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. 1806171 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, PHD 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 today is the era of social media with nearly around 192 million daily active users on twitter alone. With increase in the number of people online , individuals inclined towards racism , misogyny have found instances that can reinforce their views and lead them to violence.Not only this it also causes harm psychologically to its victims and physically when it incites violence.

Hate speech acts as a challenge for modern liberal societies which are committed to freedom of expression and social equality. Hate speech is not only insulting but also perpetuates the oppression of historically oppressed minorities by causing the victims and the society to internalize the hateful messages and act according to it.Typical hate spreading speech involves harsh words and slurs , they promote malicious stereotypical thoughts, and it is intended to incite hatred or violence against a group.Thus there is an ongoing debate 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 hate speech spreaders on twitter as first step towards preventing hate speech from being propagated on social media platforms.

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 emergence of social media platforms have given rise to an unparalleled level of hate speech in public conversations. The number of tweets containing hate speech and targeting one or other user is appearing 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 no less than a felony that has been continuously and abruptly 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 demonstrate their opinions publicly have contributed to multiplication of hate speech also. 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 take on inappropriate or violent behaviour due to the anonymity provided by these environments. Since this type of prejudice can cause extreme harm to the society, government, and social network platforms such as twitter and facebook 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 tough task for the users, especially those who 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.

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.For that we have grouped together all the tweets of the same id together. Our ultimate target is profiling those authors who spread hate speech based on the number of tweets containing any hateful content that they spread, for two languages - English and spanish. This will allow the social media platforms to identify hate speech spreaders on Twitter as an initial step towards preventing hate speech from spreading among social media users and preventing it from influencing the lives and work of 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.

While evaluating different models we have compared their accuracies on different basis like count vectorizer , tf idf vectorizer and on the basis of ngram range. We trained and predicted the accuracies using different ngram ranges and then compared for which range it gave better result . Similarly we compared whether a model was performing better with count vectorizer or tf idf vectorizer.

Here is a snapshot of the word cloud of a merged tweet from english and spanish dataset which depicts the term with the highest frequency in the tweet .

trump
obama
donald trump
president
inauguration
meryl
twitterorder
obamacare
women
melania
immigration
speech day
look
state
watch
daughter
exercises
things
amp
ways
guide
red
gifts
youre
never
christmas
hair
home
shares
kids
tips
ideas

qu
progr
solo
mierda
puta
ms
terrorista
mejor
libertad
farsa
toda
amigo
acuerdo
gobierno
fun
si
jode
pued
madrid
espero
ms
gent
gracia
mucha
slo
si
nimo
bien
est
estn
hospit
incendio
hashtag
covid

2. RELATED WORK

2.1 SVM with RBF Kernel[1]

SemEval -2019 task was about Detection of Hate speech against Immigrants and Women. In this task the participant has been given tweets of two languages english and spanish.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 first task the best result was with macro F1-score 0.65. This was obtained by the SVM 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 in [2] was to identify the author who is spreading the fake news, not to identify the message that it is 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 Models to Detect Hate Speech[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 many 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, 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 regarding detecting hate speech in Italian Social Media like twitter and facebook. Linear SVM using embedding as features had performed best for the given problem.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.

2.6 An N-gram and TF IDF based Machine Learning Approach for Detecting Hate Speech and Offensive Language on Twitter[9]

The problem addressed in this paper is about identifying hate speech and aggressive tweets.The dataset used in this paper is a combination of publicly available three datasets.The author used the TF-IDF vectors with different n-gram range as features.The author used the three model-Logistic Regression,Naive Bayes Classifier and SVM.Among these models Logistic Regression fed with TF-IDF vector with n-gram range (1,3) has given the best accuracy.The hyperparameter for logistic regression was - solver=liblinear,C=100.

Model-TF-IDF+Logistic Regression
Accuracy - 0.956
Year of publication - 2018

2.7 Multilingual Hate Speech Detection using Deep Learning Models[10]

In this paper , The authors have implemented deep learning models in sixteen different datasets of nine different languages. They found that for small dataset Logistic Regression fed with LASER (Language-Agnostic SEntence Representation)embedding has performed best and for the larger dataset BERT based model has given better results. In this paper the author has used features of LASER embedding and MUSE embedding.

Model - mBERT
Accuracy - 0.832
Year of publication- 2020

2.8. Hateminers : Approach to detect hate speech against women[11]

The problem addressed in this paper is a shared task at EVALITA 2018. The author has extracted the three features-TF-IDF vectors, sentence embedding and Bag Of Words Embedding. All the three features have been concatenated and fed into the different machine learning models. Among all the models, Logistic Regression has given the best accuracy with hyperparameter C = 1.0. Two subtasks were performed in this paper - first to classify whether a text is hate speech or not and, second is Category and Target Classification.

Model - Logistic Regression
Accuracy - 0.704
Year of publication - 2018

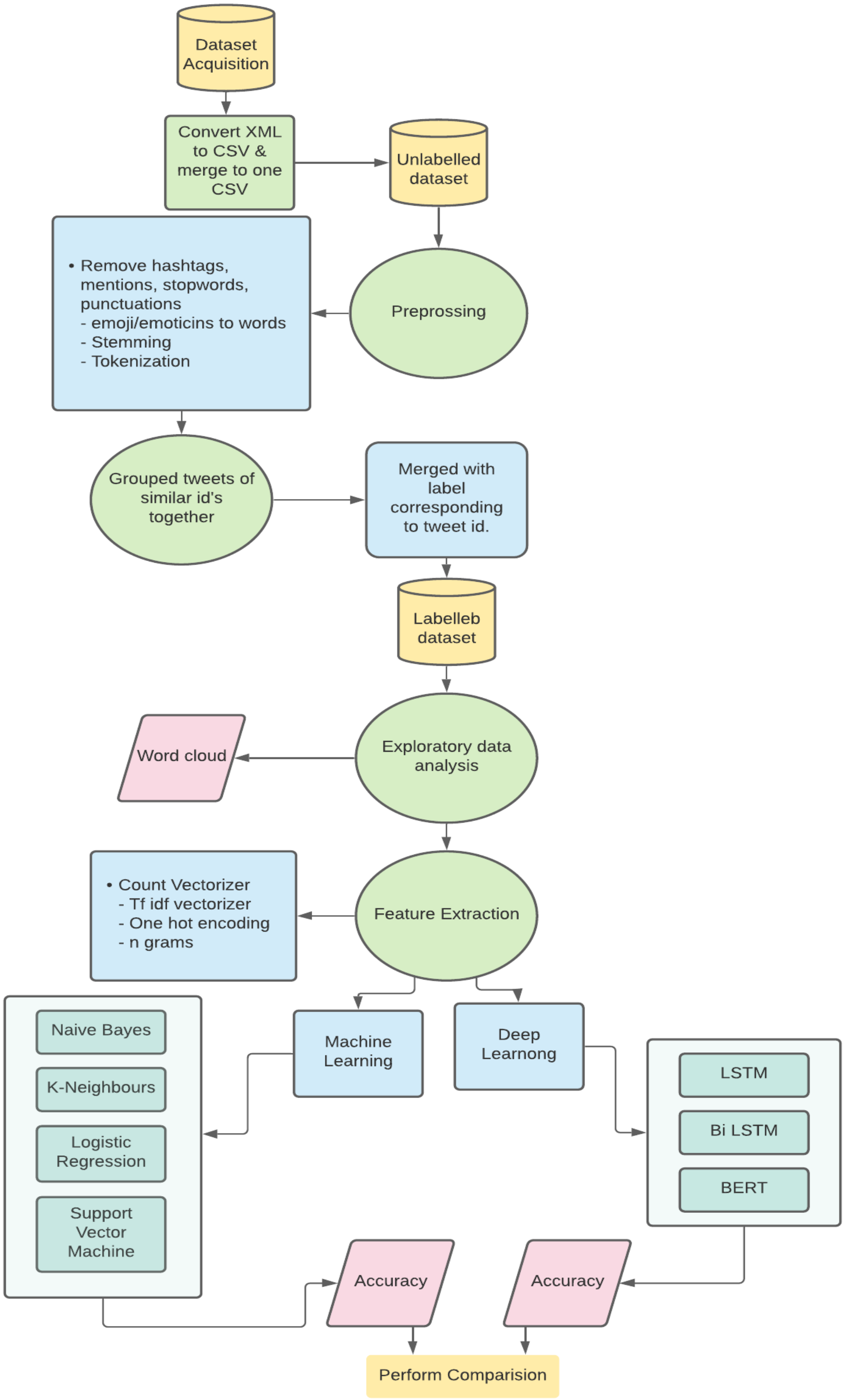
2.9 Transformer Method to Detect hate Speech in Twitter[12]

In this paper, authors claim that the performance of various machine learning algorithms to Detect hate speech is hampered by inefficient sequence transduction and the vanilla recurrent neural networks and Recurrent Neural Networks with attention also suffer from various problems such as lack of parallelization and long term dependency. Therefore, the authors proposed a transform based model and used a public dataset containing 24,783 labelled tweets. The proposed DistillBERT transformer method was compared against other transformer baselines and recurrent neural networks for Hate Speech Detection in Twitter and results showed that DistillBERT transformers outperformed other models with an accuracy of 75%.

2.10 Recurrent Neural Network approach to detect effective Hate Speech in Twitter[13]

The problem addressed in this paper is about recognizing hateful content in social media. Recurrent Neural Networks were ensembled and various user-related features were incorporated showing the users tendency towards hate speech such as racism or sexism. Word frequency vectors along with these features and data were fed as input to the classifiers. The dataset used by them was a corpus of 16,000 tweets that is available publicly. The results were compared to existing state-of-art solutions. The model can successfully differentiate racism and sexism messages from the one’s which do not fall in these categories. Finally, the highest F1-score of 0.9320 was achieved using the ensemble approach.

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 ranges . 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

Tokenization is done which is the process in which text is parsed to remove certain words to use that data for predictive modelling. These words are needed to be encoded as integers or floating point values then those encoded values are used as input . This 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 . It is a very common algorithm for transferring text into a meaningful representation of numbers which can be used to fit a machine algorithm for prediction.It evaluates the relevance of a word to a document in a collection of documents . This is done by multiplying two metrics one is how many times a word appears in a document and the other is 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 .

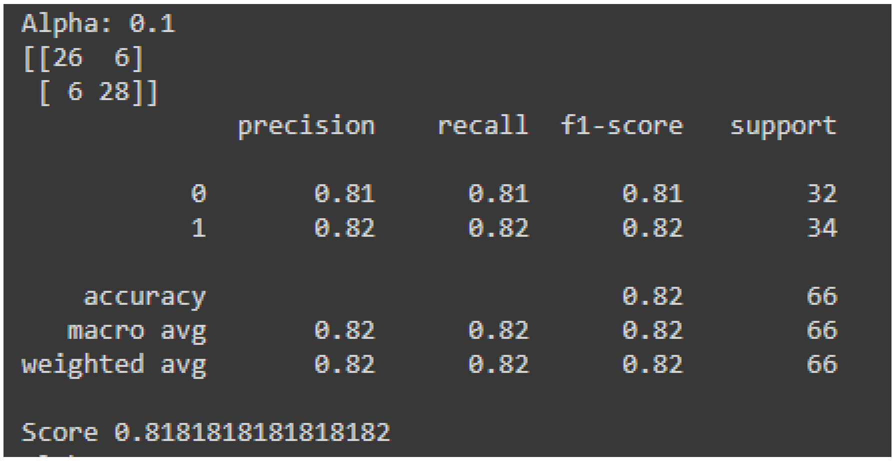
3.3 Classification

3.3.1 Different machine learning models used for classification:

3.3.1.1 Multinomial Naive Bayes :

Multinomial Naive Bayes uses frequency of each term i.e. what is the number of times a term appears in a document . Frequency of a term is often normalised by dividing the raw term frequency by the length of the document . After performing normalization on the frequency of the term it can be used to compute maximum likelihood estimates based on the training data to estimate conditional probability for predicting the output .We evaluated the classifier for different values of alpha in the range of 0 to 1.

Out of all the different classification methods naive bayes performed best with an accuracy of 74% for ngram range (1,1) and count vectorizer while 65% for ngram range (1,3).For other criteria like for tfidf vectorizer with ngram (1,1) it gave 69.6% accuracy and with ngram (1,3) it gave 65% accuracy.



```
Alpha: 0.1
[[26  6]
 [ 6 28]]
              precision    recall  f1-score   support

           0       0.81      0.81      0.81        32
           1       0.82      0.82      0.82        34

    accuracy                           0.82        66
   macro avg       0.82      0.82      0.82        66
weighted avg       0.82      0.82      0.82        66

Score 0.8181818181818182
```

3.3.1.2 KNeighborsClassifier :

In the case of the K neighbor classifier the user specifies the number of neighbors and for that value of neighbor the model will take into consideration that many neighbors and will give the prediction according to what the majority among those neighbors predict . As the name suggests this implements learning based on the k nearest neighbors . We tried to evaluate the model for different values of K i.e. for different numbers of neighbors and then compared the accuracy and considered the number of neighbors with which the model was giving the best result.

KNeighbor gave the best result for spanish dataset with count vectorizer and ngram range (1,1) and with number of neighbors 7.

3.3.1.3 Logistic Regression :

It is a statistical model that uses a logistic function to model binary dependent variables. Logistic regression estimates the parameters of a logistic model while doing regression analysis.It is used to examine the association of (in our case categorical) independent variable with one dichotomous dependent variable. It is using its conditional probability to find whether the combined tweets of a tweet id is spreading hate speech or not.

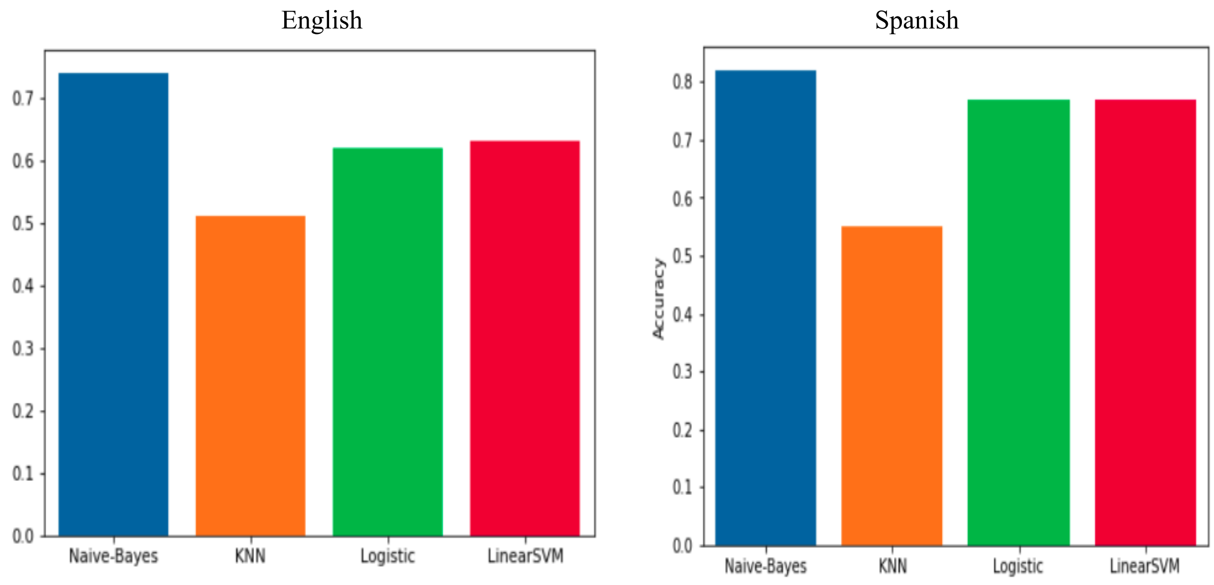
The above model gave best accuracy using count vectorizer and ngram range of (1,1) i.e 80%.

3.3.1.4 Linear SVM SVC :

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

The above model gave best accuracy using tfidf vectorizer and for both ngram range of (1,1) and ngram range of (1,3) i.e 77%.

Comparing performance of different Machine Learning models


English

Spanish

3.3.2 Different deep learning models used for classification:

3.3.2.1 LSTM :

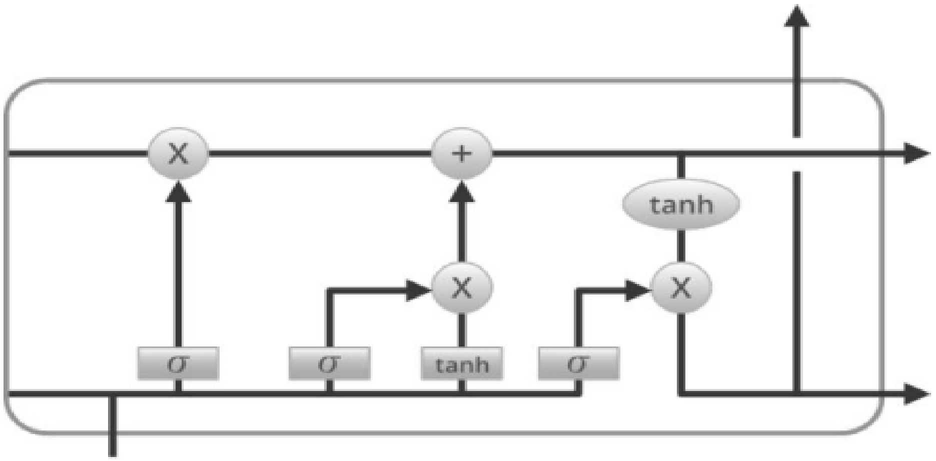
It stands for (Long Short -Term Memory) designed by Hochreiter & Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long term memory but can give more accurate predictions from the recent information. As the gap length increases 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 a chain structure that contains four neural networks and different memory blocks called cells. Information is retained by the cells and the memory manipulation is 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

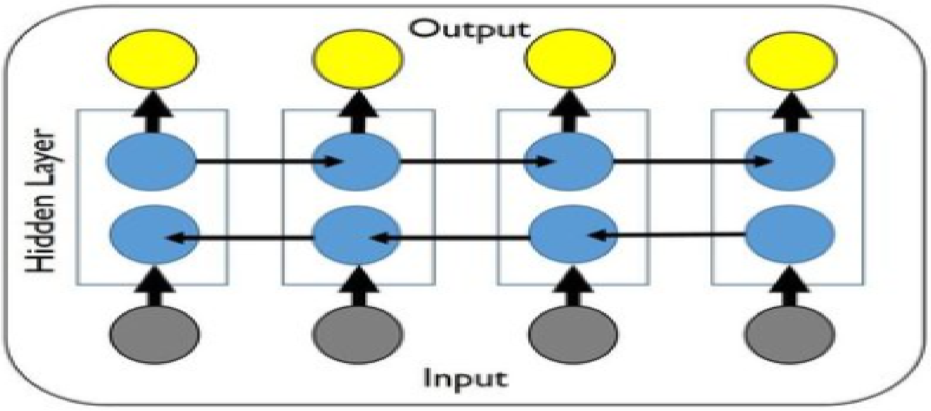
Metrics	Accuracy

3.3.2.2 BiLSTM :

Its abbreviation of bidirectional LSTM is an extension of traditional LSTM that can improve model performance on sequence classification problems . The problems in which all timesteps of input sequence are available are dealt with by this model .They 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.

All the other steps will be the same as we took in lstm. The only thing which will change is the designing of the model .



Steps we followed while training our BILSTM :

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 :

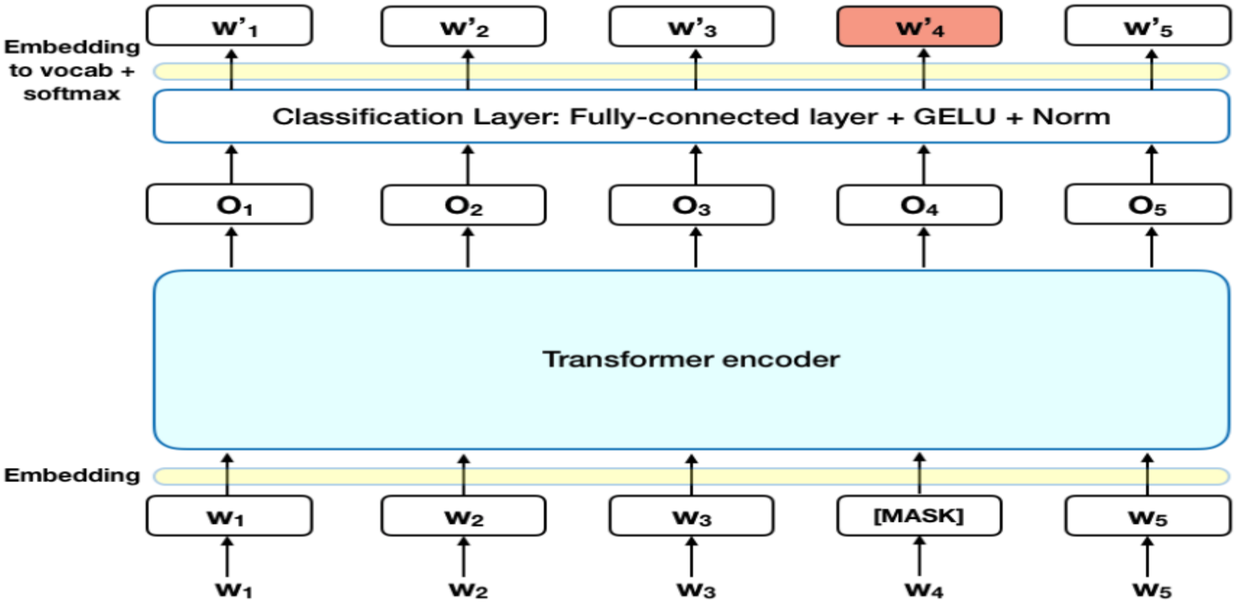
First we defined the number of vector features , we took it as 40.
The layers :

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. **Bidirectional 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.

3.3.2.3 BERT :

BERT(Bidirectional Encoder Representation From Transformers) is a great innovationF in the field of Machine Learning for Natural Language Processing. It was introduced in 2018 by the researchers of Google AI Language.BERT is different from the directional model which reads the input sequentially(left to right or right to left).

It reads the whole sentence at once.That's why it is considered bidirectional. The paper[8] showed that a bidirectionally trained model performs better than a single direction trained model.BERT can be easily fine tuned for the classification problem,question answer problem and named entity problem.


w'1
w'2
w'3
w'4
w'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
w1
w2
w3
[MASK]
w5
w1
w2
w3
w4
w5

Steps we followed while training BERT model:

1. First of all, we imported the BERT Tokenizer and Sequence Classifier.

2. Convert each row of the data into an InputExample Object.

3. We did tokenization of the InputExample objects and created the required input format from the tokens so that we can feed the data to the model.

Description Of Hyper Parameters :

Activation function	Relu
Loss function	Sparse Categorical Cross Entropy
Optimizer	Adam
Epochs	15
Batch-size	64
Metrics	Accuracy

4. RESULTS

4.1 Evaluation Metrics :

For the evaluation of proposed models we used precision , recall , F1 Score , support and accuracy.We have used classification report and confusion matrix , these metrics are widely used for evaluating supervised machine learning models for classification when the dataset is multi labelled . Suppose if a multi-labeled dataset consists 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 :

It is the ratio of accurately predicted authors as hate speech spreaders or not to the total number of predicted authors. It is computed as given in equation below. The range of precision varies between 0 and 1 , where 1 is the best value and 0 is the worst value.

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

4.1.2 Recall :

It is the ratio of accurately predicted authors as hate speech spreaders or not to the total number of authors. It is computed as is given in below equation. The range of recall varies between 0 and 1 , where 1 is the best and 0 is the worst value .

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

4.1.3 F1-Score :

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

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

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			
N	2			

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	9			

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

K Neighbours - TF-IDF				
	Precision	Recall	F1 - Score	Support
0	0.67	0.67	0.67	36
1	0.6	0.6	0.6	30
weighted avg	0.64	0.64	0.64	66
Accuracy	0.636			
N	6			

Logistic Regression - Count vect				
	Precision	Recall	F1 - Score	Support
0	0.67	0.61	0.64	36
1	0.58	0.63	0.6	30
weighted avg	0.63	0.62	0.62	66
Accuracy	0.6212			
Logistic Regression - TF-IDF				
	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			

Logistic Regression - Count vect				
	Precision	Recall	F1 - Score	Support
0	0.76	0.61	0.68	36
1	0.62	0.77	0.69	30
weighted avg	0.7	0.68	0.68	66
Accuracy	0.6818			
Logistic Regression - TF-IDF				
	Precision	Recall	F1 - Score	Support
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			

SVC SVM - TF-IDF				
	Precision	Recall	F1 - Score	Support
0	0.75	0.5	0.6	36
1	0.57	0.8	0.67	30

Linear SVC SVM - TF IDF				
	Precision	Recall	F1 - Score	Support
0	0.88	0.39	0.54	36
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			

Bi LSTM	
Accuracy	54 %
Epochs	15
LSTM	
Accuracy	44%
Epochs	15
BERT	
Accuracy	53%
Epochs	15

For spanish dataset

Spanish						Spanish				
using N-grams range - (1, 1)						using N-grams range - (1, 3)				
Naive Bayes - Count Vectoriser						Naive Bayes - Count Vectoriser				
	Precision	Recall	F1 - Score	Support			Precision	Recall	F1 - Score	Support
0	0.78	0.78	0.78	32		0	0.86	0.75	0.8	32
1	0.79	0.79	0.79	34		1	0.79	0.88	0.83	34
weighted avg	0.79	0.79	0.79	66		weighted avg	0.82	0.82	0.82	66
Accuracy	0.787					Accuracy	0.81			
Alpha	0.2					Alpha	0.1			
Naive Bayes - TF-IDF						Naive Bayes - TF-IDF				
	Precision	Recall	F1 - Score	Support			Precision	Recall	F1 - Score	Support
0	0.79	0.81	0.8	32		0	0.86	0.75	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			

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 - 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	0.79	0.72	0.75	32
1	0.76	0.82	0.79	34
weighted avg	0.77	0.77	0.77	66
Accuracy	0.77			

Logistic Regression - TF-IDF				
	Precision	Recall	F1 - Score	Support
0	0.84	0.66	0.74	32
1	0.73	0.88	0.8	34
weighted avg	0.78	0.77	0.77	66
Accuracy	0.77			

Linear SVC SVM - Count vectoriser				
	Precision	Recall	F1 - Score	Support
0	1	0.06	0.12	32
1	0.53	1	0.69	34
weighted avg	0.76	0.55	0.41	66
Accuracy	0.54			

Linear SVC SVM - Count vectoriser				
	Precision	Recall	F1 - Score	Support
0	1	0.06	0.12	32
1	0.53	1	0.69	34
weighted avg	0.76	0.55	0.41	66
Accuracy	0.54			

Linear SVC SVM - TF-IDF				
	Precision	Recall	F1 - Score	Support
0	0.79	0.72	0.75	32
1	0.76	0.82	0.79	34
weighted avg	0.77	0.77	0.77	66
Accuracy	0.77			

Linear SVC SVM - TF-IDF				
	Precision	Recall	F1 - Score	Support
0	0.84	0.66	0.74	32
1	0.73	0.88	0.8	34
weighted avg	0.78	0.77	0.77	66
Accuracy	0.77			

Bi LSTM		LSTM		BERT	
Accuracy	48.5 %	Accuracy	46.9%	Accuracy	56.06%
Epochs	15	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 ranges 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 use 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)

[8]Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova BERT: Pre-training of Deep Bidirectional Transformers forLanguage Understanding https://arxiv.org/pdf/1810.04805.pdf

[9] Aditya Gaydhani∗, Vikrant Doma†, Shrikant Kendre‡and Laxmi Bhagwat§.Detecting Hate Speech and Offensive Language onTwitter using Machine Learning: An N-gram and TFIDF basedApproach.arXiv:1809.08651v1[cs.CL]23 Sep 2018 https://arxiv.org/pdf/1809.08651v1.pdf

[10]Sai Saketh Aluru1†, Binny Mathew1†, Punyajoy Saha1, andAnimesh Mukherjee.Deep Learning Models for Multilingual HateSpeech Detection.arXiv:2004.06465v3 [cs.SI] 9 Dec 2020 https://arxiv.org/pdf/2004.06465v3.pdf

[11] Punyajoy Saha1, Binny Mathew2, Pawan Goyal2, Animesh Mukherjee2.Hateminers : Detecting Hate speech against Women.arXiv:1812.06700v1 [cs.SI] 17 Dec 2018.https://arxiv.org/pdf/1812.06700v1.pdf.

[12]RaymondTMutanga1,Nalindren Naicker2,Oludayo O Olugbara3.Hate Speech Detection in Twitter using Transformer Methods.ternational Journal of Advanced Computer Science and Applications,Vol.11,No.9,2020.https://thesai.org/Downloads/Volume11No9/Paper_72-Hate_Speech_Detection_in_Twitter.pdf

[13]Georgios K. Pitsilis Heri Ramampiaro Helge Langseth. Effective hate-speech detection in Twitter data using recurrent neural networks