PROFILING HATE SPEECH SPREADERS ON TWITTER

A Project Report

Submitted for Minor Project - CS6490 of 6th Semester for the partial fulfillment of the requirement for the award of the degree of

Bachelors in Technology
in
Computer Science and Engineering

submitted by
Rakshita Jain (1806136)
Devanshi Goel (1806180)
Prashant Sahu (1806171)

Under the supervision of

Dr. Jyoti Prakash Singh
Head of Department
CSE Department
NIT Patna

Department of Computer Science and Engineering

National Institute of Technology Patna Patna-800005

Jan-June 2021

CONTENT

राष्ट्रीय प्रौद्योगिकी संस्थान पटना

NATIONAL INSTITUTE OF TECHNOLOGY PATNA

CERTIFICATE

This is to certify that Rakshita Jain with Roll No. 1806136, Devanshi Goel with Roll No. 1806180, Prashant Sahu with Roll No. 1806172 has carried out the Minor project (CS6490) entitled as " Profiling Hate Speech Spreaders on Twitter" during their 6th semester under the supervision of Dr. Jyoti Prakash Singh,Head of Department, CSE Department, in partial fulfillment of the requirements for the award of Bachelor of Technology degree in the department of Computer Science & Engineering, National Institute of Technology Patna.

……..

Dr. Jyoti Prakash Singh

Head of Department
CSE Department
NIT Patna

राष्ट्रीय प्रौद्योगिकी संस्थान पटना

NATIONAL INSTITUTE OF TECHNOLOGY PATNA

DECLARATION

We, the students of 6th semester, hereby declare that this project entitled " Profiling Hate Speech Spreaders on Twitter" has been carried out by us in the Department of Computer Science and Engineering of National Institute of Technology Patna under the guidance of Dr. Jyoti Prakash Singh, Head of Department of Computer Science and Engineering, NIT Patna. No part of this project has been submitted for the award of degree or diploma to any other Institute.

Name	Roll no.
Rakshita Jain	1806136
Devanshi Goel	1806180
Prashant Sahu	1806171

Place	Date
NIT Patna	26th May, 2021

राष्ट्रीय प्रौद्योगिकी संस्थान पटना

NATIONAL INSTITUTE OF TECHNOLOGY PATNA

ACKNOWLEDGEMENT

We would like to acknowledge and express our deepest gratitude to our supervisor, Dr. Jyoti Prakash Singh, for the valuable guidance, sympathy and co-operation for providing necessary facilities and sources during the entire period of this project.

We would also like to thank Mr. Abhinav Kumar, PDH scholar, NIT Patna, for providing mentorship, guiding us in every possible way and helping us to clear all the queries we had during the entire project. The faculties and cooperation received from the technical staff of the Department of Computer Science & Engineering is thankfully acknowledged.

1. Rakshita Jain
2. Devanshi Goel
3. Prashant Sahu

ABSTRACT

As much of the world now communicates on social media with nearly around 192 million daily active users on twitter alone. As more and more people have moved online , experts say , individuals inclined towards racism , misogyny or homophobia have found niches that can reinforce their views and goad them to violence.Not only this it also causes psychological harm to its victims and physical harm when it incites violence.Hate speech poses a challenge for modern liberal societies which are committed to both freedom of expression and social equality. When aimed at historically oppressed minorities, hate speech is not only insulting but also perpetuates their oppression by causing the victims , the perpetrators and the society at large to internalize the hateful messages and act accordingly.Typical hate speech involves epithets and slurs , statements that promote malicious stereotypes, and speech intended to incite hatred or violence against a group.Thus there is an ongoing debate in those societies over whether and how hate speech should be regulated or censored.

It's high time that proper steps must be taken to curb this issue and one major step can be to identify people who are spreading hate speech by their hate spreading tweets on twitter. The importance of hate speech detection research cannot be overemphasised. It would be counter productive if all research efforts are not focussed and channelled towards a better tomorrow by building on top one another.

The majority of research paper has focussed on several aspect of author profiling on social media like detecting that whether a tweet is spreading hate speech or not , fake news spreaders , bot detection etc however we aim at identifying possible hate speech spreaders on twitter as first step towards preventing hate speech from being propagated among online users.

We have tried to perform the above task for two different languages english and spanish on the two dataset provided by PAN @CLEF 2021 .Initially it included the tweet id and the tweet in xml form , we converted that in csv then combined it with the target label provided in separate csv file . After that we grouped the tweets of the same id together.Performed some preprocessing on them like removing hashtags , converting emoticons to words and other text cleaning and preprocessing . For feature extraction we used count vectorizer , tf idf vectorizer,word embedding and one hot encoding in case of lstm.

We performed the above task using various machine learning models like multinomial naive bayes , Kneighbors classifier , logistic regression , linear svm and deep learning models like lstm , bilstm and bert model.We trained and tested models separately for tf idf and count vectorizer and also for different ngram ranges and out of all the above mentioned models multinomial naive bayes performed best with an accuracy of 74% for english dataset and 82% for spanish dataset.

1.INTRODUCTION

Due to the excessive use of social media platforms by people belonging to different cultures and backgrounds, toxic online content has become a major issue in today's time. The advent of social media has given rise to an unprecedented level of hate speech in public discourse. More tweets involving hate appear every year. Unfortunately, any user engaged on these platforms will have a risk of being targetted or harassed via abusing language, expressing hate towards race, colour, religion, descent, gender, antion, etc.

Hate Speech is a crime that has been growing in the recent years, and the rapidly growing availability of the online platforms and rise of social media, has led users to publish and share any content, tell their views, show their liking or hatred towards people, community, race, non-living objects, etc. in an ever growing fast way. The increased willingness of people to express their opinions online have contributed to propagation of hate speech as well. The ease of getting access to these platforms and publishing content with minimal efforts have led to an increase in the hate speech about every small thing that people criticize or do not like, influencing other people's mind and causing several negative consequences in society. On internet and social network platforms people are more likely to adopt aggressive behaviour because of the anonymity provided by these environments. Since this type of prejudice can cause extreme harm to the society, government, and social network platforms can be benefited from hate speech detection and prevention tools.

Understanding whether a tweet is hate speech or not and hence finding out whether the author is a hate speech spreader or not, is a very challenging task for the users, who in their majority are not experts. Additionally, a hate speech can also be present in the form of a sarcasm or indirect taunt, making it confusing for users to actually understand the intent behind the tweet.

Previous work

Our work is based on an assumption that an author can be classified as a hate speech spreader if while analysing a certain number of tweets of that author, we find that the majority of the tweets can be classified as hate speech content. The final goal is profiling those authors who spread hate speech depending on the number of tweets that contain hateful content that they spread, for two languages - English and spanish. This will allow for identifying hate speech spreaders on Twitter as a first step towards preventing hate speech being propagated among social media users and preventing it from influencing lives and work or target people.
We will focus on classifying authors as hate speech spreaders or not hate speech spreaders (binary classification). Examples of each of these categories - taken from author's tweet dataset(PAN21-Profiling-Hate-Speech-Spreaders-in-Twitter) is illustrated below:

- POC love talking about police brutality but noone talks about black on white crime. (hateful)
- "Hey Jamal (snickering uncontrollable) You want some (PFFF) LEMONADE!" What an IDIOT! (hateful)
- Romanian graftbuster’s firing violated rights, European court says #URL#Russian ventilators sent to U.S.(not hateful)
- #RT #USER#: "At least while Biden is bombing brown people, he\'s not being offensive on Twitter." (not hateful)

Based On messages of this kind of tweets, the present work will investigate whether an author is a hate speech spreader or not using various classification and deep learning models to compare the results of different models and determine which performs best for this task for Spanish as well as English language.

2. RELATED WORK

2.1 SVM with RBF Kernel[1]

SemEval -2019 task was about Detection of Hate speech against Immigrants and Women in english and spanish messages extracted from twitter. The task included two subtasks.The first one was about identifying the hate speech and the later one was about identifying further features such as aggressiveness and the target. For the subtask A in english the best result was obtained by Fermi Team with macro-average F1-score 0.65.They trained the SVM model with RBF kernel using embeddings from Google's Universal Sentence Encoder as features.

2.2 SVM with the combination of character n-grams and word n-grams and Ensemble Logistic Regression[2][6][7]

The main motive behind this task was to identify the author who is spreading the fake news not to identify the message that it is a fake news or not.From the evaluation of the approaches of the participants ,it has been found that SVM with the combination of character n-grams and word n-grams is the best suited approach for spanish and logistic regression ensemble of five submodels: n-grams with Random Forest,n-grams with SVM,n-grams with Logistic Regression,n-grams with XGBoost and XGBoost with features based on textual descriptive statistics ,is the best suited approach for english.The best accuracy obtained for english was 75% and for spanish was 82%.

2.3 Data-Driven and Psycholinguistics-Motivated Approaches to Hate Speech Detection[3]

In this paper ,the authors have investigated multiple approaches for the problem of hate speech,aggressive behaviour and target group recognition.They have presented different learning models including Logistic regression,Convolutional Neural Network(CNN),Deep Bidirectional Transformers(BERT) using word n-grams, character n-grams,word embedding and psycholinguistic features(LIWC).Among these models,a purely Data-Driven BERT model and to some extent hybrid psycho linguistically informed CNN outperformed all other models for all tasks in both languages english and spanish. For english ,the best F1-score(0.60) for hate speech has been found by CNN using features word embedding and LIWC.For spanish, the best F1-score(0.720 for hate speech has been found by BERT using features cased word.

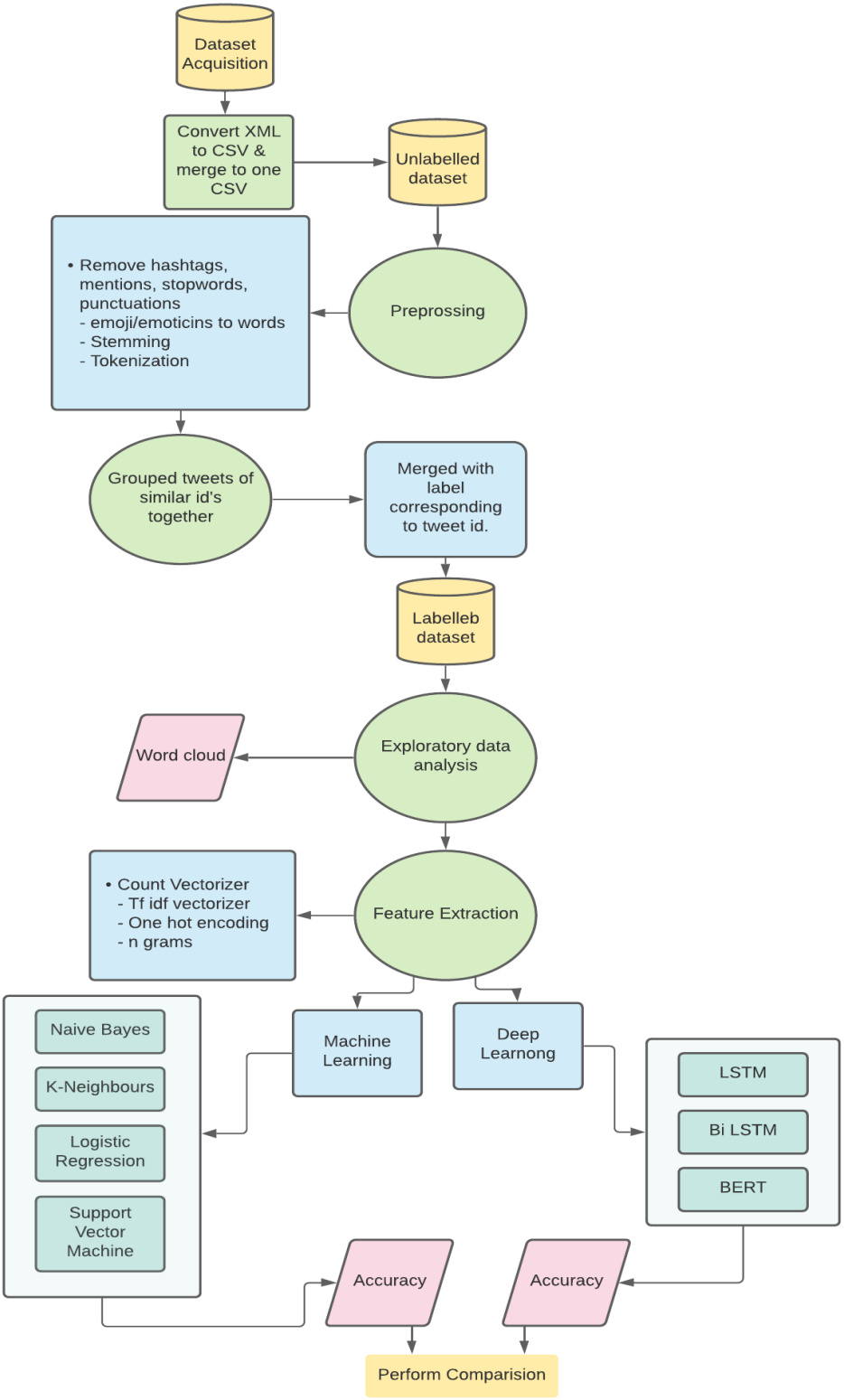
2.4 Linear SVM using Embedding[4]

At EVALITA-2018 ,Team RuG developed the model in the context of detecting hate speech in Italian Social Media like twitter and facebook. The best macro F1-score in all subtasks was obtained by linear SVM using hate-rich embedding.For twitter and facebook the best macro F1-score was 0.79 and 0.77.

2.5 Convolution-GRU Based Deep Neural Network[5]

This paper introduces a new method based on deep learning combining Convolution and Gated Recurrent Networks.Paper claims that this method has outperformed previously proposed methods for many of the twitter dataset by range of F1-score between 1 and 13.

3. METHODOLOGY



The different classification and deep learning models used for profiling hate speech spreaders learns the continuous representation of tweets and then picks features from them extracted using count vectorizer and tf idf vectorizer . Their accuracies were compared for different ngram range . In deep learning models like lstm we have used one hot encoding for feature extraction. The detailed architecture and flow of different phases in which the computation is carried out is shown in the above figure.

3.1 Data collection, preprocessing and labelling

3.1.1 Data

We have used the PAN21-Profiling-Hate-Speech-Spreaders-on-Twitter data provided by zenodo. The data contained author ids and their tweets. The data contains tweets of 200 authors each for English and spanish language. 100 tweets are provided for each author containing a combination of hate speech tweets and non hate speech tweets. Therefore, a total of 20,000 tweets.
Label information for each author is provided in a separate file classifying authors into two classes - hate speech spreader or not hate speech spreader.
Being originally a part of PAN at CLEF 2021, the data contains only the training dataset. To test and compare the performance of various classification and deep learning models, we have splitted this dataset into training and testing dataset in ration 67:33. Finally, our training dataset contains 13,400 tweets (i.e 134 authors) and the testing set contains 6,600 authors(i.e. 66 authors) for each language.

3.1.2 Preprocessing

We preprocessed the tweets to remove hashtag symbols keeping the content of the hashtag as it can be used to identify important details like the target people, emotions, intent behind the tweet. We then removed mentions and converted emoticons and emojis to text . Tweets were converted to lowercase. Punctuations and stop words were removed. Then to remove affixes from words, stemming was performed. Then finally tokenization of tweets was done.

3.1.3 Labelling

After preprocessing was completed, we merged all the tweets of a particular author into one tweet by space . Then we merged the labels with the tweets data on the basis of author id. Finally we obtained the data containing author id, combined tweets per author and label indicating whether the author is a hate speech spreader or not.

3.2 Feature extraction from tweets using tf idf vectorizer and count vectorizer

In order to use textual data for predictive modelling , the text must be parsed to remove certain words - this process is called tokenization. These words needed to be encoded as integers or floating point values , for use as input in machine learning algorithms. This process is called feature extraction or vectorization. We can form two different kind of vectors by performing feature extraction :

3.2.1 Tfidf Vectorizer

TFIDF is an abbreviation for term frequency inverse document frequency . This is a very common algorithm to transfer text into a meaningful representation of numbers which is used to fit a machine algorithm for prediction.It evaluates how relevant a word is to a document in a collection of documents . This is done by multiplying two metrics : how many times a word appears in a document and the inverse document frequency of the word across a set of documents.

TFIDF

For a term i in document j:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

3.2.2 Count Vectorizer

It is used to convert a collection of text documents to a vector of term or token counts . It also enables the preprocessing of text data prior to generating the vector representation . This functionality makes it a highly flexible feature representation module for text .

The difference between the two is that the count vectorizer gives a number of frequencies with respect to indexes of vocabulary whereas tfidf considers the overall document of weight of words.

We trained our models using both count vectorizer and tf idf vectorizer and then compared the accuracy using both .

3.3 Classification

3.3.1 Different machine learning models used for classification:

3.3.1.1 Multinomial Naive Bayes :

Multinomial Naive Bayes uses term frequency i.e. the number of times a given term appears in a document . Term frequency is often normalised by dividing the raw term frequency by the document length . After normalization term frequency can be used to compute maximum likelihood estimates based on the training data to estimate conditional probability.

3.3.1.2 KNeighborsClassifier :

The K in the name of this classifier represents the k nearest neighbors , where k is an integer value specified by the user. Hence as the name suggests this classifier implements learning based on the k nearest neighbors . The choice of the value of k is dependent on data.

3.3.1.3 Logistic Regression :

It is a statistical model that in its basic form uses a logistic function to model a binary dependent variable , although many more complex extensions exist . In regression analysis logistic regression is estimating the parameters of a logistic model.It is used to examine the association of (categorical or continuous) in our case categorical independent variable with one dichotomous dependent variable . It is like finding conditional probability of Y=1 given X or conditional probability of Y=0 given X. It is using its conditional probability to find whether the combined tweets of a tweet id is spreading hate speech or not.

3.3.1.4 Linear SVM SVC :

SVM are powerful yet flexible supervised machine learning methods used for classification , regression , and outlier's detection . SVM are very efficient in high dimensional spaces and generally are used in classification problems. SVM are popular and memory efficient because they use a subset of training points in the decision function .

3.3.2 Different deep learning models used for classification:

3.3.2.1 LSTM :

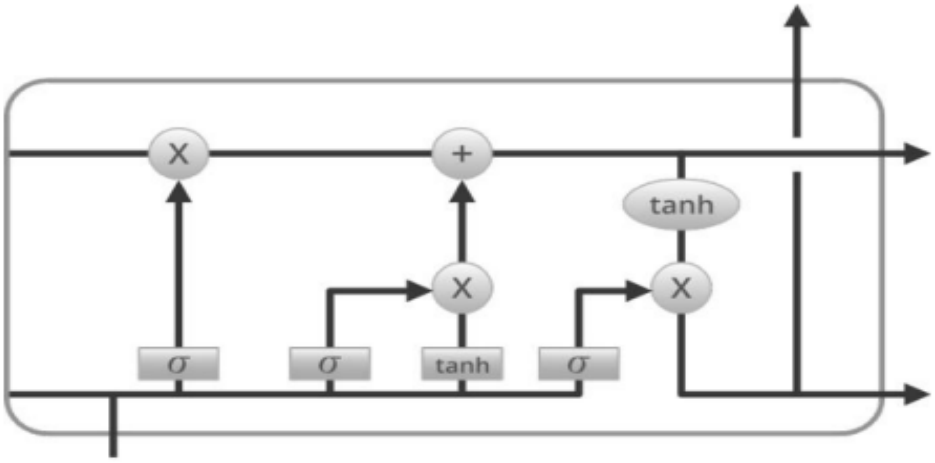
It stands for (Long Short -Term Memory) designed by Hochreiter & Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give efficient performance . LSTM can by default retain the information for a long period of time. It is used for processing , predicting , and classifying on the basis of time series data.

The success of LSTMs is in their claim to be one of the first implements to overcome the technical problems and deliver on the promise of recurrent neural networks. The two technical problems overcomed by LSTM are vanishing gradient and exploding gradient , both related to how the network is trained. The key to the LSTM solution to the technical problems was the specific internal structure of the units used in the model.

Structure Of LSTM :

LSTM has chain structure that contains four neural networks and different memory blocks called cells. Information is retained by the cells and the memory manipulation are done by the gates.

1. Forget Gate
2. Input Gate
3. Output Gate



Steps we followed while training our LSTM :

1. First we initialized a vocabulary size of 5000 which we will be using while doing the one hot representation of the tweets.
2. Using the above vocabulary it will replace each word in the tweet with its corresponding index or the index of another word having similar meaning in the vocabulary.
3. Then we passed it through the padding sequence keeping the sentence length as 2500 and padding sequence as ‘pre’ so that all our sentences are of a fixed length.
4. Now we will define our model.

Architecture of the model used :

First we defined the number of vector features , we took it as 40.
The layers :

1. **Sequential Layer** .
2. **Embedding Layer** : Here we pass the vocabulary size as our first parameter , input feature size as the second parameter and the sentence length as the third parameter which in our case is 2500. This layer will give an output which we will pass through an LSTM layer
3. **LSTM Layer** : We have used 1 lstm layer having 100 neurons.
4. **Dense Layer** : Since it is a classification problem, we will get an output from this dense layer.

Description Of Hyper Parameters :

Activation function	Relu
Loss function	Binary Cross Entropy
Optimiser	Adam
Vocabulary Size	5000
Embedding Vector Feature	40
Sentence Length	2500

Epochs	15
Batch Size	64
Validation Split	0.26
Metrics	Accuracy

It performed well for spanish dataset with an accuracy of 82.3%.

3.3.2.2 BiLSTM :

It stands for bidirectional LSTM . They are extension of traditional LSTM that can improve model performance on sequence classification problems . In problems where all timesteps of input sequence are available . BILSTM train two LSTM instead of one LSTM on input sequence. Since we have used both LSTM and BILSTM in our task to detect hate speech spreaders and the results show that since in BILSTM there is additional training of data and thus BILSTM based modelling have provided better prediction as compared to LSTM model.

3.3.2.3 BERT :

BERT(Bidirectional Encoder Representation From Transformers) is a breakthrough in the field of Machine Learning for Natural Language Processing. It is introduced in 2018 by the researchers of Google AI Language.BERT is different from previous efforts which looked at the text sequence either from the left to righ or combined from right to left and left to right.The paper[8] showed that bidirectionally trained model

4. RESULTS

4.1 Evaluation Metrics :

To evaluate the proposed models we used precision , recall , F1 Score , support and accuracy.These metrics are widely used for evaluating supervised machine learning models for classification in case of multi labelled dataset. Say a multi-labeled dataset consist of N instances each instance Ni can be represented as (xi,yi) , where xi is the set of attributes and yi is the set of labels . Suppose yi and yi' represent the true and predicted label respectively for ith instance then the metrics can be described for the ith instance by the given formulae.

4.1.1 Precision :

This is the number of accurately predicted location words to the total number of predicted location words. It is computed as given in Eq(1). The range of precision varies between 0 and 1 , where 1 is the best and 0 is the worst value.

$$\text{Precision} = \frac{\text{Number of accurately predicted location words}}{\text{Total number of predicted location words}} = \frac{|yi \cap yi'|}{|yi'|}$$

4.1.2 Recall :

This is the number of accurately predicted location words to the total number of actual location words in the tweet. It is computed as is given in Eq(2). The range of recall varies between 0 and 1 , where 1 is the best and 0 is the worst value .

$$\text{Recall} = \frac{\text{Number of accurately predicted location words}}{\text{Total number of actual location words}} = \frac{|yi \cap yi'|}{|yi|}$$

4.1.3 F1-Score :

This is the harmonic mean between Precision and Recall , which gives the balanced equation between them . It can be represented by Eq (3) . The range of F1-score varies between 0 and 1 , where 1 is the best and 0 is the worst value.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Result Table for Different Classifiers For English & Spanish

Using Different Feature & NGram Ranges

For english dataset

English				
N-grams range - (1, 1)				
Naive Bayes - Count vectorizer				
	Precision	Recall	F1 - Score	Support
0	0.79	0.72	0.75	36
1	0.7	0.77	0.73	30
weighted avg	0.75	0.74	0.74	66
Accuracy	0.742			
Alpha	0.2			
Naive Bayes - TF-IDF				
	Precision	Recall	F1 - Score	Support
0	0.72	0.72	0.72	36
1	0.67	0.67	0.67	30
weighted avg	0.7	0.7	0.7	66
Accuracy	0.696			
Alpha	0.1			

English				
N-grams range - (1, 3)				
Naive Bayes - Count vectorizer				
	Precision	Recall	F1 - Score	Support
0	0.76	0.53	0.62	36
1	0.59	0.8	0.68	30
weighted avg	0.68	0.65	0.65	66
Accuracy	0.65			
Alpha	0			
Naive Bayes - TF-IDF				
	Precision	Recall	F1 - Score	Support
0	0.72	0.58	0.65	36
1	0.59	0.73	0.66	30
weighted avg	0.67	0.65	0.65	66
Accuracy	0.65			
Alpha	0			

K Neighbours - Count Vectorizer				
	Precision	Recall	F1 - Score	Support
0	0.55	0.72	0.63	36
1	0.47	0.3	0.37	30
weighted avg	0.52	0.53	0.51	66
Accuracy	0.53			

K Neighbours - Count Vectorizer				
	Precision	Recall	F1 - Score	Support
0	0.62	0.58	0.6	36
1	0.53	0.57	0.55	30
weighted avg	0.58	0.58	0.58	66
Accuracy	0.575			

N	2				**N**	9			
K Neighbours - TF-IDF					**K Neighbours - TF-IDF**				
	Precision	**Recall**	**F1 - Score**	**Support**		**Precision**	**Recall**	**F1 - Score**	**Support**
0	0.73	0.61	0.67	36	**0**	0.67	0.67	0.67	36
1	0.61	0.73	0.67	30	**1**	0.6	0.6	0.6	30
weighted avg	0.68	0.67	0.67	66	**weighted avg**	0.64	0.64	0.64	66
Accuracy	0.66				**Accuracy**	0.636			
N	6				**N**	6			

Logistic Regression - Count vect					**Logistic Regression - Count vect**				
	Precision	**Recall**	**F1 - Score**	**Support**		**Precision**	**Recall**	**F1 - Score**	**Support**
0	0.67	0.61	0.64	36	**0**	0.76	0.61	0.68	36
1	0.58	0.63	0.6	30	**1**	0.62	0.77	0.69	30
weighted avg	0.63	0.62	0.62	66	**weighted avg**	0.7	0.68	0.68	66
Accuracy	0.6212				**Accuracy**	0.6818			
Logistic Regression - TF-IDF					**Logistic Regression - TF-IDF**				
	Precision	Recall	F1 - Score	Support		**Precision**	**Recall**	**F1 - Score**	**Support**
0	0.75	0.5	0.6	36	**0**	0.88	0.39	0.54	36
1	0.57	0.8	0.67	30	**1**	0.56	0.93	0.7	30
weighted avg	0.67	0.64	0.63	66	**weighted avg**	0.73	0.64	0.61	66
Accuracy	0.636				**Accuracy**	0.636			

SVC SVM - TF-IDF					**Linear SVC SVM - TF IDF**				
	Precision	**Recall**	**F1 - Score**	**Support**		**Precision**	**Recall**	**F1 - Score**	**Support**

0	0.75	0.5	0.6	36
1	0.57	0.8	0.67	30
weighted avg	0.67	0.64	0.63	66
Accuracy	0.636			

0	0.88	0.39	0.54	36
1	0.56	0.93	0.7	30
weighted avg	0.73	0.64	0.61	66
Accuracy	0.636			

Bi LSTM	
Accuracy	85.7 %
Epochs	15

Bi LSTM	
Accuracy	82.93 %
Epochs	15

For spanish dataset

Spanish				
using N-grams range - (1, 1)				
Naive Bayes - Count Vectoriser				
	Precision	**Recall**	**F1 - Score**	**Support**
0	0.78	0.78	0.78	32
1	0.79	0.79	0.79	34
weighted avg	0.79	0.79	0.79	66
Accuracy	0.787			
Alpha	0.2			
Naive Bayes - TF-IDF				
	Precision	**Recall**	**F1 - Score**	**Support**
0	0.79	0.81	0.8	32
1	0.82	0.79	0.81	34
weighted avg	0.8	0.8	0.8	66
Accuracy	0.8			
Alpha	0.1			

Spanish				
using N-grams range - (1, 3)				
Naive Bayes - Count Vectoriser				
	Precision	**Recall**	**F1 - Score**	**Support**
0	0.86	0.75	0.8	32
1	0.79	0.88	0.83	34
weighted avg	0.82	0.82	0.82	66
Accuracy	0.81			
Alpha	0.1			
Naive Bayes - TF-IDF				
	Precision	**Recall**	**F1 - Score**	**Support**
0	0.86	0.75	0.8	32
1	0.79	0.88	0.83	34
weighted avg	0.82	0.82	0.82	66
Accuracy	0.82			
Alpha	0			

K Neighbours - Count Vectorizer				
	Precision	Recall	F1 - Score	Support
0	0.74	0.81	0.78	32
1	0.81	0.74	0.77	34
weighted avg	0.78	0.77	0.77	66
Accuracy	0.77			
N	6			

K Neighbours - Count Vectorizer				
	Precision	Recall	F1 - Score	Support
0	0.71	0.78	0.75	32
1	0.77	0.71	0.74	34
weighted avg	0.75	0.74	0.74	66
Accuracy	0.74			
N	7			

K Neighbours - using TF-IDF				
	Precision	Recall	F1 - Score	Support
0	0.75	0.56	0.64	32
1	0.67	0.82	0.74	34
weighted avg	0.71	0.7	0.69	66
Accuracy	0.69			
N	2			

K Neighbours - using TF-IDF				
	Precision	Recall	F1 - Score	Support
0	0.8	0.25	0.38	32
1	0.57	0.94	0.71	34
weighted avg	0.68	0.61	0.55	66
Accuracy	0.6			
N	6			

Logistic Regression - Count vect				
	Precision	Recall	F1 - Score	Support
0	0.77	0.84	0.81	32
1	0.84	0.76	0.8	34
weighted avg	0.81	0.8	0.8	66
Accuracy	0.8			
Logistic Regression - TF-IDF				
	Precision	Recall	F1 - Score	Support
0	0.79	0.72	0.75	32

Logistic Regression - Count vect				
	Precision	Recall	F1 - Score	Support
0	0.76	0.81	0.79	32
1	0.81	0.76	0.79	34
weighted avg	0.79	0.79	0.79	66
Accuracy	0.79			
Logistic Regression - TF-IDF				
	Precision	Recall	F1 - Score	Support
0	0.84	0.66	0.74	32

1	0.76	0.82	0.79	34	**1**	0.73	0.88	0.8	34
weighted avg	0.77	0.77	0.77	66	**weighted avg**	0.78	0.77	0.77	66
Accuracy	0.77				**Accuracy**	0.77			

Linear SVC SVM - Count vectoriser					**Linear SVC SVM - Count vectoriser**				
	Precision	**Recall**	**F1 - Score**	**Support**		**Precision**	**Recall**	**F1 - Score**	**Support**
0	1	0.06	0.12	32	**0**	1	0.06	0.12	32
1	0.53	1	0.69	34	**1**	0.53	1	0.69	34
weighted avg	0.76	0.55	0.41	66	**weighted avg**	0.76	0.55	0.41	66
Accuracy	0.54				**Accuracy**	0.54			
Linear SVC SVM - TF-IDF					**Linear SVC SVM - TF-IDF**				
	Precision	**Recall**	**F1 - Score**	**Support**		**Precision**	**Recall**	**F1 - Score**	**Support**
0	0.79	0.72	0.75	32	**0**	0.84	0.66	0.74	32
1	0.76	0.82	0.79	34	**1**	0.73	0.88	0.8	34
weighted avg	0.77	0.77	0.77	66	**weighted avg**	0.78	0.77	0.77	66
Accuracy	0.77				**Accuracy**	0.77			

Bi LSTM		**Bi LSTM**	
Accuracy	59 %	**Accuracy**	82.93 %
Epochs	15	**Epochs**	15
LSTM		**LSTM**	
Accuracy	82.93 %	**Accuracy**	82.93 %
Epochs	15	**Epochs**	15
BERT		**BERT**	
Accuracy	82.93 %	**Accuracy**	82.93 %
Epochs	15	**Epochs**	15

5.CONCLUSION

The prediction of whether an author is spreading hate speech or not from his combined tweets was a challenging task as tweets have various noise in terms of grammatical mistakes, spelling mistakes, and non standard abbreviations.Alongwith that when the different tweets of a single author were merged together the sentiments of a particular tweet might counter the effect of other, this problem was also encountered. We trained classification models using tf idf and count vectorizer as feature values . We have shown a comparative study of machine learning algorithms with respective feature sets. We have compared their accuracies for different ngram range i.e (1,1) and (1,3) and also for tf idf and count vectorizer. We have shown the accuracy estimated in each case in the result section.

We achieved our best result with an F1-score of 0.74 for english dataset when we used multinomial naive bayes with ngram range (1,1) and count vectorizer and of 0.82 for spanish dataset again for multinomial naive bayes with ngram range (1,3) and for both tf idf and count vectorizer.

This system can be utilized by different social media platforms to identify hate speech spreaders and remove such hate speech spreaders from their platform. As for now we have developed the system for english and spanish language. We can extend it to use it for other languages as well by changing the stopwords used while preprocessing.

6. References

[1] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Dora Nozza, Viviana Patti, Francisco Rangel, Paolo Rosso, Manuela Sanguinetti (2019). SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. Proc. SemEval 2019

[2] Francisco Rangel, Anastasia Giachanou, Bilal Ghanem, Paolo Rosso. Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter. In: L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings.CEUR-WS.org, vol. 2696

[3] Samuel Caetano da Silva, Thiago Castro Ferreira, Ricelli Moreira Silva Ramos, Ivandre Paraboni (2020). Data-driven and psycholinguistics motivated approaches to hate speech detection. Computación y Sistemas, 24(3): 1179–1188

[4] Cristina Bosco, Felice Dell'Orletta, Fabio Poletto, Manuela Sanguinetti, Maurizio Tesconi (2018). Overview of the EVALITA 2018 hate speech detection task. Proc. EVALITA 2018

[5]Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting hate speech on twitter using a convolution GRU based deep neural network. The Semantic Web, Springer International Publishing, Cham, pp. 745–760

[6]Pizarro, J.: Using N-grams to detect Fake News Spreaders on Twitter. In: Cappel-lato, L., Eickhoff, C., Ferro, N., N´ev´eol, A. (eds.) CLEF 2020 Labs and Workshops,Notebook Papers. CEUR-WS.org (Sep 2020)

[7]Buda, J., Bolonyai, F.: An Ensemble Model Using N-grams and Statistical Featuresto Identify Fake News Spreaders on Twitter. In: Cappellato, L., Eickhoff, C., Ferro,N., N´ev´eol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2020