A Machine Learning Approach to Identifying Hate Speech on Social Media

Jarod DeWeese

November 2020

Introduction

The rise of social media has facilitated connection and engagement between people like never before. Unfortunately, these mediums also allow for hateful and offensive language to proliferate, often at the expensive of marginilized communties that may use such platforms as a main way to connect with those of similar backgrounds. This report focuses on identifying hate speech in social media settings. (Davidson et al. (2017)). Using NLP methods and Machine Learning models, we can work towards automating this categorization.

I was inspired to choose this project after our university diversity event was disrupted by members of an organization that spammed our meetings with hate speech. A friend of mine was targeted by and deeply impacted by the hateful speech that occured during that event.

Related Work

To famialarize myself with this issue, and the work already done in the area, I looked at many papers and code snippets out there already:

In Davidson et al. (2017), the authors explain legal implications of hateful speech in parts of the world, as well as explaining how and why certain kinds of hateful speech are more likely to be seen as just offensive, but not hateful. Additionally, they explain that while just looking for offensive words may initially make it easier to identifiy hate speech, it can make it harder to identify hate speech that does not specifically use those terms.

I found Davidson, Bhattacharya, and Weber (2019) which shares 2 authors with Davidson et al. (2017). This paper specifically focuses on racial bias as it exists in datasets and models. This paper found that because of this bias, models are more likely to predict that tweets authored by a community are hateful **towards their own community**. As discussed in the paper, this could result in victims being penalized for speaking out, potentially silencing the very communties that are using these platforms to speak out. This phenomenon could be partly explained by reclamation of words, and specific vernacular a group may use within their community.

Waseem and Hovy (2016) went into detail explaining why using character n-grams can be useful, and in conjunction or in replacement of a word n-gram approach. What's more, this paper identified which of these character n-grams were most likely to indicate certain types of offensive speech.

- For the purposes of this project, we use the definition used by Davidson et al. (2017): "that it is speech that targets disadvantaged social groups in a manner that is potentially harmful to them"

Unsophisticated algorithms may actually act unfavorably to those groups who are the victims of hate speech. (Davidson, Bhattacharya, and Weber (2019))

Of the literature I reviewed that focused on automatic classification of this speech online, many found that Logistic Regression proved to be consistently effective at determining whether or not a Tweet is hate speech. Specifically, Davidson et al. (2017) used uni, bi and

trigrams of vocabulary and Part of Speech tagged features, as well as counts for urls, mentions etc.

Dataset

The dataset I used for this project consisted of two columns: tweet id and coded value.[1] Original dataset had 80,000 entries, but of those, only the contents of 63,708 could be reached using the Twitter API

The following is an exerpt from the data: (Founta et al. (2018))

Tweet ID	maj label
848306464892604416	abusive
850010509969465344	normal
850433664890544128	hateful
847529600108421121	abusive
848619867506913282	abusive

Of this data set the data breaks down as follows:

Category	Number	Percent
normal	35998	84.88%
abusive	4530	10.68%
hateful	1881	4.44%

Methodology

1. Obtain pre-annotated ID dataset
    1. I first downloaded the dataset from the source, but this dataset only contained the tweet ID and coding value.
2. Preprocessing: Obtain actual tweet information[2]
    1. Grab the corresponding tweets, using the Twitter API. Features obtained from this step include: Tweet author, text, time the Tweet was sent and other basic information.
3. Clean data in the pipeline
    - Replace mentions, URLS, hashtags, etc
    - Fix encoding irregularities
    - Fix hanging whitespace
    - Remove emojis, and put them in their own column of the DataFrame
4. Feature selection
    1. TFIDF matrix of vocabulary
    2. TFIDF of POS
    3. TFIDF of characters
    4. Number of people the tweet had mentioned
    5. Flesch reading ease score
    6. Compound Sentiment Score
5. SMOTE-y data selection

    Because there was such a wide class imbalance, I chose a roughly equivalant number of records from each of the 3 classes to train on.
6. ML

    LinearSVC and LogisticRegression proved to be the most reliable for this application. I trained the model using roughly 5,000 features per Tweet, after considering the TFIDF matrices features. After some research, I believe that LR works best for my use case.
7. Grid search for parameters

Evaluation

Because there is such wide class imbalance in the real world with only about 5% of Tweets being hateful, and 10% being abusive it would be important that our model doesn't incorrectly tag normal tweets as abusive or hateful. If the model does incorrectly misclassify large portions of the normal tweets, a moderator or support staff may entirely abandon the system. My baseline for comparison is the metrics mentioned in Davidson et al. (2017): "overall precision 0.91, recall of 0.90, and F1 score of 0.90," "the precision and recall scores for the hate class are 0.44 and 0.61 respectively" This is the baseline to which I will compare my results. If I can get a precision even close to .8 I would be very happy.

[1] The dataset is available here: https://dataverse.mpi-sws.org/dataset.xhtml?persistentId=doi:10.5072/FK2/ZDTEMN

[2] TODO ADD MY GITHUB URL HERE

Results

No part of speech

TODO INCLUDE non POS TAGGED RESULTS, AND COMPARED TO THE POS TAGGED

TODO INCLUDE chars vs no chars

POS tagged confusion matrices

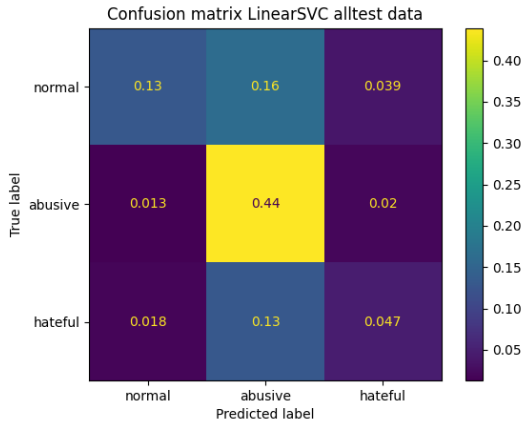

Figure 1: plotLinearSVC_tweet_coding_cleaned_no_flags_hinge_i_10000

Classification report for SVC

	precision	recall	f1-score	support
abusive	0.60	0.93	0.73	408
hateful	0.45	0.24	0.31	169
normal	0.81	0.39	0.52	288
accuracy			0.62	865

macro avg 0.62 0.52 0.52 865 weighted avg 0.64 0.62 0.58 865

Classification report for LR

	precision	recall	f1-score	support
nice	0.75	0.73	0.74	288

unpleasant 0.87 0.88 0.87 577

accuracy			0.83	865

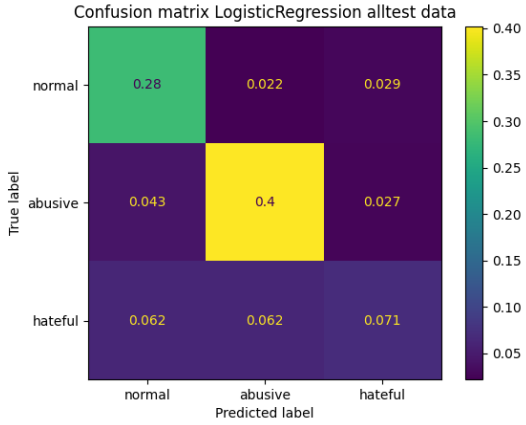

Figure 2: plotLinearSVC_tweet_coding_cleaned_no_flags_hinge_i_

macro avg 0.81 0.80 0.81 865 weighted avg 0.83 0.83 0.83 865

What I hope to get done between now and final project submission

Between now and the final project submission, I hope to be able to clean a few more data irregularities that I am experiencing, use GridSearch to better tune my parameters for my models, explore using a binary classification system instead of using the 3 class approach I'm doing now, and see if the trade-offs are worth it. I may also create a word cloud, or do further analytics on trends with these tweets. I also need to fix the formatting in this document, and make sure it adheres to AAAI standards.

Conclusion

So far I have a working model, that performs decently well. I also have some of the infrastructure in place to auto-tune parameters via GridSearch, and dump metrics into the file system for me to examine. I was originally hoping that maybe I could deploy this somewhere and be able to read in real time tweet replies to people that are known to be targets of hate speech,

and implement something like the GoodnessBot[3] that exists on Twitter.

Because of reasons discussed in Davidson, Bhattacharya, and Weber (2019), these results should be interpreted with context, and understanding of the inherent bias that exists in data collection, annotation, modeling, and context to cultural norms.

I hope to be able to do more work to find which subset of features are most effective for classifying hate speech.

Acknowledgements

- https://gist.github.com/maxogden/97190db73ac19fc6c1d9beee1a6e4fc8#file-paper-md
- https://miki725.com/2019/10/15/markdown-to-pdf-ieee.html
- Xiang (2019) whose code helped to provided useful code samples of how the techniques discussed in the research can be implemented with real model code.

References

Davidson, Thomas, Debasmita Bhattacharya, and Ingmar Weber. 2019. "Racial Bias in Hate Speech and Abusive Language Detection Datasets." *CoRR* abs/1905.12516. http://arxiv.org/abs/1905.12516.

Davidson, Thomas, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. "Automated Hate Speech Detection and the Problem of Offensive Language." In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, 512–15. ICWSM '17. Montreal, Canada.

Founta, Antigoni-Maria, 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." Root. https://doi.org/10.5072/FK2/ZDTEMN.

Waseem, Zeerak, and Dirk Hovy. 2016. "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter." In *Proceedings of the NAACL Student Research Workshop*, 88–93. San Diego, California: Association for Computational Linguistics. http://www.aclweb.org/anthology/N16-2013.

Xiang, Tong. 2019. "Hate Speech Detection for Cosc586." https://github.com/mathfather/Hate-Speech-Detection-For-COSC586.

[3] https://twitter.com/GoodnessBot/