Model comparison of hate speech detection across different social media platforms

Swati Thapa
210151488
Peiling Yi
MSc in Big Data Science: Queen Mary University of London

***Abstract* – Hate speech detection on social media platforms has been an extensive research area. There has been immense research where researchers have tried to find the solution in their unique ways. Along with text data, different feature extraction like sentiment score, n-grams etc., techniques have been used as input to machine learning models or trying to create ensemble machine learning models or using deep learning; there have been multiple ways in which solutions to this issue has been initiated.**

BERT is one of the revolutionary inventions in the field of NLP it has mostly outperformed any traditional NLP model [21] . In this field of study, many different techniques have been used however, there is not much relevant work comparing BERT with other deep learning models. Therefore in our study, I propose models where BERT embedding and GLOVE embeddings are used with different deep learning models. Each model is tested across different social media platforms. This experiment is done on three datasets namely FormSpring, YouTube and Twitter dataset. We will compare different evaluation metrics and try to get the best model across all platforms.

***Keywords* – Hate speech, abusive words, NLP,transformers, Glove, word embedding, BERT, hate words,LSTM**

Disclaimer: This paper may contain strong and abusive language for educational purpose.

I. INTRODUCTION

We all are aware due to the easy availability of fast internet, most of us love spending time on social media. These platforms help us to connect to our loved ones or even celebrities, but with its pros, there is always a con. Researchers have done the study and have found that the same platforms are the major cause of depression, anxiety and suicidal thoughts in millennials and gen z [22] . One of the most common reasons is hate speech. It is effortless to write hate speech comments on social media as there is no strict law to tackle this issue yet. Major social media platforms like Twitter, Facebook and Instagram are trying different methods to tackle this issue; however, it has not achieved much in this area. I see this issue as one of the major issues in the world of the internet because it affects a person's self-esteem and low self-esteem makes a victim's life difficult, or even worst victim might commit some horrific act such as suicide. Another issue is there are many countries in the world where we hear the news that sharing a hate comment on social media platforms makes that person targeted by other communities, and there have been actual incidents where the victim has lost his life due to sharing a hate comment.

There has been ample research in this field with many different solutions [11]. Though there has been an amazing model with the solution still, this area of study has some limitations due to which it becomes difficult to implement it practically; for instance, most of the models trained in this research [10] uses publicly available data instead of live data. The data available publicly doesn't give much insight on what bases the data has been labelled; for instance, was it reviewed by some expert like in [1] or was it just labelled based on one person's intuition. Most of the solution ignores that the dataset is highly imbalanced and the model trained on it can be highly biased; though to overcome this issue, there have been some fantastic solution like in [5] GAN is used; however, just to implement it requires high computational power. Most models created work on one platform, namely Twitter but failed on other platforms. We have discussed more limitations under "Critical Analysis and Related work". In this paper, from the above-mentioned limitation, I have worked on the data imbalanced issue and have tried to make my model work across different platforms with a good macro F1 score. To overcome an imbalanced dataset, I'm using simple class weights. It is achieved by giving more attention to the minority class (hate speech data in our case). For my project, I have chosen four different models [20]; where my first model is the BERT model, which is also my baseline model [16]due to its outperformance over any traditional NLP model [21] . My second model consists of BERT embeddings on text data which is fed to a simple multi-perceptron layer; similarly, my third model is also BERT embeddings on text data but is fed to BiLSTM, and finally, my proposed model is GLOVE embedding of text and hate word feature data which is later concatenated and is used as the input layer to LSTM [19]. From the models mentioned above, we will see how our proposed model outperforms all the other models and is capable of performing well across different platforms.

II. Critical Analysis and Related work

For the past few years, there has been a lot of research into identifying hate speech. In this area of study, numerous researchers have published some truly stunning architecture and their original methodologies. I have discussed some of the papers in this part. While going through the papers, I have insight into how different approaches have their own advantages and disadvantages. At the end of this part, I have mentioned my overall analysis.

The paper [1] which is one of the earliest papers in the field of hate detection, uses a Twitter dataset which was

retrieved and annotated manually and was also reviewed by a student studying gender studies and is also a non-activist feminist. Authors have used 16k tweets labelled as sexist, racist and none. Grid search was applied for different feature combinations, and it was observed that character n-grams outperformed all. In this experiment, character n-gram of length up to 4 long with gender, which was used as an additional feature, gave the best results.

A. Combination techniques:

In paper [2] authors have tried different combinations of the model to get better precision and recall. Firstly authors have used some baseline models such as char n-gram with logistic regression, TF-IDF with SVM, TF-IDF with GBDT (Gradient boosting decision tree) and more. Later authors tried different model combinations, such as CNN with random embedding and CNN with GLoVe, but LSTM+Random Embedding+GBDT achieved the best result. This whole training and testing was done in 16k annotated tweets made available by authors [1]

B. Ensemble techniques:

Not only a single model, there have been many types of research where multiple machine learning models have been used either sequentially or in parallel as discussed in [3] where authors of this paper have come up with unique architecture where they make a multi-view SVM model which is made by multiple-view stacked SVM. In this, each type of feature is feed to an individual Linear SVM classifier. Therefore, each view classifier learns to categorise the sentence based on a particular type of feature rather than merging all features into a single feature vector. Creating a view-classifier to learn each component of the pattern separately. This paradigm, however, encounters difficulty when the attitude toward the subject shifts over time and in the context of historical events. For instance

". . .The merciless Indian Savages, whose known rule of warfare, is an undistinguished destruction of all ages, sexes and conditions. . ."

The model says it's hate speech; however, this is a well-known line from the Declaration of Independence. Given the historical context, the person must not have planned to spread hate speech when they uploaded it; instead, they likely only wished to paraphrase the historical passage for another reason. It suggests that context and user intent are key factors in recognition of hate speech.

In the paper [4] authors have used machine learning models in a very creative way; they have divided the training of models into two sets, one as base models and another as a meta-model. Support vector machine (SVM), logistic regression (LR), and XGBoost classifier (XGB) are the three classifiers that make up this base model. They are trained concurrently utilising word2vec and universal encoding features on the training set. A meta classifier is just a logistic regressor whose input is the accumulation of base classifier outputs and development set feature extractions.

C. Unsupervised techniques:

Generally, in hate speech detection, the dataset in which training is done is highly imbalanced. For instance, from the complete dataset, just five per cent of the data is a dataset which contains hate content. Due to this high imbalance, it becomes difficult to train a highly efficient model. This issue is tackled in [5] and [6]

Roy Lee and Rui Cao are [5] create an idea called HateGAN, which employs a GAN (Generative Adversarial Network) architecture based on reinforcement learning to produce hate speech for data augmentation. Short hate speech is produced as a result, which is added to the data to help hate speech detection.

This study [6] was written to address the problem of an imbalanced dataset, which prevents the model from receiving enough training. The AngryBERT model, which employs the shared-private method, is introduced in this study. To carry out primary and secondary tasks, this model distinguishes between task-dependent and task-invariant (shared) features. Using emotion classification and target identification, the authors of this study have trained a model to extract emotions concealed in tweets. A model is trained to identify targets in the text. The MTL (Multitask learning) models can extract emotions and target groups or individuals from tweets when they are co-trained with these two secondary objectives, making it easier to detect indirect hate speech.

D. Transfer Learning techniques:

Hate speech is identified using a transfer learning method developed by the authors of [8] that relies on a BERT model that has already been trained. When a pre-trained model is being fine-tuned, bias in training is taken into account using a bias alleviation technique. The high correlation between features and class labels is reduced using the regularisation method. The "Hate Speech Detection module"and the "Bias Mitigation module"are the two main components that make up the architecture in this article.

E. Critical Analysis:

After going through different papers on hate speech, I have concluded that:

1. Most of the datasets are highly imbalanced because hate speech is detected in a setting where the majority of interactions are neutral, and hate speech is one of a few unique causes. Though we discussed above, few papers have tried using different techniques, such as in HateGAN generating synthesized tweets using GAN. But in reality, it is a technical hindrance to train GAN from scratch for hate speech data as it has many number of parameters.
2. The majority of the models used in the paper are supervised learning, meaning it requires human annotation for the label who have intense knowledge of hate speech and who can decide which conversations are hate and which are not.
3. The meaning of "hate"is pretty vague. For instance, some authors consider hate as racist, sexist conversation, cyberbullying, and harassment. Until now, the research community have not come up with one term meaning for hate speech, which makes research results vary from one hate conversation to another.
4. Most models discussed above use training Twitter datasets. However, some paper has tested in the different dataset but the majority is on Twitter [13], making the model biased for one platform. For instance, one tweet's word limit

is 280, whereas the word limit on YouTube is undefined. So using LSTM might be good for the Twitter dataset, but there are high chances it will fail on the YouTube dataset.

5. There are some major drawbacks in the models discussed; for instance, many of these models have overfitting issues making them not transferable to production.

6. Most of the research was done mostly on the Twitter dataset, especially from [1] which was written in 2016. Most hate slang must have been evaluated by now, meaning it is important to train our model from time to time for better results.

III. Methodology

A. Method used:

1) BERT: BERT (Bidirectional Encoder Representations from Transformers). Its training is performed in two stages:

a. To understand the given language BERT is pretrained.

b. To learn specific problems, Fine Tune is performed in BERT.

Pretraining stage: In this, BERT learns what language is and its context. It can be achieved by simultaneously training on two unsupervised tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP).

Fine Tuning stage: BERT can be trained to perform extremely particular tasks, such as the classification of hate speech and non-hate speech in our example. All we need to do in this case is swap out the network's fully linked output layers for new ones that can label the output according to our specifications. Then, since just the output parameters need to be learned from scratch, supervised training may be carried out using our dataset. Training time is shortened due to the slight fine-tuning of the remaining model parameters.

2) Multi Layer Perceptron: A multilayer perceptron (MLP) is a fully connected feed-forward neural networks. It consists of many layers such as input layer, output layer and hidden layers. Our experiment's input layer will be BERT embeddings, just one hidden layer and an output layer that will give our required prediction.

3) LSTM: Long short term memory is a variety of Recurrent Neural Network (RNN) which overcome the drawback of short term memory of RNN. LSTM is capable of having long term memory as it becomes essential for problem which relies on the sequence of the text. LSTM consist of three gates namely forget, input and output gates. These gates help regulate the flow of information and can decide which memory to keep and which memory to throw away.

4) BiDirectional LSTM: The enhanced LSTM is known as BiDirectional LSTM. Because the past sequence is the only input that a unidirectional LSTM has ever seen, past memory is retained. However, because the BiDirectional LSTM processes inputs in two directions—from the future to the past and from the past to the present—it preserves both past and future memory.

5) Embeddings: a. BERT embeddings: In this embedding, each word is embedded based on the context of the text. For instance:

"This is a river bank"

"Robbers robbed the bank"

Here bank has a different meaning with respective to its context; therefore same word-embedding representing both banks is not valid; therefore, we want word-embeddings based on the context, which means here, the word embedding of the bank in both the sentence should be different; therefore using this embedding gives more insight about the text data to the model.

b. GLOVE embeddings: This embedding is done based on the co-occurrence of words together. It is embedded based on the relationship between word pairs instead of just a single word. Hence try to get the context.

B. Experiments:

1) Dataset:: For this experiment, I have collected three types of datasets publicly available from three different platforms, namely FormSpring, YouTube and Twitter. I am using these datasets to check across different platforms. For instance, Formspring is an anonymous question-answer platform; therefore, the dataset contains text in the form of questions and answers. Twitter is a very famous dataset; as mentioned above, most of the research uses Twitter dataset focused on this area as it is easy to extract and tweet is of a limited character. Additional to these datasets, I will be using the YouTube dataset. I am using this dataset because, unlike Twitter, it does not have any word limit for the comment.

These are labelled datasets which were available publicly. However, some of these datasets were available with more hate categories such as racism, sexism, offensive, hate, non-hate etc. I am just using two labels for this experiment: hate and non-hate speech. For convenience, I am considering racism, sexism, and offensive or cyberbullying as hate speech and others as non-hate speech in the dataset provided. In the dataset, '1' indicates hate speech and '0' indicates non-hate speech.

FormSpring dataset: This dataset contains features like bio, text, Cyberbully words, location userid etc. It has column severity, labelled from 0-9. '0' indicates no kind of hate speech, and the other represents the severity of hate text.

YouTube dataset: It contains features such as userid, number of comments, commentator's age, membership duration and the label (hate or non-hate).

Twitter dataset: This dataset contains tweets and annotations such as sexism, racism and none. Another feature is the label, where '0' indicates no hate and '1' indicates sexism and racism.

2) Preprocessing:: As mentioned above, each dataset has its feature, which concerning its platform is extractable. However, we cannot expect the same feature extraction from different platforms. Therefore for my experiment, I'll just use text and the label. It is to be noted that in my proposed model, one extra feature I have used is hate words. In the FormSpring dataset it is named as cyberbully words; however, I manually created this feature for the YouTube dataset. There are a few preprocessing step which was common for all three datasets. The preprocessing is as follows:

a. Removed punctuation.
b. Removed stopwords.
d. Lower-cased the text feature.
c. Removed words starting with '@' in the Twitter dataset as most tweets start with tagging another user using '@'. Also removed RT from the Twitter dataset as it simply means retweet, which doesn't hold much meaning while detecting hate speech.
d. It is to be observed that I didn't remove 'Q' and 'A' from the FormSpring dataset, which means the start of question and answer, respectively. I chose to keep it as I want my model to learn when the question starts, and the answer starts when given in the same line.
e. Removed duplicate entries from the dataset.

After doing those mentioned above preprocessing, the table below represents the hate speech and non-hate speech data in each dataset.

Dataset	Hate Speech	Non Hate Speech	Total
FormSpring	3411	13616	17027
YouTube	316	3144	3460
Twitter	5347	11477	16824

TABELA I
Data distribution

This preprocessing was applied to both text and hate word data.

3) Design: My input parameter is just the text data. The aim is to label it as hate or non-hate speech as all machine learning and deep learning models don't accept any form of input other than numeric, so it becomes a mandatory step to convert the text data to numeric form. This is achieved using the word embeddings method, where depending on the chosen embedding, each word in the text data is later converted to numeric data. In this paper, I'll use two types of embedding where the text data is converted into numeric form and later passed through to different models.

Model 1 (Base line model: BERT): My baseline model is simple BERT (uncased) from hugging face. I'm using uncased as I had tried all my models with cased BERT but uncased BERT outperformed hence selecting uncased BERT. This simple BERT model consists of 12 layers, 768 hidden sizes, 12 self-attention heads and 110M parameters.

Input to this model is just text which may or may not contain hate speech. Output to this model is predicted label whether the text is hate speech or non-hate speech.

Model 2 (BERT embeddings + MLP) I'm using a simple Multilayer Perceptron in this model where the input layer is BERT embeddings, one hidden layer and one output layer. Here I'm using 'pooled output' from the BERT embeddings. The pooled output is the embeddings from sequence output which is further fed to a linear layer and Tanh activation function. Here dropout layer is also used to avoid overfitting.

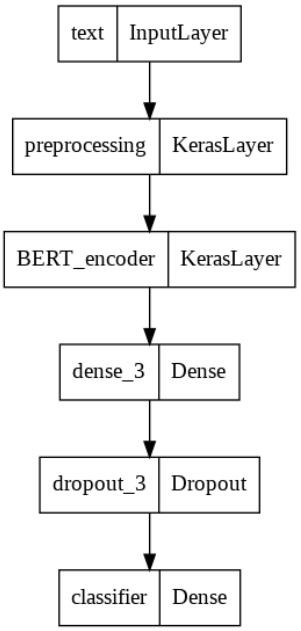

Fig. 1. Model 2 architecture (BERT embeddings + MLP)

Model 3 (BERT embeddings + BiLSTM): In this model, I'm using simple BiLSTM. Here the input is the BERT embeddings 'sequence-output'. This embedding is later fed on BiDirectional LSTM. The reason for using sequence output is that it is the output of the last layer of BERT; it contains positional embeddings, too, which will help make BiLSTM learn better contextually. Further, I'm using BiLSTM because now, since embedding is contextual embedding, and BiLSTM provides input in two ways, backwards and forward, making our model more robust. It is to be noted that BiLSTM output is again feed to a single layer MLP and dropout is used here again to avoid overfitting.

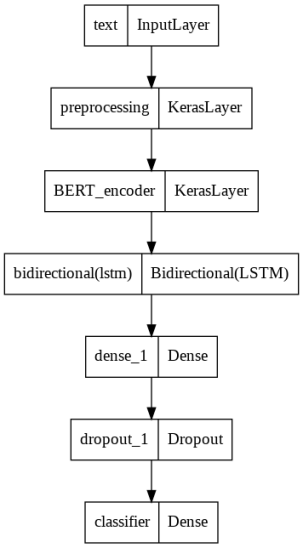

Fig. 2. Model 3 architecture (BERT embeddings + BiLSTM)

Model 4 architecture (GLOVE embeddings + hate words + LSTM): This is the proposed model where, besides text data, I'm using one more extra feature, hate words. These are the words that make our given text hate speech. For instance,

if the text is:

"How are you all?? You should kill yourself" then the hate words will be *"kill yourself"*

This feature was available for the FormSpring dataset only. However, I have manually created this feature for the YouTube dataset. Due to time constraints, I couldn't manually extract it for the Twitter dataset. Therefore this model has been tested only on FormSpring and YouTube datasets.

For embedding in this model, I'll be using GLOVE embeddings instead of BERT embeddings because BERT embeddings give the contextual embeddings. In contrast, GLOVE embeddings give fixed embeddings for a word irrespective of the context. In my model, if each word in text has been given a fixed embedding, then the same word in the hate words feature will also be given the same embedding making my model easily detect the hate comments. Further, I have used LSTM, where I have concatenated these two feature embeddings and have used them as an input layer to LSTM. I'm choosing LSTM because it is reliable for long-term dependencies, which is especially required for the youtube dataset as some contain comments that are longer than 400 words.

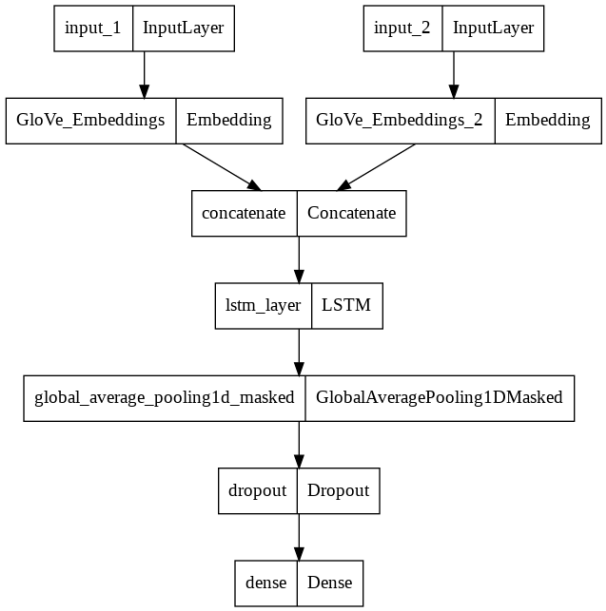

Fig. 3. Model 4 architecture (GLOVE embeddings + hate words + LSTM)

C. Handling imbalanced dataset:

Since the dataset is highly imbalanced meaning number of hate speech labelled is significantly low (mentioned in Table 1). If we don't balance the data, then our trained model will be more biased toward the majority class (in our case, non-hate speech). To overcome this issue, there are many different ways, for example, under-sampling, over-sampling, SMOTE, using class weights etc. In our experiment, it is not logical to use under sampling because even though non-hate speech labelled is more, it is still valuable information and deleting most of the comments so that our labelled data are in the same proportion will not make our model robust. Similarly, using oversampling and duplicating hate speech data will make our model overfit as it will perform well in the training set but will perform poorly in unseen data. Using SMOTE won't be that helpful because numeric vectors created from the text are of very high dimensionality. Therefore I'm using class weights to give more weight to our minority class (i.e. hate speech); this is achieved by using:

```
weight_for_0 = (1 / non_hate_speech)*(total)/2.0
weight_for_1 = (1 / hate_speech)*(total)/2.0
```

D. Training:

Here each dataset is split into training, validation and testing datasets. Where training data set is used for training, validation dataset to validate the model and testing dataset to finally test the model in complete unseen data. The chosen dataset is highly imbalanced; therefore, only 10% of data is used for testing and 10% of training data is used as a validation dataset. For example (fig 4):

			Text	hate_words	Text_clean	hate_words_clean
category	Label	data_type				
hate_speech	1	test	32	32	32	32
		train	256	256	256	256
		val	28	28	28	28
non_hate_speech	0	test	314	314	313	314
		train	2546	2546	2543	2546
		val	284	284	283	284

Fig. 4. YouTube dataset distribution

Here `non_hate_speech` is the number of non-hate speech labelled in the dataset, `hate_speech` is the number of hate speech labelled in the dataset, whereas the total is the sum of both hate and non-hate speech in the dataset. I'm dividing it by two because we have two class labels.

IV. Results

All the mentioned model has been trained by applying early stopping for True positive and loss. I have chosen True positive because the hate speech label is our true positive class. Since our ultimate aim is to detect hate speech, evaluating the model based on True positive becomes more important. In this, I have used early stopping for max validation true positive and min loss. If there is no value change from the last two epochs, it automatically saves the best model with high validation True positive and low loss. However, my ultimate evaluation is based on macro F1 score, precision and recall. The reason for choosing these as my evaluation method is because since it is a highly imbalanced dataset, choosing accuracy is not suitable. After all, accuracy fails to differentiate between the numbers of correctly labelled instances of different classes.

Precision:

$$\frac{TP}{TP+FP}$$

Recall:

$$\frac{TP}{TP+FN}$$

macro-F1:

$$\frac{1}{n} \times \sum_{n=1} \frac{2 * Precision * Recall}{Precision + Recall}$$

Here abbrevation are: TP: True Positive FP: False Positive FN: Flase negative n : number of class

It can be observed from the (Fig 4) we have very less hate speech dataset for YouTube data. Below is the confusion matrix of baseline model (BERT) on YouTube dataset.

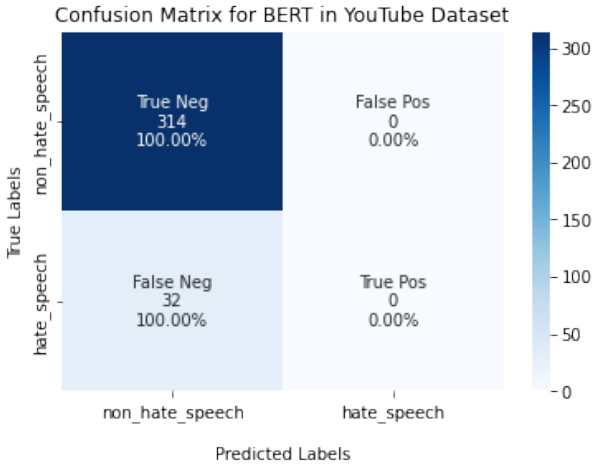

Fig. 5. Confusion matrix for BERT (Model 1) for YouTube Dataset

From the above figure, it is to be noted that our baseline model can detect 100% negative class (non-hate speech); however, it has failed to detect positive class (hate speech) hence 0% True positive. From our experiment point of view, this model is not very useful to us.

If we observe the confusion matrix (Fig.6) for BERT Embeddings + MLP on the YouTube dataset though we might not have 100% on True negative (non-hate speech), we have 6.25% for True positive class (Hate speech). This might not be the best model for us, but it is surely better than the baseline model as we aim to detect True Positive instead of True Negative, unlike in Fig.5. Similar matrix was generated for BERT Embeddings + BiLSTM.

Now, if we observe (Fig.7) the same dataset for our proposed model (GLOVE embeddings + hate words+ LSTM), we can see not only was our model able to predict the True Positive class (hate speech) 100% accurately it was also able to predict 92.04% for True Negative class(non-hate speech). This is the best model by far now as it is able to predict our required class perfectly. It even has macro F1 score of 0.839 (Table II) Now, if we observe (Fig.8) the same model in a different platform that is FormSpring, it can be observed that it has performed well for this dataset, too, by predicting approx 99.34% True negative class and 94.13% True positive class. From the above analysis and the table (Table II), it can be observed that the proposed model has outperformed all the models on different platforms. Hence we can say that feature extraction of hate words from the comments is one of

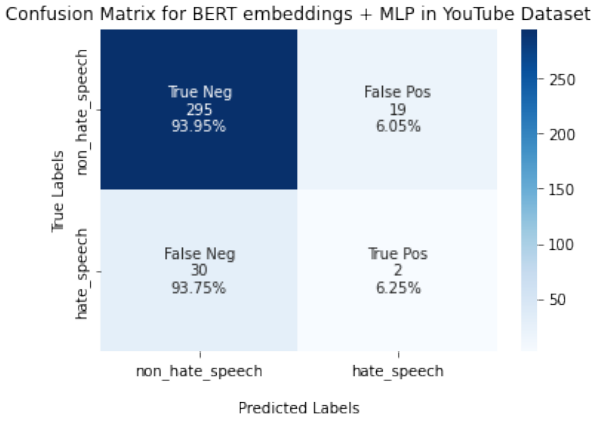

Fig. 6. Confusion matrix for BERT Embeddings + MLP (Model 2) in YouTube Dataset

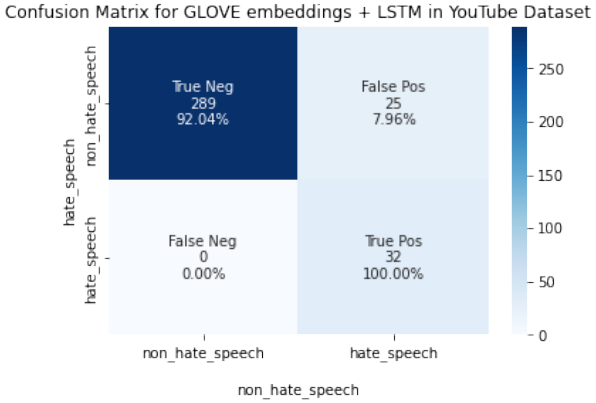

Fig. 7. Confusion matrix for proposed model (Model 4) in YouTube Dataset

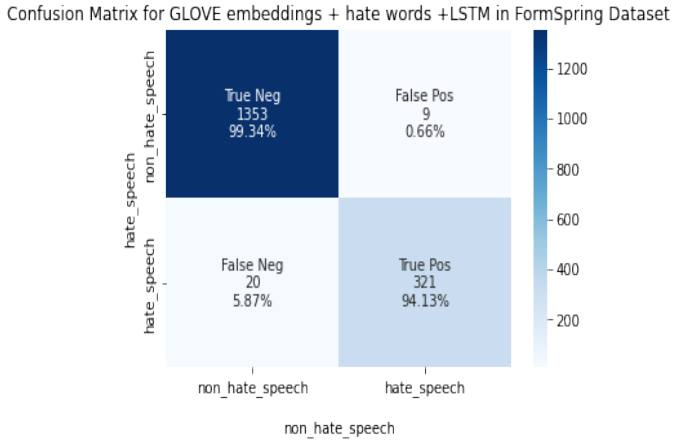

Fig. 8. Confusion matrix for proposed model (Model 4) in FormSpring Dataset

TABELA II
Macro F1 score of different models

Model	Dataset	macro F1 score	Precision	Recall
Model 1 (Baseline)	FormSpring	0.519	0.519	0.521
	YouTube	0.476	0.454	0.500
	Twitter	0.490	0.492	0.491
Model 2	FormSpring	0.479	0.479	0.484
	YouTube	0.499	0.501	0.501
	Twitter	0.506	0.506	0.506
Model 3	FormSpring	0.470	0.473	0.469
	YouTube	0.477	0.474	0.481
	Twitter	0.493	0.493	0.493
Model 4	FormSpring	0.973	0.979	0.967
	YouTube	0.839	0.781	0.96

the crucial features of detecting hate speech.

V. Conclusion

In this paper, our main aim was to detect hate speech in a text. For this, we experimented with four different types of models. Our baseline model was just a simple BERT implemented using hugging face, whereas our macro F1 score was approximately around 0.50, meaning it hasn't performed that well. Even after experimenting with other models such as BERT embeddings + MLP and BERT embeddings + BiLSTM macro F1 score and other evaluation methods have more or less increased or decreased, which indicates adding a model with BERT embeddings didn't make any difference as compared to our baseline model(BERT). However, our proposed model has given better evaluation results for all three evaluation methods. This was achieved because, in our proposed model, I have used one extra feature as hate words which were actual words from the given text where the label is hate speech. Since the same model has given both datasets a pretty good macro F1 score, we can also conclude that the same model can be used across different platforms. Here is my analysis for my proposed model:

(i). It is to be noted that this extra feature was created manually for FormSpring and YouTube dataset, which in real life is not feasible. However, it can be achieved automatically via unsupervised learning if we train our model in a given FormSpring and YouTube dataset to extract hate words when text data and label are given as input and later use the same model on Twitter dataset to extract the hate words.

(ii). I have observed some of the text labellings are still confusing; for instance, *"You are fucking awesome"* should be considered as hate or non-hate speech? How do we overcome the intention of a person and can state it as hate or non-hate speech? Well, this is again a very broad topic to discuss.

(iii). In the BERT model, the maximum length of text it accepts is 512, meaning it can take a maximum of 512 words in a sentence [23]. In our text data, especially in the YouTube dataset, we have the text of some length of more than 700. So when the text of more than 512 is introduced to BERT or just BERT embeddings, it just accepts the first 512 words discarding the remaining words hence losing information due to which macro F1 score is low. For my proposed model in the YouTube dataset, I have deleted data with a length above 500 and made GLOVE embeddings input as 500 as compared to other datasets where I have used max length as 200. I deleted these rows because, in our YouTube dataset, there were hardly 10 data containing more than 500 words. Hence limiting my model max length to 500.

However, from the above analysis, we can conclude that the proposed model gave the best result in all the evaluation methods.

VI. Future Work

Due to time constraints, I was able to manually extract the hate words for the YouTube dataset. However, to better evaluate the model, we can extract the hate words for a Twitter dataset, too, which can be achieved by unsupervised model techniques. Such as, we can use unsupervised learning to detect the preposition pattern around hate words or train an unsupervised model on FormSpring and YouTube data where text data and label are input, then later use this model on Twitter data to extract the hate words.

In the future, we can also deploy the model [9] on a cloud platform and can use the model in real time practise either as API or as website.

Additionally, we can also train the model in different languages to be robust for multilingual languages. As hate is not just limited to the English language, there is still another part of the world where a regional language is given more priority and even spreading hate in the regional language is highly possible. [14] [15]

In this paper, I have just used one extra feature [17] and the evaluation result improved significantly. Similarly we can also try more different features along with text data.

It is also observed how hate speech is not just limited to text but also images. Along with text, we can work on detecting hate on images simultaneously [18]

VII. Reference

1. Waseem, Z. and Hovy, D., 2016, June. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In Proceedings of the NAACL student research workshop (pp. 88-93).

2. Badjatiya, P., Gupta, S., Gupta, M. and Varma, V., 2017, April. Deep learning for hate speech detection in tweets. In Proceedings of the 26th international conference on World Wide Web companion (pp. 759-760).

3. MacAvaney, S., Yao, H.R., Yang, E., Russell, K., Goharian, N. and Frieder, O., 2019. Hate speech detection: Challenges and solutions. PloS one, 14(8), p.e0221152.

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

5. Cao, R. and Lee, R.K.W., 2020, December. Hategan: Adversarial generative-based data augmentation for hate speech detection. In Proceedings of the 28th International Conference on Computational Linguistics (pp. 6327-6338).

6. Awal, M.R., Cao, R., Lee, R.K.W. and Mitrović, S., 2021, May. Angrybert: Joint learning target and emotion for hate speech detection. In Pacific-Asia conference on knowledge discovery and data mining (pp. 701-713). Springer, Cham.

7. Al-Makhadmeh, Z. and Tolba, A., 2020. Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing, 102(2), pp.501-522.

8. Mozafari, M., Farahbakhsh, R. and Crespi, N., 2020. Hate speech detection and racial bias mitigation in social media based on BERT model. PloS one, 15(8), p.e0237861.

9. Anagnostou, A., Mollas, I. and Tsoumakas, G., 2018, July. Hatebusters: A Web Application for Actively Reporting YouTube Hate Speech. In IJCAI (pp. 5796-5798).

10. Schmidt, A. and Wiegand, M., 2017, April. A survey on hate speech detection using natural language processing. In Proceedings of the fifth international workshop on natural language processing for social media (pp. 1-10).

11. Jahan, M.S. and Oussalah, M., 2021. A systematic review of hate speech automatic detection using natural language processing. arXiv preprint arXiv:2106.00742.

12. Rajput, G., Punn, N.S., Sonbhadra, S.K. and Agarwal, S., 2021, December. Hate speech detection using static BERT embeddings. In International Conference on Big Data Analytics (pp. 67-77). Springer, Cham.

13. Qian, J., ElSherief, M., Belding, E.M. and Wang, W.Y., 2018. Leveraging intra-user and inter-user representation learning for automated hate speech detection. arXiv preprint arXiv:1804.03124.

14. Al-Hassan, A. and Al-Dossari, H., 2019, February. Detection of hate speech in social networks: a survey on multilingual corpus. In 6th International Conference on Computer Science and Information Technology (Vol. 10, pp. 10-5121).

15. Sohn, H. and Lee, H., 2019, November. Mc-bert4hate: Hate speech detection using multi-channel bert for different languages and translations. In 2019 International Conference on Data Mining Workshops (ICDMW) (pp. 551-559). IEEE.

16. Mutanga, R.T., Naicker, N. and Olugbara, O.O., 2020. Hate speech detection in twitter using transformer methods. International Journal of Advanced Computer Science and Applications, 11(9).

17. Alrehili, A., 2019, November. Automatic hate speech detection on social media: A brief survey. In 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA) (pp. 1-6). IEEE.

18. Das, A., Wahi, J.S. and Li, S., 2020. Detecting hate speech in multi-modal memes. arXiv preprint arXiv:2012.14891.

19. Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C. and Patel, A., 2019, December. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In Proceedings of the 11th forum for information retrieval evaluation (pp. 14-17).

20. Zhou, Y., Yang, Y., Liu, H., Liu, X. and Savage, N., 2020. Deep learning based fusion approach for hate speech detection. IEEE Access, 8, pp.128923-128929.

21. Raj, B., 2019. Understanding BERT: Is it a Game Changer in NLP?. [online] Medium. Available at: <https://towardsdatascience.com/understanding-bert-is-it-a-game-changer-in-nlp-7cca943cf3ad>.

22. Quandt, T., Klapproth, J. and Frischlich, L., n.d. Dark social media participation and well-being. [online] Available at: <https://doi.org/10.1016/j.copsyc.2021.11.004>.

23. Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

MSc Project - Reflective Essay

Project Title:	Model comparison of hate speech detection across different social media platforms.
Student Name:	Swati Thapa
Student Number:	210151488
Supervisor Name:	Peiling Yi
Programme of Study:	ECS750P

Disclaimer: This essay might contain some strong and abusive language.

In this project, our ultimate goal was to achieve a hate speech or non-hate speech label from the text data. These texts are the comments extracted from different social media platforms. This project explored different models with a different set of word embedding. We observed how just using word embedding (I have used BERT embedding) with different models didn't make much difference. In this essay, we will discuss the strength and weaknesses of our model, followed by ethical issue and how we can improvise it further in future work.

The strength and Weakness of the project:

In this paper, my proposed model consists of text data and hate word feature as input and hate speech or non-hate speech as labelled data. Here it is to be noted that hate words are those words in the text which make the text data as hate speech. For instance:

"How are you now? You should kill yourself."

Here text data will be the whole sentence, and *"kill yourself"* will be this sentence's hate word feature.

Since this whole project is a text problem and any machine learning and deep learning model doesn't accept any string as input, it becomes essential to convert this text into a numeric dimension. Hence, it is feasible for our required model. Therefore for this task, GLOVE embedding is used. This converts the text data and hates word feature into numeric form, and later these two features are concatenated and used as input to the LSTM input layer. It is clear from the result discussed in our paper (as mentioned above) that our proposed model has outperformed all the tested models indicating the importance of the hate word feature for hate detection. This single model was tested on the dataset from two different platforms and still performed extremely well in both datasets, where the main difference in the text data was the length of the comment. For instance, the text length in the FormSpring dataset was pretty short compared to the comment length in the YouTube dataset, where the comment length was even more than 700 words. Through structural differences in both the platforms, the proposed model was robust in both the platform dataset. The simplicity of the model makes our model more user-friendly.

However, this model has its weakness; one of the major weaknesses is extracting hate words from the text data manually. This is extremely time-consuming. However, for our project, the FormSpring dataset already had this as a feature, and I created the hate words feature manually in the YouTube dataset; due to time constraints, I couldn't do it manually for

Twitter data; therefore, our model is only compared to two different dataset meaning it has been evaluated with only two different platforms.

However, it should be noted that while using GLOVE embedding, the input shape used is 500 meaning any text longer than 500 words length that text is truncated to 500 words and the remaining words are lost hence information loss. Therefore if the text data are longer than 500 words, it is a little difficult for the model to predict the correct label. Just not in GLOVE embedding, BERT embedding used in other models also faces a similar issue because, by default, BERT expects a maximum of 512 words of long text data.

Future work:

As mentioned above, one of the features is hate words which have been extracted manually. Since extracting it is not feasible, an unsupervised technique can be used to extract hate words. For instance, this could be achieved by first extracting the preposition of each word and trying to learn if the hate words followed any similar preposition track or not. Another way is to train a model to extract hate words from given FormSpring and Youtube datasets where input is text data and label and later use the same model on Twitter dataset to extract the hate words.

In my project, I have just used one feature, but the model is not limited to just one feature; we can try some other features, too, for instance, the age of the user who has generated the text or semantic score of each text. As discussed in related work, researchers have previously used many feature combinations with their model's similar way; we can also try different meaningful combinations for our chosen model.

We can also deploy the model as API or a normal extension. This could be very impactful in real life; for instance, if a user decides to use this extension with their browser and can choose to block all hate speech comments on the desired platform, which will ultimately save the user from all the negative effects of hate speech. However, the practical implication is more challenging.

In this project, we have discussed hate speech detection only in text data, but in today's world, hate speech is not just limited to text but also images. There have been many instances where users try to bully others using memes or other types of images. Along with text, we should focus on hate detection in images too. Though for doing this, we require both computer vision and NLP knowledge.

Difference between theoretical and practical implementation of the model:

There has been much research on this topic; however, practical implementation of the model in this area is still considered challenging. Several challenges hinder most of the practical implementation, for instance, labelling the data as hate speech or non-hate speech. On what bases can we say the text is hate or not? Example:

"You are fucking awesome."

In this sentence, should we consider this hate speech or not? Because the word *"fucking"* is considered abusive, but if you read the overall sentence, some may debate that it cannot be considered hate speech, and some may completely disagree. This sentence suggests that the user's intention is not to hurt someone's sentiment, but if we keep the sentiment aside, it can arguably be a hate comment. So the question arises what exactly hate speech is? How do we

consider a user intention with the data because understanding by a human can be easy, but doing the same task by a machine is very challenging?

Most of the research in this field includes a similar dataset (Twitter dataset) or open source dataset, which had been extracted some years back. How much can we rely on these data anymore? As the days pass, abusive language slang changes too. For instance, the words *"stupid" and" idiot"* were considered hate words a few years ago, but now, most users don't consider these as hate words. So, how do we keep updating our model in real-time with the ongoing trend of hate words?

It can be considered that if we keep extracting the desired current Social media platform text live, it might solve the issue mentioned above; however, as data privacy is a serious concern nowadays, it becomes difficult to extract the comments from social media platforms. For instance, Twitter API is open-sourced, and it is easy to extract the tweets; however, it is difficult to extract Instagram and Facebook comments due to companies changing policies. Therefore updating our model based on current data of social media platforms is difficult. Hence theoretical approach and real-time practical implementation of most models in this area are considered extremely challenging.

Legal, Social Ethical issues:

In hate speech detection or any AI model, there is always a concern about how ethical our model is. It has been observed that an AI model has the power to impact human behaviour; for instance, it has been accused that how big data was used to manipulate the election in a country. Similarly, there is a high chance hate speech detection model can also have a negative impact with its positive implementation. As discussed earlier, it is still unclear to many researchers what exactly the definition of hate speech is then how we can rely on AI discussion. We don’t know what exactly influences the AI to detect it as hate speech or not. For instance, on the Twitter platform, it has been observed that many users have been banned from the platform stating that they tried to spread the hate comment but where in fact, all they were doing was tweeting the awareness. This shows how AI can hinder freedom of speech and human rights.

To regulate AI, stricter laws should be implemented, and humans should regulate all the decisions made by the AI model. AI can be used to label the text, but eventually, human involvement should make the decision. This way, we can make sure AI discussion is not having any negative impact on the users.

Conclusion:

In this essay, we discussed the proposed model and discussed its strength and weakness. We also discussed how this model could be modified for betterment in future work. We compared the theoretical and practical implementation and discussed the different challenges the model will face in practice. In the end, we discussed the legal and ethical issues related to this model and how an AI model can jeopardize freedom of speech and human rights.

References:

1. Cortiz, D. and Zubiaga, D., n.d. *View of Ethical and technical challenges of AI in tackling hate speech | The International Review of Information Ethics*. [online] Informationethics.ca. Available at: <https://informationethics.ca/index.php/irie/article/view/416/389>.