# From Prejudice to Parity: A New Approach to Debiasing Large Language Model Word Embeddings

**Anonymous ACL submission**

## Abstract

Embeddings play a pivotal role in the efficacy of large language models. They are the bedrock on which these models grasp contextual relationships and foster a more nuanced understanding of language and consequently perform complex tasks that require a fundamental understanding of human language. Given that these embeddings themselves often reflect or exhibit bias, it stands to reason that these models may also inadvertently learn this bias. In this work, we build on the aforementioned seminal work of (Bolukbasi et al., 2016) and (Gonen and Goldberg, 2019) and propose *DeepSoftDebias*, an algorithm that uses a neural network to perform 'soft debiasing'. We exhaustively evaluate this algorithm across a variety of state-of-the-art datasets, accuracy metrics, and challenging NLP tasks. We find that *DeepSoftDebias* outperforms the current state-of-the-art methods at reducing bias across gender, race, and religion.

## 1 Introduction

Word embeddings are a foundational element in the architecture of Large Language Models (LLMs). They act as the basis for these models to understand and subsequently, generate human-like language. However, it has been shown that these word embeddings themselves may reflect or exhibit bias (Dev et al., 2020; May et al., 2019; Caliskan et al., 2017). Given the exponential increase in the use of LLMs on a plethora of downstream tasks, these representations can amplify bias and result in discriminatory actions, especially when it comes to the fields of education, healthcare, and justice. Existing work in this field has looked most commonly into gender bias (Kotek et al., 2023; Bordia and Bowman, 2019; de Vassimon Manela et al., 2021), racial bias (Mozafari et al., 2020; Omiye et al., 2023; Tang et al.), and religious bias (Baligudam, 2022; Kirk et al., 2021). In this work, we build on the seminal work of (Gonen and Goldberg, 2019), which brought attention to the inherent biases present in traditional GloVe embeddings (Pennington et al., 2014). This study prompted the NLP community to reevaluate the fundamental choices underlying our word representation models. Specifically, we present *DeepSoftBias*: an algorithm that furthers the application of their methodology, by diverging from the conventional GloVe embeddings and delving into the word embeddings produced by the best-performing models on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2022) leaderboard. By employing these advanced embeddings on the same set of words as used in GloVe embeddings, we seek to investigate whether these state-of-the-art (SoTA) models inherently exhibit reduced bias.

Our primary objective is twofold: first, to de-bias the embeddings from these selected models, and second, to rigorously assess the effectiveness of the bias removal process. Our proposed approach, *DeepSoftDebias*, is an innovative methodology to de-bias LLM word embeddings which involves integrating a neural network into the soft debiasing approach developed by (Bolukbasi et al., 2016). This novel amalgamation is driven by the aspiration to enhance the debiasing process and contribute to the ongoing discourse on creating fair and ethically sound language models. To this end, our work answers the following research questions:

**RQ1:** Compared to traditional methods, does our proposed methodology attain better performance metrics when it comes to debiasing SOTA model embeddings?

**RQ2:** How do parameters of the model (size, complexity) interact with various SOTA debiasing techniques? What effect do they have on each other?

**RQ3:** To what extent do various SOTA debiasing techniques influence the performance of models on different downstream tasks?

**RQ4:** How does the type of bias (gender, race, religion) affect the effectiveness of the debiasing process?

To answer the above questions, we make the following contributions through this research:

**Our Contributions**

- We provide, to the best of our knowledge, the first comprehensive study of how various debiasing methods work on SoTA LLM word embeddings
- We present a novel methodology, *DeepSoftDebias*, for debiasing LLM word embeddings, which beats SoTA debiasing methods across multiple bias formats including gender, race, and religion.
- We perform an exhaustive quantitative analysis, establishing SoTA baselines and leveraging multiple evaluation metrics to provide a comparison against accessible SoTA baselines.

We illustrate our pipeline in Fig. 1. We find that *DeepSoftDebias* not only outperforms the state-of-the-art methods at reducing bias across gender, race, and religion but also does so while preserving the full information of the original embedding (which is an additional improvement on previous methods). Further, we find that model performance on challenging downstream tasks like NER and sentiment analysis remains largely unaffected when we test using our debiased embeddings.

## 2 Related Work

**INLP** Iterative Null-space Projection (INLP) (Ravfogel et al., 2020) is a post-hoc debiasing method that operates at the representation level. The INLP methodology debiases representations by iteratively projecting them into a linear classifier's null space. This technique is particularly effective for handling intersectional groups, which are defined by combinations of sensitive attributes[2]. INLP seeks to learn a hidden representation that is independent of the protected attributes. This approach is beneficial in scenarios where an attempt to make a model fairer towards some group results in increased unfairness towards another group. Therefore, INLP emerges as a robust and effective strategy for mitigating bias in language models, promoting fairness across multiple protected attributes.

**Self-Debias** Self-Debiasing (Schick et al., 2021) is a novel approach to mitigating bias in language models. The methodology, first coined by Schick et al. (2021), is based on the concept of self-diagnosis. In this approach, pretrained language models recognize their undesirable biases and the toxicity of the content they produce. Based on this self-diagnosis, a decoding algorithm is proposed that reduces the probability of a language model producing problematic text. This approach, referred to as self-debiasing, does not rely on manually curated word lists, nor does it require any training data or changes to the model's parameters. While it does not completely eliminate the issue of language models generating biased text, it is an important step in this direction. The self-debiasing approach demonstrates the potential of language models to self-regulate and reduce their inherent biases.

**Sentence Debias** SentenceDebias (Liang et al., 2020) is a debiasing methodology that operates at the sentence level. It is a projection-based method that identifies a linear subspace associated with a specific bias. The sentence representations are projected onto this bias subspace, and the projection is subtracted from the original representations. This process effectively debiases the sentence representations. SDB is particularly useful for mitigating biases related to gender, race, and religion. It offers a comprehensive comparison between models that adjust weights for debiasing and those employing test-time surgical interventions. The SDB method signifies a significant advancement in debiasing strategies, promoting a more equitable representation in language models.

**Counterfactual Data Augumentation** Counterfactual Data Augmentation (CDA) (Yadav et al., 2023) is a data-based debiasing strategy often used to mitigate gender bias. The CDA methodology involves re-balancing a corpus by swapping bias attribute words (e.g., he/she) in a dataset. This technique is part of a broader set of debiasing techniques that also includes Dropout, Self-Debias, SentenceDebias, and Iterative Nullspace Projection. CDA has been applied to various language models, including BERT, with the goal of diminishing stereotypical biases while maintaining the model's performance on downstream tasks. However, it's important to note that while CDA has the potential to improve the fairness of NLP models, it may not be effective in eliminating all biases and may even introduce new biases or errors in the model[3].

**FineDeb** FineDeb (Saravanan et al., 2023) is a two-phase debiasing framework for language models. In the first phase, FineDeb debiases the model by modifying the embeddings learned by the language model. This process involves contextual de-

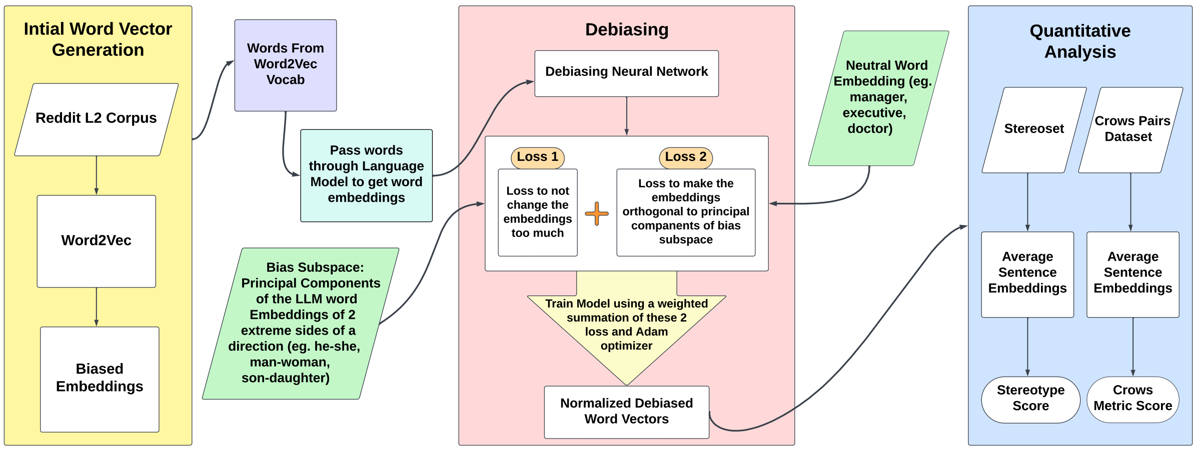


Figure 1: A step-by-step visualization of the pipeline for *DeepSoftDebias*. Our pipeline has 3 major components, Initial Word Vector Generation, Debiasing, and Quantitative Analysis. The Debiasing stage leverages the *DeepSoft-Debias* network.

biasing of these embeddings. In the second phase, the debiased model is fine-tuned on the language modeling objective. This methodology is effective for demographics with multiple classes. The FineDeb approach demonstrates its effectiveness through extensive experiments and comparisons with state-of-the-art techniques. It offers stronger debiasing in comparison to other methods, which often result in models as biased as the original language model. Thus, FineDeb emerges as a robust and effective framework for mitigating bias in language models.

## 3 Data

This study leverages several datasets to examine and address biases in word embeddings and language models, focusing on the representation and perpetuation of stereotypes within these systems.

**L2-Reddit Corpus** We utilize the L2-Reddit[1] (Rabinovich et al., 2018) corpus, a collection of Reddit posts and comments by both native and non-native English speakers, featuring approximately 56 million sentences. This dataset serves as the foundation for training word embeddings, aiming to capture the nuanced and inherently biased linguistic patterns present in social media discourse. In our study, we employ the Reddit L2 corpus as the source for our initial Word2Vec (Mikolov et al., 2013) word embeddings. Subsequently, we leverage the vocabulary derived from these word vectors to obtain the word embeddings from the LLMs. We utilize Word2Vec on the Reddit-L2 corpus to obtain the vocabulary. This vocabulary comprises the words for which we aim to extract embeddings from the LLMs. The primary objective of this approach is to ensure a consistent set of words across all our LLMs. This consistency allows each of our LLMs to be tested on the same set of words.

**StereoSet** StereoSet (Nadeem et al., 2020) stands out as a critical dataset for measuring stereotype bias in language models, containing around 17,000 sentences across demographic dimensions like gender, race, religion, and profession. It introduces the Context Association Tests (CAT) for evaluating model preferences and biases, providing a structured approach to assess and quantify biases in popular models like BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2020). In our work, we use the Stereoset dataset to benchmark our debiasing method.

**CrowS-Pairs** CrowS-Pairs (Nangia et al., 2020), designed to assess social biases in masked language models (MLMs), comprises 1,508 examples covering nine bias types, including race, religion, and age. It contrasts sentences related to historically disadvantaged and advantaged groups in the U.S., with annotations from crowd workers highlighting the degree of stereotyping. In our study, we obtain debiased word embeddings for sentences by computing the average sentence vector for both less and more stereotypical or anti-stereotypical directions. We then compare these embeddings against each other to calculate the Crows Metric score.

[1] https://github.com/ellarabi/reddit-l2

## 4 Methodology

In this section, we delve into the domain of debiasing word embeddings, presenting both an established and a newly proposed methodology for mitigating biases in word vector representations. These biases span across gender, racial, and religious lines and are encoded inadvertently within language models.

### 4.1 Bias Identification and Data Structure

To quantitatively assess bias in word embeddings, we measure the projection of word vectors onto a gender-specific axis, defined by the vector difference between the terms 'he' and 'she.' The magnitude of this projection serves as an indicator of bias. We use a structured vocabulary with its associated vector representations from the Word2Vec model to facilitate the identification of biases. For a comprehensive evaluation, we utilize additional data files that include definitive sets of gender-associated word pairs, analogy templates that list occupational roles often linked with specific genders, and a set of neutral terms used as evaluation targets. These resources are crucial for the systematic identification and rectification of biases in word embeddings. The words used for the BiasSpace are present in AppendixA.

### 4.2 Soft Debiasing: The Baseline Approach

The initial method as seen in (Manzini et al., 2019) leverages a method called soft debiasing. We recap its algorithm in Algorithm 1. Soft debiasing involves learning a projection of the embedding matrix that preserves the inner product between biased and debiased embeddings while minimizing the projection onto the bias subspace of embeddings mentioned in 4.1. Given embeddings $W$ and $N$ which are embeddings for the whole vocabulary and the subset of bias-neutral words respectively, and the bias subspace $B$ obtained in Section 3, soft debiasing seeks a linear transformation $A$ that minimizes the following objective defined in Eq. (1) as follows:

$$\left\|(AW)^T(AW) - W^TW\right\|_F^2 + \lambda \left\|(AN)^T(AB)\right\|_F^2 \quad (1)$$

Minimizing the first term preserves the inner product after the linear transformation $A$, and minimizing the second term minimizes the projection onto the bias subspace $B$ of embeddings. $\lambda$ is a tunable parameter that balances the two objectives. **W** here refers to the matrix of word embeddings and **N** refers to the matrix of the embeddings of the neutral space i.e. words that aren't influenced by any bias.

**Algorithm 1:** Transformation Matrix Approach

**Input:** Biased word embeddings ($\text{emb}_{\text{biased}}$), Bias Subspace (BiasSpace), Neutral word embeddings ($\text{emb}_{\text{neutral}}$)

**Output:** Debiased word embeddings

Perform Singular Value Decomposition (SVD) on $\text{emb}_{\text{biased}}$ to obtain singular values ($s$) and left singular vectors ($u$);
Precompute $t1 = s \cdot u^T$ and $t2 = u \cdot s$;
Compute norm1 as $\|t1 \cdot (T^T \cdot T - I) \cdot t2\|_F$;
Compute norm2 as $\|\text{emb}_{\text{neutral}}^T \cdot T^T \cdot \text{BiasSpace}\|_F$;
Total loss is a weighted combination of norm1 and norm2;
Optimize transformation matrix using SGD;
Output debiased word embeddings after recomputing using $T$ and normalizing;

### 4.3 *DeepSoftDebias*: Our Proposed Approach

In the original approach introduced by (Bolukbasi et al., 2016), a transformation matrix is utilized and optimized by an optimizer to enable a direct mapping between input and output embeddings. To enhance performance, we propose *DeepSoftDebias*. In this approach, we replace the transformation matrix with a neural network, leveraging its capability to represent a sequence of transformation matrices. This adaptation enables the algorithm to handle more complex functions mapping between input and output embeddings. We use the same loss functions as mentioned in the section 4.2. Furthermore, we transition from stochastic gradient descent (SGD (Robbins and Monro, 1951)) to the Adam (Kingma and Ba, 2017) optimizer, resulting in enhanced efficiency, speed, and optimization quality. We describe our full algorithm in Algorithm 2. While these modifications were implemented, the fundamental aspects of the method remain unaltered, ensuring minimal alterations in embeddings and preserving orthogonality with the bias space.

Unlike the baseline, which relies on singular value decomposition (SVD) and incurred information loss, *DeepSoftDebias* preserves the full information of the original matrix. Moreover, unlike the baseline, *DeepSoftDebias* can handle large embedding dimensions of more than 4.5k. We demonstrate the effectiveness of *DeepSoftDebias* on various datasets and tasks, and show that it outperforms the state-of-the-art methods in terms of accuracy and efficiency. The reason for the need of a fixed *Biasspace* is that we adopt the methodology proposed by Bolukbasi et al. for the derivation of the bias subspace.

The process of creating the *BiasSpace* commences with the identification of word vectors representing opposing concepts, such as 'he' versus 'she', or 'man' versus 'woman'. For each pair, we compute the mean vector, which encapsulates the shared semantic space. Subsequently, we subtract this mean vector from the original word vectors, yielding vectors that exclusively represent the bias components. These bias vectors are then concatenated to form a matrix, referred to as the bias subspace. This bias subspace plays a pivotal role in the training of our neural network. Specifically, we ensure that the output of the word embeddings, upon being processed through the neural network, is orthogonal to the bias subspace Fig. 2 presents a visualization of our approach to downstream testing.

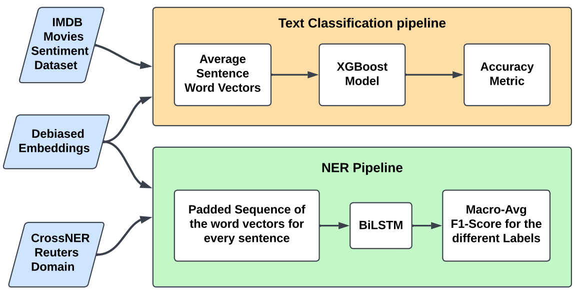


Figure 2: A step-by-step visualization of our downstream testing process to effectively evaluate *DeepSoftDebias*.

## 5 Effects of LLM Size and Dependency of Network Size

The debiasing performance of word embeddings depends on the size of the embeddings and the depth of the debiasing neural network, rather than the number of parameters of the language model. We observe in 11 Smaller models, such as bge-small (Xiao et al., 2023) and DeBERTa-v3-base (He et al., 2023) or DeBERTa-v3-large, can be debiased effectively by a single-layer neural network.

**Algorithm 2:** Neural Network Approach

**Input:** Biased word embeddings ($\text{emb}_{\text{biased}}$), Bias Subspace (BiasSpace), Neutral word embeddings ($\text{emb}_{\text{neutral}}$)

**Output:** Debiased word embeddings

Initialize neural network $NN$ with input dimension as embedding dimension and output dimension as embedding dimension;

Pass $\text{emb}_{\text{biased}}$ through $NN$ to obtain transformed embeddings;

Compute $T^T$ as the matrix multiplication of the transpose of outputs of $NN$ and the outputs;

Compute norm1 as $\|(T^T \cdot T - I)\|_F$;

Compute norm2 as $\|\text{emb}_{\text{neutral}}^T \cdot T^T \cdot \text{BiasSpace}\|_F$;

Total loss is a weighted combination of norm1 and norm2;

Optimize $NN$ using an Adam optimizer;

Output normalized embeddings obtained after passing $\text{emb}_{\text{biased}}$ through $NN$;

Larger models, such as Llama-2 (Touvron et al., 2023), Alpaca (Taori et al., 2023) and Yi-6b (01.ai, 2024) need a more complex debiasing neural network. For embeddings with embedding length of around 2000, a two-layer neural network is sufficient, while for larger embedding dimensions, a three-layer neural network is required to achieve good debiasing results. In addressing the second research question, we delve into the intricacies of neural network complexity necessary for debiasing embeddings of varying sizes. While our discussion highlights the effectiveness of larger neural networks in mitigating bias within Language Model (LM) embeddings with substantial dimensions, it is imperative to substantiate this observation. We would like to point out that we draw inspiration from the conceptual framework of DeepSoftDebias. Building upon the foundational work by Bolukbasi et al., which employed a transformation matrix for word embedding debiasing, our approach replaces this matrix with a neural network. This neural network can be conceptualized as a series of interconnected matrices. Specifically, when de-biasing larger LMs with embedding dimensions exceeding 4096, we augment the neural network by increasing the number of layers and adjusting layer sizes.

This augmentation enables us to model the intricate dependencies inherent in debiasing processes for larger embedding dimensions. Consequently, deeper neural networks emerge as more efficacious tools for addressing bias in such expansive models. Additionally, the debiasing neural network and the optimization algorithm need to be hyperparameter-tuned, such as adjusting the learning rate, to get optimal results. The hyperparameters may vary depending on the model size, the embedding dimension, and the debiasing task. Therefore, a systematic search for the best hyperparameters is necessary to ensure the effectiveness of the debiasing process.

## 6 Results

In this section, we provide an extensive analysis of our proposed methodology, complete with a comprehensive evaluation against multiple metrics, tasks, and datasets. We provide the results of additional downstream testing and ablation experiments in Appendix D and Appendix F, respectively. We also provide our hypothesis of why there is a variation in bias across LLMs in Appendix E.

### 6.1 Mean Average Cosine Similarity

Mean Average Cosine Similarity (MAC) (Manzini et al., 2019) is a metric used to quantify semantic associations between word classes and attributes. MAC takes word embeddings, targets (representing classes), and attributes as inputs. By computing the mean cosine distance between target words and attribute sets, MAC offers a concise measure of semantic proximity. This metric provides valuable insights into the contextual semantics encoded within word embeddings. Table 1 shows that the word embeddings debiased in the direction of race and gender have comparable increases in their average MAC of **0.64**, whereas word embeddings debiased in the direction of religion have an increase in MAC of **0.61**. We see that our debiasing procedure categorically moves MAC scores closer to 1.0. This indicates an increase in cosine distance. Further, the associated P-values indicate these changes are statistically significant. This demonstrates that our approach for multiclass debiasing decreases bias in the word embeddings. We provide visual representations of the efficiency of *DeepSoftDebias* at removing gender bias, racial bias, and religion bias in Appendix B.

In our research, we have chosen to utilize Mean Average Cosine Similarity (MAC) as our primary metric for assessing bias in word embeddings. This decision is informed by the work of (Manzini et al., 2019), who posit that MAC can be viewed as an extension of the Word Embedding Association Test (WEAT), specifically adapted for a multiclass setting. The MAC and WEAT serve distinct, yet complementary purposes. While WEAT is designed to focus on specific associations between word vectors and predefined concepts (such as gender or race), MAC provides a broader perspective by measuring overall similarity patterns across different groups. This makes MAC less sensitive to specific word choices, thereby revealing biases that might be overlooked by WEAT. In essence, both metrics contribute to a comprehensive understanding of bias in word embeddings. However, the use of MAC is particularly beneficial in our research as it complements the findings of WEAT, providing a more holistic view of bias in the data. This approach allows us to capture a wider range of biases, thereby enhancing the robustness of our analysis.

### 6.2 Stereotype Score

Our research focuses on evaluating and mitigating stereotypical bias in NLI tasks using the Stereoset dataset. This dataset comprises pairs of sentences differing only in the substitution of words related to social groups like gender, race, or religion. The objective is to predict their relationship as same, entailment, or contradiction. We introduce a method aimed at reducing bias in word embeddings, with **Stereotype Score** $SS$ values closer to 50 indicating decreased bias. Table 2 presents *DeepSoftDebias*'s results alongside existing approaches on the Stereoset dataset. Notably, *DeepSoftDebias* achieves the lowest $SS$ across all social groups, demonstrating its effectiveness in bias reduction. Particularly impressive is *DeepSoftDebias*'s performance in the gender and race categories, where it significantly outperforms existing methods. For instance, with the SFR-Embedding-Mistral (Jiang et al., 2023) model, *DeepSoftDebias* achieves an $SS$ of 50 for gender and 50.409 for race using the Llama-2-7b model. Additionally, *DeepSoftDebias* attains a score of 51.282 for the Zephyr-7b-beta (Tunstall et al., 2023) or 48.717 for Alpaca-7b (Taori et al., 2023). We present these score in 2 an illustration of these scores in Fig. 6.

| Model | Variant | Topic | BMAC | NSMAC | SS | CMS | CSS | CAS |
|---|---|---|---|---|---|---|---|---|
| Yi | Yi-6B | | 0.148 | 0.964 | 55.372 | **49.620** | 58.970 | 37.250 |
| Alpaca | Alpaca-7B | | 0.612 | 0.816 | 53.306 | 48.850 | 57.690 | 37.250 |
| BAAI | bge-base-en-v1.5 | | 0.471 | **0.997** | **50.000** | 48.090 | 42.310 | 58.820 |
| BAAI | bge-large-en-v1.5 | | 0.404 | 0.983 | 49.174 | 50.380 | 50.640 | 51.960 |
| Zephyr | Zephyr-7B-beta | Gender | 0.393 | 0.981 | 52.893 | 46.950 | 59.620 | 29.410 |
| Mistral | e5-mistral-7b-instruct | | 0.343 | 0.971 | 52.893 | 48.090 | 55.770 | 38.240 |
| Llama 2 | Llama-2-7b-hf | | 0.182 | 0.964 | 48.347 | 44.660 | 57.690 | 26.470 |
| Salesforce | SFR-Embedding-Mistral | | 0.343 | 0.971 | **50.000** | 45.420 | 50.000 | 40.200 |
| Falcon | falcon-7b | | 0.011 | 0.964 | 51.240 | 48.850 | 60.900 | 32.350 |
| Gemma | Gemma-2b | | 0.058 | 0.971 | 47.107 | 48.470 | 57.69 | 36.27 |
| Gemma | Gemma-7b | | 0.553 | 0.976 | 49.173 | 51.53 | 63.46 | 35.29 |
| GritLM | GritLM-7B | | 0.379 | 0.999 | 51.239 | 48.470 | 57.05 | 37.25 |
| mxbai | mxbai-embed-large-v1 | | 0.467 | 0.994 | 51.652 | 55.34 | 60.9 | 49.02 |
| Yi | Yi-6B | | 0.111 | 0.964 | 46.209 | 64.150 | 66.170 | 53.660 |
| Alpaca | Alpaca-7B | | 0.655 | 0.938 | 52.357 | 41.280 | 41.540 | 46.340 |
| BAAI | bge-base-en-v1.5 | | 0.496 | 0.992 | **49.590** | 44.770 | 46.250 | 36.590 |
| BAAI | bge-large-en-v1.5 | | 0.404 | 0.990 | 50.922 | 40.890 | 40.690 | 51.220 |
| Zephyr | Zephyr-7B-beta | Race | 0.419 | 0.992 | 49.283 | 42.250 | 41.330 | 60.980 |
| Mistral | e5-mistral-7b-instruct | | 0.380 | **0.999** | 50.922 | 52.520 | 52.680 | 60.980 |
| Llama 2 | Llama-2-7b-hf | | 0.175 | 0.990 | **50.410** | 45.930 | 46.680 | 46.340 |
| Salesforce | SFR-Embedding-Mistral | | 0.381 | 0.994 | 51.639 | **49.030** | 50.750 | 39.020 |
| Falcon | falcon-7b | | 0.010 | 0.985 | 50.922 | 46.710 | 46.900 | 53.660 |
| Yi | Yi-6B | | 0.147 | 0.984 | 52.564 | 47.620 | 48.480 | 33.330 |
| Alpaca | Alpaca-7B | | 0.676 | 0.823 | 51.282 | 80.000 | 82.830 | 33.330 |
| BAAI | bge-base-en-v1.5 | | 0.497 | 0.990 | 46.154 | 59.050 | 61.620 | 16.670 |
| BAAI | bge-large-en-v1.5 | | 0.406 | 0.985 | **51.282** | 60.000 | 61.620 | 33.330 |
| Zephyr | Zephyr-7B-beta | Religion | 0.465 | 0.996 | **51.282** | 48.570 | 50.510 | 16.670 |
| Mistral | e5-mistral-7b-instruct | | 0.436 | 0.985 | 52.564 | 52.380 | 51.520 | 66.670 |
| Llama 2 | Llama-2-7b-hf | | 0.202 | 1.003 | 44.872 | 64.760 | 66.670 | 33.330 |
| Salesforce | SFR-Embedding-Mistral | | 0.437 | 0.988 | **51.282** | 40.950 | 39.390 | 66.670 |
| Falcon | falcon-7b | | 0.009 | **0.998** | **48.718** | **50.480** | 51.520 | 33.330 |

Table 1: Quantitative analysis for *DeepSoftDebias* using BiasedMAC (BMAC), New SoftMAC (NSMAC), StereotypeScore (SS), Crows Metric Score (CMS), Crows Stereotype Score (CSS), Crows Antistereotype Score (CAS). The best performance is highlighted in **bold**.

| Stereotype Score (SS) | | | |
|---|---|---|---|
| **Stereoset** | **Gender** | **Race** | **Religion** |
| FineDeb | <u>53.27</u> | <u>50.82</u> | **50.39** |
| CDA | 59.61 | 56.73 | 58.37 |
| INLP | 57.25 | 57.29 | 60.31 |
| Self-Debias | 59.34 | 54.30 | 57.26 |
| Sentence Debias | 59.37 | 57.78 | 58.73 |
| *DeepSoftDebias* | **50.00** | **50.41** | <u>51.28</u> |

Table 2: StereoSet evaluation. Closer to 50 is better for SS. The best performance is highlighted in **bold** while the next best is <u>underlined</u>).

## 6.3 Crows-Pairs Dataset

Our study evaluates social bias in natural language generation tasks using the CrowS Pairs dataset, comprising pairs of sentences differing in their degree of bias. By ranking these sentences according to bias level, we quantify the effectiveness of various methods in reducing bias in word embeddings. But as our work is based on word embeddings instead of getting the log-likelihood of the next token from the language model, we compute the average sentence vector for the common parts shared between two sentences. Next, we compare the similarity of this average sentence vector with the uncommon part (i.e., the modified tokens) using word embeddings. By doing so, we capture the semantic differences between stereotypical and non-stereotypical components within the sentence pairs. The rest of the metric remains the same.

Table 3 presents *DeepSoftDebias*'s results alongside existing approaches on the CrowS Pairs dataset. Notably, *DeepSoftDebias* achieves scores closest to 50 across all social groups, indicating a significant reduction in social bias. The metric used here is defined in Eq. (2) as follows:

$$\text{Metric score: } \frac{(\text{stereo\_score}+\text{antistereo\_score})\times 100}{N} \quad (2)$$

where **Crows Pair Stereotype Score (CSS)** is the number of stereotypical samples that agree with their label direction and **Crows Pairs Anti-stereotype Score (CAS)** is the number of anti-stereotypical samples that agree with their label direction. Label direction refers to the label given

| Crows Pairs Metric Score (CMS) | | | |
|---|---|---|---|
| **Crows Pairs Dataset** | **Gender** | **Race** | **Religion** |
| FineDeb | 54.58 | 65.24 | <u>44.76</u> |
| CDA | 56.11 | <u>56.70</u> | 60.00 |
| INLP | <u>51.15</u> | 67.96 | 60.95 |
| Self-Debias | 52.29 | <u>56.70</u> | 56.19 |
| Sentence Debias | 52.29 | 62.72 | 63.81 |
| *DeepSoftDebias* | **50.38** | **49.07** | **50.48** |

Table 3: Crows Pairs evaluation. Metric score for every demographic. Closer to 50 is better for the metric (**best**; <u>next best</u>).

the pair of sentences whether they are stereotypical or anti-stereotypical. In our evaluation we get the average sentence vector of the context and the more and less (anti-)stereotypical sentence. We then see whether the context vector is closer to the more (anti-)stereotypical sentence or the less (anti-)stereotypical sentence. If it is closer to the more (anti–)stereotypical sentence,, then we state that it agrees with the (anti–)stereotype, i.e., the label direction. Particularly noteworthy is *DeepSoftDebias*'s superior performance in the gender and religion categories. For instance, with the Yi-6B model, *DeepSoftDebias* achieves a score of 49.62 for gender and 50.48 for religion with the falcon-7b model. Similarly, using the SFR-Embedding-Mistral model, *DeepSoftDebias* achieves a score of 49.03 for race biasing the SFR-Embedding-Mistral model. These results underscore the effectiveness of *DeepSoftDebias* in mitigating social bias in word embeddings. We present these score in 3 and depict the variation of these scores in Fig. 8.

We also report the CSS and CAS score which refer to the CrossNER Stereotype score, i.e., the number of times the model agrees with the more stereotypes statement when the label direction is stereotype, and the CrossNER Anti-stereotype score, which refers to the number of times the model agrees with the more anti-stereotyped statement when the label direction was anti-sterotype.

## 7 Discussion

In this section, we summarise the answers to our research questions.

**RQ1** We find that *DeepSoftDebias outperforms state-of-the-art methods, and does so without negatively affecting downstream task performance.* We make this conclusion after exhaustive testing on several models, and datasets and evaluating several metrics.

**RQ2** We find that *size and complexity do affect the ability of debiasing models.* Specifically, we make the following observations about *DeepSoftDebias*:

- A single layer neural network can effectively de-bias embeddings with dim $\leq 1024$.
- A two-layer neural network can effectively debias embeddings with dim $\leq 2048$.
- A two-layer neural network with an increased layer size can effectively de-bias embeddings with dim $\leq 4096$.
- A three-layer neural network can effectively debias embeddings with dim $\leq 4450$.

As a step for future work, we are curious to investigate scaling patterns to a further extent.A visualization of this is provided in Fig 11

**RQ3** While debiasing techniques in general can affect the downstream performance of models, we test *DeepSoftDebias* on multiple challenging downstream tasks and report that *our proposed approach, to a large extent, does not negatively influence the performance of different downstream tasks. Remarkably, we see an improvement when using our debiased embeddings for some downstream tasks.*

**RQ4** We find that while *DeepSoftDebias* is *effective at reducing bias across gender, race, and religion.* We conclude this after testing on multiple embeddings, and multiple datasets and evaluating on multiple performance metrics. As a step for future work, we are curious to investigate whether our proposed approach works towards other forms of bias as well.

## 8 Conclusion

In this paper, we propose *DeepSoftDebias*, an approach that leverages neural networks to reduce bias in large language model embeddings. We perform an exhaustive series of tests using multiple performance metrics, state-of-the-art datasets, and downstream tasks to ensure that our debiasing technique is robust, efficient, and accurate. In the future, it would be interesting to see how this method translates to multilingual datasets since bias is language and culture-specific. We hope that this research paves the way for future endeavors that look to make LLMs fair, ethical, and bias-free.

## 9 Limitations

While we do perform exhaustive analysis to test our proposed methodology, our study is monolingual and covers datasets only in English. Consequently, our downstream tasks are also tested only in English. Further, we were unable to conduct test on API-based models at this time. Our testing was also constrained by the limitations of GPU VRAM, which prevented us from extending our testing to larger models such as Llama-65B. These models could not be accommodated within the GPU VRAM, even after applying quantization to 8 bits. Consequently, the largest model that we were able to test was the Gemma-7B model.

## 10 Ethics Statement

We understand that bias can be defined in various ways, and it's not necessarily ideal for a language model to treat all users exactly the same without considering demographics. There are situations where certain topics require careful handling to avoid perpetuating harmful stereotypes against marginalized communities. Using specific bias metrics might suggest they encompass all negative social impacts across different groups, but we recognize that existing metrics may not capture all nuances in treatment across demographics. Therefore, any benchmark for bias needs to continually evolve to better understand and address these issues as they affect different communities.

The definitions of morality and bias are shaped by cultural perspectives, resulting in diverse interpretations among individuals. Consequently, we do not claim that this work provides an objective or exhaustive measure of any of these concepts.

## References


01.ai. 2024. Yi. 2024.

R Baligudam. 2022. A systematic study of gender and religion bias in stories. Master's thesis, University of Twente.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29.

Shikha Bordia and Samuel R Bowman. 2019. Identifying and reducing gender bias in word-level language models. *arXiv preprint arXiv:1904.03035*.

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

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM.

Daniel de Vassimon Manela, David Errington, Thomas Fisher, Boris van Breugel, and Pasquale Minervini. 2021. Stereotype and skew: Quantifying gender bias in pre-trained and fine-tuned language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2232–2242.

Sunipa Dev, Tao Li, Jeff M Phillips, and Vivek Srikumar. 2020. On measuring and mitigating biased inferences of word embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7659–7666.

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

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *arXiv preprint arXiv:1903.03862*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Hannah Rose Kirk, Yennie Jun, Filippo Volpin, Haider Iqbal, Elias Benussi, Frederic Dreyer, Aleksandar Shtedritski, and Yuki Asano. 2021. Bias out-of-the-box: An empirical analysis of intersectional occupational biases in popular generative language models. *Advances in neural information processing systems*, 34:2611–2624.

Hadas Kotek, Rikker Dockum, and David Sun. 2023. Gender bias and stereotypes in large language models. In *Proceedings of The ACM Collective Intelligence Conference*, pages 12–24.

Paul Pu Liang, Irene Mengze Li, Emily Zheng, Yao Chong Lim, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2020. Towards debiasing sentence representations.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2020. Crossner: Evaluating cross-domain named entity recognition.

Thomas Manzini, Yao Chong Lim, Yulia Tsvetkov, and Alan W Black. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2020. Hate speech detection and racial bias mitigation in social media based on bert model. *PloS one*, 15(8):e0237861.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. Crows-pairs: A challenge dataset for measuring social biases in masked language models.

Jesutofunmi A Omiye, Jenna C Lester, Simon Spichak, Veronica Rotemberg, and Roxana Daneshjou. 2023. Large language models propagate race-based medicine. *NPJ Digital Medicine*, 6(1):195.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Ella Rabinovich, Yulia Tsvetkov, and Shuly Wintner. 2018. Native language cognate effects on second language lexical choice. *Transactions of the Association for Computational Linguistics*, 6:329–342.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection.

Herbert Robbins and Sutton Monro. 1951. Stochastic approximation and recursive algorithms and applications. *Biometrika*, 37(1/2):62–79.

Akash Saravanan, Dhruv Mullick, Habibur Rahman, and Nidhi Hegde. 2023. Finedeb: A debiasing framework for language models.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp.

Fuliang Tang, Kunguang Wu, Zhendong Guo, Shuaishuai Huang, Yingtian Mei, Yuxing Wang, Zeyu Yang, and Shiming Gong. Large language model (llm) racial bias evaluation.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Nishant Yadav, Mahbubul Alam, Ahmed Farahat, Dipanjan Ghosh, Chetan Gupta, and Auroop R. Ganguly. 2023. Cda: Contrastive-adversarial domain adaptation.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020.

Xlnet: Generalized autoregressive pretraining for language understanding.

İlhan Tarımer, Adil Çoban, and Arif Emre Kocaman. 2019. Sentiment analysis on imdb movie comments and twitter data by machine learning and vector space techniques.

## Frequently Asked Questions (FAQs)

1. **Is this method effective at removing all kinds of bias?**
   We acknowledge that bias has multiple forms that vary by different social factors, language, culture, and various other factors. We evaluated *DeepSoftDebias* on gender bias, racial bias, and religious bias and it has proved effective at reducing all of them. We hope that in the future, this method will prove effective in reducing other kinds of biases as well.

2. **Why isn't GPT analyzed in this paper?**
   Given that GPT is an API-based model, we were unable to test it at this time. We hope that one day, this method can be tested even on API-based LLMs.

3. **Is the proposed approach open-sourced?**
   Yes, we plan to make all our code available on a GitHub repository.

# Appendix

This section provides supplementary material in the form of additional examples, implementation details, etc. to bolster the reader's understanding of the concepts presented in this work.

## A Table of words and bias they represent

| Bias Direction | | Biased Words |
|---|---|---|
| Gender | Male | "manager", "executive", "doctor", "lawyer", "programmer", "scientist", "soldier", "supervisor", "rancher", "janitor", "firefighter", "officer" |
| | Female | "secretary", "nurse", "clerk", "artist", "homemaker", "dancer", "singer", "librarian", "maid", "hairdresser", "stylist", "receptionist", "counselor" |
| Race | Black | "slave", "musician", "runner", "criminal", "homeless" |
| | Asian | "manager", "executive", "redneck", "hillbilly", "leader", "farmer" |
| | Caucasian | "doctor", "engineer", "laborer", "teacher" |
| Religion | Jew | "greedy", "cheap", "hairy", "liberal" |
| | Christian | "judgemental", "conservative", "familial" |
| | Muslim | "violent", "terrorist", "dirty", "uneducated" |

Table 4: List of Words related to sub-categories in the bias directions explored

## B MAC Scores of *DeepSoftDebias*

Figures 3, 4, and 5 illustrate how *DeepSoftDebias* reduces bias in LLM embeddings.

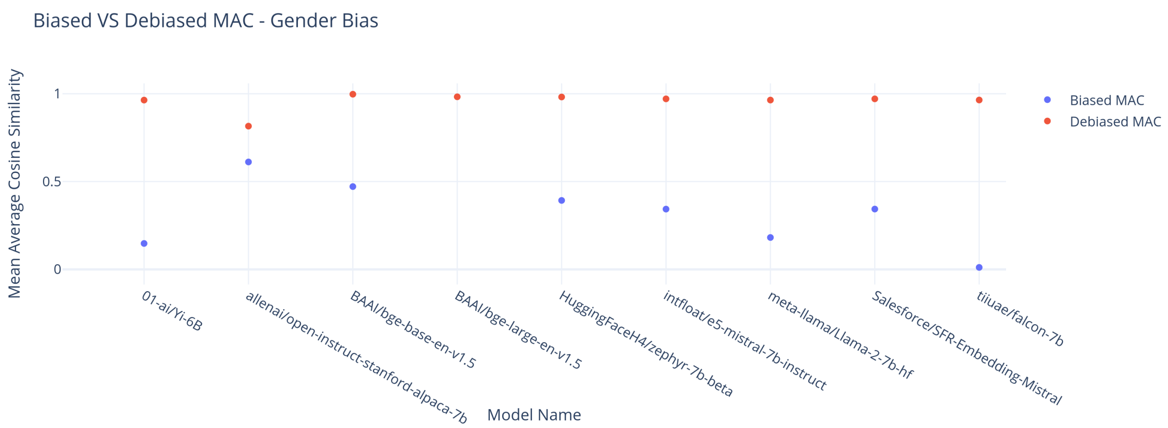


Figure 3: A visual representation of how *DeepSoftDebias* reduces gender bias in large language model embeddings.

## C Stereoset Scores of *DeepSoftDebias*

Figures 6 and 8 provide an illustration of word vectors debiased using *DeepSoftDebias* and their stereoset scores and Crows Metric scores respectively.

## D Downstream Testing Results

In our research, we primarily focus on the debiasing of word embeddings derived from Language Learning Models (LLMs). We aim to investigate the impact of this debiasing on the performance of these embeddings when subjected to identical training and testing methodologies. Our objective

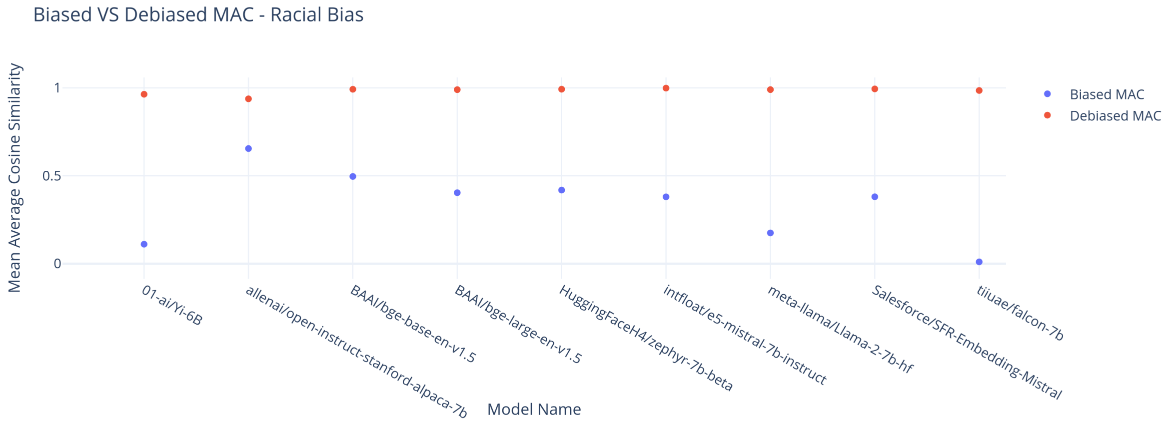


Figure 4: A visual representation of how *DeepSoftDebias* reduces racial bias in large language model embeddings.

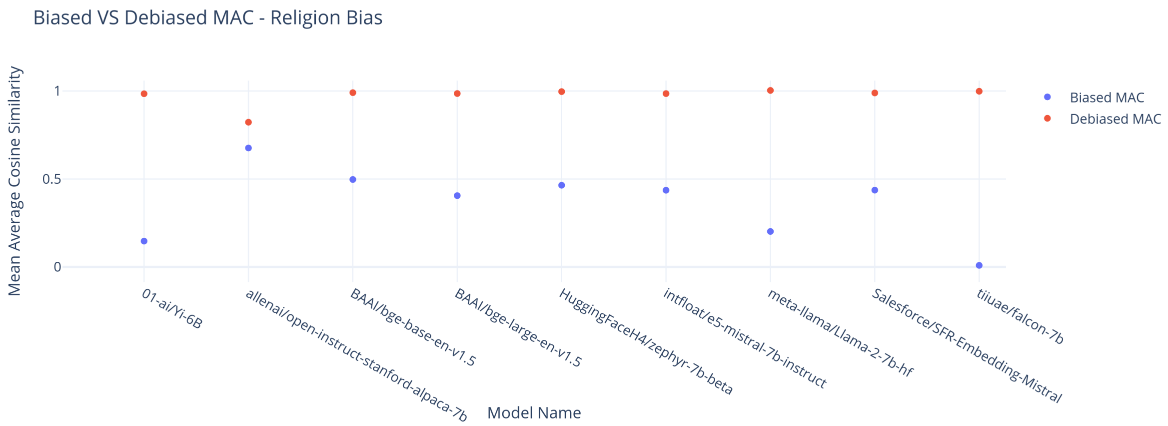


Figure 5: A visual representation of how *DeepSoftDebias* reduces religion bias in large language model embeddings.

| **Model** | **Variant** | **Baseline Debiased Text Class. Acc.** | **DSB Debiased Text Class. Acc.** | **Baseline Debiased NER Macro F1 Avg.** | **DSB Debiased NER Macro F1 Avg.** |
|---|---:|---:|---:|---:|---:|
| Gemma | gemma-2b | 0.7655 | 0.7964 | 0.469 | 0.484 |
| BAAI | bge-base-en-v1.5 | 0.8296 | 0.822 | 0.458 | 0.421 |
| Mistral | SFR-Embedding-Mistral | 0.8297 | 0.821 | 0.404 | 0.428 |
| Gemma | gemma-7b | 0.516 | 0.8032 | 0.198 | 0.475 |
| Zephyr | zephyr-7b-beta | 0.7997 | 0.81 | 0.403 | 0.429 |
| mxbai | mxbai-embed-large-v1 | 0.8366 | 0.7903 | 0.461 | 0.455 |

Table 5: Downstream testing results comparison with embeddings debiased using *DeepSoftDebias* and the baseline SoftDebais Method. The first two columns represent results for downstream performance on sentiment analysis. The second two columns represent results for downstream performance on NER.

is to quantitatively measure any performance fluctuations (increase or decrease) on the downstream tasks that we test. For this purpose, we trained simple models on top of these word embeddings. For instance, we used an XGBoost model without any hyperparameter tuning for the classification task, and a straightforward bidirectional LSTM for the Named Entity Recognition (NER) task. It is important to note that our goal in presenting our results on these two tasks is not to establish a benchmark for debiased embeddings. Instead, we aim to demonstrate the effect of debiasing on the performance of word embeddings in downstream tasks, as seen in the seminal work of (Gonen and Goldberg, 2019). This approach allows us to provide a more comprehensive understanding of the implications and potential benefits of debiasing word embeddings.

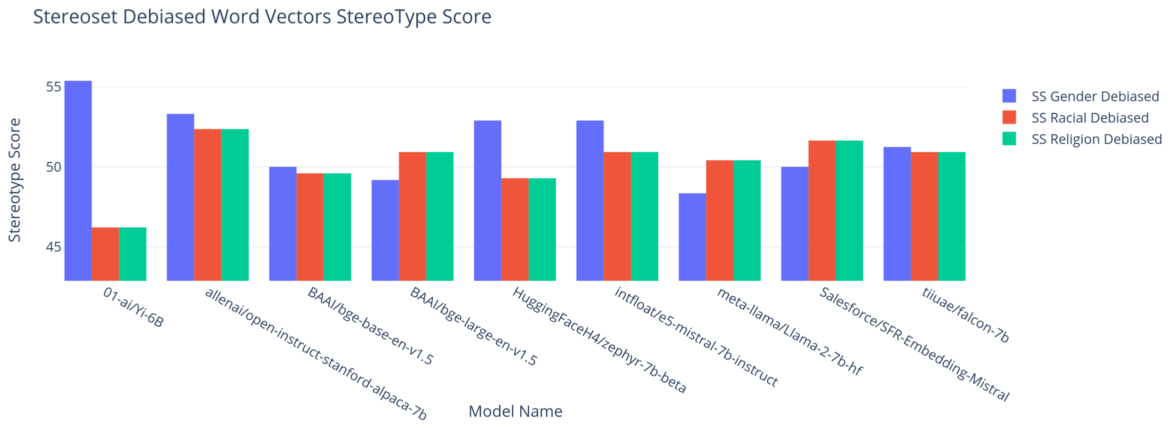


Figure 6: A visual representation of word vectors debiased using *DeepSoftDebias* and their stereotype scores across gender, race and religion respectively.

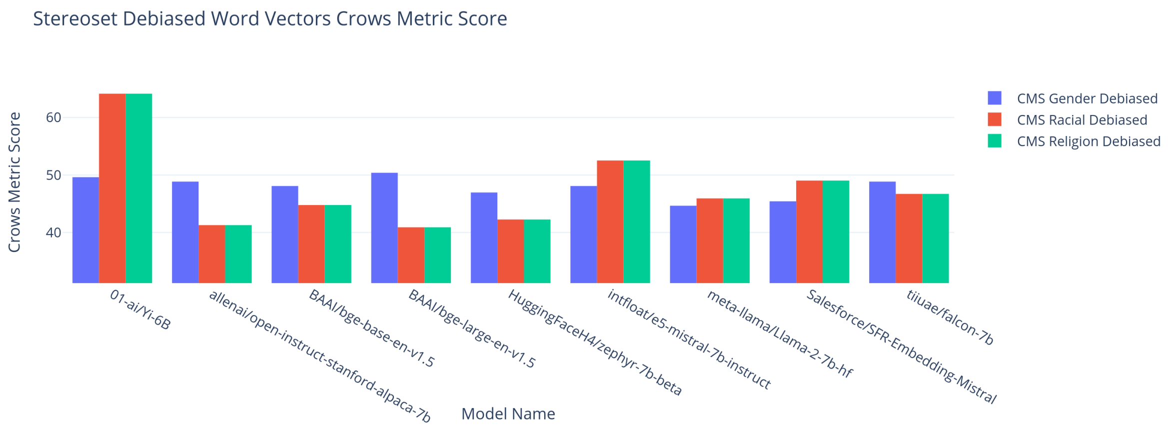


Figure 7: A visual representation of word vectors debiased using *DeepSoftDebias* and their Crows Metric score across gender, race and religion respectively.

### D.1 Sentiment Classification

In our study, we employ downstream testing to assess the utility of embeddings debiased using *DeepSoft-Debias* across two key natural language processing tasks: text classification and named entity recognition (NER). Utilizing the IMDB Sentiment Classification dataset (İlhan Tarımer et al., 2019) and Stanford TreeBank Dataset for text classification, featuring labeled movie reviews as positive or negative, we compute the average sentence vectors using both original and debiased embeddings. Training XGBoost (Chen and Guestrin, 2016) classifiers on these vectors, we compare their accuracy on the test set, recognizing accuracy as a straightforward metric for binary classification tasks like sentiment analysis. Notably, our results reveal a performance improvement when debiasing in the gender and religion directions, whereas a slight decrease in performance is observed in the case of race debiasing. We provide these results in Table:6 for IMDB Sentiment classification and Table:7 for Stanford Sentiment Treebank. A visual representation of these results in Fig. 9.

### D.2 Named Entity Recognition (NER)

In our research, we examine the performance of debiased embeddings in the domain of named entity recognition (NER) using the Reuters subset of the CrossNER (Liu et al., 2020) dataset. This dataset comprises news domain sentences annotated with four entity types: person, location, organization, and

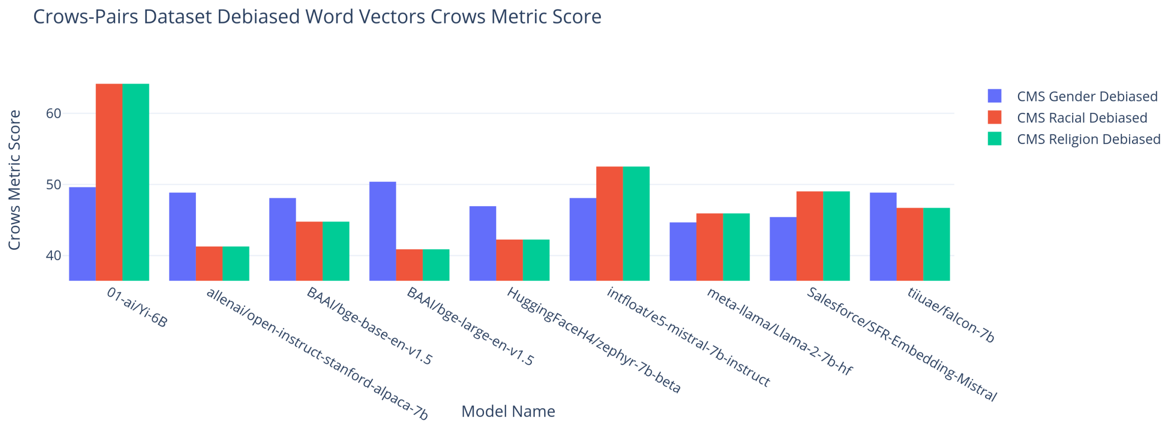


Figure 8: A visual representation of word vectors debiased using *DeepSoftDebias* and their Crows Metric scores across gender, race and religion respectively.

product. Employing a simple BiLSTM model, we input padded arrays of embeddings for each sentence and trained the model on the dataset. We evaluate the models' performance on the test set using the macro-averaged F1-score, a metric that balances precision and recall, crucial for accurate entity identification and classification. To mitigate potential bias towards more frequent entity types, we adopt macro-averaging, allotting equal importance to each entity type. Remarkably, our findings indicate a slight performance boost when using debiased embeddings in all three directions compared to biased embeddings. We provide these results in Table:6and a visual representation of these results in Fig. 10.

### D.3 Semantic Textual Similarity

In our research, we evaluate the performance of debiased embeddings for the Semantic Textual Similarity (STS) task using the STS-B dataset. This dataset, a component of the General Language Understanding Evaluation (GLUE) benchmark, is a valuable resource for the STS task. The task aims to quantify the semantic similarity between two sentences, assigning a score from 1 to 5 based on their degree of semantic equivalence. The STS-B dataset, comprising examples from diverse sources, includes human annotations for sentence pair similarity, contributing significantly to the broader field of natural language understanding by facilitating the measurement of meaning equivalence across sentences. To utilize the embeddings for the task, we train a dual-head neural network. We perform cosine similarity after passing the average sentence vector of the two sentences through the network, followed by a Fully Connected layer to obtain the actual score. The performance of our approach is evaluated using Pearson's correlation and Spearman's correlation as metrics. This methodology allows us to develop and evaluate models' ability to understand nuanced semantic relationships in text effectively. We provide our results in this task in Table:7

Figures 9 and 10 present an illustration of the results of various downstream tasks and their performance evaluation.

## E Variation of Bias in the Different LLMs

The presence of biases in has drawn significant attention from researchers and practitioners. These biases can inadvertently emerge during the training process due to the characteristics of the initial training data. In this study, we explore the factors contributing to bias variation among LLMs, focusing on three prominent models: Llama, Mistral, and Gemma. Our analysis reveals that biases, including those related to gender, race, and culture, are often inherited from the training data. For instance, historical texts may perpetuate gender stereotypes or racial prejudices present in their source material. Llama and Mistral, trained on diverse corpora containing web documents, source code, and mathematical text, exhibit varying degrees of bias. Gemma, released by Google, further demonstrates the impact of training data size, with

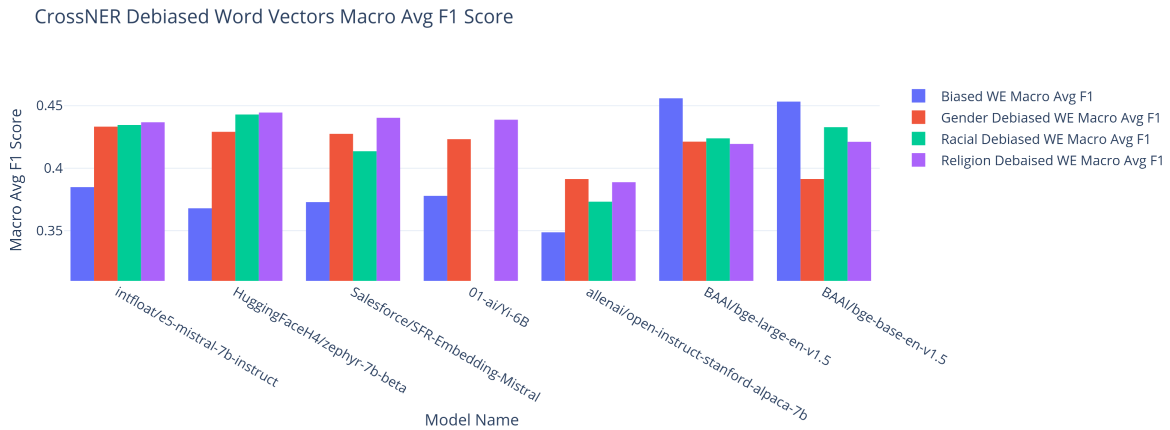


Figure 9: An illustration of the results of downstream testing on NER. We compare the performance of biased and debaised embeddings in the directions of gender, race, and religion respectively.

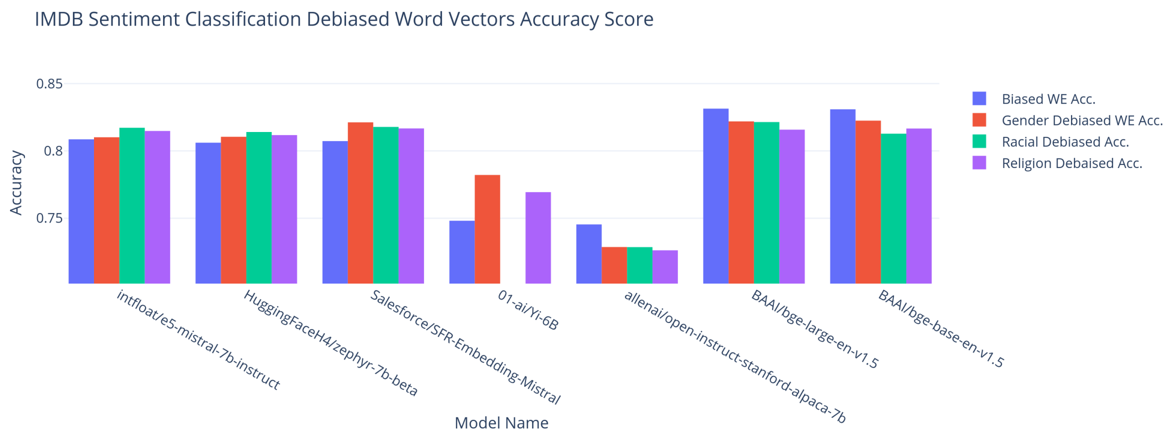


Figure 10: An illustration of results of downstream testing on sentiment analysis. We compare the performance of biased and debaised embeddings in the directions of gender, race, and religion respectively.

both 2B and 7B variants drawing from an extensive pool of up to 6 trillion tokens.

## F Ablation Experiments

In our study, we conduct ablation experiments to assess the effectiveness of various debiasing techniques in the realm of natural language processing. These techniques encompassed five distinct scenarios: the utilization of debiased embeddings, the application of the original soft debiasing method, the original debiasing method with the Adam optimizer, *DeepSoftDebias* with the SGD optimizer, and finally, *DeepSoftDebias* with the Adam optimizer. These experiments were gauged based on MAC as the evaluation metric.

Through rigorous experimentation across three biasing directions, we systematically analyze the performance of each method. Our results reveal a consistent trend of incremental improvements as we transitioned from one method to the next. Notably, *DeepSoftDebias*, emerged as the standout performer, boasting the highest mean average cosine similarity score across all evaluated scenarios. In addition, our analysis revealed that substituting the transformation matrix with our neural network approach resulted in the most significant enhancement in the efficacy of the debiasing method. This observation underscores the pivotal role played by neural networks in maximizing the effectiveness of the debiasing techniques. Table 8 presents a visualization of the results of our ablation experiments.

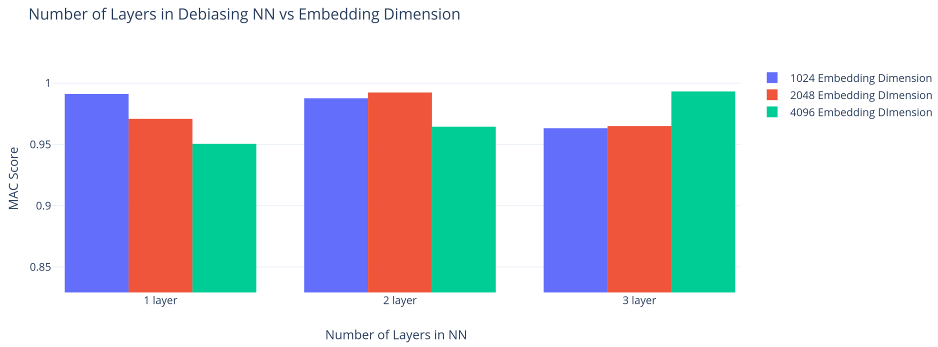


Figure 11: An illustration analysis of number of layers in debiasing neural network vs. embedding dimension. We can see the varying performance of the 3 different sizes according to the embedding dimension of the LM it is used with.

| Model | Variant | Bias Direction | Biased Text Class. Acc. | Debiased Text Class. Acc. | Biased NER Macro F1 Avg. | Debiased NER Macro F1 Avg. |
|---|---|---|---|---|---|---|
| Mistral | e5-mistral-7b-instruct | | 0.809 | 0.810 | 0.385 | 0.433 |
| Zephyr | Zephyr-7B-beta | | 0.806 | 0.810 | 0.368 | **0.429** |
| Salesforce | SFR-Embedding-Mistral | | 0.807 | 0.821 | 0.373 | 0.428 |
| Yi | Yi-6B | Gender | 0.748 | **0.782** | 0.378 | 0.423 |
| Alpaca | Alpaca-7B | | 0.745 | 0.729 | 0.349 | 0.391 |
| BAAI | bge-large-en-v1.5 | | 0.831 | 0.822 | 0.456 | 0.421 |
| BAAI | bge-base-en-v1.5 | | 0.831 | 0.822 | 0.453 | 0.392 |
| Mistral | e5-mistral-7b-instruct | | 0.811 | 0.817 | 0.362 | **0.435** |
| Zephyr | Zephyr-7B-beta | | 0.806 | **0.814** | 0.393 | 0.443 |
| Salesforce | SFR-Embedding-Mistral | | 0.810 | 0.818 | 0.382 | 0.413 |
| Yi | Yi-6B | Race | 0.748 | 0.782 | 0.378 | 0.423 |
| Alpaca | Alpaca-7B | | 0.751 | 0.729 | 0.358 | 0.373 |
| BAAI | bge-large-en-v1.5 | | 0.832 | 0.821 | 0.473 | 0.424 |
| BAAI | bge-base-en-v1.5 | | 0.830 | 0.813 | 0.457 | 0.433 |
| Mistral | e5-mistral-7b-instruct | | 0.808 | 0.815 | 0.385 | 0.437 |
| Zephyr | Zephyr-7B-beta | | 0.805 | 0.812 | 0.389 | **0.444** |
| Salesforce | SFR-Embedding-Mistral | | 0.808 | 0.817 | 0.389 | 0.440 |
| Yi | Yi-6B | Religion | 0.750 | **0.769** | 0.384 | 0.439 |
| Alpaca | Alpaca-7B | | 0.751 | 0.726 | 0.364 | 0.389 |
| BAAI | bge-large-en-v1.5 | | 0.830 | 0.816 | 0.457 | 0.419 |
| BAAI | bge-base-en-v1.5 | | 0.830 | 0.817 | 0.459 | 0.421 |

Table 6: Downstream testing results with embeddings debiased using *DeepSoftDebias*. The first two columns represent results for downstream performance on sentiment analysis. The second two columns represent results for downstream performance on NER. The best performance is highlighted in **bold**.

This empirical evidence underscores the robustness and efficacy of our proposed approach in mitigating bias within natural language processing systems. By combining state-of-the-art debiasing techniques with advanced optimization strategies, we have unlocked a powerful methodological framework for enhancing the fairness and accuracy of language models.

| Model Name | Topic | STS-B ↑ Baseline Debiased PCC | STS-B ↑ *DeepSoftDebias* Debiased PCC | SST Biased Acc. | SST Baseline Debiased Acc. | SST *DeepSoftDebias* Debiased Acc. |
|---|---|---|---|---|---|---|
| BAAI/bge-base-en-v1.5 | Gender | 0.088 | 0.001 | 0.730 | 0.725 | 0.693 |
| BAAI/bge-large-en-v1.5 | | 0.159 | 0.105 | 0.727 | 0.710 | 0.705 |
| google/gemma-2b | | -0.060 | 0.154 | 0.686 | 0.677 | 0.678 |
| google/gemma-7b | | -0.059 | 0.017 | 0.675 | 0.544 | 0.691 |
| GritLM/GritLM-7B | | -0.125 | 0.044 | 0.711 | 0.702 | 0.697 |
| HuggingFaceH4/zephyr-7b-beta | | -0.129 | 0.097 | 0.706 | 0.687 | 0.699 |
| intfloat/multilingual-e5-large-instruct | | -0.037 | 0.096 | 0.729 | 0.720 | 0.724 |
| meta-llama/Llama-2-7b-hf | | 0.009 | -0.032 | 0.701 | 0.692 | 0.686 |
| openai-community/gpt2-large | | 0.042 | -0.038 | 0.664 | 0.665 | 0.669 |
| openai-community/gpt2-xl | | 0.041 | 0.071 | 0.666 | 0.667 | 0.669 |
| tiiuae/falcon-7b | | -0.116 | 0.066 | 0.686 | 0.672 | 0.694 |
| BAAI/bge-base-en-v1.5 | Race | 0.094 | 0.092 | 0.730 | 0.709 | 0.683 |
| BAAI/bge-large-en-v1.5 | | 0.104 | 0.099 | 0.727 | 0.727 | 0.695 |
| google/gemma-2b | | -0.041 | 0.164 | 0.686 | 0.665 | 0.686 |
| google/gemma-7b | | -0.055 | 0.133 | 0.675 | 0.549 | 0.678 |
| GritLM/GritLM-7B | | -0.133 | -0.057 | 0.711 | 0.714 | 0.690 |
| HuggingFaceH4/zephyr-7b-beta | | -0.127 | 0.062 | 0.706 | 0.687 | 0.697 |
| intfloat/multilingual-e5-large-instruct | | 0.053 | 0.120 | 0.729 | 0.730 | 0.730 |
| meta-llama/Llama-2-7b-hf | | -0.058 | 0.113 | 0.701 | 0.699 | 0.705 |
| openai-community/gpt2-large | | -0.019 | 0.024 | 0.664 | 0.670 | 0.680 |
| openai-community/gpt2-xl | | 0.149 | 0.180 | 0.666 | 0.665 | 0.692 |
| tiiuae/falcon-7b | | -0.192 | -0.027 | 0.686 | 0.664 | 0.693 |
| BAAI/bge-base-en-v1.5 | Religion | 0.054 | 0.078 | 0.730 | 0.716 | 0.694 |
| BAAI/bge-large-en-v1.5 | | 0.153 | 0.175 | 0.727 | 0.718 | 0.697 |
| google/gemma-2b | | 0.118 | 0.278 | 0.686 | 0.679 | 0.682 |
| google/gemma-7b | | 0.127 | 0.194 | 0.675 | 0.548 | 0.685 |
| GritLM/GritLM-7B | | -0.002 | 0.077 | 0.711 | 0.702 | 0.703 |
| HuggingFaceH4/zephyr-7b-beta | | -0.130 | 0.118 | 0.706 | 0.693 | 0.686 |
| intfloat/multilingual-e5-large-instruct | | 0.201 | 0.194 | 0.729 | 0.728 | 0.735 |
| meta-llama/Llama-2-7b-hf | | -0.103 | 0.032 | 0.701 | 0.679 | 0.710 |
| openai-community/gpt2-xl | | 0.247 | 0.251 | 0.666 | 0.671 | 0.679 |
| tiiuae/falcon-7b | | 0.126 | 0.265 | 0.686 | 0.671 | 0.703 |

Table 7: Downstream testing results on Stanford Sentiment Treebank and STS-B Semantic Similarity Dataset. PCC here refers to the Pearson's Coefficient and we report the gain in positive PCC from the Biased embeddings to the debiased embeddings. SST is Stanford Sentiment Treebank and STS-B is the Semantic Textual Similarity Benchmark

| Debiasing Direction | Biased | Baseline | Baseline + Adam | DeepSoftBias + SGD | DeepSoftBias + Adam |
|---|---|---|---|---|---|
| **Gender** | 0.390 | 0.623 | 0.799 | 0.893 | 0.982 |
| **Race** | 0.404 | 0.656 | 0.824 | 0.984 | 0.987 |
| **Religion** | 0.406 | 0.623 | 0.812 | 0.966 | 0.983 |

Table 8: Ablations to characterize various design decisions in the development of *DeepSoftDebias*. We start with the transformation matrix, then make incremental additions till we reach the proposed architecture of the *DeepSoftDebias* network.