Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

IDENTIFICATION OF HATE SPEECH USING NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING

B.Dilip Chakravarthy1, M.Sravana Sandhya2, C.Varalakshmi3, N.Tharun4, Y.Tharun5, Y.Suchitha6

1 Assistant Professor in Department of Computer Science and Engineering, Annamacharya Institute of Technology and Sciences(Autonomous), Rajampet, Andhra Pradesh, India.

2,3,4,5,6Department of Computer Science and Engineering, Annamacharya Institute of Technology and Sciences(Autonomous), Rajampet, Andhra Pradesh, India.

Abstract:

Twitter's main objective is to make it possible for everyone to create and exchange information, as well as to freely express their beliefs and presuppositions. The purpose of Twitter is to facilitate public discourse, which necessitates the representation of a variety of viewpoints. Nonetheless, it does not encourage violence against, openly attack, or undermine people based on their ethnicity, country, public cause, rank, sexual orientation, age, incapacity, or real sickness. Hate speech may be harmful to an individual or a community. Thus, using hate speech is not suitable. Hate speech is now widely employed on social media platforms as a result of a rise in their use. Hence, it is impossible to recognise hate remarks by hand. Hence, it is essential to create a model for automatically detecting hate speech, and this study illustrates many methods of using Natural Language Processing to classify hate speech using machine learning algorithms.

Keywords: Hate Speech, Random Forest, Tf-Idf, Logistic Regression, and Bag of Words.

I.INTRODUCTION

People are adopting social media platforms to share their opinions due to social media's growing popularity. It might be challenging to express harsh or disrespectful thoughts to someone face-to-face. Many thus believe it is acceptable to abuse others or publish objectionable content online. As a result, people feel comfortable sharing such material online. As a result, hate speech on social media is becoming more prevalent every day. In order to manage such a big number of users on social media, technologies for automated identification of hate speech are needed. In this study, we classify whether or not a statement constitutes hate speech using machine learning techniques[13]. There are many uses for machine learning, and text-based categorization is one of them. The same set of characteristics utilised by machine learning algorithms is employed to represent each instance, or in this case, each tweet. Machine learning algorithms may address two different sorts of problems: supervised and unsupervised. The task of training a model using a given dataset that contains both a collection of features and labels is known as supervised learning. Even though the training system function for unsupervised learning uses data sets that are neither labelled nor categorised[3]. Based on the labels in the dataset, supervised learning is further separated into two types: regression and classification. Here, our sole focus is

categorization. To categorise the class or category of the unknown instance, classification machine techniques employed categorical datasets. Tasks that may be configured as supervised are included in many machine learning applications. With a labelled dataset of hate speech, we want to complete this job using supervised classification techniques including support vector machines, logistic regression, and random forests. Each instance is represented as a vector, whose length depends on the technique. as an illustration of tweets. In this study, we employed two different vector representation methods for tweets: bag of words and term frequency-inverse document frequency (tf-idf).

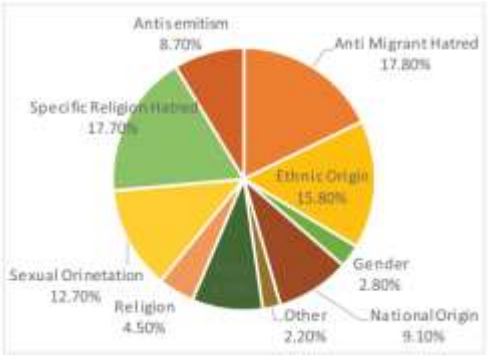

Fig : 1 Types of hate content removed from platforms that have signed the EU Code of Conduct (January 2018)

A.Hate Speech

Hate Speech used to only be allowed in person-to-person interactions. Yet as social media channels proliferate, hate speech is increasingly more prevalent. Many feel that they are concealed online. People feel comfortable using hate speech as a result, and because it is difficult for humans to detect hate speech on social media, we need automated tools to do so. On the other hand, people are more willing to express their opinions online, which encourages the spread of hate speech. Policymakers and social networking sites may benefit from monitoring and preventative measures given that this kind of biassed communication may be especially harmful to society[6]. Hate speech is often defined as any communication that disparages a person or group because of attributes including race, ethnicity, gender, sexual orientation, nationality, or religion[8].

B. Definition of hate speech

Paula Fortuna and Sergia Nunes define hate speech as "language that attacks or diminishes, that incites violence or hate against groups, based on specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual orientation, gender identity, or other," and they add that it can take many different linguistic forms, even subtly or when humour is used. [6].

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

II RELATED WORK

There are many approaches for detection of hate speech. But they differ from each other based on the output they obtained in Ref. 8 hate speech was classified into three classes race, nationality and religion. Ref. 8 uses sentiment analysis technique for detection of hate speech but just not detecting but they also classified into one of the three classes and also rate the polarity of speech. We found two survey papers for automatic hate speech detection [6],[14]. In Ref. 6 motivation for hate speech detection is shown and why it became necessary to develop more robust and accurate models for automatic hate speech detection. The problem of hate speech detection is more often researcher keep data private while collecting it and there are less open source code available which make it difficult for comparative study [6] . This degrades the progress in this field. Different features related to hate speech are described in Ref. 14, like simple surface feature which includes bag of words, unigrams or n-grams. Both training set and testing set need to have same predictive word but it is problem as detection of hate speech is applied on very small piece of text so to overcome this issue word generalization is applied [14].

Knowledge of annotator for hate speech was examined in Ref. 15. Authors produce some very good results in amateur annotation in comparison to expert annotations. Also, Waseem provide its own dataset and its evaluation. To penalize misclassification on minority classes weighted F1-score is suggested as an evaluation measure. Nowadays with development in deep learning, CNN can be used for hate speech detection [2],[1]. Word-vector also known as word embedding can be trained on relevant corpus of the domain. This pretrained word-vectors are used in CNN [2] . Most of machine learning models uses bag-of words which fails to capture patterns and sequences. It can be understood by the example in Ref. 2. “if a tweet ends saying if you know what I mean” here each word can be considered as hate speech but it is most likely that this sentence is hate speech. This type of features cannot be handled by bag of words which degrades the performance of traditional machine learning algorithm.

III. DATASET DESCRIPTION

Dataset was obtained from the online social media platform twitter. It can be easily found on the GitHub where previous researchers have uploaded different datasets for hate speech detection.

Table 1. Classwise distribution of dataset:

Class	No. Of Instances
Class 0	29720
Class 1	2242

[a] Class 0 – non hate speech

[b] Class 1 – hate speech

20 % of data is used as testing and 80% of as training from each class

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

Table 2. Classwise distribution of Training Dataset

Class	No. of Instances
Class 0	23775
Class 1	1794

TABLE 3. CLASSWISE DISTIBUTION OF TESTING DATASET

Class	No. of Instances
Class 0	5945
Class 1	448

Table 4. Top 11 words with Highest Frequency in Hate speech

Amp	Hate
Trump	Women
White	Might Libtard
Allahsoil	Libtard Libtard
Black	Libtard Sjw
Racist	

Table 5. Term frequency

	am	and	breeze	broken	canopy	forest	here	home	leaves	my	please	pollen	the	to	trees
I am the trees the forest my home	1	0	0	0	0	1	0	1	0	1	0	0	2	0	1
I am the pollen and the broken leaves	1	1	0	1	0	0	0	0	1	0	0	1	2	0	0
I am canopy	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
I am the breeze	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0
I am here to please	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0

Table 6. Inverse Document Frequency

	am	and	breeze	broken	canopy	forest	here	home	leaves	my	please	pollen	the	to	trees
For Entire Corpus	1	2.098612	2.098612	2.098612	2.098612	2.098612	2.098612	2.098612	2.098612	2.098612	2.098612	2.098612	1.405465	2.098612	2.098612

Table 7. Multiply Tf and Idf

	am	and	breeze	broken	canopy	forest	here	home	leaves	my	please	pollen	the	to	trees
I am the trees the forest my home	1	0	0	0	0	2.09861	0	2.09861	0	2.09861	0	0	2.81093	0	2.09861
I am the pollen and the broken leaves	1	2.09861	0	2.09861	0	0	0	0	2.0986	0	0	2.09861	2.81093	0	0
I am canopy	1	0	0	0	2.09861	0	0	0	0	0	0	0	0	0	0
I am the breeze	1	0	2.09861	0	0	0	0	0	0	0	0	0	1.40547	0	0
I am here to please	1	0	0	0	0	0	2.09861	0	0	0	2.09861	0	0	2.09861	0

Table 8. Normalize Tf and idf

	am	and	breeze	broken	canopy	forest	here	home	leaves	my	please	pollen	the	to	trees
I am the trees the forest my home	0.1942	0	0	0	0	0.40753	0	0.40753	0	0.40753	0	0	0.54586	0	0.40753
I am the pollen and the broken leaves	0.1942	0.4075	0	0.4075	0	0	0	0	0.40753	0	0	0.40753	0.54586	0	0
I am canopy	0.4302	0	0	0	0.90275	0	0	0	0	0	0	0	0	0	0
I am the breeze	0.3681	0	0.7725	0	0	0	0	0	0	0	0	0	0.51738	0	0
I am here to please	0.2653	0	0	0	0	0	0.55667	0	0	0	0.55667	0	0	0.55667	0

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

IV. FEATURE EXTRACTION

Feature Extraction is a method to convert each tweet into a fixed set of attributes so that it can be easily interpreted by machine learning models. In feature extraction, a vocabulary of words is being generated and vocabulary depends upon the method we use for feature extraction. This vocabulary is used to convert each tweet into a vector form [9]. It is an important stage in Text Based classification which offers significant details based on text such as the term duration for each tweet. Feature extraction is one of the key preprocessing techniques used in data mining and text classification that measures the context of documents [16].

A.Tf-idf

TF-IDF is widely recognized and is often used as a weighting strategy and its efficiency is equivalent to modern techniques. Documents are known to be variables in the word weighting. Selecting a function for a function selection procedure is considered to be the key preprocessing process necessary for indexing documents [11].

TF: Term Frequency, which measures how much a word appears in a text. Although each document is specific in duration, it is likely that the word will occur far more often in longer documents than in shorter ones.:

$$\text{T f(t)} = \frac{\text{Count of term t in a document}}{\text{T otal count of terms in the document}} \qquad (1)$$

IDF: Inverse Document Frequency, which measures the importance of a term while calculating TF. Usually, all terms as considered to be equal in terms of importance. However, when certain terms like" is"," of", and" the", may occur a greater number of times but have little importance. Thus, it is necessary to give weight to each and every term.

$$\text{IDF(t)} = \frac{\log_e \text{count of documents}}{\text{Count of documents containing term t}} \qquad (2)$$

If we break td-idf into steps then we get following three steps:

Step 1 : Derive term frequency

Step 2: Derive inverse document frequency

Step 3: Aggregate the above two values using multiplication and normalization

Corpus used as a example for tf-idf

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

```
corpus = ["I am the trees the forest my home",
          "I am the pollen and the broken leaves",
          "I am canopy","I am the breeze",
          "I am here to please"
          ]
```

Step 1: Calculate the term frequency

Calculating term frequency is a pretty straight forward it is calculated as the number of times the word or a term appear in a document.

	am	the	trees	forest	my	home
I am the trees the forest my home	1	2	1	1	1	0

	am	the	trees	forest	my	home
I am the trees the forest my home	1	1.405465	2.098612	2.098612	2.098612	2.098

Table 5. term frequency matrix was generated using CountVectorizer in python.

Step 2:Inverse Document Frequency

If we only use the term frequency as measure than there is no difference between the important word like greatness and the common word like you. If a word is in all document or in most of documents then it play very less role in differentiating between the documents.So we need a mechanism to tone down the importance of the word that appear most frequently. Similar to term frequency,

Inverse Document frequency = total number of documents / number of documents with the term t Table 6.shows the idf values for all the words. Note that there is only one IDF value for a word in the corpus.

Step 3:Multiply and normalize

In Tf-IDF as the name implies it's a combination of tf and idf i.e multiplying the two values. The sklearn implementation then normalize the product of tf and idf.

Step 3a: Multiply Tf and idf

When multiplying two matrices together, we take elment wise multiplication of term frequency matrix and Inverse Document Frequency matrix Example of multiplication for first sentence is calculated as below.

Applying this to the corpus we get the table 7.

Step 3b: Normalize:

Normally, normalize is often used because we can compare it easily as we use percentage or proportions. So table 8 shows the td-idf vector for each document.

B. Bag of words

We need a method to interpret textual data for the machine learning algorithm, and the bag-of-word model allows us to accomplish that target. The bag of words model is easy to understand and apply. In this method we create tokens of each sentence and the calculate frequency of each token [12].

For example.

“i am thankful for having a paneer today”

“i get to see my daddy today!”

First step: Each sentence is assumed to be a separate document and by making a vocabulary from this two documents excluding punctuation we get,

‘I’, ‘am’, ‘thankful, ‘for’, ‘having’, ‘a’, ‘paneer’, ‘today’, ‘get’, ‘to’, ‘see’, ‘my’, ‘daddy’

Second step:

In this step we create vectors, obtained from text which are used by the machine learning algorithms. for example — “i am thankful for having a paneer today” and we check the frequency of words from the already generated vocabulary.

“I” = 1
“am” = 1
“thankful” = 1
“for” =1
“having” = 1
“a” = 1
“paneer” = 1
“today” = 1
“get” = 0
“to” = 0
“see” = 0
“my” = 0
“daddy” = 0

Rest of the documents will be:
“i am thankful for having a paneer today” = [1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0]
“i get to see my daddy today!” = [1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1]

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

V. MACHINE LEARNING MODEL

A. Support Vector Machine (SVM)

Using either nonlinear or linear mapping SVM converts the original lower dimensional data into a higher dimension. Within this new dimension, it looks for the linear optimal dividing hyper-plane to distinguish the tuples between the sets [5]. For sufficient nonlinear scaling to an appropriate high dimension, information from two array scans are always differentiated by a hyperplane. The SVM finds this hyperplane with the help of support vectors. Support vectors are those instances which are nearer to the margins. An unlimited number of separating lines that could be drawn here. The target is to classify the "highest" one with the least classification error on previous unseen tuples [4]. Maintaining the Integrity of the Specifications.

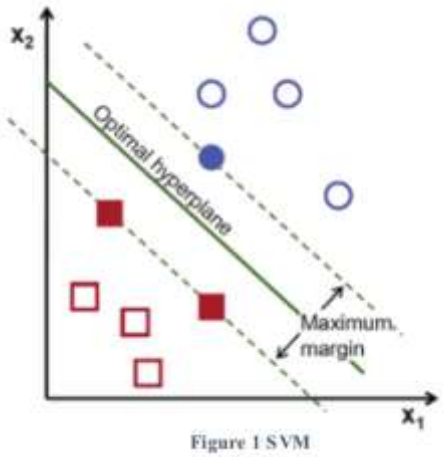

Figure 1 SVM

B. Logistic Regression (LR)

Logistic regression is a supervised machine learning method which is similar to linear regression but instead of using a linear equation it uses a sigmoid function (Eq. (1)) which makes the output value in range [0,1] which is used for text classification also [7].

$$f(n) = \frac{1}{1+ e^{-n}} \qquad (3)$$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

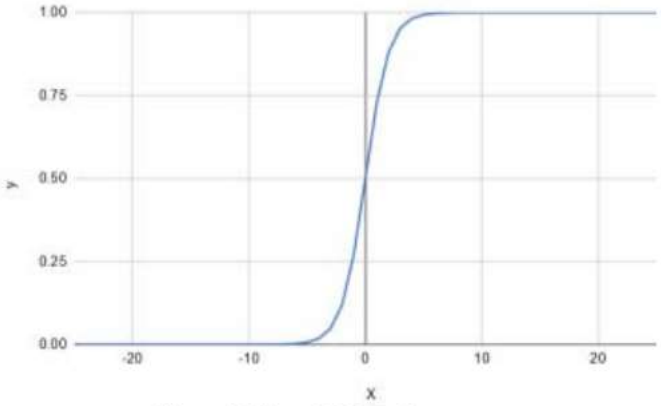

Figure 2 Sigmoid Function

$$X = (X_1, X_2, \cdots, X_n) \quad (4)$$

Logistic Regression Equation

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots\ldots + \beta_n X_n \quad (5)$$

if we substitute "Eq. (5)" in "Eq. (3)" we get,

$$f(n) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots\ldots + \beta_n X_n)}} \quad (6)$$

Where "Eq. (6)" is probability of the datapoint belonging to anyone class. Then cutoff point (mostly 0.5) is used divide between the two classes. Cutoff point is not fixed it can be change according to the dataset.

C. Random Forest (RF)

In Machine Learning, Classification is a major part. There are various techniques for classification such as Logistic regression, decision tree, Artificial Neural Network, Support Vector Machines, etc. A collection of trees is known as forest. Similarly, Random forest is a collection of decision trees and it is called random because it is a collection of relatively uncorrelated trees operating as a single model. The basic principle of random forest is that each and every tree speaks out its prediction and based on the majority decision the random forest predicts its decision. So, in our application of Detecting hate speech the trees classify the statement as hate speech or not and based on the majority decision the random forest predicts whether the statement is a hate speech or not.

VI. PREPROCESSING

Various method for machine learning model are used to achieve higher evaluation measure. Methods Used are explained below.

A.Tokenizing

Tokenizing is process in which each sentence is divided into words. This is used to create vocabulary for our dataset. This vocabulary is used represent each tweet in our dataset representation is based on choice of our method like tf-idf or bag of words.

B. Stop Words

Stop words are those words which has less meaning or are useless for e.g, a,the,of which often occurs in most sentences. So it is required to remove this stop words otherwise it will cause misclassification.

C. Stemming

Stemming is process in which the prefix or suffix of a word is removed to make similar in common form.For e.g processing, process and processed have basically same meaning if we ignore the tense so it is required to convert all this word in similar form. For this well known english stemmer, porter stemmer is used here.

D. Case Folding

In case folding all word are changed in lowercase. This is used to make the vocabulary as small as possible.

VII. RESULTS

First SVM, logistic regression and random forest are used with default parameters with bag of words and tf-idf representation without any preprocessing. Size of feature vector in all representation, with or without preprocessing is shown in below table We can see that using preprocessing size of vectors are decreased which decreases the computation time.

Table 9. Length of vector

Without preprocessing	Tfidf	(1,588381)
	Bag of Words	(1,12514)
With preprocessing	tfidf	(1,402289)
	Bag of Words	(1,10013)

A. Data Without Preprocessing

1) Confusion Matrix

Table 10. SVM with Tfidf

	Not Hate Speech	Hate Speech
Not Hate Speech	5941	4
Hate Speech	300	148

Table 11. SVM with BOW

	Not Hate Speech	Hate Speech
Not Hate Speech	5939	6
Hate Speech	275	173

Table 12. LR with Tfidf

	Not Hate Speech	Hate Speech
Not Hate Speech	5939	6
Hate Speech	377	71

Table 13. LR with BOW

	Not Hate Speech	Hate Speech
Not Hate Speech	5903	42
Hate Speech	210	238

Table 14. RF with Tfidf

	Not Hate Speech	Hate Speech
Not Hate Speech	5942	3
Hate Speech	278	170

Table 15. RF with BOW

	Not Hate Speech	Hate Speech
Not Hate Speech	5928	17
Hate Speech	220	228

2) Accuracy Score & F1-Score

Table 16. accuracy score F1 score

Model	F1-Score	Accuracy-Score
SVM with tfidf	0.4933	0.9524
SVM with BOW	0.5518	0.9560
LR with tfidf	0.2704	0.9401
LR with BOW	0.6538	0.9605
RF with tfidf	0.5475	0.9560
RF with BOW	0.6580	0.9629

B. Data with preprocessing and using Gridsearchcv

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

1) Gridsearchcv results

Table 17. Best parameter from GridSearchcv

Model	Best Parameter
SVM with tf-idf	'C': 2, 'kernel': 'linear'
SVM with BOW	'C': 1, 'kernel': 'linear'
LR with tf-idf	'C': 400, 'maxiter': 100
LR with BOW	'C': 10, 'maxiter': 50
RF with tf-idf	'maxdepth': None, 'nestimators0 : 5
RF with BOW	'maxdepth': None, 'nestimators0 : 1000

confusion Matrices using data with preprocessing and above paramters for machine learning model

2) Confusion Matrix

Table 18. SVM with Tfidf

	Not Hate Speech	Hate Speech
Not Hate Speech	5865	80
Hate Speech	132	316

Table 19. SVM with BOW

	Not Hate Speech	Hate Speech
Not Hate Speech	5868	77
Hate Speech	159	289

Table 20. LR with Tfidf

	Not Hate Speech	Hate Speech
Not Hate Speech	5869	76
Hate Speech	145	303

Table 21. LR with BOW

	Not Hate Speech	Hate Speech
Not Hate Speech	5872	73
Hate Speech	164	248

Table 22. RF with Tfidf

	Not Hate Speech	Hate Speech
Not Hate Speech	5931	32
Hate Speech	269	179

Table 23. RF with BOW

	Not Hate Speech	Hate Speech
Not Hate Speech	5914	31
Hate Speech	199	249

3) Accuracy Score & F1-Score

Table 24. accuracy score F1 score

Model	F1-Score	Accuracy-Score
SVM with tfidf	0.7488	0.9668
SVM with BOW	0.7101	0.9630
LR with tfidf	0.7327	0.9654
LR with BOW	0.7055	0.9629
RF with tfidf	0.5432	0.9529
RF with BOW	0.6840	0.9640

VIII.CONCLUSION

As social media use grows in daily life, everyone tends to believe they have the freedom to say or publish anything they want. This way of thinking has led to a rise in hate speech, making it essential to automate the process of categorising the data. We have employed a machine learning technique to identify hate speech using the data from Twitter in order to streamline the classification of hate speech. To do this, we extracted features from the tweets using the tf-idf and bag of words approaches. We have used machine learning techniques like SVM, Logistic Regression, and Random Forest to categorise hate speech from the tweets. We can infer from the findings that Random Forest with bag of words performs best when utilising Data without preprocessing and machine learning models with default settings, with F1 scores of 0.6580 and Accuracy Scores of 0.9629. Yet as was previously said, when working with unbalance class datasets, just attaining the best accuracy is insufficient. In order to do so, we have employed the F1 score, which is rather low for raw data. To make this better, we utilised gridsearch and a few preprocessing steps to find the ideal parameters for the machine learning model. SVM with Tf-IDF provides the greatest result with a 0.7488 F1 Score and 0.9668 Accuracy Score after preprocessing and utilising gridsearch. As compared to the bag of words model, the tf-idf feature extraction model achieves superior accuracy since the bag of words model just counts the frequency of words and utilises it as a vector, but the tf-idf model employs the ratio of term frequency to document frequency. This method's limitations include the fact that it can only be used with the Twitter dataset, making it difficult to identify hate speech in huge data.

F1 accuracy and score may be increased in the future. Exploration of further machine learning methods is necessary. Moreover, a separate approach must be used to manage the imbalance class dataset.

REFERENCES

[1] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In Proceedings of the 26th International Conference on World Wide Web Companion, pages 759–760, 2017.

[2] Md Abul Bashar and Richi Nayak. Qutnocturnal@ hasoc'19: Cnn for hate speech and offensive content identification in hindi language. In Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation (December 2019), 2019.

[3] Pete Burnap and Matthew L Williams. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. Policy & Internet, 7(2):223–242, 2015.

[4] Fabio Del Vigna12, Andrea Cimino23, Felice Dell'Orletta, Marinella Petrocchi, and Maurizio Tesconi. Hate me, hate me not: Hate speech detection on facebook. In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), pages 86–95, 2017.

[5] Shimaa M Abd El-Salam, Mohamed M Ezz, Somaya Hashem, Wafaa Elakel, Rabab Salama, Hesham ElMakhzangy, and Mahmoud ElHefnawi. Performance of machine learning approaches on prediction of esophageal varices for egyptian chronic hepatitis c patients. Informatics in Medicine Unlocked, 17:100267, 2019.

[6] Paula Fortuna and S´ergio Nunes. A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 51(4):1–30, 2018.

[7] Purnama Sari Br Ginting, Budhi Irawan, and Casi Setianingsih. Hate speech detection on twitter using multinomial logistic regression classification method. In 2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), pages 105–111. IEEE, 2019.

[8] Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering, 10(4):215–230, 2015.

[9] Ammar Ismael Kadhim. Term weighting for feature extraction on twitter: A comparison between bm25 and tf-idf. In 2019 International Conference on Advanced Science and Engineering (ICOASE), pages 124–128. IEEE, 2019.

[10] Harpreet Kaur, Veenu Mangat, and Nidhi Krail. Dictionary-based sentiment analysis of hinglish text and comparison with machine learning algorithms. International Journal of Metadata, Semantics and Ontologies, 12(2- 3):90–102, 2017.

[11] M Ramya and J Alwin Pinakas. Different type of feature selection for text classification. International Journal of Computer Trends and Technology, 10(2):102–107, 2014

[12] Joni Salminen, Maximilian Hopf, Shammur A Chowdhury, Soon-gyo Jung, Hind Almerekhi, and Bernard J Jansen. Developing an online hate classifier for multiple social media platforms. Human-centric Computing and Information Sciences, 10(1):1, 2020.

[13] TYSS Santosh and KVS Aravind. Hate speech detection in hindi-english code-mixed social media text. In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, pages 310–313, 2019.

[14] Anna Schmidt and Michael Wiegand. A survey on hate speech detection using natural language processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, pages 1–10, 2017.

[15] Zeerak Waseem. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In Proceedings of the first workshop on NLP and computational social science, pages 138–142, 2016.

[16] Tingxi Wen and Zhongnan Zhang. Effective and extensible feature extraction method using genetic algorithm-based frequency-domain feature search for epileptic eeg multiclassification. Medicine, 96(19), 2017

[17] Abro, S., Sarang Shaikh, Z. A., Khan, S., Mujtaba, G., & Khand, Z. H. Automatic Hate Speech Detection using Machine Learning: A Comparative Study. Machine Learning, 10, 6,2020.