International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 11, November-2020

Detection of Hate Speech using Text Mining and Natural Language Processing

G. Priyadharshini
Department of Computer Science and Engineering
Bharathidasan University
Tiruchirappalli, India

***Abstract*— In today's modern world, technology connected with humanity is doing wonderful things. On the other hand, people inclined to social networks where they have anonymity are bringing out the very nastiest of people in the form of hate speech. Social media hate speech is a serious societal problem which can contribute to magnify the violence ranging from lynching to ethical cleansing. One of the critical tasks of automatic detection of hate speech is differentiating it from the other context of offensive languages. The existing works to distinguish the two categories using the lexical methods showed very low performance metrics values which led to major misclassification. The works with supervised machine learning approaches indeed gave significant results in distinguishing hate and offensive but the presence or absence of certain words of both the classes can serve as both merit and demerit to achieve accurate classification. In this paper, a ternary classification of tweets into hate speech, offensive and neither is performed using multi class classifiers. Among the four classifiers: Logistic Regression, Random forests, Support Vector Machines (SVM) and Naïve Bayes. It can be seen that Random Forest classifier performs significantly well with almost all feature combinations giving maximum accuracy of 0.90 for TFIDF feature technique.**

Keywords—Hate speech; Natural Language Processing; Text Preprocessing; Feature Extraction; Text Classification; Machine Learning

I. INTRODUCTION

Social media corporations such as the facebook and twitter that are curbing the hate speech are pushed to deal with the questions of infringing on rights of free speech. These online communication platforms are pushed to deal with various standards and legal systems of the countries around the world and facing investigations by their governments. This made the social media giants to take on the responsibility of what to censor and what not to. The need for censorship lead to the formulation of baseline content which questions what is hate speech and how it differs from other instances of offensive languages. No outright definition prevails but there is a consensus that any speech referring to racism, misogyny, or homophobia constitutes hate speech. Correspondingly, when pinpointing the hate speech, there is a need to exclude some instances of other offensive language because people tend to use terms that are highly offensive but in a qualitatively different manner which has no specific hate undertone against a group. For example, the native English speakers often use words like b*tch and h*e in everyday language. Nowadays, even the song lyrics consists of slurs such as f*g and n*gga which is frequently quoted in online communication. The widespread of such kind of language increases the need to control the extent of freedom of speech in social media.

In order to scrutinize the hate speech, there is a need for some specific terminologies which help in hate speech identification. There are research works like Fortuna and Nune 2018 where certain main rules for detection have been listed. In this manner, (Waseem and Hovy 2016) have specified 11 parameters to differentiate hate speech specifically on twitter platforms. Officially, Twitter's hate speech policy elucidates as "Hateful conduct: You may not promote violence against or directly attack or threaten other people on the basis of race, ethnicity, national origin, sexual orientation, gender, gender identity, religious affiliation, age, disability, or disease. We also do not allow accounts whose primary purpose is inciting harm towards others on the basis of these categories". Based on this policy, in this paper, a model is trained to classify the tweets as hate, offensive and neither. The results showed that the feature extraction plays a dominant role in the task of hate speech detection and exposes the challenges to achieve accurate classification.

II. TEXT PREPROCESSING

A. Data

Insufficient data is one of the major issues for automatic hate speech detection in different languages. However, there are decent collections of dataset for English language; few problems persist in those available resources. To mention it,

- Unavailability of publicly accessible data
- Irregular data collection and annotation, making it tougher for regulated hate speech detection.
- Unavailability of in-depth annotation and classification, that is, in most cases the classes targeted are solely hate and non-hate although the crucial target is addressing different classes (e.g. racism, sexism, bullying).

When it comes to hate speech detection, twitter is the most relied social media as it contains enormous linguistic diversity in the content. Most of the publicly available hate speech annotated data in English are from twitter. The dataset which is used in this paper is a publicly available hate speech dataset on CrowdFlower which has been used previously in Davidson and Warmsley (2018) that consists of 25K tweets in English. This publicly available dataset consisting of 25k tweets are categorized into three classes with class labels

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 11, November-2020

such as "hate", "offensive" and "neither" by the CrowdFlower's manual coders. Based on this manually coded label, the features are constructed from training tweets of each class and used them to train a classifier

B. Data Preprocessing

In the process of detection and classification of hate speech in the social media, it is mandatory to have noiseless clean data in order to get high accuracy of the machine learning algorithms. While working with twitter dataset, the tweets are mostly brief with useless or unknown characters and used with informality.

Specifically, tweets have a distinct format, containing usernames, URL's, hashtags, which needs to be removed or parsed.

In this paper, each tweet is cleaned by removing the extra spaces, mentions, links, punctuations and numbers. The cleaned tweets are then lowercased and tokenized. Next, the stop words are removed. Lastly, the stemming is performed using the porter stemmer.

III. FEATURE EXTRACION

In the feature extraction process, values or features are derived from the input data in order to generate distinctive properties which are informative and non-redundant so that the accuracy of classification process is improved.

The initial feature extraction technique that is implemented is TFIDF representation. Term Frequency Inverse Document Frequency (TFIDF) is calculated to estimate the contribution of particular words in the hate speech data corpus. Then the sentiment analysis of the data corpus is done using the polarity scores as features. The doc2vec features are also found.

The extracted important features from the tweets using different feature extraction techniques are merged into different sets for the objective to compare and analyze the performance of different classification models with respect to each feature. .

The different combinations of features from F1 to F5 are:

F1:Only TFIDF features.

F2:TFIDF features are combined with additional features from sentiment analysis

F3:TFIDF features with sentiment scores and doc2vec

F4:TFIDF features and doc2vec

F5:Additional features from sentiment analysis and doc2vec

IV. CLASSIFICATION

As concluded in many research works, a single classifier cannot give best performance on all kinds of datasets. Mostly the hate speech detections are done by supervised classification algorithms. In this paper, four different classifiers: Logistic Regression, Random Forest, Naïve Bayes and SVM are used. These classifiers are considered as these are the ones which have been largely used in prior works. All the models were performed using scikit-learn.

V. METHODOLOGY

A. Data Splitting

In this paper, the preprocessed data is split into 80-20 ratio, that is., 80% for Training Data and 20% for Test Data. The Table I shows the class-wise distribution of the overall dataset as well as data set after splitting. The 80% of training data is used to train the classification model to learn classification rules and the 20% of test data **is** further used to evaluate the classification model.

TABLE I. Data Split

	Class	Total Instances	Training Set	Test Set
0	**Hate Speech**	1430	1140	290
1	**Offensive Speech**	19190	15858	3832
2	**Neither**	4163	3328	835
	Total	24783	19826	4957

B. Experimental Setup

The experimental process is started by preprocessing the text. In this preprocessing step, the tweets in the dataset are tokenized, lowercased and cleaned by removing the stop words, extra spaces, mentions, links, punctuations and numbers. As for the next step, three types of master feature representations namely TFIDF, Sentiment Scores and Doc2vec are extracted from the preprocessed data and combined into different sets. Hence, a total of five feature representations F1, F2, F3, F4 and F5. Lastly, four different ML algorithms were applied to the created five feature vectors of the preprocessed data. Hence, overall 20 analyses (5 feature vectors x 4 ML algorithms) were evaluated to check the effectiveness of classification models.

TABLE II. Confusion Matrix (Features : TFIDF, Classifier : Random Forest)

Class	Classified as		
	Hate	Offensive	Neither
Hate	**0.16**	**0.74**	**0.10**
Offensive	**0.02**	**0.97**	**0.01**
Neither	**0.00**	**0.32**	**0.68**

TABLE.III Accuracy of the Classifiers with five different feature sets

Classifiers	F1	F2	F3	F4	F5
LR	0.8910	0.8946	0.8946	0.8792	0.5186
RF	0.9009	0.8842	0.8739	0.8728	0.7486
NB	0.6491	0.6501	0.6501	0.6475	0.7103
SVM	0.8932	0.8914	0.8900	0.8832	0.7621

TABLE.IV Recall of the Classifiers with five different feature sets

Classifiers	F1	F2	F3	F4	F5
LR	0.64	0.646	0.646	0.638	0.316
RF	0.71	0.626	0.610	0.619	0.409
NB	0.55	0.553	0.553	0.556	0.627
SVM	0.686	0.686	0.686	0.629	0.329

VI. EXPERIMENTAL RESULTS

In the process of classification, the two performance metrics are taken into account to evaluate the classifiers with correspondence to different feature combination sets. These metrics are accuracy and recall. In all the analysis, the Random Forest algorithm works considerably well with almost all the feature sets particularly with F1 feature reaching the highest accuracy of 0.9009. However, the performance metrics are highly influenced when TFIDF scores are not included in the F5 feature set. This is evident by accuracy and recall decreasing to 0.7486 and 0.409 respectively for F5 with Random Forest. Similarly, Logistic Regression shows significant performance for all feature combinations except for feature F5 as the precision, recall and F1 score for "hate label" results attain almost a zero. Support Vector Machines also works the same way as Logistic Regression by giving extremely low metrics scores for F5.

On the other hand, Naïve Bayes was found to have under performance in the classification of hate and offensive labels with all the feature sets but unlike the other classifiers, it performs pretty well for F5 having high accuracy than with the other feature sets.

In feature representation, TFIDF scores obtained the best performance as compared to other combined features. Sentiment scores has also worked significantly well in the process of identifying hate speech from other instances of offensive language. Doc2vec is found to be trivial as it makes far less difference even when it is removed from the feature combinations. F5 feature set, from which the TFIDF scores are excluded, shows poor performance for all classifiers except for the Naïve Bayes classifier. In text-classification models, the Random forest classifier performed the best among all the four classifiers. However, the Logistic Regression and SVM classifiers results were lesser than Random Forest results but their results were much better than Naïve Bayes results.

Furthermore, Table.II shows the confusion matrix of the best-performing model which is the Random Forest classifier using TFIDF features. In this confusion matrix, out of 290 tweets belonging to hate speech class, only 16% was correctly classified and the remaining 84% were misclassified. Out of these 84%, 74% were incorrectly classified as offensive and only 10% were falsely classified as Neither. The 3832 instances belong to the second (Offensive) class, almost 97% of the tweets were correctly classified leaving 3% misclassified. Among these 3%, 2% was misclassified as hate speech and 1% were misclassified as neither. The remaining 835 Neither instances out of 4957 test set, the Random Forest classifier correctly classified the 68% tweets as Neither. 32% of instances were misclassified into offensive speech and surprisingly none of the tweets were falsely classified as hate speech, that is, 0% of neither tweets were misclassified as hate speech.

VII. DISCUSSION

In this paper four classifiers are evaluated over Five different feature sets, giving 20 different analyses over hate speech dataset containing three classes. Our experimental results showed that the Random Forest algorithm with the TFIDF technique showed the best results. The reasons behind the results are analyzed.

A. Feature Extraction

In the process of text classification, feature selection is an important aspect. In this paper, five different combinations of feature extraction techniques such as TFIDF, Sentiment Scores and Doc2vec are used. From the experimental results, it is certain that the Feature F1 containing only the TFIDF scores outperformed. This may be because some words which are considered as hate or slang are used so frequent in day-to-day lives. The tweets containing these words might not actually be deliberated as a hate speech. In order to detect whether the slang/abusive word is intended as hate or not, TFIDF approach is used since it assigns a low weightage to such words. Also, the feature combination without TFIDF scores showed very poor performance. In addition to that, several research papers have experimented and proved that TFIDF feature extraction technique has given higher performance than other binary and term frequency feature extraction methods [7]. The sentiment scores of this dataset are analyzed with the domain specific dictionary. A collection of words and phrases that were identified as hate speech by web users compiled by Hatebase.org is the hate speech lexicon used as the domain specific dictionary. This might be the reason for the sentiment scores' considerable contribution for the classification of hate speech and offensive language.

On the other hand, Doc2vec does not create any improvement when it is removed from the feature combination. The size of the dataset is limited to 25K tweets which might be the reason for Doc2vec's performance.

B. Machine Learning Classifiers

There is an absolute necessity to compare different classification algorithms in order to identify the best performing classifier in the dataset taken. There are also many research papers which proved that there is no one permanent algorithm which gives high performance on all dataset. Hence, on our dataset, we used four different ML algorithms.

A large class imbalance dataset is used in this paper. In this case, considering only the accuracy may be misleading. Thus, recall is considered which gives the proportion of hate tweets that are correctly classified. If the recall is high, it refers that a large number of hate tweets are correctly classified as hate tweets, thus favoring the classification model. Hence the models which have high scores for both accuracy and recall would be chosen as best performing model.

From the experimental results, although SVM and Logistic Regression has good results, Random Forest classifier is chosen as the best performer as it has the highest values for both the accuracy and recall for F1. This high performance might be because Random Forest has methods for balancing error among classes in an unbalanced data set. Random Forest has the ability to minimize the overall error rate.

Naïve Bayes, on the other hand shows the least performance for all the feature sets except for the feature F5. This might be

because of more complication on the conditional dependence of the NB due to the increase in number of features.
SVM also performs well as it gets trained independently irrespective of the number of features [3,13]. SVMs have the ability to generalize in spite of having high dimensionality feature space as it uses the hyperplane as decision boundary for classification.

C. Class Wise Performance

In this experimental work, as discussed before, ternary classification is performed. The three classes' labels are "hate", "offensive" and "neither". Among these classes, the features and classifiers performed extremely well for "offensive" class. But for the "hate" all the combinations showed low performance. Mostly, 71% of true hate speech tweets were misclassified as offensive speech. This might be because the misclassified tweets do not contain any terms that strongly exhibit the hate speech such as f*ggot and n*gger. For example, tweets like "monsoon lot of rain, too bad it wasn't enough to wash away the f*ckin white trash in the state" and "You can be Seminoles but not a*shole redskin shits" contains strong hate words such as "white trash" and "redskin shits" that are not explicit. Hence these are more likely to be classified into class "Offensive" because of the words f*ckin and a*shole rather than to class "Hate Speech".
Considering the true neither tweets, that are incorrectly classified as offensive. This is because of the occurrence of potentially offensive words. For example, tweets like "He has given a damn great performance. It's a great improvement to see a gay man, an openly queer actor casted in the highly expected music video." This tweet contains the potentially offensive words "gay" and "queer" but uses them to deliver a optimistic sense.

VIII. CONCLUSION

In this paper, the process of hate speech detection is carried out using the text classification methodology involving the preprocessing techniques, feature extraction techniques and machine learning algorithms. The performance of four different classifiers employed with five different combinations of four feature engineering techniques is performed. The experimental results exhibited that the TFIDF features showed better performance as compared to Sentiment Scores and Doc2Vec. Moreover, Random Forest algorithm showed the best performance. SVM and Logistic Regression also performed better. The lowest performance was observed in Naïve Bayes.
It can be seen from the results that while classifying the offensive class tweets, the models achieved relatively high accuracy. The neither tweets are also correctly classified to a considerable extent. But while detecting hate speech, the model showed lower performance.
There are certain fixed strong racial and homophobic slurs which are particularly used for labeling a tweet as hate speech such as f*ggot and n*gger. While using these terms in tweets makes the detection of hate speech easier, whereas the tweets that do not contain these kinds of strong instances of hate speech tend to get misclassified. Also, there are certain words which appear in both the hate and offensive categories like b*itch, f*g and n*gga. This is where most hate tweets tend to get misclassified as offensive. In order to avoid this misclassification and acquire more accurate classification results, more hate speech training data that are without the explicit particular racial or homophobic slurs should be used. Also, it is more likely that people label only the racist and homophobic slurs as hateful but consider the sexiest descriptive words as offensive. Terms like b*tch, p*ssy, andh*e also exhibit true hate speech that is sexist and derogatory towards women (Waseem and Hovy 2016). But, people tend to use these sexiest words so commonly in their day to day social life that makes them consider it as barely offensive.
Hate speech is a tough topic to handle and define due to its abstractness. The detection of hate speech depends on we people's subjective understanding of what hate speech is. From the results, it is clear that people give best performance in identifying only some of the extremely appellant hate speech like the anti-black racism, insulting Turkishness and homophobia. So the future objective is to understand the social biases and make the model to correctly spot these biases

REFERENCES

[1] *Abro,S., Shaikh,S., Khand,H,Z.,Ali,Z., Khan,S., Mujtaba,G., Automatic Hate Speech Detection using Machine Learning: A Comparative Study. International Journal of Advanced Computer Science and Applications (IJACSA),2020, Volume 11, Issue 8.*
[2] Burnap, P. and M.L. Williams, Us and them: identifying cyber hate on Twitter across multiple protected characteristics. EPJ Data Science, 2016. 5(1): p. 11.
[3] Cavnar, W.B. and J.M. Trenkle. N-gram-based text categorization. in Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval. 1994. Citeseer.
[4] Davidson, T., Warmsley, D., Macy,M., and Weber, I.. Automated hate speech detection and the problem of offensive language. arXiv preprint arXiv:1703.04009, 2017.
[5] Fortuna, P. and S. Nunes, A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 2018. 51(4): p. 85.
[6] Kwok,I.,andWang,Y. 2013. Locate the hate: Detecting tweets against blacks. In AAAI.
[7] Mujtaba, G., et al., Prediction of cause of death from forensic autopsy reports using text classification techniques: A comparative study. Journal of forensic and legal medicine, 2018. 57: p. 41-50.
[8] Wang, W.,Chen, L., Thirunarayan, K., and Sheth, A. P. 2014. Cursing in english on twitter. In CSCW, 415–425.
[9] Warner, W., and Hirschberg, J. 2012. Detecting hate speech on the world wide web. In LSM, 19–26.
[10] Waseem, Z. and D. Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. in Proceedings of the NAACL student research workshop. 2016.
[11] Waseem,Z.,.2016. Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In Proceedings of the First Work shop on NLP and Computational Social Science. Association for Computational Linguistics, Austin, Texas, 138–142.
[12] Z. Zhang and L. Luo, "Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter," vol. 1, no. 0, pp. 1–5, 2018.
[13] Zhang, M.-L. and Z.-H. Zhou, A k-nearest neighbor based algorithm for multi-label classification. GrC, 2005. 5: p. 718-721.