HATE TEXT RECOGNITION IN SOCIAL MEDIA

Y. Sri Lalitha Information Technology, GRIET, Hyderabad
V N Rama Devi Basic Sciences, GRIET, Hyderabad

Abstract-The fast development of social media content on net has driven certain very unpleasant tendencies, such as the development of abusive and filthy language on the Internet. The Effective Hate speech Detection in Twitter is presented in this Work. The goal of this work is to see if the Twitter streams are spreading hate dialog. Hate speech is described as statement that characterizes a person or a group based on a trait llike colour, racism, origin, gender, faith, nationality, or gender identity. As a first step in preventing hate dialog from spreading among internet users, the purpose of this research is to identify potential hate speech on Twitter. The work explores SVM, MultinominalNB, Logistic Regression, Random Forest and noticed that the average precision, recall and Accuracy on PAN CLEF 2021 are 64%, 74% and 65.7 respectively. It is observed that our experimentation has achieved high accuracy with these models.

Keywords : Hate dialog, Machine Learning, Text classification

I INTRODUCTION

The rising prevalence of hatred text on social media, as well as the imperative necessity for effective remedies, has attracted major investment from governments, businesses, and researchers in recent years. On the internet, a great variety of approaches for detecting automated hate speech have been created. This seeks to categorise textual information as non-hatred or hatred text, in which case the approach may also recognize the hate speech's targeting features (such as racism and faith). However, we see a substantial variation in the two's performance (Non-Hatred Vs Hatred Text). For practical reasons, we suggest in this paper that the latter problem should be prioritized. It is evident that identifying the hate dialog in text is a challenging task as it lacks discriminating factors for non-hate dialog from hate dialog especially in long text content. One negative word can change the whole essence of the sentence and categorize to different class.

II RELATED WORK

In recent years Hatred speech identification in text has attracted many researchers and has witnessed an increase in study. As a result, terminology like 'offensive, profane, and abusive languages,' as well as 'cyberbullying,' frequently coexist or get intermingled with the term 'hate speech'. To separate them, we describe Hate text 1) As a statement that bulls individual or a group of people based on personalities and behaviors; 2) shows a strong purpose to damage or spread hatred; and 3) may or may not include harsh or profane language. [2,4,6,14]
'Assimilate?' for example.

i. “No, they must all return to their own nations”. #BanMuslims Please accept my apologies if someone strongly disagrees.'
ii. 'All you spoils (excluding me) who trolled today should quit this communication platform' says another.

Existing approaches treat the problem largely as a supervised document categorization problem. It includes steps like feature Extraction, selection of important features, train the data using Machine learning models such as Decision Trees, Logistic Regression Or Support Vector Machine etc, or Deep Learning Methods using Artificial Neural networks of Multiple Hidden

Layers to learn the features from the raw documents automatically and perfor document categorizations. Traditional approaches need manual design[11]. It is clear that certain works target a connected topic rather than hatred content in text. Feature vectors are created by encoding instance properties into feature vectors, which are subsequently used by classifiers.

Certain other works learnt from unlabeled corpora via clustering, topic modelling, and word embeddings. The word representation are utilized to create message feature vector.

Specific negative terms (such as slurs, insults, and so on) in communications are frequently looked up using lexical resources [6]. Information such as Part of Speech (PoS) and specific dependence connections are used as linguistic characteristics[12]. Meta-information refers to information regarding communications, such as user's gender identification or a high occurrence of disrespectful phrases in a user's posts in past. In addition, multimodal information such as picture captions and pixel characteristics and knowledge-based features such as communications linked to stereotyped notions in a knowledge base were employed in cyberbullying detection, although only in a very limited context.[13,15]

The best results were obtained using KNN and SVM in [13] uni-class classifier for identifying online hate speech on different Twitter datasets. It incurred an accuracy of 70% and similar results were received by others too who had used slightly different techniques.

ISSUES IN IDENTIFYING HATRED AND INVASIVE TEXT

The problem of automatically recognizing dislike or derogative communication, in social websites has multiple dimensions. Some of them include Spelling Mistakes, Language usage, slang, the mode of casual communication that happens between a group of people, sarcasm that prevails in a social community etc are diverse and is difficult to extract the real intension of such communication on twitter or social media.

Secondly, certain letters in text may be mystified for example a smile is now a days replaced by a symbol‿ .‿., E is written as 3, I is written as 1, characters are written with similar looking digits etc. preprocessing them is a tedious task.

Another challenge is to identify the hatred dialogue from the text content by using key-word based features selection and text classification. Most of the efficient works in the field of Text Classification are based on VSM model of Data representation and training the model for Classification. Some of the works addressed the semantic similarities of words and classification[16].

Furthermore, many terms are not intrinsically objectionable but might be when used in the wrong context. However, not only can different slurs have varying degrees of offence, but the offence can also vary depending on time (Initially harmless phrases can become derogative phrases after some time) as well as homonyms (‘different meaning of the same word’), diverse perspective of user on a social platform, all of these contribute to various challenges in understanding whether the speech is hateful or normal. One suggestion for reducing bias is to actively prepare annotators for it.

Another challenge in Hate speech recognition works include availability of consistently labelled data as there is no universally acknowledged description of Hatred Content. Let alone one that is productive (a remark on which numerous publications concur)[8].

Even if we manually prepare one such dataset, we cannot incorporate all the solutions to the issues stated above in the dataset. Distinguishing the different forms of communications among a certain selected groups of people with diverse perspectives is not trivial considering the attributes of local conditions, Contextual Circumstances of individual, Individual or group expressions that contribute to the factors of hateful speech are difficult to determine.

III OUR APPROACH

The objective of this work is in Categorizing tweets into two categories: "hateful”," and “not hateful". The approach is depicted in Figure 1. Data collection, Pre-processing, Identify Features, Prepare Train and Test sets, Develop Model (Classifier) and Evaluate the Model are the six major steps in this work. The next sections go through each stage in great depth.[4]

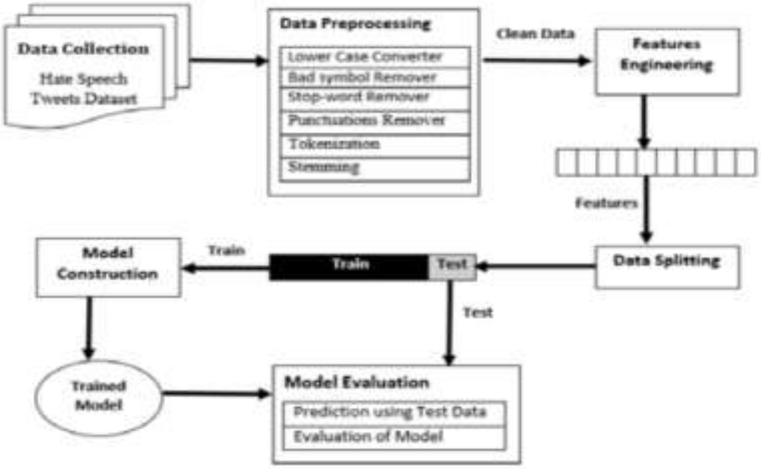


Figure 1 : System Architecture

A DATA COLLECTION

The data was obtained through PAN CLEF 2021, where a 300 record data was provided for training and a 100 record data was given for testing. The tweets for each user were provided in a different .txt file which were then converted into a table format afterward.

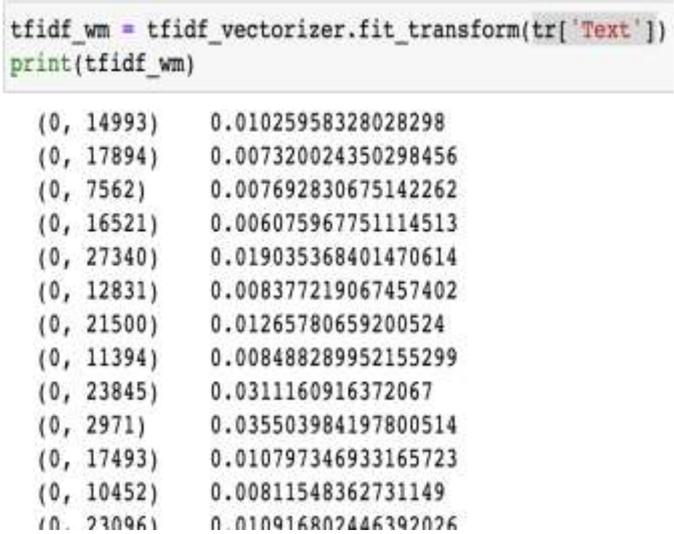
B PRE-PROCESSING

Pre-processing is a crucial step in Machine Learning Process. Better results are achieved with pre-processed data. Different text pre-processing techniques are studied to sift noisy and non-informative text from the dataset. Several Text cleaning methods are applied to extract crisp, error-free data from large collection of Tweeter data. The challenging part in social media, the communication is casual, unofficial and often we see gibberish content such as ‘wefwfwe’ or ‘qjndkqx’ used by people in order to express their mood or opinion about some topic or a person, we might not know the exact meaning of those words therefore we can remove such words to attain maximum efficiency without any addition of unknown data into the model [1,2,3,4]. Our work applied techniques like changing all the text content into lowercase, removing hyperlinks, urls, special characters such as “#,@,$,<,>, spaces, single and double quotes” to obtain text that is meaningful to express hatred or non-hatred content in text. Later applied Tokenization to form words from the collection of tweets, performed stop word removal to discard words that will not help in discriminating the text, finally applied Stemming algorithm to determine root words and thus formed the collection of features.

C IDENTIFYING FEATURE

The categorization rules cannot be deduced from the raw text by machine learning techniques. To determine the classification rules, the features must be expressed using numeric values. This stage involves extracting essential features from raw text and numerically expressing the extracted features. We used three different feature engineering strategies in this study: n-gram with tfidf, word2vec, and doc2vec.

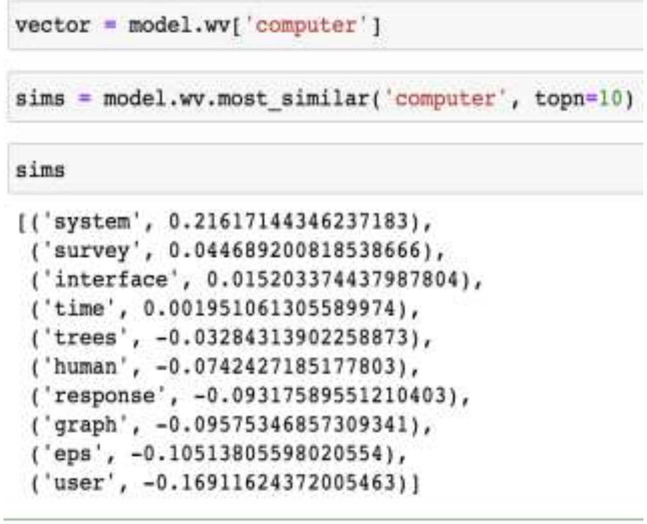
TFIDF: TF-IDF indicates "Term Frequency-Inverse Document Frequency" a method to calculate the significance of a word in a collection of text or documents. If a sentence such as the one given below is fed into the tfidf vectorizor, it will calculate the frequency of the word in a sentence and characterize the importance of each word. Now each word in the sentences contained in the list is converted to a vector and assigned values based on its importance and frequency of occurrence.



```
tfidf_wm = tfidf_vectorizer.fit_transform(tr['Text'])
print(tfidf_wm)
```

```
  (0, 14993)    0.01025958328028298
  (0, 17894)    0.007320024350298456
  (0, 7562)     0.007692830675142262
  (0, 16521)    0.006075967751114513
  (0, 27340)    0.019035368401470614
  (0, 12831)    0.008377219067457402
  (0, 21500)    0.01265780659200524
  (0, 11394)    0.008488289952155299
  (0, 23845)    0.0311160916372067
  (0, 2971)     0.035503984197800514
  (0, 17493)    0.010797346933165723
  (0, 10452)    0.00811548362731149
  (0, 23096)    0.010916802446392026
```

Word2Vec: Word2vec generates vectors, which are distributed numerical representations of word properties like context of individual words. It accomplishes this without the need for human interaction. Word2vec can produce very accurate assumptions about a word's meaning based on previous appearances if given enough data, use, and circumstances. In the figure below all the data inside sample is vectorized and an assumption of its each word is made by the Wrod2vec model. These assumptions can then be used by the model to understand the meaning of test data provided later. The example below shows how similar the word is to other words in its understanding.



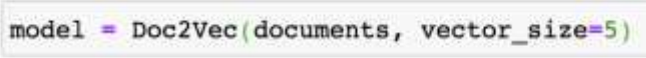
```
vector = model.wv['computer']
```

```
sims = model.wv.most_similar('computer', topn=10)
```

```
sims
```

```
[('system', 0.21617144346237183),
 ('survey', 0.04468920081853866),
 ('interface', 0.015203374437987804),
 ('time', 0.001951061305589974),
 ('trees', -0.03284313902258873),
 ('human', -0.0742427185177803),
 ('response', -0.09317589551210403),
 ('graph', -0.09575346857309341),
 ('eps', -0.10513805598020554),
 ('user', -0.16911624372005463)]
```

Doc2vec: Doc2vec is very similar to Word2vec, Word2Vec calculates a feature vector for every word in the dataset, Doc2Vec calculates a feature vector for every document in the dataset.



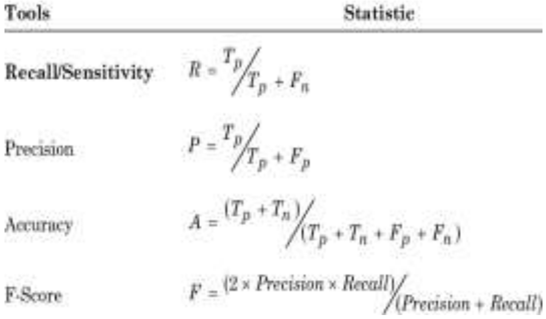
```
model = Doc2Vec(documents, vector_size=5)
```

D Cross-validation

All the machine learning algorithms were trained using 5-fold cross-validation. The “PAN CLEF 2021” training data set is split into five development sets with no overlap, preserving the training set's class distribution as precisely as feasible. The training set in each example was made up of the remaining training data (data not included in the development set). Then we trained five distinct models for each machine learning approach (that we used here), each using a training set for parameter optimization and development set for validation. Calculating the anticipated probabilities for all the models, averaging these probabilities, and then categorizing each event using the label with the highest expected probability yielded the final choice for each technique. Then incorporated the new training data to all training folds when using additional corpora.

E Classifier Evaluation

The Developed models predicts the class of unknown text based on the test set (i.e. "hate speech, not hate speech"). The classifier's performance is measured using true negatives (TN), false positives (FP), false negatives (FN), and true positives (TP). The performance of the selected classifier is evaluated using a range of performance measures. Here are a few common text categorization performance measures that are briefly discussed.



Tools	Statistic
Recall/Sensitivity	$R = {T_p}/{T_p + F_n}$
Precision	$P = {T_p}/{T_p + F_p}$
Accuracy	$A = {(T_p + T_n)}/{(T_p + T_n + F_p + F_n)}$
F-Score	$F = {(2 \times Precision \times Recall)}/{(Precision + Recall)}$

1. *Figure 2: Precision, Recall, F-score Measures*

IV Results

The work modelled different classifiers their results are depicted in the Table 1

Table 1 Comparision of Models

MODELS	Accuracy	Recall	Precision	ROC	AUC
SVM	69.40	80.00	64.50	67.50	0.69
Logistic Regression	65.50	80.00	62.02	65.50	0.67
MultinomialNB	66.40	74.00	65.49	67.50	0.69
RandomForest	66.00	61.00	67.78	66.00	0.66
ModelSGD	66.00	79.00	62.70	66.00	0.67

The best results were obtained in a support vector machine (SVM) algorithm with a linear kernel. The accuracy obtained was 69.4, Recall score of 80%, a Precision score of 64.5%, and an AUC of 68.7%. The Area under curve observed the highest in case of the SVM method. AUC indicates the degree or measure of separability, whereas ROC is a probability curve. The AUC metric indicates how well a model can differentiate across classes that is how well the model predicts 0 class as 0 and 1 class as 1.

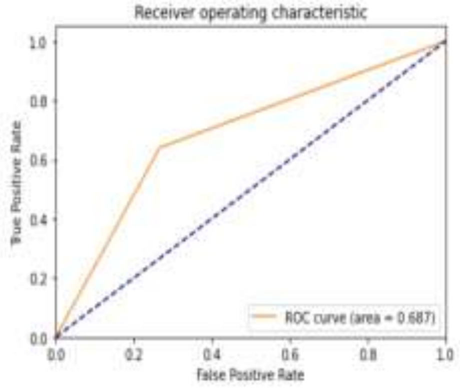


Figure 3- SVM ROC curve

The other models gave out the following results:

The figure 4 is the result obtained from using logistic regression on the dataset. The AUC of the curve is 0.669 which is clearly below the result from SVM model, the accuracy is also comparatively lower at 65.5%. Logistic Regression model requires the dependent variable to be binary, multinomial or ordinal in nature. It necessitates that the observations be unrelated to one another. As a result, the findings should not be based on repeated measurements.

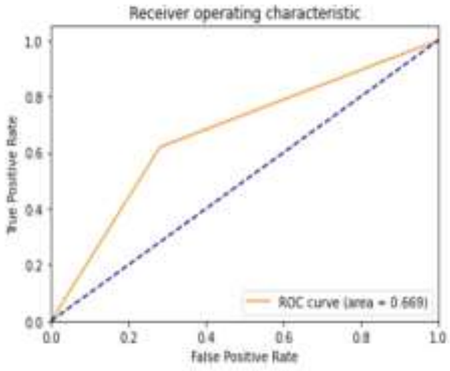


Figure 4: Logistic Regression

The following figure 5 is the result of Multinomial NB model with AUC score of 0.678 and an accuracy of 66.4%. The multinomial Naive Bayes classification model is best for distinct feature classification. Integer feature are best suitable for classification, real valued features such as tf-idf, can also be used.

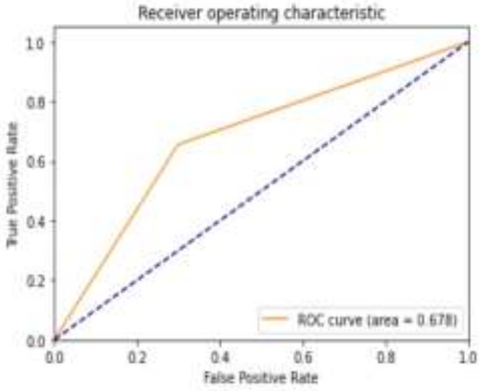


Figure 5 : Multinominal NB ROC Curve

The result of Random Forest model is shown in fig6 where the AUC score is 0.660 and the accuracy obtained is 66%. The random forest splits the training data into subsets and applies decision tree classifier, takes the average of the predicted accuracies of the multiple classifiers and determines the accuracy of the entire dataset. It often presents better result in comparison to application of a tree algorithm on whole dataset.

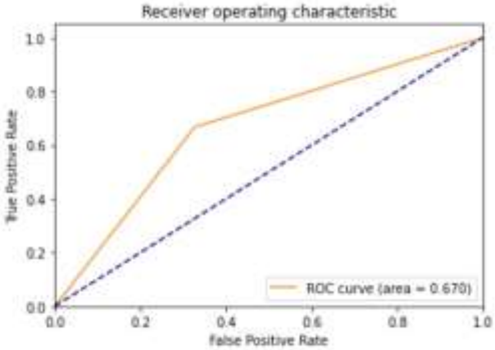


Figure 6 : Random Forest

V CONCLUSION

We summarized the models for the Profiling Hate Speech Spreaders on Twitter task at PAN CLEF 2021. Typically, this work presented supervised learning Models on Twitter data. Bags of words, Word 2 vector embeddings and Doc2Vector representation of Data are rather general features that consistently produce decent classification results. Later modelled Machine Learning Algorithms such as SVM, Multi-nominal NB, Logistic Regression and Random Forest to study the effect of classification on twitter data. It is observed that atmost 67% of accuracy, with AUC score of 0.678 is achieved using SVM and MultinomialNB models and around 62-65% accuracy with other models and around 0.65 AUC Score. There is scope to improve the accuracy by considering various NLP methods for preprocessing, identifying hateful content not only at token level but considering word n-gram features, character n-gram features for experimentation. The following conclusions are drawn from the findings. Character level hate speech content determination will be more effective than approaches at the token level. Lexical resources, such as a list of slurs, can aid categorization, but only when used in conjunction with other characteristics. Various complicated features that need additional linguistic expertise, such as dependency parse information, or features that mimic specific verbal structures, such as imperatives or politeness, have also been demonstrated to be useful. Text indication alone may not be enough to specify the presence of hatred content. It might be accompanied with meta-data or data from other modalities. Many of the complex features are difficult to judge in terms of their overall effectiveness because they are usually only evaluated on individual data sets, the majority of which are not publicly available and often only address a subtype of hate speech, such as bullying of specific ethnic minorities. We propose for a benchmark data set for hate speech identification to improve the comparison of different characteristics and algorithms.

REFERENCES

[1] Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, Narayan Bhamidipati, “Hate speech detection using comment embeddings”, in 24th ICWWW, 2015.

[2] Sanjan Sharma, Saksham Aggarwal and Manish Srivastava “Degree based Classification of Harmful Speech using Twitter Data”, in “First Workshop on Trolling, Aggression and Cyberbullying”, 2018.

[3] Thomas Davidson, Dana Warmsley, Michael Macy,Ingmar Weber. “Automated Hate Speech Detection and the Problem of Offensive Language” , in ICWSM 2017

[4] Shervin Malmasi Marcos Zampieri , "Detecting Hate Speech in Social Media" in "Recent Advances in Natural Language Processing",2017

[5] Hugo Lewi Hammer, "Automatic detection of hateful comments in online discussion", "International Conference on Industrial Networks and Intelligent Systems", 2017

[6] Kelvin Kiema Kiilu, George Okeyo, Richard Rimiru, Kennedy Ogada, " Using Naïve Bayes Algorithm in detection of Hate Tweets", in international journal of Scientific and Research Publications 2018

[7] 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.

[8] Ramos, J "Using tf-idf to determine word relevance in document queries". in Proceedings of the first instructional conference on machine learning. 2003. Piscataway, NJ.

[9] Mikolov, T., et al. "Distributed representations of words and phrases and their compositionality" in Advances in neural information processing systems. 2013.

[10] Le, Q. and T. Mikolov. Distributed representations of sentences and documents. in International conference on machine learning. 2014.

[11] Kotsiantis, S.B., I.D. Zaharakis, and P.E. Pintelas, Machine learning: a review of classification and combining techniques. Artificial Intelligence Review, 2006. 26(3): p. 159-190.

[12] Y. Sri Lalitha, J Sirisha Devi, L. Sukanya, N.V. Ganapathi Raju, "Analysis of Parts of Speech Tagging in Text Clustering", International Journal of Innovative Technology and Exploring Engineering, June 2019, Volume 8, Issue : 8, pp : 2287-2291.

[13] Swati Agarwal and Ashish Sureka, "Naive (Bayes) at forty: The independence assumption in information retrieval", in European conference on machine learning. 2018. Springer.

[14] Xu, B., et al., An Improved Random Forest Classifier for Text Categorization. JCP, 2012. 7(12): p. 2913-2920.

[15] Joachims, T. "Text categorization with support vector machines: Learning with many relevant features," in European conference on machine learning. 1998. Springer.

[16] Y. Sri Lalitha, A. Govardhan "Semantic Framework for Text Clustering with Neighbors" in Intelligent Systems and Computing 249, © Springer International Publishing Switzerland December 2013 pp.261-271.