Bengali Sentence Completion Using Long Short Term Memory.

.

A Thesis
by

Hassan Tufik Imam
Mohammad Hadiul Islam
Md. Fuad Hasan

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DHAKA UNIVERSITY OF ENGINEERING & TECHNOLOGY, GAZIPUR

JULY 2021

Bengali Sentence Completion Using Long Short Term Memory.

A Thesis

By

Hassan Tufik Imam
Student No.: 154009

Mohammad Hadiul Islam
Student No.: 154030

Md. Fuad Hasan
Student No.: 154072

Supervisor

Dr. Fazlul Hasan Siddiqui
Professor

Submitted to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DHAKA UNIVERSITY OF ENGINEERING & TECHNOLOGY, GAZIPUR

In partial fulfillment of the requirements for the degree

of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

JULY 2021

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge, it does not contain any idea, material or resources published or written by another person except otherwise explicitly mentioned and properly referenced. We also declare that neither a part nor a whole of this work has been submitted for an award of any other degree of a university or other institute of higher education.

Signature of authors

....
(Hassan Tufik Imam)

....
(Mohammad Hadiul Islam)

....
(Md. Fuad Hasan)

Student of department of Computer Science and Engineering of Dhaka University of Engineering & Technology, Gazipur.

Signature of Thesis Supervisor

…………………………………..
(Supervisor)
Dr. Fazlul Hasan Siddiqui
Professor
Department of Computer Science and Engineering
Dhaka University of Engineering & Technology, Gazipur.

ACKNOWLEDGEMENT

All the praises are due to almighty Allah for giving us the ability to successfully complete this thesis and project.

We would like to acknowledge the supports provided by different people in different aspects of our research.

First of all we express our sincere gratitude to our thesis supervisor **Dr. Fazlul Hasan Siddiqui**, Professor Department of Computer Science and Engineering, DUET, Gazipur, for providing us time to time advices and instructions in progression of the research. We are also thankful to him insightful review during the editing and writing of this dissertation.

We wish to the all the teacher who have directly or indirectly contributed towards the completion of the thesis work.

Last but not least, we are grateful to our parents and to our families for their patience and support during our studies.

July, 2021

Authors

Hassan Tufik Imam
Mohammad Hadiul Islam
Md. Fuad Hasan

ABSTRACT

Writing is an important medium for people to communicate. People with different needs have to write on different platforms. Such as in official documents, online chatting, search, mail, book, etc. In these cases, time and effort consuming, mistakes happen. Where a person falling in very big danger sometimes. So in order to make this writing easier, researchers around the world are doing a variety of research. That's why when writing, the user can be helped to complete the sentence by predicting the next word. So it will prevent take less time, spelling mistakes also efforts. Although work was done in other languages, not much work was done in Bengali. So we chose to work with the Bengali language. We have set up a model that will help the user to complete his desired sentence in the Bengali language.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Introduction

Sentence completion is to measure grammatical and semantic correctness to choose the most appropriate sentence or word. Sentence Completion is a highly discussed topic in current domains in NLP research. Word prediction model helping tool that auto completes writing, both by completing current words and suggesting relevant word sequences. Smart typing assistant is the one that can suggest more than just the current word. Based on the context, it is able to suggest the next few words that are relevant. In this work, we propose such a model using the recent advancement of the recurrent neural network (RNN). It is a context-aware model that can predict the next few words conditioning on the context user is writing. This is a general purpose solution to typing assistant system. Basically RNN are amazingly able to handle the long-term dependencies that's why it is might be difficult which is showed in Sanzidul Islam et.ai (2019). But RNN has a vanishing gradient problem. Fe-lix et.al used LSTM to solve tasks that were previously unsolvable by RNN. (LSTM)Long short term memory networks are a special kind of RNN. All recurrent neural network have the form of a chain of a repeating modules of neural networks so LSTM also have this chain like structure but the repeating Module have a different structure instead of having a single neural network layer .But LSTM are actually made for avoiding the long short term dependency issue by keeping information in periods of time is their actual default behavior.

1.2 Motivation

There are many types of research and development works in this field, but we can hardly find text prediction-related work for the Bengali language. Bengali is the sixth-largest language in the world. This language is spoken by about 260 million people. This huge number of people use this Bengali language on their various platforms. For the benefit of this huge number of people, we suggest technique a that can help the user type more efficiently and faster than usual. In this work, we propose an intelligent typing assistant which is based on artificial intelligence derived by Long-

Short Term Memory (LSTM) model. This model can give suggestions based on the context the writer is writing.

.

1.3 Objectives with specific aims and possible outcomes

The objectives of our proposed work

a. To Increase writing fluency, allowing students to generate more writing.\newline

b. To Support vocabulary development and increase variety and complexity of words used in writing.

The possible outcomes of our proposed work:

a. Be able to predict the words of the sentence

b. Time will be saved in writing.

c. There will be less mistakes in writing and much more efficient

1.4 Our Contributions

To address research question we study different machine learning (ML) algorithm and natural language processing (NLP) techniques that can be prospective candidates for the concerned issue. Through this research, using contemporary ML and NLP tools we propose a semantic automatic marking approach. Our introduced technique can automatically assign appropriate marks to the students' responses although responses are syntactically different but semantically similar. To implement this technique we overhauled the automatic marking system of [1], and integrated one modules in the marking engine. The new modules can measure semantic similarity through finding synonyms of model answer and also check overall grammatical errors. Also we worked on the dataset, increased the size and represented in standard csv format. This now allows us to automatically feed the data to the marking system.

1.5 Thesis Outline

We have provided the introduction of our thesis in the first chapter .This chapter also includes the research topic, our aim and objectives .The second chapter contains the literature review, where we have mentioned related works and focused on background study. Third chapter is about methodology, where detailed explanation of the working process and evaluation method of our system are given. In fourth chapter result and analysis are discussed. Fifth chapter is about conclusion and future work.

CHAPTER 2: LITERATURE REVIEW

In this chapter, we have discussed details about our background study and related work To our thesis. In section 2.1 we have described NLP, NN, Learning, RNN, LSTM, Sequencing, Word2vec.In section 2.2 we have mentioned some related works.

2.1 Background Study

Here we have briefly described those topics that are related to our thesis.

2.1.1 Neural Network

An artificial neural network is a supervised learning algorithm that contains input layer output layer and hidden layers. It is mandatory one layer for input and one layer for output but there is no rule for hidden layer. The number hidden layer depend on expected result. Input data is called as independent variable and output details called dependent variable.

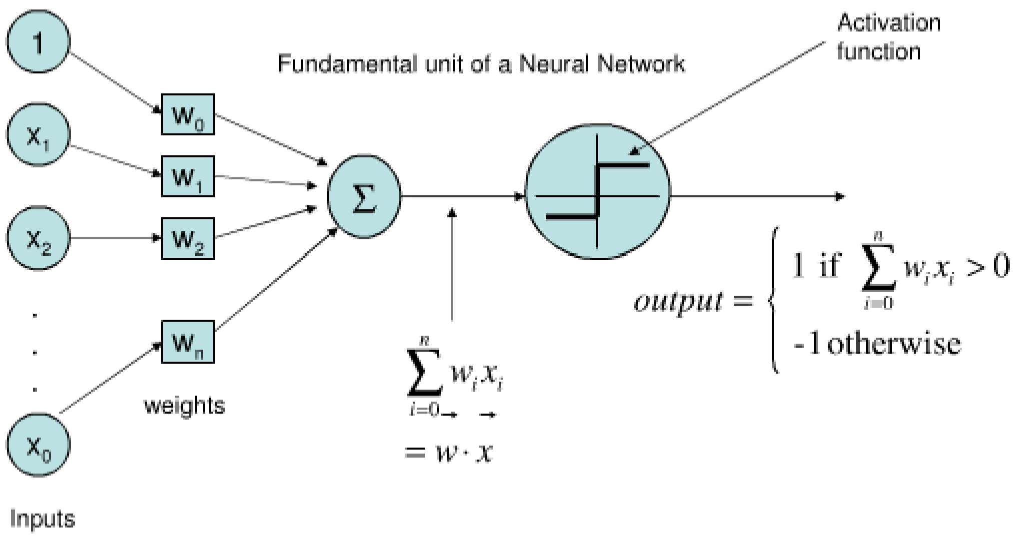

Figure 2.1.1: A Single Layer Perceptron [9]

There also show on a multilayer Neural Network

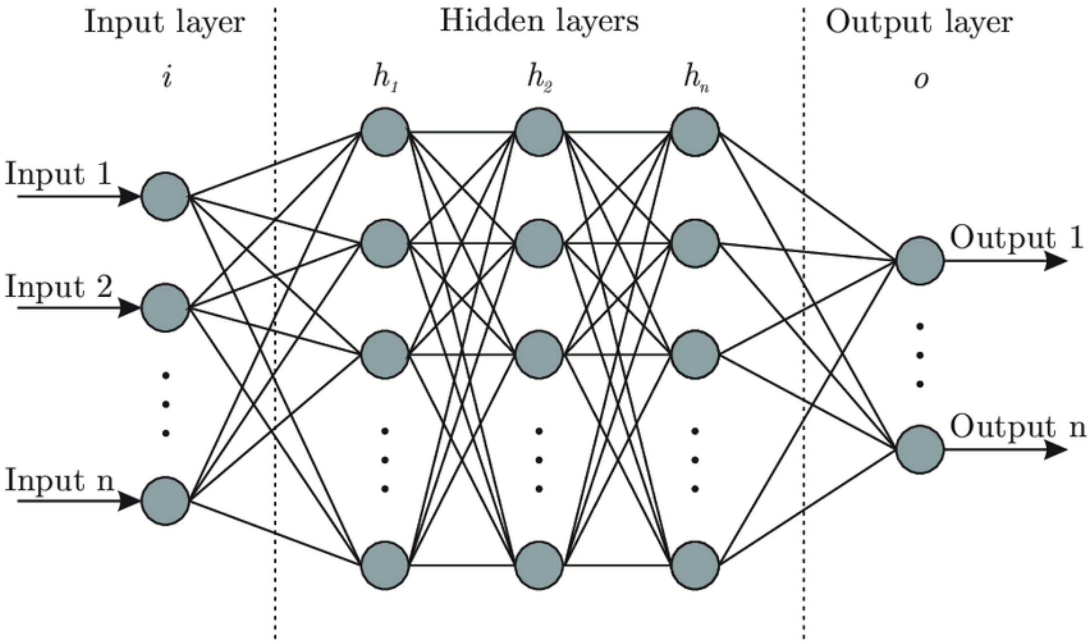

Figure 2.1.2: A Multilayer Perceptron [9]

2.1.2 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is a type of Neural Network where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

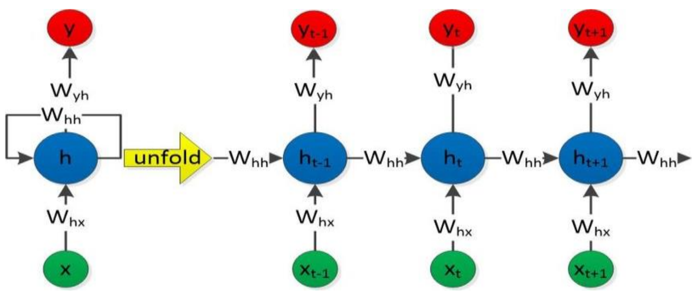

Figure 2.1.2: The-architecture-of-RNN [11]

2.1.3 Learning Process

Artificial NN can predict randomly then predictions are compared with the correct output and the error. The difference is called cost function or error. Training a NN actually refers to minimizing the error. Optimizer uses the loss value to update the networks weight.

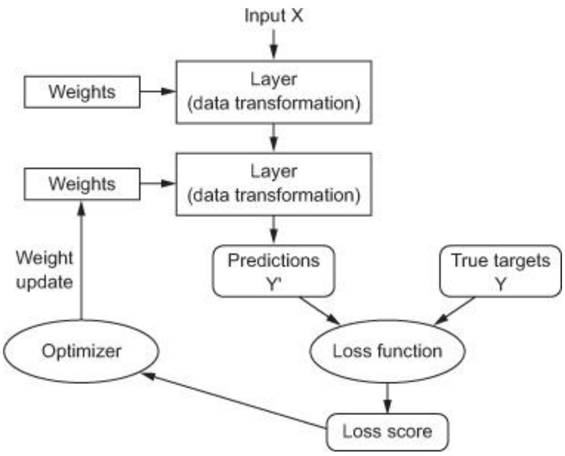

Figure 2.1.3: The Learning Process [14]

2.1.4 Vanishing Gradient

The vanishing gradient problem is an issue that sometimes arises when training machine learning algorithms through gradient de-scent. This most often occurs in neural networks that have several neuron layers such as in a deep learning system, but also occurs in recurrent neural networks. The key point is that the calculated partial derivatives used to compute the gradient as one goes deeper into the network. Since the gradients control how much the net-work learns during training, if the gradients are very small or zero, then little to no training can take place, leading to poor predictive performance.

2.1.5 Long-Short Term Memory

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more. LSTMs are a complex area of deep learning. LSTM remove the vanishing gradient problem. For this task, we use an LSTM since we would like to predict each word by looking at words that come before it and LSTM can maintain a hidden state that can transfer information from one time step to the next. See diagram below for how LSTM works

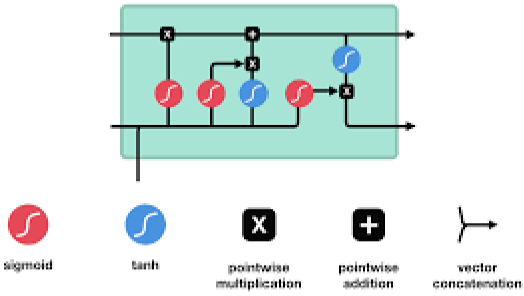

Figure 2.1.5: LSTM Model [12]

Input gate -Discover which value from input should be used to modify the memory. Sigmoid - Function decides which values to let through 0, 1. tanh- Function gives weightage to the values which are passed deciding their level of importance ranging from-1 to 1. Forget gate - Discover what details to be discarded from the block. It is decided by the sigmoid function. It looks at the previous state (ht-1) and the content input (Xt) and outputs a number between0and 1(keep this) for each number in the cell state Ct1.Output gate - The input and the memory of the block is used to decide the output. Sigmoid-function decides which values to let through 0, 1.tanh-function gives weightage to the values which are passed deciding their level of importance ranging from-1 to 1 and multiplied without put of Sigmoid.

2.2 Related Work

Robert Ostling and Johannes Bjervas proposed a model which was constructed with sequence-to-sequence artificial neural network and LSTM architecture that was a big attention to enthusiasts. Robert Ostling and Johannes Bjervas proposed a model which was constructed with sequence to sequence model of Bahdanauet is proper for this task. Abu Kaiser Mohammad Masum and et al. working with Bengali Text generations using Bi directional RNN Which is an extended version of recurrent neural network.[1]. Praveen Krishnan and et al. introduce an OCR system which pursues a combined architecture in seven different languages of India and a segmentation free method [4]. Their system was proposed to assist continuous learning in the time of being usable, like continuous user input. They worked with the BLSTM method, another form of general LSTM. Masami Nakamura and et al. Working Neural Network Approach towords Category Prediction for English text where they use Netgramand improve the word recognition rate from 81.0 percent to 86.9percent[5].N. Siddique et al. they proposed two methods for suggestion task one for single sequences of words, other suggests multiple next works. First one is an encoder-decoder based model that takes the already written text to understand the context and generate the word sequence next to it. Their context-based model is very accurate and good for understanding and remembering context word long before in the sequence. And second method is a character level sequential model to suggest the next words. This method is a little variation of other that uses beam search to suggest more than one word based on nearby context but the downside is, it cannot capture long-term dependency and this method is not very good for suggesting long sequence. But it can suggest many possible candidates for the next word. In the case of the first method, They combined one most probable long sequence suggestion and traditional frequency based suggestion system to give rest of the suggested word in the assistant system.[8] She Takase et al. proposed a method that constructs word embed-dings from character n-gram embedding and combines them with ordinary word embedding. Their proposed method achieves the best perplexities on the language modeling datasets: Penn Tree-bank, WikiText-2, and WikiText-103. Moreover, They experiments 10 on application tasks: machine translation and headline generation. The experimental results indicate their proposed method also positively affects these tasks. AMNA (Automated Marking of Narrative Answer) system was proposed in [1], by Miah et al for narrative type responses. This system performed easy marking and comprehensive feedback within a short time. By obtaining quick feedback and results of examination or assignment or reports students' can identify and understand the areas of their weakness and address them immediately in order to improve their learning. Author conducted a preliminary experiment using document similarity (Cosine similarity), Spelling check, Cosine similarity of important keyword, Similarity without stop word by using advanced Natural Language Processing and Machine Learning technologies. For measure efficiency of the system authors obtain human given marks to which machines gen-rated mark will be compared, Author uses two human markers for the students' answers and were separately marked. By doing this the system's generated mark were compared with accurate marks of each responses and average marks of inter-rater-agreement. Authors keep the Semantic Similarity, Grammar check, latent semantic similarity, ontology (knowledge) based similarity techniques as a venture for future work. Based on those techniques or proper combination of some techniques and with some additional measurement we can implement such a system that can provide an effective assessment of student responses which was able to measure semantic similarity and grammar check of student answer.

CHAPTER 3: METHODOLOGY

In this chapter we have explained details about our system. In section 3.1 we have discussed details working procedure of our system. In section 3.2 evaluation method is provided. Language Modeling is the most significant piece of present-day NLP. Here we build a model for the prediction of the next word for sentence completion.

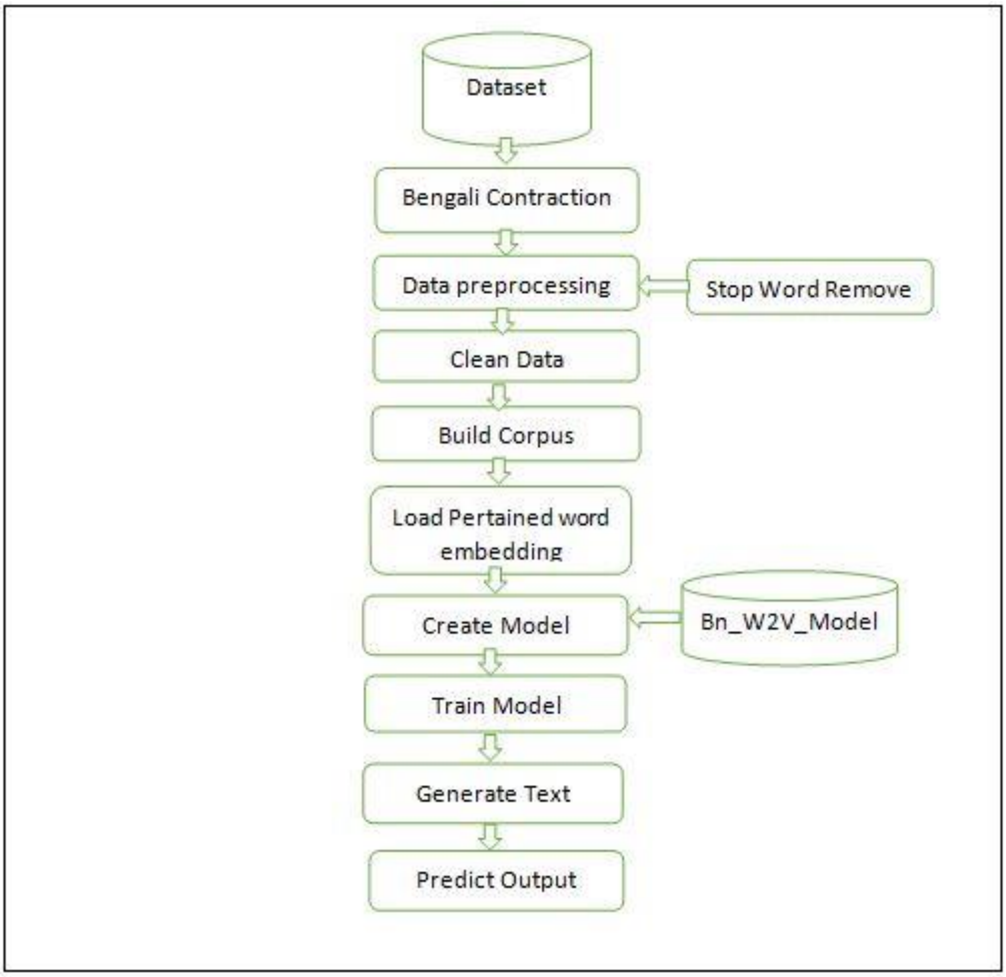

Figure 3.1: Working Flow of Text Prediction

3.1 Data Download and Preprocessing

Data preprocessing is a data mining technique that involves converting the raw data into a comprehensible format. Data from various sources is not compatible with input in neural network architecture. Formatted the data is important to achieve better and consistent results.

3.1.1 Data load

We have assembled a large amount of data in Bangla language to assessment the new approach that we have proposed, and the total amount of gathered data is 170 thousand, which were collected from different sources.[Bangla Data set Example]We manage the Bengali dataset from an online newspaper. After collecting datasets from different sources. There is a huge amount of data so we split line by line. Then we have to clean the dataset using the cleaning function to remove the unwanted objects. This cleaning function also helps to turn the initial dataset into a standard one. Afterward, every of data is stored as a token. Then we measure the length of the token needed for pre-processing.

3.1.2 Stop word Removing

Stop words are a set of commonly used words in a language. Examples of stop words in English are “a”, “the”, “is”, “are” and etc. The intuition behind using stop words is that, by removing low information words from text, we can focus on the important words instead.

3.1.3 Punctuations and White Space Removing

To remove all special characters, punctuation and spaces from string, iterate over the string and filter out all non-alpha numeric characters .For example

3.1.4 Generating Word Vector

In NLP, one the first tasks is to replace each word with its word vector as that enables a better representation of the meaning of the word. That’s why we sequence the token and store these in an array.

In semantic similarity measure between model answer and each student answer we require a model answer and student answer. These answers are passed through preprocessing module as discussed above. Since different students' response are in different pattern but it may be semantically similar with model answer. So if we compare directly with model answer then it may not similar in terms with the model answer but could be similar in concept. For solution to this problem we find the synonyms of model answer without stop words. Then we perform union of those synonyms with original terms in model answer and find cosine similarity. To find cosine similarity we need to vectories both student and model answer.

3.1.5 CLEAN DATASET

Performing the following task for cleaning:

1 .Remove the unnecessary attributes

2 .Select the essential attributes

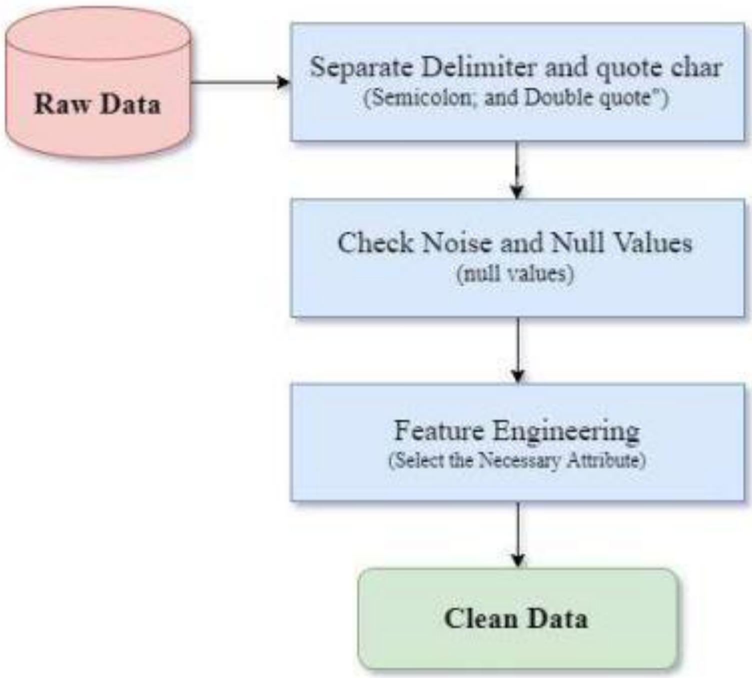

Figure 3.1.5: The Block Diagram of Data Cleaning

3.1.6 LSTM Learning Process

Ensemble methods generate multiple learners and aggregate them to provide a composite prediction. Among them, the Bagging and Boosting method are most popular. The diversity of individual learner is an important issue for ensemble model, which can be achieved by selecting and combining the training examples or the input features, injecting randomness into the learning algorithm [34,36].The proposed LSTM algorithm is an ensemble method utilizing LSTM as base learner. Two random strategies are employed to produce different training subsets, hence constructing a number of base LSTM classifiers. All predictions are integrated to give a comprehensive estimate of the outcome. Given a training set with N training instances, each instance can be represented as (V, Y). V is a matrix containing values of D variables and T sequences. It can be written as, as expressed in equation (1). Is a vector given in equation (2).represents the value of the d-th variable at t-th time step. And Y is the target label for the instance taking 0 (negative) for survival and 1 (positive) for death. The ratio of negative and positive group size is denoted as: LSTM has the advantage of capturing temporal information and is popular to be adopted in time series modeling. Detailed structure of the LSTM block is illustrated in Figure 1.

$$V = [X_1, X_2, X_3, \ldots, X_t, \ldots, X_T], \tag{1}$$

$$X_t = [x_t^1, x_t^2, x_t^3, \ldots, x_t^d, \ldots, x_t^D]. \tag{2}$$

For this task, we use an LSTM since we would like to predict each word by looking at words that come before it and LSTMs can maintain a hidden state that can transfer information from one time step to the next. See diagram below for how LSTM works [4]. Then, the output of hidden layer, namely, the current hidden state , is computed as follows: where and are the forget, input, and output gates, respectively. Is the previous hidden state. And are previous and current cell memories.15

The weight matrices and the bias vectors and are model parameters. The symbol is the sigmoid function and hyperbolic tangent function. The symbol denotes matrix multiplication and element wise product. A sigmoid layer is applied on the output of the LSTM block at final step for binary classification. The predicted score is computed as equation (4). The loss function is the weighted cross entropy of real label and predicted score with positive instances weighted and negative ones weighted 1. The parameters within the net are updated over several iterations to reach the minimum loss value:

$$\bar{y} = \sigma\left(w_{ho} \cdot h_T + b_{ho}\right). \qquad (4)$$

3.2 Proposed Method

In general an LSTM network is complex comparative to other methods. It consume a much power in hardware machines capability. The whole interior activities and logic flow could be presented as below-

1) Input: Firstly, The input is squashed with the tanh activation function between-1 and 1. This could be expressed by-

$$g = \tanh\left(b^g + t_t U^g + h_{t-1} V^g\right) \ldots\ldots 2$$

Where U_g and Vg are the previous weights of cell output and inputs. In other side b_g is performing as an input bias. Remember, the exponents (g) is only considering as input weights.

$$i = \sigma\left(b^i + X_t U^i + h_{t-1} V^i\right) \ldots\ldots 3$$

The equation 4 is considered as output of LSTM input section-

$$g \circ i \qquad \ldots\ldots 4$$

Here the $\circ$ is elements-wise multiplication.

2) Forget state loop and gate: the output forgotten gate expression is-

$$f = \sigma\left(b^f + X_t U^f + h_{t-1} V^f\right) \ldots\ldots 5$$

The product output shows the position of previous state and forgotten gate. The equation for this calculation is-

$$S_{t-1} \circ f \qquad \ldots\ldots 6$$

The output of forgotten loop is calculated in another strategy. For different time frame-

$S_t = S_{t-1} \circ f \ + g \circ i$ ……………………………………………………7

3) Output gate: Necessary output gate is evolved as-

$\circ = \sigma \, (b\circ \, X_t \, U^0 + h_{t-1} V^0)$………………………………………………..8

So that the cell final output, with tanh squashing, can be expressed as –

$h_t = \tanh\,(s_t) \circ O$ ……………………………………………………9

Finally, a very common form of LSTM networks equation can be written from Colahs famous blog post-

$$z_t = \sigma(W_z.[h_{(t-1),x_t}])$$
$$z_t = \sigma(W_r.[h_{(t-1),x_t}])$$

$$h_{t1} = \tanh\,(W.[(r_t * h_{(t-1),x_t}])$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * h_{t1}$$

That’s how the Long-short-term-memory (LSTM) network do the operations sequentially. That’s why it perform superior in any type of sequential data. The LSTM network activity flow could be presented as the figure given below. There we can notice some time evaluation term what’s for LSTM is different.[10]

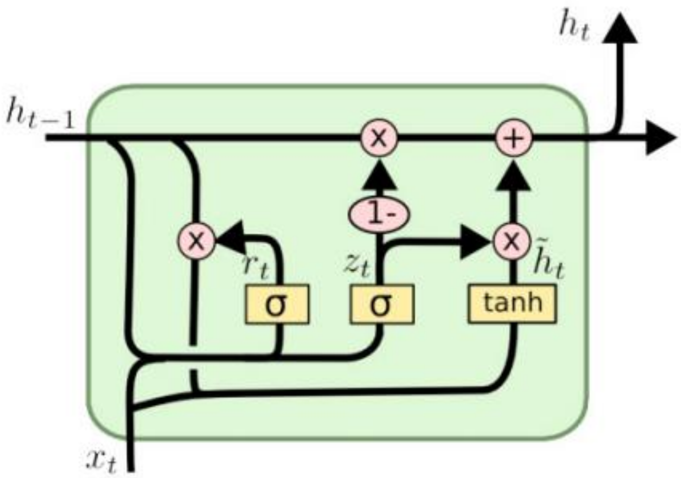

Fig 3.2: LSTM networks activity flow [10]

3.2.1 Layer Description

Generally a neural network contains three layers for taking input, doing calculation and giving decision. An input embedding layer was taken as initial layer of neural network as input layer. Here a single line of text is being trained one after one and sequentially. Then the hidden layer was taken place. It could be explain as the main LSTM layer and did it for 100 units. The final and output layer is described now. An activation function is applied here named softmax. Softmax calculates the probability of event distribution over n events. This function generally calculates the probabilities of each target class across all possible target classes.

3.2.2 Model Validation

The LSTM model is little different in validation perspective. Performance determination with cross validation or train-test accuracy in general like CNN model is not practical. It actually better to test the model with real data and its output. We trained news paper corpus for having limitation of hardware limitation. And finally did test with different Bengali words, then the model generated some text according to previous text. Here are two generated Bangla sentences with our model.

```
print(generate_seq(model, tokenizer, 'নিশ্চিত ', 6))
```

```
Epoch 500/500
1/1 - 0s - loss: 0.0999 - accuracy: 0.9643
/usr/local/lib/python3.7/dist-packages/keras/engine/sequ
  warnings.warn('`model.predict_classes()` is deprecated
নিশ্চিত  হয়ে গেছে জানাচ্ছে ক্রিকেটবিষয়ক ওয়েবসাইট ক্রিকইনফো
```

CHAPTER 4: EXPERIMENTAL RESULT AND DISCUSSION

In this chapter we have described our data collection, required tools, experimental results and discussion of them. In section we 4.1 have discussed data collection. In section 4.2 we have focused on required tools of our system. In section 4.3 and 4.4, our experiment, results and their analysis are given.

4.1 Data Set

We used Bengali text data for our system. We have collected data from different source of newspaper. But it is difficult task because of there are too noisy and not suitable for working with machine learning or deep learning process. We did some making processing work our dataset noise free. We used two types dataset, one is short length dataset and another is large.

4.2 Required Tools

TensorFlow is used [TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.] to implement our system model. It is a platform used for building Python programs that work with human language data for applying in statistical NLP (Natural Language Processing). We need a programming language to implement our system. We have used **Python 3.6.8,** which is one of the versions of python programming language that enables us to work more quickly and integrate systems more effectively. We have used **Memory 8GB** RAM, Processor i7 in our personal computer to get a speedy processing implementation. Though we implemented our system in **Windows 10** Operating System, users can easily use this system by installing required libraries on any kind of operating system.

4.3Neural Network Parameters:

4.3.1 ACTIVATION FUNCTION

It is the function that you use to get the output of node (yes or no). It is also known as Transfer function. It maps the resulting values in between 0 to 1 or -1 to 1 etc. (depending upon the function) [17]. The Activation Functions can be basically divided into 2 types

1. Linear Activation Function
2. Non-linear Activation Functions

4.3.1.1 RELU (RECTIFIED LINEAR UNIT)

ReLU is half rectified. If f(z) is zero when z is less than zero and f(z) is equal to z when z is above or equal to zero. Range: (0 to infinity) [17]

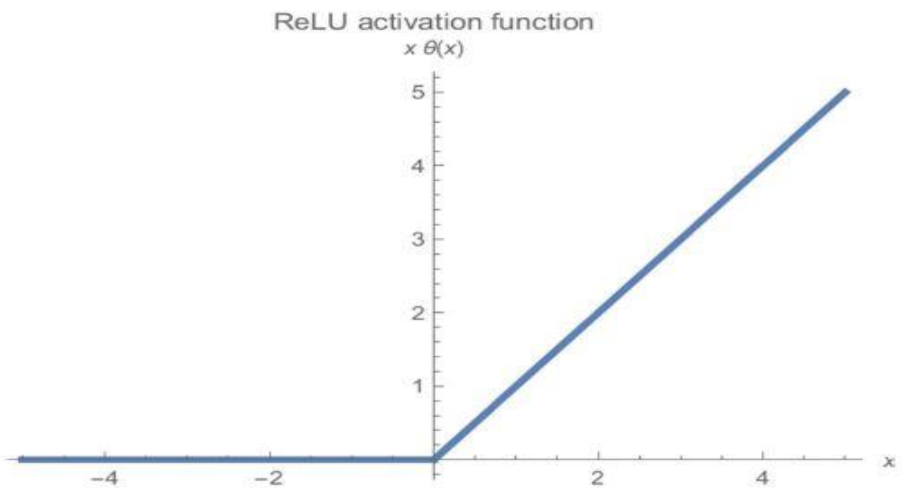

Figure 4.3.1.1 RELU (Rectified Linear Unit) [13]

4.3.1.2 SOFTMAX

Softmax function used the final layer for multiple classes. normalizes the outputs for each class between 0 and 1, and divides by their sum, giving the probability of the input value being in a specific class [17].

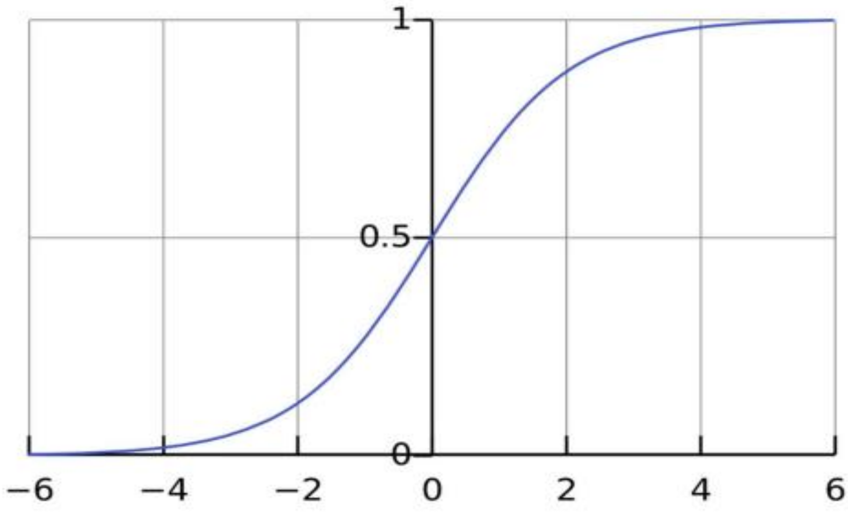

Figure 4.3.1.2 Softmax Function[17]

4.3.2 OPTIMIZER

Optimization is a process of searching for parameters that minimize or maximize our Functions. During the training process, we tweak and change the parameters (weights) of our model to try and minimize that loss function, and make our predictions as correct as possible [16].

4.3.2.1 ADAM OPTIMIZER

The Adam optimization algorithm is an extension to stochastic gradient descent. Adam is an optimization algorithm that can used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data. The authors describe Adam as combining the advantages of two other extensions of Stochastic gradient descent. Specifically:

Adaptive Gradient Algorithm (AdaGrad)

Adagrad maintains a per-parameter learning rate that improves performance on Problems with sparse gradients (e.g. natural language and computer vision problems).

Root Mean Square Propagation (RMSProp)

RMSProp also maintains per-parameter learning rates that are adapted based on the average of recent magnitudes of the gradients for the weight (e.g. how quickly it is 26 changing). This means the algorithm does well on online and non-stationary problems (e.g. noisy).[15]

4.4 Experiment Datasets

We have experiment with three types of dataset. We see that result of accuracy and loss are varied with different kind of dataset. Firstly we used small length dataset that have vocabulary 25, Total Sequence is 25.Figure is given below:

```
Vocabulary Size: 25
Total Sequences: 25
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_5 (Embedding)      (None, 1, 10)             250
_________________________________________________________________
lstm_5 (LSTM)                (None, 50)                12200
_________________________________________________________________
dense_5 (Dense)              (None, 25)                1275
=================================================================
Total params: 13,725
Trainable params: 13,725
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.4.1 Configuration of dataset 1

Secondly we used medium length dataset that have vocabulary 100, Total Sequence is 117. Figure is given below:

```
Vocabulary Size: 100
Total Sequences: 117
Model: "sequential_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_8 (Embedding)      (None, 1, 10)             1000
_________________________________________________________________
lstm_8 (LSTM)                (None, 50)                12200
_________________________________________________________________
dense_8 (Dense)              (None, 100)               5100
=================================================================
Total params: 18,300
Trainable params: 18,300
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.4.2 Configuration of dataset 2

Finally we used medium length dataset that have vocabulary 1259, Total Sequence is 3936. Figure is given below:

```
Vocabulary Size: 1259
Total Sequences: 3936
Model: "sequential_12"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_12 (Embedding)     (None, 1, 10)             12590
_________________________________________________________________
lstm_12 (LSTM)               (None, 50)                12200
_________________________________________________________________
dense_12 (Dense)             (None, 1259)              64209
=================================================================
Total params: 88,999
Trainable params: 88,999
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.4.3 Configuration of dataset 3

4.5 Result and Accuracy Analysis

To ratify the proposed approach, it's essential to run the experiments and analyzed the outcome earnestly. Hence, we have appraised our proposed approach on a corpus dataset having identical structures until 500 epochs. We used datasets 1 and found loss 0.1632 and accuracy 92% .The following iteration of train loss, train accuracy produced when we train our different model:

```
Epoch 491/500
1/1 - 0s - loss: 0.1667 - accuracy: 0.9200
Epoch 492/500
1/1 - 0s - loss: 0.1663 - accuracy: 0.9200
Epoch 493/500
1/1 - 0s - loss: 0.1659 - accuracy: 0.9200
Epoch 494/500
1/1 - 0s - loss: 0.1655 - accuracy: 0.9200
Epoch 495/500
1/1 - 0s - loss: 0.1651 - accuracy: 0.9200
Epoch 496/500
1/1 - 0s - loss: 0.1647 - accuracy: 0.9200
Epoch 497/500
1/1 - 0s - loss: 0.1644 - accuracy: 0.9200
Epoch 498/500
1/1 - 0s - loss: 0.1640 - accuracy: 0.9200
Epoch 499/500
1/1 - 0s - loss: 0.1636 - accuracy: 0.9200
Epoch 500/500
1/1 - 0s - loss: 0.1632 - accuracy: 0.9200
```

Figure 4.5.1 Epoch of dataset 1

Loss and Accuracy Curve for dataset 1 are given below:

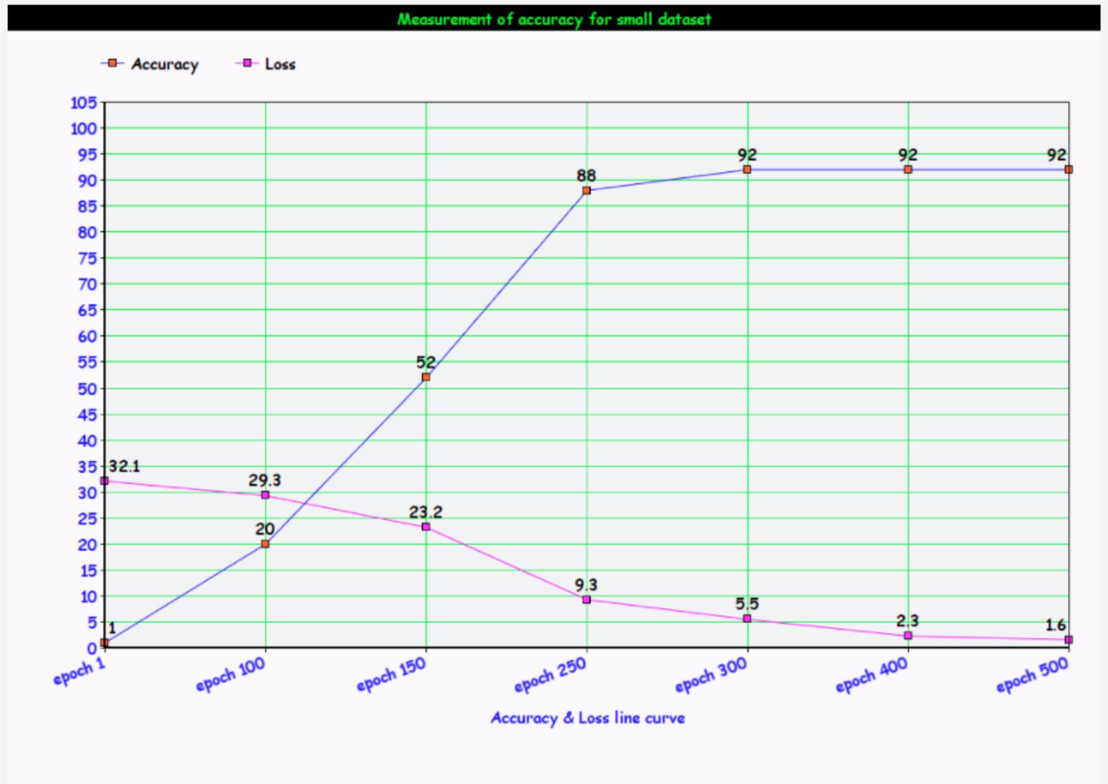

Figure 4.4.1 Curve of dataset 1

We used datasets 2 and found loss 0.1632 and accuracy 84.62% .The following iteration of train loss, train accuracy produced when we train our different model:

```
Epoch 490/500
4/4 - 0s - loss: 0.2577 - accuracy: 0.8291
Epoch 491/500
4/4 - 0s - loss: 0.2572 - accuracy: 0.8376
Epoch 492/500
4/4 - 0s - loss: 0.2572 - accuracy: 0.8291
Epoch 493/500
4/4 - 0s - loss: 0.2572 - accuracy: 0.8376
Epoch 494/500
4/4 - 0s - loss: 0.2571 - accuracy: 0.8376
Epoch 495/500
4/4 - 0s - loss: 0.2567 - accuracy: 0.8291
Epoch 496/500
4/4 - 0s - loss: 0.2566 - accuracy: 0.8205
Epoch 497/500
4/4 - 0s - loss: 0.2569 - accuracy: 0.8291
Epoch 498/500
4/4 - 0s - loss: 0.2567 - accuracy: 0.8376
Epoch 499/500
4/4 - 0s - loss: 0.2566 - accuracy: 0.8120
Epoch 500/500
4/4 - 0s - loss: 0.2566 - accuracy: 0.8205
```

Figure 4.4.1 Epoch of dataset 2

Loss and Accuracy Curve for dataset 2 are given below:

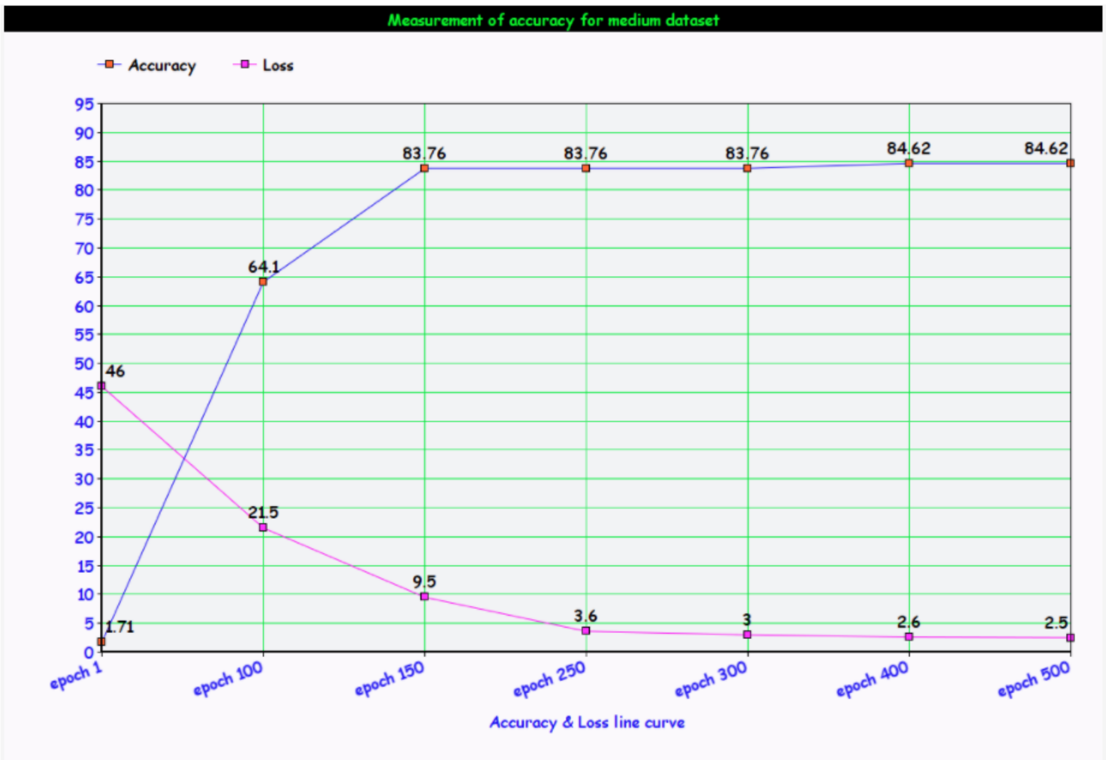

Figure 4.4.1 Curve of dataset 2

We used datasets 3 and found loss 1.29 and accuracy 50.48% .The following iteration of train loss, train accuracy produced when we train our different model:

```
Epoch 491/500
123/123 - 0s - loss: 1.2993 - accuracy: 0.5086
Epoch 492/500
123/123 - 0s - loss: 1.2983 - accuracy: 0.5051
Epoch 493/500
123/123 - 0s - loss: 1.2995 - accuracy: 0.5053
Epoch 494/500
123/123 - 0s - loss: 1.2992 - accuracy: 0.5030
Epoch 495/500
123/123 - 0s - loss: 1.2994 - accuracy: 0.5046
Epoch 496/500
123/123 - 0s - loss: 1.2983 - accuracy: 0.5013
Epoch 497/500
123/123 - 0s - loss: 1.2989 - accuracy: 0.5046
Epoch 498/500
123/123 - 0s - loss: 1.2998 - accuracy: 0.5051
Epoch 499/500
123/123 - 0s - loss: 1.2987 - accuracy: 0.5048
Epoch 500/500
123/123 - 0s - loss: 1.2976 - accuracy: 0.5048
```

Figure 4.4.1 Epoch of dataset 3

Loss and Accuracy Curve for dataset 3 are given below:

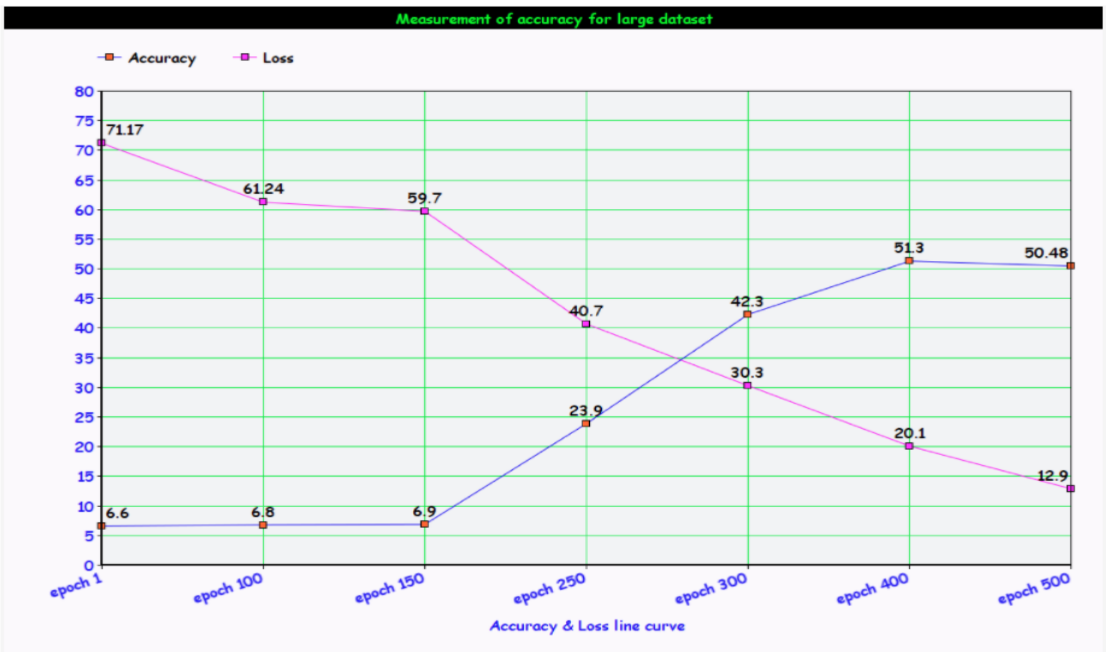

Figure 4.4.1 Curve of dataset 3

Overall view of loss and accuracy curve:

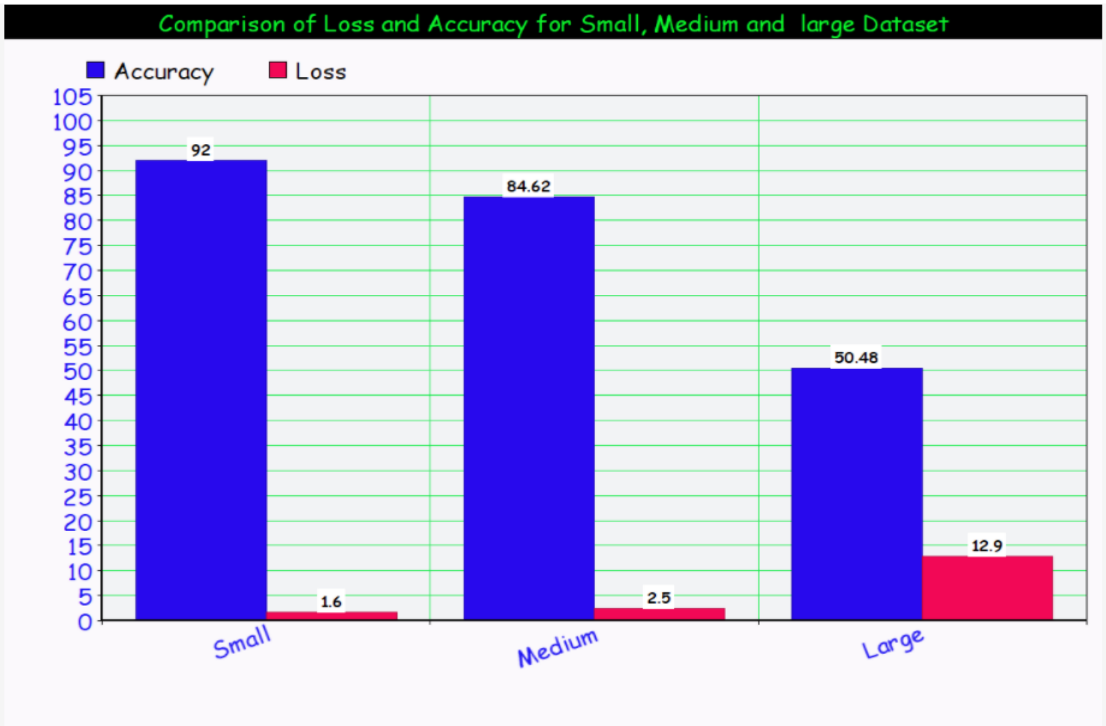

Figure 4.4.1 Comparison curve of loss and accuracy

4.6 EXPERIMENTAL TOOLS

Some experimental tools we used in the model building that are describe shortly below:

GOOGLE COLAB

Google colab is python 3 google computing engine backend.it has 12.69 GB ram and 107.72 GB disk.

ANACONDA

Anaconda is free and open source distribution the python programming language. Package versions are managed by the package management system conda.

PANDAS

Pandas is a software library that was written in python programming language. It is widely used for data manipulating and analysis. It is easy to use with dataset.

KERAS

Keras is high level Neural Networks API. It is written by python programming language and also capable of running on top of Tensorflow, Theano, CNTK. Here we used tensorflow backend. It runs seamlessly on both CPU and GPU.It is used for various experiment on Neural Networks with speedy processing. It allows for easy and fast prototyping[8].

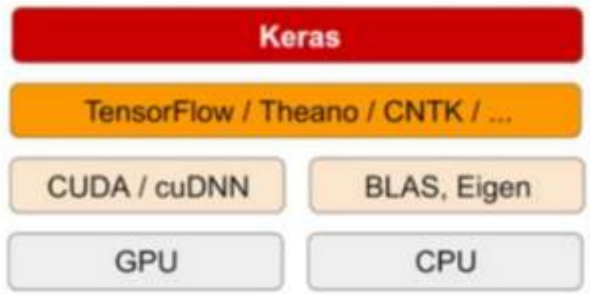

Figure 4.6 The deep learning software and hardware stack [8]

NUMPY

Numpy is a python package for scientific computing. There are some main feature in numpy such as n dimensional object, sophisticated functions, random number generator, Fourier transform, linear algebra etc. It is also efficient in multi-dimensional container data.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion

We have described our research about Bengali Sentence Completion. We have implanted a

neural network model for Bengali sentence prediction on the basis of previous seed text. We have used our model for different kind of dataset that's why we observed variation in our result accuracy level. A noticeable thing model execution time would be different based on different platform and machine capacity. Word prediction model helping tool that auto completes writing, both by completing current words and suggesting relevant word sequences. Smart typing assistant is the one that can suggest more than just the current word. Our system achieved a highest accuracy of 75.51%. Average accuracy based on our dataset is of 50.1%. We used Google Colab platform, in this platform provide 12 GB RAM that's why when we try to large dataset then system would be fall for insufficient RAM.

5.2 Future Work

In this paper we worked with less data, due to hard ware limitations .After wards we will enhance our dataset .In future we will improve the model for achieving multitask sequence to sequence text generation and multi way translation like Bengali articles ,caption generation. Furthermore, we would aim to pursue the possibility of extending our model to Bangla regional languages. We also has plan to work with Bangla Sign Language[23] generation with sequential image data as like general people language.

Reference

[1] Sheikh Abujar, Abu Kaisar Mohammad Masum, SM Mazharul Hoque Chowdhury, Mahmudul Hasan, and Syed Akhter Hossain. Bengali text generation using bi-directional rnn. In 2019 10th International Conference on Computing,Communication and Networking Technologies (ICCCNT), pages1{5. IEEE, 2019.

[2] Pauziah Mohd Arsad, Norlida Buniyamin, et al. A neural network stu-dents' performance prediction model(nnsppm). In2013 IEEE InternationalConference

on Smart Instrumentation, Measurement and Applications (IC-SIMA),pages 1–5. IEEE, 2013.Pauziah Mohd Arsad, Norlida Buniyamin, et al. A neural network stu-dents' performance prediction model (nnsppm). In2013 IEEE InternationalConference on Smart Instrumentation, Measurement andApplications (IC-SIMA), pages 1–5. IEEE, 2013.

[3] Partha Pratim Barman and Abhijit Boruah. A rnn based approach for next word prediction in assamese phonetic transcription.Procedia computer science, 143:117–123, 2018.

[4] Md Sanzidul Islam, Sadia Sultana Sharmin Mousumi, Sheikh Abujar, and Syed Akhter Hossain. Sequence-to-sequence bangla sentence generation with lstm recurrent neural networks.Procedia Computer Science, 152:51–58,2019.

[5] Masami Nakamura, Katsuteru Maruyama, Takeshi Kawabata, and KiyohiroShikano. Neural network approach to word category prediction for english texts. InCOLING 1990 Volume 3: Papers presented to the 13th International Conference on Computational Linguistics, 1990.

[6] Omor Faruk Rakib, Shahinur Akter, Md Azim Khan, Amit Kumar Das,and Khan Mohammad Habibullah. Bangla word prediction and sentence completion using gru: an extended version of rnn on n-gram language model.In2019 International Conference on Sustainable Technologies for Industry4.0 (STI), pages 1–6. IEEE, 2019.

[7] Md Shovon, Hedayetul Islam, and Mahfuza Haque. An approach of improv-ing students academic performance by using k means clustering algorithm and decision tree.arXiv preprint arXiv:1211.6340, 2012.

[8] F. Chollet, "Deep Learning with Python," p. 58, 2017.

[9] "https://machinelearningmastery.com/how-to-choose-loss-functions-when training-deep-learning-neural-networks/."

[10] "https://shrikar.com/deep-learning-with-keras-and-python-for-multiclass classification/."

[11] "https://dev.to/hydroweaver/single-label-multiclass-classification-using-keras 45jl."

[12] http://www.mit.edu/~9.520/spring09/Classes/multiclass.pdf.

[13] https://michielstraat.com/talk/mastertalk/.

[14] https://uclanlp.github.io/CS269-17/slides/CS269-03.pdf.

[15] https://machinelearningmastery.com/adam-optimization-algorithm-for -deep-learning/."

[16] https://algorithmia.com/blog/introduction-to-optimizers.

[17] https://towardsdatascience.com/activation-functions-neural-networks1cbd9f8d91d6.

[18] https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-%0Aand-examples-bf025f872908 %0A

[19] https://www.statisticshowto.datasciencecentral.com/inter-rater-reliability/.

[20] https://dev.to/hydroweaver/single-label-multiclass-classification-using-keras45jl

.