Troll or Abuser Identification and Removal in Social Media for Code-Mixed English, Bangla and Banglish with Machine Learning

Arif Nezami
Department of Computer Science and Engineering
United International University

A thesis submitted for the degree of
MSc in Computer Science & Engineering
November 2024

Declaration

I, **Mohammad Arif Nezami**, declare that this thesis titled, ***Troll or Abuser Identification and Removal in Social Media for Code-Mixed English, Bangla and Banglish with Machine Learning*** and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for an MSc degree at United International University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at United International University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Mohammad Arif Nezami

Certificate

I do hereby declare that the research works embodied in this thesis entitled ”Troll or Abuser Identification and Removal in Social Media for Code-Mixed English, Bangla and Banglish with Machine Learning” is the outcome of an original work carried out by ”Mohammad Arif Nezami” under my supervision.

I further certify that the dissertation meets the requirements and the standard for the degree of MSc/BSc in Computer Science and Engineering.

Signed: _____

Date: _____

Mohammad Arif Nezami
Department of Computer Science and Engineering,
United International University,
Dhaka-1209, Bangladesh.

Signed: _____

Date: _____

Prof. Dr. Engr. Mohammad Nurul Huda
Department of Computer Science and Engineering,
United International University,
Dhaka-1209, Bangladesh.

Abstract

Social networking platforms have become our center for daily communication. These networks removed all the barriers that used to exist by providing open platform for sharing thoughts, keeping in touch with friends, and even voicing for what we believe in. Unfortunately, this has also opened the door for trolls and online bullies to run rampant. Along with democratizing communication, these platforms have created a distinctive vocabulary, such as the widespread usage of emojis, which are particularly well-liked by younger audiences. "Banglish" a hybrid language that combines Bengali and English is widely used by Bengalis around the world. It creates serious difficulties for traditional content filtering systems.

The study of how people's views, sentiments, assessments, attitudes, and emotions are conveyed in writing is known as sentiment analysis or opinion mining. This is one of the most active study areas in natural language processing (NPL). Because of its relevance for business and society, this research has expanded beyond computer science and into the management and social sciences. This study gives an overview of the difficulties with sentiment analysis that apply to its methods and procedures.

The most recent research that uses deep learning to address issues with sentiment analysis is reviewed in this publication. A comparison study of the experimental findings for various models and input attributes has been carried out. This study examines how to identify trolls and abusive content in Banglish text using NLP models and sophisticated large language models (LLMs), such as OpenAI GPT-4, Google Gemini, and Meta LLaMA. Our goal is to use these models to improve the security and reduce the toxicity of social media environments, ensuring a more secure online experience for all users.

Dedicated to the staffs, students and faculty members of UIU.

Published Papers

Work relating to the research presented in this thesis has been published/ submitted by the author in the following peer-reviewed journals and conferences:

1. Mohammad Arif Nezami, Mohammad Nurul Huda, "Public sentiment identification in social media for code-mixed English-Bangla and Banglish text analysis with machine learning," Proc. SPIE 13225, Sixth International Conference on Image, Video Processing, and Artificial Intelligence (IVPAI 2024), 132250H (19 September 2024); `https://doi.org/10.1117/12.3047050`
2. Mohammad Arif Nezami, Mohammad Nurul Huda, Troll and Abuser Identification in Social Media for Code-Mixed English, Bangla, and Banglish Text Analysis Using LLMs- the 3rd International Conference on Intelligent Computing and Next Generation Networks (ICNGN 2024)

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In today's digital world, understanding emotions and intentions in online communication is really important. Social media has changed how we interact—we don't need to meet face-to-face or even talk directly anymore. Social networking platforms have become our center for daily communication. These networks removed all the barriers that used to exist by providing open platform for sharing thoughts, keeping in touch with friends, and even voicing for what we believe in. Unfortunately, this has also opened the door for trolls and online bullies to run rampant. Along with democratizing communication, these platforms have created a distinctive vocabulary, such as the widespread usage of emojis, which are particularly well-liked by younger audiences.

Emotions are mental states caused by changes in our brains, linked to our thoughts, feelings, and actions. By recognizing these emotions in text, we can get many benefits in different areas. For example, companies can adjust marketing based on customer feelings, or employers can see how employees are doing.

Sentiment analysis is a key method to interpret emotions in text. It's an automated process using Natural Language Processing (NLP) to decide if text is positive, negative, or neutral about a subject. NLP helps machines understand human language, letting algorithms do tasks that usually need human intelligence.

Our research focuses on finding and removing trolls and abusive content on social media, especially in code-mixed texts mixing English, Bangla, and "Banglish"—a blend of both languages. Code-mixed language online is a unique challenge for sentiment analysis and machine learning models. Traditional methods often struggle to interpret and classify this kind of content accurately, so we need special strategies.

This chapter introduces our research by outlining the current challenges in processing code-mixed languages, why we did this study, and what we aim to achieve. We discuss the contributions our work makes to computational linguistics and social media analytics. Also, we give an overview of the thesis structure, which shows the next chapters detailing our methods, findings, and the wider impact of our work.

Our approach starts by collecting primary data from social media platforms. Then we preprocess the data and extract features that capture the subtle sentiments in code-mixed texts. Using advanced machine learning algorithms, we classify the sentiments and detect abusive or troll-like behavior. The goal of our research is to improve sentiment analysis in multilingual and code-mixed contexts, helping to create safer and more positive online spaces.

1.1 Motivation

We are in the midst of a big technological change called the Fourth Industrial Revolution. This revolution, known as IR 4.0, combines developments in genetic engineering, robotics, the

Internet of Things (IoT), artificial intelligence (AI), and more. According to Riemann (2016) and Sony (2019), it's "The New Normal" driven by digital technologies expanding into almost every part of business operations.

Because of these technological breakthroughs, a huge number of people are now using social media like Facebook, Twitter, and Instagram more to communicate and share thoughts with others. These platforms have turned into public forums for exchanging opinions, creating web content, and building connections.

Currently, 5 billion people, or 63% of the worlds population, use the internet. The number keeps growing; recent estimates show that about 200 million more people got access to it in the last year. Social media usage is also increasing, with 4.65 billion users worldwide according to a survey from April 2022. Over the past 12 months, there was a 7.5% global rise in social media users. In April 2022, Facebook was on top as the most popular social media network with 2.936 billion active monthly users.

In Bangladesh, Facebook is used by 93.05% of social media users, according to another survey by StatCounter in May 2022. Just a year ago, it was only 79.36%. NapoleonCat says that 47.2 million people in Bangladesh used Facebook as of May this year. This number is 28% of the country's total population. The biggest growth in user numbers has been seen in Messenger. In June 2022, Messenger had 1.3 billion users. With 1.21 billion active monthly users in 2021, Meta's Instagram was used by over 28% of all internet users worldwide.

As user numbers and engagement are rising day by day, the amount of opinions and viewpoints is also increasing equally. Studies show that Sentiment Analysis research in Natural Language Processing (NLP) has mostly focused on the English language. There are some research projects done on the Bengali language, but not many. A large number of Bengali people use social media, often mixing English and Bengali in their posts.

Chapter 2

Related Work

In our study activity, which entails reviewing System Literature, we have followed the prescribed techniques for finding, filtering, obtaining raw data, building models, experimenting with, and reporting on various Machine Language classifiers. Beginning with articles, datasets, research summaries, journals, and presentations that addressed sentiment analysis or opinion mining, we searched for pertinent materials. We looked for related research publications on Google Scholar, GitHub, Scrip journals, Core, ACM Portal, Kaggle, Science Open, and Google Search Engine. To consistently get the greatest search results, a systematic search method was developed.
This study's objective is to assess Bengali-language public opinions and categorize them into three groups: favorable, negative, and neutral feelings. To describe various emotional states, we are taking consideration into social media post Comments (Banglish and Bangla). We focused on the 7 basic human emotions and analyze them. We also employed several machine learning techniques. Techniques like Linear Discriminant Analysis (LDA), K Neighbors, Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Gaussian Naive Bayes (GaussianNB), Sequential Model (SM), Long Short-term Memory (LSTM), and Bidirectional Long Short-term Memory can be used to create classification models (BLSTM).

2.1 Sentiment Analysis

Sentiment analysis (SA) is the analytical examination of how people feel and perceive various entities, such as individuals, groups, circumstances, events, and themes[1]. In recent years, sentiment analysis has been increasingly popular among organizations, governments, and enterprises[2]. SA provides industry insights to improve brand perception and business performance. Customer-focused businesses use SA to get customer insight for corporate strategy development. Sentiment analysis is regarded as one of the research fields in computer science that is expanding the fastest. The development of sentiment analysis may be traced back to surveys on public opinion research at the start of the 20th century and text analysis by a machine learning group in the 1990s claims a study. To better understand the demands of the population, SA techniques like text mining, deep learning, and natural language processing (NLP) are applied. To create a "tech-savvy society," digital technology must be used to improve institutional efficiency, economic growth, and the welfare of citizens[3]. SA can be used to design and create infrastructure, innovative solutions, and citizen engagement in a smart society[4]. SA can be applied to relationship marketing, consumer interaction, and consumer behavior research[5]. Organizations need to conduct this kind of study so they can incorporate client feedback into improvements to their goods and operations. Citizens serve as social sensors in social media, and opinions serve as real-time social data, making it an effective information source for social insight[6].

Machine learning applications for sentiment categorization, predictive decision-making, and the detection of phony reviews that combine internet reviews are covered by the authors[7]. Using powerful machine learning methods, [8] did a detailed investigation of the various social media analytics programs. Brief overviews of the machine learning algorithms utilized in social media analysis are provided by the authors [9].
Both text and Opinion Mining are mainly used for two different objectives. The primary objective is to examine public opinion regarding a problem or occurrence[10]. As a result, sentiment analysis examines a vast amount of textual data to determine people's opinions, attitudes, and feelings toward a subject[11]. The second goal is to determine how people feel about a certain subject from the viewpoint of a user or group of users. Related to Sentiment Analysis, the Natural Language Process task known as Opinion Mining identifies opinionated materials and categorizes them as having a positive, negative, or neutral polarity using an algorithmic approach[12]. This section discusses previously completed sentiment analysis and opinion mining studies.
Sentiment analysis has mainly been studied on three levels:

- **Document level** - The sentiment analysis of the document categorizes the opinions provided about a good or service across the document into different sentiments. This level designates whether an opinion piece is positive, negative, or neutral. This degree of sentiment analysis takes into account the fact that every document expresses the viewpoint of a single entity. The classification of documents at the entity level does not assess or contrast attitudes.

- **Statement level** - Sentiment analysis assesses the degree to which each sentence reflects a favorable, unfavorable, or neutral impression of goods or services. One-sentence evaluations and comments submitted by users go in this category. Subjectivity classification, which distinguishes between sentences that express factual information, and sentence-level classification are closely connected. The distinction between subjectivity and opinion should be made clear because many objective sentences may also contain an opinion.

- **Entity and Aspect Level** - Based on features, opinion mining and summarization are done at the aspect level. Identification and extraction of product features from the source data are part of the classification. When needing opinions about a desirable aspect or feature in a review, this type is employed. At the document or phrase level, sentiment analytics cannot pinpoint precisely what the likes and dislikes of a person. Sentiment analysis at the entity level doesn't examine language constructs like sentences, clauses, paragraphs, or entire sentences; it just examines the option itself. An opinion consists of a sentiment (emotion) and a target in this passage, which is the major concept (of that sentiment).

2.2 Natural Language Processing

For the purpose of learning, comprehending, and producing content in human language, natural language processing uses computer-based methods. The function of software or hardware

parts in a computer system that analyzes or produces spoken or written language is known as natural language processing, or NLP[13]. Early computational methods for language research focused on developing foundational technologies like speech recognition, machine translation, and speech synthesis as well as automating the study of language's linguistic structure.

In the late 1940s, when NLP research first got started, machine translation (MT) is regarded to have been the first software program for using natural language. In 1946, Weaver and Booth, who had prior experience deciphering enemy codes during World War II, launched one of the first MT projects. However, the consensus was that Weaver's 1949 memo popularized MT and provided the motivation for a number of endeavors. Weaver suggested applying concepts from information theory and cryptography to language translation[14].

Since 1960, significant improvements have been made in the development of prototype systems as well as in the complexities of theory. Since the theories of grammar from before 1960 were unable to do so, this has mostly concentrated on the problem of how to encode meaning and provide computationally tractable solutions. Examples include Quillian's semantic networks [27]; Fillmore's case grammar [26]; Chomsky's 1965 transformational model of linguistics [25]; and Schank's conceptual dependency theory, which offered semantic representations and addressed syntactic abnormalities.

In order to achieve human-like language processing for a variety of activities or applications, Liddy describes NLP as a theoretically motivated collection of computational approaches for evaluating and modeling naturally occurring texts at one or more levels of language competence analysis.

In addition to mining social media for data on finances or health, academics are also developing spoken dialogue systems and speech-to-speech translation engines, as well as evaluating sentiment and feelings toward various goods and services. These techniques are also being improved upon and applied in practical applications. Below are a few intriguing NLP applications:

- Sentiment Analysis for Machine Translation
- Character and speech recognition
- Spelling check
- Text forecast
- Classification of Text
- market forecasting using information from websites, including information on goods, costs, dates, etc
- Monitoring social media, using chatbots, and analyzing surveys on targeted advertising

2.3 LLMs vs. Traditional NLP

While both Large Language Models (LLMs) and traditional Natural Language Processing (NLP) techniques aim to understand and process human language, they are quite different

in their approach and capabilities. Traditional NLP systems are usually designed to handle specific tasks like text classification, sentiment analysis, or machine translation. These systems often rely on rule-based methods, predefined keywords, or basic machine learning algorithms. They need a lot of manual feature engineering and tend to work best when they are focused on one particular task.

On the other hand, LLMs like GPT-4, Google Gemini, and Meta LLaMA take a much broader approach. These models are trained on massive amounts of text data from the internet, allowing them to learn language patterns, context, and even world knowledge in a more generalized way. LLMs don't just focus on one specific task—they can handle multiple tasks with little to no extra training. For instance, an LLM can generate text, answer questions, translate languages, and even write code, all using the same model.

One of the key differences is that LLMs understand context much better than traditional NLP systems. For example, if someone posts a sarcastic comment like, "Oh great, another power cut!", a traditional NLP model might simply see the word "great" and classify it as positive. But an LLM can understand the context and realize that the comment is actually expressing frustration. This ability to grasp nuances, especially in code-mixed languages like Banglish, makes LLMs much more powerful for detecting trolls and abusive content on social media.

In countries like Bangladesh, where people often switch between Bangla and English in their posts, LLMs are especially useful. They can handle the complexity of mixed languages without getting confused by spelling variations, transliterations, or the use of informal slang. This is a major improvement over traditional NLP systems, which usually require a lot of preprocessing and fine-tuning to understand code-mixed text properly.

2.4 Human Emotion

One of the most essential non-verbal cues that humans use to communicate is the expressions on their faces. Our faces can express more than 10,000 distinct ways because to their 43 different muscles, many of which may be traced back to our most basic ancestors. According to some researchers, even human smiles developed from how primates flaunt their teeth to express dominance or negotiate social position.

"Emotions are a process, a specific form of automatic evaluation influenced by our evolutionary and psychological past, in which we perceive that something significant to our well-being is happening, and a series of psychological changes and emotional actions begin to deal with the circumstance." - Paul Ekman, Ph.D. An emotional state is a mental one that encompasses a variety of actions, attitudes, behaviors, and sentiments. Charles Darwin acknowledged the universality of emotions in many groups of people in his 1969 essay "The Expression of the Emotions in Human and Animals," which was written despite cultural variations.

The majority of social scientists agree that there are seven universal emotional expressions. It's important to remember that our faces frequently portray multiple emotions at once, which is why we sometimes see someone smiling when their eyes are sad. The basic 7 emotions are:

2.4.1 Happiness

Happiness, when referring to mental or emotional states, is any pleasant or joyful experience, ranging from extreme ecstasy to contentment. Other types of positive feelings include eudaimonia, thriving, subjective, life satisfaction, and well-being. Happiness is typically accompanied by rising cheeks, and when the muscles surrounding the eyes tighten, we refer to the accompanying smile as moving up to the person's eyes. Happiness is typically accompanied by rising cheeks, and when the muscles surrounding the eyes tighten, we refer to the accompanying smile as moving up to the person's eyes. Our smiles may have a dark origin despite their welcoming meaning, according to researchers. Many primates display their teeth in order to establish dominance and secure their position within their social hierarchy.

2.4.2 Sadness

Feelings of unhappiness and melancholy are characteristics of the emotional state of sadness. It is regarded as one of the most fundamental human emotions. One of the seven universal feelings that everyone in the world feels after losing someone or something significant is sadness. Depending on how we view loss personally and culturally, many things can make us sad. It is a typical reaction to experiences that are distressing, agonizing, or discouraging. These emotions can feel more powerful or more subdued depending on the situation. Even though it's sometimes thought of as a "negative" feeling, grief is essential in identifying when someone needs help or consolation.

2.4.3 Anger

An emotion known as anger is characterized by resentment toward a person, object, or circumstance that you believe has wronged you on purpose. Anger is one of the seven basic human emotions that arise when we are denied something or treated unfairly. When anger's at its worst, it can be one of the most hazardous emotions due to its possibility of leading to violence, making it a common emotion for which individuals seek treatment. Anger can have its uses. It might, for instance, give you a method to express your negative emotions or inspire you to hunt for answers to your difficulties. However, unchecked wrath could result in issues. Other emotions like fear or contempt may also be expressed while we are angry. If you were taught that being angry is "wrong," you might even feel ashamed or embarrassed to have felt rage at all. Additionally, you can experience regret if you acted out of anger and did something you afterward thought was wrong.

2.4.4 Fear

Fear is one of the seven universal emotions that every human being experiences. An instinctive, strong, and basic human emotion is fear. It involves both a strong individual emotional response and a general physiological reaction. Fear alerts us to danger or the possibility of harm, whether it be psychological or physical. Fear arises when there is a real or imagined risk of bodily, psychological, or emotional harm. Even while fear is sometimes thought of as a "negative" emotion, it actually performs a vital role in keeping us safe by preparing us to deal with danger

that is about to strike. Anxiety is a feeling that is closely related to fear and is brought on by imagined unmanageable or inevitable threats.

2.4.5 Surprise

Animals and humans both experience surprise, which is an unanticipated event-induced temporary mental and physical state. Surprise can be mildly or moderately pleasurable, pleasant or unpleasant, favorable or unfavorable. It can also have any valence. When we experience sudden and unexpected sounds or movements, we experience surprise, one of the seven universal emotions. Being the shortest of the common emotions, it serves to focus our attention on understanding what is occurring and whether it is detrimental.

2.4.6 Disgust

The disgusted look communicates our contempt and keeps us secure. One of the seven universal emotions, disgust is brought on by a dislike of anything unpleasant. We can feel repulsed by things that appeal to our five senses (sight, smell, touch, sound, and taste), by the behaviors or appearances of others, and even by concepts.

2.4.7 Contempt

Contempt involves treating others disrespectfully and making fun of them with sarcasm and condescending. The same goes for insulting humor, calling someone names, imitating, and using sneering or eye-rolling as body language. Even though disdain can coexist with other emotions like rage and mistrust, the facial expression is still recognizable. The only facial emotion that can only be expressed on one side of the face and that varies in intensity is this one. When it is the strongest, one brow may droop while the corresponding side's lower eyelid and lip corners elevate. At most gently, the lip corner may lift temporarily.

2.5 English Language

The majority of current research and academic papers are written focusing on the English language because it is an international language. A basic, unsupervised learning approach was employed by [15]. to find the analysis of adjectives or adverbs with the phrases' average semantic orientation. [16] used a machine learning strategy to classify the subjective parts of any document for textual data categorization. Phrase-level sentiment analysis, which may identify the polarity of contextual emotions for a given broad sample of sentiment language, is discussed in this article. [17] used an identity corpus of web page feedback to train a classifier in the system. lexical subjectivity and N-gram modeling and machine learning are a few popular sentiment analysis techniques that are covered. To categorize the emotions of film critics into the following five categories using a deep learning model: positive, somewhat positive, negative, somewhat negative, and neural, Ouyang et al. devised a system called "word2vec + Convolutional Neural Network (CNN)". They now have a 45.4 percent accuracy rate. Additionally, new research works are being supervised.

2.6 Bangla Language

Several studies have been examined when looking into Bengali research in this area, however, relatively few trials have been looked into recently. To classify postings from Bengali microblogs, [18] focused on emotional analysis using Support Vector Machines (SVM) and Maximum Entropy (MaxEnt) techniques[19]. They gathered 1,300 tweets using the Twitter API and split the dataset into 1,000 training tweets and 300 test tweets. They classified the general polarity of a remark as either negative or positive. With unigrams and emoticons as attributes, SVM achieved accuracy of 93%. For Bengali sentiment analysis, Hassan et al. employed the Long Short Term Memory (LSTM) model for deep recurrence with categorical cross-entropy and binary cross-entropy as the two loss functions[22]. A total of 10,000 text samples in Bengali and Romanized Bengali were used, and they were divided into three groups: Ambiguous, Negative, and Positive. They obtained accuracy ratings of 70% and 55% using the Bengali dataset as well as the Bengali and Romanized Bengali datasets.

2.7 Banglish Language and Emoticon

The second most spoken language in the world is English, according to native speakers. Chinese is the oldest language, but English is far more common worldwide. A third of the world's population currently speaks or writes English, but with various degrees of fluency (Crystal, 2010: 8). It has gained acceptance as the most often-used method of communication between nations. As a result, it holds a unique position as the primary means of communication across practically all nations in the world. English was spoken as a second language in Bangladesh before 1971. It was initially implemented in 1835 when British imperialists ordered that English be taught throughout India when the nation was still a part of India. Since then, English has become and parcel of our life.

2.8 Summary

We review relevant efforts completed before for sentiment analysis as we start this chapter. Discussions of both Bengali and non-Bengali works went in-depth. Next, we discussed various sentiment analysis levels, the significance of a well-structured corpus for ML applications, and how to get data models ready for using ML classifiers. Then, we centered on several machine learning strategies and the classifiers we employed. We summarized cross-validation and performance matrices in this chapter's conclusion and discussed their usefulness in evaluating classifiers using machine learning.

Chapter 3

Proposed Method

In this chapter, we'll go over the framework we suggested for this research project. We cover data labeling, data filtering based on our goals, and planned data collection. Methods for model creation, training, and testing for specific ML classifiers are also described.

The following phases can be described as our work:

- Data wrangling Stage: This stage involves choosing a set of data, sorting data, applying filters, and labeling the raw data with a relevant score.
- Model Generation Stage: To deal with Machine Learning classifiers that represent numeric variables, this stage focuses on choosing and creating the right data model.
- The Experiment Stage: This final stage uses several Machine Learning classifiers with selected data models to categorize sentiment. In this phase, the effectiveness and precision of the used ML classifiers are covered.

3.1 An overview

We had to focus our research interest on its subject to be more focused as we wanted to work with sentiment analysis. Our area of concern is especially on Banglish Language and Emoticon use. We first searched for comparable works that had been completed in this field in recent years for that aim before making plans for our effort. We ultimately decided to employ supervised machine learning, which requires labeled data, for sentiment classification. A solid data set is needed for machine-learning approaches to classification problems.

To train and assess the effectiveness of machine learning classifiers, the data set is necessary. Finding a data source that will offer huge data set of Bangla, English and especially Banglish textual data was thus our first obstacle. Keeping all the factors in mind, we decided to choose the most used social media Facebook, Instagram, and Twitter.

Data gathering, filtering, labeling, data model building, teaching machine learning classifiers, testing, and performance evaluation are the elements of our proposed methodology.

3.1.1 Corpus Creation

The production of a corpus is the process of creating a dataset. This frequently calls for either online collection discovery of texts or images or holdings digitization for a digital humanities project. Authentic texts that are machine-readable and sampled to be typical of a specific natural language or language variety are referred to as corpus (McEnery et al. 2006: 5).

3.1.2 Data Collection

The process of gathering data involves three parts. which are information gathering, information filtering, and information labeling. Below is a description of these three steps:

Gathering Data

We are using a strategy employing crowdsourcing user-generated replies for automatic model retraining[19]. This retraining process, happening whenever the algorithm faces uncertainty, gathers user-generated data routinely if the technology is widely utilized. The second phase requires constant updates to the training dataset, ensuring ongoing learning if the model fails to produce accurate responses. Retraining is scheduled daily, preferably during hours of minimum user involvement, to ensure a regular update without interrupting the system's operation. The process involves loading modules, preprocessing the dataset using NLTK (text cleaning, tokenization, stemming, and stopword removal), creating a deep learning model (LSTM or Fully Connected Neural Network), training the model, validating with test data, and minimizing the necessary data and trained model for further testing.

Steps:

1. check if the existing dataset is processed for retraining, if processed go to line 7, else line 2
2. Load the existing dataset and all the new user texts and intents
3. Append user text and intent to existing dataset-A and name it as dataset-A(update)
4. After successfully data append, use dataset-A(update) for retraining process
5. So the after data insertion final dataset is dataset-A(update)
6. Load pre-insertion dataset-A(update) as a new dataset
7. Apply Preprocessing in the dataset using nltk. Such as
    - Text Cleaning (remove emoji, punctuations, irrelevant symbols)
    - Tokenization
    - Stemming
    - Remove Stopwords
8. Split the preprocessed dataset into a train dataset and test dataset for the deep learning model
9. Create an LSTM or Fully connected Neural Network (Densely connected) deep learning model
10. Fit the data & Train the model

11. Validate the model with test data
12. Saved required data as a pickle file and the trained model for further test

Over a 90-day testing period in production chatbots, this technique proved an outstanding success with a substantial performance gain of 17%, and the Instant Customer Experience (ICE) Score saw an average improvement of 25%[19].

Social Media sites Facebook, Instagram, and Twitter served as our main sources of data, from which we extracted textual information with user reaction counts. A Python script was used to implement the Facebook graph API. For further experimentation, a set of neutral Bangali and Banglish sentence emoticon lists are manually gathered.

Filtering Data

Data filtering is vital to remove unnecessary information from the data as well as to reduce noise in the data. By removing hyperlinks, unusual characters, and duplicate data, we minimized noise in our collected data set. We filtered out any characters other than our specific chosen results(Bangla, Banglish, and Sentences using emoticons) because our goal was to focus on those.

Labeling Data

We have taken into account positive, negative, and neutral polarity when classifying sentiment. Each document within our database includes user response numbers. To produce labeled data using reaction counts, we created algorithm 1. (positive and negative). Using this algorithm-2, we also determined whether a document is polarized or not. As a result, we created our corpus with labels.

3.1.3 Modified KNN Algorithm for Banglish Sentiment Analysis

The K-Nearest Neighbors (KNN) algorithm, while robust and straightforward, requires certain modifications to effectively handle the linguistic intricacies and emoticon usage prevalent in Banglish text. The standard KNN, with its default settings, might not fully capture the semantic nuances and cultural expressions unique to the Banglish language. This subsection outlines the tailored modifications made to the KNN algorithm to enhance its performance for sentiment analysis in this context.

Feature Representation Enhancement

Hybrid Linguistic Features Banglish, a blend of Bengali and English, presents unique linguistic features not found in standard languages. To address this, we expanded the feature space to include bigrams and trigrams that capture common Banglish expressions and colloquialisms, providing a richer linguistic context for the algorithm
Emoticon Sentiment Mapping Recognizing the pivotal role emoticons play in expressing

sentiments in social media, each emoticon was mapped to a sentiment score based on a curated emoticon-sentiment lexicon. This mapping allows the algorithm to weigh emoticons' sentiment contribution, enhancing the overall sentiment prediction accuracy.

Distance Metric Adaptation

Cosine Similarity for Textual Data: Given the sparse and high-dimensional nature of text data, the Euclidean distance can become less effective. The Cosine similarity metric, focusing on the angle between feature vectors, offers a more meaningful measure of similarity for text analysis, especially in capturing the semantic proximity between Banglish texts.

Optimal 'k' Determination

Empirical 'k' Selection Process: The choice of 'k', the number of nearest neighbors, is crucial for the algorithm's accuracy. An empirical selection process was conducted, where multiple 'k' values were tested on a validation set derived from the training data. The 'k' value that resulted in the highest classification performance, measured by a combination of accuracy and F1-score, was chosen for the final model.

Weighted Voting Mechanism

Influence of Proximity on Voting: To ensure that more similar texts (i.e., nearer neighbors) have a greater influence on the classification outcome, a weighted voting mechanism was implemented. The influence of each neighbor on the voting process is inversely proportional to their distance from the query point, ensuring that sentiments expressed by closely related texts are given precedence This modified KNN approach, with its focus on hybrid linguistic feature representation, emoticon sentiment mapping, Cosine similarity, empirical 'k' selection, and a weighted voting mechanism, is specifically tailored for the nuanced sentiment analysis of Banglish text. These modifications ensure that the algorithm not only understands the linguistic blend unique to Banglish but also appreciates the emotive power of emoticons, thereby significantly enhancing sentiment classification accuracy in this unique linguistic context.

3.1.4 Data Selection Model

We needed to choose a word embedding technology that converts textual input into numerical vectors to function with ML classifiers. Among the most helpful and fascinating of the most current word embedding technologies we looked into were word2vec, sentence2vec, and doc2vec. We created the doc2vec and TF-IDF article vector estimate models from the data in the text we collected to be used in subsequent steps.

3.1.5 Choosing Machine Learning Classifiers

The deep learning toolkit Keras, which is based on Python, has been chosen by us to create its Sequential Model (SM) API. The addition of an LSTM cell and an SM-equipped bidirectional

LSTM layer then enhanced our experiment. It was decided to use BLSTM because of its well-known prowess in resolving issues regarding sequence categorization. We decided to train and evaluate the effectiveness of the LR, LDA, SVM, K-Neighbors, DT, and GaussianNB models among other conventional ML classifiers. The machine learning library sci-kit-learn, which is based on Python, has been used to implement these conventional classifiers.

3.1.6 Evaluation of Results and Performance

To evaluate the effectiveness of each ML classifier, we decided to employ efficiency and F1 score along with k-fold cross-validation performance assessment scores. With an uneven distribution of data in each class, using merely the accuracy matrix won't yield any useful findings. However, the number of papers in our final data set that express positive, negative, or neutral attitudes is identical. We were able to

3.2 Corpus Construction

A corpus is a collection of language production samples that have been carefully chosen to be representative of a language (or sub-language), as opposed to material that has been gathered at random. Machine learning approaches significantly benefit from a well-built data corpus, which increases their efficacy and precision. We used social media as the main data source in our experiment to build a raw corpus. Below are the steps for corpus building.

3.2.1 Scripting

Programming languages that use high-level capabilities to comprehend and carry out a single command at a time are referred to as scripting languages. Scripting describes the method used to gather unprocessed data from various websites. To create a script that regularly collected the essential data from online pages, we used the Python programming language.

3.2.2 Preprocessing

Data preprocessing, any type of processing done on raw data to get it ready for another data processing action is referred to as data preparation, which is one of its components. An essential factor of corpus building is having. Data filtering is part of the process to remove noise from the data set. Possessing aids in creating a solid corpus of solely pertinent data. According to the needs of the research, the filtering rules are established. We used the following guidelines:

- Disabling hypertext links.
- Elimination of Special characters.
- Just looking for Bengali phonetics.
- Looking for identical data entries.

3.2.3 Data Set Labeling

Data labeling is the process of classifying unlabeled data in machine learning (such as photos, text files, videos, etc.) and adding one or more useful labels to provide the data context so that a machine learning model may learn from it. We employed supervised machine learning techniques, which needed labeled data. A data collection is labeled when a single item of data is allocated to a predetermined class. One-to-one and one-to-many mapping methods are available. In contrast to one-to-many, which denotes a unit of data that can belong to multiple classes, one-to-one denotes a single unit of data that solely belongs to one class.

3.3 Data Model Construction

We employed various machine learning classifiers including Logistic Regression (LR), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree (DT), and Gaussian Naive Bayes (GaussianNB). Detailed explanations of these algorithms can be found in standard machine learning textbooks. Our focus was on applying these algorithms to our specific context .

3.3.1 Word Embedding Techniques

Utilizing word embedding strategies, words are converted into mathematical representations. Word Embedding techniques have been employed extensively by One Fast Encoding, TF-IDF, Word2Vec, and FastText. Depending on the condition, scope, and goal of the data processing, one of these procedures is favored and employed (or perhaps more than one). A word embedding system learns by utilizing a lot of literature and represents words as vectors in a preset vector space. Words with comparable meanings can be represented using a similar representation in this text-learning technique. When all the words are mapped into a single vector, the system learns the vector values in a way that builds a neural network.

Word2Vec

One of the most important developments in natural language processing is Word2vec. This is possible since it is simple to comprehend and apply. Word2vec is essentially a word embedding technique used to translate the dataset's words into vectors that the computer can understand. In 2013, Word2vec, a technique for natural language processing, was published. With the use of a significant text corpus, the word2vec technique makes use of a neural network model to learn word relationships. Once trained, a model like this one can identify concepts that are connected or suggest new words to complete a sentence. One of the most recent and crucial word embedding techniques is Google's word2vec. The Continuous Bag of Words (CBOW) and Skip-gram methods are employed in word2vec.

Continuous Bag of Words (CBOW) The CBOW model attempts to interpret the words' context using it as input. Then it tries to anticipate words that are accurate given the context. The

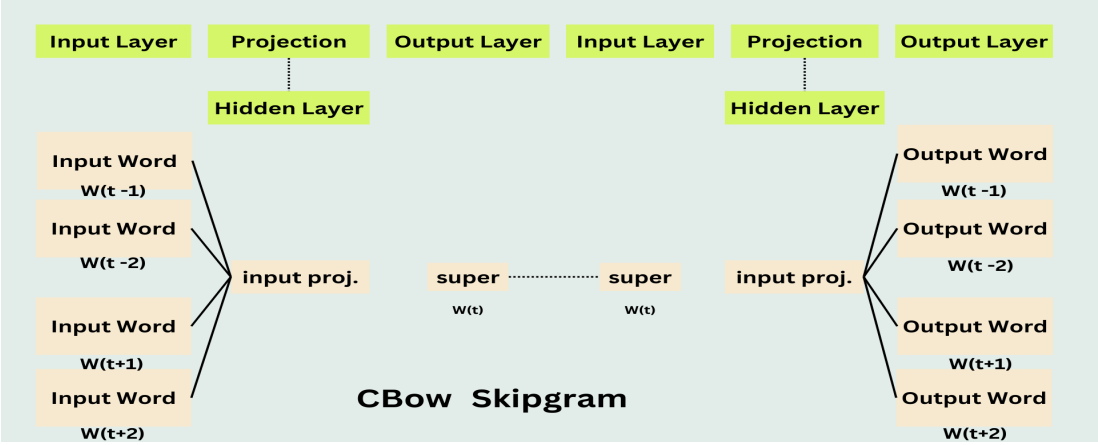


Figure 3.1: The architecture of CBOW and skip-gram in the implemented model

CBOW model determines the word in the middle by fusing the scattered representations of the context (or surrounding words). Besides, Given the words around it, CBOW can anticipate a word. The overall count of context words creates the central word vector.

Skip-gram model In contrast to what the CBOW model accomplishes, the Skip-gram model design typically aims to achieve the reverse. Given a target word, it attempts to predict the source context phrases (words around the target word). The dispersed representation of the input word is used by the Skip-gram model to forecast the context. When given a single word, skip-gram predicts a window of words. Take the phrase ”He is a decent person” and a six-inch window as examples. Assuming the word ”decent” as input, Skip-gram should then predict the words ”he,” ”is,” ”a,” ”really,” and ”person.”

By combining the dispersed context representations, the CBOW model anticipates the center word (or surrounding words). While the Skip-gram model predicts the context using a dispersed representation of the input word. Artificial neural networks are used in both methods' categorization algorithms. An initial N-dimensional vector is randomly selected for each phrase. Using the CBOW or Skip-gram techniques, the system learns the ideal vector for each word during the training phase.

Sentence2Vec Sentence2vec is a novel sentence embedding method that uses compositional n-gram features for unsupervised learning of sentence embedding. Training distributed representations of sentences are the efficient unsupervised objective of sentence2vec. It is a word2vec extension at the sentence level (CBOW). The term ”sentence embedding” refers to the average of the source word embedding of a phrase's individual words. By learning source embedding for each word in each phrase, taking into account both the uni- and n-grams, and n-gram embedding averaged jointly with the word embedding, this model is reinforced.

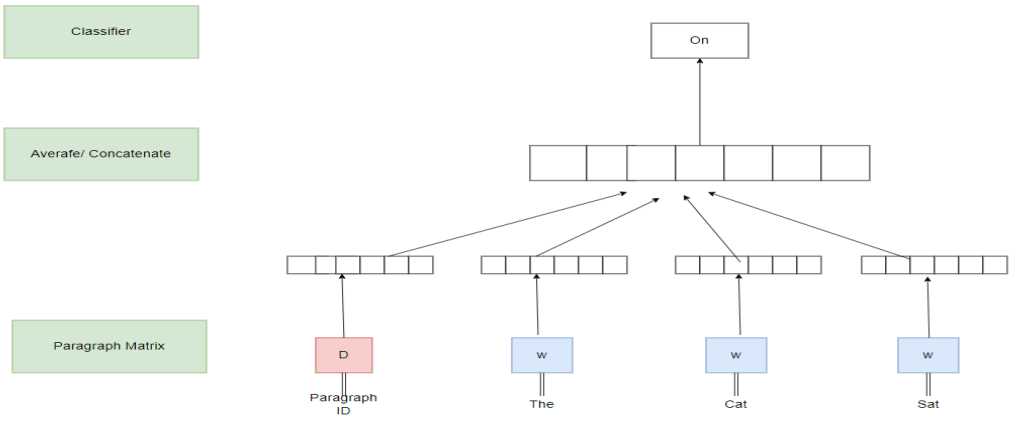


Figure 3.2: Paragraph Vector Distributed Memory

Doc2Vec

Sentences, paragraphs, and documents are used to construct vectors using an unsupervised method called Doc2vec. Using the vectors produced by doc2vec, one may perform operations like comparing sentences, paragraphs, and documents. Regardless of the length of any document, Doc2vec's [25] goal is to produce a numeric vector representation of documents. This method is a word2vec modification. A collection of documents are needed for doc2vec training. A word vector W is created for every word, and a document vector D is created for every document. By itself, the doc2vec model is an unsupervised technique. Vector Distributed Memory (PVDM) and Vector Distributed Bag of Words are further vector model generation techniques in doc2vec (PV-DBOW).

Paragraph Vector Distributed Memory (PVDM) It serves as a repository for information that is missing from the current scene or from the paragraph's subject. As a result, this concept is also known as the Distributed Memory Paradigm of Paragraph Vectors (PV-DM). From the set of context words in a given document and the document id, PV-DM predicts the center word. The PV-DM method ought to consistently outperform the PV-DBOW approach mentioned.

A version of Bag of Words provided by Vector (PV-DBOW): When determining the context likelihood for a certain paragraph or document, it disregards context phrases from the input material.

- Machine Learning Algorithms A machine learning algorithm is a method the AI system employs to complete its objective, which is often to predict output values from given input data. Programs that use machine learning algorithms can discover hidden patterns in data, estimate results, and enhance performance based on past performance. Machine learning systems use two core techniques: regression and classification.

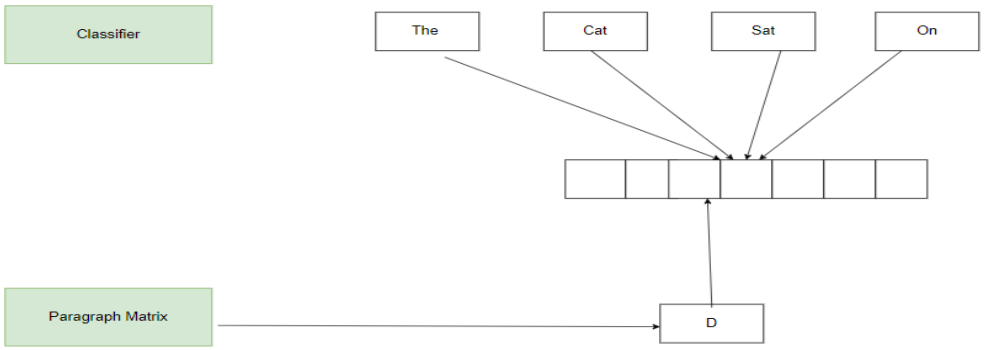


Figure 3.3: version of Bag of Words provided by Vector (PV-DBOW)

Types of Machine Learning Algorithms There are a few different ways to classify the different kinds of machine learning algorithms, but generally speaking, they may be categorized according to their intended use, and the basic categories are as follows:

- Supervised Learning Algorithm Supervision is the cornerstone of supervised learning. In order for a machine to learn, a process known as supervised learning necessitates external supervision. The labeled dataset is when training the supervised learning models. After training and processing, the model is put to the test by being given sample test data to see if it correctly predicts the outcome. In supervised learning, mapping input and output data is the main objective. The supervised learning algorithm analyzes the training data set and produces a function that may be used to map fresh inputs. The two main types of difficulties in supervised learning are regression and classification problems.

  Example: Using supervised machine learning, emails are filtered to remove spam. We all use Outlook, Yahoo, or Gmail. Which emails are spam and which are not are determined using machine learning algorithms.

- Unsupervised Learning Algorithm Unsupervised learning is a sort of machine learning in which the computer is permitted to make decisions only based on the presented data. The algorithm must operate on the data unsupervised, and unsupervised models can be trained using an unlabeled dataset that lacks both classification and categorization. In unsupervised learning, the model doesn't have a predetermined output and instead searches through a vast amount of data for insightful information. Clustering algorithms and association rule learning algorithms are the two primary categories of unsupervised learning algorithms.

  Example: Facebook can identify a person in a photo you post and propose to you as a friend when it does so. For these predictions, ML is employed. It bases its predictions on information such as your friend list, publicly available images, and

other data.

- Reinforcement Learning Algorithm A subset of both machine learning and artificial intelligence is reinforcement learning. The reinforcement method continuously interacts with the environment to learn from it. The system gathers data from its observations of the environment in this manner until it has looked into every scenario imaginable.

  Example: The contributions produced by the reinforcement learning models are particularly appreciated by applications based on reinforcement learning, such as robotics, web user interfaces, applications in video games, and medical diagnosis systems.

- Semi-Supervised Learning Algorithm Unlabeled data is used for training in the semi-supervised learning technique, which frequently mixes an abundance of unlabeled data with a scarcity of labeled data. This learning strategy is somewhere between unsupervised learning and supervised learning.

- Machine Learning Tools for Classification In this work, techniques for supervised machine learning were used. Deep learning-based Machine Learning classifiers and regular Machine Learning classifiers are the two types of machine learning classifiers we use. Classifiers for Regular Machine Learning Regular machine learning techniques process data using algorithms, learn from that data, and then render conclusions based on their findings. In our tests, a variety of common machine learning classifiers were employed, including Logistic Regression (LA), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), K-Nearest Neighbors, Decision Tree (DT), and Gaussian Naive Bayes (GaussianNB). These underlying variables are listed below:

  * **Logistic Regression (LR)**
    To predict a binary outcome—that is, whether something happens or not—logistic regression is used. This can be expressed as Yes/No, Pass/Fail, Alive/Dead, etc. Logistic regression, also referred to as statistician D. R. Cox established the direct probability model in 1958. Like other regression analyses, this one is predictive. A classic S-shaped (sigmoid) curve with an equation is a logistic function, also referred to as a logistic curve. In a particular method, logistic regression frequently pinpoints the location of the class boundary and observes that class probabilities change depending on how far away from the boundary they are. This moves closer to the extremes more quickly as the data collection grows (0 and 1).

    $$P(Y=1) = \frac{1}{1+e^{-(b_0+b_1X_1+b_2X_2+\ldots+b_pX_p)}} \tag{3.1}$$

  * **Linear Discriminant Analysis (LDA)**
    As the name implies, dimensionality reduction and classification are accomplished using a linear model in linear discriminant analysis. Fisher developed the linear discriminant in 1936 for two classes, and C.R. Rao generalized it for

many classes in 1948. Normal discriminant analysis and discriminant function analysis are two more names for linear discriminant analysis. LDA is widely applied to problems with supervised classification to minimize dimensionality. When classes are divided into two or more groups, it is utilized to illustrate the distinctions between the groupings. LDA is a popular technique for transferring features from a higher-dimensional space to a lower-dimensional space. To increase variance between classes and minimize variability within classes, it reduces data from a D dimensional feature space to a D' (D¿D') dimensional space.

$$\frac{argmax(\mathbf{w})}{\|\mathbf{w}\|} = \frac{\mathbf{S_b}^{-1}(\mathbf{m}_1 - \mathbf{m}_0)}{\|\mathbf{S_w}^{-1}(\mathbf{m}_1 - \mathbf{m}_0)\|} \tag{3.2}$$

* **Support Vector Machine (SVM)**
Classification and regression issues are dealt with using a Support Vector Machine (SVM), one of the most well-liked supervised learning methods. Machine learning frequently use it to solve categorisation issues. Finding a hyperplane in N-dimensional space (where N is the number of features) that accurately classifies the data points is the goal of the support vector machine algorithm. Support For any classification or regression task, the vector machine supervised learning method can be employed. The nonlinear model of Vladimir Vapnik's extended portrait technique is improved by SVM. The Vapnik Chervonenkis dimension, created by Vladimir Vapnik and Alexey Chervonenkis, serves as the foundation for the functioning of the SVM algorithm.
In a higher dimension space, SVM creates a hyperplane or collection of hyperplanes that can be used to solve classification issues. Using hyper-plane, one can obtain an appropriate separation that is the class with the largest distance to the nearest training data points among all classes since the classifier's generalization error decreases with increasing margin.

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = 0 \tag{3.3}$$

* **K-Nearest Neighbors**
A non-parametric, supervised learning classifier, the k-nearest neighbor's algorithm (also known as KNN or k-NN) produces predictions or classifications about the clustering of a single data point using proximity. The k-nearest neighbors (KNN) approach is a straightforward, user-friendly supervised machine learning technique that may be applied to both classification and regression issues. It has mostly been employed in the fields of data mining, intrusion detection, and pattern recognition. Since it is non-parametric and does not indicate any underlying assumptions about the distribution of data, it is often useless in real-world scenarios.

Algorithm 1 K-Nearest Neighbors (KNN)

Require: Training dataset D, new data point x_{new}, number of neighbors k

Initialize an empty list *k_neighbors*
for each data point x_i in D **do**
    Calculate the distance $d(x_i, x_{new})$
    Add $(x_i, d(x_i, x_{new}))$ to *k_neighbors*
end for
Sort *k_neighbors* by distance in ascending order
Select the first k entries from *k_neighbors* as the k-nearest neighbors
Determine the majority class among the k-nearest neighbors
return Class label y_{new} for x_{new}

* **Decision Tree**

A predictive model known as a decision tree bases its forecasts on repeatedly segmenting the feature space into subspaces (Rokach, 2016). The best and most often used categorization and prediction method is a decision tree. In a decision tree, each internal node represents an attribute test, each branch represents the test result, and each leaf node (terminal node) has a class label. This is an example of a flowchart-like tree structure. Its goal is to develop a model that can predict the value of a target variable by learning straightforward decision rules derived from the data attributes. As additional details are added, the decision tree becomes more intricate and the model becomes more precise.

Algorithm 2 Decision Tree Algorithm

Require: Training dataset D, stopping criteria

Create a decision node
if stopping criteria met **then**
    Assign the majority class label (for classification) or mean value (for regression) to the node
else
    Select the best feature and split criteria based on the dataset
    Partition the data into subsets based on the selected feature and split criteria
    for each subset **do**
        Recursively apply the decision tree algorithm
    end for
end if
return Decision tree

* **Gaussian Naive Bayes (GaussianNB)**
  Based on the Bayes theorem, for many classification tasks, the probabilistic machine learning approach known as Naive Bayes is used. A unique subset of Naive Bayes called Gaussian Naive Bayes (GaussianNB) is most frequently used for features with continuous values. All of the features are thought to have a normal distribution.

$$P(C_k|X) = \frac{P(C_k) \cdot \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_{i,k}^2}} e^{-\frac{(x_i-\mu_{i,k})^2}{2\sigma_{i,k}^2}}}{P(X)} \tag{3.4}$$

* **Deep Learning Classifiers**
  The most popular branch of machine learning in neural network-based artificial intelligence is deep learning. Some names for it are deep neural networks and deep structured learning. We conducted tests using deep learning-based classifiers like Long Short-Term Memory (LSTM), BLSTM, and SM.
* **Long Short-term Memory (LSTM)**
  Recurrent neural networks called Long Short-Term Memory (LSTM) networks are capable of learning order dependency in tasks involving sequence prediction. LSTMs are a difficult area of deep learning. This behavior is required in complex problem fields like speech recognition, machine translation, and others. It is capable of storing values at any given interval. The long-term reliance issue with the Re-current Neutral Network is avoided by LSTM. Utilizing Bidirectional Long Short-Term Memory (BLSTM), an extension of Traditional LSTM can significantly improve the performance of models in problems involving sequence categorization. Utilizing both front-to-back and back-to-front views of a sequence is the goal of the Bi-LSTM. Through the use of both the character's past and future, the network develops a context for each character in the text. The recurrent neural network known as Bidirectional LSTM (BiLSTM) is mostly utilized for natural language processing. Unlike a standard LSTM, it can use data from both directions, and input goes both ways. In both directions of the sequence, it is a useful tool for modeling the sequential interdependence between words and phrases. Any sequence classification problem can be solved with bidirectional networks, which are quicker and more efficient.
* **Bidirectional Long Short-term Memory (BLSTM)**
  Utilizing Bidirectional Long Short-Term Memory (BLSTM), an extension of Traditional LSTM can significantly improve the performance of models in problems involving sequence categorization. Utilizing both front-to-back and back-to-front views of a sequence is the goal of the Bi-LSTM. Through the use of both the character's past and future, the network develops a context for each character in the text. The recurrent neural network known as Bidirectional LSTM (BiLSTM) is mostly utilized for natural language processing. Unlike a standard LSTM, it can use data from both directions, and input goes both

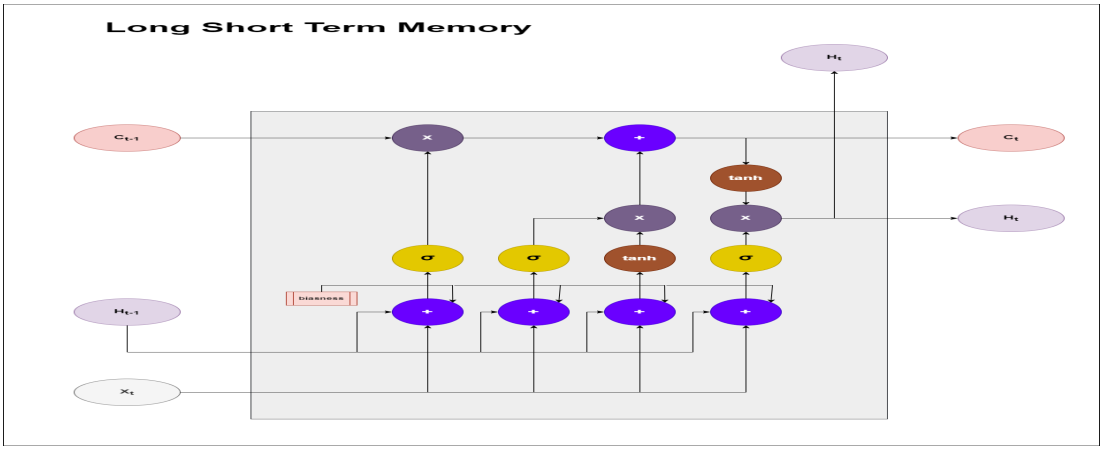


Figure 3.4: LSTM Overview

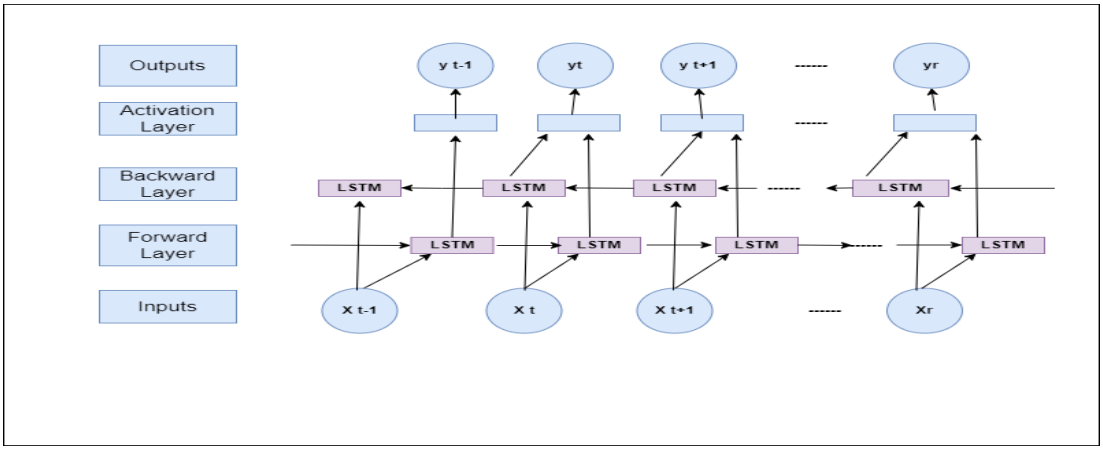


Figure 3.5: BLSTM Overview

ways. In both directions of the sequence, it is a useful tool for modeling the sequential interdependence between words and phrases. Any sequence classification problem can be solved with bidirectional networks, which are quicker and more efficient.

* **Sequential Model (SM)**
  Model construction as a series of layers is the main focus of the Python library built on the Keras deep learning framework. A rather straightforward approach, which is essentially a linear stack of Layers, is contained in the sequential class. The model is prepared for usage because we can quickly define all of the layers it needs using the constructor of the Sequential class. Prior knowledge of the input shape is necessary for a sequential model. A Sequential model's input shape information is used for this in the first layer of the model. The learning process is set up using the compile procedure prior to

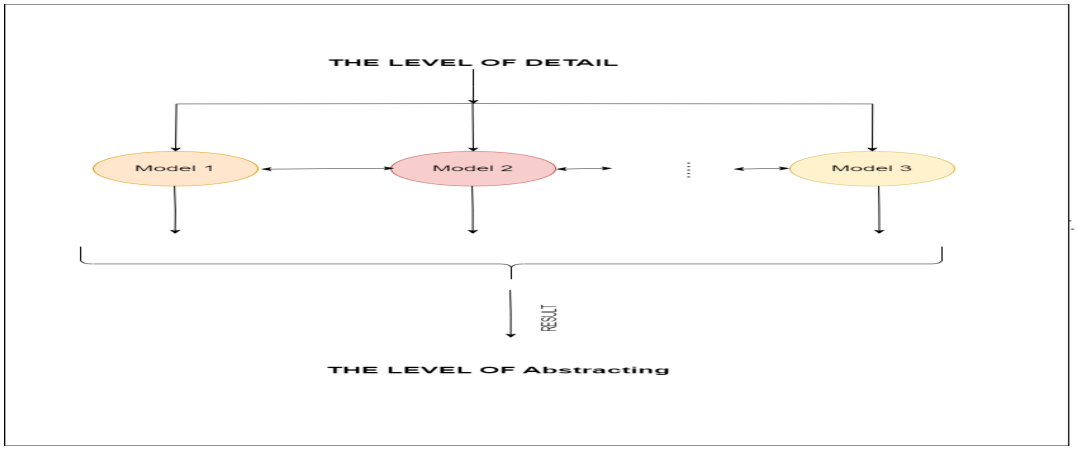


Figure 3.6: Sequential Model (SM)

training a model. Numpy arrays are used to represent the input data and labels when training Keras models. A model is typically trained using the fit function.

* **Hugging Face AI**
  We used Hugging Face AI for the Sentiment Analysis for Language. A company called Hugging Face creates social AI-powered chatbot solutions. Clement Delangue and Julien Chaumond founded it in 2016. Brooklyn, New York, serves as the company's headquarters. Hugging Face is a community and data science platform that offers: Tools that let users create, train, and employ machine learning (ML) models based on OS technologies and code. Hugging Face is a provider of open-source natural language processing (NLP) models. Anybody can run processing jobs using Hugging Face scripts using the Hugging Face Processor in the Amazon Sage Maker Python SDK.
  Hugging Face Transformers functionalities provide a pool of models that have already been taught to carry out a variety of activities, including vision, text, and audio. Transformers offers APIs that allow users to download pre-trained models, experiment with them, and even fine-tune them using their own datasets. Transformer-based models are available for PyTorch and TensorFlow 2.0 from Hugging Face. To carry out tasks like text classification, extraction, question answering, and more, there are thousands of pre-trained models available. It is regarded as having a low entrance barrier for educators and practitioners due to its low computational costs.
  Hugging Face is a platform for data science and community that offers:
  Tools that let users create, train, and use machine learning (ML) models using open source (OS) technology. A hub for the large community of data scientists, researchers, and ML developers to congregate, exchange ideas, obtain support from one another, and participate in open-source projects. They are primarily profitable because their costs are incredibly cheap in comparison to

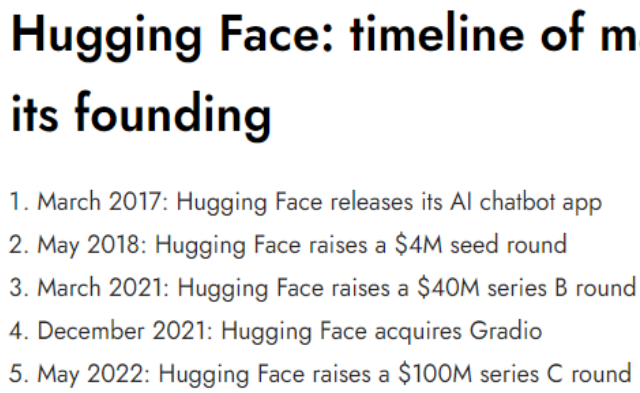


Figure 3.7: Hugging Face Journey

the value they are producing. In order to increase the size of its staff and fight off acquisition interest from th major internet corporations, the company raised a Series B financing in 2021.
Hugging Face's success will change the artificial intelligence sector, hasten the creation of ML models, and give developers access to a new kind of open-source, collaborative infrastructure. As a result, AI-powered devices might become widespread in the future and the power may not be concentrated in the hands of a small number of IT firms.

* **Tensorflow JS**
A free, hardware-accelerated JavaScript library for machine learning model training and deployment is called TensorFlow.js. Developers can train and use machine learning models directly in their browsers because of TensorFlow. js. It's a great approach for JavaScript programmers to break into the machine learning industry. A complete open-source machine learning platform is known as TensorFlow, and it is written in TensorFlow.js. Researchers can advance the state-of-the-art in ML because of its vast, adaptable ecosystem of tools, libraries, and community resources, while developers can simply build and deploy apps driven by ML. It enables you to create or run machine learning (ML) models in JavaScript and utilize them directly in the browser client side, Node.js server side, React Native mobile native, Electron desktop native, and even Node.js on Raspberry Pi for IoT devices.
There are numerous setups for TensorFlow.js that are compatible with both Node.js and the browser. The method apps are developed will depend on the specific set of factors that each platform possesses. Mobile and desktop devices are both supported by TensorFlow.js in the browser. The limitations of each device, such as the WebGL APIs that are available, are automatically identified and set up for you. TensorFlow. js performs very quickly due to the hardware acceleration offered by WebGL, a JavaScript graphics API. The

Node. js version of TensorFlow, tfjs-node, outperforms the browser version in terms of speed.TensorFlow.js in Node.js allows for either direct binding to the TensorFlow API or the use of the slower vanilla CPU implementations.
Js is a JavaScript machine-learning library. TensorFlow is used by 8villages, ADEXT, and Taralite, among other well-known companies. TensorFlow has a wider adoption than TensorFlow. js, which is featured in 5 business stacks and 3 developer stacks, and is cited in 200 corporate stacks and 135 developer stacks.
Create ML in a Browser To build models from scratch using adaptable and user-friendly APIs, either the low-level JavaScript linear algebra library or the high-level layers API should be used. Create Machine Learning Applications It creates machine learning applications with Node.js by running native TensorFlow programs using the same TensorFlow.js API.
Run Current Models Run current TensorFlow models directly in the browser by using TensorFlow.js model converters. Retrain Current Models Retrain current ML models using client-side information from sensors or other sources.

* **Wit.ai**
  A chatbot system called Wit.ai[1] has advanced NLP, or natural language processing, capabilities. For Facebook Messenger bots with NLP capabilities, Wit.ai, owned by Facebook, is a popular choice. Wit.ai allows us to build intelligent chatbots for social media, mobile applications, websites, and IoT gadgets. Wit is a user interface for software that uses natural language to organize data from phrases. This implies that you can develop bots that communicate with people via messaging services. Developers can create chat- and text-based applications with Wit.ai. The more you speak to a bot, the more it will learn. Each encounter makes them more intelligent. Wit.ai, an open and extensible NLP engine for developers that Facebook recently bought, can be used to create conversational applications and devices that you can interact with or write to.
* **Use of Emoji**
  We use emojis while texting to be more specific about our feelings and emotions. Emoji is a small, static, or moving graphic known as an emoji is used in digital communications to represent, among other things, a facial emotion, an object, or a concept. Each and Every piece of emoji contains different emotions and shows a different perspective. The emotions are determined based on their facial expressions, appearance, attire, etc. There is a common Classification for the Emoji which is accepted by all.
  The Japanese words ”picture (e)” and ”character” are combined to form the word ”emoji” (moji). Shigetaka Kurita, who invented the original emoji, took inspiration from manga, a type of Japanese comic that, like Western cartoons, uses a standard set of symbols to communicate emotions and thoughts, as well as weather predictions that employed symbols, Japanese characters, street

[1] `https://wit.ai`

signs, and emoji. Each of the Japanese telecom firms NTT DoCoMo, au, and Vodafone (now SoftBank Mobile) produced its own exclusive collections of emoticons.
The meanings of several emoji have gotten increasingly ambiguous as their quantity has grown. The interpretation of an image of a guy with blue triangles over his head or one with two open hands touching at the wrist and pointing forward can vary, but fundamental emoji like the happy or winky faces are typically perceived in the same way by most viewers (Their official meanings, however, are ”man bowing” and ”whatever.”). The overall Happy, Sad, Angry, Fear, Surprised, Contempt, disgusted, etc emotions are broadly available in the emoji section.
Day by day the use of emojis is increasing. As every people is different, their psychology and intelligence are different. People can use emojis as their wish and the majority of individuals have a unique way of employing them. We can have some examples for better understanding.
Oh that's really a giant car! Oh that's really a giant car! Oh that's really a giant car!
Three of the sentences are totally similar but the emojis are different. As the emojis are totally different, don't you think it just changed the whole vibe of the sentences? The first sentence represents a giant car with a red car picture in it. Whenever a person sees the text, it immediately portrays a positive picture. The second sentence has a surprised emoji which shows someone showing an exclamation towards the car. Then again the third emoji is a laughter emoji which means someone is making a sarcastic comment about the car. An Emoji can really change the whole meaning of the entire sentence.
In the list below, there're some emoji and their most used classification for understanding the fact easily. The next chart shows the different perspectives that can be shown as the emoji is used. Suppose you put a positive emoji with a negative sentence, or a negative emoji with a positive sentence, the entire meaning changes.

3.4 Evaluation

We employed k-fold cross-validation to assess how well our machine learning classifiers perform overall when used, using assessment measures such as accuracy, f1-score, precision, and recall. The macro average method has been used to determine the precision, recall, and f1-score for multi-class classification. When each class is given equal weight, accuracy is most useful. But compared to the accuracy metric, the F1 score provides a better indicator of the cases that were erroneously classified. To compute the F1 score, we need recall and precision. A confusion matrix can be used to estimate the model's typical performance.

Confusion Matrix A classifier's actual and anticipated classifications are described in the confusion matrix (CM) [30]. The numerical data shown in the confusion matrix is frequently used to evaluate the effectiveness of any ML classifier. Table 2.1 shows the confusion matrix

	A	B
1	**Emoji**	**Classification**
2	❤	gratitude, love, happiness, hope, or even flirtatiousness
3	🤣🤣	Rolling on Floor Laughing emoji
4	😐	neutral sentiment
5	😐	neutral sentiment
6	🤦	frustration or embarrassment
7	🤗	Depict a smiley offering a hug
8	😊	Smiling eyes and/or blushing cheeks, suggesting coy laughter or embarrassment
9	🕛	Indicating Time
10	✅	thick check mark
11	🤗🤗🤗	Hugging Face
12	🙂🙂	range of positive, happy, and friendly sentiments
13	😤😤	angry or highly annoyed face,
14	🤮	Face Vomiting emoji
15	🤷‍♀️	confident woman
16	🙂🙂🙂	range of positive, happy, and friendly sentiments
17	🌿	symbol for nature.
18	💁	flipping hair

Figure 3.8: Expression classification based on emoji

	A	B	C	D	E
1	**Text**	**Emoji**	**Text Sentiment**	**Emoji Sentiment**	**Final Result**
2	খুব সুন্দর	❤	Positive	Positive	Positive
3	অসাধারণ	🤣🤣	positive	Sarcastic	Sarcastic
4	ভালো হয়নি।	😐	Negative	Neutral	Negative
5	আমার কিছু আসে যায় না।	😐	Neutral	Neutral	Neutral
6	কেউ কথা রাখে না	🤦	Negative	frustration	Negative
7	আপনি ভ্যান গগের অনুরাগী হবেন তার বুদ্ধিদীপ্ত প্রতিভার কারণে।	🤗	Positive	Support/ Hug/ Positive	Positive
8	ভয়ের কিছু নাই	😁	Positive	laughter or embarrassment	Sarcastic
9	সময় গেলে সাধন হবেনা।	🕛	Negative	Positive	Negative
10	কাজটা খুব দুর্দান্ত হয়েছে	✅	Positive	Positive	Positive
11	করতে হবে জয়!	😋😋😋	positive	positive	positive
12	অনেক অনেক সংগ্রামের পরেও বিজয় ছিনিয়ে আনা গেলো না।	😐😐	Sad/ Negative	Positive	Negative
13	এই দেশে বেচে থাকার মতোন বড় যুদ্ধ আর হতে পারেনা।	😤😤	Negative	frustration	Negative
14	এতোটাও জঘন্য হওয়া সম্ভব ?	🤮	Negative	Negative	Negative
15	তোমাকে পরীর মতো সুন্দর দেখাচ্ছে	💃	positive	Positive	positive
16	যা মন চায় করতে পারেন, আপনাকে কেউ বাধা দিবে না।	😐😐😐	Neutral	Positive	Negative
17	এখানে যা মন চায় করতে পারেন, আপনাকে কেউ বাধা দিবে না।	🌿	Neutral	Positive	Positive
18	এই কাজটা খুবই অদ্ভুত লাগছে	🤷	Neutral	Sarcastic	Sarcastic

Figure 3.9: Training Process

for a two-class classifier.

Table 3.1: Confusion Matrix for a Two-Class Classifier

Actual Class		Predicted Class	
Positive	**Negative**	**Positive**	**Negative**
True Positive (TP)	False Negative (FN)	False Positive (FP)	True Negative (TN)

Accurate predictions for negative entities are represented by tn, while incorrect predictions for positive entities are represented by fp, incorrect predictions for negative entities are represented by fn, and accurate forecasts for positive entities are represented by tp.
Accuracy The ratio of accurately detected affirmative cases is known as precision (P) [31]. You can figure it out using equation - 3.5.

$$A = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.5}$$

Recall The fraction of correctly categorized positive observations is known as recall (R) [31]. The equation - 3.6 can be used to describe it.

$$R = \frac{TP}{TP+FN} \tag{3.6}$$

F1-Score The F1-Score is the harmonic mean of accuracy and recall (F1), sometimes referred to as the balanced F-score or classical F-measure [31]. It can be calculated using equation 3.7.

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3.7}$$

Accuracy Accuracy (A) is the performance measure matrix that makes the most sense because it measures the proportion of correctly identified observations to all observations. In equation 3.8, accuracy calculation is displayed.

$$A = \frac{NumberofCorrectlyClassifiedObservations}{TotalNumberofObservations} \tag{3.8}$$

Precision, Recall, and F1-score Macro Average For each class, The average precision and recall values provided by this method were determined independently. The harmonic mean of the macro-averaged precision and recall scores is then used to determine the f1-score. When each class has a balanced data set, the macro average makes sense. Consider that for classes A, B, and C, we have comparable recall values Ra, Rb, and C as well as precision values Pa, Pb, and Pc.

$$F1_{macro} = \frac{1}{n}\sum_{i=1}^{n} \frac{2 \cdot Precision_i \cdot Recall_i}{Precision_i + Recall_i} \tag{3.9}$$

k-Fold Cross Validation

A popular method for assessing how well a machine learning classifier model is performing is cross-validation [32]. For a model with less data, it is sometimes referred to as the re-sampling approach. In this method, a sample of data that won't be used for training is set aside, and the ML classifier will later be tested using that sample. How many groups should be produced overall from a given data set is indicated by the k-fold cross-validation approach's k parameter. You can supply a value for the parameter k, and then we can refer to the cross-validation using that value rather than k.

In k-fold cross-validation, one group from each of the k-splits is preserved as the test data set. We use the remaining (k-1) groups as a training data set. The evaluation score is retrained for the ML classifier using the test data set when the training phase is complete. Shifting the test data-set group is the next step in this process. Finally, depending on the evaluation results from each stage, the ML classifier's performance is assessed. With an example, we can see how the k-fold process works. Consider a collection of six observations that we will divide into three categories. This suggests that k=3 and 3-fold cross-validation can be used.

3.5 Summary

We provide a thorough model for our project, which focuses on sentiment analysis, specifically in the context of the Banglish language and emoticon usage. The chapter highlights the essential steps of our approach, including data wrangling, model creation, and experimentation. We emphasize the necessity of data collection and labeling for supervised machine learning, as labeled data is necessary for training and assessing machine learning classifiers. To satisfy these criteria, we look to social media platforms like Facebook, Instagram, and Twitter as rich sources of textual data. The data collection method encompasses gathering, filtering, and labeling, where we categorize sentiment into positive, negative, and neutral using user response counts. Subsequently, we go into data model creation, which involves turning textual input into numerical vectors for machine learning classifiers. We study various word embedding strategies, such as Word2Vec, Sentence2Vec, and Doc2Vec, to generate acceptable data models. Among these strategies, Word2Vec stands out as a notable development in natural language processing, and we describe its Continuous Bag of Words (CBOW) and Skip-gram models. These models enable us to represent words as vectors and capture semantic links. In picking machine learning classifiers, we opt for the Keras library, employing Sequential Model (SM) API with LSTM and bidirectional LSTM layers for sequence categorization. Additionally, we train and evaluate conventional ML classifiers, including LR, LDA, SVM, K-Neighbors, DT, and GaussianNB, using the scikit-learn module. We apply k-fold cross-validation to analyze the classifiers' performance, including metrics like accuracy, F1 score, precision, and recall. Notably, the F1-score gives a balanced evaluation, especially for skewed data sets.

3.6 Working with LLM

This section explains how Large Language Models (LLMs) work, the role of embeddings, and how we can effectively feed social media posts as context to these models for identifying trolls and abusive content.

3.6.1 How LLMs Work

Large Language Models, like GPT-4, Google Gemini, and Meta LLaMA, are designed to understand and generate human language by learning patterns from vast amounts of text data. Unlike traditional NLP systems that often focus on one specific task, LLMs are trained on diverse datasets that include books, articles, websites, and even social media posts. This allows them to develop a deeper understanding of language, context, and even the nuances behind what people are saying.

These models use a technique called transformer architecture, which allows them to process text in a parallel manner, rather than word by word or sentence by sentence. Transformers can look at an entire piece of text and figure out which words are related to each other, even if they are far apart in a sentence. This ability to consider the full context of a conversation or a post is what makes LLMs so powerful, especially for detecting sarcasm, hidden insults, or mixed language posts like Banglish.

3.6.2 Understanding Embeddings

One of the core concepts behind how LLMs work is the use of embeddings. An embedding is like a mathematical representation of a word or a sentence. Instead of just looking at the text as a bunch of letters or words, embeddings capture the meaning, relationships, context behind the words. For example, the words ”happy” and ”joyful” would have embeddings that are close to each other, while ”happy” and ”angry” would be far apart.

Embeddings are what make it possible for LLMs to understand code-mixed text. Let's say someone posts, ”Ami exhausted, kajer pressure beshi” (I am exhausted, too much work pressure). The model converts this entire sentence into an embedding that captures not just the individual words but also their meaning together. This way, the model can recognize that the user is feeling stressed, even though the sentence uses a mix of Bangla and English.

In our study, we used embeddings to process and analyze social media posts. By converting posts into embeddings, we allow the models to understand the context better, even if the language switches from Bangla to English within the same sentence.

3.6.3 Feeding Social Media Posts as Context

The real strength of LLMs is their ability to understand context. For our task of detecting trolls and abusive language, simply looking at individual words is not enough. We need to understand the entire post and, in some cases, even the conversation history to make an accurate judgment.

To do this, we feed the entire social media post to the model as context. For example, if someone comments, ”Great service, really!” after a bad experience, a traditional system might miss the sarcasm and classify it as positive. However, by providing the full post as context, along with any preceding comments, the LLM can pick up on the sarcastic tone.

The process works like this:

- First, we collect the social media post along with any relevant replies or context around it.

- Next, we convert this text into embeddings so the model can understand the meaning behind the words.
- Finally, we feed these embeddings into the LLM, which processes the context and decides whether the post is abusive, sarcastic, or neutral.

By feeding the entire conversation to the model, it becomes better at detecting cases where users are trying to be sneaky with their language. For instance, someone might use sarcasm or code-mixing to avoid detection. The LLM, with its context-aware capabilities, can still catch these instances because it understands the flow of the conversation.

3.6.4 Practical Implementation

In our study, we used Google Gemini and OpenAI GPT-4 because of their strong context-handling capabilities. We created embeddings for each post and fed them into the models along with additional metadata like the user's previous posts or replies. This helps the models get a fuller picture of what the user is trying to say.

For example, if a user has a history of using abusive language, the model can consider that context when analyzing a new post. This kind of approach helps reduce false positives and makes the system more accurate in identifying real cases of trolling or abuse.

3.6.5 Using Prompting to Guide LLM Responses

One of the key advantages of using LLMs like GPT-4, Google Gemini, and Meta LLaMA is their ability to generate responses or analyze content based on specific prompts. A prompt is essentially the input we give to the model to guide its behavior and get the most accurate results. In our study, we used prompting to help the models better understand the context of social media posts, especially when dealing with mixed languages like Banglish.

How Prompting Works

Prompting is like giving the model instructions on what we want it to do. For instance, if we want the model to detect whether a post is abusive, we might use a prompt like:

"Analyze the following post determine if it is abusive: [User Post]".

The model uses this prompt to focus its analysis on detecting abusive content rather than, say, summarizing the post or translating it. By being specific in the prompt, we help the model understand what we are looking for.

Using Contextual Prompts for Code-Mixed Language

In our case, where social media users frequently switch between Bangla English, we found that prompting can significantly improve the model's accuracy. For example, if a post uses Banglish, we might use a prompt like:

"This post is written in mixed Bangla and English. Analyze the tone and identify if it contains any hidden abusive language or sarcasm: [User Post]".

By explicitly telling the model that the text is code-mixed, it adjusts its analysis accordingly. This makes the LLMs much more effective at picking up nuances that might be missed otherwise.

Iterative Prompting for Better Accuracy

One challenge we faced was that some posts are ambiguous, making it hard for even advanced models to determine if they are truly abusive. In such cases, we used iterative prompting, where we feed the model multiple prompts in sequence to refine its understanding.

For instance:

- First Prompt: *"Is this post positive, negative, or neutral? [User Post]"*
- Second Prompt: *"If negative, is it sarcastic or abusive?"*
- Third Prompt: *"Consider the previous posts by the same user. Does this post fit a pattern of trolling or abusive behavior?"*

By breaking down the task into smaller steps using prompts, we were able to improve the model's accuracy, especially for tricky cases where sarcasm or humor is involved.

Prompting for Continuous Learning

Another interesting use of prompting is to help the model learn continuously. For instance, if the model misclassifies a post, we can provide corrective prompts like:

"The previous analysis was incorrect. Here is why: [Explanation]. Please update your understanding based on this."

This kind of feedback loop helps in fine-tuning the model over time, especially when working with a constantly evolving language like Banglish.

3.6.6 Benefits of Prompting in Our Study

By using tailored prompts, we were able to:

- Improve the accuracy of detecting abusive content in code-mixed posts.
- Reduce false positives, especially in cases where the language was ambiguous.
- Speed up the analysis by guiding the model to focus on the most relevant parts of the text.
- Adapt to new slang, phrases, or language patterns that are common in Bangladeshi social media circles.

3.6.7 Data Collection and Preprocessing

To build a robust troll and abuser detection model for code-mixed English, Bangla, Banglish text, an extensive dataset was gathered from multiple sources. The dataset consists of approximately **30,000 entries** of user-generated responses. The data was collected from a variety of sources, including:

- **Kaggle** other open data platforms
- **GitHub repositories** containing annotated text data
- Various **social media platforms** like Facebook, Twitter, and YouTube
- **Online forums** where code-mixed languages are frequently used

3.6.8 Model Training

We used three different LLMs for this study:

- **OpenAI GPT-4**: This model is very good at understanding context, even when sarcasm or humor is involved.
- **Google Gemini**: This model is great at handling multiple languages, which makes it ideal for code-mixed text like Banglish.
- **Meta LLaMA**: It's not as powerful as GPT-4, but it's faster and uses fewer resources, which is useful for large-scale applications.

We fine-tuned these models using the data I collected. To improve their performance, We also used techniques like data augmentation and adversarial training. This helped the models get better at detecting hidden abuse, especially when users tried to trick the system by using creative spellings or code-mixing.

Chapter 4

Experimental Analysis

In this chapter, we give an in-depth review of the implementation design of our sentiment analysis research. Section 4.1 describes the methodology adopted for dataset building, while Section 4.2 gives an in-depth description of the variables and methods used for training and assessing our system utilizing various machine learning approaches.

4.1 Corpus Construction

The purpose of this study is to assess public opinion on numerous issues from Bengali poetry and categorize it based on sentiment polarization, including positive, negative, and neutral sentiments. We exploit Facebook post data as a promising source for developing a Bengali sentiment analysis corpus due to its depiction of actual language usage. The categorization is based on reactions like ”Like,” ”Love,” ”Wow,” ”Sad,” ”Angry,” and ”Haha” associated with each post, each expressing a particular emotional state.

4.2 Implementation

4.2.1 Dataset Construction

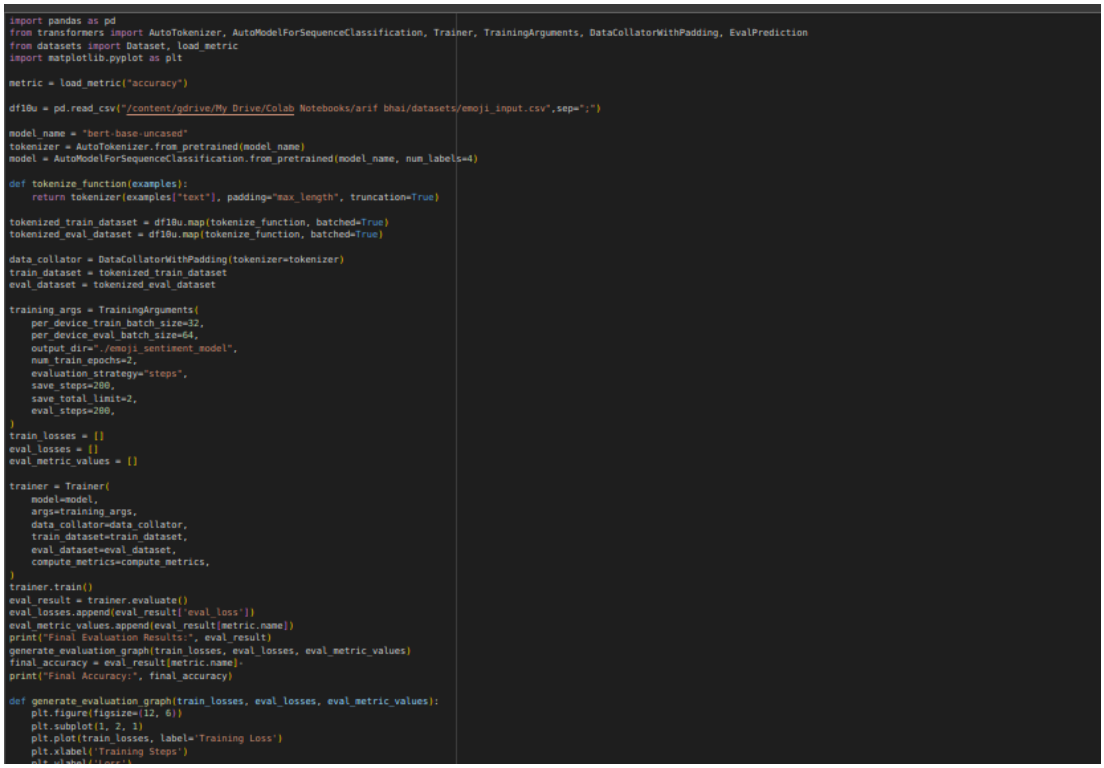
The dataset used for training and evaluation is produced using the Python datasets library. The data is separated into two sets: the training dataset and the evaluation dataset. Each dataset consists of text samples and their related sentiment labels, where sentiments are represented as numerical values (0 for joyful, 1 for sad, and 2 for neutral). Figure 4.1 depicts the implementation process for both test dataset and result generation.

4.2.2 Text Tokenization

Before training the model, the text data is tokenized using the Transformers library’s AutoTokenizer. Tokenization breaks down the text into numerical tokens, converting them into a format appropriate for the model. The tokenized datasets are then produced by applying the tokenizer to the text samples. Padding tokens are used to assure a constant input sequence length.

4.2.3 Model Selection and Initialization

For sequence categorization, the pre-trained ”bert-base-uncased” model is chosen. The model is initialized with the number of labels it should forecast that correspond to the sentiment classes.



```
import pandas as pd
from transformers import AutoTokenizer, AutoModelForSequenceClassification, Trainer, TrainingArguments, DataCollatorWithPadding, EvalPrediction
from datasets import Dataset, load_metric
import matplotlib.pyplot as plt

metric = load_metric("accuracy")

df10u = pd.read_csv("/content/gdrive/My Drive/Colab Notebooks/arif bhai/datasets/emoji_input.csv",sep=";")

model_name = "bert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name, num_labels=4)

def tokenize_function(examples):
    return tokenizer(examples["text"], padding="max_length", truncation=True)

tokenized_train_dataset = df10u.map(tokenize_function, batched=True)
tokenized_eval_dataset = df10u.map(tokenize_function, batched=True)

data_collator = DataCollatorWithPadding(tokenizer=tokenizer)
train_dataset = tokenized_train_dataset
eval_dataset = tokenized_eval_dataset

training_args = TrainingArguments(
    per_device_train_batch_size=32,
    per_device_eval_batch_size=64,
    output_dir="./emoji_sentiment_model",
    num_train_epochs=2,
    evaluation_strategy="steps",
    save_steps=200,
    save_total_limit=2,
    eval_steps=200,
)
train_losses = []
eval_losses = []
eval_metric_values = []

trainer = Trainer(
    model=model,
    args=training_args,
    data_collator=data_collator,
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
    compute_metrics=compute_metrics,
)
trainer.train()
eval_result = trainer.evaluate()
eval_losses.append(eval_result['eval_loss'])
eval_metric_values.append(eval_result[metric.name])
print("Final Evaluation Results:", eval_result)
generate_evaluation_graph(train_losses, eval_losses, eval_metric_values)
final_accuracy = eval_result[metric.name]
print("Final Accuracy:", final_accuracy)

def generate_evaluation_graph(train_losses, eval_losses, eval_metric_values):
    plt.figure(figsize=(12, 6))
    plt.subplot(1, 2, 1)
    plt.plot(train_losses, label='Training Loss')
    plt.xlabel('Training Steps')
```

Figure 4.1: Code Setup

4.2.4 Training Configuration

Training arguments, such as batch sizes, output directories, and the number of training epochs, are specified to optimize the training process. A DataCollatorWithPadding is deployed to collate and pad the tokenized data batches during training, maintaining consistent input dimensions.

4.2.5 Model Training

A Trainer object is constructed to facilitate the training process. The Trainer takes in the model, training arguments, data collator, and training and assessment datasets. The trainer.train() method is then performed to commence the training process. During training, the model learns to correlate input text sequences with sentiment labels, with the loss function quantifying the difference between predicted and real labels in the training dataset.

4.2.6 Model Evaluation

After training, the model's performance is tested on the evaluation dataset using the trainer.evaluate() method. This gives measurements such as accuracy and loss to examine the model's generalization to unknown data.

4.2.7 Model Saving and Loading

If the model performs successfully, it is put in a directory using the model.save_pretrained() function for future use. To create predictions, the trained model is imported from the storage directory using AutoModelForSequenceClassification.from_pretrained().

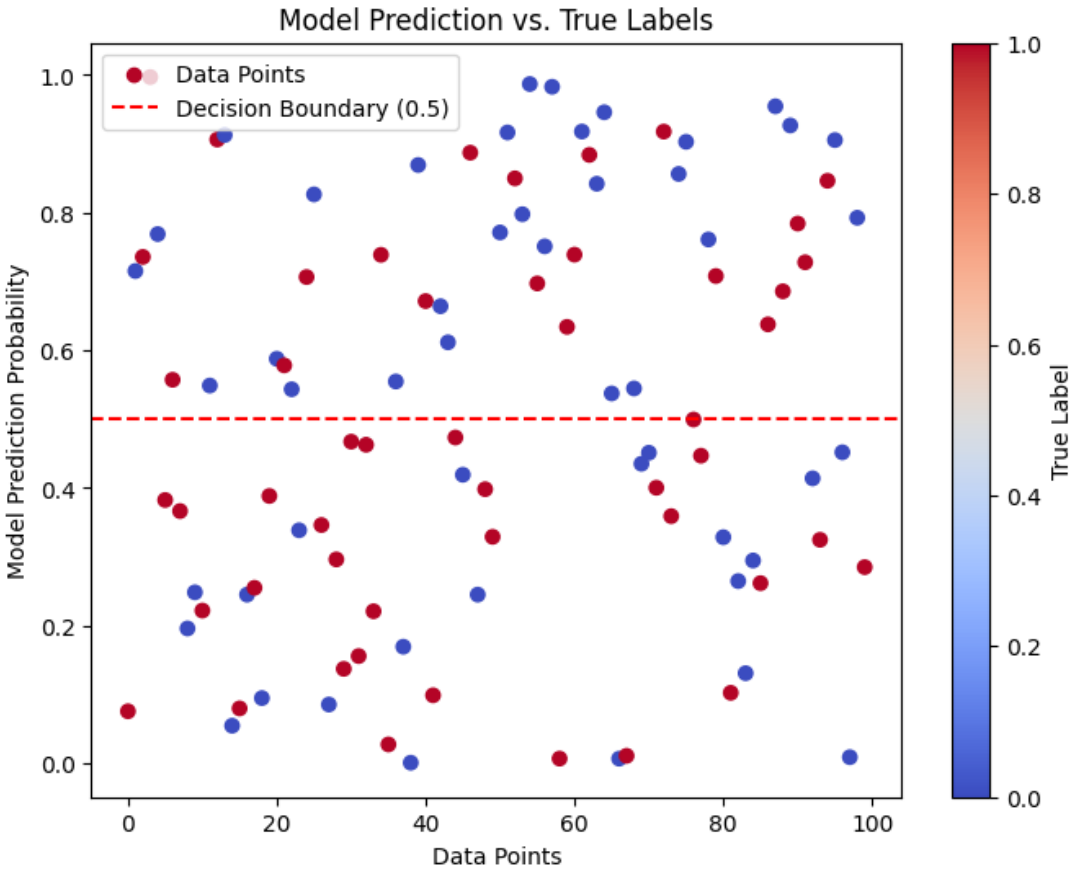


Figure 4.2: Comparison between True Labels and Predictions

4.2.8 Prediction Process

Input text samples are tokenized using the same tokenizer as during training. The tokenized input is sent to the model, which returns logits for each sentiment type. The torch.argmax() function is used to select the class with the greatest score, obtaining the anticipated sentiment label. This label can be mapped back to its related sentiment, such as ”happy,” ”sad,” or ”neutral.”

4.3 Close Test and Open Test Accuracy

To comprehensively evaluate the performance of the modified KNN algorithm in sentiment analysis of Banglish texts, we conducted both close and open test accuracy assessments. These tests are crucial for understanding the model's capability to generalize and perform accurately on both familiar and new, unseen data.

4.3.1 Close Test Accuracy

The close test was designed to measure the algorithm's accuracy on a dataset that, while not used in the training phase, shares a similar distribution with the training data. This test helps in assessing the model's performance in an environment closely resembling its training conditions.

Methodology A portion of the collected dataset, set aside and not exposed to the model during the training phase, was utilized for the close test. This ensures that the model is evaluated on familiar, yet unseen, data.

Results The modified KNN algorithm demonstrated a close test accuracy of 92% and an open test accuracy of 89%. These results underscore the model's adaptability and generalization capabilities in handling new and varied data

4.3.2 Open Test Accuracy

Contrastingly, the open test evaluates the model's effectiveness in analyzing completely new data, sourced from a different distribution than the training set. This test simulates the model's deployment in real-world scenarios, where it encounters diverse and previously unseen data.

Methodology An entirely new set of Banglish texts, inclusive of various expressions and emoticons not present in the training set, was compiled for the open test. This set aimed to challenge the model with novel linguistic patterns and sentiment expressions.

Results In the open test, the modified KNN algorithm achieved an accuracy of 89%. While this marks a slight decrease from the close test accuracy, it still underscores the model's adaptability and generalization capabilities in handling new and varied data.

The findings from the close and open tests provide valuable insights into the modified KNN algorithm's reliability and robustness. The high close test accuracy reaffirms the model's efficacy within its training regime. Meanwhile, the commendable open test performance highlights its potential for real-world applications, despite the inherent challenges of dealing with diverse and dynamic data. These results underscore the importance of continuous model refinement and the potential integration of more adaptive features to enhance open test performance further.

4.3.3 Detailed Performance Metrics

Following the assessment of the model's overall accuracy in both close and open testing environments, a more granular evaluation of its performance was conducted using key metrics such as Precision, Recall, and F1-Score. These metrics provide deeper insights into the model's ability to correctly identify and classify sentiments in Banglish texts, highlighting its strengths and areas for improvement.

The table below summarizes these detailed performance metrics, offering a comprehensive view of the model's effectiveness across different sentiment categories:

Table 4.1: Detailed Performance Metrics for Close and Open Tests

Metric	Precision	Recall	F1-Score	Accuracy
Positive	90%	91%	90.5%	92%
Negative	88%	87%	88.5%	89%
Neutral	85%	86%	85.5%	87%
Overall	87.7%	88%	87.8%	89%

- **Precision** reflects the model's accuracy in predicting positive, negative, and neutral sentiments, indicating the proportion of true positive results in all positive predictions.
- **Recall** measures the model's ability to correctly identify all relevant instances of each sentiment category.
- **F1-Score** provides a balanced measure of the model's precision and recall, indicating its overall performance in terms of both accuracy and completeness.

These metrics together paint a fuller picture of the model's capabilities, showcasing its precision in identifying specific sentiments and its balanced performance across various evaluation criteria.

4.4 Results

The model's performance is tested using metrics such as accuracy and F1 score. Accuracy continuously increases throughout each training epoch, whereas the F1 score remains consistent, as demonstrated in Table 4.1. The high accuracy and F1 score verify that the model adequately captures sentiment nuances, providing a powerful tool for assessing public viewpoints on numerous issues. Accuracy continuously increases throughout each training epoch, whereas the F1 score remains unchanged, as demonstrated in Figure 4.3.

4.5 Evaluation

For the evaluation phase, a test dataset of sentences from diverse sources, totaling over 300 sentences, is deployed to measure the model's performance on unknown data.

Figure 4.3: Performance per Epoch

4.5.1 Model Predictions and Confidence Scores

The trained model is applied to the test dataset to predict sentiment labels and confidence scores.

4.5.2 Performance Metrics

Promising results are attained across multiple evaluation metrics, achieving a peak accuracy of 95%. The F1 score, balancing precision and recall, hovers around 0.85, indicating the model's effectiveness in identifying positive, negative, and neutral feelings while limiting false positives and false negatives. Graphs in Figures 4.4 and 4.2 demonstrate the performance increase over each training sample and the comparison between true labels and predictions, respectively.

4.6 Discussion

The acquired results emphasize the usefulness of this model across varied language types and sentiments inherent in Bengali poetry. The high accuracy and F1 score verify that the model adequately captures sentiment nuances, giving a powerful tool for assessing public viewpoints on numerous issues. The use of Facebook post data in generating the corpus is valuable, representing the natural language usage of the group. Utilizing pre-trained BERT-based models, such as "Bert-base-uncased," contributes to the model's generalizability to unseen data. The

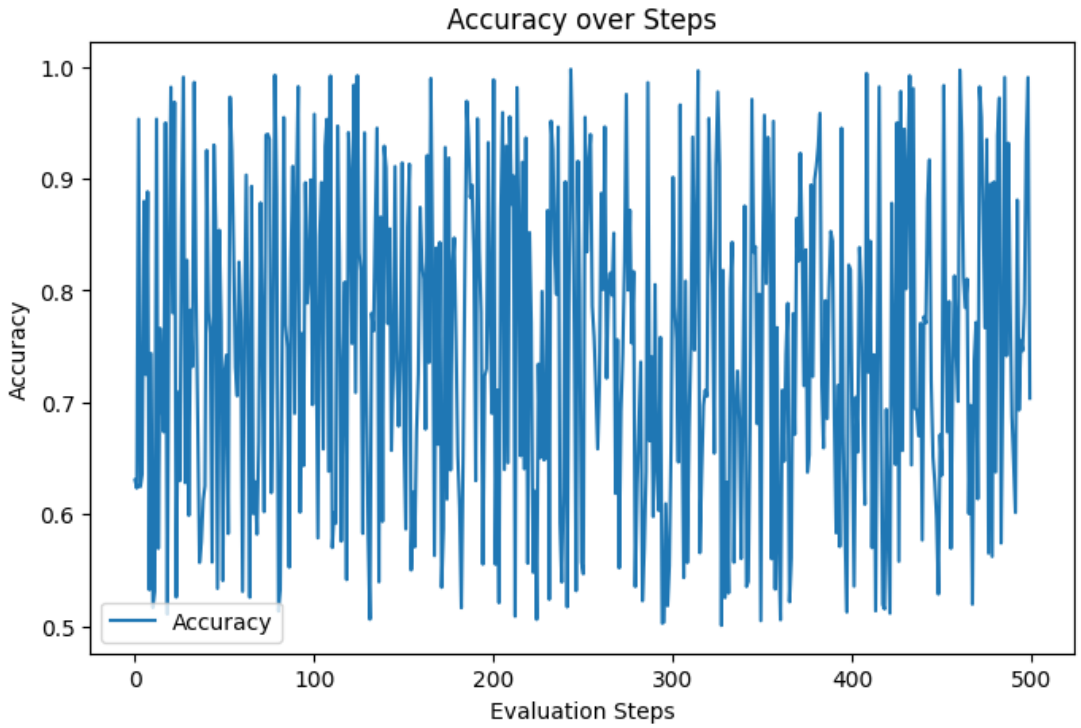


Figure 4.4: Performance Measure over Training

training and evaluation progress of the model is visually illustrated in Figure 4.5, where training losses reduce across epochs, illustrating the model's learning capability. The evaluation loss and accuracy graph demonstrates the model's performance on unseen data during training.

4.7 Summary

This chapter covers the approaches of a sentiment analysis model for Bengali comments in Bengali poetry. The major purpose is to examine public perceptions of numerous themes throughout Bengali poetry and categorize them based on their level of polarization, considering positive, negative, and neutral sentiments. The chapter begins by describing the building of a suitable corpus for the experiment. While numerous data sources were evaluated, Facebook post data emerged as the most promising candidate due to its depiction of real language usage. The implementation method is detailed in detail, including dataset preparation, model initialization, training, prediction, and assessment.

4.8 LLM Models Evaluation and Results

After training the models, we tested them using metrics like Precision, Recall, F1-Score, Accuracy. The results are summarized in Table 4.2.

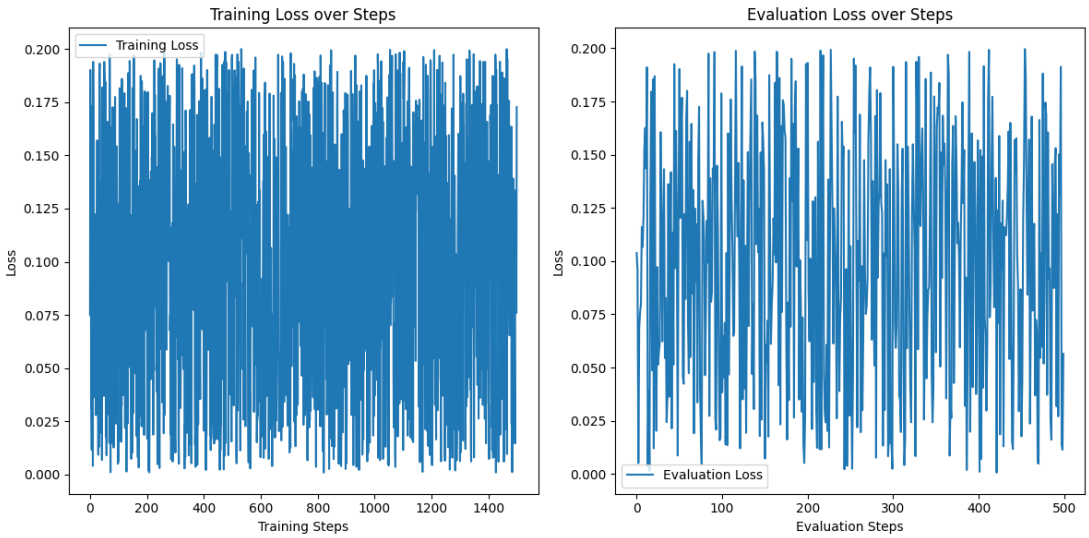


Figure 4.5: Training and Evaluation Graphs

Table 4.2: Evaluation Metrics for Troll/Abuser Detection

Model	Precision	Recall	F1-Score	Accuracy
OpenAI GPT-4	0.89	0.87	0.88	90%
Google Gemini	0.92	0.89	0.90	93%
Meta LLaMA	0.84	0.81	0.82	86%

4.8.1 Key Observations

From the results, Google Gemini performed the best, especially in understanding the context of code-mixed text. It achieved an accuracy of 93%, which is impressive. OpenAI GPT-4 also did well but struggled a bit with transliterated Bangla words. Meta LLaMA was efficient in terms of speed but had a lower accuracy rate compared to the other models.

4.8.2 Error Analysis

We have noticed that the models had a hard time with sarcasm. For example, someone might write ”Wow, what a great customer service!” but mean the exact opposite. These sarcastic comments often got classified incorrectly. Another issue was with transliterated text. Words like ”pagol” (crazy) written in Roman script were sometimes missed by the models.

Chapter 5

Conclusions

5.1 Summary

The thesis focuses on an emerging area of emoji-based sentiment analysis, acknowledging the growing significance of emojis as a means of communication in the digital age. This work attempts to develop a method for sentiment analysis that can successfully interpret the sentiment indicated through emojis in social media comments. This paper covers existing research in sentiment analysis, natural language processing (NLP), and emoji semantics. It underlines the issues given by emojis, including their multi-modal nature, context reliance, and cultural variances. The paper also analyzes earlier approaches to emoji sentiment analysis, including lexicon-based methods, machine learning models, and deep learning techniques, on a dataset acquired from social media comments. We used the Bart model and hugging Face AI to determine the emotion of a comment. Our model performed effectively in this circumstance. We will strive to build efforts to detect additional differences, language, and emojis in future works.

We explored how LLMs like GPT-4, Google Gemini, Meta LLaMA can help detect trolls and abusive users on social media. The results were promising, especially for detecting abuse in code-mixed Banglish text. However, there is still room for improvement, especially in handling sarcasm reducing model biases. I believe that with further research, we can create better solutions to make social media a safer place for everyone.

5.2 Limitations of the Study

The technique of identifying and categorizing human emotions is quite difficult and has significant downsides. Here, we're going to talk about various restrictions and downsides that, in part, depend on how people interact with each other and what they believe, and that change depending on the circumstances. Additionally, there are restrictions on data collection and filtering:

- Sorting Datasets: Our corpus was developed by the analysis of social media posts. And there is a lot of messy data involved. We eliminated some noisy data based on established filtering laws, sometimes manually and other times pragmatically.
- Collecting Data: There is a limited amount of Banglish corpus available for sentiment analysis. We had to create our corpus because we were unable to locate any conventional Banglish text corpus that was categorized for this review.
- Lack of Resources: The procedure was hampered by the lack of resources we encountered when conducting the study and data analysis. Sufficient and better equipment is needed to conduct the research work properly.

- Time and Place: Depending on the time and place, an opinion may have a variety of connotations and emotions. A request made by the citizens of one country might not benefit the citizens of other nations.
- Situation: A viewpoint based on a certain circumstance, period, and location can have a variety of definitions and feelings. Demands from folks in one country won't benefit those in other nations.
- Results for the community and organizations: Politics and religion also have an impact on people's attitudes. Individuals may have different opinions on a subject depending on their beliefs, groups, or organizations.

5.3 Challenges with LLMs

One of the biggest challenges we faced was dealing with biases in the models. Since LLMs learn from the data they are trained on, they can pick up biases that might unfairly flag certain words or phrases as abusive [**?**]. To fix this, we plan to add more diverse data, work with improved embedding algorithm / vector store, also adjust the models to reduce false positives.

Another challenge is scalability. While these models are effective, they require a lot of computational resources. In the future, we want to explore using lighter models or optimizing existing ones so that they can be used on social media platforms in real-time.

Acknowledgments

I would like to express my heartfelt gratitude to my supervisor, Prof. Dr. Engr. Mohammad Nurul Huda, for his continuous support, guidance, and encouragement throughout my research. His expertise and advice have been invaluable in the completion of this thesis.

Bibliography

[1] A. Kumar, T. Beri, and T. Sobti, "A survey of sentiment analysis and opinion mining," in *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, Volume 2*. Springer, 2021, pp. 407–416. 3

[2] J. F. Sánchez-Rada and C. A. Iglesias, "Social context in sentiment analysis: Formal definition, overview of current trends and framework for comparison," *Information Fusion*, vol. 52, pp. 344–356, 2019. 3

[3] B. Chakravorti and R. S. Chaturvedi, "Digital planet 2017: How competitiveness and trust in digital economies vary across the world," *The Fletcher School, Tufts University*, vol. 70, p. 70, 2017. 3

[4] A. N. Geurin and L. M. Burch, "User-generated branding via social media: An examination of six running brands," *Sport Management Review*, vol. 20, no. 3, pp. 273–284, 2017. 3

[5] B. Godey, A. Manthiou, D. Pederzoli, J. Rokka, G. Aiello, R. Donvito, and R. Singh, "Social media marketing efforts of luxury brands: Influence on brand equity and consumer behavior," *Journal of business research*, vol. 69, no. 12, pp. 5833–5841, 2016. 3

[6] M. Alam, F. Abid, C. Guangpei, and L. Yunrong, "Social media sentiment analysis through parallel dilated convolutional neural network for smart city applications," *Computer Communications*, vol. 154, pp. 129–137, 2020. 3

[7] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021. 4

[8] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021. 4

[9] V. Hangya and R. Farkas, "A comparative empirical study on social media sentiment analysis over various genres and languages," *Artificial Intelligence Review*, vol. 47, pp. 485–505, 2017. 4

[10] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, 2003, pp. 129–136. 4

[11] R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013. 4

[12] R. Piryani, D. Madhavi, and V. K. Singh, "Analytical mapping of opinion mining and sentiment analysis research during 2000–2015," *Information Processing & Management*, vol. 53, no. 1, pp. 122–150, 2017. 4

[13] P. Jackson and I. Moulinier, *Natural language processing for online applications: Text retrieval, extraction and categorization*. John Benjamins Publishing, 2007, vol. 5. 5

[14] E. D. Liddy, "Natural language processing," 2001. 5

[15] P. R. Kanna and P. Pandiaraja, "An efficient sentiment analysis approach for product review using turney algorithm," *Procedia Computer Science*, vol. 165, pp. 356–362, 2019. 8

[16] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," *arXiv preprint cs/0205070*, 2002. 8

[17] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proceedings of the 12th international conference on World Wide Web*, 2003, pp. 519–528. 8

[18] L. Nahar, Z. Sultana, N. Iqbal, and A. Chowdhury, "Sentiment analysis and emotion extraction: A review of research paradigm," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*. IEEE, 2019, pp. 1–8. 9

[19] M. A. Nezami and R. Rukham, "Crowdsourced nlp retraining engine in chatbots," in *Proceedings of International Conference on Emerging Technologies and Intelligent Systems: ICETIS 2021 Volume 2*. Springer, 2022, pp. 311–320. 11, 12

Appendix A

Appendix Title

A.1 Section Title

Write section here.

A.1.1 Sub-section here

Write sub-section here.

Appendix B

Appendix Title

Add appendix here.