AMITY UNIVERSITY MUMBAI

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

Department of Computer Science & Engineering

Academic Year 2020-21

A Project Report on

Automated Detection of Hate Speech

Bachelor of Technology

(Computer Science & Engineering)

Submitted By

Rittik Basu

Shaurya Sood

Sanah Saleem

Rehmat Irphany

Declaration of Academic Integrity

We declare that this written submission conveys our ideas in our own words. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/date/fact/source in our submission.

We understand that any violation of the above will be cause for disciplinary action by the institute and they can evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Rittik Basu, Rehmat Irphany, Sanah Saleem, Shaurya Sood

(Student Signature)

Date: 07-05-2021

Approval

This is to certify that Rittik Basu, Rehmat Irphany, Shaurya Sood, Sanah Saleem have satisfactorily completed their project on **Automated Detection of Hate Speech** during the academic term 2020-2021 and their report is approved for final submission.

Examiners
Dr. Deepa Parasar

Date: 07-05-2021
Place: Mumbai

Certificate

This is to certify that the project entitled "**Automated Detection of Hate Speech**" is a bonafide work of Rittik Basu, Rehmat Irphany, Shaurya Sood, Sanah Saleem submitted to the Amity School of Engineering and Technology, Amity University Mumbai in partial fulfilment of the requirement for the degree of B.Tech Computer Science & Engineering.

Dr. Deepa Parasar
(Coordinator, Department)

Dr.Shrikant Charhate
(Director, ASET)

Acknowledgements

We would like to express our special thanks of gratitude to our professor and mentor Dr. Deepa Parasar for giving us the constant motivation, support and the opportunity to work on this project on *Automated Detection of Hate Speech*. Through the course of this project we’ve learned a lot of things by doing a great deal of quality research and really educating ourselves on the topic.

We would also like to extend our sincere thanks to everyone associated with the project without whom it would not have been possible.

Abstract

A key challenge for automatic hate-speech detection on social media is the separation of hate speech from other instances of offensive language. Lexical detection methods tend to have low precision because they classify all messages containing particular terms as hate speech and previous work using supervised learning has failed to distinguish between the two categories. Therefore in order to create a hate-speech-detecting algorithm, we used Python-based NLP machine learning techniques.

- Firstly, we used a crowd-sourced hate speech lexicon to collect tweets containing hate speech keywords. The data is labelled into three categories: those containing hate speech, only offensive language, and those with neither.

- Then, using a NLP (or Natural Language Processing) technique called Tf-Idf vectorization, we'll extract keywords from the dataset that convey importance within hate speech.

- Finally, based on a machine learning technique called logistic regression, which is popular for probability calculations, we'll train the model to classify hate speech.

Table Of Contents

Introduction

In the broad spectrum, hate speech can be described as a speech act that antagonizes or marginalizes people based on their identification with a particular social or demographic group. It is the result of communication manners that compromise human dignity, equality, and human rights. Nevertheless, definitions of hate speech can be contentious and problematic, given tensions between the labeling of a harmful speech act as such, which has the potential to antagonize or marginalize an individual or group, and those speech acts that contribute to a pluralistic debate on a particular issue.

Detecting hate speech is a challenging task, however. First, there are arguments on how hate speech should be defined. This means that some content can be recognized as hate speech to some and not to others, based on their respective interpretations. Drawing upon certain definitions, we define hate speech as a 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.

Another concern that arises in the interpretation of hate speech is the potential praising of a group that is hateful. For example, praising XXX is hate speech, however praising another group can be non-hate speech. In this case, it is essential to know what crowds are hate crowds and what exactly is being praised about the crowd as some praising is unquestionably, and unfortunately, accurate. For instance, the Nazis were very efficient in terms of their “Final Solution”.

Thus, praise processing solely is, at times, difficult. For a more modern-day example, it has come to our notice that social media contributes to almost 80% of the hate crime out there. More than 8,000 tweets per second are posted every day which amounts to something like 260 billion tweets per year. This astonishing mass of text encloses an invaluable amount of information from which important insight could be deduced. The process of examining text information belongs to the area of Natural Language Processing (NPL). The purpose of this paper is to propose a practical, and easy to implement, method to distinguish hateful and prejudiced content.

Data

We begin with a hatebase twitter dataset from Kaggle which is a set of 24,802 tweets provided by Thomas Davidson. Their procedure for creating the dataset was as follows. First they took a hate speech lexicon from Hatebase.com and searched for tweets containing these terms, resulting in a set of tweets from about 33,000 users. Next they took a timeline from all these users resulting in a set of roughly 85 million Tweets. From the set of about 85 million tweets, they took a random sample, of 25000 tweets, that contained terms from the lexicon. Via crowdsourcing, they annotated each tweet as hate speech, offensive, or neither hate speech nor offensive. If the agreement between annotators was too low, the tweet was excluded from the set.

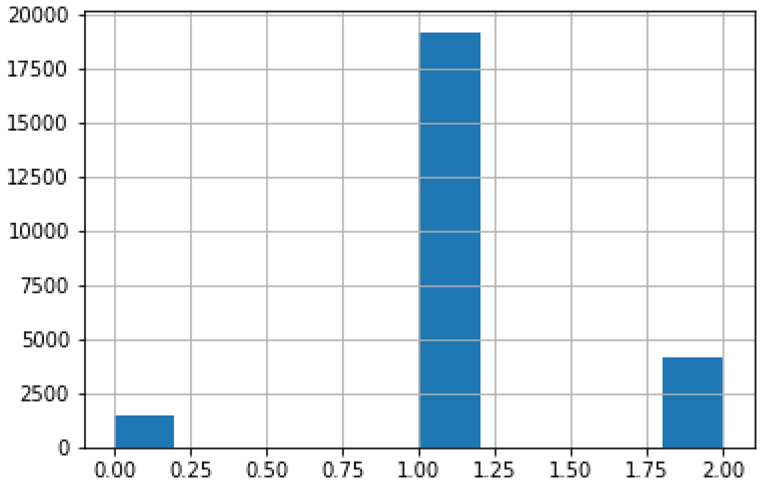


Figure 1

The histogram shown in *Figure 1* shows the distribution of tweets in the dataset. It contains about 5% of hate speech, 76% of offensive speech and 17% neither hate nor offensive speech.

The dataset is divided into 3 classes, ‘0’ for hatespeech, ‘1’ for offensive language and ‘2’ for neither. Classifying the data helps the predictive model to approximate the mapping function from input variables to discrete output variables. The main goal here is to identify which class/category the new data will fall into.

Features

Text Preprocessing

Several research studies have explained that using text preprocessing makes better classification results So, in our dataset, we applied different preprocessing-techniques to filter noisy and non-informative features from the tweets. In preprocessing, we changed the tweets into lower case. Also, we removed all the URLs, usernames, white spaces, hashtags, punctuations and stop-words using pattern matching techniques from the collected tweets. Besides this, we have also performed tokenization and stemming from preprocessed tweets.

Data Splitting

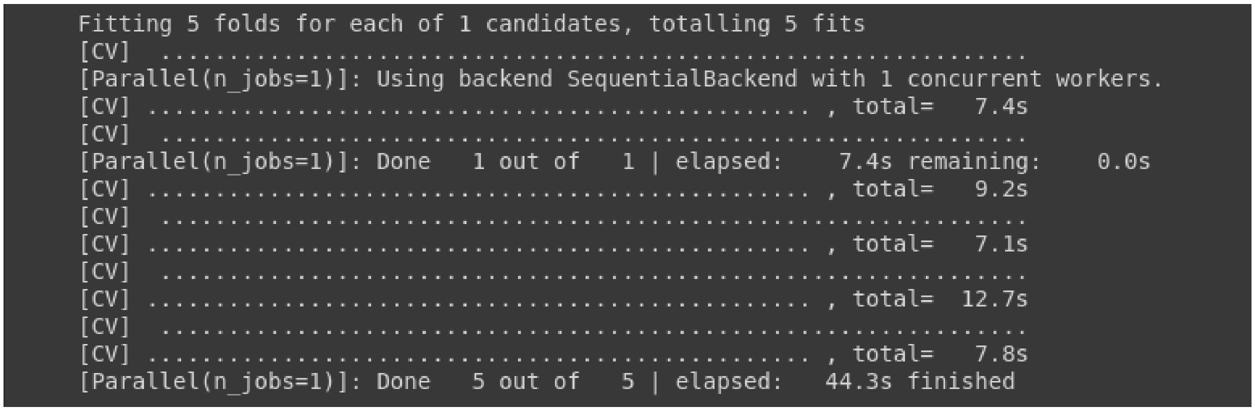
The dataset after splitting (i.e. Training set and Test set). We have used the 80-20 ratio to split the preprocessed data (i.e. 80% for Training Data and 20% for Test Data). The training data is used to train the classification model to learn classification rules.

Tf-idf classifier

The classifier performance is evaluated by calculating true negatives (TN), false positives (FP), false negatives (FN) and true positives (TP). These four numbers constitute a confusion matrix.
This technique is suitable for extracting words and converting them into numbers allowing us to train the machine appropriately.
The technique works by assigning scores to particular words in the document using a specific formula.

Model

We begin by using logistic regression with L1 regularization to reduce the dimensionality of the data. Dimensionality reduction basically reduces the number or variables under consideration and is related to feature extraction. It is useful when the data set has similar measurements with different units. Regularization is a technique used to avoid overfitting, it introduces additional information in order to solve a problem in a better way.



```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[CV]  ...............................................................
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[CV] ................................................ , total=   7.4s
[CV]  ...............................................................
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    7.4s remaining:    0.0s
[CV] ................................................ , total=   9.2s
[CV]  ...............................................................
[CV] ................................................ , total=   7.1s
[CV]  ...............................................................
[CV] ................................................ , total=  12.7s
[CV]  ...............................................................
[CV] ................................................ , total=   7.8s
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:   44.3s finished
```

Figure 1

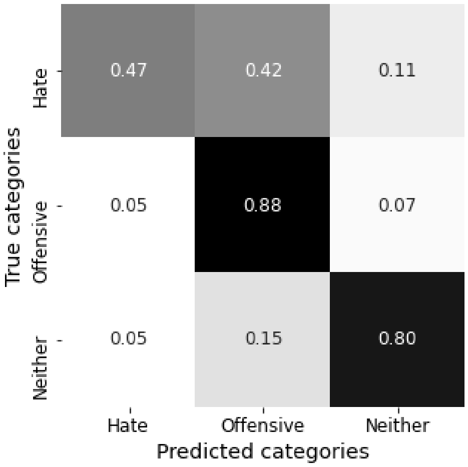
We initially started working with the Naive Bayes model as it was easier to implement but soon realised it was inefficient and not a good fit for our project. We then tested a variety of models like logistic regression, decision trees, random forests, and linear SVMs by using 5 fold cross validation as shown in *Figure 1*, holding out 10% of the sample for evaluation to help prevent overfitting. After using a grid-search to iterate over the models and parameters we find that the Logistic Regression and Linear SVM tended to perform significantly better than other models.

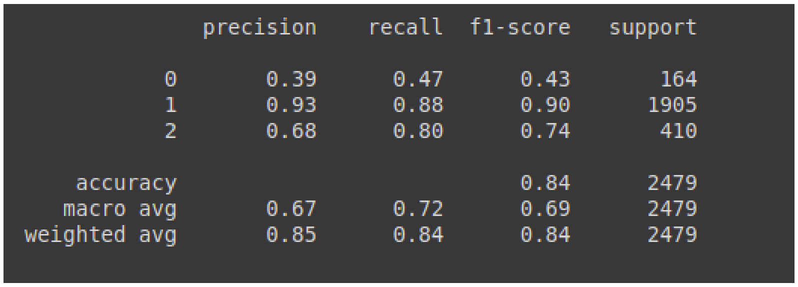
We decided to use a logistic regression with L2 regularization for the final model as it more readily allows us to examine the predicted probabilities of class membership and also upon exploring and researching about other projects which were similar to ours we saw that logistic regression with L2 regularization gave the best results.

We trained the final model using the entire dataset and used it to predict the label for each tweet. We use a one-versus-rest frame- work where a separate classifier is trained for each class and the class label with the highest predicted probability across all classifiers is assigned to each tweet

Results

The best performing model has an overall Precision of 0.85, Recall of 0.84, and an F1 score of 0.84. However, we see that almost 40% of hate speech is misclassified.Most of the misclassification occurs in the upper triangle of this matrix, suggesting that the model is biased towards classifying tweets as less hateful or offensive than the human coders.Far fewer tweets are classified as more offensive or hateful than their true category;approximately 5% of offensive and 5% of innocuous tweets have been erroneously classified as hate speech.





```
              precision    recall  f1-score   support

           0       0.39      0.47      0.43       164
           1       0.93      0.88      0.90      1905
           2       0.68      0.80      0.74       410

    accuracy                           0.84      2479
   macro avg       0.67      0.72      0.69      2479
weighted avg       0.85      0.84      0.84      2479
```

The tweets in the category that have been misclassified as hate or offensive tend to mention race,sexuality, and other social categories that are targeted by hate speakers. Most appear to be misclassifications that appear to be caused by the presence of potentially offensive language which is used in a positive sense.Some tweets contain terms that can be considered racist and sexist,terms like n*gga and b*tch but it is apparent than many Twitter users use this type of language in their everyday communications.

Turning to true hate speech classified as offensive it appears that tweets with the highest predicted probability of being offensive are genuinely less hateful and were perhaps mislabeled, for example When you realize ‘*how curiosity is a b*tch #CuriosityKilledMe’* may have been erroneously coded as hate speech if people thought that curiosity was a person.This problem illustrates the importance of taking context into account. Understanding the context of each tweet or sentence will reduce the percentage of misclassifications.

Limitations of the Model

A challenge faced by automatic hate speech detection systems is the changing of attitudes towards topics over time and historical context.

The user who posted a particular sentence may not have intended the hate speech result, but instead meant to quote someone or a movie or use it in a different context, but eventually be flagged as hate or offensive

Another remaining challenge is that automatic hate speech detection is a closed-loop system; individuals are aware that it is happening, and actively try to evade detection.Users who desired to spread the hateful messages quickly found ways to circumvent these measures by, for instance, posting the content as images containing the text, rather than the text itself. Although optical character recognition can be employed to solve the particular problem, this further demonstrates the difficulty of hate speech detection going forward. It will be a constant battle between those trying to spread hateful content and those trying to block it.

Another limitation is the flagging of offensive language as "hate". It is detected that only 5% of the offensive language is actually hate speech,so the model can be deceived by flagging "offensive" speech as "hate".
It is also observed that hate against women or misogyny is less or isn't classified as hate but rather as just offensive as compared to racist,casist language

The biggest limitations however will always remain human bias and context,all this information is scraped from words written by humans and no 2 humans can think alike regarding every matter, this difference in opinion causes the programmed model to have 2 sentences having opposite contexts or meanings but be flagged as offensive or hate.

Human tolerance for offensive language is different for every single person and therefore that will remain the biggest challenge faced by our model,as something that is offensive to Person A may not be offensive or hateful to Person B.

Another bothering fault is the model only being 80-85% accurate,this may not seem as a very large inaccuracy but when filtering millions and millions of sentences and metric data on the internet, the 15% inaccuracy can lead to no detection of massive amounts of unlawful language.

Summary

Given the legal and moral implications of hate speech it is important that we are able to accurately distinguish between hate speech and offensive language. Lexical methods are effective ways to identify potentially offensive terms but are inaccurate at identifying hate speech. While automated classification methods can achieve relatively high accuracy at differentiating between these different classes, close analysis of the results shows that the presence or absence of particular offensive or hateful terms can both help and hinder accurate classification.

Certain terms are particularly useful for distinguishing between hate speech and offensive language. While f*g, b*tch, and n*gga are used in both hate speech and offensive language, the terms f*ggot and n*gger are generally associated with hate speech.

Many of the tweets considered most hateful contain multiple racial and homophobic slurs. While this allows us to easily identify some of the more egregious instances of hate speech it means that we are more likely to misclassify hate speech if it doesn't contain any curse words or offensive terms. To more accurately classify such cases we should find sources of training data that are hateful without necessarily using particular keywords or offensive language.

People identify racist and homophobic slurs as hateful but tend to see sexist language as merely offensive.

Our results also illustrate how hate speech can be used in different ways: it can be directly send to a person or group of people targeted, it can be espoused to nobody in particular, and it can be used in conversation between people. Future work should distinguish between these different uses and look more closely at the social contexts and conversations in which hate speech occurs. We must also study more closely the people who use hate speech, focusing both on their individual characteristics and motivations and on the social structures they are embedded in.

Hate speech is a difficult phenomenon to define and is not monolithic. Our classifications of hate speech tend to reflect our own subjective biases.It is important that we are cognizant of the social biases that enter into our algorithms and future work should aim to identify and correct these biases.

Future Scope

Hate speech is a complicated phenomenon to establish and is not monolithic. Our classifications of hate speech tend to ponder our own idiosyncratic biases. People identify racist and homophobic slurs as hateful but tend to see sexist language as merely offensive. While our results show that people perform well at identifying some of the more egregious cases of hate speech, it is important that we are cognizant of the social biases that enter into our algorithms and future work should aim to identify and correct these biases.

By working with a diverse and advanced data-set we can increase the accuracy of the program as well as make a better classification between hate and offensive content. This research only used the tweets written in the English language, which can be further extended by incorporating other languages such as Hindi, Marathi, Tamil, etc. Eventually, we can develop a model which captures all hate speech content from the OSN. To build a general framework using deep learning models, the training dataset must have sufficient samples, in the future, the current dataset may be extended to achieve better accuracy

In the future, we look forward to building this project for a much larger real-time data set. We will initially start with building a website embedded with this very code and will incorporate features such as typing in a certain URL and obtaining how much hate and offensive content is present in it. Later, this code can also be implemented on certain social-web platforms where hateful and foul languages are detected even before the user can put them up on the in

Bibliography

Mr. Nemo & Data (Yoo Byoung Woo). 2019. *My First Machine Learning Project: Designing a Hate Speech Detecting Algorithm*. towardsdatascience.

Bird, S.; Loper, E.; and Klein, E. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Burnap, P., and Williams, M. L. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet* 223–242.

Djuric, N.; Zhou, J.; Morris, R.; Grbovic, M.; Radosavljevic, V.and Bhamidipati, N. 2015. Hate speech detection with comment embeddings. In *WWW,* 29–30.

Gitari, N. D.; Zuping, Z.; Damien, H.; and Long, J. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering* 10:215–230.

Hutto, C. J., and Gilbert, E. 2014. VADER: A parsimonious rule based model for sentiment analysis of social media text. In ICWSM. Jacobs, J. B., and Potter, K. 2000. Hate crimes: *Criminal Law and Identity Politics*. Oxford University Press.

Kwok, I., and Wang, Y. 2013. Locate the hate: Detecting tweets against blacks. In *AAAI*.

Nobata, C.; Tetreault, J.; Thomas, A.; Mehdad, Y.; and Chang, Y. 2016. Abusive language detection in online user content. In *WWW*, 145–153.

Pedregosa, F., et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Silva, L. A.; Mondal, M.; Correa, D.; Benevenuto, F.; and Weber, I. 2016. Analyzing the targets of hate in online social media. In *ICWSM*, 687–690.
Walker, S. 1994. *Hate Speech: The History of an American Controversy*. U of Nebraska Press.