,

Project Report on
Hate Speech Detection using BERT Based Models

by

Racha Adithyavardhan	*420225*
Nandam Sai Saketh	*420212*
Machkuri Dishanth	*420202*

Under the guidance of

Mr. D. Prasad

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH
TADEPALLIGUDEM-534101, INDIA

MAY-2023

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH

Certificate

It is certified that the work contained in the thesis titled “HATE SPEECH DETECTION USING BERT BASED MODELS” by “Racha Adithyavardhan, bearing Roll No: 420225”, “Nandam Sai Saketh, bearing Roll No: 420212” and “Machkuri Dishanth, bearing Roll No: 420202” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Mr. D. Prasad
Computer Science and Engineering
N.I.T. Andhra Pradesh
May 2023

Place: Tadepalligudam
Date:

May 10, 2023

Table of Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

Notation	Meaning
i.e.,	that is
etc.,	etcetera
NLP	Natural language processing
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
BiLSTM	Bidirectional Long Short-Term Memory
HOF	Hate or Offensive
NOT	Not Hate, Offensive

Abstract

Hate speech has become a prevalent issue on social media platforms as individuals use these platforms to express discriminatory and abusive opinions. Detecting and combating hate speech is a growing challenge, particularly in the context of social media and online platforms, where it can spread rapidly and anonymously. The main objective of our work is to detect hate speech in the Hindi language. We have used Hasoc 2021 acombinedHasoc (Hasoc2019, Hasoc2020, Hasoc2021) datasets to evaluate our models. Our study describes adding deep learning models as a classification layer to pre-trained models can improve the performance and give better results. We developed a model BERT-CNN which is a combination of fine-tuning with BERT and passing through CNN layers for hate speech detection. We have also described how we can add other deep learning models like bi-LSTM in combination with BERT. The proposed model BERT-CNN performs better than the base model with an F1 score - 0.81 whereas mBERT F1 score - 0.75 .

Chapter 1

Introduction

Social Media is a platform where many users can openly express their thoughts and opinions. Online hate speech can be of many forms including threatening or harassing messages or offensive comments. Hate speech can target specific groups based on their gender, orientation, race, religion, ethnicity, etc. Online hate speech can be a problem since it can lead to real-world harm and can make people feel excluded or unsafe. According to a study in India, online risks such as hate speech have a huge increase from previous years. From 2016 to the present it almost got doubled and increased to 26%. Many social media platforms use fast checkers which detect hate speech and remove such content to provide a positive and better environment. However, they are often biased and sometimes used to target political opponents. So one of the effective ways to handle this is to use ML or deep learning-based detection system. Hate speech detection is a Natural Language Processing(NLP) task. As the introduction of pre-trained language models such as BERT, GPT, T5, etc has achieved state-of-the-art on various NLP tasks by providing a powerful and flexible tool for understanding and analyzing the data. These BERT-based transformer models can identify patterns and contexts that are indicative of hateful language.

Our project aims to detect hate speech content in the Hindi language. Our task is to classify tweets into two categories hate-speech(HOF) and not hate-speech (NOT). In our project, hate speech classification is done by fine-tuning the multilingual-BERT trained on a large corpus of text data from multiple languages to our classification task. We are adding deep learning techniques such as CNN and BiLSTM to the multilingual BERT model as a classification layer to enhance and improve the performance of the model. These models are fine-tuned and evaluated on the Hasoc data set. Our evaluation metrics for evaluating the performance of the model are Accuracy and F1 Score.

Chapter 2

Literature Survey

Hate Speech Detection in Hindi Language using BERT and Convolution Neural Network published in 2022,[1] discusses detecting hate speech in the Hindi language using the Hasoc 2020 and Hasoc 2021 datasets. They used a model that combined BERT and Convolutional Neural Network (CNN). The BERT encoder was used to transform the text data into contextualized embeddings. They used the last four layers of BERT as their target output and generate embeddings. The results of the experiments conducted by the authors showed promising results for hate speech detection in the Hindi language using BERT and CNN. Their approach achieved an accuracy of 80.47% on the Hasoc 2020 dataset and 75.43% on the Hasoc 2021 dataset. Their work demonstrated the effectiveness of using contextualized embeddings and a CNN for hate speech detection in the Hindi language.

A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media discusses the use of the BERTbase model for hate speech detection in two datasets: Waseem and Hovey and Davidson et al.[2] The goal was to classify each tweet as either racism, sexism, neither, or hate/offensive. The paper explores several models, including BERT base, BERT-Nonlinear layers, BERT-CNN, and BERT-CNN. In the BERT-CNN model, the outputs of all transformer encoders are used instead of using the output of the latest transformer encoder.

Hate Speech Targets Detection in Parler using BERT presents a pipeline for detecting hate speech and its targets in Parler social media platform.[3] The pipeline is composed of two models: one for hate speech detection and the other for target classification. Our work focuses on four main minorities, including People of color, Muslims, Jews, and LGBT. To accomplish this task, we utilized an annotated Parler dataset provided by Israeli and Tsur for the hate speech detection task, while HateXplain and Dialoconan datasets were used for the target detection task. These two datasets were used for training, and the Toxigen dataset and a new Target Annotated Parler dataset, which we annotated, were used for evaluation. Our paper offers a unique approach to identifying hate speech and its targets in Parler, which has been a controversial platform for extremist content. By fine-tuning BERT on different datasets, we were able to accurately classify the targeted minority groups and provide insights into the distribution of hate speech in Parler.

HATECHECK: Functional Tests for Hate Speech Detection Models published in 2022, addresses the challenge of detecting multilingual hate speech in social media.[4] To evaluate the performance of hate speech detection models, the authors used the HateCheckHIn dataset, which contains code-mixed text in multiple languages and covers a variety of hate speech types. They employed the m-BERT model for the evaluation. In addition, the authors used two publicly available datasets to assess the generalizability of their approach: the Mandl et al. (2021) dataset, which consists of 6,126 tweets annotated as hateful, offensive, profane, or neither, and the Bhardwaj et al. (2020) dataset, comprising 8,192 tweets annotated as fake news, hate speech, offensive, defamation, or non-hostile. These datasets provide a diverse range of hate speech types and help to evaluate the performance of models in different contexts.

Large Annotated Dataset for Multi-Domain and Multilingual Hate Speech Identification paper presents a new multilingual hate speech analysis dataset for English, Hindi, Arabic, French, German, and Spanish languages for multiple domains across hate speech - Abuse, Racism, Sexism, Religious Hate, and Extremism.[5] This paper describes how the dataset is created and annotated at a high level and low level for different domains and how they are used to test the current state-of-the-art multilingual and multitasking learning approaches approaches.

Chapter 3

Methodology

3.1 BERT Architecture

Given input text is first tokenized into subwords using a workpiece (subword) tokenizer. We add special tokens [CLS] and [SEP] to the input sequence. [CLS] is added at the beginning of the text and used for representing contextual information.

[SEP] is added at the end of every sentence. Then these tokens are converted to IDs. BERT expects all input sequences to be of the same length, so the input sequence is padded with zeros to the maximum length of the sequence. Attention mask ensures that real word tokens have mask 1 and padded will have mask 0. These token ids and attention masks are then fed to the BERT model for encoding.

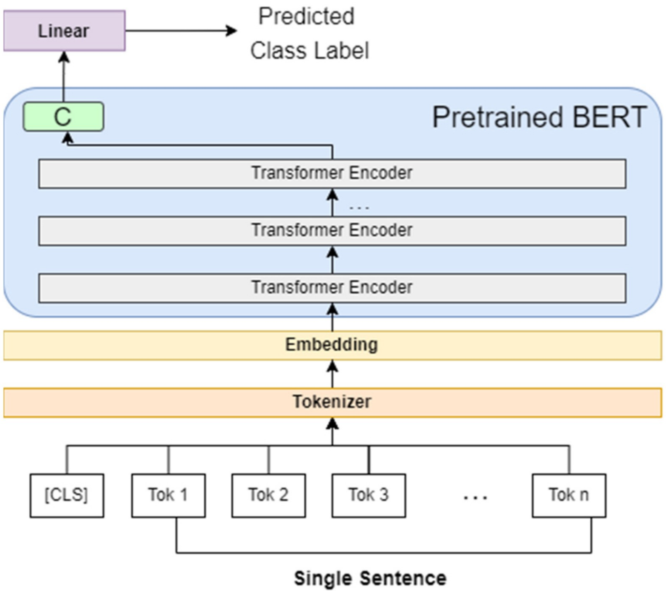


Figure 3.1: Bert Architecture

These are passed through several transformer layers. The BERT base contains 12 transformer encoder layers, each layer has multi-head attention and feed-forward network sub-layers. Each transformer layer takes in the hidden states from the previous layer as input and outputs a new set of hidden states of the same dimension. The encoded representation of the [CLS] token which represents the entire input sequence is passed through a linear layer to obtain a fixed-size representation of the input sequence which can be passed through the soft-max layer to get the final outputs.

3.2 Fine Tuning Methods

3.2.1 *Bert base with BiLSTM*

In this model, we use the BERT embeddings obtained by passing our input sequences to the BERT model and we pass the last hidden layer output to a bi-directional LSTM layer, which processes the sequence in both forward and backward directions to capture temporal dependencies between the tokens. LSTM output is a sequence of hidden states, one for each token in the input sequence.

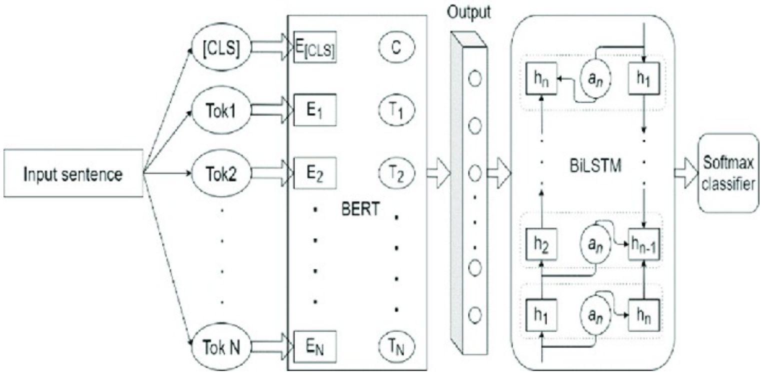


Figure 3.2: BERT BiLSTM

LSTM networks are a type of recurrent neural network capable of learning long-term dependencies. It consists of 3 gates - input, forget, output gates, and a memory cell. Each hidden layer takes the hidden state output from the previous LSTM cell as input and produces a new hidden state as output. A bidirectional-LSTM processes the input sequence both forwards and backward through time. Here the current hidden state has representations from previous hidden states from both forward and backward computation. This allows a network to capture information from both past and future contexts. The last hidden state is passed through a softmax layer to predict the outputs.

3.2.2 BERT Base with CNN

This model utilizes the last hidden state(Hn) from the layers of BERT. The last hidden state is the encoded output that contains contextualized representations of the input text. This encoded output is reshaped and passed through a stack of convolutional layers. Each convolution layer has different kernel sizes that capture local features and patterns in the text. The output from the convolution layer is then passed through a max pooling layer to obtain the maximum value across the feature map for each channel. This reduces the dimensionality of the output and preserves the most important information.

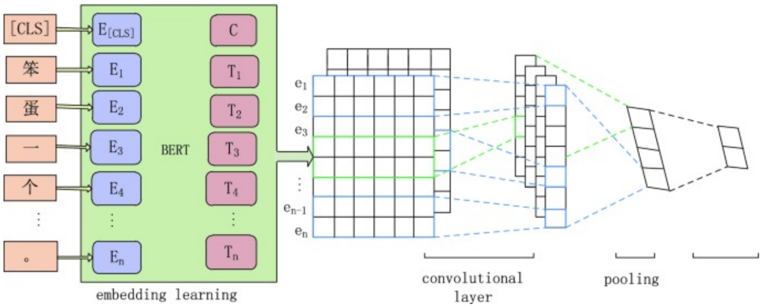


Figure 3.3: BERT CNN

The output from the pooling layer is passed to a fully connected layer to produce the final output logits. The final output logits are passed through a softmax activation function to produce a probability distribution over the classes. The class with the highest probability is the predicted class i.e. Hate or Not Hate for the input text.

Chapter 4

Experimental Setup and Results

4.1 System Architecture

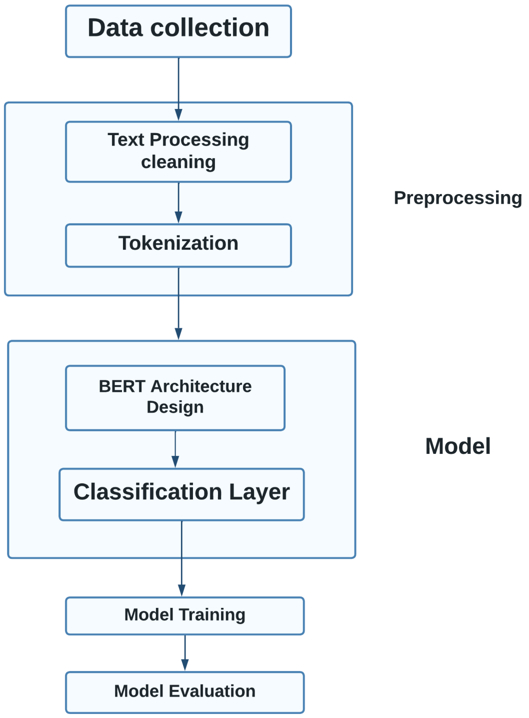


Figure 4.1: System Architecture

4.1.1 Dataset Overview

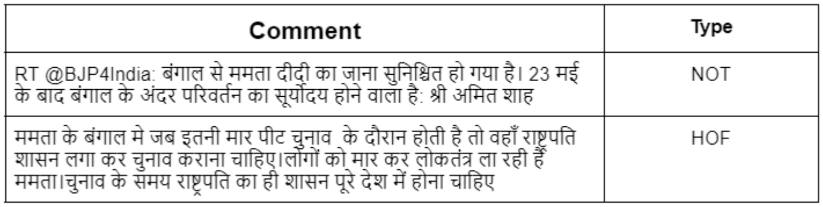
For our project, we worked with the Hasoc dataset. The Hasoc dataset is available in several languages including English, German, Hindi, and other languages. The dataset has been extracted from various social media platforms like Twitter and Youtube. The dataset includes 2 sub-tasks -

- Task A - Binary Hate Speech Detection (HOF/NOT) - HOF - Hate or Offensive NOT - Not hate, offensive
- Task B - Multilabeled Hate Speech Detection (HATE, OFFN, PROFANE)

DATASET	HOF	NOT	TOTAL
Hasoc 2020	847	2116	2963
Hasoc 2021	1433	3161	4594
Combined Hasoc Dataset (Hasoc 2019,2020,2021)	5354	8186	13345

Table 4.1: Dataset

In our project, we use the task A dataset for binary classification. For this, we have considered Hasoc 2021 Dataset and a combined dataset(Hasoc 2019,2020,2021).



Comment	Type
RT @BJP4India: बंगाल से ममता दीदी का जाना सुनिश्चित हो गया है। 23 मई के बाद बंगाल के अंदर परिवर्तन का सूर्योदय होने वाला है: श्री अमित शाह	NOT
ममता के बंगाल मे जब इतनी मार पीट चुनाव के दौरान होती है तो वहाँ राष्ट्रपति शासन लगा कर चुनाव कराना चाहिए।लोगों को मार कर लोकतंत्र ला रही हैं ममता।चुनाव के समय राष्ट्रपति का ही शासन पूरे देश में होना चाहिए	HOF

Figure 4.2: Dataset sample

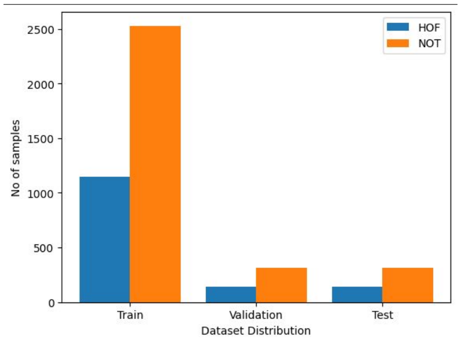


Figure 4.3: Dataset splitting of Hasoc 2021

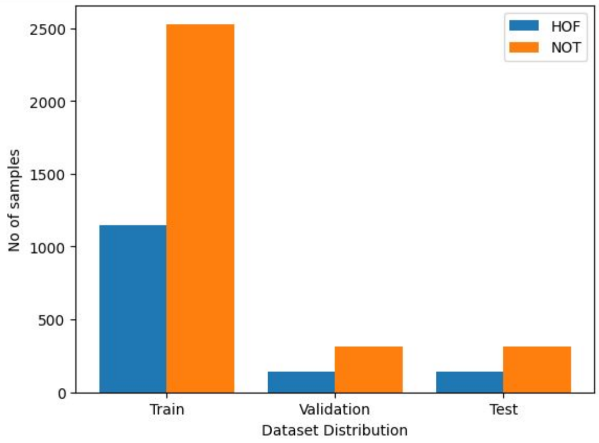


Figure 4.4: Hasoc Combined Datasets

4.1.2 Preprocessing

The first step in our work involved preprocessing, which aims to remove unnecessary elements and clean the data from the dataset. This stop includes:

- Removing of usernames
- Removeing punctuations
- Removing stopwords in Hindi

We retained the emojis and hashtags since they provide information for hate speech classification.

4.1.3 Implementation

4.1.3.1 BERT

The first step of implementation is adding the last classification layer to the pre-trained BERT model. We used the logits layer as the final classification layer. Then it is trained on the dataset. The input text is converted into corresponding tokens using a tokenizer and passed to BERT as input. The maximum input sequence length of BERT is 512 tokens. During training, the weights of the BERT model and the classification head are updated using backpropagation based on the error between the predicted output and the true labels. The gradients of the loss function are computed and used to update the weights using an Adam optimization algorithm. Once training is complete, the model can be evaluated on a separate validation set to determine its performance on the text classification task. The number of epochs this model trained on is 8.

4.1.3.2 BERT CNN

The last hidden layer of the BERT is unsqueezed to increase the tensor dimension and passed through a list of convolution layers of different kernel sizes(1 and 2 in our work). Each convolutional layer has a specified number of filters. i.e. 3, which produces feature maps. Then a ReLU activation function is applied to each feature map produced by the convolutional layer. The dimension is squeezed and passed to the 1-D max-pooling layer as input. Max-pooling operation is applied along all the layers and they are concatenated along with the second dimension. A dropout layer is applied to the concatenated tensor to prevent overfitting. The final linear layer produces a tensor of output classes. A softmax activation function is applied to the output tensor to convert the logits to a probability distribution over the output classes.

4.1.3.3 BERT BiLSTM

The pre-trained BERT model processes the input sequence and produces a tensor of output embeddings, which is then passed through an LSTM layer. The LSTM layer processes the sequence of embeddings and produces a tensor of hidden states that represent the learned context of the sequence. The LSTM layer has an input_size of 768, which is the size of the BERT output embeddings, hidden_size of 768, which determines the size of the hidden state vectors, and num_layers of 1, specifies the number of LSTM layers to use. We are using bidirectional LSTM in which the LSTM layer processes the input sequence in both directions and concatenates the results. The output of the BiLSTM layer is then passed through a linear classifier layer, which maps the hidden state tensor to a tensor of shape (batch_size, 2), where the second dimension represents the two possible classes in the hate speech detection task. Finally, a softmax activation function is applied to the output tensor to obtain a probability distribution over the two classes.

4.1.4 Loss Function

The cross-entropy loss function is used in the backpropagation step for training the model. The cross-entropy loss measures the difference between the predicted probability distribution and the actual probability distribution of the target labels. The equation for cross-entropy loss is:

$$\text{Cross-Entropy Loss = -}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log(p_{i,j})$$

Here y is the ground truth and p is the predicted output of the model. In our classification number of classes, C is 2 (Hate or Not Hate).

4.1.5 Evalution Metrics

We evaluated the model performance using Accuracy and F1-Score. F1-Score is the harmonic mean of precision and recall, calculated as follows:

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

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

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

Accuracy measures the proportion of correct predictions among all predictions, calculated as follows:

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP}$$

4.2 Results

We have implemented hate speech using Bert, BertCNN models. For finetuning Bert we have run it for 8 epochs and BertCNN for 4 epochs which give the following results -

	Accuracy	**F1**
BERT	0.78	0.75
BERT CNN	0.70	0.81

Table 4.2: Models Result on Hasoc 2021 dataset

We have also trained over models on the HasocCombined Dataset which is made by concatenating Hasoc datasets(Hasoc 2019,Hasoc 2020,Hasoc 2021). For this HasocCombined Dataset, the BERT model gives an accuracy of 0.80 and F1 score of 0.804.

4.2.1 Error Analysis

In this project, we conducted an error analysis on a combined dataset of 13,345 comments and a Hasoc 2021 dataset of 4,594 comments to improve the performance of a binary classification model for hate speech detection. The dataset contained a mix of hate speech and non-hate speech comments, and the goal was to develop a model that accurately classified each comment as either hate speech or non-hate speech. First, we split the dataset into training and validation sets, with 80% of the data used for training, 10% used for validation, and 10% used for testing. We trained our model on the training set and evaluated its performance on the test set using metrics such as accuracy and F1-score.

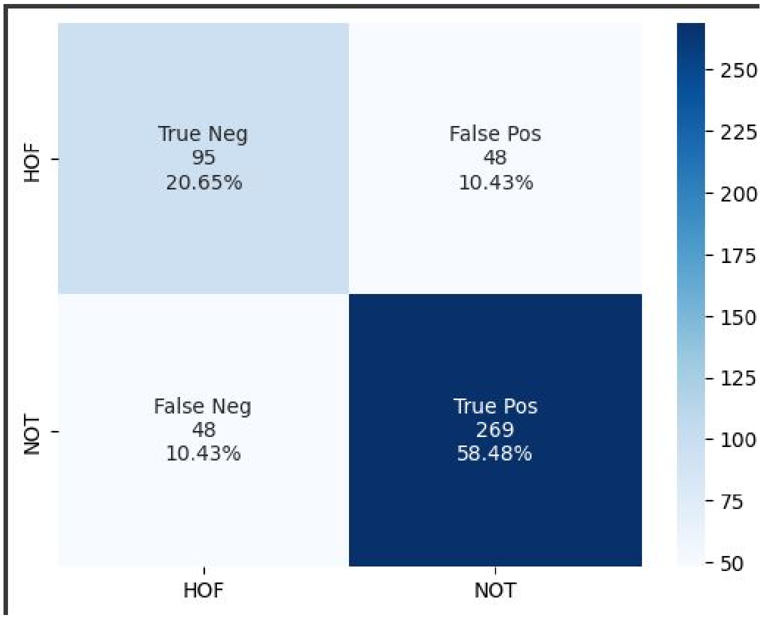


Figure 4.5: Error Analysis on BERT

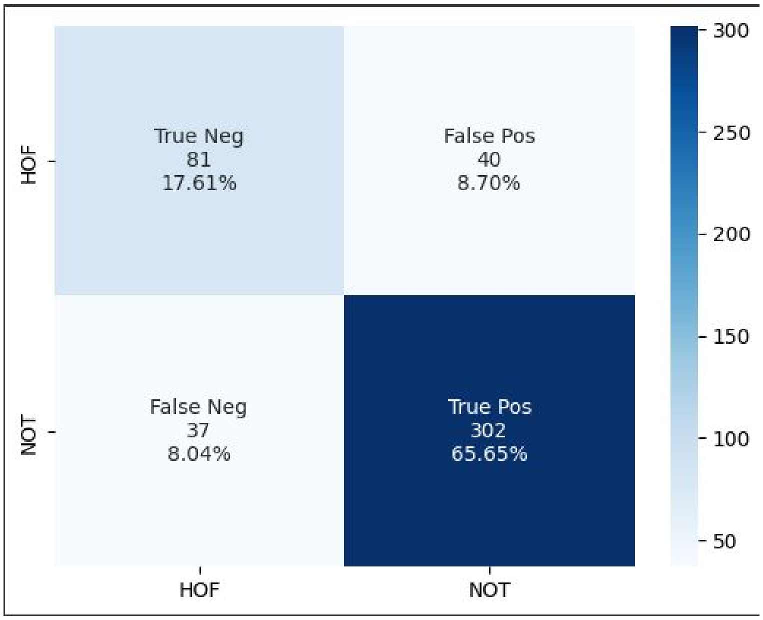


Figure 4.6: Error Analysis on BERTCNN

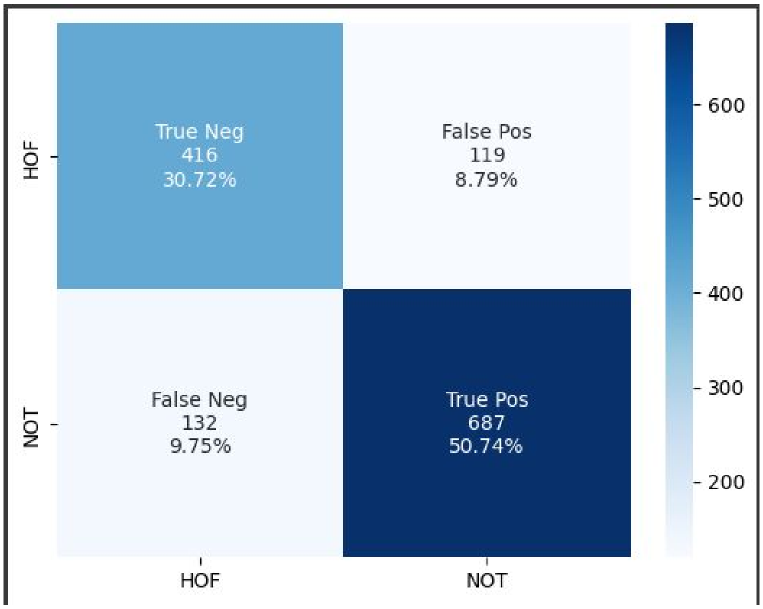


Figure 4.7: Error Analysis on HasocCombined Dataset(Hasoc 2019, Hasoc 2020, Hasoc 2021)

Chapter 5

Conclusion

Hate speech detection is an important task and continues to be a societal problem. In this project, we have described how deep learning models can be added as a classification layer for the BERT model. We have used mBERT with CNN as a classification layer. Our results demonstrate that BERT-based models can effectively capture semantic and contextual information and outperform traditional methods. We have also described how BERT can be combined with biLSTM for our task.

Chapter 6

Future Scope

Hate speech detection is an important task and research continues in this field. Adding extra layers to BERT is not only limited to CNN, and bi-LSTM. Other deep learning techniques can also be incorporated into our task. We can extend this to other low-resource languages like Telugu and detect hate speech in code-mixed language datasets. Further, our proposed method can also be used for detecting fine-grained hate speech where hate speech can be targeted to a particular group, race, gender, etc are detected.

Bibliography

[1] Shubham Shukla, Sushama Nagpal, and Sangeeta Sabharwal, “Hate Speech Detection in Hindi language using BERT and Convolution Neural Network” Netaji Subhas University of Technology Sector-3 Dwarka, Delhi, India.

[2] Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi “A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media” Telecom SudParis, Institut Polytechnique de Paris, Evry, France.

[3] Nadav Schneider, Shimon Shouei, Saleem Ghantous, and Elad Feldman “Hate Speech Targets Detection in Parler using BERT”.

[4] Mithun Das, Punyajoy Saha, Binny Mathew, and Animesh Mukherjee “HateCheckHIn: Evaluating Hindi Hate Speech Detection Models” Indian Institute of Technology, Kharagpur West Bengal, India – 721302

[5] Ankit Yadav, Shubham Chandel, Sushant Chatufale and Anil Bandhakavi, “LAHM: Large Annotated Dataset for Multi-Domain and Multilingual Hate Speech Identification“ ogically.ai, Brookfoot Mills, Brookfoot Industrial Estate, Brighouse, HD6 2RW, United Kingdom

[6] T. Mandl et al., “Overview of the Hasoc Subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages under Creative Commons License Attribution 4.0 International (CC BY 4.0),” 2021.

[7] A. Schmidt and M. Wiegand, “A Survey on Hate Speech Detection using Natural Language Processing,” Soc. 2017 - 5th Int. Work. Nat. Lang. Process. Soc. Media, Proc. Work. AFNLP SIG Soc., pp. 1–10, 2017, doi: 10.18653/V1/W17-1101.