Take its Essence, Discard its Dross! Debiasing for Toxic Language Detection via Counterfactual Causal Effect

Junyu Lu[†], Bo Xu[†], Xiaokun Zhang[†], Kaiyuan Liu[†], Dongyu Zhang[‡], Liang Yang[†], Hongfei Lin[†]*

[†]School of Computer Science and Technology, Dalian University of Technology, China,
[‡]School of Foreign Chinese, Dalian University of Technology, China
{dutljy, kun, 1154864382}@mail.dlut.edu.cn
{xubo, zhangdongyu, liang, hflin}@dlut.edu.cn

Abstract

Current methods of toxic language detection (TLD) typically rely on specific tokens to conduct decisions, which makes them suffer from lexical bias, leading to inferior performance and generalization. Lexical bias has both "*useful*" and "*misleading*" impacts on understanding toxicity. Unfortunately, instead of distinguishing between these impacts, current debiasing methods typically eliminate them indiscriminately, resulting in a degradation in the detection accuracy of the model. To this end, we propose a **C**ounterfactual **C**ausal **D**ebiasing **F**ramework (***CCDF***) to mitigate lexical bias in TLD. It preserves the "*useful impact*" of lexical bias and eliminates the "*misleading impact*". Specifically, we first represent the total effect of the original sentence and biased tokens on decisions from a causal view. We then conduct counterfactual inference to exclude the direct causal effect of lexical bias from the total effect. Empirical evaluations demonstrate that the debiased TLD model incorporating CCDF achieves state-of-the-art performance in both accuracy and fairness compared to competitive baselines applied on several vanilla models. The generalization capability of our model outperforms current debiased models for out-of-distribution data.

***Disclaimer**: The samples presented by this paper may be considered offensive or vulgar.*

Keywords: Toxic Language Detection, Lexical Bias, Causal Inference

1. Introduction

In recent years, researchers have introduced natural language processing techniques to detect toxic language. However, due to biased training, current toxic language detection (TLD) methods are prone to relying on lexical bias to perform decisions. The lexical bias associates toxicity with the presence of biased tokens (*e.g.*, identity mentions, insults, and markers of African American English) (Davidson et al., 2019; Zhang et al., 2020), which undermines the fairness of minorities (Thiago et al., 2021; Hutchinson et al., 2020). As an example, as shown in Figure 1, the TLD model tends to classify all samples containing "*n*gga*" (a cordial phrase for dialogue between Africans) as toxic language, due to its frequent occurrence in toxic samples during training. This actually compromises the freedom of expression of Africans (Sap et al., 2019). Meanwhile, lexical bias also affects the generalization ability of the TLD model, resulting in limited detection performance of the model for out-of-distribution (OOD) data (Vidgen et al., 2019; Ramponi and Tonelli, 2022; Zhou et al., 2021b).

Researchers have presented several methods to mitigate lexical bias in TLD. Due to the expensive labor costs of constructing unbiased datasets (Dinan et al., 2019), many studies have attempted to weaken lexical prior while training with original data, and enable models to make decisions without the impact of the bias (Swayamdipta et al., 2020; Chuang et al., 2021; Ramponi and Tonelli, 2022). However, these methods fail to distinguish the "*useful impact*" and "*misleading impact*" of lexical bias for understanding toxicity. In fact, lexical bias has positive effects on TLD, which was viewed as an effective surface feature for identifying toxic language in earlier work (Abney, 2014; Dinakar et al., 2015). As shown in Table 1, biased tokens are used to express toxic semantics in considerable comments. Therefore, interpreting lexical bias as a detriment to TLD and directly eliminating the bias can lead to a significant reduction in the accuracy of debiased models (Zhou et al., 2021b). To maintain detection performance while debiasing, it is necessary to examine how lexical bias influences model decisions from the dual characteristics.

Token	Toxic	Non-Toxic	Ratio (%)
black	244	76	76.25
n*gga	541	17	96.95
f*ck	878	46	95.02
ass	1592	132	92.34

Table 1: Proportion of toxic samples containing several biased tokens in the dataset (Founta et al., 2018), which are crawled from Twitter.

In this work, we propose a novel **C**ounterfactual

* Corresponding author

LREC-COLING 2024, pages 15566–15578

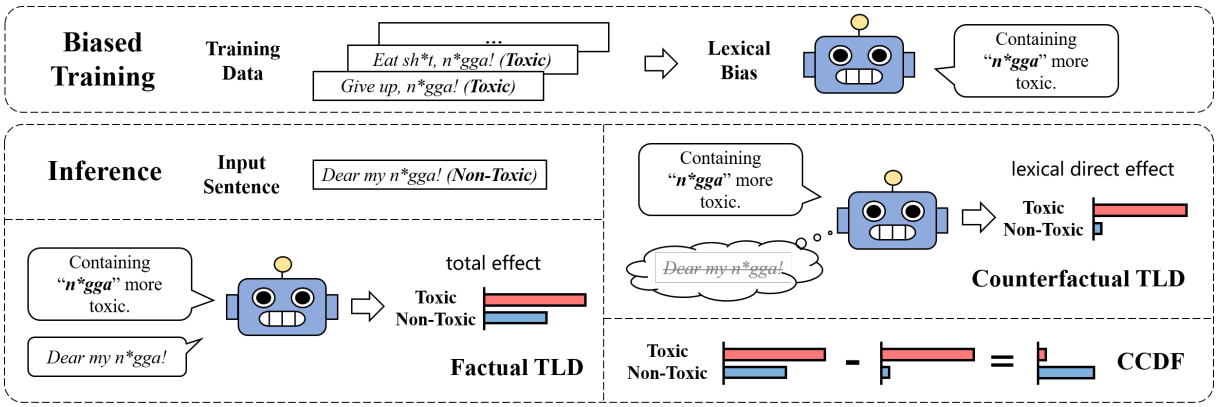

Figure 1: Due to the biased training, the TLD model is prone to identify all samples containing biased tokens, such as "*n*gga*", as toxic language. In this paper, we present a Counterfactual Causal Debiasing Framework to mitigate lexical bias by excluding the direct causal effect of biased tokens on model decisions from the total effect.

Causal **D**ebiasing **F**ramework (CCDF) to mitigate lexical bias for TLD. We employ causal learning techniques to examine the "*useful*" and "*misleading*" impact of lexical bias since it is applicable to estimating the effects of variables on model decisions (Pearl and Mackenzie, 2018). We formulate the "*useful impact*" of lexical bias as the causal effect of biased tokens combined with context information on decisions, while the "*misleading impact*" refers to the direct causal effect of biased tokens without introducing any context. As shown in Figure 1, two scenarios are constructed to calculate the causal effect of variables in TLD. Specifically, we design Factual TLD to estimate the total effect of biased tokens and the input sentence on detection, which jointly influence the predicate logit of the model. The Counterfactual TLD is proposed to estimate the direct causal effect of biased tokens, where the model is invariant to the changes of the sentence and only relies on the lexical bias to make decisions. We then conduct counterfactual inference to exclude the direct causal effect of biased tokens from the total effect, thus preserving the positive effects of lexical bias and mitigating the negative effects.

We evaluate the performance of the debiased TLD model incorporating CCDF for in-distribution data and out-of-distribution data. The experimental results demonstrate that the debiased model achieves state-of-the-art in both accuracy and fairness on several vanillas. And its migration ability outperforms current models. We further discuss the rationales of CCDF with empirical experiments.

The main contributions of this work are summarized as follows:

- We examine the positive and negative effects of lexical bias on model decisions in toxic language detection from the causal view.
- We present a Counterfactual Causal Debiasing Framework to retain the positive effects of lexical bias and mitigate negative effects, improving fairness while maintaining accuracy.
- We perform an empirical evaluation and demonstrate the effectiveness of our proposed framework in both in-distribution and out-of-distribution data[1].

2. Related Work

2.1. Debiasing for Toxic Language Detection

Toxic language is viewed as a rude, disrespectful, or unreasonable comment that is likely to make someone leave a discussion (Dixon et al., 2018). In recent years, researchers have tackled the problem of toxic language detection (TLD) using techniques of natural language processing (AlKhamissi et al., 2022; Tekiroglu et al., 2020; Zhou et al., 2021a; Mathew et al., 2021; Caselli et al., 2020; Hanu and Unitary team, 2020; Min et al., 2023; Lu et al., 2023a,b). Despite excellent performance on specific datasets, however, these methods over-rely on lexical bias in decision making, resulting in harm to the fairness of minority groups (Thiago et al., 2021). To mitigate the bias in TLD, many research efforts have been presented. The straightforward method is to balance the biased data, including using adversarial data (Xia et al., 2020; Dixon et al., 2018), filtering (Bras et al., 2020; Swayamdipta et al., 2020), relabeling (Zhou et al., 2021b) and counterfactual data augmentation (Sen et al., 2021, 2022). However, the application of these methods is challenging due to the significant costs of human annotation and the uncontrollability of data selection.

In view of this, several debiasing methods that weaken the influence of lexical priors have been

[1]Codes of this paper are available at https://github.com/DUT-lujunyu/Debias

presented, which can be applied directly to the original data. (Kennedy et al., 2020; Attanasio et al., 2022) calculated additional penalty loss for samples containing biased tokens to mitigate lexical bias. InvRat (Chuang et al., 2021) aimed to maintain invariant predictions, regardless of whether the model determines the sample contains biased tokens, thereby removing the bias. Badjatiya et al. (2019) and Ramponi and Tonelli (2022) remove and mask biased tokens directly on the original sample during the training phase, respectively. Motivated by LMixin (Swayamdipta et al., 2020), Zhou et al. (2021b) designed a separate branch which only makes decisions based on lexical bias while training, and directly excluded it in the test phase. While these methods have a certain degree of debiasing effect, they fail to leverage the positive effect of lexical bias, resulting in a decrease in the accuracy of debiased TLD models. In contrast, our CCDF performs counterfactual reference to ensure the detection performance of the model while mitigating the lexical bias.

2.2. Debiasing for Other NLU Tasks

In natural language understanding (NLU) tasks, some studies have focused on mitigating social biases in pre-trained language models to improve the fairness of models (Zmigrod et al., 2019; Liang et al., 2020; Cheng et al., 2021; Garimella et al., 2021; Kaneko and Bollegala, 2021; Guo et al., 2022; He et al., 2022; Webster et al., 2020). However, these methods are not applicable to debiasing for TLD due to the substantial difference in purpose. Specifically, they aim to eliminate the unbalanced model behaviors on socially sensitive topics, such as the spurious correlations between gender and careers, while the purpose of debiasing for TLD is to mitigate the dependence of model decisions on lexical bias. In addition, the sentences in TLD datasets are crawled from online platforms and contain more flexible word variants compared with the samples in ordinary NLU datasets (Wang et al., 2014), which also brings challenges for debiasing (Zhou et al., 2021a).

3. Preliminaries

3.1. Causal Learning

Causal learning aims to estimate the impact of variables on model decisions, which has been widely applied in various fields (Niu et al., 2021; Tang et al., 2020; Tian et al., 2022; Choi et al., 2022; Qian et al., 2021). Here we introduce the basic concepts of causal learning. For distinction, we use the uppercase letter to denote the variable (*e.g.*, *X*) and the lowercase refers to its observed value (*e.g.*, *x*),

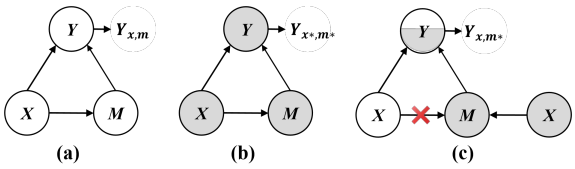

Figure 2: Illustration of causal graph. (a) Factual scenario; (b, c) Counterfactual scenario. Where white nodes denote variables with observed values and gray nodes denote variables with counterfactual values

while the lowercase letter with "*" represents the counterfactual value (*e.g.*, x^*).

Causal graph is a directed acyclic graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ represents the set of variables and $\mathcal{E}$ refers to the set of causal relationships from independent variables to dependent variables. An example of the causal graph is shown in Figure 2(a), which has three variables, *X*, *M*, and *Y*. In this case, *M* is a mediator between *X* and *Y*. Meanwhile, *X* has a direct causal effect and an indirect causal effect on *Y*, *i.e.*, $X \rightarrow Y$ and $X \rightarrow M \rightarrow Y$. Therefore, *Y* can be denoted as $Y_{x,m} = Y(X = x, M = m)$, where $m = M(X = x)$ in the factual scenario.

Causal effects reflect a comparison between the potential outcomes of the same individual in either treatment or not. Here we take variable *X* as an example. In the factual scenario, *X* is under treatment condition and gets observed value. And *M* and *Y* can respond to the variations of *X*. In the counterfactual scenario, *X* is under no-treatment condition and cannot directly affect its successor nodes. Furthermore, for a given variable in the causal graph, Total Effect (TE) refers to the sum of its predecessors' causal effect on it. By comparing Figure 2(a) and Figure 2(b), the TE on *Y* can be written as follows:

$$TE = Y_{x,m} - Y_{x^*,m^*}, \tag{1}$$

where $Y_{x^*,m^*} = Y(X = x^*, M = M(X = x^*))$.

In causal learning, TE consists of the natural direct effect (NDE) and total indirect effect (TIE). Where NDE estimates the direct causal effect of *X* on *Y* by blocking *M*, resulting in *M* failing to respond to the variations of *X*, as shown in the comparison between Figure 2(c) and Figure 2(b). NDE can be written as follows:

$$NDE = Y_{x,m^*} - Y_{x^*,m^*}. \tag{2}$$

Then TIE is the remaining effect of X and M on Y after excluding the direct causal effect of X on Y, which can be obtained by comparing TE and NDE, denoted as:

$$TIE = TE - NDE = Y_{x,m} - Y_{x,m^*} \tag{3}$$

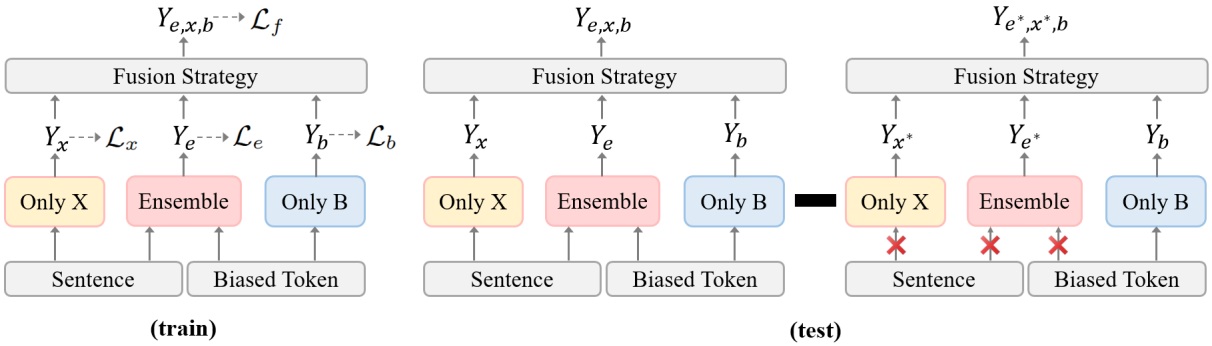

Figure 3: The model diagram of CCDF, where *Only X*, *Only B* and *Ensemble* represent different branch models, *i.e.* $\mathcal{F}_E$, $\mathcal{F}_X$, and $\mathcal{F}_B$, respectively. The vector representations of the original sentence and biased tokens are obtained by the same encoder. $\mathcal{L}_f$, $\mathcal{L}_x$, $\mathcal{L}_e$, and $\mathcal{L}_b$ respectively refer to the loss values between each predicate logit (*i.e.* $Y_{e,x,b}$, Y_x, Y_e, and Y_b) and the ground-truth label.

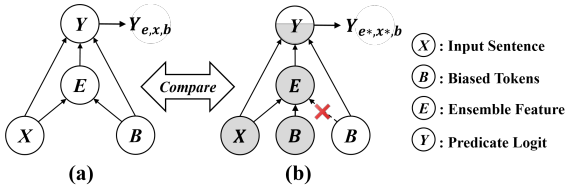

Figure 4: Comparison between (a) Factual TLD and (b) Counterfactual TLD using causal graph.

3.2. Problem Formulation

Let $X = \{w_1, ..., w_n\}$ a sentence containing n tokens. These tokens consist of both biased tokens, *e.g.* identity mentions, denoted as $B = \{b_1, ..., b_m\}$, and unbiased tokens. To recognize the biased tokens, a public toxic lexicon ToxTrig[2] (Zhou et al., 2021b) is introduced. For an *X* with a ground-truth label $y \in \{0, 1\}$, the TLD models aim to predict whether *X* is toxic or non-toxic, where the prediction is denoted as *Y*.

4. Methodology

4.1. Overview

We first introduce our Counterfactual Causal Debiasing Framework (CCDF) from a causal view, and analyze the total effect of biased tokens and original sentence on model decisions during the biased training (*i.e.* Factual TLD). Counterfactual reference is then performed in the test phase to make debiased predictions by excluding the direct causal effect of lexical bias from the total effect (*i.e.* Counterfactual TLD). The diagram of our CCDF is presented in Figure 3.

4.2. Causal View of CCDF

In the CCDF, we first propose an ensemble feature *E* integrating the original sentence *X* and biased tokens *B*. This facilitates the model to more adequately employ contextual information to determine whether biased tokens are used to express toxic semantics, maximizing the positive effects of lexical bias on model decisions. Then, several separate branch models are respectively utilized to obtain the logits for the three variables, *i.e.*, *E*, *X*, and *B*. We further incorporate these logits with a fusion function to generate the final predictions. The causal graph of CCDF is shown in Figure 4 and its details are as follows.

Node *X and B*. These two nodes denote the original sentence *X* and biased tokens *B*, respectively. We employ the same encoder to obtain the vector representations of the two inputs. The corresponding separator (*e.g.*, "[SEP]" in BERT (Devlin et al., 2019)) is utilized to separate each b_i.

Node *E*. It refers to the ensemble feature of *X* and *B*. As a mediator from *X* and *B* to *Y*, *E* can be written as follows:

$$E_{x,b} = E(X = x, B = b) \tag{4}$$

In this work, we employ Cross Attention to integrate *X* and *B* to obtain *E*:

$$E = softmax(\mathbb{X}^T \cdot \mathbb{B})\mathbb{X}, \tag{5}$$

where $\mathbb{X}$ and $\mathbb{B}$ refer to the vector representations of *X* and *B*, respectively.

Link *X* → *E and B* → *E*. Both *X* and *B* have direct causal effects on *E* due to *E* being built with the information of *X* and *B*.

Link *E* → *Y, X* → *Y, and B* → *Y*. These links denote the process by which each branch model outputs the predicate logit separately during the biased training phase. Therefore, *E*, *X*, and *B* have direct causal effects on *Y*. The branch models are represented as $\mathcal{F}_E$, $\mathcal{F}_X$, and $\mathcal{F}_B$, respectively.

Node *Y*. It refers to the final prediction result of CCDF, which integrates the outputs of three branch models with a fusion function. In the scenario of Factual TLD, all the input variables get observed

[2] https://github.com/XuhuiZhou/Toxic_Debias/blob/master/data/word_based_bias_list.csv

values. Therefore, branch models can respond to the variation of *E*, *X*, and *B*. And *Y* can be written as follows:

$$Y_{e,x,b} = Y(E = e, X = x, B = b), \tag{6}$$

where $e = E_{x,b}$ integrates the information of both lexical bias and context information.

4.3. Debiasing Inference with Casual Effect

As the definition of total effect (TE) shown in Section 3.1, we compare the Factual TLD and no-treatment condition to get TE of *E*, *X*, and *B* on *Y*, which can be written as:

$$TE = Y_{e,x,b} - Y_{e^*,x^*,b^*}, \tag{7}$$

where e^*, x^*, and b^* denote the corresponding variables under no-treatment condition.

Furthermore, based on the casual graph, the effect of biased tokens *B* on the predicate logits *Y* can be divided into two parts: the direct causal effect via $B \rightarrow Y$ and the indirect causal effect via $B \rightarrow E \rightarrow Y$ which incorporates the context information. Due to the significance of maintaining detection performance while debiasing, it is necessary to address the effect of *B* from both sides. To mitigate the negative effects of lexical bias, the direct causal effect of *B* on *Y*, *i.e.*, $B \rightarrow Y$, has to be eliminated from the total effect. Meanwhile, to exert the positive effects of the bias, the indirect causal effect, *i.e.*, $B \rightarrow E \rightarrow Y$, should be reserved. The scenario of Counterfactual TLD is designed to estimate the direct causal effect of *B*, and counterfactual inference is then conducted. Specifically, we block the direct causal effect of *E* and *X* on *Y*, causing the branch model $\mathcal{F}_E$ and $\mathcal{F}_X$ invariant, which cannot respond to the variation of input variables *E* and *X*. This leads to the TLD model only relying on the lexical bias to make decisions, which is the natural direct effect (NDE) of *B* on *Y*.

$$NDE = Y_{e^*,x^*,b} - Y_{e^*,x^*,b^*}. \tag{8}$$

Then the total indirect effect (TIE) of variables on *Y* can be calculated by excluding NDE from TE:

$$TIE = TE - NDE = Y_{e,x,b} - Y_{e^*,x^*,b}. \tag{9}$$

And we use TIE as the debiased prediction.

4.4. Other implementation details

In the implementation, we employ three separate MLPs as branch models. As shown in Figure 3, $\mathcal{F}_X$, $\mathcal{F}_E$, and $\mathcal{F}_B$ are running in the Factual TLD, while $\mathcal{F}_X$ and $\mathcal{F}_E$ are blocked in the Counterfactual TLD. Therefore, the output of each branch model can be defined as follows:

$$Y_b = y_b = \mathcal{F}_B(b), \tag{10}$$

$$Y_x = \begin{cases} y_x = \mathcal{F}_X(x) & \text{if } X = x \\ y_x^* = c_x & \text{if } X = \varnothing \end{cases}, \tag{11}$$

$$Y_e = \begin{cases} y_e = \mathcal{F}_E(x, b) & \text{if } X = x \\ y_e^* = c_e & \text{if } X = \varnothing \end{cases}, \tag{12}$$

where c_x and c_e refer to the invariant responses of $\mathcal{F}_X$ and $\mathcal{F}_E$, respectively, which can be trained or set as hyperparameters. And $\varnothing$ denotes the no-treatment condition.

To obtain the final predicate logit, we utilize the harmonic function to integrate Y_e, Y_x, and Y_b. The fused score $Y_{e,x,b}$ is as follows:

$$Y_{e,x,b} = h\left(Y_e, Y_x, Y_b\right) = \log \frac{Z_{e,x,b}}{1 + Z_{e,x,b}}, \tag{13}$$

where $Z_{e,x,b} = tanh\left(Y_e\right) \cdot tanh\left(Y_x\right) \cdot tanh\left(Y_b\right)$.

In the training phase, we utilize cross-entropy to calculate the difference between the predicate logits, including the output of each branch (*i.e.* Y_e, Y_x, and Y_b) and the fused score $Y_{e,x,b}$, and the ground-truth label y. The final loss function is defined as follows:

$$\begin{aligned} \mathcal{L}_{all} &= \mathcal{L}_f + \mathcal{L}_e + \mathcal{L}_x + \mathcal{L}_b \\ &= \mathcal{L}(Y_{e,x,b}, y) + \mathcal{L}(Y_e, y) + \mathcal{L}(Y_x, y) + \mathcal{L}(Y_b, y). \end{aligned} \tag{14}$$

The parameters of TLD models are optimized by minimizing $\mathcal{L}_{all}$. In addition, since *X* and *B* share the same encoder, we do not backpropagate $\mathcal{L}(Y_b, y)$ to the encoder, preventing the encoder from learning lexical bias directly. And $\mathcal{L}(Y_b, y)$ is only used to update parameters of $\mathcal{F}_B$.

5. Experiments

5.1. Datasets and Evaluation Metrics

For fair comparisons with baselines of debiasing methods for the TLD model, we followed Zhou et al. (2021b) and selected the same benchmarks in both in-distribution and out-of-distribution data. Specifically, we first conducted the main experiment on (Founta et al., 2018), which has 32K toxic and 54K non-toxic samples crawled from Twitter. Referenced by Zhou et al. (2021b), we focused on three kinds of lexical biases, including non-offensive minority identity (NOI), *e.g.*, *gay*, offensive minority identity (OI), *e.g.*, *n*gga*, and offensive non-identity (ONI), *e.g.*, *f*ck*. Overall accuracy (Acc) and F_1 are used to measure the detection performance

of TLD models. Then F_1 and false positive rate (FPR) on the samples containing nOI, OI, and ONI are respectively reported, evaluating the degree of lexical bias in the model. Intuitively, the lower the FPR, the less the model relies on lexical bias in decision making, and the fairer the model.

We then evaluated the performance of trained models on OOD data. We used the test set of (Dinan et al., 2019) as the adversarial dataset, which contains 6k artificial sentences (including 600 toxic samples). The language style of these artificially constructed samples is quite different from the in-distribution data crawled from Twitter and has a more standardized character. In addition, many of the toxic samples in this dataset are implicit and do not contain insults towards minorities. This presents a serious challenge to the generalization capability of TLD models. Here we use the accuracy and weighted F_1 as evaluation metrics.

5.2. Baselines and Experimental Settings

We conducted various baselines to mitigate lexical bias in TLD models, including both weakening lexical prior with original data and unbiased training with data filtering. For methods of weakening lexical prior, we selected Masking (Ramponi and Tonelli, 2022), LMixin (Swayamdipta et al., 2020) and InvRat (Chuang et al., 2021). We evaluated the methods on three commonly used PLMs, including BERT-base (Devlin et al., 2019), RoBERTa-base and RoBERTa-large (Liu et al., 2019). For methods of data filtering, two data filtering methods were introduced and applied on RoBERTa-large, including AFLite (Bras et al., 2020) and DataMaps (Swayamdipta et al., 2020). The filtered training data size is 33% of the original training set.

In the experimental stage, we trained TLD models on the training set of (Founta et al., 2018) and saved their best parameters on the validation set. Then we respectively evaluated the performance of models on the test set and adversarial dataset. To further prove the generalization of our CCDF, we also evaluated its performance on balanced training data filtered by AFLite and DataMaps, respectively. We did not perform any pre-processing of the datasets, or any hyperparameter search, but followed all the settings in Zhou et al. (2021b). We use one NVIDIA GeForce RTX 3090 to perform the experiments. AdamW is used as the optimizer for model training.

5.3. Quantitative Results

5.3.1. Main Discussions

Table 2 shows our empirical evaluation results of both in-distribution and OOD data. And we have the following findings:

RQ1: Performance of Weakening Lexical Prior Methods on In-distribution Data. Overall, whereas most debiasing methods of weakening the lexical prior can improve the fairness of TLD models, they also lead to a reduction in the accuracy of models' detection for in-distribution data. Here we take LMixin as an example, which is a competitive baseline for mitigating lexical bias. After the adoption of LMixin, the accuracy and F_1 value of the models decrease on average by almost 2.5% for the samples of the test set.

We also notice that models introducing Masking have little performance degradation on in-distribution data and have almost the worst ability to mitigate the bias. This is because Masking itself is an incomplete method of debiasing, and the biased tokens are only masked in the training set and remain in the validation and test sets. Therefore, even though it prevents the model from learning the lexical bias during the training phase, the model can still make decisions on the test set based on the lexical prior learned during pre-training. Since the PLM itself has already learned the semantics of these tokens, the overall impact of Masking on model performance is minimal.

In contrast, with the introduction of our CCDF, TLD models outperform the original vanilla in detecting in-distribution data, and achieve state-of-the-art debiasing effects for the lexical bias of nOI and OI. This demonstrates that our method can effectively mitigate bias while preserving model detection performance, achieving a trade-off between model accuracy and fairness.

RQ2: Performance of Weakening Lexical Prior Methods on Out-of-distribution Data. Compared to the original vanilla, TLD models introducing the debiasing method exhibit a better detection performance on OOD data. This indicates that removing spurious associations between biased tokens and labels can improve the generalization ability of the models. Moreover, our CCDF significantly outperforms other debiasing methods to provide maximum benefit to the model, with an improvement of approximately 4.5% in the F_1 value. Meanwhile, We also find that the accuracy of TLD models in detecting OOD data is much higher than the F_1 value. This is because the adversarial dataset contains implicitly toxic samples that the models often misclassify as non-toxic, leading to a decrease in F_1. We will further conduct an error analysis to illustrate these samples in Section 5.4 below.

RQ3: Comparison with Debiasing Methods of Data Filtering. Models trained on the balanced data have high performance on in-distribution data. This is because data filtering methods are able to select the most efficient samples, enabling models to achieve optimal results while utilizing minimal data. However, the debiasing effect of these mod-

Method	Test (12893)		nOI (602)		OI (553)		OnI (3236)		OOD (6000)	
	$Acc\uparrow$	$F_1\uparrow$	$F_1\uparrow$	$FPR\downarrow$	$F_1\uparrow$	$FPR\downarrow$	$F_1\uparrow$	$FPR\downarrow$	$Acc\uparrow$	$F_1\uparrow$
weakening lexical prior with BERT-base:										
Vanilla†	$93.53_{0.1}$	$91.39_{0.1}$	$89.29_{0.3}$	$9.22_{0.4}$	$98.90_{0.0}$	$85.71_{3.4}$	$97.19_{0.0}$	$66.34_{1.4}$	$91.28_{0.1}$	$81.43_{2.4}$
Masking	$93.24_{0.1}$	$91.08_{0.1}$	$\mathbf{89.33}_{0.2}$	$9.56_{0.4}$	$\mathbf{98.80}_{0.2}$	$83.33_{3.4}$	$\mathbf{97.24}_{0.0}$	$64.88_{0.5}$	$91.55_{0.0}$	$81.71_{0.6}$
LMixin	$91.85_{0.5}$	$89.51_{0.5}$	$87.19_{0.2}$	$10.24_{2.7}$	$98.33_{0.0}$	$\mathbf{74.52}_{0.0}$	$97.08_{0.1}$	$59.51_{1.8}$	$91.58_{0.1}$	$83.67_{3.2}$
CCDF(ours)	$\mathbf{93.75}_{0.0}$	$\mathbf{91.59}_{0.0}$	$88.54_{0.7}$	$\mathbf{4.10}_{0.9}$	$98.62_{0.8}$	$78.57_{0.0}$	$97.15_{0.2}$	$\mathbf{59.02}_{2.3}$	$\mathbf{91.63}_{0.1}$	$\mathbf{85.81}_{1.2}$
weakening lexical prior with RoBERTa-base:										
Vanilla†	$94.04_{0.1}$	$91.70_{0.1}$	$90.10_{0.3}$	$8.40_{0.4}$	$98.60_{0.0}$	$81.00_{3.4}$	$97.00_{0.0}$	$63.40_{1.4}$	$92.19_{0.1}$	$81.78_{2.4}$
Masking	$93.91_{0.1}$	$\mathbf{92.01}_{0.1}$	$\mathbf{89.60}_{0.2}$	$5.30_{0.4}$	$\mathbf{98.21}_{0.2}$	$80.95_{3.4}$	$\mathbf{97.32}_{0.0}$	$62.76_{0.5}$	$92.27_{0.0}$	$82.35_{0.6}$
LMixin	$92.05_{0.5}$	$90.53_{0.5}$	$87.51_{0.2}$	$6.35_{2.7}$	$97.93_{0.0}$	$\mathbf{71.43}_{0.0}$	$97.14_{0.1}$	$63.09_{1.8}$	$91.92_{0.1}$	$83.11_{3.2}$
InvRat†	-	$91.00_{0.5}$	$85.50_{1.6}$	$3.40_{0.6}$	$97.50_{1.0}$	$76.20_{3.4}$	$97.20_{0.2}$	$61.10_{1.5}$	-	-
CCDF(ours)	$\mathbf{94.05}_{0.0}$	$91.86_{0.0}$	$85.91_{0.7}$	$\mathbf{2.85}_{0.9}$	$97.69_{0.8}$	$\mathbf{71.43}_{0.0}$	$97.12_{0.2}$	$\mathbf{57.23}_{3.3}$	$\mathbf{92.39}_{0.1}$	$\mathbf{86.12}_{1.2}$
weakening lexical prior with RoBERTa-large:										
Vanilla‡	$94.21_{0.0}$	$92.33_{0.0}$	$89.76_{0.3}$	$10.24_{1.3}$	$98.84_{0.1}$	$85.71_{0.0}$	$97.34_{0.1}$	$64.72_{0.8}$	$92.20_{0.1}$	$82.20_{2.0}$
Masking	$93.67_{0.1}$	$91.75_{0.1}$	$\mathbf{87.56}_{0.7}$	$8.19_{1.1}$	$98.40_{0.5}$	$83.33_{3.4}$	$97.40_{0.1}$	$61.79_{2.3}$	$91.93_{0.2}$	$84.01_{2.2}$
LMixin‡	$90.44_{0.7}$	$86.94_{1.1}$	$85.47_{0.3}$	$11.15_{1.7}$	$97.64_{0.3}$	$\mathbf{71.43}_{0.0}$	$90.41_{1.8}$	$\mathbf{44.55}_{1.5}$	-	-
LMixin	$91.67_{1.1}$	$89.58_{1.1}$	$86.76_{0.8}$	$6.94_{0.7}$	$98.12_{0.3}$	$78.57_{2.9}$	$96.95_{0.1}$	$56.10_{1.2}$	$91.95_{0.1}$	$85.35_{1.9}$
CCDF(ours)	$\mathbf{94.15}_{0.1}$	$\mathbf{92.07}_{0.1}$	$86.65_{0.9}$	$\mathbf{3.75}_{1.0}$	$\mathbf{98.49}_{0.3}$	$78.57_{0.0}$	$\mathbf{97.42}_{0.1}$	$58.54_{3.2}$	$\mathbf{92.33}_{0.0}$	$\mathbf{86.40}_{1.6}$
balancing training data with RoBERTa-large:										
AFLite	$93.86_{0.1}$	$91.94_{0.1}$	$90.21_{0.4}$	$8.22_{1.1}$	$98.90_{0.0}$	$85.71_{0.0}$	$97.32_{0.1}$	$62.44_{0.0}$	$91.34_{0.2}$	$79.61_{2.3}$
w/ CCDF	$93.85_{0.1}$	$91.83_{0.0}$	$86.36_{0.6}$	$3.83_{0.7}$	$98.78_{0.2}$	$78.57_{0.0}$	$97.31_{0.1}$	$59.35_{2.8}$	$91.73_{0.1}$	$82.56_{1.9}$
DataMaps	$94.33_{0.1}$	$92.45_{0.1}$	$89.16_{0.7}$	$7.39_{1.0}$	$98.87_{0.1}$	$85.71_{0.0}$	$97.54_{0.0}$	$64.39_{1.4}$	$91.54_{0.3}$	$81.62_{1.3}$
w/ CCDF	$94.25_{0.0}$	$92.20_{0.1}$	$88.11_{0.4}$	$3.75_{0.9}$	$98.34_{0.1}$	$78.57_{0.0}$	$97.13_{0.1}$	$60.49_{1.1}$	$92.20_{0.3}$	$83.64_{1.8}$

Table 2: Evaluation on the test set of (Founta et al., 2018) and adversarial dataset (Dinan et al., 2019). Results show the mean and s.d. (subscript) of Acc and F_1 across 3 runs, as well as F_1 and FPR towards test samples containing specific mentions in TOXTRIG, including NOI, OI, and ONI. The **best** results of debiasing methods that weaken lexical prior are highlighted in each column. †: results reported in Chuang et al. (2021); ‡: results reported in Zhou et al. (2021b). ↑: greater the better; ↓: lower the better.

els is limited, especially for the lexical bias of OI and OnI. This is due to the fact that data filtering only relies on the confidence probability of the model on the sample without effectively eliminating samples containing lexical bias. Furthermore, for OOD data, we reach the same conclusion as Zhou et al. (2021a) that data filtering methods have poor generalization ability compared to debiasing methods that weaken lexical prior due to insufficient training data (only 33% of the original data after filtering). Besides, these methods require a relatively high time overhead to perform additional training rounds for data selection. In contrast, our method is more efficient by training on the original dataset and has a better effect of mitigating bias.

Furthermore, we evaluate the performance of CCDF on balanced training data. The results indicate that the incorporation of CCDF can further enhance the fairness of the model trained on the balanced data, resulting in a significantly lower FPR. Meanwhile, the debiased model retains the benefits of the filtering data method, maintaining detection performance on in-distribution data. The results of OOD data demonstrate its enhanced capacity for generalization.

Method	Test	nOI	OOD
	$F_1\uparrow$	$FPR\downarrow$	$F_1\uparrow$
RoBERTa-base	$91.70_{0.1}$	$8.40_{0.4}$	$81.78_{2.4}$
CCDF	$\mathbf{91.86}_{0.0}$	$\mathbf{2.85}_{0.9}$	$\mathbf{86.12}_{1.2}$
w/o $\mathcal{F}_e$	$91.82_{0.1}$	$3.57_{1.2}$	$83.69_{1.7}$
w/o $\mathcal{F}_x$	$91.86_{0.1}$	$3.04_{0.8}$	$84.23_{1.5}$
w/o $\mathcal{F}_x$ & $\mathcal{F}_b$	$91.76_{0.1}$	$7.64_{0.5}$	$82.24_{1.7}$

Table 3: Ablation experiments on branch model, where w/o $\mathcal{F}_x$ & $\mathcal{F}_b$ refers to the model which only relies on $\mathcal{F}_e$ to make decisions.

5.3.2. Ablation Experiments

Here we further conduct ablation experiments on the branch model of our CCDF. The experimental results are shown in Table 3. From the results, we obtain the following conclusions:

(1) Ablation of branch model $\mathcal{F}_x$ or $\mathcal{F}_e$ has lit-

	Sentence	Label	Vanilla	LMixin	CCDF
(a)	@user @user You don't have to pay for their ***bullshit*** read your rights read the law I don't pay fo. . .	✋	⚠	✋	✋
(b)	RT @user: my ex so ugly to me now like...i'll beat that ***hoe ass***.	⚠	⚠	✋	⚠
(c)	@user Stop that, it's not your fault a ***scumbag*** decided to steal otems which were obviously meant for someone i. . .	⚠	⚠	✋	✋
(d)	He should go back to his lad.	⚠	✋	✋	✋

Table 4: Examples from the test set and adversarial dataset with predictions from vanilla (RoBERTa-base), LMixin, and our CCDF. ⚠ denotes toxic label and ✋ denotes non-toxic label. Biased tokens in examples are highlighted.

tle effect on the performance of the model on in-distribution data. This is because counterfactual reasoning can still be performed on the ablated model to remove the negative effects of lexical bias and improve fairness. Meanwhile, the accuracy of the ablated model on OOD data is significantly reduced. This reflects that joint training of multiple branches facilitates the generalization ability of the model. In addition, whether ablating branch model $\mathcal{F}_x$ or $\mathcal{F}_e$, ablated CCDF has a more competitive performance than baselines, illustrating the effectiveness of our framework.

(2) We also find that CCDF without the branch model $\mathcal{F}_x$ has higher accuracy and better fairness than without $\mathcal{F}_e$, and has stronger generalization on OOD data. Meanwhile, compared with vanilla, $\mathcal{F}_e$ performs better. This reflects the fact that integrating context information and biased tokens for making decisions can fully exploit the positive effects of lexical bias, demonstrating the significance of ensemble features to TLD.

5.4. Qualitative analysis

In this section, we further illustrate the capability of our CCDF in mitigating the lexical bias for TLD, providing several examples shown in Table 4. We choose RoBERTa-base as vanilla and list predictions from LMixin and our CCDF for comparison.

For Exp. (a), the vanilla incorrectly predicts this non-toxic sentence as toxic, due to the biased token "*bullshit*", which is itself a high-frequency swear word, even though in context it does not express the semantics of toxic. And the vanilla introducing either LMixin or CCDF can perform a debiased decision. For Exp. (b), we find that the vanilla and our CCDF correctly predict the label, while LMixin does not. This is because it ignores the positive effects of biased tokens (i.e., "*hoe*" and "*ass*"). In contrast, our CCDF integrates context information and lexical bias to preserve the positive effect and provide more accurate predictions.

To gain more insights into the performance of our model, we list two sentences that our model misclassifies, Exp. (c) and (d) shown in Table 4, and conduct an error analysis. For Exp. (c), both debiased models introducing LMixin and CCDF incorrectly predict this toxic sentence as non-toxic, while vanilla correctly predicts the label. This is because the toxicity degree of "*scumbag*" is small, leading debiased models to consider the sample as non-toxic. And for Exp. (d), which implicitly expresses sarcasm towards LGBTQ, vanilla and LMixin also make incorrect predictions. This result reflects that current TLD models still lack enough world knowledge to capture potential toxicity, resulting in poor detection of samples that do not contain insults.

6. Conclusion

In this paper, we propose a Counterfactual Causal Debiasing Framework to mitigate lexical bias in toxic language detection. We formulate the bias as the causal effect of biased tokens on decisions and build a causal graph to analyze the causal relationship between variables and predicate logits. In the training phase, our framework integrates context information to leverage the positive effects of lexical bias, guaranteeing the detection performance of the model. The negative effects of the bias are then removed from the total effect by performing counterfactual reasoning during the testing phase. In the experiments, we show that our framework significantly outperforms the state-of-the-art debiasing methods on both accuracy and fairness for TLD. Furthermore, we demonstrate that the debiased model employing our framework has an excellent generalization capability in out-of-distribution data. In the future, we will further evaluate the debiasing effect of CCDF for other NLU tasks, as well as the performance applied to large language models.

Limitations

In this work, we utilized a toxic lexicon as prior knowledge to recognize biased tokens. However,

since lexical bias is generated during model training rather than being artificially proposed (Hutchinson et al., 2020), the external lexicon may not align with the biased tokens learned by the TLD model during training. Consequently, an incomplete lexicon could result in new lexical bias, which would negatively impact the fairness of the model (Joshi and He, 2022). Furthermore, our study did not investigate sentence-level dialectal bias, such as African American English (AAE), which is officially considered a less appropriate language variety, and this exacerbates racial bias (Sap et al., 2019). Based on the above, it is imperative to acknowledge that CCDF should not be perceived as a universal solution to mitigate all the bias in TLD. Instead, it should be regarded as an innovative attempt to highlight certain aspects of a complex, elusive, and multifaceted problem. In the future, we will investigate techniques for identifying lexical bias from the standpoint of model training and examine strategies for alleviating dialectal bias in the TLD model.

Ethics Statement

Due to the research field of this work, we recognize that the examples provided in the paper may have a negative impact on certain minority groups. However, our values include honesty, integrity, respect, fairness, and responsibility. We are committed to treating all individuals with dignity and respect, and to promoting a culture of inclusivity and diversity. Therefore, the views and conclusions presented in these examples should not be interpreted as reflecting the opinions or beliefs expressed or implied by the authors. Our hope is that the advantages of this research outweigh any potential risks.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (No. 62376051, 62076046, 62076051, 62066044), and Liaoning Province Applied Basic Research Program (No. 2022JH2/101300270). We would like to thank all reviewers for their constructive comments.

Bibliographical References

Steven P. Abney. 2014. Semi-supervised learning and domain adaptation in natural language processing. *Mach. Transl.*, 28(1):61–63.

Badr AlKhamissi, Faisal Ladhak, Srinivasan Iyer, Veselin Stoyanov, Zornitsa Kozareva, Xian Li, Pascale Fung, Lambert Mathias, Asli Celikyilmaz, and Mona Diab. 2022. ToKen: Task decomposition and knowledge infusion for few-shot hate speech detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2109–2120, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Giuseppe Attanasio, Debora Nozza, Dirk Hovy, and Elena Baralis. 2022. Entropy-based attention regularization frees unintended bias mitigation from lists. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1105–1119. Association for Computational Linguistics.

Pinkesh Badjatiya, Manish Gupta, and Vasudeva Varma. 2019. Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 49–59. ACM.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1078–1088. PMLR.

Tommaso Caselli, Valerio Basile, Jelena Mitrovic, and Michael Granitzer. 2020. Hatebert: Retraining BERT for abusive language detection in english. *CoRR*, abs/2010.12472.

Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. 2021. Fairfil: Contrastive neural debiasing method for pretrained text encoders. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Seungtaek Choi, Myeongho Jeong, Hojae Han, and Seung-won Hwang. 2022. C2L: causally contrastive learning for robust text classification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10526–10534. AAAI Press.

Yung-Sung Chuang, Mingye Gao, Hongyin Luo, James Glass, Hung-Yi Lee, Yun-Nung Chen, and Shang-Wen Li. 2021. Mitigating biases in toxic language detection through invariant rationalization. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 114–120.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35.

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: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Karthik Dinakar, Rosalind W. Picard, and Henry Lieberman. 2015. Common sense reasoning for detection, prevention, and mitigation of cyberbullying (extended abstract). In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 4168–4172. AAAI Press.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4536–4545. Association for Computational Linguistics.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA, February 02-03, 2018*, pages 67–73. ACM.

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018*, pages 491–500. AAAI Press.

Aparna Garimella, Akhash Amarnath, Kiran Kumar, Akash Pramod Yalla, Anandhavelu Natarajan, Niyati Chhaya, and Balaji Vasan Srinivasan. 2021. He is very intelligent, she is very beautiful? on mitigating social biases in language modelling and generation. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4534–4545. Association for Computational Linguistics.

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 (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1012–1023. Association for Computational Linguistics.

Laura Hanu and Unitary team. 2020. Detoxify. Github. https://github.com/unitaryai/detoxify.

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 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 9681–9702. Association for Computational Linguistics.

Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. 2020. Social biases in NLP models as barriers for persons with disabilities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5491–5501. Association for Computational Linguistics.

Nitish Joshi and He He. 2022. An investigation of the (in)effectiveness of counterfactually augmented data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3668–3681. Association for Computational Linguistics.

Masahiro Kaneko and Danushka Bollegala. 2021. Debiasing pre-trained contextualised embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1256–1266. Association for Computational Linguistics.

Brendan Kennedy, Xisen Jin, Aida Mostafazadeh Davani, Morteza Dehghani, and Xiang Ren. 2020. Contextualizing hate speech classifiers with post-hoc explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July*

5-10, 2020, pages 5435–5442. Association for Computational Linguistics.

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 2020, Online, July 5-10, 2020*, pages 5502–5515. Association for Computational Linguistics.

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.

Junyu Lu, Hongfei Lin, Xiaokun Zhang, Zhaoqing Li, Tongyue Zhang, Linlin Zong, Fenglong Ma, and Bo Xu. 2023a. Hate speech detection via dual contrastive learning. *IEEE ACM Trans. Audio Speech Lang. Process.*, 31:2787–2795.

Junyu Lu, Bo Xu, Xiaokun Zhang, Changrong Min, Liang Yang, and Hongfei Lin. 2023b. Facilitating fine-grained detection of chinese toxic language: Hierarchical taxonomy, resources, and benchmarks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 16235–16250. Association for Computational Linguistics.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14867–14875. AAAI Press.

Changrong Min, Hongfei Lin, Ximing Li, He Zhao, Junyu Lu, Liang Yang, and Bo Xu. 2023. Finding hate speech with auxiliary emotion detection from self-training multi-label learning perspective. *Inf. Fusion*, 96:214–223.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual VQA: A cause-effect look at language bias. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12700–12710. Computer Vision Foundation / IEEE.

Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic books.

Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma, and Pengjun Xie. 2021. Counterfactual inference for text classification debiasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5434–5445. Association for Computational Linguistics.

Alan Ramponi and Sara Tonelli. 2022. Features or spurious artifacts? data-centric baselines for fair and robust hate speech detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3027–3040. Association for Computational Linguistics.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1668–1678. Association for Computational Linguistics.

Indira Sen, Mattia Samory, Fabian Flöck, Claudia Wagner, and Isabelle Augenstein. 2021. How does counterfactually augmented data impact models for social computing constructs? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 325–344. Association for Computational Linguistics.

Indira Sen, Mattia Samory, Claudia Wagner, and Isabelle Augenstein. 2022. Counterfactually augmented data and unintended bias: The case of sexism and hate speech detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4716–4726. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 9275–9293. Association for Computational Linguistics.

Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. 2020. Unbiased scene graph generation from biased training. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 3713–3722. Computer Vision Foundation / IEEE.

Serra Sinem Tekiroglu, Yi-Ling Chung, and Marco Guerini. 2020. Generating counter narratives against online hate speech: Data and strategies. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1177–1190. Association for Computational Linguistics.

Dias Oliva Thiago, Antonialli Dennys Marcelo, and Alessandra Gomes. 2021. Fighting hate speech, silencing drag queens? artificial intelligence in content moderation and risks to lgbtq voices online. *Sexuality & culture*, 25(2):700–732.

Bing Tian, Yixin Cao, Yong Zhang, and Chunxiao Xing. 2022. Debiasing NLU models via causal intervention and counterfactual reasoning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11376–11384. AAAI Press.

Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. 2019. Challenges and frontiers in abusive content detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P. Sheth. 2014. Cursing in english on twitter. In *Computer Supported Cooperative Work, CSCW '14, Baltimore, MD, USA, February 15-19, 2014*, pages 415–425. ACM.

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.

Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. 2020. Demoting racial bias in hate speech detection. In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media, SocialNLP@ACL 2020, Online, July 10, 2020*, pages 7–14. Association for Computational Linguistics.

Guanhua Zhang, Bing Bai, Junqi Zhang, Kun Bai, Conghui Zhu, and Tiejun Zhao. 2020. Demographics should not be the reason of toxicity: Mitigating discrimination in text classifications with instance weighting. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4134–4145. Association for Computational Linguistics.

Xianbing Zhou, Yang Yong, Xiaochao Fan, Ge Ren, Yunfeng Song, Yufeng Diao, Liang Yang, and Hongfei Lin. 2021a. Hate speech detection based on sentiment knowledge sharing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7158–7166.

Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Yejin Choi, and Noah A. Smith. 2021b. Challenges in automated debiasing for toxic language detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 3143–3155. Association for Computational Linguistics.

Ran Zmigrod, S. J. Mielke, Hanna M. Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1651–1661. Association for Computational Linguistics.

A. Experimental Details

A.1. Hyperparameter Setting

We use one NVIDIA GeForce RTX 3090 to perform the experiments. AdamW is used as the optimizer for model training. The invariant responses of $\mathcal{F}_X$ and $\mathcal{F}_E$ in the Counterfactual TLD, i.e. c_x and c_e, are obtained by training. Other details of hyperparameters are directly followed (Zhou et al., 2021b), listed in Table A1. While under non-optimal hyperparameters, debiased models with CCDF still obtain state-of-the-art performance in both accuracy and fairness across most datasets compared with other methods.

Hyperparameter	Value
epochs	3
saved steps	1000
batch size	8
learning rate	1e-5
dropout	0.1
hidden state of MLPs	256
padded length of sentence	128
padded length of biased tokens	16

Table A1: The hyperparameters of the experiments.

A.2. Baseline Introduction and Implementation

Here we further introduce the baselines and their implementation details. The same hyperparameters are utilized as our CCDF.

Masking (Ramponi and Tonelli, 2022): In the training set, the biased tokens are masked, while in the validation set and test set, they are still retained.

LMixin (Zhou et al., 2021b): In the training phase, model decisions depend on the outputs of two branch models, whose inputs are the original sentence and biased tokens respectively, like our CCDF without $\mathcal{F}_E$. In the test phase, the model makes predictions only based on the sentence.

InvRat (Chuang et al., 2021): InvRat is a three-player framework consisting of an environment-agnostic predictor, an environment-aware predictor, and a rationale generator. Therefore, three independent RoBERTa models are running at the same time during the training phase.

AFLite (Bras et al., 2020): An ensemble of simple linear classifiers is trained and tested on the dataset. Samples that are correctly classified by most of the classifiers in the ensemble are considered to contain lexical bias and are discarded. The algorithm is iterative until the remaining data reaches the target size.

DataMaps (Swayamdipta et al., 2020): For a specific model, there are different regions in a dataset, including easy, hard, and ambiguous regions. These regions are identified based on the confidence of the model in the true category of examples, and the variation of this confidence during the training phase. DataMaps-Easy, DataMaps-Ambiguous, and DataMaps-Hard subsets of the dataset are then created (Founta et al., 2018).

Following (Zhou et al., 2021a), the size of filtered subsets is set to 33% of the original training set for both filtering methods, and label proportions are preserved. Then a RoBERTa-large classifier is fine-tuned on filtered subsets.

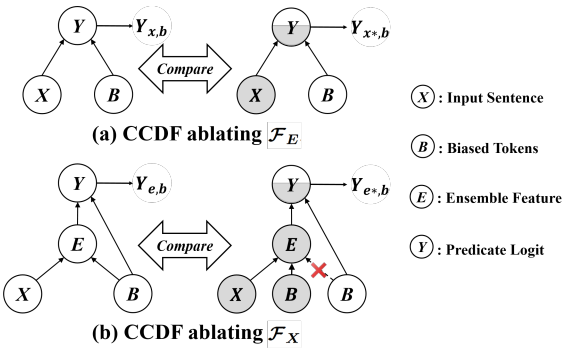

Figure B1: Causal graph of ablated CCDF.

B. Causal View of Ablated CCDF

Here we introduce the ablated CCDF from the causal view. As shown in Figure B1(a), for CCDF ablating branch model $\mathcal{F}_e$, only the original sentence *X* and biased tokens *B*, but not ensemble feature *E*, directly affect the model decisions *Y*. Therefore, TE of variables on *Y* can be written as:

$$TE = Y_{x,b} - Y_{x^*,b^*}. \tag{15}$$

And NDE of *B* on *Y* is:

$$NDE = Y_{x^*,b} - Y_{x^*,b^*}. \tag{16}$$

We then obtain the TIE by comparing TE and NDE as debiased predicate logits:

$$TIE = TE - NDE = Y_{x,b} - Y_{x^*,b}. \tag{17}$$

Similarly, the causal graph of CCDF ablating $\mathcal{F}_x$ is shown in Figure B1(b) and the causal effect is calculated as follows:

$$TE = Y_{e,b} - Y_{e^*,b^*}. \tag{18}$$

$$NDE = Y_{e^*,b} - Y_{e^*,b^*}. \tag{19}$$

$$TIE = TE - NDE = Y_{e,b} - Y_{e^*,b}. \tag{20}$$