Minimize Unintended Bias in Toxicity Detection though a Multi-task Learning Mechanism

Jinghan Xu Jiatong Zhu Lu Chen

Abstract

Toxicity detection has long been an important task for machine learning. However, due to the presence of bias in the training datasets, some words tend to be associated with negative texts, even if they do not carry negative meaning. As a result, many existing models tend to misclassify the text with these words as toxic ones.

In this project, we design a fine-tuned BERT model based on multi-task learning with additional labels to mitigate the effects of this bias. Meanwhile, due to the scarcity of training data with additional labels, we train a tagger from existing labeling information to supplement the remaining labeling of the data, and then train the model with the expanded dataset. Our model model achieves **89.12% accuracy** with **89.89% recall**, which greatly reduces the unintended bias.

1 Introduction

Toxic texts are common in online conversations, including discrimination, assault and ridicule, etc. This urges for advanced technologies for toxicity detection and mitigation. The machine learning (ML) methods have been adapted to toxicity detection, modeling it as a binary classification task and achieving great accuracy. However, previous studies ignore the bias in training datasets: **If a word (e.g., "gay") always occur in the toxic training texts, then texts with this word tend to be recognized as toxic**, which called **unintended bias** (Figure 1). Handling the unintended bias is significant for network supervision, enabling more accurate and fine-grained texts classification and reducing the incidence of false locking and warnings of user accounts, thus improving the user experience.

Based on the completed Kaggle competition "Jigsaw Unintended Bias in Toxicity Classification" [1], our project aims to address unintended bias in toxicity. This Kaggle contest encourages participants to use extra labels to improve model performance. We use the Jigsaw dataset as the project dataset. It's worth pointing out that our goals are different with other Kaggle competitors. For the generous awards provided by sponsors, the competitors used many tricks to improve the Kaggle scores, many of which lacks a reasonable experimental design (e.g., train with unbalanced datasets, high precision but low recall). As for us, we are seeking for a more reasonable and reproducible solutions for the unintended bias instead of the high scores.

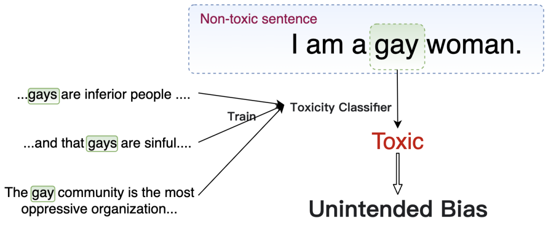
Non-toxic sentence
I am a gay woman.
...gays are inferior people ....
Train
Toxicity Classifier
...and that gays are sinful....
The gay community is the most oppressive organization...
Toxic
Unintended Bias

Figure 1: Definition of unintended bias

In our work, for better evaluating the model performance, we partition the training dataset provided by Kaggle, one part as a dataset and the other as a training set, to realize local evaluation. For the training set, we perform an equalization operation. In this task, we are not only concerned with precision, but also with recall, which we believe can better reflect the model ability for detecting unintended bias.

Our main contributions are as follows:

- **We apply a multi-task learning BERT,fine-tuned by Low-Rank Adaption (LoRA) (Hu et al., 2021), to reduce the unintended bias.** The model learns to handle a series of binary theme classification problems in training, one of them is toxicity detection. We verify that this method can greatly alleviate the unin-

[1] https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification/overview

tended bias.

- **We train a tagger to expand our training dataset.** As a result of scarce human tagged data, we train a tagger to expand our training dataset for robust model performance.
- **We achieve 89.89% recall by ensemble learning,** which means our solution to minimize the unintended bias is effective.

2 Related Work

2.1 Offensive Language Detection

Research on offensive language detection on social media has grown substantially in recent years. Previous studies have examined various types of offensive language, particularly toxic language (Borkan et al., 2019; Burnap and Williams, 2015). Several machine learning methods have been introduced to detect toxic language, with the mainstream of work using Neural Networks for toxicity classification, yielding impressive results (Georgakopoulos et al., 2018; Chen et al., 2019). Among these, the best-performing approaches include Transformers and LSTMs (Dinan et al., 2019; Kumar et al., 2018). However, some machine learning methods are criticized for biases in classification (Wiegand et al., 2019; Kirchner and Mattu, 2016), leading to unfair model decisions. Certain "social identity terms" are disproportionately labeled as toxic, indicating a false-positive bias due to the model's tendency to overgeneralize from training data (Argyriou et al., 2008).

Significant efforts have been made to mitigate unintended bias in text classification. Some researchers have proposed data balancing techniques to reduce such biases, employing additional real-world or synthetic data to balance the dataset (Reichert et al., 2020). Adversarial learning has also been used to mitigate biases in text classification (Zhang et al., 2018).

Furthermore, related research includes well-designed metrics to evaluate and quantify biases. The ROC metrics introduced by the Google Conversation AI Team (Borkan et al., 2019) provide insights into how unintended bias manifests.

2.2 Relevant Techniques

Multi-Task learning Since its invention, Multi-task learning (Caruana, 1998; Argyriou et al., 2008) has been widely used in NLP tasks. It leverages shared representations and information across multiple related tasks, improving generalization and performance by reducing overfitting and enhancing the model's ability to capture relevant features. Overall, multi-task learning can enhance model performance on various NLP tasks (Zhang and Yang, 2018). However, its effectiveness is highly dependent on the specific task and framework.

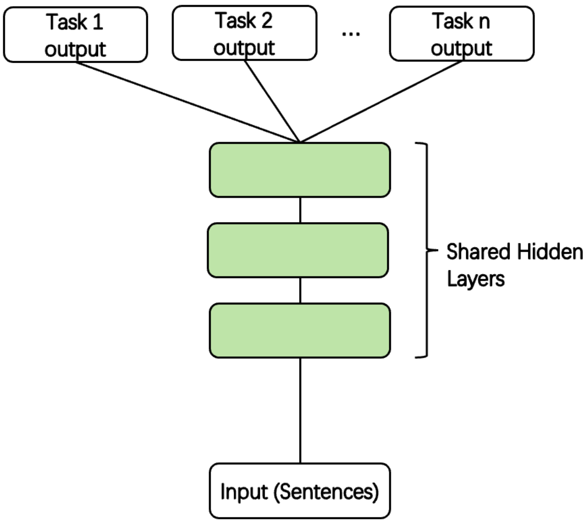

Figure 2: Multi-task Learning Mechanism

Low-rank adaptation (LoRA, see Figure 3) Fine-tuning is a common approach to adapt a large model to a specific field. However, the parameters of large models pretrained in general domains are extremely numerous. Using Low-Rank Adaptation, the pretrained weights are frozen, and only low-rank decomposition matrices are injected into the model architecture (Hu et al., 2021). LoRA can significantly reduce operational costs while maintaining model performance.

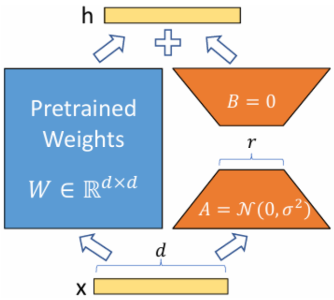
h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Figure 3: LoRA Fine-tuning Mechanism (Hu et al., 2021)

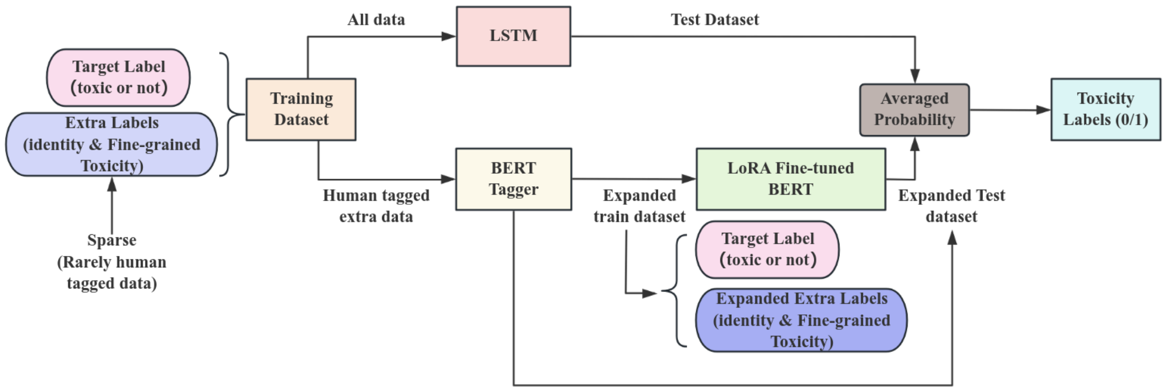
All data
LSTM
Test Dataset
Target Label (toxic or not)
Extra Labels (identity & Fine-grained Toxicity)
Training Dataset
Averaged Probability
Toxicity Labels (0/1)
Sparse (Rarely human tagged data)
Human tagged extra data
BERT Tagger
Expanded train dataset
LoRA Fine-tuned BERT
Expanded Test dataset
Target Label (toxic or not)
Expanded Extra Labels (identity & Fine-grained Toxicity)

Figure 4: Framework of our Solution

3 Problem Definition

To address the challenge of toxicity classification while minimizing unintended bias, we adopt a multi-task learning approach leveraging additional human-labeled annotations (provided by Kaggle). The primary goal is to develop a robust model capable of accurately categorizing toxic contents, while mitigating biases that may arise due to cultural, linguistic, or demographic factors present in training data.

Traditional toxicity classification models often struggle with detecting nuanced forms of toxicity or exhibit biases towards certain demographic groups or types of content. These biases can lead to inaccurate predictions and potentially harmful outcomes, such as unfair moderation or reinforcement of existing biases in online discourse.

In this study, we define our problem as follows:

Toxicity Classification : Given a text input, classify it as toxic or non-toxic.

Multi-Task Learning : Utilize additional labeled data from related tasks (e.g., sentiment analysis, topic classification) to enhance the toxicity classification performance.

Bias Minimization: Develop techniques to minimize unintended biases present in the training data and ensure the model's predictions are fair and unbiased across different subgroups.

Our approach aims to achieve superior performance in toxicity classification while addressing biases through the integration of multi-task learning and bias mitigation strategies. By leveraging diverse sources of annotated data, we intend to build a model that is not only accurate but also robust against biases prevalent in online content moderation.

4 Methodology

4.1 Framework

Our solutions basically includes 2 part: The fine-tuned BERT for multi-tasking, and the tagger to enhance overall quality for the dataset (See figure 4).

4.1.1 BERT Tagger

Among the 1,804,874 comments in the training set, only 405,130 are annotated for their identities, and only 534,630 are annotated for the additional toxicity labels. This large number of untagged comments limits the performance of our multi-task learning model.

To address this, we fine-tuned a BERT-base-uncased model as a multi-task tagger. We used the annotated comments and their identity labels as the training set to fine-tune the full parameters of the BERT pretrained model. This model was then used to predict the labels for the remaining 1,500,000 unannotated comments.

Since the comments with identity labels differ from those with aux-labels, we followed a similar approach to fine-tune another BERT-base-uncased model using the toxicity-labelled comments. This second model was used to predict the labels for the remaining 1,400,000 comments without toxicity labels.

Table 1 shows the effectiveness of the tagger. Although some comments clearly reflect certain identities, the identities are not shown due to the lack of annotation. With the tagger, reasonable labels are generated.

Table 1: Example label made by Bert Tagger

Comment Text	Predicted Homosexual, Gay or Lesbian
They should be, but when they encourage bullying of gay youth and their own ennui and that leads to suicide, the intent is missed.	0.976087
Most couples, gay and straight, talk about what's for dinner.	0.975312

4.1.2 Fine-tuned BERT

After tagging, we then fine-tuned the BERT for the multi-labeling problem. In training stage, we care about the BCE loss across all the labels, and the model update itself with the overall loss. But in reasoning stage, we only concentrated on first labeling problem (i.e., "toxic" or "non-toxic"), and the accuracy, recall and ROC are all measuring the only problem.

We've ever tried to fine-tune all the parameters in BERT, but that cost a lot of computational resources and didn't achieve great model performance (hard to convergence). Thus we use LoRA to **modify only the parameters for Values in transformer Q, K, and V matrices , and also the linear layers**, which account for only 0.08% of total parameters.

4.2 Evaluation

We apply many traditional ML metrics for evaluation in our task, including **Accuracy, Recall, and Area under Curve (AUC)**.

$$Recall = \frac{\#\{truly\,toxic\,and\,predicted\,toxic\}}{\#\{predicted\,toxic\}}$$

We prioritize recall over accuracy, for the reason that recall evaluates the proportion of truly toxic samples correctly identified among all predicted toxic samples. **Recall is more consistent with the requirement to minimize unintended bias** : Enabling models to take responsibility for text they judge to be toxic. Thus the neural words are tend to be held as non-toxic.

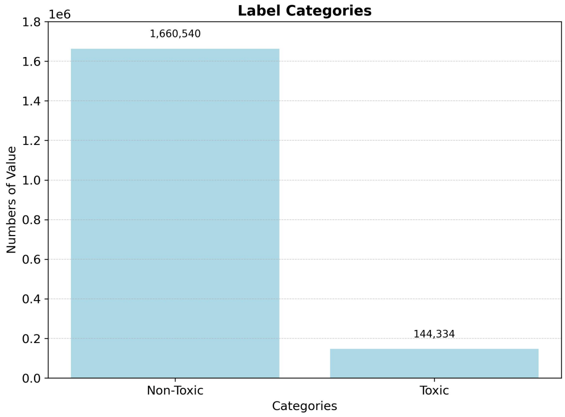

Figure 5: The number of toxic or non-toxic comments

5 Experiment

5.1 Dataset

In our work, we use the dataset from the Kaggle competition, which was published by Google and Jigsaw in 2019. The dataset contains 1,804,874 comments from the Civil Comments platform. The comments are annotated by the Civil Comments platform, and Jigsaw extended the annotations with additional labels provided by human raters. Each comment is assigned a toxicity label based on the percentage of annotators that categorize it as toxic. For evaluation purposes, comments with a target value greater than or equal to 0.5 are considered toxic.

Each comment also comes with a variety of identity labels (e.g., male, female, homosexual, Jewish) and five subtypes of toxicity labels: severe toxicity, obscene, threat, identity attack, and insult. The details of the dataset are shown in Table 7. Figure 5 demonstrates that the distribution of toxicity labels is highly unbalanced, with over 92% of comments classified as non-toxic.

During training, we split the original training set into training and test sets at a ratio of 9:1 for local evaluation.

In real use, we have equalized the dataset to balance the number of toxic and non-toxic examples. **The models trained on the equalized dataset get worse performance in accuracy, but there is a significant improvement in recall (Table 2)**. However, to get better Kaggle scores, many previous participants didn't adopt the equalizing method. We believe that training after equalizing the dataset is the only way to really improve the model's ability to handle bias, rather than just achieving higher scores.

Table 2: Results of simple LSTM (without multi-task learning) on different datasets.

Dataset	accuracy (%)	recall (%)
Equalized	88.71	**86.93**
Unequalized	**95.43**	76.54

5.2 Experimental Settings

We leased GPUs on the AutoDL platform for model training. An RTX 4090 with 24GB of RAM was used for training.

5.3 Pre-Experiment and Baseline

5.3.1 Long Short-Term Memory (LSTM)

We start with the LSTM, which has a good performance in the toxicity classification task, and verify the positive impact of multi-task learning and integrated learning on the task results on the LSTM.

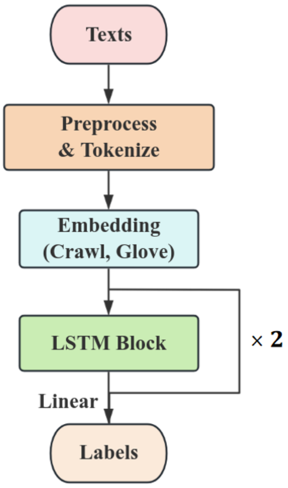

Figure 6: Structure of LSTM

To facilitate effective training, we use the concatenation of 2 pre-trained embeddings (i.e., crawl-300d-2M & glove.840B.300d) as the input of LSTM. See Figure 6 for the structure of our LSTM. The output label is the probability of "belonging to the label".

Table 3 are the results for multi-task learning and ensemble learning. For multi-task learning, we take advantage of 7 extra labels. Even though there are many default values in the labels (i.e., data not labeled by the human tagger), learning with these labels can lead to better model performance. For ensemble learning, we compare the performance of single model and the combination of two LSTMs, where we take the average of their output probabilities. Our experimental results demonstrate the effectiveness of multi-task learning and ensemble learning in this task.

5.3.2 Logistic Regression (LR)

As a complement, we explored simpler models (i.e., Logistic Regression) as well, and set it up as our baseline. We also use the two pre-trained embeddings as model input, take the mean of all the words embedding in the text as the text embedding, and put them through a linear layer and a Sigmoid layer. The results are also present in Table 3.

5.4 Results

5.4.1 Selection for Low-Rank Parameterization Target Modules in LoRA Model

In the parameters of the LoRA model, the most critical parameter is target_modules, which specifies the modules in the LoRA model that require low-rank parameterization. The `target_modules` include the following parameters:

query: The query matrix, used in the attention mechanism to compute the query vector representations.

key: The key matrix, used in the attention mechanism to compute the key vector representations.

value: The value matrix, used in the attention mechanism to compute the value vector representations.

classifier.bias: The bias term for the classifier, used as the bias parameter in the linear classifier.

classifier.weight: The weight parameter for the classifier, used as the weight matrix in the linear classifier.

dense: The dense layer, referring to fully connected layers or densely connected parameters in the model.

The impact of applying low-rank parameterization to the query, key, and value parameters is crucial. Therefore, prior to conducting experiments on different taggers, we adjusted various combinations of them to identify the optimal training results. The specific combinations and results are shown in Table 5. Based on these results, we decided to use the strategy that yielded the appropriate accuracy and the highest recall in subsequent experiments, which involves applying low-rank parameterization only to the value parameter.

Table 3: Results of LSTM and LR on equalized dataset

Model	accuracy (%)	recall (%)
LSTM with multi-task	89.23	87.63
LSTM w.o. multi-task	88.71	86.93
ensemble LSTM with multi-task	**90.12**	**88.25**
ensemble LSTM w.o. multi-task	89.59	88.07
LR w.o. multi-task (**baseline**)	82.25	79.81
ensemble LR w.o. multi-task	82.32	80.00

Table 4: Results of LoRA fine-tuned BERT (After trained 4 epoch)

Model	accuracy (%)	recall (%)
No tagger	85.70	88.07
identity-tagger	85.55	87.81
auxlabel-tagger	84.05	**89.41**
all-tagger	85.61	87.41

Table 5: Parameters Applying Low-rank Parameterization Result After 1 epoch)

Parameter(s)	accuracy (%)	recall (%)
query	84.73	83.27
key	84.69	82.85
value	**84.03**	**89.79**
query, key, value	85.42	87.61

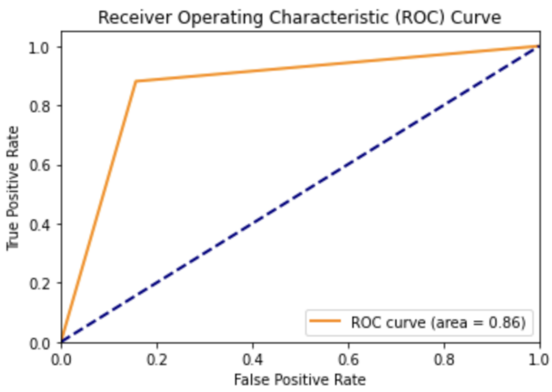

Figure 7: Receiver Operating Characteristic (ROC) Curve

5.4.2 Results of LoRA fine-tuned BERT

We classify the extra labels into two categories, aux-label and identity-label, based on their characteristics. These two types of labels are trained using the BERT Tagger to generate respective labelings. Subsequently, we form four datasets by combining these labels in different ways. Fine-tuned BERT is then trained on each of the four datasets separately. The training models, along with the obtained accuracy and recall data, are presented in Table 4, and the ROC curves are shown in Figure 7 (taking LoRA fine-tuned BERT with aux-label as an example)

Based on the findings in Table 4, we observe a 1% improvement in recall with the aux-label tagger. Given our primary focus on recall performance, this suggests that aux-labels, which provide a more detailed breakdown of toxicity categories, are effective in minimizing unintended bias. The AUC for this tagger is also 0.86, demonstrating strong performance.

However, in the identity-tagger and all-tagger datasets, the role of the tagger is not well presented. We believe that the unsatisfactory performance of the tagger can be attributed to several reasons. Firstly, we observed that the tag values predicted by the tagger are generally small, which may not significantly improve the classification criteria. Additionally, an analysis of the dataset revealed that the quantity of data with identity-labels and aux-labels is much lower than that of the prediction set. Furthermore, the BERT model used to train the tagger was fully fine-tuned without freezing any layers, and the loss during the tagger training process was approximately 0.014. This suggests potential overfitting and poor generalization ability during training, leading to the tagger's suboptimal performance on the prediction set. Consequently, this affected the accuracy of the LoRA fine-tuned BERT, resulting in no significant improvement or even a slight reduction in accuracy. Due to these reasons, we failed to achieve the results in (Vaidya et al., 2020)

5.5 Hyperparameters

5.5.1 LoRA Fine-tuning Parameters

LoRA fine-tuning enhances the pretrained BERT model by introducing specific enhancements through configurable parameters.

r: Set to 4, specifying the low-rank dimensionality for reducing attention matrix dimensionality.

lora_alpha: Set to 8, scaling LoRA activations relative to pretrained weight activations to maintain model fidelity.

lora_dropout: Set to 0.1, controlling dropout probability within LoRA layers to mitigate overfitting during training.

5.5.2 Training-Related Hyperparameters

Training incorporates parameters crucial for optimizing model performance across multiple epochs.

Learning Rate: Fixed at 5e-4, governing step size in model parameter updates during training.

Batch Size: Configured at 8 for both training and evaluation batches per device, balancing computational efficiency and model convergence.

No. of Training Epochs: Set to 4 epochs, determining iterative model updates and convergence towards optimal performance.

Weight Decay: Implemented at 0.01, facilitating regularization by penalizing large model weights during optimization.

6 Ensemble Learning

For our specific implementation, we selected two models with promising results: LSTM with multi-task learning and LoRA fine-tuned BERT with all-tagging. These models were integrated through ensemble learning to derive a strong learner. We separately trained and validated each model, averaging their predictions on the target labels of the validation set to obtain the ensemble prediction result:

$$y_{prediction} = \frac{y_{LSTMprediction} + y_{BERTprediction}}{2}$$

The obtained results were then compared against the ground truth labels, yielding an accuracy of 89.12%, recall of 89.89%, and an AUC of 0.89. A comparative analysis with individual models is presented in Table 6.

Table 6: Ensemble Learning Result

Model	accuracy (%)	recall (%)
LSTM	89.23	87.63
LoRA	85.61	87.41
Ensemble	**89.12**	**89.89**

We observed that the ensemble learning approach achieved a 4% higher accuracy compared to LoRA fine-tuned BERT, albeit slightly lower than LSTM with multi-task learning. Notably, the recall metric demonstrated significant improvement with the ensemble approach, surpassing both individual models by more than 2%. And the AUC metric also improved from 0.86 to 0.89.

Given our focus on minimizing unintended bias in toxicity classification, ensuring comprehensive coverage of potentially toxic reviews is paramount to avoid misclassifying them as non-toxic. Therefore, the notable enhancement in recall metrics underscores the effectiveness of our ensemble learning model.

7 Discussion and Conclusion

Through training, our model achieves recall 10% higher than baseline on the test dataset, which means that our strategy is effective. Our model tends to adopt a more conservative judgmental strategy and thus can largely avoid unintended bias.

However, we are still not perfect. The tagger doesn't achieve our expectation: It was supposed to have a larger improvement in the model's performance. We attribute it to simple training strategy and limited training resources. Besides, our training framework is step-by-step: labeling is first supplemented by Tagger, and then the classifier is trained. It is not an end-to-end training mode, which may also lead to some bias in the tagger, making it not highly aligned with our training object.

References

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2008. Convex multi-task feature learning. *Machine learning*, 73:243–272.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500.

Pete Burnap and Matthew L Williams. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & internet*, 7(2):223–242.

Rich Caruana. 1998. *Multitask Learning*, pages 95–133. Springer US, Boston, MA.

Hao Chen, Susan McKeever, and Sarah Jane Delany. 2019. The use of deep learning distributed representations in the identification of abusive text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 125–133.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. *arXiv preprint arXiv:1908.06083*.

Spiros V Georgakopoulos, Sotiris K Tasoulis, Aristidis G Vrahatis, and Vassilis P Plagianakos. 2018. Convolutional neural networks for toxic comment classification. In *Proceedings of the 10th hellenic conference on artificial intelligence*, pages 1–6.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Jeff Larson Lauren Julia Angwin Kirchner and Surya Mattu. 2016. Machine bias: Thereâăźs software used across the country to predict future criminals. and itâăźs biased against blacks.(may 2016).

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018)*, pages 1–11.

Elizabeth Reichert, Helen Qiu, and Jasmine Bayrooti. 2020. Reading between the demographic lines: Resolving sources of bias in toxicity classifiers. *arXiv preprint arXiv:2006.16402*.

Ameya Vaidya, Feng Mai, and Yue Ning. 2020. Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 683–693.

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.

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

Yu Zhang and Qiang Yang. 2018. An overview of multitask learning. *National Science Review*, 5(1):30–43.

A Appendix

Table 7: Dataset Composition

Data Item	**Category**	**Data Item**	**Category**
id	-	hindu	identity label
target	-	homosexual_gay_or_lesbian	identity label
comment_text	-	intellectual_or_learning_disability	identity label
severe_toxicity	aux label	jewish	identity label
obscene	aux label	latino	identity label
identity_attack	aux label	male	identity label
insult	aux label	muslim	identity label
threat	aux label	other_disability	identity label
asian	identity label	other_gender	identity label
atheist	identity label	other_race_or_ethnicity	identity label
bisexual	identity label	other_religion	identity label
black	identity label	other_sexual_orientation	identity label
buddhist	identity label	physical_disability	identity label
christian	identity label	psychiatric_or_mental_illness	identity label
female	identity label	transgender	identity label
heterosexual	identity label	white	identity label
publication_id	-	created_date	-
article_id	-	parent_id	-
funny	-	rating	-
sad	-	wow	-
disagree	-	likes	-
identity_annotator_count	-	sexual_explicit	-
		toxicity_annotator_count	-