Removing Bias from Toxic Comment Classifiers

Benjamin Bowman
George Washington University
bbowman410@gwu.edu

Cole Weinhauer
George Washington University
ceweinhauer@gmail.com

Christian Cleber Masdeval Braz
George Washington University
masdeval@gwu.edu

***Abstract*—In the last several years it has become increasingly obvious that machine learning (ML) and artificial intelligence (AI) algorithms have systematic social bias built into the models. This has been found to either mirror existing bias, or even in some cases amplify it. As algorithms continue to play an increasingly import role in our society, it is imperative that we come up with techniques for both measuring fairness in ML, and also techniques to remediate the identified fairness deficiencies.**

A standard use case in ML and specifically natural language processing (NLP) is the classification of textual content as either "toxic" or "non-toxic". Toxic comments plague many web-based community forums to the point where their utility is diminished. There is often too many comments for human moderators to manually evaluate for toxicity. Thus it is necessary that we have machine-based algorithms for quickly and accurately identifying and removing such comments.

It has recently been discovered that many state-of-art toxic comment classifiers are highly biased against certain groups. For example, online forums within the LGBTQ community may use the word "gay" frequently in a non-toxic way. However, state-of-art toxic comment classifiers will likely label this content as toxic, which disproportionately effects members of this community in an adverse way. In this project, we will participate in a Kaggle competition [1] with the goal of building a fairness optimized toxic comment classifier. We will evaluate and explore the dataset provided by the competition and determine likely causes of bias, and we will utilize techniques learned in the course of this class to train a ML classification algorithm with fairness in mind.

I. Introduction

Machine learning (ML) algorithms are increasingly being used to solve real-world problems often with better rates of accuracy than humans [5]. However, recently it has been shown by many works [4], [10], [2] that the ML algorithms we train are not immune to human bias and often times exhibit similar if not enhanced systematic bias against particular groups. For example, it has been shown that in Caliskan et. al. [4] that word embeddings generated by popular ML algorithms have encoded in them significant gender bias. As these word embeddings are utilized in real-world problems such as resume screening, it is possible that the algorithms trained on these embeddings may treat people differently based on their gender.

Another problem related to word embeddings is the problem of toxic comment classification. Online social media forums are highly popular and used by many individuals as well as organizations. There is often more comments than moderators can read, and thus it is important that we have algorithms capable of identifying comments which violate a platforms terms of service. Many platforms prohibit hate speech and other forms of comments that seek only to insult or hurt a person or a group. This broad category of comments have come to be known as "toxic comments". In order to detect and consequently remove toxic comments, many platforms utilize a ML approach to automatically classify comments based on their level of toxicity. Comments that are deemed to be highly toxic may be censored and/or removed from the platform.

Recently it has been discovered by the Jigsaw group at Google that many such algorithms are biased against certain groups of people. For example, members of the LGBTQ community who may often use the word "gay" in a non-toxic setting, may erroneously get flagged as being a toxic comment due to biased ML algorithms. This is due to the fact that the word "gay" is more often used in a toxic way to insult another person or group. Thus, a machine learning trying to optimize accuracy will learn to classify comments with the word "gay" more often in the toxic category than not. As this is a very relevant problem, there is a current Kaggle competition [1] seeking to find new ways to eliminate bias from toxic comment classifiers. In this work, we will participate in the Kaggle competition, and attempt to both understand sources of bias in the provided dataset, as well as generate a ML model trained specifically with fairness in mind.

In summary, the contributions oft this work are as follows:

- A comprehensive analysis of the dataset provided in the Kaggle competition and identification of where bias could effect the fairness of a trained classifier
- Two different ML approaches for improving fairness in toxic comment classifiers
- An evaluation of fairness performance of our fairness-priority models vs. a baseline models

II. Problem Specification and Bias Definition

Text based platforms are all over the Internet and having efficient tools to monitor such data plays an important role in many current applications. Understanding unstructured data imposes several difficulties, representing an important topic of research in Machine Learning (ML) and Natural Language Processing (NLP). One of the main draws of social media is the interaction with other users, which is achieved mainly via exchanging textual comments.

The Conversation AI team, among other things, has focused in building machine learning models that can identify toxicity in online conversations, where toxicity is defined as anything rude, disrespectful or otherwise likely to make someone leave a discussion. Recently, however, it has been discovered that these toxic comment classification algorithms have differing levels of performance across groups associated with certain identities.

This is due to the fact that ML algorithms are trained on data which has inherent societal bias built in. As ML techniques are based on statistics, bias in the training dataset will frequently get mirrored in the trained model.

The problem at hand is to fit a model to detect toxic comments while at the same time minimize this type of unintended bias with respect to mentions of identities. The metrics used to evaluate accuracy and bias are:

- Overall ROC-AUC of the model.
- Subgroup AUC: ROC-AUC specific for each identity subgroup.
- BPSN (Background Positive Subgroup Negative): restrict the dataset to the non-toxic examples that mention the identity and the toxic examples that do not. This metric represents how likely the model is to mistake a negative sample with associated with an identity as a toxic sample (a.k.a false-positive).
- BNSP (Background Negative Subgroup Positive): restrict the dataset to the toxic examples that mention the identity and the non-toxic examples that do not. This metric represents how likely the model is to mistake a positive sample associated with the identity for a non-toxic sample (a.k.a false-negative).

The proposed single final metric to evaluate regarding bias is defined as:

$$score = w_0 AUC_{overall} + \sum_{a=1}^{A} w_a M_p(m_{s,a}) \quad (1)$$

where the Generalized Mean of Bias AUC $M_p(ms)$ is defined as:

$$M_p(m_s) = ((1/N)\sum_{s=1}^{N} m_s^p)^{1/p} \quad (2)$$

Thus, throughout this work we compute the final score, as well as the subgroups metrics, of our models in order to evaluate their unintended bias and compare their performance.

III. Related Work

There has been an explosion of research within the last several years focusing on the fairness, or lack-there-of, in machine learning algorithms. Olteanu et al. published a comprehensive survey of how and why bias exists in our ML algorithms [8]. They show how bias can be introduced at almost every stage of a machine learning pipeline, from the data acquisition to the final inference. Individual machine learning domains were evaluated for ML bias such as NLP [4], and image processing [11]. Many studies were performed to evaluate how these biased ML algorithms were adversely effective real people in real scenarios such as racial disparities in police stops [10].

There have additionally been many suggested ways to remediate this problem of fairness in ML. Hardt et al. suggested

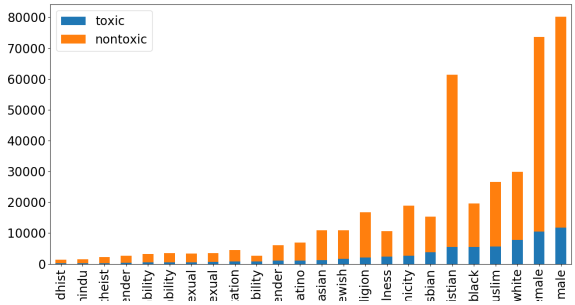


Fig. 1: Toxic vs. Non-Toxic comment counts for various identity classes. We can see there is a significant imbalance of data both from toxic to non-toxic, but also from one identity group to another.

techniques such as providing equalized-odds or equalized-opportunity across different protected groups to achieve fairness [7]. Dwork et. al. proposes a technique of requiring the treatment of similar individuals in a similar manner to achieve fariness [6]. Other techniques actively remove bias from models by systematic removal of any information that would contribute to bias [3], or the modification and constraining of the dataset on which models are trained [12].

IV. Dataset

The dataset provided in the Kaggle competition is roughly 1.8 million public comments from the now-defunct platform Civil Comments. This was a commenting platform built with the goal of essentially crowd sourcing comment toxicity classification, by requiring all commenter to first rate other comments prior to their own comment being rated by others and eventually posted. As part of this process, Civil Comments amassed a large amount of comments with associated toxicity scores. In addition, the dataset also includes some additional information such as the type of toxicity, as well as identity attributes extracted from the comment itself.

The dataset comes with 45 features, and are divided into 3 major categories. The first is general informative information about the comment: the comment itself, the id, the parent id, when it was created, and the number of major types of reactions it got, including sad, happy, wow, and likes. The second set defines the toxicity. This includes 5 major types of toxicity: severe toxicity, obscene, identity insult, threats, and has values between 0 and 1 to signify how much the comment is representative of each of these things. Also, this includes the final toxicity score. The third category is the identities. This includes gender (female, male or transgender), race (asian, black, latino, or white), religion (atheist, buddhist, christian, hindu, jewish, or muslim), sexual orientation (heterosexual or homosexual), as well as intellectual, physical disability, and psychiatric or mental illness. This also includes features for other religions, race, sexual orientation, disability, and gender. These features also go from 0 to 1 to represent how much the identity is represented in the comment.

In Figure 1, we see the ratio of toxic to non-toxic for each of the itentity clases. We defined a comment as toxic if its

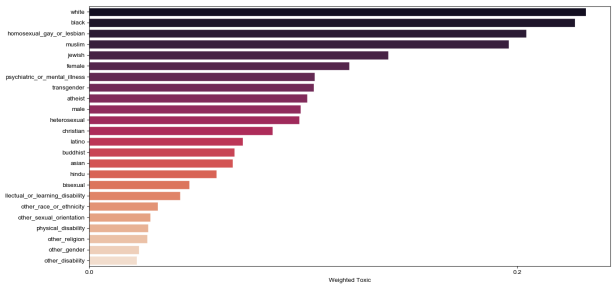


Fig. 2: Heat map indicating the relative ratio of toxic to non-toxic comments for each identity attribute. We can see several identities have much more toxicity than others.

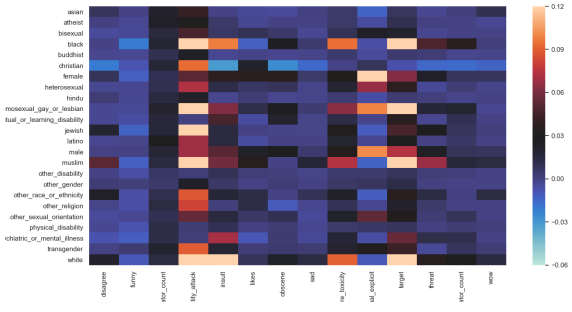


Fig. 3: Correlations of Identities and Characteristics

toxicity is above .5, and nontoxic if its toxicity is below .5. As we can see, there is a much higher percentage of nontoxic comments, which could show some initial bias in the dataset by way of the class imbalance. In addition, we see that the ratio of toxic to non-toxic is very different across identity class. Figure 2 shows the per-class percentage of toxic comments. The values range from very few toxic comments, 2.5%, up to many toxic comments, 23%. This is yet another possible source of model bias, where the identities with the most toxicitiy will dictate what it means to be toxic or not.

Figure 3 shows the correlations between all of our identities and each characteristic defined in the dataset. Most have little correlation, evident by the large number of blue boxes. However, some have correlation above .1, like black and identity attack and female and sexually explicit. This again speaks to the fact that the dataset at hand clearly has some implicit bias built in to the data.

V. Approach

This section presents our techniques for building toxic comment classifiers while minimizing bias across the identity groups discussed previously. We present two techniques for attempting to remove bias from toxic comment classifiers. The first technique is based on [3] and actively tunes embeddings to remove bias. The second technique is based on a mixture of experts model where each expert learner is trained on identity specific data.

TABLE I: Embedding technique model comparison

Embedding Technique	**Overall Test ROC-AUC**	textbfFinal Bias AUC Score
Bag-of-Words	88.7%	83.4%
GLOVE Embeddings	83.7%	80.2%

TABLE II: Toxicity predictions of the embedding model for identity words.

identity	**toxicity score**
black	0.99004923
christian	0.00011506
female	0.9390484
homosexual	0.99395751
gay	0.88179089
lesbian	0.93650436
jewish	0.2304727
male	0.84353626
muslim	0.99171696
white	0.99365251

A. BOW and Embeddings

Bag-of-words is a common approach to processing textual data. It is a technique to extract suitable features from text to be used in machine learning algorithms. We thus fit a binary BOW using a Maximum Entropy (ME) classifier with a vocabulary of 10,000 words. The metrics computed for it are in Table I. We can see that the BOW model performs very well considering it is such a simple technique.

Next we try a similar approach where, instead of a vector of 10,000 features, we use word embeddings. Word embeddings are dense vector representation of words. The semantics of a word is encapsulated in its vector, and words that share similar meaning/context have similar vectors. This allow several interesting operations between them as, for example, the cosine similarity. To create the features, we use the 200 size vector from the Glove [9] implementation as follows:

- Iterate over all examples and tokenize the comment of each one.
- Given the tokens of an example, for each one get its embedding counterpart (if it exists) and sum them.
- Return the average of this sum.

We then train, again, a ME classifier with L2 regularization, obtaining the results shown in Table I.

We can see that our two baseline models are similar regarding their accuracy and bias score, being the BOW model slightly better. In the next next section we discuss our first attempt in trying to reduce the bias in the model.

B. Word Vector de-biasing

Recall that the bias in this problem is related to the classifier predicting high toxicity for neutral or pleasant sentences, just because the presence of some identity word, as depicted in Figure 4. Hence, a natural question that arise is, what is the predicted toxicity for specific identity words of our embedding model. Table II shows these probabilities for the main identity words addressed in this work.

We can see that the classifier learned to associate very high or very low toxicity for certain identities, showing a

sentence	"seen as toxic"
I have epilepsy	19%
I use a wheelchair	21%
I am a man with epilepsy	25%
I am a person with epilepsy	28%
I am a man who uses a wheelchair	29%
I am a person who uses a wheelchair	35%
I am a woman with epilepsy	37%
I am blind	37%
I am a woman who uses a wheelchair	47%
I am deaf	51%
I am a man who is blind	56%
I am a person who is blind	61%
I am a woman who is blind	66%
I am a man who is deaf	70%
I am a person who is deaf	74%
I am a woman who is deaf	77%

Fig. 4: Sentences referencing regularly targeted identity groups return higher toxicity scores.

strong trend of bias. This led to the intuition for our first experiment to handle the unintended bias. Using techniques similar as demonstrated in [3], we try to debias the identity words by adding pleasant meaning to one highly toxic, and subtracting pleasant meaning to one highly non toxic. If the classifier learned to associate the number signature found in vectors with high toxicity, maybe if we, during training, replace identities vectors for something else in the toxic cases, this might help to reduce bias. This replacement should be close to the original in order to not compromise accuracy. The way we do that is creating a pleasant and unpleasant word vector, and then adding or subtracting a fraction of them from each identity word. By softening the vector signature pattern when they occur in toxic cases, in the end the strong association with toxicity will also be softened.

We generate a $pleasant\_vector$ as an average of pleasant words such as: freedom, health, peace, cheer, gentle, gift, honor, miracle, sunrise. Similarly, we generate an unpleasant vector as the average of unpleasant words such as: filth, poison, stink, ugly, evil, kill rotten, vomit, negative, bad. Finally, to balance out overly-negative bias, we actively subtract the unpleasant vector, and add the pleasant vector. Similarly, to balance out overly-positive bias, we add the unpleasant vector and subtract the pleasant vector.

Once more, these replacement vectors should resemble enough the original one, with the risk that if this is not the case, the solution may hurt the accuracy. Following this, every time a toxic example containing an identity word occur, the replacement will be used instead. For non toxic examples, the original vector is kept. If the replacement resemble enough the original, during the test phase, when only the original vectors take place, the classifier is still be able to detect that, for instance, the original black vector left a signature in a toxic example strong enough to the classifier predict the toxicity correctly.

Following this methodology, it turns out that it is not enough to change only each identity vector independently. For example, suppose in an extreme case one swap BLACK for CHRISTIAN each time they appear during training in toxic cases. The result is that the final classifier still predicts BLACK being highly toxic and CHRISTIAN being non toxic at all. Well, if this is the case, we can argue that the feature vectors created for toxic cases are them similar to BLACK, and when we test for toxicity using only the BLACK vector, it is still high. Therefore, it seems to be necessary to debias other words that occur together with the identity words in toxic comments as well. To accomplish that, we devise a random optimization procedure that works as follow:

```
DO FOR TOXIC COMMENTS
  FOR EVERY WORD IN COMMENT
    IF word IS SIMILAR SOME IDENTITY
      IF word IS TOO TOXIC
        REPEAT UNTIL GOOD ESTIMATION FOR
        ADD AND SUB
          add = random()
          sub = random()
          replace_word = word - sub *
                         UNPLEASANT +
                         add  * PLEASANT
          IF NEW TOXICITY OF WORD IS FINE
            IF replace_word IS CLOSE
            ENOUGH TO ORIGINAL
              USE replace_word
              INSTEAD OF ORIGINAL
  COMPUTE TOXICITY FOR THE
  OBTAINED FEATURE VECTOR
WHILE PREDICTION IS NOT TOXIC ENOUGH
```

A similar procedure is used for too non toxic words. This is our debiasing algorithm textbfby similarity. We also tried a slightly different procedure where we debias any word that is too toxic or too non toxic, weather or not it is similar to an identity word or not. We call this version the debiasing **by toxicity**.

In both cases, once we find the best local estimative for the weights (add and sub variables in the pseudocode) for a given word, we save and use them from then on. We train using only the examples that belong to some identity. This is equivalent to 11.2% of the train data, with 1.4% and 9.8% of toxic and non toxic comments respectively. Even with this small proportion of the train dataset, it takes around eleven hours to run in a high performance computing environment. Table III shows the main metrics for the three classifiers.

C. Mixture of Fair Experts

Our second technique for building a fair toxic comment classifier is based on a mixture of experts approach. The mixture of experts is a term from the ML literature for models which divide a problem into smaller homogenous subproblems. Our idea is to utilize a similar approach to create a specific toxic comment classifier for each of the identity groups in the dataset. In addition, we will create an identity predictor for each identity group. This way we will be able to predict when a comment belongs to a specific identity, and then utilize our expert predictor to perform the comment classification.

TABLE III: Metric results for the three classifiers

No Debias		Optimizing by Toxicity		Optimizing by Similarity	
Overall ROC-AUC: 83.7%	Bias AUC: 81.2%	Overall ROC-AUC: 81.2%	Bias AUC: 78.4%	Overall ROC-AUC: 78.3%	Bias AUC: 75.7%
black	0.99	black	0.99	black	0.99
christian	0.00	christian	0.00	christian	0.00
female	0.94	female	0.63	female	0.23
homosexual	0.99	homosexual	0.99	homosexual	0.99
gay	0.88	gay	0.98	gay	0.74
lesbian	0.94	lesbian	0.99	lesbian	0.90
jewish	0.23	jewish	0.00	jewish	0.00
male	0.84	male	0.45	male	0.15
muslim	0.99	muslim	0.66	muslime	0.11
white	0.99	white	0.99	white	0.55

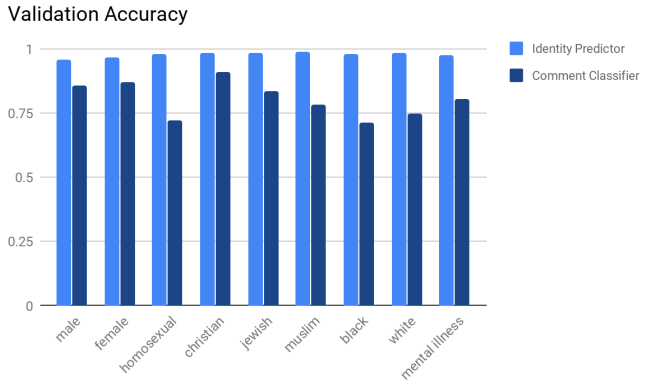


Fig. 5: Validation set accuracy metrics for the identity predictors and identity comment classifiers.

1) Model: We utilize a single Neural Network architecture for all of our models as part of the Mixture of Fair Experts. The model is a simple 3-layer convolutional model with a max-pooling layer after each convolution. The output section consists of a single dense layer, followed by a 2-class softmax layer, where the classes depend on what type of predictor we are building. Word embeddings are generated for each comment utilizing the pre-trained Glove [9] word-embeddings. Each comment is truncated/padded to 250 words. We utilize a cross-entropy loss function and optimize with rmsprop.

2) Training: We split our data into a 80% training set, and a 20% testing set. We train a baseline model using all of the training data. We use this to compare with our Mixture of Experts model. Next, we train a set of identity predictors, where each is trained on all of the comments associated with a certain identity in the training data, and a random sampling of comments not associated with that identity. We sample the negative comments such that the classes are balanced.

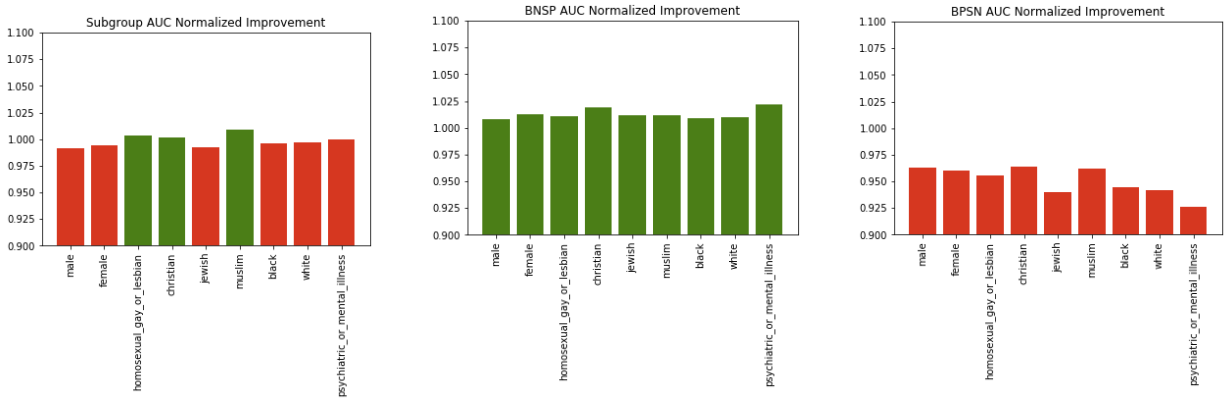
Next, we train a set of identity-specific toxicity comment classifiers (ICCs), where each classifier is trained only on the samples in the training data that were labeled as being relevant to the particular identity. Figure 5 show the validation accuracy scores for each identity predictor and comment predictor. We can see in most cases the identity predictors worked very well, able to achieve over 90% accuracy for all identities. However, we see that each individual identity-specific comment classifier performance ranges from above 90% to below 75%. This will likely effect the overall performance of our mixture of experts model.

3) Testing: Now that we have our per-identity comment classifiers, as well as the ability to predict the identities associated with a particular comment, we are ready to build the final prediction model. Pseudocode for the final prediction routine is shown below:

```
baseline_prediction=
  baseline_model.predict(X)
for identity in all_identities:
  confidences.append(
    identity_predictors[identity].predict(X))
  values.append(
    expert_classifier[identity].predict(X))
return average(
          average(values, weights=confidences),
          baseline_prediction)
```

For each sample in our test set, we compute the baseline prediction, as well as the identity predictions and identity-expert comment classification. After this is complete, we identify if any identity predictors were highly confident in determining an identity. If they were, we perform a weighted average across the identity comment classifier experts for the identified identities, where the weights are the identity prediction confidence scores (note that a single example might be related to multiples identities). Finally, we perform a final unweighted average with the baseline model score. In the event that there were no identities predicted for a particular sample, we simply use the baseline prediction as-is.

4) Results: We compute the identity-specific AUC, BNSP AUC and BPSN AUC for both the mixture of experts model, as well the baseline model alone. We show the normalized scoares compared with the baseline model alone in Figure 6. We can see that in a few cases, the mixture of experts was able to help fairness of some identity groups (e.g., homosexual, muslim, christian). However, this was at the expense of loss in accuracy for other identity groups. Additionally, we can see that for all classes the BPSN scores were lower, and the BNSP samples were higher. This means that, overall, the mixture of experts model produced less false negatives (better BNSP than baseline), but more false-positives (worse BPSN). We can argue that the expert models were actually more aggressive in their labeling scheme, more often labeling samples associated with an identity as toxic than the baseline model. This is likely due to the fact that we train our expert comment classifiers only on the data associated with a particular identity. They are likely missing out on a large amount of useful knowledge from other identities which helps improve the performance as in the baseline model (although, for each subgroup, there are more nontoxic examples). In future work we would like to explore identifying additional training data for the comment classifiers from other identity groups that would likely not interfere with learning the identity specific behavior (e.g., by only selecting training samples that do not have significant word overlap with the words typically associated with the identity the expert is focused on).



(a) Subgroup AUC

(b) Background Negative, Subgroup Positive

(c) Background Positive, Subgroup Negative

Fig. 6: Comparison of bias metrics for the mixture of experts model. Each metric is normalized against the baseline model. We can see that some identity groups did become more fair, yet at the expense of others. Overall the model seems to be more prone to false positives, while lessening false negatives.

VI. Conclusion

In this project we designed and evaluated two potential problems for mitigating bias in toxic comment classification algorithms. We explored the data provided by the Kaggle competition titled "Jigsaw Unintended Bias in Toxicity Classification" and identified where and why bias would creep into our ML models. We develop two separate techniques for mitigating bias. In one, we actively debias identity specific words using known pleasant and unpleasant word vectors. In the other, we build a mixture of experts model, where each expert is specialized to operate on a single identity group. We show in both cases how we are able to improve bias for some identity groups, however often at the expense of other identity groups.

References

[1] "Jigsaw unintended bias in toxicity classification," https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification, note = Accessed: 2019-04-17.

[2] E. Bakshy, S. Messing, and L. A. Adamic, "Exposure to ideologically diverse news and opinion on facebook," *Science*, vol. 348, no. 6239, pp. 1130–1132, 2015.

[3] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Advances in neural information processing systems*, 2016, pp. 4349–4357.

[4] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[5] R. M. Dawes, D. Faust, and P. E. Meehl, "Clinical versus actuarial judgment," *Science*, vol. 243, no. 4899, pp. 1668–1674, 1989.

[6] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*. ACM, 2012, pp. 214–226.

[7] M. Hardt, E. Price, N. Srebro *et al.*, "Equality of opportunity in supervised learning," in *Advances in neural information processing systems*, 2016, pp. 3315–3323.

[8] A. Olteanu, C. Castillo, F. Diaz, and E. Kiciman, "Social data: Biases, methodological pitfalls, and ethical boundaries," 2016.

[9] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[10] E. Pierson, C. Simoiu, J. Overgoor, S. Corbett-Davies, V. Ramachandran, C. Phillips, and S. Goel, "A large-scale analysis of racial disparities in police stops across the united states," *arXiv preprint arXiv:1706.05678*, 2017.

[11] A. Torralba, A. A. Efros *et al.*, "Unbiased look at dataset bias." in *CVPR*, vol. 1, no. 2. Citeseer, 2011, p. 7.

[12] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang, "Men also like shopping: Reducing gender bias amplification using corpus-level constraints," *arXiv preprint arXiv:1707.09457*, 2017.