Towards an Unbiased Language Model for Hate Speech Classification

Athiya Deviyani
Carnegie Mellon University
adeviyan@cs.cmu.edu

Abstract

Throughout the past decade, we have observed the rapid increase in social media content, and with it, the presence of online hate speech becomes more prominent. This paper will introduce a language model based on the Recurrent Convolutional Neural Network (R-CNN) architecture which aims to automatically detect hate speech as well as a penalty-based method aimed at mitigating the biases learned from our final model.

1 Introduction

The most commonly accepted formal definition is adapted from the Encyclopedia of the American Constitution, defined by Nockleby (1994) as *any communication that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic.* To cope with the massive scale of online platforms, there is an increased need for automated offensive speech detection methods. While language models are becoming more advanced, studies conducted by Yasin (2018) and Guynn (2020) show that current hate speech and toxic language detection systems exhibit problematic and discriminatory behavior. This causes them to have a disproportionately negative impact on minority user demographics. In this paper, we will first evaluate the PerspectiveAPI using various metrics. PerspectiveAPI is a tool released by Alphabet that assigns a toxicity score to a text sequence. Then, we will compare the performance of the PerspectiveAPI model with basic machine learning algorithms for binary text classification. For our advanced analysis, we will explore how the neural language models improve the classification performance. All our models will also be assessed on how biased they are at classifying toxicity across different user demographics. Additionally, we propose a penalty-based method to debias the model across the various user demographic groups in order to mitigate the performance disparities and overcome the bias-accuracy trade-off. Finally, we will discuss our findings and talk through the corresponding ethical issues and challenges that accompany the deployment of a machine-learning based hate speech detection system.

2 Methodology

2.1 Datasets

The dataset that we are using for our experiments is taken from the 2019 SemEval task (Zampieri et al., 2019) on offensive language detection. The training data consists of over 10K tweets labeled OFF (offensive) and NOT (not offensive) and an accompanying validation set of 5K tweets labeled similarly. We were also given a subset of the TwitterAAE dataset (Blodgett et al., 2016) containing non-offensive tweets and their demographic labels: WHITE, AA (African American), HISPANIC, and OTHER.

2.2 Text preprocessing

Since we are given raw data from an informal language corpora, we are often met with sentences which contain various spelling errors. It is crucial for us to perform appropriate text preprocessing so we can obtain as many 'hateful signals' as possible. We have decided to use the `ekphrasis` Python library which was specifically built to clean Twitter text data using a FastText language model trained on 1 billion tweets. It automatically detects spelling errors such as elongated words (e.g. "wooow" → "wow") and separates hashtags to individual words (e.g. "#BlackLivesMatter" → "Black Lives Matter"). Finally, we perform basic preprocessing using NLTK to get rid of residual noise such as punctuations, emojis, user tokens, and more.

2.3 Model choice and evaluation metrics

For our basic analysis, we built a pipeline for a binary text classification task using two popular methods available in scikit-learn. The tweets are tokenized and transformed into feature vectors using CountVectorizer, passed through a TfIdfTransformer to downscale words that are occur in many documents, and finally for our classification model we evaluated LogisticRegression and MultinomialNB (Multinomial Naive Bayes).

To evaluate the performance of our model, we will report the overall accuracy and F1-score over the validation set. We will extend our evaluation to observe the performance disparity of our models by calculating their false positive rates and standard deviation over the different demographic groups in our additional dataset. Throughout this paper, we will use this standard deviation as our primary bias metric. The False Positive Rate (FPR) denotes how often the model misclassified nontoxic speech as toxic. In both F1-score and FPR, we have taken the NOT label as the positive label. Finally, we have also evaluated the performance of the PerspectiveAPI at classifying offensive language using similar metrics to assess how our models fare against the current state of the art hate speech detection model.

3 Analysis of results

3.1 Basic analysis

The results of our experiments are presented in Table 1. At a glance, we notice that there exists a trade-off between accuracy and the standard deviation between the FPRs of the different demographic groups. This trade-off exists because there are different consequences to prioritizing one metric over another. We strongly believe that a model with a low accuracy exhibits poor performance over hate speech classification, thus not accurate enough to be useful. On the other hand, an imbalanced model with a high standard deviation across the demographic FPRs may result in a model which imposes racial biases and correlates linguistic style adopted by a certain demographic to toxicity.

The PerspectiveAPI has an accuracy of 76.44% with an F1-score of 0.85 on the positive class. However, out of all the demographic groups, the FPR tends to be staggeringly higher for the AA class compared to the other classes, with the PerspectiveAPI having a value of 0.19. This means that the model has somehow learned an unwanted correlation between linguistic style and toxicity.

It is interesting to note that our simple text classification pipeline based on Logistic Regression performs almost as well, with an accuracy of 75.23% and an F1-score of 0.84. Although the Multinomial Naive Bayes model received a much lower accuracy of 72.05% and an F1-score of 0.82, we can observe that it is the least biased model as it has the lowest FPR on the AA class and the lowest overall standard deviation across the FPRs of the different demographic groups.

3.2 Error analysis

We decided to investigate further by performing a detailed error analysis on misclassified non-offensive text. We used the WordCloud Python library which visualizes the top occurring words within a corpus in an intuitive manner as shown in Figure 1.

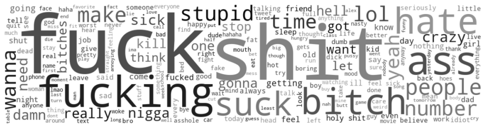

Figure 1: Common words in misclassified tweets

We can observe that most of the non-offensive tweets that were misclassified as offensive often contain profanity and words that are indicative of hate such as ‘suck’, ‘crazy’, and ‘stupid’. This means that our models put a high weightage on the presence of profanity to predict toxicity, which is not always the best intuition when operating on a corpus of largely colloquial text. This is an example of shortcut learning, where our model attempts to identify the simplest solution or a ’shortcut’ to solve a given problem. For example, our models would classify the tweet “Your music is lit as f***!” as hate speech while the general sentiment of the tweet is positive and complimentary. This is also observed by Davidson et al. (2017), where they noted that not all tweets containing offensive language in the form of profanity are hateful or offensive per se, and the task of determining which tweet is which is not a trivial task. Additionally, we also generated word clouds for the tweets coming from the different demographic groups and found that tweets labeled AA often contain profanity, however the use of profanity may not always be in a hateful context. This issue was also addressed by Zhou et al. (2021), where they observed biased associations between toxicity and dialectical markers, specifically in African American English.

Model	Acc. (%)	F1	White FPR	Hispanic FPR	AA FPR	Other FPR	FPR SD
PerspectiveAPI	76.44	0.8472	0.0732	0.1015	0.1898	0.0118	0.0740
LogisticRegression	75.23	0.8357	0.1051	0.1284	0.2139	0.0000	0.0880
MultinomialNB	72.05	0.8235	0.0649	0.0746	0.1747	0.0294	0.0623
R-CNN (LSTM)	78.32	0.8456	0.1469	0.1672	0.2681	0.0000	0.1106
R-CNN (GRU)	78.92	0.8506	0.1511	0.1701	0.2681	0.0000	0.1108
R-CNN with λ (GRU)	79.00	0.8551	0.1419	0.1672	0.2229	0.0000	0.0949
R-CNN (2 GRUs)	79.68	0.8564	0.1445	0.1701	0.2651	0.0059	0.1071
R-CNN with λ (2 GRUs)	78.93	0.8549	0.1403	0.1642	0.2410	0.0000	0.1005

Table 1: Performance metrics of the various models trained to perform the task of hate speech classification

We determined that this is one of the main reasons why our models have a relatively poorer FPR on the AA demographic group, however there may be other underlying reasons which were not detected.

4 Advanced analysis

4.1 Neural language models

We will implement various neural network architectures which are often used for language modeling to improve the overall hate speech classification performance. We decided to employ a different feature representation method by using GloVe word embeddings (Pennington et al., 2014) with 100 dimensions that were specifically trained on a Twitter corpus.

Following the work presented by Badjatiya et al. (2017), we will investigate hybrid neural network architectures which are combinations of Convolutional Neural Networks (CNNs) (LeCun et al., 2015) and Recurrent Neural Networks (RNN) (Cho et al., 2014). CNNs are great at interpreting data that does not come in a sequence, such as indicating the presence of a certain feature within an input. RNNs, on the other hand, are great at interpreting temporal or sequential information, such as a structured sentence or tweet. Therefore, by designing a model based on the Recurrent Convolutional Neural Network (R-CNN) architecture, we would have a powerful feature extractor that is able to capture contextual information from sequential data. For the RNN, we have also evaluated the performance of using bidirectional LSTM (Long Short-Term Memory) and bidirectional GRU (Gated Recurrent Unit) layers.

We preprocessed the input data the same way we did for the basic analysis. We tokenized the tweets using the Tensorflow Keras tokenizer and padded the sequences. The tokenized tweets are then passed through the Embedding Layer of our neural network. We used the RMSprop optimizer with a learning rate of 0.0001 and a binary cross-entropy loss. We trained each network for at most 50 epochs with early stopping to prevent the model from overfitting. Finally, we made predictions over the validation set and obtained the accuracy and F1-score. We also made predictions over the additional dataset to see how the FPR varies across the different demographic groups.

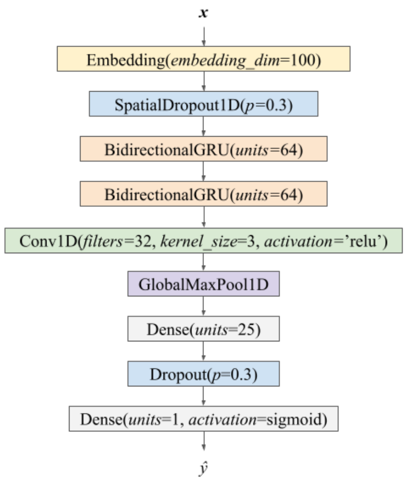
x
Embedding(embedding_dim=100)
SpatialDropout1D(p=0.3)
BidirectionalGRU(units=64)
BidirectionalGRU(units=64)
Conv1D(filters=32, kernel_size=3, activation='relu')
GlobalMaxPool1D
Dense(units=25)
Dropout(p=0.3)
Dense(units=1, activation=sigmoid)
ŷ

Figure 2: Recurrent CNN with two bidirectional Gated Recurrent Unit (GRU) layers

The performance of the models is presented in Table 1. We can see that all of our R-CNN-based models have an accuracy that surpasses PerspectiveAPI's. The best performing model consists of two bidirectional GRU layers with an accuracy of 79.68% and an F1-score of 0.86. The architecture of the model is shown in Figure 2.

However, the results show that the neural network-based models are more biased than the simpler models evaluated earlier in this paper, with our best model having an AA FPR of 0.27 and an FPR standard deviation of 0.11. The FPR on the AA class is amplified by the other neural language models as well, thus increasing the overall FPR standard deviation. This provides more evidence for the bias-accuracy trade-off.

4.2 Penalty-based debiasing method

We will explore a method that penalizes the model when it makes a toxicity classification on a tweet that is highly indicative of linguistic style. This idea is motivated by the presence of a correlation between linguistic style and toxicity seen from the consistently higher FPR of tweets from the AA demographic group. We would like to test the hypothesis that "truly" toxic tweets are more likely to be written in a more race-agnostic or general dialect. To do this, we propose a modified inference process involving a bias penalty term for our toxic speech classification pipeline shown in Figure 3.

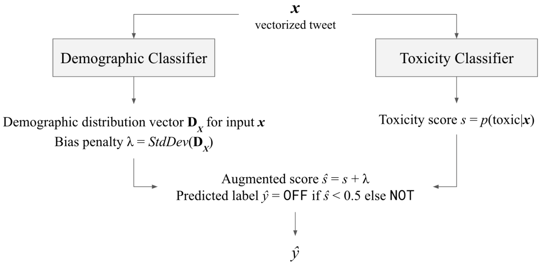
$\boldsymbol{x}$
vectorized tweet
Demographic Classifier
Toxicity Classifier
Demographic distribution vector $\mathbf{D}_x$ for input $\boldsymbol{x}$
Bias penalty $\lambda = StdDev(\mathbf{D}_x)$
Toxicity score $s = p(\text{toxic}|\boldsymbol{x})$
Augmented score $\hat{s} = s + \lambda$
Predicted label $\hat{y}$ = OFF if $\hat{s} < 0.5$ else NOT
$\hat{y}$

Figure 3: Penalty-based inference

We have trained a demographic classifier using Logistic Regression on the TwitterAAE dataset to an accuracy of around 72% that outputs a vector $\mathbf{D_X}$ containing the probabilities of a tweet coming from a user belonging to a particular demographic group. A tweet that has an agnostic linguistic style will have a vector with uniform distribution. Therefore, for each toxicity score s that is output from our R-CNN offensive speech classifier, we will add a bias penalty λ computed through the standard deviation of the vector $\mathbf{D_X}$. This implies that if a tweet is highly indicative of a certain linguistic style, we will push the toxicity classification towards the non-offensive label. The final predicted label will depend on the augmented score $\hat{s}$.

We tried this method with the two best performing R-CNN models to check whether or not the FPRs across the various demographic groups will drop, particularly in the AA class. From the results reported in Table 1, we notice that in both models the standard deviation of the FPRs reduces, with a drop of more than 2% in the FPR of the AA class. We observed a small accuracy drop on the R-CNN model with two bidirectional GRU layers, however there is an increase on the R-CNN model with a single bidirectional GRU layer. Further work should involve investigating the effect of this method on accuracy across various models with different architectures and data distributions, as well as exploring other metrics to compute the bias penalty λ, such as using the variance of $\mathbf{D_X}$ instead of the standard deviation.

5 Discussion

From the results presented in both the basic and advanced analysis, we can observe that one of the biggest ethical implications of using machine learning to combat abusive language is the performance disparity across the linguistic styles adapted by users of different demographic groups. Thus, carelessly deploying a biased hate speech detection model for downstream NLP tasks can negatively impact users who adapt the African American linguistic style in their writing. From further evaluation, we notice that most of our models are very likely to flag a tweet as toxic if it contains profanity. However, tweets containing profanity are not necessarily offensive in nature. This largely depends on how we define offensive language: Davidson et al. (2017) suggests that the most telling sign of offensiveness is whether it targets disadvantaged social groups in a potentially harmful manner.

The annotation process of the dataset is also something worth investigating. A qualitative study done by Zhou et al. (2021) shows the presence of many annotation errors in a similar offensive speech dataset (Founta et al., 2018). Offensiveness is inherently subjective; certain demographic groups are likely to have different offensive speech thresholds. Therefore, the demographic of the annotators performing the offensiveness annotation on the Twitter corpus should also ideally be balanced.

6 Conclusion

From the analysis of results and discussion above, it is evident that one of the main challenges of developing a toxic language detection system is the performance disparity across various user demographics, potentially having a disproportionate negative impact on minority user demographics. Therefore, careful considerations need to be taken when deploying such models at a global scale. Future work should aim to explore methods to mitigate biases learned by the model and decouple toxicity classification and linguistic style.

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. *CoRR*, abs/1706.00188.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.

KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *CoRR*, abs/1703.04009.

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*.

Jessica Guynn. 2020. What civil rights groups want from facebook boycott: Stop hate speech and harassment of black users.

Yann LeCun, Y Bengio, and Geoffrey Hinton. 2015. Deep Learning. *Nature*, 521:436–444.

John T Nockleby. 1994. Hate speech in context: The case of verbal threats. *Buff. L. Rev.*, 42:653.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Danyaal Yasin. 2018. Black and banned: Who is free speech for?

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. In *Proceedings of NAACL*.

Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2021. Challenges in automated debiasing for toxic language detection. *CoRR*, abs/2102.00086.