RESEARCH ARTICLE

SAMRIDDHI Volume 14, Issue 4, 2022 Print ISSN: 2229-7111 Online ISSN: 2454-5767

Evaluating Effectiveness of Feature Extraction Technique in Gujarati Hate Speech Detection

Abhilasha Vadesara[1],Purna Tanna[2]

[1]Gujarat Law Society University, Ahmedabad, Gujarat, India

[2]Faculty, Gujarat Law Society University, Ahmedabad, Gujarat, India

Abstract

Natural language processing has been built so much im-portance in recent years. NLP and machine learning can recognize the hidden feature from a tremendous volume of text data for text classification and sentiment analysis. Twitter has become one of the most popular microblogging services for sharing and receiving ideas and views world-wide. However, users sometimes post the incidence of ag-gression and related incidents like trolling, cyberbullying, flaming, spreading hate etc. For that reason, the detection of hate speech is required for many social media services.

In this paper, we experimented with different feature ex-traction approaches like BoW (Bag of Word) and TF-IDF to extract the feature from Gujarati hate speech. The exper-iment was done on 12K tweets. We implemented the pre-processing technique, such as removing unnecessary sym-bols, URLs, characters, and stop words to improve the clas-sification accuracy in the machine learning model. Fleiss's Kappa technique is used to check inter agreement between 25 annotators, who annotate the whole corpus and have achieved 0.87% accuracy.

Keywords: Sentiment Analysis, Pre-processing, Feature Extraction, Fleiss's Kappa.

SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology (2022); DOI: 10.18090/samriddhi.v14i04.24

Introduction

According to the survey, hate speech is considered a crime,[1,2] which spreads face to face and on social media like Facebook, Twitter, WhatsApp, Snapchat, Instagram, etc. The number of social media users increases every day, and it was estimated in 2019, there will be up to 2.77 billion social media users worldwide.[4] Nowadays, Twitter has become one of the most popular microblogging services on the internet in a few years. Tweets are small text-based messages of up to 140 characters that users can send and read. Abbreviations, hashtags, and emoticons are used on social media to convey the author's message in a few words. It enables fast communication and easy access to sharing and receiving ideas and views from worldwide. But at the same time, this freedom of expression has led to a continuous rise in hate speech on social media.

As people's interaction on social media has increased, the incidence of aggression and related incidents like trolling, cyberbullying, flaming, spreading hate, etc., has also increased worldwide. Much of this hateful language has given unprecedented power and influence to affect the lives of billions of people on the internet. It has been conveyed that these happenings have created mental and psychological suffering for web users. Still, it has forced people to deactivate their accounts and, in rare instances, commit suicide. Thus, hateful and offensive posts must be detected and removed from social platforms as soon as possible because such posts spread very quickly and harm humankind.

Hate speech detection has become a popular research topic in recent years. The language used in social media is often very different from traditional print media. It has various linguistic features. Thus, the challenge in automatically detecting hate speech is very hard. The systems need to be more intelligent and nuanced to be helpful in many cases. Moreover, the system should also be able to recognize incidents of both overt as well as covert aggression. Many Different ML techniques are used to detect hate speech in different languages like Hindi, English, Arabic, German, etc.

The ML technique requires the related feature to accomplish the task from the unstructured data. Text mining is the most common way of finding helpful information from unstructured text. Feature extraction technique eliminates unwanted variations from the Twitter

Corresponding Author: Abhilasha Vadesara, Gujarat Law Society University, Ahmedabad, Gujarat, India, e-mail: abhilashavadesara@gmail.com

How to cite this article: Vadesara, A., Tanna, P. (2022). Evaluating Effectiveness of Feature Extraction Technique in Gujarati Hate Speech Detection. *SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology,* 14(4), 147-152.

Source of support: Nil

Conflict of interest: None

data, escapes the computational expense and increases the classification accuracy. Before the feature extraction technique is implemented, the initial step is to clean the data or implement the pre-processing technique. Then the feature extraction can be done on pre-processed data and finally, the ML algorithm detects the hate speech from the extracted feature.

This paper represents the feature extraction and ML algorithm to identify hate speech from the Gujarati language corpus. The corpus of Gujarati hate speech isn't available at this point, so we gathered around twelve thousand tweets of different categories like politics, celebrity, religion, sports, etc. The techniques like TFIDF and BOW have been widely used to extract the features as well as reduce error rate. SVM (Support vector machine) algorithms help classify the data into two groups.

Review of theoretical and empirical literature

Extracting meaningful information from the text is required for text classification. Many Studies have been led about feature extraction techniques. It is the process of dimensionality reduction which transforms the raw data into the important feature.

Naseem, Usman, *et al.*[4] Twelve alternative pre-processing methods on pre-classified hate speech datasets were explored and their influence on the classification tasks they support was seen. They applied the TFIDF and Glow word-level feature extraction models on three separate datasets with standard and deep learning classifiers.

Raj C, Agarwal A, *et al.*[5] introduced two real-world cyberbullying datasets, built a neural network framework with optimization and tested eleven classification methods: four traditional machine learning and seven shallow neural networks. They evaluated the effect of feature extraction and word-embedding techniques in natural language processing and found that Logistic Regression produced the best results in ML classifiers. The classic ML strategy with TF-IDF and their suggested shallow neural networks obtained 95 and 98% accuracy with F1 scores.

Ahuja, Ravinder, *et al.*[6] investigated the SS-Tweet dataset of sentiment analysis, TF-IDF word level and N-gram were examined. They utilized that TF-IDF word level sentiment analysis performance is 3-4% greater than using N-gram features. They looked at six classification algorithms (support vector machine, K-nearest neighbor, decision tree, random forest, Naive Bayes, and logistic regression) and their performance characteristics (F-score, accuracy, precision, and recall). As a result, logistic regression is the best algorithm for sentiment analysis, and both feature extraction techniques work well.

Suhasini and vimla[7] used ML classification methods to detect bogus news on Twitter data. They employed a combination of TFIDF and N-gram approaches, resulting in the greatest accuracy compared to individual TF-IDF and N-Gram approaches.

Kasri, Mohammed, *et al.*[8] examine the effect of feature extraction approaches such as Bag-of-Words, TF-IDF, and word2vec on sentiment analysis performance in Arabic. Many machine learning techniques, such as Logistic Regression and Support Vector Machine, are used to analyze the retrieved features. When compared to BoW and AraVec, the TF-IDF technique produced the best results with the most classifiers. Logistic regression, on the other hand, achieved good results regardless of the approach utilized.

Rusli, Andre, *et al.*[9] used the Multi-Layer Perceptron (MLP) in order to detect fraudulent news items and separate them from the real ones using a binary text classification approach. They compared the TF-IDF model, bag of words and N-gram model for feature extraction and achieved accuracy with 0.87% f1-score.

Rathi, Megha, *et al.*[10] implemented a hybrid classification approach using svm, adaboost decision tree, and decision tree and achieved higher accuracy than the traditional approach.

Thavareesan, Sajeetha, and Sinnathamby Mahesan[11] evaluated five corpora using different feature representation strategies to discover the best strategy to perform SA in Tamil literature. They compared characteristics such as word count and punctuation count, as well as classic features such as Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TFIDF), and discovered that the UJ Corpus Opinions Nouns corpus using fastText had the highest accuracy of 79% for the supervised Machine learning-based technique.

Romadon, Annalisa Wahyu, *et al.* used feature extraction methods to automate job interview grading to reduce bias and human mistakes. TFIDF, a popular feature extraction method, is compared against word embedding to determine the best method and parameters for categorizing interview verbatim using an ANN classifier. According to the test findings, TFIDF surpasses word embedding by 85.22% against 74.88%, respectively. The findings reveal that the average TF-IDF accuracy is 80.55% and the average word embedding accuracy is 71.22%. According to the average accuracy findings, TF-IDF and Artificial Neural Networks classification are better for text classification than Word Embedding in this study.

The Table 1 details the related datasets of hate speech in different language with used different features and algorithm.

Research Methodology and Data Collection

The proposed work, as illustrated in Figure 1. For the experimentation, the data was gathered from Twitter using API which contains 12000 tweets. To increase the accuracy of the ML model, the data was cleaned by NLP pre-processing techniques and extracted the features were using

TF-IDF and Bag-of-Word techniques.[8] The extracted features are then evaluated using the Support vector machine (SVM) classifier for hate speech detection tasks.

Data and Data Pre-processing

Hate speech datasets are available for many different languages like Hindi, English, Arabic, German etc., but the dataset of Gujarati language is not available yet. Therefore, we gathered the hate speech of Gujarati language from Twitter using Twitter API for the period of Jan 2020 to January 2021. The dataset was collected on different categories like politics, celebrity, religion, and sports. The dataset contains twelve thousand tweets but initially, it was in the form of Unicode, so it was converted into Gujarati language with the help of Python language. Since there is a lot of noise in the dataset, it was necessary to clean it, so we implemented the NLP pre-processing technique as follows.[4]

- Remove all the non-Gujarati tweets
- Remove all the whitespace, null, blank value
- Remove all URLs (e.g. www.abl.com), special characters, emoticons, symbols, or numbers (e.g.! #, $, *, 1234, etc.)
- Remove the duplicate tweets
- Normalize the lengthening words. For example, the word 'અરે રેરેરેરે' will be normalized into 'અરે રે'.
- Remove Stop words
- Tokenize the tweets

Data Annotation

The annotation task for identifying hate speech was completed by twenty-five annotators with a strong Gujarati language background. The selected category of annotators are based on different age groups. 28% of people are in the graduate age range 41 to 49 years. 32% of people are postgraduate age range 29 to 36 years and 40% of people were all college students and language experts with the age range of 19 to 24. The guidelines and instructions were given to annotators to classify the tweet into hate and non-hate categories. The data is hard to trace and biased as it is annotated manually, therefore, to check the inter agreement between annotators, Fleiss's kappa technique is used. We achieved the value of kappa k 0.87 which denotes almost perfect agreement between annotators.[19] After the annotation task, we found 69.3% of tweets as hate speech, while 30.7% were non-hate speech from the whole corpus presented in Table 2.

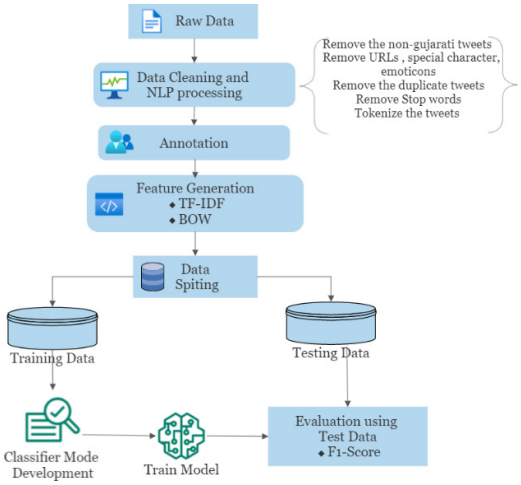

Figure 1: Methodology

Table 1: Related dataset of hate speech

Paper	*Language*	*Dataset*	*Features*	*Algorithm*
[13]	English	Twitter, Facebook, Google	unigram, bigram	Naive Bayes, Max Entropy, Support Vector Machine
[14]	Malayalam, Tamil	YouTube, Twitter	TF-IDF, Word Embedding	Support Vector Classifier (SVC), Multinomial Naive Bayes (MNB), Logistic Regression (LR), AdaBoost, Decision Tree Classifier (DTC) Random Forest Classifier (RFC)
[15]	English	Twitter	N-gram	Naive Bayes, Support Vector Machine, Random Forest
[16]	Arabic	Twitter	unigram, word-ngram, char-ngram, word embedding, Random Embedding, FastText, mBert, AraVec	Naive Bayes, Support Vector Machine, Logistic Regression, Convolution Neural Network, Long Short-Term Memory, Gated Recurrent Unit
[17]	English, German, Spanish, French, Greek	Twitter	N-gram, skip-gram	CNN, LSTM, Skipped CNN, Ensemble
[18]	English	Twitter	Word gram, unigram, Glove, Embedding, trigram	Fuzzy Logic, Support vector machine, ANN, Deep Learning, Hybrid Method

Table 2: Gujarati hate and non-hate dataset

Numeric representation	Class	Total instance
0	Hate	6930
1	Non-hate	3070
	Total	10000

Feature Extraction

TF-IDF

TF-IDF stands for "Term Frequency Inverse Document Frequency". This algorithm can locate the content of a document quickly. It provides the accurate information and quick results required by people and performs massive data classification.[20] The TF-IDF algorithm is used as a weighting factor in search of text mining and information retrieval and user modeling. To evaluate words in a bunch of records this technique is used. It can be defined as the computation of how a word is related to a particular corpus or series. It can be divided into two parts TF (term frequency) and IDF (inverse document frequency).[21]

TF

Term Frequency estimates the number of times a specific term t appears in document d. When the term has appeared various times, the frequency will increase. The TF is evaluated by taking the ratio of the number of time t appears in d, where is the raw count of a term in a document d, to the total number of terms t in the document d.[22]

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (1)$$

IDF

The inverse document frequency is a proportion of how much information the word gives, i.e., if it is rare in the whole corpus. In the dataset, some terms like stopwords appear multiple times but it may not be very important. Therefore the IDF measured the importance of terms that rarely occurred in documents. N stands for total number of documents in the corpus. dD:td defined the number of documents where the term t appears (i.e., tft,d0. If the term is not in the corpus, this will lead to a division by zero. It is, therefore, common to adjust the denominator to 1+dD:td.[23]

$$idf(t,D) = \log \frac{N}{|\{d \in D: t \in d\}|} \quad (2)$$

Bag of word (BoW)

The bag of words model is a summarizing representation used in natural language processing and information retrieval (IR). This model removes grammar and even word order while maintaining multiplicity and represents a text (such as a sentence or document) as the bag (multiset) of its words.

BoW = No. of times word w occurred (3)

This research transformed requirements into numerical vectors so that each document would have its own vector (row). As an illustration, the following document feature vectors were generated using the BoW model for the statements of 'આ તમારા નેતા હદ્દિઓને ડફોળ બનાવવવા નીકળ્યા છો.', and 'સમજદાર તો નેતા જ છે, દુનયિા અને પબ્લકિ તો ડફોળ છે.' The unique terms in the corpus included after pre-processing the data' નેતા હદ્દિઓને ડફોળ બનાવવવા નીકળ્યા સમજદાર દુનયિા પબ્લકિ', therefore the size of vocabulary will be 8 words out of a total of 10 words. The vector representation of this corpus is shown in Table 3. It is clear from the table that each row represents one of the corpus documents and that each column or dimension in the feature vectors represents a word from the corpus. The value in any cell indicates how many times that term appears in the particular documents represented by the row. A corpus of documents will thus contain an N-dimensional vector for each document if there are N unique words present in the corpus.

Machine learning Classifier

Textual features were extracted using BoW and TF-IDF in the above step. These features serve as the input for machine learning techniques that train classifiers. The supervised machine learning algorithm SVM classifier is used in this research. The idea of classification focuses on creating a model that divides data into two classes, "hate" and "non-hate," to identify hate speech. In order to learn the algorithm, this model is created by entering a collection of training data for which the classes are already annotated as "hate" or "non-hate". The ratio of training and testing the data has been kept 80–20.

Support vector machine is a supervised and associated learning technique for pattern recognition and data analysis. The objective of the support vector machine algorithm is to locate the best two-dimensional line or hyperplane in an N-dimensional space that clearly classifies the data points, i.e., where all of the data points on one side of the line represent one category and all of the data points on the other side of the line represent a different category. SVM works well because it chooses a line that separates the data and is as far away from the nearest data points as is practical. Margin, support vectors, and hyper-planes are further terminology related to this.

Classifier Evaluation

In this stage, we assess the classifier's effectiveness in predicting the outcomes of the unlabeled test dataset, i.e., "hate" and "non-hate." Calculating the precision, recall, and F1 measure is used to evaluate the accuracy. The classification

Table 3: Example of bag of word feature extraction technique

નેતા	હદ્દિઓને	ડફોળ	બનાવવવા	નીકળ્યા	સમજદાર	દુનયિા	પબ્લકિ
1	1	1	1	1	0	0	0
1	0	1	0	0	1	1	1

Table 4: Confusion matrix for classification evaluation

hate	*non-hate*		
True Negative (TN)	False Negative (FN)	hate	Predict
False Positive (FP)	True Positive (TP)	non-hate	

performance of the following equations is assessed as shown in Table 4.

FP (False Positive). If the classification statistics are inaccurate, it represents the number of non-hate examples.

TP (True Positive). It displays the number of positive samples and the outcome of a positive classification.

FN (False Negative). In the instance that the classification outcomes are incorrect, it indicates the number of hate cases.

TN (True Negative). If the classification statistics are accurate, it indicates the number of negative cases.

Precision. Precision measures the percentage of accurate predictions given inside examples that the detector has detected as hate speech.

$$Precision = \frac{True\ Positive}{(True\ Positive + False\ Positive)} \quad (4)$$

Recall. The percentage of real cases of hate speech that were automatically labelled as such is known as recall.

$$\text{Recall} = \frac{True\ Positive}{(True\ Positive + False\ Negetive)} \quad (5)$$

F1-score. The weighted average of recall and precision.

$$\text{F1} = \frac{2 * Precision * Recall}{(Precision + Recall)} \quad (6)$$

Accuracy. The percent of labels that were accurately predicted to all labels.

$$\text{Accuracy} = \frac{(True\ Positive + True\ Negative)}{(True\ Positive + True\ Negative + False\ Positive + False\ Negative)} \quad (7)$$

Empirical Findings

We used the Twitter API to obtain the Gujarati dataset for our study, which was used to identify hate speech. We considered two different classifications for the dataset which are hate and non-hate. The entire dataset includes four distinct categories: politics, celebrity, religion, and sports. Twelve thousand datasets have been collected, however, because they were unprocessed and noisy, NLP pre-processing has been used. We used a dataset that 25 different individuals annotated. At the end of the annotation process, we had 6930 hate and 3070 non-hate datasets and measured the degree of agreement between the annotators using Fleiss's kappa and came up with a score of 0.87%, which is an almost perfect agreement based on the kappa's table.

The accuracy discovered from testing the classifiers using the feature extraction techniques is shown in Table 5. The result shows that the BOW method performs well as compared to TF-IDF with SVM classifiers. In common classification tasks, evaluation metrics like accuracy, precision, recall, and F-score are used. In our example, accuracy and precision are used to determine how many comments are correctly classified into different classes and how many accurately identified comments among those labeled as expressing hate. A low recall indicates that many relevant opinions are left unrecognized. F-score is the arithmetic mean of precision and recall. Recall indicates how many comments have been correctly predicted in the entire collection.

Table 5: Accuracy performance of SVM classifier

Feature extraction method/ML Classifier	*TF-IDF*	*BOW*
SVM Classifier	0.64	0.79

Conclusion

In this paper, we have discussed and implemented the different feature extraction methods like TF-IDF and BoW on the collected dataset, which was pre-processed by different techniques. We implemented a support vector machine classification method to identify hate and non-hate speech with TF-IDF and BoW methods and achieved 0.64 and 0.79 accuracies, respectively. The result shows that the BoW method performs well as compared to TF-IDF with SVM classifiers on our dataset. Further, we will implement different word embedding techniques and improve the accuracy of the classifier.

References

[1] Fbi. 2015. 2015 hate crime statistics. Retrieved from https://ucr.fbi.gov/hate-crime

[2] Ciftci, tuba & gashi, liridona & hoffmann, rené & bahr, david & ilhan, aylin & fietkiewicz, kaja. (2017). Hate speech on facebook.

[3] Vadesara, tanna and joshi (2021). Hate speech detection: a bird's-eye view, data science and intelligent applications, vol 52. Springer, singapore.

[4] Naseem, U., Razzak, M. I., & Eklund, P. W. (2021). A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. Multimedia Tools and Applications, 80(28–29), 35239–35266. https://doi.org/10.1007/s11042-020-10082-6.

[5] Raj, Chahat, et al. "Cyberbullying Detection: Hybrid Models Based on Machine Learning and Natural Language Processing Techniques." Electronics, vol. 10 https://doi.org/10.3390/electronics10222810

[6] Ahuja, R., Chug, A., Kohli, S., Gupta, S., & Ahuja, P. (2019). The Impact of Features Ex-traction on the Sentiment Analysis. Procedia Computer Science, 152, 341–348. https://doi.org/10.1016/j.procs.2019.05.008

[7] V, Suhasini, and Dr N.vimala, (2021, September 7). A Hybrid TF-IDF and N-Grams Based Feature Extraction Approach for Accurate Detection of Fake News on Twitter Data, vol. 12, https://turcomat.org/index.php/turkbilmat/article/view/10885,

[8] Kasri, M., Birjali, M., & Beni-Hssane, A. (2019). A comparison of features extraction methods for Arabic sentiment analysis. In International Conference Big Data and Internet Things. https://doi.org/10.1145/3372938.3372998

[9] Rusli, A., Young, J. C., & Iswari, N. M. S. (2020). Identifying Fake News in Indonesian via Supervised Binary Text Classification.

In 2020 IEEE International Conference on In-dustry 4.0, Artificial Intelligence, and Communications Technology (IAICT). https://doi.org/10.1109/iaict50021.2020.9172020.

[10] Rathi, M., Malik, A., Varshney, D., Sharma, R., & Mendiratta, S. (2018). Sentiment Anal-ysis of Tweets Using Machine Learning Approach. In International Conference on Con-temporary Computing. https://doi.org/10.1109/ic3.2018.8530517.

[11] Thavareesan, S., & Mahesan, S. (2019). Sentiment Analysis in Tamil Texts: A Study on Machine Learning Techniques and Feature Representation. In 2019 14th Conference on Industrial and Information Systems (ICIIS). https://doi.org/10.1109/iciis47346.2019.9063341

[12] Romadon, A. S., Lhaksmana, K. M., Kurniawan, I., & Richasdy, D. (2020). Analyzing TF-IDF and Word Embedding for Implementing Automation in Job Interview Grading. In In-ternational Conference on Information and Communication Technology. https://doi.org/10.1109/icoict49345.2020.9166364.

[13] Kharde, V., & Sonawane, S. S. (2016b). Sentiment Analysis of Twitter Data: A Survey of Techniques. International Journal of Computer Applications, 139(11), 5–15. https://doi.org/10.5120/ijca2016908625.

[14] Pathak, v.m., joshi, m.r., joshi, p.a., mundada, m.r., & joshi, t. (2020). Kbcnmujal@hasoc-dravidian-codemix-fire2020: using machine learning for detection of hate speech and of-fensive codemix social media text. Fire.

[15] Martins, R. S., Gomes, M., Almeida, J., Novais, P., & Henriques, P. R. (2018). Hate Speech Classification in Social Media Using Emotional Analysis. In Brazilian Conference on Intelligent Systems. https://doi.org/10.1109/bracis.2018.00019.

[16] Alsafari, S., Sadaoui, S., & Mouhoub, M. (2020). Hate and offensive speech detection on Arabic social media. Online Social Networks and Media, 19, 100096. https://doi.org/10.1016/j.osnem.2020.100096

[17] Charitidis, P., Doropoulos, S., Vologiannidis, S., Papastergiou, I., & Karakeva, S. (2020). Towards countering hate speech against journalists on social media. Online Social Networks and Media, 17, 100071. https://doi.org/10.1016/j.osnem.2020.100071.

[18] Ayo, F. E., Folorunso, O., Ibharalu, F. T., & Osinuga, I. A. (2020). Machine learning tech-niques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions. Computer Science Review, 38, 100311. https://doi.org/10.1016/j.cosrev.2020.100311.

[19] R. Artstein and m. Poesio (2008). inter-coder agreement for computational linguistics. Computational linguistics 34(4):555-596.

[20] Prasanth s n, r aswin raj, adhithan p, premjith b, and soman kp. (2022). Cen-tamil@dravidianlangtech-acl2022: abusive comment detection in tamil using tf-idf and random kitchen sink algorithm. In proceedings of the second workshop on speech and lan-guage technologies for dravidian languages, pages 70–74, dublin, ireland. Association for computational linguistics.

[21] Guo, A., & Yang, T. (2016). Research and improvement of feature words weight based on TFIDF algorithm. In 2016 IEEE Information Technology, Networking, Electronic and Au-tomation Control Conference. https://doi.org/10.1109/itnec.2016.7560393.

[22] Xiao, y. (2009). Study of tfidf algorithm. Journal of computer applications.

[23] Tf–Idf." Wikipedia, 6 Mar. 2023. Wikipedia https://en.wikipedia.org/wiki/tf%e2%80%93idf

[24] Sarkar, chatterjee, das, datta (2015) Text classification using support vector machine. In-ternational Journal of Engineering Science Invention. vol 4, PP.33-37