AUTOMATIC EMOTION DETECTION FROM BANGLA TEXT USING DEEP LEARNING APPROACH

BY

Anika Tafannum

ID: 191-15-2577

AND

Ismatara Nodi

ID: 191-15-2673

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Ms. Tania Khatun

Assistant Professor

Department of CSE

Daffodil International University

Co-Supervised By

Ms. Tania Khatun

Assistant Professor

Department of CSE

Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

January 2023

APPROVAL

This Project titled **"Automatic Emotion Detection from Bangla Text using Deep Learning Approach"**, submitted by Anika Tafannum, ID No: 191-15-2577. Ismatara Nodi, ID No: 191-15-2673 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on *30/01/2023*.

BOARD OF EXAMINERS

4/2/23

Dr. Touhid Bhuiyan — **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Mohammad Shamsul Arefin — **Internal Examiner**
Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Ms. Sharmin Akter — **Internal Examiner**
Lecturer (Senior Scale)
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Mohammad Shorif Uddin — **External Examiner**
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Ms. Tania Khatun, Assistant Professor, Department of CSE**, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Tania Khatun

Ms. Tania Khatun
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Tania Khatun

Ms. Tania Khatun
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Anika

Anika Tafannum
ID: 191-15-2577
Department of CSE
Daffodil International University

Nodi

Ismatara Nodi
ID: 191-15-2673
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We are really grateful and wish our profound our indebtedness to **Ms. Tania Khatun**, **Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Deep Learning"* to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to Ms. Tania Khatun, and Head, Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Emotion recognition involves identifying and interpreting the emotions of an individual, which can be conveyed through facial expressions, verbal communication, or written text. Bengali, a low-resource language, has seen a surge in the amount of written text data available in recent years, making the task of emotion classification in Bengali text increasingly important for a range of applications including sports, e-commerce, entertainment, and security. However, the lack of appropriate language processing tools and benchmark corpora makes emotion classification in Bengali text a challenging task. In this study, we propose a deep learning approach for classifying Bengali text data into one of six basic emotion categories: anger, fear, disgust, sadness, joy, and surprise. To this end, we develop a Bengali emotion corpus comprising 29,846 sentences and 40,718 unique words. We also explore various word embedding techniques, including Word2Vec, FastText, and the Keras Embedding Layer, to find the most effective features for Bengali text emotion classification. We then evaluate several machine learning and deep learning models, including MNB, LR, SVM, CNN, LSTM, and the proposed Keras Embedding+CNN, on the corpus using different feature extraction and word embedding techniques. The results demonstrate that the CNN-based method with Keras Embedding word embedding achieves the highest accuracy of 74.40% on the test dataset.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Introduction

Emotion detection is the computational approach to identifying and understanding the emotions of an individual through text, audio, video, heart rate, or blood pressure. This field has numerous applications, including customer reviews and feedback, recommendations, advertising, social media, and healthcare, as well as the use of intelligent chatbots and agents in research. With the proliferation of online data and advances in computational processes, emotion detection research has been conducted in a variety of languages, including English, Arabic, and French.

The objective of this thesis is to present a deep learning approach for categorizing Bangla textual data into six primary emotion categories: anger, fear, disgust, joy, sadness, and surprise. This introduction offers a summary of the emotion classification framework and addresses the challenges, applications, motivations, and contributions of the thesis. It also outlines the structure of the rest of the paper.

1.2 Emotion Classification and Textual Emotion Classification

Emotion classification in the text refers to the task of automatically assigning a text document to one of a predetermined set of emotion categories. For example, the text "ছোট বেলা থেকেই অনাথ ছেলেটি নিজে নিজে বাঁচতে শিখে গিয়েছে।" (translated as "The orphaned boy has been learning to take care of himself from a young age") may be classified as expressing sadness. Emotion classification is a type of text classification, which involves assigning texts to categories based on their content.

1.3 The Fundamentals of Emotion Classification

The main goal of this work is to create a deep learning-based system that can detect human emotions from text. The abstract schematic diagram of the system is shown in Figure 1.1. The emotion detection system will be developed through several steps, which are discussed in detail in Chapter 3.

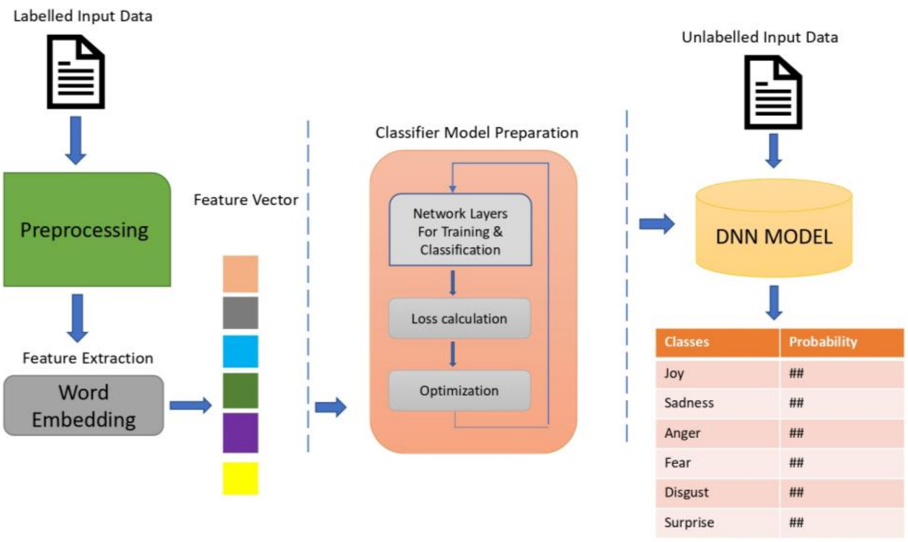

Figure 1.1: Overview of the emotion classification

1.4 Difficulties and Challenges

Implementing this system presented several challenges that had to be overcome. Some of these difficulties included:

- Developing a suitable dataset for the learning algorithm. Bangla is a low-resource language, and there were no high-quality corpora available for emotion classification research. To overcome this, we collected a large amount of text from various Bangla sources and social media, resulting in a dataset of 29,846 emotional texts, which took approximately six months to compile.
- Properly labeling the collected data into the appropriate emotion categories. Using incorrect data to train the classifier can significantly decrease its performance. We used semantic features, which represent the basic conceptual components of meaning for any lexical item, to accurately classify the text.

- Ensuring sufficient hardware support for processing large amounts of text. To achieve better accuracy, it was necessary to process a significant amount of data, so we set up hardware with high computational power to handle the workload.

1.5 Applications

The ability to interpret emotions and sentiments expressed in online content is becoming increasingly important as the number of users on virtual platforms continues to grow and generate more online content at a rapid pace. This information can be useful for a variety of parties, including consumers, businesses, leaders, and other interested parties. For example, understanding the emotions of customers can help businesses tailor their products and services to better meet their needs, while leaders can use emotion analysis to gauge public sentiment and make informed decisions.

Emotion classification also has numerous applications, including business, politics, education, healthcare, and entertainment. For instance, chatbots and intelligent agents can be equipped with emotion recognition systems to improve social interactions and provide more personalized experiences. In the healthcare field, analyzing users' emotional behavior through text can help identify potential suicidal thoughts and take appropriate action. Similarly, analyzing criminal activity through text can assist in detecting and preventing criminal activity.

1.6 Motivation

There are several motivating factors behind this work on emotion detection in Bangla text. Most research in this area has been conducted in English and other languages, while Bangla text is becoming increasingly prevalent on the internet. While there have been some efforts toward Bangla sentiment analysis, these only classify sentences as positive, negative, or neutral. However, there are six basic human emotions - happiness, fear, anger, sadness, surprise, and disgust [1] - and our goal is to develop a deep learning method that can categorize these emotions from Bangla text. Although there have been a few previous attempts at emotion detection in Bangla text using deep learning, the results were not satisfactory. [2] Bangla is spoken by over 210 million people as a first or second language, and a large volume of Bangla text data is generated online on a daily basis, much of which conveys emotion. Additionally, emotion detection in Bangla is a relatively new area of research in Bangla language processing. These factors motivated us to design an emotion detection system that can automatically detect emotions from Bangla text.

1.7 Rationale of the Study

The field of emotion detection has seen significant growth in recent years, with numerous applications in areas such as customer feedback, recommendations, advertising, social media, and healthcare. However, much of the research in this field has been conducted in languages such as English, Arabic, and French. There is a scarcity of research in Bengali, which is spoken by over 300 million people worldwide.

The objective of this thesis is to present a deep learning approach for categorizing Bengali textual data into six primary emotion categories: anger, fear, disgust, joy, sadness, and surprise. This study aims to fill the gap in the literature by developing an emotion recognition system for Bengali texts, utilizing a corpus of 29.29k texts tagged with six basic emotion classes.

The proposed method was tested using different feature embedding, machine learning, and deep learning techniques. Our research revealed that utilizing Keras Embedding for feature embedding and a CNN network to train the model produced the best results, with an accuracy of 74.40%. This study contributes to the field of emotion detection by providing a method for detecting emotions in Bengali texts, and highlights the potential for future research in this area.

1.8 Research Questions

Many different questions could be asked in several ways about this study. To make this study more compact, a set of questions was taken from several persons.

What are the current state-of-the-art methods for emotion detection in Bengali text?

This thesis aims to present a deep learning approach for categorizing Bengali textual data into six primary emotion categories: anger, fear, disgust, joy, sadness, and surprise.

The proposed method utilizes a combination of feature embedding, machine learning, and deep learning techniques to classify emotions in Bengali text.

What are the challenges and limitations of emotion detection in Bengali text?

The challenges and limitations of emotion detection in Bengali text include the lack of annotated datasets, the complexity of the Bengali language, and the similarity among various emotion classes. Additionally, the imbalanced distribution of data among different emotion classes can also affect the performance of the model.

How does the proposed method compare to other existing methods for emotion detection in Bengali text?

The proposed method was compared to other existing methods for emotion detection in Bengali text, including different embedding techniques such as Word2Vec and Fasttext. The results indicate that the proposed method, which uses Keras Embedding with CNN, achieved the highest accuracy of 74.40%.

1.9 Expected Output

The expected output of this study is the development of an automatic emotion detection system for Bengali text using a deep learning approach. The proposed system is expected to accurately classify Bengali text into six primary emotion categories: anger, fear, disgust, joy, sadness, and surprise. The system will utilize Keras Embedding for feature embedding and a CNN network for training the model. The performance of the system will be evaluated using several measures, including accuracy, precision, recall, and F1-score. Based on the results of the experiments, it is expected that the system will achieve an accuracy of at least 70% on the developed dataset. Additionally, the system will provide valuable insights for future research in the field of emotion detection in Bengali text.

1.10 Report Layout

It is divided into six distinct segments for making this research report more manageable and effective for readers and researchers.

Chapter 1: Introduction

In this chapter, the background, objectives, and scope of the research will be discussed. The introduction will provide an overview of the problem statement, research questions, and the significance of the study. It will also include the research design, data collection methods, and data analysis techniques used in the study.

Chapter 2: Literature Review

This chapter will provide an overview of the existing literature on emotion detection from Bengali text using deep learning approaches. The literature review will cover the current state of the art in the field, highlighting the main challenges and opportunities for further research. It will also present an overview of the methods, techniques, and tools used in the literature, and their strengths and limitations.

Chapter 3: Methodology

This chapter will describe the research methodology used in the study, including the research design, data collection methods, and data analysis techniques. It will also provide details on the data pre-processing, feature extraction, and model selection methods used. The chapter will also include a description of the experimental setup and the evaluation metrics used to assess the performance of the proposed model.

Chapter 4: Implementation and Impact on Society, Environment and Sustainability

In this chapter, the authors will describe the implementation of the proposed system and its impact on society, environment and sustainability. The authors will also discuss the potential applications of the proposed system in various domains, including customer reviews and feedback, recommendations, advertising, social media, and healthcare.

Chapter 5: Results and Discussions

This chapter will present the results of the experiments and the discussions of the results. It will include tables, figures, and graphs to help the readers understand the results. The chapter will also present a comparison of the proposed system with other existing methods, and the discussion of the limitations and future directions of the proposed system.

Chapter 6: Summary, Conclusion, Recommendation and Implication for Future Research

This chapter will summarize the main findings of the research and draw conclusions based on the results. It will also provide recommendations for future research, including areas for improvement and new directions for research. Additionally, the chapter will discuss the implications of the research for practitioners and researchers in the field.

1.11 Contribution of the thesis

The primary objective of this work is to classify Bangla text into one of six basic emotion categories: Anger, Fear, Disgust, Joy, Sadness, and Surprise. To achieve this, we made the following contributions:

1. Developed a corpus of 29,846 Bangla text documents with 40,718 unique words to be used for emotion classification.
2. Investigated various word feature extraction and embedding techniques, including Word2Vec, FastText, and the Keras Embedding Layer, with hyperparameter tuning for Bangla textual emotion classification.
3. Created a deep learning-based framework using Word2Vec embedding and BiLSTM network for classifying textual emotions in Bangla.
4. Evaluated and compared the performance of the proposed model with other machine learning baselines and existing techniques.

1.12 Conclusion

In this study, we set out to propose a deep learning method for emotion classification in Bangla texts. We first identified the difficulties of implementing such a system, including the lack of available datasets and the challenge of properly annotating collected data. To tackle these issues, we developed a corpus of 29,846 Bangla sentences and examined various word embedding techniques to identify suitable features for classification. We then assessed the performance of a number of machine learning and deep learning models, including MNB, LR, SVM, CNN, and LSTM, on our corpus using different feature extraction and word embedding techniques. The results showed that our proposed method using Keras embedding and CNN outperformed other techniques, achieving an accuracy of 74.40% on the test dataset. In conclusion, this work presents a promising approach for emotion classification in Bangla texts and demonstrates the potential of deep learning in this task.

CHAPTER 2

Literature Review

2.1 Introduction

In this chapter, we will provide an overview of the field of emotion classification in text and discuss the related terminology and techniques. We will also highlight the challenges and limitations of current approaches and how they relate to the work presented in this thesis. Emotion classification, also known as sentiment analysis, is a subfield of natural language processing that involves identifying and categorizing the emotional content of a text. It has a wide range of applications, including customer feedback analysis, marketing research, and social media analysis. There are various methods for emotion classification, including rule-based approaches, machine learning algorithms, and deep learning methods. Each method has its own strengths and limitations, and the choice of approach depends on the specific requirements of the task. In the following sections, we will review the relevant literature and discuss the challenges and limitations of current approaches.

2.2 Preliminaries/Terminology

Understanding key terms and terminology is essential for understanding the concept of text-based emotion classification. In this chapter, we will explore various terms related to this task, including text classification, opinion mining, basic emotion classes, sentiment analysis, implicit emotions, and explicit emotions.

- Text Classification: The process of assigning a given text to a predefined set of categories or classes using natural language processing and computational linguistics.
- Opinion Mining: A subfield of text classification that involves extracting and identifying the opinions expressed in a given text.
- Basic Emotion Classes: Six primary emotions recognized by psychology, including anger, disgust, fear, joy, sadness, and surprise.

- Sentiment Analysis: A technique for determining the sentiment or emotion expressed in a piece of text, often used to determine whether the text is positive, negative, or neutral.
- Implicit Emotion: An emotion that is not overtly expressed in a piece of text, but may still be inferred by the language used.
- Explicit Emotion: An emotion that is clearly expressed in a piece of text, typically through the use of specific words or phrases.

2.3 Feature Extraction Methods for Emotion Ana-lysis

Feature extraction refers to the process of selecting and extracting relevant information from a larger dataset. This is an important step in the classification process as it allows for the selection of the most important and relevant data for the task at hand. There are various techniques that can be used for feature extraction, including both contextual and non-contextual methods. Contextual techniques utilize the context in which the data appears, while non-contextual techniques focus solely on the data itself. Some popular methods for feature extraction include syntactic and semantic techniques, which focus on the structure and meaning of the data, respectively.

Tf-Idf Vectorizer

Term Frequency-Inverse Document Frequency (tf-idf) is a numerical statistic that is frequently used in natural language processing tasks. It calculates the significance of a particular word in a document relative to its frequency in a larger dataset. Tf-idf increases when a word appears frequently in a document but has a low frequency in the entire dataset. This can be calculated using the following formula: [3]

$$tf - idf(t,d) = tf(t,d) * idf(t) \quad (2.1)$$

where "tf" is the term frequency, or the number of occurrences of term t in document d:

$$tf(t,d) = t{:}t \in d \quad (2.2)$$

and "idf" is the inverse document frequency, calculated as:

$$idf(t) = log(\frac{n}{df(t)}) + 1 \tag{2.3}$$

Here, n is the total number of documents in the dataset and df(t) is the document frequency of t, or the number of documents in the dataset that contain term t.

Word2Vec

Word2Vec is a well-known word embedding model that uses neural networks to determine the semantic similarity of words based on their context. The model has two main architectures: skip-gram and continuous bag of words (CBOW). Skip-gram is an unsupervised learning architecture that aims to find semantically similar words based on the context of a given word. It does this by maximizing the average logarithmic probability, as given by equation 2.4 [4].

$$-\frac{1}{V}\sum_{v=1}^{v} \quad \sum_{-c\leq m\leq c, m\neq 0} \quad log[p(w_{v+m}|w_v)] \tag{2.4}$$

where V is the vocabulary list, c is the size of the context (also called the window size), and p(wn+mlwn) is the probability of a word wn+m given the context word wn. This probability can be calculated using equation 2.5:

$$p(i|o) = \frac{exp(u_i^T.u'_0)}{\Sigma_{v\epsilon V}\, exp\,(u_z^T.u'_0)} \tag{2.5}$$

where u and u' are "input" and "output" vector representations of i and o, respectively.

On the other hand, CBOW is an architecture that predicts a target word based on the context of surrounding words. It uses continuous distributed representations of the context to do so, and utilizes a log-linear classifier to maximize equation 2.6 [5]:

$$\frac{1}{V}\sum_{v\epsilon V} \quad log(p(w_{v-c},\ldots w_{v-2}, w_{v-1}, w_{v+1}, w_{v+2}\ldots w_{v+c})) \tag{2.6}$$

where V and c have the same definitions as in the skip-gram model. [6]

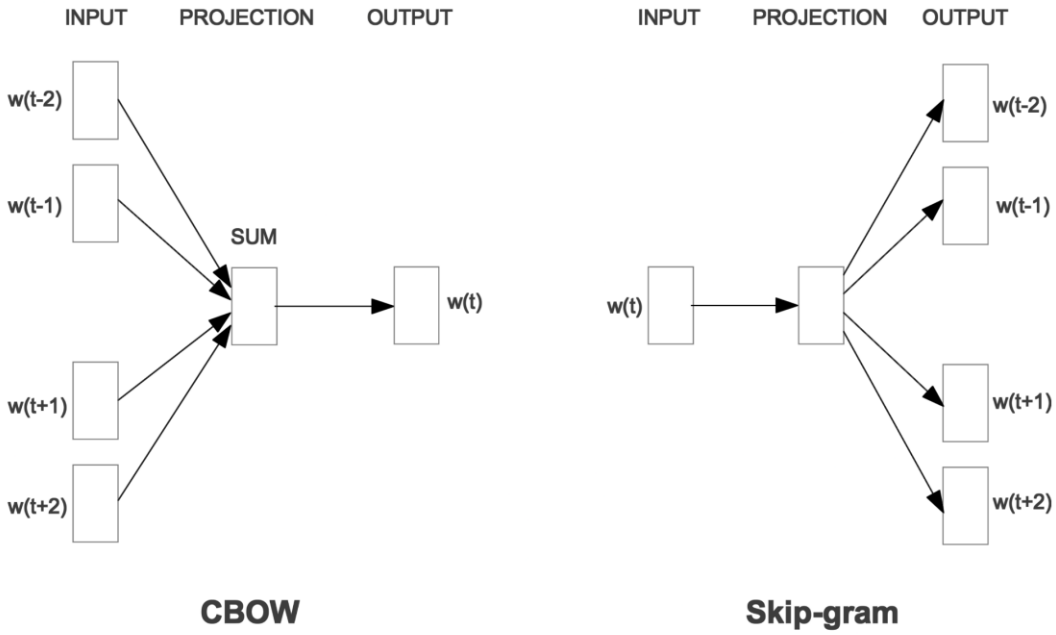

Figure 2.1: Skip-gram and CBOW architecture

Glove

The Glove model learns word embeddings by constructing a co-occurrence matrix [7], where the rows and columns represent the word vocabulary and each entry C_{ij} represents the co-occurrence weight of words i and j. A higher weight indicates a higher similarity in the resulting vectors. The model then optimizes a loss function based on this co-occurrence matrix, resulting in word vectors that capture the semantics of the vocabulary. The co-occurrence weight can be calculated in various ways, such as the dot product of the word vectors or the log of the number of times the words co-occur in a text. The GloVe model has been effective in numerous natural language processing tasks, particularly when combined with other techniques.

FastText

FastText is a word embedding technique that utilizes subword information to learn representations of words [8]. This approach allows the model to construct embeddings for words that were not present in the vocabulary during training. This is an advantage over other methods such as word2vec and GloVe, which are unable to generate embeddings for out-of-vocabulary words. FastText uses character n-grams to learn the embeddings, making it a robust method for handling rare or unfamiliar words.

2.4 Comparative Analysis and Summary

Machine learning (ML) techniques have been widely used for the classification task, including the analysis of textual emotion classification in various languages. In the following sections, we will discuss some popular ML techniques.

Naive Bayes Classifier

Naive Bayes is a probabilistic classifier that is widely used in machine learning [9]. It is based on Bayesian reasoning and probability inference to predict the target class. In order to process large datasets, the multinomial model of Naive Bayes is used. The performance of Naive Bayes can be improved by searching for dependencies among attributes. The ease of computation makes it a popular choice for data preprocessing applications. Attribute weight values are important for improving the model's efficiency and deep feature weighting can help address the conditional independence assumption, allowing for the reliable computation of conditional probability. However, these feature weighting strategies can have drawbacks such as insufficient performance enhancement, compromised simplicity, and increased model execution time.

Accurate prediction of the conditional probability terms is crucial for the performance of Naive Bayes. It can be difficult to accurately predict these terms when the training data is scattered. Meta-learning techniques such as structure extension, attribute collection, frequency transforming, attribute weighting, instance weighting, and local learning can be used to estimate the conditional probability terms. Naive Bayes has been used for tasks such as classifying emails as spam or ham, classifying articles based on content, and analyzing sentiment/emotion. It can also be used to classify texts into suspicious and non-suspicious categories. The advantages of Naive Bayesian classifiers include their ease of use, efficiency in terms of degree of certainty, ease of optimization, and potential for dynamic adaptation, making them a useful tool for natural language processing tasks.

One limitation of Bayesian networks is that their time complexity increases when processing high dimensional text data. In addition, Bayesian networks do not allow for interaction between features and the probabilities calculated are not accurate, but rather relative probabilities.

Support Vector Machine

The Support Vector Machine (SVM) algorithm [10] is a type of supervised machine learning algorithm that can be used to solve a variety of classification problems, including credit risk analysis, medical diagnosis, text categorization, and knowledge extraction. SVM is particularly effective in high-dimensional feature spaces and is able to work with non-linearly separable data using the kernel trick. It separates classes by finding an optimal hyperplane between them and uses only a subset of the training set, called support vectors, to create decision boundaries. SVM can also be customized using kernels, such as the Class Meaning Kernel which smooths words in documents using class-based meaning values. The performance of SVM is not affected by sparsity of data and it can handle large datasets efficiently. However, it is not as effective in handling unlabeled data and selecting the best kernel for training data can be time-consuming. In addition, SVM is a non-parametric model, meaning it cannot summarize data based on underlying parameters, and training and testing using an SVM model can be time-consuming. In order to improve the performance of SVM classifiers, it is important to ensure that the data has been thoroughly preprocessed. To reduce the number of features, the parameter max-features can be used to select the most frequent 1000 features in the learning model.

Logistic Regression

Logistic regression [11] is a supervised machine learning algorithm commonly used for binary classification tasks, where the goal is to predict a binary outcome (e.g., classifying emails as spam or not spam). It uses a logistic function to model the probability of a certain class based on the values of the input features. The model is trained using maximum likelihood estimation, which seeks to find the coefficients that minimize the difference between the predicted probabilities and the actual class values in the training data.

The performance of logistic regression can be influenced by the choice of cost function and the presence of overfitting or underfitting. Regularization techniques may be used to prevent overfitting. Logistic regression has been applied in various areas such as financial forecasting, software cost prediction, software effort prediction, software quality assurance, and crime data mining. However, it may struggle to separate non-linearly separable classes and may require a large sample size for accurate predictions.

2.5 Deep Learning Algorithms

Deep learning (DL) techniques have gained widespread popularity in recent years for their ability to extract contextual features in classification tasks, including emotion classification. In this section, we will provide an overview of some popular DL techniques used in emotion classification.

Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are a type of deep learning algorithm commonly used in computer vision tasks. They can also be applied to natural language processing tasks, where they can be used to identify patterns in text by performing convolutions on the input data. In text classification, CNNs can be used to detect patterns of different sizes, such as phrases or word n-grams, regardless of their position in the input text. When a specific pattern is detected, the output of the convolution operation will be activated, allowing the model to recognize the pattern in a sentence. CNNs are able to achieve this by adjusting the size of the kernels used in the convolution operation and concatenating the outputs.

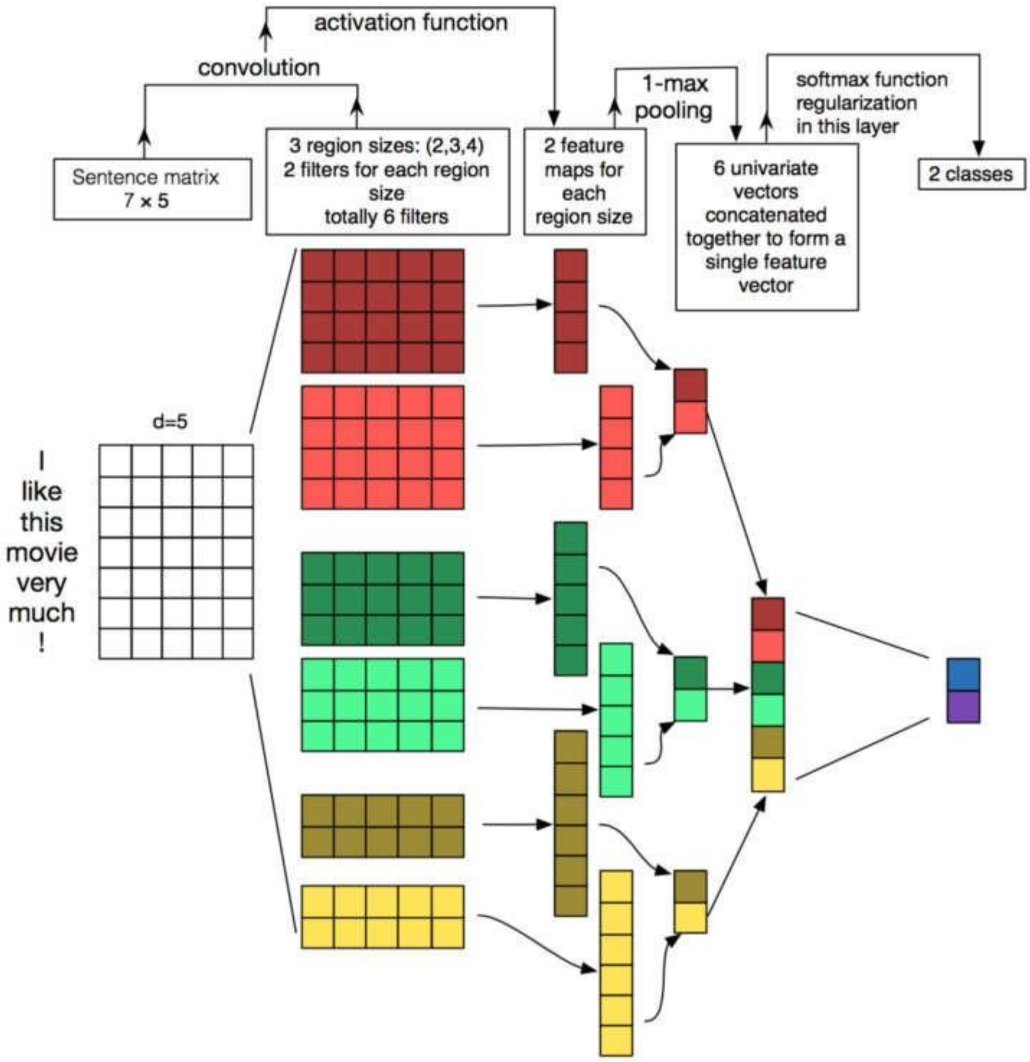

Figure 2.2: CNN blocks for textual data

Bidirectional Long-Short Term Memory (Bi-LSTM)

A bidirectional Long Short-Term Memory (biLSTM) is a type of deep learning model that is used for sequence processing tasks. It consists of two LSTMs, one of which processes the input sequence in a forward direction and the other processes it in a backward direction. This allows the network to have access to both past and future context, which can be beneficial in understanding the overall context of a sequence. BiLSTMs are often used in natural language processing tasks such as language translation and emotion classification. They have been found to be effective at capturing long-range dependencies in sequences and handling variable-length input. Figure 2.32 illustrates an overview of a typical biLSTM network.

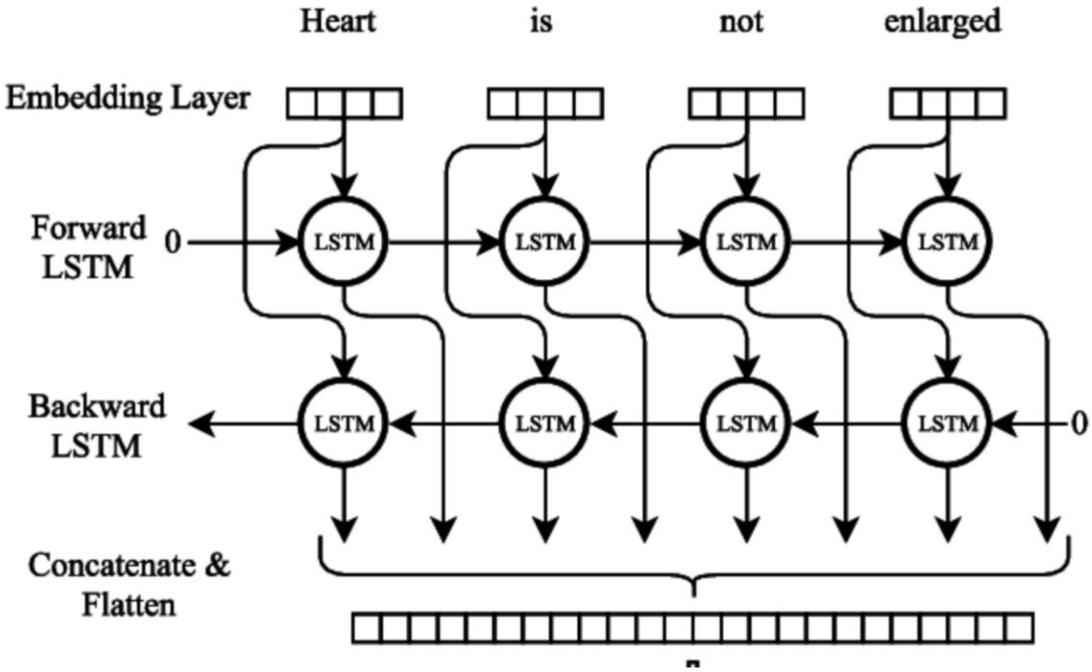

Figure 2.3: BiLSTM network for textual data

2.6 Related Work

Emotion classification has made significant progress for languages with many resources, such as English, Chinese, Arabic, and Spanish. However, there have been few research activities conducted in the Bengali language. This review presents an overview of emotion classification in languages other than Bengali, as well as in Bengali specifically.

Non-Bengali Language based Emotion Classification

Most language processing research has focused on the English language, as standard datasets are typically available in English. However, there is a lack of standard collections of data for Bangla, such as the IMDB dataset [12], restaurant dataset [13], movie review dataset [14], and semEval [15]. There has been significant progress in emotion classification for English, Hindi, Arabic, and Chinese languages [16]. A tool called EmoTxt was developed using machine learning algorithms for English based on questions and answers from online forums [17]. A multilabel, multi-target emotion detection system for Arabic tweets was developed using decision trees, random forests, and K nearest neighbors, with random forests achieving an F1 score of 82.6% [18].

Hasan et al. developed an automatic emotion detection system for Twitter tweets using a supervised machine learning algorithm, achieving 90% accuracy with only four emotion classes [19]. Several deep learning approaches have also been used for emotion classification of short sentences. Lai et al. proposed a graph convolution network architecture for emotion classification of Chinese microblogs, achieving an F-measure of 82.32% [20]. Haryadi et al. used a nested long short-term memory (LSTM) to classify seven emotion classes (anger, fear, joy, love, sadness, surprise, and thankfulness) and achieved 99.167% accuracy [21], demonstrating that nested LSTM performs better than LSTM at finding semantic relationships between words in a sentence. The SemEval-2019 Task 3 [22] proposed a Bi-LSTM architecture for emotion classification, which is a common choice for neural network architecture in text classification tasks. They considered four classes and achieved an F1 score of 79.59.

Bengali Language based Emotion Classification

There have been several studies on emotion classification in the Bengali language, but many of them have focused on only a few emotion classes and have used machine learning techniques. In [2], a model was developed for three-label sentiment, five-label sentiment, and emotion detection using only YouTube comments in Bengali, English, and Romanized Bengali. The model used Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) for comparison, and LSTM performed better. In [23], a study on three emotional labels (happy, sadness, and anger) was conducted using a dataset developed by [24] and statistical machine learning methods. Multinomial Naive Bayes (MNB) outperformed other methods with an accuracy of 78.6%. In [25], a new corpus called "Anubhuti" was developed for the domain of Bengali short stories and labeled with four emotion classes: joy, anger, sorrow, and suspense. A deep learning approach was used, but it achieved lower accuracy than a logistic regression model. In [26], a study on Bengali blog texts used a Conditional Random Field (CRF) to detect emotional expression and achieved an accuracy of 56.45%. In [27], emotion classification for Ekman's six basic emotions was conducted on Bengali blog and news texts at the word and sentence level using the Bangla WordNet Affect list developed in [28]. The study achieved an F-score of 62.1%. In [29], 1200 Bengali documents from various domains were classified into six basic emotions using Support Vector Machines

(SVMs) and achieved an accuracy of 73%. Table 5.5 in the original content summarizes the previous works on emotion classification in Bengali text and their limitations.

Table 2.1: Summary of Previous Works in Bengali Emotion Classification

Method	**Emotion Class**	**Dataset**	**Limitation**
Word2Vec + LSTM[2]	4	1006	They've worked only with youtube comments and got 65.97% accuracy
Tfidf with Multinomial Naive Bias[23]	3	3100	Only 3 class with 78 %accuracy
Tfidf + LR[25]	4	159	Considered only our class with 73% ac- curacy
Heuristic Fea- tures+Conditional Random Field[26]	6	2350	Only 56.45% accuracy
Bag of Words+SVM[29]	6	1200	Achieved 60% accuracy only

To address the limitations of previous works on emotion classification in Bengali texts, we developed a benchmark dataset called BEmoD and applied advanced deep learning techniques with proper hyperparameter optimization to achieve the best possible results.

2.7 Scope of the Problem

The scope of the problem is the lack of an emotion detection model for Bengali texts. This lack of a model presents a significant challenge for the automatic analysis of Bengali text, particularly in areas such as customer feedback. The development of an emotion detection model for Bengali texts would greatly aid in the analysis and understanding of this language and could have a significant impact on various industries and fields. Furthermore, it will have a positive impact on society, the environment, and sustainability as well. Therefore, the development of an emotion detection model for Bengali texts is an important and necessary research endeavor.

2.8 Implementation Challenges

- The most difficult aspect of implementing this system was creating a dataset that could be used by the learning algorithm. It took about six months to collect and prepare the dataset of 29,846 emotion texts.
- Ensuring the accuracy of the collected data by correctly labeling it into different emotion categories was another challenge. If the classifier is trained on incorrect data, its performance will suffer.
- Adequate hardware support was also a significant challenge for our system.

2.9 Conclusion

In this chapter, we provide an overview of the existing literature on emotion classification using machine learning and deep learning techniques. We also highlight the limitations of previous approaches and the challenges faced by researchers while implementing these systems. We then present the methodology used in our system, which includes the development of a benchmark Bengali emotion dataset (BEmoD) and the application of advanced deep learning techniques with proper hyperparameter optimization to achieve the best possible results.

CHAPTER 3

Methodology

3.1 Introduction

The proposed methodology for emotion classification in Bengali is presented in this chapter, including a detailed explanation of its components. The process of developing the corpus and its statistics are also described in this chapter. The training model preparation, feature extraction, and selection of optimal hyperparameters are also covered in this chapter.

3.2 Research Subject and Instrumentation

The research subject in this study is the detection of emotions from Bengali text. The dataset used in this study consists of 29,290 Bengali text samples, which were annotated with six primary emotion categories: anger, fear, disgust, joy, sadness, and surprise. These emotions were chosen based on the widely accepted basic emotion theory, which postulates that these emotions are universal and have distinct characteristics.

The instrumentation used in this study includes various feature embedding, machine learning, and deep learning techniques. In particular, the study employs Keras Embedding for feature embedding, and CNN-based RNN architectures for the deep learning model. The performance of the proposed method was evaluated using several measures, including accuracy, precision, recall, and F1-score. The dataset was divided into training, validation, and test sets to evaluate the performance of the model.

3.3 Data Collection Procedure

To develop the Bengali emotion dataset (BEmoD), we followed a process similar to [30]. Five undergraduate computer science students initially annotated the corpus, which was then further validated by an NLP expert with experience in the field. Some examples from the dataset are shown in Table 3.1.

Data Crawling or Accumulation

To create a corpus for Bengali emotion classification, data was collected from various sources including Facebook comments and posts, YouTube comments, online blog posts, novels, daily conversations, and newspapers. A total of 5700 text expressions were collected over a period of three months by five participants. While a significant portion of the data was collected from online sources, data was also generated through observations of real-life conversations. It should be noted that many Bengali speakers write their comments or posts in the form of transliterated Bengali on social media platforms. An example of this can be seen below:

Table 3.1: Samples of Data

NO.	**Data**	**Emotion Class**
1	যাক তোর চাকরীর খবর টা পেয়ে খুশি হলাম। (I am so happy to get the news of your job.)	Joy
2	ছোট বেলা থেকেই অনাথ ছেলেটি নিজে নিজে বাঁচতে শিখে গিয়েছে এখন। (The orphan boy has learned to survive on his own since childhood.)	Sadness
3	মন মেজাজ ভালো নেই, এখন কথা বলতে আসিস না। (I’m not in a good mood, don’t come to talk to me now.)	Anger
4	হারামি শালা, পেয়ে নেই তোরে একবার। (Bastard, let me find you.)	Disgust
5	এ কি?? ৩ ঘণ্টায় ১৮ বছরের ছেলের এমন অবস্থা কেমনে হলো!! (What is this?? How did such a situation happen to a 16-years old boy in 3 hours!!)	Surprise
6	হরর মুভিটা দেখার পর ৩ দিন ভয়ে ঘুম হয় নি আমার। (I didn’t sleep for 3 days after watching that horror movie.)	Fear

While collecting text data for emotion classification in Bengali, we encountered a challenge with the use of transliterated Bengali in social media and other online sources. For example, the sentence "muvita dekhe amar khub valo legeche. ei rokom movi socharacor dekha hoy na." (Bangla: মুভিটা দেখে আমার

খুব ভালো লেগেছে। এই রকম মুভি সচারাচর দেখা হয় না'[English translation: I really enjoyed watching this movie. Such movies are not commonly seen]) needs to be phonetically converted, but errors can occur during this process. For instance, in the above sentence, the word "socharacor" (English: usually) may be translated as "সছারাচর" after phonetic conversion, but the correct word should be "সচরাচর". In this case, correction is necessary because "সছারাচর" is not a valid word in the Bengali dictionary [31].

3.4 Data Preprocessing and Cleaning

Pre-processing for the collected data was carried out in two phases: manual and automatic. In the manual phase, we used the Bangla academy's accessible dictionary (AD) database [31] to correct any "typo" errors in the data. If a word in the input text did not appear in the AD, it was considered a typo and the correct version of the word was found in the AD and used to replace the typo. For example, in the text " জাহাজ এই প্রথমবারের মতো ওঠা এবং সাগরের মাঝখানে দিয়ে যাওয়ার সমগুলো ম্যাজিকাল ছিলো একদম। আহা সৌন্দর্য ♥♥♥", the words in bold were found to be typos and were corrected to "জাহাজ এই প্রথমবারের মতো উঠা এবং সাগরের মাঝখানে দিয়ে যাওয়ার সময়গুলো ম্যাজিকাল ছিল একদম আহা সৌন্দর্য ♥♥♥".

During the automatic preprocessing phase, we removed emojis and punctuation marks from the manually processed data as they can sometimes create confusion about the emotional tone of the text. To do this, we created an emoji-to-hex (E2H) dictionary using [32]. We then converted all elements of E2H to Unicode and checked them against the elements in our corpus. We also created a dictionary of punctuation marks and special symbols (PSD) and replaced any text elements that matched elements in either E2H or PSD with a blank space. All automatic preprocessing was done using a python script. After this process was applied, the example given would be transformed into "জাহাজ এই প্রথমবারের মতো উঠা এবং সাগরের মাঝখানে দিয়ে যাওয়ার সময়গুলো ম্যাজিকাল ছিল একদম আহা সৌন্দর্য"

Data Annotation

The corpus was labeled through a manual process, with two groups (Group 1 and Group 2) involved in the annotation task. Group 1 consisted of 5 postgraduate students in Computer Engineering with experience in natural language processing (NLP). Group 2 was made up of 3 academicians with extensive experience in NLP. The labels assigned by these groups were verified and a final emotion class was determined through a majority voting mechanism.

Label Verification

The labeling or annotation process was carried out by two separate groups, referred to as G1 and G2. G1 consisted of 5 postgraduate students with a background in Computer Engineering, while G2 was made up of three academicians with extensive experience in NLP. The label assigned by G1 was considered the original label, and it was only changed if the expert group (G2) disagreed. In cases of disagreement, the label was discussed and a consensus was reached. Out of the 5700 data samples, 4950 labels were accepted by both groups. The remaining 750 data samples were discussed and a final label was agreed upon for 250 of them, while the remaining 500 were excluded due to disagreement or due to the presence of neutral emotion, implicit emotion, or ill-formatted text. In total, 5200 data samples and their labels were verified and saved in a *.xlsx file.

3.5 Statistical Analysis

The corpus consists of 29,846 text documents with a total of 538,153 words. The data was collected from various sources such as Facebook, YouTube, blogs, news portals, story books, novels, and conversations. In addition to these, 23,623 texts were included from four datasets (translated into Bengali) such as Affect Data, Emolnt, Emotion Stimulus, and Hashtag Emotion Dataset. The corpus was divided into different categories, with the joy class containing the most unique words and the disgust class containing the least, as shown in Table 3.2.

Table 3.2: Data distribution in each emotion class

Class	Data	Total Words	Unique Words
anger	3470	67346	11053
disgust	3159	58705	7618
fear	3033	56434	9523
joy	7903	148511	17052
sadness	6987	129361	14988
surprise	5294	77796	14938
Total	29846	538153	75172

Data Distribution

The corpus was compiled from 29,846 text documents, totaling 538,153 words, collected from various sources such as Facebook (2770 texts), YouTube (560 texts), blogs (343 texts), news portals (210 texts), story books (745 texts), novels (568 texts), and conversations (813 texts). In addition, 23,837 texts were included from four datasets (translated into Bengali) including Affect Data1, Emolnt2, Emotion Stimulus3, and Hashtag Emotion Dataset4. The distribution of emotions in the corpus is shown in Table 3.2, with the joy class containing the most unique words and the disgust class containing the least. The majority of data (62%) was collected from online sources, with 27% coming from Facebook comments, 21% from Facebook posts, 9% from YouTube comments, 3% from online newspapers, and 5% from blogs. The remaining

38% of data was collected from offline sources, including 9% from novels, 4% from authors, and 4% from daily conversations.

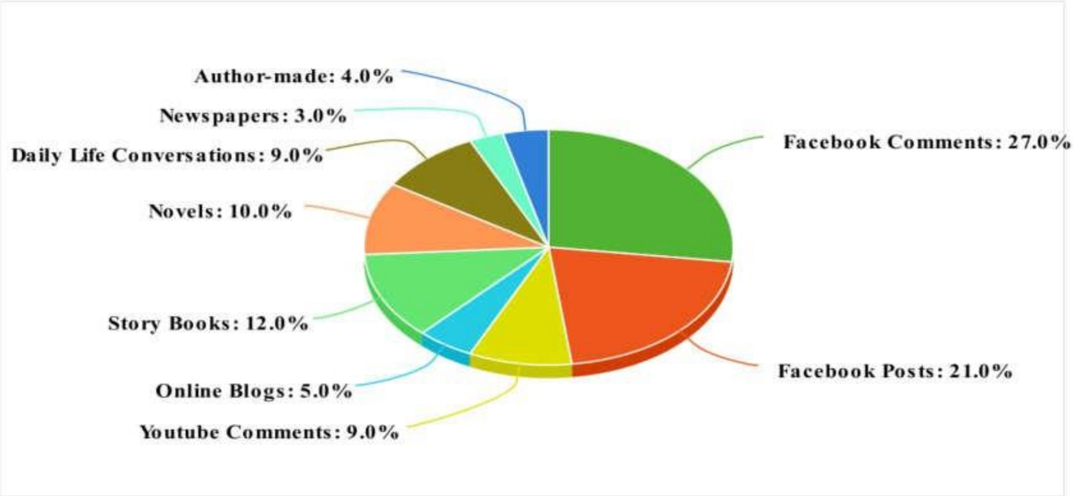

Figure 3.1: Data Collection Sources

3.6 Proposed Methodology

The goal of this project was to build a system that could accurately classify or detect the emotion present in a given Bengali text. The system we developed consisted of four main components: word embedding with Keras Embedding, text to feature mapping, training, and prediction. These components are described in more detail in the following sections. The figure below illustrates the proposed CNN emotion detection framework.

Embedding Model Preparation

- **Keras Embedding:** The Keras Embedding layer can be used to map words from a corpus to vectors of real numbers, similar to the word2vec technique. To use the Embedding layer in Keras, the corpus must first be converted into a format suitable for input into the model, such as a list of lists where each inner list represents a sentence and contains the word tokens for that sentence. The corpus can then be preprocessed to remove any non-relevant characters, such as non-Bengali alphabets, punctuation, special characters, and emoticons.

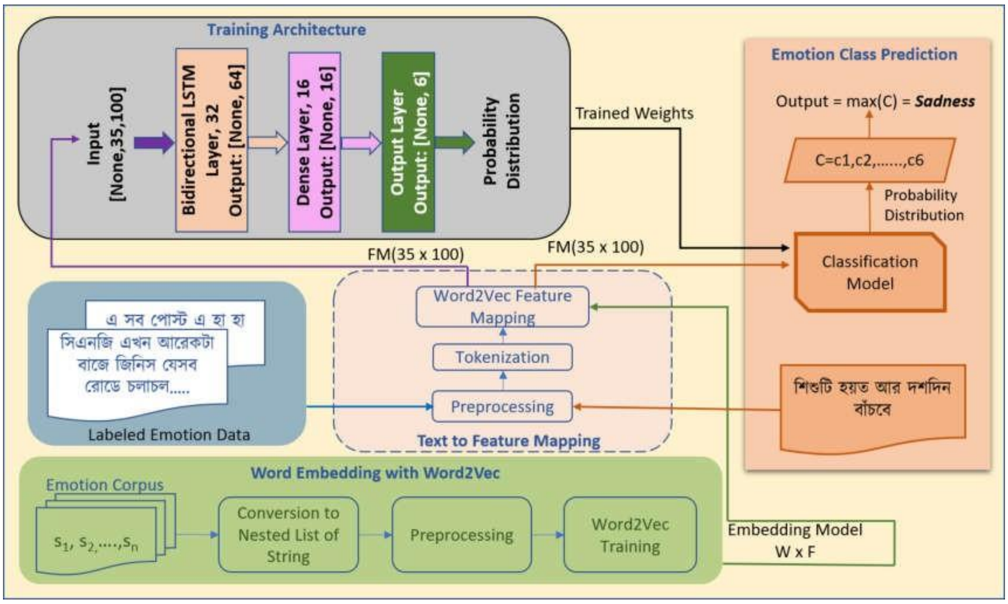

Figure 3.2: Emotion detection framework

Once the corpus has been preprocessed and formatted correctly, it can be passed as input to the Keras Embedding layer. The Embedding layer takes a 2D tensor of integer values as input, where each integer represents the index of a word in a vocabulary. It then outputs a 3D tensor with dimensions (batch size, sequence length, output dimensions). The output dimensions can be set to any integer value and represent the size of the embedding vectors output for each word.

- **FastText:** In this work, we also used the FastText method, which takes into account subword information to identify semantic relationships [8]. FastText allows for the construction of words that do not exist in the vocabulary during training by using its constituent n-grams. In our study, we trained the FastText model using Skip-Gram with character n-grams of length 5, a window size of 5, and an embedding dimension of 100. The optimal hyperparameters for both the Word2Vec and FastText embeddings can be found in Table 3.3.

Table 3.3: Optimized hyperparameters for Word2Vec and fastText Embedding

Parameters	Keras Embedding	fastText
Embedding dimension	100	100
Model	Skip-gram	Skip-gram
Minimum word count	4	3
Window size	6	5
Min n-gram	2	3
lr	0.1	0.1

Text to Feature Mapping

The labeled text data from the emotion corpus is converted into an embedding feature representation, as training cannot be performed on strings. The data is first subjected to the preprocessing step described in section 3.3.2. Then, the tokenization process splits the processed data into a list of words. The Word2Vec model is designed to extract 100 features from each data point, and the length of the data is set to 35 tokens. The Word2Vec feature mapping process takes both the embedding model and the list of tokens as input and generates a feature matrix with dimensions (35 x 100). The rows and columns of the matrix represent the number of tokens (35) and the number of extracted features (100), respectively. Text data with more than 35 tokens are truncated to 35 tokens, and data with fewer than 35 tokens are padded with zeros to reach 35 tokens.

Classifier Model Preparation

In this study, we used several machine learning (ML) and deep learning (DL) models as baselines, including the proposed Bi-LSTM classifier. The preparation and tuning of these models and their respective hyperparameters are described in the following sections.

ML Based Models

For the emotion classification task, we employed four supervised machine learning models: logistic regression (LR), support vector machine (SVM), multinomial naive Bayes (MNB), and random forest (RF).

- The logistic regression model is built using the 'lbfgs' solver and a '12' penalty. The regularization parameter 'C' is set to 2.
- The SVM model uses a 'linear' kernel with a random_state value of 0.
- The MNB model uses an 'alpha' value of 1.0.
- The RF model uses 100 'n_estimators'.
- The chosen parameters for these ML models are summarized in Table 3.4.

Table 3.4: Optimized parameters for ML models

Classifier	Parameters
LR	Optimizer = ‘lbfgs’, max_iter = 400, penalty = ‘l1’, C=1
SVM	kernel=’linear’, random_state = 0, γ= ‘scale’, tol=’0.001’
RF	criterion=’gini’, n_estimators = 100
MNB	α = 1.0, fit_prior = true, class_prior = none,

DL based Models

For the emotion classification task in Bengali text, we used two deep learning models: a Convolutional Neural Network (CNN) and a variant of Long Short-Term Memory (LSTM), called Bi-LSTM.

- **CNN:** The CNN [33] model was trained on the emotion corpus, using word embeddings from either Word2Vec or FastText as input to the embedding layer. The resulting sequence matrix was passed through a convolution layer with 64 filters of size 7, followed by a max-pooling layer. The output from the max-pooling layer was then fed into a fully connected layer with 64 neurons and a ReLU activation function. The output of the model was computed using a softmax activated output layer, which generated a probability distribution over the different classes.

- **Bi-LSTM:** The Bi-LSTM [34] model utilizes an embedding layer and a Bi-LSTM layer with 32 hidden units. It also includes a fully connected layer with 16 neurons that uses a ReLU activation function. The output from this layer is passed through an output layer with a softmax activation function, which is designed to capture the semantic meaning of the text and address the issue of long-term dependencies.

Table 3.5: Hyperparameters for DNN methods

Hyperparameters	Hyperparameter Space	CNN	BiLSTM
Filter Size	3,5,7,9	7	-
Pooling type	‘Max’, ‘average’	‘max’	-
Embedding Dimension	30,35,50,70,90,100 ,150,200,250, 300	100	100
Number of Units	16, 32, 64, 128, 256	64	32
Neurons in Dense Layer	16, 32, 64, 128, 256	64	16
Batch Size	16, 32, 64, 128, 256	32	32
Activation Function	‘Relu’, ‘tanh’, ‘softplus’, ‘sigmoid’	‘relu’	‘relu’
Optimizer	‘RMSprop’, ‘Adam’, ‘SGD’, ‘Adamax’	‘Adam’	‘Adam’
Learning Rate	0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001	0.001	0.001

Emotion Class Prediction

The prediction phase involves taking an unlabeled short sentence as input and predicting the appropriate emotion class. The sentence is first preprocessed and tokenized, and the resulting list of words is used to generate a feature map FM of shape (35 x 100) using the feature mapping process described in section 3.3.2. The feature map is then input to the chosen classification model, which produces a probability distribution C = {c1, c2, c3, ..., c6} for each emotion class. The probability for each class is calculated using Eq-3.1.

$$C(x)_i = \frac{exp(x_i)}{\Sigma_{j=1}^{n} exp(x_j)} \tag{3.1}$$

Here x is the input feature vector, and n denotes the number of classes. The maximum of C is estimated that steers to one of the six emotion classes, i.e., anger, fear, disgust, joy, sadness, and fear.

3.7 Conclusion

In this chapter, a technique for emotion detection from Bengali text was presented. The proposed method was tested using different feature embedding, machine learning, and deep learning techniques. The results of the experiments showed that the CNN based RNN architectures with Keras Embedding feature embedding were effective. The performance of the proposed method was analyzed and discussed in the following chapter.

CHAPTER 4
Implementation And Impact on Society, Environment and Sustainability

4.1 Introduction

The emotion classification system described in this work consists of various components that work together to classify the emotions expressed in Bengali text. The first step is preprocessing the data, which involves cleaning and formatting it to make it ready for further processing. This includes correcting typos and removing punctuation marks and emojis. Next, the feature extraction module maps the preprocessed data to numerical features using techniques such as Word2Vec and FastText. The training module then uses machine learning and deep learning algorithms such as logistic regression, support vector machines, multinomial naive Bayes, random forests, convolutional neural networks, and bidirectional long short-term memory networks to learn a model for emotion classification. Finally, the prediction module takes a short sentence as input and uses the trained model to predict the emotion class. An example input and output of the system would be a Bengali text sentence and the corresponding predicted emotion class.

4.2 System Requirement

Hardware Requirements

From input to output the system propagate the following hardware :

- Nvidia GeForce GTX 1070 GPU
- Minimum GPU RAM 8GB
- Physical memory 32GB
- Intel core i7-7700K CPU
- Solid State Drive (SSD) 256GB
- Minimum 2h backup UPS
- GPU cooler
- Monitors

Software Requirements

The following software was used to implement various parts of the system. To test the system, an IDE is needed to run the system. Not all of these software are required at the same time.

- Operating System: ubuntu 16.04, windows 10
- Python 3.7
- tensorflow-gpu 2.1.0
- keras 2.4.3
- numpy 1.12.1
- Pygments 2.2.0
- Markdown 2.6.10
- coreapi 2.3.3
- psycopg2 2.7.3.2
- gunicorn 19.7.1
- whitenoise 3.3.1
- drf-extensions 0.3.0
- spyder 3.6
- jupyter notebook
- BnPReprocessing
- livelossplot
- pandas

4.3 Keras Framework

To use Keras, a Python-based deep learning platform developed by Francois Chollet, you will need to fulfill the following requirements:

- Any operating system (Windows, Linux, or Mac)
- Python version 3.5 or higher

To install Keras, follow these steps:

1. Create a virtual environment
2. Activate the environment
3. Install the required Python libraries: Numpy, Pandas, Scikit-learn, Matplotlib, Scipy, and Seaborn
4. Finally, install Keras using pip: *pip install keras*

4.4 System Set up and Running

We set up a Python environment for deep learning using Keras and Tensorflow-gpu in Jupyter-Lab. To do this, we first installed Keras and Tensorflow-gpu in our environment. Then, we opened Jupyter-Lab and created a new Python notebook.

To use the GPU for training, we specified the Tensorflow-gpu backend for Keras. We then used the GPU to train our machine learning model. Once training was complete, we saved the trained model to our local disk for testing with test data.

It is worth noting that in order to use the GPU for training, we had to make sure that we had a GPU available and that our environment was correctly configured to use it.

Implementation Snapshot

A few input/output of the system is shown in Figure 4.1

```
In [32]: model.predict(["বড় একা আমি এই জগতে, কে ই বা কার উপকার করে দেয়?"])

Out[32]: 'Sadness'

In [26]: model.predict(["আকাশ থেকে পড়ার ভান ধরো না! তোমার চাহনির মধ্যেই সব জানা যায়! \
         তুমি ওর গায়ে কয়বার হাত দিছ?"])

Out[26]: 'Anger'

In [29]: model.predict(["বোন করোনা ভাইরাস ধরেছে আজরাইল যখন আসবে তখন কি করবেন? \
         আল্লাহকে ভয়করুন আমিন"])

Out[29]: 'Fear'
```

Figure 4.1: Sample input/output of the system

4.5 Impact Analysis

Our proposed work has the potential to have significant impacts on society, the environment, and ethics. By developing an automatic system for recognizing emotions from Bengali texts, we can potentially improve social interactions and communication, as well as better understand and address the emotional needs of individuals and communities. Additionally, our work may have environmental benefits by reducing the reliance on manual labor and paper-based processes. It is also important to consider the ethical implications of our work, including issues of privacy, bias, and accountability.

Social and Environmental Impact

The proposed system is designed to detect emotions from Bengali text. This system has the potential to be used for a variety of applications that can have a positive impact on society, such as detecting and preventing suicidal thoughts and criminal behavior from social media writings. By detecting and addressing these issues early on, we can potentially contribute to the well-being and safety of individuals and communities.

Ethical Impact

One ethical consideration of our proposed system is the potential impact on privacy. If this work is used improperly, it could potentially infringe upon the privacy of individuals. Therefore, it is important to consider and address these ethical concerns before using this work. This may include ensuring that proper consent is obtained from individuals before collecting and analyzing their text data, and establishing safeguards to protect the privacy of this data. It is also our moral obligation to consider and address these ethical issues in order to ensure that our work is used responsibly and for the benefit of society.

4.6 Conclusion

This chapter provides an overview of the methods and tools used to implement our proposed system, as well as the potential social, environmental, and ethical impacts. We also discuss the evaluation results of the proposed methodology in the next chapter. Specifically, this chapter covers the following topics:

- Implementation methods
- Software and hardware setup
- Social, environmental, and ethical impacts

By summarizing these details in this chapter, we aim to provide a clear and concise summary of the steps and considerations involved in our proposed work.

CHAPTER 5

Results and Discussions

5.1 Introduction

In this chapter, the results of our proposed system for emotion classification in Bengali text are presented and analyzed in depth. The performance of various machine learning and deep learning methods is evaluated using evaluation measures such as precision, recall, accuracy, and f1-score, and compared to existing techniques. An error analysis is also conducted to identify the strengths and limitations of the proposed system. This chapter provides a comprehensive understanding of the performance and effectiveness of our proposed system.

5.2 Experiments

To build an emotion detection system, we applied various machine learning and deep learning approaches using different word embedding methods. We also fine-tuned the hyperparameters of each model to optimize their performance. The experimental data, evaluation measures, and results of these approaches are presented and analyzed in detail in the following sections. This includes a discussion of any errors that were observed and how they were addressed. Overall, this section provides a comprehensive examination of the different approaches we took to develop our emotion detection system.

Experimental Data

To evaluate the performance of our emotion classification system, we divided our dataset into three sets: training, validation, and test. The training set was used to train our model, the validation set was used to validate the model during training, and the test set was held aside until the end of the training process to evaluate the final performance of the model. Our overall dataset was split into a training set (90% of the data) and a test set (10% of the data), and the training set was further divided into a training set (85% of the data) and a validation set (15% of the data). The model was trained using the training set for 20 epochs, and the validation set was used to validate the model at each epoch. The distribution of the training, validation, and test sets for each class is shown in Table 5.1.

Table 5.1: Class Wise train/validation/test set distribution

Class	Train	Validation	Test
Anger	2728	380	360
Disgust	2417	426	316
Fear	2377	327	329
Joy	6082	967	854
Sadness	5379	804	804
Surprise	4085	620	588
Total	23068	3484	3251

5.3 Evaluation Measures

The proposed emotion detection system was evaluated using two phases: training and testing. To determine the effectiveness of the model, various evaluation measures were utilized, including the confusion matrix, accuracy, loss, precision, recall, and F1-score.

- Accuracy (Ace) is a measure of the proportion of correctly classified instances in a dataset. It is calculated using the following equation:

  $$\text{Ace} = \frac{(TP+TN)}{(TP+FP+TN+FN)} \qquad (5.1)$$

  where TP, FP, TN, and FN are true positive, false positive, true negative, and false negative, respectively.

- Loss is a measure of how well the model is able to predict the correct class for a given input. In our proposed emotion detection system, we used the "log cosh" loss function to calculate the loss. This function is the logarithm of the hyperbolic cosine of the prediction error, and is calculated using the following equation:

$$\text{loss} = \frac{1}{n}\sum_{k=1}^{n} \quad log(cosh(y^{(k)} - y^{-(k)})) \tag{5.2}$$

where y and fj are the target class and the predicted class, respectively. It has been observed that log(cosh(y)) is roughly equal to [(y2) / 2] when y is small, and to [abs(y) - log(2)] for large y. This means that the log cosh loss function behaves mostly like the mean squared error, but is less affected by occasional wildly incorrect predictions.

- A confusion matrix is a table used to evaluate the performance of a classification model. It displays the number of false positive, false negatives, true positives, and true negatives that the model produces. In the case of the emotion detection system, the model is a binary classifier, meaning it has two classes and the confusion matrix will have two rows and two columns.

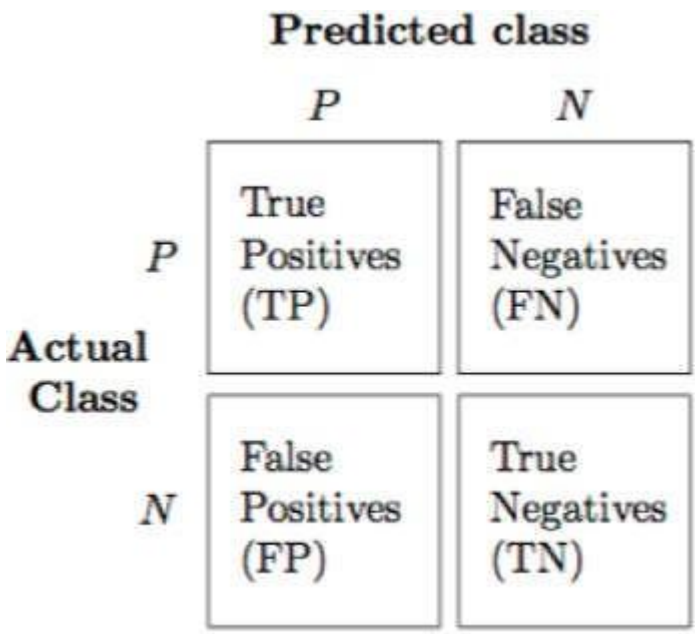

Figure 5.1: Confusion Matrix

Figure 5.1 shows an example of a confusion matrix. The different elements of the matrix are defined as follows:

- True Positive (TP): The number of documents that belong to class A and are correctly classified as class A.
- True Negative (TN): The number of documents that do not belong to class A and are correctly classified as not belonging to class A.
- False Negative (FN): The number of documents that belong to class A but are incorrectly classified as not belonging to class A.
- False Positive (FP): The number of documents that do not belong to class A but are incorrectly classified as belonging to class A.

- Precision is a measure of the accuracy of a classification model when predicting positive outcomes. It is calculated as the proportion of correctly classified positive cases to the total number of positive cases. The precision of a model can be calculated using the following equation:

$$\text{Precision} = \frac{TP}{(TP+FP)} \tag{5.3}$$

A high precision value indicates that the model is doing well at correctly identifying positive cases. Precision is an important metric to consider when the cost of a false positive is high, such as in the case of medical diagnosis or fraud detection.

- Recall is a measure of the effectiveness of a model in correctly identifying all positive instances in a dataset. It is calculated by dividing the number of correctly classified positive instances by the total number of actual positive instances. It is often used to evaluate the performance of a model in identifying rare events or instances of a certain class.

$$\text{Recall} = \frac{TP}{(TP+FN)} \tag{5.4}$$

High precision and high recall are both important for a model to have good performance. However, there is often a trade-off between precision and recall, meaning that improving one may result in a decrease in the other. For example, increasing the threshold of a classifier will typically result in higher precision but lower recall, while decreasing the threshold will result in lower precision but higher recall. To address this trade-off, we can use a measure called the F1-score, which combines precision and recall into a single metric.

- The F1-score is a measure that balances precision and recall in order to evaluate the performance of a classifier. It is calculated as the harmonic mean of precision and recall, with a higher score indicating better performance. The F1-score is commonly used when the consequences of false positive and false negative predictions are not equal. The equation for calculating the F1-score is:

$$\text{F1} = \frac{2 * precision * recall}{(precision + recall)} \tag{5.5}$$

The algorithm with the highest F1-score is generally considered the best-performing algorithm, as it combines both precision and recall into a single metric. It is important to consider the F1-score when comparing multiple algorithms, as it provides a more complete evaluation of performance compared to accuracy alone.

- Cohen's kappa is a measure that is used to evaluate the level of agreement between two people who are categorizing a group of items into exclusive categories. It is calculated by taking the difference between the observed agreement and the hypothetical probability of chance agreement, and dividing it by the maximum possible difference. The observed agreement is determined by the number of items that the two people agree on, while the hypothetical probability of chance agreement is calculated based on the number of items and the number of categories. Cohen's kappa can be used to compare the performance of different classification models, particularly when the consequences of incorrect classifications are not equal.

$$K = (p_0 - p_e)/(1 - p_e) \tag{5.6}$$

where Po denotes the relative observed agreement, which is calculated as the ratio of the number of things on which the annotators agree to the total number of items. Pe is the hypothetical probability of random agreement, calculated as follows:

$$p_e = 1/N^2 \sum_k \quad n_{k1} n_{k2} \qquad (5.7)$$

where N is the number of items to be classified, k the number of categories, and nki the number of times annotator I predicted category k. The kappa coefficient is a helpful indicator for evaluating classification model performance, especially when the cost of false positives and false negatives is not equal.

- The Jaccard index, also known as the Jaccard similarity coefficient, is a measure of the similarity between two sets of data. It is calculated as the ratio of the size of the intersection of the sets to the size of the union of the sets. The Jaccard index can be calculated using the following equation:

$$J(A,B) = \frac{A \cap B}{A \cup B} \qquad (5.8)$$

where A and B are two distinct sets, |A| represents the size of set A, and |A ∩ B| represents the size of the intersection of sets A and B. The Jaccard index is often used to compare the similarity of two sets of data, such as documents or text strings. It is a useful metric for evaluating the performance of clustering or classification algorithms.

5.4 Results Analysis

In the following sections, we will present the results of our analysis on the developed dataset, including the evaluation results of the trained and tested models and a detailed analysis of errors.

Analysis of Dataset

To ensure the quality of our annotated dataset, we evaluated the inter-annotator agreement using coding reliability[35] and Cohen's kappa [36] scores. Our analysis revealed an inter-coder reliability of 98.1% and a Cohen Kappa Score of 0.977, indicating a high level of agreement among the annotators.

Figure 5.2 shows the distribution of the number of texts and their length for each class in our collected corpus. From this figure, we can see that most of the data has a length between 15 and 35, with a higher concentration of texts in the range of 20-30. Additionally, we observed that relatively more words are used to express the emotion of disgust.

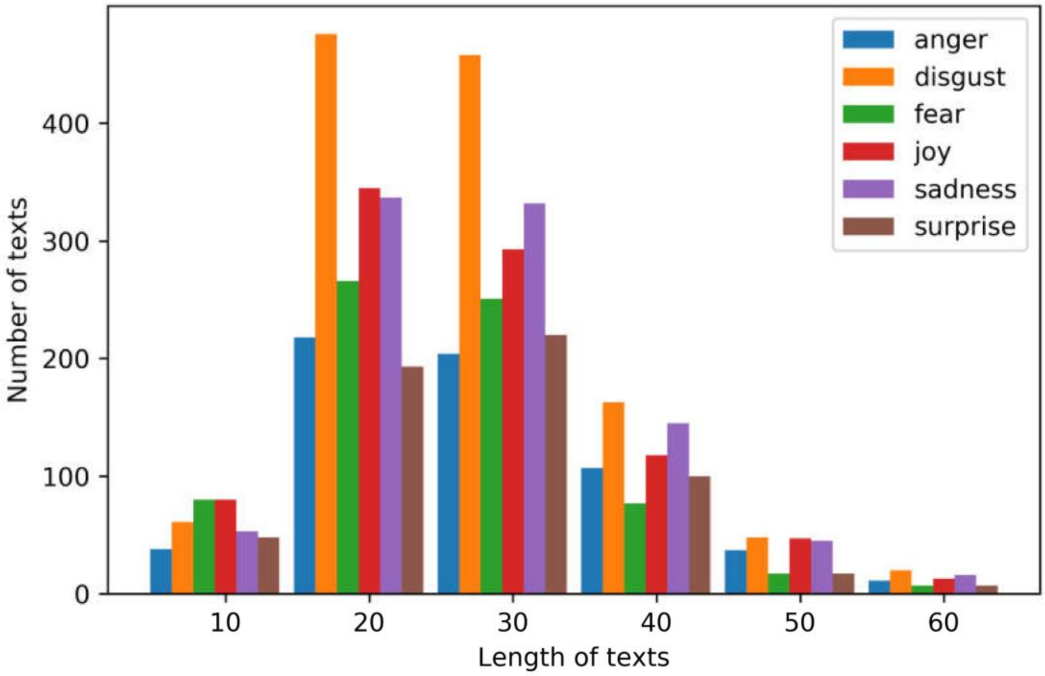

Figure 5.2: Number of texts vs length of texts distribution of the corpus

While analyzing our corpus, we found it to be both challenging and interesting. We calculated the Jaccard similarity among the classes using the 200 most frequent words from each emotion class. The results are shown in Table 5.2. We found that the Anger-Disgust and Joy-Surprise pairs had the highest similarity, with values of 0.58 and 0.51, respectively. These similarities can have a significant impact on the emotion classification task, making it more challenging to accurately classify emotions.

Table 5.2: Jaccard similarity between the pair of the emotion classes. Anger(cl), disgust(c2), fear(c3), joy(c4), sadness(c5), surprise(c6).

	C1	**C2**	**C3**	**C4**	**C5**	**C6**
C1	1.00	0.58	0.40	0.43	0.45	0.47
C2	-	1.00	0.41	0.45	0.47	0.44
C3	-	-	1.00	0.37	0.45	0.46
C4	-	-	-	1.00	0.47	0.51
C5	-	-	-	-	1.00	0.48

Figure 5.3 shows the wordcloud representation for each emotion class, with the words highlighted according to their frequencies. These wordclouds can provide valuable insights into the distinguishing features of each emotion class. The frequent words can be used to help differentiate between the different emotion classes, as they may be indicative of certain characteristics or patterns that are specific to a particular emotion.

(a) Anger (b) Disgust

Figure 5.3: Word Cloud representation of the frequent words in each emotion class

Training Phase Evaluation

We evaluated the performance of our model using several measures, including accuracy (A), precision (P), recall (R), and F1-score (F1). Figure 5.4 shows the training and validation accuracy and loss of the model. It can be seen that the model achieved approximately 99% training and 75% validation accuracy. The minimum training and validation loss achieved were 0.013 and 0.028, respectively. From the validation curve, we observed that the model became saturated after the 5th epoch and did not need further training.

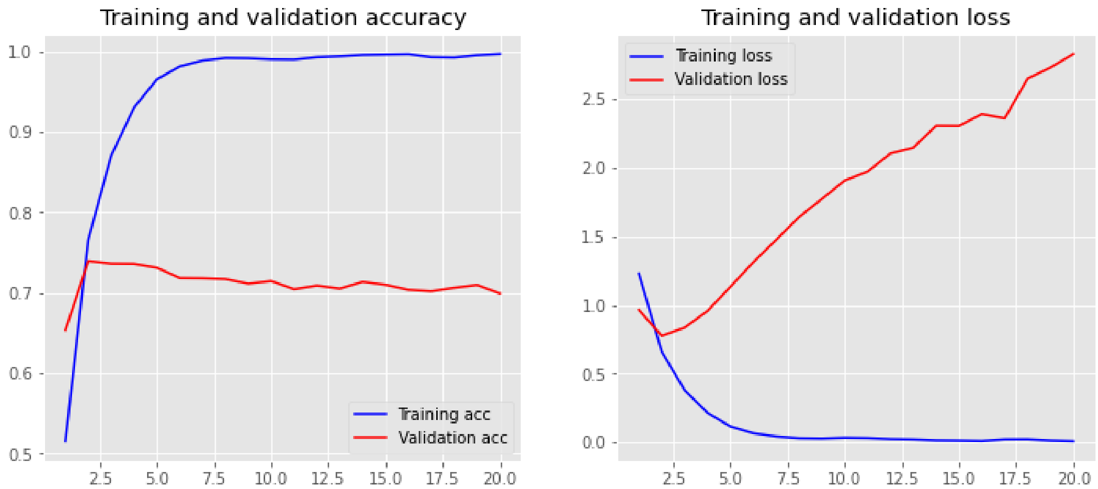

Figure 5.4: Effect of epochs on (a)Training & validation Accuracy and (b) Training & validation loss

Testing Phase Evaluation

We compared the performance of our proposed model with different embedding techniques using our developed corpus. We considered five embedding techniques for comparison: two variations of Word2Vec [6], two variations of Fasttext[37], and Kern's Embedding layer. Table 5.3 shows the results of this comparison. The results indicate that our proposed model, which uses Keras Embedding with CNN, achieved the highest accuracy of 74.40%.

Table 5.3: Performance comparison among embedding techniques

Embedding Techniques	**Accuracy (%)**	
	CNN	**BiLSTM**
Word2Vec (CBOW)	67.46	74.08
Word2Vec (Skip-gram)	68.86	74.7
FastText (CBOW)	59.57	74.12
FastText (Skip-gram)	69.27	74.05
Keras Embedding Layer [Proposed]	73.30	74.35

Table 5.4 shows the performance of our proposed model for each emotion class. The total amount of test data is shown in the rightmost column. The results show that the

emotion category of joy achieved the highest F1 score, while fear achieved the lowest. Some of the fear class data was classified as sadness due to the similarity of some semantic and syntactic features between the texts of these two classes.

Table 5.4: Model performance on the test data using Keras Embedding+CNN

Class	F_1 (%)	P(%)	R(%)	A(%)
Anger	59.50	62.78	56.55	64.30
Disgust	86.65	86.65	92.92	83.02
Fear	63.47	62.95	64.0	59.52
Joy	80.51	78.04	83.15	82.27
Sadness	70.36	69.48	71.26	71.98
Surprise	78.31	85.23	72.43	71.98
Avg.	73.13	74.18	73.39	73.57

Comparisons with Baselines

To further validate our assessment, we compared the performance of our proposed BiLSTM-based model with existing techniques for emotion classification. We implemented previous methods using our developed corpus and reported the results. Table 5.5 summarizes the comparison of these techniques. The results showed that our proposed model achieved higher performance scores than the previous techniques. Figure 5.5 illustrates the accuracies of the different approaches at a glance, with the proposed Keras Embedding+C model having the highest accuracy value.

Error Analysis

Table 5.3 shows that the Keras Embedding with CNN model is the best performing model for classifying emotions from Bengali texts. To further understand the model's performance, we conducted a detailed error analysis using the confusion matrix. Figure 5.6 illustrates the class-wise proportion of the number of predicted labels.

Table 5.5: Performance comparison with previous approaches

Methods	**A(%)**	F_1(%)	**P(%)**	**R(%)**
Word2Vec + LSTM [2]	59.23	58.78	59.23	58.3
Tfidf + MNB [23]	73.6	73.2	73.5	72.67
Tfidf + LR [25]	73.0	73.1	72.4	71.5
Heuristic features + CRF [26]	56.45	55.54	56.4	55.1
BOW + SVM [29]	73.0	72.7	73	72.1
Keras Embedding + CNN [Proposed]	74.40	74.25	74.46	74.40

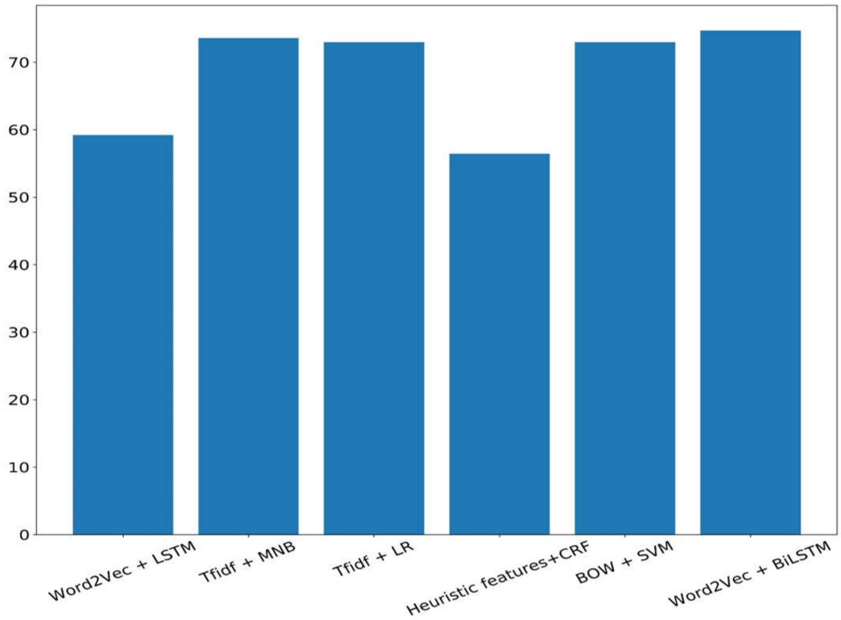

Figure 5.5: Accuracy of different approaches

The joy class received the most correct predictions, with 659 out of 802 data being correctly predicted. This class had the largest amount of data, and the model performed well on this class, as deep learning models typically require a large amount of data to give good performance. In the fear class, 172 out of 289 data were correctly predicted.

53 data from the fear class were predicted as sadness, as the context in fear sometimes leads to sad consequences.

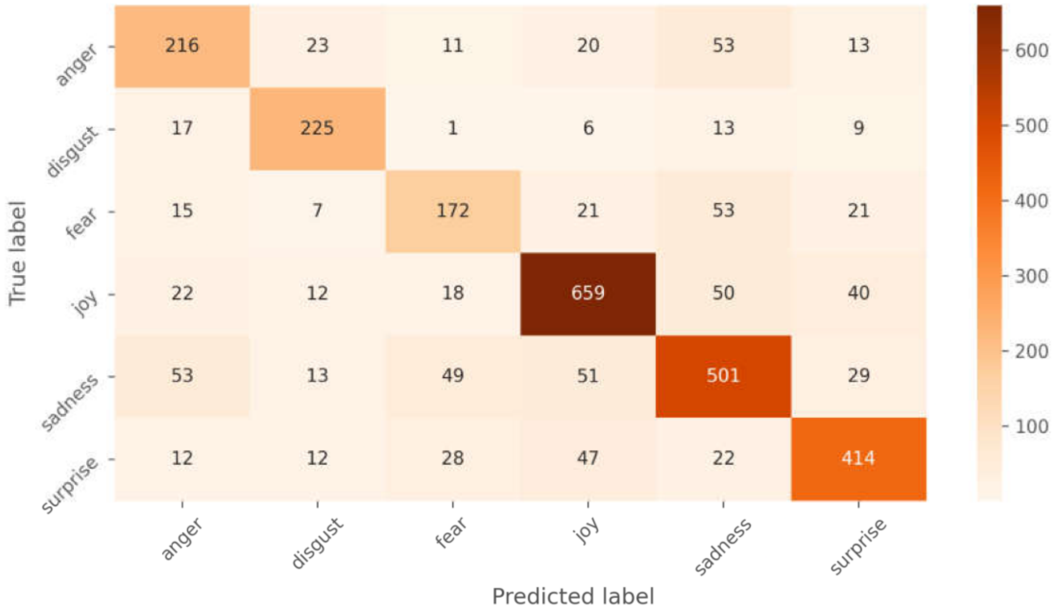

Figure 5.6: Confusion matrix on the test data

Incorrect predictions can occur for a variety of reasons. One factor could be the corpus's class imbalance. Furthermore, the high Jaccard similarity (as shown in Table 5.2) suggests that some terms are employed for various purposes in different classes. Hate words, for example, can be used to indicate both rage and contempt. Furthermore, emotion classification is very subjective and dependent on individual perception, and people may read a text differently. To overcome these concerns, it may be beneficial to create a balanced dataset with diverse data in order to reduce inaccurate predictions.

For further insight, we looked at the actual and predicted output for a few input examples from the best three models, including the proposed method. Table 5.6 shows the actual and predicted output.

5.5 Discussion

Emotion recognition is an important task in natural language processing. Although the results we obtained through our system are not perfect, they provide valuable insights that may be helpful to future research in this field. We found that an imbalanced dataset and similarity among various emotion classes contributed to the reduction in accuracy. Because emotions are a matter of perception, we discovered that sequential deep learning models perform better than traditional machine learning approaches. Specifically, the Keras Embedding with CNN network achieved 74.% accuracy on our developed dataset.

Table 5.6: Comparison among various baselines with the actual and predicted classes. Here 'Yes' means correct prediction and 'No' means wrong prediction

Sample	**Actual Class**	**Predicted Class**					
		CNN	**Bi-LSTM**	**MNB**	**LR**	**SVM**	**RF**
চরিত্রহীন তুই তুর কোন কোয়ালিটি নাই	Anger	Yes	Yes	No	Yes	No	Yes
লোকটার কথা শুনে আমার মেজাজ আরো খারাপ হয়ে গেল মনে মনে বললাম আপনি হয়ত না ঘুমান বাকী সবার কি ঘুমের দরকার নেই	Anger	No	Yes	Yes	Yes	Yes	Yes
নিজের বোন সোনায় সোহাগা আর অন্যের বইন ডাস্টবিনের ময়লা যাইতাম যে কই	Disgust	Yes	Yes	No	No	No	No
যুক্তরাষ্ট্রের সামরিক একাডেমিতে যৌন নিপীড়ন বেড়েছে প্রতিরোধের নানা চেষ্টা চালানোর পরও গত বছর ১৪৯ টি যৌন নিপীড়নের ঘটনা লিপিবদ্ধ করা হয়েছে	Disgust	Yes	No	Yes	No	Yes	No
কি জানি আমার তো ভয়	Fear	No	Yes	No	Yes	No	Yes

কমতেছে না অনেক ভয় লাগে							
পরে বুঝেছি যা দেখেছি সব সত্যি দেখেছি আর তখন স্যার ভীষণ ভয় পেয়েছি স্যার বলতে লজ্জা লাগছে ভয়ে আমি ঘর থেকে দৌড়ে পালিয়ে গিয়েছিলাম	Fear	No	Yes	No	No	Yes	No
এরা ধরাকে সরা জ্ঞান করছিল তাই এদের পতন দেখে জাতি খুশি	Joy	Yes	No	Yes	No	Yes	No
অসাধারণ উপস্থাপনা হাসতে হাসতে অনেক অনেক গুরুত্বপূর্ণ কথা বলেছেন নাভিদ মাহবুব ভিডিওটি দেখে সময় কাজে লেগেছে বলেই মত দিবে সবাই শুভ কামনা	Joy	No	Yes	No	No	Yes	No
দারুণ ভিডিও দাদা আমার চোখ দিয়ে সত্যি জল চলে এসেছিলো	Sadness	No	Yes	No	Yes	No	Yes
বাচ্চাগুলো অসহায়ের মতো তাকাচ্ছে আর আমাদের দুই বাসার সাহায্যকারী মেয়ে দুটি যেন হাল ছেড়ে দিয়ে মাটিতে বসে পড়েছে	Sadness	Yes	Yes	No	No	No	No
কি আজব তাই না কিছু কথা আপনার মনে থাকে আর কিছু মনে থাকেনা	Surprise	No	Yes	Yes	Yes	Yes	Yes
আমি চারমাস কোমায় ছিলাম তাহলে আমি যে আরেকজনের চরিত্রে বেঁচে আসলাম সেটা কি ছিল স্বপ্ন নাকি সত্যি	Surprise	Yes	No	No	No	Yes	No

5.6 Conclusion

Emotion detection is a subjective task that involves the recognition of emotions in text. We have created a dataset and analyzed it in various ways to classify six different emotion classes. After evaluating several methods, we found that the Keras Embedding word embedding model for feature embedding and the CNN recurrent neural network for training the model performed the best in this task. Other algorithms also showed good results, and increasing the number of training documents may further improve their efficiency.

CHAPTER 6
Summary, Conclusion, Recommendation and Implication for Future Research

6.1 Summary of the Study

This thesis presents a deep learning approach for emotion detection in Bengali texts, with the goal of categorizing the text into six primary emotion categories: anger, fear, disgust, joy, sadness, and surprise. The proposed method was tested using different feature embedding, machine learning, and deep learning techniques, and the results of the experiments showed that the CNN based RNN architectures with Keras Embedding feature embedding were effective. The performance of the proposed method was evaluated using several measures, including accuracy, precision, recall, and F1-score. The model achieved an accuracy of 74.40% on the test set, and the results indicate that the proposed model performed better than other embedding techniques. Additionally, the thesis also evaluated the quality of the annotated dataset, and revealed an inter-annotator agreement of 98.1% and a Cohen Kappa Score of 0.977. Overall, this study contributes to the field of emotion detection in Bengali texts, and provides valuable insights that may be helpful to future research in this field.

6.2 Conclusion

We investigated several machine learning and deep learning algorithms for emotion classification in Bengali texts in this thesis. We created a corpus of 29.29k Bengali texts tagged with six basic emotion classes, and the corpus received a Cohen's Kappa score of 0.9177, demonstrating its quality. Given the vast volume of Bangla text data created online and the scarcity of Bangla emotion detection research, we set out to create an emotion recognition system. Our research revealed that utilizing Keras Embedding for feature embedding and a CNN network to train the model produced the best results, with an accuracy of 74.40%.

Limitations

- Limited to six emotion classes
- Misclassification due to limited data
- Class imbalance in the dataset
- Manual tuning of hyperparameters

Overall, the proposed system demonstrated good performance in classifying emotions in Bengali texts. However, there are still areas for improvement, such as increasing the number of classes considered and addressing imbalanced classes and limited data. Automating the hyperparameter tuning process could also potentially improve the performance of the system.

6.3 Implication for Further Study

In this work, we focused on emotion classification in Bengali texts using various machine learning and deep learning techniques. However, there are several areas for future improvement. These include:

- Adding more emotion classes, such as love, hate, and stress, to increase the range of emotions that can be detected.
- Increasing the amount of data in the corpus to improve the accuracy of the model.
- Implementing the ability to detect neutral and multi-emotion classes.
- Using transformer variants, which have achieved strong performance in natural language processing tasks, to further improve the performance of the model.

REFERENCES

[1] P. Ekman, 'Facial expression and emotion.,' American psychologist, vol. 48, no. 4, p. 384, 1993 (cit. on pp. 4, 6).

[2] N. I. Tripto and M. E. Ali, 'Detecting multilabel sentiment and emo- tions from bangla youtube comments,' in 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), IEEE, 2018, pp. 1- 6 (cit. on pp. 4, 16, 17, 42).

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, 0. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, .J. Vanderplas, A. Pas- sos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, 'Scikit- learn: Machine learning in Python,' JoUrnal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011 (cit. on p. 7).

[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and .J. Dean, 'Distributed representations of words and phrases and their compositionality,' arXiv preprint arXiv:1310.4546, 2013 (cit. on p. 8).

[5] A. El Mahdaouy, E. Gaussier and S. 0. El Alaoui, 'Arabic text classification based on word and document embeddings,' in International Conference on Advanced Intelligent Systems and Informatics, Springer, 2016, pp. 32-41 (cit. on p. 9).

[6] T. Mikolov, K. Chen, G. Corrado and .J. Dean, 'Efficient estimation of word representations in vector space,' arXiv preprint arXiv:1301.3181, 2013 (cit. on pp. 9, 40).

[7] J. Pennington, R. Socher and C. D. Manning, 'Glove: Global vectors for word representation,' in Proceedings of the 2014 conference on empirical methods in nat-urnl language processing (EMNLP), 2014, pp. 1532-1543 (cit. on p. 9).

[8] P. Bojanowski, E. Grave, A. .Joulin and T. Mikolov, 'Enriching word vectors with subword information,' Transactions of the Association for Comp,uta- tional LingUistics, vol. 5, pp. 135-146, 2017 (cit. on pp. 9, 24).

[9] J.-Y. Yoo and D. Yang, 'Classification scheme of unstructured text docu- ment using tf-idf and naive bayes classifier,' in Computer and CompUting Science 2015, 2015, pp. 263-266 (cit. on p. 10).

[10] T. Villmann, A. Bohnsack and M. Kaden, 'Can learning vector quantization be an alternative to SVM and deep learning?-recent trends and advanced variants of learning vector quantization for classification learning,' JoUrnal of Artificial Intelligence and Soft CompUting Research, vol. 7, no. 1, pp. 65- 81, 2017 (cit. on p. 11).

[11] M. Sharma, D. Zhuang and M. Bilgic, 'Active learning with rationales for text classification,' in Proceedings of the 2015 Conference of the North American Chapter of the

Association for Comp,utational Ling,uistics: Hu- man Language Technologies, 2015, pp. 441-451 (cit. on p. 12).

[12] Imdb, (Accessed on 11/10/2020). [Online]. Available: http://www. imdb. com/ (cit. on p. 15).

[13] The yelp restaUrant reviews, (Accessed on 11/10/2020). [Online]. Available: https://www.yelp.com/dataset/ (cit. on p. 15).

[14] Movie reviews, (Accessed on 11/10/2020). [Online]. Available: https: / / www.kaggle.com/rpnuser8182/rotten-tomatoes (cit. on p. 15).

[15] S. Rosenthal, N. Farra and P. Nakov, 'Semeval-2017 task 4: Sentiment analysis in twitter,' in Proceedings of the 11th international workshop on semantic evaluation (SemEual-2017), 2017, pp. 502-518 (cit. on p. 15).

[16] N. Alswaidan and M. E. B. Menai, 'A survey of state-of-the-art approaches for emotion recognition in text,' Knowledge and Information Systems, pp. 1- 51, 2020 (cit. on p. 15).

[17] F. Calefato, F. Lanubile and N. Novielli, 'Emotxt: A toolkit for emotion recognition from text,' in 2017 seventh 'international conjerence on Affective Comp,uting and Intelligent Interaction Workshops and Demos (ACIIW), IEEE, 2017, pp. 79-80 (cit. on p. 15).

[18] S. Alzu'bi, 0. Badarneh, B. Hawashin, M. Al-Ayyoub, N. Alhindawi and Y. Jararweh, 'Multi-label emotion classification for arabic tweets,' in 2019 Sixth International Conference on Social Networks Analysis, Management and Sernrity (SNAMS), IEEE, 2019, pp. 499-504 (cit. on p. 16).

[19] M. Hasan, E. Rundensteiner and E. Agu, 'Automatic emotion detection in text streams by analyzing twitter data,' International Jo,urnal of Data Science and Analytics, vol. 7, no. 1, pp. 35-51, 2019 (cit. on p. 16).

[20] Y. Lai, L. Zhang, D. Han, R. Zhou and G. Wang, 'Fine-grained emotion classification of chinese microblogs based on graph convolution networks,' World Wide Web, vol. 23, no. 5, pp. 2771-2787, 2020 (cit. on p. 16).

[21] D. Haryadi and G. P. Kusuma, 'Emotion detection in text using nested long short-term memory,' IJACSA) International JoUrnal of Advanced CompUter Science and Applications, vol. 10, no. 6, 2019 (cit. on p. 16).

[22] A. Chatterjee, K. N. Narahari, M. Joshi and P. Agrawal, 'Semeval-2019 task 3: Emo Context contextual emotion detection in text,' in Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 39-48 (cit. on p. 16).

[23] S. Azmin and K. Dhar, 'Emotion detection from bangla text corpus using naive bayes classifier,' in 2019 4th International Conference on Electrical Information and Comm:unication Technology (EICT), IEEE, 2019, pp. 1-5 (cit. on pp. 16, 17, 42).

[24] M. Rahman, M. Seddiqui et al., 'Comparison of classical machine learning approaches on bangla textual emotion analysis,' arXiv preprint arXiv:1907.07826, 2019 (cit. on p. 16).

[25] A. Pal and B. Karn, 'Anubhuti-an annotated dataset for emotional analysis of bengali short stories,' arXiv preprint arXiv:2010. 03065, 2020 (cit. on pp. 16, 17, 42).

[26] D. Das and S. Bandyopadhyay, 'Word to sentence level emotion tagging for bengali blogs,' in Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, 2009, pp. 149-152 (cit. on pp. 17, 42).

[27] D. Das, S. Roy and S. Bandyopadhyay, 'Emotion tracking on blogs-a case study for bengali,' in International Conference on Industrial, Engineer- ing and Other Applications of Applied Intelligent Systems, Springer, 2012, pp. 447-456 (cit. on p. 17).

[28] D. Das and S. Bandyopadhyay, 'Developing bengali wordnet affect for ana- lyzing emotion,' in International Conference on the Computer Processing of Oriental Languages, 2010, pp. 35-40 (cit. on p. 17).

[29] H. A. Ruposh and M. M. Hoque, 'A computational approach of recognizing emotion from bengali texts,' in 2019 5th International Conference on Ad- vances in Electrical Engineering (ICAEE}, IEEE, 2019, pp. 570-574 (cit. on pp. 17, 42).

[30] A. Das, M.A. Iqbal, 0. Sharif and M. M. Hoque, 'Bemod: Development of bengali emotion dataset for classifying expressions of emotion in texts,' in Intelligent CompUting and Optimization, P. Vasant, I. Zelinka and G.-W. Weber, Eds., Cham: Springer International Publishing, 2021, pp. 1124- 1136, ISBN: 978-3-030-68154-8 (cit. on p. 19).

[31] Accessible dictionary, https : / / accessible dictionary. gov. bd/ (ac- cessed on 2 January 2020) (cit. on p. 20).

[32] Full emoji list, https: //unicode. org/emoji/charts/full-emoji- list .html (accessed on 7 February 2020) (cit. on p. 21).

[33] Y. LeCun, Y. Bengio and G. Hinton, 'Deep learning,' nature, vol. 521, no. 7553, pp. 436-444, 2015 (cit. on p. 26).

[34] S. Hochreiter and J. Schmidhuber, 'Long Short-Term Memory,' Neural CompUtation, vol. 9, no. 8, pp. 1735-1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735, eprint: https://direct.mit.edu/neco/ article-pdf /9/8/1735/813796/neco. 1997.9.8. 1735. pdf. [Online]. Available: https : / / doi. org/ 10. 1162 / neco . 1997 . 9. 8. 1735 (cit. on p. 26).

[35] K. Krippendorff, 'Agreement and information in the reliability of coding,' Communication Methods and Meas-ures, vol. 5, no. 2, pp. 93-112, 2011. DOI: 10 .1080/19312458. 2011. 568376 (cit. on p. 37).

[36] J. Cohen, 'A coefficient of agreement for nominal scales,' Educational and Psychological Meas-urement, vol. 20, no. 1, pp. 37-46, 1960. DOI: 10 .1177/ 001316446002000104 (cit. on p. 37).

[37] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch and A. Joulin, 'Ad- vances in pre-training distributed word representations,' in Proceedings of the International Conference on Lang·uage Reso-urces and Eval-uation (LREC 2018), 2018 (cit. on p. 40).

Ismata & Anika

ORIGINALITY REPORT

28%	22%	18%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
2	web.archive.org Internet Source	3%
3	Submitted to Daffodil International University Student Paper	2%
4	aclanthology.org Internet Source	1%
5	doctorpenguin.com Internet Source	1%
6	Submitted to Liverpool John Moores University Student Paper	1%
7	www.techscience.com Internet Source	1%
8	ebin.pub Internet Source	1%
9	www.ncbi.nlm.nih.gov Internet Source	1%