Bachelor of Science in Computer Science & Engineering

A Machine Translation Framework for Translating English Sentences into Bangla Class Dialect Sentences

by

Abu Naim Khan

ID: 1504080

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

May, 2021

A Machine Translation Framework for Translating English Sentences into Bangla Class Dialect Sentences

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

by

Abu Naim Khan
ID: 1504080

Supervised by

Dr. Mohammad Ibrahim Khan
Professor
Department of Computer Science & Engineering

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

The thesis titled '**A Machine Translation Framework for Translating English Sentences into Bangla Class Dialect Sentences**' submitted by ID: 1504080, Session 2020-2021 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. Mohammad Ibrahim Khan
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

____________________________________ Member (Ex-Officio)

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

____________________________________ Member (External)

Dr. Kaushik Deb
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.

Signature of the candidate

Date:

Acknowledgements

First and foremost, I would like to thank Almighty Allah for giving me the strength to finish this work. The satisfaction that accompanies the successful completion of this thesis would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. I am grateful to my honorable thesis Supervisor Dr. Mohammad Ibrahim Khan, Professor, Department of Computer Science and Engineering, Chittagong University of Engineering and Technology(CUET). Without his guidance, it would not be possible for me to complete this task. I would also like to extend my gratitude to all of my teachers for their valuable guidance in every step of mine during the four years of the learning stage. Finally, I would like to thank my family, friends, seniors, juniors, and the staff of the department for their valuable suggestions and assistance that has helped in the successful completion of the project

Abstract

Machine translation (MT) is a computer program that converts text from one language to another. To implement this work, we used Recurrent neural network.Recurrent neural networks are rapidly being employed in machine translation in natural language processing. The Bangla language, in addition to other languages, has a wide vocabulary. Improved English to Bangla machine translation would make a substantial contribution to the processing of the Bangla language. We had to create a data collection with several types of Class dialects. We have given a comparison of neural network accuracy using different features in this work, which will be useful to others. The architecture of an English to Bangla machine translation system is described in this work. The encoder-decoder recurrent neural network was used to create the system. For the mapping of English and Bangla terms, the model employs a knowledge-based context vector. The experiment shows average accuracy of 86 percent for Recurrent Neural Network. By increasing the dataset size, the system's accuracy can be improved.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Bangladesh is a land of mystery. There is eight-division in Bangladesh. Dialects vary by geographical location. In Chittagong Sylhet Hill Tracts, there are lots of indigenous people and they have their languages. This language is also considered a dialect. All of the languages are related to other languages around the world in some way, and many of them are closely linked to Bengali. According to MRGI, more than 2 million indigenous people are live here and they use their languages[1]. These people are the citizen of Bangladesh. So it is very essential to develop a media that can help to understand their languages.

In the process of computerization of any language, one of the most important tasks is to develop an efficient and accurate machine translation framework. Despite the wide use and popularity of the Bangla language, there exists very little research for overall Bangla language processing, compared with other popular languages such as English.

The overview of the framework will be described in this chapter, along with the inherent difficulties of this problem. Motivation and contribution of this specific thesis will be discribed in this chapter

1.1.1 Natural Language Processing (NLP)

Natural Language Processing is one of the major branch of artificial intelligence that deals with the interaction between computer and human. Usually known as NLP, it helps creating system for computers in understanding human language.It usually works by applying identification algorithm and extracting rules from the data so that raw language data can be converted in a form that computers can understand. It is the driving force in many application such as google translate, word process, artificial personal assistant such as Siri, Cortana etc.

1.1.2 Bangla Language Processing (BLP)

Bangla language processing is a sub field of natural language processing. Bangla is a language for which the nation sacrificed their lives. In 1952, through language movement we have found Bangla as our mother language and it is recognized asInternational Mother Language day. Bangla is spoken by about 245 million people of Bangladesh and two states of India. Today, most of the computer based resources and technical journals are in English. Due to the language barrier, the common masses face big obstacle to enjoy the optimum benefits of modern communication and information technology (ICT) as well as huge enriched English knowledge database around the globe. Language processing in mother tongue is the only technology that can be used to remove this barrier. The research work on Bangla language processing (BLP) was started in late 1980s in Bangladesh and it already produced some tangible results. The success of BLP may have a huge impact particularly to learn and use ICT in Bangla for the common people which will enhance their socio-economic life greatly

1.2 Framework Overview

In the field of machine translation in natural language processing Recurrent Neural Network (RNN) is one of the most advanced technology. The RNN architecture is made up of two parts: an encoder network that absorbs input text and a decoder network that produces translated output text.The recurrent neural network (RNN) model takes sequential input[2]. The encoder's task is to extract a dense representation of the various length input texts in a fixed size. Based on the dense representation, the decoder's task is to produce the corresponding text in the destination language (fig. 1.1) using both RNNs and LSTM network[3].

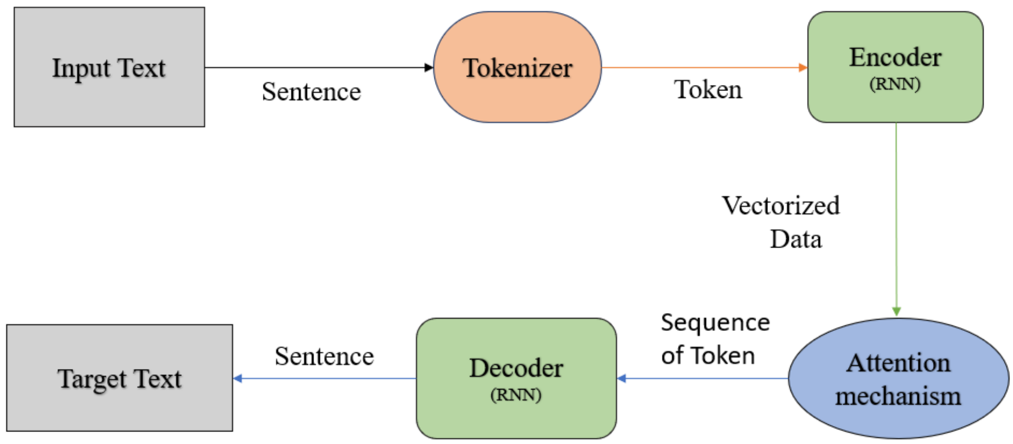

Figure 1.1: Block diagram of RNN Machine translation Framework.

1.3 Difficulties

Lacking of standard resources and datasets are the main obstacle of this work. This type of work hardly seen in Bangla language. So, it is tough to collect datasets and works through it. For the implementation of the system was to develop a dataset which can be used by our learning algorithm. We know that for any machine learning algorithm a well-furnished dataset a key. If the dataset is accurate then the accuracy of the output becomes perfect. Bangla is low resource language. We try to overcome this problem by collecting a large amount of text from different Bangla sources.Characters of Bangla language which is completely different from English or similar language style. Because it is used Sanskrit based style. There are 50 basic characters, 10 numerals, and over 260 compound characters. Along with the class, dialects are partially different from the Bengali language, so it is difficult to describe their sound with the Bangla alphabet. The major problems are described below :

- The major problem is to collect the Dataset of class dialect.Because there is no available Dataset.
- To communicate with tribal is so difficult for collecting data.
- Complex Structure of Bangla class dialect sentences.

1.4 Applications

Machine Translation is a long studied topic in natural language process. There are many aspects in modern world which can be made efficient with the help of machine translation technology. Some of them are,

- The tranlation from English to Bangla class dialects.
- Automation of any kind of registration system for indigenous people.

1.5 Motivation

Motivation for working on Bangla language arises from the story of sacrificed life for a language. Bangla has a rich historical and cultural background. To keep Bangla history, culture, Literature existent and to introduce it globally we have to digitize the Bangla language. Bangla is the fourth largest language in the world and despite having over 245 million native speakers still, now Bangla language has a negligible amount of work on authorship detection. Our Bangladeshi people have dreamed to make a digitized Bangladesh. The government of Bangladesh is highly concentrated to make digitization successful. Without Bangla Language Processing (BLP) the digitization of a nation is not possible. The research work for Bangla language processing Although, there are lots of works [survey] in Bangla Language Processing like in the field of syntax analysis, semantic analysis, machine translation, speech recognition, character recognition, there are few works on stylometric analysis in Bangla. Researchers in the past have worked with famous English writers like Shakespeare, Jane Austen, and Charles Dickens, etc. However, three has not been any analysis on Bengali writers. There are many talented and skillful writers in our country and we often find interesting styles and patterns in their writings. By analyzing their writing we can distinguish among the characteristics and features of individual writers. Stylogenetics helps to detect who is the actual author of specific writing and may assist in identifying fraudulent writing or in cases where multiple writers claim the ownership of a particular writing. For this reason, we are intended to work on the machine translation of the Bangla class dialect.

1.6 Contribution of the thesis

The purpose of a thesis or research project is to fulfill a certain set of objectives, such as defining a new methodology or improving an existing one. In this thesis, the main focus was given to develop a framework to translate Bangla class dialect. The primary contribution of this thesis is the following:

- Developed the data set of various class dialects from different indigenous people groups.
- Implemented the system with the encoder-decoder recurrent neural network that uses a knowledge-based context vector for the mapping of English and Bangla words.
- Evaluation of the overall performance measure of the proposed method.

1.7 Thesis Organization

This thesis is divided into five chapters. This chapter briefly discussed a general overview of topic related to this thesis. In addition, the motivation and objectives of this thesis are presented. In the next chapter 2, an overview of our project related terminologies related to the project and contains brief discussion on previous works that is already implemented with their limitations. Chapter 3 describes elaborately the working procedure of our proposed system with appropriate figure and tables. We also explain authorship detection mechanism with appropriate iteration and figure. we have illustrated our implementation of the project and explain the implementation step by step. The graphical representation, abstract view of the system is explained here with necessary figures. In this chapter we also specify the system requirements of the proposed model. Chapter 4 focuses on the experimental result of the proposed system. In order to evaluate the system, we have used subjective as well as quantitative measures. In this chapter a shown effective performance analysis system of language processing. The thesis concludes with a summary of research contributions and future plan of our work in chapter 5.

1.8 Conclusion

In this chapter, an overview of Bangla class languages and Bangla machine translation is provided. Along with the difficulties, the summary of the Bangla machine translation framework using recurrent Neural Network is described in this chapter. The motivation behind this work and contributions are also stated here. In the next chapter, background and present state of the problem will be provided.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we will shortly describe the history of machine translation and learn about different machine translation techniques which are useful for classifying text. This chapter also contains a brief discussion on related previous works.

Machine translation (MT) is an application of computers that translating text from one language to another. The primary objective of Bangla class dialect machine translation is to build a system that evaluates, recognizes, and generates natural human languages, with the intention of ultimately be able to address the computer as if it were another person. The translation of language from one language is not an easy task. There are so many factors that can be effected on translating.

Machine Translation, also known as robotized interpretation, is a process in which a computer program interprets text from one language to another without the involvement of a human. Machine translation, at its most basic level, is the simple replacement of atomic words in one distinctive language with words in another. Machine translation tecnique comes in a variety of forms.

- Statistical Machine Translation or SMT.
- Rule-based Machine Translation or RBMT.
- Hybrid Machine Translation or HMT.
- Neural Machine Translation or NMT.

2.2 Related Literature Review

Machine translation from English to Bangla is still in its infancy. There is currently no work being done to translate English into the Bangla class dialect. However, many translations from Bangla to English and vice versa have been completed.

A few studies have been conducted on the creation of dictionaries[4].Hoque et al.[5] proposes a technique to parse sentences using CFG rules.

Statistical approaches for fixing the Bangla G2P issue are relatively uncommon. Google has unveiled their Bangla version of their search engine. The year of the Text-to-Speech (TTS) system [6] is 2016. Machine learning was used to create a G2P model. They're compiled a lexicon of 65,000 terms, 37,000 of these were utilized to train the G2P model. Chowdhury et al.[7] were able to attain a WER of 18.5 percent.

Natural language processing is used to make machines intelligent. Language translation is improving, however machine translations from English to Bangla have not improved significantly. Many machine translation solutions have been proposed by researchers. Two encoder-decoder neural machine translation architectures, convolutional sequence and neural machine translation, are employed for English to Hindi translations[8] The Nematus framework was used to train the RNNS2S model, while the ConvS2S model was used to train the ConvS2S model. Fairseq-5, an open-source library created by Fairseq, was used to train. Convolutional neural networks are being used by Facebook for neural machine translation. Networks that use a convolutional neural network (CNN) or a recurrent neural network (RNN). The end product is ConvS2S performed better on English to Hindi translations. a translation that might assist us in resolving our issue. The translations are solved using a corpus-based technique that uses one topic file and one verb file. Each topic has a flag associated with its verb, and the most appropriate and meaningful sentences are chosen for final translations. When compared to Google Translator[9], the results were better.

A feed-forward back-propagation artificial neural network was employed for another English to Hindi translation [10] . Apart from the neural network model, which was built in Matlab, java was used as the main programming language to implement the rules and all of the modules.The Encoder, which is primarily developed in Java, encodes the training data into numeric form. To calculate the system's score, they employed BLEU [11] . The training models are also tested using BLEU scores. RNN encoder-decoder framework methods are used as another way for English to French neural machine translation [12],In order to implement both of the models, specific training processes and datasets are used. The RNN search produced a better result than the traditional RNN encoding after the test.

2.3 Conclusion

A detailed review is discussed in this chapter. This discussion has been separated into basic components of translating English sentences to another language for ease of use. The researchers' feature extraction methodologies and classifier are explained in this section. The proposed methodology of the machine translation framework of English into bangla dialects is explained in detailed in the following chapter.

Chapter 3

Methodology

3.1 Introduction

Bangla is one of the most difficult languages in the world. Furthermore, the characters of this language are intrinsically cursive and depict a complex shape.In this chapter, we will discuss about our proposed methodology and will try to explain each module of the system. We will also try to discuss our used algorithms. In the end entire system will be reviewed with an example.

3.2 Overview of Framework

Our proposed system takes English sentences as input and the output of the system is probable Bangla class dialect sentences. For this system to work we have normalized and then sampling the data set then divided the total data for training and testing. Then input data is passed through an encoder that vectorizes the input and then vectorized data passed through an attention model to predict the sentences. Finally, another RNN is used as a decoder to get the final output. Here we have given an overall system overview in figure 3.1 and figure 3.2.

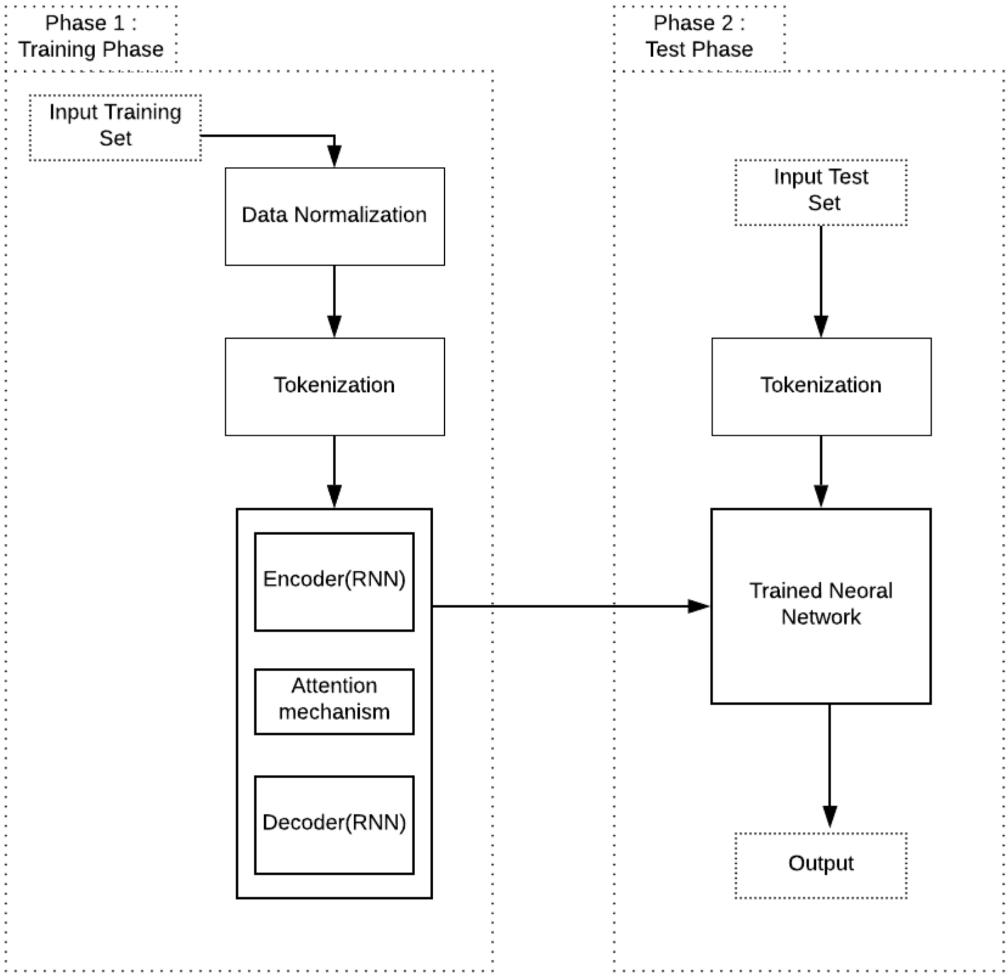

Figure 3.1: Abstract view of Proposed Framework.

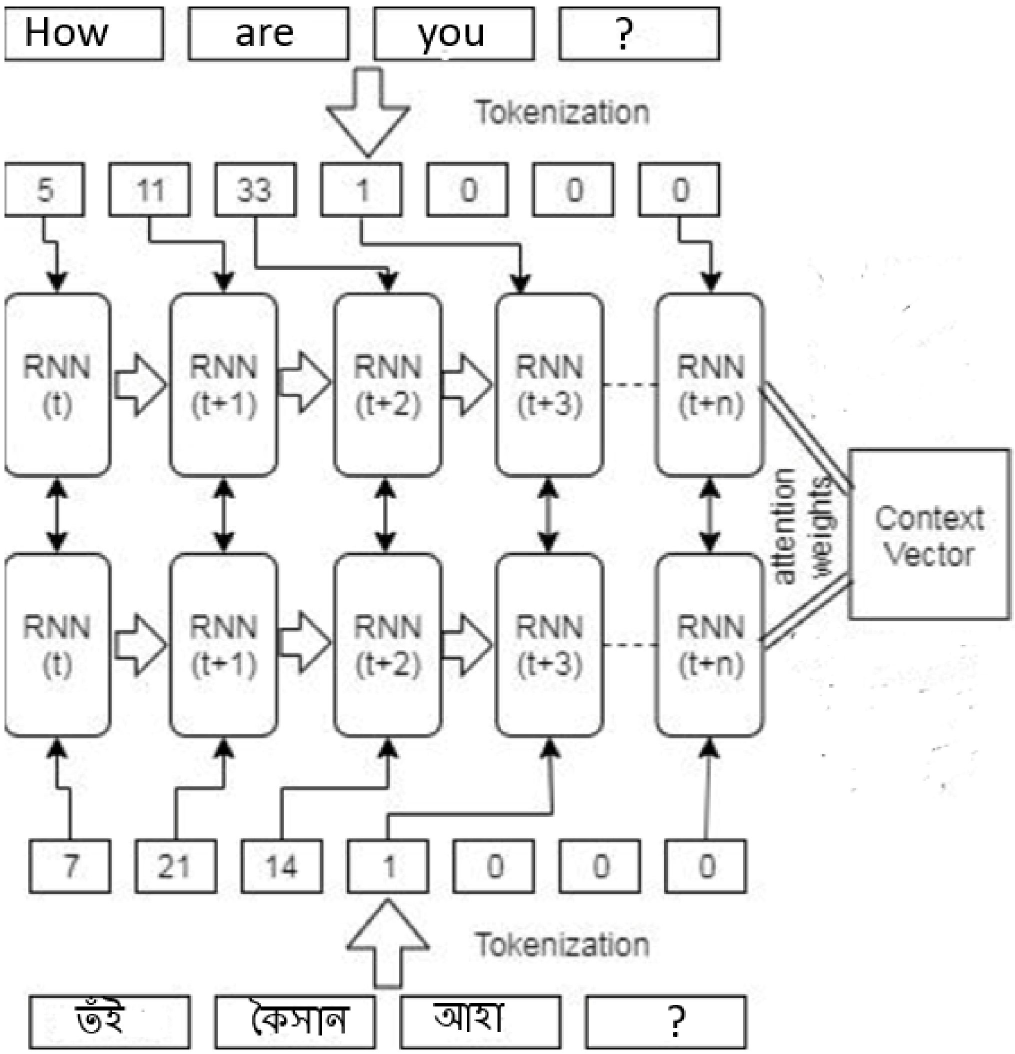

Figure 3.2: Abstract view of Recurrent neural network.

3.3 Detailed Explanation

To build this framework we have normalized and then sampling the data set then divided the total data for training and testing. Then input data is passed through an encoder that vectorizes the input and then vectorized data passed through an attention model to predict the sentences. Finally, another RNN is used as a decoder to get the final output. In this section we explained the process briefly.

3.3.1 Data Collection

Datasets are collected for analysis or training with a machine learning algorithm. The parallel sentences in English and Bangla are the key dataset for our study. To train and test the intelligent system, we require some co-responding Bangla sentences for each English sentence. The data collection was compiled from articles that were handwritten in English and Bangla by people.

3.3.2 Prepossessing

It is impossible to create a corpus without any error and inconsistency. We get a usable corpus after removing redundancies and unnecessary words and punctuation. Data prepossessing is a proven method of resolving such issues. Data prepossessing is required to ll in missing values, smooth noisy data, identify or remove the outliers, and resolve inconsistencies. Words with no signicance must be removed from the text. For cleaning the raw data we have to remove the stop words. Which are basically a set of commonly used words in any language. The reason why stop words are critical to many applications is that, if we remove the words that are very commonly used in a given language, we can focus on the important words instead. Eliminating these common terms can do two things for the classifier:

- Learning can become much faster, since we are reducing the total number of features in use.
- Translation become more accurate since we are eliminating noise or distracting. features.

3.3.3 Tokenizations

Some text preprocessing processes are performed in order to normalize the dataset. All of the letters in a sentence are changed to lowercase, and all punctuation is eliminated. Characters that do not belong in the English or Bangla alphabets are also removed.In its initial state, the dataset must be tokenized. All words in each English and Bangla sentence are tokenized based on their frequency. Tensorflow includes a tokenizer module that may be used to map a word to an integer number.

Table 3.1: Tokenization of a sentence.

Sentence	Tokenized Mapping			
how are you ?	how	are	you	?

3.3.4 Context Vector

English tokenized sentences could be anticipated using the stored context vector. To create a context vector, tokenized English and Bangla sentences are provided as input.Attention weights are applied to mapped tokens in English and Bangla. The encoder and attention weights are assessed. The attention of the Bangla tokenized sequence over the English tokenized sequence is represented by attention weights.

$$Score = sigmoid(Dlayer + Hlayer) \tag{3.1}$$

$$AttentionWeight = Softmax(score) \tag{3.2}$$

$$contexVector = (atentionscore * output) \tag{3.3}$$

Equation(3.1,3.2,3.3) are used to create the context vector. English-Bangla parallel sentences are used as training sequence inputs. The RNN embedding layer normalizes token sequences and outputs them as GRU/LSTM layer input. Both GRU and LSTM are implemented and tested to assess performance.

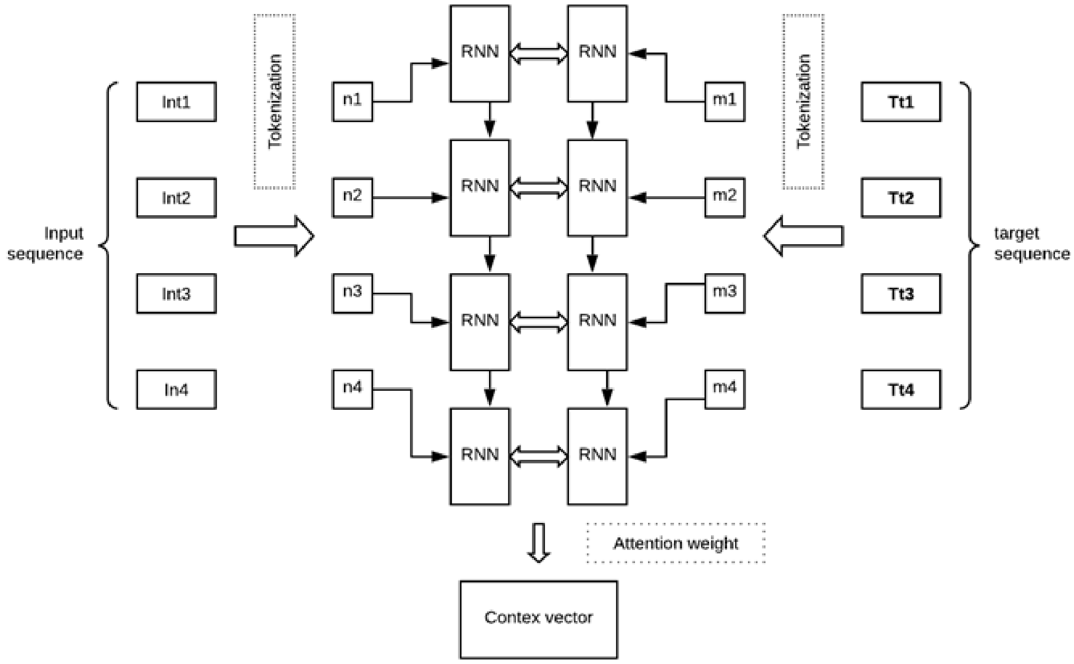

Figure 3.3: Training Model to make the context vector.

3.3.5 Training Phase

Training phase has 3 modules. Those modules will perform their task independently. Input to the training phase is a set of text file . Output of the training phase is a trained Recurrent Neoral network model that will be used in testing phase. Block diagram of the training phase is shown in Figure 3.3

3.3.6 Recurrent Neural Network Model

We used RNN both encoding and decoding. Sequential input is used in the recurrent neural network (RNN) model[12]. A node's output is used as an input bias for another node. Because the words in a sentence have a corelational meaning. we used the RNN model in our work.Figure 3.4 described the Recurrent neural network.

The encoder is made up of a few input embedding layers, a GRU layer, and some hidden input layers. The dataset is normalized using the embedding layer. The second layer employs a Gated Recurrent Unit (GRU) [13]. Instead of GRU, Long Short Term Memory (LSTM)[14] is utilized to measure performance. The

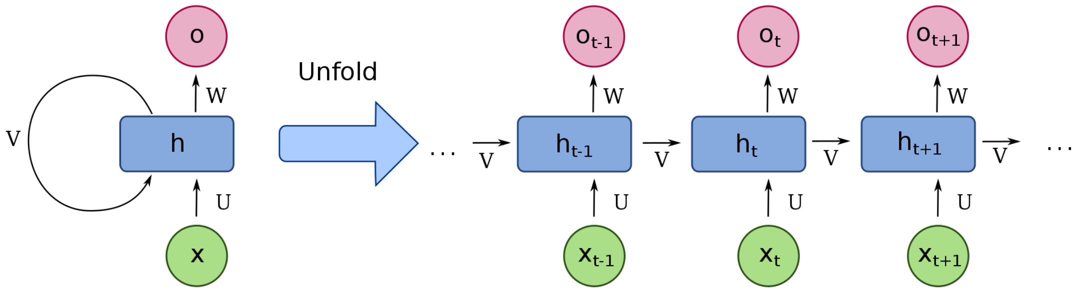

Figure 3.4: Diagram of Recurrent neural network.

model's performance is measured using several activation functions.

The embedding layer is used to create the initial layer of the decoder. The encoder is then replaced with GRU. On the GRU layer, there is an activation function. Following GRU, a dense layer with the complete vocabulary size is created. To create the context vector, the Bahdanau Attention Theory is applied [12].

3.3.7 Attention Mechanism

Our approach analyzes the attention approach. Attention weights are utilized to determine which English words are focused on and which Bangla terms are measured . To normalize inputs with two dense layers and evaluate score weights, the sigmoid activation function is used. Softmax or sigmoid activation functions are also used to normalize the weights.Some activation functions are used to compare performance. The activation function's primary function is to normalize the input sequence.

$$F(x) = [Exp(2X) - 1]/[Exp(2X) + 1] \tag{3.4}$$

where x is the value of the sequence.

Linear Activation Function:

$$F(Xi) = WiXi + b \tag{3.5}$$

Softmax Activation Function:

$$F(Xi) = Exp(Xi)/\sum_{j=1}^{k} Exp(Xj) \tag{3.6}$$

Sigmoid Activation Function:

$$F(X) = 1/[1 + \sum_{j=1}^{k} Exp(Xj)] \tag{3.7}$$

Where i=1,...,k and X=(X1,....,Xk)€R^k.

All these activation functions(3.4,3.5,3.6,3.7) are used in the Encoder GRU layer, Decoder GRU layer and attention layer for finding the attention weights.

In figure 3.5 Expalined the error back propagation algorithm where,(A) compute the error function, (B) propagate the local gradient from the output layer L to (C) hidden layer L1 and (D) until input layer 1.

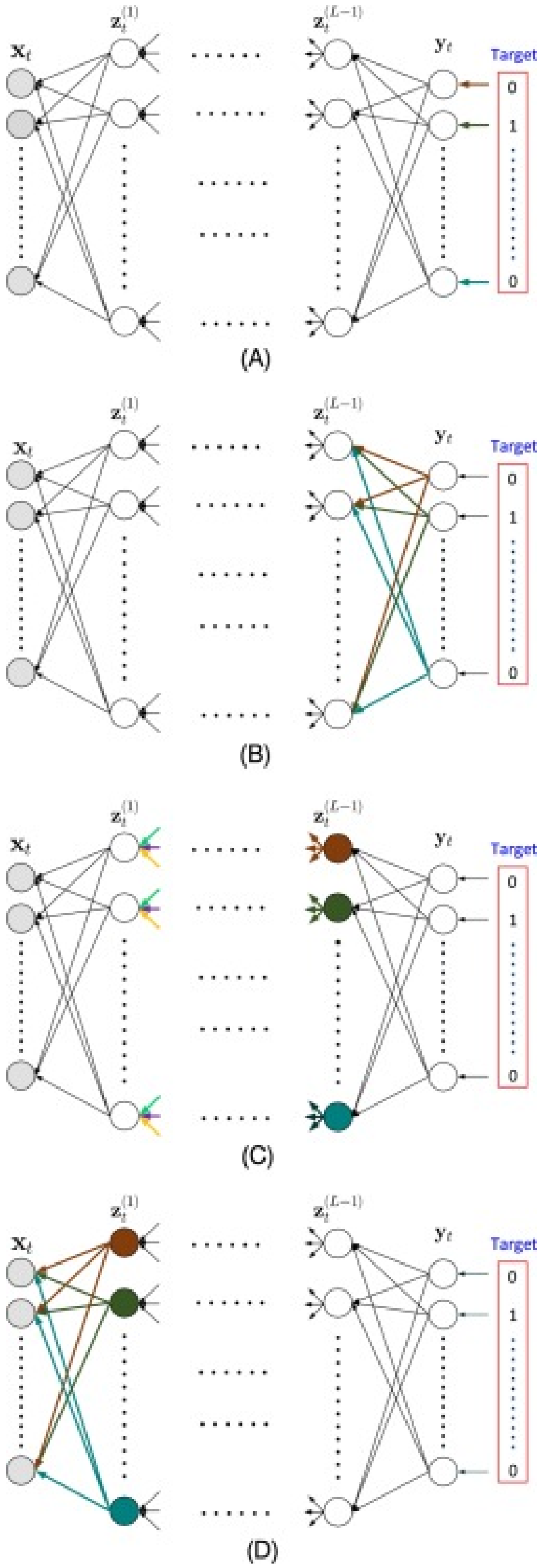

Figure 3.5: Procedure in the error back propagation algorithm.

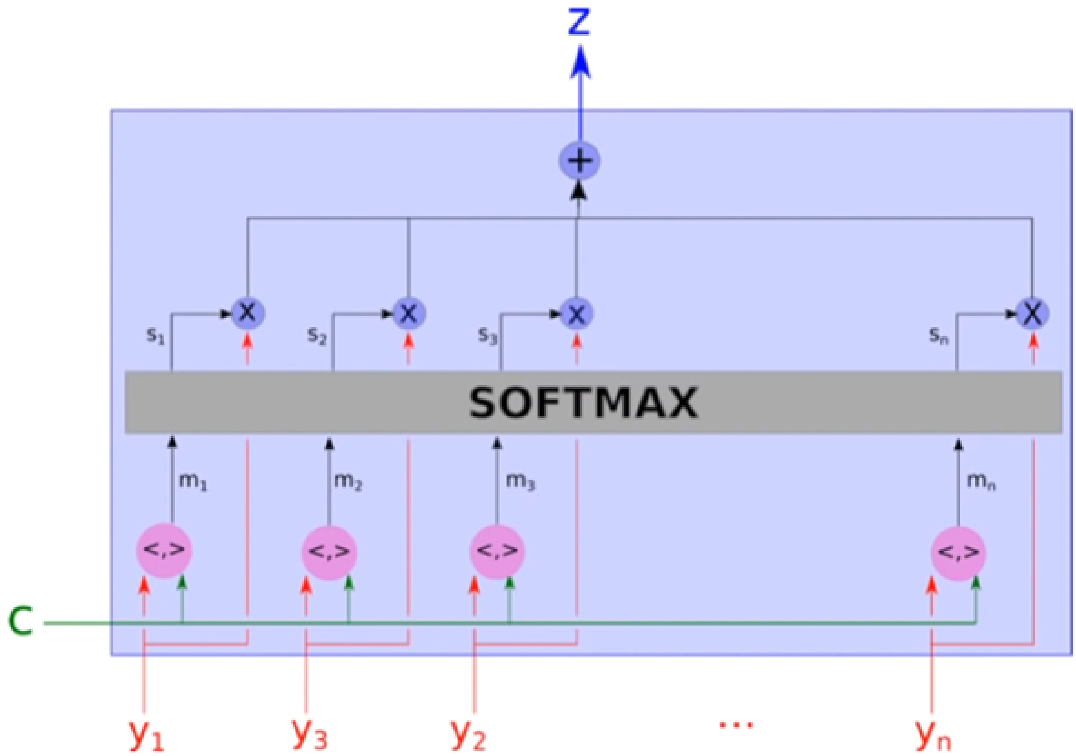

Figure 3.6: Closer view of attention mechanism.

3.4 Conclusion

In this chapter, a methodology for developing a framework to translate English to Bangla class dialect is discussed. For feature extraction, two methods are designed. Features obtained from each of them constitute the final feature descriptor. As for translating recurrent neural network is used. The next chapter is about the experimental result analysis of the proposed framework.

Chapter 4

Results and Discussions

4.1 Introduction

A full explanation of the proposed framework for English to Bangla class dialect was given in the previous chapters. The performance of the proposed methodology is examined in this chapter.

This framework was implemented in TensorFlow 2.0 open-source python library with Keras API. It is unfortunate that there is no standard dataset of Bagla class dialect. For this thesis work, we made a dataset with the help of the book "Kamalgonj-er Vasha Boichitra"[15]. Besides that to evaluate the performance we used tatoeba's Bangla-English dataset, which is an open-source data repository.

4.2 Dataset Description

There were some important works done in Machine translation in Bengali before us but there is no contribution about bangla class dialects. We have focused on neural machine translation in keeping with prior studies of Siddique et al.[2] It is unfortunate that there is no standard dataset of Bagla class dialect. For this thesis work, we made a dataset with the help of the book "Kamalgonj-er Vasha Boichitra". From that book we have collect 200 sententens of 5 class. Besides that to evaluate the performance we used tatoeba's Bangla-English dataset, which is an open-source data repository. That has 4332 no. of data.

Table 4.1: Data Summary For Bangla Dataset.

Number of Samples	4332
Number of unique input token	71
Number of unique output token	91

Table 4.2: Data Summary For Bangla Class Dataset.

Number of Samples	352
Number of unique input token	45
Number of unique output token	48

Table 4.3: Data Summary For Train and test In Bangla Dataset.

	Training	Testing
Number of Class	1	1
Number of Documents	3820	418

Table 4.4: Data Summary For Train and test In Bangla Class Dataset.

	Training	Testing
Number of Class	5	5
Number of Documents	320	50

4.3 Evaluation of Framework

There are many types of performance measurement technique which are help to measure the level of accuracy of the system. We work with the 5 type of accuracy measurement technique. Performance evaluation technique of the system are discuss below:

4.3.1 Activation function

Some activation functions are used to compare performance. The activation function's main purpose is to normalize the input sequence.To normalize inputs with two dense layers and evaluate score weights, the sigmoid activation function is used. Softmax or sigmoid activation functions are also used to normalize the weights.Some activation functions are used to compare performance. The activation function's primary function is to normalize the input sequence.

$$F(x) = [Exp(2X) - 1]/[Exp(2X) + 1] \tag{4.1}$$

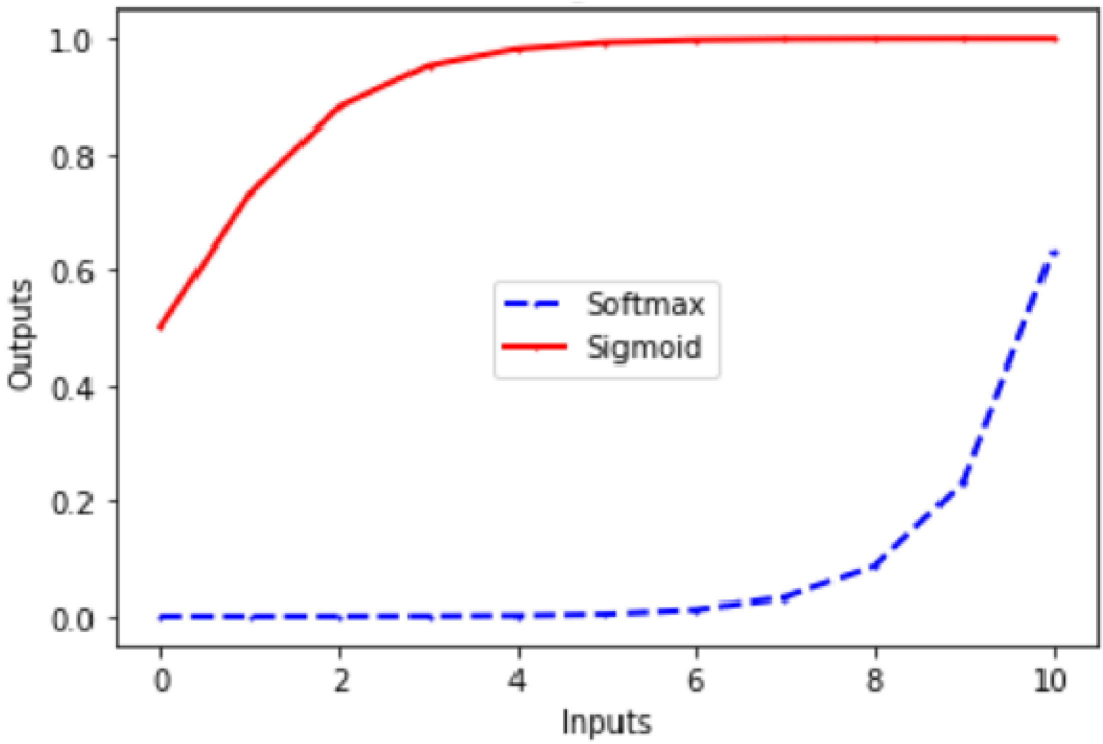

Figure 4.1: Sigmoid and Softmax activation function.

where x is the value of the sequence.

Linear Activation Function:

$$F(Xi) = WiXi + b \tag{4.2}$$

Softmax Activation Function:

$$F(Xi) = Exp(Xi)/\sum_{j=1}^{k} Exp(Xj) \tag{4.3}$$

Sigmoid Activation Function:

$$F(X) = 1/[1 + \sum_{j=1}^{k} Exp(Xj)] \tag{4.4}$$

Where i=1,...,k and X=(X1,....,Xk)€R^k.

All these activation functions(4.1,4.2,4.3,4.4) are used in the Encoder GRU layer, Decoder GRU layer and attention layer for finding the attention weights.In figure 4.1 shows the Sigmoid and softmax activation function.

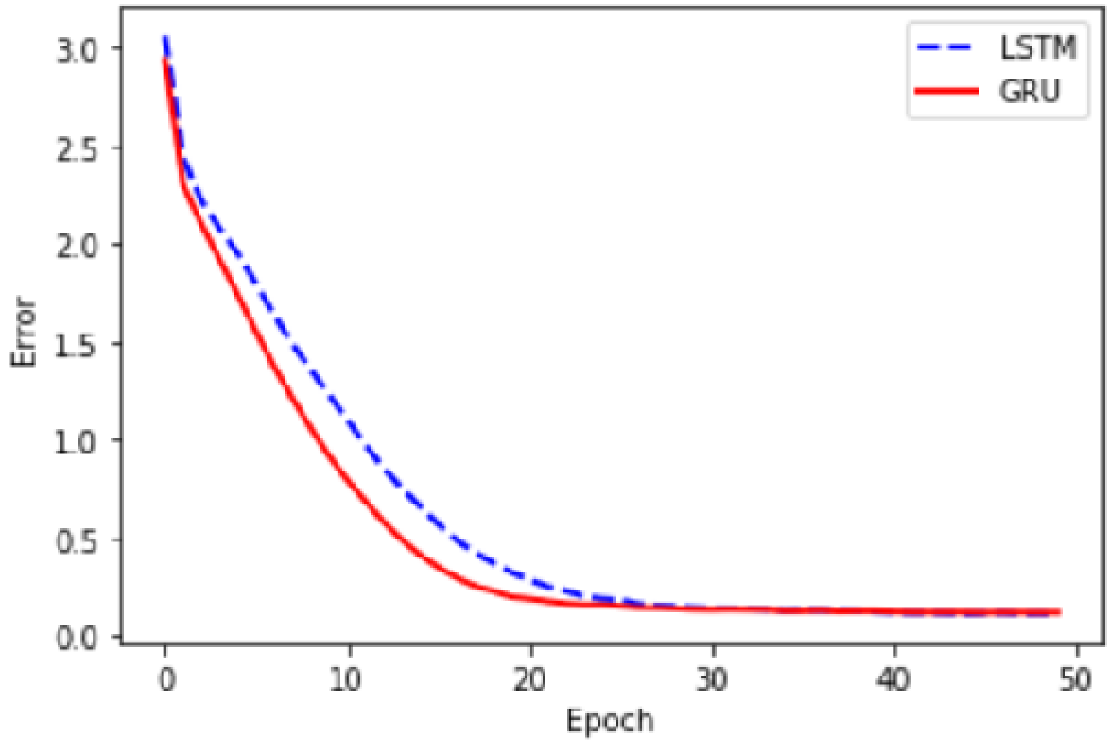

Figure 4.2: Loss of model based on attention layer activation function.

4.3.2 Attention layer

The attention layer consists of two activation functions. One is for input and the other one is for normalizing the outputs as attention weights. SoftmaxEq. (4.3) and Sigmoid Eq. (4.4) are used here. All combinations of sigmoid and softmax are tried to evaluate the best performance.Figure 4.2 represents, the sigmoid function gives the best performance for the attention layer. Then sigmoid for inputs of attention layer and softmax for output attention layer are efficient.

4.3.3 LSTM and GRU

After the embedding layer, the GRU layer is implemented. Instead of using GRU, the LSTM layer can be also used. 50 epochs are executed to generate the Fig. 2 and the best performing activation functions are used from the inputoutput layer and attention layer. But if the epochs have increased, the errors and losses of Gated recurrent unit(GRU) decrease. To get the best performance, some parameters like- Central processing unit (CPU), Randomaccess memory (RAM)

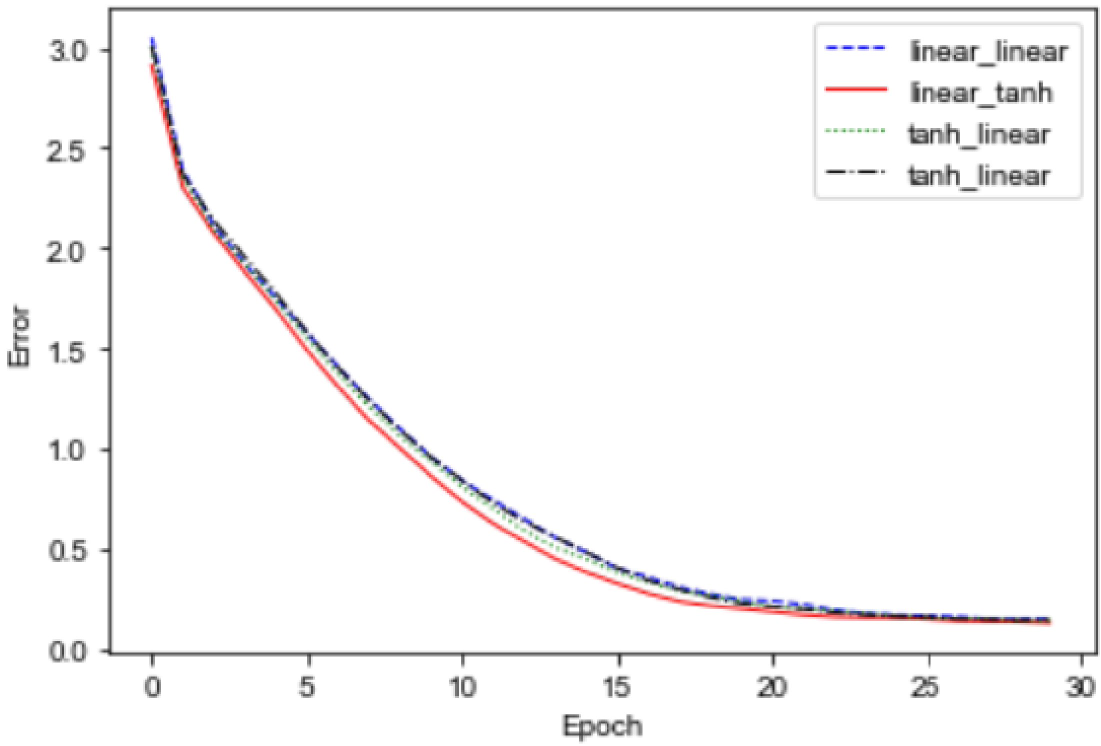

Figure 4.3: Performance of the model (LSTM-GRU).

and Graphics processing unit (GPU) and obviously the number of datasets as sample training matters.

Here in Figure 4.3, the performance of GRU looks better than LSTM. That why our final model added GRU layer. The mean error of GRU is 0.508, which is more efficient than LSTM 0.602

4.3.4 Error minimization per epoch

To evaluate the minimizing of error for each epoch, 0 epochs are performed. The first 50 epochs and the second 50 epochs are separated into two portions. Figure 4.4 demonstrates that the satisfaction level of the error has decreased during the last fifty epochs.

The standard deviation for the first 50 epochs is around 0.680, according to Table Figure 4.5,4.6. For the first 50 epochs, the error drops by about 0.680 each epoch. The mean error drops to 0.17 after 50 epochs, which is a decent result with 0.003 standard deviations.

```
Epoch 73/100
55/55 [==============================] - 46s 839ms/step - loss: 0.1296 - accuracy: 0.9624 - val_loss: 1.1990 - val_accuracy: 0.7749
Epoch 74/100
55/55 [==============================] - 46s 840ms/step - loss: 0.1278 - accuracy: 0.9628 - val_loss: 1.1894 - val_accuracy: 0.7767
Epoch 75/100
55/55 [==============================] - 46s 844ms/step - loss: 0.1267 - accuracy: 0.9627 - val_loss: 1.1962 - val_accuracy: 0.7759
Epoch 76/100
55/55 [==============================] - 46s 841ms/step - loss: 0.1261 - accuracy: 0.9632 - val_loss: 1.2082 - val_accuracy: 0.7780
Epoch 77/100
55/55 [==============================] - 46s 830ms/step - loss: 0.1247 - accuracy: 0.9633 - val_loss: 1.2053 - val_accuracy: 0.7792
Epoch 78/100
55/55 [==============================] - 46s 835ms/step - loss: 0.1248 - accuracy: 0.9631 - val_loss: 1.2188 - val_accuracy: 0.7765
Epoch 79/100
55/55 [==============================] - 47s 851ms/step - loss: 0.1223 - accuracy: 0.9640 - val_loss: 1.2451 - val_accuracy: 0.7742
Epoch 80/100
55/55 [==============================] - 46s 837ms/step - loss: 0.1229 - accuracy: 0.9640 - val_loss: 1.2396 - val_accuracy: 0.7751
Epoch 81/100
55/55 [==============================] - 46s 841ms/step - loss: 0.1207 - accuracy: 0.9642 - val_loss: 1.2447 - val_accuracy: 0.7735
Epoch 82/100
55/55 [==============================] - 46s 841ms/step - loss: 0.1198 - accuracy: 0.9641 - val_loss: 1.2379 - val_accuracy: 0.7765
Epoch 83/100
55/55 [==============================] - 47s 849ms/step - loss: 0.1195 - accuracy: 0.9645 - val_loss: 1.2445 - val_accuracy: 0.7760
Epoch 84/100
55/55 [==============================] - 46s 834ms/step - loss: 0.1181 - accuracy: 0.9648 - val_loss: 1.2805 - val_accuracy: 0.7735
Epoch 85/100
55/55 [==============================] - 46s 842ms/step - loss: 0.1183 - accuracy: 0.9647 - val_loss: 1.2656 - val_accuracy: 0.7754
Epoch 86/100
55/55 [==============================] - 46s 839ms/step - loss: 0.1169 - accuracy: 0.9654 - val_loss: 1.2615 - val_accuracy: 0.7760
Epoch 87/100
55/55 [==============================] - 46s 834ms/step - loss: 0.1158 - accuracy: 0.9652 - val_loss: 1.2559 - val_accuracy: 0.7763
Epoch 88/100
55/55 [==============================] - 46s 832ms/step - loss: 0.1158 - accuracy: 0.9655 - val_loss: 1.2879 - val_accuracy: 0.7749
Epoch 89/100
55/55 [==============================] - 46s 842ms/step - loss: 0.1143 - accuracy: 0.9657 - val_loss: 1.2700 - val_accuracy: 0.7770
Epoch 90/100
55/55 [==============================] - 46s 843ms/step - loss: 0.1145 - accuracy: 0.9655 - val_loss: 1.3205 - val_accuracy: 0.7717
Epoch 91/100
55/55 [==============================] - 46s 839ms/step - loss: 0.1143 - accuracy: 0.9654 - val_loss: 1.2841 - val_accuracy: 0.7764
Epoch 92/100
55/55 [==============================] - 46s 841ms/step - loss: 0.1117 - accuracy: 0.9659 - val_loss: 1.2952 - val_accuracy: 0.7740
Epoch 93/100
55/55 [==============================] - 46s 840ms/step - loss: 0.1124 - accuracy: 0.9662 - val_loss: 1.2777 - val_accuracy: 0.7757
Epoch 94/100
55/55 [==============================] - 46s 840ms/step - loss: 0.1120 - accuracy: 0.9659 - val_loss: 1.2883 - val_accuracy: 0.7778
Epoch 95/100
55/55 [==============================] - 46s 846ms/step - loss: 0.1115 - accuracy: 0.9659 - val_loss: 1.3155 - val_accuracy: 0.7750
Epoch 96/100
55/55 [==============================] - 46s 843ms/step - loss: 0.1107 - accuracy: 0.9662 - val_loss: 1.2931 - val_accuracy: 0.7771
Epoch 97/100
55/55 [==============================] - 46s 846ms/step - loss: 0.1099 - accuracy: 0.9663 - val_loss: 1.3084 - val_accuracy: 0.7786
Epoch 98/100
55/55 [==============================] - 47s 849ms/step - loss: 0.1096 - accuracy: 0.9663 - val_loss: 1.2991 - val_accuracy: 0.7755
Epoch 99/100
55/55 [==============================] - 47s 856ms/step - loss: 0.1083 - accuracy: 0.9668 - val_loss: 1.3246 - val_accuracy: 0.7751
Epoch 100/100
55/55 [==============================] - 47s 860ms/step - loss: 0.1094 - accuracy: 0.9665 - val_loss: 1.3145 - val_accuracy: 0.7784
```

Figure 4.4: Performance increase per epoch.


```
Epoch 73/100
55/55 [==============================] - 46s 839ms/step - loss: 0.1296 - accuracy: 0.9624 - val_loss: 1.1990 - val_accuracy: 0.7749
Epoch 74/100
55/55 [==============================] - 46s 840ms/step - loss: 0.1278 - accuracy: 0.9628 - val_loss: 1.1894 - val_accuracy: 0.7767
Epoch 75/100
55/55 [==============================] - 46s 844ms/step - loss: 0.1267 - accuracy: 0.9627 - val_loss: 1.1962 - val_accuracy: 0.7759
Epoch 76/100
55/55 [==============================] - 46s 841ms/step - loss: 0.1261 - accuracy: 0.9632 - val_loss: 1.2082 - val_accuracy: 0.7780
Epoch 77/100
55/55 [==============================] - 46s 830ms/step - loss: 0.1247 - accuracy: 0.9633 - val_loss: 1.2053 - val_accuracy: 0.7792
Epoch 78/100
55/55 [==============================] - 46s 835ms/step - loss: 0.1248 - accuracy: 0.9631 - val_loss: 1.2188 - val_accuracy: 0.7765
Epoch 79/100
55/55 [==============================] - 47s 851ms/step - loss: 0.1223 - accuracy: 0.9640 - val_loss: 1.2451 - val_accuracy: 0.7742
Epoch 80/100
55/55 [==============================] - 46s 837ms/step - loss: 0.1229 - accuracy: 0.9640 - val_loss: 1.2396 - val_accuracy: 0.7751
Epoch 81/100
55/55 [==============================] - 46s 841ms/step - loss: 0.1207 - accuracy: 0.9642 - val_loss: 1.2447 - val_accuracy: 0.7735
Epoch 82/100
55/55 [==============================] - 46s 841ms/step - loss: 0.1198 - accuracy: 0.9641 - val_loss: 1.2379 - val_accuracy: 0.7765
Epoch 83/100
55/55 [==============================] - 47s 849ms/step - loss: 0.1195 - accuracy: 0.9645 - val_loss: 1.2445 - val_accuracy: 0.7760
Epoch 84/100
55/55 [==============================] - 46s 834ms/step - loss: 0.1181 - accuracy: 0.9648 - val_loss: 1.2805 - val_accuracy: 0.7735
Epoch 85/100
55/55 [==============================] - 46s 842ms/step - loss: 0.1183 - accuracy: 0.9647 - val_loss: 1.2656 - val_accuracy: 0.7754
Epoch 86/100
55/55 [==============================] - 46s 839ms/step - loss: 0.1169 - accuracy: 0.9654 - val_loss: 1.2615 - val_accuracy: 0.7760
Epoch 87/100
55/55 [==============================] - 46s 834ms/step - loss: 0.1158 - accuracy: 0.9652 - val_loss: 1.2559 - val_accuracy: 0.7763
Epoch 88/100
55/55 [==============================] - 46s 832ms/step - loss: 0.1158 - accuracy: 0.9655 - val_loss: 1.2879 - val_accuracy: 0.7749
Epoch 89/100
55/55 [==============================] - 46s 842ms/step - loss: 0.1143 - accuracy: 0.9657 - val_loss: 1.2700 - val_accuracy: 0.7770
Epoch 90/100
55/55 [==============================] - 46s 843ms/step - loss: 0.1145 - accuracy: 0.9655 - val_loss: 1.3205 - val_accuracy: 0.7717
Epoch 91/100
55/55 [==============================] - 46s 839ms/step - loss: 0.1143 - accuracy: 0.9654 - val_loss: 1.2841 - val_accuracy: 0.7764
Epoch 92/100
55/55 [==============================] - 46s 841ms/step - loss: 0.1117 - accuracy: 0.9659 - val_loss: 1.2952 - val_accuracy: 0.7740
Epoch 93/100
55/55 [==============================] - 46s 840ms/step - loss: 0.1124 - accuracy: 0.9662 - val_loss: 1.2777 - val_accuracy: 0.7757
Epoch 94/100
55/55 [==============================] - 46s 840ms/step - loss: 0.1120 - accuracy: 0.9659 - val_loss: 1.2883 - val_accuracy: 0.7778
Epoch 95/100
55/55 [==============================] - 46s 846ms/step - loss: 0.1115 - accuracy: 0.9659 - val_loss: 1.3155 - val_accuracy: 0.7750
Epoch 96/100
55/55 [==============================] - 46s 843ms/step - loss: 0.1107 - accuracy: 0.9662 - val_loss: 1.2931 - val_accuracy: 0.7771
Epoch 97/100
55/55 [==============================] - 46s 846ms/step - loss: 0.1099 - accuracy: 0.9663 - val_loss: 1.3084 - val_accuracy: 0.7786
Epoch 98/100
55/55 [==============================] - 47s 849ms/step - loss: 0.1096 - accuracy: 0.9663 - val_loss: 1.2991 - val_accuracy: 0.7755
Epoch 99/100
55/55 [==============================] - 47s 856ms/step - loss: 0.1083 - accuracy: 0.9668 - val_loss: 1.3246 - val_accuracy: 0.7751
Epoch 100/100
55/55 [==============================] - 47s 860ms/step - loss: 0.1094 - accuracy: 0.9665 - val_loss: 1.3145 - val_accuracy: 0.7784
```


Figure 4.6: Performance increase per epoch.

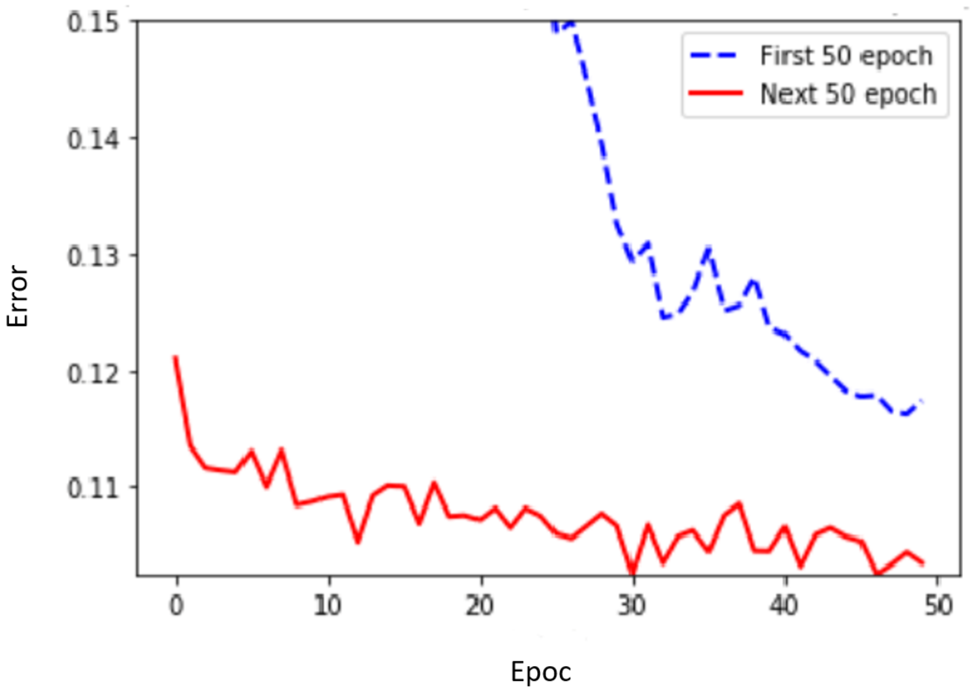

Figure 4.7: Performance increase per epoch.

4.4 Evaluation of Performance

In this Section we evaluated the performance of our developed framework. We have tested our system with test dataset and obtained overoall 72 percent accuracy. we also test the system without attention mechanism. The comparison of this two system are given in this section.and we also give the BLEU analysis to evaluate our performence.

4.4.1 Success Rate

In this thesis work we are tested our system with two two neural Network, one is RNN with attention mechanism and another is RNN without attention mechanism.we have tested 60 sentences in different length and obtain a 86 percent accuracy. Table 4.2 shows the result of various testing. in table 4.2 we have shown the accuracy of various sentences with different length.we also show the comparison with attention model.

In order to understand the performance easily we discuss the result in figure 4.8,

Table 4.5: Success rate for different type of sentences.

Model	Sentences length	No. of sentences	No. of correctly translated sentences	overall success %
RNN without Attention	2	25	24	96
	3	20	17	86
	4	12	8	72
	5	7	4	50
	6	5	2	42
RNN with Attention	2	25	23	93
	3	12	9	85
	4	8	5	75
	5	7	4	68
	6	5	3	64

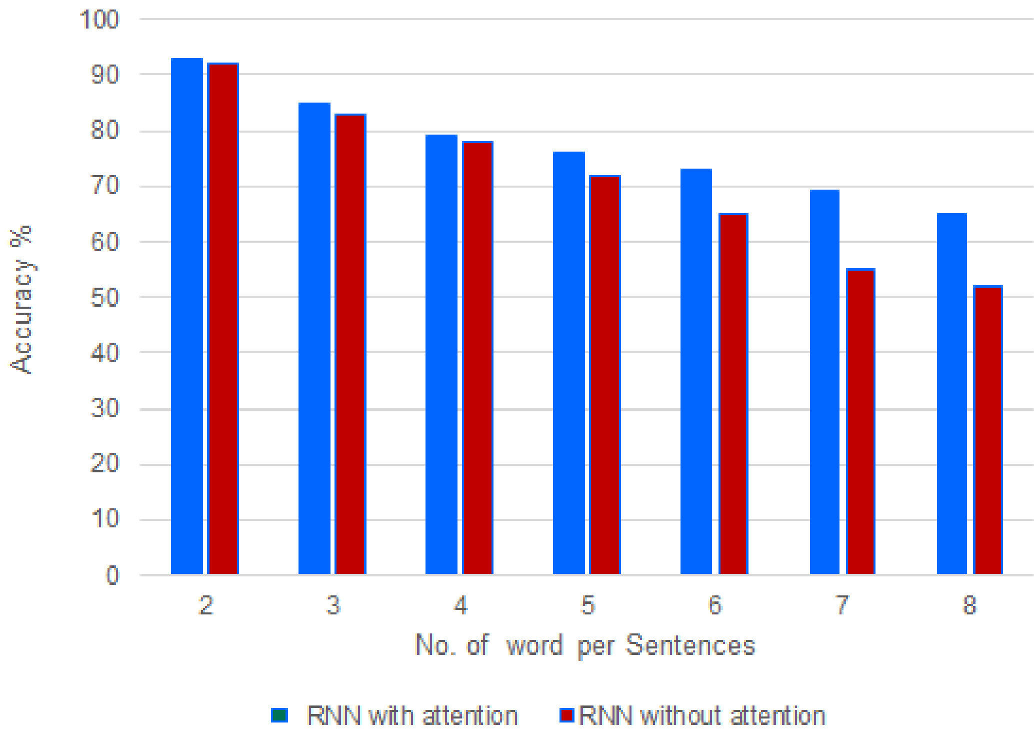

Figure 4.8: Success rate for different types of sentences.

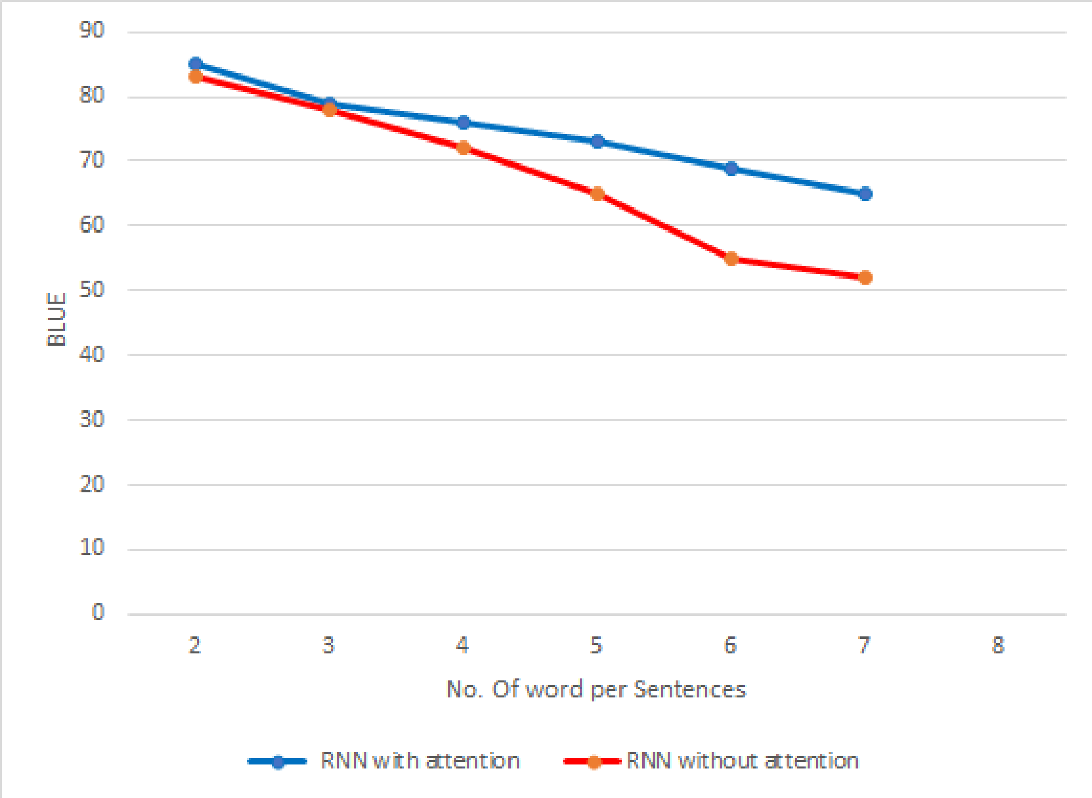

Figure 4.9: BLEU Score vs. sentence length for different types of sentences.

where along x-axis indicates the number of word per sentences and y-axis indicates the accuracy.The blue section indicates the accuracy of model with attention mechanism and orange section indicates the framework of simple neural network.

4.4.2 BLEU Score

The BLEU (Bilingual Evaluation Understudy Score) is a statistic for comparing a generated sentence to a reference sentence. The score was created to assess the accuracy of automated machine translation systems predictions[16].

In figure 4.9 shows the BLEU Score vs. sentence length of sentences. We shown that there is a slice change among two model where sentence length is up-to three words but in higher order sentences attention mechanism showed a higher accuracy.

4.5 Conclusion

This chapter shows the result of translating English sentences to bangla class dialects sentences . Performance of the proposed framework is also discussed here. As shown by the results, the proposed framework provides better accuracy. In the next chapter, the conclusion is drawn on this thesis work.

Chapter 5

Conclusion

5.1 Conclusion

Machine translation has significant advantages. It saves time, can translate multiple languages, and so forth. This article attempted to build and develop a machine translation system from English to Bangla, We had to create a data collection with several types of Class dialects. We have given a comparison of neural network accuracy using different features in this work, which will be useful to others. The architecture of an English to Bangla machine translation system is described in this work. The encoder-decoder recurrent neural network was used to create the system. For the mapping of English and Bangla terms, the model employs a knowledge-based context vector. The experiment shows average accuracy of 86 percent for Recurrent Neural Network. By increasing the dataset size, the system's accuracy can be improved. In compared to other implemented approaches found in various studies, our English to Bangla RNN-based technique produces better outcomes.

5.2 Future Work

After running the simulation, various flaws were discovered that will be addressed in the future, such as dealing with a big vocabulary can increase efficiency, and increasing the number of epochs can enhance accuracy. Working with these kinds of problems necessitates additional processing power and memory. With the seq2seq approach, performance can be improved by creating numerous dense layers.Instead of causing problems, this research will help to improve the structure of English to Bangla universal machine learning methods.

References

[1] wikipedia. (2019). ‘wikipedia,’ [Online]. Available: `https://en.wikipedia.org/wiki/Indigenous_peoples_in_Bangladesh#cite_note-1` (visited on 30th Mar. 2021) (cit. on p. 1).

[2] S. Siddique, T. Ahmed, M. R. A. Talukder and M. M. Uddin, ‘English to bangla machine translation using recurrent neural network,’ *International Journal of Future Computer and Communication*, vol. 9, no. 2, 2020 (cit. on pp. 2, 20).

[3] T. Mikolov, M. Karafiát, L. Burget, J. Černock and S. Khudanpur, ‘Recurrent neural network based language model,’ in *Eleventh annual conference of the international speech communication association*, 2010 (cit. on p. 2).

[4] S. Ismail, M. S. Rahman and M. A. Al Mumin, ‘Developing an automated bangla parts of speech tagged dictionary,’ in *16th Int’l Conf. Computer and Information Technology*, IEEE, 2014, pp. 355–359 (cit. on p. 8).

[5] M. M. Hoque and M. M. Ali, ‘A parsing methodology for bangla natural language sentences,’ in *Proceedings of International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh*, 2003, pp. 277–282 (cit. on p. 8).

[6] A. Gutkin, L. Ha, M. Jansche, K. Pipatsrisawat and R. Sproat, ‘Tts for low resource languages: A bangla synthesizer,’ 2016 (cit. on p. 8).

[7] S. A. Chowdhury, F. Alam, N. Khan and S. R. Noori, ‘Bangla grapheme to phoneme conversion using conditional random fields,’ in *2017 20th International Conference of Computer and Information Technology (ICCIT)*, IEEE, 2017, pp. 1–6 (cit. on p. 8).

[8] D. Gupta, A. Ekbal and P. Bhattacharyya, ‘A deep neural network framework for english hindi question answering,’ *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 19, no. 2, pp. 1–22, 2019 (cit. on p. 8).

[9] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, ‘Google’s neural machine translation system: Bridging the gap between human and machine translation,’ *arXiv preprint arXiv:1609.08144*, 2016 (cit. on p. 8).

[10] S. Khan and R. Mishra, ‘A neural network based approach for english to hindi machine translation,’ 2012 (cit. on p. 9).

[11] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, ‘Bleu: A method for automatic evaluation of machine translation,’ in *Proceedings of the 40th annual*

meeting of the Association for Computational Linguistics, 2002, pp. 311–318 (cit. on p. 9).

[12] D. Bahdanau, K. Cho and Y. Bengio, 'Neural machine translation by jointly learning to align and translate,' *arXiv preprint arXiv:1409.0473*, 2014 (cit. on pp. 9, 15, 16).

[13] R. Dey and F. M. Salem, 'Gate-variants of gated recurrent unit (gru) neural networks,' in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, IEEE, 2017, pp. 1597–1600 (cit. on p. 15).

[14] M. Sundermeyer, R. Schlüter and H. Ney, 'Lstm neural networks for language modeling,' in *Thirteenth annual conference of the international speech communication association*, 2012 (cit. on p. 15).

[15] mohammad mahmudul haque, 'Kamalgonj-er vasha boichitra, Kamalgonj-er vasha boichitra,' 2018 (cit. on p. 20).

[16] N. Mathur, T. Baldwin and T. Cohn, 'Tangled up in bleu: Reevaluating the evaluation of automatic machine translation evaluation metrics,' *arXiv preprint arXiv:2006.06264*, 2020 (cit. on p. 28).