Data-Centric Explainable Debiasing for Improving Fairness in Pre-trained Language Models

Yingji Li[1], Mengnan Du[2], Rui Song[1], Xin Wang[3], Ying Wang[1,4]*
[1]College of Computer Science and Technology, Jilin University, Changchun, China
[2]Department of Data Science, New Jersey Institute of Technology, Newark, USA
[3]School of Artificial Intelligence, Jilin University, Changchun, China
[4]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, China
{yingji21,songrui20}@mails.jlu.edu.cn, mengnan.du@njit.edu, {xinwang,wangying2010}@jlu.edu.cn

Abstract

Human-like social bias of pre-trained language models (PLMs) on downstream tasks have attracted increasing attention. The potential flaws in the training data are the main factor that causes unfairness in PLMs. Existing data-centric debiasing strategies mainly leverage explicit bias words (defined as sensitive attribute words specific to demographic groups) for counterfactual data augmentation to balance the training data. However, they lack consideration of implicit bias words potentially associated with explicit bias words in complex distribution data, which indirectly harms the fairness of PLMs. To this end, we propose a **Data**-Centric **Debias**ing method (named Data-Debias), which uses an explainability method to search for implicit bias words to assist in debiasing PLMs. Specifically, we compute the feature attributions of all tokens using the Integrated Gradients method, and then treat the tokens that have a large impact on the model's decision as implicit bias words. To make the search results more precise, we iteratively train a biased model to amplify the bias with each iteration. Finally, we use the implicit bias words searched in the last iteration to assist in debiasing PLMs. Extensive experimental results on multiple PLMs debiasing on three different classification tasks demonstrate that Data-Debias achieves state-of-the-art debiasing performance and strong generalization while maintaining predictive abilities.

1 Introduction

In recent years, pre-trained language models (PLMs), such as the BERT series (Devlin et al., 2019; Sanh et al., 2019; Liu et al., 2019) and GPT series (Radford et al., 2018, 2019; Brown et al., 2020) have seen vigorous development in natural language processing. By training on massive text datasets, PLMs can acquire rich linguistic knowledge. However, these training datasets contain human-like social biases and stereotypes (Zhao et al., 2019). PLMs can learn and amplify biases against certain demographic groups, leading to unfair decisions that harm vulnerable groups Therefore, mitigating social biases in PLMs is a critical issue and necessary to improve the fairness of natural language systems.

Current debiasing methods usually take the model-centric strategies (Liang et al., 2020; Cheng et al., 2021), seeking to improve fairness by modifying model architectures or adding regularization during training without considering the defects of the training data (Han et al., 2021). However, they ignore critical issues of training data, including real-world stereotypes, discrimination against vulnerable groups, and imbalanced samples across demographic groups. Data quality issues fundamentally enable language models to learn biased semantics. Furthermore, some methods are difficult to generalize to large-scale PLMs due to the limitations of the fine-tuning model.

In contrast to the model-centric strategies, the data-centric debiasing strategies focus on addressing defects in training data to improve data quality (Zha et al., 2023). The most general approach utilizes counterfactual data augmentation (Lu et al., 2020) to alleviate the imbalance of samples from different demographic groups. These debiasing methods generate synonymous sentences based on priori sensitive attribute words specific to different demographic groups (e.g., *male/female*, *white/black*). Although they consider sensitive attribute words that directly cause model biases, they ignore potentially harmful associations present in the training data that indirectly cause unfairness. We refer to sensitive attribute words specific to demographic groups as *explicit bias words*, and to tokens in the training data that have potentially harmful associations with explicit bias words as *implicit bias words* (illustrative examples are given in Figure 2). We formulate the following hypoth-

* Corresponding author

Findings of the Association for Computational Linguistics: ACL 2024, pages 3773–3786
August 11-16, 2024

esis: **Under the indirect effect of implicit bias words, PLMs capture the spurious associations between demographic groups and certain class labels to make unfair and biased decisions.**

Based on the above assumptions, in this work, we propose a **Data**-Centric **Debias**ing method (named Data-Debias), which uses an explainability method to search for implicit bias words to assist debiasing PLMs. Specifically, we compute the feature attribution scores for all tokens in the training data using Integrated Gradients (Sundararajan et al., 2017) which is a post-hoc explainability method. Implicit bias words are selected as tokens with large differences in feature attribution scores across demographic groups, and then used to assist debiasing. To more precisely target implicit bias words that are strongly associated with explicit bias words, we iteratively train a bias-amplified model. Within each iteration, we train the model with the implicit bias words to amplify the model's bias, and then re-search and update the implicit bias words with the biased model. After the iteration, the last updated more biased implicit bias words are used to assist debiasing to improve the fairness of PLMs.

Our main contributions are summarized as follows: 1) We propose a data-centric debiasing framework, which mines implicit bias words that have potentially harmful associations with explicit bias words, to improve the quality of the training data more comprehensively for stronger debiasing effects. 2) The obtained implicit bias words are interpretable and precise, ensured by searching via explainability methods and iteratively training the bias-amplified model. 3) Experiments on classification tasks on multiple PLMs demonstrate that Data-Debias outperforms state-of-the-art baseline models in debiasing while preserving predictive abilities. 4) Debiasing experiments on large-scale language models under the prompting paradigm verify the generalization of implicit bias words.

2 Methodology

In this section, we introduce the proposed Data-Debias debiasing framework, as shown in Figure 1. Data-Debias has three stages: 1) Searching for implicit bias words via an explainability method. 2) Iterative training of a bias-amplified model using implicit bias words. 3) Debiasing training using the final implicit bias words.

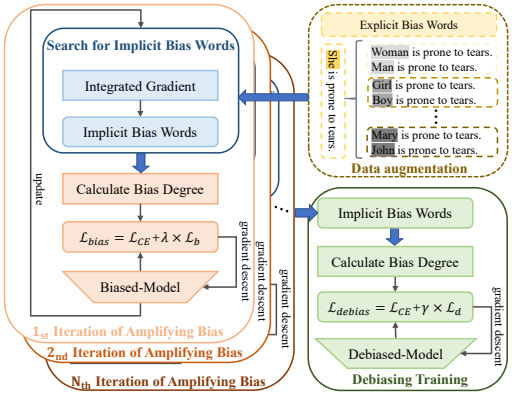

Figure 1: The Framework of Data-Debias. After data augmentation based on explicit bias words, it searches for implicit bias words via an explainability method, and then iteratively trains a biased model and updates the implicit bias words. The implicit bias words updated in the last iteration will be used to assist debiasing.

2.1 Search for Implicit Bias Words via Explainability

In this stage, we aim to mine implicit bias words that have potentially harmful associations with the explicit bias words. The motivation is that in the model's decision process, if a token has inconsistent attention among the sample pairs containing the explicit bias words, there is a harmful association between it and the explicit bias words that indirectly causes the model to make an unfair decision. To this end, we first perform data augmentation based on explicit bias words to generate sample pairs specific to demographic groups. We then apply the explainability method proposed by Slack et al. (2021) to the augmented sample pairs to search for tokens with inconsistent attention as implicit bias words.

For a certain social bias, we augment the dataset with prior explicit bias words. We focus on gender bias in this paper[1]. Following previous work (Bolukbasi et al., 2016; Liang et al., 2020; Cheng et al., 2021), we set the gender-specific list of explicit bias word pairs as: {FEMALE, MALE}={(woman, man), (girl, boy), (female, male), (she, he), (mother, father), (daughter, son), (gal, guy), (her, his), (herself, himself), (Mary, John)}, alongside plural forms.

We use the samples containing the explicit bias words as the original samples for the subsequent amplifying bias training and debiasing training. The augmented samples of the original samples

[1]While it may be different in the real-world, we chose the binary $(male, female)$ as our target demographic group.

are used to mine the implicit bias words. Specifically, for each original sample x, we replace all explicit bias words in x by each word pair in the list, obtaining a set of augmented sample pairs $\{(x_1^f, x_1^m), \cdots, (x_j^f, x_j^m), \cdots, (x_N^f, x_N^m)\}$, where N is the number of augmented sample pairs and also the number of explicit bias word pairs.

To mine explainable implicit bias words, we adopt Integrated Gradients (IG) (Sundararajan et al., 2017) which is a versatile explainability method with low time complexity. IG's idea is to aggregate gradients along the input that fall on the line between the baseline and the input. Given a sample x, the feature attribution of the k_{th} token t in x is defined as the path integral of straight path between baseline x'_k and input x_k:

$$attri_t ::= (x_k - x'_k) \int_{\alpha=0}^{1} \frac{\partial F_y(x' + \alpha(x - x'))}{\partial x_k} d\alpha, \tag{1}$$

where the function F_y represents the PLM-based classification model that predicts the true label y of the sample x, $\frac{\partial F_y(x)}{\partial x_k}$ represents the gradient of $F_y(x)$ on the k_{th} feature. The standard practice for x' uses zero embedding vectors.

Based on our hypothesis, implicit bias words indirectly cause PLMs to capture spurious associations between demographic groups and class labels. Therefore, in IG algorithm, the same implicit bias word gets different attention in different gender samples. Specificly, for each augmented sample pairs (x_j^f, x_j^m) of the original sample x, we compute the attribution $(attri_{j,t}^f, attri_{j,t}^m)$ of token t in both samples. To obtain the impact gap of the token t between female and male samples, the two attributes are subtracted and averaged across all augmented sample pairs. This impact gap is the *bias score* of token t, which is defined as:

$$\text{Score}_t = \frac{1}{N} \sum_{i=1}^{N} (attri_{j,t}^f - attri_{j,t}^m). \tag{2}$$

To obtain biased words that have a discriminative impact on different gender groups, we filter out tokens that are smaller than the bias threshold θ and add the remaining tokens to the implicit bias word list $\mathcal{W} = \{w_1, w_2, \cdots\}$.

2.2 Iteratively Training Bias-amplified Model

The initial list of implicit bias words can be directly used to debias PLMs. However, given the stability of search results, we iteratively train a bias-amplified model to identify implicit bias words more precisely. Previous research has shown that smaller models tend to be more biased than larger models (Sanh et al., 2021; Ahn et al., 2022; Silva et al., 2021). Therefore, we consider Tiny-BERT (Bhargava et al., 2021), a small variant of BERT, as the biased model. This provides the benefits of amplifying bias while reducing training overhead.

The iterative process is as follows. In the first iteration, we apply IG to the original Tiny-BERT to obtain an implicit bias word list. This list assists in training Tiny-BERT to amplify the model bias. The trained Tiny-BERT is then reapplied IG to search and update the implicit bias word list. In subsequent iterations, the list of the last iteration is used to further amplify Tiny-BERT's biases and is updated again. This process is repeated to iteratively grow a more precise list of implicit bias words.

The training objective is to ensure the model's accuracy in downstream tasks while making unfair decisions for different demographic groups. Given a batch of original samples $\{x_i|_{i=1}^B\}$, the task loss is calculated by the cross-entropy of the predicted label and the true label, defined as follows:

$$\mathcal{L}_{ce} = \frac{1}{B} \sum_{i=1}^{B} \text{CrossEntropy}(y_i, F_y(x_i)), \tag{3}$$

where B is the batch size, y_i and $F_y(x_i)$ denote the true label and the predicted label for the model output. We then add a bias regularization term that applies Jensen-Shannon divergence (JSD) and reweights the training samples to amplify the model bias. For each sample x_i, we define a *bias degree* to reweight it by summing the bias scores of all implicit bias words contained in x_i and then globally normalizing. The bias degree is calculated as:

$$D_i = \text{Norm}(\sum_{w \in \mathcal{W}'} \text{Score}_w), \tag{4}$$

where w is the implicit bias word in x_i and Score_w is its bias score, $\mathcal{W}'$ is obtained by sorting the implicit bias words in $\mathcal{W}$ from largest to smallest by their bias score Score_w and then keeping the top $\epsilon\%$. $\text{Norm}(\cdot)$ is the max-min normalization function computed as $\frac{\text{value}-\text{min}}{\text{max}-\text{min}}$, where max is the maximum value in the batch and min is the minimum value in the batch. The bias degree of original sample is used as the weight of its augmented sample pairs. The bias regularization term is:

$$\mathcal{L}_b = \frac{-1}{NB} \sum_{j=1}^{N} \sum_{i=1}^{B} D_{i,j} \times \text{JSD}(p_{i,j}^f, p_{i,j}^m), \tag{5}$$

where N is the number of augmented sample pairs, $p^f_{i,j}$ and $p^m_{i,j}$ are probability distributions of the j_{th} female augmented sample and male augmented sample of x_i, respectively. JSD$(\cdot)$ measures the consistence between the two gender distributions, and its larger value indicates that the two distributions are less similar. Therefore, to differentiate the two distributions so that the model makes inconsistent decisions, we take negative values for the results of JSD.

The overall loss function for amplifying bias training is as the combination of the cross-entropy loss $\mathcal{L}_{ce}$ and the bias regularization term $\mathcal{L}_b$: $\mathcal{L}_{bias} = \mathcal{L}_{ce} + \lambda \times \mathcal{L}_b$, where λ is the hyperparameter that regulates the weight of the bias regularization term. The trained biased model is then applied IG again to search for more precise implicit bias words. In turn, more precise implicit bias words catalyze the process of amplifying model bias. In the last iteration, we obtain the implicit bias words that have a strong harmful association with the explicit bias words.

2.3 Debiasing Training

In the debiasing stage, we use the implicit bias words searched in the last iteration to assist the model debiasing. We expect to improve the model's fairness without compromising the predictive ability in downstream tasks. Consistently, the task loss is the same as Eq.(3) and the debiasing term reweights the JSD using the bias degree. Differently, the goal of the debiasing term is to align the probability distribution of the augmented samples of two genders. Thus, the debiasing regularization term is defined as:

$$\mathcal{L}_d = \frac{1}{NB} \sum_{j=1}^{N} \sum_{i=1}^{B} D_{i,j} \times \text{JSD}(p^f_{i,j}, p^m_{i,j}). \quad (6)$$

The overall loss function of debiasing training is: $\mathcal{L}_{debias} = \mathcal{L}_{ce} + \gamma \times \mathcal{L}_d$, where γ is a hyperparameter that regulates the weight of the debiasing regularization term.

Samples with higher bias degrees contain more harmful associations related to gender. Reweighting changes the attention of samples, so that the PLM focuses on potentially harmful biased samples during training. The task loss function ensures the predictive ability of PLM in downstream tasks, and the debiasing regularization term encourages the model to learn consistent representations and thus make fair decisions. Therefore, PLM is improved for fairness with auxiliary training of implicit bias words.

3 Evaluation Methods

We evaluate the performance of Data-Debias on the task metrics and the fairness metrics.

Task Metrics. Four general accuracy metrics evaluate the predictive abilities of PLMs on all classification tasks, which are *Accuracy*, *F1 score*, *Precision*, and *Recall*. They are all calculated based on the predicted and true labels of the samples in the test set, and a higher value indicates the better predictive abilities of PLM.

Fairness Metrics. In the existing debiasing research, the external metrics to measure fairness are specific to a certain task and are not applicable to other tasks. To this end, we propose two fairness metrics common to all classification tasks, which are calculated based on the prediction results of each augmented sample set.

The first metric we propose to evaluate the fairness of PLMs is $FalseRatio/TrueRatio$ defined as $FR/TR = \frac{num(y_m \neq y_f)/N_{test}}{num(y_m = y_f)/N_{test}} = \frac{num(y_m \neq y_f)}{num(y_m = y_f)}$, where y_m and y_f are the labels that the model predicts for male and female augmented samples, N_{test} is the size of test set, and $num(\cdot)$ is the number statistics function. An ideal language model should make decisions based on the semantics and nothing else. Based on this, a fully unbiased model should give the same predicted label for augmented samples of the same sample. A lower value of FR/TR indicates a higher fairness of the model.

We propose a second metric to jointly evaluate the fairness and predictive abilities of PLMs, which is the $TruePositiveRatio$ defined as $TPR = \frac{num(y_m = y_f = y_{true})}{num(y_m = y_f)}$, where y_{true} is the true label of the sample. We expect the model to improve fairness while maintaining the accuracy of the predictions. TPR will decrease as the fairness increases when the prediction ability of the model is constant. We observe the change of TPR in the experiment to measure the overall performance of the model.

In addition, we also choose *F1 score*, *Precision*, and *Recall* as fairness indicators, whose inputs are all the prediction results of the male augmented sample y_m and the prediction results of the female augmented sample y_f.

4 Experiments

In this section, we experimentally analyze the performance of the Data-Debias framework to answer

the following research questions. **Q1** : How explainable are the detected implicit bias words? **Q2** : Compared to baseline methods, how effective is applying Data-Debias to debias PLMs in downstream tasks? **Q3** : What is the role of iteratively training the bias-amplified model? **Q4** : Whether implicit bias words generalize to debiasing PLMs in the prompting paradigm?

4.1 Experimental Setting

Downstream Tasks. For downstream debiasing, we chose three classification tasks corresponding to three datasets with different data sizes. Each dataset is matched and augmented based on a given list of explicit bias word pairs, and the train-test split is 50/50.

Stanford Sentiment Treebank (SST2)[2] (Socher et al., 2013) is a manually annotated binary sentiment analysis dataset to judge single-sentence movie reviews, which matches a total of 4,133 original samples containing gender attributes.

ToxiGen[3] (Hartvigsen et al., 2022) is an implicit toxic text classification dataset automatically generated based on prompts with revealed information, which matches 38,000 original samples containing explicit bias words. The prompt label for each sample is taken as the true label.

Bias in Bios[4] (De-Arteaga et al., 2019) is a third person biography dataset with 28 occupation categories and 250,000 original samples scraped. The task objective is to predict the category of occupation for each biography with the occupation information removed.

Comparing Baselines. We choose three model-centric debiasing methods *INLP* (Ravfogel et al., 2020), *Sent-Debias* (Liang et al., 2020), as well as *FairFil* (Cheng et al., 2021), and three data-centric debiasing methods *CDA* (Lu et al., 2020), *Auto-Debias* (Guo et al., 2022), as well as *MABEL* (He et al., 2022) as baselines. They are described as follows:

- INLP (Ravfogel et al., 2020) proposes an Iterative Null-space Projection debiasing method, which removes sensitive information from the neural representation by repeatedly training a linear classifier to predict the debiasing target and then projecting the representation on the null space.
- Sent-Debias (Liang et al., 2020) contextualizes predefined explicit bias words to the given sentence templates, and then debiasing by estimating the bias subspace of the sentence representation and eliminating projections on the bias subspace.
- FairFil (Cheng et al., 2021) proposes a fairness filter neural network to filter biases in sentence representations, which minimizes the correlation between filtered embeddings and explicit bias words via contrastive learning.
- CDA (Lu et al., 2020) enables debiasing research based on counterfactual data augmentation, which employs causal interventions to generate a copy of each sample by replacing the target word with a synonym.
- Auto-Debias (Guo et al., 2022) automatically searches for biased prompts in the top 5,000 most frequent Wikipedia vocabulary to debias by identifying the words that elicit the maximum JSD divergence between explicit bias words and stereotype words.
- MABEL (He et al., 2022) uses the priori explicit bias words to generate gender-balanced counterfactual entailment pairs in the natural language inference dataset, and introduces contrastive learning to narrow the representation between entailment pairs.

Implementation Details. We verify the effectiveness of debiasing on three PLMs: *BERT* (Devlin et al., 2019), *DistilBERT* (Sanh et al., 2019), and *RoBERTa* (Liu et al., 2019). The checkpoints of all models use `bert-base-uncased`, `distilbert-base-uncased`, and `roberta-base` implemented by the Huggingface Transformers library (Wolf et al., 2020). The [*CLS*] output from the last layer of PLM's encoder is pooled and fed to a classifier with one linear layer for prediction. All PLMs use Tiny-BERT to search for implicit bias words. In all experiments on three downstream tasks, the batch size B is set to 32, the learning rate is set to $5e-5$, and the hyperparameters λ and γ are set to 1 and 10. The number of steps M of the interpretability algorithm is set to 50 and the bias threshold θ is set to 0. The iteration number of amplifying bias training is chosen from 0 to 6, and the implicit bias words list $\mathcal{W}'$ in the amplifying bias training stage is selected to be the top $\epsilon = 30\%$.

[2] https://dl.fbaipublicfiles.com/glue/data/SST-2.zip
[3] https://github.com/microsoft/TOXIGEN
[4] https://github.com/microsoft/biosbias

Model	Task metrics (%)				Fairness metrics (%)				Overall (%)
	Acc ↑	F1 ↑	Precision ↑	Recall ↑	FR/TR ↓	F1 ↑	Precision ↑	Recall ↑	TPR ↑
BERT	91.78	91.69	91.51	92.08	2.45	97.60	97.80	97.49	92.25
+INLP	92.08↑0.30	91.85↑0.16	92.38↑0.87	91.51↓0.57	0.98	98.96	99.15	98.84	↓0.19 92.06
+Sent-Debias	91.30↓0.54	91.24↓0.45	91.13↓0.38	91.88↓0.20	1.62	98.40	98.41	98.43	↓1.62 90.63
+FairFil	91.88↑0.10	91.63↓0.06	92.32↑0.81	91.21↓0.87	1.48	98.47	98.78	98.20	↓0.24 92.01
+CDA	91.98↑0.20	91.74↑0.05	92.34↑0.83	91.37↓0.71	0.84	99.13	99.24	99.02	↓0.40 91.85
+Auto-Debias	91.78↑0.00	91.60↓0.09	91.76↑0.25	91.47↓0.61	1.32	98.67	98.84	98.52	↓0.20 92.05
+MABEL	91.49↓0.29	91.24↓0.45	91.83↑0.32	90.87↓1.21	1.13	98.83	99.08	98.61	↓0.47 91.78
+Data-Debias (Ours)	91.93↑0.15	91.73↑0.04	92.05↑0.54	91.49↓0.59	**0.39***	**99.60***	**99.62***	**99.58***	↓0.17 92.08
DistilBERT	92.13	92.04	91.85	92.39	2.39	97.65	97.84	97.54	92.50
+INLP	91.78↓0.35	91.61↓0.43	91.71↓0.14	91.52↓0.87	1.19	98.80	98.93	98.69	↓0.36 92.14
+Sent-Debias	92.17↑0.04	92.00↓0.04	92.11↑0.26	91.92↓0.47	1.05	98.94	99.06	98.83	↓0.33 92.17
+FairFil	91.64↓0.49	91.42↓0.62	91.80↓0.05	91.15↓1.24	1.26	98.70	98.93	98.50	↓0.93 91.57
+CDA	91.54↓0.59	91.45↓0.59	91.27↓0.58	91.86↓0.53	1.63	98.39	98.48	98.34	↓1.24 91.26
+Auto-Debias	91.88↓0.25	91.78↓0.26	91.60↓0.25	92.10↓0.29	2.34	97.71	97.90	97.61	↓0.52 91.98
+MABEL	-	-	-	-	-	-	-	-	-
+Data-Debias (Ours)	92.32↑0.19	92.14↑0.10	92.39↑0.54	91.94↓0.45	**0.52***	**99.47***	**99.51***	**99.43***	↓0.11 92.39
RoBERTa	92.37	92.22	92.28	92.16	5.52	94.68	94.92	94.51	93.50
+INLP	92.62↑0.25	92.45↑0.23	92.62↑0.34	92.31↓0.15	5.40	94.70	95.33	94.26	↓0.45 93.05
+Sent-Debias	92.32↓0.05	92.23↑0.01	92.05↓0.23	92.59↑0.43	4.98	95.24	95.25	95.22	↓1.32 92.18
+FairFil	93.06↑0.69	92.88↑0.66	93.21↑0.96	92.64↑0.48	4.93	95.12	95.64	94.73	↓0.50 94.00
+CDA	92.86↑0.49	92.67↑0.45	93.06↑0.78	92.40↑0.24	4.26	95.54	96.12	95.11	↑0.26 93.76
+Auto-Debias	93.40↑1.03	93.26↑1.04	93.32↑1.04	93.21↑1.04	6.73	93.50	94.15	93.09	↑1.18 94.68
+MABEL	92.57↑0.20	92.36↑0.14	92.83↑0.45	92.05↑0.11	4.69	95.29	96.13	94.71	↓0.10 93.40
+Data-Debias (Ours)	93.15↑0.78	92.97↑0.75	93.36↑1.08	92.70↑0.54	**3.22***	**96.77***	**96.93***	**96.62***	↓0.10 93.40

Table 1: Debiasing results on SST2. The best result is indicated in **bold**. ↓ and ↑ indicate decrease and increase in performance over biased BERT. * represent statistically significant ($\rho < 0.05$). MABEL does not provide code to run on DistilBERT.

confidence, ireland, stick, critically, eroded, dead, fails, virtues, white, tormented, enthusiasm, horribly, killed, tenderly, bilingual, tricky, amusing, scratches, affluent, naive, wicked, charming, writers, bully, rob, resist, sick, insulting, comedian, gun, dude, steal, damn, horror, miserable, rude, divine, hilarious, passionate, delicate, witch, elderly, pianist, strongest, suspects, sensitive, overwhelmed, hispanic, vulnerability, doomed, physician, sensual, raped, lip, tearing, black, indian, sympathetic, ghetto, argentine, sexy

Figure 2: Examples of implicit bias words from SST2. *Yellow* highlights race-related words. *Blue* highlights identity-related words. *Green* highlights words may have gender stereotypes. More examples are given in the Appendix A.

We implement INLP, Sent-Debias, and CDA using the code provided in (Meade et al., 2022), implement FairFil and Auto-Debias using the code provided by their authors, and MABEL using the checkpoints provided by its authors. CDA uses the same list of explicit bias word pairs as Data-Debias. All baselines are fine-tuned in our tasks and consistent with our experimental setup.

4.2 Implicit Bias Words

To answer research question **Q1**, we exhibit the top-scoring implicit bias words searched from the SST2 and rank them by bias score, as shown in Figure 2. From the reported results, the detected implicit bias words are intuitively explainable because they appear to have potential associations with gender. We highlight some representative words with different colors. The occurrence of race-related words marked in yellow indicates that the PLMs bias against gender groups is influenced by racial groups. These words are interpretable, for example *black women* are always discriminated against more than *white women* in the real-world. Blue highlights identity-related words, many of which contain gender stereotypes such as *comedian*, *witch*, and *physician*, which can induce gender bias in PLMs. Green highlights words that may have gender stereotypes, such as *sensual* being associated with women and *rude* being associated with men. PLM's focus on these words may lead to unfairness in the decision making. Other unhighlighted words may be more implicit or seemingly irrelevant. In general, it is difficult to notice the associations between these words and gender and ignore them, resulting in unsatisfactory debiasing results. Our proposed Data-Debias can comprehensively compensate for the potential defects in the data by searching for implicit bias words, thus achieving more powerful debiasing performance.

4.3 Debiasing Ability Analysis

To answer **Q2**, we apply Data-Debias to three PLMs in three downstream tasks. For all experiments, we evaluate their results for both task and fairness, as shown in Tables 1, 2, and 3. We report the results of Data-Debias using implicit bias words obtained in the fourth iteration of amplifying bias training. We show the results of original PLMs and present the change in the task metrics for debiasing model compared to the original model.

Table 1 shows the debiasing results in SST2. All three PLMs fit well and exhibit excellent accuracy and fairness due to the small size of the SST2 dataset. Interestingly, Data-Debias improves fairness even further. For both BERT and DistillBERT, it shrinks FR/TR to near 0 and improves

Model	Task metrics (%)				Fairness metrics (%)				Overall (%)
	Acc ↑	F1 ↑	Precision ↑	Recall ↑	FR/TR ↓	F1 ↑	Precision ↑	Recall ↑	TPR ↑
BERT	72.46	72.25	72.45	72.20	11.36	89.41	88.78	91.05	73.55
+INLP	71.92↓0.54	71.35↓0.90	72.44↓0.01	71.36↓0.84	8.81	91.31	90.24	93.42	↓1.57 71.98
+Sent-Debias	71.78↓0.68	70.51↓1.74	72.86↓0.68	70.85↓1.35	7.91	91.62	90.32	93.51	↓1.87 71.68
+FairFil	72.24↓0.42	71.34↓0.91	72.50↑0.05	71.47↓0.73	9.38	90.10	88.21	93.62	↓1.61 71.94
+CDA	71.54↓0.92	70.69↓1.56	72.57↑0.12	70.81↓1.39	7.82	91.52	90.01	93.72	↓3.03 70.52
+Auto-Debias	71.47↓0.99	70.05↓2.20	72.85↑0.30	70.48↓1.72	7.63	92.91	89.34	94.23	↓2.58 70.97
+MABEL	71.65↓0.81	70.93↓1.32	72.44↓0.01	70.99↓1.21	5.71	94.07	93.06	95.52	↓2.08 71.47
+Data-Debias (Ours)	72.03↓0.43	71.22↓1.03	73.06↑0.59	71.32↓0.88	**2.46***	**97.37***	**97.45***	**97.28***	↓1.71 71.84
DistilBERT	71.28	70.61	71.92	70.66	10.61	88.99	87.19	92.41	71.02
+INLP	71.73↑0.45	71.26↑0.65	72.07↑0.15	71.24↑0.58	7.09	92.75	91.85	94.01	↑0.47 71.49
+Sent-Debias	71.55↑0.27	70.92↑0.31	72.16↑0.24	70.95↑0.29	8.31	91.36	90.19	93.06	↑0.37 71.39
+FairFil	71.27↑0.01	70.70↑0.09	71.75↓0.17	70.72↓1.20	8.45	91.14	89.76	93.23	↓0.21 70.81
+CDA	70.96↓0.32	70.14↓0.47	71.87↓0.05	70.26↓0.40	7.73	91.52	89.98	93.74	↓0.37 70.65
+Auto-Debias	70.50↓0.78	68.57↓2.04	73.92↑2.00	69.34↓1.32	6.29	91.51	88.57	95.88	↓2.05 68.97
+MABEL	-	-	-	-	-	-	-	-	-
+Data-Debias (Ours)	71.12↓0.16	70.43↓0.18	71.79↓0.13	70.49↓0.17	**4.01***	**95.80***	**95.36***	**96.30***	↓0.13 70.89
RoBERTa	72.99	72.89	72.91	72.88	19.79	83.38	83.83	85.06	75.38
+INLP	71.14↓1.85	71.13↓1.86	71.59↓1.32	71.47↓1.41	18.82	84.16	84.17	84.17	↓0.91 74.47
+Sent-Debias	70.37↓2.62	70.17↓2.72	70.30↓2.61	70.13↓2.75	22.20	81.60	81.83	83.53	↓1.02 72.54
+FairFil	69.13↓3.86	69.13↓3.76	69.26↓3.65	69.29↓3.59	22.28	81.77	81.87	81.97	↓3.82 71.56
+CDA	72.93↓0.06	72.77↓0.06	72.87↓0.04	72.73↓0.15	17.85	84.14	84.41	83.92	↑0.01 75.39
+Auto-Debias	69.61↓3.38	69.60↓3.29	69.69↓3.22	69.74↓3.14	19.11	83.94	84.10	84.29	↓3.35 72.03
+MABEL	70.51↓2.48	70.51↓2.38	70.69↓2.22	70.71↓2.17	23.45	80.96	80.98	81.22	↓0.85 74.53
+Data-Debias (Ours)	72.58↓0.41	72.57↓0.32	72.67↓0.24	72.72↓0.16	**13.70***	**87.67***	**88.25***	**87.36***	↓0.31 75.07

Table 2: Debiasing results on ToxiGen. The best result is indicated in **bold**. ↓ and ↑ indicate decrease and increase in performance over biased BERT. * represent statistically significant ($\rho < 0.05$). MABEL does not provide code to run on DistilBERT.

Model	Task metrics (%)				Fairness metrics (%)				Overall (%)
	Acc ↑	F1 ↑	Precision ↑	Recall ↑	FR/TR ↓	F1 ↑	Precision ↑	Recall ↑	TPR ↑
BERT	84.15	78.23	77.94	78.99	6.16	91.82	92.26	92.19	86.25
+INLP	83.88↓0.27	78.32↑0.09	81.70↑3.76	76.38↓2.61	4.58	93.51	93.89	93.99	↓0.93 85.32
+Sent-Debias	83.69↓0.46	77.78↓0.45	80.77↑2.83	76.70↓2.29	5.02	93.29	94.03	93.47	↓1.02 85.23
+FairFil	84.04↓0.11	78.55↑0.32	82.76↑4.82	75.61↓3.38	3.76	94.56	95.24	94.38	↓1.15 85.10
+CDA	84.06↓0.09	78.45↑0.22	81.32↑3.38	76.49↓2.50	4.62	93.68	94.14	93.83	↓1.06 85.19
+Auto-Debias	84.50↑0.35	78.77↑0.54	79.50↑1.56	78.67↓0.32	3.75	94.93	95.15	95.06	↓1.19 85.06
+MABEL	84.22↑0.07	78.99↑0.76	81.67↑3.73	76.96↓2.03	4.28	93.59	94.40	93.55	↓1.22 85.03
+Data-Debias (Ours)	84.62↑0.47	79.17↑0.94	79.83↑1.89	79.19↑0.20	**2.69***	**96.07***	**96.15***	**96.20***	↓0.90 85.35
DistilBERT	83.12	78.30	77.69	79.28	7.22	91.09	92.05	91.40	84.50
+INLP	83.63↑0.51	77.79↓0.51	79.06↑1.37	77.71↓1.57	4.59	93.11	93.73	93.42	↑1.05 85.55
+Sent-Debias	83.62↑0.50	77.83↓0.47	78.52↑0.83	78.28↓1.0	4.44	93.72	94.32	93.90	↑0.90 85.40
+FairFil	83.68↑0.56	77.94↓0.36	78.85↑1.16	77.97↓1.31	4.38	93.59	94.13	93.94	↑1.05 85.55
+CDA	83.63↓0.51	77.91↓0.39	79.54↑1.85	77.50↓1.78	4.31	93.84	94.32	94.12	↑0.83 85.33
+Auto-Debias	84.44↑1.32	78.74↑0.44	79.39↑1.70	79.25↓0.03	3.89	94.01	94.32	94.38	↑1.39 85.89
+MABEL	-	-	-	-	-	-	-	-	-
+Data-Debias (Ours)	84.30↑1.18	78.94↑0.64	81.08↑3.39	77.71↓1.57	**3.15***	**95.63***	**95.84***	**95.79***	↑1.08 85.58
RoBERTa	83.44	77.89	78.31	78.19	6.49	90.95	90.81	91.92	84.92
+INLP	83.13↓0.29	77.28↓0.61	77.80↓0.51	77.75↓0.44	5.06	92.95	93.08	93.36	↓0.01 84.91
+Sent-Debias	83.03↓0.31	77.43↓0.46	81.00↑2.69	75.21↓2.98	4.33	93.50	93.08	94.59	↓0.38 84.54
+FairFil	83.43↓0.01	77.64↓0.25	77.18↓1.13	78.69↑0.38	4.96	93.03	92.62	94.10	↑0.15 85.07
+CDA	83.78↑0.34	78.02↑0.13	78.20↑0.11	78.88↑0.69	5.62	92.02	92.40	92.49	↑1.07 85.99
+Auto-Debias	84.00↑0.56	78.22↑0.23	79.42↑1.11	77.97↓0.22	6.49	93.27	93.75	93.16	↑1.08 86.01
+MABEL	83.92↑0.48	77.99↑0.10	78.58↑0.27	78.43↓0.24	5.25	92.13	92.28	92.75	↓1.04 85.96
+Data-Debias (Ours)	83.49↑0.05	77.66↓0.23	78.42↑0.11	77.87↓0.32	**3.89***	**94.87***	**94.89***	**95.08***	↑0.05 84.97

Table 3: Debiasing results on Bias in Bios. The best result is indicated in **bold**. ↓ and ↑ indicate decrease and increase in performance over biased BERT. * represent statistically significant $\rho < 0.05$). MABEL does not provide code to run on DistilBERT.

F1, precision, and recall to close to the ideal score of 100%, while the overall accuracy of the task is not damaged. In the case of RoBERTa, Data-Debias reduces bias and even improves predictive ability. The debiasing results on ToxiGen in Table 2 show that all debiasing methods impair the predictive ability to varying degrees. This is inevitable because fairness and accuracy are difficult to achieve at the same time. Data-Debias minimizes the damage to accuracy while greatly debiasing, which benefits from our debiasing strategy that combines the task objective and the debiasing objective. Table 3 shows the debiasing results on Bias in Bios. Compared with other debiasing methods, Data-Debias not only obtains the fairest score, but also performs the best overall in the task metrics.

From the debiasing results of the three tasks, all the baseline methods perform the debiasing ability to some extent. Overall, the three data-centric baselines CDA, Auto-Debias, and MABEL are more stable in debiasing than the three model-centric baselines INLP, Sent-Debias, and FairFil, achieving more effective debiasing with less degradation in task performance. However, the gap is marginal, because while data-centric baselines capture the direct harm caused by explicit bias words, they ignore potentially harmful associations in the data. Data-Debias achieves more robust debiasing and greater retention of performance, which benefits from the implicit bias words we mine to more deeply alleviate potentially harmful associations in the data.

TPR is an overall metric that we propose for joint task accuracy and fairness, which decreases as fairness is promoted with constant task accuracy. We expect the TPR to decrease as little as possible after debiasing. In the three downstream tasks,

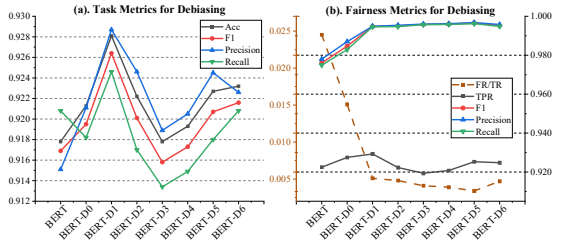

Figure 3: On the SST2 task, the results of debiasing BERT with implicit bias words from each iteration.

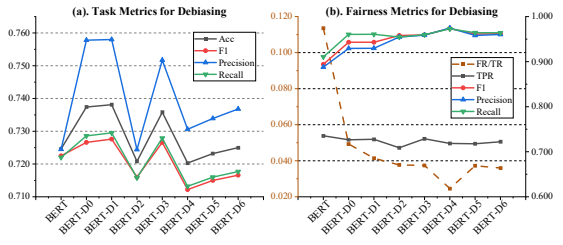

Figure 4: On the ToxiGen task, the results of debiasing BERT with implicit bias words from each iteration.

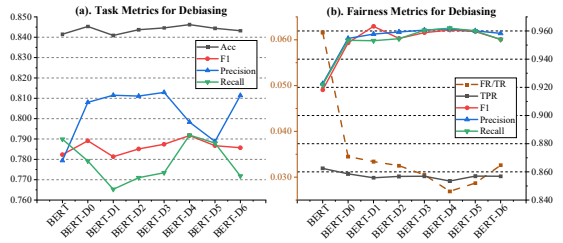

Figure 5: On the Bias in Bios task, the results of debiasing BERT with implicit bias words from each iteration.

Data-Debias achieves the least TPR drop compared to the baseline and even improves the original DistillBERT and RoBERTa on Bias-in-Bios task.

In summary, our proposed debiasing method Data-Debias has outstanding performance in different classification tasks and different PLMs, and it can extremely improve the fairness of PLMs while preserving the accuracy to the greatest extent.

4.4 Ablation Analysis

To answer research question **Q3**, we study ablation experiments with iteration numbers 0-6, and the results of debiasing in BERT are shown in Figures 3, 4, and 5. The results of bias-amplified Tiny-BERT are provided in the Appendix B.

For debiasing training, we report task metrics and fairness metrics for debiasing BERT using the implicit bias words searched in each iteration in three tasks. `BERT-DN` on the x-axis represents the BERT debiased using the implicit bias words searched in the N_{th} iteration. Note that `BERT-D0` is applied to implicit bias words searched by the 0_{th} iteration (i.e., original Tiny-BERT). According to the Figures 3(b) to 5(b), we find that the fairness of BERT is extremely improved at `BERT-D0` even without amplifying Tiny-BERT's bias, which verifies the effectiveness of our proposed Data-Debias in debiasing performance. Then, BERT's bias gradually decreases as the number of iterations increases. This shows that implicit bias words become more precise with amplifying bias training, leading to better debiasing effects. It is important to note that the elimination of bias is not endless, and the fairness peaks at the 4_{th} or 5_{th} iteration and then decreases. We analyze that excessive iterative training destroys the language modeling ability of the biased model, which in turn affects the performance of searching for implicit bias words.

Furthermore, we observe that the overall TPR metric fluctuates without significant decrease within a small range of the original BERT scores. This suggests that Data-Debias alleviates bias without negatively affecting the performance of the model. It can also be verified based on Figures 3(a) to 5(a). In most iterations, the scores of all task metrics are improved over the original BERT, indicating that moderate debiasing is beneficial to the accuracy of the model.

In summary, it is necessary to iteratively train the bias-amplified model, which plays a role in advancing the debiasing of the PLMs. Note that the degree of debiasing, i.e., the number of iterations, needs to be carefully chosen to avoid damaging the language modeling capability.

4.5 Generalization Analysis

To answer research question **Q4**, we experiment with zero-shot and few-shot tasks on two large-scale PLMs, that is *T5-Large* (Raffel et al., 2020) and *LLaMA-7B* (Touvron et al., 2023), to probe generalization of implicit bias words. We test on the validation set of three tasks.

We first find the augmented sample pairs that cause the model biases, then match them with implicit bias words. The top three implicit bias words with the highest bias scores in each biased sample are taken to generate auxiliary prompts, which are sent to the model to guide fair decisions. The template for the auxiliary prompts is as: *Reduce the focus on keywords 'bias_word$_1$', 'bias_word$_2$', and 'bias_word$_3$'*. We expect that simple and direct prompts can guide the model to pay less attention

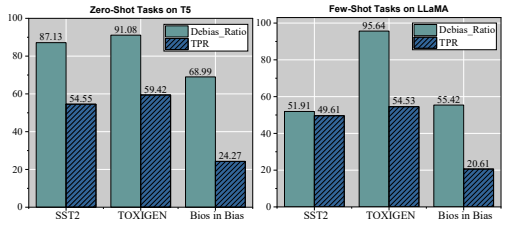

Figure 6: The results of zero-shot tasks on T5 and few-shot tasks on LLaMA with auxiliary prompt. *Debias_Ratio* denotes the rate at which the auxiliary prompts debias the unfair sample pairs. *TPR* denotes the ratio of correctly predicted sample pairs among the debiased sample pairs.

to implicit bias words, thereby weakening the potentially harmful associations in the data.

We report the results of zero-shot tasks on T5 and few-shot tasks on LLaMA, as shown in Figure 6. Excitingly, the experimental results show that providing PLMs with simple auxiliary prompts can effectively improve the fairness of decision. From the test results of three downstream tasks, we observe that T5 and LLaMA are sensitive to implicit bias words, especially in ToxiGen dataset, with an effective rate of 91.08% and 95.64%, respectively. Furthermore, the TPR scores achieve decent results, indicating that the model maintains predictive abilities while improving fairness. Although these implicit bias words are not re-searched by applying IG to T5 and LLaMA, they still perform exceptionally well in debiasing PLMs. More experimental details can be found in the Appendix C.

Experiments demonstrate that implicit bias words have strong generalization and can be applied to large-scale PLMs under the prompt paradigm without fine-tuning the model. It is extremely exciting for large-scale PLMs, we can improve the fairness of decision by simply adding an auxiliary prompt that contains implicit bias words.

5 Related Work

According to the training strategy, we divide existing debiasing methods into two types: model-centric debiasing and data-centric debiasing.

5.1 Model-Centric Debiasing

The model-centric debiasing strategies focus on identifying more effective models to improve the fairness of the models. Biases are mitigated by changing the model structure or training strategies (Webster et al., 2020), and introducing projection-based (Ravfogel et al., 2020) or contrastive learning (Liang et al., 2020; Cheng et al., 2021) to disregard sensitive attributes in representations. Model-centric debiasing strategies have two fatal flaws: 1) task-specific models are highly specialized and difficult to generalize to more tasks; 2) the quality of training data is ignored. There are many defects in the training data from messy sources. Unfiltered utterances on the Internet contain a large amount of discrimination, prejudice, and stereotypes of human society, and even a synthetic corpus is difficult to guarantee that samples are balanced.

5.2 Data-Centric Debiasing

The data-centric debiasing strategies aim to improve data quality and pursue data excellence. Representative methods are a range of extended methods for counterfactual data augmentation (Lu et al., 2020), such as Auto-Debias (Guo et al., 2022), MABEL (He et al., 2022), and CCPA (Li et al., 2023), which augment original training data with explicit bias words to compensate for imbalanced samples. However, these methods only take into account explicit bias words that directly cause bias, and ignore other unfavorable factors that may be mixed in the training data with complex distribution. In this paper, we define implicit bias words that have potentially harmful associations with explicit bias words, considering both direct and indirect causes that affect model fairness.

6 Conclusion

This paper mitigates social biases from a data perspective to improve fairness in PLMs. We propose a data-centric explainable debiasing method, which identifies implicit bias words that have potentially harmful associations with explicit bias words in the training data and reduces PLMs' focus on implicit bias words to alleviate biases. Implicit bias words are guaranteed to be interpretable by searching with Integrated Gradient and precise by iteratively amplifying bias training. Extensive experiments on three classification tasks demonstrate that Data-Debias can extremely improve the fairness of PLMs while maintaining the predictive abilities. The implicit bias words have strong generalization and can be applied to large-scale PLMs under the prompt paradigm without fine-tuning the model.

Limitations

In this work, we focus on debiasing the gender bias for PLMs. In the future, we will try to mitigate social biases other than gender, such as race and religion. In this paper, our proposed Data-Debias is specific to classification tasks, and in the future we plan to extend it to more downstream tasks, such as natural language inference and generative tasks. In addition, we will further explore the application of implicit bias words to debiasing on more large-scale PLMs.

Ethics Statement

This paper has been thoroughly reviewed for ethical considerations and has been found to be in compliance with all relevant ethical guidelines. The paper does not raise any ethical concerns and is a valuable contribution to the field.

Acknowledgments

We express gratitude to the anonymous reviewers for their hard work and kind comments. The work was supported in part by the National Natural Science Foundation of China (No.62272191, No.62372211), the International Science and Technology Cooperation Program of Jilin Province (No.20230402076GH, No.20240402067GH), the Science and Technology Development Program of Jilin Province (No.20220201153GX).

References

Jaimeen Ahn, Hwaran Lee, Jinhwa Kim, and Alice Oh. 2022. Why knowledge distillation amplifies gender bias and how to mitigate from the perspective of distilbert. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 266–272.

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 4349–4357.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems, NeurIPS*.

Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. 2021. Fairfil: Contrastive neural debiasing method for pretrained text encoders. In *Proceedings of the 9th International Conference on Learning Representations, ICLR*.

Maria De-Arteaga, Alexey Romanov, Hanna M. Wallach, Jennifer T. Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Cem Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT*, pages 120–128.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL*, pages 4171–4186.

Yue Guo, Yi Yang, and Ahmed Abbasi. 2022. Autodebias: Debiasing masked language models with automated biased prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 1012–1023.

Xudong Han, Timothy Baldwin, and Trevor Cohn. 2021. Diverse adversaries for mitigating bias in training. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, pages 2760–2765.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 3309–3326.

Jacqueline He, Mengzhou Xia, Christiane Fellbaum, and Danqi Chen. 2022. MABEL: attenuating gender bias using textual entailment data. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 9681–9702.

Yingji Li, Mengnan Du, Xin Wang, and Ying Wang. 2023. Prompt tuning pushes farther, contrastive learning pulls closer: A two-stage approach to mitigate social biases. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, ACL*, pages 14254–14267.

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

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. *CoRR*, abs/1907.11692.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2020. Gender bias in neural natural language processing. In *Proceedings of the Logic, Language, and Security - Essays Dedicated to Andre Scedrov on the Occasion of His 65th Birthday*, volume 12300 of *Lecture Notes in Computer Science*, pages 189–202.

Nicholas Meade, Elinor Poole-Dayan, and Siva Reddy. 2022. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 1878–1898.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative ppre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 7237–7256.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, A distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M. Rush. 2021. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *Proceedings of the 9th International Conference on Learning Representations, ICLR*.

Andrew Silva, Pradyumna Tambwekar, and Matthew Gombolay. 2021. Towards a comprehensive understanding and accurate evaluation of societal biases in pre-trained transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2383–2389.

Dylan Slack, Anna Hilgard, Sameer Singh, and Himabindu Lakkaraju. 2021. Reliable post hoc explanations: Modeling uncertainty in explainability. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 9391–9404.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, volume 70, pages 3319–3328.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. *CoRR*, abs/2010.06032.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 38–45.

Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. 2023. Data-centric artificial intelligence: A survey. *CoRR*, abs/2303.10158.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL*, pages 629–634.

A Implicit Bias Words

As shown in Figure 10, we exhibit the top-scoring implicit bias words searched from the three datasets and rank them by bias score.

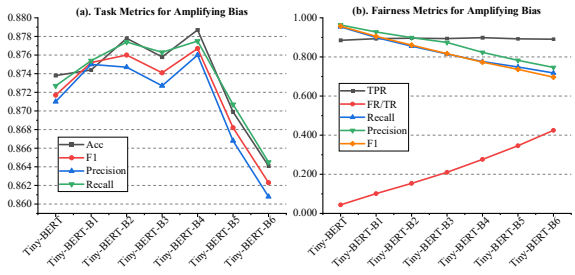

Figure 7: On the SST2 task, the results of iteratively amplifying bias training Tiny-BERT with implicit bias words from each iteration.

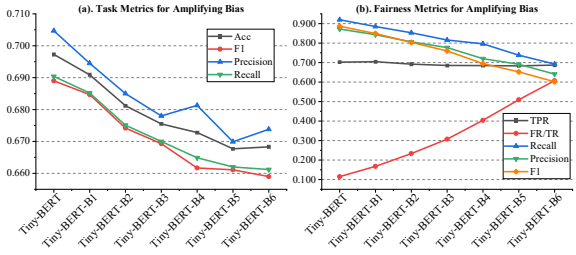

Figure 8: On the ToxiGen task, the results of iteratively amplifying bias training Tiny-BERT with implicit bias words from each iteration.

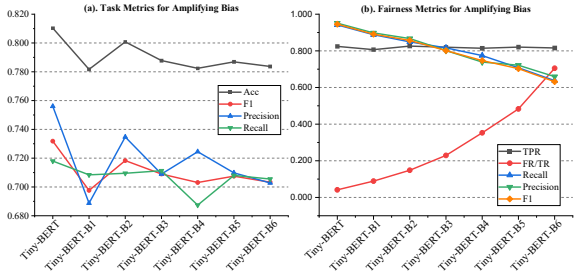

Figure 9: On the Bias in Bios task, the results of iteratively amplifying bias training Tiny-BERT with implicit bias words from each iteration.

B Results of Bias-amplified Tiny-BERT

For amplifying bias training, we report the task metrics and fairness metrics of Tiny-BERT with different iteration numbers in three downstream tasks, as shown in Figures 7, 8, and 9. `Tiny-BERT-BN` on the abscissa represents Tiny-BERT trained with N iterations of amplifying bias. It is observed from Figures 7(b) to 9(b) that Tiny-BERT's fairness gets worse as the iteration numbers increases, as expected from the amplifying bias training phase. At the same time, the task accuracy and overall index TPR of the model shows a general downward trend with the increase of the iteration number. Although the performance of Tiny-BERT is increasing before the 4_{th} iteration in the SST2 task, more iterations still cause performance degradation. This suggests that social biases within a model can damage its language modeling ability on downstream tasks.

Task	LLM	Acc↑	FR/TR↓	TPR↑
SST2	T5	94.87	1.20	94.57
	LLaMA	86.31	5.55	86.32
ToxiGen	T5	58.01	2.31	56.25
	LLaMA	51.37	2.22	51.43
Bios-in-Bias	T5	59.16	5.15	59.49
	LLaMA	49.34	12.52	48.20

Table 4: Original results for testing zero-shot tasks on T5 and few-shot tasks on LLaMA without auxiliary prompts.

Task	LLM	bias_num	ap_num	debias_num	tpr_num
SST2	T5	290	101	88	48
	LLaMA	1291	497	258	128
ToxiGen	T5	5626	2465	2245	1334
	LLaMA	5403	1904	1821	993
Bios-in-Bias	T5	73831	63620	43892	10652
	LLaMA	167699	143735	79659	16417

Table 5: The results of debiasing T5 and LLaMA with auxiliary prompts. *bias_num* represents the number of biased augmented sample pairs, *ap_num* represents the number of sample pairs in *bias_num* with auxiliary prompts, *debias_num* represents the number of sample pairs in *ap_num* that are debiased, and *tpr_num* represents the number of sample pairs in *debias_num* that are consistent with the predicted result and the true label.

C Generalization Analysis

We experimentally analyze the generalization of implicit bias words in two large-scale pretrained language models (LLMs). We use T5-Large[5] to test zero-shot tasks and LLaMA-7B[6] to test few-shot tasks implemented by the Huggingface Transformers library. For T5, we simply input the prompts with the task instructions and let the model predict the label of the sample. For LLaMA, we input two demonstration samples to help the model understand the goal of the task. The adopted prompt for each task is shown in Table 6.

We measure task accuracy by the agreement between the predicted labels of the original samples

[5]https://huggingface.co/t5-large

[6]https://huggingface.co/huggyllama/llama-7b

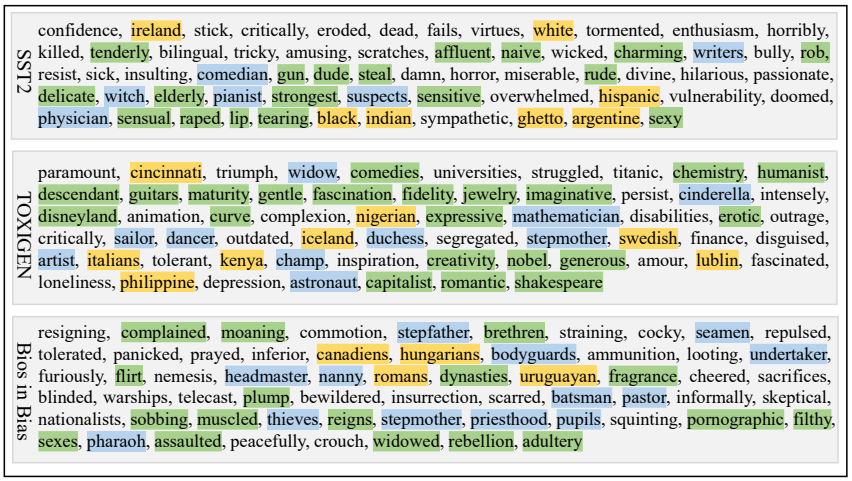

Figure 10: Examples of implicit bias words. *Yellow* highlights race-related words. *Blue* highlights identity-related words. *Green* highlights words that may have gender stereotypes.

and the true labels, and measure fairness by the agreement between the predicted results of the augmented sample pairs, and calculate the TPR score of the fair sample pairs. We report the original results for the undebiased T5 and LLaMA in Table 4. The results show that T5 performs better than LLaMA in terms of accuracy and fairness on all three tasks even though it only has 770M parameters. We analyze that T5, which reconstructs all NLP tasks into a unified text-to-text format for training, is better adapted to classification tasks than LLaMA, which is not fine-tuned on downstream tasks. For the obtained biased sample pairs, we guide LMs debiasing with the implicit bias word generation assisted prompts, and Table 5 reports the specific values corresponding to Figure 6.

LLM	Task	Prompt
T5	SST2	Determine whether the sentiment of the sentence is positive or negative: *Sample*
	ToxiGen	Determine whether the sentence is toxic or non-toxic: *Sample*
	Bios-in-Bias	Occupation classification, selected from [accountant, architect, attorney, chiropractor, comedian, composer, dentist, dietitian, dj, filmmaker, interior designer, journalist, model, nurse, painter, paralegal, pastor, personal trainer, photographer, physician, poet, professor, psychologist, rapper, software engineer, surgeon, teacher, yoga teacher]: *Sample*
LLaMA	SST2	Tweet: I hate it when my phone battery dies. Sentiment: Negative Tweet: My day has been good. Sentiment: Positive Tweet: *Sample* Sentiment:
	ToxiGen	Tweet: Black people are monkeys. Toxic: Yes Tweet: Women are good drivers. Toxic: No Tweet: *Sample* Toxic:
	Bios-in-Bias	Occupation classification, selected from [accountant, architect, attorney, chiropractor, comedian, composer, dentist, dietitian, dj, filmmaker, interior designer, journalist, model, nurse, painter, paralegal, pastor, personal trainer, photographer, physician, poet, professor, psychologist, rapper, software engineer, surgeon, teacher, yoga teacher]. Tweet: He teaches in high school. Occupation: teacher Tweet: She graduated from computer college and is good at programming. Occupation: software engineer Tweet: *Sample* Occupation:

Table 6: Prompts adopted for the zero-shot task on T5 and the few-shot task on LLaMA. *Sample* denotes the test sample.