Expanding minBERT's Scope: Integrating SimCSE, Ensemble Learning, and PANDA

Stanford CS224N Default Project

Yasmina Abukhadra
Department of Computer Science
yasabukh@stanford.edu

Samantha Liu
Department of Computer Science
szwliu@stanford.edu

Hannah Norman
Department of Computer Science
hnorman@stanford.edu

Abstract

We introduce BERT SCOPE: minBERT with Simple COntrastive learning, Perturbation augmentation, and Ensemble learning. BERT SCOPE is designed as a multi-task model that targets sentiment analysis, paraphrase detection, and semantic textual similarity ranking. Starting with minBERT, an adaptation of the original BERT model, we enhance it using contrastive learning, ensemble learning, and the Perturbation Augmentation NLP Dataset (PANDA). Our approach includes extensive hyperparameter tuning to optimize performance. The results demonstrate strong performance across all tasks, with ensembling being particularly effective. Our best model achieves a mean performance of 78.1% on the test set, significantly surpassing baseline results.

1 Key Information

TA Mentor: Neil Nie *Sharing Project:* No *External Collaborators:* No

2 Introduction

In the field of Natural Language Processing (NLP), a significant and ongoing challenge tackles the generation of high-quality sentence embeddings for use in a variety of tasks. Sentence embeddings encode semantic information into numerical form and enable models to perform tasks such as sentiment analysis, paraphrase detection, and semantic textual similarity ranking. Our work builds upon the foundational Bidirectional Encoder Representations from Transformers (BERT) model, which revolutionized NLP with its use of bidirectional word representation [1]. However, despite BERT's groundbreaking advancements, limitations remain in its ability to generate robust sentence embeddings for optimal performance across several tasks at once.

We seek to address these challenges by leveraging and optimizing the minBERT architecture. We integrate advanced techniques such as contrastive learning with SimCSE [4], ensemble learning, and perturbation augmentation with PANDA [7]. By expanding minBERT's scope in this manner—via our novel architecture, BERT SCOPE—our approach improves BERT's performance across a number of tasks. Specifically, our results demonstrate substantial improvements across sentiment analysis, paraphrase detection, and semantic textual similarity ranking, and our model ultimately achieves a test set performance score of 78.1%. This demonstrates the effectiveness of our methods in enhancing sentence embedding generation within the BERT framework, paving the way for future research into improved model robustness for multi-task objectives.

3 Related Work

In describing our final model, BERT SCOPE, we consider foundational work in the areas of contrastive learning, ensemble learning, and bias mitigation. Specifically, we discuss the SimCSE framework [4], the QuarBERT ensemble architecture [5], and the PANDA dataset [7].

Contrastive Learning Gao et al. propose the SimCSE framework, a method which advances the use of contrastive learning for sentence embeddings in both supervised and unsupervised settings [4]. Contrastive learning aims to improve the objective:

$$l_i = -\log \frac{e^{\mathrm{sim}(h_i, h_i^+)/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(h_i, h_j^+)/\tau}}$$

for a mini-batch of N pairs, where τ is a temperature hyperparameter, sim() is the cosine similarity, and each pair h_i, h_i^+ represents a positive pair of sentence embeddings to be considered similar [4]. The authors of the SimCSE paper analyze the embeddings from the perspective of *alignment* and *uniformity* [4]. In contrastive learning terms, embeddings of positive pairs should stay close and embeddings for random instances should scatter in the vector space [4]. Gao et al. propose that contrastive learning is effective because its inherent goal is to avoid *anisotropy* (where the embeddings form a narrow cone) [4]. The model's performance in the SimCSE paper demonstrates that alleviating the anisotropy problem is crucial in making the sentence embeddings more expressive representations of the human language [4].

In terms of implementation, the SimCSE paper presents a simple yet effective framework for unsupervised learning of sentence embeddings: to use standard dropout on the same sentence twice to obtain two different embeddings that become a positive pair for contrastive learning [4]. Negative pairs, on the other hand, are selected from other sentences in the minibatch during training [4]. Gao et al. show that this method produces better word sentence embeddings than previous methods, such as predicting next sentences and discrete data augmentation (e.g., word deletion and replacement) [4]. As for supervised learning, the authors leverage the supervised natural language inference (NLI) datasets, using sentences that represent entailment as positive pairs and those that represent contradiction as negative pairs [4]. After training towards the contrastive learning objective, further improvement on the model's performance is again observed [4]. For BERT SCOPE, we implement supervised contrastive learning, using only positive pairs and not hard negatives.

Ensemble Learning Ensemble learning has been shown to improve the performance of BERT, with the caveat of higher computational demands. A combination of variations of BERT models have been used for specific tasks such as for semantic similarity matching for patent documents [8] and hate speech classification [3]. Methods of combining the predictions of the models include taking a weighted average and majority voting. For the purpose of multitask classification, however, the question remains of how we can create enough diversity among the models and combine them effectively for improvements across all tasks. Gu et al. proposed QuarBERT, which uses four BERT models with one that is trained for multitask classification and three that specialize in each of the tasks [5]. This has been shown to enhance the performance of a single multitask model; our implementation takes inspiration from this architecture.

Bias Mitigation One key area of research of NLP is that of reducing and mitigating bias. Qian et al. created an NLP dataset designed to help address the issues of fairness and bias: Perturbation Augmentation NLP Dataset (PANDA) [7]. The dataset consists of example pairs containing an original example and a perturbed example, where a demographic word is changed to reflect a different demographic group [7]. Qian et al. found that fine-tuning models on perturbed data (or “fairtuning") caused them to perform better on metrics of fairness [7]. One technique to mitigate bias is to use conterfactually augmented data such as this to perform contrastive learning; the goal of this technique is to promote the closeness of the sentence embeddings despite the demographic word differences [2]. Therefore, in order to help mitigate potential bias, we implement an extension to fine-tune the model on the PANDA dataset using contrastive learning, considering the perturbed pairs to be positive pairs.

4 Approach

4.1 minBERT Architecture

We begin by implementing minBERT, a simplified version of the original BERT model presented by Devlin et al. [1]. This involves implementing parts of BERT multi-head self attention, the BERT transformer layer, and the Adam Optimizer (used for stochastic optimization). Further, we implement multi-task classification, which includes tasks of sentiment classification, paraphrase identification, and semantic textual similarity ranking. In order to fine-tune the minBERT model for multi-task classification, we sum together the calculated losses for each downstream task, described previously.

In our first iteration of minBERT multi-task classification, we applied forward propagation and dropout on each set of embeddings, before generating logits by feeding these embeddings into a linear layer. For paraphrase prediction and similarity analysis, we utilized absolute difference and element-wise multiplication between the two embedding sets to generate input for the linear layer. Also, we fed the similarity analysis logits into a non-linear ReLU layer. Finally, during training, we calculated cross-entropy loss with the sentiment logits, binary cross-entropy loss with the paraphrase logits, and MSE loss with the similarity embeddings, before back-propagation. However, due to suboptimal performance (see Table 1), we recognized a need to reorganize our model architecture.

In the current version of minBERT multi-task classification, we apply forward propagation and dropout on each set of embeddings, feed them into a linear hidden layer and apply dropout once again, before feeding this iteration of embeddings into a final linear predictor layer to generate logits. Since the paraphrase identification and semantic textual similarity ranking tasks receive sentence pairs, these inputs are immediately concatenated before being fed into forward propagation. Finally, during training, the generated logits are used for cross-entropy loss, binary cross-entropy loss, and MSE loss respectively for the tasks of sentiment classification, paraphrase identification, and semantic textual similarity ranking. With this relatively simple design, our model performance improves significantly, and so we build the rest of our framework atop it.

4.2 Contrastive Learning

In order to perform supervised contrastive learning, we need positive pairs of examples—i.e., those that should be considered the same or similar. In order to extract these positive pairs from our existing datasets, we iterate through the data from the Quora dataset for paraphrase detection and the SemEval dataset for semantic textual analysis. Positive pairs for the Quora dataset are those labeled as paraphrases (with a label of 1), and positive pairs for the SemEval dataset are pairs with similarity scores higher than 2 (on a scale of 5). The positive pair examples from each dataset are then used to create two new SentencePairDatasets, and these are loaded in the same manner as the other datasets.

For each dataset, we then calculate the cross-entropy loss on the cosine similarities of the sentence embeddings for each batch, as in the SimCSE paper [4]. We base our implementation of the CL loss calculation on that of Gao et al., making some adjustments of the structure of the loss calculation code to fit within our framework [4]. To compute the overall loss, the CL losses are multiplied by a weight value and then added to the sum of the loss values computed for each original dataset.

4.3 Hyperparameter Tuning

As we iterate through our model, we perform extensive hyperparameter tuning. For the baseline multi-task architecture, we focus on optimizing the number of hidden layers, testing configurations with one and two layers, while prioritizing time and space efficiency. For contrastive learning (CL), we tune the temperature hyperparameter using the same test values from the original SimCSE paper [4]: 0.001, 0.01, 0.05, 0.1, and 1.0. Additionally, we adjust the CL weight across values of 0.1, 0.5, and 1.0 to find the most effective balance. The learning rate is fine-tuned, starting from the default 1e-5 and incrementing up to 5e-5 in steps of 1e-5, to ensure optimal training conditions. With the ideal hyperparameters determined, we then proceed to integrate ensemble learning and the PANDA dataset, enhancing the model's robustness and performance across all tasks.

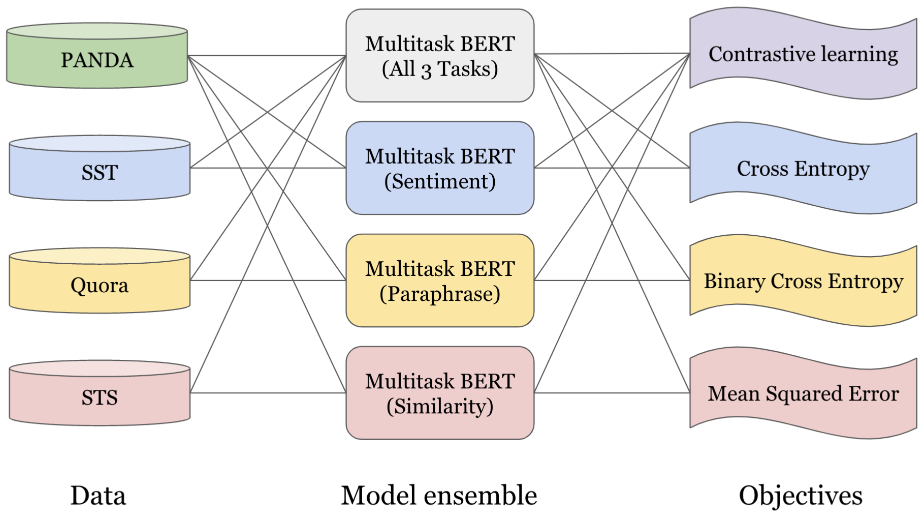

Figure 1: **BERT SCOPE model architecture.** Our architecture combines contrastive learning, ensembling, and perturbation augmentation so as to optimize multi-task performance.

4.4 Ensemble Learning

Another extension we implement is ensemble learning, which combines several models in order to generate more robust predictions. To do this, we follow the approach of Gu et al. [5], using four instances of the base minBERT models. Model 0 is trained on the losses from all three tasks while models 1, 2, and 3 are trained on only the loss of their respective tasks—sentiment, paraphrase, or semantic similarity classification. For every task, we then take the average of the predictions of the two models, with the goal of achieving more balanced predictions in terms of multitasking and specialization. Furthermore, we expand on this approach by considering the combination of contrastive learning and ensemble methods and weighing the CL objective individually for each of the models. Every model was trained on some linear combination of the CL loss and the multitask losses. See Figure 1 for a visualization of our architecture.

4.5 PANDA

Finally, we explore contrastive learning (CL) in the context of bias mitigation, using the perturbation pairs from the PANDA dataset as positive pairs for CL [7]. In order to easily download and use the PANDA dataset, we use the Hugging Face Hub and associated Datasets library [6]. We build off of the existing SentencePairDataset in the default project's `datasets.py` file in order to create a class for PANDA pair data. The original and perturbed examples are taken from the PANDA dataset and stored as positive pairs. Then, the loss as calculated by the objective for contrastive learning is added to Model 0, weighted by the same CL weight objective.

5 Experiments

5.1 Data.

To perform sentiment analysis with the BERT SCOPE model, we use the SST and CFIMDB datasets. For the paraphrase detection task, we use the Quora dataset. And for semantic textual analysis, we use the SemEval dataset.

We additionally perform contrastive learning on the Perturbation Augmentation NLP Dataset (PANDA), a dataset designed to help reduce bias in NLP models by providing equivalent example pairs with perturbations of demographic terms [7].

5.2 Evaluation method.

We use `evaluation.model_eval_multitask()` in `evaluation.py` to evaluate our model on each sentence-level task, both with and without the extensions. The datasets described above (SST, Quora, and SemEval) are utilized, which in turn use standard accuracy and the Pearson correlation of true versus predicted similarity values as evaluation metrics.

5.3 Experimental details.

We develop our model locally, and run it on a L4 GPU on Google Colab. We run both `classifier.py` and `multitask_classifier.py` with batch size 64, though we use occasionally resort to batch sizes of 32 or 8 for multi-task classification of the full model due to CUDA memory constraints. For fine-tuning the last linear layer, we use a learning rate of `1e-3`, and for fine-tuning the full model, we use a learning rate of `1e-5` and later `3e-5`. Each initial configuration runs between 15 and 25 minutes, though after adding ensemble learning, we observe a runtime of around 90 minutes, which reflects the fact that we are training four models at the same time.

Note that during development, we test and benchmark our model against the DEV leaderboard. All results that follow, unless explicitly stated, represent numbers achieved on the DEV set.

5.4 Results.

First, we provide results for our initial and reworked minBERT model architectures on the three tasks of sentiment analysis, paraphrase detection, and semantic textual similarity ranking.

	Sentiment	Paraphrase	Similarity	Overall
Initial model	0.521	0.629	0.213	0.586
Reworked	**0.511**	**0.806**	**0.857**	**0.749**

Table 1: **DEV results.** On minBERT architectures.

The switch in model architecture design immediately offers a significant boost in performance. For further enhancement, we consider adding a second hidden layer to each task's predictive method.

	Sentiment	Paraphrase	Similarity	Overall
One Hidden Layer	**0.511**	**0.806**	**0.857**	**0.749**
Two Hidden Layers	0.500	0.824	0.853	0.750

Table 2: **DEV results.** On minBERT hidden layer variations.

With an additional hidden layer, the results demonstrate a significant increase in paraphrase task performance matched by moderate decreases in sentiment and similarity task performances. As a consequence, the overall performance rating increases by 0.01, but we decide to remain with the single hidden layer architecture due to an interest in space and time efficiency.

At this point, we incorporate the first extension: contrastive learning (CL). We fine-tune the temperature hyperparameter while doing so and include the results in Table 3.

Temperature	Sentiment	Paraphrase	Similarity	Overall
0.001	0.474	0.781	0.853	0.727
0.01	**0.515**	**0.816**	**0.849**	**0.752**
0.05	0.515	0.816	0.849	0.752
0.1	0.512	0.817	0.846	0.751
1.0	0.508	0.819	0.849	0.751

Table 3: **DEV results.** On contrastive learning temperature fine-tuning.

Although a temperature of 0.01 and 0.05 achieve identical results, we choose to maintain a temperature value of 0.01 in line with the authors of the original SimCSE paper [4].

With temperature chosen, we investigate tuning the CL weight parameter. For extraneous reasons, we conduct this stage of hyperparameter tuning on the earlier, initial version of the minBERT architecture, with the CL extension implemented in its same form. We hypothesize that any demonstrated trends in the results would extend to our current minBERT architecture.

Weight	Sentiment	Paraphrase	Similarity	Overall
0.1	**0.500**	**0.637**	**0.589**	**0.644**
0.5	0.448	0.635	0.609	0.629
1.0	0.411	0.635	0.656	0.625

Table 4: **DEV results.** On CL weight fine-tuning, tested with the initial architecture.

Now, we proceed to the final round of hyperparameter tuning: learning rate (LR). We return to fine-tuning on the current minBERT model architecture from this point forward.

LR	Sentiment	Paraphrase	Similarity	Overall
1e-5	0.515	0.816	0.849	0.752
2e-5	0.516	0.805	0.858	0.750
3e-5	**0.523**	**0.810**	**0.863**	**0.755**
4e-5	0.523	0.807	0.863	0.754
5e-5	0.515	0.811	0.865	0.753

Table 5: **DEV results.** On learning rate fine-tuning.

Up to this point, we have determined that the ideal version of the current architecture with contrastive learning (CL) implemented utilizes one hidden layer, a CL temperature of 0.01, a CL weight of 0.1, and an overall learning rate of 3e-5. In seeking a final performance boost, we implement two more extensions: ensemble learning and PANDA. Results are provided below.

Extension	Sentiment	Paraphrase	Similarity	Overall
Ensembling	0.531	0.850	0.889	0.775
PANDA	0.523	0.850	0.885	0.772

Table 6: **DEV results.** On ensembling and PANDA extensions.

Despite the model ensembling extension on its own achieving a higher overall accuracy, we submit the version of our BERT SCOPE model with the PANDA extension implemented to the TEST leaderboard. The results surpass expectations.

Sentiment	Paraphrase	Similarity	Overall
0.548	0.852	0.887	0.781

Table 7: **TEST results.** On final BERT SCOPE model.

Shown above, the overall accuracy achieved by our BERT SCOPE model on the TEST set is 78.1%.

6 Analysis

minBERT Architecture We designed our minBERT model in two iterations, ultimately finding a relatively simple architecture to be the most successful for our multi-task BERT models targeting sentiment analysis, paraphrase detection, and semantic textual similarity ranking. This simplified design utilizes one hidden layer and two iterations of dropout, with an initial forward propagation call and final linear predictive layer to generate logits. This straightforward architecture led to significant performance improvements, forming the foundation of our framework, as illustrated in Table 1.

The performance uptick observed can be attributed to several factors. Firstly, using dropout at multiple points likely aided in model regularization, reducing overfitting, and improving generalization to unseen data. Moreover, utilizing only a single hidden layer prioritizes computational efficiency while avoiding potential pitfalls of overly complex architectures, such as vanishing gradients. The minimal performance difference observed when testing two hidden layers versus one reinforces that added complexity did not yield significant benefits, validating the efficacy of our simpler architecture. Additionally, concatenating sentence pairs for tasks like paraphrase identification and semantic textual similarity ranking ensures that the model captures the relational context between sentences, enhancing its ability to discern subtle nuances in meaning. This design choice likely contributed to better performance in these tasks.

When it comes to the tuning of learning rates, 3e-5 performs the best. However, it is worth noting that the other learning rates still offered competitive performances, indicating that our model is robust across different hyperparameter settings. This is demonstrated in Figure 2.

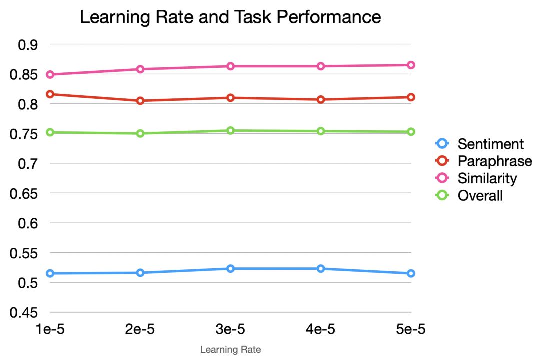

Figure 2: **LR on multi-task.** Effect of learning rate on downstream task performance.

Contrastive Learning We choose sentence pairs that are paraphrases or semantically similar to construct positive pairs for contrastive learning (CL), in hopes that this aligns the semantically-related positive pairs more closely and makes the whole representation space more uniform, as described by Gao et al. [4]. Furthermore, while tuning the weight of the contrastive learning loss with respect to the losses for the prediction tasks, we discovered the a higher CL weight improves paraphrase and semantic similarity classifications, but is not beneficial for sentiment classification. See Figure 3.

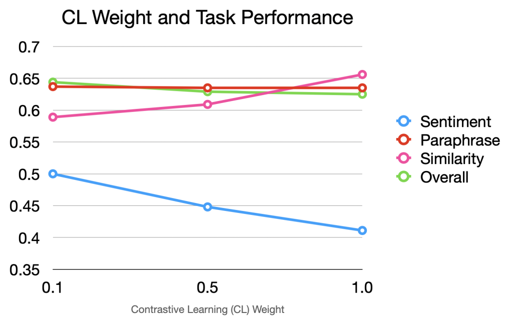

Figure 3: **CL weight on multi-task.** Effect of CL weight on downstream task performance.

We suspect this is because the data for the paraphrase and similarity tasks align more closely with our CL framework's objective, particularly as we use the Quora and SemEval datasets to generate positive and negative pairs. Positive pairs are derived from sentences that are paraphrases or semantically similar, so when their representations are made close through CL, the predictor more accurately classifies them as paraphrase/similar. In contrast, sentiment prediction lacks this correlation between task data and CL data, leading to decreased performance.

Model Ensembling Adding ensemble learning as an extension allows us to set individualized weights for the CL objective for each model in the ensemble. We explore various combinations of the weights and find that there is no significant effect in performance (see Table 8); in fact, adding the CL loss to models 1-3 hurts the overall performance. This surprising result suggests that the CL objective may introduce noise into the losses of specialized models. For example, even if its predictions are accurate, the paraphrase predictor would be penalized for not pushing closer sentence pairs that are semantically similar, which constitutes some of our positive pairs for CL. This shows that finding datasets that align well with the tasks to be done by the model is crucial for incorporating CL.

w1	w2	w3	w4	Sentiment	Paraphrase	Similarity	Overall
0.1	**0**	**0**	**0**	**0.531**	**0.85**	**0.889**	**0.775**
0.1	0	2	2	0.531	0.85	0.882	0.774
0.1	0.1	0.1	0.1	0.531	0.85	0.881	0.774
0.1	0.05	0.5	0.5	0.531	0.85	0.878	0.773
0.1	0.5	5	5	0.531	0.85	0.877	0.773

Table 8: **Ensemble fine-tuning.** Tuning the CL objective weights for each model in the ensemble.

PANDA When we fine-tune the model using contrastive learning on the PANDA dataset, we do not observe significant negative impacts on performance. In fact, with a 1e-5 learning rate, the PANDA fine-tuned model slightly outperforms the EM model, although this does not hold for the 3e-5 learning rate (see Table 9). Similar to Qian et al.'s successful use of PANDA for pre-training and fine-tuning, we find that the fairness dataset can be used for fine-tuning with negligible negative impact on task performance [7]. Perturbation-augmented datasets may even enhance model performance by introducing more variation in the training data, leading to better generalization to unseen examples.

Extension	Sentiment	Paraphrase	Similarity	Overall
Ensembling	0.521	0.845	0.868	0.767
PANDA	**0.527**	**0.844**	**0.869**	**0.769**

Table 9: **DEV results.** On ensembling and PANDA extensions with 1e-5 LR.

7 Conclusion

Our study underscores the effectiveness of extending minBERT for specialized tasks through the integration of contrastive learning, ensemble learning, and the PANDA dataset. We observe significant performance enhancements across sentiment analysis, paraphrase detection, and semantic textual similarity tasks. We find ensembling to be the most pivotal strategy for improving model accuracy, with our approach demonstrating notable improvements in both overall and task-specific performance. Our work lays a solid foundation for future advancements in leveraging minBERT and advanced techniques to tackle diverse and complex NLP challenges with improved accuracy and fairness.

Future work includes experimenting with unsupervised contrastive learning (CL) for sentiment classification. Due to the lack of effective positive pair generation from the SST dataset, CL was not performed with examples from this dataset. We can also generate negative pairs for CL to enhance the objective of distinguishing dissimilar pairs. Additionally, generating perturbation-augmented versions of the original datasets could help mitigate bias and result in a fairer model.

8 Ethics Statement

The ethical and social implications of this project primarily center around bias in NLP. Social biases in NLP can be grouped into two categories: “representational harms," which are those that reinforce negative social representations or attitudes, and“allocational harms," which are those that cause unfair treatment of groups [2]. Let us explore examples of these ethical challenges and societal risks within the context of our movie review datasets targeting sentiment analysis tasks with BERT.

One “representational harm” would be that social biases in the training data lead to negative sentiment being associated with certain identities more often. For example, movies with strong female leads might be reviewed more negatively, causing a model to classify similar movies more negatively in testing data. An “allocational harm” would occur if the biased model were used to score or rank movies on a website, leading to movies with unfairly low scores receiving less viewership. Continuing the previous example, movies with female main characters might receive lower ratings and consequently be watched by fewer people.

There are various approaches to mitigating bias at different stages of the processing/training cycle in NLP [2], but we will focus on pre-processing and in-training approaches. One interesting technique, relevant to our focus on contrastive learning, is counterfactual data augmentation. This method replaces certain identifying words, such as changing male pronouns to female pronouns, creating equivalent sentences for training [2]. In contrastive learning, this technique is compelling because it allows for the creation of positive pairs differing only in specific identifying characteristics. The model learns to prioritize the similarity of these pairs, indicating that changing these characteristics does not alter the underlying meaning, thereby reducing bias [2]. To explore this, we used the Perturbation Augmentation NLP Dataset to fine-tune our model and mitigate bias [7]. In the future, it might be more effective to apply counterfactual data augmentation to the original datasets to create fairer training data for semantic classification, paraphrase, and similarity tasks.

Another intriguing approach that involves in-training mitigation is Entropy Based Attention Regularization, which aims to maximize the entropy of the attention weights by adding a regularization term to the loss function [2]. This is hypothesized to help mitigate bias since it will help prevent strong associations with a few stereotypical words in a sentence and emphasize a more holistic interpretation of the sentence [2]. Taken together, we recognize the diverse ethical and social implications of our model, and though we did explore implementing one strategy for bias mitigation, there are many further extensions which could aid in producing a fairer model.

9 Contributions

Work on this project was shared equally across team members. Yasmina took the lead with the Adam Optimizer implementation, contrastive learning (CL) dataset generation, CL implementation, and PANDA extension. Samantha took the lead with initial minBERT architecture design, minBERT testing, CL fine-tuning, and implementing the ensemble learning extension. Hannah took the lead with the minBERT multi-headed self-attention and transformer layer implementations, the reworked minBERT architecture design, hyperparameter tuning, and final report revisions. All three team members supported each other in their respective tasks and wrote the project proposal, milestone, and final report jointly. We feel the work was shared fairly.

10 Acknowledgements

We wish to thank Neil Nie, our CS 224N mentor, for his feedback and support during this project.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[2] Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. Bias and fairness in large language models: A survey. *arXiv preprint arXiv:2309.00770*, 2023.

[3] Amrita Ganguly, Al Nahian Bin Emran, Sadiya Sayara Chowdhury Puspo, Md Nishat Raihan, Dhiman Goswami, and Marcos Zampieri. Masonperplexity at multimodal hate speech event detection 2024: Hate speech and target detection using transformer ensembles, 2024.

[4] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

[5] Siyi Gu, Zixin Li, and Ericka Liu. Quarbert: Optimizing bert with multitask learning and quartet ensemble, 2024.

[6] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[7] Rebecca Qian, Candace Ross, Jude Fernandes, Eric Smith, Douwe Kiela, and Adina Williams. Perturbation augmentation for fairer nlp, 2022.

[8] Liqiang Yu, Bo Liu, Qunwei Lin, Xinyu Zhao, and Chang Che. Semantic similarity matching for patent documents using ensemble bert-related model and novel text processing method, 2024.