Prediction of offensive language and hate speech on social media

Purva Kolhe

Abstract

The social media platforms are flooded with hate speech. Cyberbullying, cyberhate are proven to have prolonged psychological and physical effects on individuals. Flagging and removing such offensive social media posts that can psychologically harm individuals or incite violence has gained a lot of significance in recent years. The data-set used here is from kaggle website which is classified by hate speech, offensive language and neither. It roughly contains 26k tweets. In this paper, I have pre-processed the data using natural language processing techniques such as NLTK and used this data for running the classical model. For this project, SVM model was used as baseline to perform sentiment analysis on tweets. After extracting sentiment polarity score, classical model has been implemented using TFIDF features and polarity features that were obtained from sentiment analysis to improve the accuracy. Logistic regression and random forest outperformed all the other models using TFIDF feature set and the sentiment analysis features with an accuracy around 90%.

Keywords— offensive language, Natural language processing, NLTK, Sentiment Analysis, SVM, Logistic regression, Naive Bayes, Random forest.

1 Introduction

The use of social media in the current days has risen exponentially, everyone has a smartphone and social media platforms accounts which results in generation of Tera/Peta bytes of data every minute. This information can be about someone's cat or an opinion about a political party/religion/culture etc., In certain occasions, these opinions contain derogatory and offensive content towards individuals or groups.These offensive posts are leading to a spectrum of problems ranging from mental depression in youngsters to political unrest/protests in many countries.

There have been studies and automated models that have been developed to predict the offensive posts using the magic of Machine Learning. These predictions are used to flag the account and also remove these offensive posts from the platform before it does any harm. The models and research that has gone into this problem have been aimed at solving the binary classification problem i.e., to predict whether a post is offensive or not. The current approach is explained in detail in this paper : Preprocess tweets using NLTK library , regex command and perform classical models on preprocessed tweets. The dataset has been taken from the kaggle website which is classified by hate-speech, offensive language, and neither. using the baseline model like SVM, I obtained 89% accuracy for the processed tweets. I also performed the sentiment analysis on tweets and run models using TFIDF with features from sentiment analysis. Additionally, subsequent sections show the analysis performed with the help of different models. The results are presented in the last section. The results are presented in "Result and Discussion" section followed by conclusion and future work .

2 Related Work Survey

A wide range of approaches were studied that aim to identify offensive language on social media. There are multiple models used to train on a dataset to identify such offensive tweets. However, there is enough work done to develop computational models. Early approaches inclined towards feature engineering which combined with traditional machine learning classifiers such as SVM(support vector machine) [2], Random forest classifier, Naive Bayes [4], On the other hand, recently, neural networks and CNN such as LSTMs, GRU and combination of both have proved to outperform traditional machine learning classifiers to predict offensive tweets [1]. In many research papers, transformer models like BERT have been applied to detect offensive language content on social media [3]. These models performed very well in terms of accuracy and give highest accuracy compared to traditional machine learning approaches.

In the current approach, I am performing some classical models on tweets to identify the offensive post on social media. Along with the classical models, I performed the sentiment analysis on tweets and ran models using TFIDF with features from sentiment analysis to improve the accuracy.

3 Dataset

Table 1: Count of labels in class

Label	Count
Hate Speech	1430
Offensive text	19190
neither	4163
Total tweet count	24782

This dataset[1] has been taken from the kaggle website to used for research hate speech detection. This is publicly available dataset provided by CrowdFlower. The text is classified as: hate-speech, offensive language, and neither. Due to the nature of the study, this dataset contains text that can be considered racist, sexist, homophobic, or generally offensive. The dataset contains the different column containing count, hate-speech, offensive language, neither, class and tweets. The classification of lables and count of labels such as hate speech,offensive text and neither is shown in Table 1.

- Count- Number of CrowdFlower users who coded each tweet(Minimum 3).
- Hate Speech- Number of CrowdFlower users who judged the tweet to be hate speech
- Offensive Language- Number of CF users who judged the tweet to be offensive
- Neither- Number of CF users who judged the tweet to be neither offensive nor non-offensive
- Class Labels- class labels assigned as 0 for hate speech, 1 for offensive language and 2 for neither tweet.

4 Method/Implementation

4.1 Data cleaning and pre-processing

The flow “Fig. 1” of the work starts with the analysis of dataset followed by text pre-processing to achieve a cleaner dataset that can be used for feature engineering. The data collected from the social media contains a lot of noise as well as URLs, mentions and emojis. There was a need to clean the data and keep only the most relevant information in the final data set. A lot of the data contained URLs, special symbols, emojis and ’@USERNAME’ tags. The ’@USERNAME’ tags, with ’USERNAME’ here being a stand-in for an actual username, had to be removed due to privacy concerns of Twitter users.

- All numerical characters were removed and replaced with blanks.
- All the user-names and mentions were replaced with a blanks.
- All the punctuation,special characters were removed and replaced with blanks. The characters that were removed from the data-set were -’[]#$%&̂;’.
- Removed STOP-WORDS from all the tweets. I used nltk.stop-word list to remove unwanted words from the tweet. This list is extended to include other words used in twitter such as retweet(rt) etc in stopwords.
- Removed white space with a single space in tweet.
- Tokenization and stemming has been done on the preprocessed tweets.
- Removed all emojis and emoticons.

The processed text is passed further for the feature engineering extraction where features like n-gram tf-idf weights, sentiment polarity scores, and other scores are extracted and concatenated in different sets to fit into different classifications models.

[1]https://www.kaggle.com/mrmorj/hate-speech-and-offensive-language-dataset

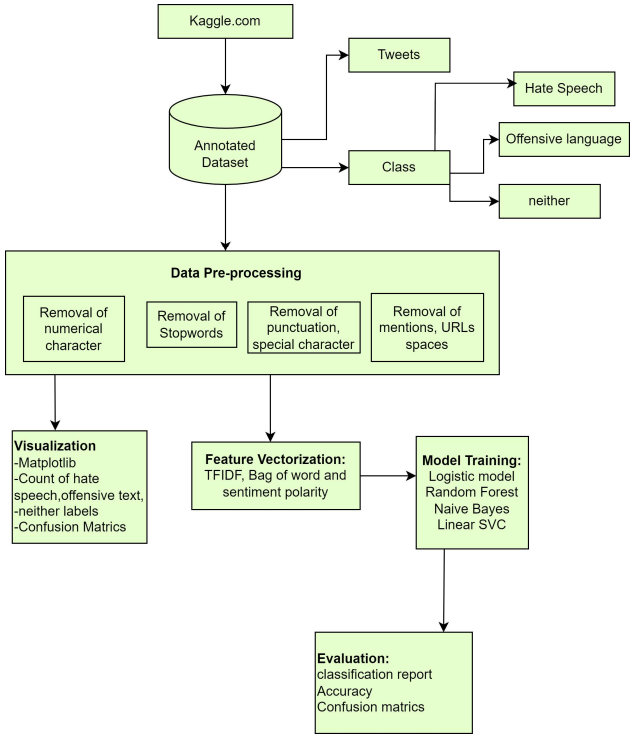

Figure 1: Architecture Diagram

4.2 Experiments

Different techniques like TF-IDF vectorisation, Bag-of-words were used to run various experiments on the data. I extract some unique and important feature and combine them in different sets for the purpose of comparison and analysis of the performance of various machine learning classification algorithms with regard to each feature set. I used several classical Machine Learning Models like Logistic model, Random Forests,Naive Bayes, and LinearSVC. After running model using TFIDF and Bag of word vectorization, I ran the model with the polarity score as a enhanced features from the sentiment analysis using the SentimentIntensityAnalyzer from nltk.vader. After concatenation of TFIDF, Bag of words and sentiment analysis features in a pipeline, I ran several classical models to improve the accuracy score of the previous models. These classification models are evaluated on the basis of accuracy and f1-scores in regards to different feature sets.

5 Results and Discussion

	Hate Speech(0)			Offen sive speech(1)			Neither(2)			
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Accuracy**
Logistic	0.60	0.18	0.28	0.92	0.96	0.94	0.85	0.85	0.85	**0.90**
RandomForest	0.51	0.13	0.21	0.91	0.97	0.94	0.85	0.84	0.84	0.89
NaiveBayes	0.10	0.39	0.16	0.89	0.68	0.77	0.54	0.59	0.56	0.65
SVM	0.45	0.26	0.33	0.92	0.95	0.94	0.83	0.85	0.84	0.89

Each model was trained in the same way using the same techniques such as word tokenization, Bag of words, TFIDF, n-gram mentioned in the Experiment. The TFIDF was performed on the tweet column and pre-processed tweets were added as a new column with additional features from sentiment analysis polarity score. These features were used to get the vectors from the class labels and feature extraction was completed successfully on 24k instances of tweets. I used train_test_split from the sklearn model selection library and split it into test and train data and labels from respective levels of classification. As we can see from the above table of classification report-

- The logistic regression algorithm works consistently well with all feature sets as precision, recall, and f1-score.
- Random Forest classifier works considerably well when it comes to F1 and also shows a significant performance

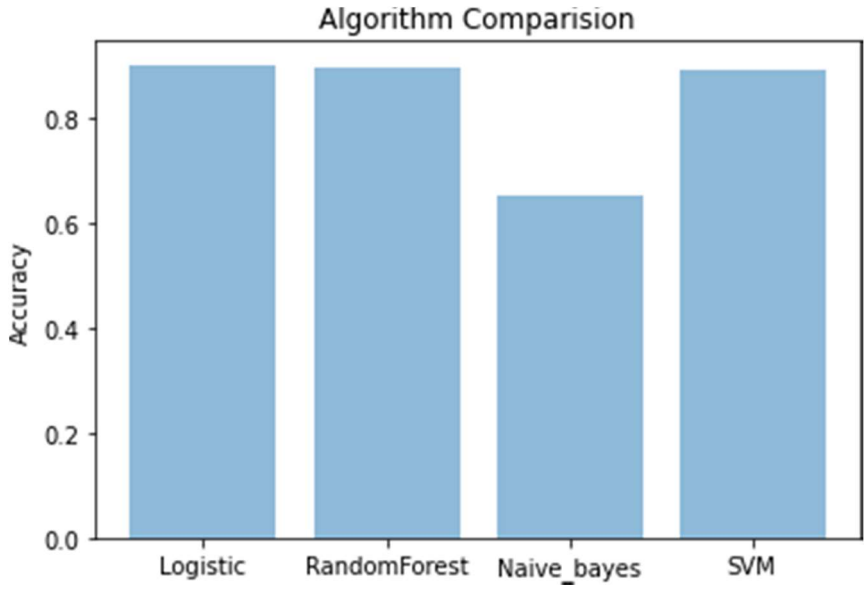

Figure 2: Comparision between the models

in all other feature sets but its performance is hugely impacted when tf-idf scores are not included in the feature set

- The overall performance of the Na¨ıve Bayes classifier is found to be less significant for the purpose of classifying tweets into hate, offensive or neither labels but its performance better in label of offensive including all feature sets.
- SVM classifier also seems to be consistent throughout all feature sets except for feature set from sentiment analysis.
- From the above results it can be inferred that the most important feature is tf-idf scores which helps in better classification of hate speech.
- After analyzing the graph presented in "Fig. 2", its concluded that Logistic regression and random forest algorithm performed better than other.

6 Conclusion

To conclude, this project was challenging from the start as the very first step of collecting/identifying hate speech was tricky. As social dynamics is constantly shifting, what's considered a hate speech also changes drastically over a short period. After pre-processing of the dataset, the text was passed further for feature extraction where features like n-gram tf-idf weights, sentiment polarity scores were extracted and concatenated in different sets to fit into different classification models. This approach also demonstrated that the features set that used TFIDF and sentiment polarity score gave promising results on all levels as opposed to feature set that uses TFIDF or sentiment polarity individually. Out of the experiments performed on different models, Random Forest classifier outperformed the rest.

The results clearly show that differentiating hate speech and offensive language is a challenging task. It also indicates the benefits of using the proposed features and provides a valuable resource for detecting toxic language on twitter.

7 Future Work

As per the approach taken in this project, classical models performed satisfactorily on all levels of classes using TFIDF and sentiment polarity feature set. To further enhance the results, deep leaning models such as LSTM and RNN can be explored as possible solutions. Transformers models like BERT should also be tested to inspect if it can provide

better results. Additionally, usage of NLP libraries such as indicNLP[2] can provide more accurate text preprocessing in INDO-ARYAN languages which will improve overall accuracy of the results. Further work on this project involves the predictions from English to the other language using cross-lingual contextual embed-dings and Transfer Learning (TL).

Acknowledgment

I would like to express my gratitude to profession Cecilia for guiding and advising me through this research. I would also like to thank teaching assistants who was instrumental in solving technical issues faced during this research. Also, a special appreciation should go to the authors of all the research papers listed in reference section that shaped my understanding of this area.

References

[1] Md. Abul Bashar and Richi Nayak. Qutnocturnal@hasoc'19: CNN for hate speech and offensive content identification in hindi language. *CoRR*, abs/2008.12448, 2020.

[2] Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[3] Tharindu Ranasinghe and Marcos Zampieri. Multilingual offensive language identification with cross-lingual embeddings, 2020.

[4] Amir Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. Offensive language detection using multi-level classification. pages 16–27, 05 2010.

[2] https://anoopkunchukuttan.github.io/indic_nlp_library/