Hate Speech Detection

Navneet Nikhil
CSE(DBIT, VTU)
navneetnikhil007@gmail.com

Nitesh Kumar
CSE(DBIT, VTU)
niteshsingh6206@gmail.com

Sanglap Karmakar
CSE(DBIT, VTU)
sanglapkarmakar2001@gmail.com

Department of Computer Science and Engineering, Don Bosco Institute of Technology, Bangalore, India, 560074

Abstract - The massive expansion of social networking websites has made it easier for people with various cultural and psychological backgrounds to communicate directly with one another. It has led to an increase in online conflicts between them. This paper proposes an approach to detect hate speech. A publicly available dataset of tweets in language is used. Data preprocessing includes the removal of stopwords , punctuations , emojis , numbers , URLs etc. and feature extraction is carried out using tokenization , lemmatization and POS tagging. The performances of XGBoost, Random Forest , Logistic Regression and SVM have been compared in this study for the detection of hate speech. XGBoost classifier provided highest accuracy of 74.93%.

Keywords: Hate speech, Hate tweets, Machine learning.

I. Introduction

Internet users are becoming more and more interested in online social media. The services offered by social networking providers like Twitter ,Instagram and Facebook are extremely popular among internet users. Due to their prominence in the social networking space, they frequently struggle to handle rude and hateful language. Hence, such companies need to invest a lot of attention and resources to tackle and to provide a permanent solution to this problem.

Hate speech is the use of hostile, violent, or offensive language directed at a certain group of people who share a characteristic, such as gender, ethnicity, race, or religious views. There is a critical need to propose a solution to detect hate speech automatically. This would automate decision-making to turn social networking sites into a welcoming space for information sharing.

In this work, hate speech detection is carried out. Four classes hate , offensive , profane and neutral have been analyzed.

II. Literature Survey

In this paper, the authors have studied research works carried out between the years 2017 to 2022. An effective approach for detecting hate speech patterns and most prevalent unigrams has been proposed in [1].The tweets are classified into three classes mainly clean ,offensive and hateful. Features like semantic,sentiment,unigram and pattern are extracted in this approach. The accuracy achieved for binary classification was 87% whereas it was 78.4% for ternary classification.

A method based on a deep neural network combining convolutional and gated recurrent networks (GRU) is proposed in [2]. The use of GRU over LSTM helped achieve better accuracy. This approach classifies tweets based on sexism and racism. Three Deep Neural Network Architectures are suggested by [3] to identify hate speech on Twitter: GRU, which is strong at capturing sequence orders, CNN, which is good at feature extraction, and ULMFiT, which employs transfer learning. The ULMFiT model provided the best results with an accuracy of 97.5%.

A supervised learning model has been proposed in [4] to classify hate towards women on twitter. Turkish tweets based on women’s clothing has been used and machine learning algorithms achieved a maximum accuracy of 72%. Flesch KinCaid level and Flesch Reading Ease scores are used to assess the quality of the tweets.

Automatic detection of the hate tweets using machine learning using bag of words and the TFIDF is proposed in [5].A publicly available dataset from kaggle based on English tweets has been used for experimentation. An accuracy of 94% is obtained using both the above mentioned features separately. The logistic regression classifier is used to classify whether the content is hateful or not. The approach in [6][7] used n-grams as features and passed their TF IDF values to different machine learning models. An accuracy of 95.6% was achieved using this approach.

South African English tweets [8] are used to detect speech that is hateful and offensive. Word n-gram, character n-gram, negative emotions and syntactic-based features were extracted and analyzed. Gradient Boost classifier achieved an accuracy of 80.3% for hate speech.

A method of classifying online hate using machine learning that makes use of word embeddings such as Distributed Bag of Words (DBOW) and Distributed Memory Mean (DMM), as well as Word2vec Convolutional Neural Networks (CNNs) is proposed in [9].Two publicly available datasets consisting of 35000 and 25000 tweets are used in this approach to classify hate and non-hate tweets.

Hate speech detection was carried out on a dataset consisting of Urdu tweets in [10].Variable Global Feature Selection Scheme for dimensionality reduction and Synthetic Minority Optimization Technique for class imbalance were used to get better performance. [11] uses a tweet dataset to classify the text as hate and non-hate. The text is encoded using encoding techniques like BagOfWords, n-gram, Word2Vec.TF-IDF feature extraction technique is implemented and an accuracy of 77% was achieved. Subjective and semantic features are considered in [12] and a lexicon is created from hate and semantic features which is further used for developing a hate speech detection model.

III. Methodology

A. Dataset Description

The dataset consists of 25,000 tweets with four main categories : hate,offensive,profane and neutral. Each category has 5300 tweets as shown in Fig 1.

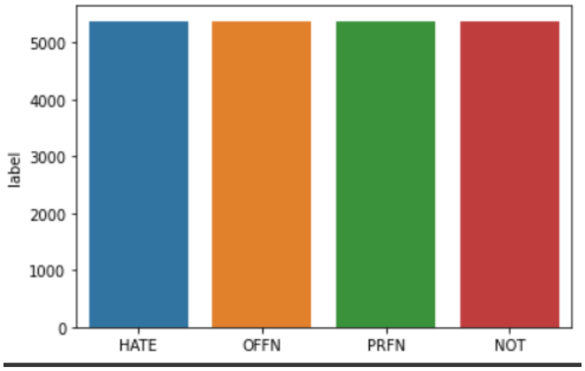

Fig.1. Class wise tweet count bar plot

B. Preprocessing

Typically, the data is presented in phrases or paragraphs, which is how people naturally communicate. Therefore, the data must be changed and cleaned up before analysis in order for the computer to interpret it in the proper language.

The first step in the preprocessing of the data was noise removal. The noise in the dataset includes the URL links and Twitter handle names. Using regular expressions, anything that comes after http/https is removed from URL links and Twitter handle names after the previous stage. Then punctuation and special characters are eliminated, followed by stop words like a, an, the, is, etc. Since they have no real significance, it is not necessary to include these stop words in order to understand the statement's sentiment.

The python strip() method is capable of eliminating these extra spaces from the beginning and end of each line. The sentiment is not discernible from the punctuation or special characters. The distribution of tweet length and tweet character count for hate and profane classes is shown in Fig 2 and for offensive and neutral classes is shown in Fig 3.

.

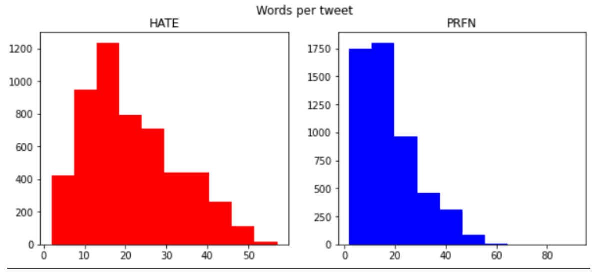

Fig.2. Distribution of words in HATE and PRFN

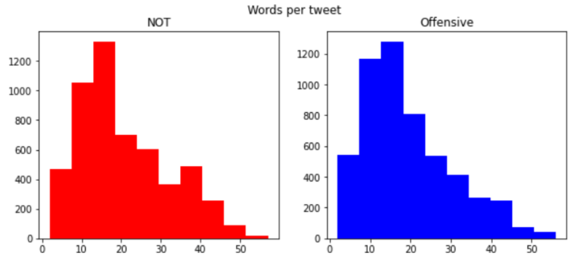

Fig.3. Distribution of words in NOT and OFF

C. Feature Extraction

To extract features from text,Part of speech tagging has been used.Unigram model is used to predict the probability of words. The following assumptions are made by the unigram model:

1. Each word's probability stands alone from those that came before it.

2. It depends on how frequently the word appears overall in the training text.

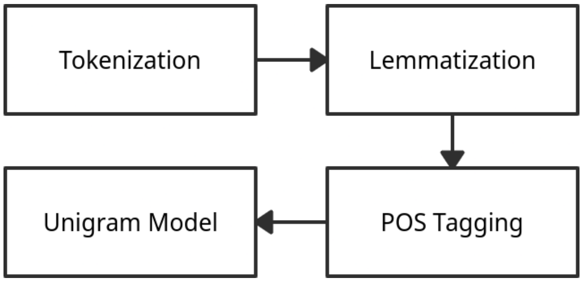

Fig.4. Feature extraction process for hate speech detection

The raw text is divided up into its component words and this process is called tokenization.Then the individual words are tagged using Part of speech tagging (POS tagging).It is also called grammatical tagging in which the words are marked corresponding to the part of speech(noun,pronoun,adjective) based on the context.
The process of feature extraction is shown in Fig 4.

Once the tokens are tagged , they are then passed to the Unigram model which gives us the probability of individual words.The words are then lemmatized to remove unnecessary processing. Lemmatization allows us to group together different forms of the same word to the base word. The system workflow is shown in Fig 5.

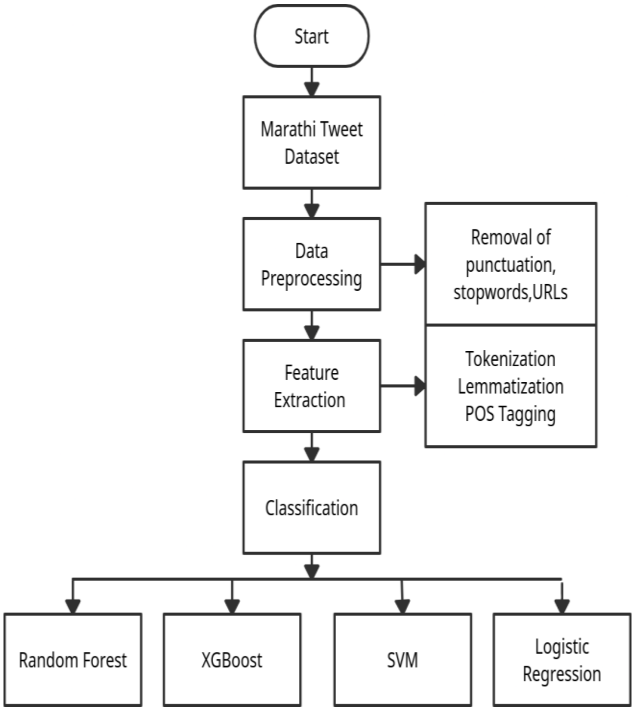

Fig.5. Overall workflow of Hate Speech Detection

These were treated as a series of words and the features were then extracted. For feature extraction, Count vectorizer is used. We used a unigram model to map the POS tags and then form the corresponding sentences. The sentences are broken down into words using the Count vectorizer to tokenize the text, and this vocabulary is then used to encode new texts.

D. Classification

There are four classes of speech i.e. hate, offensive, profane and neutral. Classification is performed using four algorithms: XGBoost, Logistic regression, Random Forest and SVM.
The first classifier used is XgBoost The equation for XGBoost is given by eq.1.

$$\widehat{y_i} = \sum_{k=1}^{k} f_k(x_i), f_k \in F \qquad (1)$$

where k, f and F denote the number of trees, functional space of F and set of CARTS respectively.

To improve the predicted accuracy of a dataset, Random Forest mixes a number of decision trees on various subsets of the data and averages the results. The equation for Random Forest is given by eq.2.

$$RF = \frac{\sum c}{T} \qquad (2)$$

where, c is the entropy of all trees and T is the total count of trees in the forest.

SVM is used for both classification and regression. Classification in SVM is carried out by finding the hyperplane which differentiates the two classes. The equation for the hyperplane is given by eq.3.

$$w.x + b = 0 \qquad (3)$$

where x is the data point, w is the vector normal to the hyperplane, and b is the bias.

In logistic regression, the dependent variable is modelled using a logistic function. The hyperparameter used is a random state whose value has been taken as 0. The equation of logistic regression is given by eq.4.

$$ln(\frac{p}{1-p}) = \beta_0 + \beta_1 X_1 + \ldots\ldots + \beta_k X_k \qquad (4)$$

IV. Results

The testing accuracies achieved by the classifiers XGBoost, Random Forest, Logistic Regression and SVM are 74.93, 74.80 ,72.32 ,70.12, respectively.

Table 2 represents the testing accuracies along with other performance evaluation metrics.

Table 1 : Performance evaluation metrics

Classifier	Accuracy %	Precision %	Recall %	F1 score %
Logistic Regression	72.32	73.23	74.34	74.00
Random Forest	74.80	80.20	70.42	75.41
Support Vector Machine	70.12	74.15	70.35	71.05
Xgboost	74.93	80.23	75.45	70.24

Recall, precision and F1 score are the other performance evaluation metrics used. By comparing these values it is observed that XGBoost gives the higher results with a f1 score 70.24%, recall 75.45% and precision 80.23%.

V. Conclusion

This paper presents a novel method for hate speech detection from tweets. This machine learning approach classifies the tweets as hate , offensive , profane and neutral. A comparative study using four machine learning models (Random Forest, SVM , XGBoost , Logistic regression) is performed and the XGBoost classifier provides a maximum accuracy of 74.93% .

VI. References

[1] Watanabe, Hajime, Mondher Bouazizi, and Tomoaki Ohtsuki. "Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection." IEEE access 2018, pp. 13825-13835.

[2] Zhang, Ziqi, David Robinson, and Jonathan Tepper. "Detecting hate speech on twitter using a convolution-gru based deep neural network." In

European semantic web conference, 2018, pp. 745-760.

[3] Amrutha, B. R., and K. R. Bindu. "Detecting hate speech in tweets using different deep neural network architectures." In 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 923-926. .

[4] Şahi, Havvanur, Yasemin Kılıç, and Rahime Belen Sağlam. "Automated detection of hate speech towards woman on Twitter." In 2018 3rd international conference on computer science and engineering (UBMK), 2018, pp. 533-536.

[5] Koushik, Garima, K. Rajeswari, and Suresh Kannan Muthusamy. "Automated hate speech detection on Twitter." In 2019 5th International Conference on Computing, Communication, Control And Automation (ICCUBEA), 2019,pp. 1-4.

[6] Gaydhani, Aditya, Vikrant Doma, Shrikant Kendre, and Laxmi Bhagwat. "Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tf idf based approach." arXiv preprint arXiv:1809.08651 (2018).

[7] Davidson, Thomas, Dana Warmsley, Michael Macy, and Ingmar Weber. "Automated hate speech detection and the problem of offensive language." In Proceedings of the international AAAI conference on web and social media, 2017, vol. 11, no. 1, pp. 512-515.

[8] Oriola, Oluwafemi, and Eduan Kotzé. "Evaluating machine learning techniques for detecting offensive and hate speech in South African tweets." IEEE Access 8 (2020): 21496-21509.

[9] Ketsbaia, Lida, Biju Issac, and Xiaomin Chen. "Detection of hate tweets using machine learning and deep learning." In 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2020, pp. 751-758.

[10] Ali, Muhammad Z., Sahar Rauf, Kashif Javed, and Sarmad Hussain. "Improving hate speech detection of Urdu tweets using sentiment analysis." IEEE Access 9 (2021): 84296-84305.

[11] Gajbhiye, Disha, Swapnil Deshpande, Prerna Ghante, Abhijeet Kale, and Deptii Chaudhari. "Machine Learning Models for Hate Speech Identification in hate speechLanguage." In Forum for Information Retrieval Evaluation (Working Notes)(FIRE), CEUR-WS. org. 2021.

[12] Gitari, Njagi Dennis, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. "A lexicon-based approach for hate speech detection." International Journal of Multimedia and Ubiquitous Engineering 10, no. 4 (2015): 215-230.