Language Classification: Hate Speech Detection

Lauren Lawless | IST 664: Natural Language Processing | September 15, 2020

# Introduction

Hate speech is among the greatest challenges faced by social media companies in the United States. While hate speech is protected under the First Amendment to the Constitution, private companies like Twitter, Facebook, and YouTube are not bound by this legislation, and instead aim to protect their users from potential harassment by regulating the use of hate speech on their platforms. The degree to which these companies are successful in regulating hate speech, however, is highly variable. For example, there was a recent controversy involving Twitter’s content policy wherein a Twitter account that posts word-for-word copies of Donald Trump’s tweets was temporarily suspended after tweeting on May 29, 2020:

….These THUGS are dishonoring the member of George Floyd, and I won’t let that happen. Just spoke to Governor Tim Walz and told him that the Military is with him all the way. Any difficulty and we will assume control but, when the looting starts, the shooting starts. Thank you! (Gilbert, 2020)

The tweet violates Twitter’s policy against the glorification of violence, a form of hate speech (Gilbert, 2020). Twitter’s response was justified by a spokesperson who pointed out a difference in policy in handling tweets by world leaders and other users that had been in place since the summer of 2019 (Gilbert, 2020).

Twitter uses a combination of autoregulation by analytic models and manual regulation by user-reporting to identify qualifying content in violation of its policies (Twitter, 2020). The analysis that follows attempts to replicate the automatic detection of hate speech that might be used to identify content that violates Twitter’s user policies.

# Data

The data used for this classification task come from tweets containing hate speech keywords, which often includes language that is classified as offensive but not hate speech, a distinction which, historically, has presented a challenge to lexical detectors (Davidson, Warmsley, Macy, & Weber, 2017). A random sample of 25,000 of the original 85.4 million tweets were retained for analysis and manually annotated by at least three CrowdFlower workers, where the majority decision became the final annotation. The published data were annotated using three classes (hate speech, offensive language, and neither), which was converted to a binary annotation of either hate speech or not hate speech for this analysis. The annotations had an intercoder agreement score of 92%, and tweets that did not have a majority decision for annotation were excluded, leaving a dataset of 24,802 tweets, 5.77% of which are classified as not hate speech.

# Experimentation

Since the dataset is so unbalanced between positive (hate speech) and negative (not hate speech) observations, the baseline accuracy (if all examples were classified as not hate speech) is 94.23%. In terms of text classifiers, this accuracy is artificially high, and it is not helpful to our task to assume that no tweets contain hate speech, so measures of precision (rate of success among positive predictions) and recall (rate of success among positive examples) will be more important for evaluation of the experiments.

Each experiment uses the multinomial Naïve Bayes algorithm with 10-fold cross-validation for the model with various combinations of factors.

The tokenizer used for the tweet data is NLTK’s TweetTokenizer, which performs particularly well on tweets containing emojis, by separating them into individual tokens, even when they are not separated by spaces, resulting in aggregated data that does not differentiate between the use of “😂” and “😂😂😂😂😂😂😂😂😂😂😂😂😂😂” as unique tokens, for example.

## Experiment 1

The first experiment is the most basic in approach. It identifies the top 2,000 most common words in the tokenized tweet corpus of 42,993 tokens, with the only data transformation being the conversion of emojis from html format.

Table 1: Results of Experiment 1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | Not | Hate Speech |
| Actual | Not | 22,840 | 513 |
| Hate Speech | 1,020 | 410 |
|  | Accuracy | 0.9381 |  |
|  | Precision | 0.4442 |  |
|  | Recall | 0.2867 |  |
|  | F1 | 0.3485 |  |

While the accuracy is lower than the baseline, precision and recall are greater than 0, which is more desirable than the artificially high accuracy of assuming all negative classifications.

## Experiment 2

The second experiment transforms the tweet text with the following operations:

* All text lowercase
* Replace instances of other users’ Twitter handles with *USERNAME\_MENTION*
* Replace instances of *“USERNAME\_MENTION:* with *RETWEET\_SOURCE* in cases where the mention is due to a quote tweet, where a user retweets the source while adding personal commentary
* Replace instances of links with *URL\_LINK*

The resulting corpus contains 22,735 unique tokens: nearly half the size of the Experiment 1 corpus. The top 2,000 tokens are used as factors.

Table 2: Results of Experiment 2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | Not | Hate Speech |
| Actual | Not | 22,847 | 506 |
| Hate Speech | 1,006 | 424 |
|  | Accuracy | 0.9390 |  |
|  | Precision | 0.4559 |  |
|  | Recall | 0.2965 |  |
|  | F1 | 0.3593 |  |

All performance metrics improved from Experiment 1 to Experiment 2, indicating that these corpus-generalizing operations were beneficial to the model.

## Experiment 3

Experiment 3 applies even more cleaning to the tweet text, primarily focused on abbreviations for common words, and again uses the top 2,000 common words. The additional operations include:

* Replace “\n” with a space
* Replace “&” with “ and ”
* Replace ’ (special apostrophe) with ’ (generic apostrophe)
* Replace “ or ” (opening or closing quotation marks) with “ (generic quotation marks)
* Replace … (single ellipsis character) or a sequence of two or more . with ... (a sequence of three periods)
* Replace instances of “bout” (with word boundaries on either side) with “ about ”
* Replace instances of “r” (with word boundaries on either side) with “ are ”
* Replace instances of “b” (with word boundaries on either side) with “ be ”
* Replace instances of “cant” (with word boundaries on either side and with an optional apostrophe between the *n* and *t*) with “ cannot ”
* Replace instances of “dont” (with word boundaries on either side and with an optional apostrophe between the *n* and *t*) with “ do not ”
* Replace instances of “hes” (with word boundaries on either side and with an optional apostrophe between the *e* and *s*) with “ he is ”
* Replace instances of “im” (with word boundaries on either side and with an optional apostrophe between the *i* and *m*) with “ i am ”
* Replace instances of “pls” (with word boundaries on either side) with “ please ”
* Replace instances of “plz” (with word boundaries on either side) with “ please ”
* Replace instances of “shes” (with word boundaries on either side and with an optional apostrophe between the *e* and *s*) with “ she is ”
* Replace instances of “dat” (with word boundaries on either side) with “ that ”
* Replace instances of “da” (with word boundaries on either side) with “ the ”
* Replace instances of “wont” (with word boundaries on either side and with an optional apostrophe between the *n* and *t*) with “ will not ”
* Replace instances of “w/” (with word boundaries on either side) with “ with ”
* Replace instances of “u” (with word boundaries on either side) with “ you ”
* Replace instances of “ur” (with word boundaries on either side) with “ your ”
* Replace instances of “youre” (with word boundaries on either side and with an optional apostrophe between the *u* and *r*) with “ you are ”

The new corpus size is 22,685 tokens.

Table 3: Results of Experiment 3

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | Not | Hate Speech |
| Actual | Not | 22,855 | 498 |
| Hate Speech | 1,014 | 416 |
|  | Accuracy | 0.9390 |  |
|  | Precision | 0.4551 |  |
|  | Recall | 0.2909 |  |
|  | F1 | 0.3549 |  |

This experiment has slightly lower performance compared to Experiment 2, but the differences are nearly negligible, and the cleaning will be beneficial in removing stop words.

## Experiment 4

Experiment 4 leaves all prior data cleaning intact and removes stop words, using the NLTK English stop words list. The original list was extended to include:

* .
* ,
* rt
* -
* :

These tokens all appeared in the 2,000 most common words list but are ultimately not very helpful in determining hate speech in the Twitter context.

Table 4: Results of Experiment 4

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | Not | Hate Speech |
| Actual | Not | 22,921 | 432 |
| Hate Speech | 1,064 | 366 |
|  | Accuracy | 0.9396 |  |
|  | Precision | 0.4586 |  |
|  | Recall | 0.2559 |  |
|  | F1 | 0.3285 |  |

Although these results show the highest accuracy and precision of any experiment so far, the recall and F1 score are the lowest of all experiments, indicating that removing stop words might be helpful in detecting hate speech, but the list likely removed one or more tokens that provide a benefit to the classifier.

## Experiment 5

Since hate speech involves the targeting of specific groups of people, there is reason to believe that the inclusion of pronouns in a tweet would be indicative, but pronouns were removed as stop words in the previous experiment, so Experiment 5 leaves all prior data cleaning intact and removes stop words (in addition to the extended stop words from Experiment 4), but alters the stop words list to exclude any second- or third-person pronouns before application to the tweets.

Table 5: Results of Experiment 5

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | Not | Hate Speech |
| Actual | Not | 22,909 | 444 |
| Hate Speech | 1,045 | 385 |
|  | Accuracy | 0.9399 |  |
|  | Precision | 0.4644 |  |
|  | Recall | 0.2692 |  |
|  | F1 | 0.3402 |  |

Leaving the specified pronouns in the corpus restores the recall and F1 measures nearly to prior levels and produces the highest levels of accuracy and precision seen so far.

## Experiment 6

Several hateful terms or phrases involve more than one word, so identifying their presence and impact on hate speech classification requires the inclusion of n-gram word features. In addition to the unigram features used previously, Experiment 6 uses the top 1,000 most common n-grams from the set of bigrams, trigrams, and quadrigrams. Before gathering word features for this model, additional data preparation measures were taken: multiple occurrences of “USERNAME\_MENTION” in a row were replaced with a single instance, and the TweetTokenizer parameter *reduce\_len* was set to True, meaning that instances of four or more repeated characters was replaced with three instances of the repeated character (e.g., “yesssssssss!!!!!!!!” would be replaced by “yesss!!!”).

Table 6: Results of Experiment 6

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | Not | Hate Speech |
| Actual | Not | 22,706 | 647 |
| Hate Speech | 951 | 470 |
|  | Accuracy | 0.9355 |  |
|  | Precision | 0.4254 |  |
|  | Recall | 0.3350 |  |
|  | F1 | 0.3748 |  |

This approach leads to a higher overall occurrence of predictions of hate speech, meaning that while accuracy and precision decrease slightly, recall increases significantly, leading to a record high F1 score.

## Experiment 7

The final experiments use data from Hatebase, “a service built to help organizations and online communities detect, monitor and analyze hate speech” (Hatebase, 2020). After registering for an academic license, one can use the Hatebase API to query data on hate speech. For Experiment 7, the Hatebase vocabulary was queried for all terms and associated “average offensiveness” scores with a language attribute of “eng” (English). Once obtained and saved as a comma-delimited file, the terms were loaded into the classification program. Any term with an average offensiveness score of “None” was removed, and the complete list was compared to each tweet using regular expressions to identify any Hatebase terms contained within a tweet. The average offensiveness scores of each Hatebase term were summed, and the sum was divided by the number of tokens in the tweet to obtain an average hate score. This score was added as an additional feature to the feature set used in Experiment 6.

Table 7: Results of Experiment 7

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | Not | Hate Speech |
| Actual | Not | 21,818 | 1,535 |
| Hate Speech | 790 | 640 |
|  | Accuracy | 0.9062 |  |
|  | Precision | 0.2943 |  |
|  | Recall | 0.4476 |  |
|  | F1 | 0.3551 |  |

This experiment resulted in the lowest accuracy and precision scores but highest recall so far, indicating that there is some significance to the use of the Hatebase average offensiveness scores in detecting hate speech, but this calculation may not make the best use of it.

## Experiment 8

Experiment 8 uses the same features as Experiment 7 except the denominator of the hate score calculation is the number of Hatebase terms identified in the tweet rather than the number of tokens.

Table 8: Results of Experiment 8

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | Not | Hate Speech |
| Actual | Not | 22,145 | 1,208 |
| Hate Speech | 737 | 693 |
|  | Accuracy | 0.9215 |  |
|  | Precision | 0.3645 |  |
|  | Recall | 0.4846 |  |
|  | F1 | 0.4161 |  |

Although the accuracy is not the highest observed, Experiment 8 produces the best F1 score (and thus the best combination of precision and recall. Compared to the results of Experiment 7, this indicates that the presence of hate speech relies less on the average across all words and more so on the offensiveness scores of a few words in particular.

# Limitations of the Model

A study conducted at Cornell University by the publishers of this dataset showed that there is significant racial bias against users believed to be Black in the flagging of content as hate speech on all five datasets analyzed, including the one used in this analysis (Davidson, Bhattacharya, & Weber, 2019). The researchers conclude that although Twitter is unlikely to use those same datasets in their own hate-speech detectors, the presence of the bias in all five datasets suggests that it could be a widespread issue (Davidson, Bhattacharya, & Weber, 2019). User race and ethnicity is not included as a factor in the published version of the dataset, so racial bias is able to be neither evaluated nor mitigated, but in the interest of data ethics and integrity, measures should be taken, when available, to reduce bias in the analysis by implementing checks against implicit bias, especially in text and language analysis, where the “gold standard” for classification is often highly subject to the potential biases of the data annotators.

# Conclusion

The final model, produced by Experiment 8, had the best overall performance in hate speech detection. Although the accuracy never reached the baseline of 94.23%, identifying almost half of the hate speech examples is preferable to none, however, an organization using such a model would have to consider their approach to false positives, or tweets that are classified as hate speech that should not have been (most of the free-to-use platforms have an appeals process if a user believes his or her content to be mistakenly classified).

Although n-grams through quadrigrams were used, the most helpful appeared to be unigrams and bigrams. Removing stop words allowed the model to prioritize words that were more likely to influence the score, but second- and third-person pronouns, which are usually excluded by stop words lists, were significant to the context of hate speech identification, since it is language that targets and individual or group (“you”, “them”, etc.)

The list of most helpful features for either class remained relatively constant across experiments, with the most dramatic shift occurring with the inclusion of bigrams, trigrams, and quadrigrams. The most impactful features from Experiment 8 are summarized below in Table 9. Unigram features have the prefix “V\_” and other n-grams have the prefix “N\_”.

Table 9: Most Indicative Features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Positive Class (Hate Speech) Indicators** | |  | **Negative Class (Not Hate Speech) Indicators** | |
| **log Probability** | **Feature** |  | **log Probability** | **Feature** |
| -4.5232 | V\_spic |  | 3.4506 | V\_bird |
| -3.3192 | N\_fucking\_faggot |  | 3.1946 | V\_charlie |
| -3.0499 | N\_nigger\_" |  | 2.3845 | V\_sex |
| -2.9248 | V\_niggers |  | 2.3269 | N\_USERNAME\_MENTION\_lol |
| -2.8492 | V\_#faggots |  | 2.3269 | N\_...\_" |
| -2.8233 | N\_white\_trash |  | 2.2291 | V\_😍 |
| -2.6444 | N\_faggot\_ass |  | 2.2291 | V\_birds |
| -2.6041 | V\_faggots |  | 2.1412 | V\_#yankees |
| -2.5773 | V\_whitey |  | 2.1309 | V\_$ |
| -2.5616 | V\_chink |  | 2.0449 | N\_😂\_URL\_LINK |
| -2.5308 | V\_beaner |  | 2.0392 | N\_bad\_bitch |
| -2.4438 | N\_USERNAME\_MENTION\_ugly |  | 2.0336 | V\_brownies |
| -2.4438 | V\_beaners |  | 1.9751 | V\_basic |
| -2.4148 | V\_fags |  | 1.9629 | V\_females |
| -2.3102 | V\_trailer |  | 1.9257 | V\_😎 |
| -2.2950 | V\_nigger |  | 1.8870 | V\_lost |
| -2.2496 | V\_racist |  | 1.8603 | V\_👀 |
| -2.2431 | N\_you\_faggot |  | 1.8467 | V\_bae |
| -2.2045 | V\_faggot |  | 1.8467 | N\_ain't\_loyal |
| -2.1925 | N\_black\_people |  | 1.8189 | N\_bad\_bitches |

The helpful features for the positive class are primarily slurs referring to race or sexual orientation, while the features on the negative class list are either standard offensive language or more obscure slurs that have more common alternative meanings, such as “bird” and “brownies”. These dual-purpose words likely led to much of the misclassification of positive examples seen in the models. Note that neither the hate score feature nor the pronoun features appear on the list. Although they are a helpful features, they are almost exclusively helpful when these other features are present, thus the higher magnitude log probability scores of the slur features.

While these automated models saw some success in hate speech detection, they only scratch the surface of the potential for the use of NLP in text classification, and the models in use today by social media companies are much more complex and capable.

# Works Cited

Davidson, T., Bhattacharya, D., & Weber, I. (2019, May 29). Racial Bias in Hate Speech and Abusive Language Detection Datasets. Retrieved September 10, 2020, from Cornell University: https://arxiv.org/abs/1905.12516

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. *Proceedings of the 11th International AAAI Conference on Web and Social Media*, (pp. 512-515). Montreal, Canada.

Gilbert, D. (2020, June 5). *A Twitter Account That Copies Trump's Tweets Word For Word Got Suspended Within 3 Days*. Retrieved September 8, 2020, from Vice: Trump's Tweets Word For Word Got Suspended Within 3 Days

Hatebase. (2020, September 11). About. Toronto, Canada.

Twitter. (2020). *Hateful conduct policy*. Retrieved September 8, 2020, from Twitter: https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy