Hate Speech detection in English and Malayalam Code-Mixed Text using BERT embedding

2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS) | 978-1-6654-6883-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IC3SIS54991.2022.9885339

Deepasree Varma P
Department of Computer Application
Scms School of Engineering and Technology
Cochin, India
deepasree@scmsgroup.org

Dr Vinod P
Department of Computer Application
Cochin University of Science and Technology
pvinod21@ gmail.com

Dr M Nandakumar
Department of Electrical and Electronics Engineering
Christ College of Engineering
Irinjalakkuda
mnkumar.tcr@gmail.com

Akshay K
Department of Computer Application
SCMS College Of Engineering and Technology
Karukkutty, Ernakulam, Kerala ,India
akshayanil16@gmail.com

Akhil Madhu
Department of Computer Application
SCMS College of Engineering and Technology
Karukkutty, Ernakulam, Kerala ,India
akhilmadhu81@gmail.com

***Abstract*—Hate speech detection is a very popular research area for past few years. Hate speech is given various definition by various researchers. In this paper we try to analyse the use of BERT embedding in hate speech detection in low resource language like Malayalam. The crucial challenge faced by researchers in this area are that most non-English languages are represented in code-mixed form in Social media. Here we work with transformer-based models to classify tweets as hate or nonhate content. Hence this is a novel approach that uses BERT in non-English text.**

***Keywords*—hate speech, social media, BERT, Codemix**

1. INTRODUCTION

Nowadays, it is evident that social media is intertwined with many areas of a person's day-to-day existence. The abundance of information available on the internet, as well as the ability to engage with peers or prominent figures through various social media platforms, has fuelled the exponential expansion of social media usage. Hate speech, cyberbullying, and other types of unpleasant language are common in online forums across the world.

Despite the increased convenience that social media has brought to our lives, there have been numerous examples of public personalities being subjected to social media abuse. Hate speech, according Sanguinetti et al [1], is any act that fosters hatred, incites violence, or threatens an individual's safety, dignity, or freedom based on their protected qualities (e.g., race, gender, sexual orientation, etc.).

Even though social media is frequently utilised in hundreds of languages and dialects, most studies on the automated recognition of such information focus on English alone. Bidirectional Encoder Representations from Transformers (BERT) is an acronym for Bidirectional Encoder Representations [1]. It's intended to use both left and right context conditioning to pre-train deep bidirectional representations from unlabeled text. It also offers DistilBERT, a distilled version produced by HuggingFace's team, in addition to BERT[2].

Here we use DistilBERT to train a model that will detect hate speech in codemix language. This approach is novel as there is no previous work done in code mix language using DistilBERT. The

978-1-6654-6883-1/22/$31.00 ©2022 IEEE

earlier researches [5] use various NLP techniques to detect hate speech.

The organisation of the paper is as follows. In next section related works are discussed followed by description of the dataset. Following section describe the methodology in detail. Final two sections discuss the result and conclusion.

II. RELATED WORK

A. Hate speech detection

Hate speech, which can take the shape of text, photos, or video, is the most common form of damaging content on social media. It is defined as an insult directed at a person or group based on qualities such as race, sexual orientation, origin, nationality, religion, or other factors. Hate speech is a huge hazard to communities, as it instils animosity in young people toward others and encourages criminal conduct or violence against others. As a result, detecting hate content on social media is a required and crucial requirement for social media platforms[4]. For a safer social atmosphere, social media providers work hard to remove this content. Detecting hateful content is one of the most difficult NLP jobs since the content may target/attack individuals or groups based on a variety of attributes using numerous hate terms and phrases.

B. code-mixed language approaches

Code-mixing is the blending of two or more languages or linguistic variations in speech. Some scholars refer to this as "code-mixing." Code mixing is getting more common as the number of language linkages increases. As a result of the phasing out of many languages, they are getting polluted. Multilingualism causes languages to clash, resulting in a varied spectrum of languages. When speakers move between two or more languages in the same discourse, this is known as code switching. It depicts a speaker who starts out speaking in one language and ends up speaking in another[5].

C. BERT based models

Bidirectional Encoder Representations from Transformers (BERT) is a language model based on contextual representations that has been trained on massive amounts of data[1]. BERT is made up of feature extraction layers that include word embedding and a model layer. BERT is the most recent language model, and it provides state-of-the-art outcomes for many NLP tasks when compared to other language models[6]. The BERT word embedding training technique is distinct from other word embedding models. It generates a bidirectional representation of words that can be learned in both directions. Approaches to word embedding like Word2Vec and GloVe[7] only look in one direction, resulting in static word representations that don't change with context[17].

III. DATASETS

Two public datasets have been carefully vetted and are of small size. The properties and creation procedure of each of these datasets are discussed in the following sections.

a) The data [11] in the online social media platform can be found in Google searches and Github repositories.

The annotators have labelled it into pre-defined hate speech categories and subcategories. The samples from Google searches and Github repositories are based on keyword matching with hate-related phrases. These have been marked as hateful and annotated. This task must be completed manually and takes a long time. For code-mixed projects, the same procedure was used (Table2).

b): The dataset [12] used to classify English text is Dynamically Generated Hate Speech Dataset, which contains synthetic training dataset taken from Kaggle and excluded some features like Id, Type, Model_wrong, Round, split, status Db.model_pred , Annotator and changed the label of the dataset from Hate to 1 and Non-hate to 0 ,each labeled as either Hate (has the value 1) or non-hate (has the value 0) (Table1).

class	Training	Testing	Total
Not offensive	592	195	787
offensive	908	305	1213
Total	1500	500	2000

Table1*:Training and test split of english*

class	Training	Testing		Total
Not offensive	1315	328		1643
offensive	185	172		357
Total	1500	500		2000

Table2*: Training and test split of code-mix*

IV. METHODOLOGY

A. Preprocessing

We started the work with English hate speech dataset. We removed user names, numbers, hashtags, URLs, and common emoticons etc. We replaced tokens with their textual equivalents in hashtags that comprise certain tokens without any space between them.(eg:"hatespeech" to "hate speech"). We removed all punctuation marks, unknown unicodes, and unnecessary delimiting letters, and stop words. All text are also converted to lower case[19]. In case of malayalam dataset which is in code-mixed form no preprocessing was performed except that every character was converted from uppercase to lower case.

B. Algorithms used

The BERT algorithm was given the preprocessed dataset. The BERT algorithm comes in a variety of forms. DistilBERT was the one we utilised. It's a smaller, less expensive version of BERT that has 40% of the parameters and 95% of the performance of BERT. The authors of [1] used a smaller language model and knowledge distillation that was pre-trained to achieve equivalent performance on downstream NLP tasks with less inference time. Knowledge distillation is a compression approach based on the student-teacher model, in which the learner (small model) learns the behaviour of the instructor (large model) by distillation loss. The BERT embeddings are used as a training set for a variety of machine learning methods. The algorithms that were used were:

1) Logistic Regression:

LR is a probabilistic classifier that is used to classify data and is one of the traditional machine learning methods [23]. This is essentially the logistic function converted version of linear regression [14]. As a result, it accepts input of real-valued characteristics, multiplies them by a weight, and then feeds the resulting sum to the sigmoid function. Use the logistic function, often known as the logistic function [15], to get the class probability. Based on the threshold value, a decision is taken. Since neural networks may be thought of as a stack of several LR classifiers, they are closely related to each other. Naive Bayes is a generative classifier, whereas LR is a discriminative classifier [25]. LR seems to be more resilient to linked traits while Nave Bayes expects great conditional independence. It means that the weight W will be split among the characteristics as W1, W2, and W3 in accordance if there are several features that are absolutely connected, such as F1, F2, and F3.

2) Support vector machine (SVM):

Support Vector Machines (SVMs) are a supervised machine learning technology that may be used for both classification and regression. The goal of an SVM is to find the hyperplane in an N-dimensional space that categorizes data points clearly [16]. It means that this strategy clearly distinguishes between data points that belong to a given category and those that do not. This is true for any data that is encoded in a vector format. As a result, we can use SVM to acquire the answers we desire if we can generate acceptable vector representations of the data we have [16].

3) Multinomial Naive bayes:

The naive assumption of conditional independence of features serves as the foundation for this Bayesian classifier. Since real data cannot be independent of other inputs, this implies that each input is independent of the others[20]. For multinomially distributed data, we test a Naive Bayes classifier. Based on an observed occurrence, the Bayes Theorem calculates the likelihood of a future event[11]. The MNB variant of Naive Bayes functions best with text documents. Instead of modelling a text as the presence and absence of individual words as basic naive Bayes does, MNB directly models the word counts and modifies the underlying computations to account for them. As a result, the input text data is treated as a collection of words, with word position being ignored and only the frequency of occurrences (frequency) taken into account.

C. Experiment

Basic pre-processing was completed as described in the preceding section. We split the training data into conventional train-validation splits in the ratio of 80:20 for each of the sub-tasks. DistilBERT analyses the sentence and transfers some of the information it gleans to the next model. The following model is a basic Logistic Regression model that takes the outcome of DistilBERT processing and categorises the sentence as hate or non-hate (1 or 0, respectively). Only the logistic regression model will be trained. since DistilBERT is a pre-trained model. The next phase is tokenization, and the first step is to use the BERT tokenizer to tokenize the sentences. The specific tokens required for sentence categorization are then added ([CLS] at the beginning of the sentence and [SEP] at the end)[16]. The output of the Distilbert model would be a vector for each input token. Each vector has 768 numbers in it (floats). We disregard all but the first vector because this is a corpus categorization exercise (the one associated with the [CLS] token)[21]. In the case of code-mix languages, we use the one vector as input to the logistic regression model, and in the case of English datasets, we use the Ensemble model. This input is then used by the logistic regression and Ensemble models respectively to classify the vector based on what they learnt during their training phase.

The training set is fitted to each weak learner, and predictions are obtained. The ultimate prediction result is calculated by integrating all of the weak learners' scores[18].

In this approach, voting is hard so each model's prediction is a vote. In maximum voting, the final result is determined by the prediction that receives the most votes.

E. Model Distillation:

While most past research has focused on employing distillation to generate task-specific models, we show that a BERT model can be lowered by 40% using knowledge distillation during the pre-training phase. As Transfer Learning from large-scale pre-trained models grows more prominent in Natural Language Processing, operating enormous models on-the-edge and/or under restricted computational training or inference constraints remains difficult (Fig1). It introduces DistilBERT, a method for pre-training a smaller general-purpose language representation model that can then be fine-tuned to perform effectively on a range of tasks, in the same way that larger versions can [24]. While the majority of past research has focused on employing distillation to generate task-specific models, this study use knowledge distillation during the pre-training phase[22]. It's a lighter, faster version of BERT with similar performance

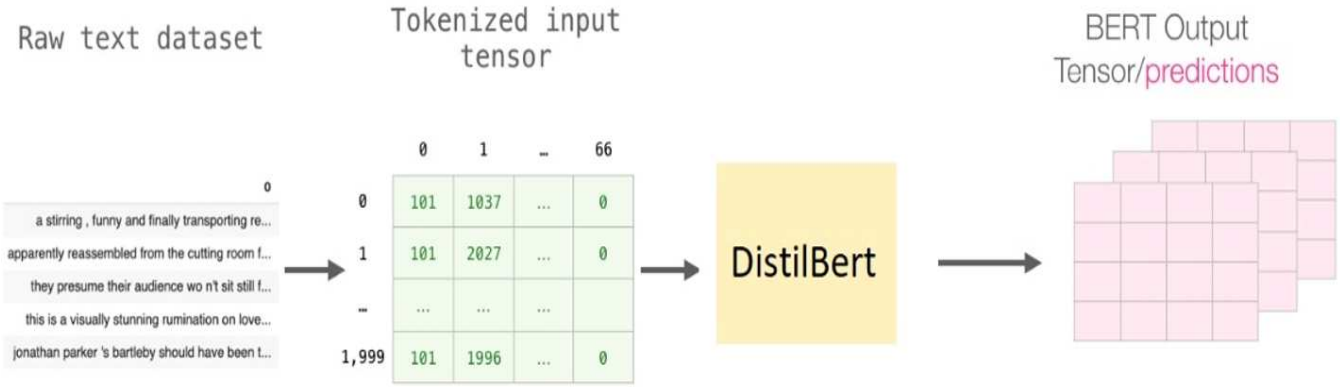

Fig1: *Processing with DistilBERT*

D. Ensemble:

Ensemble learning is the use of many machine learning models to solve a single problem. Weak learners are the name given to these models. The idea is that if numerous weak learners are together, they can become strong learners.

V. RESULTS

In comparison to other models, the logistic regression model augmented with the BERT model produced relatively acceptable results. The

BERT model is a cutting-edge, high-performance model. Among all the models, it is the most accurate.

MODEL	ACCURACY
SVM	90.4
Naïve Bayes	80.8
Logistic Regression	90
Random Forest	55.5
Ensemble Model	93

Table3:*Accuracy score of english dataset*

The maximum accuracy score for the Code-mix dataset is presently 90% using Logistic Regression(Table4), with DistilBert and an accuracy score of 93 percent when using the English dataset as an input of Ensemble model with DistilBert Ensemble model is the combination of Svm, Naïve Bayes, Logistic Regression and Random Forest (Table3).

MODEL	ACCURACY
Logistc Regression	85
Logistc Regression(Grid Search)	90.5
SVM	78
Naïve Bayes	84

Table4:*Accuracy score of code-mix dataset*

VI. CONCLUSION AND FUTURE WORK

Finally, for hate speech detection, DistilBERT architecture provides an effective feature extraction and classification approach. DistilBERT combines the advantages of domain-agnostic and domain-specific word embedding by training the model on a large amount of data before adding an extra layer that is trained on domain-specific data

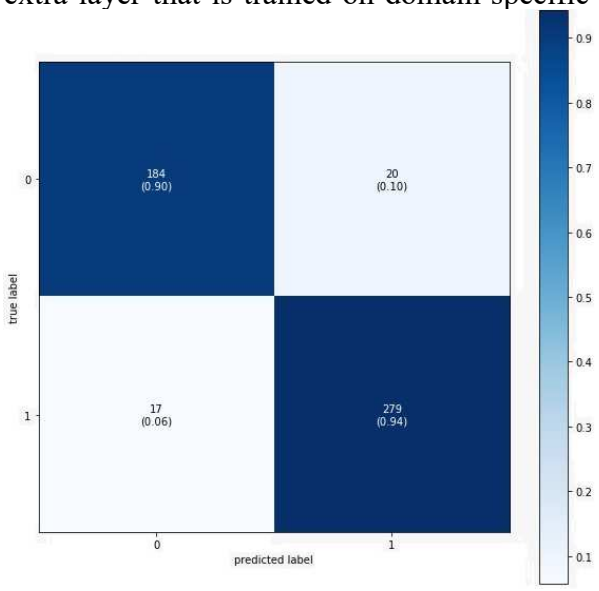

Fig2: *confusion matrix of English dataset*

(fine-tuning). DistilBERT also saves time and effort when creating an embedding model from the groundup.

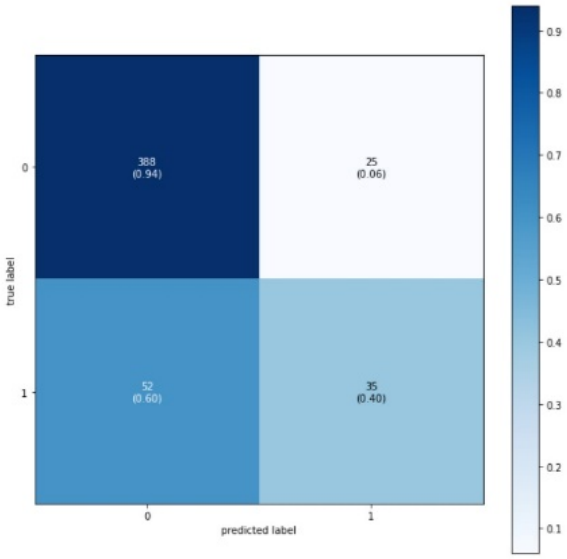

Fig3:*confusion matrix of code-mix*

The goal of this research was to develop a model that could be trained on a huge dataset of multiple languages. We demonstrate that our models attain equivalent or greater performance to a wide range of baseline monolingual models by experimenting on an aggregated dataset containing three datasets in English, Malayalam, and Code-mixed Malayalam. For future research, this study can be expanded to detect multi-class hate speech (Fig 2,3).

VII. References

[1] Poletto, Fabio, et al. "Resources and benchmark corpora for hate speech detection: a systematic review." Language Resources and Evaluation 55.2 (2021): 477-523.

[2] Koroteev, M. V. "BERT: a review of applications in natural language processing and understanding." arXivpreprintarXiv:2103.11943 (202).

[3] Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv preprint arXiv:1910.01108 (2019).

[4] Jahan, Md Saroar, and Mourad Oussalah. "A systematic review of Hate Speech automatic detection using Natural Language Processing." arXiv preprint arXiv:2106.00742 (2021).

[5] Ahmad, Gazi Imtiyaz, et al. "Machine Learning Techniques for Sentiment Analysis of Code-Mixed and Switched Indian Social Media Text Corpus: A Comprehensive Review." Machine Learning 13.2 (2022)

[6] Rana, Aneri, and Sonali Jha. "Emotion Based Hate Speech Detection using Multimodal Learning." arXiv preprint arXiv:2202.06218 (2022).

[7] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[8] Saurabh Gaikwad, Tharindu Ranasinghe, MarcosZampieri,ChristopherM.Homan(2021), arXiv:2109.03552v1 [cs.CL] 8 Sep 2021

[9] Mutanga, Raymond T., Nalindren Naicker, and Oludayo O. Olugbara. "Hate speech detection in twitter using transformer methods." International Journal of Advanced Computer Science and Applications 11.9 (2020).

[10] Vashistha, Neeraj, and Arkaitz Zubiaga. "Online multilingual hate speech detection: experimenting with Hindi and English social media." Information 12.1 (2020): 5.

[11] Kanakaraj, Monisha, and Ram Mohana Reddy Guddeti. "Performance analysis of Ensemble methods on Twitter sentiment analysis using NLP techniques." Proceedings of the 2015 IEEE 9th international conference on semantic computing (IEEE ICSC 2015). 2015.

[12] https://www.kaggle.com/datasets/usharengaraju/dynamically-generated-hate-speech-dataset

[13] Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv preprint arXiv:1910.01108 (2019).

[14] Park, H. (2013). “An introduction to logistic regression: From basic concepts to interpretation withparticular attention to nursing domain.” Journal of Korean Academy of Nursing, 43(2), 154–164.

[15] Shah, Kanish, et al. "A comparative analysis of logistic regression, random forest and KNN models for the text classification." *Augmented Human Research* 5.1 (2020): 1-16.named entity recognition using support vector

[17] de Sousa, João Guilherme Routar. "Feature extraction and selection for automatic hate speech detection on Twitter." (2019).

[18] Ganaie, M. A., and Minghui Hu. "Ensemble deep learning: A review." *arXiv preprint arXiv:2104.02395* (2021).

[19] Jamdade, Priya, et al. "Hateful Speech Detection on Social Media using Deep Learning: An Overview."

[20] Mohiyaddeen, Mr, and Sifatullah Siddiqi. "Automatic hate speech detection: A literature review." *Available at SSRN 3887383* (2021).

[21]https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

[22]https://huggingface.co/docs/transformers/model_doc/distilbert

[23] Genkin, Alexander, David D. Lewis, and David Madigan. "Large-scale Bayesian logistic regression for text categorization. 49.3 (2007): 291-304.

[24] Arora, Gaurav. "Gauravarora@ hasoc-dravidian-codemix-fire2020: Pre-training ulmfit on synthetically generated code-mixed data for hate speech detection." *arXiv preprint arXiv:2010.02094* (2020).

[25] Winkler, William E. *Methods for record linkage and bayesian networks*. ,Statistical Research Division, US Census Bureau, Washington, DC, 2002.