Bias Mitigation for Toxicity Detection via Sequential Decisions

Lu Cheng
School of Computing and Augmented Intelligence, Arizona State University
Tempe, AZ, USA
lcheng35@asu.edu

Ahmadreza Mosallanezhad
School of Computing and Augmented Intelligence, Arizona State University
Tempe, AZ, USA
amosalla@asu.edu

Yasin N. Silva
Department of Computer Science, Loyola University Chicago
Chicago, Illinois, USA
ysilva1@luc.edu

Deborah L. Hall
School of Social and Behavioral Sciences, Arizona State University
Glendale, AZ, USA
d.hall@asu.edu

Huan Liu
School of Computing and Augmented Intelligence, Arizona State University
Tempe, AZ, USA
huanliu@asu.edu

ABSTRACT

Increased social media use has contributed to the greater prevalence of abusive, rude, and offensive textual comments. Machine learning models have been developed to detect toxic comments online, yet these models tend to show biases against users with marginalized or minority identities (e.g., females and African Americans). Established research in debiasing toxicity classifiers often (1) takes a static or batch approach, assuming that all information is available and then making a one-time decision; and (2) uses a generic strategy to mitigate different biases (e.g., gender and racial biases) that assumes the biases are independent of one another. However, in real scenarios, the input typically arrives as a sequence of comments/words over time instead of all at once. Thus, decisions based on partial information must be made while additional input is arriving. Moreover, social bias is complex by nature. Each type of bias is defined within its unique context, which, consistent with intersectionality theory within the social sciences, might be correlated with the contexts of other forms of bias. In this work, we consider debiasing toxicity detection as a sequential decision-making process where different biases can be interdependent. In particular, we study debiasing toxicity detection with two aims: (1) to examine whether different biases tend to correlate with each other; and (2) to investigate how to jointly mitigate these correlated biases in an interactive manner to minimize the total amount of bias. At the core of our approach is a framework built upon theories of sequential Markov Decision Processes that seeks to maximize the prediction accuracy and minimize the bias measures tailored to individual biases. Evaluations on two benchmark datasets empirically validate the hypothesis that biases tend to be correlated and corroborate the effectiveness of the proposed sequential debiasing strategy.

CCS CONCEPTS

- Security and privacy → Social aspects of security and privacy;
- Human-centered computing → Collaborative and social computing;　Collaborative and social computing design and evaluation methods;
- Computing methodologies → Supervised learning.

KEYWORDS

unintended bias, toxicity detection, sequential decision-making, social media

ACM Reference Format:

1 INTRODUCTION

Current machine learning models for toxicity detection exhibit problematic and discriminatory performance, resulting in poorer prediction [10, 33] and negatively impacting disadvantaged and minority groups [14, 21, 43]. That is, Instagram sessions that include comments with swear words can be flagged as toxic even when the swear words are used inoffensively and tweets containing words related to minority groups are more likely to be identified as toxic. For example, the widely-used Perspective API has been found to identify “Wussup, n”ggal!” and “F”cking love this.” as toxic comments with high probability, revealing swear-words-based lexical bias [43] and potential dialect-based racial bias against African American English (AAE), respectively.

Despite promising efforts to debias toxicity detection and related tasks (e.g., cyberbullying detection), most research to date (e.g., [9, 14, 42]) is based on two assumptions: (1) bias mitigation is a “static” problem where the model has access to all of the information and makes a one-time decision; and (2) different types of biases are independent of one another. Yet, comments/words in social media often come in a sequence instead of all at once. In
This environment, conventional batch-processing can be impractical. Further, the relations among different biases are complex. As shown in Fig. 1, sessions containing comments written in Hispanic English (HE) or AAE with swear words contribute larger portions of toxic sessions than those without swear words in a benchmark dataset. Recent work (e.g., [21]) also showed evidence of intersectional bias within toxicity detection: AAE tweets were 3.7 times as likely and African American male tweets were 77% more likely to be labeled as toxic [21]. In the social sciences, intersectionality is the idea that multiple identity categories (e.g., race and gender) combine interactively in ways that contribute to greater bias than the bias associated with each category alone. Informed by these findings, we first hypothesize that biases tend to be correlated in toxicity detection.

To effectively mitigate potentially correlated biases with a sequential input, we address two challenges: (1) making sequential decisions based on partial information, e.g., comments observed so far, given that “static” debiasing may cause unnecessary delay and is less responsive when the input is changing (e.g., topic diversion); and (2) characterizing unique aspects of individual biases to reduce the overall bias. Conventional debiasing strategies provide a generic and one-size-fits-all solution. A straightforward approach is to add multiple fairness constraints w.r.t. different biases to the training process of a toxicity classifier. However, it overlooks the unique characteristics of each bias and confronts challenging optimization problems. This leads to our second hypothesis: with sequential input, sequential bias mitigation strategies that include bias measures tailored to individual biases can improve the debiasing performance in the presence of potentially correlated biases.

To test our hypotheses, we study the novel problem of joint bias mitigation for toxicity detection via sequential decision-making. The goal is to effectively detect toxicity and mitigate potentially correlated biases as comments arrive sequentially. This work proposes a sequential debiasing strategy for toxicity detection – Joint – built on theories of sequential Markov Decision Processes (MDP) [2] and a pairwise comparison bias measure that compares every two groups sampled from the same bias type. Joint is model-agnostic (i.e., any standard toxicity classifier can be the input model) and focuses on debiasing with sequential inputs.

The major contributions of this paper are the following:

- We investigate the novel and practical research question of joint bias mitigation for toxicity detection via sequential decisions;
- We propose two hypotheses that offer new theoretical and practical insights for research on bias and fairness in AI;
- We propose a sequential bias mitigation approach that takes sequential input seeking to maximize the prediction accuracy and minimize the bias measures tailored to individual biases.
- Empirical evaluation on two benchmark datasets shows that our approach can effectively reduce the total amount of bias and present competitive prediction accuracy.

2 RELATED WORK

We briefly review two lines of research closely related to our work – toxicity detection and bias mitigation in text classification.

2.1 Toxicity Detection

Toxicity detection has received considerable attention as a tool for mitigating the detrimental impact of toxicity online (see, e.g., [5, 25]) – where toxicity is defined as “a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion” [1]. Toxicity is also used as an umbrella term for different kinds of toxic language, including language that is “hateful” [31, 35, 39], “offensive” [32, 41], “cyberbullying” [12, 24], or “abusive” [25, 28].

An early work in hate speech detection [38] extracted n-grams and part-of-speech tags from comments as features to train a Support Vector Machine (SVM) classifier. Subsequent work [25] examined the effectiveness of various linguistic features, such as character/word n-grams and word embeddings. Experimental results found that models trained with the combination of all features achieved the best predictive performance. Other promising approaches have emerged from research on cyberbullying detection. For instance, studies that augmented textual features with rich social media information, such as social network [36] and other multi-modal information (e.g., time, location) [8], were found to improve the performance of cyberbullying detection significantly. Notably, there has been growing interest in models designed specifically for toxicity detection (e.g., [5, 15]). For example, because toxic users may continually modify their content to circumvent comment filters, a recent work [5] proposed using sentiment information to help detect toxicity, based on the hypothesis that it is harder for toxic users to hide their sentiments. Their results showed that sentiment information has a positive impact on toxicity detection.

Recent work [6, 29] has highlighted the role of semantic context in detecting toxicity. Inspired by the hierarchical attention network proposed in [40], Cheng et al. [6] used a hierarchical structure to model a social media session with a sequence of comments and attention weights to differentiate the word/comment importance. Surprisingly, others (e.g., Pavlopoulos et al. [29]) have found a lack of evidence that context improves the performance of toxicity classifiers. These inconsistent findings point to the need for in-depth analyses of context-aware toxicity detection.
2.2 Bias Mitigation in Text Classification

Computational methods can reinforce and even propagate unintended biases in text classification tasks that stem from datasets [14], contextual word embeddings [22], distributed word embeddings [4, 16], machine learning algorithms [9, 42], and human annotators [18]. In a pioneering work [4], word embeddings trained on Google News articles were found to exhibit gender stereotypes to an alarming extent. Yet, only a handful of studies [9, 17, 42] have focused on mitigating these unintended biases in text classification, broadly, and toxicity detection, specifically. For example, a recent survey [43] identified and mitigated three types of biases in toxicity detection: identity (e.g., “gay”), swear words (e.g., “f*k”), and racial biases (e.g., AAE). One approach for mitigating bias in text classification—and mitigating demographic bias, in particular—is data augmentation [14, 27, 34]. This approach seeks to reduce data bias stemming from the lower weight and/or under-representation of minority (relative to majority) groups by balancing the training data sets. Specifically, one can add external labeled data [14], swap gender-related terms [27], or assign different weights to instances from various groups [23]. The primary drawback of these data manipulation methods is their impracticality (e.g., costliness of labeling data).

Recent work by Zhang et al. [42] sought to address these limitations. The authors assumed that there are discriminative and non-discriminative data distributions and sought to reconstruct the non-discriminative data distribution from discriminative ones by instance weighting. Another approach formulates the task as a constrained optimization problem [17]. The basic idea is to impose a fairness constraint w.r.t. a single bias type during model training. Critically, however, prior research takes static or batch approaches, assuming all information it needs is available. Furthermore, it overlooks the unique aspects within each bias and does not distinguish the debiasing strategies for different biases.

Our work complements prior work by: (1) studying multiple potentially correlated biases with sequential input; (2) providing the first sequential bias mitigation strategy to jointly mitigate these biases; and (3) conducting an in-depth analysis of the impact of the size of historical information on sequential debiasing performance. Our findings offer new insights for both the theoretical and practical aspects of research on bias and fairness in AI.

3 PROBLEM DEFINITION

Given a corpus C of N samples, a sample i ∈ {1, 2, ..., N} can be a social media session Si (e.g., an Instagram session) consisting of a sequence of C comments {c1, c2, ..., cC}. There is also a pre-defined set of K sensitive attributes P = {P1, P2, ..., PK}. For instance, K = 2 and P = {gender, race} when the considered sensitive attributes are gender and race. A sample i can also be a comment (e.g., a tweet) comprised of a sequence of W words {w1, w2, ..., wW}. Moreover, every sample (a session or a comment) is labeled as “non-toxic” (Yi = 0) or “toxic” (Yi = 1). Let D be the number of dimensions of the extracted features xi for every comment ci in a session or every word wi in a comment. Our proposed sequential debiasing process for toxicity detection aims to learn a binary classifier that jointly mitigates the overall bias w.r.t. the set of sensitive attributes P and accurately identifies if a session or comment is toxic or not, with sequential input:

\[ F : \mathcal{P} \cup \{x_1, x_2, ..., x_L\} \in \mathbb{R}^D \rightarrow Y \in \{0, 1\}, \]

where L denotes the number of comments in a session or words in a comment. To simplify the presentation, we will use social media sessions in the Method section for illustration. Experimental results will be given for both data consisting of sessions and comments.

4 METHOD

Existing research in debiasing toxicity classifiers assumes that the model observes all the comments and then makes a one-time decision regarding the prediction and bias mitigation. However, social media users interact in a sequential manner. Conventional debiasing approaches, therefore, are less responsive when the conversations between users are changing. A desired debiasing strategy should be able to process sequentially revealed comments and make dependent decisions. In addition, prior research studied different biases either individually (i.e., debiasing one type of bias at a time) or independently (i.e., debiasing multiple biases that are independent from one another). Nevertheless, bias is complex by nature and different biases might be correlated, as we will show in Sec. 5. Therefore, it is important for the sequential debiasing strategy to identify and capture the unique aspects of various biases. In this section, we first discuss how to measure bias in the presence of multiple types of biases. Then, we detail the proposed joint bias mitigation approaches for toxicity detection via sequential decisions.

4.1 Measuring Bias

Measuring bias is key for addressing unfairness in NLP and machine learning models. This section presents two categories of bias metrics that quantify the differences in a classifier’s behaviour across a range of groups within the same identity, e.g., {female, male, other} for gender. They are the Background Comparison Metric (BCM) and the Pairwise Comparison Metric (PCM) [11].

4.1.1 Background Comparison Metric. The core idea of BCM is to compare a measure m (e.g., False Positive/Negative Rate) of a group over the sensitive attribute p with the group’s background score using the same measure m. The background score is defined based on the task at hand and the scientific questions being asked. In this study, it is defined as measure m over the overall evaluation set. We use the following common bias metrics in debiasing text classification as measure m in the toxicity classifier: False Negative Equality Difference (FNED) and False Positive Equality Difference (FPED) [14]. FNED and FPED are defined based on the False Positive Rate (FPR) and False Negative Rate (FNR). Formally, we define the BCM-based fairness metrics, FPEDCM and FNEDCM, as follows:

\[ \text{FNED}_{\text{BCM}} = \sum_{z \in \mathcal{P}} \left| \frac{\text{FNR}_z - \text{FNR}_\text{overall}}{\text{FNR}_\text{overall}} \right| \]

\[ \text{FPED}_{\text{BCM}} = \sum_{z \in \mathcal{P}} \left| \frac{\text{FPR}_z - \text{FPR}_\text{overall}}{\text{FPR}_\text{overall}} \right| \]

where z denotes the values that a sensitive attribute p ∈ P can be assigned to. For example, in case of p = {male, female, other}, the FNRz and FPRz are calculated for every group z ∈ p. They are then compared to FNRoverall and FPRoverall which are calculated...
4.2.2 Pairwise Comparison Metric. BCM allows us to investigate how the performance of a toxicity classifier for particular groups differs from the model’s general performance. When applied to settings with multiple biases, BCM can be less effective, as it tends to cancel out the differences of these distinctive biases. In addition, when a toxicity classifier presents low performance, the BCM-based metrics may underestimate the bias [11]. Here, we present PCM that quantifies how distant, on average, the performance for two randomly selected groups $z_1$ and $z_2$ within the same attribute $p$ is. This metric examines whether and to what extent the groups differ from one another. For example, given the sensitive attribute Race with three groups [White-American, African-American, Asian], we consider performance differences for White-American vs African-American, White-American vs Asian, African-American vs Asian. We formally define the PCM-based metrics as follows:

$$
FNED_{PCM} = \sum_{z_1,z_2 \in (p)} |FNR_{z_1} - FNR_{z_2}|.
$$

$$
FPED_{PCM} = \sum_{z_1,z_2 \in (p)} |FPR_{z_1} - FPR_{z_2}|.
$$

In both Eq. 4 and Eq. 5, we measure the difference between every possible pair of groups in $p \in P$. This forces the algorithm to focus on the particular aspects of this sensitive attribute, which otherwise will be averaged out in the overall population.

4.2 Sequential Bias Mitigation

When comments come in a sequence, a toxicity classifier needs to make decisions based on incomplete information, i.e., comments observed so far. The current decision will, in turn, influence both future prediction results and debiasing strategies. In addition, in the presence of multiple biases, debiasing a toxicity classifier can be more challenging due to the need to capture the unique characteristics of each bias and potential correlations among biases. To tackle these challenges, in this section, we present a sequential bias mitigation approach that leverages a reinforcement learning (RL) framework that seeks to maximize prediction accuracy and minimize bias measures tailored to individual biases at each timestep.

4.2.1 Debiasing via Sequential Decision Making. As comments arrive in a sequence, a debiased toxicity classifier needs to respond in a timely manner based on partial information. This process might also involve the trade-off between debiasing and prediction accuracy. Our proposed solution is an RL framework built upon theories in sequential MDP, which allows learning to trade off competing objectives in a principled way [19]. It considers two tasks at each state of the decision-making process: (1) predicting whether the session is toxic and (2) minimizing the total amount of biases. In a typical RL framework, an agent $A$ interacts with the environment over time. At timestep $t \in \{1, 2, ..., T\}$, $A$ chooses an action $a_t$ in response to the current state $s_t$, which causes the environment to change its state and returns the reward value $r_{t+1}$. Formally, we represent every interaction as an experience tuple $M_t = (s_t, a_t, s_{t+1}, r_{t+1})$ used to train $A$.

In sequential bias mitigation for toxicity detection – *Joint*. The agent is a biased toxicity classifier that takes an action $a_t$ (i.e., predicting the label) based on the current state $s_t$ (i.e., the comments observed so far). By maximizing the reward value returned by the reward function – consisting of the bias measures and prediction error – the biased classifier is forced to improve the prediction performance and reduce the biases on the selected session from the evaluation set, which is a subset of the training set.

In sequential bias mitigation for toxicity detection, the environment includes all of the training sessions and $A$ is a biased toxicity classifier $F$. State $s_t$ is a sequence of $t$ comments $A$ has observed so far. $A$ selects an action $a \in \{\text{toxic, non-toxic}\}$ based on an action-selection policy $\pi(s_t)$, which outputs the probability distribution over actions based on $s_t$, $\pi(s_t, a_t)$ represents the probability of choosing action $a_t$ when observing $t$ comments, i.e., $s_t$. After selecting $a_t$, the environment returns a reward value $r_{t+1}$ based on the state-action set $(s_t, a_t)$. The reward values defined by the toxicity prediction error and bias metrics are then used to calculate the cumulative discounted reward $G_t$ (i.e., the sum of all rewards received so far) and optimize the policy $\pi(s_t)$. At each state $s_t$, the RL framework maximizes the expected cumulative reward until $t$ to force the agent to improve accuracy and mitigate bias. Essentially, the agent is making dependent decisions to adjust to the sequential input. Fig. 2 depicts an overview of the RL framework.

4.2.2 Reward Function. As the key element in the RL framework, the reward function is designed to assess the performance of the classifier $F$ and jointly mitigate various types of biases over time. BCM-based bias metrics seek to compare the performance measure $m$ of a specific group $z$ (e.g., FNR$_z$) with that over the entire population (e.g., FNR$_{overall}$), resulting in a solution that might cancel out the differences of biases. However, as biases are complex (e.g., correlated) and often defined within different contexts, it is important to distinguish the unique aspects of various biases. Therefore, we hypothesize that PCM is more appropriate for reducing multiple biases because it compares $m$ of every pair of groups belonging to the same sensitive attribute $p$ instead of across all attributes in $P$. For example, when $P = \{\text{race, gender}\}$, PCM requires that the FNR of all subgroups in gender be similar while BCM seeks for similar FNR across all groups in both gender and race. With PCM, we only consider sessions related to a certain bias type instead of all of the
Algorithm 1 The Optimization Algorithm for Joint

Input: The dataset \( \{D, \mathcal{P}\} \) with labels \( y \in Y \), discount rate \( \gamma \), bias importance values \( a_i \in \Gamma \), learning rate \( \eta \), number of episodes \( E \), terminal time \( T \).

Output: Debiased classifier/agent (\( A \))

1: Initialize memory \( M \)
2: Initialize agent \( A \) with parameters \( \theta \) and \( \pi_0 \)
3: while Episode \( e < E \) do
4: Initialize \( s_0 \) by selecting a random session
5: for \( t \in \{0, 1, \ldots, T\} \) do
6: \( A \) selects an action \( a_t \) according to \( \pi(s_t) \)
7: \( s_t \leftarrow s_t + (s_t, a_t, r_{t+1}, s_{t+1}) \)
8: for each timestep \( t \), reward in \( M_e \) do
9: \( G_t \leftarrow \sum_{t'=1}^{T} \gamma^{t-t'} \cdot r_{t+1} \)
10: end for
11: Calculate the policy loss using
12: \( \mathcal{L}(\theta) = \log(\pi_0(s_t, a_t) \cdot G_t) \)
13: Update the agent \( A \) using \( \Delta \theta = -\eta \nabla_{\theta} \mathcal{L}(\theta) \),
14: end for
15: end while

sessions. We thus propose to use PCM-based metrics to capture the unique information of \( p \).

Formally, we describe the RL framework by defining the environment, the state, the action, and the reward function. The environment contains the training dataset \( D \) in which every session includes a sequence of comments. At \( t \), the environment randomly selects a session and passes the first \( t \) comments of that session to the agent \( A \), which is a toxicity classifier \( F \) that outputs a decision probability \( q_t \). We convert \( q_t \) into an action \( a_t \) using the following criterion:

\[
    a_t = \begin{cases} 
        \text{toxic} & \hat{q}_t \geq 0.5 \\
        \text{non-toxic} & \hat{q}_t < 0.5,
    \end{cases} 
\]

Finally, we define the reward function using PCM-based bias metrics to jointly evaluate various types of biases as follows:

\[
    r_{PCM}^t = -1 - \sum_{S_p \in S} a_{p_i} \cdot \frac{1}{|S_p|} \sum_{z_{1:2} \in \mathcal{S}_{p_i}} [\text{FPR}_2^t - \text{FNR}_2^t] \]  

where \( r_{PCM}^t \) denotes the binary prediction loss (e.g., log loss) of the toxicity classifier \( F \), \( a_{p_i} \) represents the importance value of bias related to the sensitive attribute \( p_i \), and \( \mathcal{S}_{p_i} \) denotes the sessions with sensitive attribute \( p_i \).

Similarly, the reward function using BCM-based bias metrics can be defined as follows:

\[
    r_{BCM}^t = -1 - \sum_{S_p \in S} a_{p_i} \cdot \sum_{z_{1:2} \in \mathcal{S}_{p_i}} [\text{FPR}_2^t - \text{FNR}_2^t] \]  

4.2.3 Optimization Algorithm. We aim to learn an optimized action-selection policy \( \pi(s) \) that maximizes the cumulative rewards based on Eq. 7-8. We consider the agent as a neural network with weights \( \theta \) (e.g., a trained recurrent neural network). We use an optimization algorithm similar to [9] to force a biased toxicity detection model to converge to a more equitable solution. We denote the two sequential debiasing models as \( \text{Joint}_g \) and \( \text{Joint}_p \), respectively. Algorithm 1 shows the high-level training process of the \( \text{Joint} \) model\(^\dagger\). During each episode, the environment selects a random session and returns the first \( t \) comments in every step. Considering the observed comments as state \( s_t \), the agent \( A \) selects an action according to \( \pi(s_t) \). To select the action according to the action-selection policy, we use the multinomial distribution to sample from the action probabilities \( \pi(s_t, a_t) \). The performed action results in reward \( r_{t+1} \) and state \( s_{t+1} \). We use experiences \( M \) to calculate the cumulative reward \( G_t \).

finally, we update the agent’s parameters using the following loss function and its gradient:

\[
    \mathcal{L}(\theta) = \log(\pi_0(s_t, a_t) \cdot G_t), \quad \Delta \theta = \eta \nabla_{\theta} \mathcal{L}(\theta). \]

5 EXPERIMENTS

To test the proposed hypotheses for sequential bias mitigation in toxicity detection, we run experiments on two benchmark datasets to answer the following research questions:

\textbf{RQ. 1.} With multiple types of bias present in toxicity detection, will different biases \textit{tend to be correlated} with each other?

\textbf{RQ. 2.} If ‘Yes’ to \textbf{RQ. 1.}, will a sequential and joint bias mitigation strategy tailored to individual biases outperform conventional static and generic debiasing approaches?

\textbf{RQ. 3.} How do the size of the historical information and parameter \( \alpha \) influence the performance of the sequential debiasing strategy?

5.1 Data

We use two publicly available datasets collected from two platforms: Jigsaw and Instagram. They differ on data format, studied bias types, sample size, and overall proportion of toxic samples. Particularly, the Jigsaw dataset consists of comments, in which words are observed over time; and the Instagram dataset consists of sessions, in which comments come in a sequence. The data statistics are shown in Table 1. Note that compared to Instagram data, each demographic group in the Jigsaw data has a much smaller portion of positive instances. We detail the two datasets below.

- **Jigsaw** [37]: The Perspective API’s Jigsaw dataset\(^3\) consists of comments – extracted from the Civil Comment platform – with toxicity and identity annotations. We consider gender and race as the sensitive attributes due to the relatively small number of comments associated with other identities.

- **Instagram** [20]: Instagram is a top-ranked social networking site with the highest percentage of users reporting experiences of cyberbullying [37]. Each Instagram sample is a social media session comprised of a sequence of comments in temporal order. As this dataset has no annotated identities (i.e., sensitive attributes), we detail the process of identifying potential attributes regarding swear words and dialect in the Experimental Setup subsection.

\(\dagger\)The source code and data can be found at https://github.com/GitHubLuCheng/DebiasTD via Sequential Decisions.

\(\ddagger\)https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification
Table 1: Statistics of the Instagram and Jigsaw datasets: percentages of every sensitive attribute and the proportion of toxic samples within each group.

<table>
<thead>
<tr>
<th></th>
<th>Group Type</th>
<th>Swear Words</th>
<th>Dialect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Present</td>
<td>Not Present</td>
</tr>
<tr>
<td>Instagram</td>
<td></td>
<td>34.80%</td>
<td>65.19%</td>
</tr>
<tr>
<td></td>
<td>% toxicity</td>
<td>15.14%</td>
<td>13.79%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Group Type</th>
<th>Gender</th>
<th>Race</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Female</td>
<td>Male</td>
</tr>
<tr>
<td>Jigsaw</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>% data</td>
<td>12.92%</td>
<td>83.06%</td>
</tr>
<tr>
<td></td>
<td>% toxicity</td>
<td>1.87%</td>
<td>8.56%</td>
</tr>
</tbody>
</table>

Figure 3: Instagram: The total biases (FPED + FNED) w.r.t. swear words and dialect. The x-axis denotes the number of comments (t) the agent has observed. SOTA-D in (a) shows the total bias w.r.t. Swear Words in the Dialect-debiased SOTA.

Figure 4: Jigsaw: The total biases (FPED + FNED) w.r.t. gender and race. The x-axis denotes the number of words (t) a model has observed. SOTA-R in (a) shows the total bias w.r.t. Gender in the Race-debiased model.

5.2 Experimental Setup

We consider two commonly studied types of bias with the Instagram dataset: swear words bias and dialectal bias. For swear-words-related bias, we used a set of predefined toxic keywords suggested in the psychology literature [26, 36]. A comment containing these terms is labeled as 1, otherwise as 0. To infer the dialect of a comment, we employed a lexical detector of words associated with AAE or WE [3], as used in previous work studying racial bias [13, 34].

5.2.1 Baselines. As methods such as data augmentation are not suitable for sessions with a sequence of comments and sensitive attributes with multiple groups, we consider the following baselines:

- **Biased Models.** These are standard machine learning models commonly used for toxicity classification. We consider the hierarchical attention network (HAN) [40] and the popular commercial model Google Jigsaw’s Perspective API. As Perspective only works on a single comment, to assign the toxicity label for each Instagram session, we first use Perspective to label every comment in the session. The session label predicted by Perspective is then the majority vote of the comments’ labels.
- **Debiasing with Fairness Constraints.** The Constraint model [17] is a debiased toxicity detection model that imposes fairness constraints of a single bias type on standard classifiers.
- **Debiasing Models for Sequential Data.** This is the state-of-the-art sequential bias mitigation model (SOTA) for cyberbullying detection [9]. SOTA is built upon an RL framework in which
the reward function considers both the predictive error and the harmonic mean of bias amount measured by FPED and FNED.

For fair comparison, we further extend the BCM-based Constraint into PCM-based (denoted as Constraintp) and use HAN as the backbone model of Constraint, Constraintp, SOTA, and Joint. Also, Constraint, Constraintp, and SOTA are individual debiasing methods, therefore, they have several variants when considering multiple biases. For instance, when both gender and racial biases are present, there are SOTA-gender (SOTA-G, i.e., gender-debiased SOTA) and SOTA-race (SOTA-R, i.e., race-debiased SOTA). Detailed parameter settings of all approaches can be found in Appendix A.

5.2.2 Evaluation Metrics. Evaluations in bias mitigation for toxicity detection typically focus on two aspects: prediction accuracy and bias removal. We adopt standard metrics for binary classification, including Precision (Prec.), Recall (Rec.), F1, and Accuracy (Acc.). Following [14, 17], we use FPED, FNED, and total bias (FPED+FNED) to assess debiasing effectiveness. An effective model should present low bias and high prediction performance. For all compared models, we use pre-trained GloVe word embeddings [30] and 10-fold cross validation with 80% data for training and the rest for testing. We also perform McNemar’s test to examine the statistical significance of the difference between models. Unless otherwise noted, the differences between our model and the baselines are statistically significant. We highlight all of the best results.

5.3 Results

5.3.1 Do different biases tend to be correlated? Essentially, we ask how a toxicity classifier debiased for one bias influences the results for other biases. We first show in Fig. 3-4 the total biases (i.e., FPED + FNED) of the two bias types in each dataset, respectively, at each timestep. For clear comparisons, all of the selected baselines are specifically designed for sequential data. Observe that for Instagram data, mitigating an individual bias can also reduce the other bias. For example, in Fig. 3(a), the total bias of swear words in SOTA-D (i.e., SOTA debiased for Dialect) is reduced significantly compared to that of the biased model, HAN. This finding appears less clear for Jigsaw data as shown in Fig. 4, in part due to the extremely low percentage of positive instances in each group in Jigsaw. However, a deeper investigation of separate results for FPED and FNED in Fig. 5 shows that SOTA-R presents lower FPED and larger FNED w.r.t. gender than the biased model HAN. This indicates that an individual debiasing method designed for race can help mitigate the FPED of gender but increases its FNED. Therefore, we empirically validate our first hypothesis that biases tend to correlate.

5.3.2 How does Joint fare against generic and static debiasing methods? First, we observe in Fig.3-4 that sequential and joint bias mitigation strategies with PCM (Jointp) consistently outperforms the biased model HAN and the generic debiasing approaches regarding all bias metrics over time. Further, Jointp appears to be more effective than Joint. Here, we consider both bias and classification measures of biased models (HAN and Perspective), generic and individual debiasing models (Constraint, Constraintp, and SOTA), and sequential and joint debiasing models (Jointp and Joint). Results for both Instagram and Jigsaw datasets are shown in Table 2-3.

We observe the following: First, it is challenging to reduce both FPED and FNED within the same or across different sensitive attributes due to the potential trade-off between the two bias metrics and correlations among biases. Second, Jointp significantly reduces total bias in both datasets. Regarding prediction, it achieves the best recall and F1 and competitive accuracy performance (see the last row in Table 2 as an example). By sacrificing precision, Jointp focuses more on identifying positive instances (i.e., protecting the victims), which is more desirable in toxicity detection [7]. Third, for the baselines, we again observe correlations between biases, supporting findings observed in RQ. 1. For example, in Table 2, the dialect-debiased Constraint (i.e., Constraint-D) presents the lowest swear word FNED while in Table 3, the race FNED of Constraint-G is largely amplified. Lastly, PCM-based models are more effective than BCM-based models w.r.t. both total bias mitigation and prediction performance, as we also showed in RQ. 1.

For RQ. 2, we conclude that (1) our joint debiasing strategy outperforms conventional approaches in terms of both bias mitigation and detection performance, as also shown by previous studies using sequential debiasing strategy [9]. This suggests the potential benefits of debiasing in a sequential manner; (2) PCM is a more effective measure than BCM regarding jointly mitigating potentially correlated biases. Therefore, with correlated biases, it is important to consider strategies tailored to individual biases.

5.3.3 Ablation Study. How does the size of historical information influence the sequential bias mitigation performance? First, the results in Fig. 3 show that as the models observe more comments in a social media session, the overall bias is progressively reduced. For Jigsaw in Fig. 4-5, we see an overall increased bias with more words observed. However, for Jointp and Joint, the bias measures become more stable and even decrease after the agent observes...
approximately 50 words. We believe this is partly because a comment typically contains more semantically richer information than a word and biases and semantics are inherently related. For a deeper understanding, we vary the size of the observed historical information at each timestep, i.e., “window size” $AL$. For example, $AL = 5$ means that at $t$, the agent uses the comments observed from $t - 5$ to $t$ to take an action. In this experiment, we use both datasets to examine the influence of window size on reduced total bias $\Delta$ (FPED + FNED) w.r.t. each bias type, compared to the biased model HAN. The results are shown in Fig. 6-7.

We observe that the proposed Jointp model consistently outperforms the baselines when the window size increases. Moreover, as window size increases, i.e., more historical comments are observed at each timestep, most approaches tend to remove more bias, especially for models debiased for the same target sensitive attributes. For example, in Fig. 7(a), where gender is the sensitive attribute, we can see that the difference in gender bias between SOTA-G and HAN increases. The same trend is observed in Jointp and Jointb. This observation is less clear for the results of Instagram, as shown in Fig. 6, e.g., the difference in swear word bias between HAN and all the debiasing models is comparatively stable as the window size grows. There are multiple potential reasons: (1) the number of comments in a session is significantly smaller than the number of words in a comment (see the x-axis range for the two datasets); (2) the annotation of sensitive attributes for Instagram is noisier, as it is automatically generated whilst that for Jigsaw is human-coded; and (3) words and comments provide different levels of semantic information, and the dependency among words is often stronger than comments. Future experiments are needed to test these hypotheses.

How does $a_i$ influence the performance of Joint? We examine the impact of $a_i$ which controls the importance of individual biases in PCM, as shown in Eq. 7. We use Instagram for illustration purposes. Specifically, for every bias $i \in \{\text{Dialect, Swear Words}\}$, we set $\alpha_{i,j} = 0.5$ while varying the $\alpha_i$ parameter as $\alpha_i \in \{0.0, 0.25, 0.50, 0.75, 1.0\}$. We show both the prediction and bias removal results in Fig. 8. We can see a clear trade-off between the performance w.r.t. the two biases: by increasing the weight $\alpha_i$ of bias $i$, Jointp shows an increase in bias $j$ and a decrease in bias $i$. For example, when $\alpha_{5}$ (i.e., weight for Swear Word bias) increases from 0.75 to 1.0, Jointp prioritizes the swear word bias, resulting in an increase in the dialect bias. In this figure, we observe the overall robustness of Joint to changes in $\alpha_i$ as well as the trade-off between mitigating multiple biases.

5.3.4 Case Studies. In addition to the above qualitative analyses, we show some case studies performed on Instagram data in Fig. 9. Several observations further support the previous quantitative results. In this non-bullying session, Jointp weighs less the importance of swear words as well as comments including these words, therefore, making the correct prediction. In contrast, the biased

| Table 2: Instagram: Bias and classification performance of various models, “S”=swear words, “D”=dialect. |
|---|---|---|---|---|---|---|---|
| | Bias Metrics ($) | Classification Metrics ($) |
| | FPED-S | FNED-S | FPED-D | FNED-D | Total | ACC. | Prec. | Rec. | F1 |
| Biased Models | | | | | | | | | |
| HAN | 0.0113 | 0.0664 | 0.2229 | 0.6441 | 0.9447 | 0.7800 | 0.6370 | 0.5576 | 0.5947 |
| Perspective | 0.0097 | 0.0654 | 0.2149 | 0.3865 | 0.6764 | 0.4829 | 0.3113 | 0.5708 | 0.4029 |
| Constraint-S | 0.0331 | 0.0524 | 0.1555 | 0.6675 | 0.8885 | 0.8532 | 0.7152 | 0.9089 | 0.8004 |
| Constraint-D | 0.0741 | 0.0316 | 0.1730 | 0.1170 | 0.3957 | 0.8578 | 0.7923 | 0.8994 | 0.8424 |
| Constraint-S-D | 0.0106 | 0.0638 | 0.1601 | 0.4203 | 0.6548 | 0.8465 | 0.7370 | 0.9315 | 0.8230 |
| Constraint-R-D | 0.0032 | 0.0748 | 0.1521 | 0.0797 | 0.3098 | 0.8741 | 0.7463 | 0.9217 | 0.8247 |
| SOTA-S | 0.0817 | 0.1075 | 0.1260 | 0.2214 | 0.5366 | 0.8747 | 0.7230 | 0.9190 | 0.8093 |
| SOTA-D | 0.0692 | 0.1336 | 0.0815 | 0.1869 | 0.4712 | 0.8981 | 0.9159 | 0.7737 | 0.8388 |
| Jointb | 0.0274 | 0.1265 | 0.1541 | 0.1527 | 0.4607 | 0.9017 | 0.7819 | 0.9159 | 0.8436 |
| J oin t p | 0.0021 | 0.0654 | 0.1619 | 0.0756 | 0.3050 | 0.9008 | 0.7567 | 0.9688 | 0.8497 |

| Debiased Models | | | | | | | | | |
| Jointp | | | | | | | | | |
| Jointb | | | | | | | | | |

| Table 3: Jigsaw: Bias and classification performance of various models, “G”=gender, “R”=race. |
|---|---|---|---|---|---|---|---|
| | Bias Metrics ($) | Classification Metrics ($) |
| | FPED-G | FNED-G | FPED-R | FNED-R | Total | ACC. | Prec. | Rec. | F1 |
| Biased Models | | | | | | | | | |
| HAN | 0.1231 | 0.1204 | 0.2942 | 0.4064 | 0.9441 | 0.8881 | 0.4724 | 0.4411 | 0.4562 |
| Perspective | 0.0419 | 0.0319 | 0.2011 | 0.5649 | 0.8398 | 0.4820 | 0.5693 | 0.3105 | 0.4019 |
| Constraint-G | 0.0147 | 0.0465 | 0.1941 | 0.6920 | 0.9473 | 0.8893 | 0.5102 | 0.3790 | 0.4349 |
| Constraint-R | 0.0873 | 0.0991 | 0.1669 | 0.3722 | 0.6255 | 0.8712 | 0.4946 | 0.4513 | 0.4719 |
| Constraint-S-G | 0.0135 | 0.0398 | 0.1863 | 0.4320 | 0.6716 | 0.8893 | 0.5378 | 0.3810 | 0.4460 |
| Constraint-S-R | 0.0970 | 0.0879 | 0.1135 | 0.1843 | 0.4824 | 0.8945 | 0.5421 | 0.3417 | 0.4191 |
| SOTA-G | 0.0894 | 0.1153 | 0.2862 | 0.4173 | 0.9082 | 0.8828 | 0.4793 | 0.4184 | 0.4468 |
| SOTA-R | 0.1096 | 0.1468 | 0.1788 | 0.2432 | 0.6784 | 0.8851 | 0.4900 | 0.4003 | 0.4407 |
| Jointb | 0.0519 | 0.0761 | 0.1279 | 0.2674 | 0.5233 | 0.8937 | 0.4396 | 0.4540 | 0.4495 |
| Jointp | 0.0517 | 0.0772 | 0.1198 | 0.2110 | 0.4597 | 0.8921 | 0.4667 | 0.4880 | 0.4771 |
Figure 6: Instagram: $\Delta$(FPED + FNED) w.r.t. swear words and dialect biases for different window sizes. For example, $\Delta$(FPED + FNED) of SOTA-S is the difference between the total bias of SOTA-S and that of HAN. A larger value of $\Delta$ indicates a greater reduction in bias.

Figure 7: Jigsaw: $\Delta$(FPED + FNED) of racial and gender biases for different window sizes. For example, $\Delta$(FPED + FNED) of SOTA-G is the difference between the total bias of SOTA-G and that of HAN. A larger value of $\Delta$ indicates a greater reduction in bias.

Figure 8: Instagram: Bias measures and classification performance (F1 score) using different values of $\alpha_i$.

(a) Biased model.

(b) $\text{Joint}_P$ with window size = 5.

(c) $\text{Joint}_P$ with window size = 10.

Figure 9: Case studies on Instagram. “Probability” denotes the model’s output probability of predicting the true label. Darker color indicates larger attention weight.

In summary, while most of the empirical findings in RQ. 1 and RQ. 3 suggest that the size of accessible historical information is critical for bias mitigation, future research is warranted to obtain more conclusive findings.

6 CONCLUSION

In contrast to the static and generic bias mitigation approaches in the debiasing toxicity classifier literature, this paper studies the novel problem of joint bias mitigation in the presence of potentially correlated biases with sequential input. In particular, we first empirically show that different biases tend to correlate. We then develop an effective solution that leverages the strengths of the theories
in sequential MDP and the PCM-based bias measure to maximize prediction accuracy and jointly minimize total bias. PCM forces the model to focus on the differences between various biases. Empirical evaluation with real-world datasets corroborates the effectiveness of the sequential bias mitigation approach with sequential input. Given our finding that word- and comment-level semantics impact the performance differently, future research can incorporate such hierarchical structure and mitigate biases in a hierarchical manner. Our work could also be adapted to other applications/domains to examine the generalizability of the approach and findings. Lastly, the proposed approach could benefit from additional studies about the ways semantic context influences sequential debiasing.

ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation under the grant #2036127. The views, opinions and/or findings expressed are the authors’ and should not be interpreted as representing the official views or policies of the U.S. Government.

A REPRODUCIBILITY

The compared approaches include two biased models (HAN and Prospective), three debiased models (Constraint, Constraint+, and SOTA), and the proposed joint debiasing models (Jointp and Jointg). We detail the parameter settings for each of these approaches below.

- **HAN**: We implemented the work of [40] with the parameter settings in Table 4 to pre-train the model.
- **Prospective**: We used Google’s Perspective API to gather the toxicity probabilities for every comment in a social media session in Instagram and in Jigsaw. We applied the Sigmoid function to convert the predicted toxicity probabilities to binary labels, i.e., \( \hat{y} = \text{Sigmoid}(Pr(\text{toxic}|x)) \). For Instagram, we used the majority vote to obtain the “toxic” label for every session.
- **Constraint**: We used the implementation of [17] and set the parameters of the regularizations terms to 0.005.
- **Constraint+**: We used the same implementation as Constraint, with the constraints changed to PCM-based metrics.
- **SOTA**: We used the implementation of [9]. The backbone model is a HAN following the same parameter settings as in Table 4. Parameters used in the RL framework can be found in Table 5. We set \( \beta \) – the parameter that balances between the prediction error and bias measures – to 1.
- **Joint**: We used the same backbone model (i.e., HAN) and parameter settings for training the RL framework as in SOTA. We set the alpha values as 0.5 for all bias types in Jointp and Jointg.

REFERENCES


Table 4: The details of the parameters of the HAN classifier.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Instagram</th>
<th>Jigsaw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Rate</td>
<td>3e-3</td>
<td>3e-3</td>
</tr>
<tr>
<td>Batch Size</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Word Embedding</td>
<td>GloVe</td>
<td>GloVe</td>
</tr>
<tr>
<td>Embedding Dim.</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Sentence Length</td>
<td>150</td>
<td>-</td>
</tr>
<tr>
<td>Number of Comments</td>
<td>100</td>
<td>512</td>
</tr>
<tr>
<td>Word Attention Dim.</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Sentence Attention Dim.</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Number of Epochs</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5: The details of the parameters of the RL algorithm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Instagram</th>
<th>Jigsaw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Rate</td>
<td>1e-5</td>
<td>1e-5</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Number of Episodes</td>
<td>150</td>
<td>150</td>
</tr>
</tbody>
</table>


