Debiasing Large Language Models Using Attention Alteration

Priyal Belgamwar
psb872
priyal.belgamwar@utexas.edu

Abstract

The issue of bias in natural language processing (NLP) models results in unequal benefits for individuals in positions of privilege. Biases in the data used for training NLP models propagate into their predictions which hinders the socio-economic and cultural advancement of society. While previous methods for mitigating bias in NLP models have focused on manipulating word embeddings and data the proposed approach involves modifying the attention mechanism of pre-trained transformer-based language models. Specifically, the attention mechanism is modified to give equal weights to all demographic groups in the input data. The proposed method aims to reduce harmful associations by calibrating the attention of the context on identity terms which is applied to text encoders as a general representation layer. The method is different from previous approaches and aims to maintain the semantic understanding of the original encoder while reducing biases. The proposed approach utilizes attention and modifies it to reduce bias in NLP models.

1 Introduction

As society progresses technologically, the benefits of these developments are not reaped equally – those in positions of privilege tend to enjoy the most benefits. This can be attributed to machine learning models that are becoming increasingly integrated into my daily lives. Since these models rely highly on data, biases inherently present in the data propagate into their predictions. This issue is especially prominent in NLP-based models, where real-world data is essential in order to make meaningful predictions. This, in turn, hinders the socio-economic and cultural advancement of society. NLP models have been observed to show biases in various social constructs such as gender, religion, race etc. Hence it is important to take measures toward conscious mitigation of these biases and increase fairness in models while striving to maintain considerable accuracy. While significant progress has been made toward the mitigation of bias in NLP models, most methods focus on the manipulation of word embeddings and data. Pre-trained transformer-based language models, which have a wide range of applications in modern NLP tasks, have not been modified internally to remove bias in the operation of the layers of these models. My approach involves diving deeper into the attention mechanism of transformers, specifically that of pre-trained text encoders. Inspired by the paper [6], my approach involves modifying the attention mechanism to give equal weights to all demographic groups in the input data. This helps to reduce biases that might be present in the data, ensuring that the model does not disproportionately focus on certain groups. I also strive to ensure that the semantic information of the original encoder is not affected by the alterations made to the attention mechanism. This can be done by distilling the original attentions into the model I wish to debias from an unaltered model [8]. I propose to conduct attention alterations only on certain values of number of heads and layers. These values can be determined based on the bias calculated. The magnitude of alteration on attention layers will depend on the calculated bias and a predetermined bias threshold. This will help in reducing the number of operations and keeping the semantic understanding of the original encoder intact to a higher degree.

2 Related Work

There is a lot of work out there that deals with quantifying bias. For example, [3] and [13] compared the cosine similarity of group and attribute representations discovering unequal similarities between groups. This is representation based and involves learning vector relationships between inputs. Another approach is likelihood-based in which they analyze how frequently text encoders favor stereotypes over anti-stereotypes. Preferences are de-

fined in terms of higher likelihoods produced by language models using embeddings of the text encoders under study (Kurita et al., 2019 [9]; Nadeem et al., 2020 [14]; Nangia et al., 2020 [15]; Gaci et al., 2022b [7]). I also have an Inference-based approach that uses text encoders in downstream NLP tasks such as natural language inference [4], sentiment analysis [5], or language generation ([17]; [12]). Bias in such settings is identified as the difference in outcome when models are tested with the same input sentence varying only in social groups. Debiasing Pretrained Text Encoders by Paying Attention to Paying Attention proposed metric specifically encodes bias in the attention layer which is what I use and extend in my models.

The field of Natural Language Processing (NLP) has made significant strides in reducing bias from static word embeddings. Various techniques have been developed to address the issue such as projecting embeddings onto bias-free dimensions using adversarial attacks or training models from scratch with fairness constraints. However, there has been comparatively less attention given to debiasing large-scale text encoders until recently. To address this gap, researchers have explored extending existing techniques to work with large text encoders. For example, one approach involves contextualizing words within sentences before applying existing debiasing techniques by [11]. Another approach involves adding bias-reduction objectives to the loss functions of the models by ([16]; [2]; [12]). Some researchers have also used CDA to balance gender correlations in training data [10]. While others have used adapters to reduce training time. Other debiasing techniques include contrastive learning, zero-shot learning, dropout, or regularizing the entropy of attention. These approaches aim to discourage models from basing their classifications on identity terms or reduce harmful associations with disadvantaged groups. The proposed method discussed in this text is different from the previous approaches. It aims to reduce harmful associations by calibrating the attention of the context on identity terms. This method is applied to text encoders as a general representation layer while the approach proposed by [1] is proposed specifically for hate-speech classification models.

3 Proposed Methodology

My goal for this project is to develop a method that could mitigate the bias in transformer based architectures and leverage the self-attention computed by each attention head in each layer. I primarily focus on transformer based encoders and gauge the bias present in the text representations outputted by these encoders like BERT. To facilitate this, I conduct experiments based on the BERT base model, which has 12 layers and 12 attention heads. In this study, I propose a model which utilizes attention correlation bias to debias BERT. I use the attention values that words in a sentence allocate to other words to calculate the bias. I also use these attention values to remove the bias in the encoder by altering these attention values.

I first create an augmented dataset as created by the authors in [6]. This dataset contains information about demographic groups (eg. gender) to enable me to measure the difference in attention certain words allocate to the words belonging to this demographic group (eg. he, she). My training was conducted on the bert_uncased model using the News-commentary-v15 dataset.

To calculate the bias present in the attention allocations, I make use of different correlation metrics. To enhance the model's robustness, I incorporated additional correlations to compute the bias metrics such as Spearman and Kendall correlation in addition to Pearson correlation. These metrics were chosen for their ability to handle outliers and non-normal distributions. Next, I use these values to gauge the bias of the model and debias it. To do this, I train the model to minimize a certain cost function which consists of two distinct losses. The first loss strives to equalize the attention allocations and thus reduce the bias while the second loss strives to reduce the loss of semantic information in the encoder so that the model can still create meaningful text representations.

To increase the model's flexibility, I made a significant modification to the loss function by introducing two distinct values - one static and the other dynamic. This allowed me to fine-tune the model for better performance and faster training.

I also experimented on different methods of training, which focus on debiasing the model on different areas of the architecture. Through extensive analysis, I determined that the most efficient debiasing approach was to focus exclusively on the top l most biased layers, top k most biased heads, and

top m heads of each layer. Further details on my analysis can be found in the subsequent sections.

4 Bias Metric Using Attention

Due to the bias present in data that the BERT model is trained on, this bias is propagated into the predictions it makes. Consider the example: "The CEO asked the receptionist a question. He looked at her with confusion." The BERT model automatically attributes "he" to "CEO" and "her" to "receptionist". In other words, the word "CEO" allocates more attention to "he" while receptionist allocates more attention to "her". This indicates that the model is biased. To calculate the amount of bias present, I use attention that a particular word (CEO, receptionist) allocates to different demographic attributes (he, her).

I first start by identifying different biases. In this project, I have included the following biases: gender, age, religion, race and sexual orientation. For this, I define a set of tuples for each bias.

Gender	Age	Race
boy, girl	old, young	White, Black
he, she	elderly, youth	African,
man, woman	adult, child	Caucasian

Table 1: Examples of group tuples per bias

Table 1 shows the examples of group tuples. Each tuple shows the attributes of a social group. I then create the augmented dataset using the News-commentary-v15 dataset based on the selected bias type. Each training example consists of two sentences s1 and s2. s1 is the original sentence while s2 is created by appending a sentence of keywords from one of these tuples. This is done to gauge the attention allocated by words in s1 on the biased keywords in s2. I pass the augmented input into the BERT (bert-base-uncased) encoder to find the attention weights. Attention weights are based on the importance that a certain token gives to another token. There is a bias present if tokens allocate different attention values to different social group attributes. The example in Figure 1. shows gender bias in the model. The words CEO and Receptionist pay more attention to "he" and "her" respectively indicating gender bias in the model.

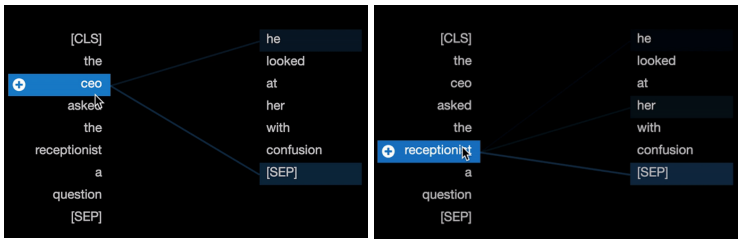
$$Bias(S,G) = \frac{1}{|S||G|} \sum_{s \in S} \sum_{s_g \in G} \frac{1}{|\binom{G}{2}|} \sum_{i,j \in \binom{G}{2}} \rho(A_s^{g_i}, A_s^{g_j}) \tag{1}$$

Here S and G are sets of sentences and social groups, respectively, and ρ is the Pearson correlation used to measure attention bias. The $\binom{G}{2}$ term produced all possible pairs of social groups given a tuple. The attention vector (A_s) represented the allocation of attention that the sentence (s) gave to group g_i. If the resulting value is close to 0, it indicates that attention exhibits bias since the average correlation across sentences and groups is nearly 0.

While Pearson correlation measures the linear relationship between two variables Spearman and Kendall correlations measure the monotonic relationship between two variables. Spearman correlation is calculated by first ranking the values of each variable, then calculating the Pearson correlation coefficient between the ranks of the two variables. Kendall correlation, on the other hand, is based on the number of concordant and discordant pairs in the data. A pair is considered concordant if the ranks of the two variables are in the same order and discordant if the ranks are in opposite orders. They provide a more accurate measure of the monotonic relationship between social groups in the attention weights, which may not be linear. They are also less sensitive to outliers which can be present in attention weights due to noisy data or specific language use. By replacing the Bias equation with these different correlation types I get different indicators/metrics of bias.

5 Debiasing

To perform debiasing and increase fairness, I finetune the text encoder to alter the attention of the model. Specifically, I aim to ensure that the biased attention heads provide equal attention allocation to all the groups in the augmented input eliminating any preference. This is done by adding an equalizing term to the loss function which ensures that the sentences are allocating equal attention to all demographic attributes. Altering the attention allocation can cause a loss in the semantic and syntactic abilities of the model. To curb this problem, I also minimize a semantic loss function that encourages the model to learn the original semantics of the input from an unaltered teacher model, which is accomplished by copying its internal attention. This approach enables me to mitigate bias while preserving the model's ability to capture the original semantics of the input. Thus, debiasing process



(a) (b)

Figure 1: Gender Bias in Baseline Model

involves two steps:

1. equalizing the attentions
2. preserving the semantic information.

5.1 Equalizing attentions for debiasing

The goal of attention equalization is to remove any biases or preferences towards particular attributes of social groups that may be present in the text encoder. This is accomplished by minimizing the difference in attention scores that each token in s1 pays to the tokens in s2. To achieve this, a fabricated sentence s2 is created by adding words related to a particular social group (as described in Section 4). Then, the attentions of s1 towards each group in s2 are evaluated, and the differences are minimized by tuning the encoder's parameters. The loss function is given by the sum of squared differences for all layers and heads over the training corpus.

$$L_{equ} = \sum_{s \in S} \sum_{l=1}^{L} \sum_{h=1}^{H} \sum_{i=2}^{||s_g||} ||A^{l,h,s,s_g}_{:\sigma,\sigma+1} - A^{l,h,s,s_g}_{:\sigma,\sigma+i}||_2^2 \tag{2}$$

The process involves identifying the top k1 layers and k2 heads in the encoder that exhibit the most bias. The attention scores of s1 towards the first social group token in s2, which immediately follows the special token [SEP], are used as the pivot vector. Then, the attention scores of s1 towards the other social group tokens in s2 are compared to the pivot vector, and the mean square error between them is calculated. The resulting equalization loss, denoted as L_{equ}, is given by Equation 2. L_{equ} is dependent on the number of layers L and heads H in the text encoder, as well as the number of social groups $||s_g||$ in s2. The special token [SEP] is used to mark the boundary between s1 and s2, and its position is denoted by σ. By minimizing the difference in attention scores between the first social group token and the other social group tokens, L_{equ} forces the encoder to pay equal attention to all social groups, thus eliminating biases and preferences.

5.2 Preserving Semantics Using Knowledge Distillation

To ensure that the debiased encoder does not lose its original semantics, I use knowledge distillation, a technique in which I train a student model to mimic the behavior of a well-performing teacher model (Hinton et al., 2015; Gou et al., 2021). In this setting, I use the original model as the teacher and the debiased encoder as the student. The teacher model is not subjected to my debiasing technique since it serves as a reference to the original unaltered language representations. To accomplish this, I require the student to learn the same attention distribution as the teacher model for each input in the training set S.

To minimize the loss of semantic information, I use another loss L_{distil} . To do this, I take an untouched BERT encoder for the target model to copy it. I do this by minimizing the MSE of the attention scores between the two models.

$$L_{distil} = \sum_{s \in S} \sum_{l=1}^{L} \sum_{h=1}^{H} ||A^{l,h,s,s_g}_{:\sigma,:\sigma} - O^{l,h,s,s_g}_{:\sigma,:\sigma}||_2^2 \tag{3}$$

The training objective can be expressed as a linear combination of the previously defined losses where the hyperparameter is utilized to regulate the importance of de-biasing versus semantic preservation.

$$\begin{aligned} Loss &= L_{distil} + \lambda L_{equ} \\ where \quad \lambda &= \frac{2L_{equ}}{L_{equ} + L_{distil}} \end{aligned} \tag{4}$$

Here, A(l,h,s,s_g) represents the attention scores of the student model at layer l and head h when

s and s_g are provided as input. Similarly, I use O(l,h,s,s_g) to denote the attention scores of the teacher model. The preservation of semantic information is formalized as a regularizer, which minimizes the squared L2 distance between the attention scores of the student and the teacher. By minimizing this distance, I ensure that the student model learns to attend to the same linguistic features as the teacher, thus preserving the original semantics of the language.
The training objective can be expressed as a linear combination of the previously defined losses where the hyperparameter λ is utilized to regulate the importance of debiasing versus semantic preservation.

6 Implementation

In order to conduct comprehensive experiments and gain a deeper understanding of the impact of different training methods on model performance, I believe it is necessary to create multiple models and compare their results. To achieve this, I implement several variations of the training function, with a specific focus on addressing biases in specific parts of the model.
My approach involves performing attention alteration on different parts of the model, taking into account the bias in different layers and heads of each layer. By creating variations in the training function, I aim to mitigate the effects of bias and improve the overall performance of my models. These biases are measured based on the metric described in Section 4. I hypothesize that debiasing all attention heads in all attention layers is not required to obtain a substantial reduction in bias. This is because the bias is localized in certain regions of the model and operating only on the attentions of those layers could be sufficient. To test my hypothesis, I implement variations of models based on careful selection of certain hyperparameters and logical choices. The aim of these experiments is to find an optimal tradeoff between obtaining the least bias, developing the best accuracy on downstream NLP tasks as well as improving efficiency.

Model Details

For all the following models, I utilize the News-commentary-v15 corpus to generate positive and negative sentences, with 80% of the data used for training comprising 133,892 examples, and the remaining 20% utilized for validation. The models are trained using the Adam optimizer with a learning rate of 5e-6, and training is conducted over 3 epochs. Manual hyperparameter tuning is employed during the training process.

6.1 Standard Model

In this approach, I retrain the BERT encoder by implementing a technique that addresses bias at the all attention heads of every transformer layer, resulting in a more comprehensive and thorough debiasing process.
I calculate the bias by using my attention bias metric for each attention head by finding the correlation of the attention in different groups. Then the model is trained to mitigate the bias in each self-attention head. In other words, I equalize the attention produced by all attention heads in all layers of the model. By targeting the entirety of the model's bias rather than focusing on the bias present in specific regions, I aim to improve the overall performance and accuracy of my model. Through this approach, I are able to mitigate biases that may exist in specific parts of the model, while also ensuring that all attention heads are treated equally.
My experiments were conducted on the BERT base model, consisting of 12 transformer layers and 12 self-attention heads. Thus, this method performs debiasing on all 12*12=144 heads. Given that this debiasing approach involves retraining the model on every head of each layer, it is important to acknowledge that this process is computationally intensive.

6.2 Debiasing k most biased heads

This debiasing approach adopts a targeted strategy to mitigate the influence of attention bias by focusing on the most biased attention heads. By addressing these specific attention heads, I aim to achieve a more efficient debiasing process that can improve the overall performance of the model.
The methodology involves measuring the attention bias correlation (as mentioned in Section 4) for the attention heads in each transformer layer. Using these values, I calculate the bias in all the attention heads in the model based on the attention bias metric. By setting a value for k, I use these attention bias values to retrieve a list of the k most biased heads. This list is represented as a set of tuples (l,h) where l denotes the layer number and h denotes the head number.
The BERT base model is then trained using this list ensuring that only the attention heads present in the list are debiased, i.e. attention values are

equalized for different groups, while the attention values of the remaining heads remain externally unchanged. After hyperparameter tuning, I determined that k=40 is the optimal value for achieving a good tradeoff between the amount of bias and training time. This method does not account for the layer that the biased attention head belongs to and only makes a holistic consideration of the most biased heads.

6.3 Debiasing l most biased layers

In this approach, I adopt a layer-level debiasing strategy that focuses on entire attention layers in the transformer, rather than individual self-attention heads. Essentially, this means that I examine all the heads in a given layer and use that as the basis for mitigating bias.

To achieve this, I first measure the attention bias for each attention head in every transformer layer. Then, I calculate the average bias of all the heads in each layer, thus obtaining the bias present in each layer. From there, I create a list of the l most biased layers based on these values. This list is represented as a list of size l, where each value denotes the corresponding layer number. Using l=6, I train the BERT model such that only these 6 layers are debiased. This means that the attention produced by all the heads in these 6 layers is equalized. My experimentation revealed that selecting l=6 achieves a significant reduction in bias while also cutting the computation time by half compared to the standard model.

I also observed that the bias is much more pronounced in the intermediate layers of the BERT model and decreases as I move further into the network (as discussed in more detail in Section 7.1.1). This finding indicates that debiasing the top-most biased layers is an efficient strategy, as it addresses the most critical areas of bias without expending resources on later layers.

6.4 Debiasing m most biased heads in each layer

This approach aims to strike a balance between the standard model and the top-k heads model. Similar to the standard model, I address the bias present in all layers of the transformer encoder, but unlike the standard model, I focus on debiasing only the most biased heads of each layer. This allows me to comprehensively address the bias across the entire encoder while also optimizing computational resources.

In this method, I first measure the attention bias for each attention head in each transformer layer, similar to the other models. Then, I calculate the bias of all the attention heads in each layer based on the attention bias metric. Using hyperparameter tuning, I select a value for m, and retrieve a list of the m most biased attention heads in each layer. This is represented in the form of a mapping l:[h]. Here, l denotes the layer number and [h] denotes a list of size m representing the heads which are most biased in layer l. I then train the BERT base model such that only these heads corresponding to each layer are debiased, while leaving the rest of the attention heads externally unaltered.

This approach ensures that every layer of the transformer is accounted for in the debiasing process, while also reducing computational expense by focusing only on a select few attention heads within each layer.

7 Results and Evaluation

In the previous sections, I described the implementation details and the experimental setup. In this section, I evaluate my method mainly based on two viewpoints:

1. Evaluation of Fairness
2. Evaluation of Semantics

By conducting these two types of evaluations, I can obtain a comprehensive understanding of the performance of the model. However, due to limitations in terms of time and resources, I have primarily focused on the first type of evaluation for the current study.

7.1 Evaluation on Fairness

I conducted fairness evaluation by measuring the attention correlation bias (metric described in Section 4). This is an intrinsic method of evaluation, which means that it does not consider the application of the text representations but rather evaluates the bias in the text representations themselves.

As previously mentioned, I use three types of correlations to calculate the bias in the model using attention correlation bias: Pearson correlation, Spearman correlation and Kendall correlation. I calculate the bias for five types of bias namely gender, race, religion, age and sexual orientation for each of these correlation types. I compare these biases by performing these evaluations on

Correlation/Bias Type	Gender	Race	Religion	Age	Sexual Orientation
Pearson	0.8452	0.7956	0.8257	0.8567	0.8694
Spearman	0.7845	0.6956	0.7732	0.8080	0.8012
Kendall	0.6541	0.6550	0.6215	0.6683	0.6551

Table 2: Bias in Baseline model for different bias types

Correlation/Bias Type	Gender	Race	Religion	Age	Sexual Orientation
Pearson	0.9069	0.9011	0.9018	0.9013	0.9112
Spearman	0.8656	0.8556	0.8566	0.8584	0.8842
Kendall	0.7267	0.7303	0.7280	0.7183	0.7345

Table 3: Bias in Standard model for different bias types

Models/Correlations	bert_base_uncased	Standard	Top_heads	Top_layers	Top_heads_per_layer
Pearson	0.8452	0.9069	0.8867	0.8812	0.8905
Spearman	0.8007	0.8656	0.8510	0.8404	0.8573
Kendall	0.6483	0.7267	0.6889	0.6925	0.7220

Table 4: Gender bias for all models

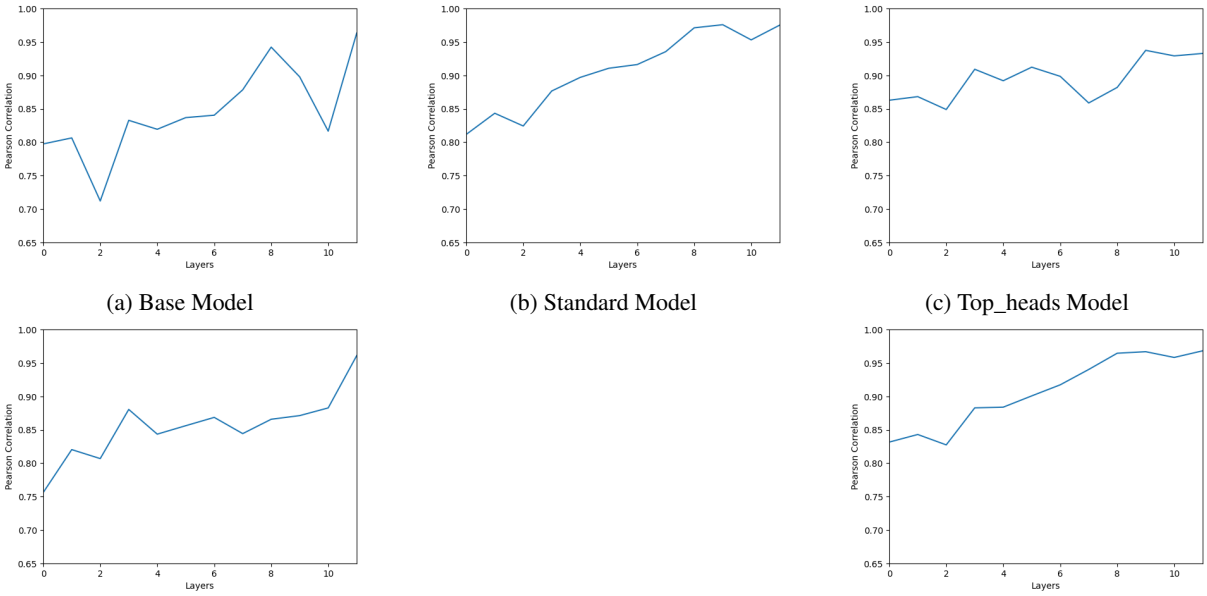
the unaltered biased BERT as the baseline model and my debiased BERT standard model (Section 6.1). The results of the evaluation can be seen in Table 2 and 3. Table 2 shows the bias in baseline biased BERT model using all three correlations across all the biases used during training while Table 3 shows the same for the debiased BERT model. A higher score indicates lesser bias.

The attention correlation bias scores in Table 2 and Table 3 clearly indicate that the Standard model is capable of performing debiasing, as evidenced by the higher values in Table 3 compared to the corresponding values in Table 2. This shows that equalizing attention scores and removing bias from the attention of the encoder clearly helps in debiasing the transformer model.
There is no evidence as to which method of correlation is better suited for calculation of bias. A further investigation on this topic would be necessary to determine the best correlation to be put to use. While Pearson correlation is more reliable in capturing statistical associations [18] Spearman and kendall correlations are robust to outliers and do not assume that the variables correlated are normally distributed which makes them better indicators of attention bias. Overall, I observe that using Pearson correlation gives the highest values of attention correlation bias, while Spearman gives slightly lower values followed by Kendall correlation, using which I obtain the lowest bias correlation values.
To test my various models (as mentioned in Section 7), I focus on only one demographic bias, namely Gender. Table 4 illustrates the gender attention bias values for the different models. While the standard model outperforms the rest of the models, it is worth noting that the lack of significant difference makes the other models a preferable option in terms of training time. This finding also supports my hypothesis that training all layers and all heads is redundant and unnecessary. For example, if I consider the results using Pearson correlation, the top_1_layers model has a 2.22% higher bias as compared to the standard model but the training time is cut down by half (considering l=6). Table 4 shows that training top_heads_per_layer outperforms top_heads and top_layers model and is in accordance with my hypothesis.

7.1.1 Bias in Layers

Figure 2 presents the results of computed bias on the baseline unaltered BERT base model as well as the standard, top_heads, top_layers and top_heads_per_layer models. The graphs show the attention correlation bias in each layer for each of the models. Bias in each layer is calculated by calculating the average of the attention correlation bias in each head of that layer. Only Pearson correlation is used here for the purpose of comparison.

Figure 2a shows the attention correlation bias



(a) Base Model

(b) Standard Model

(c) Top_heads Model

(d) Top_layers Model

(e) Top_heads_per_layer Model

Figure 2: Line Graphs indicating attention correlation bias versus layers for all models

in the original unaltered BERT base model. Ideally, the graph should be a straight horizontal line cutting the y-axis at 1. This denotes that none of the words in the sentence show any bias against attributes of the social groups. In the following graphs I can see that the graphs are far from ideal, but this is because of the distillation loss term which was added to the total loss in order to cater for semantic preservation.I can clearly see the pattern of the bias in each layer – the trend of the correlation bias is upwards. The value is lower in the intermediate layers of the model and increases as I go further into the later layers. This means that the bias is much more pronounced in the intermediate layers of the encoder and decreases as I go further into the network. This could be because the lower layers are closer to the input layer and receive a direct exposure to the text data. As a result, they may pick up on more nuanced biases that are present in the language data. On the other hand, the top layers have more abstract representations of the text, and may be less influenced by the biases present in the data. Since the lower intermediate layers are responsible for capturing more syntactic and semantic information, they contribute more towards encoding the biases in the model.

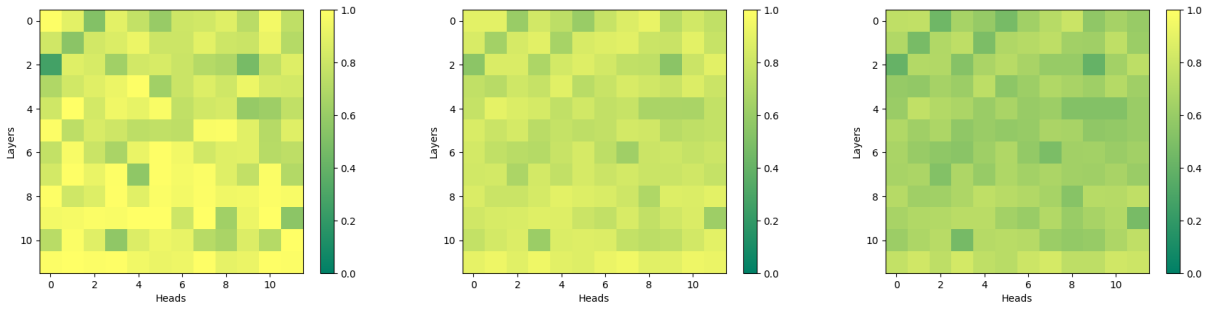
Figure 2b, 2c, 2d and 2e show the same data evaluated on my debiased models. I see that all the debiased models show better results than the baseline model with higher values of attention bias correlation. I can also observe that the slope of the graphs is reduced (the graph is flatter than the baseline), i.e. the difference in the bias between the intermediate layer and the top layers has reduced. Thus, my models were able to successfully remove more bias from the intermediate layers than the top layers.

7.1.2 Bias in Heads

To visualize how each head of the encoder contains bias and how it changes after debiasing, I also create heat maps. These heat maps show the attention bias correlation in each self-attention head of the layer of the encoder. They show the results of computed bias broken out by layers and heads. I have used Pearson, Spearman, and Kendall correlation to evaluate the bias, and have included heat maps for all models: the baseline unaltered BERT base model, as well as the standard, top_heads, top_layers, and top_heads_per_layer models.

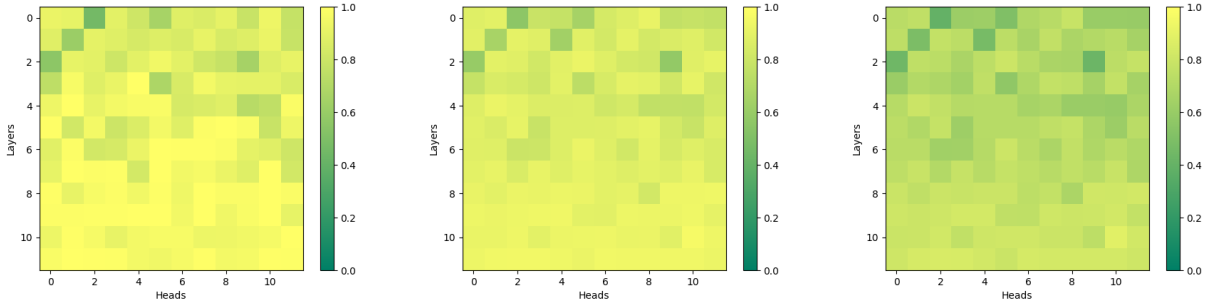
In Figure 3, I present a visualization of the attention correlation bias in the original BERT model. The darker colors indicate lower attention correlation bias and higher bias for the corresponding layer and head. In contrast, lighter colors indicate lower bias. I observe that the layers that are closer to the input exhibit higher bias (represented by green), while the layers closer to the output have lower bias (represented by yellow). This finding reaffirms the information presented in Section 7.1.1.

Figure 4 shows the heat maps for my standard



(a) Pearson (b) Spearman (c) Kendall

Figure 3: Bias in Baseline Model



(a) Pearson (b) Spearman (c) Kendall

Figure 4: Bias in Standard Model

model. I can clearly see that the map appears to be much less biased as compared to the baseline model indicated by the change in color of the map. Although the overall bias seems to have reduced over the entire encoder, the bias is still more concentrated on the first few layers as compared to the later layers.

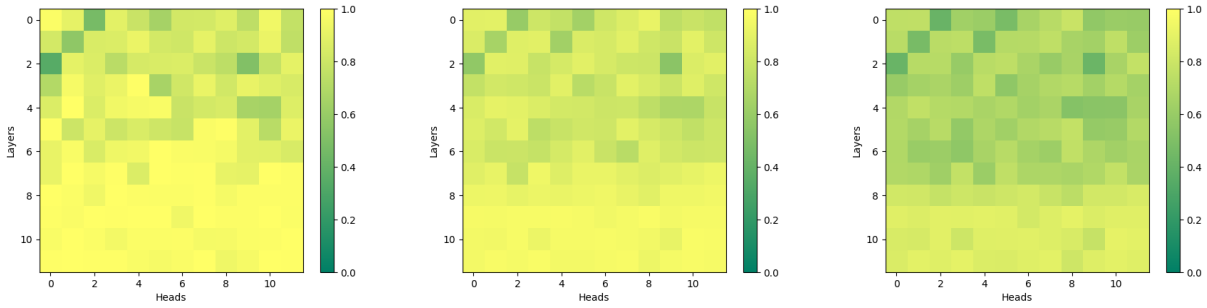
Figure 5 shows the heat maps for the top_heads model, which involves training by debiasing only the k most biased heads. It is evident that by focusing only on the most biased heads, I were able to achieve a reduction in bias not only in those heads but also in other heads that were not debiased. This can be attributed to the fact that by debiasing only a few heads that had the most significant contribution to bias, I were able to stop the propagation of this bias to other heads and layers, resulting in an overall debiased model. However, I see that the difference in the bias in the upper and lower layers of the transformer is much more stark compared to the standard model. This could be due to the fact that some of the heads in the first few layers were not considered for debiasing.

In Figure 6, I present the heat maps of the top_layers model, which was trained by debiasing only the l most biased layers. In most cases, these layers were the first l layers of the encoder. The heat maps show that the bias is more uniform across different layers than the other models, indicating that the debiasing influenced the earlier layers more than the later layers.

Figure 7 represents the heat maps for the top_heads_per_layer model, which was trained by debiasing only the m most biased heads in each layer of the encoder. This model also gives great results in terms of debiasing and performs better than the other models barring the standard model.

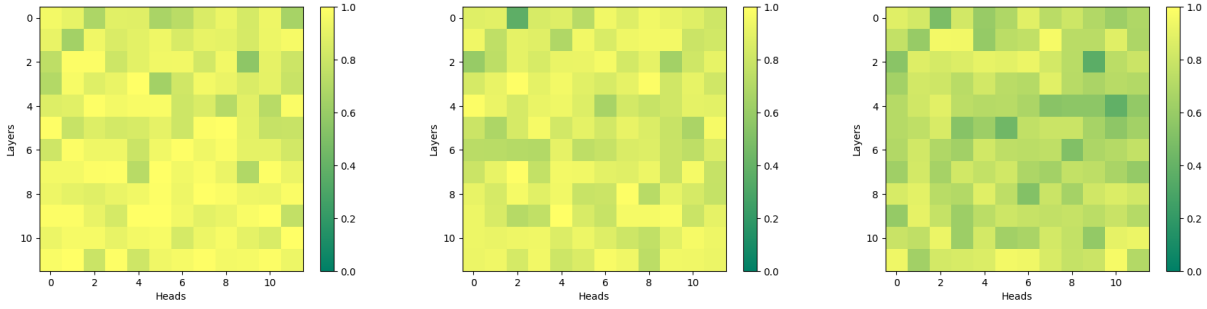
7.2 Evaluation of Semantics

I use the GLUE benchmark (Wang et al., 2018) to test whether the debiased text encoder maintains sufficient semantic information for downstream NLP tasks. GLUE is a reliable measure of natural language understanding capabilities for NLP models and is a suitable tool to evaluate the semantic preservation of my model. One of the GLUE tasks I used is the SST-2 task, which is a binary classification task that predicts the sentiment of a sentence as either positive or negative. I fine-tune my debiased models on this task and demonstrate task accuracy preservation against some state of the art models in Table 4. By examining the table, I can



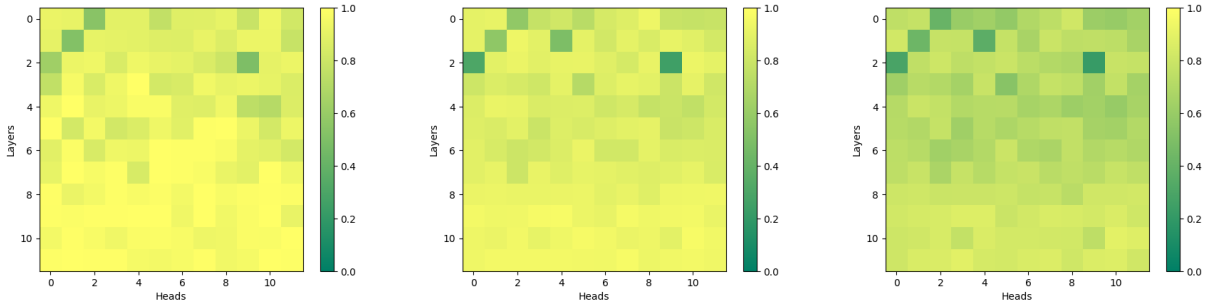
(a) Pearson (b) Spearman (c) Kendall

Figure 5: Bias in Top_Heads model



(a) Pearson (b) Spearman (c) Kendall

Figure 6: Bias in Top_layers model



(a) Pearson (b) Spearman (c) Kendall

Figure 7: Bias in Top_heads_per_layer model

observe that the decrease in accuracy in my models is not significant enough. Moreover, the model is proficient in preserving the semantic information. Considering the trade-off between training time and accuracy, I believe that my models can achieve a reasonable level of accuracy while reducing the training time.

Model	sst2 task accuracy
BERT	92.78
Sent-D	91.63
Kaneko	91.97
Standard	92.66
Top_Heads	92.68
Top_Layers	92.61
Top_heads_per_layer	92.64

Table 5: Standard model / all biases / all correlations

7.3 Conclusion

My proposed model is designed to address bias in attention layers which can occur when certain demographic groups are given more or less attention than others. To tackle this issue, I utilize a set of unique and dynamic loss functions that enable me to quantify bias in the attention allocation for demographic group words. This approach is more effective than simply training the model on a dataset that has been balanced for demographic groups as it ensures that the model itself is designed to recognize and correct any biases that may exist. To calculate equalized losses, I use Pearson, Spearman, and Kendall Correlation between the attention allocation for demographic group words. These metrics help me to determine how much attention is being allocated to each group and how evenly it is being distributed. By using a combination of these metrics, I can ensure that the model is sensitive to variations in attention allocation across different demographic groups. In addition to addressing bias, my model also preserves semantic information using Distill loss. This approach ensures that the model is able to capture the underlying meaning of words and phrases, even as it works to correct any biases that may exist in the attention allocation. To further improve the performance of my model, I use a linear combination of the loss functions and a dynamic lambda. My research also indicates that training all layers and heads is unnecessary, as it can be time-consuming. Instead, I find that training a smaller number of heads per layer yields the best results. This is because the first layers of the model are directly impacted by the input, while later layers receive transformed input. By training a smaller number of heads per layer, I can strike a balance between accuracy and train time without compromising performance.

8 Future Work

Even though my model gives a good amount of performance it should be reproduced on the large BERT. Firstly the models consider three dimensions of social divisions, and it may not be clear which or how many groups to include. Thus, the experiments are restricted to common use cases. Secondly, certain words may have inherent biases, which do not require debiasing, and compiling lists of related words for every social group can be expensive. Thirdly, the models currently deal with debiasing discrete words, and it is not obvious how to extend this approach to implicit bias or biases towards finer-grained groups. I don't consider intersectional bias which is prominent in the real world. To address these limitations, further research is needed, including exploring different kinds of template structures and developing approaches to treat implicit biases and biases towards finer-grained groups.

References

[1] Giuseppe Attanasio, Debora Nozza, Dirk Hovy, and Elena Baralis. Entropy-based attention regularization frees unintended bias mitigation from lists. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1105–1119, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[2] Shikha Bordia and Samuel R. Bowman. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[3] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, apr 2017.

[4] Sunipa Dev, Tao Li, Jeff Phillips, and Vivek Srikumar. On measuring and mitigating biased inferences of word embeddings, 2019.

[5] Mark Diaz, Isaac Johnson, Amanda Lazar, Anne Marie Piper, and Darren Gergle. Addressing

age-related bias in sentiment analysis. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14, New York, NY, USA, 2018. Association for Computing Machinery.

[6] Yacine Gaci, Boualem Benatallah, Fabio Casati, and Khalid Benabdeslem. Debiasing pretrained text encoders by paying attention to paying attention. pages 9582–9602, December 2022.

[7] Yacine Gaci, Boualem Benatallah, Fabio Casati, and Khalid Benabdeslem. Masked language models as stereotype detectors? In Julia Stoyanovich, Jens Teubner, Paolo Guagliardo, Milos Nikolic, Andreas Pieris, Jan Mühlig, Fatma Özcan, Sebastian Schelter, H. V. Jagadish, and Meihui Zhang, editors, *Proceedings of the 25th International Conference on Extending Database Technology, EDBT 2022, Edinburgh, UK, March 29 - April 1, 2022*, pages 2:383–2:387. OpenProceedings.org, 2022.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. 2015.

[9] Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172, Florence, Italy, August 2019. Association for Computational Linguistics.

[10] Anne Lauscher, Tobias Lüken, and Goran Glavaš. Sustainable modular debiasing of language models, 2021.

[11] Paul Pu Liang, Irene Mengze Li, Emily Zheng, Yao Chong Lim, Ruslan Salakhutdinov, and Louis-Philippe Morency. Towards debiasing sentence representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5502–5515, Online, July 2020. Association for Computational Linguistics.

[12] Sheng Liang, Philipp Dufter, and Hinrich Schütze. Monolingual and multilingual reduction of gender bias in contextualized representations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5082–5093, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[13] Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[14] Moin Nadeem, Anna Bethke, and Siva Reddy. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online, August 2021. Association for Computational Linguistics.

[15] Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online, November 2020. Association for Computational Linguistics.

[16] Yusu Qian, Urwa Muaz, Ben Zhang, and Jae Won Hyun. Reducing gender bias in word-level language models with a gender-equalizing loss function. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 223–228, Florence, Italy, July 2019. Association for Computational Linguistics.

[17] Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online, July 2020. Association for Computational Linguistics.

[18] Vitalii Zhelezniak, Aleksandar Savkov, April Shen, and Nils Hammerla. Correlation coefficients and semantic textual similarity. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 951–962, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.