

# Conversion of Bengali Speech to Text Using Long Short-Term Memory(LSTM)

Mohammad Fahim Chowdhury

17101293

Zakia Sultana

17301182

Nusrat Jahan

17101332

Safkat Hasin Alavi

17101515

This thesis report is submitted to the Department of Computer Science  
and Engineering in partial fulfillment of the requirements for the degree of B.Sc. in  
Computer Science

Department of Computer Science and Engineering

Brac University

January, 2021

© 2020. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis that is being submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

## Student's Full Name & Signature:

*Fahim*

---

Mohammad Fahim Chowdhury  
17101293

নুসরাত জাহান

---

Nusrat Jahan  
17101332

*Zakia Sultana*

---

Zakia Sultana  
17301182

*Safkat*

---

Safkat Hasin Alavi  
17101515

# Approval

The thesis project titled “Conversion of Bengali Speech to Text Using Long Short-Term Memory(LSTM)” submitted by

1. Mohammad Fahim Chowdhury (17101293)
2. Zakia Sultana (17301182)
3. Nusrat Jahan (17101332)
4. Safkat Hasin Alavi (17101515)

Of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 15, 2021.

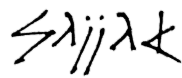
## Examining Committee:

Supervisor:  
(Member)

---

Mohammad Zavid Parvez, PhD  
Assistant Professor  
Department of CSE Coordinator of Software Engineering and HCI(SE-HCI)  
Research Group, BRAC University

Co-Supervisor:  
(Member)



---

Sajjad Ahmed  
Data Scientist  
Axiata Analytics, Axiata

Program Coordinator:  
(Member)

---

Md. Golam Rabiul Alam, PhD  
Associate Professor  
Department of Computer Science and Engineering  
BRAC University

Head of Department:  
(Chair)

---

Mahbulul Alam Majumder, PhD  
Professor and Dean, School of Data and Sciences  
Department of Computer Science and Engineering  
Department of Mathematics and Natural Sciences  
BRAC University

## Abstract

Speech to text conversion is a remarkable topic in the field of Artificial Intelligence which is undoubtedly a significant medium of expressing human feelings and thoughts. However, if we compare it with text to speech, work in speech to text conversion has been done less. Among those works, many languages got priority but the numerical value of work in Bengali language is little. Previously a similar work has been done in that language where they got 82.35% accuracy using LSTM[15]. Our approach was to avail more accuracy in speech to text conversion using Neural Network models. We build a novel dataset for research purposes. We tried both GRU and LSTM and focused on LSTM later on. The reason behind it is, GRU showed an unstable and started fluctuating where LSTM is much more stable and minimized errors in case of loss function and the accuracy was also less compared to LSTM. An increasing number of datasets was giving better accuracy and on the whole dataset, the accuracy on testing data is around 90%. In terms of loss function, testing loss is less than 40%. We did data testing manually to justify the result with the given output and we got 90% accuracy rate in a dataset which the model never fed before. In the future, we would like to work with automatic sentence recognition, the process of preparing the response basis of the statement, and also changing sentiment depending on it.

## **Acknowledgement**

We would like to start by thanking Almighty Allah for His blessings on us, without which nothing was possible. Then we will thank our respected supervisor Mohammad Zavid Parvez sir and co-supervisor Sajjad Ahmed for their tremendous support throughout the research. Lastly, I would like to mention about the guideline, support, and knowledge we got from all the faculties and our parents helped us a lot in many ways.

# Contents

<b>Declaration</b>	<b>i</b>
<b>Approval</b>	<b>ii</b>
<b>Abstract</b>	<b>iv</b>
<b>Dedication</b>	<b>v</b>
<b>Acknowledgment</b>	<b>v</b>
<b>Table of Contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>x</b>
<b>Nomenclature</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Problem statement: . . . . .	1
1.3 Research Objective: . . . . .	2
1.4 Workflow: . . . . .	2
1.4.1 Research and planning . . . . .	2
1.4.2 Implementation . . . . .	4
1.4.3 Result analysis . . . . .	4
1.5 Timeline: . . . . .	4
<b>2 Related Work</b>	<b>6</b>
<b>3 Background Analysis</b>	<b>10</b>
3.1 Recurrent Neural Network . . . . .	10
3.1.1 Sequence Data . . . . .	10
3.1.2 Sequential Memory . . . . .	11
3.1.3 Vanishing Gradient . . . . .	15
3.2 Gated Recurrent Unit . . . . .	19
3.2.1 Update Gate . . . . .	20
3.2.2 Forget Gate . . . . .	20
3.2.3 Input Gate . . . . .	21
3.2.4 Reset Gate . . . . .	21

3.3	Long Short Term Memory . . . . .	21
3.3.1	Basic Concept of LSTM . . . . .	23
3.3.2	Steps of LSTM . . . . .	24
3.4	Confusion Matrix for evaluating the model . . . . .	27
<b>4</b>	<b>Methodology and Implementation</b>	<b>28</b>
4.1	Data Collection and Dataset . . . . .	28
4.1.1	Data Collection Process . . . . .	28
4.1.2	Dataset Description . . . . .	29
4.1.3	Exploratory Data Analysis(EDA) . . . . .	31
4.2	Pre-processing . . . . .	35
4.2.1	Dataset Splitting . . . . .	35
4.2.2	Standard Scaling . . . . .	35
4.2.3	Scaling the feature range . . . . .	35
4.3	Implementation . . . . .	35
<b>5</b>	<b>Result Analysis</b>	<b>39</b>
5.1	Comparison Between GRU and LSTM . . . . .	39
5.1.1	Loss Comparison . . . . .	39
5.1.2	Accuracy Comparison . . . . .	40
5.2	LSTM Analysis and Hypothesis . . . . .	41
5.2.1	Loss Function Analysis . . . . .	41
5.2.2	Accuracy Analysis . . . . .	42
5.3	Benchmark model using Confusion Matrix . . . . .	44
5.4	Limitations . . . . .	45
5.5	Error Analysis and Hypothesis . . . . .	45
<b>6</b>	<b>Conclusion</b>	<b>47</b>
6.1	Conclusion . . . . .	47
6.2	Future Work . . . . .	47
	<b>Bibliography</b>	<b>49</b>



# List of Figures

1.1	Test conversion from direct audio input . . . . .	2
1.2	Work Plan . . . . .	3
1.3	Timeline for actual work . . . . .	5
3.1	Working process of Recurrent Neural Network . . . . .	10
3.2	Present State . . . . .	11
3.3	Present State with previous value . . . . .	11
3.4	Audio spectrogram chopped into chunks[23] . . . . .	11
3.5	Typical Neural Network(Feed-Forward) . . . . .	12
3.6	Pass Hidden State to next state . . . . .	12
3.7	Segmentation of a sentence . . . . .	13
3.8	Feed the first word into RNN . . . . .	13
3.9	Feed the model maintaining the sequence . . . . .	14
3.10	Feed the full input by maintaining the sequence . . . . .	14
3.11	Make the final prediction . . . . .	15
3.12	Final hidden state . . . . .	15
3.13	Loss Function . . . . .	16
3.14	Calculating Lost Function[18] . . . . .	17
3.15	Calculating gradients through back-propagation . . . . .	18
3.16	Back-propagation over time . . . . .	18
3.17	GRU cell and its gate . . . . .	20
3.18	Operations of Forget Gate . . . . .	20
3.19	Input Gate Operations . . . . .	21
3.20	Single Layered LSTM . . . . .	22
3.21	Four Interacting Layered LSTM . . . . .	22
3.22	Notations used in LSTM . . . . .	23
3.23	Cell State of LSTM . . . . .	23
3.24	Gates of LSTM . . . . .	24
3.25	Removing information from cell state . . . . .	24
3.26	Adding information in cell state . . . . .	25
3.27	Multiplication of sigmoid and tanh layer . . . . .	25
3.28	The resultant step of LSTM . . . . .	26
3.29	Confusion matrix . . . . .	27
4.1	Gender ratio over dataset . . . . .	29
4.2	Gender wise age frequency . . . . .	30
4.3	Age wise population percentage . . . . .	30
4.4	Time wise Population . . . . .	31
4.5	Waveform of a stereo channel . . . . .	32

4.6	Waveform of a mono channel . . . . .	33
4.7	Male-Female ratio of dataset . . . . .	33
4.8	Age ratio of dataset . . . . .	34
4.9	Data points of each label . . . . .	34
4.10	Gender variance on each label . . . . .	35
4.11	Model Architecture . . . . .	38
5.1	Loss Function of GRU . . . . .	39
5.2	Loss Function of LSTM . . . . .	39
5.3	Accuracy of GRU . . . . .	40
5.4	Accuracy of LSTM . . . . .	40
5.5	Loss Function on one-third dataset . . . . .	41
5.6	Loss Function on two-third dataset . . . . .	42
5.7	Loss Function on full dataset . . . . .	42
5.8	Accuracy on one-third dataset . . . . .	42
5.9	Accuracy on two-third dataset . . . . .	43
5.10	Accuracy on full dataset . . . . .	43
5.11	Confusion Matrix on Manually tested dataset . . . . .	44
5.12	Spectrogram of same word . . . . .	45

# List of Tables

4.1	Hyperparameters . . . . .	36
5.1	Comparison of loss function and accuracy based on number of data .	44

# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

*AI* Artificial Intelligence

*ANN* Artificial Neural Network

*CNN* Convolutional Neural Network

*DNN* Deep Neural Network

*EDA* Exploratory Data Analysis

*FFT* Fast Fourier Transform

*GRU* Gated Recurrent Units

*HMM* Hidden Markov model

*LSTM* Long Short-Term Memory

*MRN* Home Team Run Rate

*NER* Named Entity Recognition

*NLP* Natural Language Processing

*RNN* Recurrent Neural Network

# Chapter 1

## Introduction

### 1.1 Introduction

Modern society needs a modern solution. As we are from a modern era, prediction is no longer confined to random guesses. It has validity, accuracy, and sequence. LSTM plays an important role in this case. LSTM, a recurrent neural network architecture, can work with sequences of data that have feedback connections. LSTM is highly recommended for time series prediction, speech recognition, next word or sentence prediction, text prediction, even handwriting prediction, prediction in medical care pathways, and many more. The best example can be the stock price predictor which is used to predict stock prices. It is done by following the LSTM model and some neural network algorithm. So, our approach is to convert a speech into text by using LSTM.

### 1.2 Problem statement:

Human beings are different from all other creatures as humans can talk. Moreover, human has the power to do innovations using their intelligence. After realizing it, they started to work on the field of their interest, curiosity, and also for the solutions to make an easier, better, and relaxing life for people.

In 1877, Thomas Alva Edison made the first device to record and reproduce sound by the phonograph. But it was not that effective. Later on, many people worked on voice recognition as people depend on technology more day by day and they want the easiest solution for everything. For example – robot Sophia. Hanson Robotics, a Hong Kong based company, introduced a robot that can interact with people and continue conversation with a human. But Sophia cannot reply in Bengali. So, works related to these fields based on the Bengali language with a high accuracy rate is still not up to the mark.

People have worked on speech to text less compared to text to speech. Moreover, there is not sufficient work on the Bengali language as people have focused on other international and national languages more. On the contrary, there is a huge number of people who speak in Bengali. We are already overpopulated, and all our country's people use the Bengali language to communicate. Even many people from our neighbor country India use this language. Not only that, now people prefer to go

to foreign countries to earn for their livelihoods. And most of them are uneducated. Moreover, around 27 percent of people are still illiterate in Bangladesh. That’s why they face difficulty when they have to write anything. Most importantly, we want to develop a system that will be helpful to the deaf people to understand.

After looking on these problems, we decide to build a system which will take audio input from human and convert the audio into a Bengali line which means it will convert the audio speech into Bengali text. It will help the uneducated to express their statement, the deaf people to understand others, and also it will help people to make their lifestyle easier and fast as they can use it in many ways to save their time, such as they can use the system to take notes, also when they are busy and have to write a text, they simply can say the message and send the text.

### 1.3 Research Objective:

In the beginning, we will create a speaker-independent dataset where we will record an audio clip from different speakers. At the same time, we will create another dataset based on the speech in a text format.

Our system will try to match the audio signal with our text-based dataset and translate the audio speech into text like Fig. 1.3. If in any case, it is not able to find the word, it will give the nearest best possible answer. So, our motto is to train our system to gain higher accuracy.

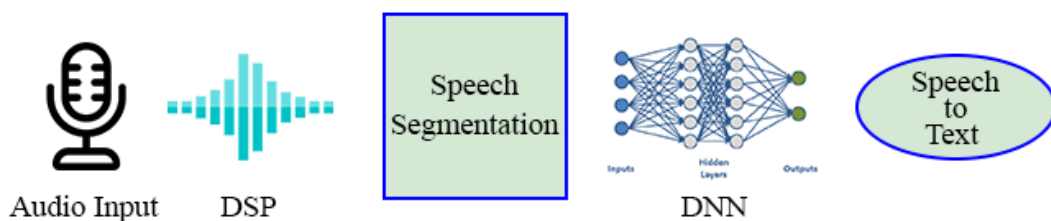


Figure 1.1: Test conversion from direct audio input

### 1.4 Workflow:

The workflow arrange that after whereas working on this proposal is given underneath:

#### 1.4.1 Research and planning

The research and planning stage are the fundamental portion. So we partitioned the assignment into two parts:

## 1.Literature Review

The subject of discourse to content conversion utilizing RNN which is an exceptionally well-known subject for inquiring. The writing survey incorporates doing inquire about almost past works done in this field. We attempted to discover related work in different places such as online libraries, journals, and institutional research papers.

## 2.Recent Research works

As time changes the work speech to text on RNN gets to be for progress. So we need to attempt to do our best to work agreeing for the progressions of our extend exceptionally vital portion.

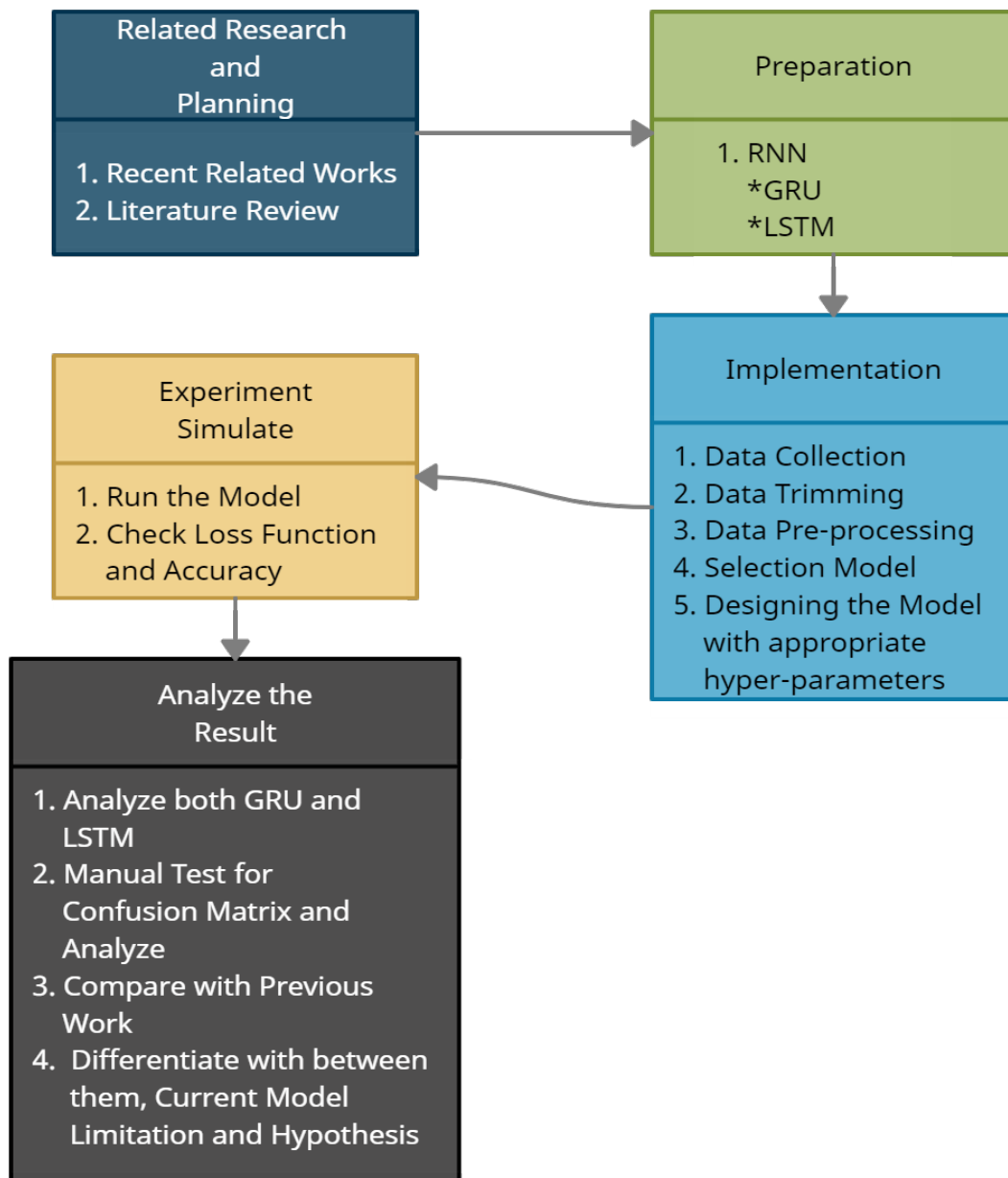


Figure 1.2: Work Plan

## 1.4.2 Implementation

- **Data collection:** We have to begin our work by collecting data to train our model. There are different sources for discourse to content information collection however as our work is in Bengali language we barely found any Data that is able to utilize. In this manner, we have chosen to make our own novel dataset that we'll utilize for our work.
- **Data pre-processing:** As we are aiming to do speech to text conversion so we have to do some data pre-processing before we fit it into our model. This handles incorporates changing over them to mono, trimming each word, scaling them, naming and keeping each word in a separate folder, and evacuating the word which is able to be articulated off-base.
- **Designing the model:** For designing the model, we have to first get the dataset prepared for work, and then we need to make sure that the environment is ready for coding. Here fundamental challenges are planning the architecture in such a way that we get mistakes as much less as conceivable when we run our model within the dataset. For that, we have to make the dataset clean and select the most excellent corresponding hyper-parameters to generate the most excellent result within the conclusion.

## 1.4.3 Result analysis

- **Justify the Result:** After getting the result we have to justify its accuracy and loss. We used Deep Neural Network as model and we will get best accuracy and less loss based on the performance. So we have to analysis those results and come up with a hypothesis how well the model is performing.
- **Compare the result with our hypothesis:** Our aim is to build a user independent system which can classify what a user is saying. So after preparing the model we have to manually justify if the model actually is working with user independent data or not. For that purpose we have to use completely different user who are not present in dataset.

## 1.5 Timeline:

We have planned the timeline of our venture for way better efficiency. We managed to do it on time as well. The timeline is appeared in Figure 1.2



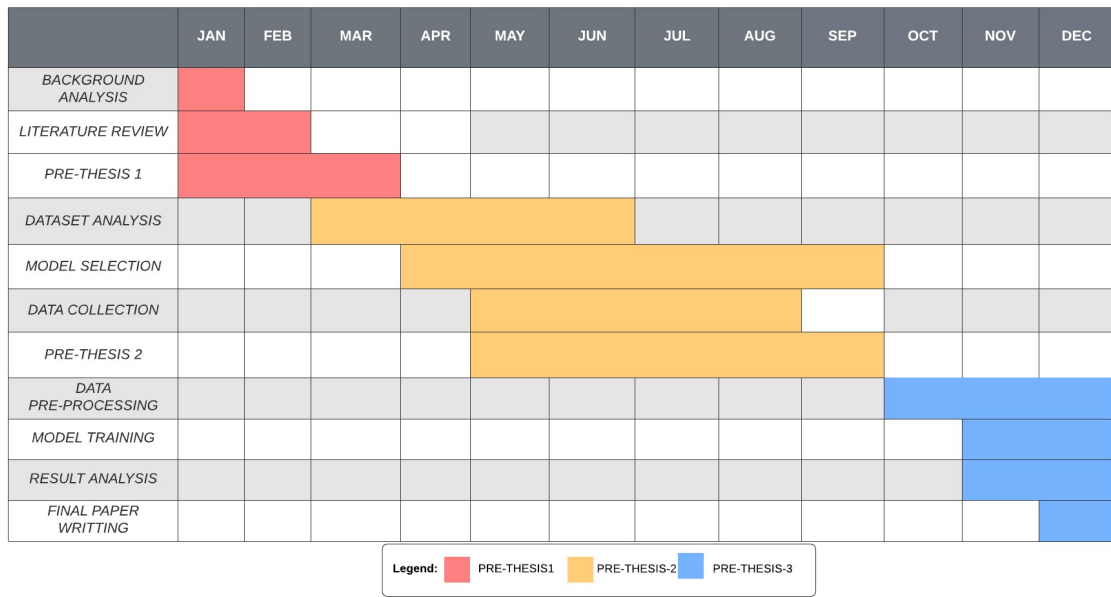


Figure 1.3: Timeline for actual work

# Chapter 2

## Related Work

The process of voice recognition is pretty interesting. Let's see some relevant research papers. In one of the papers[7], Vanajakshi and Mathivanan researched on discourse acknowledgment, a methodology where they used human discourse by the PC and making string yield fit as a fiddle. The database contains discourse signs of 92 speakers having age extends between 18 to 30, out of which 40 signs have a place with males and staying 50 signs has a place with a female. Graves et al. worked on speech to text conversion[2]. They mainly used Recurrent Neural Network. They combined BRNNs and LSTM. Their main purposed was end-to-end training. The model they used is Connectionist Temporal Classification and the softmax function is also available. CTF and a separate RNN are combined by an RNN transducer. It was decoded with beam search which is both effective and faster. Moreover, two regularization— early stopping and weight noise had been used for good performance. They experimented using the TIMIT corpus. They were able to decrease the error rate which was 17.7%. Similarly, we found an interesting paper[3] done by Viji et al. about a speaker-independent speech to text converter for a mobile user where they said speech recognition is the translation of spoken words into text. When everything is done, the procured discourse signal of a 1-second term (8000 examples) starts and finishes with a silence that represents almost 6000 samples. The examples were gathered from both male and female speakers. The precision for prepared speaker was 96%, untrained was 85%. Similarly, Dave et al. researched on speech recognition using recurrent neural networks[8]. To convert the speech into text, they did a sampling of the signal by using theorem Nyquist and also did pre-processing to divide into groups because these steps can result in better accuracy and time consumption. By using RNN, a sequence was maintained and it helped to predict words. To reduce long term dependencies, specifically LSTM had been used. The paper[9] of Gupta and Joshi is based on recognition of speech using an artificial neural network for both supervised and unsupervised learning. Bidirectional Recurrent Neural Network and Restricted Boltzmann Machine had been mainly used. LSTM also had been used as the solution of vanishing gradient and exploding gradient. To update the weights, they used the cross-entropy function. Data preprocessing, training, and testing data had been done to get higher accuracy. Moreover, Chen et al. worked on speech recognition by jointly learning structures[13]. There they talked about speech recognition and its success in many applications. It was discovered that the previous one was more better. In the starting investigations on the TIM IT data set, just 2.1 hours of total discourse information(in which 2500 spoken

words were annotated) gave a word blunder pace of 44.6% and this number can be decreased to 34.2% if 4.1 hours of discourse information (in which 20000 spoken words were explained) were given. Paul et al. paper[20] they used A.I. embedded speech to text. Their paper shows that deep speech is an open-source speech to text engine. Deep speech v0.6.0 works better with an embedded device but the first two versions, v0.1.0 and v0.1.1 are effective much to give the right output of words while doing speech-to-text conversion. V0.6.0 has the best features because of its less CPU usage, memory usage, processing time, loading model, interference time compared to the other two versions. But in terms of calculating accuracy rate, the v0.1.0 and v0.1.1 have 0% word error and v0.6.0 has a 7.50% word error.

Also, we found a technique in Kim and Smiths' paper[1] for acquiring highlights from discourse signals for strong investigation and acknowledgment of the non-uniform direct forecast Cepstrum. These outcomes in features that successfully model the pinnacles of the distorted vocal-tract-range, which bend viewed as perceptually important. There they found the exact calculation for discourse which includes LPC Cepstrum 53.50 percent, PLP Cepstrum 60.14%, Mei Cepstrum 68.69%, NLP Cepstrum 69.00%. In this paper[17], Raihan and Latifur talked about an effective Question Bank Similarity Searching System to search for similar questions and rate investigating results based on NLP and saving information techniques. For Bangla document processing, stemming by automata rules, lexical analysis, and elimination of stop - words have been used. Analyzing the data structure based on clustered indexing in a sorted file with a B+ tree data structure and enhanced weighted functionality TF-IDF algorithm were the most difficult QB3S procedures. The Vector Space Model was constructed from the TF-IDF weighted matrix. Quality precision of the output of the quest is satisfactory. Chen et al. pre-trained a system in their paper[19]. There they said that communication in language understanding frameworks is answerable for extricating semantic data from an information voice question. "Pre-train text NLU" has a further 2% gain and 1% gain over "pre-train content". The outcome out-plays the baseline, which is completely prepared on the whole QR set, by around 7% or around 5% with or without NLU speculations in pre-preparing. Moreover, Riad et al. paved a way for identification of stuttered speech in a paper[24]. With the standard sample, both the DNN and SVM had been discussed (Detection Recall at 0.014 and 0.086 separately). The system missed various disfluent occasions (Low Detection Recall 0.063). However, keep up with a decent accuracy comparing to the baselines.

Also, there was one survey paper[5] created by Sultana et al. where they talked about Bengali speech to text recognition. They talked about some methods-Phonemes, Direct forecast investigation, MFCC in addition to GMM. This framework obtained 96% recognition accuracy for managing novel speaker and 84.24 percent for more than one speaker. Similarly, Shrouded Markov Model, Artificial Neural Network(ANN) are also discussed. In this case, the calculation proposed by Nipa et al. has the best solution. Amin et al. demonstrated on their paper[11] about how GMM-HMM and DNN-HMM based models have been actualized in the Kaldi toolbox, for consistent Bengali discourse distributed corpus was used which is called SHRUTI. The best word error rate (WER) had been accomplished on SHRUTI was 15% utilizing CMU-SPHINX-based GMM-HMM. However, this research indicates

that utilizing Kaldi based component extraction plans with DNN-HMM and GMM-HMM acoustic models have accomplished exhibitions WER 0.92% and WER 2.02% separately. Another finding is, the WERs of the two models are exceptionally alike because the corpus's size is little. Hossain et al. proposed another sort of text age of Bangle language utilizing the bi-directional RNN[12]. Here they accomplished their work by using n-gram and Bi-directional Recurrent Neural Network. The training model right around 3 hours gives better precision of 98.766% with misfortune 0.0430. Moreover, when we attempt to relate this paper to our work there are numerous similarities, for example, they are additionally chipping away at the following conceivable word in a Bangle text and they likewise utilized NLP for their work. Tausif et al. present a paper[10] where a total discourse to message transformation framework for the Bangle language utilizing Deep Recurrent Neural Networks. Broken Language Format has been also proposed which depends on properties of the Bangle Language for decreasing the training season of the network. A straight forward profound intermittent neural organization engineering has been utilized for discourse acknowledgment. It was prepared with gathered information and which yielded over 95% exactness in the event of preparing information and half-precision if there should arise an occurrence of testing information. Saurav, Summit, and Fardia demonstrate in their paper[4] about the end to end Parts of Speech tagging and Named Entity Recognition (NER) to solve complex NLP tasks and primary objective for data acquisition. In this research, two distinct POS tagging and NER datasets were configured and numerous deep neural network initiatives were examined to overcome these two tasks separately. Bi-directional long short-term memory (BLSTM), Convolutional Neural Network (CNN), and Conditional Random Field (CRF) these deep neural network models show an end approach by obtaining an overall of 93.86% on POS and a strict f1 score of 0.6285 on NER. Fabliha, Motaleb, and Hashem describe in the paper[14] that text mining is a computational survey that deals with the views, sentiments, and behavior of humans, one of the core aspects of Natural Language Processing(NLP) research. Their method begins with actual data preprocessing and then features various N-gram techniques for extraction. Vectorization with HashingVectorizer, CountVectorizer, and TF-IDF vectorizer is then applied to the information. Aisha et al. illustrate in this paper[16] seeks to resolve this problem of puzzlement and introduces a neural language word-based model with the AWD-LSTM architecture and several similar approaches appropriate for Bangla language training. The classifier is constructed by Bangla articles having word tokens more than 28.5 million. Because of the scarcity of resources, work in this field is very limited using Bangla language, except very recent neural network proposed models and some count-based models, but all are based on words and confined in specific work resulting in high befuddlement. Rukaiya and Mushfiq show in the paper[25] that the purpose of the paper is to find out the disruptive trends of Bangla sentence extraction relationships. From FreeBase, we can get seed tuples. Freebase is a broad shared knowledge base and database of general, publicly accessible, organized information. Therefore, creating a new free base for the Bangla language and to use it to free the noisy patterns. In a new approach, Mehedi, Niaz, and Kaykobad propose a technique to parse Bangla phrases using context-free grammar rules that acknowledge all sorts of Bangla phrases, including complex, compound, exclamatory and optative phrases[21]. The suggested grammar also enables all five statement divisions to be parsed according to Bangla pronunciation. In

Bangla parsing, verb inflection plays a very important role, as well as significant attention has been paid to decomposing the verb. All syntactically correct Bangla sentences must be acknowledged by the parser, each of them parsed using the rules of the proposed grammar.

# Chapter 3

## Background Analysis

### 3.1 Recurrent Neural Network

We have decided to use RNN which stands for Recurrent Neural Network for our system. This is a specialized extended model of Deep Neural Network(DNN) which can perform speech recognition very well. Now let's see how RNN architecture works in Fig. 3.1.

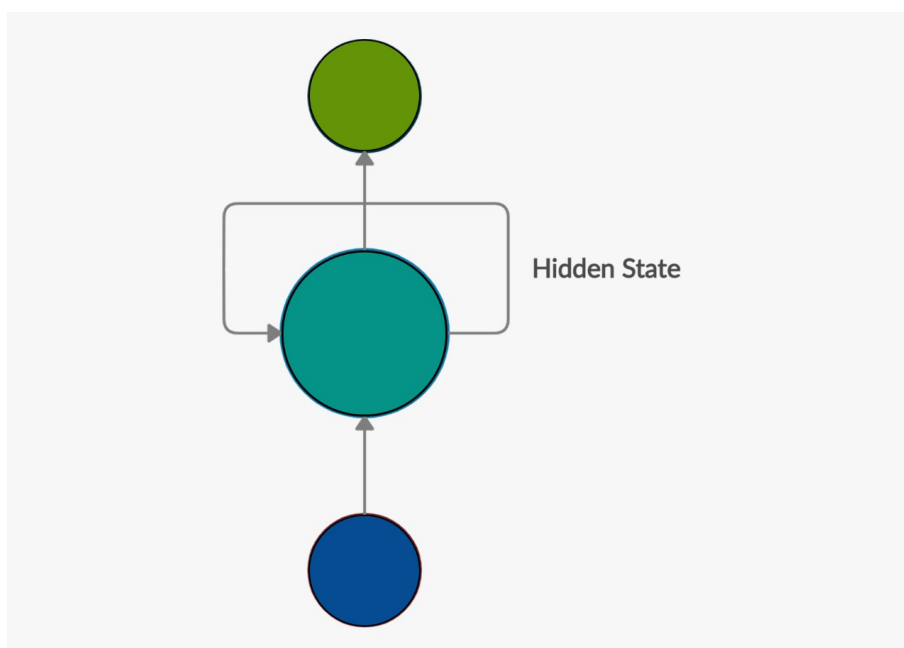


Figure 3.1: Working process of Recurrent Neural Network

For the most part, in a neural network, we take an autonomous variable  $x$  (or set of free factors) and a needy variable  $y$ , and afterward the model learns the planning  $x$  to  $y$  (training), and once the preparation is done, we give the reliant variable to decide the free factor. Now, let's talk about some of the factors about the model.

#### 3.1.1 Sequence Data

RNN is one of the foremost prevalent strategies within the later world which can successfully demonstrate the foremost important data. It may be a lesson of neu-

ral systems that are supportive in modeling grouping information. Inferred from feed-forward networks, RNNs display comparative behavior to how human brains function. To see how RNN capacities with progression data we ought to explore a demonstrate.

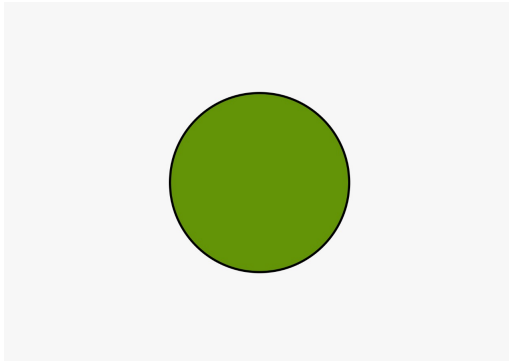


Figure 3.2: Present State

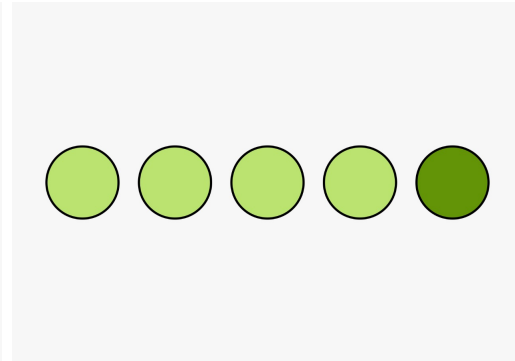


Figure 3.3: Present State with previous value

The job is to determine the direction of the ball. From the first Fig. 3.2 all that the model can do is “a guess”. Because there is no previous state or data of the ball. But still it is a random guess. For that the exactness of that figure is flawed. Presently let’s see at another Fig. 3.3 . It comprises numerous depictions of the ball’s position in progression. So the model has sufficient data to create an improved expectation. So usually a grouping, a particular arrangement in which one thing is taking after another. With this data, it is unsurprising that the ball is moving to the proper direction.

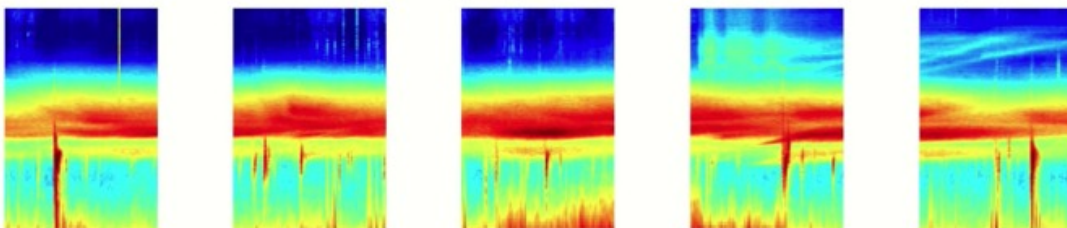


Figure 3.4: Audio spectrogram chopped into chunks[23]

Within the cutting-edge world, sequence data comes in various structures. We know that audio could be a characteristic arrangement. Here sound spectrograms can chop up into chunks to feed the RNN. RNN deliver presciently comes about in successive information that other calculations can’t do properly. The chopped audio spectrograms similar as Fig. 3.4

### 3.1.2 Sequential Memory

RNN are acceptable at preparing succession information for expectations. Sequence memory is an instrument that makes a framework simpler to perceive a grouping design.

RNN uses this concept of sequential memory. Let's talk about how the model replicates the concept. To begin with, let us see at a bidirectional neural network which is additionally known as the feed-forward neural network at Fig. 3.5. In this model there are different sort of layer such as input layer, covered up layer and yield layer.

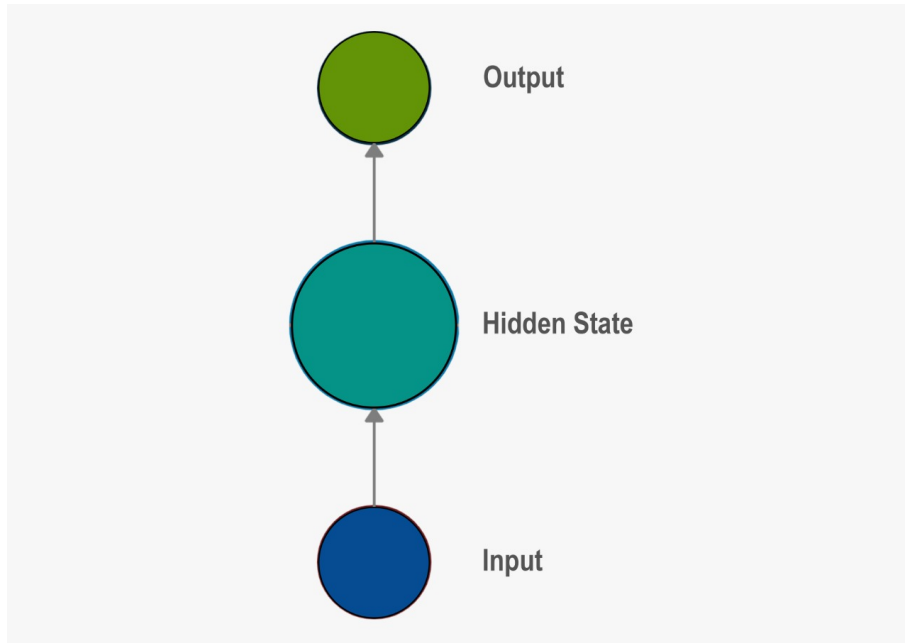


Figure 3.5: Typical Neural Network(Feed-Forward)

Directly in here, we require a circle inside the concealed state so that it can make it prepared to utilize past information to affect afterward ones which are appeared in Fig. 3.6.

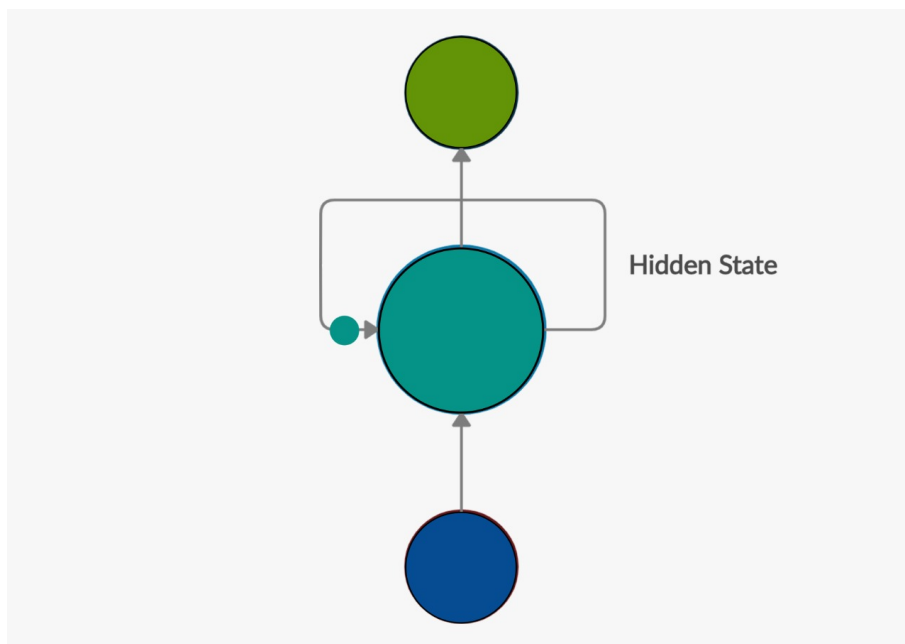


Figure 3.6: Pass Hidden State to next state



Moreover, that's fundamentally how an RNN works in this sort of situation. An RNN features a circling component that goes around as an interstate to allow information to spill out of one stage to the taking after. This information is then covered state, which may be a depiction of past sources of info. Let us look at how a chatbot can characterize objectives from the client's inputted text. At first, we ought to encode the progression of content utilizing an RNN. At that point, we'll look out of the RNN abdicate into a feed-forward neural network which can bunch the objectives. Assume a client sorts in “আমাদের দেশের নাম বাংলাদেশ।” To begin with, we partitioned each sentence into solitary words. As RNN's work progressively in this matter so we feed each word in turn which is appeared at Fig. 3.7.

আমাদের দেশের নাম বাংলাদেশ।

আমাদের দেশের নাম বাংলাদেশ।

Figure 3.7: Segmentation of a sentence

In here, the first step is to feed “আমাদের” into the network. After that the RNN encodes “আমাদের” and give an output which is appeared in Fig. 3.8.

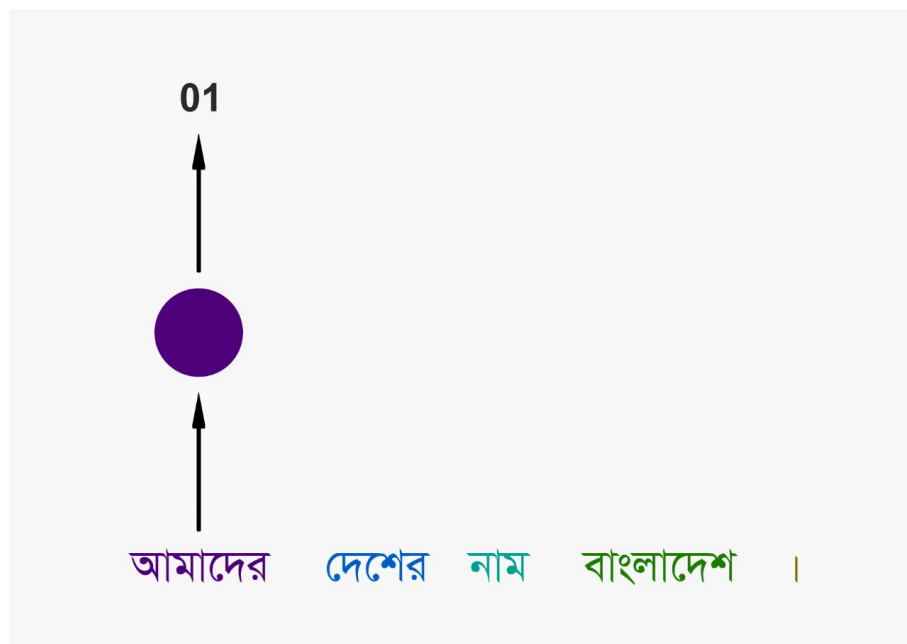


Figure 3.8: Feed the first word into RNN

After that within another step, we feed the word “দেশের” and the covered up state from the previous step. The RNN presently has data on both the word “আমাদের” and “দেশের” that we will see at Fig. 3.9 .

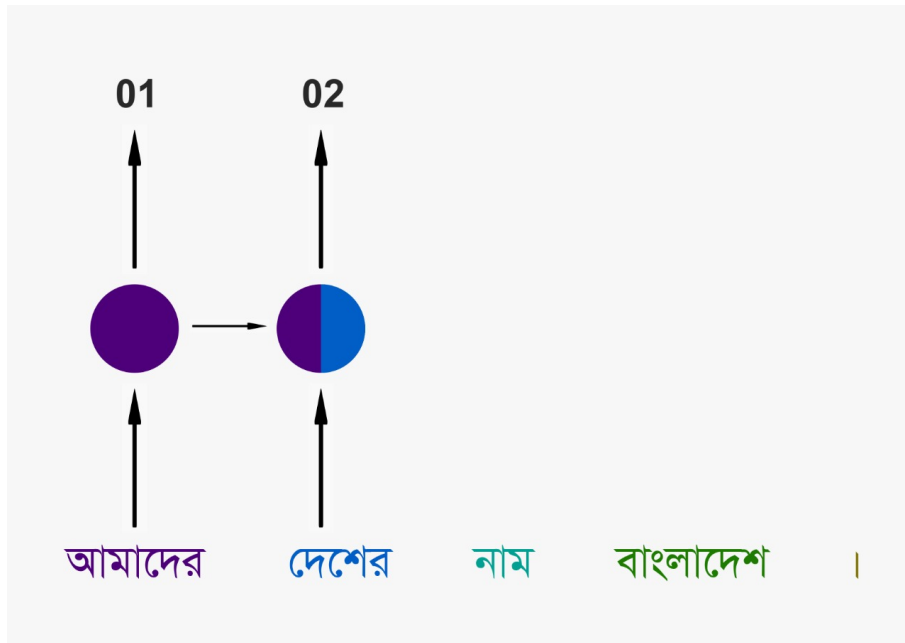


Figure 3.9: Feed the model maintaining the sequence

At that point we repeat this process, until we reach the ultimate step. You’ll be able see by the final step that, the network has encoded data from all the words in past steps which is appeared in Fig. 3.10 .

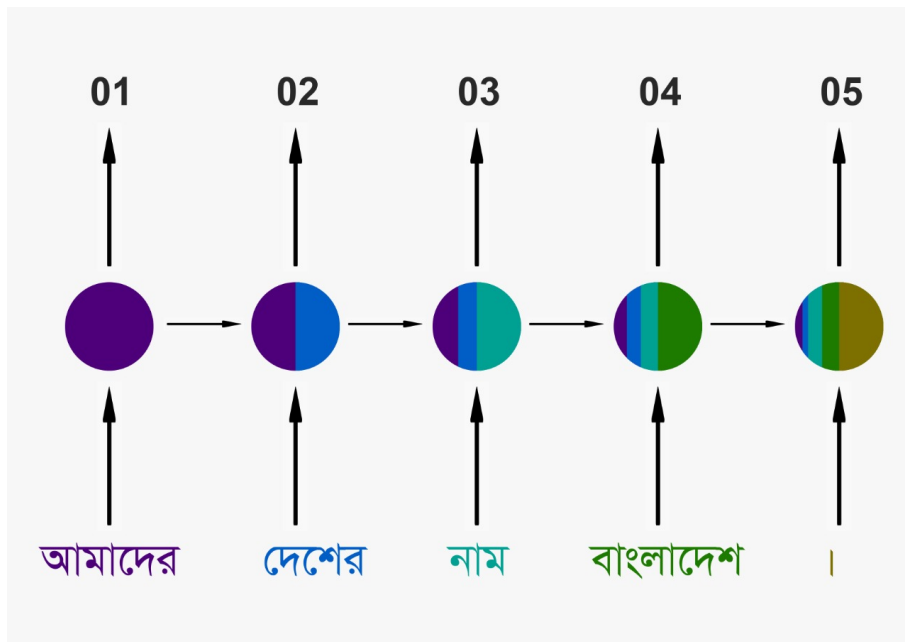


Figure 3.10: Feed the full input by maintaining the sequence

Since the ultimate yield we got was made from the rest grouping, we should be then able to require the ultimate yield and pass it to the feed-forward layer so

that able to classify an expectation. And this can be how a RNN predicts the yield which is appeared at Fig. 3.11.

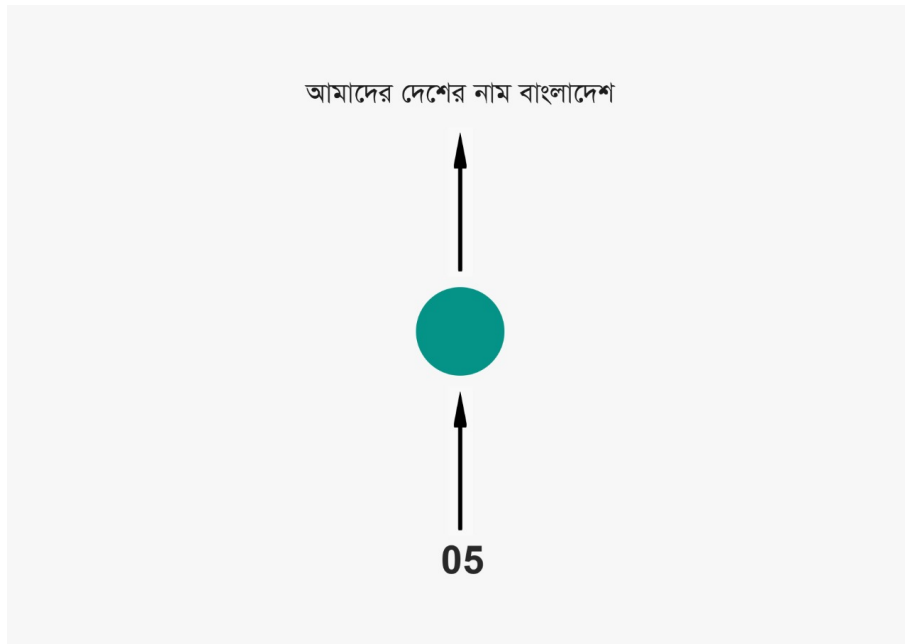


Figure 3.11: Make the final prediction

### 3.1.3 Vanishing Gradient

Within the occasion where we explore there, we will see the odd conveyance of hues within the covered states. This happens since it represents an issue with RNN known as short-term memory which is obvious at Fig. 3.12.

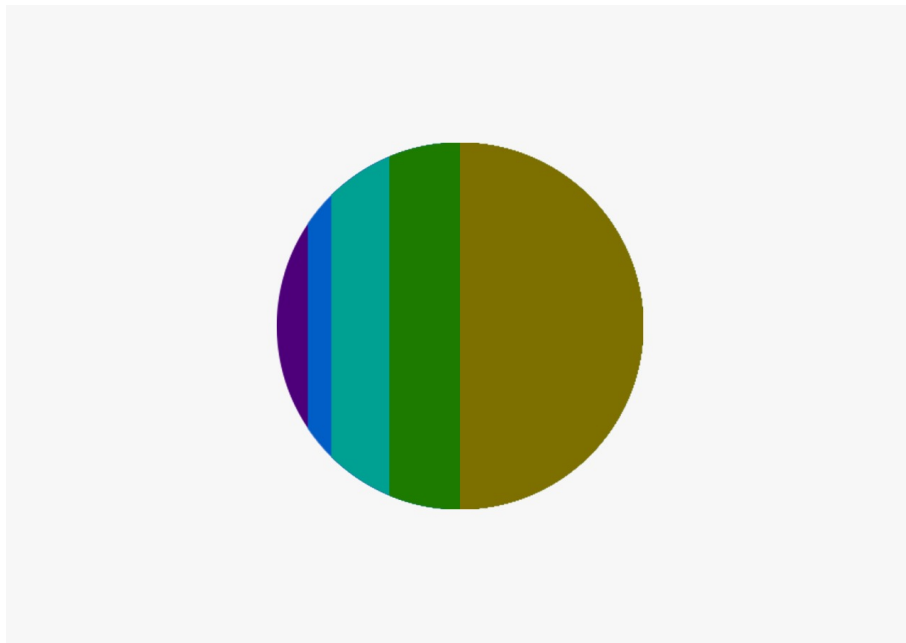


Figure 3.12: Final hidden state

Here the short-term memory is brought approximately by the scandalous disap-

peering inclination issue, which is moreover common in other neural arrange models. As we know that the RNN measures more propel, it encounters trouble holding information from past propels. As you'll be able to see, the data from the word “আমাদের” and “দেশের” is nearly non-existent at the ultimate time step. In here Short-term Memory and the vanishing gradient is happening since the thought of back-propagation; a calculation used to get ready and progress neural systems. To comprehend why usually, how around we explore the impacts of back propagation on a significant feed-forward neural arrange.

For planning a neural network it has three critical advances. To begin with, it does a forward pass and makes a forecast. Moment, it compares the forecast to the ground truth employing a misfortune work. The loss function yields blunder esteem which is an assessment of how ineffectively the network is performing is alluding at Fig. 3.13. Finally, it employs that mistake esteem to do back propagation which calculates the gradients for each hub within the arrange.

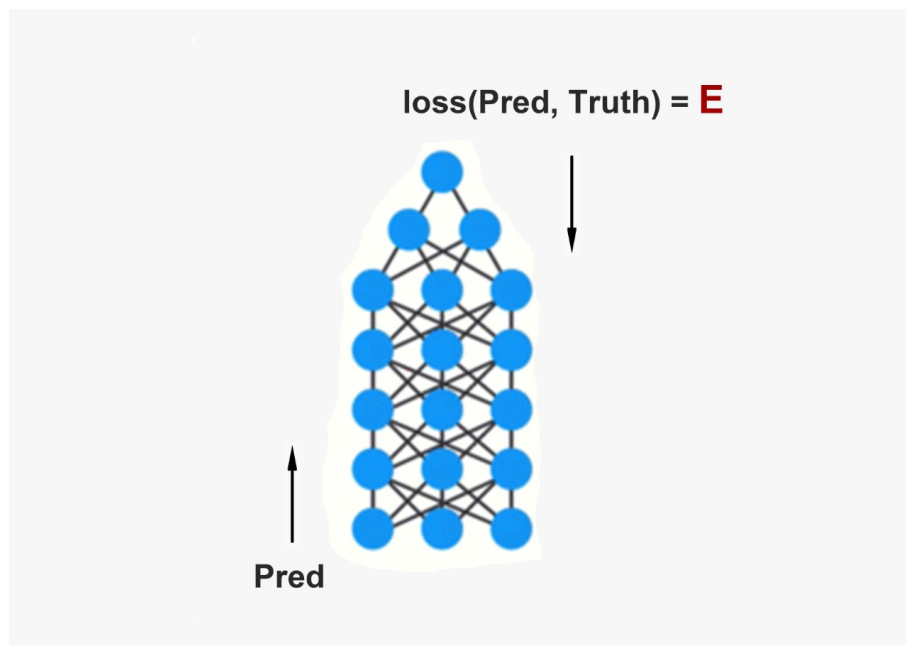


Figure 3.13: Loss Function

With the knowledge of graph unrolling and parameter sharing, we currently build up the RNN. We assume the hyperbolic tangent activation function. A natural way to regard output is by giving unnormalised log probabilities, we apply softmax as a post-processing step.

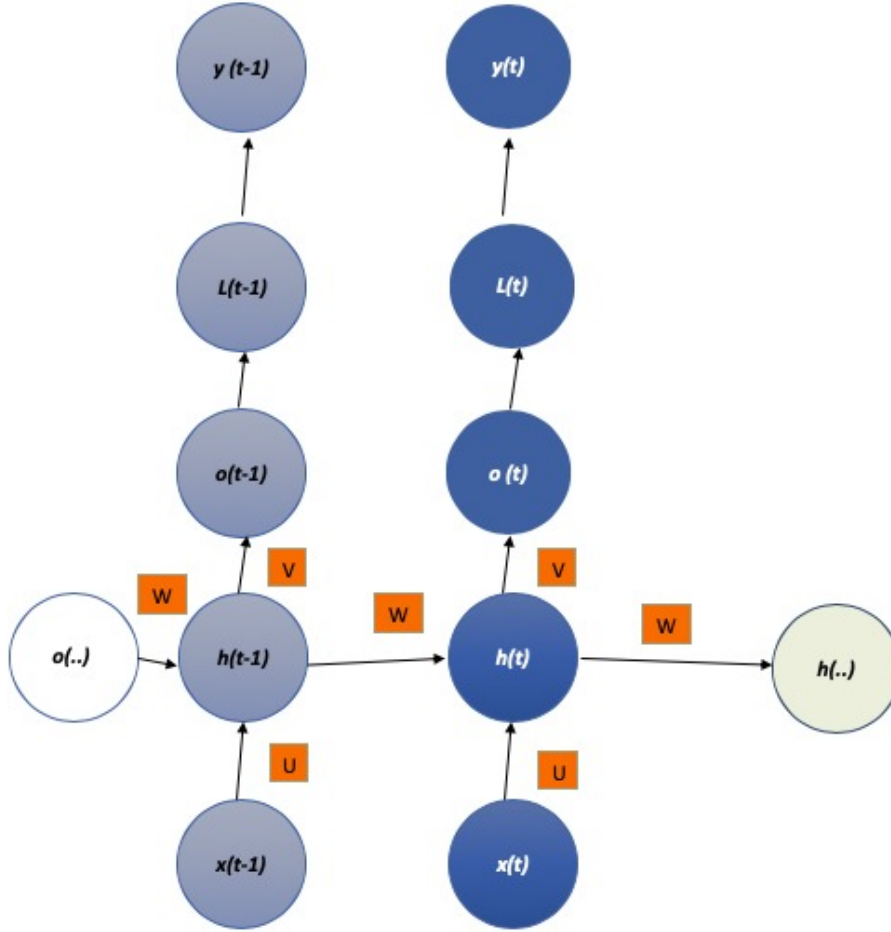


Figure 3.14: Calculating Lost Function[18]

Computation Chart is utilized to compute the preparing misfortune of recurrent neural networks. There, the arrangement of yield values  $o$  is compared to the preparing targets  $y$ , at that point this leads to the computation of the misfortune function. Here we accept  $o$  is the unnormalized log probabilities shown in Eqn. 3.2. The misfortune work  $L$  inside computes  $y \hat{=} \text{softmax}(o)$  and compares this to target  $y$ . Then the RNN has input to cover up associations parameterized by a weight framework  $U$ , parameterized by a weighted network  $W$ , and covered up to yield association parameterized by a weight framework  $V$  which is appeared in Fig. 3.14.

Here is a picture of RNN that maps input grouping to yield grouping of same length. In this the overall loss would be the sum of losses over time.  $L(t)$  may be a negative log probability of  $y(t)$ [18].

$$\mathbf{L}(\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\tau)}\}, \{\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(\tau)}\}) = \sum_t \mathbf{L}^{(t)} \quad (3.1)$$

$$= - \sum_t \log p_{\text{model}}(\mathbf{y}^{(t)} | \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\tau)}\}) \quad (3.2)$$

Presently let's examine the gradient. The gradient is the esteem that is utilized to alter the network's inner weights, permitting the organization to memorize. The

greater the slope, the greater the alterations, and the other way around. Here is the put the troublesome falsehoods. Whereas doing backpropagation, when each node in a layer computes its slope it concerns approximately the impacts of the slope within the layer sometime recently it. So within the occasion that the changes in agreement with the layers sometime recently it is small, at that point acclimations to the current layer will be significantly smaller.

That produces gradients exponentially contract because it back proliferates down. The earlier layers come up short to do any learning as the insides weights are scarcely being adjusted since of little angles. Moreover, that's the vanishing angle issue in Fig. 3.15.

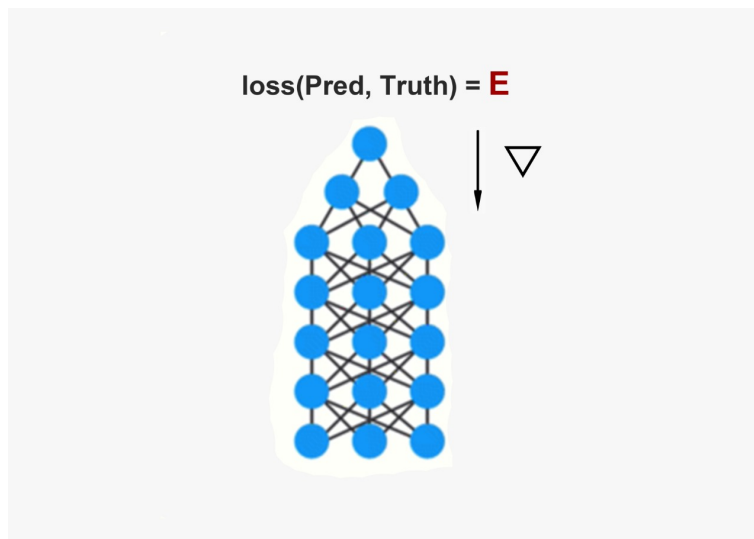


Figure 3.15: Calculating gradients through back-propagation

Now let us see how this applies to recurrent neural systems. We can think of each time a step in a repetitive neural network as a layer. So to prepare a repetitive neural network, we will utilize an application of back-propagation called, back-propagation through time. At that point, the gradient values will exponentially wither because it proliferates through each time step which is visible at Fig. 3.16.

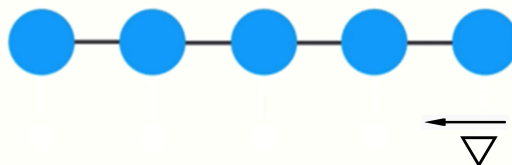


Figure 3.16: Back-propagation over time

To ascertain these gradients we utilize the chain rule of differentiation. That is

the backpropagation calculation when applied in reverse beginning from the error. We'll use as an example, just to have concrete numbers to work with[6].

$$\begin{aligned}
 \frac{\partial A_3}{\partial U} &= \frac{\partial A_3}{\partial \hat{t}_3} \frac{\partial \hat{t}_3}{\partial U} \\
 &= \frac{\partial A_3}{\partial \hat{t}_3} \frac{\partial \hat{t}_3}{\partial w_3} \frac{\partial w_3}{\partial V} \\
 &= (\hat{t}_3 - t_3) \otimes p_3
 \end{aligned} \tag{3.3}$$

Again, the gradient is utilized to form alterations within the weights of the neural system in a like manner allowing it to memorize. Little gradients mean little adjustments. That causes the early layers not to memorize. For vanishing gradients, the RNN doesn't learn the long-range conditions over time steps, which means there's a possibility that the word “আমাদের” and “দেশের” are not considered when attempting to anticipate the user's purpose. The network at that point has got to make the most excellent guess with “নাম বাংলাদেশ।”. That's truly dubious and would be troublesome on any occasion, indeed for a human. As a result not being able to learn on prior time steps causes the organization to have a short-term memory. So we have to utilize a specialized form of RNN which is the Gated Recurrent Unit(GRU) and Long Short-Term Memory(LSTM). GRU essentially works like RNN, however, they're fit for learning long-term conditions utilizing instruments called “gates”. These entryways are distinctive tensor operations that can learn which data to include or remove in covered up state. In case of the limit, short-term memory isn't a huge issue for them.

## 3.2 Gated Recurrent Unit

GRU is the exceptionally modern part of the family. GRU has liberated a few extraordinary state and utilized the covered up state to control the data[22]. It moreover just has two gates for it's utilize, one is called reset gate and other one is called update gate Fig. 3.17.

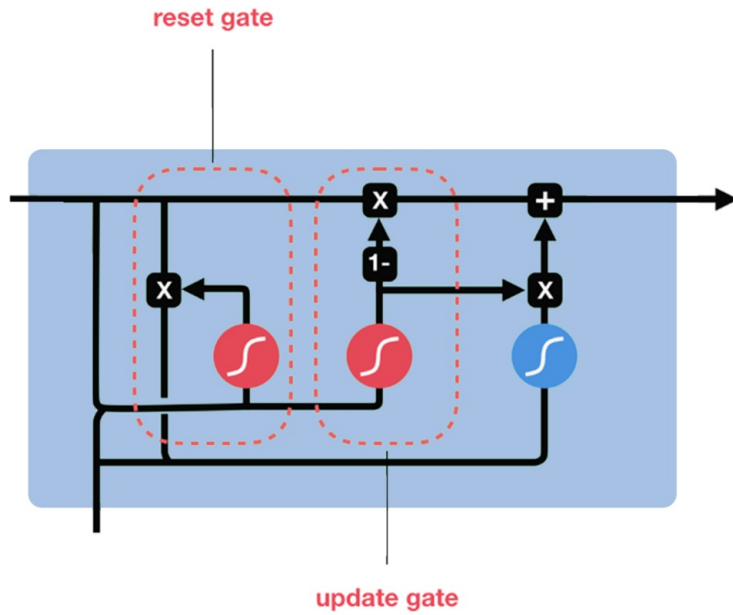


Figure 3.17: GRU cell and its gate

### 3.2.1 Update Gate

This specific gate, update gate, acts at 2 phrases. First one is Forget Gate and the second one is Input Gate. Let's talk about them.

### 3.2.2 Forget Gate

To start with, we have the forget gate which is shown at Fig. 3.18. This door chooses what information ought to be erased or kept. The previous information from the previous hidden state and current state is gone through the sigmoid activation function. The result is always between 0 and 1. The more like 0 means to forget the information and the more like 1 likes to keep the information.

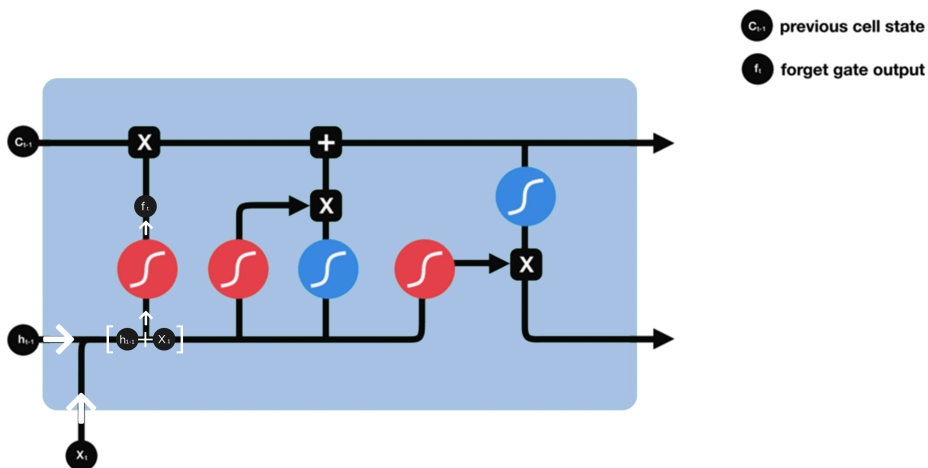


Figure 3.18: Operations of Forget Gate



### 3.2.3 Input Gate

In this step, to start the cell again we have input gate which is referring Fig. 3.19. To start with, previous hidden state is passed along with the current input into a Sigmoid activation function. Which values will be updated by transforming the values to be between 0 and 1 is chosen by that activation function. Here, not significant is represented by 0 and significant is represented by 1.

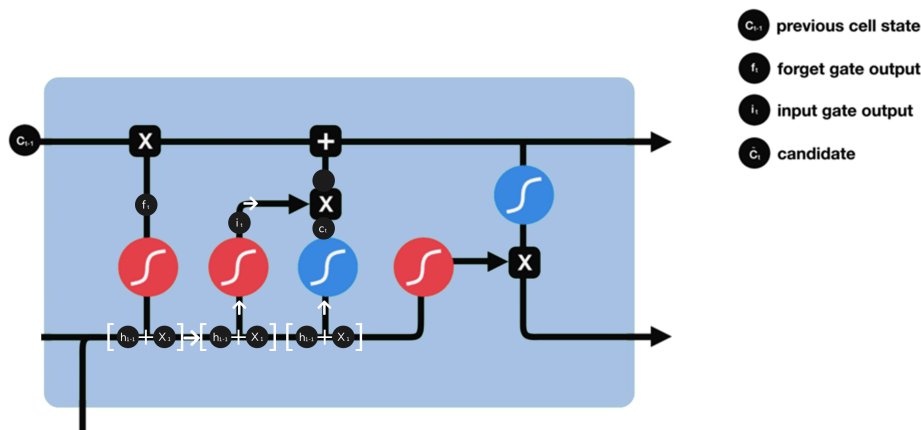


Figure 3.19: Input Gate Operations

We likewise pass the hidden state and current input into an activation function named Tanh function to keep values from -1 to 1 to help to maintain the network. Now we multiply the activation function with another activation function called Sigmoid function. Which information is necessary to keep from the output from Tanh function is decided by the Sigmoid output.

### 3.2.4 Reset Gate

This gate helps the model to learn how much past information it can forget. In a certain period of time the model starts to forget some of its old information and its determined by this gate.

## 3.3 Long Short Term Memory

The most recent version of RNN is LSTM. LSTM works really well in time series data which means it can be a good fit for our dataset since speech is one kind of time series data. Let's talk about briefly how LSTM mechanism works LSTMs are unequivocally intended to evade the drawn out reliance issue. Recollecting data for significant stretches of time is for all intents and purposes their default conduct, not something they struggle to learn! If we recall how RNN works we can easily sum up that one of the best disadvantages of that was the memory issue which is resolved in LSTM, and that's the main reason why LSTM is preferred over RNN.

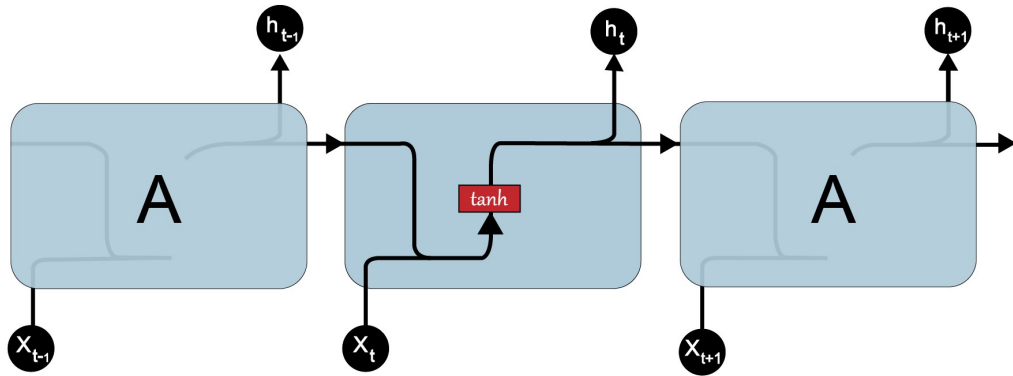


Figure 3.20: Single Layered LSTM

LSTMs have chain-like structure which is shown in Fig. 3.21, yet the rehashing module has an alternate structure. Rather than having a solitary neural network layer, there are four, collaborating in an extremely unique way.

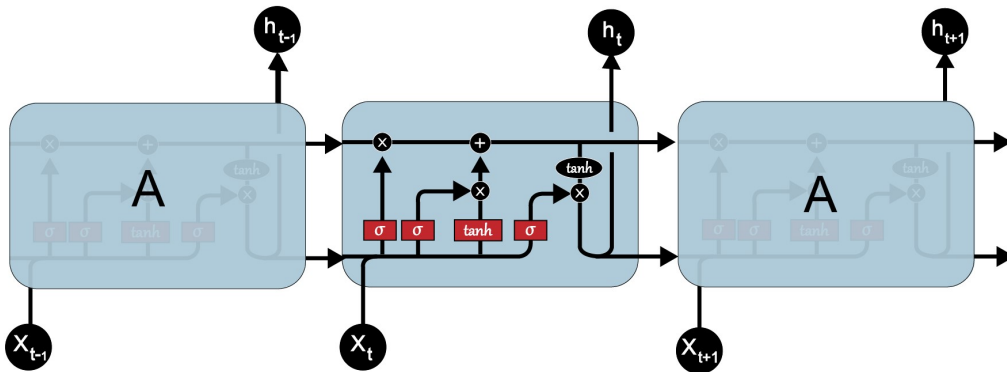


Figure 3.21: Four Interacting Layered LSTM

we are using some notations to describe the graphical architecture of the model, in below those notations are given in Fig. 3.22.

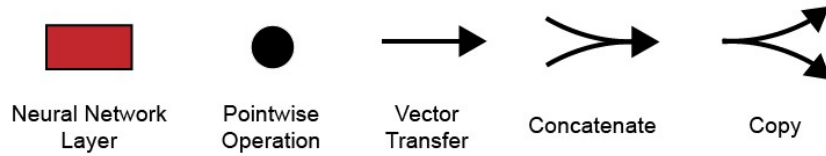


Figure 3.22: Notations used in LSTM

In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The black circles represent pointwise operations, like vector addition, while the red boxes learned neural network layers. Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

### 3.3.1 Basic Concept of LSTM

Cell state is the key of LSTM. The following figure Fig. 3.23 shows a cell state. It works straight and do some linear interactions. LSTM cannot add or remove anything from cell state.

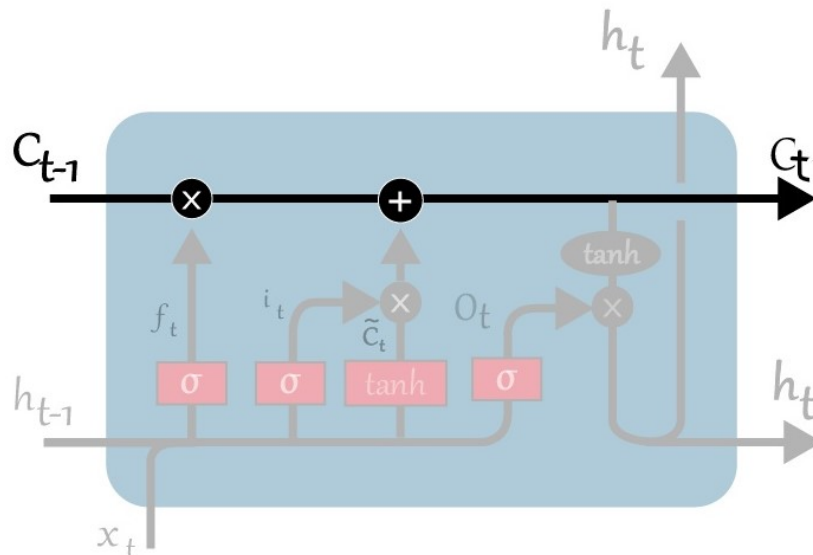


Figure 3.23: Cell State of LSTM

Some basic structures of LSTM have done by using gates just like GRU. Gates basically let information go through it. LSTM has various gates. These gates output value 0 or 1. 0 means no information can pass through it and 1 means all information can pass through it. After going through the gate, there is a pointwise multiplication operation which is given at Fig. 3.24.

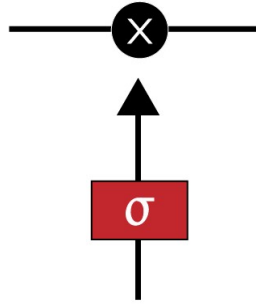


Figure 3.24: Gates of LSTM

### 3.3.2 Steps of LSTM

At the very beginning, we need to finalize what will we remove from cell state. For this, a sigmoid layer means that the forget gate layer will be generated. In the figure 3.25 we can see that from  $h_{t-1}$  and  $x_t$ , they will give either 0 or 1 in  $C_{t-1}$  cell state.

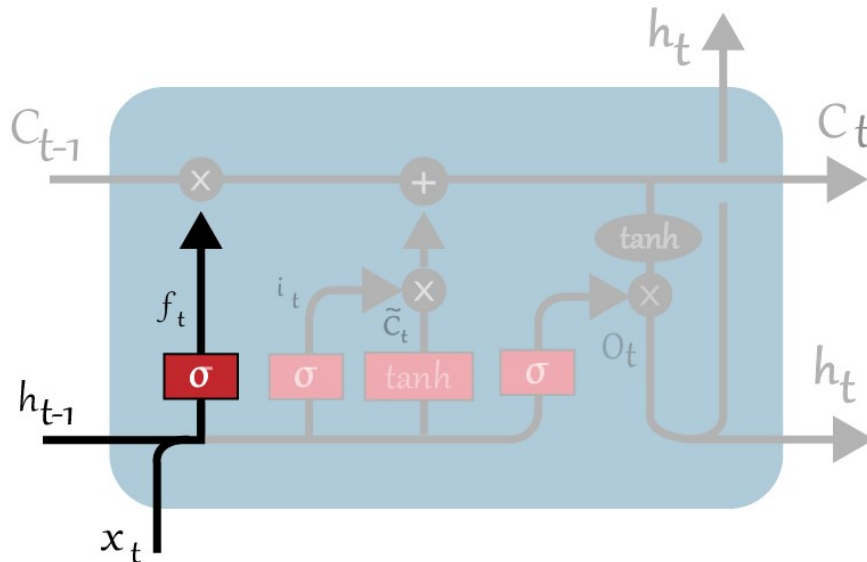


Figure 3.25: Removing information from cell state

$$f_t = \sigma(G_f \cdot [h_{t-1}, x_t] + k_f) \quad (3.4)$$

Next, we have to decide what new thing will be added in cell state. For this, a sigmoid layer will work as input gate layer and a tanh layer will work on vector for new candidate values,  $C_t$ . Then it will update the cell state which is shown in figure 3.26

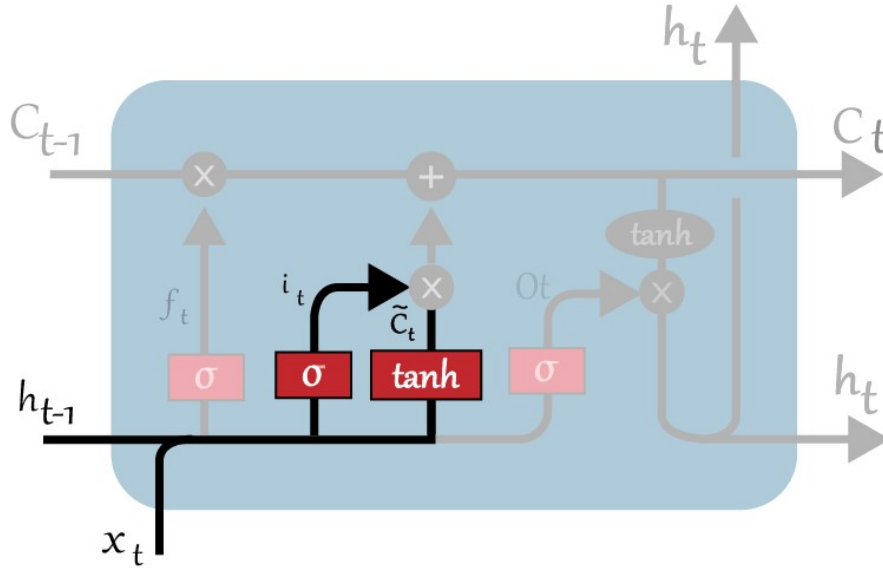


Figure 3.26: Adding information in cell state

$$\begin{aligned} i_t &= \sigma(G_i \cdot [h_{t-1}, x_t] + k_i) \\ \tilde{C}_t &= \tanh(G_C \cdot [h_{t-1}, x_t] + k_C) \end{aligned} \quad (3.5)$$

Now, we will execute a sigmoid layer and put a tanh layer. The following figure makes the operation more clear.

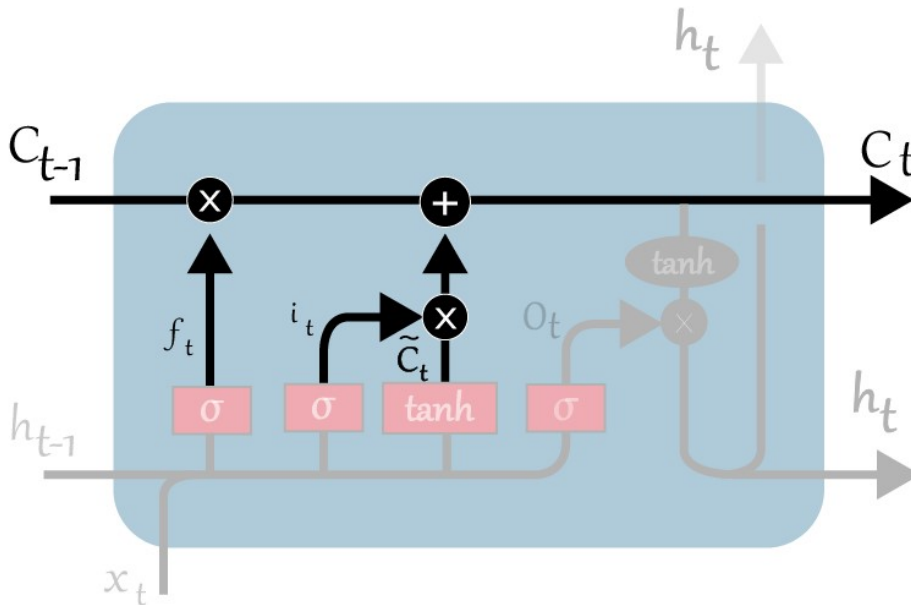


Figure 3.27: Multiplication of sigmoid and tanh layer

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (3.6)$$

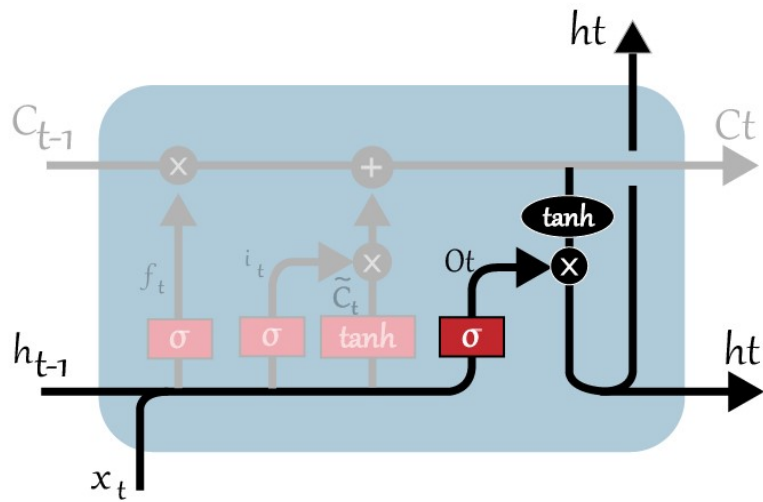


Figure 3.28: The resultant step of LSTM

$$\begin{aligned}
 o_t &= \sigma(G_o[h_{t-1}, x_t] + k_o) \\
 h_t &= o_t * \tanh(C_t)
 \end{aligned}
 \tag{3.7}$$

Finally, we will get the result by multiplying both it is shown in the above Fig 3.28

### 3.4 Confusion Matrix for evaluating the model

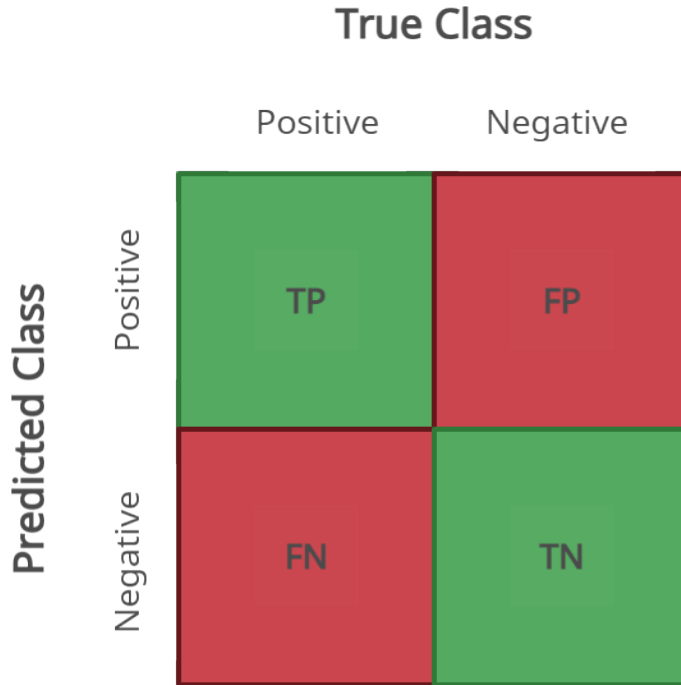


Figure 3.29: Confusion matrix

Confusion matrix is a  $N \times N$  matrix used to determine the performance of any classification method. Where  $N$  is the number of both true classes and predicted classes by the model in Fig 3.29. Confusion matrix basically helps us to visualize which actual class is predicting as which class by the model. There are a few parameter by which we get the performance result of the model.

- **Sensitivity:** Sensitivity measures the positive outcome. So, the higher sensitivity brings more accuracy to our model. The measuring equation is eqn 3.8

$$TPR = TP / (TP + FN) \tag{3.8}$$

- **Precision:** It defines how precise it is in case of calculating the positive outcome. Better model has high precision. We calculated the precision following the equation eqn 3.9

$$PPV = TP / (TP + FP) \tag{3.9}$$

- **Accuracy:** Accuracy means the positive feedback, how much validity we get. Needless to say, how important better accuracy is for any model. We measure it by eqn 3.10

$$ACC = (TP + TN) / (P + N) \tag{3.10}$$

# Chapter 4

## Methodology and Implementation

### 4.1 Data Collection and Dataset

#### 4.1.1 Data Collection Process

Our primary task was to collect a good dataset. In our research, the higher number of datasets can enrich our model. It will also increase the accuracy rate. Our thesis topic is Conversion of Bengali Speech to Text Using Long Short-Term Memory(LSTM). For this, we needed an audio type dataset. A lot of work has been done in the English language but very few have been done in the Bengali language. During audio data collection, we found some data available in the Bengali language. These available datasets gave us some information about how we should go ahead in the process but were not useful for our research purpose. That's why we have made our novel dataset to move forward with our work. Therefore, we decided that we had to collect our dataset of Bengali speech.

We live in a modern era. We are inventing new methods and ways every day to make our life much simpler and easier. Even village people are also trying to keep pace with the time. In Bangladesh, mostly villagers are poor and illiterate. For being uneducated, sometimes they face difficulty to cope up with the modern world and many wicked people try to take advantage of it. So, our motto is to make the system available for them in the long run so that their illiteracy cannot be a curse for them anymore. For now, we chose some basic words from our everyday life to generate the necessary text for further process.

We have used a passage with 100 unique words for our audio dataset. Also we wrote it on our own. We try to keep various types of words in our passage so that there will be diversity in our word collection. It is put together with 11 compound words, 20 singular words, 6 plural words. There is pretty much every kind of words like nouns, adjectives, pronouns, and many more in our passage.

We have two types of datasets. One is training and another is testing dataset. The training dataset is to train our model the voice of different people and match with our selective words. Moreover, testing dataset will help us to test our model and identify the accuracy.



## 4.1.2 Dataset Description

To collect our dataset, we took the help of social media. By using the platform, many of our friends, mutual friends, relatives have given us the voice recording of our selected passage. They recorded the voice on their mobile and send us on mail, messenger, whatsapp according to their convenience. Some also used the recorder of messenger and whatsapp to record the voice.

Till now, we have collected more than 216 voice recordings and still collecting. After collecting the voice recording, we all followed a format to save it. We all took a number range separately so that we could know who saved which recording. We divided the range among four of us into this format: 0001-1000, 1001-2000, 2001-3000 and 3001-4000. We also added the age and the gender while saving the recording so that we can easily differentiate the recording based on that. Therefore, we decided to save our recorded file as “pNumber\_Age\_Gender”. This way we will easily know about the voice record whether it belongs to male or female and also the age range from the collected data. We have 103 male voice data, 113 female voice data. The average number of male and female voice collection: male 47.69%, female 52.31% which is plotted at Fig. 4.1. Also, there was diversity in the age range during the collection of data. The highest age was 50 years and the lowest age was 11 years.

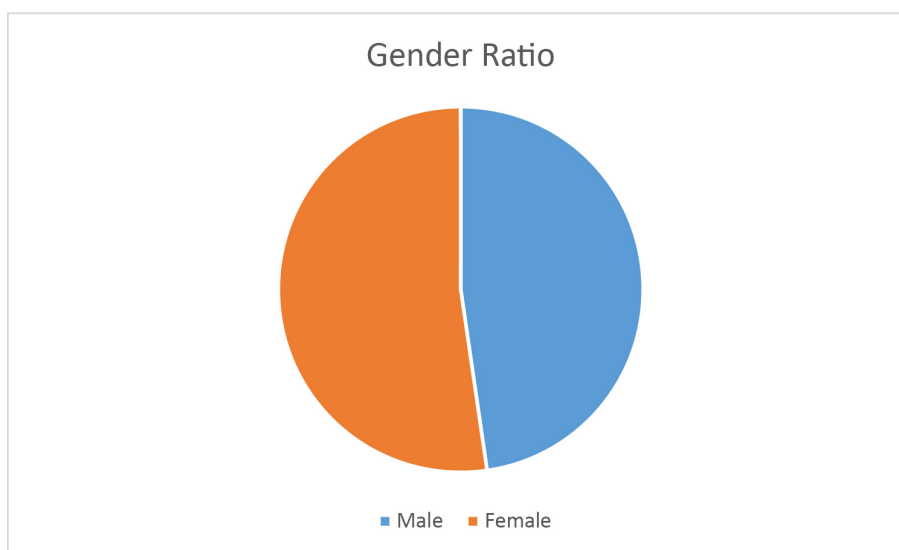


Figure 4.1: Gender ratio over dataset

However, when we tried to differentiate the age range among male and female the number of female were more than female in our data collection. The highest male age was 50 years and the lowest was 12 years and the average age for male was 23.76 years. Also among the females highest age was 49 years and the lowest was 11 years and the average age was 22.83 years. During our research, we saw that the frequency of women was lowest when the age range was 26-28 years and highest from 20-22 years. Also for males, the lowest frequency was when it was between the age of 29-31 years and highest from 20-22 years as you can see from Fig. 4.2. However if we want to compare each age range we saw that when the age range in 11-13 years male had the highest frequency, when it was 17-19 years female had highest one, also when it was 20-22 age range female had the highest frequency, but

when it was 23-25 age range male had the highest one, also when the range was 26-28 years and 29-31 years male had highest however when the age range was 31 and more female had the highest one.

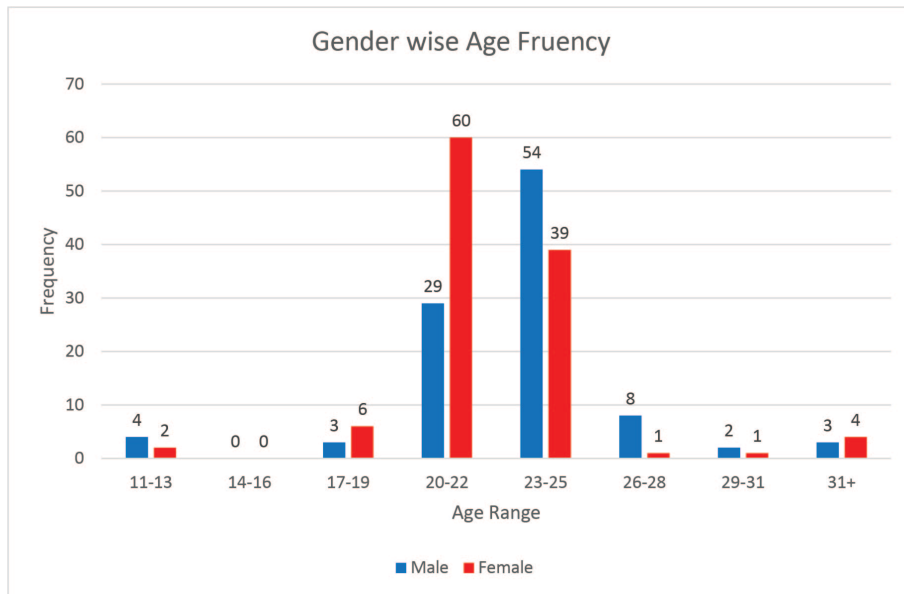


Figure 4.2: Gender wise age frequency

When we tried to see the age range there were various ages in our collection which we plotted as a bar chart at Fig. 4.3. Most of the collected recordings of people age belong to 20-25 years. We got the lowest recordings of the age 11-13 years people. However among the age range 11-13 years, 17-19 years, 26-28 years, and 31 and more we got a little more than age range 11-13 years.

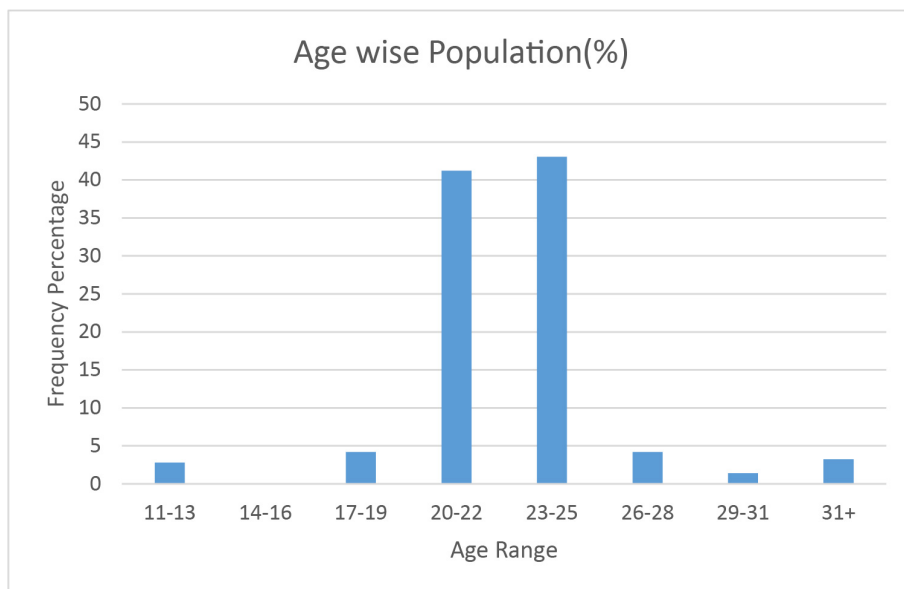


Figure 4.3: Age wise population percentage

In this process when we try to divide the collection according to the time they took 15 people took 39-45 seconds, 42 people took 46-50 seconds, 51 people took

51-56 seconds, 68 people took 56-60 seconds, 22 people took 61-65 seconds, 10 people took 66-70 seconds, 8 people took the highest time which was 71-76 seconds. The longest voice record was 76 seconds up until now as you can see in the pie chart at Fig. 4.4.

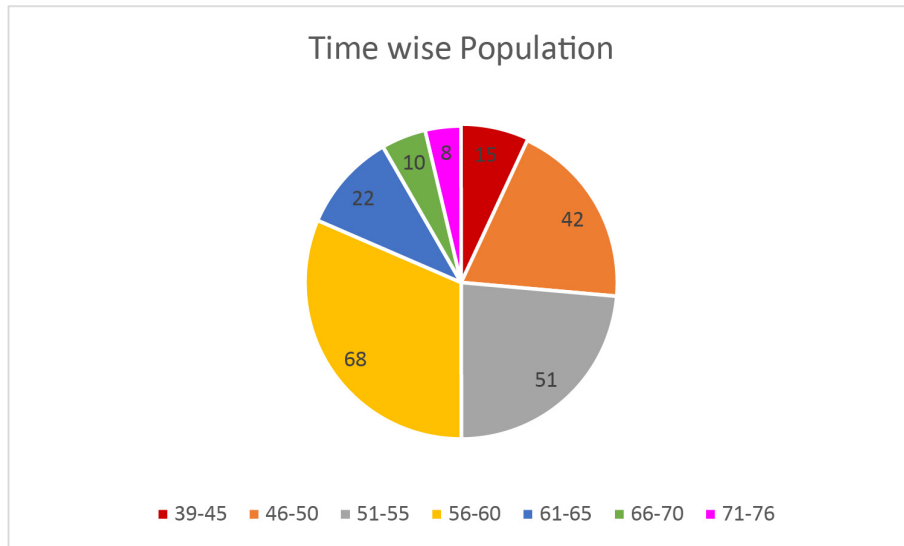


Figure 4.4: Time wise Population

### 4.1.3 Exploratory Data Analysis(EDA)

At the most essential level, sound is spoken to by a surge of tests, each determining the huge amount of the sound waveform as estimated for a given cut of the general waveform of the sound sign. There are a few organizations utilized for the individual examples inside a sound document. Most sound documents utilize 16-cycles marked numbers for each example, yet others utilize 32-digits coasting point esteems or 24-bits or 32-bits whole numbers. Furthermore, tests may utilize marked or unsigned qualities, too. The size of an individual example is known as the example size.

The situation of every sound source inside the sound sign is known as a channel. Each channel contains an example demonstrating the abundance of the sound being delivered by that source at a given second in time. For example, in sound system sound, there are two sound sources: one speaker on the left, and one on the right. Each of these is spoken to by one channel, and the quantity of diverts contained in the sound sign is known as the channel check. During our data collection process we collected our data using stereo channel which contain frequency like Fig. 4.5. Playback frameworks that utilize two speakers are alluded to as sound systems. Sound system sound records, for example, sound system MP3 and WAV documents, contain left channel and right channel data that tell the left and right speaker when to push and pull air.

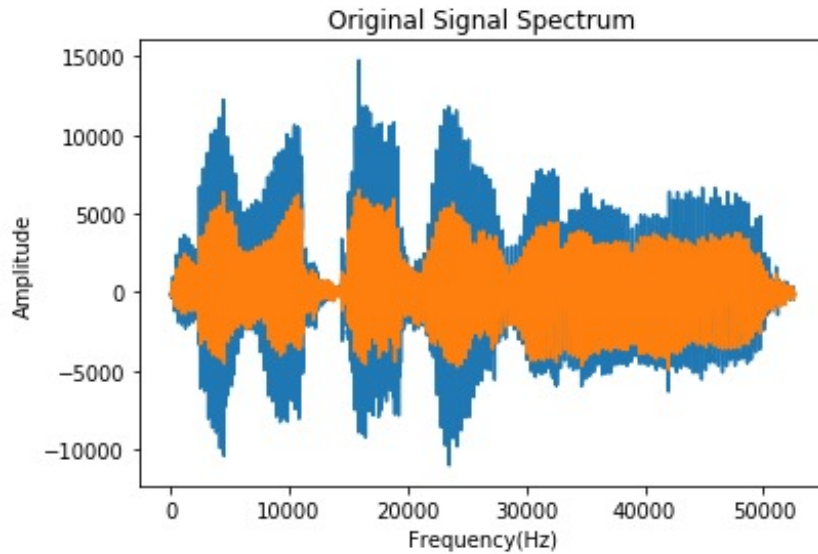


Figure 4.5: Waveform of a stereo channel

Sound systems are fit for making the impression of sound source confinement. Sound source confinement alludes to the human capacity to find the situation of a sound source inside a space. For model, on the off chance that you hear a canine yapping, it's moderately simple to decide the course the sound is originating from, and the distance away the sound source(the canine) is. The vast majority ought to have the option to limit sounds with conventional exactness, even with their eyes closed. It bodes well to expect that you'd see the sound delivered by a sound system to originate from two particular sound sources; the left speaker and right speaker. In certain circumstances, you will see sound originating from two distinct bearings, yet this isn't generally the situation. Profundity is influenced by a sound's relative level, powerful range, and resonation sum. Sounds with a powerless level and less unique range will in general show up rearward of the sound system field, as do sounds containing extreme reverb.

For this reason we have to convert our dataset recordings to mono channel which contain frequency like Fig. 4.6. Mono audio files only contain a single audio channel. In mono, otherwise called monaural sounds, there is just one hotspot for the sound. While mono sounds can likewise be shipped to various speakers, the sign that is played will in any case be the same. Here, the sound originates from just one sound and even though it is moved to two distinct speakers, the substance of the sign is consistently the equivalent.

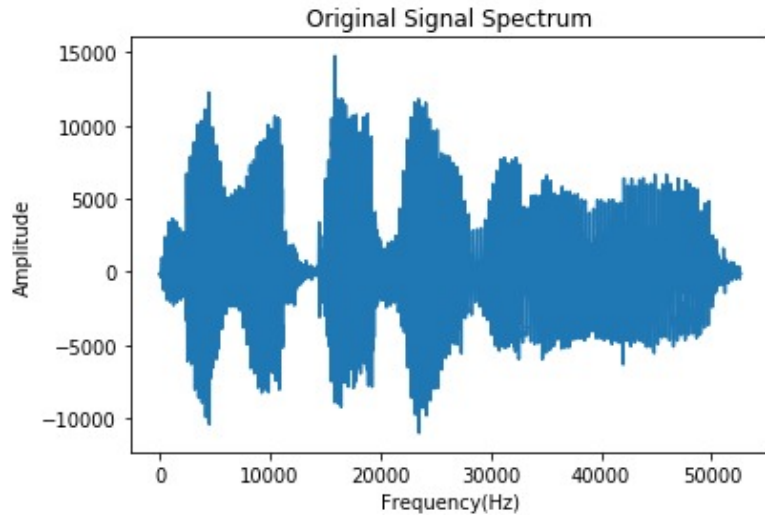


Figure 4.6: Waveform of a mono channel

At first, we gathered 216 number of audio files of unique subjects as our dataset, having 100 unique words each. But for time shortage, we extract the dataset into a smaller form in which we include 160 individual subjects and 56 unique words. Around 47% are male and 53% are female. So, female has the majority which is shown at Fig 4.7.

In our novel dataset, we have taken the voice of various subjects. Within the dataset, variety of age, gender, age- all are present. Between male and female ratio, female members have the majority as you can see in the following figure.

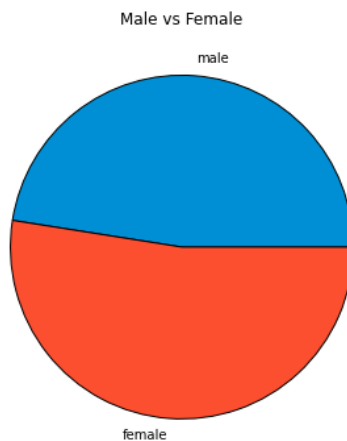


Figure 4.7: Male-Female ratio of dataset

Besides gender, we have also looked at the age. Subjects are of different ages starting from teenager to aged people. The highest age range is between 20-24 which covers 6720 data points. 784 data points are from age 45-50. 579 data points are 25-29. 158 data points are from 30-34 and 107 data points 40-44. Lastly, very few people which is 54 data points are from 35-39 in Fig 4.8.

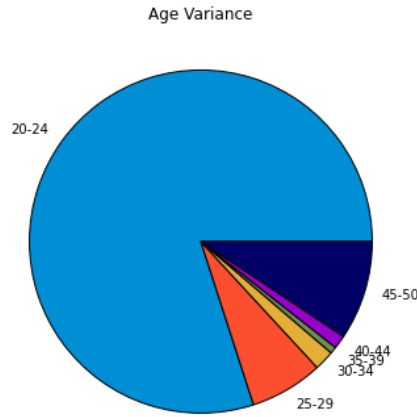


Figure 4.8: Age ratio of dataset

By data points, we are trying to mention those unique 56 words. We have trimmed 56 words from each 160 files. But all those words were not possible to use. Some were not clear, whereas some have some background sound, and also some didn't pronounce some words correctly. For these issues, the number of data points varies. The highest number of unique data points is nearly 150. On the contrary, the lowest number of some unique data points is 130. No doubt, these data points are the core element of our research which we plot at Fig 4.9.

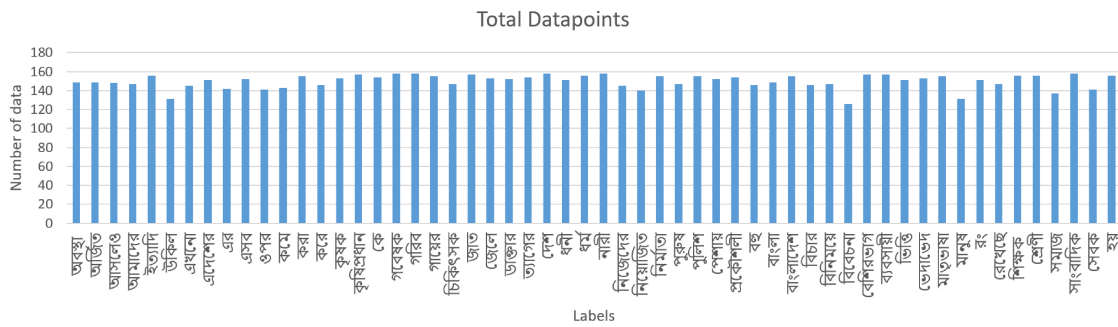


Figure 4.9: Data points of each label

Gender variance is visible clearly in each data point. When we collected those audio files, we preferred collecting audios from both male and female. After collecting those, we saw the majority of females. As mentioned earlier, 53% are female and the rest are male. Within 8402 total number of data points, 3992 are male and 4410 are female. The red marks are the portion of females and the blue marks are male in the figure Fig 4.10 where gender variance is easily seen.

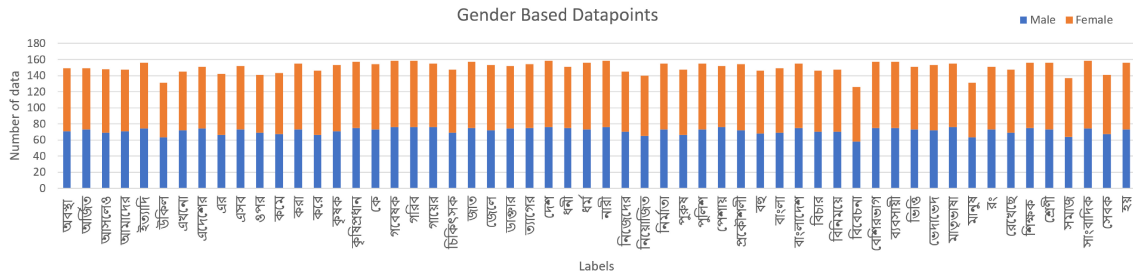


Figure 4.10: Gender variance on each label

## 4.2 Pre-processing

After completing the novel dataset we followed the following pre-processing steps before any further steps.

### 4.2.1 Dataset Splitting

So far we considered all the data points as a whole dataset. It's time to split the dataset into training and testing data so that we can get the accuracy, value loss, and everything. We have split the whole dataset into 80% training data and 20% testing data which is the standard form of any kind of machine learning research.

### 4.2.2 Standard Scaling

At this point, we take the initiative to erase mean and scale unit variance for standardizing because it's a common requirement. We followed the equation  $z=(x-u)/s$ , where  $u$  is the mean and  $s$  is the standard deviation of training samples.

### 4.2.3 Scaling the feature range

We already explained the standard scaling we have done. For each unique subject, data points are ranged between -1 to +1 on the basis of frequency. A unique subject will have all the data points according to its frequency. The lowest frequency will be scaled as -1 and the highest will be scaled as +1. The rest will be sorted according to its frequency in ascending order. The reason behind it is we wanted to put the frequency in a standard range instead of some random value. So that, it's lowest data is -1 and highest is +1.

## 4.3 Implementation

We used Tensorflow as a deep learning framework and Keras as a backend. Keras contains all the APIs for deep learning. Without further talking let's proceed to the actual work. We use the Input function with the parameter of our feature size which is 8000. it is a 1-dimensional data since audio lies under time-series data. After that, we normalize our data to standardized the input of each mini layer. It also helps to reduce the number of epochs to train our system. Then we create 3 layers of a 1-dimensional convolution process. In each step, we use Convolution for

1-dimensional data so that it can segment all the features into some portion, and after that those portions are sent to the max-pooling function to figure out the most significant features among them. The last process of this step is the dropout function which saves our model from overfitting. After these 3 steps batch normalization was again used to again standardized the input. Then comes the part of LSTM where the model feeds itself not only the input data but the previous output so that it can memorize the pattern in its memory. Once again the normalization process is again used to normalize those outputs of LSTM to standardize them. Here comes the part of the dense layer whose main aim is to feed the output of the previous layer to its all neurons. Dense is consider as a most basic layer of a neural network since the neural network good at feeding the data to neurons. An activation function is also added known as rectifier or in short ReLU. An activation function must help the network to learn a complex pattern. And the ReLU is a linear function that will output directly if the curve is in a positive direction. At last, we again use the Dense function as our output layer with the parameter of all the labels since it's a classification model. Another activation function being used this time, it is softmax. This activation function helps us to predict the multinomial probability distribution and it will output only one value for each node .Table 4.1 basic hyper parameters of the model.

Layer	Output Shape	Activation Function
InputLayer	[(None, 8000, 1)]	-
BatchNormalization <sub>1</sub>	(None, 8000, 1)	-
Conv1D <sub>1</sub>	(None, 7988, 8)	ReLu
MaxPooling1D	(None, 2662, 8)	-
Dropout	(None, 2662, 8)	-
Conv1D <sub>2</sub>	(None, 2652, 16)	ReLU
MaxPooling1D	(None, 884, 16)	-
Dropout	(None, 884, 16)	-
Conv1D <sub>3</sub>	(None, 876, 32)	ReLU
MaxPooling1D	(None, 292, 32)	-
Dropout	(None, 292, 32)	-
BatchNormalization <sub>2</sub>	(None, 292, 32)	-
Bidirectional <sub>1</sub> ( <i>GRU/LSTM</i> )	(None, 292,128)	sigmoid, tanh
Bidirectional <sub>2</sub> ( <i>GRU/LSTM</i> )	(None, 292,128)	sigmoid, tanh
Bidirectional <sub>3</sub> ( <i>GRU/LSTM</i> )	(None, 128)	sigmoid, tanh
BatchNormalization <sub>3</sub>	(None, 128)	-
Dense <sub>1</sub>	(None, 256)	ReLU
Dense <sub>2</sub> ( <i>Output</i> )	(None, 56)	softmax

Table 4.1: Hyperparameters

Our batch size was 32 and epoch was 100 with early stopping. Epoch helped us to divide the whole dataset into small chunks. Batch size determines the number of training samples in each forward or backward pass. Since memory requires increases proportional to batch size we kept the batch size a bit lower. The early stop callback function helped us to stop the training at a point when the validation data starts to degrade. Adam was used as optimizer. Optimizer helps us to change the attributes



like weights and learning rate to reduce the loss. Adam works really good with GRU and LSTM, that's why we choose it. Output shape and layer along with its activation function is given in Tab. 4.1. Finally we get the whole architecture when we integrate Tensorboard with the notebook. As notebook and for getting GPU support we used Google Colab. Since we were using DNN, GPU support was mandatory for us. In Fig. 4.11 the whole architecture of the system is given which we generated from Tensorboard.

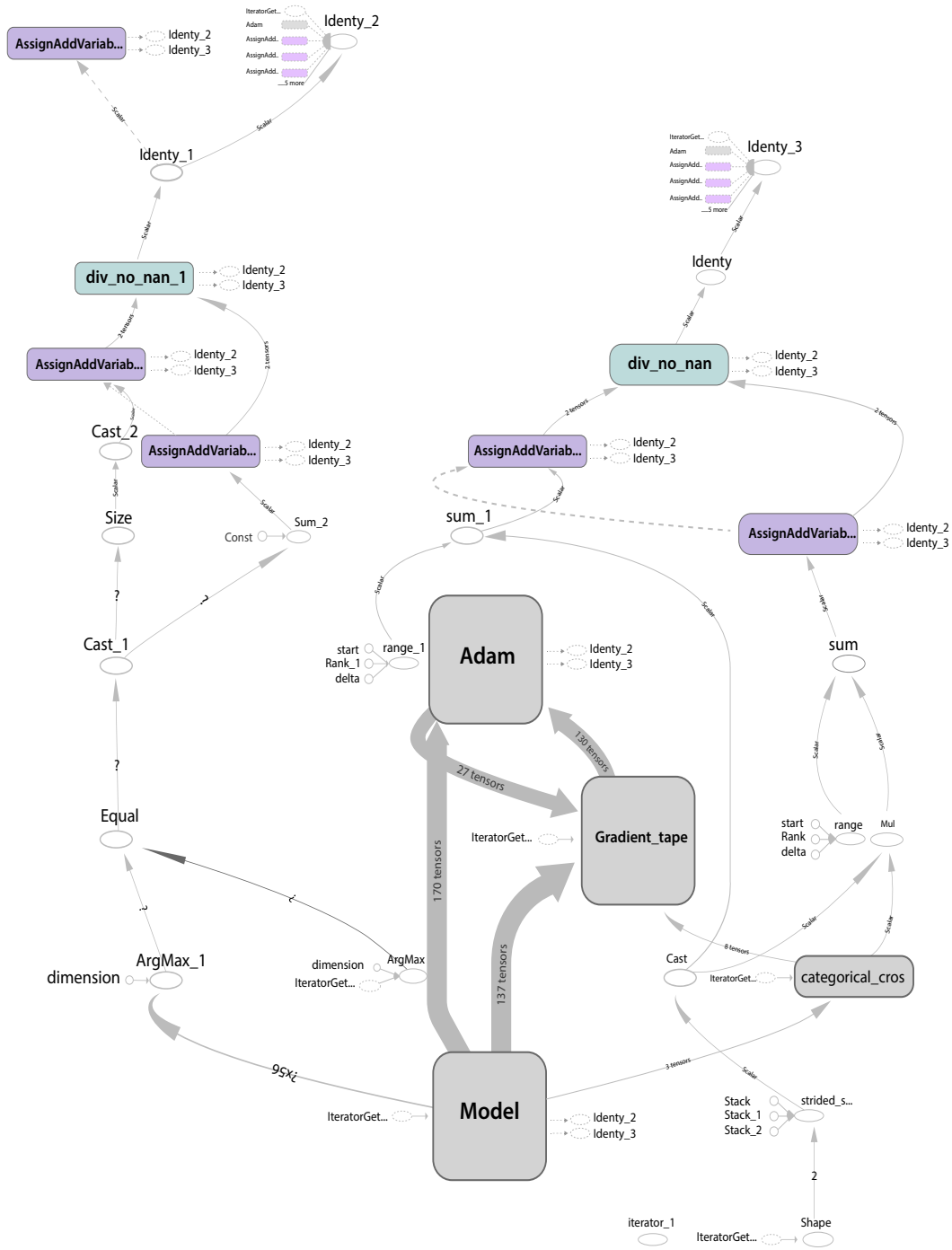


Figure 4.11: Model Architecture

# Chapter 5

## Result Analysis

### 5.1 Comparison Between GRU and LSTM

Since we did some research on both GRU and LSTM, we want to make a comparison of these two methods practically. So, we implement both models to determine which one is giving more accurate results.

#### 5.1.1 Loss Comparison

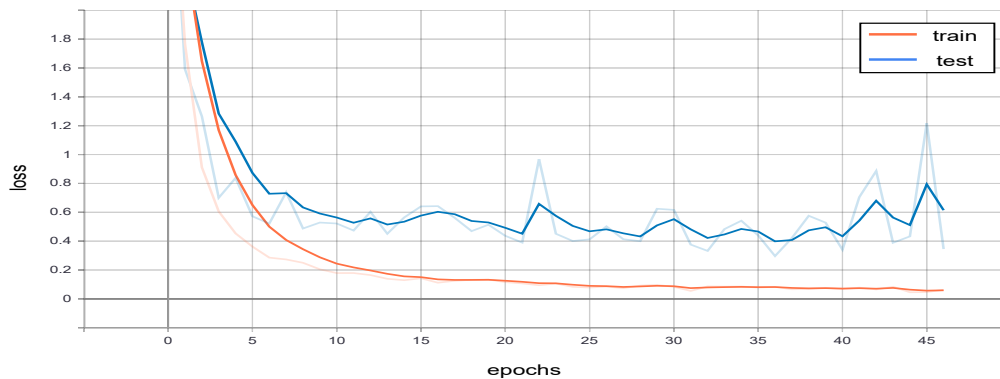


Figure 5.1: Loss Function of GRU

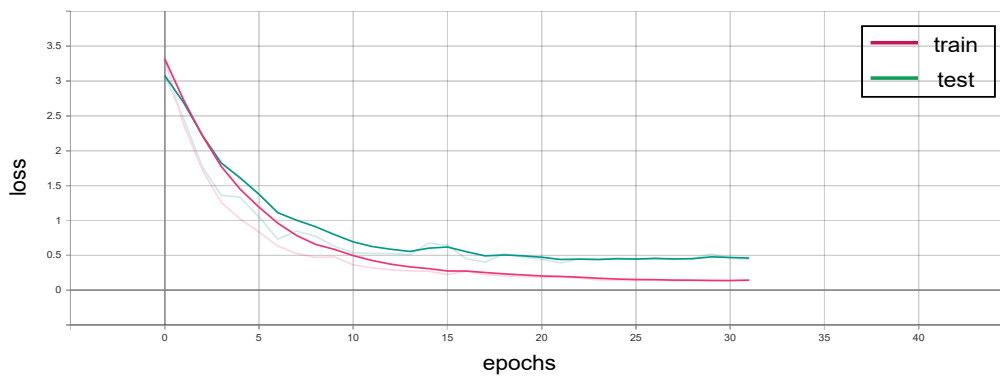


Figure 5.2: Loss Function of LSTM

In Neural Network, we seek to minimize the error which generally refers to loss function also sometimes known as cost function. At the beginning of any model, the

loss function used to be high, and over time, the loss will decrease which means the model is losing its errors. For the GRU model, in Fig. 5.1, it is clearly visible that the loss is decreasing but it is not stable. With a closer look, it is clearly verified that the loss was pretty decent till 7 epochs but it has started fluctuating when the epochs have started increasing. Though for the training data, the model maintained to minimize the loss but the problem arises when it comes to the testing data. But for LSTM, in Fig. 5.2, we found a very decent loss curve which really impressed us. The loss for both training and testing data were pretty much the same at the early epoch though it gains some extra error in testing data which is because it gets more training data than testing data. So in order to Loss the comparison, we can say that LSTM did a good job than GRU.

### 5.1.2 Accuracy Comparison

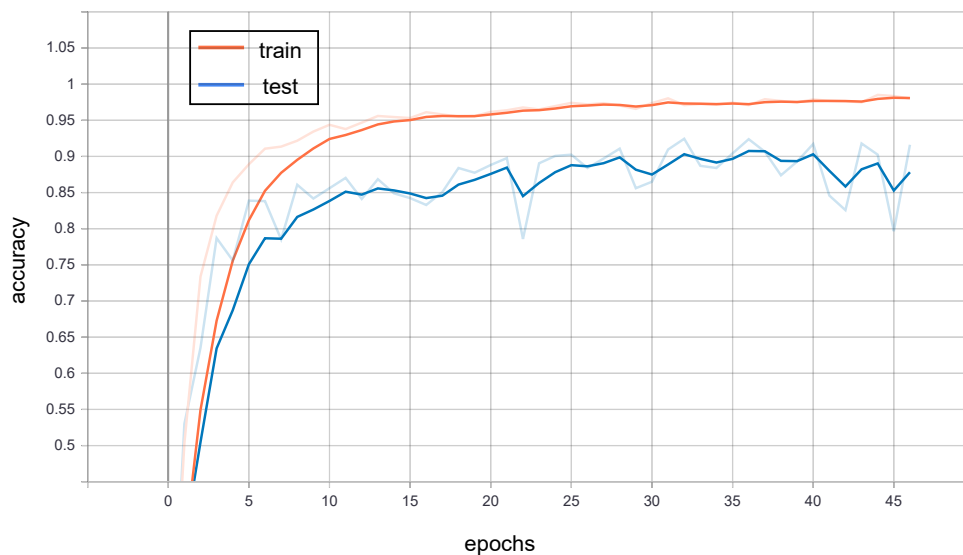


Figure 5.3: Accuracy of GRU

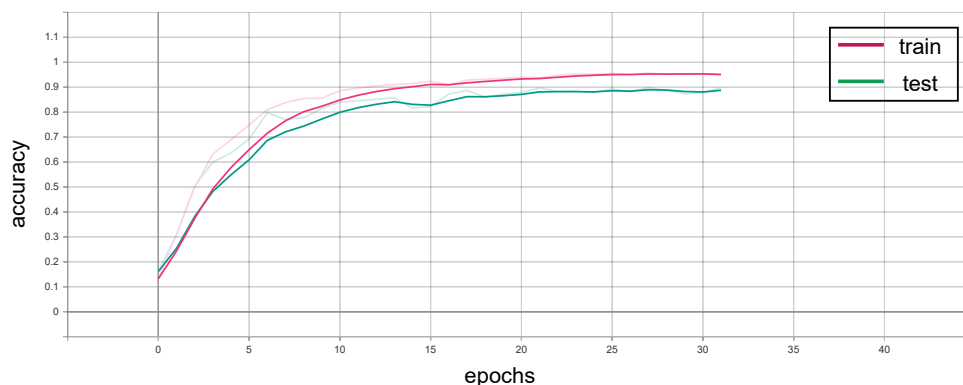


Figure 5.4: Accuracy of LSTM

Now it's time to compare the accuracy for both GRU and LSTM models. In Fig. 5.3, it is defining the accuracy for the GRU model which does not bring much satisfaction. Just like loss, it has started to do the same for training and testing but all of a sudden

it has started losing its accuracy over epochs, and that is reflected on the curve. Though it nearly 97% in training data but the accuracy is very poor on validated data which is only around 87%. Besides, in Fig. 5.4 LSTM did surprisingly well by maintaining both training and testing accuracy at the same pace. Though it is not the same for training and testing but it has a good explanation that is testing has very little data than training, that's why the model couldn't get as much accuracy as training data. But the best part is, there was no fluctuation in over epoch of traversal and the model ends up gaining around 97% accuracy for training data and 89% accuracy over testing data. So it is very clear that LSTM is performing better for our dataset and we proceed further based on the LSTM model.

## 5.2 LSTM Analysis and Hypothesis

### 5.2.1 Loss Function Analysis

In this stage, we did some experiments with the model to clarify how well a model can perform based on the amount of data it is feeding.

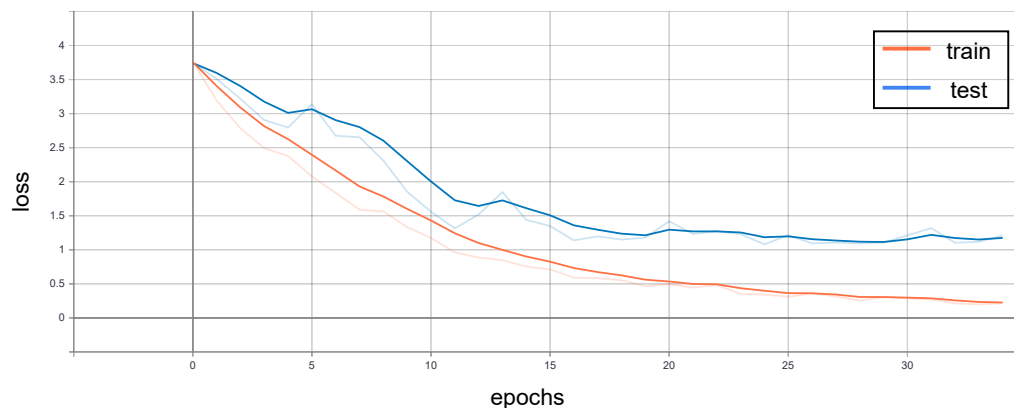


Figure 5.5: Loss Function on one-third dataset

At first, we separate the dataset into one-third to feed the model. In Fig. 5.5, it is a loss of the model based on that one-third dataset. No wonder the model will perform very poorly because, for the time being, the training data is only about 2240, and the training data was 560. And the reason for the high loss is the model needs more data to minimize its errors.

Now it's time for feeding more data into the network to see the impact of data. So this time we feed the network two-third of a dataset which makes the training data 4480 and the testing data 1120 in total for every label. Surprisingly, we can see a lower curve this time at Fig. 5.6. We can declare the model is improved by seeing its loss curve. The training loss has reduced to nearly 10% from 20% and training loss reduced to 60% from 120% which is a great success at this stage.

Now it is the time for a moment of truth where we will find the model is holding up or not. And yes! The training loss is more less which is only 10% and the testing loss is less than 40% which is at Fig. 5.7. This time we feed the network our whole dataset and it doesn't disappoint us. Maybe more data can reduce the loss even more which means the more data we can feed a network, the more it will be able to minimize the errors.

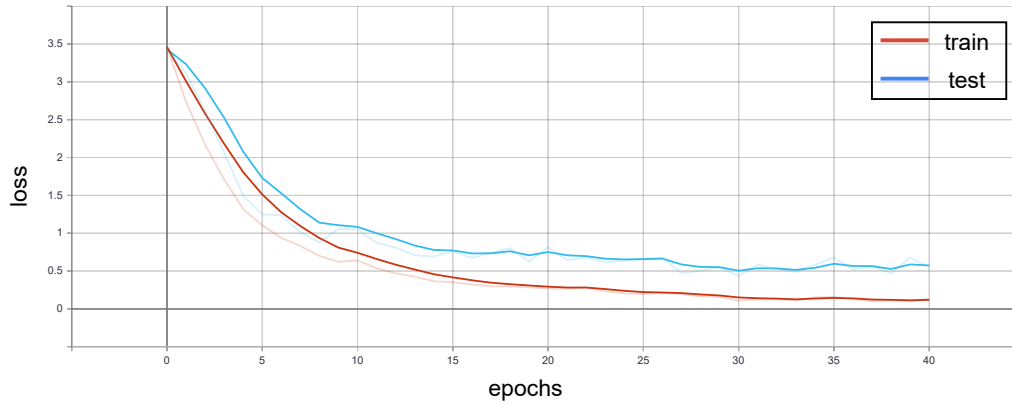


Figure 5.6: Loss Function on two-third dataset

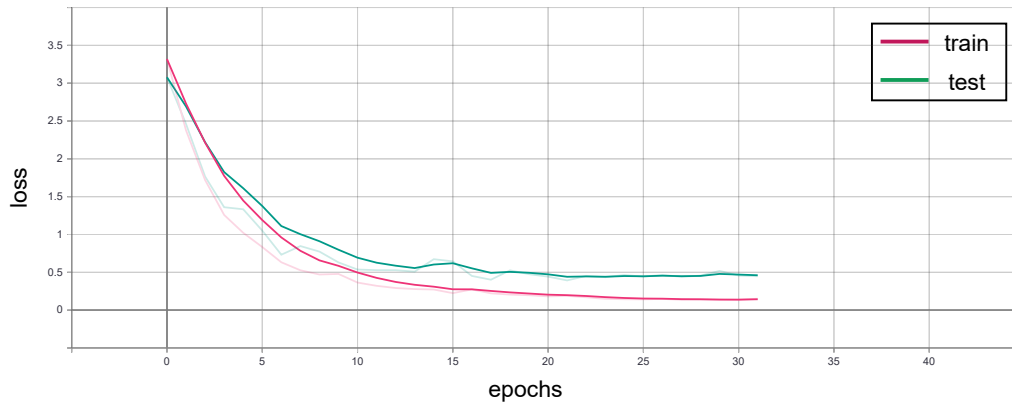


Figure 5.7: Loss Function on full dataset

## 5.2.2 Accuracy Analysis

This time we apply the same procedure to analysis the accuracy of the model. Just like loss function, the accuracy was determined to see the performance of the model over data. On the first phrase, when the feeding data was one-third, the accuracy

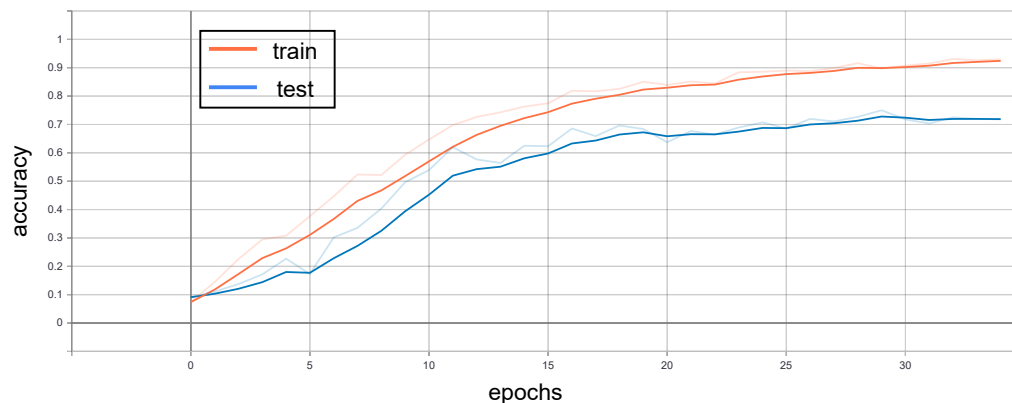


Figure 5.8: Accuracy on one-third dataset

was very poor. In Fig. 5.8, it is visible that the model is not performing well. Though it has managed to gain over 90% accuracy in training data, it has passed barely 70% accuracy in validation data. The only reason behind this poor curve is

the poor amount of training and testing data.

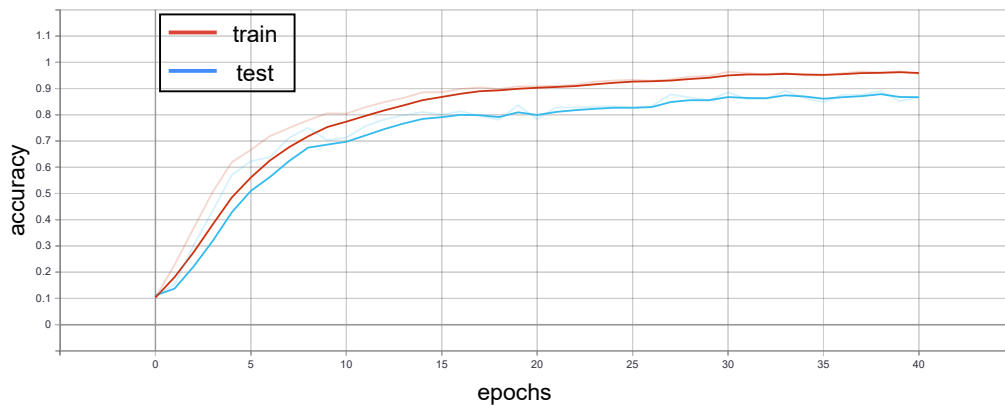


Figure 5.9: Accuracy on two-third dataset

After that, we increased the number of datasets by 2 which means this time we experiment with two-third of our dataset. The model seems to be stable this time in Fig. 5.9. It managed to gain around 96% accuracy over training data and nearly 87% accuracy on testing data. This is satisfactory cause the model performs much better this time.

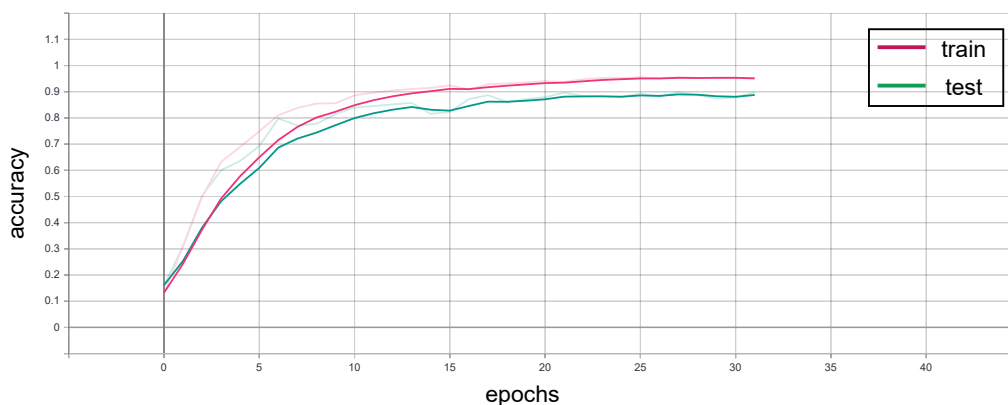


Figure 5.10: Accuracy on full dataset

This time we tried to measure the accuracy of our model by feeding it the whole dataset. In Fig. 5.10, the result is visible. The model did an excellent job though there are still some gaps between training and testing accuracy. The training accuracy managed to gain over 96% while the testing accuracy was around 90%. After all this analysis, we can conclude by saying that if we had more data points, the model may perform better cause it is visible that the model managed to perform well by depending on the number of data it has fed. Table. 5.1 describes the condition of LSTM model over dataset.

Weight	TrainingSamples	TestingSamples	Loss	Accuracy	Condition
One-Third	2240	560	Over 100%	70%	VeryUnstable
Two-Third	4480	1120	60%	87%	Unstable
Full	6721	1681	40%	90%	Stable

Table 5.1: Comparison of loss function and accuracy based on number of data

### 5.3 Benchmark model using Confusion Matrix

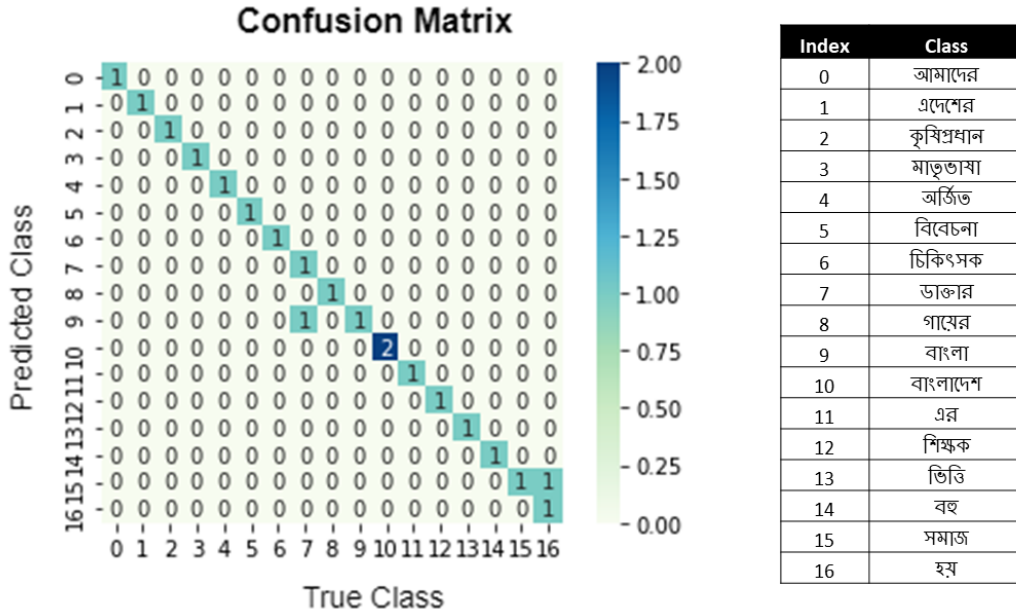


Figure 5.11: Confusion Matrix on Manually tested dataset

We test some cases manually to determine how well the model is performing in real-world data. At this stage, we test the model with 20 cases of 17 different classes. We tried a variety of classes based on their length, complexness, noise, and gender variant. Among those 20 cases, 2 cases were predicted incorrectly. Those two false-negative was “বাংলা” and “সমাজ” but predicted as “ডাক্তার” and “হয়” accordingly. The interesting part is, both wrong prediction was for female voice but the model predicted correctly while we tried the same word for a male voice. This is an interesting finding. Except for this, we tried the longest word of our dataset which is “কৃষিপ্রধান” and the model predicted that correctly. Complex word like “চিকিৎসক” and “শিক্ষক” are also tested by the model and the model predicted them correctly as well. The most amazing part was the model successfully classify the word “এর” which is barely possible even for a human being to understand by listening to the audio. One point we want to mention all those testing data are unique which means those subjects didn’t participate in dataset.

We would like to consider a confusion matrix in Fig. 5.11 to measure how well the model is performing. The indexes of the labels are given in Fig. 5.11. So based on the matrix, we get 18 true positives and 2 false negatives. We got 90% sensitivity, which indicates a positive outcome of the model. We also measured the precision rate of our model which is 100%. Precision determines how precise the model is, so



the higher the precision rate means the better model. Since we got 100% precision rate we can say the model is very precise. Last but not the least, we got 90% accuracy which is good cause it justifies the accuracy measurement which we get previously. This is how we determine how well the model is working where the user is unique and through this, it is proven that it is a user-independent.

## 5.4 Limitations

Though the model accuracy and testing results of the model is seems fine for the time being, there are still some limitations of the model. The loss is around 40% which means the model is still needed to minimize the errors but it needs more data. We worked on 160 unique subjects but that's not enough. From the previous discussion, we came to a point that we need more data to make the model works more fluently. Another reason could be the data which we trained were pretty much about from age 18 to 30. We need more data points from a younger age and older age so that the model can learn any pitch of sound. Another issue we faced is the duration of every word was not the same. The issue arose because the sampling rate started to vary from word to word and to cover up that we need to take a standard value and fill those words with 0 values which could not reach the standard point and had to remove the extra points which overfill the rate. Maybe by improving that the model can perform even better and an accuracy of 100% can be gained with this mode

## 5.5 Error Analysis and Hypothesis

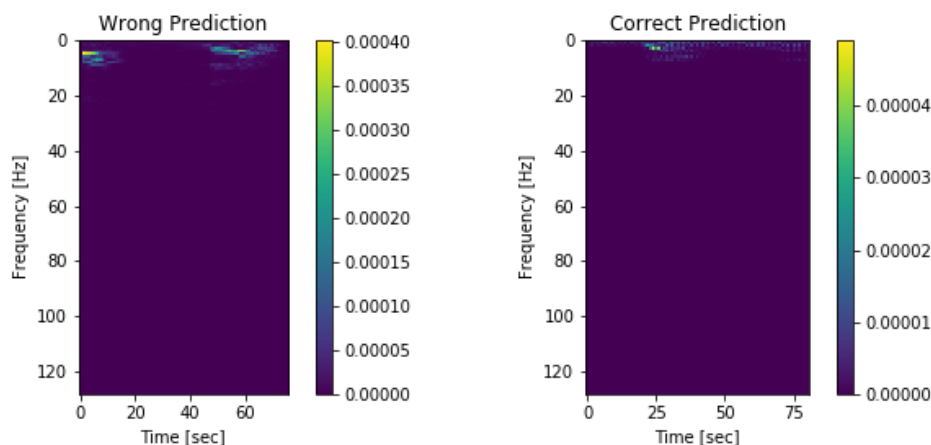


Figure 5.12: Spectrogram of same word

From manual testing we found some interesting findings. For example, same word is predicted in different classes. In first case that word predicted wrong while we tried the case for a female subject. Again we tried that case for a male subject and we get the correct prediction. Let's look at the spectrogram of both case so that we can stand a hypothesis which is given in Fig. 5.12. With a closer look we can find that in wrong prediction word there was frequency gap but in the correct predicted

word there was no such gap. So we can say that that word shouldn't contain any such gap in order to predict correctly.

# Chapter 6

## Conclusion

### 6.1 Conclusion

From the starting, we desired to train the model so perfectly that it can predict any given real-world word. But there was no such dataset that suits us. So, we had to make one of our own. Manually arranging that novel user-independent dataset is our core achievement. The most difficult part was the inequality of amplitude of the audio because of using various mediums to record it and there can be many more factors behind it. But it didn't affect our model that much which reflects on the accuracy of the model. We have a mindset to make it available in the public domain. Moreover, our achieved work matches with our hypothesis which is we can build a network with LSTM which can predict real-world data in real-time. Before concluding, we want to add, we can further work on it to predict a whole real world sentence in real-time.

### 6.2 Future Work

We are at the edge of the research where we successfully implement a system that can predict 56 Bengali words from a given speech of any independent subject. The number of predicted words can be increased by increasing the training labels. Also, we would like to predict a full sentence with this model which can be implemented if we can separate each word from a sentence and try to predict them one by one. Though we are not sure about the time and space complexity but we can work on that too once we successfully predict the sentence. After that, the most challenging part will come which is preparing an answer for the sentence that the system predicted. After accomplishing the task, only one phrase will remain which is output the reply with machine voice. In this step, we want to do something unique which has never done before. We want to build a system that can change the sentiment depending on the reply. So far, all the advanced machines can successfully speak just like a human being but it can't express the emotions like humans. This is the sector where a machine is different from a human. So, we want to erase these static feelings and make a perfect system which can express its feeling through voice; based on the reply.

# Bibliography

- [1] Yoon Kim and J. O. Smith, “A speech feature based on bark frequency warping-the non-uniform linear prediction (nlp) cepstrum”, in *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. WASPAA '99 (Cat. No.99TH8452)*, 1999, pp. 131–134.
- [2] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks”, *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 38, Mar. 2013. DOI: 10.1109/ICASSP.2013.6638947.
- [3] R. Sandanalakshmi, P. A. Viji, M. Kiruthiga, M. Manjari, and M. Sharina, *Speaker independent continuous speech to text converter for mobile application*, 2013. eprint: arXiv:1307.5736.
- [4] K. A. Hasan, M. Hozafa, S. Dutta, and R. Z. Rabbi, “A framework for bangla text to speech synthesis”, in *16th Int'l Conf. Computer and Information Technology*, IEEE, 2014, pp. 60–64.
- [5] R. Sultana and R. Palit, “A survey on bengali speech-to-text recognition techniques”, in *2014 9th International Forum on Strategic Technology (IFOST)*, 2014, pp. 26–29.
- [6] D. Britz, *Recurrent neural networks tutorial, part 3 – backpropagation through time and vanishing gradients*, Apr. 2016. [Online]. Available: <http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-part-3-backpropagation-through-time-and-vanishing-gradients/>.
- [7] P. Vanajakshi and M. Mathivanan, “A detailed survey on large vocabulary continuous speech recognition techniques”, in *2017 International Conference on Computer Communication and Informatics (ICCCI)*, 2017, pp. 1–7.
- [8] A. Amberkar, P. Awasarmol, G. Deshmukh, and P. Dave, “Speech recognition using recurrent neural networks”, Mar. 2018, pp. 1–4. DOI: 10.1109/ICCTCT.2018.8551185.
- [9] A. Gupta and A. Joshi, “Speech recognition using artificial neural network”, in *2018 International Conference on Communication and Signal Processing (ICCSP)*, 2018, pp. 0068–0071.
- [10] M. T. Tausif, S. Chowdhury, M. S. Hawlader, M. Hasanuzzaman, and H. Heickal, “Deep learning based bangla speech-to-text conversion”, in *2018 5th International Conference on Computational Science/ Intelligence and Applied Informatics (CSII)*, 2018, pp. 49–54.

- [11] S. Abujar, A. Mohammad Masum, S. M. M. H. Chowdhury, M. Hasan, and S. Hossain, “Bengali text generation using bi-directional rnn”, Jul. 2019. DOI: 10.1109/ICCCNT45670.2019.8944784.
- [12] M. A. A. Amin, M. Islam, S. Kibria, and M. Rahman, “Continuous bengali speech recognition based on deep neural network”, Feb. 2019, pp. 1–6. DOI: 10.1109/ECACE.2019.8679341.
- [13] Y.-C. Chen, S.-F. Huang, H.-y. Lee, and L.-s. Lee, *From semi-supervised to almost-unsupervised speech recognition with very-low resource by jointly learning phonetic structures from audio and text embeddings*, 2019. eprint: arXiv:1904.05078.
- [14] F. Haque, M. M. H. Manik, and M. Hashem, “Opinion mining from bangla and phonetic bangla reviews using vectorization methods”, in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, IEEE, 2019, pp. 1–6.
- [15] N. S. I. Nirjhor M. Abidur, “Bangla speech recognition using 1d cnn and lstm with different dimension reduction techniques”, Brac University, 2019.
- [16] A. Khatun, A. Rahman, H. A. Chowdhury, M. S. Islam, and A. Tasnim, “A subword level language model for bangla language”, in *International Joint Conference on Computational Intelligence*, Springer, 2019, pp. 385–396.
- [17] M. R. Mia and A. S. M. L. Hoque, “Question bank similarity searching system (qb3s) using nlp and information retrieval technique”, in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, IEEE, 2019, pp. 1–7.
- [18] A. Rai, *Recurrent neural network-head to toe*, Oct. 2019. [Online]. Available: <https://towardsdatascience.com/recurrent-neural-network-head-to-toe-d58ff2f2dab3>.
- [19] Z. Chen, X. Fan, Y. Ling, L. Mathias, and C. Guo, *Pre-training for query rewriting in a spoken language understanding system*, 2020. eprint: arXiv:2002.05607.
- [20] M. H. Firmansyah, A. Paul, D. Bhattacharya, and G. M. Urfa, *A.i. based embedded speech to text using deepspeech*, 2020. eprint: arXiv:2002.12830.
- [21] L. Mehedy, S. M. N. Arifin, and M. Kaykobad, “Bangla syntax analysis: A comprehensive approach”, Oct. 2020.
- [22] M. Phi, *Illustrated guide to lstm’s and gru’s: A step by step explanation*, Jun. 2020. [Online]. Available: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>.
- [23] —, *Illustrated guide to recurrent neural networks*, Jun. 2020. [Online]. Available: <https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9>.
- [24] R. Riad, A.-C. Bachoud-Lévi, F. Rudzicz, and E. Dupoux, *Identification of primary and collateral tracks in stuttered speech*, 2020. eprint: arXiv:2003.01018.
- [25] R. Habib and M. M. Anwar, “Finding out noisy patterns for relation extraction of bangla sentences”,