Bachelor of Science in Computer Science & Engineering

Automatic Detection of Human Emotion from Bangla Text Using Deep Learning

by

Avishek Das

ID: 1504018

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

April, 2021

Automatic Detection of Human Emotion from Bangla Text Using Deep Learning

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Avishek Das
ID: 1504018

Supervised by

Dr. Mohammed Moshiul Hoque
Professor
Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.

The thesis titled '**Automatic Detection of Human Emotion from Bangla Text Using Deep Learning'** submitted by ID: 1504018, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

______________________________ Chairman

Dr. Mohammed Moshiul Hoque
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

______________________________ Member (Ex-Officio)

Dr. Asaduzzaman
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

______________________________ Member (External)

Prof. Dr. Mohammad Shamsul Arefin
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

AVISHEK DAS

Signature of the candidate

Date: 19-4-2021

Acknowledgements

The joy that comes with the successful completion of this project would be incomplete if it weren't for the people whose unwavering cooperation made it possible, and whose relentless guidance and encouragement ensured the success of all efforts. Dr. Mohammed Moshiul Hoque, Professor, Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, is my honorable project supervisor, and I am grateful for his guidance, inspiration, and constructive suggestions in the preparation of this work. I am thankful for his many crucial questions, his exalted support throughout the entire time, and motivating me to see things from diverse perspectives on Bengali language processing domains.

I owe my gratitude to Omar Sharif, Lecturer, Department of Computer Science & Engineering, Chittagong University of Engineering & Technology (CUET) who took a keen interest in my work and guided me by providing the necessary information. My sincere thanks to Professor Dr. Asaduzzaman, Head, Department of CSE, CUET for providing his valuable support. I am thankful for the endless encouragement, support, and guidance from all Teaching staff of the department.

I would also like to express my gratitude to all my teachers, data crawlers, annotators and supporting staffs of CSE Department for their helpful cooperation, and assistance, which contributed to the thesis's successful completion.

This research conducted under this thesis was funded by Directorate of Research & Extension, CUET, whose support is greatly appreciated.

Abstract

Emotion detection is a computational approach to find the distinct emotion or feeling of an individual. It can be expressed through facial expression, verbal communication, or textual representation. An enormous amount of textual data is generated over the globe due to the endless use of Web 2.0 applications. Although Bengali considered as a low resource languages the amount of textual data has increased rapidly in recent years. Emotion classification in the Bengali texts is also gradually being considered as an important task for sports, e-commerce, entertainments, and security applications. However, unavailability of necessary language processing tools and and deficiency of benchmark corpora makes the emotion classification task in Bengali more complicated. This thesis proposes a deep-learning approach to classify Bengali text data into one of the six basic emotion categories: anger, fear, disgust, sad, joy, and surprise. Due to the unavailability of benchmark dataset, this work develops a Bengali emotion corpus consisting of 29,290 sentences with 40,718 unique words to perform the emotion classification task. This thesis explores several word embedding techniques such as Word2Vec, FastText, Keras Embedding Layer to find the appropriate features for Bengali textual emotion classification. To investigate the performance of Bengali textual emotion classification task, several machine learning and deep learning models (such as MNB, LR, SVM, CNN, and LSTM) including the proposed (Word2Vec+BiLSTM) is evaluated on the developed corpus with various feature extraction/word embedding technique. The comparative analysis revealed that BiLSTM-based method with Word2Vec word embedding outperforms the other techniques with achieving a highest accuracy of 74.7% on test dataset.

***Keywords*— Natural language processing, Resource constrained language, Textual emotion classification, Bengali language processing, Sentiment classification, Emotion corpus**

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The original goal of this thesis is to use deep learning techniques to create a method for detecting emotion in Bangla texts. Emotions are feelings that are generated by certain situations and surroundings. The computational process of categorizing human emotions from text, audio, video, heart rate, blood pressure is known as Emotion Detection. Emotion detection system can be used in various fields. Customer reviews and feedbacks, recommendations, advertising, social media, healthcare are some of the fields. Intelligent chat-bots and agents are another most active research area involving human emotions. Availability of huge amount of online data and advancement of computational process have accelerated the development of emotion detection research on different languages like English, Arabic and French etc. This thesis proposes deep learning-based method that can classify a Bangla textual data into six emotion classes: anger, fear, disgust, joy, sadness, and surprise. The overview of the emotion classification framework is explained in this Chapter. This Chapter also explains the difficulties, applications, motivation, and contribution of the thesis. Finally, the organization of the thesis is presented at the end of this Chapter.

1.2 Emotion Classification and Textual Emotion Classification

Emotion classification in the text signifies to the task of automatically attributing an emotion category to a textual document selected from a set of predetermined emotion categories. Such as: "ছোট বেলা থেকেই অনাথ ছেলেটি নিজে নিজে বাচতে শিখে গিয়েছে "

-this text can be an expression of *Sadness* category. Emotion classification is an application of text classification.

1.3 General Framework of Emotion Classification

The key objective of our work is to develop a system that can detect human emotion from text using deep learning. Figure 1.1 shows the abstract schematic diagram of the system. The detection system will be made using several steps. Detailed processing of the system is discussed in Chapter-3.

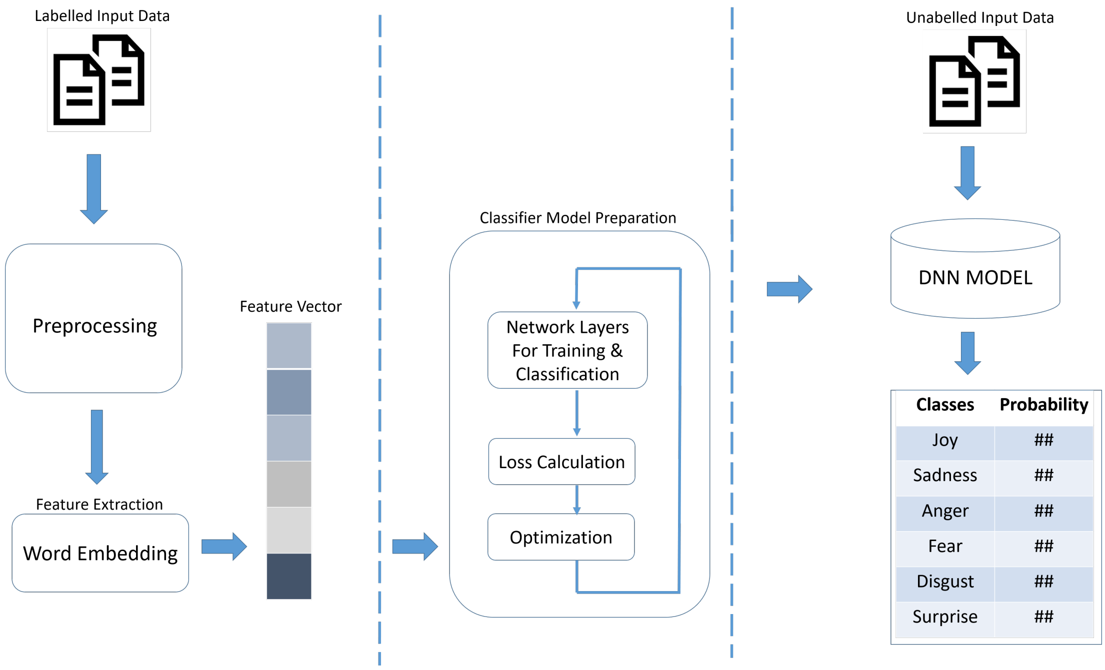


Figure 1.1: Overview of the emotion classification

1.4 Difficulties and Challenges

We have to face several difficulties to implement this system. Some of this difficulties are listed below,

- First difficulty, for the implementation of this system the most challenging task was to develop a dataset which can be used by our learning algorithm. We know that for any deep learning algorithm a well-furnished dataset is a key. Bangla is a low resource language and there are no good quality corpora is available for research on emotion classification.. We try to overcome this problem by collecting a large amount of text from different Bangla source and social media. We collect about 29.29K emotion text and it takes us around six months to prepare this dataset.

- The second challenge was to label the collected data into different emotion category. If we use wrong data to train the classifier, then performance will decrease. Semantic feature is used to classify text properly. Semantic features represent the basic conceptual components of meaning for any lexical item. An individual semantic feature constitutes one component of a word's intention, which is the inherent sense or concept evoked.

- Hardware support was another vital challenge for our system. For getting better accuracy we have to process huge amount of texts. We set up a high computation power hardware which able to process large amount of texts.

1.5 Applications

With the growing amount of users in virtual platforms generating online contents steadily as a fast-paced, interpreting emotion or sentiment in online contents is great importance for consumers, enterprises, business leaders, and other parties concerned.

- There are various implications of classifying emotion where interpreting the emotions can play a great significance such as business, politics, education, healthcare and entertainment

- Chatbots and intelligent agents can also be used an emotion recognition system for better social interaction.

- Also suicidal attempts can be detected by analysis user emotional behaviour and necessary steps can be taken.

- Criminal activities through texts can be detected and prevented accordingly.

1.6 Motivation

Most of the research works done on emotion detection are in English or other languages, while Bangla text, now-a-days is being used widely over the internet. However, there are several works done on Bangla sentiment analysis which represent only if a sentence(s) is/are positive, negative or neutral. Regarding emotions, there are six types of basic human emotions – happiness, fear, anger, sadness, surprise & disgust [1]. We will try to produce a deep learning method to categorize these basic types of emotions from Bangla text. Though there are a small number of works on emotion detection from Bangla text using deep learning, their result was not so satisfactory [2]. Bangla is spoken by more than 210 million people as first or second language [1]. Every day huge amount of Bangla text data is being generated online. Almost all of them indicates to some kind of emotion. Besides Bangla emotion detection is relatively new work in Bangla language processing field. This motivated us to design emotion detection system which will automatically detect emotion from Bangla text.

1.7 Contribution of the thesis

The principle purpose of this work is to classify a Bangla text into an emotion class. This specific contributions of the work illustrates in the following:

- Develop a corpus containing 29.29K Bangla text documents with 40,718 unique words to classify six basic emotions: mention the classes Anger, Fear, Disgust, Joy, Sadness, and Surprise.
- Investigate various word feature extraction/embedding techniques including Word2Vec, FastText, Keras Embedding Layer with hyperparameters tuning for Bangla textual emotion classification.
- Develop a deep learning-based framework using Word2Vec embedding and BiLSTM network to classify textual emotions in the Bangla.

[1] https://www.britannica.com/topic/Bengali-language

- Investigate and compare the performance of the proposed model with other ML baselines and existing techniques.

1.8 Thesis Organization

This thesis is organized into five chapters. The thesis is organized as follows:

- Chapter two contains brief discussion on previous works that is already implemented, their limitations and their role on text classification using machine learning and deep learning.
- Chapter three describes proposed system with necessary diagrams. An overall system architecture is given on this chapter. Also our implementation of the project in details have been illustrated.
- Chapter four focuses on the experimental results of the system. Evaluation measures and results of our system are described in this chapter.
- Chapter five consists of conclusion with the summary of our system and the future plan of our system.

1.9 Conclusion

In this chapter, we discussed about some introductory readings on emotion classification, a few challenges of implementation of our work, and the contributions we made. The motivation behind this work is also stated here. In the next chapter, background and present state of the problem will be discussed.

Chapter 2

Literature Review

2.1 Introduction

Analysis of textual emotion has gained much attention among the NLP researchers in recent years. Chapter 2 explains a few basic terminology and terms related to emotion detection/classification. Moreover, this Chapter discuss the various textual emotion classification techniques that is closely related to the proposed work. Few implementation challenges also highlights at the end of this Chapter.

2.2 Important Terms and Terminology

There are several critical terms and terminology related to the text-based emotion classification task which is described in the following:

- **Text Classification**: Text classification is the task of assigning a text into a set of predefined classes automatically. Because of the rapid growth of online information, text classification has become more challenging and more important as well. Text classification can be described into two categories.
- **Opinion Mining**: Opinion mining is a text analysis system that employs computational linguistics and natural language processing to classify and extract opinion from text automatically.
- **Basic Emotion Classes:** According to Ekman, there are six basic emotions: anger, disgust, fear, joy, sadness and surprise[1].
- **Sentiment Analysis:** Sentiment analysis, a branch if opinion mining, is a technique for determining whether data is positive, negative, or neutral using natural language processing.

- **Implicit Emotion:** Implicit emotion denotes the situation where the emotion is typically unapparent within the expression.
- **Explicit Emotions:** In Explicit emotion, the emotion is typically clear in the expression.

2.3 Feature Extraction Methods for Emotion Analysis

Extracting important features is a key step in any classification task. Several contextual and non-contextual techniques have been used to extract relevant syntactic/semantic features from text expression. Some of the most popular feature extraction techniques are described in the following:

2.3.1 Tf-Idf Vectorizer

Tf-idf, a short form of "Term-frequency time's inverse document-frequency" is a numerical statistic used widely in NLP tasks. It calculates the significance of a word in a given document concerning the overall occurrence of that word in a dataset. Tf-idf rises when a word/term is more frequent in a document but less frequent in a whole dataset of documents. The formula to compute tf-idf of a term t in document d in a document set is calculated form equation-2.1 [3]:

$$tf\_idf(t,d) \;=\; tf(t,d) * \; idf(t) \tag{2.1}$$

where "tf" is the term frequency that is the total number of term t in document d calculated from equation-2.2:

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

and "idf" is inverse document frequency calculated as equation-2.3:

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

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

2.3.2 Word2Vec

Word2Vec is a novel and widely used word embedding model. It uses neural networks to find the semantic similarity of the context of the words. Two inversely related architectures are used in word2vec, skip-gram, and continuous bag of words. Skip gram is an unsupervised learning architecture used to find semantically similar words based on the context of a given word. Skip gram calculates the maximum average logarithmic probability in equation 2.4 [4]

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

from given training words w1,w2,w3.....wN. Here c is the size of the context also called the window size. The probability $p\left(w_{n+m}|w_n\right)$ can be calculated using equation-2.5:

$$p\left(i|\ o\right) = \frac{\exp\left(u^T{}_i.u'_o\right)}{\sum_{v\in V} exp\left(u^T{}_z.u'_o\right)} \tag{2.5}$$

where V is the vocabulary list and u, u' are 'input' and 'output' vector representations of i, o respectively. Continuous Bag of Words (CBOW) is another architecture highly used in language processing tasks. Unlike Skip-Gram, CBOW predicts a target word based on the context of some given words as input. CBOW uses continuous distributed representations of the context. In CBOW a fixed window is constructed with some sequence of words and the model tries to predict the middle word of the window based on the future and history words using log-linear classifier. The model tries to maximize equation-2.6 [5] to achieve the predicted word wt.

$$\frac{1}{V}\sum_{v\in V} log\left(p\left(w_v|\ w_{v-c},\ \ldots w_{v-2}, w_{v-1},\ w_{v+1},\ w_{v+2}\ldots w_{v+c}\right)\right) \tag{2.6}$$

Where V and c represent the same as Skip-Gram model. Figure 2.1 [6] represents both the model in a nutshell.

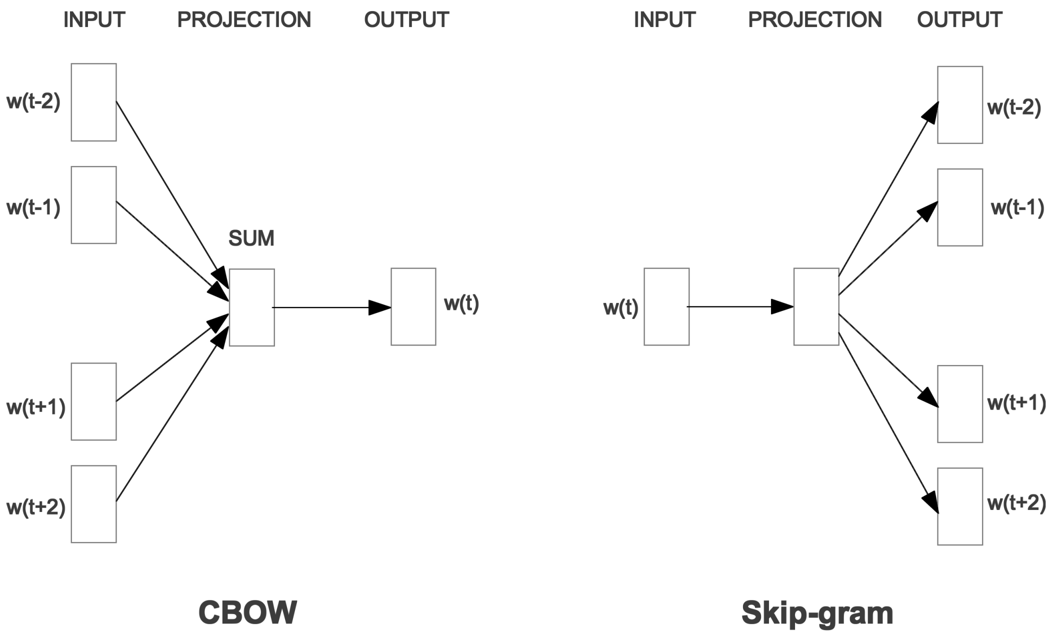


Figure 2.1: Skip-gram and CBOW architecture

2.3.3 Glove

Glove or Global Vectors learns the embeddings from word co-occurrences[7]. It creates a co-occurrence matrix C where the rows and columns denote the word vocabulary and each entry in C, that is Cij, denotes the co-occurrence weight of word i and j. Higher weight results in higher similarity in the resultant vectors.

2.3.4 FastText

FastText is one of the most robust techniques used for word embedding by using subword information [8]. This model learns the embeddings from character n-grams of the training words. As a result, a non-existing word in the vocabulary during training time can be constructed from its constituent n-grams. This surpasses the limitation of word2vec and GloVe where a non-vocab word can't be obtained after training.

2.4 Machine Learning Methods for Emotion Analysis

Nowadays, various machine learning (ML) techniques have been employed for the classification task. These techniques also most commonly used in analysis textual emotion classification in many languages. Some popular ML techniques are explained in the following subsections.

2.4.1 Naive Bayes Classifier

Naive Bayes is a probabilistic classifier that is widely used in machine learning[9]. Bayesian classifiers are both statistical and learnable. For processing large dataset multinomial model of Naive Bayes is used. By searching the dependencies among attributes the performance of Naive Bayes could be enhanced. Due to its ease of computation, it is primarily used in data preprocessing applications. In order to predict the target class, Bayesian reasoning and probability inference are used. When using a probabilistic model to classify data, attributes are crucial. As a result, attribute weight values play an important role in improving the model's efficiency. Deep feature weighting solves the conditional independence assumption, which is a significant improvement over the Naive Bayesian classifier and reliably computes conditional probability. However, these feature weighting strategies have flaws such as insufficient performance enhancement, compromised simplicity, and increased model execution time, among others.

The precision of the predicted conditional probability terms determines how well Naive Bayes performs. It's difficult to predict conditional probability terms correctly when the training data is scattered. To estimate conditional probability terms, several meta learning methods are used. Structure extension, attribute collection, frequency transforming, attribute weighting, instance weighting, and local learning are some of the meta-learning techniques used to enhance this. The Naive Bayes classifier is used to classify emails as spam/ham, to classify articles based on content, and to analyze sentiment/emotion. It can also be used to identify texts into suspicious and non-suspicious categories in our project. The benefit of Naive Bayesian classifiers is that they are easy and efficient in terms of degree of certainty, optimization is easier, and dynamic adaptation is possible. These characteristics make them a more appealing choice for dealing with natural language processing issues.

The main limitation of Bayesian networks is that the time complexity increases when high dimensional text data is processed using these networks. Moreover, in Bayesian networks interaction between features can not be achieved and the probabilities calculated are not accurate but relative probability.

2.4.2 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. Credit risk analysis, medical diagnosis, text categorization, and knowledge extraction are all possible applications. Since the complexity of the classifiers is determined by the number of support vectors rather than the data dimensions, they generate the same hyper plane for repeated training samples and have good generalization abilities. It separate classes by placing hyper plane between classes. It selects optimal hyper plane from which distance of classes are maximized. The performance of SVM does not decrease with sparsity of data. SVM is a really powerful tool for processing data and extracting information when dataset in huge. SVM's performance can be improved by using customized kernels. Class Meaning Kernel is one such customizable kernel that is used to smooth words in documents using class-based meaning values. SVM has a number of distinguishing characteristics that have led to it being regarded as the state-of-the-art in classification tasks. Text classification, handwritten digit identification, and a variety of other classification tasks have all been accomplished using SVM. It has some unique characteristics, such as the ability to function well in a very high-dimensional feature space, the use of only a subset of the original training set to create decision boundaries called support vectors, and the ability to work with non-linearly separable data (it uses kernel trick). SVM we can select maximum features length for our model during learning.Using the parameter max-features in our SVM learning model, we selected the 1000 most frequent features.

The limitation of Support Vector Machine is it still lags in handling unlabeled data. It has to be verified that data is thoroughly preprocessed to increase the performance of classifiers. In addition, selecting the best kernel among available kernels to train data is time consuming. Training and testing using SVM model is time consuming. As SVM is non parametric model it could not summarize data based on underlying parameters.

2.4.3 Logistic Regression

Logistic regression [11] is a methodology borrowed from statistics by machine learning. It's the form of choice for binary classification issues (problems with two class values). The central method used in the logistic function is called Logistic Regression. Max likelihood estimation is used to learn logistic regression. Since it makes assumptions about the distribution of your results, maximum-likelihood estimation is a popular learning method used by a number of machine learning algorithms. The best coefficients would result in a model that predicted a value for the default class that was very close to 1 and a value for the other class that was very close to 0. The idea behind maximum-likelihood logistic regression is that a search procedure looks for coefficient values that minimize the difference between the model's expected probabilities and those in the results.

The performance of logistic regression depends on cost function. If we can reduce the value of cost function then the model will perform better. At the time of fitting the parameters to the model we must careful about over-fitting or under-fitting characterstics of the model. If the model overfits it will perform well on training set but performs poorly on test set. Regularization techniques often used to prevent over-fitting. Logistic regression model can be used in Financial forecasting, Software cost prediction, software effort prediction, Software quality assurance, Crime data mining etc.

The main drawback of Logistic Regression is that it could not separate non-linearly separable classes. In addition to ensure better accuracy with Logistic Regression large sample space is required.

2.5 Deep Learning Algorithms

Recently, the deep learning (DL) techniques extensively used in many classification problem due to its capability of contextual feature extraction. In the following subsection, a few popular DL techniques used in emotion classification is explained.

2.5.1 Convolutional Neural Network (CNN)

CNNs are a type of deep, feed-forward artificial neural network where the correlations between nodes do not form a cycle and which employ a multilayer perceptron variation that requires little preprocessing. These were influenced by the visual cortex of animals. CNNs are typically used in computer vision, but they've recently been applied in a wide variety of NLP tasks, with encouraging results.

Let's look at a figure 2.2[1] to see what happens when we use CNN on text info. When a specific pattern is detected, the outcome of each convolution will fire. By adjusting the size of the kernels then concatenating the outputs, we can detect patterns of different sizes (2, 3, or 5 adjacent words). Patterns may be phrases (word ngrams?) such as "I despise," "very nice," and thus CNNs will recognize them in a sentence regardless of their place.

[1] Image Source:http://www.wildml.com/2015/11/understanding-convolutional\-neural-networks-for-nlp/

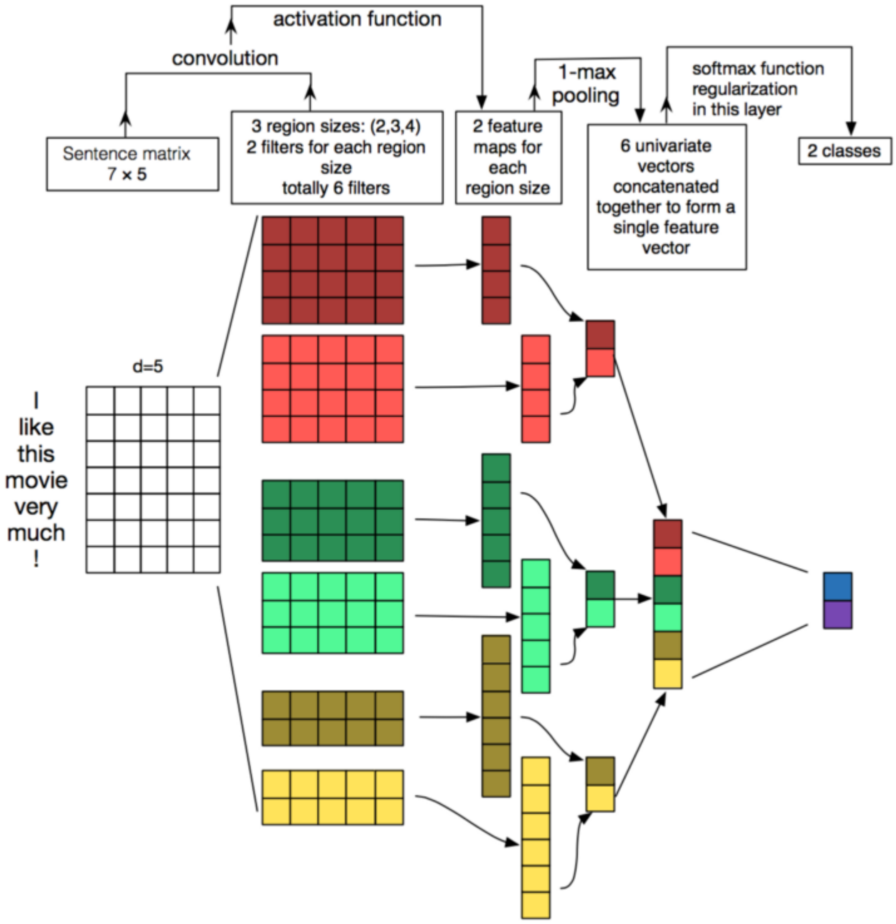


Figure 2.2: CNN blocks for textual data

2.5.2 Bidirectional Long-Short Term Memory (Bi-LSTM)

A bidirectional LSTM, also known as a biLSTM, is a sequence processing model that consists of two LSTMs, one of which takes the input forward and the other backward. BiLSTMs essentially increase the amount of data accessible to the network, allowing the algorithm to better understand the context (e.g. knowing what words immediately follow and precede a word in a sentence). Figure 2.3[2] shows the overview of a typical BiLSTM network.

[2]Image Source: Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks, Cornegruta et al

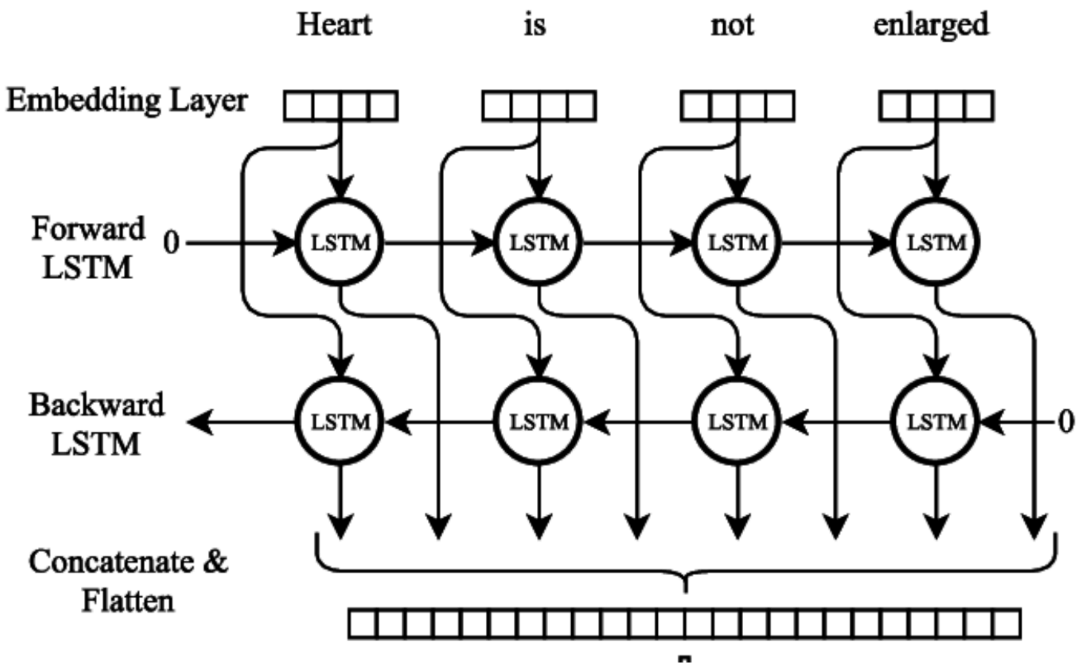


Figure 2.3: BiLSTM network for textual data

2.6 Related Work

Although the textual emotion classification has achieved an enormous progress concerning the highly resources languages (i.e., English, Chinise, Arabic, and Spanish), there is a very few research activities have been conducted in Bengali language. Based on the survey of previous literature, we presents the reviews on emotion classification concerning non-Bengali language based emotion classification and Bengali language based emotion classification.

2.6.1 Non-Bengali Language based Emotion Classification

Most of the language processing works are highly concentrated on the English language as standard datasets are available in English. It is so unfortunate that there is no standard collection of data on Bangla, such as – the IMDB dataset[12], restaurant[13] or movie review dataset [14], and semEval[15]. There is significant progress in emotion classification in English, Hindi, Arabic, and Chinese languages [16]. A toolkit named EmoTxt was developed using machine learning algorithms for English languages based on questions-answers from online forums[17]. A multilabel multi-target emotion detection of Arabic tweets was accomplished using Decision Trees, Random Forest, and K nearest neighbor, where random forest gave 82.6% F1 score [18]. An automatic twitter tweets-based emotion detection system was developed by Hasan et al. [19]. They applied supervised machine learning algorithm and got 90% accuracy on their proposed model but only four classes of emotion were considered. Several deep learning approaches were also considered for emotion classification of short sentences. Lai et. al [20] proposed a graph convolution network architecture for emotion classification of Chinese microblogs and their proposed system achieved a F-measure of 82.32%. In [21], Nested Long-Short Term Memory(LSTM) was used by Haryadi et. al to classify 7 emotion classes: anger, fear, joy, love, sadness, surprise, and thankfulness and 99.167% accuracy was achieved. This work showed, nested LSTM outperforms LSTM in finding semantic relationships between words in a sentence. SemEval-2019 Task 3[22] proposed a Bi-LSTM architecture for emotion classification which is the most common choice of neural network architecture in terms of text classification. They considered four classes and achieved 79.59 F1 score.

2.6.2 Bengali Language based Emotion Classification

There have been several works on sentiment analysis and a small number of works on emotion analysis from Bangla text. Among them, a research work on detecting multi-label sentiment and emotions from Bangla, English & Romanized Bangla sentences was done using only youtube comments in [2]. They claimed to have developed an emotion dataset containing 1006 Bangla emotional youtube comments. A model was developed by them for three label sentiment, five label sentiment and emotion detection with accuracy 65.97%, 54.24%, 59.23% respectively. They used both LSTM and Convolutional Neural Network(CNN) to compare and LSTM performed better. Another work with three emotional labels i.e. happy, sadness and anger was carried out in [23]. They used a dataset developed by [24] and applied some statistical machine learning methods. Multinomial Naïve Bias (MNB) outperformed others with an accuracy of 78.6%. A new corpus was developed in [25] named 'Anubhuti' which is not publicly available yet. The domain of the corpus is mainly concentrated on Bangla short stories only. They labeled the corpus with four emotion classes: joy, anger sorrow, and suspense and got an accuracy of 73% applying logistic regression. Deep learning methods were also applied but less accuracy was achieved. Das et al. [26] conducted a study to identify emotions in Bengali blog texts. They used conditional random field (CRF) to detect emotional expression from blogs and attained 56.45% accuracy. Another work by them in [27] showed Ekman's six basic emotion classification done on Bengali blogs and News texts at word and sentence level. They used the Bangla wordnet affect list developed in [28] which is a translated version of the English wordnet affect list and got F-score of 62.1%. A recent work by Ruposh et. al classified six basic emotions by Ekman on 1200 Bangla documents from different domains using SVM [29] and obtained 73% accuracy. Most of the previous works are concentrated on three or four classes of emotion through machine learning approaches. In contrast with them, we have developed a deep learning-based system that can detect six types of basic emotions, and it outperforms previous approaches taken on Bangla emotion detection.

Table 5.5 summarizes the previous works done on emotion classification from Bengali texts along with 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% accuracy
Heuristic Features+Conditional Random Field[26]	6	2350	Only 56.45% accuracy
Bag of Words+SVM[29]	6	1200	Achieved 60% accuracy only

To solve the limitations of previous works, we developed a benchmark Bengali emotion dataset(BEmoD) and apply advanced deep learning techniques with proper hyperparameter optimization, to get the best achievable result.

2.7 Implementation Challenges

- For the implementation of this system the most challenging task was to develop a dataset which can be used by our learning algorithm. We collected about 29.29K emotion text and it takes us around six months to prepare this dataset.
- The second challenge was to label the collected data into different emotion category. If we use wrong data to train the classifier, then performance will decrease.
- Hardware support was another vital challenge for our system.

2.8 Conclusion

This chapter contains a thorough summary of the literature review. We briefly discussed on previous works that is already implemented, their limitations and their role on text classification using machine learning and deep learning. The researchers used a variety of feature extraction techniques and classification techniques, which are detailed here. The methodology for the whole system is thoroughly explained in the following sections.

Chapter 3

Methodology

3.1 Introduction

This chapter presents the proposed methodology for textual emotion classification in Bengali and explains in details with its constituents. The corpus development procedure and its statistics also presented in this Chapter. The details description of the training model preparation, feature extraction and selecting optimum hyperparameters included in Chapter 3.

3.2 Corpus Development

Due to the unavailability of standard emotion dataset in Bengali, we developed a corpus. The dataset preparation steps are explained in detail here. These steps are adopted from [30]. The corpus has been developed and initially annotated by five undergrad computer science students and further validated by an NLP expert who is an academician and working on NLP for several years.

Some samples data is shown in table-3.1.

3.2.1 Data Crawling or Accumulation

Bengali text data were accumulated from several sources such as Facebook comments/posts, YouTube comments, online blog posts, storybooks, Bengali novels, daily life conversations, and newspapers. Five participants were assigned to collect data. They manually collected 5700 text expressions over three months. Although most of the data collected from online sources, data can be created by observing people's conversations. In social media, many Bengali native talkers wrote their comments or posts in the form of transliterated Bengali. For example,

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 বাচতে শিখে গিয়েছে এখন (The orphan boy has learned to survive on his own since childhood.)	Sadness
3	মন মেজাজ ভালো নাই, এখন কথা বলতে আসিস না।(I'm not in a good mood, don't come to talk now.)	Anger
4	হারমী শালা, পেয়ে নেই তরে একবার(Bastard, let me find you)	Disgust
5	এ কি??৩ ঘণ্টায় ১৮ বছরের ছেলের এমন অবস্থা কেমনে হল!! (What is this ?? How did such a situation happen to a 16-year-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

a transliterated sentence, "muvita dekhe amar khub valo legeche. ei rokom movi socharacor dekha hoy na." (Bangla: 'মুভিটা দেখে আমার খুব ভাল লেগেছে। এই রকম মুভি সচারাচর দেখা হয় না'[English translation: I really enjoyed watching this movie. Such movies are not commonly seen]. This type of texts demands to be converted phonetically by the phonetic conversion. However, errors may take place during phonetic conversion. For instance, in the above texts, the word "socharacor (English: usually) could be translated in Bengali as, "সছারাচর"after phonetic conversion whereas the accurate word should be "সচরাচর".Therefore, correction should handle because there is no such word like সছারাচরin Bengali Dictionary[31].

3.2.2 Data Preprocessing and Cleaning

Pre-processing performed in two phases: manual and automatic. In the manual phase, "typo" errors eliminate from the collected data. We took Bangla academy supported accessible dictionary (AD) database [31] to find the appropriate form of a word. If a word existed in input texts but not in AD, then this word was considered to be a typo word. The appropriate word searched in AD and the typo word was replaced with this corrected word. For example, the text,"জাহাজ এই প্রথমবারের মতো ওঠা এবং সাগরের মাজখানে দিয়ে যাওয়ার সময়গুলো ম্যজিকাল ছিলো একদম। আহা সৌন্দর্য ❤❤❤" . In this example, the bold words indicate the typo errors that need to be corrected by using AD. After replacing, the sentence turned into "জাহাজ এই প্রথমবারের

মতো উঠা এবং সাগরের মাঝখানে দিয়ে যাওয়ার সময়গুলো ম্যাজিকাল ছিল একদম। আহা সৌন্দর্য ❤❤❤" .

It has been observed that emojis and punctuation marks sometimes create perplexity about the emotional level of the data. That why in the automatic phase, these were eliminated from the manually processed data. We made an emoji to the hex (E2H) dictionary from [32]. Further, all the elements of E2H were converted to Unicode to cross-check them with our corpus text elements. A dictionary was introduced, which contains punctuation marks and special symbols (PSD). Assume any text element matched with elements in E2H or PSD substituted with blank space. All the automatic preprocessing was done with a python-made script. After automatic preprocessing, the above example comes into "জাহাজ এই প্রথমবারের মতো উঠা এবং সাগরের মাঝখানে দিয়ে যাওয়ার সময়গুলো ম্যাজিকাল ছিল একদম আহা সৌন্দর্য".

3.2.2.1 Data Annotation

The whole corpus was labeled manually, followed by majority voting to assign a suitable label. The labeling or annotation tasks were performed by two separate groups (G1 and G2). G1 consists of 5 postgraduate students having a Computer Engineering background and working on NLP. An expert group (G2) consists of three academicians and working on NLP for several years. They performed label verification by selecting an appropriate label. The majority voting mechanism uses to decide the final emotion class of emotion expression.

3.2.2.2 Label Verification

The majority voting by the G1 annotators has decided the original label of data. The original label was considered the ultimate if this label matched the expert label (G2). When the label of G1 and G2 was mismatched, then it was sent to the groups for further discussion. Both groups accepted a total of 4950 data labels among 5700 data. The remaining 750 data was sent for discussion. Both groups are agreed about 250 data label after discussion and added to BEmoD. About 500 data have been precluded due to the disagreement between groups. This exclusion may happen due to the texts with neutral emotion, implicit emotion, and ill-formatted. Holding verifying 5200 data, including their labels saved in *.xlsx format.

3.2.3 Data Statistics

The corpus consists of a total of 29,290 text documents with 538,153 words in total. A different amount of data were accumulated from several sources such as Facebook (2720 texts), YouTube (510 texts), blogs (283 texts), news portals (170 texts), story books (680 texts), novels (568 texts), and conversations (736 texts). In addition to that a total of 23,623 texts are included from four datasets (which are translated to Bengali) such as Affect Data[1], EmoInt[2], Emotion Stimulus[3], and Hashtag Emotion Dataset[4].

Table 3.2 shows the categorical distribution of the developed corpus. The *joy* class contains the most unique words and the *disgust* class contains the least.

Table 3.2: Data distribution in each emotion class

Class	Data	Total Words	Unique Words
anger	3468	67346	11053
disgust	2606	58705	7618
fear	3033	56434	9523
joy	7903	148511	17052
sadness	6987	129361	14988
surprise	5293	77796	14938
total	29,290	538,153	75,172

3.2.4 Data Distribution

Bengali text data are collected from several sources. Figure 3.1 represents the proportion of collected data as per their sources. Majority amount of data (about 62%) collected from the online sources. For example, 27% of data was collected from the Facebook comments and 21% data accumulated from the Facebook posts (21%). Youtube comments, online newspapers and blogs contributed 9%, 3% and 5% respectively. On the other hand, the offline sources contained 38% amount of data into the corpus. A significant proportion of data comes from the novels

[1] http://people.rc.rit.edu/~coagla/affectdata/index.html
[2] http://saifmohammad.com/WebPages/EmotionIntensity-SharedTask.html
[3] http://www.site.uottawa.ca/~diana/resources/emotion_stimulus_data/
[4] http://saifmohammad.com/WebPages/SentimentEmotionLabeledData.html

(9%) whereas the authors contributed only 4% data and 4% accumulated from the daily conversation.

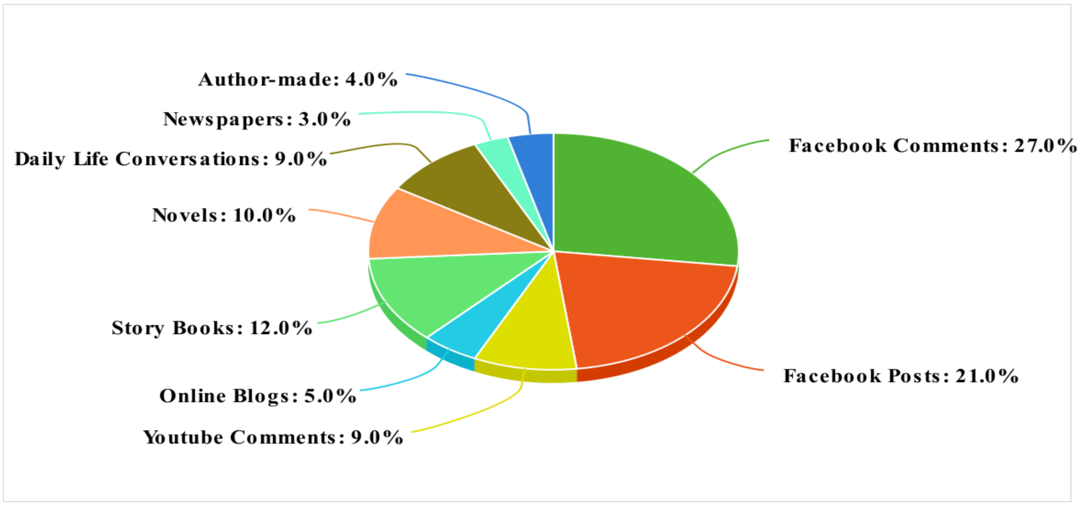


Figure 3.1: Data Collection Sources

3.3 Proposed Framework of Emotion Classification

The main objective of this work is to develop an emotion classification system that can classify or detect the appropriate emotion of a given Bengali text. The proposed classification framework covers four crucial components—word embedding with word2vec, text to feature mapping, training, and prediction. These are covered in the following sections. Figure 3.2 portrays the proposed BiLSTM emotion detection framework.

3.3.1 Embedding Model Preparation

- **Word2Vec**: Word embedding denotes some feature learning techniques where the words of a corpus are mapped to vectors of real numbers. One of the popular embedding techniques is Word2Vec, which is used in this work to generate the embedding model from the emotion corpus(EC). EC consists of texts such as EC = $\{t_1, t_2, t_3,...,t_n\}$ where n is the length of EC and each text t_i has word tokens $w_1, w_2, , w_k$ where k is the length of t_i. The corpus is now converted to a nested list of sentences and tokens that has the structure

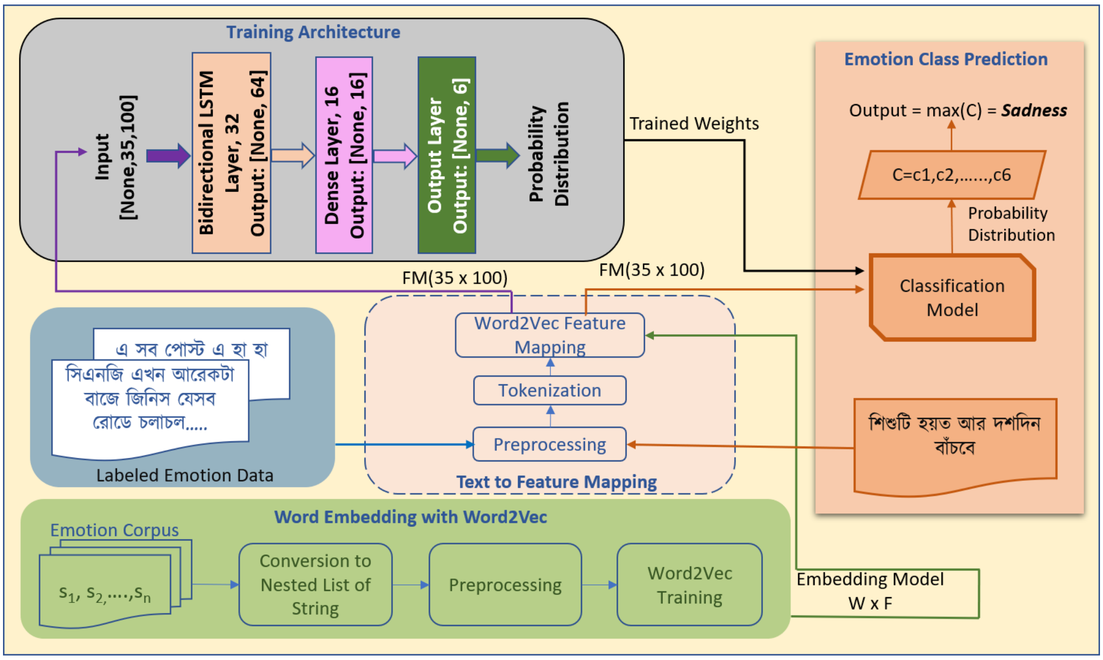


Figure 3.2: Emotion detection framework

[$[tw_{11},\ tw_{12}, \ldots, tw_{1k}], \ldots, [tw_{n1}, tw_{n2}, \ldots, tw_{nk}]$]. This is the required input format for word2vec model generation. The preprocessing step removes all the non-Bengali alphabets, punctuations, special characters, and emoticons from the corpus. The word2Vec algorithm produces an embedding model with the dimension ((W x F) ϵ (5362 x 100)). Here W is the number of unique words in the embedding model, and F denotes the embedding dimension.

- **FastText**:

  This technique uses subword information to find the semantic relationships [8]. In this approach, if a word does not exist in the vocabulary during training, it can be constructed from its constituent n-grams. We trained FastText on Skip-Gram with character n-grams of length 5, windows size of 5, and embedding dimension of 100.

The optimized hyperparameters for Word2Vec and fastText Embedding are reported in Table 3.3

Table 3.3: Optimized hyperparameters for Word2Vec and fastText Embedding

Parameters	**Word2Vec**	**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

3.3.2 Text to Feature Mapping

Labeled text data from the emotion corpus needs to be converted to embedding feature representation as the training cannot be performed on strings. The data passes through the same preprocessing step described in section **??**. Then the tokenization process splits the processed data into a list of words. Word2vec model is designed to extract 100 features from each data, and the length of the data is set to 35 tokens. Word2Vec feature mapping process takes both the embedding model and the list as input and generates a feature matrix FM with dimension (35 x 100). The row and column of the matrix represent the number of tokens(35) and the number of extracted features(100), respectively. Any text data having more than 35 tokens are truncated to 35 tokens and data having less than 35 tokens are zero-padded to 35 tokens.

3.3.3 Classifier Model Preparation

Several ML and DL models are prepared as the baselines including the proposed classifier (Bi-LSTM). The preparation of the models and its several tuned hyper-parameters described in the following:

3.3.3.1 ML Based Models

Four popular ML-based techniques are used to perform the emotion classification task such as LR, SVM, MNB and RF.

- **LR:**

  Logistic Regression(LR) is constructed by using 'lbfgs' solver along with 'l2' penalty. The regularization parameter C is set at 2.

- **SVM:** The Support Vector Machine (SVM) algorithm is a type of supervised machine learning algorithm that can be used to solve a variety of classification problems. For SVM, 'linear' kernel is utilized with random_state value of 0 is chosen.

- **MNB:** Multinomial Naive Bayes (MNB) is a probabilistic classifier that is widely used in machine learning. Bayesian classifiers are both statistical and learnable. '*alpha=1.0*' is chosen for MNB.

- **RF:** The random forest (RF) is a decision tree-based classification algorithm. For RF '*n_estimators*' is set to 100.

A summary of the parameters chosen for ML models are provided 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,

3.3.3.2 DL based Models

Two most commonly used DL models are prepared to perform the emotion classification task in Bengali text including CNN and and a variant of LSTM (i.e., Bi-LSTM).

- **CNN:** Convolutional Neural Network (CNN) [33] is tuned over the emotion corpus. The trained weights from the Word2Vec/FastText embeddings are fed to the embedding layer to generate a sequence matrix. The sequence matrix is then passed to the convolution layer having 64 filters of size 7. The convolution layer's output is max-pooled over time and then transferred to a fully connected layer with 64 neurons. 'ReLU' activation is used in the corresponding layers. Finally, an output layer with softmax activation is used to compute the probability distribution of the classes.

- **Bi-LSTM:** Bidirectional Long-Short Term Memory (BiLSTM) [34] is a variation of recurrent neural network (RNN). BiLSTM captures the semantic meaning of the texts as well as solves the problem to long-term dependency. The developed BiLSTM network consists of an Embedding layer similar to CNN, a BiLSTM layer with 32 hidden units, and a fully connected layer having 16 neurons with 'ReLU' activation. An output layer with 'softmax' activation is used.

Table 3.5 illustrates the details of the hyperparameters used in the DNN models.

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	16	16
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

3.3.4 Emotion Class Prediction

The prediction segment takes an unlabeled short sentence as input and predicts an appropriate emotion class. First of all, the sentence passes through the feature mapping process described in section 3.3.2 and generates a feature map FM of shape(35 x 100). Then FM is fed to the classification model that produces a softmax probability distribution C=$\{c_1, c_2, c_3, ..., c_6\}$. Here c_i denotes the probability of each emotion class. Eq-3.1 is used to calculate the distribution.

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

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.

3.4 Conclusion

This chapter goes into the technique for emotion detection from Bangla texts. The suggested method has been tested for various feature embedding , machine learning and deep learning techniques. This experiment yielded a CNN and RNN based BiLSTM architecture with Word2Vec and FastText feature embedding. The experimental result analysis of the proposed methodology is discussed in the following chapter.

Chapter 4

Implementation

4.1 Introduction

The emotion classification system implementation consist of different modules.In this chapter we will see the implementation details of this modules. Finally we will see sample input and output of our system.

4.2 System Requirements

4.2.1 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

4.2.2 Software Requirements

textcolorredMove this part in the Implementation Chapter 4 We implement our system in a specific software environment. Required software lists are,

- 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

All of this softwares are not required at a time. To implement different part of our system this softwares were used. To test our system one just need an IDE where the system could be run.

4.3 Keras Framework

Keras is a python-based deep learning platform that is open source. Francois Chollet, a Google artificial intelligence researcher, came up with the idea. To setup Keras we must fulfill the following requirements –

- Any kind of OS (Windows, Linux or Mac)
- Python version 3.5 or higher.

Installation Steps of Keras:

Step 1: Create virtual environment

Step 2: Activate the environment

Step 3: Python libraries Keras depends on the following python libraries.

- Numpy
- Pandas
- Scikit-learn
- Matplotlib
- Scipy
- Seaborn

Step 4: Finally install using pip: *pip install keras*

4.4 System Set up and Runing

We set up the whole environment in python jupyter-lab. Keras with Tensorflow-gpu backend is installed initially. The training architecture was trained utilizing the gpu. The output model is saved in local disk to be tested with the test data.

4.4.1 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 can have a great impact on social and environment as well as on ethics. As we are recognising emotion from Bengali texts by means of automatic system this has a huge impact on society. They are briefly discussed in this section.

4.5.1 Social and Environmental Impact

The proposed system can detect emotion from Bengali text. This system can be used to detect and prevent suicidal thoughts, criminal behaviours from the writings on social media and thus contribute to the society.

4.5.2 Ethical Impact

The ethical issues are if this work is used in wrong, it will hamper public privacy. Moral obligation before using this work is mandatory.

4.6 Conclusion

This chapter summarizes the implementation methods, software and hardware setup. Also the social, environmental and ethical impacts are discussed in this chapter. In the next chapter the evaluation result of the proposed methodology will be discussed.

Chapter 5

Results and Discussions

5.1 Introduction

Chapter 5 explains the details experimental analysis on developed corpus for various ML and DL methods. This Chapter also investigates the performance of implemented models with various evaluation measures (such as precision, recall, accuracy, f_1-score). The details comparative analysis among various methods with existing techniques also illustrated in this Chapter. For better insight, a details error analysis of the proposed model is included at end of this Chapter.

5.2 Experiments

We conducted different machine learning and deep learning approached with various word embedding methods for building the emotion detection system. We also tuned the hyperparameters to get the best out of each model. The experimental data, evaluation measures and the results along with analysis of errors are discussed in details in the following sections.

5.2.1 Experimental Data

Presets here the data split up in a table: train/test and validation set with proper explanation. Shows the each set distribution with class-wise and sub-class wise.

The total dataset was divided into 3 portions as train, validation, and test set. The train and test set split ratio was 90:10. The train set is further splitted into train and validation set with the ratio 85:15. Table 5.1 shows train/validation/test set distribution for each class. The training data was passed to the proposed network and trained with 15 epochs. The validation data was used to validate the model in each epoch.

Table 5.1: Classwise train/validation/test set distribution

Class	**Train**	**Validation**	**Test**
Anger	2729	381	360
Disgust	2028	291	287
Fear	2377	327	329
Joy	6082	967	854
Sadness	5379	804	804
Surprise	4085	620	589
Total	22680	3390	3223

5.3 Evaluation Measures

The proposed emotion detection system was evaluated two different phases: training phase and test phase. Several measures such as confusion matrix (CM), accuracy (Acc), loss, precision (P), recall (R), and F_1-score are considered for model's evaluation.

- **Accuracy (Acc)**: The accuracy(Acc) is calculated using Eq-5.1.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.1}$$

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

- **Loss**: We used the 'logcosh' loss function, which is the logarithm of the hyperbolic cosine of the prediction error. The loss is calculated using Eq-5.2.

$$Loss = \frac{1}{n}\sum_{k=1}^{n} log(cosh(y^{(k)} - \bar{y}^{(k)})) \tag{5.2}$$

  where y and $\bar{y}$ are the target class and 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. It indicates that logcosh behaves mostly like the mean squared error, only with the difference that, the occasional wildly incorrect prediction won't affect it so strongly.

- **Confusion Matrix (CF)**: A confusion matrix is a table used to assess a classification model's results. Our system's confusion matrix has two rows and two columns since we use a binary classification model. The number of false positives, false negatives, true positives, and true negatives are all recorded in this matrix.

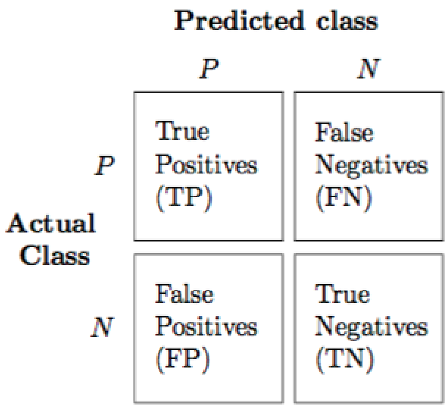


Figure 5.1: Confusion Matrix

  The figure represents the confusion matrix where

  - True Positive(TP): Number of documents that is of class A and also classified as Class A.
  - True Negative(TN): Number of documents that is not from Class A and also classified as non Class A.
  - False Negative(FN): Number of documents that is from Class A but classified as non Class A.
  - False Positive(FP): Number of documents that is not from Class A but classified as Class A.

- **Precision (P)**: Precision refers as a predictor of positive outcomes. That is the proportion of correctly classified positive cases to the total number of positive cases. The following equation can be used to calculate precision.

$$\texttt{Precision} = \frac{TP}{TP + FP} \tag{5.3}$$

  If precision is high then the algorithm is doing well.

- **Recall (R)**: Recall is the ratio of correctly classified positive instances to the total number of positive instances. It is also called true positive rate. Recall can be obtained from the following equation.

$$\texttt{Recall} = \frac{TP}{TP + FN} \tag{5.4}$$

High precision and high Recall is essential for a model. Unfortunately there is a trade-off between precision and recall. That is, improving precision typically reduces recall and vice versa.

- If threshold of a classifier is increased then it causes high precision and low recall.
- If threshold of a classifier is decreased then it causes low precision and high recall.

To sort out this problem we need another measure that is F_1measure.

- F_1 **score**: Precision and recall are weighted in the F_1 score. As a result, this score considers both false positives and false negatives. In most cases, F_1 is more valuable than accuracy. When false positives and false negatives have comparable costs, accuracy works better. It's best to look both at Precision and Recall if the cost of false positives and false negatives is quite different. To chose a learning algorithm between several algorithms we have to find F_1 Score of algorithms. The algorithm which has highest F_1 score value is chosen as our learning algorithm. The following equation can be used to calculate the F1-score:

$$F_1 = \frac{2 * precision * recall}{precision + recall} \tag{5.5}$$

- **Kappa Score (κ)**:

Cohen's kappa measures the agreement between two annotators, each classifying N items into C categories which are mutually exclusive. This coefficient determines how well one annotator agrees with another annotator. If the annotators are in complete agreement then $\kappa = 1$ but if they disagree, $\kappa \leq 0$. The κ coefficient defined by the Eq. 5.6.

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

where p_0 represents relative observed agreement which measured by the ratio of number in agreement and the Eq. 5.7 determines the hypothetical probability of chance agreement.

$$p_e = 1/N^2 \sum_k n_{k1} n_{k2} \tag{5.7}$$

k is the number of categories, N is the number of observations to categorize and n_{ki} is the number of times that the annotator i predicted category k.

- **Jacard Similarity**:

  The Jaccard index, also known as the Jaccard similarity coefficient, is a metric that is used to determine how similar and diverse sample sets are. It can be calculated using following equation, where A and B are two distinct document:

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

5.4 Results Analysis

The analysis of developed dataset, evaluation results of the training and testing models and a details errors analysis is presented in the subsequent subsections.

5.4.1 Analysis of Dataset

We evaluated the inter-annotator agreement to ensure the quality of the annotation using the coding reliability [35] and Cohen's kappa [36] scores. An inter-coder reliability of 98.1% and 0.977 Cohen Kappa Score has been achieved on the corpus that reflects the quality of the dataset.

Figure 5.2 represents the number of texts vs the length of texts distribution for each class of the collected corpus. Investigating the figure we can see that, most of the data has a length between 15 to 35. Another interesting fact is discovered that, to express some *'Disgust'* emotion relatively more words are used .

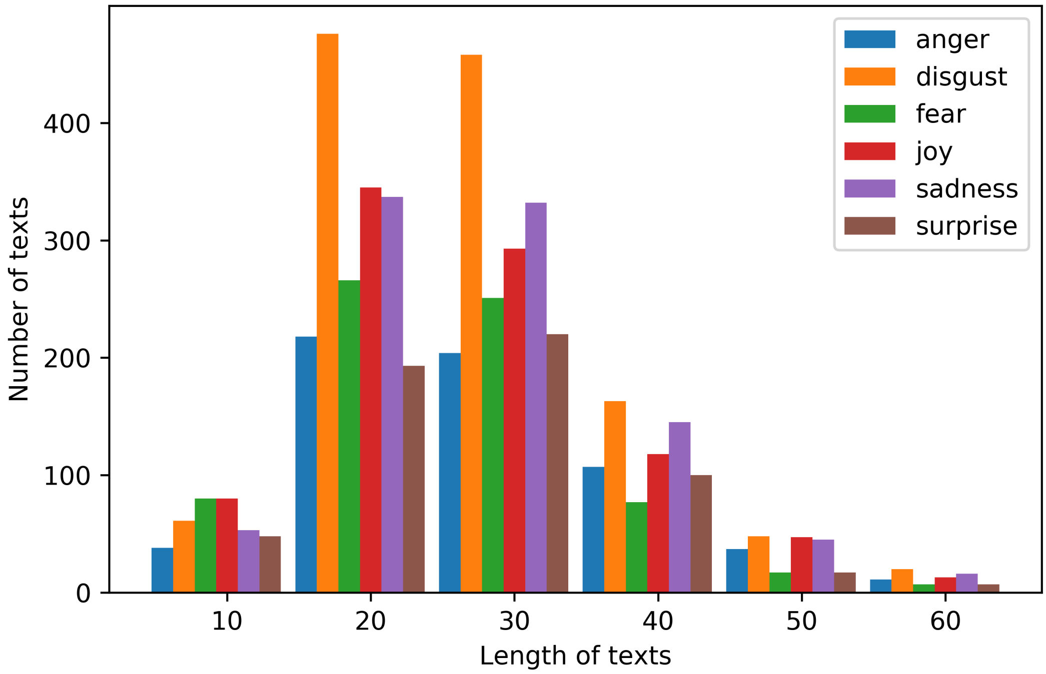


Figure 5.2: Number of texts vs length of texts distribution of the corpus

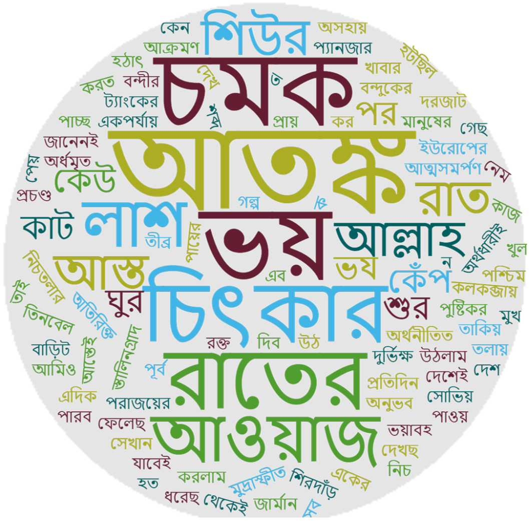
By analyzing the corpus we found it more challenging as well as interesting. A *Jaccard* similarity among the classes have been calculated using 200 most frequent words from each emotion class. The result is given in table 5.2. The *Anger-Disgust* and *Joy-Surprise* pairs hold the highest similarity of 0.58 and 0.51, respectively. These similarity issues can substantially effect the emotion classification task.

Table 5.2: *Jaccard* similarity between the pair of the emotion classes. Anger(c1), disgust(c2), fear(c3), joy(c4), sadness(c5), surprise(c6).

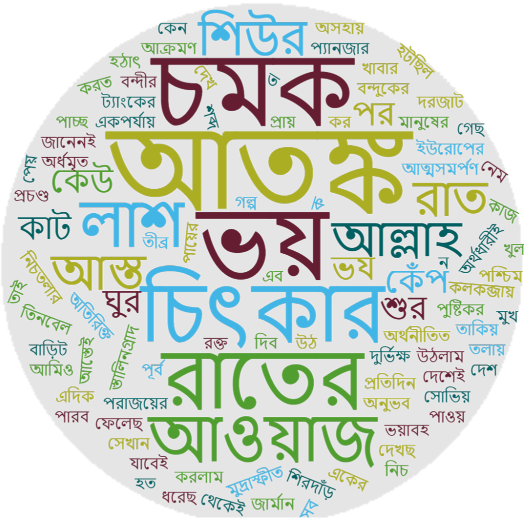
	C1	**C2**	**C3**	**C4**	**C5**	**C6**
C1	1.00	0.58	0.40	0.43	0.45	0.47
C2	-	1.00	0.41	0.45	0.47	0.44
C3	-	-	1.00	0.37	0.45	0.46
C4	-	-	-	1.00	0.47	0.51
C5	-	-	-	-	1.00	0.48

Wordcloud representation for each emotion class is drawn in Figure 5.3. Here the words are highlighted according to their frequencies. The frequent words can give significant information to distinguish the emotion classes.

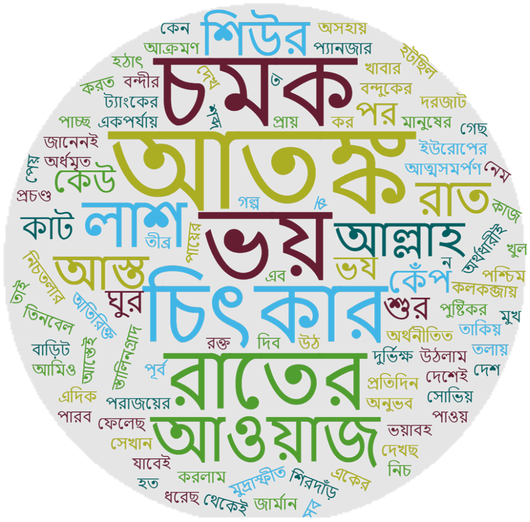
(a) Anger



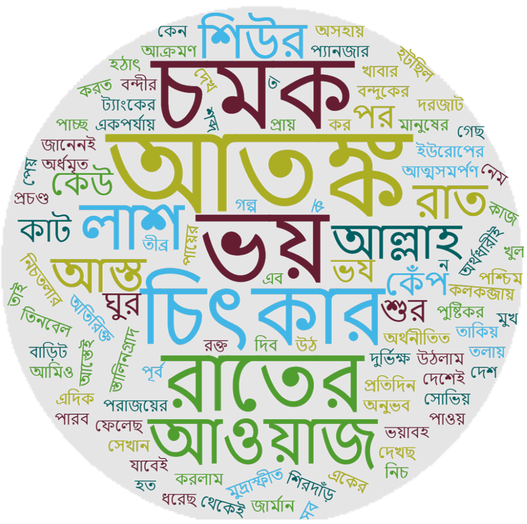
(b) Disgust



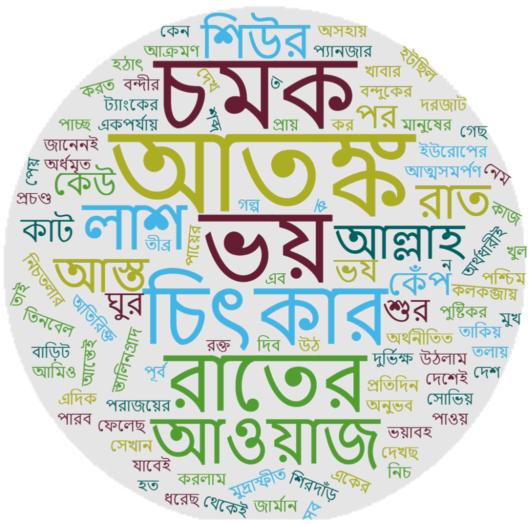
(c) Fear



(d) Joy



(e) Sadness



(f) Surprise

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

5.4.2 Training Phase Evaluation

Several measures such as accuracy (A), precision (P), recall (R), and F_1-score (F_1) are used as the evaluation measures.

From fig-5.4, it can be seen that the model achieved about 89% training and almost 75% validation accuracy. The minimum training and validation loss acquired is 0.013 and 0.028, respectively. Through the validation curve, it is observed that the model was saturated after the 10th epoch and so it was not trained further.

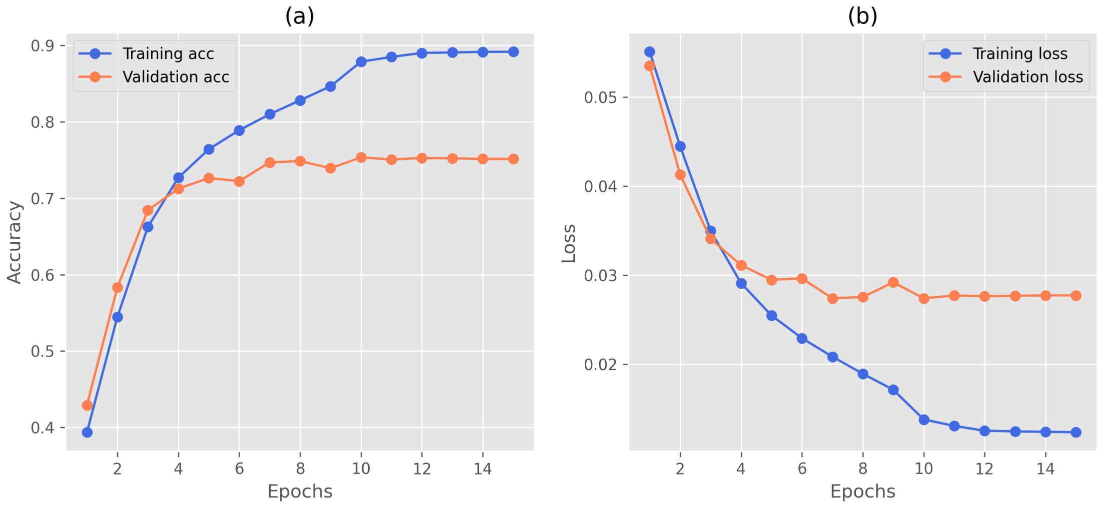


Figure 5.4: Effect of epochs on (a)Training & validation Accuracy and (b) Training & validation loss

5.4.3 Testing Phase Evaluation

The proposed model performance compared with the different embedding techniques with the developed corpus. Five embedding techniques are considered for comparison such as two variations of Word2Vec [6], two variations of Fasttext [37], and Keras Embedding layer[1]. Table 5.3 demonstrates the results of the comparison. The results indicate that the proposed model (Word2Vec (skip-gram) with BiLSTM) achieved the better accuracy (74.7%) than the other methods.

Table 5.4 represents the performance of the proposed model in each emotion class. The total amount of test data represented in the rightmost column of the table. Results indicates that the emotion category *joy* obtained the highest F_1 score,

[1] `https://keras.io/api/layers/core_layers/embedding/`

Table 5.3: Performance comparison among embedding techniques

Embedding Techniques	Accuracy(%)	
	CNN	BiLSTM
Word2Vec (CBOW)[**Proposed**]	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	73.30	74.35

and *fear* achieved the least. Some of the *fear* class data is classified as *sadness*. It occurred 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 Word2Vec+Bi-LSTM

Class	F_1(%)	P(%)	R(%)	A (%)
Anger	63.7	64.7	62.8	64.30
Disgust	80.8	78.6	83.0	83.02
Fear	60.8	62.3	59.5	59.52
Joy	81.6	81.2	82.2	82.27
Sadness	71.4	70.9	72.0	71.98
Surprise	78.3	79.3	77.4	77.38
Avg.	73.36	72.83	72.81	73.57

5.4.4 Comparisons with Baselines

To validate the assessment, the proposed BiLSTM based model compared with the existing techniques of emotion classifications. We implemented previous methods using our developed corpus and reported outcomes. Table 5.5 shows a summary of the comparison. The results revealed that the proposed model achieved the highest performance scores than the past techniques. Figure 5.5 show the accuracies of different approaches at a glance where BiLSTM+Word2Vec has the higest accuracy value.

5.4.5 Error Analysis

It is evident from Table 5.3 that Word2Vec (skip-gram) with BiLSTM is the best performing model to classify emotion from Bengali texts. A detailed error analysis is performed using the confusion matrix. Figure 5.6 illustrates a 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
Word2Vec + BiLSTM (**Proposed**)	**74.7**	74.3	74.65	73.9

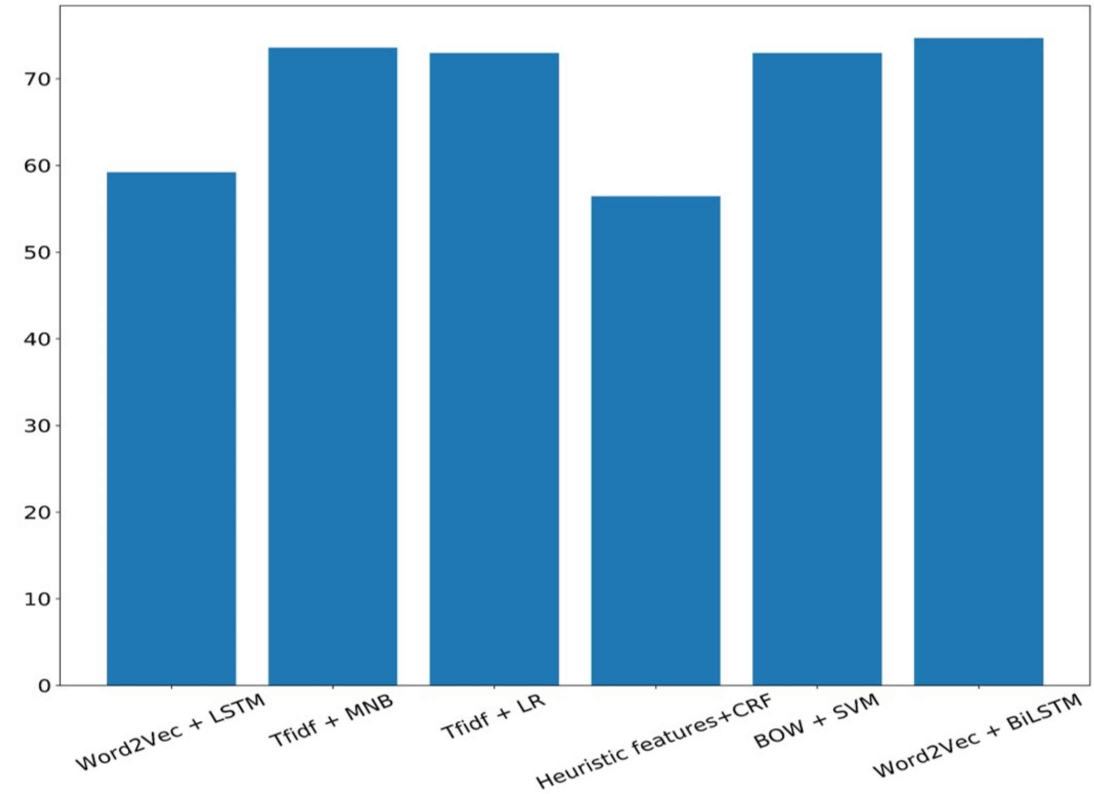


Figure 5.5: Accuracy of different approaches

The *joy* class gets the most correct predictions from the other classes, i.e., 659 from 802 data was predicted correctly. The *joy* class has maximum data, and the model performed well in this class, as deep learning models required a large amount of data to give better performance. In the *fear* class, 172 from 289 data was predicted accurately. 53 data from the fear class was predicted as *sadness* as the context in *fear* sometimes drives to some sad consequences.

The possible reason for incorrect predictions might be the class imbalance nature of corpus. However, the high *Jaccard similarity*(Table 5.2) also reveals some reasons. Few words are used multi-purposely in multiple classes. For instance, hate words can be used to express both *anger* and *disgust* feelings. Moreover,

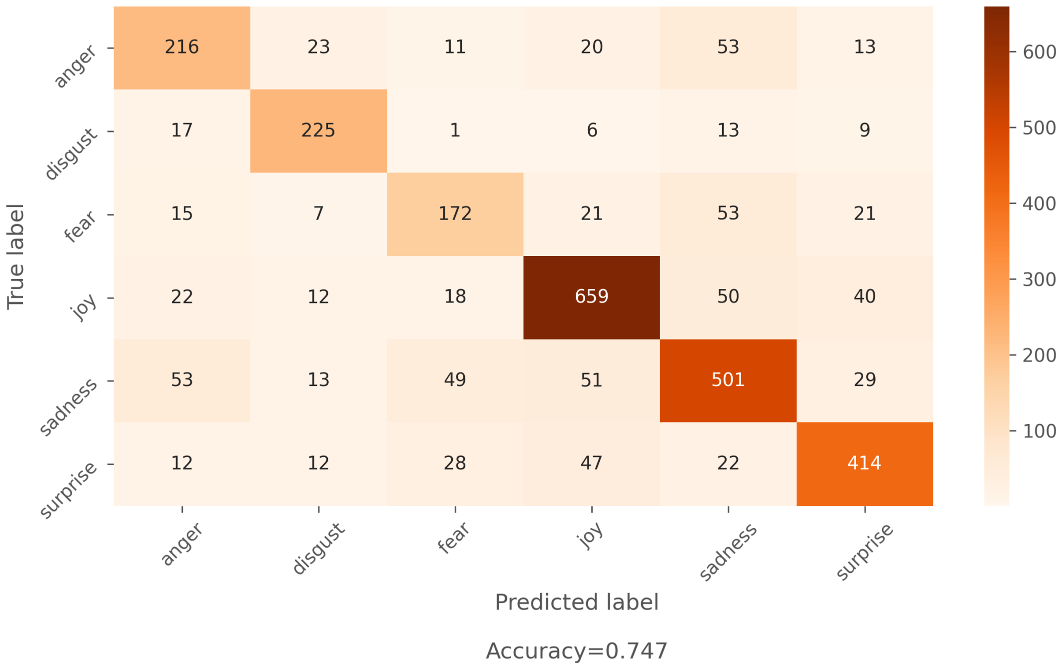


Figure 5.6: Confusion matrix on the test data

emotion classification is highly subjective, depends on the individual's perception, and people may contemplate a sentence in many ways. Thus, by developing a balanced dataset with diverse data incorrect predictions might be reduced to some extent.

For better insights, we investigate a few input for their actual class and predicted class by the best three models including the proposed method. Table 5.6 shows the actual and predicted output.

5.5 Discussion

Emotion recognition is one of the very crucial tasks in the arena of NLP. The results we obtained through the system is not perfect but has some valuable insights that might help future researches in this branch. We figured out imbalanced dataset and similarity among various emotion classes contributed to the accuracy reduction. As emotion is a matter of perception, sequential deep learning models should perform better than the traditional machine learning approaches. Word2Vec with BiLSTM network achieved 74.7% 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 মিয়া 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 ১৪৯টি যৌন নিপীড়নের ঘটনা লিপিবদ্ধ করা হয়েছে	Disgust	Yes	No	Yes	No	Yes	No
কি জানি আমার তো ভয় কমতেছে না অনেক ভয় লাগে	Fear	No	Yes	No	Yes	No	Yes
পরে বুঝেছি যা দেখেছি সব সত্যি দেখেছি আর তখন স্যার ভীষন ভয় পেয়েছিস্যার 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
দারুন 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 matter of perception. We have built a dataset and analyzed it in many angles. The developed dataset is then used to classify six types of emotion classes. After investigating several methods we came up with the conclusion that Word2Vec word embedding model for feature embedding, and RNN based BiLSTM network for training the model achieved the best result in this particular task. Other algorithms also performed quiet well. Increase in the number of training documents may increase efficiency of this algorithms.

Chapter 6

Conclusion

This Chapter summarizes the thesis with highlighting the major contributions of this work including a few weaknesses in the current implementation. This Chapter also provides the few recommendations for further improvements of the proposed system.

6.1 Conclusion

This thesis work investigated various ML and DL techniques to classify the emotion in Bengali texts. Due to the scarcity of benchmark corpus, we developed a corpus containing 29.29k Bengali texts labelled with six basic classes. Co-hen's Kappa score of 0.9177 reflects the quality of the corpus. A massive amount of Bangla text data is being generated online everyday. Almost all of them indicate some kind of emotion. Besides Bangla emotion detection is relatively new work in the Bangla language processing field. This motivated us to design an emotion detection system. Among various techniques Word2Vec word embedding model for feature embedding, and RNN based BiLSTM network for training the model gives superior result achieving 74.7% accuracy.

6.1.1 Limitations

Our system has some limitations. They are pointed out below:

- Only six emotion classes has been investigated, more could be considered.
- Mis-classification occured due to less data.
- The classes are imbalanced.

- Hyper-parameters are tuned manually, automated approach could be considered.

6.2 Future Recommendations

This work investigated various aspect of emotion classification task. But more improvements can be made in future works. Future improvements can be summarized as follows:

- Incorporating more classes such as love, hate, and stress.
- Including more data at corpus.
- Detecting Neutral and multi-emotion classes..
- Implementing transformer variants.

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 emotions 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, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, 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. O. 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:1301.3781*, 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 natural 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 Computational Linguistics*, vol. 5, pp. 135–146, 2017 (cit. on pp. 9, 24).

[9] J.-Y. Yoo and D. Yang, 'Classification scheme of unstructured text document 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 Computational Linguistics: Human 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 (SemEval-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 conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, IEEE, 2017, pp. 79–80 (cit. on p. 15).

[18] S. Alzu'bi, O. 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 Security (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 Journal 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: Emocontext 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 naïve bayes classifier,' in *2019 4th International Conference on Electrical*

Information and Communication 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, Engineering 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 analyzing 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 Advances in Electrical Engineering (ICAEE)*, IEEE, 2019, pp. 570–574 (cit. on pp. 17, 42).

[30] A. Das, M. A. Iqbal, O. 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://accessibledictionary.gov.bd/` (accessed on 2 January 2020) (cit. on p. 20).

[32] *Full emoji list*, `https://unicode.org/emoji/charts/full-emoji-list.html` (accessed on 7 Februray 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 Measures*, 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 Measurement*, 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, 'Advances in pre-training distributed word representations,' in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018 (cit. on p. 40).