Moderating Hate Speech on Social Media

DATASCI266 Final Project, Summer 2023

Anahit Hovhannisyan
ahovhann@ischool.berkeley.edu

Mark Haase
mehaase@ischool.berkeley.edu

Zachary Galante
zgalante@ischool.berkeley.edu

Abstract

In an effort to help moderators identify hate speech online and facilitate human review of hateful language, our research focuses on classifying hate speech and explaining hate speech classification decisions through random forest, CNN, and BERT models. Using a labeled dataset from Davidson et. al. [1] consisting of 24,783 tweets, which we balance into 2,860 total tweets split evenly between hate speech and non-hate speech classes, we construct, evaluate, and tune hyperparameters for several architectures. We find that the CNN model with BERT embeddings as inputs is best at classifying hate speech tweets with a recall of 0.8500 and an F_1 score of 0.8278. Additionally, we find that the 12 attention heads from the BERT model are successful at identifying hateful key-words from tweets with a ROUGE1 score of 0.6576.

1. Introduction

Modern social media sites such as Twitter (X), Reddit, Facebook, et. al. offer exciting new ways for people to form communities of interest, communicate, and exchange ideas. Social media also has significant drawbacks, such as exposing vulnerable populations to hate speech and targeted threats. The Anti-Defamation League reports [5] that 27% of survey respondents experienced “severe online harassment” in 2021.

Our research focuses on the task of moderating hate speech on social media. The end goal is to provide tools to help moderators automatically (a) identify hate speech on their platform and (b) facilitate human review by highlighting words or tokens in the content that the model considers most significant with respect to the classification decision. Building explanatory mechanisms differentiates our research from most of the academic literature, which tends to focus on classification without explanation.

Content moderation presents complexities around error analysis. False negatives (failing to take down hate speech) do not have the same cost as false positives (taking down benign speech). The best balance between precision and recall is subject to debate. From the perspective of civil rights, we want to maximize recall, i.e. remove as much hate speech from the platform as possible. On the other hand, the civil liberties perspective seeks to maximize precision, i.e. to avoid the erroneous removal of permitted speech. Our research does not take an opinionated stance on this debate. While we report classification results in terms of F_1 score, we also report precision and recall to support more generalized F_β scores.

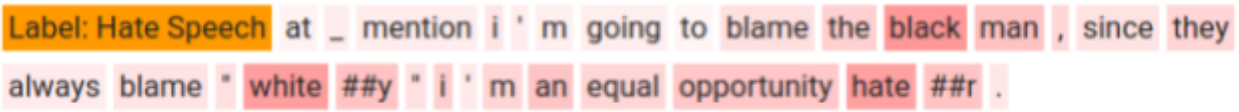

Figure 1: A tweet from the hate speech dataset with color coding indicating the cumulative attention paid to individual tokens across BERT's 12 attention heads.

Our work also incorporates advancements in the state-of-the-art for natural language classification. Prior research on classification of hate speech on social media [1] uses approaches that were mainstream in 2017, such as logistic regression and support vector machine (SVM) classifiers. In recent years, large-language models (LLMs) with pretrained weights and fine tuning have accomplished new state-of-the-art performance on a wide variety of benchmark tasks. We base our experiments on a hate speech dataset developed by Davidson et. al. [1] consisting of Twitter messages. We investigated the performance of modern architectures such as (Convolutional Neural Networks) CNNs and transformers. In particular, we investigate the applicability of BERT and other pretrained models in the BERT family. We compare these modern approaches to a baseline random forest classifier.

Hate speech is strongly associated with specific pejorative terms, such as epithets and slurs. One of the most interesting and crucial questions for building high quality classifiers for hate speech is to what extent hate speech correlates with the presence or absence of these pejoratives. For example, if hate speech correlates strongly with a small number of specific terms, then we would expect simple classifiers to perform very well, such as our baseline random forest. On the other hand, if hate speech can be constructed without using pejoratives, or benign speech can reference pejoratives, then we expect that models with deeper semantic capabilities will perform better. See **Figure 1** for an example of a tweet that does not use any strong pejoratives and yet is still labeled as hate speech. This example suggests that some contextual tokens such as "blame" and "hate" might play a useful role in classification.

The correlation of pejorative terms to hate speech classification is important in the context of moderating social media due to the scale that social media platforms operate at. When processing a hundred million messages per day, going from a simple classifier to a large transformer architecture would result in an order of magnitude increase in both capital and operating expenses.

2. Background

Hate speech is difficult to define formally, and recognizing hate speech is a challenging task even for the human experts who moderate social media sites. Hate speech can take on many shades, and it can be difficult to discern when offensive language crosses the line and becomes hate speech.

Davidson et. al. [1] focus on these core issues. They define hate speech in a careful and pragmatic manner so that they can distinguish between offensive speech and hate speech: "we define hate speech as language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group." This definition has two important aspects. First, it does not depend on the use of pejoratives; in fact, it allows for pejorative words to be used in non-pejorative ways. Second, the definition focuses on the speaker's intended effect, i.e. to

denigrate a target group. Both of these considerations make the classification task a difficult one.

Our research uses the dataset from Davidson et. al. They gathered this dataset using a crowd-sourcing platform called CrowdFlower. The annotators were trained on the definition of hate speech and then coded each tweet into one of three classes: hate speech, offensive speech, and neither. Each tweet was labeled by at least 3 different annotators; the ground truth is determined by the class receiving the most votes.

Their best model is a logistic regression with TF-IDF features and L2 regularization, with an F_1 score of 0.90. Their results are not directly comparable to ours, because they are using a three class problem and have heavy class imbalance, whereas we focus on a binary classification problem with balanced classes. Still, their work influenced the creation of our baseline model, a random forest classifier with term frequency features.

Our approach differs in two key ways. First, we are using more powerful models that were not available in 2017. Second, we are focused on the specific use case of human moderators employed by a social media platform. To that end, we collapse the 3-class problem into a binary problem (is it hate speech or not?) and investigate methods for assisting the human moderator by highlighting the tokens associated with a classification decision.

Several other papers are relevant. Fortuna et. al. [2] combines the dataset from Davidson et. al. with five other datasets for a larger corpus in a meta study about how hate speech is defined, how datasets are coded, and making recommendations for future data collection efforts. While this meta study is broader in scope than our present work, they do provide useful ideas for preprocessing of social media text (such as normalizing usernames and hashtags).

Pramanik et. al. [3] investigate harmfulness in internet memes. Their approach is multimodal (both text and images) and takes a wider scope of harmful content than we do, including deception, harm to public health, political harm, etc. In their setting, harmful content is not directed at a targeted group but can be directed at government institutions, corporations, or at society as a whole. They also use explanatory approaches such as LIME to evaluate which input features are most important to classification decisions. Our classification task is quite different from theirs, but we also seek explanatory methods using model outputs such as attention or convolutional weights.

Nguyen et. al. [4] builds a pre-trained language model based on BERT using a corpus of 850m tweets. This work highlights the effectiveness of domain-specific language models, which is an aspect that we incorporate in our work.

3. Methods

3.1 Initial Preprocessing and Baseline Model

Before creating a baseline model, we perform several preprocessing steps on our raw data. As shown in **Figure 3** in the appendix and **Table 1** below, the dataset is highly unbalanced, with only 5.77% of rows containing the label of interest (*Hate Speech*). To address this, we balance the data by combining and downsampling the *Neither* and *Offensive Language* classes to match the number of items in the *Hate Speech* class speech. We then build features by tokenizing the text and computing vectors of token counts for each tweet. Using the preprocessed data, we create a baseline random forest classifier using default hyperparameters. The baseline model performance is F_1=0.7792. Our objective with

the remaining models is to get a higher F_1 score and also to implement explanatory mechanisms.

Class	Count	Percentage
Hate Speech	1430	5.77%
Offensive Language	19190	77.43%
Neither	4163	16.80%
Total	24,783	100%

Table 1: Summary of the original dataset containing three classes with a total of 24,783 samples.

Masking Usernames.
We run an experiment where we mask usernames that appear in the text by replacing them with the special token “AT_MENTION”. We do this to avoid potential data leakage from accounts with a history of offensive tweets, and also to work around the issue of out-of-vocabulary tokens that could arise by parsing usernames. Our experiment compares the random forest performance using both unmasked and masked usernames; we find that masked usernames actually improves F_1 score.

3.2 Embedding Based Models:

Building off the preprocessing and baseline models, we use a pre-trained BERT model to tokenize and create embeddings for the tweets. These embeddings are used in a variety of models, including an improved version of our baseline model, and all subsequent BERT-based models. Using the BERT architecture, we took several approaches to create the following models.

BERT using the [CLS] token.
This model uses the learned BERT [CLS] token in the first of two hidden layers before passing results to the classification layer.

BERT with CNN head.
Our next experiment adds a CNN head on top of the last hidden layer of BERT. We accomplish this by passing the outputs of the CNN pooling layer as input to the BERT model.

BERTweet.
Along with the default BERT model, we also use BERTweet, another BERT model pre-trained on tweets. For this model we use 2 hidden layers with the addition of L2 regularization and a dropout layer.

Measuring Offensive Words.
In addition to evaluation classification performance using the F_1 score, we also want to evaluate the explanatory mechanism. For this purpose, we take a random sample of 100 hate speech tweets from the test data set and manually create reference strings containing the words most indicative of hate speech for use with the ROUGE metric.

For the BERT model, we take the final layer’s attention weights for the [CLS] token and sum across all 12 attention heads to get a 1x128 vector of attention weights corresponding to the attention that the [CLS] token pays to all of the tokens in the input. We take the slice of the vector that corresponds to actual input tokens (i.e. ignoring the [CLS] token itself as well as [PAD] tokens) and then normalize to a unit-length vector. We include all tokens that are above a chosen threshold (the threshold is determined empirically) in a candidate string and use ROUGE to compare that candidate to the reference that we coded. The BERT attention approach results in ROUGE1=0.6576.

For comparison, we also experiment with using the activations of the convolutional layer in the BERT + CNN head model to determine model importance. While this approach works to some degree, it is not as good as the BERT attention heads, resulting in ROUGE1=0.5389.

4. Results & Discussion

Each model is trained using 80% of the cleaned and balanced data, and validated using the remaining 20%. In this study, we have two goals. The first goal is to correctly predict whether a tweet contains hateful language or no hateful language (results shown in **Table 2**).

Model	Precision	Recall	F_1 Score
RF (baseline)	**0.8108**	0.7500	0.7792
RF w/ BERT embeddings	0.6486	0.6000	0.6234
CNN w/ embedding layer	0.7867	0.8036	0.7951
CNN w/ BERT embeddings	0.8068	0.8500	**0.8278**
BERT w/ CLS Token	0.7587	0.8536	0.8034
BERT, max token = 80	0.7638	0.8429	0.8014
BERT, learning rate = 0.001	0.4900	**1.0000**	0.6600
BERTweet	0.7333	0.9429	0.8250

Table 2: Precision, recall, and F_1 score for models predicting whether tweets in our dataset are hateful or not hateful.

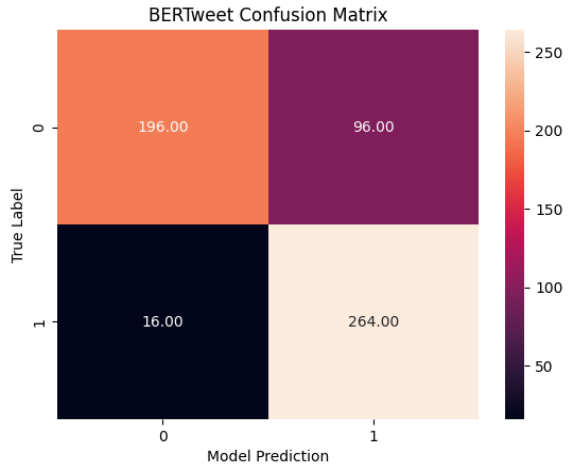

Figure 2: Confusion matrix of BERTweet results showing high correlation between true labels and model predictions.

The baseline model uses default parameters in the Sklearn implementation of random forest and is trained on bag-of-words features. It performs reasonably well on its own with an F_1 score of 0.7792. This may be due to the language used in hate speech classified tweets. **Figure 6** in the Appendix shows a list of the most frequent words from hate speech tweets that are not found in non-hate class tweets as frequently. This list consists of key words with hateful connotations, which may explain why a bag-of words approach is successful at classifying hateful vs non-hateful tweets.

The CNN model with BERT embeddings performs the best at classifying hateful vs non-hateful tweets with a recall of 0.8500 and F_1 score of 0.8278. We believe that the CNN model performs better with BERT embeddings compared to a general bag-of-words vectorization method because it can leverage BERT's pretraining to better understand which words have hateful connotations. The next best performing model is the BERTweet model with a recall of 0.9429 and F_1 score of 0.8250. This aligns with our expectations, given that BERTweet is pre-trained using tweets. The style of language used on social media is distinct enough from many other corpora; it may have a different distribution than the general English language, and this may not be represented well in BERT's pretraining. On the other hand, we find that the random forest with BERT embeddings falls short in extracting additional context with an F_1 score of 0.6234 compared to 0.7792 for the baseline. We suspect this may be due to the binary decision making used in a random forest model. Thus, subtle context weights provided by BERT may not be as easily discernible as a black and white bag-of-words embedding. Lastly, hyperparameter tuning performed on the random forest model with BERT embeddings, shown in **Table 4** in the Appendix shows no additional improvement to the F_1 score.

The second goal is to identify where the hateful language is in a given tweet, i.e. extracting context and meaning (results shown in **Table 3**).

Model	Precision	Recall	F_1 Score	ROUGE1
BERT w/ attention outputs	0.7796	0.8714	0.8229	**0.6576**
BERT w/ CNN head	0.7760	0.8786	0.8241	0.5389

Table 3: Precision, recall, F_1 score and ROUGE1 score for models identifying hateful language in tweets classified as hateful.

From the F_1 score, the two BERT models are similar at predicting where tweets are hateful. However, the attention outputs from the BERT model are more successful at matching our 100 hate speech references with a ROUGE1 score of 0.6576. **Figure 1** shows an example output from this model with darker colors corresponding to higher attention weights. We suspect this is due to better context extracted from attention rather than CNN filters.

5. Conclusion

In conclusion, we find that the CNN model with BERT embedding inputs is best at classifying tweets containing hateful language. An alternative to the CNN model with BERT embeddings is the BERTweet model. Given the nuances used in social media language, the BERT embeddings provide the CNN model with enhanced context clues to help improve its filters while the BERTweet model is pre-trained on similar language which gives it its advantage. In addition, we found that using BERT's attention heads is successful in highlighting which tokens are most indicative of hate speech, with a ROUGE1 score of 0.6576. For next steps, we would start with testing our top performing models with an updated dataset consisting of tweets from 2022-2023. We would also explore additional data cleaning methods such as special tokens, emojis, and URLs. Given the constantly changing formats and language used on social media, we are interested in evaluating our findings on current data.

References

[1] T. Davidson, D. Warmsley, M. Macy, I. Weber. Automated Hate Speech Detection and the Problem of Offensive Language. In *International AAAI Conference on Web and Social Media*, 2017.

[2] P. Fortuna, J. Soler-Company, L. Wanner. Toxic, Hateful, Offensive or Abusive? What Are We Really Classifying? An Empirical Analysis of Hate Speech Datasets. In *Conference on Language Resources and Evaluation*, 2020.

[3] S. Pramanik, D. Dimitrov, R. Mukherjee, S. Sharma, Md. S. Akhtar, P. Nakov, T. Chakraborty. Detecting Harmful Memes and Their Targets. In *ACL*, 2021.

[4] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English Tweets. In *Association for Computational Linguistics*, November 2020.

[5] Anti-Defamation League. Online Hate and Harassment: The American Experience 2021. https://www.adl.org/resources/report/online-hate-and-harassment-american-experience-2021

Appendix

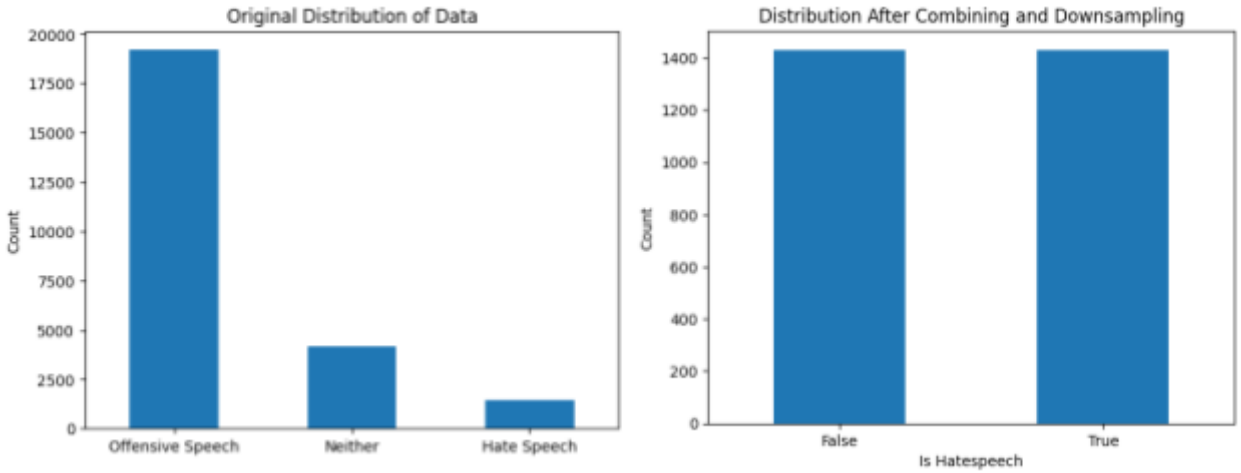

Figure 3: The distribution of the original dataset over 3 classes (left), and the distribution after combining into binary classes and downsampling (right).

Model	Precision	Recall	F_1 Score
Baseline: Random Forest	0.8108	0.7500	0.7792
Random Forest w/ BERT Embeddings	0.6486	0.6000	0.6234
Random Forest w/ BERT Embeddings (5 depth, "sqrt" num of features)	0.6256	0.4357	0.5137
Random Forest w/ BERT Embeddings (5 depth, "log2" num of features)	0.6054	0.4821	0.5368
Random Forest w/ BERT Embeddings (5 depth, 1000 num of features)	0.6341	0.4643	0.5361
Random Forest w/ BERT Embeddings (10 depth, "sqrt" num of features)	0.6695	0.5571	0.6082
Random Forest w/ BERT Embeddings (10 depth, "log2" num of features)	0.6216	0.4929	0.5498
Random Forest w/ BERT Embeddings (10 depth, 1000 num of features)	0.6711	0.5464	0.6024
Random Forest w/ BERT Embeddings (30 depth, "sqrt" num of features)	0.6223	0.6179	0.6201
Random Forest w/ BERT Embeddings (30 depth, "log2" num of features)	0.5983	0.5000	0.5447
Random Forest w/ BERT Embeddings (30 depth, 1000 num of features)	0.6130	0.5714	0.5915

Table 4: Hyperparameter tuning results for the Random Forest model using BERT embeddings. Default parameters on the random forest show the highest F_1 score indicating that limiting the tree depth and the number of features used has minimal effect on the model performance.

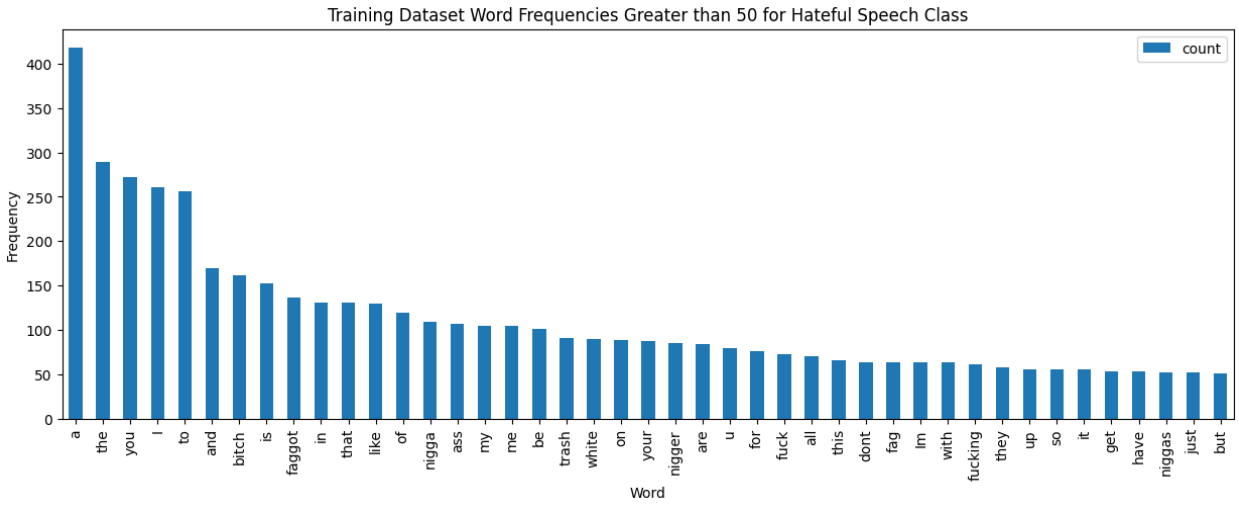

Figure 4: Most frequent words from hateful speech class tweets.

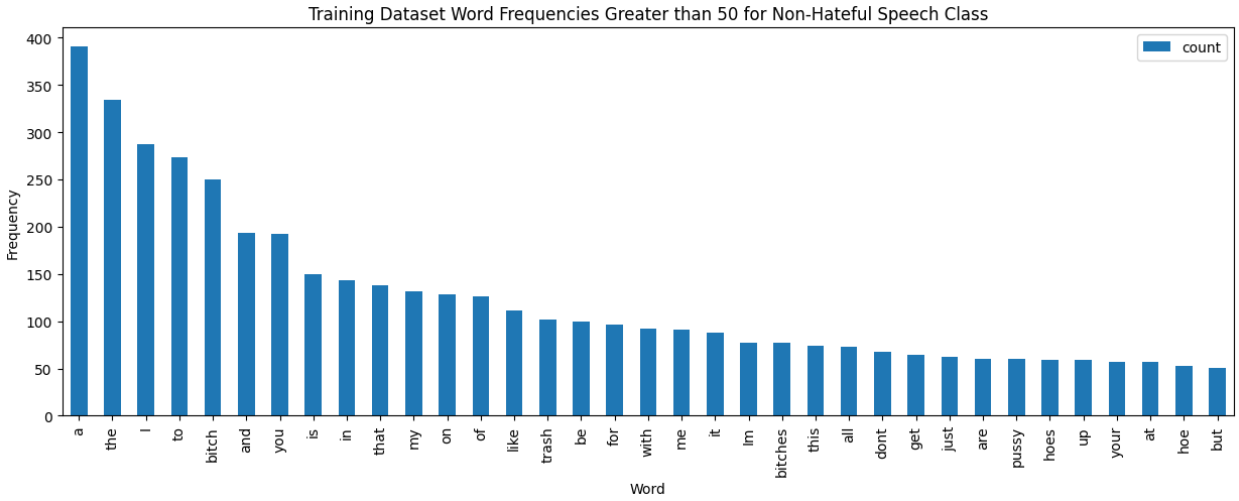

Figure 5: Most frequent words from non-hateful speech class tweets.

```
f#ggot, n#gga, a##, white, n#gger, u, f#ck, f#g, f#cking, they, so, have, n#ggas
```

Figure 6: List of frequent words in the hateful speech class, not found in the frequent non-hateful speech class.