# Hate Speech Classification

##### **Akshith Alwala, Varsha Karanam, Mitali Tavildar**

Department of Intelligent Systems Engineering

Luddy School of Informatics, Computing and Engineering

***Abstract*-** With the popularity and ease of access in digital communication, we notice a rise in use social media platforms as a mode of communication. But along with its several pros there are certain cons such as stalking, sexual harassment and even abuse towards sexuality, race, gender, etc. In the recent Hate Crime Statistics released by FBI [2], 8,263 hate crime incidents involving 11,129 offenses were identified. The first step towards preventing such cases would be identifying them. We propose a solution to classify such hate speech from the online data (specifically, tweets). We propose Convolutional Neural Network structures that capture the hate speech content and classify it into hate speech or non-hate speech. We plan to use the semantic content analysis techniques built on NLP and ML models.

***Index Terms***- Hate, Crime, Classify, NLP, Convolutional Neural Network

**Introduction**

Social media abuse and related issues are avoidable. This behavior induces hatred and affects people at such extreme extents as it could be life taking. In this modern business techniques that are followed, we believe in criticism and how it increases productivity, but criticism is different from hatred and abuse. There is special importance given to mental health which could also be negatively affected by bombarding of abusive thoughts.

To classify the hate speech from data, there needs to be identification done and there is no legal definition of same. The only type of definition that is given in few related papers is [1], any type of communication that criticizes a person or a group of people based on some identifications such as race, color, ethnicity, gender, nationality, religion, etc. This causes a problem due to subjectivity since speech & context is open to interpretation. The main step in building the hate speech classifier is to detect and bring down the hate speech online. For several years social media websites such as Facebook, Twitter have been trying to eliminate this hate speech using several other methods but were not that very successful. The basic methods used to classify hate speech were not sufficient so we came up with different approach that could classify hate speech and non-hate speech. Building effective counter measures for online hate speech requires as the first step, identifying and tracking hate speech.

We found datasets that have data annotated into hate speech or not, based on the keywords and user judgements. We created a CNN model for classification and compared Random Forest and Logistic Regression. The process started with preprocessing the tweets, followed by tokenizing as required. Using keras Sequential models, we trained the CNN model to classify hate speech.

The remaining part of this paper is structured as follows. Section 2 reviews related work on hate speech classification and other relevant fields; Section 3 describes the method that we have designed and mentions the data set that we have taken into consideration; Section 4 describes the experimental setup that we have implemented; Section 5 presents graphs and results; and finally in Section 6 we conclude our Hate Speech Classification and discuss future work.

**Related Works**

1. The paper: *Ashwin Geet d’Sa, Irina Illina, Dominique Fohr. Classification of Hate Speech Using Deep Neural Networks. Revue d’Information Scientifique & Technique, Centre de Recherche sur l’Information Scientifique et Technique (CERIST), 2020, From Data and Information Processing to Knowledge Organization: Architectures, Models and Systems, 25 (01). ffhal-03101938f* [3] clearly discusses the anti-social behaviour and proposes classification of hate speech using deep neural networks.

The discussion begins with giving out some examples of hate speech, and explains how this speech could be explicit or implicit. Following are a couple of effects described:

1. “Hate content on the Internet platform can create fear, anxiety and threat to the individuals. In the case of a company or online platform, company or platform may lose its reputation or the reputation of its product.”
2. *“A report from the news article states that during the recent crisis of COVID-19, there has been a 900 percent surge in the hate speech against people from China and other Asian origins on Twitter”*

The paper further discusses the challenges in NLP and the proposed approaches to automate hate speech classification using numerical representation of text in order to use as input to the classification model. This is then followed by the deep learning techniques like Deep learning-based LSTM and GRU model to capture results better than traditional classifiers. Multiple models were trained and were tested upon.

The result achieved: “The feature-based approach is able to correctly predict up to 31% of the hate speech tweets (table 3), while BERT fine-tuning achieved 53% (table 4)”

The blog: “**Classifying Hate Speech: an overview**” [3] discussed precisely the issue. This includes a supporting case study followed by the ML approach. It discusses the baseline multilabel classification. This is then taken forward to compare with Transfer learning and Weak Supervisor & Voting based classification.

**Method**

To achieve the mentioned objective, we implemented the following 3 algorithms:

Baseline algorithms: Logistic regression, Random Forest

Deep Neural Network: CNN using keras Sequential layers

1. Logistic regression:

Steps followed: -

* Data preprocessing – cleaning the tweets, compiling the hate speech annotations into one column(isHateSpeech) such that 1 represents hate speech and 0 represents normal text.
* Splitting the data into labels and features. Label – isHateSpeech, features – preprocessed tweets.
* Feature engineering - Vectorizing text data using tf-idf was implemented. [5]

We researched on a few feature engineering techniques used like Bag of Words, bag of n-grams and tf-idf.

Bag of words- is based on simply word count statistics and does not consider relevance/context semantically.

Bag of n-Grams- this method retains the original sequence and the text better than BoW but is computationally costly

Tf-idf- is the most efficient option, that considers both frequency and relevance.

* We fit the Logisitc Regression model and achieved an accuracy of 92.5%

1. Random Forest:

* We used the preprocessed data from the above method & split it into labels and features, same as above.
* For feature engineering, we used tokenizer. Tokenizer to tokenize each word in the tweet. This was followed by converting the tokens into sequences according to the data.

These sequences of token were then padded to maintain uniformity of the size of the sequences.

* The RandomForest classifier was used to fit our padded data for training.

This prediction resulted in an accuracy of 88.5%

1. CNN model:

* The same preprocessed data along with feature engineering as tokenizer was implemented.
* The CNN usually implements convolution and pooling. Conv1D is a keras layers that is used for text input. This is a dot-product of between the embedding values and the weight vector. Followed by pooling, which is input feature mapping used to extract representative values from each input.
* Training the whole CNN and applying the optimal parameters, we achieved an accuracy of 84%.

**Experimental setup**

Data pre-processing is done using two procedures:

1. Tokenizer

2. TF-idf

One important parameter we have set during the pre-processing phase is oov token. This is useful for prediction. There will be a case where in which the text corpus will not have one or more words present in the text sequences. We set all those words to oov. As this is a sentiment analysis one word will not make much of difference. It is the resultant meaning of all the words that is important.

The hate speech classifier model that we have designed has 4 layers that are stacked sequentially.

1. Embedding Layer
2. Convolutional Layer
3. Pooling Layer
4. Dense Layer

We first have the embedding layer that basically projects/vectorizes each word and projects it onto an n-dimensional space so that the words that have no hate components fall to one extreme end and the ones with the hate component or negative sense fall onto the other extreme. The other words fall between these extremes and their direction conveys their meaning/ intent. Below is an example figure of embedding. [6]

Chart, diagram, line chart

Description automatically generated

Fig 1. Embedding Process In 2-D

Whenever there is a new sentence, the aggregate or resultant of vectors corresponding to each word is calculated, there by classifying to hate or non-hate speech. The vocab size is 36000. We chose this value because vocab size should be more than the number of entries in the text corpus. Our text corpus has around 35200 unique words/tokens. This can be varied depending upon the size of the text corpus. We have used the embed dimension as 32. We have also tried using 64 as embed dimension but 32 works slightly better. Also, the maximum length is set to 40 as we have taken the length as 40 during padding. We are using a 1D convolution layer as it is text classification. We have used 64 filters with a kernel size of 40. We are using global average pooling to reduce overfitting in the system. We then pass this through a fully connected neural networks with relu activation and then to another dense neural network with a filter size of 1 and sigmoid activation. we are choosing sigmoid function because we want the output to be between 0 and 1 as it is a classification problem.

Data Set is taken from Kaggle it has the data taken from twitter. It has labelled data with usernames as starting with a special character@ and the tweet followed by the username. And corresponding label ‘IsHatespeech’ which if 1 means it is hate speech/negative sense and viceversa.[7]

**Results**

Our experiments could not identify a best performing candidate among the three state-of-the-art methods on all datasets, by all measures. Therefore, in the following discussion, unless otherwise stated, we compare our methods against the best results achieved by any of the three state-of-the-art methods. The experiment that we have conducted to classify hate speech into hate and non-hate speech respectively gave us results with an accuracy of 84.6%. The graph for the same is shown below:

Chart, line chart

Description automatically generated

Fig 2. Training Accuracy and Test Accuracy vs epoch

The above graph shows the epochs vs accuracy output. Each epoch is a complete iteration of one train data and one test data. From the above graph it can be inferred that the range of accuracy of the data is approximately between 80% and 86%. We have compared our result with two baseline methods to compare the accuracy with one another. The two baseline methods that we have used to compare are Random Forest and Logistic Regression. When checked, the accuracy that we have achieved to classify hate speech with Random Forest method is 88.5% and that accuracy that we have achieved to classify hate speech with Logistic Regression method is 92%.

Below is a snapshot of the demo that we have implemented to check if the classifier is able to classify the tweet into hate speech.

Graphical user interface, text, application

Description automatically generated

As we can see from the above snapshot, we have passed a tweet/string (I hate you) which is definitely a hate speech. According to the classifier that we have designed, we must get an output between 0.75 to 1 (embedding technique). We achieved the output as 0.996 which clearly tells that the tweet is a hate speech tweet.

**CONCLUSION AND FUTURE WORK**

Hate Speech continues to persist in social media these days at a very high rate. Although all the social media platforms are trying to classify the hate speech from their tweets, they are not able to classify them properly and take further steps to stop the hate speech. We were able to successfully read the data, clean it, split the same into test and train data and pass this data to the respective methods that we have implemented. The classifier that we have built and implemented is able to classify hate speech and non-hate speech successfully. Using this classifier, the social media platform will be able to classify hate speech with increased accuracy. With few more additions to the classifier that we have built, there would be more increase in the accuracy. By modifying few of the hyper parameters that we have used, or by training on large data sets, or by implementing other feature engineering methods, we should be able to build a classifier with more accuracy. Therefore, with the help of this hate speech classifier, we should be able to stop or at least try to minimize the hate speech that has become a huge problem for everyone.

**REFERENCES**

[1.] *Ziqi Zhang, Lei Luo, Hate Speech Detection*: A Solved Problem? The Challenging Case of Long Tail on Twitter, October 25, 2018.

[2.] [https://www.justice.gov/hatecrimes/hate-crime-statistics#piechart-description](https://www.justice.gov/hatecrimes/hate-crime-statistics%23piechart-description)

[3.] <https://hal.archives-ouvertes.fr/hal-03101938/document>

[4.] <https://towardsdatascience.com/classifying-hate-speech-an-overview-d307356b9eba>

[5.] <https://medium.com/analytics-vidhya/applying-text-classification-using-logistic-regression-a-comparison-between-bow-and-tf-idf-1f1ed1b83640>

[6.] <https://www.youtube.com/watch?v=BO4g2DRvL6U&ab_channel=TensorFlow>