CS 584: TOXIC COMMENT CLASSIFICATION

Brian Moon
Stevens Institute of Technology
bmoon@stevens.edu
Spring 2025

ABSTRACT

This project aims to develop a fair and robust model to classify toxic online comments, especially focusing on bias against demographic groups. While automated toxicity detection is critical for managing user-generated content, existing models often show bias, flagging false positives and requiring human input to verify the toxic flagging. The goal of the project is to train a transformer based model that balances accuracy with fairness, and maximize the lowest accuracy across all demographic biases being analyzed, and then evaluate whether the techniques used were effective.

In the end we will show
1: application of transformer based models to solve the problem, and
2: explore a different technique (worst-group-accuracy) and evaluate if it is effective in improving results.

1 Introduction

Online comment moderation tools often rely either on manually coded keyword detection or machine learning models to detect toxic or harmful speech. However, keyword-based methods struggle to adapt to evolving language and may miss context, while machine learning models often inherit social biases from their training data. These models can disproportionately flag comments that mention race, gender, or religion, even when the comment itself is not toxic.
This project aims to address this challenge by training a toxicity classifier that does not just optimize for overall performance, but seeks to improve fairness by maximizing the worst case performance across demographic identity subgroups. Our goal is to reduce disparities in prediction accuracies while still maintaining a high overall performance.

2 Related Work

Toxic comment detection relating to bias against specific demographics have been created before, but these classifiers have been shown to inherit social biases present in training data.
One approach to this is adversarial debiasing, which is to train a predictor that tries to maximize detection, then train an adversary that tries to minimize false positives ([1]).
Many models in use today still rely too much on false correlations between any demographic term and toxicity. This project aims to build on this by evaluating for each subpopulation and maximizing the worst performance.

In recent years, transformer based models such as BERT have become widely used for classification systems, including toxicity classification. Their ability to capture contextual relationships across long sequences has made them perform better compared to traditional methods like bag of words or RNN ([2]).
However, these transformer based models- although powerful- are still vulnerable to learning false correlations, such as associating demographic identity terms with toxicity when trained without fairness constraints.

3 Methodology

We can define the task as binary classification:
$f(x) = P(y = toxic|x)$
Where x is a comment and $y \in \{0, 1\}$

The key elements of my approach:

1. Data Processing: The given data is clean but raw text, and we need to set up the demographic labels as well as tokenize all the inputs.
2. Create Models: I will create 2 different models, and they will both use BERT transformer models. However, they will use different training methods that I will explain in 3.
   (a) I chose to use a transformer based model because toxic language is highly context-sensitive, and traditional models tend to overfit on individual keywords without understanding if those keywords are being used in a harmful or benign context. BERT is able to include more context, which should make it perform better than traditional methods for this task.
3. Create Custom Training Methods: I will create 2 different custom trainers for the BERT model- one will use a loss that optimizes global accuracy, and one will use a loss that optimizes worst-group-accuracy.
   (a) Although BERT is more robust to finding false correlations, it is still vulnerable to them. That is why we want to optimize for worst-group-accuracy and see if the model will capture less of these false correlations compared to overall accuracy.
   (b) We will optimize for these accuracies by using soft accuracies for both WGA and global accuracy. Soft accuracy is defined as $1 - mean(|P - L|)$ where P is the set of the predicted probabilities, and L is the set of true labels for those predictions.
   Soft accuracy is used because it will preserve more data as compared to thresholding and predicting those probabilities, so that for example, a probability of 0.7 contributes differently to the loss than 0.9.
   The soft WGA will find the soft accuracy within each identity group.
   (c) We will combine this to the loss by finding binary cross entropy loss, then apply a penalty for the soft accuracy as such: $Loss = BCELoss + \alpha * (1 - SoftAccuracy)$ where α is a fairness weight hyperparameter.
   This loss is set up like this because we want less penalty added if we have a high accuracy, and more penalty added if we have a low accuracy, and in this way, we will apply a penalty term to optimize for the soft accuracy. We will do this for both worst-group-accuracy and global accuracy to compare the two.
4. Reasoning for WGA: Optimizing for worst-group-accuracy (WGA) shifts the model's objective away from just minimizing average loss or maximizing overall accuracy, and forces it to perform consistently well across all subgroups, especially the ones that are most vulnerable to bias.

4 Experimental Setup

4.1 Data

We use a challenge-provided dataset on Kaggle ([3]) which includes user comments, toxicity labels (0,1), and binary indicators for 8 demographic identities (male, female, LGBTQ, christian, muslim, other religions, black, white).
We are provided both a train and validation dataset, so the splitting is done for us.
In order to tokenize our data to use with a BERT model, we will use BertTokenizer from Hugging Face's library (transformers).

4.2 Evaluation Metrics

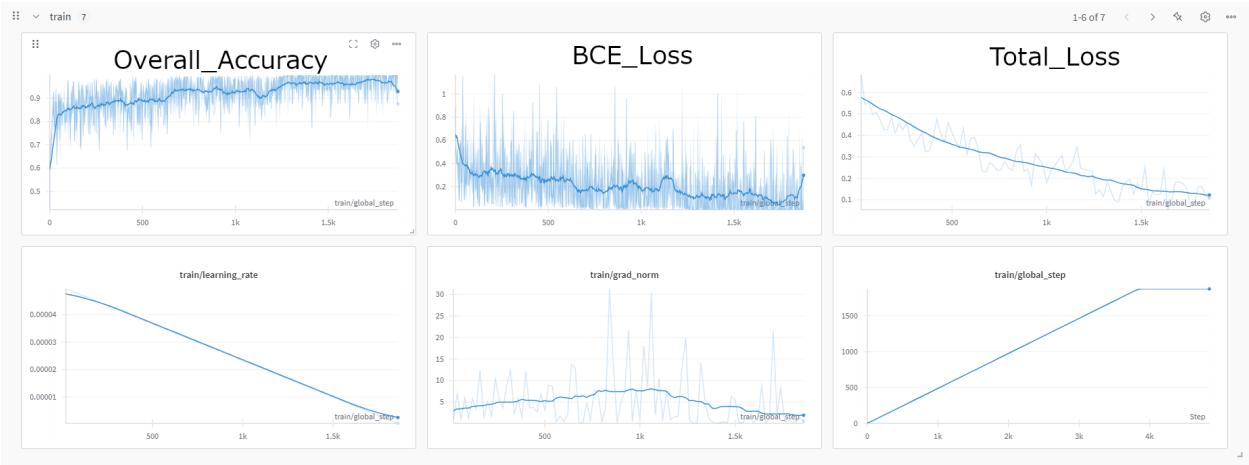
While training, we will use cross entropy loss but with a penalty term for the accuracy as described in 3.3c. The trainer will be able to use accuracy metrics and optimize for them.
While evaluating, we use global accuracy and F1-score. Global accuracy gives us a good raw accuracy, and F1-score gives more insight into false-positives and false-negatives.
After predictions, we can also further evaluate by finding the raw accuracy only within predictions that flag a toxic comment, and this raw accuracy will inform us more about how many false positives exist.

4.3 Comparison Methods

We will compare our two models by first comparing raw accuracy and f1-score which will provide some insight into the overall performance, along with some insight into false positives and negatives with the f1-score.
We will also find the accuracy of all the predictions that detect toxicity in order to evaluate how many false positives exist, and compare those two accuracies to see which model was able to limit more false positives.
We will also see the accuracy of a baseline non-BERT model and compare it to the accuracy of the BERT models I create to determine if using a BERT model improves performance.

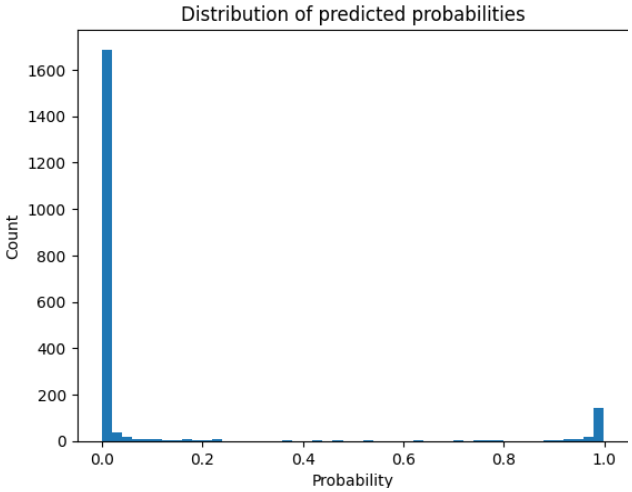
5 Results

1. Global Accuracy Optimized BERT Model:
   (a) Training Results:



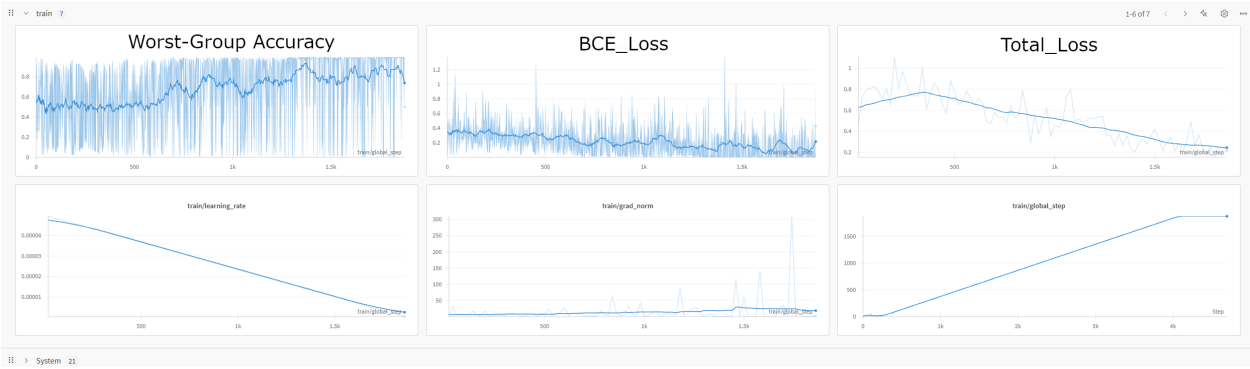
   We can see that throughout the training, the loss trended downwards, and the overall accuracy trended upwards, which indicates that the model successfully learned from the data

   (b) Prediction Distribution:



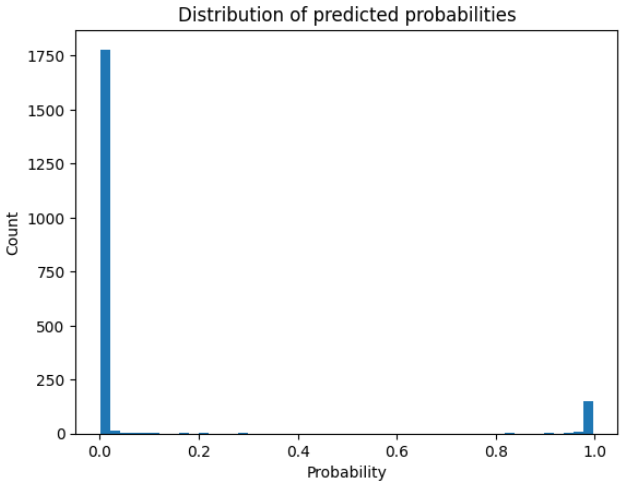
   (c) The predictions were almost all 0 or 1, with not many samples having a probability in between. This is an indication of slight overfitting, but is still useful for comparison if the other model is trained the same exact way but with WGA optimization.
   (d) We had a raw accuracy of 0.911, and an F1 score of 0.7581
   (e) For predictions that suggest non-toxic, we had a raw accuracy of 0.9488
   (f) For predictions that suggest toxic, we had a raw accuracy of 0.5743
   (g) This indicates that we detected most of the toxic comments, but we had many false positives.
2. Worst-group-accuracy Optimized BERT Model:

(a) Training Results:



We can see that throughout the training, the loss trended downwards, and the WGA trended upwards, which indicates that the model successfully learned from the data

(b) Prediction Distribution:



(c) The predictions similarly were almost all 0 or 1, which indicates slight overfitting, but it should match a similar amount of overfitting to the global accuracy model to draw a fair comparison.

(d) We had a raw accuracy of 0.9185, and an F1 score of 0.7647. This indicates that we performed slightly better in accuracy, and more importantly, slightly better on the F1 score, which indicates that we did improve on the false positives and negatives.

(e) For predictions that suggest non-toxic, we had a raw accuracy of 0.9463, which actually is a slight drop in accuracy, but this amount is expected as the model aims to have less false positives, so a slight decrease is reasonable.

(f) For predictions that suggest toxic, we had a raw accuracy of 0.6286, which is a significant increase over the other model. This indicates that we had less false positives by implementing WGA optimization.

3. Comparison table:

Accuracy Table				
Model	Total_Accuracy	Total_F1Score	Non-Toxic Predictions	Toxic Predictions
Global Accuracy Model	0.911	0.7581	0.9488	0.5743
WGA Model	0.9185	0.7647	0.9463	0.6286

4. Comparison of BERT to baseline model:
The Kaggle competition ([3]) includes a baseline classifier that uses MLP, and it results in an accuracy of about 0.7022 accuracy. We can already see that both BERT models performed much better than the baseline non-BERT model, which shows that BERT model did make a large difference in improving performance

References

[1] B. H. Zhang, B. Lemoine, and M. Mitchell, Mitigating Unwanted Biases with Adversarial Learning. 2018. [Online]. Available: https://arxiv.org/abs/1801.07593.
[2] Fields, John & Chovanec, Kevin & Madiraju, Praveen. (2024). A Survey of Text Classification With Transformers: How Wide? How Large? How Long? How Accurate? How Expensive? How Safe?. IEEE Access. PP. 1-1. 10.1109/ACCESS.2024.3349952.
[3] Leo Fillioux. Toxic comment classification 2. https://kaggle.com/competitions/toxic-comment-classification-dsba-2023-2, 2024. Kaggle.