MULTICLASS CLASSIFICATION OF BANGLA CYBERBULLYING USING BI-LSTM AND BANGLABERT

By

Md. Asaduzzaman

ID: M-200305019

Batch: 8th

Session: 2020-2021

Supervised & Co-supervised By

Dr. Ujjal Kumar Acharjee
Professor
Department of CSE
Jagannath University

Dr. Md. Manowarul Islam
Associate Professor
Department of CSE
Jagannath University

University in partial fulfillment of the requirements for the degree of

MASTERS OF SCIENCE

IN

COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering

Jagannath University

Dhaka, Bangladesh

September, 2024

Candidate's Declaration

This is to certify that the work presented in this project entitled, "Multiclass Classification Of Bangla Cyberbullying Using Bi-LSTM And BanglaBERT", is the outcome of the research carried out by Md. Asaduzzaman under the supervision of Dr. Ujjal Kumar Acharjee, Professor & co-supervision of Dr. Md. Manowarul Islam, Associate Professor, Computer Science and Engineering Department, Jagannath University, Dhaka-1100, Bangladesh.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma, or other qualifications.

.....
Dr Ujjal Kumar Acharjee
Supervisor

.....
Dr. Md. Manowarul Islam
Co-supervisor

.....
Md. Asaduzzaman
Candidate

Dedication

Dedicated
To my beloved parents

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgement

First of all, I would like to express my eternal gratitude to the Almighty Allah for the special blessing leading to the writing of this report.

I am extremely thankful and want to pay gratitude to my supervisor Dr. Ujjal Kumar Acharjee, Professor & Co-supervisor Dr. Md. Manowarul Islam, Associate Professor, Department of Computer Science and Engineering, Jagannath University, Dhaka, Bangladesh for their effective guidance, encouragement, suggestion, cooperation, and valuable advice and special supervision in the preparation of this paper. His willingness to give his time so generously has been very much appreciated.

Before I get into thick things I would like to add a few heartfelt words for the people who are part of this project in numerous ways. People who give unending support right to stage the project idea are convinced.

I also acknowledge with a deep sense of reverence, my gratitude towards my parents, and other numbers of my family and friends, who have always supported me morally.

Last but not least, I would like to thank everyone I learned something from throughout my life.

Abstract

The deep learning pipeline presented in this work is specifically designed to classify poisonous comments in Bengali. An exact description of the toxicity is determined using a multi-label classifier after a binary classification model determines whether a comment is harmful. We created a carefully curated dataset comprising 16,073 comments, with 8,488 identified as toxic. Toxic comments are categorized into one or more of the following six types: vulgar, hateful, religious, threatening, trolling, and insulting.Using a hybrid Convolutional Neural Network and Bidirectional Long Short-Term Memory (CNN-BiLSTM) model with BanglaBERT mechanism, the pipeline achieved a state of the earth model accuracy of 79.03% . Futhermore the model has shown excellent result on each of the 6 class label with accuracy 94.37%, 93.64%, 97.05%,93.79%, 92.03%& 88.45% for the following 6 categorized class labels.

Chapter 1

Introduction

Nowadays, all it takes to become tightly connected is the click of a button or social network icon. They are simultaneously exposed to the weaknesses brought about by cyberbullying. As a result, our society has imposed a difficult task on us: identifying cyberbullying in social media at an early stage.Early detection of cyberbullying will enable us to lessen its harmful effects and provide victims with the most expedient support. It will be possible to stop aggressors from posing serious risks to mental health by quickly identifying them. Additionally, early cyberbullying detection will assist the social organization in successfully rehabilitating both the victim and the predator.

1.1 Cyberbullying in Social Media

Cyberbullying is a significant and increasingly prevalent issue in our digital age, where the boundaries between private and public spaces have become blurred. With the rapid advancement and widespread use of information and communication technology, individuals are now more connected than ever before. This connectivity, while offering numerous benefits, also creates opportunities for harmful behaviors, including cyberbullying.

Cyberbullying involves using digital platforms—such as social media networks, messaging apps, forums, and e-commerce websites—to harass, intimidate, or demean others. The anonymity and reach provided by the internet can amplify the impact of offensive remarks and threats, making them more pervasive and, at times, more damaging than traditional forms of bullying. The nature of cyberbullying can vary widely, including behaviors such as:

1. **Harassment:** Persistent, offensive, and aggressive messages or comments directed at an individual, often aimed at causing emotional distress.

2. **Impersonation:** Pretending to be someone else online to spread false information or create confusion and distress.

3. **Doxxing:** Publicly revealing private or personal information about someone without their consent, such as home addresses or phone numbers, with the intent to intimidate or harm.

4. **Exclusion:** Deliberately excluding someone from online groups, social media circles, or other digital interactions as a means of social isolation.

5. **Trolling:** Posting inflammatory or provocative comments to elicit reactions or disturb others, often in a way that belittles or mocks the victim.

6. **Spreading Rumors:** Disseminating false or misleading information about someone to damage their reputation or relationships.

The impact of cyberbullying on individuals, particularly on mental health, is profound. Victims often experience heightened levels of anxiety, stress, and depression. The constant presence of digital devices means that the harassment can follow individuals everywhere, extending the reach and impact of the bullying beyond traditional boundaries. This persistent exposure can lead to severe psychological effects, including:

- **Increased Anxiety:** Victims may become fearful of using digital platforms or interacting with others online, leading to significant stress and anxiety about potential further harassment.

- **Depression:** The emotional toll of being targeted can lead to depressive symptoms, including feelings of hopelessness and worthlessness.

- **Social Isolation:** Victims might withdraw from social interactions, both online and offline, due to fear of further harassment or a diminished sense of self-worth.

- **Decreased Self-Esteem:** Repeated negative reinforcement and derogatory comments can erode an individual's self-esteem and self-confidence.

1.2 Background Scenario

Existing research [1] has demonstrated a positive relationship between engaging in cyberbullying and increasing social distancing-related loneliness. People are becoming more and more accustomed to using social media for online communication, which increases their vulnerability to becoming victims of cyberbullying. Statistics from a major mobile operator show that in Bangladesh, 49% of students experience cyberbullying and harassment, either as a result of being the target of other people's remarks or as an active participant in bullying behavior. [2]. Women in Bangladesh between the ages of 14 and 22 make up 80% of all victims of cyberbullying, according to a UNDP survey conducted during the COVID-19 pandemic. Unexpectedly, the majority of cybercriminals are between the ages of 14 and 17. Previous studies have demonstrated a strong correlation between the detrimental effects of cyberbullying and problems with depression, anxiety, suicidal ideation rates, teenage violence, and physical and mental health. Typically, the remarks made by attackers have the potential to cause racism, body shaming, sexual harassment, or trolling of any public figure or celebrity. Even so, the victims' lives may become less harmonious as a result of the bullying, which may have a long-term negative impact on their mental health. In addition, the family and society at large face difficulties in helping these traumatized victims get well. Thus, the victims appear to be overly sensitive, alone, and worthless. This causes discord and imbalance in our society, which could encourage the emergence of more predators who would disseminate cyberbullying like a "social cancer."

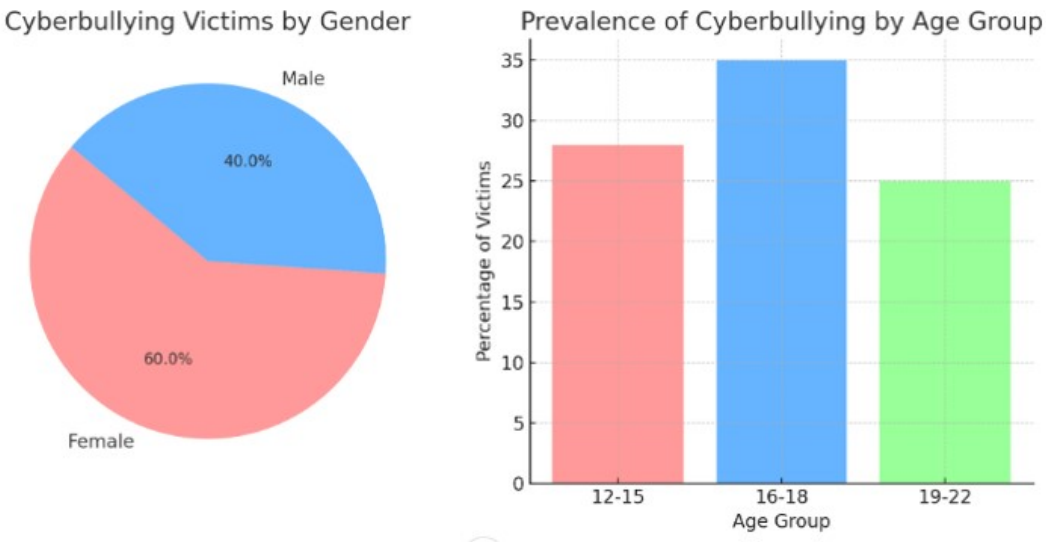

Figure 1.1: Cyberbullying in different age groups & gender

People may now more simply than ever share their ideas and feelings thanks to the development of social networks and mass media. More quickly than previously, information may now spread. However, the increased use of social media has also raised concerns about the dissemination of dangerous information in the name of unrestricted communication. Anything digital that irritates or upsets someone is toxic content. This phrase is quite vague since what one person considers to be problematic may not be to another. Toxic content often includes sexually explicit material, violent content, hate speech, offensive information, political or religious content, cyberbullying, and harassment. Such material tends to be more prominent on digital platforms as it easily captures public attention.

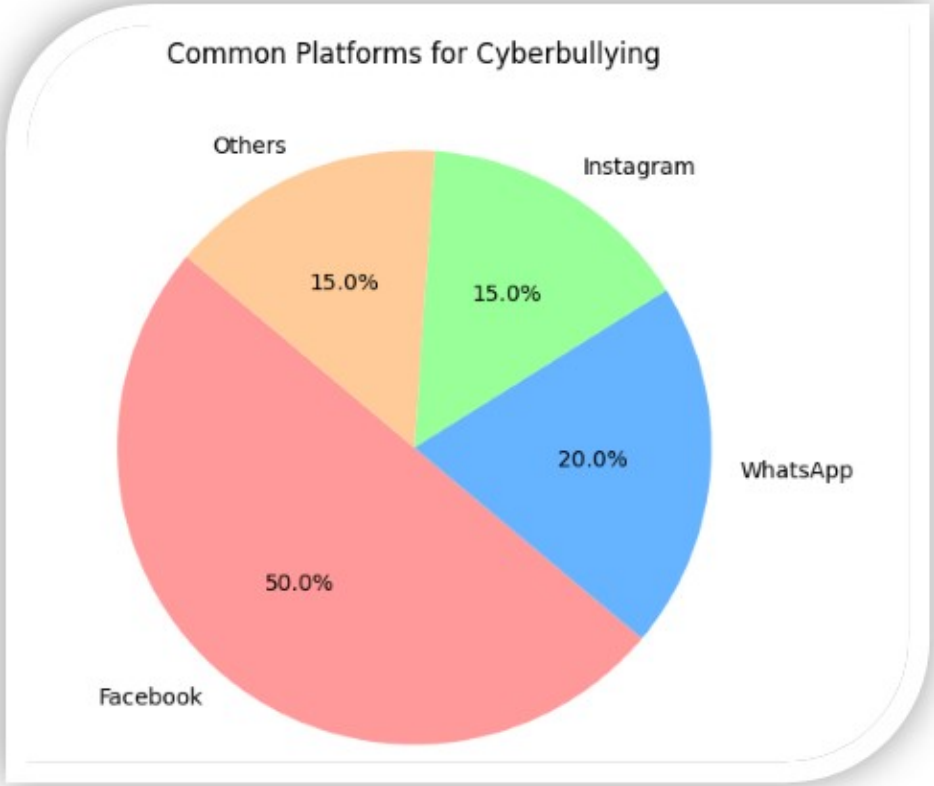

Figure 1.2: Social Media Platform in Cyberbullying

Even so, with over 229 million native speakers, Bengali is the seventh most spoken language in the world. [1], Its swift application as a derogatory and coarse language in instances of cyberbullying, harassment, and defamation crimes makes it difficult to access and enjoy, even to the point of using social media. In 2019, seventy percent of

Bangladeshi women in the fifteen to twenty-five age range said they had been the victim of online abuse. The nation's sole cybercrime tribunal heard 18 cases.% of harassment complaints as well as cases involving cyberbullying and defamation [2]. The poisonous social media comments used in cyberbullying incidents throughout the years have already resulted in a great number of suicides. hazardous comment identification [3], aggressive comment detection [4], cyberbullying detection [5], and hate speech detection [6] are just a few of the research projects that have been done to address hazardous material in the Bengali language. Most of these research suffer from insufficient data usage, and most of the time the datasets were imbalanced. Furthermore, most research papers regarded the classification of poisonous comments as a binary or multi-class problem.But a negative remark could end up in multiple courses at once. Furthermore, we found that feature representation was often performed using word embedding methods such as word2vec, fasttext, and GloVe, which is problematic given the presence of missing words, sentence context, and other factors. These attempts used transformer-based and deep learning models for classification tasks, but the models produced were not interpretable. This work provides a two-stage pipeline that employs transformer-based models and deep learning to define poison detection as a multi-label classification issue and identify dangerous comments in Bengali.

1.3 Our Contribution:

The main contributions we made in this research are, to sum up:

- We have generated a 16, 073 instance Bengali harmful comment detection dataset. Of those, 8, 488 are toxic and 7, 585 are not.Toxic instances are manually categorized into six categories: insult, troll, threat, religion, hate, and vulgar.
- We offer a pipeline that first determines whether or not a comment is harmful by using a binary classification model. If so, the toxicity kind is then classified using a multi-label classifier.
- We found that for binary classification, the LSTM model with BERT achieved the highest accuracy of 79.42%. For multi-label classification, the CNN-BiLSTM model with an attention mechanism performed best.The CNN-BiLSTM model was able to more accurately represent the relationships between the various categories because the attention layer was able to selectively focus on the most significant portions of the input sequence.

Chapter 2

Literature Review

Similar to previous sentiment analysis tasks, the researchers proposed generating intermediate vectors from the non-trainable raw texts by applying content-based feature extraction techniques like TF-IDF, count vectorizer, BOW (Bag of Words), etc. These feature extraction strategies determine the link between several words in a given document when contextual findings are absent.. Conventional machine learning classifiers performed admirably in these studies. Various probabilistic models, such as Naive Bayes classifiers and Random Forests, as well as rule-based classifiers like Decision trees, have demonstrated strong performance in sentiment analysis tasks.

Perera et al. [7] proposed to extract features from the Twitter dataset that are based on sentiment and content. Techniques for TF-IDF feature extraction were applied. Their main goal was to determine whether swear words and pronouns or the third person were related. To label the texts from the Twitter dataset, they examined various keywords associated with different forms of cyberbullying, such as racism, sexual harassment, troll, threat, etc. Once more, they used their understanding of polarity-based sentiment analysis to determine whether the comment was favorable or not. They experimented with a Support Vector Machine (SVM) to train the dataset after labeling and pre-processing nearly 1000 texts, and they discovered that cyberbullying could be identified with about 75% accuracy from English-language tweets.

Murshed et al. [8] suggested a Dolphin Echolocation Algorithm optimization to adjust the Recurrent Neural Network's parameters. To distinguish between texts that were bullying and those that weren't, they gathered 10,000 tweets. They worked with English texts and gathered information based on derogatory terms such as "rape," "bisexual," "fuck," and "moron," among others. They used 35 specific bully words—words that actually defined negative polarity—in the data collection process. To gather the data, they also concentrated on racism and bullying motivated by sexism. SMOTE was used

to balance the dataset because it was unbalanced. The suggested DEA-RNN model produced the highest classification accuracy of 90.4% when compared to other machine learning algorithms such as SVM, Random Forest, and Naive Bayes. There is no benchmark dataset for this work. The data was restricted to Twitter posts and the data collection procedure was biased towards certain specific abusive words. Information from various social media platforms could improve the model and make it more adaptable. The limit of this effort is the reduction of feature compatibility when managing massive input.

Islam et al. [9] suggested using five distinct classical machine learning classifiers to train the work dataset: Naive Bayes, decision trees, random forests, SVM, and DNN. They used BOW (Bag of Word) to transform the text into such a vector as part of the data pre-processing technique. Additionally, TF-IDF was used to produce word frequency data for text sentiment analysis. Working with a publicly available dataset, they extracted text from Kaggle that was either bullying or nonbullying. Using a DNN (Deep Neural Network), classification accuracy of about 99% was discovered.

Vijayakumar et al. [10] proposed a hybrid deep learning model that uses a convolutional neural network and LSTM (Long Short Term Memory) to detect cyberbullying. They used information from text and image data to approach this classification task. From textual data, they extracted sentiment and semantics, and from images, they extracted contextual information. Additionally, utilizing the "Kaggle Toxic Comment" classification dataset, they were able to identify toxic text with 85% classification accuracy.

Dewani et al. [11] developed a deep learning-based classifier to identify cyberbullying in Roman Urdu text. This approach involved creating a Roman Urdu slang dictionary, which was used to map slang terms following tokenization. The text underwent preprocessing and encoding before being trained with an RNN-LSTM model. The model's performance on the test set demonstrated an accuracy of 85% for detecting cyberbullying in Roman Urdu

Al-Ajlan et al. [12] suggested using the "GloVe" word embedding technique in conjunction with a convolutional neural network to identify texts that contain cyberbullying from the Twitter dataset. The word embedding layer, convolutional layer, max pooling layer, and dense layer made up this model. To identify cyberbullying, they used a content-based embedding that matched bully or bad words.

Roy et al. [13] suggested using image data to identify posts that contain cyberbullying using a 2-dimensional convolutional neural network. They gathered a dataset from Twitter that included pictures of bullies and non-bullies. To categorize photos of bullies, several pre-trained deep learning classifiers, including VGG16 and Inceptionv3, were tested. This study is unable to identify sentimental threats or trolls from textual contents, nor is it able to identify cyberbullying from texts written in multiple languages.

Bharti et al. [14] tried turning the text into a vector using a pre-trained "GloVe" word embedding model; the generated word vector identified a word as out-of-vector that wasn't present in the pre-trained model. They used character-level encoding to modify the pre-trained model in order to eliminate this bias. Using a Twitter dataset, they achieved 92% accuracy in classifying bullying text by training a Bi-LSTM model with pre-processed data.

Prior research primarily employed content-based feature extraction methods for sentiment analysis assignments. When dealing with ambiguous terms or multi-modality, contextual information can occasionally be more crucial than content.

Tan et al. [15] investigated a hybrid model that combined deep learning and transformer-based models. They used a meaningful word embedding technique to extract features from a robustly optimized BERT model. An additional element of this hybrid model was the LSTM, which effectively integrated features while maintaining contextual semantics. They conducted experiments using datasets from Twitter US Airlines, IMDB, etc. Techniques for data augmentation were investigated to lessen the constraints brought about by an unbalanced dataset. Tokens are converted by BERT into contextual word embedding vectors that the classifier can use for training. An attention mask was employed to optimize the model's fine tuning. To get the best accuracy, they experimented with various LSTM units, hyperparameters, learning rates, and optimizers. Their suggested hybrid model demonstrated 91% accuracy for this sentiment analysis task using the IMDB dataset.

Desai et al. [16] suggested using the transformer-based model BERT (created by Google AI Language researchers) to detect cyberbullying. For this sentiment analysis task, they used a bi-directional deep learning-based model to extract all syntactic, semantic, and sarcastic features. In this work, all of the embedding to convert the text into trainable parameters was done by the model's encoder. This work has demonstrated 70% accuracy in cyberbullying detection, but it is limited to a sufficiently large dataset to create a generalized model.

Hoq et al. [17] contributed to the use of Bangla in sentiment analysis. Three classes of Facebook comments were included in the dataset they prepared: angry, sad, and happy. There were a thousand texts in each class. The words were converted into vectors using a variety of embedding techniques, including GloVe, word embedding, Word2Vec, and CBOW (Continuous Bag of Words). To train this dataset, they adjusted their model. They experimented with CNN, LSTM, and a hybrid CNN/LSTM model to classify sentiments. In various situations, they obtained accuracy ranging from 85% to 90%.

Junaid et al. [18] suggested to categorize food reviews in the Bangla language using a deep learning-based model called LSTM combined with the Word2Sequence feature embedding technique. They helped compile a dataset of food reviews from websites

such as Foodpanda, Shohoj Food, Hungynaki, and others into Bangla.

Rahman et al. [18] suggested using LSTM, CNN, and Multilayer Perceptrons to examine bangla sentiment in cricket news. They gathered a dataset from the well-known newspaper centered on cricket news, bangla online news portal. Five sentiment groups were identified through the collection and classification of texts and comments: happy, sad, annoyed, neural, and advice. Their accuracy rate with this sentiment analysis task was about 70%.

Durga et al. [18] suggested examining the sentiment in texts written in Bangla and Romanized Bangla. They created a dataset using the web scraper tool that included both Bangla and romanized Bangla texts from YouTube comments because the Bangla sentiment analysis task lacked a sufficient dataset. After translating all texts into English, they categorized each sentiment as either neutral, positive, or negative. We experimented with Textblob to examine the sentences' polarity. For this task, a recurrent neural network was used to analyze sentiment.

Chapter 3

Methodology

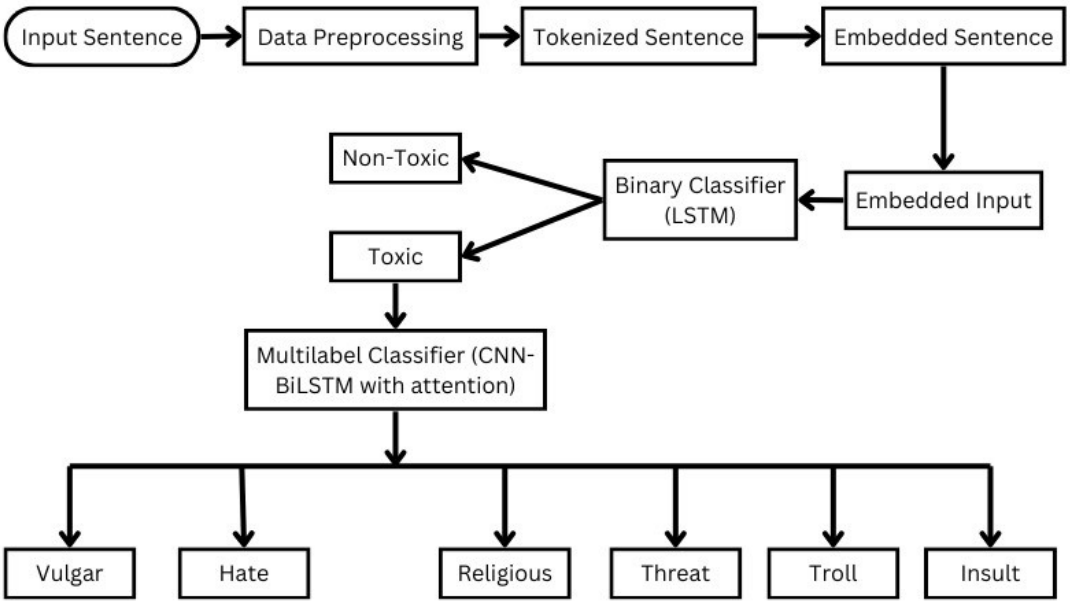

Figure 3.1: Proposed workflow

Our proposed workflow will go such way given below:
Figure shows the suggested system for classifying toxic comments. Initially, we frame the task as a binary classification task, in which we have to determine if a given input text is toxic or not. The input text will be classified into one or more of the following six classes: vulgar, hate, religious, threat, troll, and insult if it is determined that the text is toxic. This is because we are considering a multi-label classification problem. The main stages of the suggested approach are explained in more detail below.

1. **Sentence for Input:** The suggested model evaluates each raw sentence in the dataset one at a time, sequentially.

2. **Data Preprocessing:** The input sentence is purged of stop words, emoticons, hyperlinks, and punctuation.

3. **Tokenization:** Bangla BERT receives the tokenized output directly and uses it to represent each token numerically. Every token is represented by 768 real values, which adds up to a 300 x 768 embedded vector.

4. **Embedding:** The suggested model receives each raw sentence from the dataset one at a time for additional processing.

5. **Pooling:** A real-valued vector representation of size 768 per sentence is produced by max pooling, thereby reducing the dimension of the embedding vector (300 x 768).

6. **Binary Classification:** A Long Short Term Memory (LSTM) [14] network is applied to the embedding vector for binary classification in order to determine the dependency of the entire sentence. The max pooling, dropout, and dense layer layers come after the LSTM layer, in that order. In our model, the activation function for the hidden layers is a leaky rectified linear unit (Leaky ReLU), and the dense layer uses a sigmoid activation to predict the likelihood that the input text falls into the toxic or non-toxic category.

7. **Multi-label Sequencing:** When a text is flagged as toxic, it is run through a combination model that assigns it to one or more of the following six categories: threatening, vulgar, hateful, religious, troll, and insulting. Convolutional neural networks (CNNs) and bidirectional long-short-term memory (Bi-LSTM) networks with attention mechanisms are used in the combined model. The embedding vector of the poisonous text is first subjected to three one-dimensional convolutional layers, each of which is activated by a Rectified Linear Unit (ReLU). The 512, 256, and 128 filters in these layers have kernel sizes of 4, 3, and 2, respectively. Subsequently, the final pooling layer's output is fed into a 128-unit Bi-LSTM layer, where L2 regularization is used to minimize overfitting.

3.1 Data Collection

Text samples were collected from three different sources. Upon thorough analysis, it was discovered that the texts in these datasets were not consistently or accurately labeled from the beginning.Because some texts obviously fit into more than one category, it was decided to manually classify the texts into six categories:

Class	Number Of Instances
Vulgar	2505
Hate	1898
Religious	1418
Troll	1419
Threat	1643
Insult	2719

Table 3.1: Dataset Description

vulgar, hatred, religion, threat, troll, and insult. This allowed for several labels to be applied to each text. The texts must be reclassified into these new categories in order to provide more thorough insights into the type and level of toxicity that is present, as well as to improve data quality, consistency, and model performance.

3.1.1 Dataset statistics

The total number of cases in the dataset is 16, 073, of which 7, 585 are deemed non-hazardous and 8, 488 are deemed harmful. The distribution of harmful instances by class is displayed.

3.2 Dataset Preparation

The initial step in identifying cyberbullying activities is to obtain an input dataset from social media platforms. Facebook was selected for this thesis project. Text comments left on Facebook photos and posts make up the input dataset.
In order to enhance the quality of the research data, data pre-processing is applied to the input data. Hyperlinks, extra characters, and stop words are eliminated in later analytical stages. After the input data has undergone pre-processing, feature extraction is completed. By using feature extraction and word frequency analysis, features from the text such as nouns, adjectives, and pronouns are extracted. The Classification Algorithm is then provided with the extracted features.
The training data set obtained from Kaggle.com is used to first train the classfier system. The test data set is a portion of the data set—nearly 20% of it. The output of this classification algorithm shows whether or not bullying words are present in the given data (comment). 6 forms of bullying are intended to be identified by this system: Vulgar, Hate, Religious, Threat, Troll & Insult.

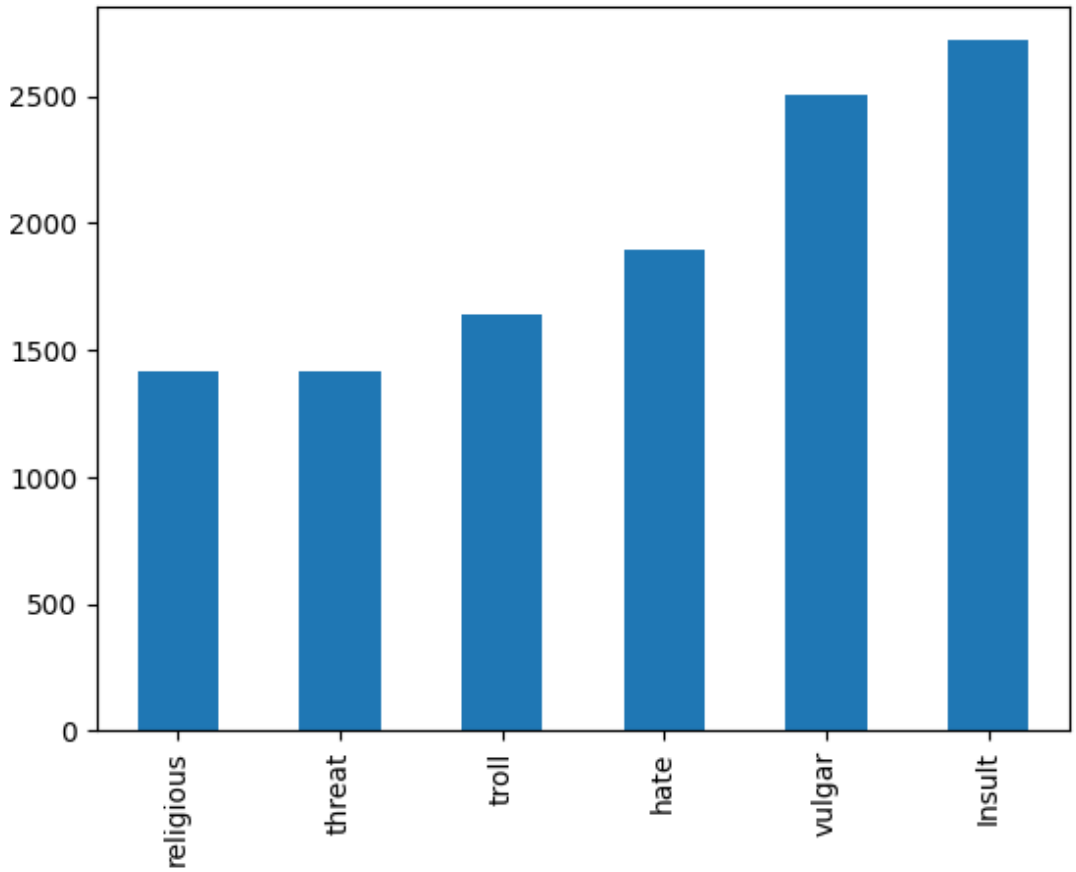

Figure 3.2: Data Statistics

3.2.1 Data Annotation

Data annotation in natural language processing (NLP) is the process of labeling text data to create structured datasets that can be used to train machine learning models. This involves tasks such as tagging parts of speech, identifying named entities, classifying sentiment, or marking toxic content. Accurate annotation is crucial as it directly influences the quality and performance of NLP models. Skilled annotators often follow predefined guidelines to ensure consistency, and inter-annotator agreement metrics like Cohen's kappa are used to validate the reliability of the annotations. High-quality annotated data is essential for developing robust NLP systems capable of understanding and processing human language effectively. Necessary data annotation as been performed here for our dataset.

3.2.2 Text Cleaning Removal of Stopwords

Text cleaning and the removal of stop words are crucial steps in natural language processing (NLP) and text mining, aimed at refining raw text data for more ef-

	text	vulgar	hate	religious	threat	troll	Insult
0	কিরে তুই কি পাবলিকের কমেন্ট পড়স না,তোরে তো কে...	0	0	0	0	0	1
1	তোর মত নাস্তিককে বাংলার জমিন থেকে জুতা মেরে বে...	0	0	1	1	0	0
2	রাজাকার শোভাযাত্রা ও সমাবেশ করা উচিত	0	1	0	0	0	0
3	তোদের পুরোহিতদের আমি হিংসা করিরে তোদের সেক্সি ...	0	0	1	0	0	0

Figure 3.3: Dataset overview

fective analysis. Text cleaning involves several preprocessing tasks to enhance data quality and manageability. This includes removing punctuation, converting all text to lowercase to ensure uniformity, stripping out numbers and special characters, and correcting spelling errors. Once the text is cleaned, the removal of stop words—common but less meaningful words such as ”and,” ”the,” and ”is”—further refines the data. Stop words are often removed to eliminate noise, reduce dataset size, and highlight more significant content. Techniques for this include using predefined lists from NLP libraries or creating custom lists tailored to specific contexts. Overall, these preprocessing steps streamline text data, making it more suitable for analysis and improving the efficiency and accuracy of NLP applications.

```
test_text = pd.DataFrame(x_test, columns = ['text'])
test_label = pd.DataFrame(y_test, columns = ['vulgar', 'hate', 'religious', 'threat', 'troll', 'Insult'])
test_df = pd.concat([test_text, test_label], axis=1, join='inner')
test_df = test_df.loc[((test_df['vulgar'] == 1) | (test_df['hate'] == 1) | (test_df['religious'] ==1) | (test_df['threat'] == 1) | (test_df['troll'] == 1) | (test_df['Insult'] ==1))]
test_df = test_df.sample(frac=1).reset_index(drop=True)
test_df
```

	text	vulgar	hate	religious	threat	troll	Insult
0	রানু মন্ডল কে দেখলে আমার শরীর রক্ত হয়ে যায় তা...	0	0	0	0	0	1
1	নিজের পুটকিতে নিজে আঙ্গুল দিয়া বয়া থাক	1	0	0	0	0	1
2	তোর ভাগে ভিক্ষার টাকা পাছ নাই তাই হাম্বা হাম্ব...	0	1	0	0	0	0
3	মুচারের পুতেরে শয়তানে লারে চারে।	1	0	0	0	0	1
4	রাত দুই টার ফোন দিয়ে বন্ধু তার ২৩৮ ব্রেকআপের ক...	1	0	0	0	1	0
...	...	...	...	...	...	...	...
1364	দুইদিনেই রাজাকার হয়ে গেলোআমারে আমির বানাইলে দে...	0	1	0	0	0	0
1365	প্রিয় সুশীল মনে পড়ে কি ২১ আগস্টের বর্বরতা	0	1	0	0	0	0
1366	পোস্ট টা যে করেছে তাকে একটু দেখা দরকার সে কি আ...	1	0	0	1	0	0
1367	যাক বঙ্গবন্ধু স্যাটেলাইট সার্থক হলো হজ্বের যাত...	0	0	0	0	1	0
1368	জুতা মারো সবাই সাফা সালিকে	0	0	0	1	0	1

Figure 3.4: Clean text without stopword

3.3 Tokenization

We began by segmenting the comments in our dataset, which is a vital process in natural language processing (NLP). Segmentation involves breaking down a text into smaller components known as tokens, a critical step for various NLP tasks, including text classification, named entity recognition, language modeling, and

machine translation. In the initial stage of our research, we applied tokenization to the raw text from our dataset. For example, tokenization was utilized on our Bangla dataset, allowing us to decompose the raw text into manageable segments or tokens. The process initiates with the raw text, which can range from a single line or phrase to an entire document or a large corpus. If the dataset's input is a multi-sentence document or paragraph, the first step is generally to tokenize the text into separate sentences. The objective is to identify sentence boundaries and split the text into distinct sentences, using punctuation marks like commas, exclamation points, and question marks as delimiters. Following sentence tokenization, the next step is word tokenization.

Once the text is split into sentences, the next step is word tokenization, which breaks the text into individual words or tokens. The simplest approach is to use spaces, tabs, or newlines as delimiters to split the text wherever a space appears. However, this method might not work well for all languages, especially those with complex word structures or those that do not use spaces to separate words. For such cases, subword tokenization can be used. This method is particularly useful for languages with complex word patterns or for handling words that are not in the standard vocabulary. Subword tokenization addresses the limitations of word-level tokenization by breaking text into smaller units, such as subwords or character n-grams. This approach is particularly useful for languages with complex morphology or for handling out-of-vocabulary (OOV) words. Subword tokenization allows the model to handle and learn from rare or previously unseen words by decomposing them into more frequent, manageable units. Techniques like Byte-Pair Encoding (BPE) or Unigram Language Models are commonly used for this purpose. Furthermore, we have fed this tokenized sequence into our word embedding techniques, including GloVe and FastText.

3.4 Sentence Embedding

3.4.1 BERT Embedding

BERT (Bidirectional Encoder Representations from Transformers) embedding is a type of word representation that captures the context of words in a text using a deep learning model based on the transformer architecture. Unlike traditional word embeddings (like Word2Vec or GloVe) that assign a single, static vector to each word regardless of its context, BERT embeddings are dynamic. They generate different vector representations for the same word depending on the context

in which it appears. This ability to understand context at a deep level makes BERT embeddings particularly powerful for tasks like text classification, sentiment analysis, and question-answering.
The process of generating BERT embeddings begins with input preparation. The first step is tokenization, where the input text is split into smaller units called tokens. BERT uses a specific tokenization method known as WordPiece tokenization, which can break down words into subword units. This is particularly useful for handling rare or unfamiliar words by splitting them into smaller, more common components. Additionally, special tokens are added to the input sequence: [CLS] at the beginning, which is used for classification tasks, and [SEP] at the end, which marks the boundary of sentences or the end of the sequence.

Once the text is tokenized, it passes through the embedding layer, where each token is converted into a fixed-size vector. These embeddings are not just simple word vectors; they also include segment embeddings to differentiate between different parts of the input (such as two sentences in a pair) and positional embeddings to encode the position of each token in the sequence. These combined embeddings are then fed into the transformer layers of BERT.
In the transformer layers, BERT employs a self-attention mechanism that allows the model to consider the relationship between each token and every other token in the sequence. This is crucial for understanding the context in which a word is used, as the meaning of a word can change drastically depending on the words around it. Unlike unidirectional models that process text from left to right or right to left, BERT is bidirectional. It reads the text in both directions simultaneously, which enables it to capture a word's full context within the sentence. In summary, BERT embeddings are powerful tools in NLP, thanks to their ability to capture nuanced, context-aware representations of text. The bidirectional self-attention mechanism enables BERT to understand the full context of words, enhancing its performance across a wide range of language tasks and applications.

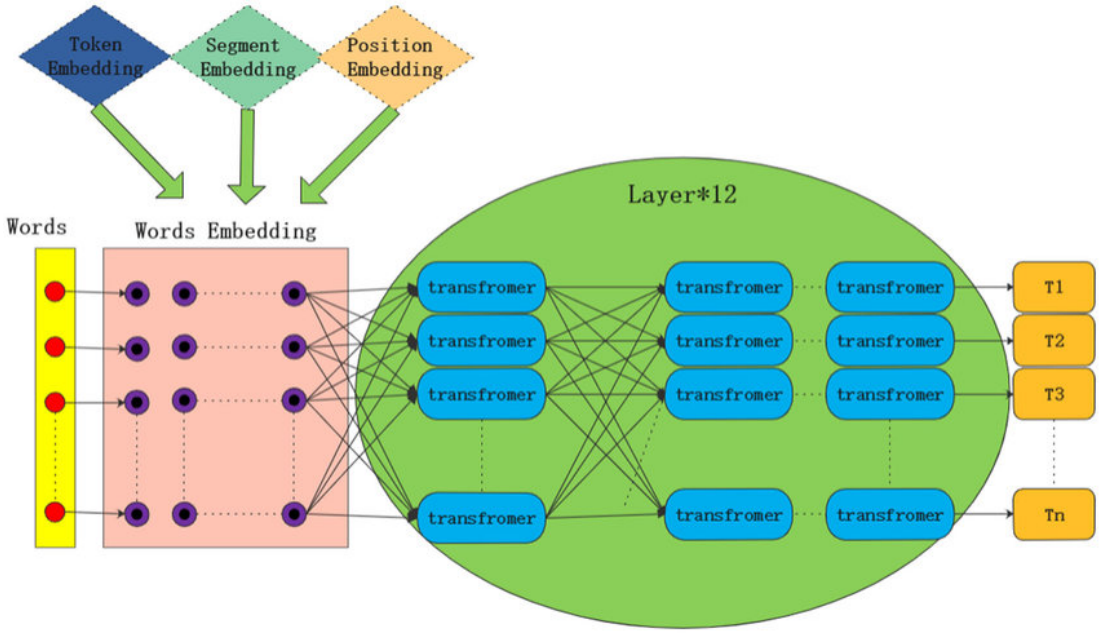

Figure 3.5: BERT Embedding Architecture

3.4.2 Attention Mask

An attention mask plays a vital role in transformer models like BERT by guiding the self-attention mechanism on where to focus during processing. Transformer models rely on self-attention to determine how much attention each token in a sequence should give to every other token. However, not all tokens in a sequence carry meaningful information—particularly in cases where sequences are padded to ensure uniform length across a batch of data.

- **Binary Encoding:** The attention mask is essentially a binary tensor (a matrix of 1s and 0s) that corresponds to the input tokens. Each token in the input sequence has an associated mask value: 1 indicates that the token is valid and should be attended to, while 0 indicates that the token is irrelevant, often because it is a padding token. Padding tokens are added to shorter sequences to make them as long as the longest sequence in the batch, but they don't carry any meaningful information.
- **Masking in Practice:** During the self-attention operation, the attention mask tells the model which tokens should be included in the calculation of attention scores. Tokens with a mask value of 1 participate fully in the self-attention process, contributing to the overall understanding of the sequence. Tokens with a mask value of 0 are effectively ignored, ensuring that the padding does not distort the model's comprehension of the actual content.

- **Impact on Training and Inference:** By applying an attention mask, the model is prevented from giving weight to padding tokens, which would otherwise skew the attention distribution and potentially degrade performance. This is particularly important during both training and inference, where the model might encounter sequences of varying lengths. The attention mask allows the model to maintain consistency in how it processes input, regardless of sequence length, leading to more accurate and reliable outputs.

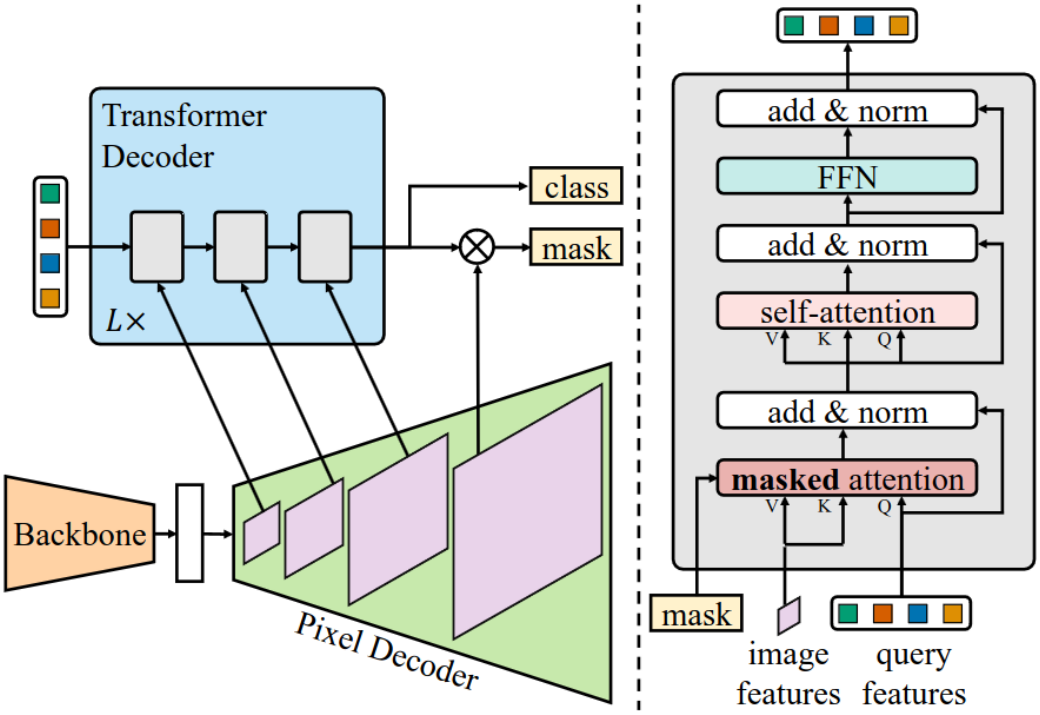

Figure 3.6: Attention Mask Architecture

An attention mask is a crucial component in transformer-based models like BERT, used to control how the model attends to different parts of the input sequence. In simple terms, the attention mask is a binary vector that indicates which tokens in the input should be considered (attended to) and which should be ignored (masked out) during the self-attention mechanism. Typically, tokens that represent meaningful text receive a mask value of 1, while padding tokens (used to ensure uniform sequence length) are assigned a value of 0. This ensures that the model focuses only on the relevant parts of the input, preventing it from learning or making decisions based on padding, which could lead to incorrect or biased outputs. By using an attention mask, models can more effectively process varying lengths of input data without being influenced by irrelevant information.

```
X_input_ids_train = np.zeros((len(x_train), max_length))
X_attn_masks_train = np.zeros((len(x_train), max_length))

X_input_ids_dev = np.zeros((len(x_val), max_length))
X_attn_masks_dev = np.zeros((len(x_val), max_length))

X_input_ids_test = np.zeros((len(x_test), max_length))
X_attn_masks_test = np.zeros((len(x_test), max_length))
```

```
from tqdm import tqdm
def preprocessing_dataset(df, ids, masks, tokenizer):
    for i, text in tqdm(enumerate(df)):
        tokenized_text = tokenizer.encode_plus(
            text,
            max_length=max_length,
            truncation=True,
            padding='max_length',
            add_special_tokens=True,
            return_tensors='tf'
        )
        ids[i, :] = tokenized_text.input_ids
        masks[i, :] = tokenized_text.attention_mask
    return ids, masks
```

```
X_input_ids_tain, X_attn_masks_tain = preprocessing_dataset(x_train, X_input_ids_train, X_attn_masks_train, tokenizer)
X_input_ids_dev, X_attn_masks_dev = preprocessing_dataset(x_val, X_input_ids_dev, X_attn_masks_dev, tokenizer)
X_input_ids_test, X_attn_masks_test = preprocessing_dataset(x_test, X_input_ids_test, X_attn_masks_test, tokenizer)
```

```
5093it [00:04, 1031.57it/s]
2066it [00:02, 792.25it/s]
1369it [00:01, 1238.92it/s]
```

Figure 3.7: Attention Masking output

3.5 Model Overview

LSTM (Long Short-Term Memory) networks enhance traditional RNN (Recurrent Neural Network) architectures by incorporating three types of gates: input, forget, and output. These gates manage the flow of information within the network. The forget gate decides which information should be discarded from the previous state, while the input gate evaluates the importance of new data and works alongside the forget gate to eliminate unnecessary information, allowing relevant data to be processed. The output gate is responsible for determining the network's next hidden state, facilitating the passage of information through the activation function. In our case, using two LSTM layers, each with 128 neurons and the "RELU" activation function, proved highly effective in capturing complex text features. The model's design remains relatively simple with only two layers, as increasing the number of layers could complicate training. Since our input data contains no negative values, the "RELU" activation function was chosen to enhance processing capabilities and uncover nonlinear relationships.

In our model construction outlines the process of building a predictive model by detail-

ing each step involved. It starts with defining the problem and selecting the appropriate data. Next, it involves preprocessing the data, including tasks like normalization and feature selection. The tree then branches into choosing the model type, such as decision trees or neural networks, and configuring its parameters. The model is trained on the prepared data, followed by evaluating its performance using metrics like accuracy or F1 score. Finally, the model is fine-tuned based on performance results and deployed for practical use.

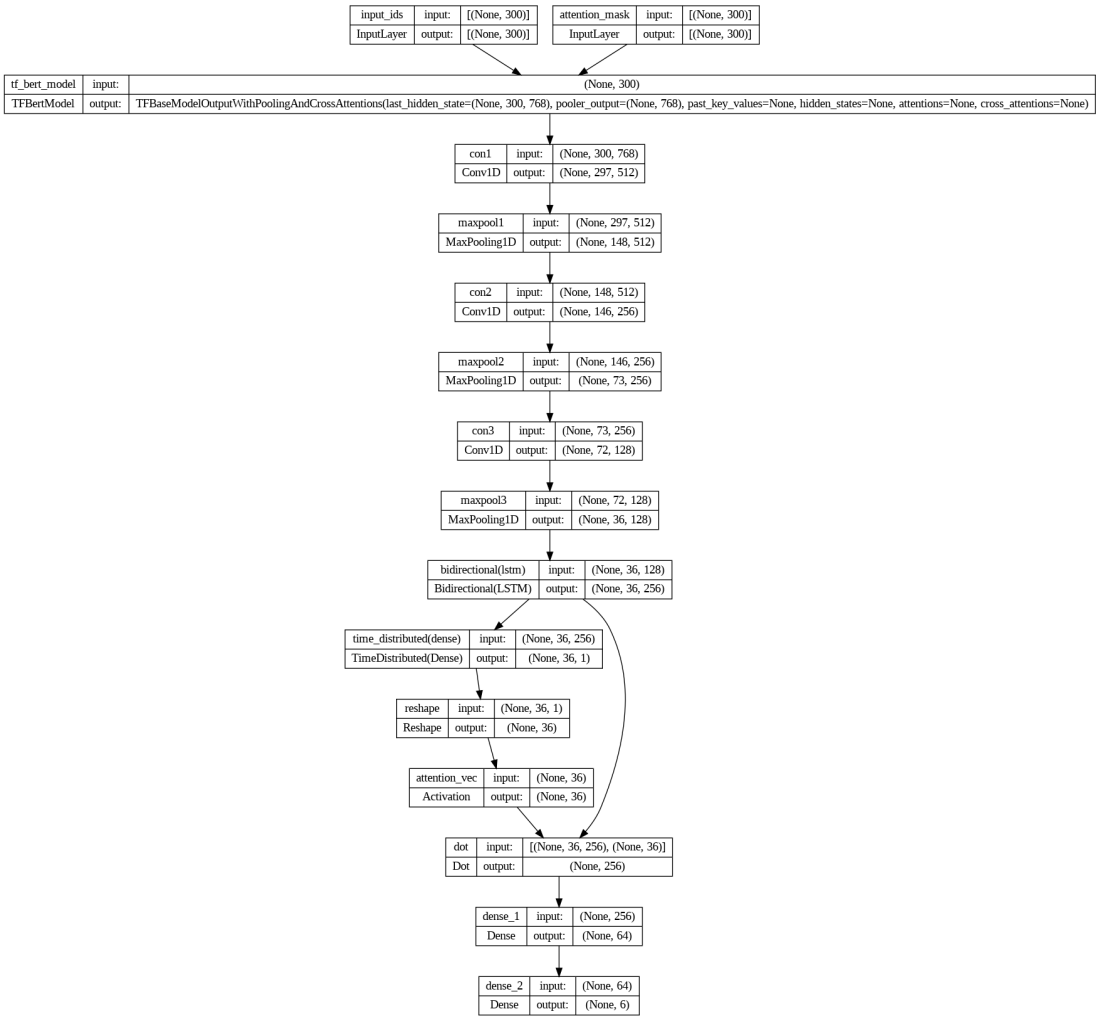

Figure 3.8: Constructed Model

In our model construction, we integrate Bi-LSTM (Bidirectional Long Short-Term Memory) networks with BanglaBERT embeddings to address Bangla language tasks effectively. The process begins with defining the problem, such as sentiment analysis or named entity recognition, which requires nuanced understanding of Bangla text. We then select a relevant dataset annotated for our specific task and proceed with preprocessing. This includes normalizing the text, tokenizing it with BanglaBERT to generate contextual embeddings, and selecting key features for the task. The core of our model

consists of two Bi-LSTM layers, each with 128 neurons, designed to capture complex text features through their bidirectional nature. These layers use the ReLU activation function to enhance processing capabilities and uncover nonlinear relationships, even though LSTMs inherently use sigmoid and tanh functions internally. The forget gate in the LSTM decides what information to discard from previous states, the input gate evaluates the importance of new data, and the output gate determines the next hidden state. We train the model using the BanglaBERT-generated embeddings and evaluate its performance using metrics like accuracy and F1 score. Based on these evaluations, we fine-tune the model's parameters and eventually deploy it for practical use, ensuring it can effectively handle new Bangla text data in real-world applications. This approach balances the richness of contextual embeddings from BanglaBERT with the sophisticated sequence modeling capabilities of Bi-LSTM, achieving a robust and effective solution for Bangla language processing tasks.

3.5.1 CNN:

A Convolutional Neural Network (CNN) architecture is made up of convolution and pooling layers. Artificial neurons are used by CNN to process inputs and transmit outputs. Text embedded within is used as input. A CNN can have a large number of hidden layers, and in order to extract features, it performs a lot of calculations. The first step in the process of extracting features from an input is the convolution layer. The output layer's classification is aided by the fully connected layer. CNNs are referred to as feed-forward networks since data only moves from inputs to outputs in a single direction.

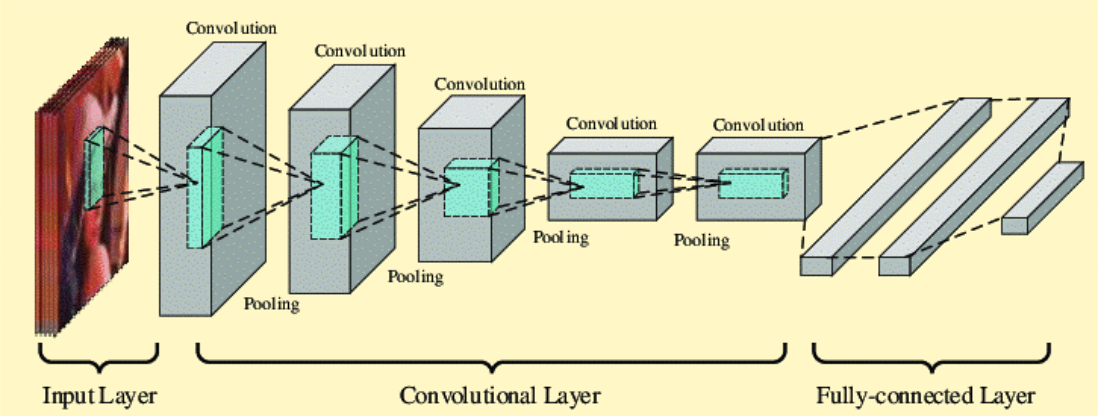

Figure 3.9: CNN Architecture

3.5.2 Bi-LSTM:

One kind of recurrent neural network (RNN) intended to capture long-term dependencies in data is called Bidirectional Long Short-Term Memory (Bi-LSTM) [15]. Bi-LSTM networks use information from both sides of the sequence to process input in both forward and backward directions, in contrast to traditional LSTM networks that only process input in one direction. Because of a few linear interactions, the cell state in an LSTM is essential for information flow through the network with the least amount of modification. Three gates are present in every LSTM unit: an input gate, an output gate, and a forget gate. The forget gate determines whether to keep or discard data from the previous cell state based on a sigmoid function.An additional LSTM layer that reverses the information flow is added to a Bi-LSTM network. This indicates that the input sequence is processed backwards in the additional LSTM layer. The information that should be forwarded to the following hidden state is then decided by the final output gate.

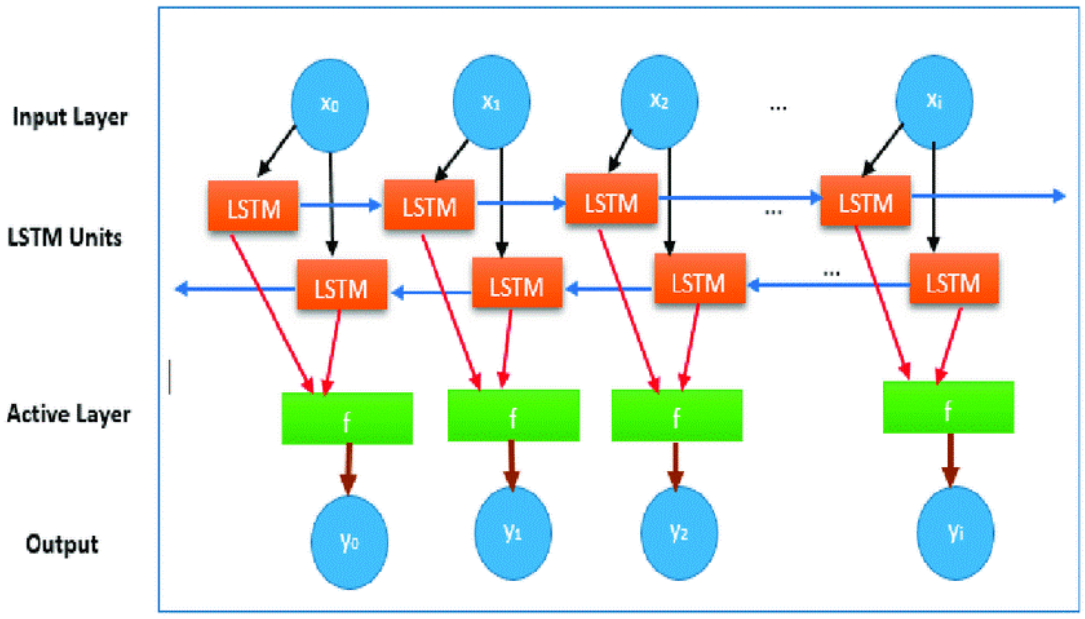

Figure 3.10: Bi-LSTM Architecture

3.5.3 Transformer Models

1. **BERT:**Recent advancements in natural language processing (NLP) have been significantly driven by powerful Transformer-based language models. Among these, BERT (Bidirectional Encoder Representations from Transformers) stands out as one of the pioneering models. BERT's strength lies in its extensive pre-training on diverse text data, which enables it to grasp both semantic relationships within sentences and the contextual nuances of language. What sets BERT apart is its bidirectional approach, which allows it to generate rich contextual embeddings. Unlike models that process text in a single direction, BERT considers both preceding and succeeding words when encoding each term in a sentence. This bidirectional encoding provides a deeper understanding of the context and meaning of words based on their surrounding text. BERT's versatility is evident in

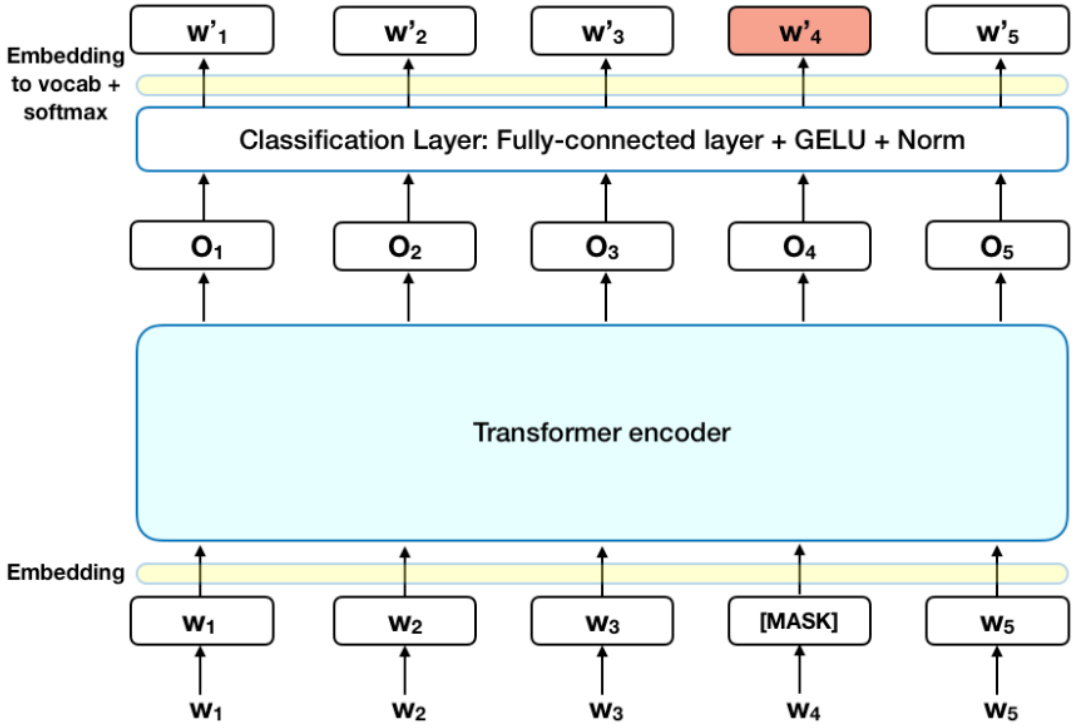

Figure 3.11: BERT Model Architecture

its ability to be fine-tuned for specific tasks, adapting its general language understanding to meet the needs of particular applications. For instance, in cyberbullying detection, BERT's pretrained representations can be refined through fine-tuning to focus on the specific linguistic patterns and contextual cues associated with harmful behavior. This targeted adaptation enhances BERT's performance in detecting nuanced instances of cyberbullying, demonstrating its effectiveness in specialized domains.

Overall, BERT's ability to leverage bidirectional context and its adaptability through fine-tuning make it a powerful tool for various NLP tasks. Its application in areas such as cyberbullying detection highlights its potential to provide accurate and contextually aware analyses, paving the way for more effective solutions in language understanding and processing.

2. **Bangla BERT:** Bangla BERT is a specialized adaptation of the BERT model designed specifically for the Bangla language. It employs a Transformer-based architecture, which allows it to process and understand text in a highly contextual and nuanced manner. This architecture is particularly effective for capturing the rich semantic and syntactic details of Bangla, making it well-suited for complex language tasks.

   The model benefits from pretraining on extensive Bangla text corpora, which helps it learn a wide range of linguistic patterns and contextual meanings unique to Bangla. During pretraining, Bangla BERT is exposed to a diverse set of texts, allowing it to build a deep understanding of the language's structure and usage. This foundational knowledge is crucial for effectively handling the intricacies of Bangla, including its various linguistic features and idiomatic expressions.

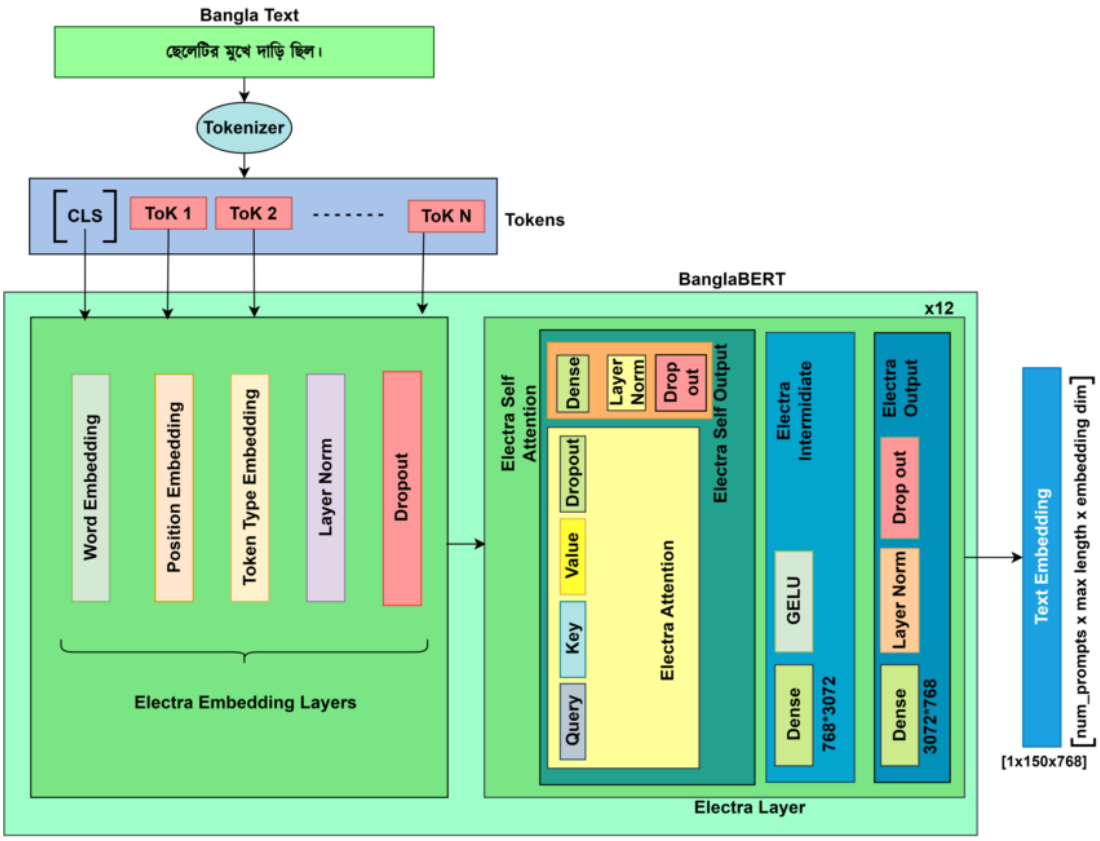

Figure 3.12: Bangla BERT Model Architecture

Once pretrained, Bangla BERT can be fine-tuned on specific datasets tailored to particular tasks, such as detecting cyberbullying in Bangla social media comments. Fine-tuning involves adjusting the model's parameters based on task-specific data, which enables it to adapt its general language understanding to the nuances of the task at hand. In the case of cyberbullying detection, this means training the model to recognize and classify instances of abusive or harmful language based on patterns learned during pretraining.

The application of Bangla BERT in natural language processing (NLP) tasks, including cyberbullying detection, has shown promising results. Its ability to grasp the subtleties of Bangla text enhances its performance in identifying contextually relevant cues that indicate cyberbullying behavior.

Chapter 4

Experiment Result & Analysis

4.1 Train-test split:

In natural language processing (NLP), the train-test-validation split is a method used to optimize and evaluate machine learning models more effectively. This approach divides the dataset into three distinct subsets: training, validation, and test sets. The training set is used to train the model, while the validation set is used to fine-tune hyperparameters and make adjustments to improve performance. Finally, the test set, which remains unseen during training and validation, provides an unbiased evaluation of the model's generalization ability. This triple-split approach helps in avoiding overfitting, selecting the best model configuration, and ensuring reliable performance metrics for real-world applications.

Total Data	Training	Testing	Validation
16073	5093	1368	2065

Figure 4.1: Total Data split

Class	Count(training)
Vulgar	1503
Hate	1139
Religious	851
Threat	851
Troll	986
Insult	1631

Figure 4.2: Data split for each class

4.2 Training model:

There are multiple crucial processes involved in training a multiclass model for the classification of Bangla cyberbullying using a CNN-BiLSTM architecture. To get ready for model training, the first step in the procedure is data preprocessing, which involves cleaning and tokenizing Bangla text. The Convolutional Neural Network (CNN) component is used to extract local features and patterns from the text sequences, leveraging its ability to capture n-gram information effectively. The extracted features are fed into a Bidirectional Long Short-Term Memory (BiLSTM) network, which processes the data in both forward and backward directions, enabling the model to capture context from both preceding and succeeding tokens. This dual approach helps the model grasp the sequential dependencies and contextual subtleties of the Bangla language, which is crucial for accurately classifying different forms of cyberbullying. The model is trained on labeled data, utilizing techniques like cross-entropy loss to manage multiclass outputs, and optimization algorithms such as Adam to enhance learning efficiency. To evaluate the model's effectiveness, metrics such as accuracy, precision, recall, and F1-score are employed, ensuring the model can accurately differentiate between various types of cyberbullying in Bangla.

```
[ ]  print("Data ->Vulgar, Hate, Religious, Threat, Troll, Insult")
     print("Train: ", y_train.sum(axis = 0))
     print("Validation: ", y_val.sum(axis = 0))
     print("Test: ", y_test.sum(axis = 0))

     Data ->Vulgar, Hate, Religious, Threat, Troll, Insult
     Train:  [1503 1139  851  851  986 1631]
     Validation:  [624 445 324 343 424 653]
     Test:  [440 294 229 224 254 469]
```

Figure 4.3: Training dataset of each class

```
Epoch 1/8
319/319 [==============================] - 397s 1s/step - loss: 0.4851 - accuracy: 0.3039 - fmeasure: 0.2546 - precision: 0.5115 - recall: 0.1862
Epoch 2/8
319/319 [==============================] - 361s 1s/step - loss: 0.3665 - accuracy: 0.6036 - fmeasure: 0.6085 - precision: 0.7549 - recall: 0.5195
Epoch 3/8
319/319 [==============================] - 360s 1s/step - loss: 0.3014 - accuracy: 0.6978 - fmeasure: 0.7226 - precision: 0.8023 - recall: 0.6633
Epoch 4/8
319/319 [==============================] - 357s 1s/step - loss: 0.2485 - accuracy: 0.7630 - fmeasure: 0.7911 - precision: 0.8461 - recall: 0.7483
Epoch 5/8
319/319 [==============================] - 360s 1s/step - loss: 0.2074 - accuracy: 0.8037 - fmeasure: 0.8381 - precision: 0.8789 - recall: 0.8043
Epoch 6/8
319/319 [==============================] - 360s 1s/step - loss: 0.1730 - accuracy: 0.8266 - fmeasure: 0.8720 - precision: 0.9062 - recall: 0.8436
Epoch 7/8
319/319 [==============================] - 360s 1s/step - loss: 0.1441 - accuracy: 0.8396 - fmeasure: 0.8964 - precision: 0.9258 - recall: 0.8717
Epoch 8/8
319/319 [==============================] - 358s 1s/step - loss: 0.1180 - accuracy: 0.8520 - fmeasure: 0.9200 - precision: 0.9477 - recall: 0.8963
```

Figure 4.4: Training the model

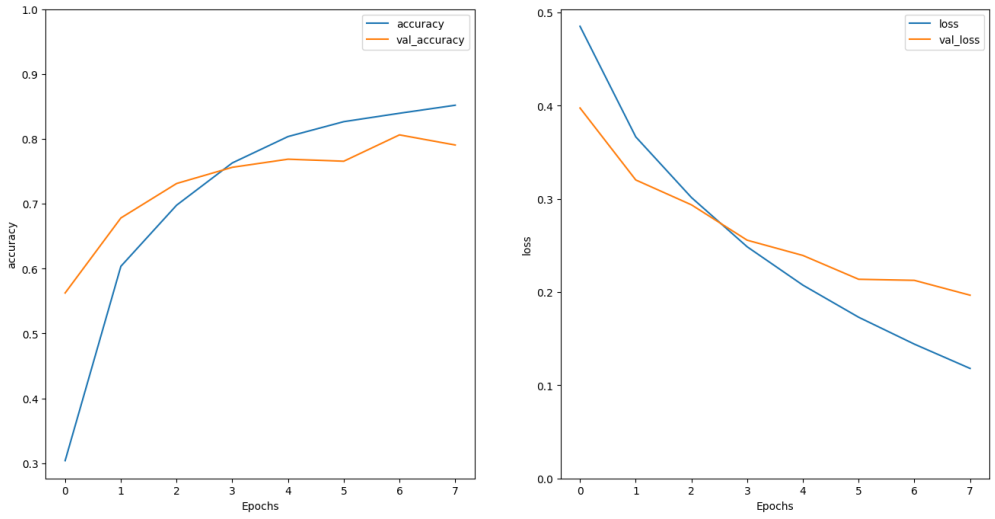

Figure 4.5: Model Accuracy vs Loss

4.3 Evaluation Metric:

We assessed Bangla-BERT on four downstream tasks for Bangla language comprehension and these are cross-lingual sentiment analysis, named entity recognition, binary Text Classification, and multi-class sentiment analysis. In addition, we have evaluated Bangla-BERT to the multilingual variant of BERT, including other enhanced neural techniques such as word2vecfor findings for each task as a baseline.

1. **Accracy:** Accuracy is a performance parameter for machine learning classification models that is defined as the proportion of true positives and negatives to the total number of positive and negative observations. In other words, accuracy is the proportion of times we anticipate our machine learning model to predict a result correctly out of the total number of times it has made predictions. Mathematically, it defines the ratio between the sum of all true positives (TP) and true negatives (TN).

$$AccuracyScore = \frac{(TP + TN)}{(TP + FN + TN + FP)}$$

2. **Precison:** The model precision score estimates the proportion of accurately predicted positive labels. Precision is also referred to as the predictive value of the positive. It indicates the mathematical ratio of true positives to the sum of true positives and false positives (FP)

$$PrecisionScore = \frac{TP}{(FP + TP)}$$

3. **Recall:** The score for model recall reflects the model's ability to correctly forecast the positives from the actual positives. It depicts the mathematical ratio of true positives to the sum of true positives and false negatives (FN).

$$RecallScore = \frac{TP}{(FN + TP)}$$

4. **F1-Score:** F1 score indicates the model score as a function of the recall and precision scores. F1 score is an alternative to Accuracy metrics that provides equal weight to both Precision and Recall when analyzing the performance of a machine learning model in terms of accuracy. It can be mathematically expressed as a harmonic mean of precision and recall score..

$$F1Score = \frac{2 * PrecisionScore * RecallScore}{(PrecisionScore + RecallScore)}$$

4.4 Result Analysis:

This section evaluates the classification model's accuracy across various content categories, providing insights into its strengths and areas for improvement. The accuracy rates help assess how effectively the model identifies and classifies different types of inappropriate or harmful content.

The table shows that the model performs best in identifying religious content, with an accuracy of 97.05%, followed by vulgar content at 94.37% and threats at 93.79%. The model also detects hate speech well, with 93.64% accuracy, and trolling behavior at 92.03%. However, the model is slightly less accurate in identifying insults, achieving 88.45% accuracy, indicating a need for improvement in this area. Overall, the model shows strong performance across most categories, with the highest accuracy in religious content and the lowest in insults.

Class Label	Accuracy(%)
Vulgar	94.37
Hate	93.64
Religious	97.05
Threat	93.79
Troll	92.03
Insult	88.45

Figure 4.6: Each label accuracy

4.5 Confusion Matrix:

An essential instrument for evaluating the efficacy of a multiclass classification model—like the one employed to identify cyberbullying in Bangla—is the classification report. For every class, it offers a thorough summary of the most important parameters, such as support, recall, F1 score, and precision. Precision measures the percentage of true positives out of all projected positives for a given class to determine how accurate the model's positive predictions are. By calculating the percentage of true positives among all actual cases, recall, also known as sensitivity, assesses the model's accuracy in identifying every real instance of a class. The F1 score provides a balanced metric that takes into account the trade-off between these two metrics. It is calculated as the harmonic mean of precision and recall.

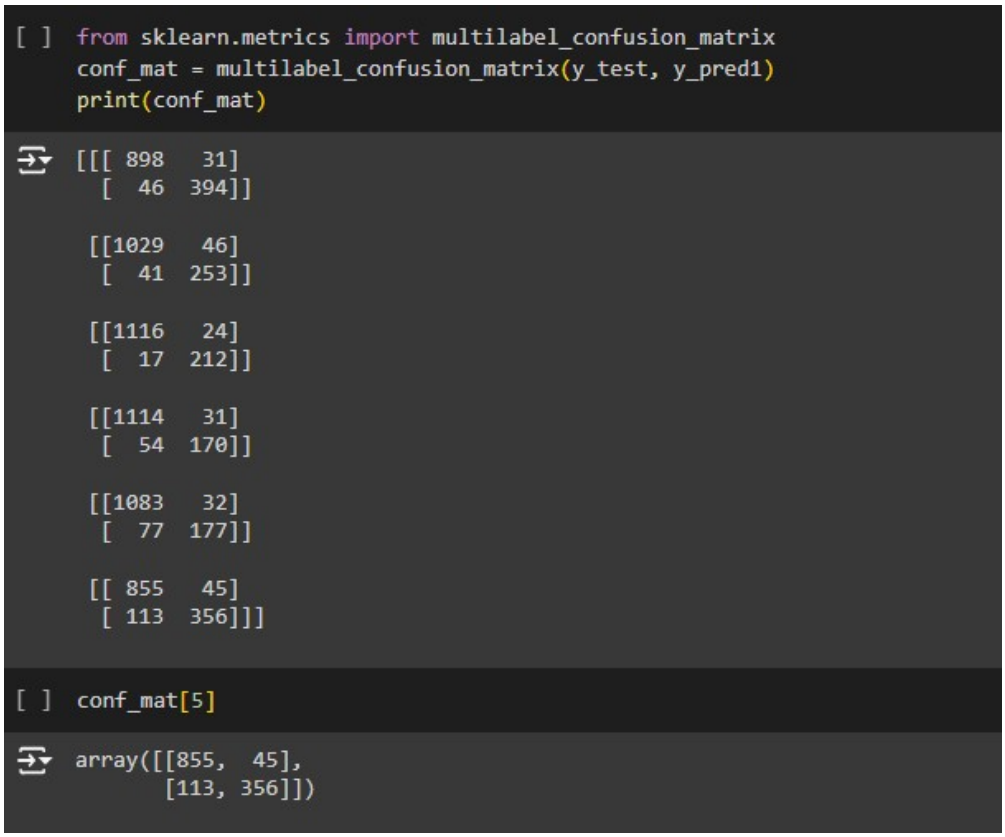

Figure 4.7: Each label accuracy

Moreover, in multiclass classification, micro, macro, and weighted averages of these metrics can be reported to provide a more comprehensive evaluation. The micro-average aggregates the contributions of all classes to compute the average metric, treating all instances equally, which is useful when class imbalance exists. The macro-average, on the other hand, computes the metric independently for each class and then takes the average, treating all classes equally, which helps in understanding the model's performance across classes without bias. The weighted average considers the support of each class, making it a balanced metric that accounts for class frequency.

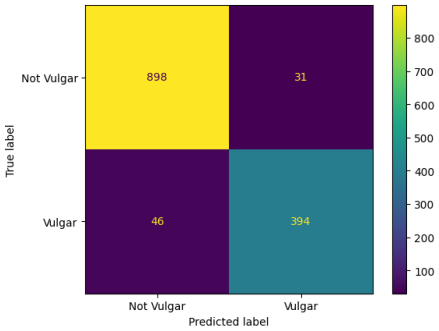

Figure 4.8: Vulgar

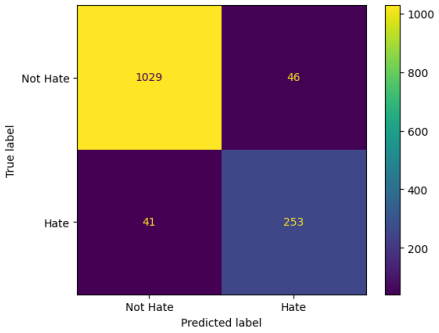

Figure 4.9: Hate

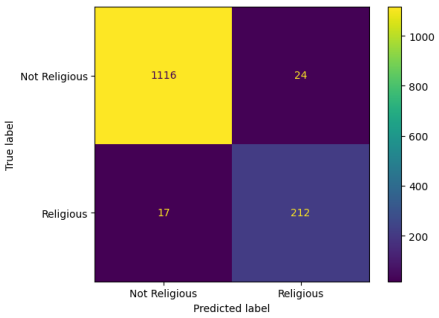

Figure 4.10: Religious

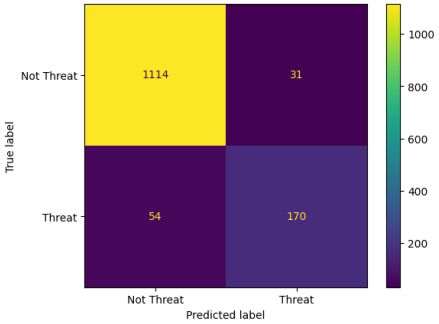

Figure 4.11: Threat

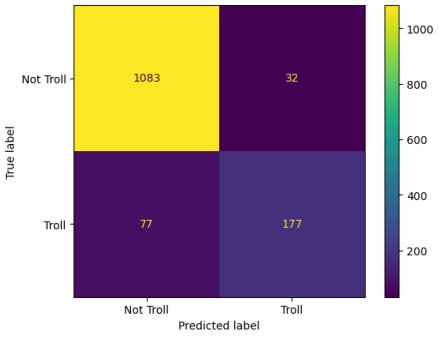

Figure 4.12: Troll

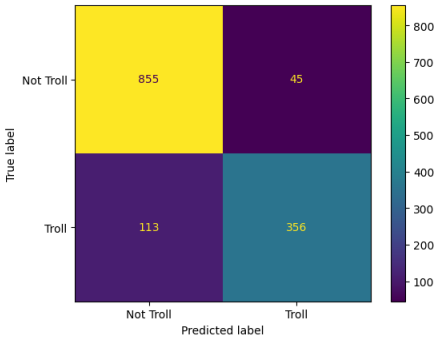

Figure 4.13: Insult

	precision	recall	f1-score	support
vulgar	0.9271	0.8955	0.9110	440
hate	0.8462	0.8605	0.8533	294
religious	0.8983	0.9258	0.9118	229
threat	0.8458	0.7589	0.8000	224
troll	0.8469	0.6969	0.7646	254
Insult	0.8878	0.7591	0.8184	469
micro avg	0.8820	0.8178	0.8487	1910
macro avg	0.8753	0.8161	0.8432	1910
weighted avg	0.8813	0.8178	0.8470	1910
samples avg	0.8598	0.8301	0.8348	1910

Figure 4.14: Classification Report Summery

4.6 ROC Curve Analysis:

Plotting the True Positive Rate (TPR) versus the False Positive Rate (FPR) yields the Receiver Operating Characteristic (ROC) curve, which is a useful tool for assessing a classification model's performance. In order to evaluate the model's capacity to discriminate between various classes, the ROC curve is expanded to a multi-class classification situation in this investigation.

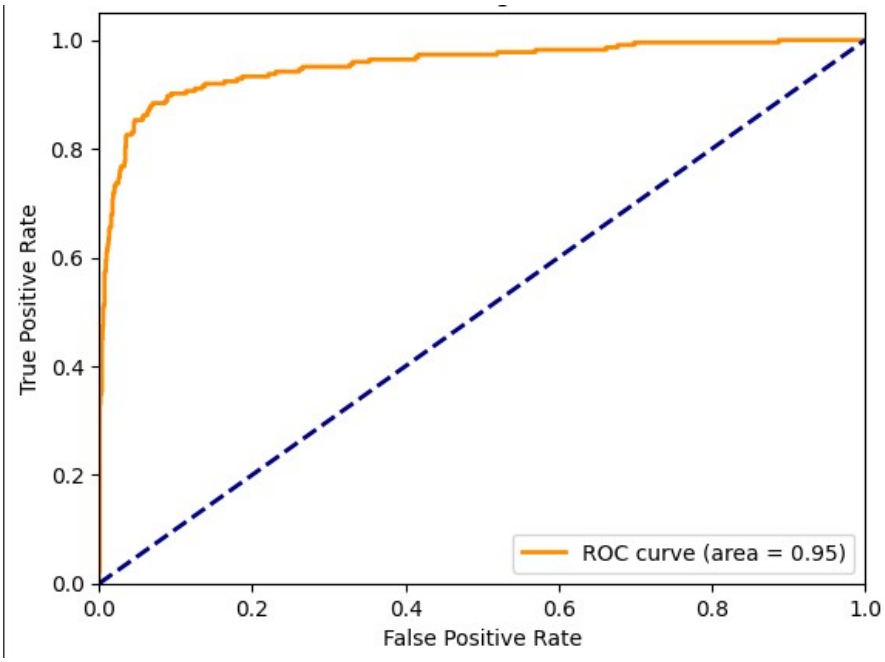

Figure 4.15: Model ROC curve

The micro-average ROC curve, encompassing all class contributions, exhibits an Area Under the Curve (AUC) value of 0.96. This shows that the model handles every instance uniformly across all classes, resulting in good overall performance. In a similar vein, the AUC of the macro-average ROC curve, which averages the ROC curves for each class, is 0.96. This implies that all classes experience the same level of performance from the model.

When analyzing the class-specific ROC curves, we observe that the ”vulgar” and ”religious” classes have the highest AUC values of 0.97, indicating the model is highly effective at distinguishing these classes from others. The ”hate” class follows closely with an AUC of 0.96, showing strong performance as well. The ”threat” class has a slightly lower AUC of 0.95, but still demonstrates robust performance. However, the model shows a slight decline in performance for the ”troll” and ”insult” classes, both with an AUC of 0.93. This suggests that the model has more difficulty in distinguishing these classes compared to others, which might indicate the need for further refinement or additional training data. Overall, the ROC curves reveal that the model performs well across most classes, with high AUC values indicating a strong ability to differentiate between different classes. The slight drop in performance for certain classes like ”troll” and ”insult” could be a focus area for improving the model’s accuracy.

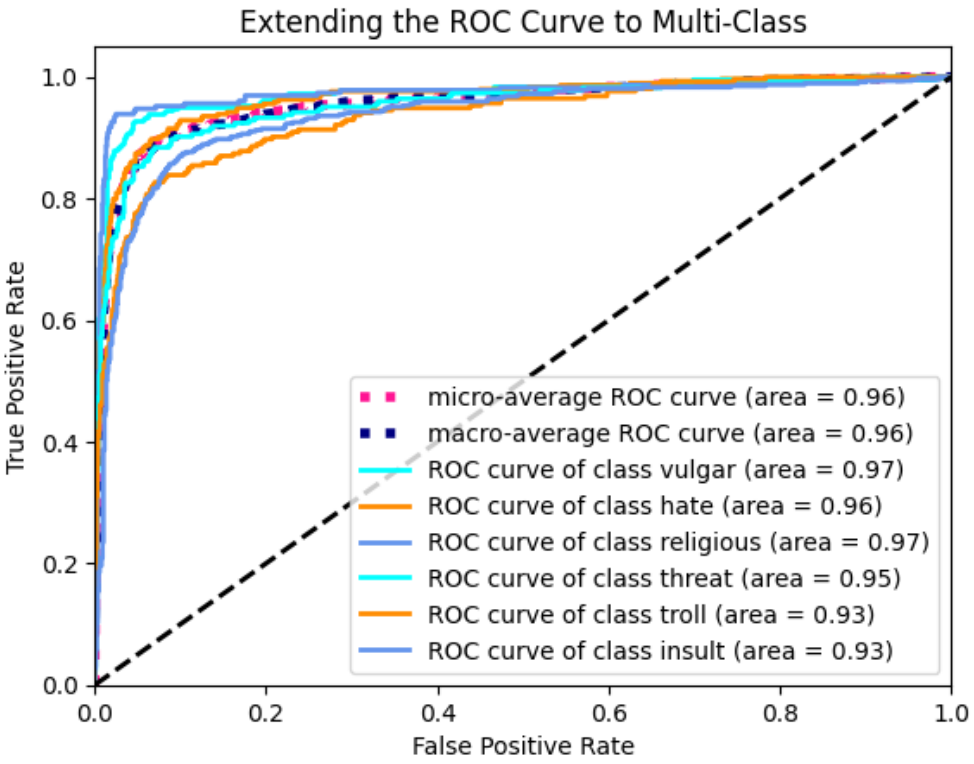

Figure 4.16: ROC curve

Chapter 5

Conclusion

5.1 Conclusion

In our study, we undertook an extensive exploration of multilingual cyberbullying detection, concentrating on Bangla texts. This research led to several significant findings and laid the groundwork for future work in this area. To ensure a well-rounded analysis, we curated a robust dataset consisting of over 16,000 Bangla social media comments, employing meticulous data collection and annotation methods to maintain balance and accuracy. Our evaluation began with traditional machine learning techniques, including ensemble methods and Support Vector Machines, which yielded promising results. However, deep learning approaches, particularly Convolutional Neural Networks (CNNs), demonstrated superior performance in detecting instances of cyberbullying.

We also introduced and assessed hybrid network models, highlighting the benefits of combining different neural network architectures to enhance detection accuracy. Notably, transformer-based models such as Bangla BERT, when integrated with Bi-LSTM (Bidirectional Long Short-Term Memory), exhibited remarkable improvements in detection precision. These advancements underscore the potential of leveraging advanced models and hybrid approaches for more effective cyberbullying detection in Bangla texts, offering valuable insights for ongoing and future research in this domain.

Additionally, our analysis explored various preprocessing techniques and feature extraction methods that further refined the model's performance. We examined the impact of different tokenization strategies and the inclusion of linguistic features, which contributed to a deeper understanding of the text's sentiment and intent. The results of this study not only demonstrate the effectiveness of advanced models in detecting cyberbullying but also provide a comprehensive framework for future research. By integrating

these insights, we aim to enhance automated systems for monitoring and mitigating online abuse in Bangla and other multilingual contexts.

5.2 Future Scope

In our research on Bangla cyberbullying classification, we plan to apply our current model to a larger dataset to enhance the model's effectiveness and credibility for real-time applications. Expanding the dataset will help improve the robustness and generalizability of our model, which is crucial for its adoption by various organizations and platforms.

We will incorporate feature extraction techniques from pre-trained models into algorithms such as Support Vector Machines (SVMs) to further enhance performance. This includes refining performance metrics, like specificity and sensitivity, to ensure accurate detection of cyberbullying instances across diverse contexts.

Additionally, our future work will involve evaluating different text preprocessing methods and feature extraction approaches to optimize the model's ability to handle complex and varied cyberbullying scenarios. We aim to leverage transfer learning to adapt existing models to better recognize and classify subtle and context-specific forms of cyberbullying in Bangla texts.

A key focus will be integrating explainable AI (XAI) techniques into our models. By implementing methods such as attention mechanisms and gradient-based visualization, we seek to make the model's decision-making process more transparent and interpretable. This will provide valuable insights into which aspects of Bangla text contribute most significantly to the classification of cyberbullying, thereby helping stakeholders understand and trust the system's predictions. Enhancing explainability will be essential for building confidence in AI-driven solutions for detecting and addressing cyberbullying in multilingual contexts.

References

[1] Bangladesh Telecommunication Regulatory Commission, "Bangladesh Telecommunication Regulatory Commission," http://www.btrc.gov.bd/site/page/347df7fe-409f-451e-a415-65b109a207f5/- (accessed on 15 January 2023).

[2] T. Mahmud, M. Ptaszynski, J. Eronen, and F. Masui, "Cyberbullying detection for low-resource languages and dialects: Review of the state of the art," *Information Processing and Management*, vol. 60, p. 103454, 2023.

[3] T. Mahmud, M. Ptaszynski, and F. Masui, "Automatic vulgar word extraction method with application to vulgar remark detection in chittagonian dialect of bangla," *Applied Sciences*, vol. 13, p. 11875, 2023.

[4] R. Pawar and R. Raje, "Multilingual cyberbullying detection system," in *Proceedings of the 2019 IEEE International Conference on Electro Information Technology (EIT)*. Brookings, SD, USA: IEEE, 2019, pp. 040–044.

[5] B. Haidar, M. Chamoun, and A. Serhrouchni, "Multilingual cyberbullying detection system: Detecting cyberbullying in arabic content," in *Proceedings of the 2017 1st Cyber Security in Networking Conference (CSNet)*. Rio de Janeiro, Brazil: IEEE, 2017, pp. 1–8.

[6] C. Okoloegbo, U. Eze, G. Chukwudebe, and O. Nwokonkwo, "Multilingual cyberbullying detector (cd) application for nigerian pidgin and igbo language corpus," in *Proceedings of the 2022 5th Information Technology for Education and Development (ITED)*. Abuja, Nigeria: IEEE, 2022, pp. 1–6.

[7] E. Mahajan, H. Mahajan, and S. Kumar, "Ensmulhatecyb: Multilingual hate speech and cyberbully detection in online social media," *Expert Systems with Applications*, vol. 236, p. 121228, 2023.

[8] S. Si, A. Datta, S. Banerjee, and S. Naskar, "Aggression detection on multilingual social media text," in *Proceedings of the 2019 10th International Conference on*

Computing, Communication and Networking Technologies (ICCCNT). Kanpur, India: IEEE, 2019, pp. 1–5.

[9] S. Roy, U. Narayan, T. Raha, Z. Abid, and V. Varma, "Leveraging multilingual transformers for hate speech detection," *arXiv*, 2021, arXiv:2101.03207.

[10] F. El-Alami, S. El Alaoui, and N. Nahnahi, "A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, pp. 6048–6056, 2022.

[11] A. Das, S. Nandy, R. Saha, S. Das, and D. Saha, "Analysis and detection of multilingual hate speech using transformer based deep learning," *arXiv*, 2024, arXiv:2401.11021.

[12] T. Mahmud, S. Das, M. Ptaszynski, M. Hossain, K. Andersson, and K. Barua, "Reason based machine learning approach to detect bangla abusive social media comments," in *Intelligent Computing & Optimization, Proceedings of the 5th International Conference on Intelligent Computing and Optimization 2022 (ICO2022)*. Hua Hin, Thailand: Springer, 2022, pp. 489–498.

[13] R. Rahman, "Robust and consistent estimation of word embedding for bangla language by fine-tuning word2vec model," in *Proceedings of the 2020 23rd International Conference on Computer and Information Technology (ICCIT)*. Dhaka, Bangladesh: IEEE, 2020, pp. 1–6.

[14] T. Mahmud, M. Ptaszynski, and F. Masui, "Vulgar remarks detection in chittagonian dialect of bangla," *arXiv*, 2023, arXiv:2308.15448.

[15] M. Wadud, M. Mridha, J. Shin, K. Nur, and A. Saha, "Deep-bert: Transfer learning for classifying multilingual offensive texts on social media," *Computer Systems Science and Engineering*, vol. 44, pp. 1775–1791, 2023.

[16] P. Sharma and P. Parwekar, "Comparing ensemble techniques for bilingual multiclass classification of online reviews," in *Proceedings of the International Conference on Emerging Trends in Expert Applications and Security*. Jaipur, India: Springer, 2018, pp. 275–284.

[17] E. Can, A. Ezen-Can, and F. Can, "Multilingual sentiment analysis: An rnn-based framework for limited data," *arXiv*, 2018, arXiv:1806.04511.

[18] M. Kowsher, A. Sami, N. Prottasha, M. Arefin, P. Dhar, and T. Koshiba, “Bangla-bert: Transformer-based efficient model for transfer learning and language understanding.”