Unbiased Hate Speech Detection; Bringing the Bias back to the Table

Aida Mostafazadeh Davani
mostafaz@usc.edu

Abstract

Approaches for unbiased classification tend to prevent models from overly relying on preserved features of the input data. However, in the case of hate speech detection, while unbalanced distribution of the social group tokens, among the positive and negative instances, causes unintended bias in the models, it is not desirable to ignore these features due to their importance in recognizing hate speech. In this paper, we study hate speech detection as a sample problem in which the task is basically dependent on preserved features. We apply an unbiased model to a subtask of the problem – recognizing offensive language – which does not directly rely on the preserved features, and aggregate the results with mentioned preserved tokens to recognize hate speech.

1 Introduction

While explicit prejudicial behavior has social costs, social media platforms provide a safe haven for extremists to voice their prejudicial or hateful rhetoric in a way that not only minimizes social costs, but gets amplified or endorsed by communication with like-minded individuals and repetition inside a closed system (Lima et al., 2018).

Given the significant adverse psychological affects of being exposed to hateful rhetoric (cite), NLP researchers have conducted studies principally to design classification models for identifying offensive language and hate speech (Wiegand et al., 2018; Zampieri et al., 2019; Davidson et al., 2017; Kennedy et al., 2018). However, exploratory studies have discovered that classification models' behaviors are affected by the disproportionate appearance of *identity terms* in positive and negative samples of hate speech (Wiegand et al., 2019). This imbalance affects models' approach to predict labels, causes violation of equality of odds and opportunities (Hardt et al., 2016), and results in what is referred to "unintended bias" (Dixon et al., 2018).

For instance, Dixon et al. (2018) demonstrate how disproportionate appearance of the word "gay" in positive instances of toxic comments of the train set leads to higher false positive rates for comments that included this token. We define *Social Group Term* (SGT) as a term that refers to a specific group of individuals with a shared social identity (e.g., Muslim, woman, Russian) and seek to analyze the impact of SGTs in unintended bias of hate speech detection.

As opposed to other approaches for addressing the issue of imbalanced datasets and mitigating the unintended bias (Madras et al., 2018; Zhang et al., 2018a; Dixon et al., 2018), we argue that in hate speech detection, the vital association between SGTs as the source of unintended bias and the definition of hate speech obstructs employing exclusively unbiased datasets or input representations.

In this work, we provide an alternative framework for unbiased hate speech detection that is based on the following definition of hate-based rhetoric:

> *Language that intends to – through rhetorical devices and contextual references — attack the dignity of a group of people, either through an incitement to violence, encouragement of the incitement to violence, or the incitement to hatred.*

Therefore, we consider hate speech as

> *Offensive language that is directed to a person or some people because of their perceived association with a social group.*

Sample	Sentence	Hate	Offensive
1	In breaking news, Hillary Clinton is still a cold, heartless c**t.	No	Yes
2	When will they start to murder those who don't agree to convert?	Yes	Yes
3	Save the environment, recycle a liberal.	Yes	Yes

Table 1: Samples from our dataset and their labels based on being offensive or hate speech.

Table 1 shows three samples from our dataset. Sample #1 shows an offensive post, which is not based on social groups and is directed towards a person. Sample #2 shows a hate speech sample which does not include any SGTs. Sample #3 shows a hate speech sample that includes a SGT (*liberal*).

While in detecting offensive language the model should not consider any bias towards specific "social groups" mentioned in the text, it is essential to account for these terms to identify hate speech. Therefore, our model (1) identifies the offensiveness of the input sentence based on its unbiased representation, (2) detects the social group mentioned in the sentence, and ultimately (3) detects the hate speech.

2 Related Work

Recent NLP studies of hate speech on social media platforms include (1) developing typologies for hate speech based on the literature of hate and prejudice in social sciences (e.g., Waseem et al., 2017; Olteanu et al., 2018), (2) annotating social media content in order to produce labeled datasets (e.g., Davidson et al., 2017; Founta et al., 2018), (3) examining individual-level psychological precedents of using hateful language online (e.g., Hoover et al., 2019), and (4) designing NLP models for detecting this type of language (e.g., Zhang et al., 2018b).

Proposed methods for reducing the unintended bias can be categorized into two classes based on whether the approach is to remove the bias from the dataset or whether the bias is mitigated in the model.

2.1 Balancing the Dataset

Dixon et al. (2018) introduced unintended bias in a model as observing different performance on subsets of the dataset that contain particular *identity terms*. By inserting additional data containing identity terms, Dixon et al. (2018) create a dataset with balanced ratio of positive and negative labels. Even though all sentences in the document are from comments in Wikipedia Talk Page, the added data are gathered from the Wikipedia articles, making the distribution of the new training set incongruous with the original dataset.

Another approach for generating a balanced dataset is to swap the SGT in sentences, to provide an equal representation of different groups Zhao et al. (2018); Park et al. (2018). In other words, these methods repeat the same sentence by substituting their identified SGTs with terms that refer to other social groups. As discussed by Wiegand et al. (2019), a disadvantage of this approach is that it ignores the other sentence-level sources of bias.

Wiegand et al. (2019) argues that datasets that are gathered by randomly sampling from a corpus are usually sparse regarding the positive labels. That leads to data gathering based on dictionaries or topics, which causes the trained models to be biased towards specific terms. Classifiers trained on biased datasets might achieve high accuracy, which should be evaluated by testing them on other datasets. It can be concluded from the analyses conducted by Wiegand et al. (2019) that the bias in the dataset is not restricted to the identity terms and can also involve the data collection method.

2.2 Debiasing the Model

Another class of approaches consider altering or extending the model by adding an adversarial attack to the model to minimizes the model's bias. The adversarial network does not have to be as complicated as the prediction model, which renders this method suitable for being complementary to other models.

Garg et al. (2019) introduced the use of counterfactuals for examining unintended bias. Counterfactuals are sentences that are generated by substituting "specific critical tokens" with other instances to test the fairness of the model. The model's loss function is then extended to minimize the difference among the error ranges for all coun-

terfactuals generated for a specific document.

Liu and Avci (2019) assume that bias in classification is due to reliance on specific SGTs and consequently use model interpretation to measure the importance of particular SGTs in predicting the label in classification tasks. The interpreted importance of these terms is considered for defining the loss function to prevent the adoption of SGT. Along with training a model that performs as accurate as of the biased models, the learned word embeddings are shown to include less bias.

Zhang et al. (2018a) train an adversarial model by minimizing its capability for predicting the preserved features from input data while maximizing the classification accuracy. By having the adversarial network trained in parallel with the classifier, the loss function drives the hidden layers to acquire less information about the mentioned SGT.

In a similar approach, Madras et al. (2018) trains an autoencoder to learn a latent representation of the documents. The latent representation is then utilized by the classifier to predict the preserved features and by the adversarial network to identify the SGT. The network is trained to minimize the autoencoder and classifier loss jointly with maximizing the adversary loss.

The advantage of the adversary models is their flexibility to define the bias under study. By manipulating the loss function, it is possible to account for different components of fairness.

3 Data

The dataset includes $\sim 23.5k$ Gab posts, randomly selected from PushShift.io. Gab purports to be a haven for free speech and has attracted a large number of users who align themselves with far-right ideologies (Benson, 2016; Anthony, 2016). Due to the over-occurrence of hate speech in Gab compared to other platforms, a randomly collected dataset is expected to have a relatively high frequency of hate speech samples.

Trained research assistants then annotated the dataset for the presence of hate speech and offensive language based on the coding manual developed by Kennedy et al. (2018). It is worth noting that, according to this manual, all hate speech instances are considered as offensive language. However, offensive language instances that are not directed to social groups are not considered hate speech. Each post is annotated by at least three well-trained annotators and an agreement of 0.66% is achieved. The label for each post is assigned based on the majority vote of the annotators.

We divide the dataset to create train, validation and test sets including 70%, 10% and 20% of the dataset respectively. We evaluate both the baseline models and the proposed model with the test set to compare overall accuracy of the models.

We create **Adversarial Gab** dataset, an evaluation set of posts sampled from the test set that includes equal positive and negative samples of hate, each containing at least one SGT. The dataset is then extended by substituting each SGT with all other SGTs under study. The set of SGTs is produced by extending the list of identity terms from Dixon et al. (2018) using the set of synonyms from WordNet (Miller, 1995). Out of the 101 tokens in this list, 68 appear in our train set. Adversarial Gab includes $66k$ instances, 50% of which were labeled as hate when containing their original SGT.

We use a second bias evaluation dataset (**Phrase Templates** dataset) introduced by Dixon et al. (2018). This dataset is generated by inserting SGTs into templates of toxic and non-toxic phrases. The data includes $77k$ samples, 50% of which are labeled as toxic.

Moreover we test our model on the **Davidson** dataset, which includes $25k$ tweets, gathered based on a hate speech lexicon compiled from HateBase and labeled based on its offensive and hateful content (Davidson et al., 2017). Davidson et al. (2017) use a typology of hate speech similar to ours, where hate speech is defined as *language that is used to expresses hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group*.

4 Bias in Hate Speech Detection

Fair classifiers are trained to predict label $Y \in \{0, 1\}$ from input data $X \in \mathbb{R}^n$ with respect to preserved features S. The classifier should perform accurate regarding Y and fair regarding S (Madras et al., 2018). Unintended bias in hate speech detection model can be observed when model results in different error rated when the input data has specific SGTs.

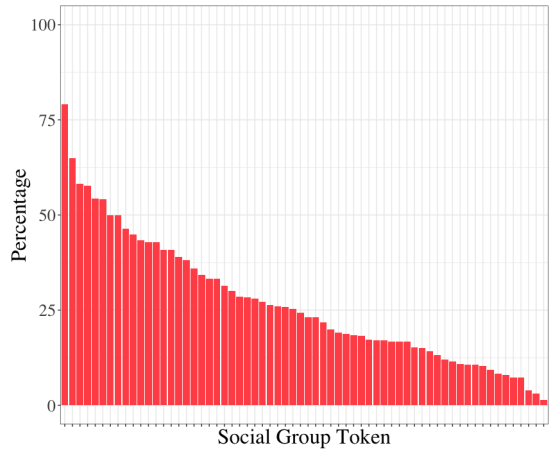


Figure 1: The proportion of each social group token appearing in positive instances of hate speech

4.1 Quantifying Bias

Based on the definition of unintended bias, we are interested in the model's performance with regards to SGTs. Unintended bias can be interpreted as high variance in false positive and false negative ratio in model's predictions on the adversarial samples that contain each SGT. In order to evaluate these error ratios we explore two different adversarial dataset introduced in section 3 and compare the false positive and false negative ratios for each SGT with overall error ratios.

Unintended bias can be observed when $Y(a_i)$ is different from $Y(Adv(a_i, s_x, s_y))$ where $s_x \in SGT(a_i)$ and $s_y \in S - \{s_x\}$. To understand the bias for each token s_y, we calculate B_y^0 as the number of times when $Y(a_i) = 1$ and $Y(Adv(a_i, s_x, s_y)) = 0$, and B_y^1 as the number of times when $Y(a_i) = 0$ and $Y(Adv(a_i, s_x, s_y)) = 1$. In other words, B_y^1 shows how many times substituting s_y in a sentence caused the model to predict it as hate.

4.2 Data Analysis

Figure 1 represents the distribution for the proportion of each SGT appearing in positive samples of Gab dataset. Ideally in a balanced dataset we expect proportions to be close to 50%, however, Figure 1 shows that for most of the SGTs, that is not the true.

We trained a vanilla LSTM model to predict the hate speech label. The model achieved F1 score accuracy of 0.64 in a 10 fold cross validation with 0.6 drop out ratio and hidden size of 512 dimensions.

SGTs with most unbalanced representation in the annotated dataset, represented in Figure **??**, are also among the most biased ones as represented in Figure 2a and 2b. This observation is in line with previous representations of bias in unbalanced datasets (Dixon et al., 2018).

5 Method

As mentioned in the Introduction, the approach we apply in this study is based on the following definition of hate speech:

> "Hate speech is a type of offensive language directed to the target based on their observed association with a specific social group."

Therefore, to identify a post as hate speech, we first need to figure out whether the post is offensive, and thereafter, whether the offense is constructed on the target's social group identity.

To decide whether a post is offensive, information about the mentioned SGTs need not be acquired (i.e., an offensive remark is offensive, regardless of the target). As we showed, the SGT information, as the delicate features of the dataset, can even cause unintended bias.

Therefore our model uses an adversarially fair representation of the sentence (Madras et al., 2018), excluding the information about SGTs, for identifying the offensive language. Importantly, this approach cannot directly be applied to hate speech detection since detecting hate speech is dependant on using the information about SGTs.

Given an input sentence X_i, the model uses a Bi-LSTM layer to generate a hidden representation $H_i \in \mathbb{R}^{d_h}$ of the input. This representation is used by different modules:

5.1 Offensive Language Classification

The offensive language label is generated by a fully connected layer and applying a sigmoid activation:

$$c_i^o = H_i \times W_i^o + b_i^o \quad (1)$$

$$o_i = sigmoid(c_i^o) \quad (2)$$

where $W^o \in \mathbb{R}^{d_h \times 1}$ and $b^o \in \mathbb{R}^1$ are parameters for the offensive language detection layer. We represent the loss function of this binary classification task as $L_{offensive}$.

5.2 Adversarial Social Group Detection

An adversarial loss function is defined to operationalize the reduction of the SGT information from hidden representation vector. This layer predicts the SGT label given H_i in an adversary manner.

$$c_i^{adv} = H_i \times W_i^{adv} + b_i^{adv} \quad (3)$$

$$adv_i = sigmoid(c_i^{adv}) \quad (4)$$

where $W^{adv} \in \mathbb{R}^{d_h \times t}$ and $b^{adv} \in \mathbb{R}^t$ (t is the number of SGT labels) are parameters for the adversarial SGT detection. We represent the loss function of this classification task as L_{SGT}.

5.3 Hate Speech Detection

Hate speech detection layer aggregated the offensive language label, extracted SGT and sentence representation to predict the hate speech label. To this end the classifier uses a representation of mentioned SGT tokens as a vector of length d_{SGT}

$$H_i^{SGT} = sigmoid(SGT \times W_i^{SGT} + b_i^{SGT}) \quad (5)$$

where SGT is a binary vector of length t and $SGT_i = 1$ if the ith SGT is mentioned in the post. $W^{SGT} \in \mathbb{R}^{t \times d_{SGT}}$ and $b^{SGT} \in \mathbb{R}^{d_{SGT}}$.

We use the concatenation of sentence representation and SGT representation ($H_{hate} = [H_i, H_{SGT}]$) to predict the hate label.

$$c_i^h = H_{hate} \times W_i^h + b_i^h \quad (6)$$

$$h_i = sigmoid(c_i^h) \quad (7)$$

where $W^h \in \mathbb{R}^{d_h + d_{SGT} \times 1}$ and $b^h \in \mathbb{R}^1$ are parameters for the hate detection. We represent the loss function of this classification task as L_{hate}. The general loss function for the debiased offensive detection model is calculated as:

$$L_{de_offensive} = L_{offensive} - L_{SGT} \quad (8)$$

6 Experiment

The model is implemented in TensorFlow 1.14.0, the hyperparameters are set as d_h = 256, d_s = 32. The model was trained for 50 epochs.

Since the dataset has a relatively small ratio of positive labels, we generated weighted batches for training the model. In doing so, we restricted the batches to have at least 10% positive labels. This balanced batch generation is also performed in the basic model to prevent additional bias.

In order to train the adversarial loss function, we optimize the $L_{de_offensive}$ for 50 epochs first. In every other epoch, we either minimize the loss function by minimizing $L_{offensive}$ or by maximizing L_{SGT}. The sentence representation vector (H_i) is relaxed during the adversarial training for maximizing L_{SGT}.

Method	Dataset	Hate	Offensive
LSTM	Gab-test	64.3	71.0
Debiased	Gab-test	41.2	53.2
LSTM	Davidson	73.6	88.7
Debias	Davidson	31.4	89.6

Table 2: F1 score of the vanilla LSTM model and the debiased model on two datasets labeled based on offensive language and hate speech

Method	Evaluation Data	Hate	Offensive
LSTM	Gab-Adv	69.1	NA
Debiase	Gab-Adv	72.2	NA
LSTM	Templates	NA	69.7
Debias	Templates	NA	76.1

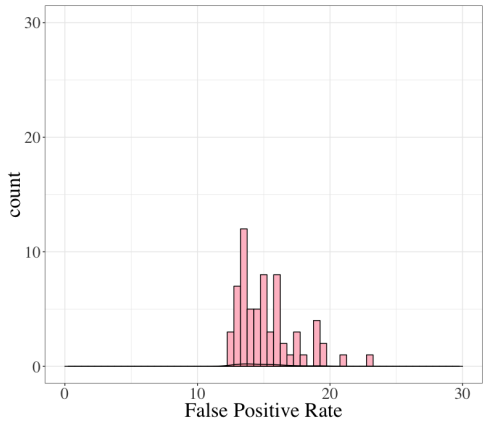
Table 3: Evaluation of the vanilla LSTM and debiased models trained on the Gab dataset on the evaluation datasets

7 Results

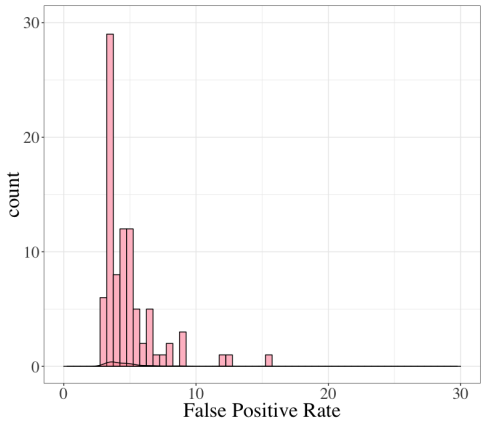
Table 2 shows the F1 accuracy score for predicting hate using the vanilla LSTM model or the debiased model on different Gab and Davidson datasets. As Table 2 shows, the debiasing model has a significantly lower F1 score comparing to the vanilla LSTM model. Although we expected a decrease in the F1 score, that is predictable based on the pre-existing bias in the dataset, it is essential to improve the accuracy score of the model with regulations.

Table 3 shows the results of evaluating the trained models on evaluation datasets (**Adversarial Gab** and **Phrase Templates**). Since the **Phrase Templates** only include offensive language labels, we evaluate the models on it to assess the reliability of the debiased offensive language detection module.

We also evaluate each method based on the number of false positive and false negative ratios for each SGT on the **Adversarial Gab** dataset. As represented in Figure 3 and 2 the evaluation

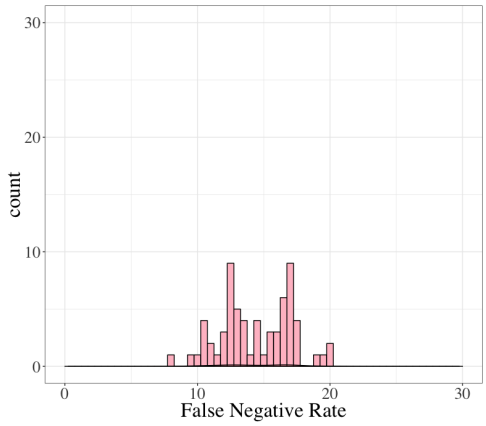


(a) Vanilla LSTM

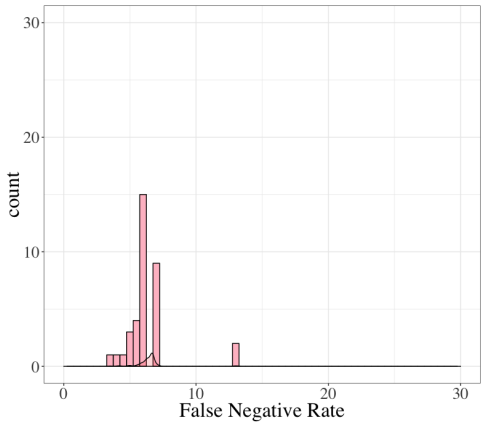


(b) Debiased Model

Figure 2: False negative ratio associated with SGTs. The y axis show the false negative ratio, the x axis shows the number of SGTs that are associated with a specific false negative rate.



(a) Vanilla LSTM Model



(b) Debiased Models

Figure 3: False positive ratio associated with SGTs. The y axis show the false positive ratio, the x axis shows the number of SGTs that are associated with a specific false negative rate.

of unintended bias, shows that the false positive and false negative ratios associated with specific SGTs been decreased after applying the debiasing approach. The figures essentially show that the the number of SGTs that are associated with high false positive and false negative ratios have decreased after performing the debiasing. Consequently, the empirical result is that no specific SGT is now associated with hate speech.

Running a t-test to assess the different between the association between SGTs and false positive and false negative errors shows that the debiased model results in significantly different error ratios comparing the to vanilla model (for both t-test $p < 0.05$).

8 Discussion

Most methods for mitigating bias in classification provide general approaches that can be applied to either balance the datasets or prohibit models from overly relying on the bias in data. However, we mitigate bias in hate speech detection by construsting a model based on the definition of hate speech. This approach can be applied to cases in which the source of bias is not to be excluded from the classification process based on the specific definition of the task under study.

The results show that the on order to achieve comparable results with the current model, we need to apply more regulation. Nevertheless, it should be mentioned that the high F1 score achieved by the vanilla model depends on how the model is trained in association with the existing bias and a debiased model cannot necessarily per-

form as well. On the other hand, the bias evaluation results demonstrate how the association between SGTs and hate speech token has been partly eliminated from model behavior.

References

A Anthony. 2016. Inside the hate-filled echo chamber of racism and conspiracy theories. *theguardian.com*.

T Benson. 2016. Inside the "twitter for racists": Gab — the site where milo yiannopoulos goes to troll now. *Salon.com*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

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*, 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 *Twelfth International AAAI Conference on Web and Social Media*.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226. ACM.

Moritz Hardt, Eric Price, Nati Srebro, et al. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323.

Joseph Hoover, Mohammad Atari, Aida Mostafazadeh Davani, Brendan Kennedy, Gwenyth Portillo-Wightman, Leigh Yeh, Drew Kogon, and Morteza Dehghani. 2019. Bound in hatred: The role of group-based morality in acts of hate. *PsyArXiv*.

Brendan Kennedy, Drew Kogon, Kris Coombs, Joseph Hoover, Christina Park, Gwenyth Portillo-Wightman, Aida Mostafazadeh Davani, Mohammad Atari, and Morteza Dehghani. 2018. A typology and coding manual for the study of hate-based rhetoric. *PsyArXiv*.

Lucas Lima, Julio CS Reis, Philipe Melo, Fabricio Murai, Leandro Araujo, Pantelis Vikatos, and Fabricio Benevenuto. 2018. Inside the right-leaning echo chambers: Characterizing gab, an unmoderated social system. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 515–522. IEEE.

Frederick Liu and Besim Avci. 2019. Incorporating priors with feature attribution on text classification. *arXiv preprint arXiv:1906.08286*.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. 2018. Learning adversarially fair and transferable representations. *arXiv preprint arXiv:1802.06309*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Alexandra Olteanu, Carlos Castillo, Jeremy Boy, and Kush R Varshney. 2018. The effect of extremist violence on hateful speech online. In *Twelfth International AAAI Conference on Web and Social Media*.

Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing gender bias in abusive language detection. *arXiv preprint arXiv:1808.07231*.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. *arXiv preprint arXiv:1705.09899*.

Michael Wiegand, Josef Ruppenhofer, and Thomas Kleinbauer. 2019. Detection of abusive language: the problem of biased datasets. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 602–608.

Michael Wiegand, Josef Ruppenhofer, Anna Schmidt, and Clayton Greenberg. 2018. Inducing a lexicon of abusive words – a feature-based approach. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1046–1056, New Orleans, Louisiana. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018a. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340. ACM.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018b. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European Semantic Web Conference*, pages 745–760. Springer.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*.