Fine-tuning and Gender Fairness of BERT Models

Stanford CS224N Default Project

Kevin Rizk
Department of Mathematics
Stanford University
krizk@stanford.edu

Abstract

In recent years, fine-tuning pre-trained models has become the preferred method for training models for downstream tasks. This paper investigates the implementation of a pre-trained BERT model (minBERT) and explores how various techniques, including regularization, parameter updates, and pre-training strategies, can enhance model performance across three distinct downstream tasks: sentiment analysis, paraphrase detection, and Semantic Textual Similarity. Specifically, we examine the impact of the Bergman proximal point optimization method, data manipulations, and cosine similarity on BERT's performance in these tasks. Additionally, we analyze gender bias within our model and explore approaches to mitigate it, considering its effects on model performance and the balance between fairness and performance.

1 Key Information to include

- TA mentor: Soumya Chatterjee

2 Introduction

Transformers Vaswani et al. (2017) marked a significant milestone in the history of Natural Language Processing (NLP), catalyzing a paradigm shift and popularizing the use of pre-trained models fine-tuned for downstream tasks. BERT Devlin et al. (2018), a prominent example of a transformer model, achieved state-of-the-art results across various downstream applications.

As AI adoption surged in societal contexts, addressing ethical considerations within NLP gained importance. Mitigating human biases encoded in NLP models emerged as a critical research area Sun et al. (2019).

In this paper, we present our methodology for enhancing BERT's performance across sentiment classification, paraphrase detection, and semantic textual similarity tasks. Our approach leverages multitask learning, incorporating techniques such as cosine similarity for semantic detection, strategic adjustments of dataset sizes for efficient training across datasets of varying scales, and SMART regularization to mitigate overfitting. Our aim is to develop an efficient model capable of harnessing the full potential of pre-trained BERT.

Moreover, we investigate the fairness of our model by analyzing gender biases while proposing mitigation strategies that preserve robust performance. In particular, we will use a datasets consisting of pair of sentences which differ by a men or woman characteristic.

Through empirical evaluation, we demonstrate the efficacy of our approach and offer insights into both its strengths and limitations.

3 Related Work

Fine-tuning BERT for various methods has been a pivotal focus in recent years, garnering substantial research attention aimed at enhancing the model's efficacy. This approach allows for the adaptation of pre-trained models to specific downstream tasks, leveraging the rich knowledge captured during pre-training.

However, the transition from a pre-trained model to task-specific applications poses challenges, particularly in scenarios where the amount of available training data is limited compared to the scale of the pre-training corpus. Aggressive updates during fine-tuning can often result in overfitting, thereby hindering the model's ability to generalize to unseen data.

To address this challenge, we implemented Jiang et al. Jiang et al. (2019) novel learning framework designed to generalize well and exhibit robustness. Their approach emphasizes the importance of maintaining a smooth learned function and employing smaller, incremental step sizes during optimization using Bregman proximal point optimization. By mitigating the risk of overfitting, their method aims to enhance the model's performance across various downstream tasks.

In the domain of sentence similarity measurement, various approaches utilizing BERT have been explored. For instance, one common strategy involves concatenating two input sentences before feeding them through the model and employing a linear classifier to predict their similarity. Additionally, we were inspired by methods used by Reimers and Gurevych Reimers and Gurevych (2019). They introduced a method using BERT to learn new embeddings for each word and then try to compare the similarity of the two embeddings by using cosine smilarity for example. Their approach offers an alternative perspective on leveraging BERT's capabilities for semantic tasks.

Addressing issues of fairness and bias in NLP models has become increasingly important in recent years. Qian et al. Qian et al. (2022) proposed a novel approach to mitigating biases by augmenting training data with perturbations designed to address societal characteristics such as gender and race. Their work demonstrates the feasibility of reducing biases in NLP models without compromising performance. While our research does not directly replicate their methodology, we draw inspiration from their approach and incorporate elements of their dataset to develop our own strategies for mitigating biases in our model.

By building upon the insights and methodologies of these previous studies, we aimed to contribute to the efforts in fine-tuning pre-trained models and addressing challenges related to bias mitigation and fairness in NLP applications.

4 Approaches

Before introducing the different approaches used for training, We will introduce the baseline architecture used for comparison.

4.1 Baseline Methods

For our baseline approach, we will utilize the "[CLS]" embedding provided by BERT and employ three different heads and loss functions tailored to each of our tasks.

For sentiment analysis (SST), we will employ a linear classifier along with cross-entropy loss. In paraphrase detection (PAR) and Semantic Textual Similarity (STS), we will concatenate the two input sentences before obtaining their combined embedding from BERT. Subsequently, we will utilize a linear layer to produce logits, employing mean squared loss for STS and binary cross-entropy loss for PAR.

Additionally, we will utilize the AdamW optimizer for training, leveraging its effectiveness in optimizing large-scale neural networks.

4.2 Different approaches for downstream tasks

Apart from the Baseline architecture, we tried to identify problems and possible improvements and see which type of improvements can be made.

SMART Regularisation: As we train further, we notice overfitting effects, especially with the sentiment analysis task.Jiang et al. (2019) proposed a method which smooth the learned function and make smaller steps at each iteration. Using their code provided in the paper, where modified it a little bit by using different norms and using Bregman proximal point optimization as specified in the paper. Mathematically, if we have some model $f(\cdot;\theta)$ and n data points (x_i, y_i), where x_i is the embedding layer of the pre-trained language model, we would like to optimize:

$$\min_{\theta} \mathcal{F}(\theta) = \mathcal{L}(\theta) + \lambda_s R_s(\theta) \tag{1}$$

where $\mathcal{L}(\theta)$ is the loss function which depends on the target task, λ_s is a tuning parameter, and where

$$R_s(\theta) = \frac{1}{n} \sum_{i=1}^{n} \max_{||\tilde{x_i} - x_i||_s \leq \epsilon} l_s(f(\tilde{x_i}, \theta), f(x_i, \theta)).$$

Here, l_s is the symmetrized KL-divergence. The $\max$ ensures that in a neighborhood of x_i, we do not have a big change in $f(x_i, \theta)$, and this will ensure that our learned function will have a sort of smoothness.

2. To solve Equation 1, we used the Bregman proximal point optimization method, where we start with θ_0 as the pre-trained parameter, and update the parameter with:

$$\theta_{t+1} = \text{argmin}_{\theta} \left(\mathcal{F}(\theta) + \mu \mathcal{D}_{\text{Breg}}(\theta, \theta_t) \right), \tag{2}$$

where μ is a tuning parameter, and

$$\mathcal{D}_{\text{Breg}}(\theta, \theta_t) = \frac{1}{n} \sum_{i=1}^{n} l_s(f(x_i, \theta), f(x_i, \theta_t)),$$

again with l_s representing the symmetrized KL-divergence. The idea of this type of update is to force each θ_{t+1} to be close to θ_t, which will lower the chance of overfitting.

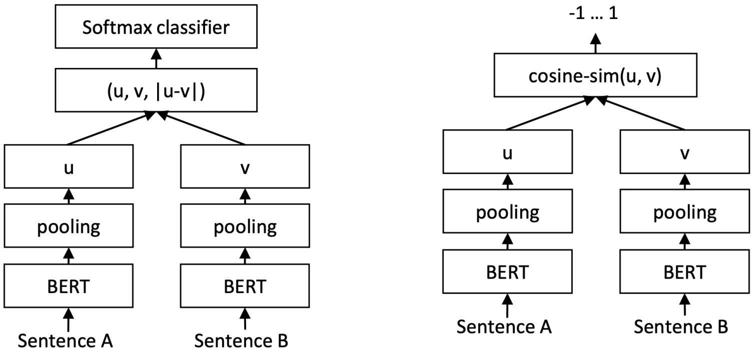
Cosine similarity for STS: Following ideas of Reimers and Gurevych (2019), we tried a new approach for STS where we took ideas from the paper and tested the two version as shown in figure 1. For the cosine similarity we projected the results into $[0, 5]$ (using different ways, squaring, linear transformation, taking 5*RELU) but finally we found that RELU gave the best result, as we considered a cosine similarity of -1 should correspond to two opposite sentence and so have a similarity score of 0.

Fine tuning Last layers vs full model: While training a pre-trained a pre-trained model, we have two options with their incovience and advantages:

- **Last Layer tuning**: Freezing the pre-trained model parameters and only updating the parameters of the added classifier layers.
- **Full model tuning**: Letting all parameters be updated, including the BERT pre-trained models.

While Last Layer tuning is way faster, it usually yields worse results. However, one significant advantage is that each head for each of the tasks will be independent of the others. Therefore, while training all together, we can save each head's parameters when it performs the best on its tasks. This would be impossible for full model tuning, as the three tasks use and change the BERT parameters. In fact, we tried investigated what happens when we use both methods: Full model tuning for some epochs and then last layer tuning in later epochs.

Data Handling: Our paragraph dataset is significantly larger than the other datasets, which could lead to problems during training. For instance, the paragraph task might overpower the other tasks.



(a) version 1

(b) version 2

Figure 1: Figure taken from Reimers and Gurevych (2019) corresponding to two different methods for sentence similarity.

To address this issue, techniques like annealed sampling Stickland and Murray (2019) have been proposed. Annealed sampling adjusts the sampling probabilities of tasks based on their dataset size, promoting a more balanced training process.

In our case, we attempted a simpler approach. We initially trained with the full paragraph dataset for a small number of epochs before reducing the size of the paragraph dataset for the remaining epochs. This approach aimed to ensure that our model considers all tasks while full utlizing all the dataset.

4.2.1 Fairness

Normally, sentiment analysis should be independent of gender, sexual orientation, or race, but bias can be incorporated into our NLP model. Here, we will discuss the approaches we came up with to mitigate biases. In fact, our goal is for two sentences with only differences in gender characteristics to have the same sentiment class. We developed different methods to train our model to reduce biases based on gender. We will outline our various approaches below.

Fairness Training: Note that in our case, for sentiment analysis, each sentence will have an associated probability P, where $P(i)$ corresponds to the probability that the sentence has sentiment class i (we will have 5 different sentiment classes). Let P_1 and P_2 represent the probabilities of two sentences, which ideally should be the same. A popular way to measure their distance is the Kullback-Leibler divergence, defined as:

$$D_{KL}(P_1||P_2) = \sum_i P_1(i) \log \left(\frac{P_1(i)}{P_2(i)} \right)$$

.

Now note that $D_{KL}(P_1||P_2) \neq D_{KL}(P_1||P_2)$, this is why we considered the symmetric Kullback–Leibler divergence defined as

$$symD_{KL}(P_1||P_2) = D_{KL}(P_1||P_2) + D_{KL}(P_2||P_1). \tag{3}$$

Here are several approaches we can use to train our model to be more robust to men versus woman biases:

- **Fair Tuning Vanilla:** We fine-tune our model (while fixing everything except the sentiment analysis head) using the symmetric Kullback-Leibler divergence as a loss function.
- **Fair Tuning Regularized:** The vanilla version may have some problems, as it could converge to $P_1 = P_2$, where $P_1(i) = 0.2$, which is not ideal. To prevent this, we add a form of regularization. Let P_1^0 and P_2^0 be the initial probabilities of P_1 and P_2 before training. We

aim to ensure that P_1 stays relatively close to P_1^0, and the same for P_2 and P_2^0. Therefore, we consider the following loss:

$$symDKL_{reg}(P1||P_2) = symD_{KL}(P_1||P_2) + \alpha D(P_1||P_1^0) + \beta D(P_2||P_2^0). \quad (4)$$

Here, α and β are tuning parameters.

- **Fair Tuning + Sentiment Tuning:** Another approach to address the problem with Fair Tuning Vanilla is to retrain both sentiment analysis and the sentence pairs using the symmetric Kullback-Leibler divergence. We then use the model that maximizes $\mu *$ Sentiment score $+ (1 - \mu) *$ fairness score on the dev set, where μ is a tuning parameter.
- **Full Training with Fair:** The last method is to train everything together and select a model that performs well based on a combination of downstream score and fairness score on the dev set, where μ is a tuning parameter. This approach is advantageous because the embedding of BERT would be fairer, potentially reducing bias in Paraphrase Detection and Semantic Textual Similarity tasks.

5 Experiments

We will present the different experiments we have done this section.

5.1 Data

We will train on three different datasets corresponding to the different downstream tasks and we will use another dataset for the

- **Sentiment analysis**: We will use Stanford Sentiment Treebank data set which consists of sentences from movie reviews labeled from 0 (Negative) to 4 (Positive). Socher et al. (2013)
- **Paraphrase Detection**: we will us the Quora Dataset for paragraph detection, where two sentences will have as label 1 if they are considered a paraphrase and 0 if not.
- **Semantic Textual Similarity** We will use SemEval STS Benchmark Dataset which consists of a pair of sentences labeled from 0 (Unrelated) to 5 (Related). Agirre et al. (2013)
- **Fairness dataset** Furthermore, for the fairness Data, we used part of the dataset provided in Smith et al. (2022). We took the part of the consisted of pair of sentences where the two sentence are almost similar except for a gender specific part which is swapped(In our case, it is woman or men) We also only considered sentences in their datasets which had a limited number of characters and not very long pair of sentences. We had 8400 train sample, 1844 dev sample, 3721 test sample.

5.2 Evaluation methods

To evaluate our model, we will use three different metrics corresponding to the different tasks. Sentiment analysis (SST accuracy) and Paraphrase Detection (PAR accuracy) will be evaluated by accuracy. For Semantic Textual Similarity (STS correlation), we will use Pearson correlation as described in the paper Agirre et al. (2013), which gives us a number between -1 and 1. The overall score will be equal to the average between PAR accuracy, SST accuracy, and $\frac{1}{2}$ (STS correlation + 1).

As for the fairness score, we calculate it by the accuracy also. In more details, it correspond to the number

5.3 Experimental details:

We employed the AdamW optimizer with a batch size of 8 for all models, allowing pre-trained BERT parameters to update by default, along with a dropout rate of 0.3.

First, we conducted an experiment where we fixed the pretrained model and only updated the final layer parameters for 10 epochs, with a learning rate of 1e-3, denoted as **Baseline (FL)**. The **Baseline** model was trained for 10 epochs, letting all pre-trained parameters update, with a learning rate of 1e-5.

For the **Cosine-method**, we tested various hypotheses. The results presented are based on two versions with architectures described in Figure 1, trained for 10 epochs with a learning rate of 1e-5.

Given the large size of the Quora dataset, we trained a baseline version (**Baseline-Red**) for 5 epochs with a learning rate of 1e-5. We selected a subset of the Quora dataset for each epoch (1/20 of its size, different subsets each epoch). Additionally, we introduced a new method, **Baseline + Baseline-Red**, where we first trained the baseline for 2 epochs using the full Quora dataset before training for 10 epochs with 1/20 of the Quora dataset, using a learning rate of 1e-5.

Lastly, **Baseline + Baseline-Red + FL** involved 2 epochs of the full Quora dataset, followed by 3 epochs of 1/20 of the Quora dataset, and then 30 epochs of fixing the BERT parameters and fine-tuning the heads , while saving the best-performing head evaluated on their respective tasks. A learning rate of 1e-5 was used throughout.

For **SMART**, we experimented with different norms but found no significant difference, so we utilized the infinity norm. The results presented used a learning rate of 1e-5. **SMART Red** was trained using 1/20 of the Quora dataset for 10 epochs with SMART update, while **Baseline + SMART Red** involved training with the baseline for 2 epochs on the full Quora dataset before running for 10 epochs using SMART.

5.4 Results

Here is a table representing the results

Models	SST accuracy	PAR accuracy	STS correlation	Overall
Baseline	0.496	**0.903**	0.818	**0.769**
Baseline (FL)	0.366	0.649	0.155	0.598
Baseline-Red	0.512	0.818	0.846	0.752
Baseline(2) + Base-Red(10)	0.505	0.897	**0.871**	0.779
Baseline(2) + Base-Red(3) + FL(30)	**0.526**	0.900	0.863	**0.786**
Cosine-method v1	0.521	0.831	0.634	0.723
Cosine-method v2	0.512	0.832	0.364	0.6753
SMART Red	0.497	0.857	0.848	0.759
Baseline(2) + SMART Red	0.489	0.893	0.864	0.771

Table 1: Experiments results on dev sets

As expected, the full tuning outperforms freezing the BERT parameters, indicating that learning better BERT embeddings for specific tasks improves performance. Notably, by reducing the influence of the Quora Set, we observe that the **Baseline-Red** model performs significantly better on sentiment analysis and STS but shows weakness in PAR accuracy. This outcome is expected since training with fewer Quora subsets reduces the influence of PAR and enhances the influence of STS and SST on MinBert parameters. Training initially for 2 epochs using the full dataset aids in this adjustment.

As anticipated, **Baseline + Baseline-Red** maximizes the use of the entire Quora dataset before downsizing it, thereby improving upon the Baseline substantially.

Moreover, **Baseline + Baseline-Red + FL** capitalizes on the fact that freezing the BERT parameters renders the three downstream heads independent. Consequently, we can select the best-performing head for each task, leading to superior results compared to the previous three models.

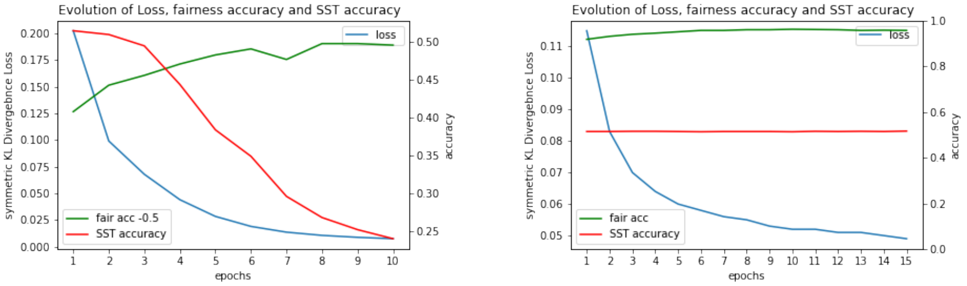
The results for SMART were unexpected; it did not mitigate overfitting. In fact, as the learning rate approached 0, the accuracy worsened from 0.512 to 0.497. Additionally, the Cosine-method yielded worse results for STS than the baseline, possibly due to the implementation of the pooling layer. Further modifications are needed to assess its impact on the score.

5.5 Test Data score

The Test Data score can be found in table 2.

Models	SST accuracy	PAR accuracy	STS correlation	Overall
Baseline(2) + Base-Red(3) + FL(30)	0.518	0.898	0.861	0.782

Table 2: Experiment result son Test set



(a) Fair Tuning Vanilla

(b) Fair Tuning + Sentiment Tuning

Figure 2: Figure taken from Reimers and Gurevych (2019) corresponding to two different methods for sentence similarity.

5.6 Fairness Experiments:

To achieve a fairer model, we conducted five experiments as described in our approach in subsection 4.2.1. For all experiments, we utilized a learning rate of $1e-5$, a batch size of 8, and AdamW optimizer and fine tuned on our BEST result model. Here are the details of the experiments:

Fair Tuning Vanilla: In this experiment, we solely fine-tuned the sentiment head using the $symDL$ loss (3) and trained for 10 epochs.

Fair Tuning Regularized: Similar to the first experiment, the only distinction here was the utilization of the $symDL_{res}$ loss (4). We conducted two rounds of training, one for 10 epochs and another for 20 epochs.

Fair Tuning Regularized with STS: For this experiment, we simultaneously trained the sentiment analysis on the STS dataset and fairness on its dataset using the $symDL$ loss. We conducted training for 15 epochs choosing $\mu = 2/3$

Full Training with Fair: For this experiment, we did a full training changing all parameters for 10 epochs, learning rate $1e-5$ and using $\mu = 9/10$.

The results are shown in table 3

5.6.1 Fairness Results and analysis:

Here is the table for our results.

Models	SST accuracy	PAR accuracy	STS correlation	Overall	Fairness
BEST	0.526	0.900	0.863	0.786	0.883
Fair Tuning Vanilla	0.252	0.900	0.863	0.6945	**0.998**
Fair Tuning Regularized (10)	0.506	0.900	0.863	0.779	0.960
Fair Tuning Regularized (20)	0.436	0.900	0.863	0.779	0.986
Fair Tuning + Sentiment Tuning (15)	0.515	0.900	0.863	0.782	0.960
Full Training with Fair	0.512	0.771	0.800	0.732	0.966

Table 3: Experiments results on dev sets

First, note how the Vanilla version substantially lowers the SST score. In fact, this phenomenon can be observed more closely in Figure 2, where the SST accuracy drops rapidly. This can be explained by the fact that the symDKL loss only ensures that the probabilities of sentences are equal without considering the initial goal. For example, having $P_1(i) = P_2(i) = 0.2$ is considered a valid solution.

This is why the Fair Tuning Regularized method performs much better. It penalizes changes where probabilities change significantly and getting away from the initial probabilites while simultantionly letting the pair of sentence converge to the same probability, resulting in better fairness results with a lower drop in performance.

Regarding Fair Tuning + Sentiment Tuning, training using only the symDKL loss and focusing solely on sentiment analysis indirectly regularizes fairness and prevent the phenoma observed in Vanilla version to happen. It forces the two tasks to find a compromise, ensuring convergence to a fair model while still performing well on sentiment analysis. In fact we see in figure 2 how the SST accuracy is more stable compared to vanilla version.

As expected, we achieved very good results with Full Training with Fair. Particularly noteworthy is the fact that in this case, bias mitigation is also incorporated into the BERT parameters, not only in the classification heads.

6 Analysis

Improving SST accuracy beyond 0.53 proved to be challenging. Upon investigating the datasets and the results, we observed that sentiments 0 and 4 appear the least, with 1092 and 1288 occurrences respectively, compared to over 1500 occurrences for the other sentiments in the training data. Consequently, sentiments 0 and 4 are underrepresented, leading the model to predict 1 and 3 more frequently (over 2000 times). Indeed, our model predicts sentiment 0 only 88 times(appears 139 times in dev set) , while predicting sentiment 1 344 times, which appears 289 times in the dev set. Similar patterns are observed for sentiments 3 and 4.Moreover, sentiment 2 represents a neutral sentiment, making it challenging to distinguish. Our model underpredicts sentiment 2 (193/229 occurrences). This underscores the difficulty in predicting extreme sentiments, as the boundary between sentiments can be subjective even for humans, presenting a challenging task for AI systems.

Additionally, we observed that the order in which we train different tasks of our model influences the final scores. Specifically, what was trained last tends to have a greater influence on the overall performance. This highlights a flaw in our model architecture. One potential mitigation strategy could involve reevaluating how we process our batches while training to mitigate this effect.

In terms of fairness, upon evaluating our best model, we conducted a direct fairness assessment and discovered a fairness metric of 0.883, indicating that 88 percent of pairs exhibited differing sentiment classes. An intriguing example is the comparison between the sentences "Bob's mood has improved" and "Bob's mood is getting better." Surprisingly, the former had a negative sentiment (0), while the latter expressed a positive sentiment (4). This unexpected result underscores the bias present in the model and the importance of mitigation. Even after fairness training, the model assigned different scores to both sentences. Despite the mitigation efforts, we still couldn't achieve agreement between the model's predictions for this pair of sentences. The fairness mitigation primarily aided pairs of sentences where the probabilities were already close, and the difference in classes was closer to 1. This could be explained by the difficulty for the model to choose the correct sentiment when there is a significant difference between the classes. In such cases, a middle ground between both sentiments may not necessarily be appropriate (as in our example, where the sentiment should be 4) and making an arbitrary choice is not easy either.

7 Conclusion

In our project, we implemented various methods to enhance BERT's performance on different downstream tasks while exploring how manipulating data could improve overall performance. Additionally, we delved into studying a specific type of fairness concerning our model and tested different approaches to mitigate bias without compromising performance. We acknowledge the potential for further investigation into addressing biases beyond gender and exploring alternative methods for bias mitigation.

8 Ethics Statement

One significant ethical challenge is the issue of bias. As we train AI models using datasets created by humans, there's a risk of incorporating societal biases related to gender, sexuality, and other factors into the model. For instance, in sentiment analysis, biases regarding what constitutes positive or negative phrases could influence the AI model's outputs and could infleunce young children, they could associate some societal category to a negative trait if they over see it through the AI.

Another concern is the potential misuse of AI models for manipulation, such as attempting to sway opinions on what is considered positive or negative by carefully selecting example to sway opinions and behaviors For example, in our case, we have seen the example "Bob's mood has improved" and "Bob's mood is getting better." Surprisingly, the former had a negative sentiment (0), while the latter expressed a positive sentiment (4) while they should in theory had the same sentiment. To mitigate these risks, we used another dataset which contained pair of senteces and we trained wehere sentiment analysis should be the same and we trained our dataset on this pair of sentences to ensure that gender biases is lowered. We can extend this techniques to include all other types of societal categories and gender. as possible.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2019. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*.

Rebecca Qian, Candace Ross, Jude Fernandes, Eric Smith, Douwe Kiela, and Adina Williams. 2022. Perturbation augmentation for fairer nlp. *arXiv preprint arXiv:2205.12586*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Eric Michael Smith, Melissa Hall, Melanie Kambadur, Eleonora Presani, and Adina Williams. 2022. " i'm sorry to hear that": Finding new biases in language models with a holistic descriptor dataset. *arXiv preprint arXiv:2205.09209*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. 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.

Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In *International Conference on Machine Learning*, pages 5986–5995. PMLR.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. *arXiv preprint arXiv:1906.08976*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

A Appendix (optional)