Hateful Speech Detection on Social Media

Israt Jahan
Department of Computer Science
The University of Memphis, USA
ijahan1@memphis.edu

Abstract

The rise of social media has brought about new challenges in identifying and combating hateful speech online. With the widespread use of social media platforms, it has become increasingly important to develop effective algorithms for detecting and mitigating the spread of hate speech. In this paper, we propose a comprehensive framework for identifying and classifying hateful speech on social media. Our approach utilizes a combination of machine learning techniques, natural language processing, and semantic analysis to identify and categorize different types of hate speech. We evaluate our approach on a large dataset of social media posts and show that our framework can accurately detect hate speech with high precision and recall. Our results demonstrate the effectiveness of our approach and its potential for real-world applications in mitigating the spread of hate speech on social media platforms.

Keywords: machine learning, hate speech detection, social media, online hate, natural language processing, sentiment analysis

1 Introduction

In recent years, social media has become a powerful tool for people to express their thoughts and opinions. While it has opened up opportunities for individuals to connect and engage with others on a global scale, it has also given rise to a new form of online abuse known as hate speech (Mathew et al., 2019). Hate speech can take many forms, from direct attacks on individuals or groups based on their race, gender, religion, or other characteristics, to more subtle forms of harassment and discrimination. The spread of hate speech on social media has become a major concern for individuals, organizations, and governments around the world.

Despite the efforts of social media companies to address this issue, hateful speech remains a persistent problem on their platforms, due in part to the sheer volume of content that is posted every day. This has led to a growing need for automated tools that can detect and flag instances of hate speech on social media, but the development of such tools is a complex and challenging task. Some of the key challenges include defining what constitutes hate speech, dealing with the nuances of language and context, and ensuring that the tools are effective and accurate without suppressing legitimate forms of expression. The development of effective hate speech detection tools is crucial to create a safer and more inclusive online environment for all users, but it seeks continuous research and innovation efforts.

In this research endeavor, we implement multiple machine learning models for detecting hate speeches on social media, including Logistic Regression, Random Forest, Naive Bayes Classifier, and Support Vector Machine (SVM). We utilize a curated dataset (Mody et al., 2023) to train, validate, and test the models. We also evaluate the performance of each model based on a list of metrices, including precision, recall, F1-score, and accuracy.

The rest of the paper is organized as follows. We include related research in section 2 and discuss machine learning models, their implementations, and the utilized dataset in section 3. In section 4, we present the experimental results and discuss the performance of the models in detail. Finally, we conclude the paper by summarizing main points in section 5.

2 Related Works

Detecting hate speech on social media is a popular research problem in the literature, and we see several research efforts to solve it. For example, (Jiang and Zubiaga, 2021) investigate the cross-lingual hate speech detection task, tackling the problem by adapting hate speech resources from one language to another. The authors, in this paper, propose a cross-lingual capsule network learning model coupled with extra domain-specific lexical semantics for hate speech (CCNL-Ex). (Cao et al., 2020) propose a novel deep learning model that combines multi-faceted text representations such as word embeddings, sentiments, and topical information, to detect hate speech in online social platforms. On the other hand, (Yang et al., 2022) propose a scalable cross-domain knowledge transfer (CDKT) framework, where the mainstream vision-language transformer can be employed as backbone flexibly. It provides a stable improvement compared with baselines and produces a competitive performance compared with some existing multi-modal hate speech detection methods.

However, the literature needs more research efforts to effectively address this hate speech issue. (Arango et al., 2019) discuss the implications for current research and re-conduct experiments to give a more accurate picture of the current state-of-the art methods. (Gröndahl et al., 2018) reproduce seven state-of-the-art hate speech detection models from prior work, and shows that they perform well only when tested on the same type of data they were trained on.

3 Technical Details

In this section, we discuss several machine learning models we utilize to detect hate speech on social media platforms. We also discuss how we process a curated dataset for training and testing the models. Finally, we conclude this section by briefly discussing how we implement the chosen models.

3.1 Models

In this research project, we use the following four popularly used machine learning models for the hate speech detection task:

1. Logistic Regression model,
2. Random Forest classifier model,
3. Naive Bayes classifier model, and
4. Support Vector Machine (SVM) model

We briefly define each of the listed model in the following paragraphs.

3.1.1 Logistic Regression Model

Logistic regression is a statistical method used to model the probability of a binary outcome based on one or more predictor variables. It uses a logistic function to estimate the relationship between the predictor variables and the probability of the outcome variable. The logistic regression model estimates the coefficients of the predictor variables that maximize the likelihood of observing the actual outcome values, given the values of the predictor variables. These coefficients can then be used to make predictions for new observations.

3.1.2 Random Forest Classifier

Random forest is an ensemble machine learning algorithm used for classification and regression tasks. It combines multiple decision trees to make a final prediction by averaging the results of each individual tree. Each tree in the forest is trained on a random subset of the input features and a random subset of the training data, which helps to reduce overfitting and improve the model's accuracy.

3.1.3 Naive Bayes Classifier

Naive Bayes is a probabilistic machine learning algorithm used for classification tasks. It is based on Bayes' theorem (Joyce, 2003), which describes the probability of an event occurring based on prior knowledge of conditions that might be related to the event. Naive Bayes assumes that the input features are conditionally independent of each other given the class variable, which simplifies the probability calculations and makes the algorithm computationally efficient.

3.1.4 Support Vector Machine

SVM is a supervised machine learning algorithm used for classification and regression tasks. It finds a hyperplane that separates the input data into different classes, maximizing the margin between the closest points of each class. SVM can handle high-dimensional data and is effective in dealing with non-linearly separable data by using kernel functions to transform the input data into a higher-dimensional space.

3.2 Dataset

To train and further test the machine learning models, we use a curated dataset for hate speech detection on social media text compiled by (Mody et al., 2023). This dataset is curated from various sources like Kaggle, GitHub, and other websites. It contains hate speech sentences in English and is confined into two classes namely *hateful* and *non-hateful* classes. It has 451,709 sentences in total. Among them, 371,452 are hate speech, and 80,250 are non-hate speech. An augmented balanced dataset with 726,120 samples is also generated to create a custom vocabulary of 145,046 words. The total number of contractions considered in the dataset is 6403. The total number of bad words usually used in hateful content is 377. The text in each sentence of the final dataset, which is utilized for training and cross-validation, is limited to 180 words.

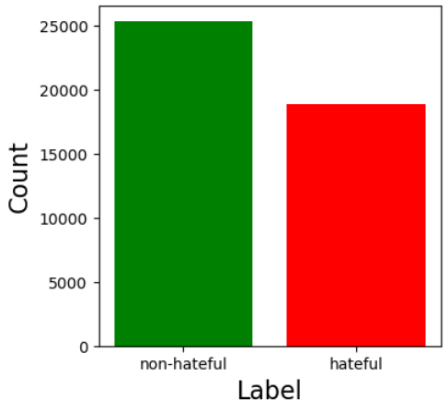


Figure 1: Class Distribution

3.3 Data Processing

Before using the dataset to train the machine learning models, we perform some data cleaning and preprocessing tasks. First, we remove all missing values from the dataset. Then, we discard irrelevant information, such as stop words, punctuation marks, and special characters, that do not add any meaningful values to the text. We perform these cleaning tasks to reduce noise in the data and improve overall quality of the text. Later, we standardize the text by 1) converting all words to lowercase, 2) correcting spelling errors, and 3) removing unnecessary spaces and duplicate words. We also use built-in NLTK (nltk, 2023) modules and packages for data preprocessing. Figure 1 shows the distribution of the two classes containing processed speech texts.

3.4 Model Implementation

Before implementing models, we extract feature vectors from the dataset using TF-IDF (Akuma et al., 2022) and sentiment analysis (Zhou et al., 2021). For TF-IDF, we use the *sklearn.feature_extract.text.TfidfVectorizer* module (Scikit-Learn, 2023) to extract features from the raw dataset. The parameter values we use for the *TfidfVectorizer* module is presented in Table 1. On the other hand, for sentiment analysis, we utilize the NLTK (nltk, 2023) library to extract four different types of sentiment features: 1) positive sentiment,

2) negative sentiment, 3) neutral sentiment, and 4) compound sentiment.

We implement all four models using Scikit-Learn (sklearn, 2023a) and Python. Prior to the training operations, we split the dataset into two parts: 1) training dataset that contains 80% of the original processed dataset, and 2) testing dataset that contains the rest 20% of the processed dataset.

Paramater	Configuration
ngram_range	(1, 2)
max_df	0.75
min_df	5
max_features	10000

Table 1: TF-IDF Vectorizer Configuration

We train our models with two separate feature sets: 1) features extracted using TF-IDF, and 2) features extracted using both TF-IDF and sentiment analysis. We run our experiment on a Windows machine with 64-bit AMD processor and 8 GB RAM.

4 Results and Discussions

We analyze the performance of the models in terms of: *precision*, *recall*, *F1 score* and *accuracy* (sklearn, 2023b). In the following paragraphs, we shortly define each performance metric before we discuss the experimental results.

4.1 Performance Metrices

After we run our experiment, we consider the following four metrices to evaulate the performance of the models.

4.1.1 Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. It measures the proportion of true positives among all the positive predictions. A high precision score indicates that the model is making fewer false positive predictions. Equation (1) presents the mathematical formula to compute the precision of a model.

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \tag{1}$$

4.1.2 Recall

Recall is the ratio of correctly predicted positive observations to the total actual positive observations. It measures the proportion of true positives among all the actual positive observations. A high recall score indicates that the model is correctly identifying most of the positive observations. The mathematical formula to compute the recall of a model is presented in the following equation (2):

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \tag{2}$$

4.1.3 F1 Score

F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is a useful metric when both false positives and false negatives are equally important. A high F1 score indicates that the model has high precision and high recall. We compute the F1 score of a model using the following equation (3):

$$\text{F1 Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

4.1.4 Accuracy

Accuracy is the ratio of correctly predicted observations to the total number of observations. It measures how well the model is predicting both positive and negative observations. However, accuracy can be misleading in imbalanced datasets, where the number of positive and negative observations is not equal. The mathematical formula of computing the accuracy of a model is presented in the equation (4).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

Hence, TP = true positive, TN = true negative, FP = false positive, and FN = false negative.

4.2 Performance Evaluation

We evaluate the performance of the machine learning models based on the performance metrices explained in section 4.1. The summary of the overall performance is tabulated in Table 2 and Table 3. We also depict it in Figure 2 to visually explain the performance of the models.

Models	**Precision**	**Recall**	**F1 Score**	**Accuracy**
Logistic Regression	**0.96**	0.85	0.90	0.92
Random Forest	0.94	0.89	**0.92**	0.93
Naive Bayes	0.77	0.58	0.66	0.75
SVM	0.94	0.87	0.90	0.92

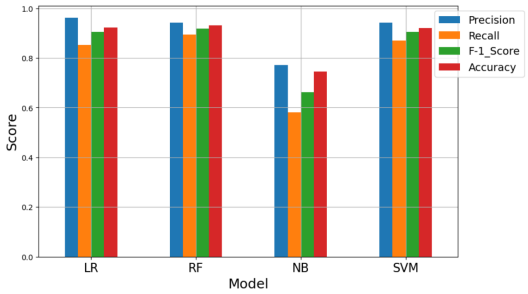
Table 2: Performance measurement for TF-IDF

Models	**Precision**	**Recall**	**F1 Score**	**Accuracy**
Logistic Regression	**0.96**	0.85	0.90	0.92
Random Forest	0.95	0.88	**0.91**	0.93
Naive Bayes	0.75	0.62	0.68	0.75
SVM	0.75	0.62	0.68	0.75

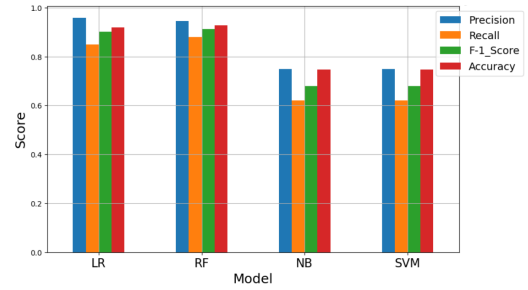
Table 3: Performance measurement for TF-IDF with Sentiment Analysis

As we see in Table 2, when the models are trained with features extracted using only TF-IDF, Random Forest model outperforms others with an F1 score of **0.92**. But Logistic Regression model gives us the highest precision score of **0.96**, i.e., it predicts very few false positives. We also gain comparable performance (see Table 3) from the models when they are trained with a feature set discovered through both TF-IDF and sentiment analysis. In summary, we can conclude that Logistic Regression is the best model to detect hate speech when we try to minimize the incorrect identification of hate speech, but if we aim to maximize the correct identification

of hate speech along with the minimization of incorrect identification, Random Forest is the best model among the chosen four models.



(a) For TF-IDF



(b) For TF-IDF with Sentiment Analysis

Figure 2: Performance measurement

5 Conclusions

In this research, we explore four different machine learning models to detect hate speech from text data: 1) Logistic Regression, 2) Random Forest, 3) Naive Bayes, and 4) Support Vector Machine (SVM). We discover feature vectors from a curated dataset through TF-IDF and sentiment analysis, and train our models with two separate feature sets. We evaluate the performance of each model based on four matrices: precision, recall, F1 score, and accuracy. It is evident that Random Forest outperforms other models achieving both the highest F1 score and accuracy score. However, if the objective of hate speech detection is to minimize the false positives, Logistics Regression is a better choice for us. In future, we aim to address some limitations that we face during this research endeavor. For example, texts generated in social platforms depend heavily on cultural and linguistic contexts, and therefore, we must consider available contexts of texts to improve the effectiveness of the detection models. The dataset we use in this research also may not be representative of all types of hate speech found on social media. Therefore, we also need to diversify the training data to allow the machine learning models learn more accurately about the characteristics of hate speech and detect hateful contents effectively on social media platforms.

Acknowledgement

I would like to thank my Instructor, Dr. Xiaofei Zhang, for his guidance and support throughout this research project.

References

Stephen Akuma, Tyosar Lubem, and Isaac Terngu Adom. 2022. Comparing bag of words and tf-idf with different models for hate speech detection from live tweets. *International Journal of Information Technology*, page 3629–3635.

Aymé Arango, Jorge Pérez, , and Barbara Poblete. 2019. Hate speech detection is not as easy as you may think: A closer look at model validation. *30th ACM Conference on User Modeling, Adaptation and Personalization*, pages 45–54.

Rui Cao, Roy Ka-Wei Lee, and Tuan-Anh Hoang. 2020. Deephate: Hate speech detection via multi-faceted text representations. *12th ACM Conference on WebScience (WebSci'20)*, pages 11–20.

Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N. Asokan. 2018. All you need is "love": Evading hate speech detection. *11th ACM Workshop on Artificial Intelligence and Security*, pages 45–54.

Aiqi Jiang and Arkaitz Zubiaga. 2021. Cross-lingual capsule network for hate speech detection in social media. *32nd ACM Conference on Hypertext and Social Media*, pages 217–223.

James Joyce. 2003. Bayes' theorem. `https://plato.stanford.edu/archives/spr2019/entries/bayes-theorem/`. Accessed: 2023-05-01.

Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. *WebSci '19: Proceedings of the 10th ACM Conference on Web Science*, pages 173–182.

Devansh Mody, Yi Dong Huang, and Thiago Eustaquio Alves de Oliveira. 2023. A curated dataset for hate speech detection on social media text. *Data in Brief*, pages 1–6.

nltk. 2023. Natural language toolkit. `https://https://www.nltk.org/`. Accessed: 2023-04-30.

Scikit-Learn. 2023. sklearn.feature_extraction.text.tfidfvectorizer. `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html`. Accessed: 2023-05-01.

sklearn. 2023a. scikit-learn. `https://scikit-learn.org/stable/modules.html`. Accessed: 2023-04-30.

sklearn. 2023b. scikit-learn. `https://scikit-learn.org/stable/modules/model_evaluation.html`. Accessed: 2023-04-30.

Chuanpeng Yang, Fuqing Zhu, Guihua Liu, Jizhong Han, and Songlin Hu. 2022. Multimodal hate speech detection via cross-domain knowledge transfer. *Proceedings of the 30th ACM International Conference on Multimedia*, page 4505–4514.

Xianbing Zhou, Yong Yan, Xiaochao Fan, Ge Ren, Yunfeng Song, Yufeng Diao, Liang Yang, and Hongfei Lin. 2021. Hate speech detection based on sentiment knowledge sharing. *59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 1–6.