ZKG INTERNATIONAL

ISSN: 2366-1313

# Detecting Hate Speech Data and Aggressive Language on Twitter using Machine Learning

SOMU SATISH KUMAR [1], Dr. SIKHAKOLLI GOPI KRISHNA [2]

[1]Assistant professor, [2] Professor

CSE Department, Sri *Mittapalli College* of *Engineering*, *Guntur*, *Andhra Pradesh*-522233

***Abstract — Toxic on-line content has become a serious issue in today's world because of Associate in Nursing exponential increase within the use of net by folks of various cultures and academic background. Differentiating hate speech and offensive language could be a key challenge in automatic detection of virulent text content. During this paper, we tend to propose Associate in Nursing approach to mechanically classify tweets on Twitter into 3 classes: hateful, offensive and clean. Victimization Twitter dataset, we tend to perform experiments considering n-grams as options and spending their term frequency-inverse document frequency (TFIDF) values to multiple machine learning models. We tend to perform comparative analysis of the models considering many values of n in n-grams and TFIDF normalization strategies. When standing the model giving the most effective results, we tend to accomplish ninety-five.6% accuracy upon evaluating it on take a look at knowledge. we tend to conjointly produce a module that is Associate in Nursing intermediate between user and Twitter.***

***Keywords— hate speech, offensive language, n-gram, tf-idf, machine learning, twitter***

## I. INTRODUCTION

In the past ten years, we've seen Associate in Nursing exponential growth within the variety of individuals victimization on-line forums and social networks. Each sixty seconds, there are 510,000 comments generated on Facebook and around 350,000 tweets generated on Twitter. The folks interacting on these forums or social networks come back from totally different cultures and academic backgrounds. At times, distinction in opinions result in verbal assaults. Moreover, ungoverned freedom of speech over the internet and the mask of obscurity that the web provides in cites folks to use racists slurs or uncomplimentary terms. This will lower the self- esteem of individuals, resulting in mental state and a negative impact on the society as an entire. moreover, virulent language will take varied forms, like cyber bullying, that was one in every of the main reasons behind suicide. This issue has shown to be progressively vital within the last decade and detective work or removing such content manually from the online could be a tedious task. Thus there's a desire of production an automatic model that is in a position to observe such virulent content on the online.

In order to tackle this issue, first off we tend to should be ready to outline virulent language. We tend to generally divide virulent language into 2 categories: hate speech and offensive language. Similar approach was used. According to Wikipedia, hate speech is outlined as "any speech that attacks an individual or cluster on the idea of attributes like race, religion, ethnic origin, national origin, gender, disability, sexual orientation, or identity." we tend to outline offensive language because the text that uses abusive slurs or uncomplimentary terms.

## II. LITERATURE STUDY

According to some research papers, we tend to propose Associate in Nursing approach to plot a machine learning model which may differentiate between these 2 aspects of virulent language. We elect to observe hate speech and offensive text on Twitter platform. By victimization in public offered Twitter datasets we tend to train our classifier model victimization n-gram and term frequency inverse document frequency

ISSN: 2366-1313

(TFIDF) as options and appraise it for metric scores. We tend to perform comparative analysis of the results obtained victimization supplying Regression, Naive Bayes and Support Vector Machines as classifier models. Our results show that supplying Regression performs higher among the 3 models for n-gram and TFIDF options when standardization the hyper parameters. We tend to conjointly build use of Twitter Application Programming Interface (API) to fetch public user tweets from Twitter for detective work tweets containing hate speech or offensive language. In addition, we tend to produce a module that is Associate in Nursing intermediate between the user and Twitter.

## III. RELATED WORK

Various machine learning approaches have been made in order to tackle the problem of toxic language. Majority of the approaches deal with feature extraction from the text. Lexical features such as dictionaries and bag-of-words were used in some studies. It was observed that these features fail to understand the context of the sentences. N-gram based approaches were also used which shows comparatively better results.

Although lexical features perform well in detecting offensive entities, without considering the syntactical structure of the whole sentence, they fail to distinguish sentences' offensiveness which contain same words but in different orders. In the same study, the natural language process parser, proposed by Stanford Natural Language Processing Group, was used to capture the grammatical dependencies within a sentence.

Linguistic features such as parts-of-speech has also been used in hate speech detection problem, these approaches consist in detecting the category of the word, for instance, personal pronoun (PRP), Verb non-3rd person.

There have been several studies on sentiment-based methods to detect abusive language published in the last few years. In some examples which applies sentiment analysis to detect bullying in tweets and use Latent Dirichlet Allocation (LDA) topic models to identify relevant topics in these texts. Also studies have been conducted for Detection of harassment on Web 2.0

More recently, distributed word representations, also referred to as word embeddings, have been proposed for a similar purposes. Deep learning techniques are recently being used in text classification and sentiment analysis using paragraph2vec approach. Convolutional Neural Network (CNN) based classification, which refers to the generation of a CNN for text classification, is being used as seen in where they experimented with a system for Twitter hate-speech text classification based on a deep-learning, CNN model.

## IV. PROPOSED APPROACH

Based on the review of features and the prominent classifiers used for text classification in the past work, we need to extract n- grams from the text and weight them according to their TFIDF values. We feed these features to a machine learning algorithm to perform classification. The aim of this work is to classify them into three categories: hateful, offensive and clean.

### *A.* Data

We have generated the data set that could be a combination of three different datasets. We can get the first dataset on Crowd flower. It contains tweets that have been manually classified into one of the following classes: "Hateful", "Offensive" and "Clean". We can get second dataset on the same class. We can get the third dataset on Github. We have used this third dataset widely in this project. In the third dataset there are two columns. They are tweet-ID and class. In this dataset, the tweets can be classified into one of the following three classes: "Sexism", "Racism" and "Neither".

### *B.* Data Preprocessing

ISSN: 2366-1313

We combine the three datasets used for this work in the data preprocessing stage. The main task is to removal of unnecessary columns from the datasets and also to enumerate the classes. We retrieve the tweets corresponding to the tweet-ID present in the dataset for the third dataset. For this purpose we have to use TWITTER API. According to definition the basic two classes such as "Sexism" and "Racism" in this dataset are considered as hate speech.

The tweets should be converted to lowercase and remove the following unnecessary contents from the tweets:

- Space Pattern
- URLs
- Twitter Mentions
- Retweet Symbols
- Stopwords

To reduce the inflectional forms of the words we have used the Porter Stemmer algorithm.

We have to shuffle randomly and the dataset has been split into two parts: train dataset containing 70% of the samples and test dataset containing 30% of the samples.

### C. Feature Extraction

We extract the n-gram features from the tweets and weight them according to their TFIDF values. The goal of using TFIDF is to reduce the effect of less informative tokens that appear very frequently in the data corpus. Experiments are performed on values of n ranging from one to three. Thus, we consider unigram, bigram and trigram features. The formula that is used to compute the TFIDF of term t present in document d is:

$$tfidf(d, t) = tf(t) * idf(d, t)$$

where n in the total number of documents. Similarly, L2 normalization is defined as:

### D. Model

We consider three prominent machine learning algorithms used for text classification: Logistic Regression, Naive Bayes and Support Vector Machines. We train each model on training dataset by performing grid search for all the combinations of feature parameters and perform 10-fold cross-validation. The performance of each algorithm is analyzed based on the average score of the cross-validation for each combination of feature parameters. The performance of these three algorithms is compared. Further, the hyper parameters of two algorithms giving best results are tuned for their respective feature parameters, which gives the best result. Again, 10-fold cross validation is per- formed to measure the results for each combination of hyper- parameters for that model. The model giving the highest cross- validation accuracy is evaluated against the test data. We have used scikit-learn in Python for the purpose of implementation.

TABLE I
COMPARISON OF THREE MODELS FOR DIFFERENT COMBINATIONS OF FEATURE PARAMETERS

| N-gram Range + TFIDF Norm | Accuracy | | |
|---|---|---|---|
| | NB | LR | SVM |
| (1,1) + L1 | 0.842 | 0.816 | 0.802 |
| (1,2) + L1 | 0.878 | 0.801 | 0.823 |
| (1,3) + L1 | 0.890 | 0.794 | 0.841 |
| (1,1) + L2 | 0.862 | 0.878 | 0.862 |
| (1,2) + L2 | 0.913 | 0.901 | 0.884 |
| (1,3) + L2 | **0.926** | **0.918** | **0.901** |

RESULTS AFTER TUNING LOGISTIC REGRESSION W.R.T REGULARIZATION PARAMETER C AND VARIOUS OPTIMIZATION ALGORITHMS (SOLVERS) FOR THE FEATURES: N-GRAM RANGE 1-3 AND TFIDF NORMALIZATION L2

ISSN: 2366-1313

| Regularization C + Solver | Accuracy |
|---|---|
| 10 + liblinear | 0.949 |
| 10 + newton-cg | 0.948 |
| 10 + saga | 0.948 |
| 100 + liblinear | 0.951 |
| 100 + newton-cg | 0.950 |
| 100 + saga | 0.950 |

TABLE III
RESULTS AFTER TUNING NAIVE BAYES W.R.T SMOOTHING PRIOR α FOR THE FEATURES: N-GRAM RANGE 1-3 AND TFIDF NORMALIZATION L2

| Alpha (α) | Accuracy |
|---|---|
| 0.01 | 0.931 |
| 0.1 | 0.934 |
| 1 | 0.925 |
| 10 | 0.877 |

## V. RESULTS

The results of the comparative analysis of Logistic Regression (LR), Naive Bayes (NB) and Support Vector Machines (SVM) for various combinations of feature parameters is shown in Fig. 1 and TABLE I.

Fig. 1 shows that all the three algorithms perform significantly better for the L2 normalization of TFIDF. However, SVM performs poorly as compared to Naive Bayes and Logistic Regression for L2 normalization. TABLE I shows that the best result for Naive Bayes, 92.6%, is obtained using n- gram range up to three and TFIDF normalization L2. Similarly, Logistic Regression performs better for the same set of feature parameters achieving 91.3% accuracy. Since both of these values are comparable, we tune both Naive Bayes and Logistic Regression, for the n-gram range up to three and TFIDF normalization L2.

TABLE II shows the results after tuning the Naive Bayes algorithm. We have considered the smoothing prior α for
tuning. α≥ 0 considers the features which are not present
smoothing and α < 1 is in the training set and in turn prevents zero probabilities called Lidstone smoothing. Naive Bayes performs better for
the α value 0.1 giving 93.4% accuracy.

TABLE III shows the performance after tuning the Logistic Regression algorithm. Here, we have considered the regularization parameter C and the optimization algorithms (solvers)
model with settings C = 100 and solver liblinear gives the – liblinear, newton-cg and saga – for performance tuning. The best accuracy 95.1%.

Comparing the best accuracy for Naive Bayes and Logistic Regression, we conclude that Logistic Regression performs better. Therefore, we evaluate Logistic Regression on test data with the settings: n-gram range 1-3, TFIDF normalization
L2, C = 100 and optimization algorithm liblinear. The classification scores are shown in TABLE IV.

TABLE IV
CLASSIFICATION SCORES OBTAINED AFTER EVALUATING THE FINAL LOGISTIC REGRESSION MODEL ON TEST DATA.

| | Precision | Recall | F-score |
|---|---|---|---|
| Hateful | 0.94 | 0.96 | 0.95 |
| Offensive | 0.96 | 0.93 | 0.94 |
| Clean | 0.96 | 0.98 | 0.97 |

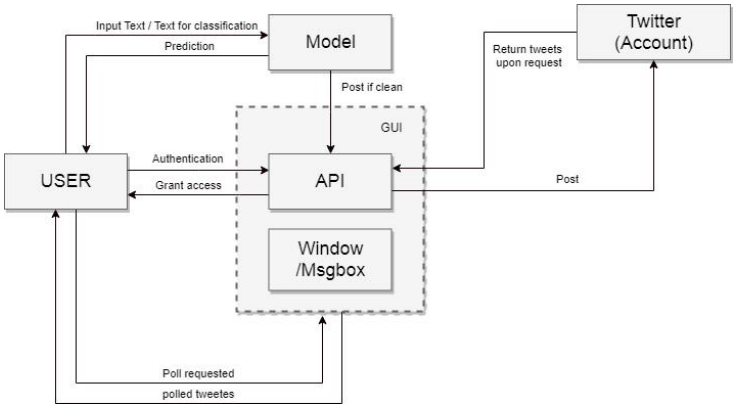
It is observed that the recall for offensive text is relatively low, 0.93. This means that 7% of the tweets that are actually offensive have been misclassified by the model. Also, the precision for the hateful class is 0.94, which signifies that 6% of the tweets that are either clean or offensive have been classified as hateful. On the other hand, the recall for clean class is 0.98, which is significantly better.

In addition to the classification scores, we also computed the confusion matrix for the test results which is shown in TABLE

The key point to notice here is that 4.8% of the tweets that are offensive have been classified as hateful. Improvements can be done in this area to further increase the scores of the model. The final testing accuracy of the model is obtained to be 95.6%.

TABLE V
CONFUSION MATRIX FOR THE EVALUATED TEST DATA ON THE FINAL LOGISTIC REGRESSION MODEL

| Class | Classified as | | |
|---|---|---|---|
| | Hateful | Offensive | Clean |
| Hateful | 0.965 | 0.021 | 0.014 |
| Offensive | 0.048 | 0.926 | 0.026 |
| Clean | 0.010 | 0.013 | 0.977 |




Architecture of the system interfacing with Twitter through Twitter API

We also create an application which acts as a module between the user and Twitter. The architecture of the application. Through our module, we are able to filter out hateful and offensive tweets being posted by an individual as well as classify the tweets posted on the user home timeline, with the only limitation being twitter read request rate limiter of 15 minutes.

## VI. CONCLUSION

In this paper, we proposed a solution to the detection of hate speech and offensive language on Twitter through machine learning using n-gram features weighted with TFIDF values. We performed comparative analysis of Logistic Regression, Naive Bayes and Support Vector Machines on various sets of feature values and model hyperparameters. The results showed that Logistic Regression performs better with the optimal n- gram range 1 to 3 for the L2 normalization of TFIDF. Upon evaluating the model on test data, we achieved 95.6% accuracy. It was seen that 4.8% of the offensive tweets were misclassified as hateful. This problem can be solved by obtaining more examples of offensive language which does not contain hateful words. The results can be further improved by increasing the recall for the offensive class and precision for the hateful class. Also, it was seen that the model does not account for negative words present in a sentence. Improvements can be done in this area by incorporating linguistic features.

# REFERENCES


**[1]** Zephoria.com, 2018. [Online]. Available: https://zephoria.com/top-15- valuable-facebook-statistics/. [Accessed: 22- Jun- 2018].

**[2]** "Twitter Usage Statistics - Internet Live Stats", Internetlivestats.com, 2018. [Online]. Available: http://www.internetlivestats.com/twitter- statistics/. [Accessed: 22- Jun- 2018].

**[3]** S. Hinduja and J. Patchin, "Bullying, Cyberbullying, and Suicide", Archives of Suicide Research, vol. 14, no. 3, pp. 206-221, 2010.

**[4]** H. Watanabe, M. Bouazizi and T. Ohtsuki, "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection", IEEE Access, vol. 6, pp. 13825-13835, 2018.

**[5]** T. Davidson, D. Warmsley, M. Macy and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language", in International AAAI Conference on Web and Social Media, 2017.

**[6]** S. Liu and T. Forss, "New classification models for detecting Hate and Violence web content," 2015.