International Journal of Recent Advances in Multidisciplinary Topics
Volume 4, Issue 6, June 2023
https://www.ijramt.com | ISSN (Online): 2582-7839

Hate Speech Detection Using Supervised Natural Language Processing for Videos and Text

Anuj Rawat[1*], Divyansh Sharma[2], Kanishk Tyagi[3], Ishaan Chadha[4], Nidhi Chandra[5]
[1,2,3,4]*Student, Amity School of Engineering and Technology, Amity University, Noida, India*
[5]*Assistant Professor, Amity School of Engineering and Technology, Amity University, Noida, India*

***Abstract*: Hate speech is an issue that frequently happens when somebody speaks with one another utilizing social media on the Web. Despite the fact that artificial intelligence frameworks are set up to ban such messages. The main issue that arises in these cases is the high false positive rates, so we need mechanisms to help counter such issues and distinguish hate speech without sabotaging the freedom of expression. Likewise, the identification of hate speech in recordings is more difficult than straightforward plain text. In this project we'll research and make a system for detection of hate speech in videos and text utilizing a classification approach in AI with different techniques such as Logistic Regression, SVM, decision tree. The execution of video to text is also something we'll work on. The recent uptick of people using social media has led to a lot of conflicting views, although healthy conversation is an important part of debates some of the content that is being shared online is offensive and can often be downright hateful. It is important to recognize which messages are actually hateful and offensive and which are not to encourage healthy debates over the internet.**

***Keywords*: Machine Learning, Hate Speech, SVM, Natural Language Processing, Decision Tree, Logistic Regression.**

1. Introduction

Hate speech is an issue that frequently happens when somebody speaks with one another utilizing social media on the Web. Despite the fact that artificial intelligence frameworks are set up to ban such messages. The main issue that arises in these cases is the high false positive rates, so we need mechanisms to help counter such issues and distinguish hate speech without sabotaging the freedom of expression. Likewise, the identification of hate speech in recordings is more difficult than straightforward plain text. In this project we'll research and make a system for detection of hate speech in videos and text utilizing a classification approach in AI with different techniques such as Logistic Regression, SVM.

It's been observed that many people are often affected by hate speech in videos as well as text. This project aims to implement hate speech recognition using AI Ml which would prove to be extremely useful in this world of social media. From using a dataset then preprocessing, classifying and creating an evaluation model for it and then to find the result, all will be done by the system.

2. Methodology Used

A. General Methodology

- SVM, Decision Tree, Random Forest, and speech recognition for the project's video to text component will all be employed as the methodology for this paper's detection of hate speech and objectionable words.
- Supervised NLP will be used to create the model.
- Using a dataset then preprocessing, classifying, and creating evaluation models for it and then find the result.
- Random Forest- Large volumes of data are classified using a variety of machine learning techniques, including Random Forest.
- One of the most well-liked supervised learning algorithms, Support Vector Machine, or SVM, is used to solve Classification and Regression problems. However, it is largely employed in Machine Learning Classification issues.
- Supervised Machine Learning techniques like decision trees involve continuously segmenting the data based on a particular parameter.
- We will also look into using NLP based API's to transcribe the audio and video files into text files.

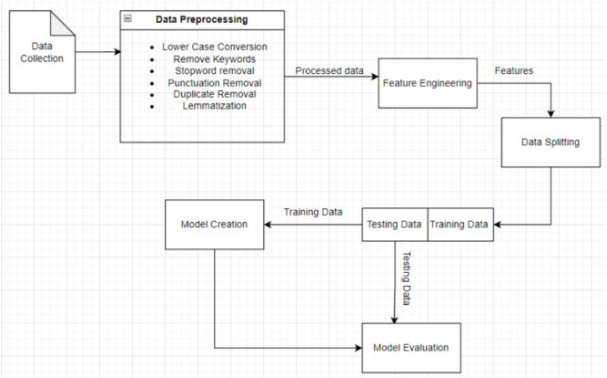

Fig. 1. Methodology flowchart

B. Data Collection

For this study we selected a public dataset which labeled data

*Corresponding author: rawatanuj10@gmail.com

into two categories with their respective classifiers. The dataset had a total of 31962 tweets and were labeled into two distinct classes, namely, hate speech and non-hate speech. The dataset consisted of about 7.04% of tweets classified as Hate Speech and 92.96\% as Non-Hate Speech. The distribution of the dataset is given in figure below.

Label	Classifier	Count
Hate Speech	1	2242
Non-Hate Speech	0	29720

Fig. 2. Data collection

C. *Data Pre-processing*

Data pre-processing is an important precursor to feature engineering, because it cleans the data and reduces the possibility of redundant, noisy and non-informative data. A lot of preprocessing functions were used to prepare the data. First the data was converted into lower case, then links and keywords like hashtags and @s were removed, then redundant symbols and stop-words were removed, then the tweets were tokenized. Then duplicates were dropped and the tokens were lemmatized meaning similar meaning words of words that were used in similar contexts were converted into their root forms.

```
#creating a function to process the data
def data_processing(tweet):
    tweet = tweet.lower()
    tweet = re.sub(r"https\S+|www\S+http\S+", '', tweet, flags = re.MULTILINE)
    tweet = re.sub(r'\@w+|\#','', tweet)
    tweet = re.sub(r'[^\w\s]','',tweet)
    tweet = re.sub(r'δ','',tweet)
    tweet_tokens = word_tokenize(tweet)
    filtered_tweets = [w for w in tweet_tokens if not w in stop_words]
    return " ".join(filtered_tweets)

tweet_df.tweet = tweet_df['tweet'].apply(data_processing)

tweet_df = tweet_df.drop_duplicates('tweet')

lemmatizer = WordNetLemmatizer()
def lemmatizing(data):
    tweet = [lemmatizer.lemmatize(word) for word in data]
    return data

tweet_df['tweet'] = tweet_df['tweet'].apply(lambda x: lemmatizing(x))
```

Fig. 3. Data preprocessing

D. *Feature Engineering/Vectorization*

Feature engineering or feature extraction is the process of converting raw data (text in this case) to vectors of numbers that can be used by machine learning models.

E. *Data Split*

The training and testing portions of the data set were randomly divided. The classification model, which employed a number of different classification algorithms, was trained using the training data, and the model's output was assessed using the testing data.

F. *Classifier Evaluation*

After training a classification model, the next step is to evaluate the results we get when we test the model using testing data to make sure that the results meet the expectations set by the creator. To evaluate the model, we use a confusion matrix. A confusion matrix is table or a matrix with 4 combinations of expected and predicted values. The four combinations include True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN).

```
X = tweet_df['tweet']
Y = tweet_df['label']
X = vect.transform(X)

x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

print("Size of x_train:", (x_train.shape))
print("Size of y_train:", (y_train.shape))
print("Size of x_test: ", (x_test.shape))
print("Size of y_test: ", (y_test.shape))

Size of x_train: (25569, 481885)
Size of y_train: (25569,)
Size of x_test:  (6393, 481885)
Size of y_test:  (6393,)
```

Fig. 4. Data split performed on the dataset

There are four majors for evaluating a model; these are Accuracy, Precision, Recall and F-score.

Precision – How accurately a model predicts depends on precision. It can alternatively be described as the proportion of accurate positive predictions to all of the model's positive predictions.

$$Precision = \frac{TP}{(TP+FP)}$$

Accuracy – Accuracy determines how correctly the model predicts classifiers. It can alternatively be described as the proportion of accurate predictions to all predictions. Or

$$Accuracy = \frac{(TP+TN)}{TP+FP+TN+FN}$$

Recall – Recall is the measure of how effective a model is at predicting the class of values. It can alternatively be characterized as the proportion of the number of values successfully predicted compared to the total number of values expected to be correctly forecasted. Or

$$Recall = \frac{TP}{(TP+FN)}$$

F1-Score – F-score is the harmonic mean of precision and recall. Or

$$F-measure = \frac{2\times(precision\times recall)}{(precision+recall)}$$

The confusion matrix for the logistic regression model is given in fig. 5.

```
style.use('classic')
cm = confusion_matrix(y_test, logreg_predict, labels=logreg.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=logreg.classes_)
disp.plot()

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1f17be5f2e0>
```

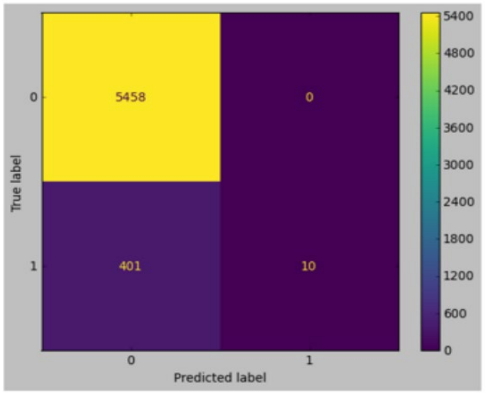

Fig. 5. Confusion matrix for logistic regression

G. Hyper Parameter Tuning

Hyper parameter tuning is the process of selecting parameters that increase the efficiency and accuracy of a machine learning model. We use hyper parameter tuning to significantly increase the accuracy of our model.

```
param_grid = {'C':[100, 10, 1.0, 0.1, 0.01], 'solver' :['newton-cg', 'lbfgs','liblinear']}
grid = GridSearchCV(LogisticRegression(), param_grid, cv = 5)
grid.fit(x_train, y_train)
print("Best Cross validation score: {:.2f}".format(grid.best_score_))
print("Best parameters: ", grid.best_params_)
```

```
Best Cross validation score: 0.96
Best parameters:  {'C': 100, 'solver': 'lbfgs'}
```

```
y_pred = grid.predict(x_test)
```

```
logreg_acc = accuracy_score(y_pred, y_test)
print("Test accuracy: {:.2f}%".format(logreg_acc*100))
```

```
Test accuracy: 95.67%
```

Fig. 6. Hyper parameter tuning performed

H. Audio Recognition

The implementation of recognizing audio-based hate speech is very straightforward. It involves first transcribing the words said in an audio file into text and then feeding the text back into our model. There are various methods to implement this, it can be done by either using built in python speech recognition libraries or by using API's like Assembly AI's API. We make the use of Assembly AI's API to transcribe the audio data to text. We do this by sending a request to Assembly AI with the audio file we need transcribed, and then receiving the transcribed text back.

I. Dataset

The dataset used in this study is a publicly available dataset that compiles a large number of tweets on the basis of two classifiers, that are hate speech and non- hate speech, which are represented by 1 and 0 respectively. After using this dataset to train our model we add the test data to improve upon our model.

3. Results

In this section, we discuss the results and findings we achieved after performing this study. The results were compiled as a confusion matrix of the different TP, TN, FP and FN obtained when testing out our model. The results were then compared to the values obtained after Hyper Parameter Tuning. There was about 2% increase in the accuracy of the model with significant increase in precision recall and F1-score. Without hyper parameter tuning the precision, recall and F1-score were 0.94, 0.94 and 0.91 respectively but after hyper parameter tuning, they showed significant improvement. The values for precision, recall and F1-score were 0.96, 0.96, 0.95 respectively.

We can clearly see that there is a significant decrease in the number of false positives and false negatives predicted by our model.

Before hyper parameter tuning about 405 tweets were incorrectly predicted but after hyper parameter tuning only 272 tweets were incorrectly predicted.

```
print(confusion_matrix(y_test, y_pred))
print("\n")
print(classification_report(y_test, y_pred))
```

```
[[5932    5]
 [ 272  184]]


              precision    recall  f1-score   support

           0       0.96      1.00      0.98      5937
           1       0.97      0.40      0.57       456

    accuracy                           0.96      6393
   macro avg       0.96      0.70      0.77      6393
weighted avg       0.96      0.96      0.95      6393
```

Fig. 7. Confusion matrix after hyper parameter tuning

4. Conclusion

In this project, we worked towards a real-world problem statement and got to learn a lot of things from our research work that we did towards drawing meaningful insights and recommending focused strategies to improve revenue and enhance customer retention. Here we derived the steps and the values that we will be requiring to implement this project which includes Understanding the problem statement, Analyzing the best approach present, Documenting the research and steps of Data Visualization, Data Pre-Processing, Vectorization and Data Split, Applying Logistic Regression, Confusion Matrix & Classification Report and Hyper Parameter Tuning. Basically, in this Project, we did all the research and documentation work and finalized the approaches and the steps which will help while implementing the methodology step by step. We implemented the methodology and made various observations which are presented in the result section of the paper.

5. Limitations

Although there are various methods to train a machine learning model it's almost impossible to achieve 100% accuracy. One more problem lies in sentiment analysis of a statement, although we can understand the sentiment of a person through a statement made by them with relatively high accuracy, some statements can be satire and sarcasm and therefore be incorrectly flagged as offensive speech as these issues can only be correctly recognized by humans.One more problem that one might come across for text-based models is the hate speech in the form of videos. Although we can use audio and convert it into text with relative ease, for videos, computer vision would also play a big part in making an accurate model. This also proposes a dataset problem where we have to make sure that our dataset is ever-expanding to accommodate new data that is constantly being added to our dataset. Thus, making training and testing the model have very high time and space complexity and require a lot more computational resource.

References

[1] Huang, X., Xing, L., Dernoncourt, F., & Paul, M. J. (2020). Multilingual twitter corpus and baselines for evaluating demographic bias in hate speech recognition.

[2] Pawar, A. B., Gawali, P., Gite, M., Jawale, M. A., & William, P. (2022, April). Challenges for Hate Speech Recognition System: Approach based on Solution. In 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) (pp. 699-704).

[3] William, P., Gade, R., Chaudhari, R., Pawar, A. B., & Jawale, M. A. (2022, April). Machine Learning based Automatic Hate Speech Recognition System. In 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) (pp. 315-318).

[4] Aljero, M. K. A., & Dimililer, N. (2021). A Novel Stacked Ensemble for Hate Speech Recognition. Applied Sciences, 11(24),11684.

[5] Del Vigna, F., Cimino, A., Dell'Orletta, F., Petrocchi, M., & Tesconi, M. (2017). Hate me, hate me not: Hate speech detection on Facebook. In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17) (pp. 86-95).

[6] Al-Makhadmeh, Z., Tolba, A. Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing 102, 501–522 (2020).

[7] indhu Abro, Sarang Shaikh, Zahid Hussain Khand, Zafar Ali, Sajid Khan and Ghulam Mujtaba, "Automatic Hate Speech Detection using Machine Learning: A Comparative Study" International Journal of Advanced Computer Science and Applications (IJACSA), 11(8), 2020.

[8] Mullah, Nanlir Sallau, and Wan Mohd Nazmee Wan Zainon. "Advances in machine learning algorithms for hate speech detection in social media: a review." IEEE Access (2021).

[9] Alfina, Ika, et al. "Hate speech detection in the Indonesian language: A dataset and preliminary study." 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2017.

[10] Kovács, G., Alonso, P., & Saini, R. (2021). Challenges of hate speech detection in social media. SN Computer Science, 2(2), 1-15.

[11] Mac Avaney S, Yao H-R, Yang E, Russell K, Goharian N, Frieder O (2019) Hate speech detection: Challenges and solutions. PLoS ONE 14(8): e0221152.