Understanding Public Sentiment on Social Media Platforms of Bangladesh: A Machine Learning Based Approach

by

Md. Monir Uddin Ananda – 180021105

Md. Asef Zahin – 180021131

Wasay Mahmood Qureshi – 180021138

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering

Islamic University of Technology (IUT)

The Organization of Islamic Cooperation (OIC)
Board Bazar, Gazipur-1704, Bangladesh

June 2023

CERTIFICATE OF APPROVAL

The thesis titled "Understanding Public Sentiment on Social Media Platforms of Bangladesh: A Machine Learning Based Approach" submitted by Md. Monir Uddin Ananda (180021105), Md. Asef Zahin (180021131), and Wasay Mahmood Qureshi (180021138) has been found as satisfactory and accepted as partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on 5th June, 2023.

Approved by:

(Signature of the Supervisor)

Mirza Muntasir Nishat
Assistant Professor
Department of Electrical and Electronic Engineering
Islamic University of Technology

DECLARATION OF CANDIDATES

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for award of any degree or diploma.

(Signature of the Candidate)
Md. Monir Uddin Ananda
Student ID: 180021105

(Signature of the Candidate)
Md. Asef Zahin
Student ID: 180021131

(Signature of the Candidate)
Wasay Mahmood Qureshi
Student ID: 180021138

DEDICATION

We would like to dedicate this thesis to our family members and everyone who have given us unwearied support throughout the entirety of our existence and every situation of our life. They have always been a source of motivation for us. They pushed us ahead and showed us how to make the correct decisions. They never fail to inspire us to work hard and move forward to overcome life's difficulties. They have provided us with the protection, wisdom, and fortitude we need to face difficult situations.

ACKNOWLEDGEMENTS

First, we would want to express our heartfelt gratitude to Almighty Allah, our Creator, for creating and instilling in us the intellect to educate ourselves with worldly knowledge and, therefore, complete our thesis research. ***Mr. Mirza Muntasir Nishat***, Assistant Professor, Department of EEE, IUT, is our respected supervisor. We owe him a debt of gratitude for his continuous advice, care, and support in our pursuit of a career in electrical and electronic engineering. He has always encouraged us to learn new things and broaden our horizons to keep our minds sharp. We would not be exploring the power electronics area if it were not for his motivation. He has encouraged us to learn the fundamentals of the specific field and shown us how to proceed in the right direction

Prof. Dr. Md. Ashraful Hoque, Professor, Department of EEE, IUT, is our co-supervisor, and we are grateful for his constant mentoring and genuine efforts in our thesis research. He devoted his considerable time guiding and motivating us to finish the work. Moreover, he gave us the most efficient technique to better understand our research. Without his help, we would become lost and unable to pick the best course of action.

Mr. Fahim Faisal, Assistant Professor, Department of EEE, IUT, served as our co-supervisor and mentored us throughout the research process. He has motivated us to study the primary goal of our project, which has given us greater confidence in our ability to build skills while working on the thesis.

We would also like to express our gratitude to ***Prof. Dr. Mohammad Rakibul Islam***, Head, Department of EEE, and all the faculty members of the EEE Department, IUT, for their unwavering support, encouragement, and assistance.

Finally, we owe a debt of gratitude to our family for encouraging and assisting us in overcoming life's challenges, as well as enchanting us with their wonderful words. Last but not least, we would like to express our gratitude to our friends for their unconditional support and for keeping our spirits upbeat throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

Abbreviated Form	Description
NLP	Natural Language Processing
SN	Social Network
SA	Sentiment Analysis
ML	Machine Learning
DL	Deep Learning
CPU	Central Processing Unit
GPU	Graphics Processing Unit
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
BiGRU	Bidirectional Gated Recurrent Unit
NLTK	Natural Language Toolkit
BNLP	Bengali Natural Language Processing
SVC	Support Vector Classifier
SVM	Support Vector Machines
MNB	Multinomial Naive Bayes
LR	Logistic Regression
DT	Decision Tree
RF	Random Forest
TF-IDF	Term Frequency-Inverse Document Frequency
CV	Cross-Validation
LDA	Latent Dirichlet Allocation
RNN	Recurrent Neural Network
RLU	Rectified Linear Unit
GRU	Gated Recurrent Unit
TP	True Positive
FP	False Positive

ABSTRACT

Sentiment analysis, also known as opinion mining, holds significant importance in today's digital age where vast amounts of textual data are generated daily. Understanding the sentiments expressed in text can provide valuable insights into public opinion, customer satisfaction, market trends, and social dynamics. While sentiment analysis has been extensively studied for major languages like English, there is a growing need for similar research in languages like Bangla. Bangla being the seventh most spoken language in the world, with a large number of users using social media to express their opinions and sentiments in this language. Analyzing sentiments in Bangla text allows for a deeper understanding of the sentiment landscape within the Bangla-speaking community. Bangla language has its own unique cultural nuances, expressions, and sentiments that may not be captured accurately by models trained on other languages. Developing sentiment analysis models specifically for Bangla ensures cultural relevance and accurate interpretation of sentiments. Also little to none work has been done to successfully train any model on a Bangla dataset having more than 6 classes. Only Positive, Negative and Neutral classifications can't portrait the exact sentiment of a sentence. So, we have built a custom Bangla Dataset which is classified into 6 classes of emotions: Joy, Sadness, Anger, Fear, Surprise and Disgust. Then we have trained different Neural Network and Machine Learning based models like: LSTM, CNN, BiLSTM, BiGRU and an Ensemble Learning Model on our dataset. Later, a comparative analysis based on the performance of the models is shown. The findings of this research contribute to sentiment analysis in Bangla language, shedding light on effective deep learning architectures for accurate emotion classification. The developed models can be utilized in various applications, such as social media sentiment analysis, customer feedback analysis, and opinion mining. The research also highlights the importance of considering linguistic and cultural nuances when designing sentiment analysis systems for specific languages.

CHAPTER 1

INTRODUCTION

We used Natural Language Processing (NLP) techniques to conduct sentiment analysis on Bengali comments collected from Facebook and YouTube for this work. Our purpose was to assess the sentiment represented in these comments in order to obtain insight into the general public's feelings about various subjects[1]. Using natural language processing, we were able to categorize the comments as joy, sadness, anger, fear, disgust, surprise, and giving vital information for understanding the perspectives and feelings of Bengali-speaking users on various social media sites.

1.1 Introduction

The development of communication technology and the availability of high-speed internet have made it possible for many people from various backgrounds to participate in Social Networks (SN) and share their thoughts and opinions on a wide range of topics. We may get a sense of how quickly individuals are utilizing the internet and other social platforms if we look at the Bangladeshi internet user scenario during the previous four years. At the beginning of 2023, when internet access was 38.9%, there were 66.94 million users in Bangladesh. In January 2023, there were 44.70 million social media users in Bangladesh, or 26.0 percent of the country's population. According to a Kepios investigation, between 2022 and 2023, the number of internet users in Bangladesh rose by 691 thousand (+1.0 percent)[2]. At the start of 2022, Bangladesh's internet access rate was 31.5 percent of the total population. According to Kepios, internet users in Bangladesh rose by 5.5 million (+11.6 percent) between 2021 and 2022. In January 2021, there were 47.6 million (28.8%) internet users in Bangladesh, while 45.0 million (27.2%) actively used social media. The percentage of internet users and social media users that are active has increased over the past year to 19.2% and 25%, respectively[2]. The number of Facebook users that use Bangla to comment, publish, or share social media is almost 41 million, and it is constantly increasing. By categorizing the feelings of SN users into more than three classes utilizing Natural

Language Processing (NLP), Machine Learning, and Multi-class SA, we are able to do this. The majority of research on multi-class SA has focused on predicting positive, negative, or neutral attitudes using ML or DL algorithms. However, deciding whether a post is positive, negative, or neutral may not accurately reflect the true sentiment it conveys[3]. To understand the underlying meaning of social media comments, it is important to automatically categorize thoughts into more than three distinct groups. Though the classification problem becomes more sophisticated as the number of sentiment classes increases, it becomes difficult to maintain a high level of accuracy when using ML algorithms. It requires a significant amount of computing effort, money, and human experience to manually develop a high-performing ML model.

To avoid over-fitting on a narrow range of viewpoints, one of the most challenging aspects of the work was locating high-quality datasets with a decent chance of capturing different perspectives on any given issue. Another crucial element that needed to be done correctly was cleaning and annotating the dataset, as a clean and well-annotated dataset produces high-quality training results for the model[4]. Finally, there was a problem with training time due to the large dataset and the sophisticated model's potential for high computational demands. Therefore, we were forced to use vertical scaling of our hardware, which required costly CPU and GPU capabilities.

TABLE 1.1. SOCIAL MEDIA USERS IN BANGLADESH

Social Media	**No. Of Users (In Millions)**
Facebook	43.25
YouTube	34.40
Facebook Messenger	20.35
Linkedin	5.90
Instagram	4.45
Twitter	1.05

According to the above table, Facebook and YouTube have the largest social media usage in Bangladesh[5]. As a result, we picked these two channels to collect feedback. Scraping comments from Facebook postings and YouTube video in Bangladesh might provide important information on the perspectives, attitudes, and behaviors of the country's social media users. We can learn a lot about popular themes, people's attitudes about certain issues, and how they interact with material on these platforms by studying text data. It's crucial to remember, though, that scraping comments might be difficult at times[6]. For instance, the comments may be noisy due to spam, useless remarks, or remarks in several languages. Additionally, the comments may not reflect the opinions of the general public or are biased against a particular user group or society. Therefore, to ensure accurate and pertinent results from the analysis, it is important to carefully clean and preprocess the data.

1.2 Problem Statement

1. Given a set of Bangla datasets that are often intermingled with English languages, emojis, and punctuation marks that are unneeded.

2. Due to a lack of adequately labeled datasets and preprocessing techniques, the Bengali language has made little progress in this sector.

3. Estimate a person's general sentiment based on their social media comments and feedback.

4. Calculating Cohen's Kappa score[7].

5. Boasting more accuracy than other existing models.

1.3 Research Challenge

- Scraping a large number of relevant comments from YouTube and Facebook was time consuming because we wanted to develop our own dataset. We scraped comments from YouTube using the Google API V3, and we manually collected comments from Facebook.

- Second, the dataset we gathered was mixed with Bangla and English terms. For the sake of efficiency, we had to translate the English into Bangla, delete troublesome terms, and remove emojis and punctuation marks.
- The third issue we encountered in our implementation was developing a model that is not overfitting and can classify unique expressions outside of training and testing datasets. As we worked on overall sentiment analysis, a large dataset was trained to provide an accurate result for any type of comment or feedback.
- Training a large dataset with our CNN, LSTM, BiGRU, BiLSTM, Ensembled Learning model was time consuming.

1.4 Thesis Objective

- Constructing a thorough dataset in Bangla with precise labels that may predict sentiment in Bengali language.
- Providing better sentiment detection and performance compared to other existing models.

1.5 Key Contribution

- One of our major achievements is the development of a high-quality dataset that fairly represents the range of public opinion in Bangladesh on various topics. Careful filtering and annotation of the dataset guarantee its high quality and applicability to the study's intended participants.
- In addition, by combining six separate sentiment categories—joy, anger, fear, sadness, disgust, and surprise—our work gives a greater comprehension of the feelings portrayed by people writing in the Bangla language. This more inclusive categorization paves the way for a more complex examination of sentiment, one that can capture a wider spectrum of feelings than just "positive" or "negative." This enhancement improves the reliability and depth of sentiment

analysis[8]; this elaboration of sentiment classifications allows for a more sophisticated comprehension of the broad spectrum of human emotions, allowing for a more nuanced study of public opinion.

- The assessment and implementation of numerous models for sentiment analysis is another major addition to our work. We have investigated many architectures and methods to find the best method for Bangla sentiment analysis by using CNN, LSTM, BiLSTM, BiGRU, and Ensembled models. A comprehensive analysis of model performance is demonstrated, and the benefits and drawbacks of each method are taken into consideration.
- Our efforts contribute to improving the state of the art in sentiment analysis for the Bangla language. Most methods and models for sentiment analysis have so far been created for and tested largely on the English language. By narrowing our attention to Bangla, we've addressed the unique difficulties and complexities of sentiment analysis in this language, meeting the demands of the local communities and laying the groundwork for future studies and applications of sentiment analysis in Bangla.

1.6 Thesis Organization

The following arrangement comprises the remaining sections of the thesis: The second chapter provides Background for the whole piece of work. The Literature Review, Dataset, Methodologies, and Results are presented sequentially in chapters 3, 4, 5, and 6. The conclusion of this thesis comes in Chapter 7.

CHAPTER 2

BACKGROUND

2.1 What is NLP?

NLP stands for Natural Language Processing. It is a branch of research that integrates artificial intelligence, computational linguistics, and computer science to enable computers to comprehend, interpret, and produce human language. NLP entails creating algorithms and models that allow computers to interpret and analyze text or speech input, extract meaning, and reply in a meaningful and human-like manner. NLP applications include language translation, sentiment analysis, speech recognition, chatbots, and information retrieval, among others[9].

2.2 Factors that influence NLP

Data Quality and Quantity: The quality and amount of the training data that are available have a significant impact on NLP performance. The accuracy and generalization abilities of NLP models may be improved with enough and a variety of data.

Domain-specific knowledge: Domain-specific knowledge and language models developed on specialized datasets can improve NLP performance. Techniques for domain adaptation can help people comprehend certain subjects or sectors better.

Ambiguity and context: Due to the numerous possible interpretations of words and phrases in natural language, ambiguity is a difficult concept to grasp. For clear interpretation and disambiguation, it is essential to comprehend the context.

Language Variety: Grammar rules, word order, and idiomatic phrases all have differing degrees of complexity in various languages. To accomplish effective understanding and generation, NLP systems must take these language-specific characteristics into consideration.

Cultural and regional variations: Language use, including dialects, slang, and idioms, can differ among cultures and locations. To efficiently analyze and produce language across many groups, NLP models must be aware of these variances.

Ethics: NLP systems must take into account issues like prejudice, privacy, and fairness. To prevent prejudices and injustices from being furthered, it is crucial to fairly reflect many languages, cultures, and viewpoints.

Advances in Technology: Technological advancements have dramatically affected the capabilities and performance of NLP systems, enabling increasingly challenging language processing tasks. These breakthroughs include improvements in computational power, machine learning strategies, and deep learning architectures.

User feedback and engagement: By using methods like active learning, reinforcement learning, or user-driven customization, NLP systems can benefit from ongoing user involvement and input.

2.3 Applications of NLP

Sentiment Analysis: Natural language processing (NLP) techniques are used to determine the sentiment or opinion conveyed in text data such as social media postings, customer reviews, or survey. This aids in public opinion analysis, brand reputation management, and market research[9].

Text production: NLP approaches enable the production of human-like text, such as chatbot answers, automated content generation, or specific suggestions[10].

Text Classification and Categorization: NLP systems can automatically categorize text into predetermined groups or themes. This is utilized in spam detection, content filtering, news classification, and customer feedback analysis[11].

Text summarizing: NLP helps in producing succinct summaries for effective information intake. It approaches enable the automated summarizing of big text documents or articles[12].

Translation of Languages: NLP supports machine translation systems that automatically convert text or speech from one language to another. It has considerably increased translation accuracy and accessibility.

Voice Recognition: NLP is used in voice recognition systems to translate spoken language into written text. It drives voice assistants, voice-controlled devices, and speech-to-text transcription services.

Chatbots: NLP is used to allow chatbots to comprehend and reply to user questions and give customized help.

Entity Recognition: NLP approaches assist in recognizing and extracting particular information such as the names of persons, organizations, locations, or dates stated in text data. It is useful for information extraction, data mining, and knowledge graph development.

Language Modeling: NLP models may learn language patterns and structures in order to create coherent and contextually appropriate content. This is utilized in text completion, auto-correction, and predictive typing.

2.4 Sentiment Analysis

The practice of applying natural language processing (NLP) methods to ascertain the sentiment or emotional tone indicated in a piece of text, such as social media postings, customer reviews, or survey replies, is known as sentiment analysis, sometimes known as opinion mining. NLP makes it possible to comprehend the context, syntax, and semantics of the language, which is essential for correctly identifying sentiment. In order to extract sentiment-related information and categorize the text into ‘Happy’, ‘Sad’, ‘Fear’, ‘Joy’, ‘Angry’; NLP techniques such as text preparation, tokenization, part-of-speech tagging, and machine learning algorithms are utilized.

To comprehend popular opinion, client sentiment, and market trends, sentiment analysis is necessary. It assists companies and organizations in learning more about how consumers feel about their brands, goods, and services. Businesses may make data-driven choices, enhance their offers, and address consumer problems by studying the sentiment. Businesses may acquire insightful information from massive amounts of text data, including social media discussions and consumer feedback, by integrating NLP with sentiment analysis, and they can base their choices on the sentiment and opinion of their customers.

2.4.1 Why Sentiment Analysis

There are several benefits to using sentiment analysis in any situation. First, sentiment analysis helps us make sense of how people generally feel about things on social media. It allows users to evaluate and discover patterns in public opinion by sifting through the massive amounts of user-generated information available on social media sites like Facebook, Twitter, YouTube and Instagram. Decision-makers can learn about the audience's dominant opinions, preferences, and worries by analyzing sentiment patterns. Organizations may use this data to improve their interaction and communication with the general public by tailoring their messages, campaigns, and services to reflect the views and priorities of target audiences.

Second, it provides the opportunity to make good choices by gaining useful information via sentiment analysis. The public's perspective can be better understood by decision-makers if they pay attention to the tone of comments, criticism, and social media posts. A favorable evaluation implies contentment, whereas a negative evaluation calls attention to shortcomings. With this information, policymakers may make well-informed decisions and adopt policies that reflect public opinion, therefore raising the probability of attaining their goals.

Finally, insights into particular difficulties can be obtained through sentiment analysis. With the use of sentiment analysis, leaders may learn how the public feels about a variety of issues and topics[5]. Organizations may use the results of this study to determine how the public feels about a certain subject, what their biggest worries are, and how serious the consequences of any decisions or actions taken will be. With this information at their disposal, leaders can better adapt to changing circumstances, address public concerns, and encourage public participation[13].

In conclusion, insights gleaned from sentiment analysis can aid in making smart choice in a number of contexts. It may help learn more about the customers, handle crises and improve brand's reputation, and spot new market opportunities, consumer preferences, and holes in the market. Decision-makers can predict public opinion, address concerns, and make educated decisions with the use of sentiment analysis, which offers significant information about specific situations. Organizations may boost their decision-making, employee engagement, and audience interactions with the use of sentiment analysis. It's useful for evaluating the efficacy of reputation management initiatives.

2.5 Proposed Solution

We went through various procedures to create a solid dataset before conducting sentiment analysis for Bangla comments and feedback using NLP techniques. Initially, we gathered feedback from numerous sources, such as Facebook and YouTube. However, we experienced difficulties during the dataset building process, such as unsuitable and out-of-context remarks, as well as one-word comments that lacked adequate context for sentiment analysis. We came up with a complete approach that included data preprocessing and meticulous annotation to solve these difficulties. To begin, we created a set of criteria or standards for identifying and filtering out undesirable remarks. This procedure aided in the removal of useless, offensive, or spam items from our dataset. Furthermore, we opted to reject comments with less than three words and more than twenty words since they either lacked the requisite context or had several contexts in a single comment for reliable sentiment analysis. We assured that our dataset had significant and valuable data for analysis by defining a minimum and maximum length criterion for comments. To get the comments ready for sentiment analysis, we then did data preprocessing. As a result, the text was made simpler and simpler to deal with during analysis by deleting punctuation marks, emojis, word embedding[14] and disregarding numerical values. Various text preparation methods, including tokenization, stop word removal, and others, were also used. These actions assisted in standardizing the content and simplifying the language, improving its suitability for sentiment analysis.

The important processes of annotation and labeling came next after data preparation. We created a precise and thorough annotation methodology to assure the dataset's quality and dependability. Six different emotion categories—joy, fear, anger, sadness, disgust, and surprise—were identified based on the remarks. "Doesn't fall under any category" and "not sure" were used to describe comments that did not fit into any of the six categories. This guideline helped the team judge sentiment labels consistently by giving annotators advice on how to do so. Annotators did not discuss the remarks with one another in order to ensure impartiality[15].

In summary, we gathered datasets from social media users that depict real-world events and sentiments more accurately. Our collection of data from Facebook and YouTube comments includes data in both Bangla and English with a variety of regional dialects. Our suggested approach entailed preparing the data by deleting improper remarks, rejecting comments with less words (less than 3 words) and long comments (more than 20 words) and doing text preprocessing to normalize the content. Following that, we meticulously annotated the dataset using a detailed guideline to ensure correctness and consistency. This work proposes a methodology for estimating the emotion of the general public on any issue, as reflected in Bangla language. The suggested model is based on the CNN, LSTM, BiLSTM, BiGRU, and Ensembled models.

CHAPTER 3

LITERATURE REVIEW

The field of natural language processing (NLP) has made significant strides in recent years, particularly in the realm of sentiment analysis. This subfield of NLP focuses on identifying and extracting opinions, attitudes, and emotions from text data, and it has numerous applications in a variety of domains, including social media analysis, market research, and customer feedback analysis, to name a few. In this literature review section, we explore recent studies on sentiment analysis that apply NLP methods. Specifically, we delve into four research papers that were published in the past four years and that have demonstrated novel techniques and approaches to sentiment analysis in diverse languages and contexts.

- The first paper that we examine is titled "Multi-class sentiment classification on Bengali social media comments using machine learning," which was published in 2023[16]. This paper investigates the classification of sentiment in Bengali social media comments using a machine learning-based approach to achieve multi-class classification of positive, negative, and neutral sentiments.
- The second paper we explore is "Anti-Islamic Arabic Text Categorization using Text Mining and Sentiment Analysis Techniques," published in 2022[17]. This paper examines the use of text mining and sentiment analysis techniques for categorizing anti-Islamic Arabic texts.
- The third paper, "Residents' sentiments towards electricity price policy: Evidence from text mining in social media," published in 2021[18], investigates the sentiment of residents towards electricity price policies by utilising text mining techniques on social media data.
- Lastly, we discuss "Sentiment Analysis on Electricity Twitter Posts," published in 2020, which explores the use of sentiment analysis on Twitter data related to electricity usage and pricing[19].

By reviewing these studies, we aim to identify common approaches, tools, and techniques used for sentiment analysis and explore the challenges faced by researchers in achieving high accuracy.

The findings from this literature review will provide insights into the state-of-the-art sentiment analysis methods and their application in different domains, contributing to the advancement of this field. Additionally, the identified gaps in the literature will guide future research directions and facilitate the development of more accurate sentiment analysis models. This review will also be useful for practitioners who are interested in implementing sentiment analysis techniques in their respective industries.

3.1 Multi-class sentiment classification on Bengali social media comments using machine learning (2023)[16]

The paper titled "Multi-class sentiment classification on Bengali social media comments using machine learning" presents a comprehensive study on multi-class sentiment analysis (SA) of Bengali social media comments using various natural language processing (NLP) techniques, feature extraction methods, machine learning (ML) algorithms, and deep learning (DL) architectures. The authors aim to provide researchers with a comparative analysis of ML and DL-based classifiers on Bengali social media comments and propose a novel CNN-based LSTM network, CLSTM, to improve the classification performance. The paper also includes the development of a web application for real-world sentiment classification.

The paper starts with an introduction to the challenges of multi-class SA in Bengali due to the absence of ground-truth datasets, inadequate data-collecting methods, and insufficient preprocessing tools. The authors then present related works on multi-class SA in English, Chinese, Urdu, and Bengali languages, summarizing the dataset size, number of classes, algorithms used, and accuracy scores. The authors highlight the scarcity of research papers with high model performance in Bengali sentiment analysis and the need for a comprehensive study to improve classification accuracy.

The authors then present the methodology used in the study, which includes data collection from publicly available datasets, data cleaning and preprocessing techniques, ML and DL-based classifiers, and performance evaluation using standard evaluation metrics. The dataset comprises 42,036 Bengali social media comments divided into four categories: politically acceptable, sexual, religious, and socially acceptable. The authors used the NLTK and BNLP packages in Python for basic and advanced NLP techniques, respectively, to preprocess the comments by removing URLs, usernames, emojis, punctuation, and digits, stemming, stop-word removal, and part-of-speech tagging. The ML classifiers used were SVC, MNB, LR, DT, and RF, and the feature extraction techniques used were TF-IDF and CV. The DL classifiers used were LSTM, BiLSTM, and BiGRU, trained on word embedding.

The authors then propose a novel CNN-based LSTM network, CLSTM, by combining a convolutional neural network (CNN) and an LSTM layer to improve the classification performance. The authors compare the performance of ML and DL classifiers with the proposed CLSTM model using standard evaluation metrics such as accuracy, macro-precision, macro-recall, and macro-F1 score. The authors observed that the proposed CLSTM model outperformed all other classifiers in terms of accuracy, macro-precision, and macro-F1 score. Finally, the authors present a web application for real-world sentiment classification using the proposed CLSTM and the highest-performing baseline (LR) models. The authors provide a demo of the web application and conclude the paper by discussing possible future works.

The paper's contributions include the development of a multi-class dataset of Bengali social media comments, a comparative analysis of ML and DL classifiers, and the proposal of a novel CLSTM model for sentiment classification. The authors observed that the proposed CLSTM model outperformed all other classifiers in terms of accuracy, macro-precision, and macro-F1 score. The authors also noted that the proposed CLSTM model addressed the issue of overfitting observed in other DL models. The web application developed by the authors can be used for real-world sentiment classification, which has implications for businesses, governments, and social media platforms.

The paper's limitations include the absence of a comprehensive analysis of the impact of hyperparameters on model performance, the absence of an analysis of the impact of data imbalance on model performance, and the use of only two publicly available datasets for data collection. The authors noted that the proposed CLSTM model suffered from class overlap between the acceptable and sexual classes, which can be resolved by collecting additional comments on acceptable and sexual sentiments. The authors also noted that the proposed CLSTM model needs further testing on larger datasets.

In conclusion, the paper presents a comprehensive study on multi-class sentiment analysis of Bengali social media comments using various NLP techniques, feature extraction methods, ML and DL-based classifiers, and a novel CNN-based LSTM network. The authors provide a comparative analysis of ML and DL classifiers and propose a novel CLSTM model that outperforms all other classifiers. The paper's contributions have implications for businesses, governments, and social media platforms and highlight the need for further research on sentiment analysis in under-resourced languages.

3.2 Anti-Islamic Arabic Text Categorization using Text Mining and Sentiment Analysis Techniques (2022)[17]

The paper titled "Anti-Islamic Arabic Text Categorization using Text Mining and Sentiment Analysis Techniques" aims to detect and classify anti-Islamic content on the web using text mining and sentiment analysis techniques. The authors propose a framework consisting of four stages: data collection and annotation, data preparation and preprocessing, feature extraction, and classification. The authors collected data from various sources, including articles, journals, and personal blogs, using Yahoo and Google search engines. They focused on formal language used in academic writing and preprocessed the data by removing punctuation, whitespaces, and Arabic stop-words.

They used TF-IDF as a feature extraction technique and SVM and Naive Bayes as classifiers. The authors conducted experiments on two Arabic language datasets: a balanced dataset and an unbalanced dataset. They used word level and tri-gram level feature extraction techniques and compared the results of SVM and Naive Bayes classifiers. They found that the supervised machine learning approach using word level feature extraction achieved the highest accuracy of 97% on the balanced Arabic dataset using SVM algorithm with TF-IDF as feature extraction.

The results of the study suggest that the proposed framework can accurately detect and classify anti-Islamic content on the web. The authors highlight the importance of feature extraction techniques such as TF-IDF and the use of supervised machine learning algorithms such as SVM and Naive Bayes for text classification tasks. The authors also discuss the limitations of their study, including the absence of a dataset containing anti-Islamic content in Arabic and the lack of an efficient Arabic preprocessing library that supports tasks such as lemmatization. They also note that some webpages promoting hate or spreading false information about Islam were blocked in Saudi Arabia, making it difficult to collect data. The implications of the study are significant, as it has the potential to prevent the spread of inaccurate and harmful information about Islam and Muslims. The authors argue that their framework can be used to automatically detect the content of websites that are hostile to Islam and transmitting extremist ideas against it, thereby reducing the risk of attacks against Muslims.

In conclusion, the paper presents a comprehensive framework for detecting and classifying anti-Islamic content on the web using text mining and sentiment analysis techniques. The authors conducted experiments on two Arabic language datasets and found that the proposed framework achieved high accuracy in detecting anti-Islamic content. The study has important implications for preventing the spread of harmful information about Islam and Muslims, and the authors suggest that future research could focus on improving Arabic language preprocessing techniques and collecting more comprehensive datasets containing anti-Islamic content.

3.3 Residents' sentiments towards electricity price policy: Evidence from text mining in social media (2021)[18]

The paper explores residents' sentiments towards electricity price policy in China using text mining techniques on social media data. The authors argue that social media data can provide real-time and cost-effective data on residents' sentiments towards policy, which can be used to evaluate and improve policy effectiveness. The paper also aims to identify the time-varying and seasonal characteristics of residents' sentiments towards electricity price policy and analyze the influencing factors of negative and positive sentiments.

Previous research has explored various aspects of electricity price policy, including its impact on residential electricity demand (Ye et al., 2016), public acceptance of tiered pricing reform (Wang et al., 2012), and the impact of tiered pricing systems on urban residential electricity consumption (Zhang and Lin, 2018). However, these studies have relied on surveys or statistical analysis of consumption data, which may not capture the real-time and nuanced sentiments of residents towards policy. The paper employs text mining techniques and sentiment analysis to analyze residents' sentiments towards electricity price policy using social media data. By constructing dictionaries and removing stop words, the authors are able to identify the sentiment of each microblog post as positive, negative, or neutral. The authors also use the Latent Dirichlet Allocation (LDA) topic model to identify the topics driving negative and positive sentiments.

The authors find that residents generally show positive sentiments towards electricity price policy, with the intensity of sentiments characterized by three stages. The authors also identify seasonal differences in residents' sentiments towards policy, with sentiments being relatively negative in summer and autumn. The LDA topic model reveals that the value of fairness and the perception of smart meters are driving factors of positive and negative sentiments, respectively. Findings of this study have practical implications for the evaluation and improvement of electricity price policy. The authors recommend that policy-makers consider seasonal characteristics and regional resources in formulating electricity price policy. They also recommend highlighting the value of fairness and smart meters in the electricity market and promoting residents' participation in policy-making.

Overall, main takeaway of this paper is to highlight the potential of social media data and text mining techniques for policy evaluation and provides insights into residents' sentiments towards electricity price policy in China. The paper's findings can inform the development of more effective and responsive electricity price policies.

3.4 Sentiment Analysis on Electricity Twitter Posts (2020)[19]

Sentiment analysis has become a popular research topic in recent years due to the increasing use of social media platforms as a means of communication. The ability to analyze public opinions and sentiments expressed on social media can provide valuable insights into various domains such as politics, business, and healthcare. In this context, this research paper titled "Sentiment Analysis on Electricity Twitter Posts" discusses the sentiment analysis of public opinions towards hike in electricity prices, using tweet data obtained from social media Twitter. The paper discusses the importance of sentiment analysis in understanding public opinions and explores various machine learning models to predict sentiment polarity (positive, negative, or neutral) of tweets related to electricity price hikes.

The authors of the paper collected a dataset composed of nearly 10,000 tweets related to energy prices from two different countries with various incomes, one developing country and one developed country, using Twitter API for academic research and the Python programming language. They used two lexicon-based sentiment analysis approaches, TextBlob and Vader, to calculate sentiment scores and infer a sentiment polarity to each tweet. They also performed data processing, which involved tokenization and stop word removal, to make the data more machine-readable. The authors used the bag-of-words model and TF-IDF (Term Frequency - Inverse Document Frequency) for feature extraction. The paper then applied and compared different machine and deep learning algorithms on the collected and curated dataset, including Naive Bayes, Decision Tree, Logistic Binary, and Random Forest.

The results of the paper indicate that the Random Forest model has a better accuracy level (i.e., 0.84) compared to using other methods, such as Naive Bayes, which only has an accuracy rate of 0.66. Other models, such as Decision tree and Logistic Regression, have good accuracy levels to determine the negative sentiment and positive sentiment achieving an accuracy of 0.83 and 0.80, respectively. The authors attribute the higher accuracy of the Random Forest model to its ability to handle non-linear relationships and interactions between features.

The study highlights the importance of sentiment analysis in understanding public opinions towards electricity price hikes. The insights gained from sentiment analysis can be used by policymakers, energy companies, and other stakeholders to make informed decisions. The paper emphasizes the potential of social media platforms like Twitter as a source of data for sentiment analysis.

In contrast to previous studies, the paper collected data from two different countries with various incomes, one developing country and one developed country, and used two lexicon-based sentiment analysis approaches, TextBlob and Vader, to calculate sentiment scores. The authors used machine learning models, including Naive Bayes, Decision Tree, Logistic Binary, and Random Forest, for sentiment analysis and compared the performance of these models. The study found that the Random Forest model achieved the highest accuracy level in predicting sentiment polarity.

The study highlights the potential of social media platforms like Twitter as a source of data for sentiment analysis. The insights gained from sentiment analysis can be used by policymakers, energy companies, and other stakeholders to make informed decisions. The study emphasizes the importance of sentiment analysis in understanding public opinions towards electricity price hikes. There are several limitations of the research that can be acknowledged, such as the small number of available tweets and the use of only bag-of-words for feature extraction. The authors suggest that future studies can expand the search to other regions and extract tweets in other languages for sentiment analysis. They also suggest using advanced embedding models and considering the semantics for improving the results. The objectives of the study are to analyze sentiment by incorporating more complex models and to include data from other social media platforms such as Facebook and Instagram.

In conclusion, sentiment analysis can provide valuable insights into public opinions towards electricity price hikes. Machine learning models such as Random Forest can be used to predict sentiment polarity with high accuracy. The paper emphasizes the potential of social media platforms like Twitter as a source of data for sentiment analysis and highlights the importance of sentiment analysis in making informed decisions. Future studies can further improve upon the accuracy of sentiment analysis by using advanced embedding models and incorporating data from other social media platforms.

CHAPTER 4

DATASET

Bangla Sentiment Analysis is a difficult undertaking because there aren't many comprehensive annotated datasets. The development of exact sentiment analysis systems requires large datasets of annotated text. There are various locations and countries where Bangla is spoken, and these dialects frequently have different vocabulary, grammar, and syntax. One of the main causes of the dearth of annotated datasets for Bangla is the absence of language standards. We mainly concentrated on creating our own dataset because a solid dataset serves as the fundamental building element for any language model. We used datasets from social media users instead of Bangla Datasets from publicly available sources since they more properly reflect events and opinions in the real world. This dataset was appropriate for our model because it includes accurate Bengali, which is more commonly used than the usual running language. We scrapped comments from some famous Bangla news channels on YouTube. We also manually collected comments from Facebook and different forums. Ultimately, a dataset of almost twenty thousand comments was built which was later processed and annotated in six categories: anger, joy, sadness, fear, surprise and disgust.

4.1 Dataset Preprocessing

After collecting all the comments in a single file, we started processing the comments in the following process:

- Google's language detection library 'Langdetect' was used to identify only the Bangla comments from the huge collection of Bangla, English and mixture of Bangla-English comments[20]. Only the pure Bangla comments were kept for the next step.
- Then we removed unnecessary punctuations and emojis from the texts.

- We counted the length of each comment. The average number of words in the comments was 15 with a standard deviation of 22. So, Comments consisting less than 3 words and more than 20 words were deleted.
- Then we removed the duplicate comments

Following preprocessing, the dataset contained 5289 comments, each of which was written in Bengali without the use of emoji or unnecessary punctuation and had three to twenty words.

4.2 Dataset Labelling

We chose to classify our dataset into 6 classes know as Ekman's 6 basic emotions[8]: joy, sadness, anger, surprise, fear and disgust by two annotators with their expertise as:

- Native Bangla speaker
- NLP Researcher

It is crucial to have multiple people annotate a same language dataset for sentiment analysis since this helps to ensure the precision and dependability of the annotations. Based on the sentiment expressed in the text, sentiment analysis entails categorizing text into the proposed categories. The sentiment, however, can be arbitrary and change depending on how the annotator reads the text. It is easier to detect and reduce the influence of individual biases and subjectivity on the annotation process when multiple annotators annotate the same dataset. We can spot anomalies and discrepancies in the sentiment labeling by comparing the annotations of various annotators, and this information can be used to improve the annotation rules and the precision of the sentiment analysis as a whole.

4.3 Proposed Dataset

Our final proposed dataset has 5289 different texts all of which were labelled as one of: anger, disgust, joy, sadness, fear and surprise. A summary of our dataset:

TABLE 4.1. SUMMARY OF DATASET

Text	Sentiment
আপনার ভাল কাজ অব্যাহত থাক এটাই আমাদের চাওয়া	joy
বিদ্যুত এর চাহিদা দিতে পারে না আলাদিনের গল্পের মতো দেশ উন্নয়ন চলছে	anger
আজকে আমার মন ভালা নাই	sadness
সম্ভবত খুবই দুঃখজনক ও ভয়ংকর পরিস্থিতির সম্মুখীন হতে হবে	fear
এই এ অনেক সুন্দর সুন্দর আছে দেখে আসেন ভালো লাগবে	joy

The most common sentiment seen throughout the dataset is ‘anger’ and the least common is ‘surprise’. Total count of each sentiment:

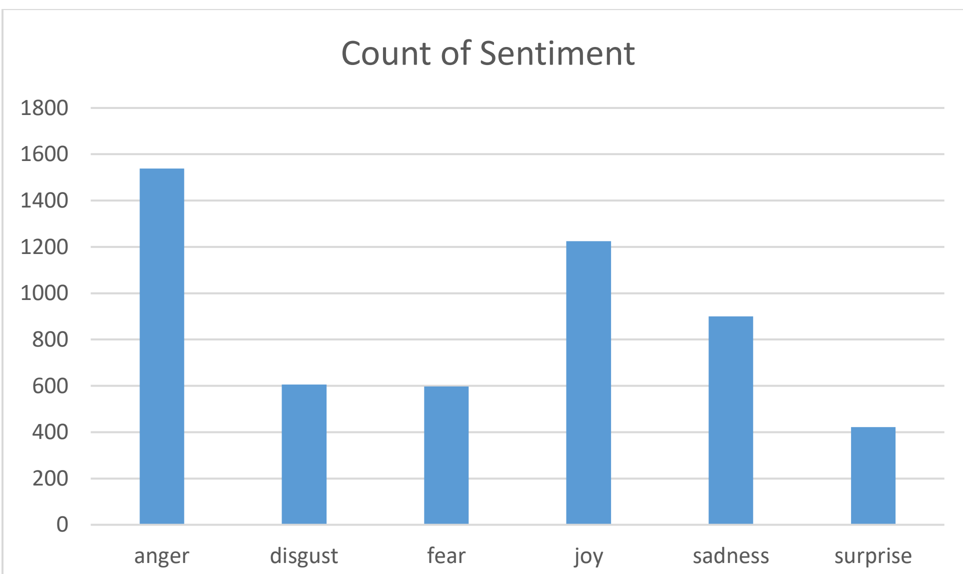


Fig. 4.1. Count of Sentiment

For model training, we counted the total number of words and also the total number unique words in the dataset under each sentiment. The unique word count gives us an idea about the dependency of sentiments on specific words.

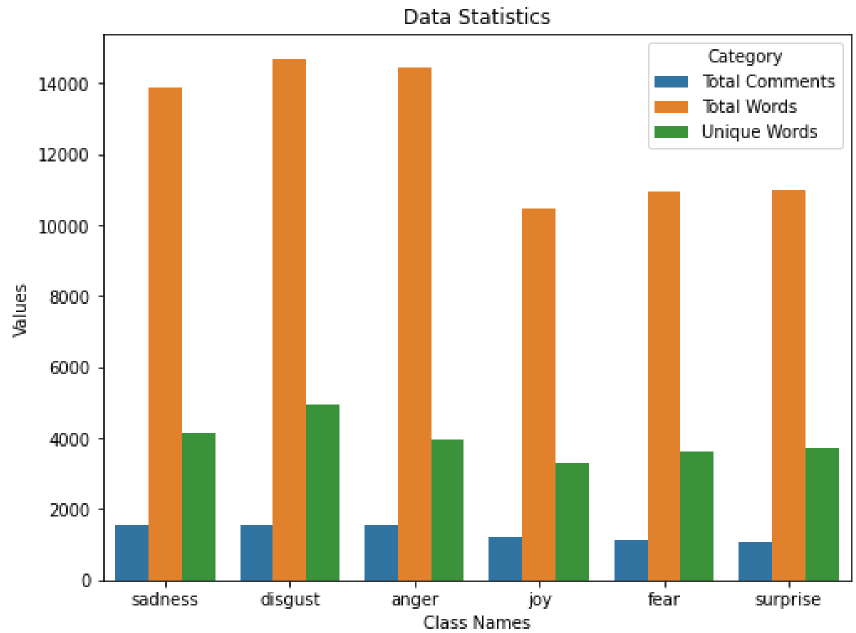


Fig. 4.2. Word Count

The dataset is built keeping the word limit of 20 per sentence and the shortest sentence has 3 words. On average, each sentence has 5-10 words.

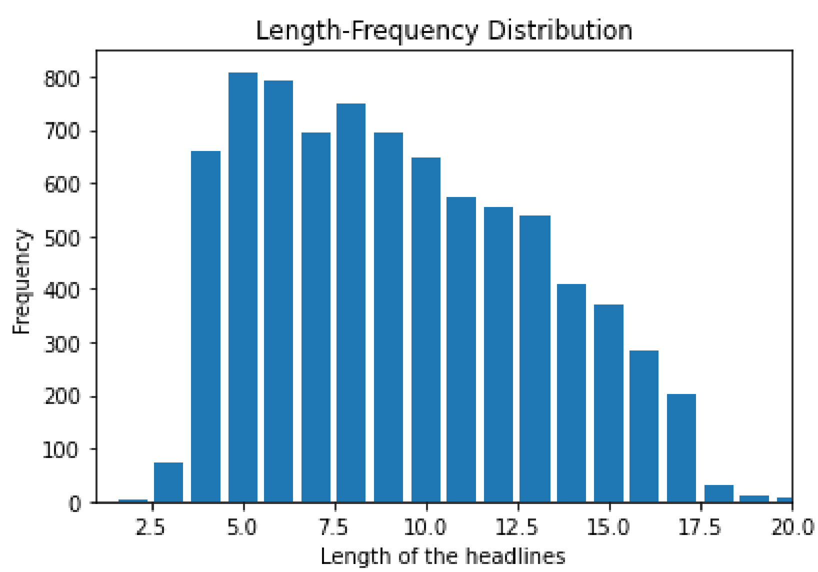


Fig. 4.3. Length Frequency Distribution

4.4 Inter-Annotator Agreement Calculation

Inter-annotator agreement, which is a metric reflecting the degree of agreement between the various annotators, can be raised by annotating a dataset by a number of different annotators. An annotation's consistency and reliability are indicated by its high inter-annotator agreement, whilst its subjectivity and unreliability are indicated by its low inter-annotator agreement.[7] One such statistical measure that is commonly used to assess the inter-annotator agreement between two or more annotators is known as Cohen's Kappa. It is a measurement of the degree of agreement among the annotators that accounts for the possibility of accidental agreement. Cohen's kappa scores vary from -1 to 1, with 1 denoting perfect agreement, 0 denoting agreement that would be expected by chance, and -1 denoting perfect disagreement. The formula for calculating Cohen's kappa is as follows:

$$\boldsymbol{\kappa = (Po - Pe) / (1 - Pe)}$$

The Cohen's Kappa value can be interpreted as follows:

TABLE 4.2. COHEN'S KAPPA VALUE INTERPRETATION:

Value Range	Cohen's Interpretation
Below 0.20	None to slight agreement
0.21-0.39	Fair agreement
0.40-0.59	Moderate agreement
0.60-0.79	Substantial agreement
0.80-0.90	Almost perfect agreement
Above 0.90	Almost perfect agreement

We calculated our Cohen's Kappa value after both the annotators labelled each comment and got a Cohen's Kappa value of 0.33 which depicts a fare rate of agreement among the annotators.

Chapter 5

Methodology and Experiment

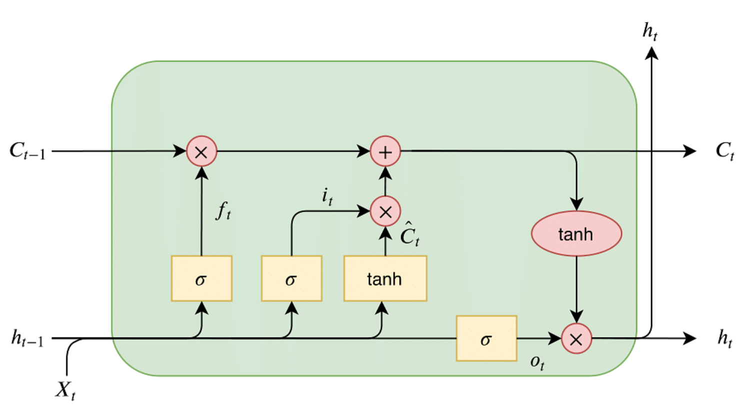
To assess the precision and efficiency of several machine learning and deep learning methods for sentiment analysis, we developed different classifier models such as LSTM, CNN, BiLSTM BiGRU and an ensembled learning method combining several models. The overall process and methodology is described below for each model:

5.1 Foundational Methodology

The basic working principle of each model used is described below.

5.1.1 LSTM

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) architecture that is commonly used for sentiment analysis tasks. The modeling of sequences with different lengths is one of the major difficulties in NLP. This problem can be solved by LSTMs because they provide variable-length input and output sequences[21]. LSTM control the information flow in a data sequence using a set of gates.



X_t: input time step

h_t: output

C_t: cell state

f_t: forget gate

i_t: input gate

O_t: output gate

Fig. 5.1. Internal Structure of an LSTM Unit

Within the LSTM network, the forget, input, and output gates act as filters and operate as independent neural networks. They control the flow of data as it enters the network, is stored, and then is eventually released.

Forget Gate: An LSTM's forget gate chooses which data from the previous cell state needs to be ignored. It generates a forget value by taking into consideration the current input, the previous concealed state, and biases.

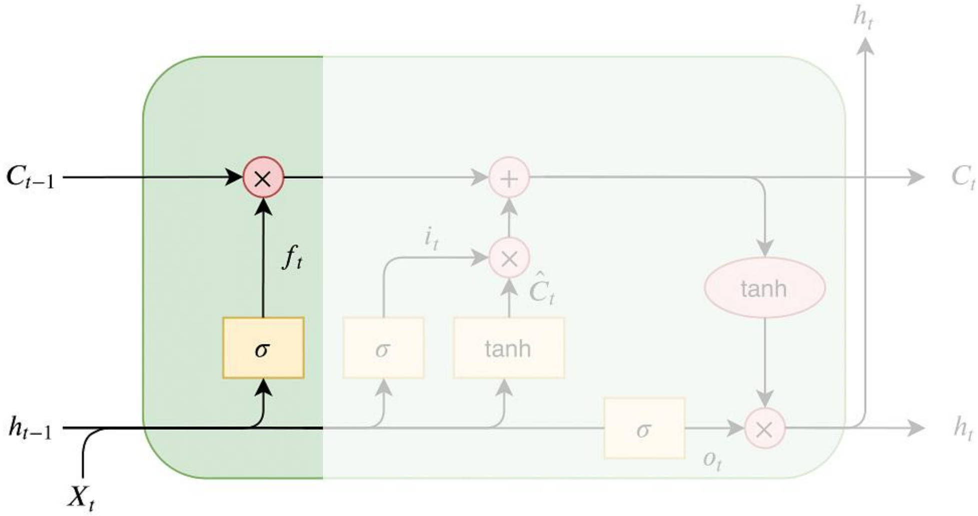


Fig. 5.2. Forget Gate

The prior hidden state (ht-1) and the new input data (Xt) are fed into a neural network that uses a sigmoid activation function to produce a vector with each element having a value between 0 and 1. This forget gate network is taught to provide a value close to 0 for information that is thought to be irrelevant and close to 1 for information that is seen to be useful. The components of this vector can be viewed as filters that open up more data as the value approaches 1[22].

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, X_t] + b_f\right)$$

The preceding cell state (Ct-1) is then multiplied element-wise by these output values. As a result, the components of the cell state that aren't important are down-weighted by a factor almost equal to 0, which lessens their influence on subsequent actions.

The forget gate, given the previous hidden state and the new input data in the sequence, basically decides which portions of the long-term memory should be forgotten. Let's elaborate the forget gate principle with an example from our dataset.

TABLE 5.1. EXAMPLE FOR LSTM

Text	Sentiment
সেকেন্ড পরপর আপনার হাতের ভঙ্গিমাটা দেখে আসলেই বিরক্তি লাগছে	anger
স্যারদ্বয় সমীপে আপনারা আমাদের আশার বাতিঘর	joy

Here the text "স্যারদ্বয় সমীপে আপনারা আমাদের আশার বাতিঘর" (Sir, you are our beacon of hope) indicates that the writer is feeling joy. But it's previous text "সেকেন্ড পরপর আপনার হাতের ভঙ্গিমাটা দেখে আসলেই বিরক্তি লাগছে" (It's really annoying to see your hand gesture every second) depicts anger. Now when the LSTM gate encounters the word "বিরক্তি লাগছে" it stores the anger sentiment in its short-term memory. When processing the 2nd text, the forget gate would determine how much of the anger sentiment from the previous cell state should be discarded[21]. A low value from the forget gate would indicate that the anger sentiment is irrelevant and should be forgotten.

Input Gate: How much new information should be stored in the cell state of an LSTM is decided by the input gate. It generates an update value by taking into account biases, the previous concealed state, and the current input. The update value is a number between 0 and 1, where 0 indicates that no new data is added and 1 indicates that all new data is recorded. An LSTM's input gate functions as a filter to isolate the useful elements of the new memory vector. It generates a vector of values from 0 to 1 using the sigmoid activation function. Through pointwise multiplication, this vector serves as a filter, allowing the input gate to choose which components of the cell state should be updated. If the input gate's output value is low, it's best to leave the corresponding cell state element alone.

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i)$$

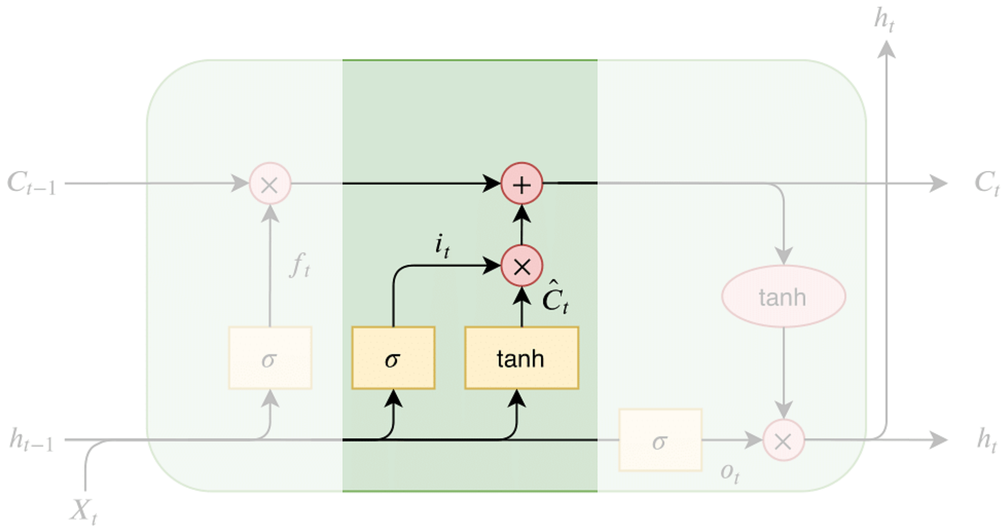


Fig. 5.3. Input Gate

The current input data and the previous hidden state are combined in the new memory network, also called the new memory update vector. It employs the tanh activation function and is trained to produce a vector carrying information from the incoming data while taking the context offered by the prior concealed state into account. This vector establishes how much the cell state—each component of the long-term memory—should be modified in light of the most recent information.

$$\hat{C}_t = \tanh\left(W_C \cdot [h_{t-1}, X_t] + b_C\right)$$

Because the tanh activation function can generate negative values, it can be used to lessen the impact of a component on the state of the cell. To precisely control the impact of various components, it is crucial to have the ability to insert negative values.

The new memory vector alone does not, however, indicate whether or not the new input material is important to recall. As a result, the input gate is used. The input gate selects which components to include in the cell state by filtering the new memory update vector using the sigmoid activation function. Through pointwise multiplication, the input gate regulates the new memory update's output. As a result, the cell state contains only the relevant parts of the new memory update. The network's updated long-term memory is represented by the updated cell state. This rule updates the internal state:

$$C_t = i_t \cdot \hat{C}_t + f_t \cdot C_{t-1}$$

Suppose we have the following input text for sentiment analysis: "স্যারদ্বয় সমীপে আপনারা আমাদের আশার বাতিঘর" (Sir, you are our beacon of hope). In this case, the input gate would decide how much of the 'joy' sentiment should be stored in the cell state. A high value from the input gate would mean that the 'joy' sentiment is crucial and should be retained in the memory.

Output Gate: The amount of information from the current hidden state that should be output depends critically on the function of the output gate in an LSTM. It serves as a filter, enabling the LSTM to manage the significance and volume of the data that is generated. The sigmoid activation function is used by the output gate to create a vector of values between 0 and 1, which represents the gate's control over each component of the hidden state.

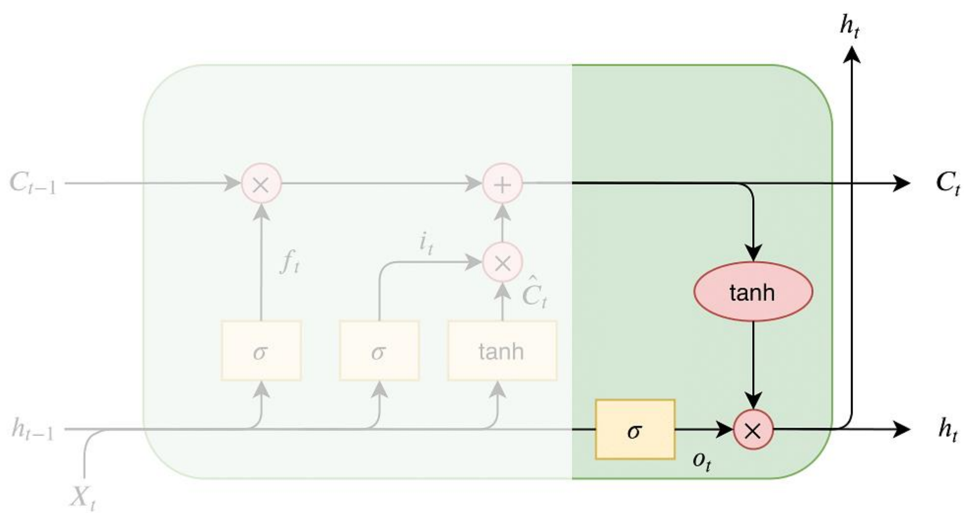


Fig. 5.4. Output Gate

The computation of a pre-activation value is the first stage in the operation of the output gate. By integrating the previous hidden state (h_(t-1)) and the current input (x_t), this pre-activation is created utilizing bias and weight vectors connected to the output gate. The input is being prepared for further processing at this stage. The values between 0 and 1 are condensed by the sigmoid activation function, which is applied after the pre-activation. The degree to which each component of the hidden state is important and should contribute to the output is determined by the ensuing output gate activation.

A high activation value denotes the significance of the information in the hidden state and recommends its inclusion in the output, whereas a low value diminishes its impact. The output gate activation is applied to the present concealed state to produce the final output. The cell state (C_t), which transfers the values to a range between -1 and 1, is passed through the hyperbolic tangent activation function to generate the hidden state. The altered hidden state is then multiplied elementally with the output gate activation. The final output of the LSTM at the specified time step is produced by this element-wise multiplication, which combines the control of the output gate with the pertinent data from the hidden state.

The LSTM can selectively focus on the pertinent information within the hidden state and control its influence on the output by altering the output gate activation. With the help of this technique, the LSTM can capture the hidden state's most important features and generate the right output for tasks like sentiment analysis, where the hidden state's processed data serves as the basis for the prediction or conclusion that is made in the end. The output gate makes sure that the LSTM delivers the data that is most pertinent and useful for the job at hand[22].

Using the same sentiment analysis example, for the text, "স্যারদ্বয় সমীপে আপনারা আমাদের আশার বাতিঘর" (Sir, you are our beacon of hope), the output gate would determine the relevance of the current hidden state in generating the final sentiment prediction. A high value from the output gate would indicate that the current hidden state contains important sentiment information that should be included in the prediction.

By utilizing the input gate, forget gate, and output gate, the LSTM model can selectively update the cell state, retain or discard information from the previous cell state, and determine the amount of information to be output. With the help of this gating mechanism, the LSTM is able to gather pertinent sentiment data over time and produce precise predictions for sentiment analysis tasks. Architecture used for training the model:

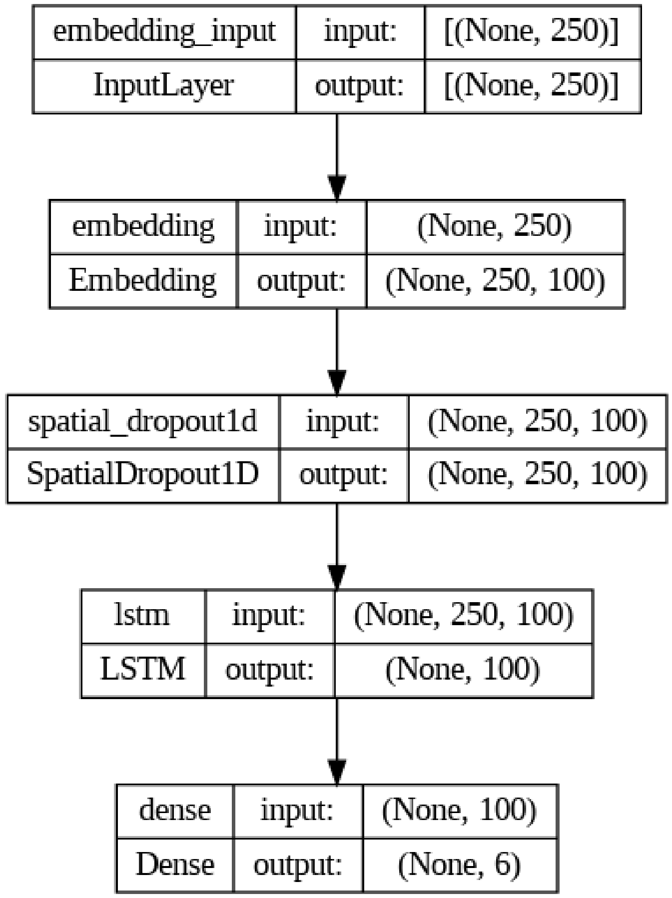


Fig. 5.5. LSTM Architecture

5.1.2 BiLSTM

Bidirectional Long Short-Term Memory (BiSTM) is an extension of the Long Short-Term Memory (LSTM) architecture, a type of recurrent neural network (RNN). BiLSTM is frequently used in natural language processing (NLP) applications where contextual information from both past and future inputs is relevant, such as sentiment analysis, machine translation, and named entity recognition. The vanishing gradient issue in conventional RNNs is addressed by LSTM networks by include memory cells that can preserve information over lengthy sequences[23]. These memory cells can use gates, such as the input gate, forget gate, and output gate, to selectively recall or forget information.

When using bidirectional, our inputs will be processed in two different directions: one being from the present to the future and the other being in the opposite direction. This method differs from unidirectional LSTM in the sense that information from the future is preserved in the LSTM that runs backward, and by combining the two hidden states, we can preserve data from both the present and the future at any given time[24]. Let's assume we want to predict the word "হাতের" (hand) in the given sentence:

সেকেন্ড পরপর আপনার হাতের ভঙ্গিমাটা দেখে আসলেই বিরক্তি লাগছে

It's really annoying to see your hand gesture every second

Conventional LSTM will only see this portion of the sentence:

"সেকেন্ড পরপর আপনার…"

"It's really annoying to see your…"

And will try to predict the next word only by this context. However, bidirectional LSTM you will be able to see information from the last part of the sentence. For example:

Forward LSTM

"সেকেন্ড পরপর আপনার…"

"It's really annoying to see your…"

Backward LSTM

"…ভঙ্গিমাটা দেখে আসলেই বিরক্তি লাগছে"

"…gesture every second."

Having access to information from the future makes it easier for the model to predict the next word and understand context of the sentence more precisely. Also, words can have different meanings based on the context of the sentence. For example:

সে বলল, “ টেডি বিয়ার আমার পছন্দের খেলনা”।

not part of person’s name

(He said, “**Teddy** bear is my favourite toy.”)

সে বলল, “ টেডি রুজভেল্ট ছিলেন মার্কিন যুক্তরাষ্ট্রের ২৬তম প্রেসিডেন্ট”।

part of person’s name

(He said, “**Teddy** Roosevelt was the 26th President of USA.”)

If the word "টেডি " is used here, it is impossible to predict whether the next word will be “বিয়ার” or “রুজভেল্ট” because it depends on the sentence's context. Thus, it has been demonstrated that BiLSTM is effective in a variety of NLP tasks where it is essential to comprehend the context of both past and future information. Bidirectional LSTM models perform better than unidirectional LSTM models by utilizing the complimentary information from both directions.

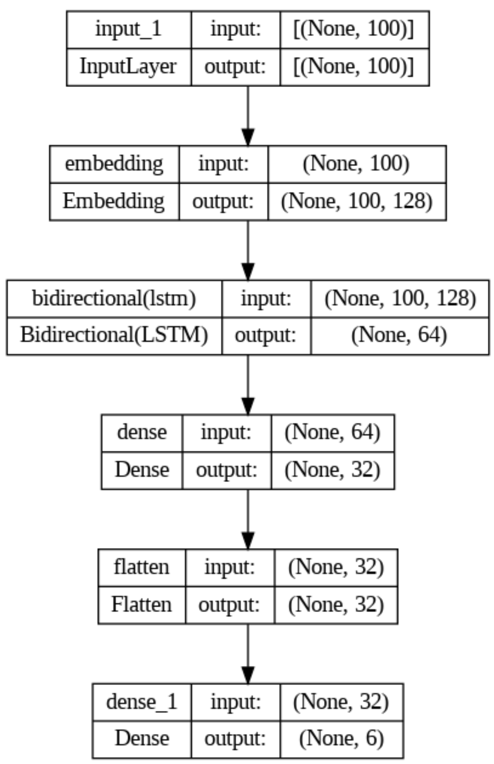


Fig. 5.6. BiLSTM Architecture

5.1.3 CNN

Convolutional Neural Networks (CNNs) epitomize a sophisticated subset of neural architectures, initially devised for tackling computer vision tasks but subsequently demonstrating immense prowess in processing a diverse array of data types. Inspired by biological paradigms, specifically the mechanisms of the human brain's visual cortex, CNNs are predicated upon the mathematical operation of convolution, establishing a salient divergence from conventional neural networks[25]. These networks are fundamentally adept at autonomously and adaptively discerning spatial hierarchies of features. Each layer in the network architecture imparts a series of transformations that progressively discern more complex and abstract feature representations.

A typical CNN is composed of an intricate arrangement of layers, including convolutional layers for salient feature extraction, Rectified Linear Unit (ReLU)[26] layers for the incorporation of non-linear transformations, pooling layers employed for the reduction of dimensionality, fully connected layers for the transformation of multidimensional data into a single dimension (a process colloquially termed "flattening"), and softmax layers for delivering probabilistic classifications. Owing to their inherent capacity to efficiently process spatial data and their compatibility with the 2D structure of inputs, CNNs have become an indelible cornerstone in advancing fields such as image recognition, video analysis, and surprisingly, even non-visual tasks such as natural language processing and speech recognition[27].

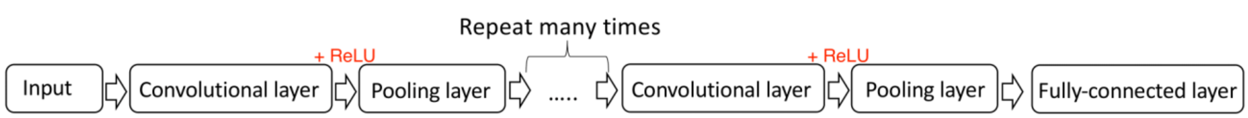


Fig. 5.7. Basic Structure of CNN

Architecture: The structural design of CNNs is inherently multi-layered, with each layer specifically designed to perform a distinct task, and is typically composed of a succession of convolutional layers, interspersed with activation and pooling layers, culminating in fully connected layers towards the end.

Convolutional Layers: Convolutional layers are the cornerstone of CNNs and are responsible for the heavy lifting in terms of computation. Each convolutional layer performs a series of mathematical operations, using a set of learnable filters (also known as kernels), which are small spatially (along width and height), but extend through the full depth of the input volume[28]. For a given layer, the filters are convolved across the width and height of the input volume, computing dot products between the entries of the filter and the input and producing a 2-dimensional activation map for each filter. Intuitively, the network will learn filters that activate when they detect some specific type of feature at some spatial position in the input.

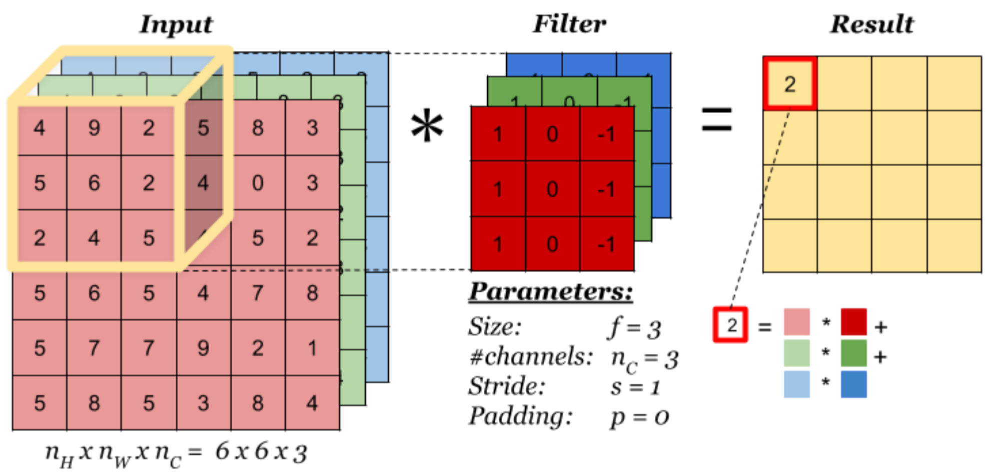


Fig. 5.8. Basic Structure of Convolutional Layer

This operation is performed for every location that the filter can fit on the input volume, producing a feature map. The depth of the output volume is a hyperparameter and it corresponds to the number of filters we use. The stride is another hyperparameter and it corresponds to the number of pixels we are shifting each time we move our filter. While performing the convolution operation, sometimes the filter goes outside the bounds of the input volume. To handle this, we often pad the input volume with zeros around the border[28]. The size of this zero-padding is a hyperparameter. The use of padding allows us to control the spatial size of the output volumes.

A key feature of convolutional layers is that they preserve the spatial relationship between pixels by learning image features using small squares of input data. Convolutional layers use a variation of matrix multiplication called the convolution operation, which makes them adept at handling input data that is grid-like (e.g., an image). The output dimension is calculated with the following formula:

$$n^{[l]} = \left\lfloor \frac{n^{[l-1]} + 2p^{[l-1]} - f^{[l]}}{s^{ll]}} + 1 \right\rfloor$$

When it comes to classify text using CNN, it involves treating text as a one-dimensional image, thereby allowing us to extract features from the "image" of text, much like how we would extract features from a real image using filters. A sentence or a document is first transformed into a matrix representation where each row of the matrix corresponds to a word or a token, represented by a word embedding (like word2vec, GloVe, or even a one-hot encoded representation)[29]. The word embeddings are learned during the training process of the deep learning model or can be initialized with pre-trained embeddings. For an m-word sentence, where each word is represented by an n-dimensional vector, the sentence can be represented by an m x n matrix. In this scenario, the filters are essentially equivalent to n-grams. Just as filters of different sizes can be applied to an image to extract features of different levels, filters (or kernels) of different sizes can be applied to a matrix to extract local features – i.e., n-grams. A kernel size k will be able to detect useful k-word phrases. These learned phrases are position-invariant, meaning that the model can recognize them regardless of their position in the text.

Let's say we have a 7-word sentence and each word is embedded into a 5-dimensional vector. This gives us a 7 x 5 matrix, to which we apply a kernel of size 2 x 5. The convolution operation will then give us a new feature map of dimension 6 x 1. The convolution operation in the context of a text is a sliding window over the input matrix. Specifically, a kernel of width k (smaller than the total number of words in the input) slides across the word vectors (the rows in our matrix), each time spanning k vectors. The kernel performs an element-wise multiplication with the elements of the vectors it spans, and all these multiplied elements are summed up to obtain a single value. This operation is repeated across the input matrix.

Activation Function - Rectified Linear Units (ReLU): Subsequent to the convolution operation in a convolutional layer, an activation function is habitually incorporated to inject non-linearity into the model's architecture. This is necessary because the operations performed during the convolutional layer are linear in nature, consisting solely of element-wise multiplications and summations[26].

The primary rationale for incorporating non-linearity pertains to the existence of non-linear interdependencies among distinct neurons. The convolutional layer is primarily designed to execute a linear operation. As a result, successive convolution layers are practically indistinguishable from a solitary convolution layer, which serves only to diminish the representational capacity of the networks. The absence of non-linearity property among neurons has not been adequately addressed, necessitating the implementation of an activation function between the convolutional layer to mitigate this concern.

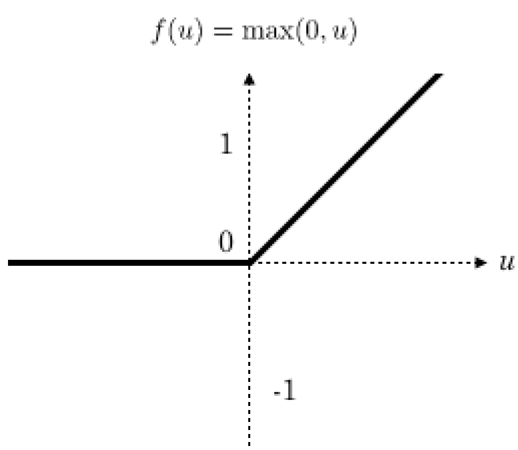


Fig. 5.9. Rectified Linear Unit

The activation function is a pivotal component in Convolutional Neural Networks (CNNs) as it executes a non-linear transformation that determines the activation or deactivation of a neuron. Numerous activation functions are accessible subsequent to the convolutional layer, including but not limited to the hyperbolic and sigmoid functions. Among these, the rectified linear unit (ReLU) is the most frequently employed activation function in neural networks, particularly in convolutional neural networks (CNNs)[30].This is due to its possession of two advantageous properties.

Non-linearity is a fundamental concept in the field of deep learning. Rectified Linear Unit (ReLU) is a commonly used activation function in neural networks. It is defined mathematically as follows:

$$R(z) = z^{+} = max(0, z)$$

The variable 'z' represents the output element of the preceding convolutional layer. The rectified linear unit (ReLU) activation function will be applied to the feature maps from the preceding layer, whereby any negative values will be rectified to zero. On the other hand, the concept of non-saturation refers to the absence of saturation in arithmetic operations. Saturation arithmetic, on the other hand, is characterised by the restriction of all operations within a predetermined range bounded by a minimum and maximum value.

- f is non-saturating iff $\left(\left|\lim_{z\to-\infty} f(z)\right| = +\infty\right) \cup \left(\left|\lim_{z\to+\infty} f(z)\right| = +\infty\right)$
- f is saturating iff f is not non-saturating

Pooling Layers: Within the architecture of a CNN, pooling layers execute a down-sampling operation across the spatial dimensions (width and height) of the input, engendering features invariant to changes in scale and orientation. Nevertheless, in the context of text classification, where data is fundamentally structured along a singular dimension (i.e., the length of the text), the purpose of a pooling layer transforms. Here, the objective becomes the reduction of the spatial size of the representation, in order to decrease the volume of parameters and computational requirements in the network, thereby aiding in the control of overfitting.

Multiple variants of pooling operations exist, inclusive of Max Pooling, Average Pooling, and Global Max Pooling, each possessing distinct characteristics. However, Max Pooling is predominantly utilized owing to its demonstrated performance advantages. Within the scope of text classification, the most prevalent form of pooling employed is the one-dimensional rendition of Max Pooling or Global Max Pooling. For a rudimentary understanding, Max Pooling operates by progressing a window across the input and selecting the maximum value within the window to represent the output. Given a 1-dimensional matrix of size [1 x n] (emerging from a convolution operation) and a pooling window of size k, Max Pooling will yield a matrix of size [1 x (n-k+1)]. In a text classification task, the pooling layer serves to capture the most pertinent feature - which could be a specific amalgamation of words or phrases - contributing most significantly to the final classification. It is vital to note that the size of the pooling window and the stride with which it traverses across the feature map are hyperparameters of the model, requiring tuning for optimal performance.

Fully Connected Layers: In the architecture of a CNN, fully connected layers are pivotal components, functioning as a critical conduit between the feature extraction phase and the model's final output. Subsequent to the processing through convolutional and pooling layers - which extract an array of local and global features from the input text - these features are typically reshaped into a singular vector of values. This vector is then funneled into the fully connected layers. Each neuron within a fully connected layer holds connections to all neurons in its preceding layer ; hence the term 'fully connected'. The primary objective of these layers is to learn non-linear combinations of the high-level features, as represented by the output of the previous layer.

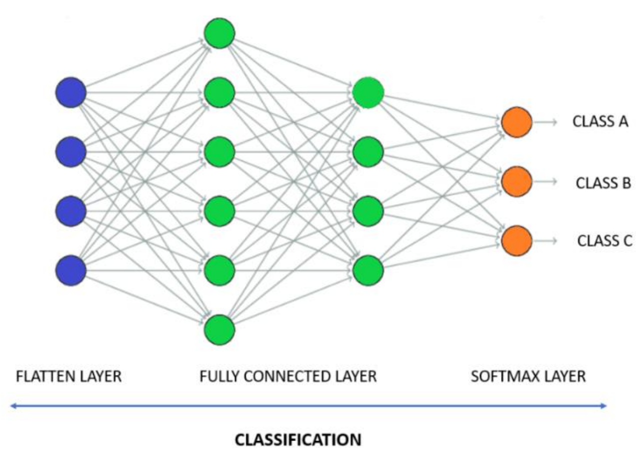


Fig. 5.10. Classification of Layers

In the context of text classification, a fully connected layer functions to identify and learn higher-level global patterns from the input text features. For instance, the convolutional layers might extract local keyword features, while the fully connected layers could learn that the presence of certain combinations of these keywords in a text indicates a specific sentiment or category. Typically, a CNN for text classification will end with one or more fully connected layers leading up to the output layer. The final fully connected layer has a number of neurons equal to the number of classes in the classification task, with each neuron representing a specific class. The activation of these neurons is calculated using the Softmax function, which produces a probability distribution over the classes, offering a probabilistic interpretation for the class prediction.

Softmax Layer: The Softmax layer, typically positioned at the end of the CNN, functions as a generalized logistic regression model, translating the non-normalized output of the previous layer (usually a fully connected layer) into a probability distribution over predefined class. This is particularly vital in multi-class classification tasks, including those in text classification[31].

Mathematically, the softmax function 'squashes' a K-dimensional vector of arbitrary real values, denoted by z, to a K-dimensional vector of real values in the range [0, 1] that add up to 1. For a given input vector z of length K, the softmax function, σ, is defined as follows:

$$\phi(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{n} \exp\left(z_j\right)}$$

$$\text{For } 0 \leq y_i(\boldsymbol{x}) \leq 1, \sum_{i=1}^{n} y_i(\boldsymbol{x}) = 1$$

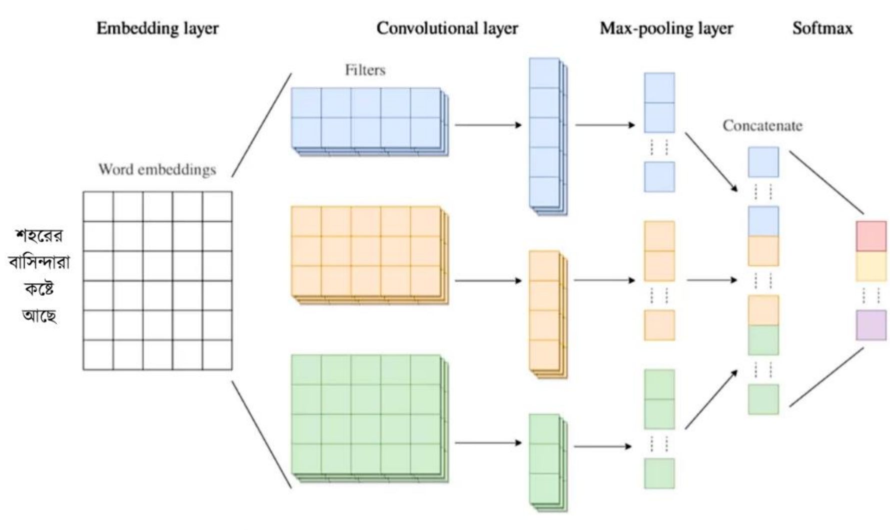


Fig. 5.11. General Structure of CNN Architecture

Here, $\exp(z_i)$ is the exponential of the i-th element of the input vector, and the denominator $\sum_{j=1}^{n} \exp(z_j) = 1.\ \sum_{j=1}^{n} \exp\left(z_j\right)$ is the sum of the exponentials of all the elements of the input vector. In a text classification task, the softmax layer acts as the final decision maker. It determines the class of the input text by choosing the class with the highest probability.

5.1.4 BiGRU

BiGRU is a type of recurrent neural network (RNN) architecture that combines bidirectional processing with Gated Recurrent Units (GRUs). GRUs are a variant of the more traditional Long Short-Term Memory (LSTM) units, both of which are commonly used in sequence modeling tasks. The bidirectional nature of BiGRU allows it to process input sequences in both forward and backward directions, capturing dependencies from past and future contexts simultaneously. Gated Recurrent Units (GRUs), in particular, offer a streamlined version of the LSTM architecture[32]. By employing two gates—the update gate and the reset gate—instead of the three gates used in LSTM models (input, forget, and output gates), GRUs reduce the number of parameters, rendering them more computationally efficient. Bidirectional Gated Recurrent Units (BiGRUs), a variation of GRUs, have been designed to process data bidirectionally. The primary innovation in BiGRUs lies in the separation of GRU neurons into two distinct directions—one propagating in the positive time direction (forward states), and the other in the negative time direction (backward states). Each of these is a fully functional GRU, and their respective outputs are concatenated at each time step.

While traditional RNNs or GRUs might suffer from an information bottleneck if crucial contextual information for a data point is situated several steps away (either before or after), a BiGRU has the potential to preserve and utilize this information more effectively. This potentially leads to superior performance in various tasks. The bidirectionality of BiGRUs thus constitutes a crucial feature for tasks that require a comprehensive context, where both past and future information is necessary to generate accurate outputs.

Architecture of the model: The Bidirectional Gated Recurrent Unit (BiGRU) introduces a remarkable extension to the original Gated Recurrent Unit (GRU) model by effectively processing sequential data in both forward and backward directions. The architecture of a BiGRU comprises two separate yet interconnected layers, each designed to handle forward and backward states. Every layer houses a sequence of Gated Recurrent Units (GRUs), and each GRU in turn employs two gating mechanisms: the update gate and the reset gate[32].

Update Gate Mechanism: The update gate plays a pivotal role in regulating the flow of information from the previous hidden state to the next. By modulating the degree of influence the past state has on the current state, the update gate essentially controls how much information from the past is carried forward. This selective memory characteristic is crucial for handling sequential data with varying dependencies over time. The update gate (Z) is computed using the following equation

$$Z = \sigma(Wz \cdot [h_{t-1}, x_t] + bz)$$

In this equation, Wz denotes the weight matrix associated with the update gate, while x_t represents the current input. The previous hidden state is given by h_{t-1} and bz stands for the bias term. The sigmoid activation function, denoted as σ, is utilized to ensure the output values of the gate lie between 0 and 1, thus representing a probability.

Reset Gate Mechanism: The reset gate mechanism provides the model with the capacity to discard information that becomes irrelevant to future predictions. This enhances the model's flexibility in controlling the amount of past information that should be remembered, which is especially critical when the model encounters a significant shift in the data sequence. The reset gate (R) is calculated using the equation:

$$R = \sigma(Wr \cdot [h_{t-1}, x_t] + br)$$

Here, Wr signifies the weights for the reset gate, while br denotes the corresponding bias term. The sigmoid function is again employed, maintaining the output within the range of 0 to 1.

Computation of the Current Hidden State: The current hidden state, often perceived as a summary of the sequence up to the current point, is formulated through a combination of the previous hidden state and the candidate hidden state, modulated by the update and reset gates. The candidate hidden state (H) is given by the equation:

$$H = \tanh\left(Wh \cdot [R \cdot h_{t-1}, x_t] + bh\right)$$

In this expression, Wh denotes the weights for the candidate hidden state, bh is the corresponding bias term, and the tanh function ensures the values are maintained between -1 and 1. The final hidden state h_{t-1} is then calculated by:

$$h_t = (1 - Z)h_{t-1} + ZH$$

This equation signifies that the new state is an interpolation between the previous hidden state and the candidate state, governed by the update gate.

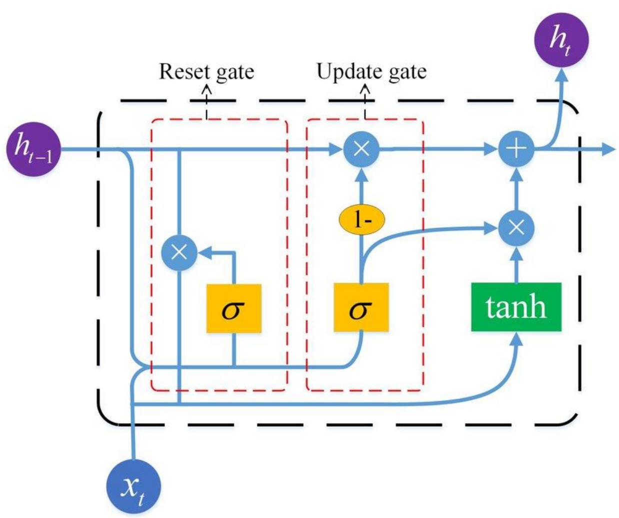


Fig. 5.12. Current Hidden State Computation Process

This equation signifies that the new state is an interpolation between the previous hidden state and the candidate state, governed by the update gate.

Forward and Backward Passes: The defining characteristic of BiGRU lies in its dual-directional information propagation. The forward pass effectively transmits information from the start of the sequence to the end, encapsulating the forward context of the sequence. Concurrently, the backward pass processes data in the reverse direction, ensuring the backward context is also comprehensively accounted for. The forward and backward hidden states at each time step are concatenated to form the final output. This two-way flow of information equips the BiGRU model with a more comprehensive understanding of the sequence, enabling superior performance on tasks requiring intricate context perception.

Interaction of Gates: A crucial aspect of the BiGRU architecture is how the update and reset gates interact in determining the hidden state at each time step. The reset gate effectively decides how much of the past information should be discarded, whereas the update gate chooses between the candidate hidden state (which carries the present input information) and the previous hidden state. By jointly operating these gates, the model is able to maintain long-term dependencies and filter out irrelevant details, improving its ability to capture patterns and make accurate predictions.

Learning Parameters: The learning parameters in a BiGRU model comprise the weights (Wz, Wr, Wh) and biases (bz, br, bh) associated with each gate and the candidate hidden state. During training, the model learns these parameters through a process of backpropagation and optimization in order to minimize the loss function. The weight matrices determine the contribution of the inputs and the hidden states to the computation of the gates and the candidate hidden state, while the bias terms offset these contributions. It's the optimization of these parameters that allows the BiGRU model to effectively learn from sequential data.

How the Model Processes the Data: To perform emotion classification using a BiGRU model, we need to preprocess the sentences, tokenize the words, and convert them into numerical inputs. Then, we can analyze how the internal layers of the BiGRU model work to classify the emotions.

Let's go through the process step by step using an example sentence: **"সরকারের এই উদ্যোগ প্রশংসাযোগ্য !!"**

Preprocessing and Tokenization: Before feeding the sentence into the BiGRU model, we need to preprocess it by removing punctuation, converting to lowercase, and handling any special characters. After preprocessing, the sentence becomes: "সরকারের এই উদ্যোগ প্রশংসাযোগ্য ". Next, we tokenize the sentence into individual words: ["সরকারের", "এই ", "উদ্যোগ ", "প্রশংসাযোগ্য"]. Each word needs to be represented by a numerical vector, known as word embeddings. These embeddings capture the semantic meaning of the words. We can use pre-trained word embeddings such as Word2Vec or GloVe, or we can initialize random embeddings and learn them during training.

For example, let's assume we have the following word embeddings for the words in our sentence:

" সরকারের ": [0.2, -0.1, 0.3, ...]

" এই ": [-0.4, 0.6, 0.1, ...]

" উদ্যোগ ": [0.7, 0.2, -0.5, ...]

" প্রশংসাযোগ্য ": [0.8, 0.4, 0.9, ...]

Now that we have word embeddings, we can represent each word in the sentence as a numerical input by looking up their respective embeddings[29]. The sentence " সরকারের এই উদ্যোগ প্রশংসাযোগ্য " becomes the following numerical input sequence:

[[0.2, -0.1, 0.3, ...], [-0.4, 0.6, 0.1, ...], [0.7, 0.2, -0.5, ...], [0.8, 0.4, 0.9, ...]]

As mentioned before, a unique attribute of the BiGRU model is its bidirectional nature, which allows it to capture information from both past and future states. The forward and backward passes each generate their own sequence of hidden states. These sequences are then concatenated at each time step to produce the final output sequence. The synthesis of these two sequences into a single output sequence allows the model to capitalize on both past and future contexts when making predictions. The model's bidirectionality, coupled with its gated memory mechanisms, afford it superior performance in numerous tasks involving sequential data, particularly those requiring the understanding of long-range dependencies and contextual information. This intricate architecture positions the BiGRU model as a significant advancement in the field of recurrent neural networks.

5.1.5 Ensemble Learning

A strategy called ensemble learning involves fitting two or more models to the same data and combining the predictions from each model. The goal of ensemble learning is to outperform each individual model using the group of models as a whole. A fusion layer is created for this first. A fusion layer is a layer that integrates the results of several earlier layers or models into a single result. It combines data from several sources to provide a fused representation that incorporates the features and collective knowledge from various models or layers[33]. When a prediction or relevant information needs to be extracted from many models or various data modalities, the fusion layer is frequently employed in ensemble learning or multi-modal learning scenarios. It enables the incorporation of supplementary data from many sources, enhancing the model's overall functionality and robustness.

We have created the fusion layer using the 'concatenate ()' function from the Keras library[34]. The 'concatenate ()' function takes a list of tensors as input and concatenates them along a specified axis. In our case, the fusion layer concatenates the outputs of three models: BiGRU_model.output, BiLSTM_model.output and CNN_model.output. The fusion layer combines the representations learned by these models, creating a fused representation that captures information from all three sources. The fusion layer is an essential component as it enables the combination of different models or modalities, allowing them to contribute to the final prediction or decision-making process. By fusing the outputs of multiple models or layers, the fusion layer provides a more comprehensive and powerful representation of the data, enhancing the model's performance and capability to learn complex patterns and relationships.

The resulting fusion layer is then passed through a dense layer with 16 units and Rectified Linear Unit (ReLU) activation. After that, the output of the dense layer is flattened using the ‘Flatten ()’ layer. Finally, the flattened output is fed into another dense layer with 6 units and softmax activation to obtain the final output. This outputs functions as the output of the newly created Ensembled learning model.

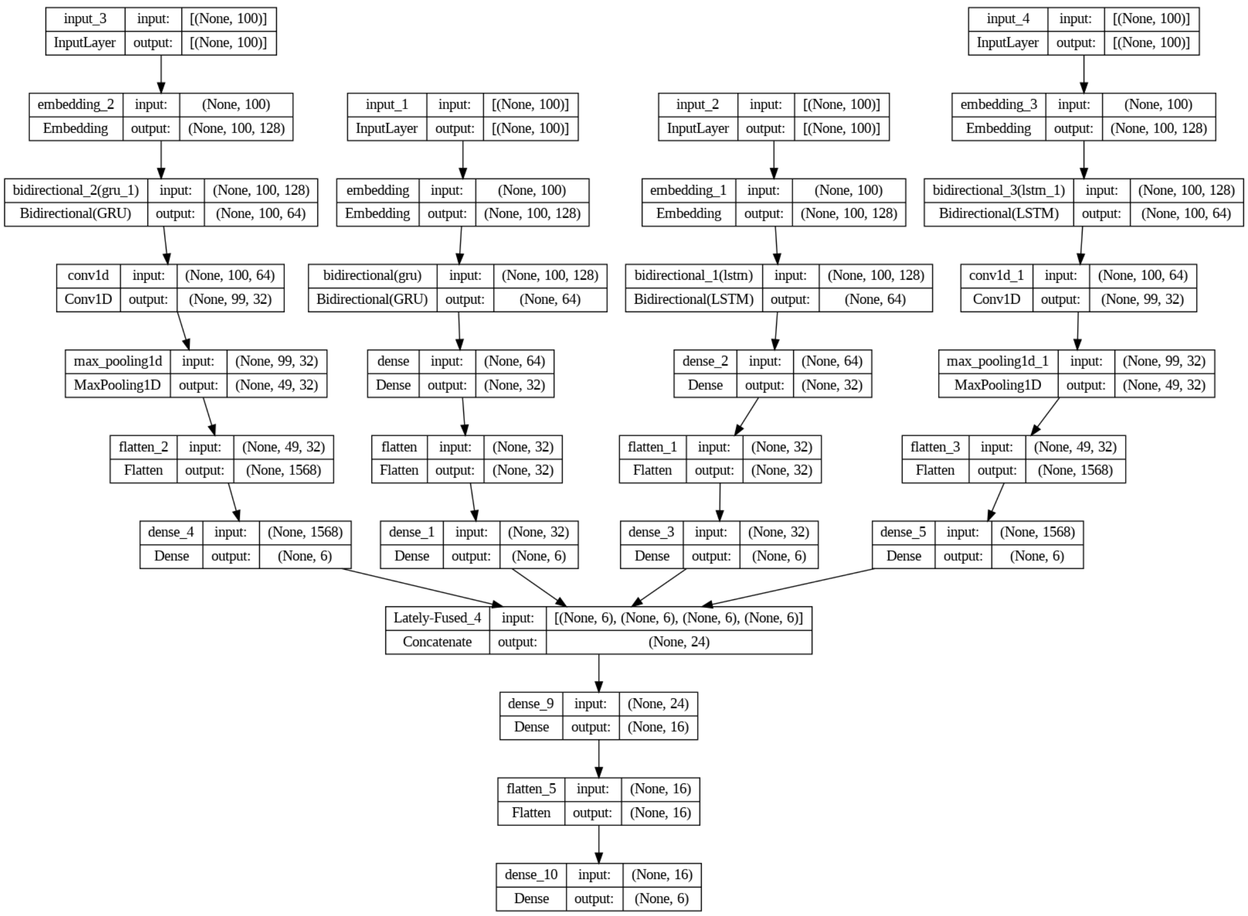


Fig. 5.13. Ensemble Learning Model Architecture

5.2 Training Arrangement

We used the TensorFlow library for Python Programming language to train our dataset. Adam optimizer was used to compile the text classifier model with the default learning rate of 0.001. Along with Categorical Cross-entropy[35] as loss function, supportive Dense Neural Network layers[36] with softmax activation were used in the overall training process. We faced difficulties while training the model as extensive computational capabilities were needed to keep the server connected and complete the training within the allotted time. We chose a batch size of 64 due to hardware constraints in order to avoid running out of memory when training.

5.2.1 Activation Function

An essential procedure used in neural networks is back-propagation, which modifies the weights and biases of neurons based on the output error. It enables the network to develop and enhance its forecasts over time. Activation functions are essential to back-propagation because they supply the gradients required to update the parameters. When dealing with multi-class classification tasks, the softmax activation function is a frequently employed activation function in the output layer of a neural network[37]. The logistic function (sigmoid) is made more inclusive to accommodate many dimensions or classes. A vector of real-valued inputs is given to the softmax function, which normalizes them into a probability distribution. The softmax equation is given by:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

$\sigma \implies$ softmax

$\vec{z_i} \implies$ input vector
$e^{z_i} \implies$ standard exponential function for input vector
$K \implies$ number of classes in the multi-class classifier
$e^{z_j} \implies$ standard exponential function for output vector

The output numbers must be non-negative and add up to 1, which represents probabilities, according to the softmax activation function. In order to enable meaningful interpretations and improve decision-making, it converts the network's unprocessed outputs into a probability distribution over the classes. The softmax activation function supplies the required gradients for the weight updates and error computation during the back-propagation procedure. Based on the discrepancy between the expected probabilities and the actual labels, the gradients are calculated. Gradients flow backward through the network when the chain rule of differentiation is used, allowing the weights and biases of the neurons to be changed in accordance with their contribution to the total error.

5.2.2 Optimizer

We have applied the stochastic gradient descent algorithm known as the Adaptive Moment Optimization Algorithm (ADAM) optimizer, which is based on adaptive estimate of first- and second-order moments. Adam focuses on optimizing the process for certain parameters within a neural network using adaptive learning rates. Adam makes sure that the optimization process is successful and efficient by dynamically calculating and modifying the learning rates. Different learning rates can be applied to various parameters using this adaptive modification, taking into account their unique needs and characteristics.

Adam carries out two crucial steps: first moment estimation (mean) and second moment estimation (variance) to achieve this adaptive learning rate. Calculating the gradients' average exponential decay is required for the first moment estimation. This is the average gradient direction throughout all repetitions, indicating the general tendency. On the other hand, the second moment estimation entails computing the squared gradient average with exponential decay[38]. This displays how the gradients have changed or spread during the iterations. Adam includes bias correction during the initial iterations to offset the moments' initialization bias. Particularly when there are few iterations, this adjustment makes sure that the moments are adjusted properly. By executing these actions, Adam successfully calculates the gradient moments, enabling a more precise and reliable learning rate adaption. The optimizer then goes on to change the neural network's parameters based on the calculated learning rates and gradients. The gradients are normalized using the first and second moments, which are also used to establish the direction and size of the parameter update. The network's parameters are improved in the best possible and most efficient way thanks to this update process. The ability of the Adam optimizer to automatically adjust the learning rate for each parameter depending on the observed gradients was probably a major factor in its selection for your NLP work. Faster convergence and better optimization performance are made possible by this adaptive nature, which is essential when working with large-scale datasets and intricate neural network topologies.

5.2.3 Training Experiments

The following models were used to train our proposed model:

- LSTM
- CNN
- BiLSTM
- BiGRU
- Ensemble Learning

5.2.3.1 LSTM With Early Stopping

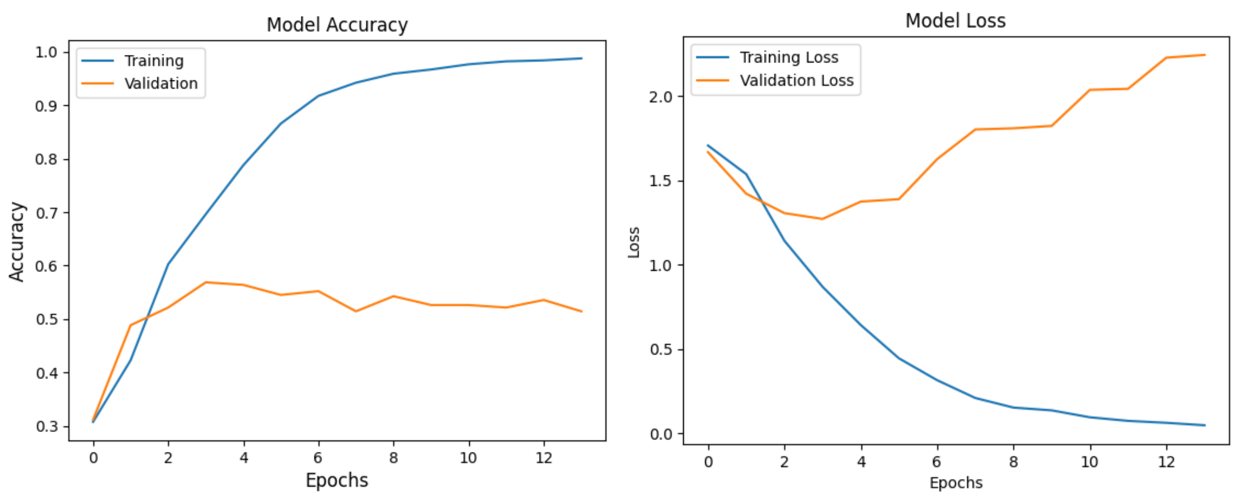


Fig. 5.14. Comparison between Model Accuracy and Model Loss (LSTM Model)

5.2.3.2 CNN With Early Stopping

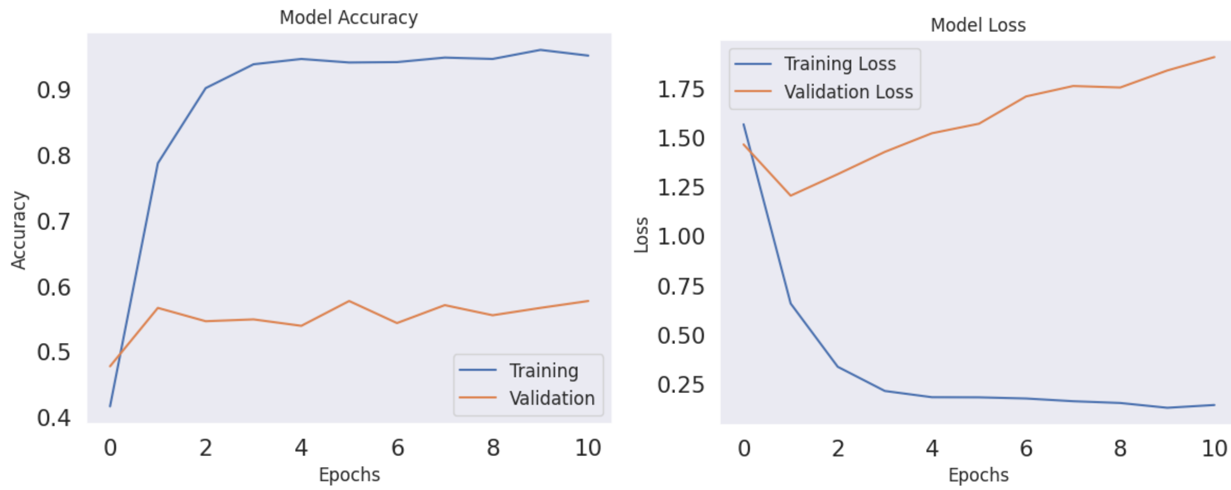


Fig. 5.15. Comparison between Model Accuracy and Model Loss (CNN Model)

5.2.3.3 BiGRU with Early Stopping

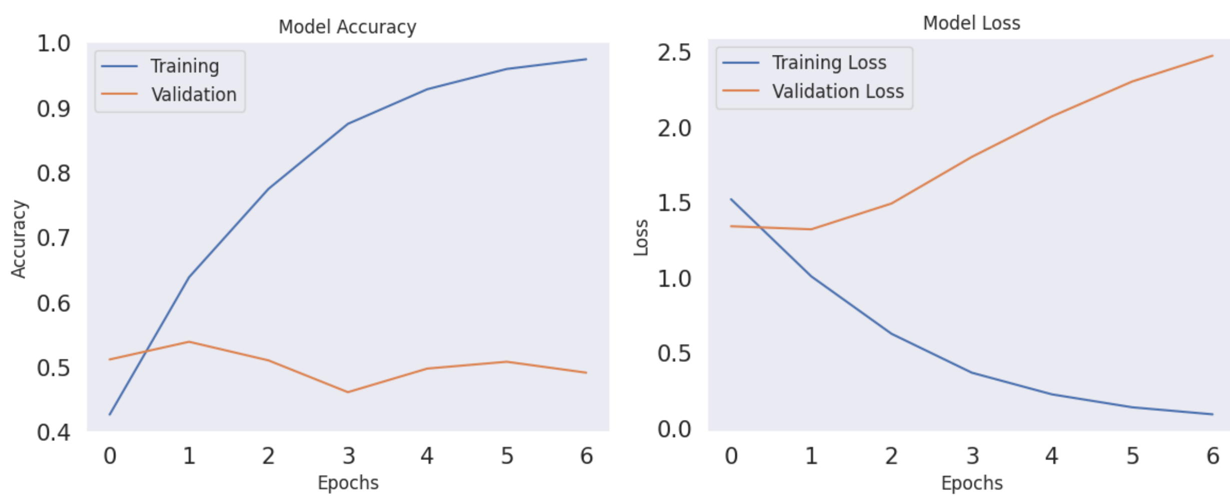


Fig. 5.16. Comparison between Model Accuracy and Model Loss (BiGRU Model)

5.2.3.4 Ensembled Model with Early Stopping

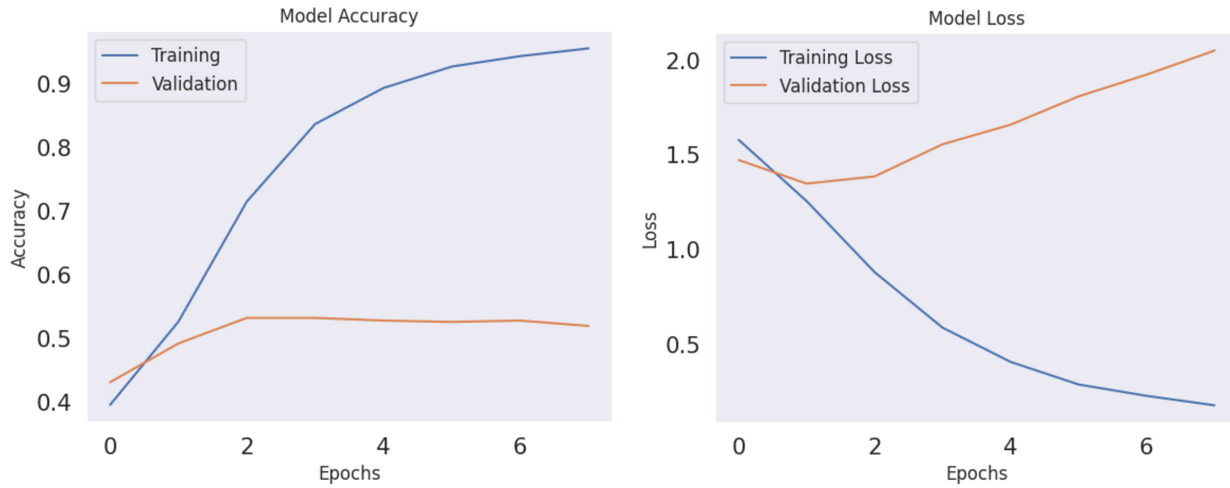


Fig. 5.17. Comparison between Model Accuracy and Model Loss (Ensembled Model)

Chapter 6

Result and Discussion

6.1 Result Metrics

The obtained results by building our model shows significant dependence on the classification we choose for our sentiment analysis. To give an overview of the results, we decided to use machine learning metrics such as Precision, Recall and F1 − score [19].

6.1.1 Precision

Precision is used in statistics and machine learning to assess how well a categorization model is performing. Out of all cases anticipated as positive, it calculates the percentage of correctly predicted positive instances. In other words, precision emphasizes the accuracy of the model's optimistic forecasts[39]. The following formula is used to determine precision:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives } + \text{ False Positives}}$$

Here:

- True Positives (TP) refers to the number of instances that are correctly predicted as positive.
- False Positives (FP) refers to the number of instances that are incorrectly predicted as positive when they are actually negative.

Precision provides insights into the model's ability to avoid false positive predictions. It indicates how well the model performs in correctly identifying positive instances without mistakenly classifying negative instances as positive[40]. A high precision value indicates that the model has a low rate of false positives, meaning it is good at identifying positive instances correctly. On the other hand, a low precision value suggests a higher rate of false positives, indicating that the model may be incorrectly predicting positive instances, leading to more false alarms or incorrect classifications.

6.1.2 Recall

Recall, sometimes referred to as sensitivity or true positive rate, is a metric used in statistics and machine learning to assess how well a categorization model performs. Out of all the real positive instances in the dataset, it calculates the percentage of accurately predicted positive instances. In other words, recall focuses on the model's capacity to recognize and record good occurrences[40]. The following formula is used to determine recall:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Here:

- True Positives (TP) refers to the number of instances that are correctly predicted as positive.
- False Negatives (FN) refers to the number of instances that are incorrectly predicted as negative when they are actually positive.

Recall sheds light on the model's capacity to steer clear of unwarranted negative predictions. It shows how effectively the model captures positive examples and reduces the proportion of cases that are mistakenly categorized as negative. A high recall number indicates that the model effectively recognizes the majority of the positive cases in the dataset and that it has a low rate of false negatives. This shows that the model is highly sensitive and can successfully identify the majority of positive cases. A low recall score, on the other hand, denotes a higher incidence of false negatives and implies that the model fails to detect a sizable proportion of positive cases[39]. This may result in missed chances or incorrectly labeling things negatively.

6.1.3 F1 Score

The F1 score is a metric frequently used in statistics and machine learning to assess the effectiveness of a classification model. It provides a balanced measurement of both measures because it is the harmonic mean of precision and recall. The F1 score takes into account the model's precision (ability to predict positive outcomes with accuracy) and recall (capability to identify positive cases)[40]. The F1 score is calculated using the following formula:

$$\text{F1 Score} = 2 * \frac{(\text{Precision} \ * \ \text{Recall})}{(\text{Precision} \ + \ \text{Recall})}$$

The F1 score has a range of 0 to 1, with 0 denoting subpar performance and 1 denoting flawless precision and recall. When precision and recall are equally crucial or when there is an imbalance between the positive and negative classes, the F1 score is very helpful. It offers a solitary metric that balances the compromise between recall and precision. The harmonic mean is used in the F1 score to give lower numbers more weight. This indicates that the F1 score will be closer to the lower value if either precision or recall are low. It discourages unbalanced performance by penalizing models with a large discrepancy between precision and recall. When the F1 score is high, it means that the model achieves both accurate positive predictions and captures the majority of the positive cases, demonstrating a strong balance between accuracy and recall. A low F1 score, on the other hand, denotes a substantial precision/recall imbalance or subpar performance in either metric.

6.1.4 Confusion Matrix

A confusion matrix, also called an error matrix, is a table that summarizes the counts of true positive, true negative, false positive, and false negative predictions to show how well a classification model performs. It is a helpful technique for assessing a classification model's performance, particularly in cases where the distribution of the classes is unbalanced. The classification problem's number of classes determines the dimensions of the confusion matrix, which is commonly a square matrix. In our case, the matrix is a 6x6 table. The examples in a predicted class are represented by each row in the matrix, and the occurrences in an actual class are represented by each column. The confusion matrix offers a thorough analysis of the model's performance, enabling us to determine its advantages and disadvantages. It reveals whether the model often commits more false positive than false negative mistakes, as well as whether it behaves differently for other classes. We can obtain understanding of the model's behavior and reach conclusions on how to enhance it by examining the confusion matrix.

6.2 Model Evaluation

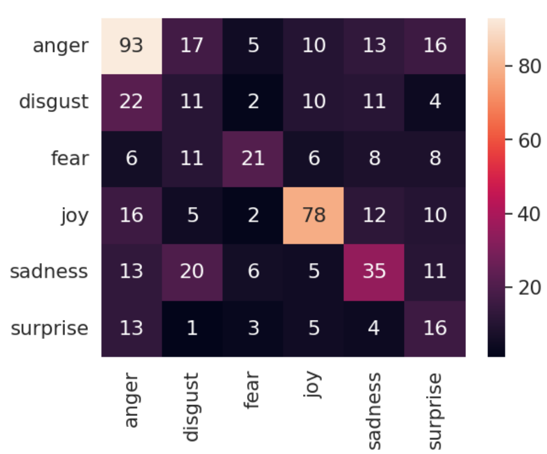
We evaluate our experimented models with the metrics mentioned in this section. The Preciosn, Recall and F1 Score of each model is given below

TABLE 6.1. PRECISION, RECALL AND F1 SCORE OF ALL MODELS

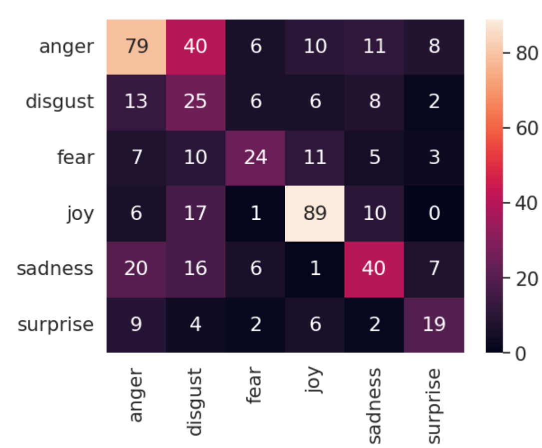
Model	Class	Precision	Recall	F1 Score
LSTM	Anger	0.56	0.52	0.57
	Disgust	0.31	0.29	0.27
	Fear	0.47	0.43	0.45
	Joy	0.66	0.69	0.61
	Sadness	0.39	0.42	0.40
	Surprise	0.32	0.35	0.33
BiLSTM	Anger	0.57	0.60	0.59
	Disgust	0.17	0.18	0.18
	Fear	0.54	0.35	0.42
	Joy	0.68	0.63	0.66
	Sadness	0.42	0.39	0.40
	Surprise	0.25	0.38	0.30
CNN	Anger	0.52	0.53	0.52
	Disgust	0.26	0.15	0.19
	Fear	0.47	0.40	0.43

	Joy	0.64	0.75	0.69
	Sadness	0.51	0.54	0.53
	Surprise	0.49	0.52	0.51
BiGRU	Anger	0.67	0.42	0.51
	Disgust	0.21	0.40	0.27
	Fear	0.38	0.48	0.42
	Joy	0.74	0.67	0.71
	Sadness	0.53	0.53	0.53
	Surprise	0.34	0.31	0.33
Ensemble Learning (CNN+BiLSTM+BiGRU)	Anger	0.51	0.48	0.50
	Disgust	0.35	0.49	0.41
	Fear	0.61	0.45	0.52
	Joy	0.61	0.62	0.62
	Sadness	0.49	0.37	0.42
	Surprise	0.43	0.47	0.45

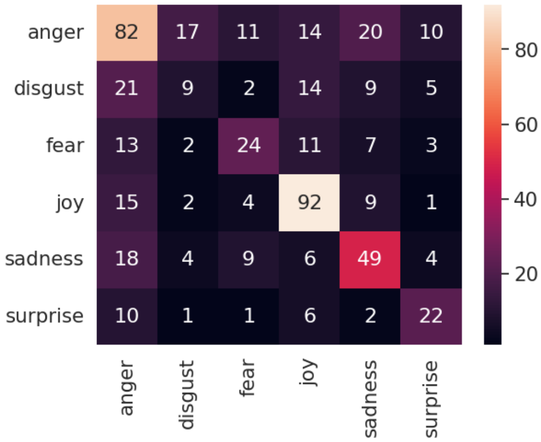
6.2.1 Confusion Matrix



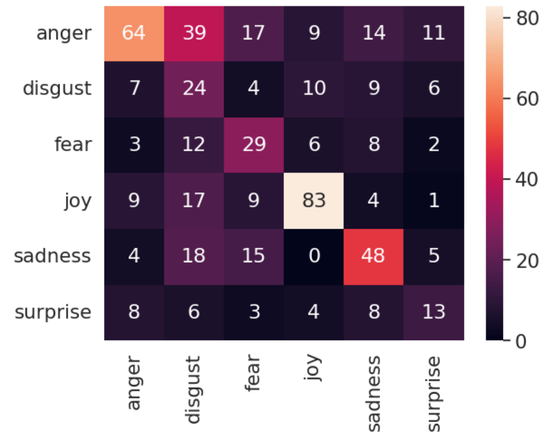
(a) LSTM



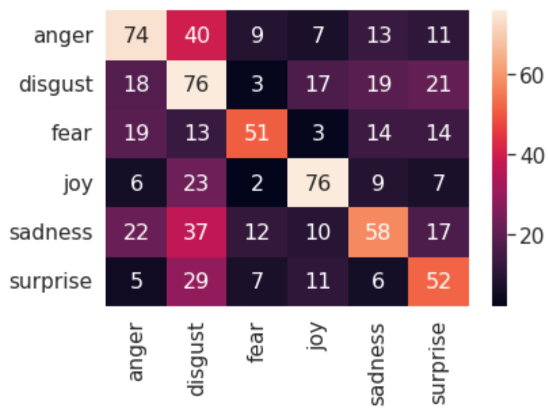
(b) BiLSTM



(c) CNN



(d) BiGRU



(e) Ensemble Learning

Fig. 6.1. Confusion Matrix of different Models

6.2.2 End to End Interface

After training the model, we've tested it by giving different random sentences as input prompt and the model predicted the sentiment of the sentence. Here are some Examples:

```
check_sentiment('নতুন বিদ্যুৎকেন্দ্র স্থাপন করা হয়েছে')
1/1 [==============================] - 0s 93ms/step
'joy'
```

```
check_sentiment('সব তেল দিয়ে দে হাঁউয়া রে')
1/1 [==============================] - 0s 89ms/step
'anger'
```

```
check_sentiment('এই হলো আমাদের ডিজিটাল বাংলাদেশ এর দশা')
1/1 [==============================] - 0s 75ms/step
'disgust'
```

```
check_sentiment('চার ঘণ্টা ধরে কারেন্ট নাই। ছেলেটা গরমে ছোটফট করছে')
1/1 [==============================] - 1s 619ms/step
'sadness'
```

```
sentiment('তুমি পরীক্ষায় এত ভালো করবে ভাবিনি')
1/1 [==============================] - 0s 30ms/step
'surprise'
```

```
check_sentiment('আমার ভয় করছে না জানে কি হয়')
1/1 [==============================] - 0s 88ms/step
'fear'
```

6.3 Discussion

From the above data and graphs it is visible that the overall accuracy of the models is relatively poor. But anomaly this can be explained by the confusion matrices. We can see that every model is getting confused between anger and disgust. In the Ensemble Learning Confusion Matrix, 40 texts that were actually labelled as ‘anger’ was predicted as ‘disgust’ by our model. Also 18 texts that were actually labelled as ‘disgust’ was predicted as ‘anger’ by our model. This problem arises due to the similarity of sentiment between ‘anger’ and ‘disgust’. In most cases, texts that are classified as ‘anger’, can also be classified as ‘disgust’ and vice versa. Due to this the overall accuracy of our model decreases. Similar anomaly can be seen between disgust and sadness.

The imbalanced nature of our dataset also leads to low accuracy. We have labelled almost 10000 texts but found only a few texts that actually indicates that the writer is surprised. As it is really difficult to gather texts that depicts ‘surprise’ as the sentiment, this causes the problem of overfitting in our models. Early stopping was the method employed to stop the overfitting side effect. More epochs result in a leveling of accuracy due to overfitting of the model because the model cannot improve accuracy on the same data, as can be seen from the learning curve in figure 5.6. The model learns more characteristics from the dataset as a result of a high number of epochs, which improves accuracy but also causes overfitting. Additionally, after a given number of iterations, the accuracy increment level flattens out and the model stops becoming better. Early stopping is utilized to lessen the impact of overfitting since early stopping will result in near to maximum accuracy without wasting time while also lowering the likelihood of overfitting. We have used early stopping in all of our models to reduce the overfitting problem as much as possible.

Chapter 7

Conclusion

7.1 Reflection

The research path into Natural Language Processing (NLP) and sentiment analysis has been a thrilling one. The amount of information obtained during this process has been immense, allowing us to establish competence in a variety of sectors.

- We have established expertise in natural language processing techniques leveraging neural networks.
- This involves learning pre-processing techniques like tokenization, stemming, and parts of speech labeling, which are essential for language analysis.
- We investigated and implemented a variety of diverse neural network modeling methodologies. Various models have been used to decipher the nuances of sentiment analysis, including CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory), BiGRU (Bidirectional Gated Recurrent Unit), BiLSTM (Bidirectional Long-Short-Term Memory), and Ensemble Model. Each of these models has distinct traits and benefits, allowing us to gain a thorough grasp of their applicability to sentiment analysis tasks.

Several problems arose during the study process that required critical analysis and problem-solving.

- One key problem was ensuring that each comment in the dataset was correctly labeled. The precision of the labeling procedure has a significant impact on the quality and dependability of the sentiment analysis results.
- Another obstacle was the model's interpretability. While great accuracy is necessary, it is also critical to comprehend and explain the model's verdicts. Creating explainable and transparent models improves the overall credibility and usefulness of the sentiment analysis system.

Although the accuracy of our proposed solution employing CNN, LSTM, BiGRU, and BiLSTM models was fair, we admit that there are alternate ways that might be studied in comparable settings.

- A bigger dataset, for example, can help decrease the risk of model overfitting while also improving generalization.
- Analyzing just a fraction of each phrase or sentence and classifying emotions based on that subset might be a feasible choice. While some contextual information is offered up in this technique, it can lead to more efficient training and greater performance.

Finally, our sentiment analysis and NLP research has provided us with significant knowledge in a variety of areas, such as neural network models, language pre-processing approaches, and understanding the issues involved with proper labeling and model interpretability. The lessons learned and alternative methodologies uncovered during this research trip will surely help to develop sentiment analysis and guide future initiatives on this interesting subject.

7.2 Future Work

In our future work, we will take into account the fact that, despite attaining commendable accuracy, we were unable to surmount certain limitations in our proposed solution, irrespective of its acceptable precision. During the annotation process, we discovered that certain comments were ambiguous, making it difficult to accurately distinguish between emotions such as sadness and anger, sadness and fear, and happiness and surprise. Given the importance of sentiment classification, even a mildly positive sentence could be misclassified as "joy or surprise." In addition, we found that the perspective of observers can be subjective, as sentences that appear negative to some can be annotated as angry or sad. So there is no concrete solution to this problem. A comment that might seem angry to someone can be seen as a sign of sadness to others. And the same goes for the cases of joy and surprise. This observation leads us to conclude that the availability of human resources, rather than the model itself, is primarily responsible for this limitation.

Next, we will incorporate explainability into the model to address these limitations. Our objective is to gain a deeper understanding of the model's behavior and how it processes text inputs to generate classified sentiments. By adding explainability, we expect to obtain insight into the inner workings of the model, enabling us to identify and rectify any potential flaws or biases. In spite of the fact that the primary focus of our present model is on increasing sentiment analysis for the Bangla dataset, we want to expand the model's reach in the near future. We have the option of obtaining a dataset that is bilingual, which would entail comments or feedback that are provided in both English and Bengali. It would be challenging to go through the steps of dataset cleaning, preprocessing, and model construction with comments that consist of several languages, but if we are able to accomplish this goal, it would provide us with a wider range of options to work with our own dataset in the years to come.

Our other objective is to compile specialized datasets on a variety of topics, such as the energy crisis, entertainment, business, athletics, daily difficulties, and so on. This is done with the intention that, in the event that emotion detection on a specific topic becomes essential, individuals will be able to use our dataset to do this task. This extension will not only widen the scope of our study, but it will also make it possible for our algorithm to categorize sentiment across a variety of Bangla datasets. We predict that our approach will be more applicable and effective in capturing sentiment subtleties in a wider variety. Our future work will rely on overcoming the restrictions associated with ambiguous annotations, including explainability into the model, working with bilingual dataset, and increasing the area of topics in order to improve sentiment analysis in the Bangla language.

7.3 Conclusion

Finally, we conducted a sentiment analysis on Facebook and YouTube comments written in Bangla. The procedure included numerous stages, commencing with the collection of data and concluding with a thorough cleaning. Afterward, a dataset was created, and two individuals labeled the data to ascertain ground-truth sentiments. We developed and evaluated several models, including CNN, LSTM, Bi-LSTM, BiGRU, and ensembled models, to analyze the sentiment of the comments. Although we were unable to achieve the highest level of precision, our models were able to recognize the emotions of individuals with a high degree of fairness. Despite constraints such as ambiguous annotations and discrimination difficulties between certain sentiment categories, our proposed method demonstrated remarkable accuracy in sentiment categorization. Our work incorporates explainability into our long-term aims, which is an important part of what we accomplish. By incorporating explainability into the model, we expect to gain a deeper understanding of how our algorithms categorize attitudes based on text inputs. This will increase our ability to detect and rectify imperfections, biases, and improvement opportunities.

In conclusion, despite not achieving the highest accuracy, our study was able to conduct sentiment analysis on Bangla comments using a multi-step strategy that included data collection, cleaning, annotation, and model construction. Our algorithms demonstrated a reasonable capacity for emotion recognition. Improving the precision and adaptability of our sentiment analysis solution in Bangla will necessitate the addition of the clarity and the expansion of the model's generalizability to new topics in the future.

References

[1] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/j.asej.2014.04.011.

[2] “Social Media and Human Well-being during COVID-19 Pandemic,” *J. Int. Bus. Manag.*, Mar. 2023, doi: 10.37227/JIBM-2023-01-5712.

[3] R. CBalabantaray, M. mohd, and N. Sharma, “Multi-Class Twitter Emotion Classification: A New Approach,” *Int. J. Appl. Inf. Syst.*, vol. 4, pp. 48–53, Sep. 2012, doi: 10.5120/ijais12-450651.

[4] K. Albishre, M. Albathan, and Y. Li, “Effective 20 Newsgroups Dataset Cleaning,” in *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Dec. 2015, pp. 98–101. doi: 10.1109/WI-IAT.2015.90.

[5] Md. S. Islam, R. Mahmud, and B. Ahmed, “Trust in Government during COVID-19 Pandemic in Bangladesh: An Analysis of Social Media Users’ Perception of Misinformation and Knowledge about Government Measures,” *Int. J. Public Adm.*, vol. 46, no. 8, pp. 570–586, Jun. 2023, doi: 10.1080/01900692.2021.2004605.

[6] M. Mancosu and F. Vegetti, “What You Can Scrape and What Is Right to Scrape: A Proposal for a Tool to Collect Public Facebook Data,” *Soc. Media Soc.*, vol. 6, no. 3, p. 205630512094070, Jul. 2020, doi: 10.1177/2056305120940703.

[7] N. J.-M. Blackman and J. J. Koval, “Interval estimation for Cohen’s kappa as a measure of agreement,” *Stat. Med.*, vol. 19, no. 5, pp. 723–741, 2000, doi: 10.1002/(SICI)1097-0258(20000315)19:5<723::AID-SIM379>3.0.CO;2-A.

[8] T. Dalgleish and M. Power, *Handbook of Cognition and Emotion*. John Wiley & Sons, 2000.

[9] S. Jusoh, “A STUDY ON NLP APPLICATIONS AND AMBIGUITY PROBLEMS,” . *Vol.*, no. 6, 2005.

[10] R. Berman and L. Verhoeven, “Cross-linguistic perspectives on the development of text-production abilities: Speech and writing,” *Writ. Lang. Lit.*, vol. 5, no. 1, pp. 1–43, Jan. 2002, doi: 10.1075/wll.5.1.02ber.

[11] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep Learning--based Text Classification: A Comprehensive Review,” *ACM Comput. Surv.*, vol. 54, no. 3, p. 62:1-62:40, Apr. 2021, doi: 10.1145/3439726.

[12] A. Nenkova and K. McKeown, “A Survey of Text Summarization Techniques,” in *Mining Text Data*, C. C. Aggarwal and C. Zhai, Eds., Boston, MA: Springer US, 2012, pp. 43–76. doi: 10.1007/978-1-4614-3223-4_3.

[13] R. Feldman, “Techniques and applications for sentiment analysis,” *Commun. ACM*, vol. 56, no. 4, pp. 82–89, Apr. 2013, doi: 10.1145/2436256.2436274.

[14] S. Hosain Sumit, Md. Zakir Hossan, T. Al Muntasir, and T. Sourov, “Exploring Word Embedding for Bangla Sentiment Analysis,” in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, Sep. 2018, pp. 1–5. doi: 10.1109/ICBSLP.2018.8554443.

[15] A. Metallinou and S. Narayanan, “Annotation and processing of continuous emotional attributes: Challenges and opportunities,” in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Apr. 2013, pp. 1–8. doi: 10.1109/FG.2013.6553804.

[16] R. Haque, N. Islam, M. Tasneem, and A. K. Das, “Multi-class sentiment classification on Bengali social media comments using machine learning,” *Int. J. Cogn. Comput. Eng.*, vol. 4, pp. 21–35, Jun. 2023, doi: 10.1016/j.ijcce.2023.01.001.

[17] R. Alraddadi and M. I. E.-K. Ghembaza, “Anti-Islamic Arabic Text Categorization using Text Mining and Sentiment Analysis Techniques,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, Jan. 2021, doi: 10.14569/IJACSA.2021.0120889.

[18] Y. Sun *et al.*, “Residents’ sentiments towards electricity price policy: Evidence from text mining in social media,” *Resour. Conserv. Recycl.*, vol. 160, p. 104903, Sep. 2020, doi: 10.1016/j.resconrec.2020.104903.

[19] P. Kaur and M. Edalati, “Sentiment analysis on electricity twitter posts.” arXiv, Jun. 10, 2022. doi: 10.48550/arXiv.2206.05042.

[20] M. Lui and T. Baldwin, “langid.py: An Off-the-shelf Language Identification Tool,” in *Proceedings of the ACL 2012 System Demonstrations*, Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 25–30. Accessed: Jun. 06, 2023. [Online]. Available: https://aclanthology.org/P12-3005

[21] Y. Yu, X. Si, C. Hu, and J. Zhang, “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: 10.1162/neco_a_01199.

[22] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” *Phys. Nonlinear Phenom.*, vol. 404, p. 132306, Mar. 2020, doi: 10.1016/j.physd.2019.132306.

[23] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF Models for Sequence Tagging.” arXiv, Aug. 09, 2015. doi: 10.48550/arXiv.1508.01991.

[24] T. Chen, R. Xu, Y. He, and X. Wang, “Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN,” *Expert Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017, doi: 10.1016/j.eswa.2016.10.065.

[25] W. Yin, K. Kann, M. Yu, and H. Schütze, “Comparative Study of CNN and RNN for Natural Language Processing.” arXiv, Feb. 07, 2017. doi: 10.48550/arXiv.1702.01923.

[26] A. F. Agarap, “Deep Learning using Rectified Linear Units (ReLU).” arXiv, Feb. 07, 2019. doi: 10.48550/arXiv.1803.08375.

[27] “A Combined CNN and LSTM Model for Arabic Sentiment Analysis | SpringerLink.” https://link.springer.com/chapter/10.1007/978-3-319-99740-7_12 (accessed Jun. 06, 2023).

[28] H.-C. Shin *et al.*, “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning,” *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1285–1298, May 2016, doi: 10.1109/TMI.2016.2528162.

[29] M. K. Dahouda and I. Joe, “A Deep-Learned Embedding Technique for Categorical Features Encoding,” *IEEE Access*, vol. 9, pp. 114381–114391, 2021, doi: 10.1109/ACCESS.2021.3104357.

[30] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors.” arXiv, Jul. 03, 2012. doi: 10.48550/arXiv.1207.0580.

[31] I. Kouretas and V. Paliouras, “Simplified Hardware Implementation of the Softmax Activation Function,” in *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, May 2019, pp. 1–4. doi: 10.1109/MOCAST.2019.8741677.

[32] “Attention-based BiGRU-CNN for Chinese question classification | SpringerLink.” https://link.springer.com/article/10.1007/s12652-019-01344-9 (accessed Jun. 06, 2023).

[33] X. Liu, L. Wang, D. F. Wong, L. Ding, L. S. Chao, and Z. Tu, “Understanding and Improving Encoder Layer Fusion in Sequence-to-Sequence Learning.” arXiv, Mar. 18, 2021. doi: 10.48550/arXiv.2012.14768.

[34] N. Ketkar, “Introduction to Keras,” in *Deep Learning with Python: A Hands-on Introduction*, N. Ketkar, Ed., Berkeley, CA: Apress, 2017, pp. 97–111. doi: 10.1007/978-1-4842-2766-4_7.

[35] S. Mannor, D. Peleg, and R. Rubinstein, “The cross entropy method for classification,” in *Proceedings of the 22nd international conference on Machine learning*, in ICML ’05. New York, NY, USA: Association for Computing Machinery, Aug. 2005, pp. 561–568. doi: 10.1145/1102351.1102422.

[36] P. Semberecki and H. Maciejewski, “Deep learning methods for subject text classification of articles,” in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sep. 2017, pp. 357–360. doi: 10.15439/2017F414.

[37] S. Sharma, S. Sharma, and A. Athaiya, “ACTIVATION FUNCTIONS IN NEURAL NETWORKS,” *Int. J. Eng. Appl. Sci. Technol.*, vol. 04, no. 12, pp. 310–316, May 2020, doi: 10.33564/IJEAST.2020.v04i12.054.

[38] Z. Zhang, “Improved Adam Optimizer for Deep Neural Networks,” in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, Jun. 2018, pp. 1–2. doi: 10.1109/IWQoS.2018.8624183.

[39] “The relationship between Recall and Precision - Buckland - 1994 - Journal of the American Society for Information Science - Wiley Online Library.” https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1097-4571(199401)45:1%3C12::AID-ASI2%3E3.0.CO;2-L (accessed Jun. 06, 2023).

[40] C. Goutte and E. Gaussier, “A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation,” in *Advances in Information Retrieval*, D. E. Losada and J. M. Fernández-Luna, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 345–359. doi: 10.1007/978-3-540-31865-1_25.