Bangla Voice Based Assistance for Mobile Device

Authors

Muhammad Atiqur Rahman Imon	Student Id: 084407
Md. Mushfiqur Rahman	Student Id: 084419

Supervisor

Md. Mohiuddin Khan
Assistant Professor
Department of Computer Science &Engineering (CSE)
Islamic University of Technology (IUT)

Co-Supervisor

Moin Mahmud Tanvee
Lecturer
Department of Computer Science &Engineering (CSE)
Islamic University of Technology (IUT)

**A Thesis submitted to the Department of Computer Science & Engineering (CSE)
in Partial Fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science & Engineering (CSE)**

Department of Computer Science & Engineering (CSE)
Islamic University of Technology (IUT)
Organization of the Islamic Cooperation (OIC)
Gazipur, Bangladesh

October, 2012

CERTIFICATE OF RESEARCH

This is to certify that the work presented in this thesis paper is the outcome of the research carried out by the candidates under the supervision of Md. Mohiuddin Khan, Assistant Professor, Department of Computer Science and Engineering, IUT and co-supervision of Moin Mahmud Tanvee**,** Lecturer, Department of Computer Science and Engineering, IUT, Gazipur. It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree or any judgment.

Authors

______________________________ ______________________________

Muhammad Atiqur Rahman Imon **Md. Mushfiqur Rahman**

Signature of Co-Supervisor

____

Moin Mahmud Tanvee
Lecturer
Department of CSE, IUT

Signature of Supervisor

____

Md. Mohiuddin Khan
Assistant Professor
Department of CSE, IUT

Signature of the Head of the Department

____

Prof. Dr. M. A. Mottalib
Head, Department of CSE, IUT

Abstract

Bangla speech recognition is a relatively young area of research and we have not seen much success so far. Pattern recognition approach is generally used for speech recognition. CMUSphinx is a framework which uses Hidden Markov Model (HMM) for pattern training and n-gram technique to build a language model from the speech corpus which can be handy for building speech recognition systems. The success of speech recognition mostly depends on the speech corpus and a well-trained acoustic model. Such a speech recognizer, implemented in mobile devices, can have tremendous implication on our day to day life. However, to build an efficient acoustic model we need an extensive amount of training data. In this thesis work, we have shown how CMUSphinx can be used to build an acoustic model for Bangla. We have built several acoustic models and tried to improve the accuracy rate. One of our trained models has achieved good accuracy rate. In the latter part of the thesis, we implemented the speech recognizer in Android platform. In this process, we investigated some problems those have to be solved to get comparable accuracy rate in Android. We have also proposed a model for the future continuation of the research.

Table of Contents

Chapter 1: Introduction

Speech recognition is a complex process of converting analog speech signal to text. If we can convert the speech signal to its corresponding text accurately then we can use the speech signal as an input mechanism in our computers and mobile devices. Research on speech recognition for English language is going on from 1970s and we already have found quite established speech recognition system for English. But research on Bangla speech recognition is a relatively new field in this arena. As Bangla is a morphologically enriched language it is not a very easy task to convert Bangla speech signal to its corresponding text. The researches done so far on Bangla speech recognition have produced poor accuracy of recognition. As there is no established speech recognition system for Bangla, we are interested to do research to improve the accuracy rate. At the first step we have tried to build a system for mobile device that can understand few Bangla commands.

1.1 Motivation

Bangla is the first language of over 200 million people. But there is no established speech recognizer system for this language. Although some systems were developed for research purpose but we have very few applications those are really helpful. We are motivated to do research on Bangla speech recognition for mobile device, to be specific we are interested to make Android phones understand Bangla and take Bangla speech as a command. Though Android phones have nicely managed GUI features with extended touch facilities sometimes it seems to be overwhelming to manage so many interfaces with the keypad and touch screen. If speech input is available then the burden on GUI is reduced. Our mass population isn’t well educated. It would be very helpful to them if their cellphones can understand their speech. Same help for the disabled persons, drivers and cooks.

1.2 State of the art

As we have mentioned before, research on Bangla speech recognition is still at its early level. At present we don’t have any established Bangla speech recognizer. Using the pattern recognition approach, only few works has been done in CRBLP (http://crblp.bracu.ac.bd/) (Centre for Research on Bangla Language Processing) of BRAC University. They have used the well-known CMUSphinx framework. They have showed some good accuracy rate in desktop computer but only the word “phonebook” was recognized in Android phone.

1.3Goal of the thesis

The first goal of our research is to understand different methodologies of implementing speech recognition system. If we get the general idea then we can try to implement it in our desktop computer and try to get a better accuracy rate. We have to develop an efficient language model and collect sufficient amount of training data for this process. Our final goal is to implement the system in mobile devices like Android phones for few commands in Bangla. For the first iteration, we have limited our domain to a fixed part of Bangla commands.

Chapter 2: Basics of Speech Recognition

This chapter is written based on our study on the basics of speech recognition [2] [3].

2.1 Terminologies

Here we are giving a brief idea of the terms we will frequently encounter in this report.

Phone: Phone is a smallest unit of a sound stream which can be identified distinctly. Generally phones are represented in IPA (International Phonetic Alphabet).

ISR: Isolated Speech Recognition, here the words are isolated. Speakers must pause between two consecutive words.

CSR: Continuous Speech Recognition, here speakers speak continuously without any pauses between words. In our thesis work we focus on continuous speech recognition.

Corpus: Corpus is the list of sentences that will be used in the speech recognition system.

Vocabularies: Vocabularies are the list of words that is recognized by speech recognition system.

Training: There are many variations in speech utterance like pitch, accent, and frequency. That's why speech recognition system uses machine learning approach. In machine learning approach we need extensive amount of training data to train the system.

2.2 Speech Recognition Approaches

There are three basic approaches for automatic speech recognition (ASR):

1. The acoustic-phonetic approach
2. The pattern recognition approach
3. The artificial intelligence approach

The acoustic-phonetic approach is based on the theory of acoustic-phonetics that postulates that there exist finite, distinctive phonetic units in spoken language and they are characterized by a set of properties that are manifest in speech signal or spectrum over time. The first step of the acoustic phonetic approach is called segmentation & labeling phase. Here the speech is segmented into discrete region (in time) where the acoustic properties of the signal are representative of one phonetic units & then attaching one or more phonetic labels to each segmented region. The second step attempts to determine a valid word from the sequence of phonetic labels produced in first step.

The pattern-recognition approach is basically one in which the speech patterns are used directly without explicit feature determination (in the acoustic-phonetic sense) &segmentation. This method has two steps – training of speech patterns & recognition of patterns via pattern comparison. The concept is that if enough versions of a pattern are included in a training set provided to the algorithm, the training procedure should be able to adequately characterize the acoustic properties of the pattern. This type of characterization of speech via training is called pattern classification. In pattern comparison stage a direct comparison of the unknown speech is done with all each possible pattern learned in the training phase & classifies the unknown speech according to the goodness of match of the patterns. Figure 1 depicts the pattern recognition approach clearly.

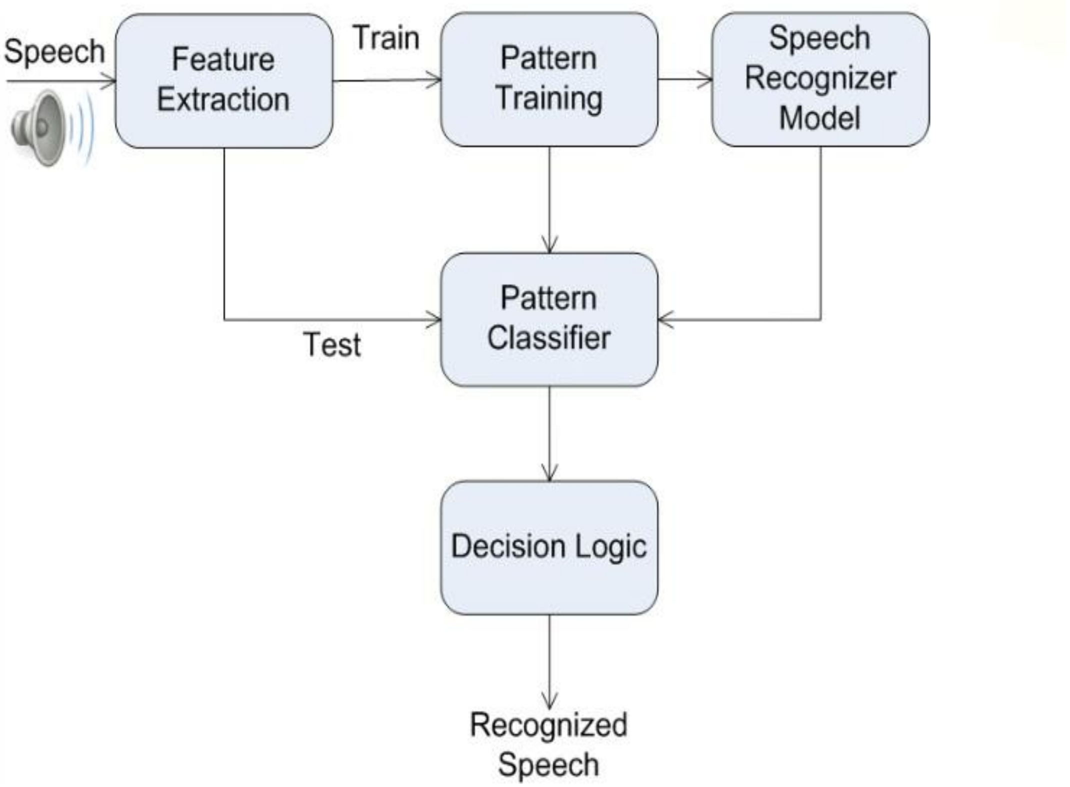


Figure 1: Pattern Recognition Approach of speech recognition [3]

The artificial intelligence approach to speech recognition is a hybrid of the acoustic phonetic approach & the pattern recognition approach. It exploits the ideas & concept of both methods. The artificial intelligence method uses expert system for segmentation & labeling; learning & adapting over time; the use of neural networks for learning relationship between phonetic events & all known inputs as well as discrimination for similar sound classes.

In our thesis work we are using the pattern recognition approach. In the following sections we will describe the steps of pattern recognition approach used in speech recognition.

2.3 Feature Extraction

The input of a speech recognizer is speech signal. We need to know how the speech signal is represented to the recognizer. The first step is to know the features of the speech signal. In feature extraction process the features of a signal is represented by different feature vectors. These vectors are known as MFCC (Mel Frequency Cepstrum Coefficient) vectors. We get the MFCC vectors through a process of applying several techniques on the input signal. The complete overview of feature extraction is shown in Figure 2.

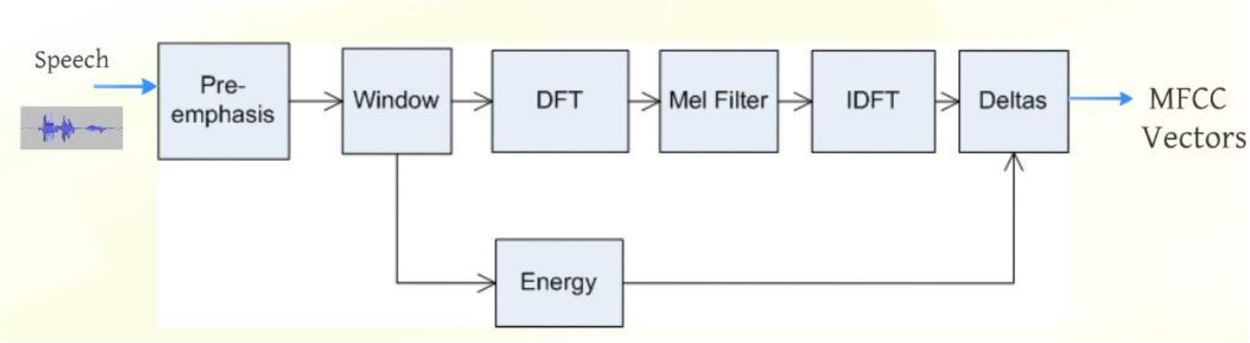


Figure 2: Feature Extraction Procedure [2]

2.3.1 Pre-emphasis

Pre-emphasis is the process of boosting up the energy level of the signal in high frequencies. Boosting the energy of higher frequencies will make it easier to acoustic model to extract the features.

2.3.2 Windowing

The main goal of feature extraction is to provide spectral features that will help us to build phone classifier. If we take the whole utterance spectrum we will not be able to extract features accurately because speech signal is non-stationary (changes very quickly). This is why we take small window for feature extraction. In a small window speech signals show stationary property.

2.3.3 Discrete Fourier Transformation

We need Discrete Fourier Transformation to get the spectral information from the windowed signal. The input to DFT is a windowed signal and for every frequency band output is the magnitude and phase of the original frequency component of the signal. We get the spectrum of the signal by plotting the magnitude against frequency.

2.3.4 Mel Filter Bank

To improve the accuracy of speech recognition system we create human like hearing by using Mel filter bank. Humans are less sensitive to higher frequencies (above 1000Hz). Mel filter scales the lower frequencies linearly and logarithmically above 1000Hz.

2.3.5 Cepstrum: Inverse Discrete Fourier Transformation

The main goal of the IDFT is to find the exact vocal tract of the phones from the signal. We know every phone has an unique vocal tract. Hence, if we can distinguish the vocal tracts from the signal we can easily determine the underlying phones of that signal. To do that the spectrum of the log of the original spectrum (DFT) is taken. That's why it is called Inverse DFT.

2.3.6 Deltas and Energy

The extraction of the cepstrum via the Inverse DFT from the previous section results in 12 cepstral coefficients for each frame. We next add a thirteenth feature: the energy from the frame. Energy correlates with phone identity and so is a useful cue for phone detection (vowels and sibilants have more energy than stops, energy and so on).

2.4 Pattern Training: HMM

Hidden Markov Model is the most important machine learning algorithm in speech recognition. To understand HMM we first need to understand Markov Chain sometimes called Observed Markov Model.

2.4.1 Markov Chain

A Markov chain is a special case of a weighted automaton in which the input sequence uniquely determines which states the automaton will go through. A Markov chain is specified by the following components:

Q = q_1,q_2,... q_N a set of N states.
A = a_{01}, a_{02},... a_{n1}... a_{nn} a transition probability matrix.
q_0, q_F a special start state & final state which are not associated with the observations.

In a first-order Markov chain, the probability of a particular state is dependent only on the previous state,

Markov Assumption: p($q_i \mid q_1$ ...q_{i-1}) = p($q_i \mid q_{i-1}$)

As a_{ij}expresses the probability p($q_i \mid q_j$) , the law of probability require that the values of the outgoing arcs from a given state must sum to 1.

Another representation is used for Markov chain that doesn't rely on a start or end state. It represents the distribution over initial states & accepting states explicitly:

π = π_1, π_2, ... π_N an initial probability distribution over states. π_i is the probability that the

Markov chain will start in state *i*. Some states *j* may have π_j = 0 indicate that they cannot be initial state.

Q_A a set of accepted states.

2.4.2 Hidden Markov Model

We can use the Markov Chain to compute the probability from a sequence of observable events. However, events may not be observable in all the cases, maybe we can have the results of the unobservable or partially observable events, then it is the case where events are hidden and here comes the Hidden Markov Model (HMM). In case of speech recognition we see the acoustic event and we need to tell about the hidden words in the acoustic signal and hence the necessity of HMM in speech recognition.

Formal definition of HMM:

Q = q_1,q_2,... q_N a set of N states

A = a_{01},a_{02},...a_{n1}... a_{nn} a transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j, and Summation of a_{ij}= 1 (j= 1 to n).

O = O_1, O_2 ,... O_T a sequence of T observations, each one drawn from a vocabulary V = $v_1, v_2, \ldots, v_V$,

B = $b_i(O_t\ )$ a sequence of observation likelihoods also called emission probabilities, each expressing the probability of an observation O_t being generated from a state i.

q_0, q_F a special start state and end (final) state which are not associated with observations, together with transition probabilities a_{01}, a_{02},... a_{n1}... a_{nn} out of the start state and $a_{1F}, a_{2F}, \ldots a_{nF}$ into the end state.

As we noted for Markov chains, an alternate representation that is sometimes used for HMMs doesn't rely on a start or end state, instead representing the distribution over initial and accepting states explicitly.

π = π_1, π_2, ... π_N an initial probability distribution over states. π_i is the probability that the Markov chain will start in state i. Some states j may have π_j = 0 indicate that they cannot be initial state and summation of π_j=1 where j=1 to n.

Q_A = {q_x,q_y,...} a set $Q_A \subset$ Q of legal accepting states.

There is two assumptions taken by first order HMM to simplify things.

First, as with a first-order Markov chain, the probability of a particular state is dependent only on the previous state:
Markov Assumption: p($q_i|q_1$ ...q_{i-1}) = p($q_i|q_{i-1}$)

Second, the probability of an output observation O_i is dependent only on the state that produced the observation q_i, and not on any other states or any other observations:
Output Independence Assumption: p($O_i|q_1 \ldots q_i, \ldots, q_T, O_1, \ldots, O_i, \ldots, O_T$)=p($O_i|q_i$)

2.4.3 Types of HMM

One way to classify types of HMM is by the structure of the transition matrix of Markov chain.

Fully Connected Or Ergodic:

In ergodic model every state can be reached from every other state of the model in a single step. In other words, there is a non-zero probability between any two states of the model.

Left-to-right Or Bakis:

In a Bakis HMM there are no transitions (or transition with zero probability) going from a higher-numbered state to a lower-numbered state. Bakis HMMs are generally used to model temporal processes like speech.

2.4.4 Three fundamental problems solved by HMM

Problem 1 (Computing Likelihood): Given an HMM and an observation sequence O, determine the likelihood P(O|HMM).

Problem 2 (Decoding): Given an observation sequence O and an HMM, discover the best hidden state sequence Q.

Problem 3 (Learning): Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

2.4.4.1 Computing Likelihood : Forward Algorithm

The first problem of HMM is to compute the likelihood of a particular observation sequence. That is given HMM λ=(A,B,π)and an observation sequence O, determine the likelihood P(O| λ).

The forward algorithm is a dynamic programming algorithm. It computes the observation probability by summing over the probabilities of all possible hidden state paths that could generate the observation sequence, but does it efficiently by implicitly folding each of these paths into a single forward trellis.

Figure 3 shows an example of the forward trellis for computing the likelihood of o_1 o_2 o_3:

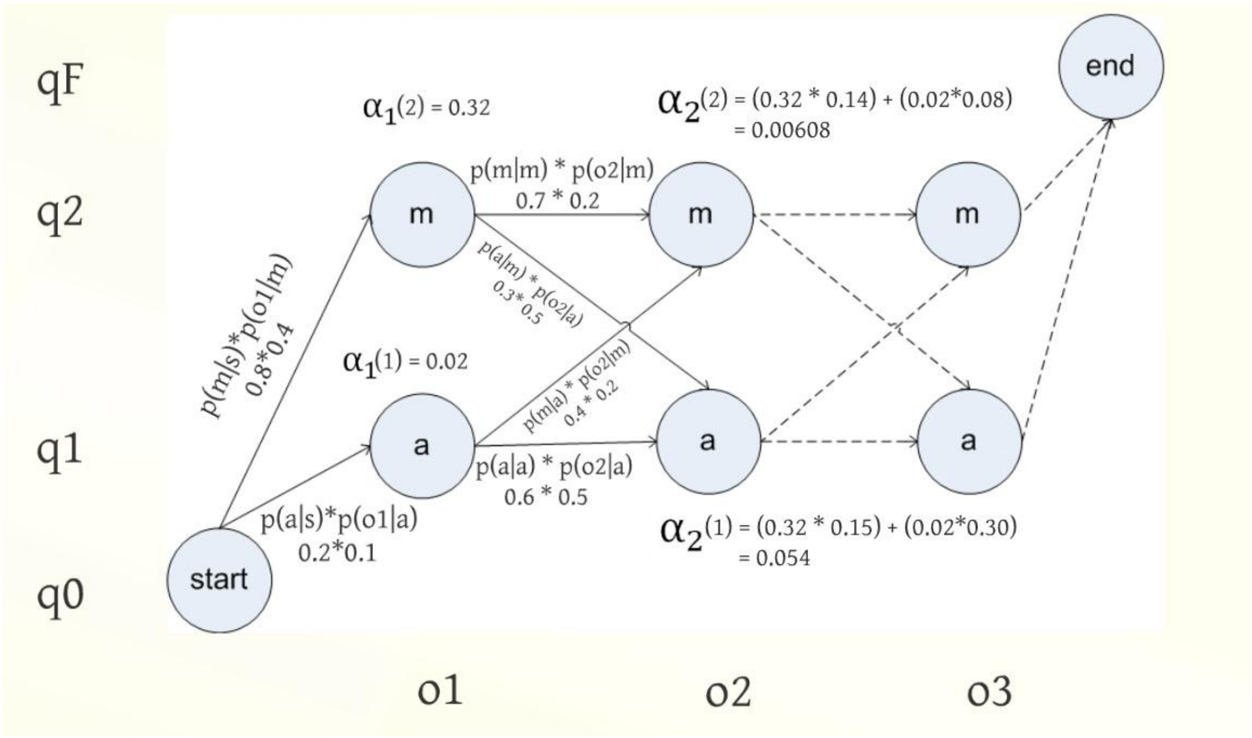


Figure 3: Calculation of Forward algorithm for word মা

Each cell of the forward algorithm trellis $\alpha_t(j)$ represents the probability of being in state j after seeing the first t observations, given the automaton λ.

The value of each cell $\alpha_t(j)$ is computed by summing over the probabilities of every path that could lead us to this cell. Formally, each cell expresses the following probability:

$$\alpha_t(j)=P(o_1,o_2...o_t,q_t=j|\ \lambda)$$

Here $q_t=j$ means the probability that the t^{th} state in the sequence of states is state j.
we compute this probability by summing over the extensions of all paths that lead to the current cell. For a given state q_j at the time t, the value $\alpha_t(j)$ is computed as:

$$\alpha_t(j)=\sum^{N}_{i=1} \alpha_{t-1}(i)a_{ij}b_j(o_t)$$

here,

$\alpha_{t-1}(i)$ the previous forward path probability from the previous time step
a_{ij} the transition probability from previous state q_j
$b_j(o_t)$ the state observation likelihood of the observation symbol o_t given the current state j.

The formal definition of forward algorithm:

1. Initialization:

$$\alpha_1(j)=a_{0j}b_j(o_t) \quad 1<=j<=N$$

2. Recursion:

$$\alpha_t(j)=\sum^{N}_{i=1} \alpha_{t-1}(i)a_{ij}b_j(o_t);\ 1<=j<=N, 1<t<=T$$

3. Termination:

$$p(O|\ \lambda)=\alpha_T(q_F)=\sum^{N}_{i=1} \alpha_T(i)a_{iF}$$

2.4.4.2 Decoding: Viterbi Algorithm

The second basic problem of HMM is decoding problem. Formally:

DECODING: Given as input an HMM $\lambda=(A,B,\pi)$and a sequence of observations $O=o_1,o_2,...,o_T$, find the most probable sequence of states $Q=q_1q_2q_3...q_T$.

The most common decoding algorithm for HMM is the Viterbi algorithm. It is a kind of dynamic programming. Figure 4shows an example of the Viterbi trellis for computing the best hidden state sequence for the observation sequence o_1 o_2 o_3.

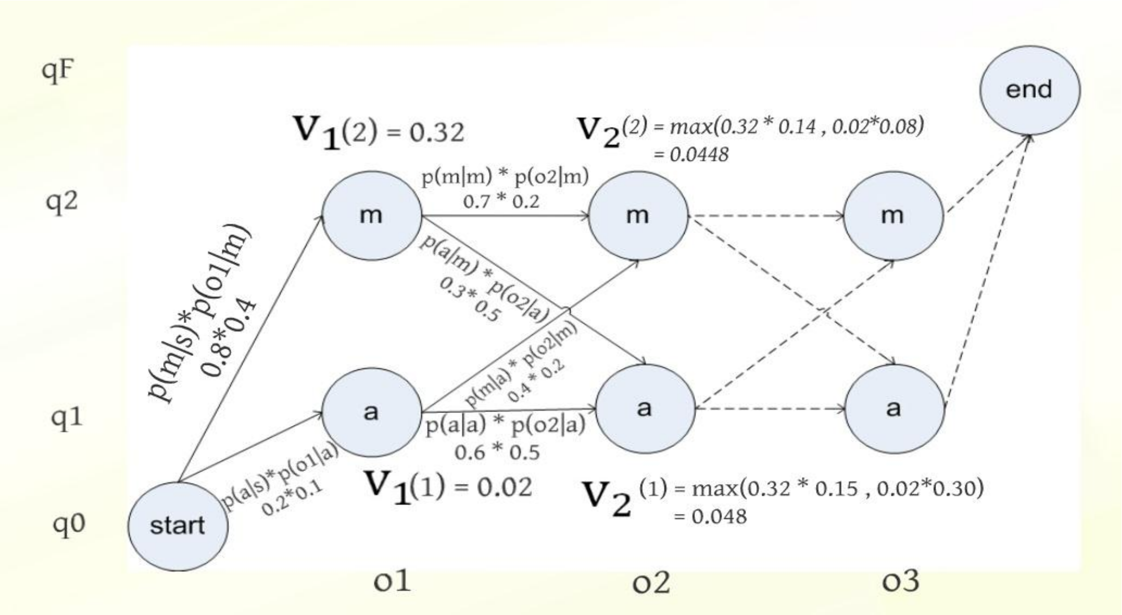


Figure 4: Calculation of viterbi algorithm for word মা

The idea is to process the observation sequence left to right, filling out the trellis. Each cell of Viterbi trellis, $v_t(j)$ represents the probability that the HMM is in state j after seeing the first t observations and passing through the most probable state sequence $q_0,q_1,...,q_{t-1}$, given the automaton λ. Formally:

$$v_t(j) = \max_{q0,q1,...,qt-1}\ P(q_0,q_1,...,q_{t-1},o_1,o_2,...o_t,q_t=j|\ \lambda)$$

Like other dynamic programming algorithms, Viterbi fills each cell recursively. For a given state q_j at time t, the value $v_t(j)$ is computed as:

$v_t(j) = \max^N_{i=1} v_{t-1}(i) a_{ij} b_j(o_t)$

where,

$v_{t-1}(i)$ the previous Viterbi path probability from previous time step.
a_{ij} the transition probability from previous state q_i to current state q_j.
$b_j(o_t)$ the state observation likelihood of the observation symbol o_t given the current state j.

Formal definition of Viterbi algorithm:

Initialization:

$v_t(j) = a_{oj} b_j(o_1)$ 1<=j<=N
$bt_1(j) = 0$

Recursion:

$v_t(j) = \max^N_{i=1} v_{t-1}(i) a_{ij} b_j(o_t)$; 1<=j<=N,1<t<=T
$bt_1(j) = \text{argmax}^N_{i=1} v_{t-1}(i) a_{ij} b_j(o_t)$; 1<=j<=N,1<t<=T

Termination:

The best Score : $P^* = v_t(qF) = \max^N_{i=1} v_T(i) a_{i.F}$
The start of backtrace :$qT^* = bt_T(qF) = \text{argmax}^N_{i=1} v_T(i) a_{i.F}$

Viterbi Algorithm Vs Forward Algorithm:

1. The Viterbi algorithm is identical to the forward algorithm except that it takes max over the previous path probabilities where the forward algorithm takes the sum.

2. The Viterbi algorithm has one component that the forward algorithm doesn't have backpointers; because the Viterbi algorithm must produce a probability and also the most likely state sequence. We compute this best state sequence by keeping track of the path of hidden states that lead to each state and at the end tracking back the best path to the beginning (Viterbi backtrace).

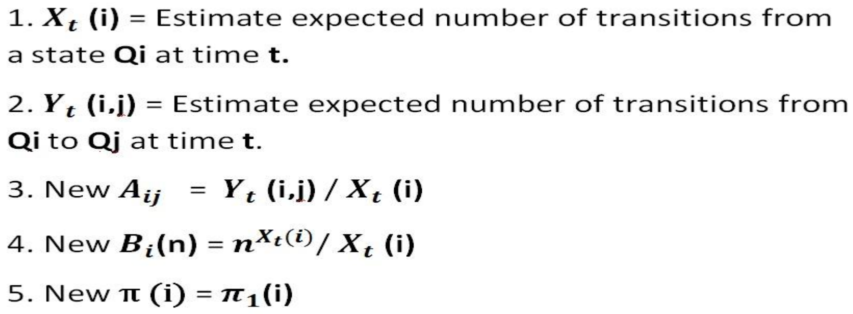
2.4.4.3 Learning: Baum-welch Algorithm

Baum-welch algorithm is used for learning the HMM model parameters. It iteratively assigns new values to A, B and π from the previous value of the parameters. The more the training data provided the more the new probability distribution becomes realistic.

- Iterative programming

- Forward, Viterbi and backward process

Here is the simplified process of algorithm [2]:



1. $\boldsymbol{X_t}$ **(i)** = Estimate expected number of transitions from a state **Qi** at time **t.**
2. $\boldsymbol{Y_t}$ **(i,j)** = Estimate expected number of transitions from **Qi** to **Qj** at time **t**.
3. New $\boldsymbol{A_{ij}}$ = $\boldsymbol{Y_t}$ **(i,j) /** $\boldsymbol{X_t}$ **(i)**
4. New $\boldsymbol{B_i}$**(n)** = $\boldsymbol{n^{X_t(i)}}$/ $\boldsymbol{X_t}$ **(i)**
5. New **π (i)** = $\boldsymbol{\pi_1}$**(i)**

2.5 N-Gram

An *n*-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a (n-1) order Markov model. We can describe N-gram model as: "given a sequence of letters/words what is the likelihood of the next letter/word?" Such statistical model of word sequence is also called Language Model.

For a sequence of words, (for example "<s> the dog smelled like a skunk </s>"), the trigrams (3-grams) would be: "<s> the dog", "the dog smelled", "dog smelled like", "smelled like a", "like a skunk", and "a skunk </s>".

For a sequence of characters, the 3-grams (sometimes referred to as "trigrams") that can be generated from "good morning" are "goo", "ood", "od ", "d m", " mo", "mor", and so forth.

Chapter 3: Proposed Mechanism and Implementation Procedure

We propose to use the pattern recognition approach to recognize Bangla speech. This approach has been used for some other languages and an acceptable accuracy rate has been achieved. However for Bangla, no such system is implemented. As Bangla is a morphologically enriched language [13] hence it is very hard to create a good transcription of speech samples and thus good acoustic model. So our approach is an iterative method that will be carried on till we get the expected accuracy rate. We expect this procedure will make the system a robust one, a user independent speech recognition system. Furthermore we will import this model into Android mobile phone which will take Bangla command as another interaction method along with touch and keypad.

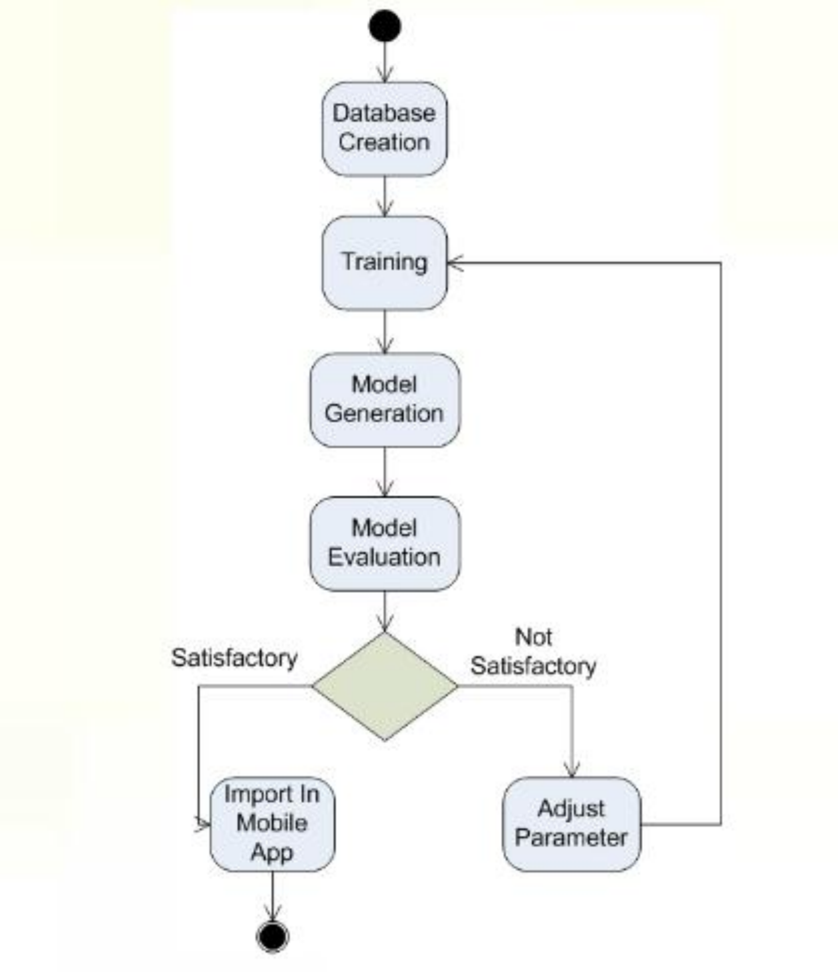


Figure 5: Activity Diagram of proposed mechanism

3.1 Choosing Framework

The available frameworks for speech recognition are CMUSphinx, HTK, Julius etc. We are using CMUSphinx for the following reasons:

- Written in C & java
- Runs on any platform
- Open source
- Pocket sphinx is dedicated framework for mobile phones
- Well documented
- Active forums

3.2 Framework Installation

The frameworks we used is consists of the following components :

- Sphinxbase
- Sphinxtrain
- Pocketsphinx
- CMULMToolkit

We install the softwares by using these commands:

```
$sudo ./autogen.sh

$sudo ./configure

$sudo make

$sudo make install
```

Then we need to export some variables. The commands are:

```
$export LD_LIBRARY_PATH=/usr/local/lib

$export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
```

To test if pocketsphinx is working go to pocketsphinx folder and write the command:

```
$pocketsphinx_continuous
```

There may be some error in doing this. Some packages may need to install:

```
$sudo apt-get install autoconfig

$sudo apt-get install libtool
```

```
$sudo apt-get install automake
```

3.3 Database Architecture

To create a speech recognizer model in CMUSphinx framework, we need to create a database. The CMUSphinx documentation helped us to understand the database architecture.The database should follow the hierarchy depicted in Figure 6.

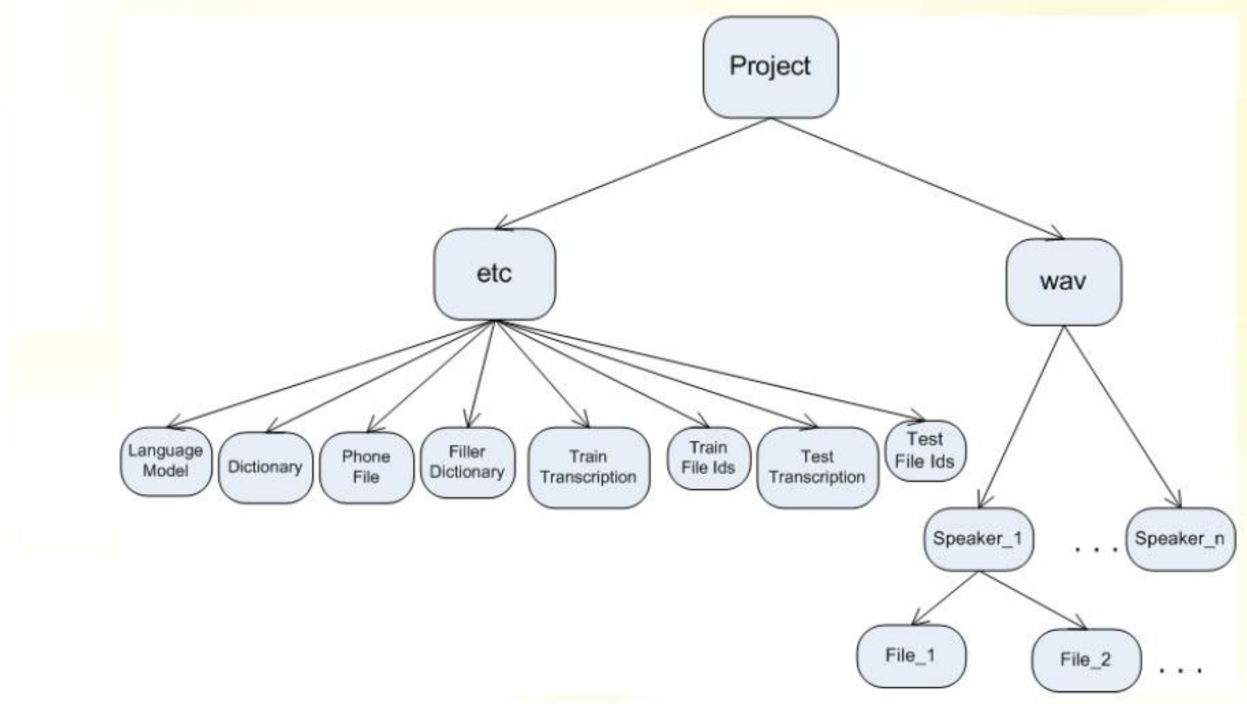


Figure 6: Database Architecture of our project

The following sections describe each component of the database.

3.3.1 Language Model (LM) Creation

Language Model is created by CMULMToolkit. For creating a language model first we have to create a corpus. Then from that corpus, using CMULMToolkit we create Language Model. The steps of creating a LM is shown in the Figure 7.

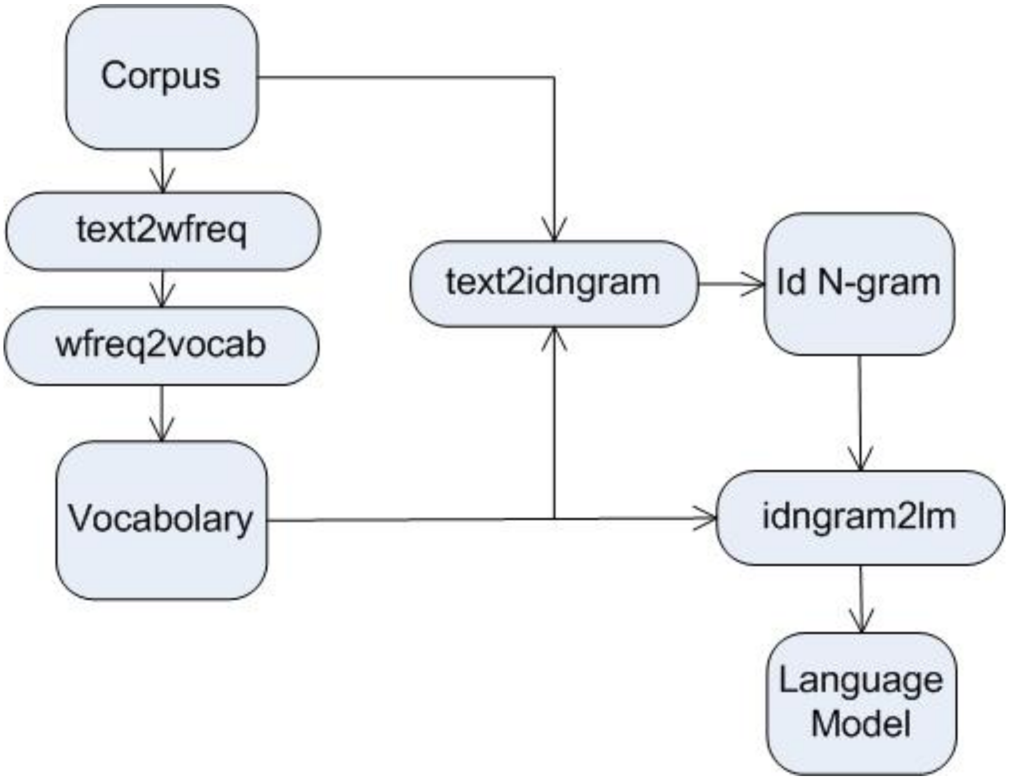


Figure 7: Language Model creation [12]

Now we will show step by step commands for creating a LM using CMULMToolkit by an example. If the corpus of our LM is in a file named *a.txt* then the following commands will generate a LM for this corpus:

```
$ cat a.txt | text2wfreq > a.wfreq
$ cat a.wfreq | wfreq2vocab -top 20000 > a.vocab
$ text2idngram -vocab a.vocab -idngram a.idngram < a.txt
$ idngram2lm -vocab_type 0 -idngram a.idngram -vocab a.vocab –arpa
a.arpa
$ sphinx_lm_convert -i a.arpa -o a.lm.DMP
```

The file *a.txt* should be in the format:

<s>আমাকে কলের তালিকা দেখাও</s>

The corpus of our system is given in the appendix.

3.3.2 Dictionary

The file dictionary contains the words with their corresponding phones. The format of the dictionary file should be like:

আমাকে a ma k e

কলের k O l e r

The dictionary of our system is given in appendix.

3.3.3 Filler Dictionary

Filler Dictionary contains filler phones (not-covered by language model non-linguistic sounds like breathe, silence, hmm or laugh). An example of filler dictionary is given below:

```
<s>  SIL

</s>  SIL

SIL  SIL
```

3.3.4 Phone File

Phone file contains one phone per line. The number of phones should match the number of phones used in the dictionary. Bangla phones are generally represented in IPA (International Phonetic Alphabet) which contains Unicode characters. However, for using CMUSphinx framework we need to use ASCII characters for representing phones. Hence, we need to convert IPA to ASCII. The idea is we have to assign a unique character or a character sequence for a particular phone. The conversion is done using IPA to ASCII conversion chart. An example of phone file is given bellow:

```
a

m

k

e
```

Another important thing is phone-set must be alphanumeric-only and all phones must be different even in case-insensitive variation (case-sensitive variants: “e” and “E”).

3.3.5 Transcription File

Transcription (train transcription and test transcription) file is a text file listing the transcription for each audio file. The transcription files for our system are given in appendix. Here, we will give an example of transcription file:

<s>আমাকে কলের তালিকা দেখাও</s> (file_1)

Here, file_1 is the utterance ID that is the name of the wav file.

3.3.6 Fileids File

Fileids (train fileids and test fileids) file is a text file listing the names of the recordings (utterance ids) one by line, for example

```
Speaker_1/file_1

Speaker_2/file_2
```

Fileids file contains the path in a file system relative to wav directory. Fileids file should have no extensions for audio files, just the names. It's critical to have exact match between fileids file and the transcription file. The number of lines in both should be identical. Last part of the file id (speaker1/file_1) and the utterance id file_1 must be the same on each line.

3.3.7 Speech Recordings

Recording files *(wav files)* must be in MS WAV format with specific sample rate - 16 kHz, 16 bit, mono for desktop application, 8kHz, 16bit, mono for telephone applications. Audio files shouldn't be very long and shouldn't be very short. Optimal length is not less than 5 seconds and not more than 30 seconds. The audio files should be in wav directory under the corresponding speaker directory.

3.4 Training

To start the training we have to change to the database directory and run the following command:

```
$sphinxtrain -t poject setup
```

This command will setup the database and create two file: sphinx_train.cfg and feat.params. We need to configure these two files. Then we will use the following command to start the training procedure:

```
$sphinxtrain run
```

Then it will go through all the required stages. It will take a few minutes to train. On large databases, training could take a month.

During the stages, the most important stage is the first one which checks that everything is configured correctly and input data is consistent. Errors reported on the first 00.verify_all step must be solved.

The next two subsections will give a detail overview of configuring these two files:

3.4.1 Configuring sphinx_train.cfg

First we need to configure model type and model parameters. To do this we need to find the following lines in the sphinx_train.cfg:

```
$CFG_HMM_TYPE = '.cont.'; # Sphinx 4, Pocketsphinx

#$CFG_HMM_TYPE  = '.semi.'; # PocketSphinx, Sphinix 3

#$CFG_HMM_TYPE  = '.ptm.'; # PocketSphinx (larger data sets)
```

As we are using PocketSphinx we need to uncomment the line .ptm.

Next we need to configure this line:

$CFG_N_TIED_STATES = 1000;

For our system the value of the tied states is much less than the given value because we have learnt that senons depends on database [9]. This value should be adjusted for better accuracy.

3.4.2 Configuring feat.params file

The default for sound files used in Sphinx is a rate of 16 thousand samples per second (16KHz). If this is the case, the etc/feat.params file will be automatically generated with the recommended values.

If we are using sound files with a sampling rate of 8KHz (telephone quality), we need to change some values in etc/feat.params. The lower sampling rate also means a change in the sound frequency ranges used and the number of filters used to recognize speech. Recommended values are :

```
-samprate 8000.0

-nfilt 31

-lowerf 200.00

-upperf 3500.00

-dither yes
```

3.5 Decoding

For sphinxtrain snapshot decoding is a part of the training process. When the recognition job is complete, the script computes the recognition Word Error Rate (WER) and Sentence Error Rate (SER). The lower those rates are the better for the system. For typical 10-hours task WER should be around 10%. For a large task, it could be like 30%. We can find exact details of the decoding, like alignment with reference transcription, speed and result for each file, in result folder which will be created after decoding.

Chapter 4: Experiment Analysis

4.1 Evaluation

In this thesis work we have done several experiment varying different parameters. We have experimented with single user and multiple user voice sample. Also we have recorded the samples in 16 KHz and 8 KHz. We also have trained the system with noisy environment and quiet environment voice data. We also trained the system with different parameters.

We have tried to build a robust model for multiple users. So, we have collected data from 82 individual speakers. The result of that experiment is given in the Table 1:

Exp No. (multiple user)	Train				Test					
	Total Sentence	Total Word	Total Phones	Number of Training Sample	Total Word	Total Word Correct	Total Word Error	Total Correct (percent)	Total Error (percent)	Accuracy (percent)
1	20	47	31	82	79	23	71	29.11%	89.87%	10.13%

Table 1: Evaluation of multiple user experiment

In this experiment sample rate was 16 KHz, 16 bit PCM, mono channel. The language model was built on 20 sentences, 47 words and 31 phones. And total training hour is 2.34. This is a very small training data set for a robust model much more training data is needed. And also the sample data variation is much as there are 82 speakers and each user gives the sample only once. Hence, the accuracy rate as we can see is only 10.13%. Only few words are recognized.

After the previous experiment, we realized that we need lot of data for a robust system. Hence, we have tried to build a model for single user. We have collected 100 training samples for a single user. Among them 50 samples are taken in quiet environment and 50 samples are taken noisy environment. And as we are trying to build a model for mobile devices we have taken the samples of sample rate 8 KHz, 16 bit PCM and mono channel. We have done the experiments with two language model of 5 sentences and 7sentences. In the table total sentences, words and phone numbers are given with their testing result.

Experiment 1, 2 and 3 are done with 5 sentences. Number of words are 14 and number of phones are 26.

In experiment 1, total training samples are 100. And we tested the samples with 1800 words and the number total of correctly recognized words is 621. And the accuracy is 30.33%.

Exp. No. (single user)	Train				Test					
	Total Sentences	Total Words	Total Phones	Number of Training Sample	Total Word	Total Word Correct	Total Word Error	Total Correct (percent)	Total Error (percent)	Accuracy (percent)
1	5	14	26	100	1800	621	1254	34.50%	69.67%	30.33%
2	5	14	26	80	360	117	262	32.50%	72.78%	27.22%
3	5	14	26	80	360	116	253	32.22%	70.28%	29.72%
4	7	18	27	50	72	36	43	50%	59.72%	40.28%
5	7	18	27	50	72	53	37	73.61%	51.39%	48.61%
6	7	18	27	50	72	38	42	52.78%	58.33%	41.67%
7	7	18	27	50	1200	741	486	61.75%	40.50%	59.50%

Table 2: Evaluation of single user experiment

In experiment 2 and 3 we have trained the model with 80 samples and tested the model with 20 samples that were not in the trained samples. And the accuracy is quite similar.

In experiment 4, 5, 6 and 7, we have used only quiet environment samples to train the model. In experiment 4, 5, 6 we tested the sample with 72 words that were not used in training and the accuracy rate is 40.28%, 48.61% and 41.67%.

In experiment 7, we tested the model with more samples both trained and untrained data and the accuracy rate found is 59.50%.

In all the experiments, we have manually adjusted different parameters like tied states. Tied states should be tuned for better performance.

4.2 Problems faced

1. Installing frameworks in Ubuntu

We have installed the necessary frameworks after studying the CMUSphinx documentation. We have written the detailed installation procedure in our blog: http://banglaspeechrecognition.blogspot.com

2. Collecting speech data is the major problem we faced because volunteers were not available when you needed them and the environment was not stationary. We thought about taking the voice sample using the Android phone as our goal is to implement the system in Android, but the Android speech file format isn't supported by the framework we are using. So we build an Android application which can record voice in our expected format. But we could not manage enough time to gather voice samples using this Android recorder application. Another major issue is utterance variation and improper utterance of the speakers.

3. Adjusting the feature parameters according to your provided voice sample is another challenge.

4. This pattern recognition approach needs extensive amount of training data. For example, for a dictionary of 20 words at least 5 hours of training data is needed. Hence, collecting huge amount of data is a time lengthy process.

Chapter 5: Android Implementation

5.1 Overview of the Android implementation

We are starting the implementation procedure discussion with the overview model of our application. We are storing the acoustic model in the memory card of the Android device. The Android program is taking the speech as input and with the help of decoder it gets the hypothesis. Decoder returns the hypothesis using the acoustic model we have in memory card. The Android program has the decision logics to decide which command to execute.

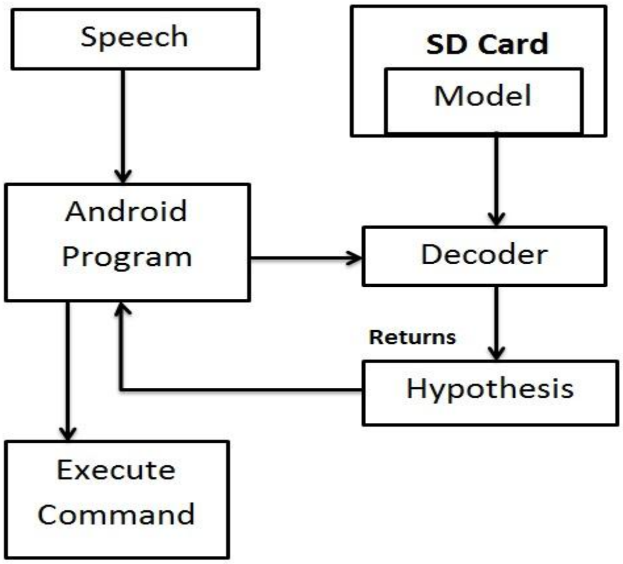


Figure 8: Android implementation overview

5.2 Building pocketSphinx on Android

To recognize speech on Android we need to have a decoder in Android. Sphinx developers have provided a demo project. In this demo project there is a decoder written in java that we used previously to test in our pc as pocketsphinx decoder. We downloaded this demo project from http://www.zachrattner.com/PocketSphinxDemo.tar.gz and we followed their instruction to run the demo project. This section is the summary of the procedure:

Open up the terminal and type to get root access:

```
$ sudo -i
```

We will need swig so we can install using the command:

```
$ apt-get install swig
```

or

```
$ yum install swig
```

Now cd into PocketSphinxDemo/jni folder and Open the Android.mk file, found in the jni folder, and change the SPHINX_PATH(line #5) to the parent folder holding pocketsphinx and sphinxbase.

From command line we have to type the following :

```
the-path-to-our-ndk-folder/ndk-build -B
```

5.2.1 Project set up in Eclipse

Now we have to import the PocketSphinxDemo in Eclipse. The following steps have to be taken:

- In the Navigator View look for PocketSphinxDemo project. Right click on it and select properties. The properties screen will pop up and we have to select Builders. In the Builders screen we will see SWIG and NDK build. Click on NDK build and edit.
- Click on NDK build and edit. In the edit screen change the field Location to point to ndk-folder we have on your machine. Click on the Refresh tab and select "The project containing the selected resource"
- Click on the Build Options tab and deselect "Specify working set of relevant resources"
- Apply changes and exit the configuration for NDK build.
- Click on SWIG and edit.
- In the refresh tab select "The folder containing the selected resource"
- In the Build Options tab deselect "Specifiy working set of relevant resources"
- Apply changes and exit the configuration for SWIG.

5.2.2 Import the model in Android device

1. Connect Android phone and create the edu.cmu.pocketsphinx folder at

```
/mnt/sdcard
```

One can do this by opening terminal command and typing:

```
$ adb shell
```

2. In shell type:

```
$ mkdir /mnt/sdcard/edu.cmu.pocketsphinx
```

3. Now cd into the edu.cmu.pocketsphinx folder that is located on phone and create the following folder structure like Figure 9:

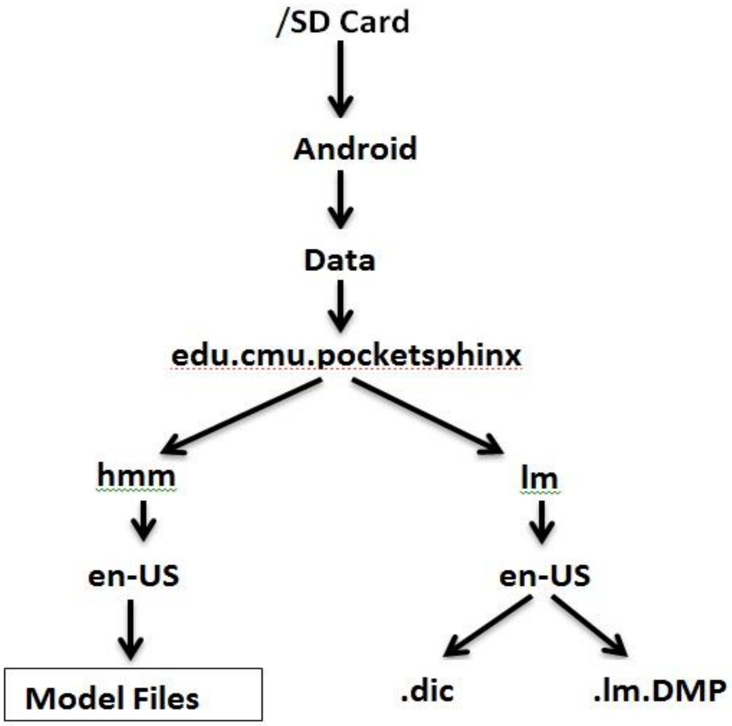


Figure 9: Directory structure to import the model in memory card [11]

4. Now type quit to leave adb shell.
5. While still in terminal we will need to push files from your computer onto the phone.

cd into pocketsphinx/model/hmm/en_US/ and type:

```
$ adb push ./hub4wsj_sc_8k
/mnt/sdcard/edu.cmu.pocketsphinx/hmm/en_US/hub4wsj_sc_8k
```

Now cd into pocketsphinx/model/lm/ in your and type:

```
$ adb push ./en_US /mnt/sdcard/edu.cmu.pocketsphinx/lm/en_US
```

6. Now build and run the project.

5.2.3 Emulator set up

In order to run the pocketsphinx demo in Android emulator we need to create a sdcard image. To do this:

1. cd into the Android sdk installation directory and to the 'tools' subdirectory from the terminal.

2. Now using mksdcard script we can create a sdcard image.

```
$ ./mksdcard 128M myCard
```

3. The image can now be mounted as a loopback device. First we have to create a folder called 'mycard' inside the "/media" directory. Then using the following command :

```
$ sudo mount -o loop <Card_name> \media\mycard
```

4. Once mounted the card can be accessed as a device.

5. Now we have to create a directory structure "./Android/data/edu.cmu.pocketsphinx" has to be created. Now the files from the 'models' folder has to be copied to Android/data/edu.cmu.pocketsphinx.

6. Then open location '/hmm/en_US/hub...' and replace the models with the ones that we have generated.

7. Now from our projects etc folder we have to Copy .dic file and .lm.DMP files to

Android/data/edu.cmu.pocketsphinx/lm/en_US/ .

8. To successfully run the pocketsphinx demo we have to edit the dicrectory given in RecognizerTask.java file. There we will find :

```
c.setString("-dict",
"/mnt/sdcard/edu.cmu.pocketsphinx/lm/en_US/hub4.5000.dic");
```

```
c.setString("-lm",
"/mnt/sdcard/edu.cmu.pocketsphinx/lm/en_US/hub4.5000.DMP");
```

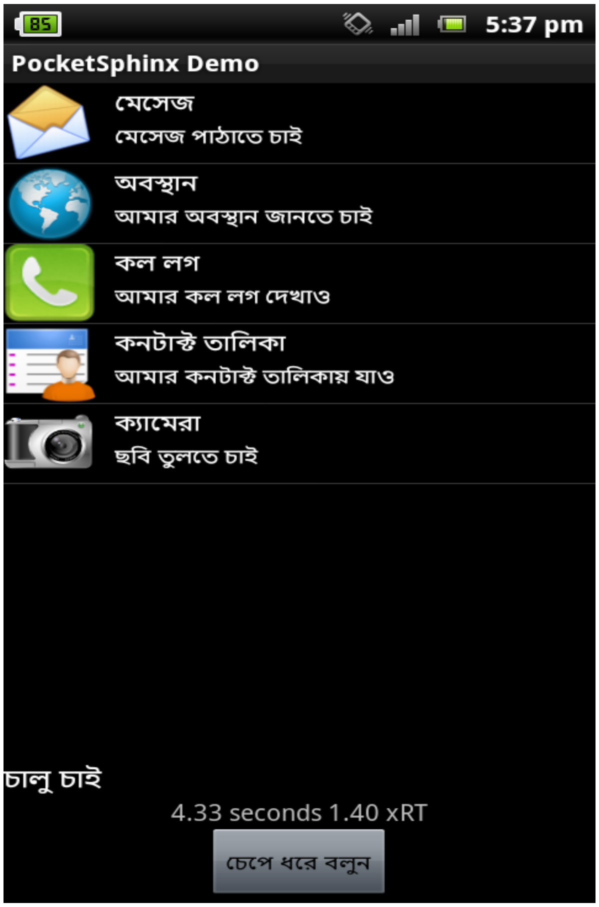
we have to adjust the name of dictionary(.dic) file and LanguageModel (.DMP) file according to our copied files.

9. Now we have to copy the 'Android' folder and its contents to the /media/mycard folder and unmount.

10. Create an Android virtual device using AVD manager in eclipse. Select an API level to 8 or above, in the place of memory card browse to the place where the memory card image was created (Android_sdk_installation_folder/tools). Add audio record and playback support and other support necessary for application and save.

5.3 Our Android Application

As our intension was to build a Bangla voice based mobile assistant hence while we were creating a dependable acoustic model for Bangla we also continued our study on Android programming. For our purpose we selected five important options that are necessary in everyday mobile using. In our application we have given options like message, contact list, call log, location support and camera. And the application is built in such a way that in future we can add other options as necessary. The application has capability of taking input both in voice and touch. A snapshot of our application is given in the Figure 10.



PocketSphinx Demo

মেসেজ
মেসেজ পাঠাতে চাই

অবস্থান
আমার অবস্থান জানতে চাই

কল লগ
আমার কল লগ দেখাও

কনটাক্ট তালিকা
আমার কনটাক্ট তালিকায় যাও

ক্যামেরা
ছবি তুলতে চাই

চালু চাই

4.33 seconds 1.40 xRT

Figure 10: Snapshot of speech recognizer application

5.4 Evaluation of the model in the Android device

Though the models give up to 59.50% accuracy, after importing the models on Android and testing it with our application we have found inconsistency in recognition. Though sometimes one or two words are recognized but total recognition is not satisfactory. After researching what may be the cause of such poor accuracy we have pointed out few reasons:

1. One big factor is environment. In our experiments training samples are taken in quiet environment. But for good recognition we have to record voice in real time environment.

2. The hardware device that we used to record the samples in computer is quite different from mobile device. The solution is to take the samples with mobile device.

After finding out the problem we have tried to record samples with Android mobile. But unfortunately we have found that Android devices don’t support recording in wav format. Hence, we have tried to find a solution and after studying we have written a program that can record in wav format. And we have taken few samples using that application but we couldn’t able to collect enough samples for training. But we have found that in the same environment the samples that are recorded using Android are noisier than our previous samples. This finding explains poor recognition in mobile device.

Chapter 6: Future Work

As from the discussion from the previous two chapters we have seen that our acoustic models are showing better accuracy rate for single user. But unfortunately the models are not giving good results in the Android devices which we have shown in the chapter 5. We have found out few reasons behind that too. From our experience through the thesis work we are proposing a 3 step process of the future continuation of work.

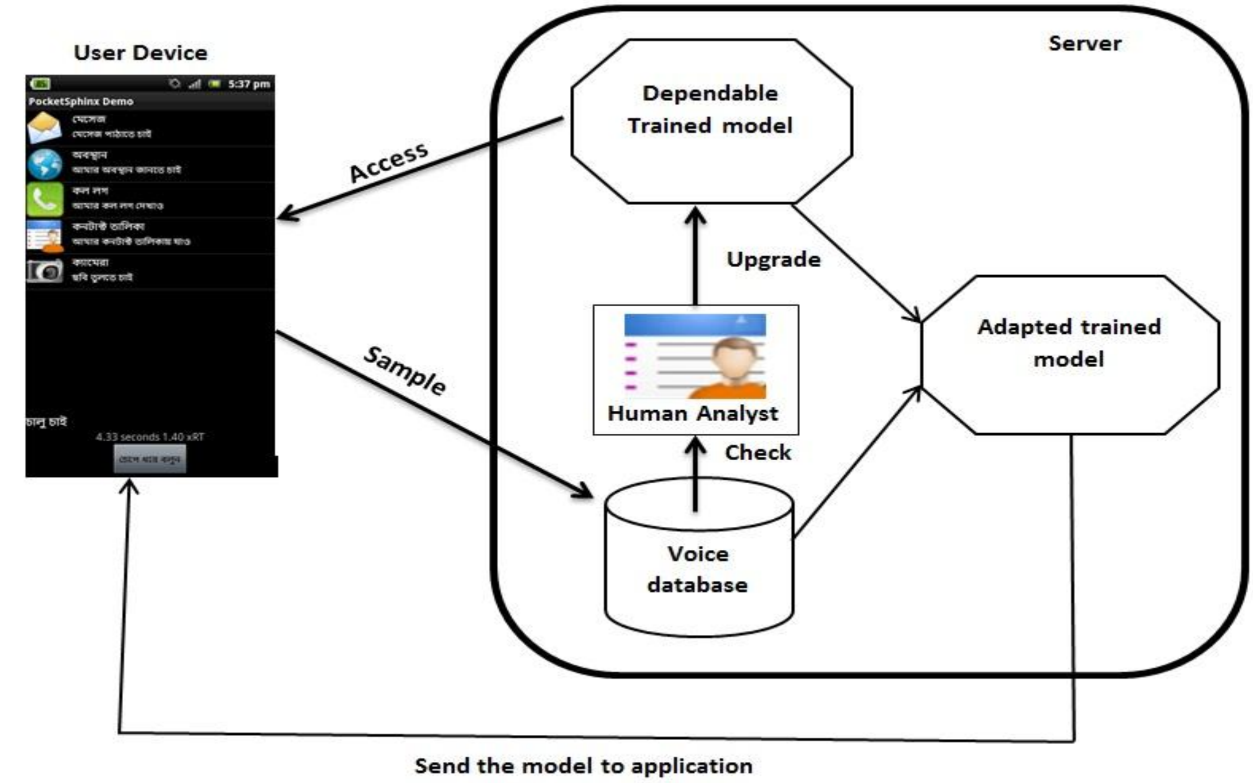
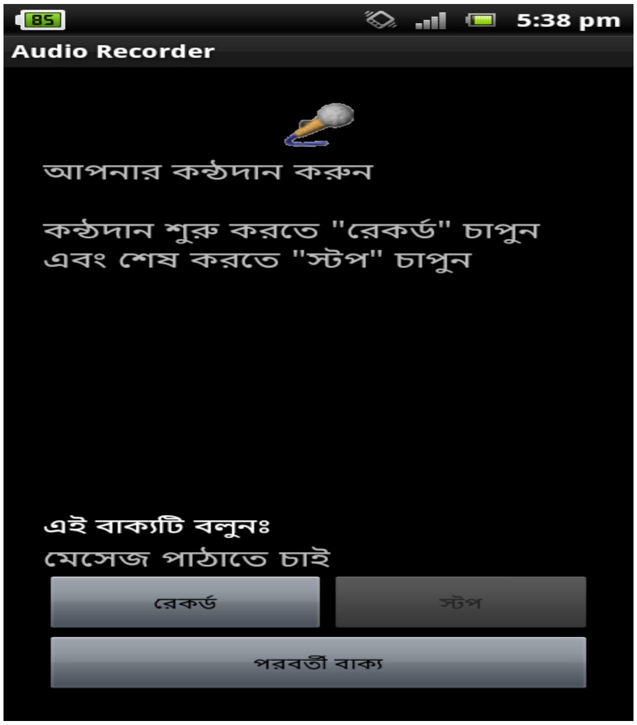


Figure 11: Proposed future system

- 1^{st} step: Build up a dependable trained model
  - Dictionary with pronunciation phoneme built by the Bangla linguistic experts
  - Well-equipped recording studio
  - Record samples from professionally trained speakers
  - Gather huge set of qualitative voice samples
  - This trained model will be stored in the server and user application access the model for decoding using the internet

- 2nd step: Make the model user specific
  - Take sample from user
  - Adapt the training model using user's sample
  - Send the model to the user device
- 3rd step: Upgrade the dependable trained model
  - Check user's sample by human analyst and upgrade the dependable trained model gradually.
- The Figure 11 depicts our proposed model of the future work. For the starting of our future work we have developed an Android application to record the voice of the user. As we don't have the server facility at present we are storing the voice samples in the memory card of the Android device. This application can be used for voice data collection. People can easily install the application to their Android mobile device and give voice sample. This sample will be sent to the server and after filtering the good samples it will be used for train the model. Thus, we can collect huge amount of voice samples that is very important for training the system. The snapshot of the recording program is given in Figure12:



আপনার কন্ঠদান করুন

কন্ঠদান শুরু করতে "রেকর্ড" চাপুন
এবং শেষ করতে "স্টপ" চাপুন

এই বাক্যটি বলুনঃ
মেসেজ পাঠাতে চাই

Figure 12: Snapshot of the Recording application

References

1. http://banglaspeechrecognition.blogspot.com, accessed on 16 April 2012.

2. Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, ComputationalLinguistics, and Speech Recognition.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

3. Lawrence Rabiner and Biing H. Juang. Fundamentals of Speech Recognition. Prentice Hall, United States Edition, April 1993.

4. P. Foster, T. Schalk. *Speech Recognition : The Complete Practical Reference Guide*. 1993.

5. *Implementation of Speech Recognition System in Bangla* - Thesis Paper by Shammur Absar Chowdhury, CRBLP, http://crblp.bracu.ac.bd/thesis_paper/2010/ASR_Thesis_Report_Shammur.pdf

6. Md. Abul Hasnat, Jabir Mowla, Mumit Khan, "*Isolated and Continuous Bangla Speech Recognition: Implementation, Performance and application perspective*", http://univ-st-etienne.academia.edu/MdAbulHasnat/Papers/681988/Isolated_and_continuous_bangla_speech_recognition_implementation_performance_and_application_perspective, accessed on 1 October 2012.

7. *MIT's Spoken Language Systems Homepage* http://groups.csail . mit.edu/sls//sls-blue-noflash.shtml, accessed on 5 April 2012.

8. *CMU – Robust Group Tutorial* http://www.speech.cs.cmu.edu/sphinx/tutorial.html, accessed on 14 April 2012.

9. *CMU Sphinx – wiki*http://cmusphinx.sourceforge.net/wiki/, accessed on 11 April 2012.

10. *Acoustic Model Creation using SphinxTrain* http://forum.visionopen.com/viewtopic.php?f=39&t=1130 accessed on 12 April 2012.

11. *PocketSphinx-Android Demo* https://github.com/ciac/cmusphinx/tree/trunk/PocketSphinxAndroidDemo accessed on 4 April 2012.

12. *The CMU-Cambridge Statistical Language Modeling Toolkit* http://svr-www.eng.cam.ac.uk/~prc14/toolkit_documentation.html accessed on 10 October 2012.

13. Tamanna Haque Nipa, Muhammad Harun-Owr-Roshid, Mohammed Zahirul Hoq Sarker, Nasrin Akhter, *"MORPHOLOGICAL ANALYSIS OF BANGLA PARTS OF SPEECH FOR MACHINE TRANSLATION SYSTEM"*, www.ijest.info/docs/IJEST11-03-02-179.pdf, accessed on 1 October 2012.

Appendix

Experiment (Multiple user) 1

Corpus:

<s>আমাকে কলের তালিকা দেখাও</s>

<s>অনুগ্রহ করে ইনবক্সে যাও</s>

<s>আমাকে মিসড়্ কলগুলো দেখাও</s>

<s>আমাকে কল লগ দেখাও</s>

<s>আমার মিউজিক প্লেয়ার চালু কর</s>

<s>আমি একটি বার্তা পাঠাতে চাই</s>

<s>আমার অবস্থান জানতে চাই</s>

<s>আজকের আবহাওয়া জানতে চাই</s>

<s>আমার ইন্টারনেট ব্রাউজার খুলো</s>

<s>আমার ব্লুটুথ চালু করি</s>

<s>গান বন্ধ কর</s>

<s>কলটিধরো</s>

<s>আমার গেমিং অপশনে যাও</s>

<s>আমার ছবির গ্যালারিতে যাও</s>

<s>আমাকে গানের তালিকা দেখাও</s>

<s>আমার ফেসবুক খুলো</s>

<s>আমার ক্যামেরা চালু কর</s>

<s>আমার রেডিও চালু কর</s>

<s>এনড্রয়েড বাজারে যাও</s>

<s>আমার ফোন সেটিংস এ যাও</s>

Dictionary:

আমাকে a m a k e

তালিকা <w> a l i k a

অনুগ্রহ o n u g g r o h o

ইনবক্সে i n oo b o k ss e

মিসড্ m i s d

গুলো g u l o

আমার a m a r

প্লেয়ার p l e a r

আমি a m I

বার্তা b a r <w> a

চাই c a I

জানতে jj a n <w> e

আবহাওয়া a b oo h a o a

ব্রাউজার b r a u jj a r

ব্লুটুথ b l u t u <h>

গান g a n

কলটি k o l o t I

গেমিং g aa m i nn

ছবির ch o b i r

গানের g a n e r

ক্যামেরা k aa m e r a

এনড্রয়েড e n oo d d r o j e d o

ফোন f o n

এ e

কলের k oo l e r

দেখাও d aa kh a o

করে k o r e

যাও jj a o

কল k oo l

লগ l oo g

মিউজিক m i u jj i k

কর k oo r

একটি e k t i

পাঠাতে p a <h> a <w> e

অবস্থান oo b o s <h> a n

আজকের a jj k e r

ইন্টারনেট i n t o a r n e t

খুলো kh u l o

চালু c a l u

বন্ধ b oo n dh o

ধরো dh o r o

অপশনে oo p o ss oo n e

গ্যালারিতে g aa l a r i t e

ফেসবুক f e ss o b u k

রেডিও r e d i o

বাজারে b a jj a r e

সেটিংস ss e t i nn ss o

Result:

```
***    আমাকে কলের তালিকা দেখাও  (SP_64-64_1_A)
দেখাও আমাকে কলের ***     ***     (SP_64-64_1_A)
Words: 4 Correct: 2 Errors: 3 Percent correct = 50.00% Error = 75.00%
Accuracy = 25.00%
Insertions: 1 Deletions: 2 Substitutions: 0
অনুগ্রহ করে ইনবক্সে যাও  (SP_64-64_2_A)
***     দেখাও কর চাই  (SP_64-64_2_A)
Words: 4 Correct: 0 Errors: 4 Percent correct = 0.00% Error = 100.00%
Accuracy = 0.00%
Insertions: 0 Deletions: 1 Substitutions: 3
***  আমাকে মিসড় কলগুলো দেখাও  (SP_64-64_3_A)
খুলো আমাকে মিসড় *** ***  দেখাও  (SP_64-64_3_A)
Words: 5 Correct: 3 Errors: 3 Percent correct = 60.00% Error = 60.00%
Accuracy = 40.00%
Insertions: 1 Deletions: 2 Substitutions: 0
***  আমাকে কল লগ দেখাও  (SP_64-64_4_A)
খুলো আমাকে কলের দেখাও খুলো   (SP_64-64_4_A)
Words: 4 Correct: 1 Errors: 4 Percent correct = 25.00% Error = 100.00%
Accuracy = 0.00%
Insertions: 1 Deletions: 0 Substitutions: 3
আমার মিউজিক প্লেয়ার চালুকর           (SP_64-64_5_A)
***  ***     খুলো আমার ইন্টারনেট  (SP_64-64_5_A)
Words: 5 Correct: 0 Errors: 5 Percent correct = 0.00% Error = 100.00%
Accuracy = 0.00%
Insertions: 0 Deletions: 2 Substitutions: 3
আমি একটি বার্তা পাঠাতে চাই          (SP_64-64_6_A)
*** ***  দেখাও আমার ইন্টারনেট  (SP_64-64_6_A)
Words: 5 Correct: 0 Errors: 5 Percent correct = 0.00% Error = 100.00%
Accuracy = 0.00%
Insertions: 0 Deletions: 2 Substitutions: 3
আমার অবস্থান জানতে চাই  (SP_64-64_7_A)
***  ***      করচাই  (SP_64-64_7_A)
Words: 4 Correct: 1 Errors: 3 Percent correct = 25.00% Error = 75.00%
Accuracy = 25.00%
Insertions: 0 Deletions: 2 Substitutions: 1
আজকের আবহাওয়া জানতে চাই       (SP_64-64_8_A)
খুলো আমাকে আমার অবস্থান  (SP_64-64_8_A)
Words: 4 Correct: 0 Errors: 4 Percent correct = 0.00% Error = 100.00%
Accuracy = 0.00%
Insertions: 0 Deletions: 0 Substitutions: 4
***    আমার ইন্টারনেট ব্রাউজার খুলো  (SP_64-64_9_A)
দেখাও আমার ইন্টারনেট ***       ***    (SP_64-64_9_A)
Words: 4 Correct: 2 Errors: 3 Percent correct = 50.00% Error = 75.00%
Accuracy = 25.00%
Insertions: 1 Deletions: 2 Substitutions: 0
আমার ব্লুটুথ চালু্ কর     (SP_64-64_10_A)
দেখাও আমার ইন্টারনেট খুলো  (SP_64-64_10_A)
```

```
Words: 4 Correct: 0 Errors: 4 Percent correct = 0.00% Error = 100.00%
Accuracy = 0.00%
Insertions: 0 Deletions: 0 Substitutions: 4
গান বন্ধ কর  ***  (SP_64-64_11_A)
দেখাও কর কর কল   (SP_64-64_11_A)
Words: 3 Correct: 1 Errors: 3 Percent correct = 33.33% Error = 100.00%
Accuracy = 0.00%
Insertions: 1 Deletions: 0 Substitutions: 2
***   ***   কলটি ধরো  (SP_64-64_12_A)
দেখাও দেখাও কল কল   (SP_64-64_12_A)
Words: 2 Correct: 0 Errors: 4 Percent correct = 0.00% Error = 200.00%
Accuracy = -100.00%
Insertions: 2 Deletions: 0 Substitutions: 2
***   আমার গেমিং অপশনে যাও    (SP_64-64_13_A)
দেখাও আমার গেমিং ***   দেখাও  (SP_64-64_13_A)
Words: 4 Correct: 2 Errors: 3 Percent correct = 50.00% Error = 75.00%
Accuracy = 25.00%
Insertions: 1 Deletions: 1 Substitutions: 1
আমার ছবির গ্যালারিতে যাও    (SP_64-64_14_A)
***  ***  দেখাও আমাকে  (SP_64-64_14_A)
Words: 4 Correct: 0 Errors: 4 Percent correct = 0.00% Error = 100.00%
Accuracy = 0.00%
Insertions: 0 Deletions: 2 Substitutions: 2
***   আমাকে গানের তালিকা দেখাও  (SP_64-64_15_A)
দেখাও আমাকে গানের ***    ***    (SP_64-64_15_A)
Words: 4 Correct: 2 Errors: 3 Percent correct = 50.00% Error = 75.00%
Accuracy = 25.00%
Insertions: 1 Deletions: 2 Substitutions: 0
***   ***   আমার ফেসবুক ***   খুলো  (SP_64-64_16_A)
দেখাও দেখাও আমার ফেসবুক দেখাও খুলো  (SP_64-64_16_A)
Words: 3 Correct: 3 Errors: 3 Percent correct = 100.00% Error = 100.00%
Accuracy = 0.00%
Insertions: 3 Deletions: 0 Substitutions: 0
***   আমার ক্যামেরা চালু কর    (SP_64-64_17_A)
দেখাও আমার ক্যামেরা দেখাও খুলো  (SP_64-64_17_A)
Words: 4 Correct: 2 Errors: 3 Percent correct = 50.00% Error = 75.00%
Accuracy = 25.00%
Insertions: 1 Deletions: 0 Substitutions: 2
***   আমার রেডিও চালু কর     (SP_64-64_18_A)
দেখাও আমার রেডিও ***  দেখাও  (SP_64-64_18_A)
Words: 4 Correct: 2 Errors: 3 Percent correct = 50.00% Error = 75.00%
Accuracy = 25.00%
Insertions: 1 Deletions: 1 Substitutions: 1
এনড্রয়েড বাজারে যাও  (SP_64-64_19_A)
***      দেখাও কর   (SP_64-64_19_A)
Words: 3 Correct: 0 Errors: 3 Percent correct = 0.00% Error = 100.00%
Accuracy = 0.00%
Insertions: 0 Deletions: 1 Substitutions: 2
***   আমার ফোন সেটিংস এ যাও  (SP_64-64_20_A)
```

```
দেখাও আমার ফোন ***     *** ***  (SP_64-64_20_A)
Words: 5 Correct: 2 Errors: 4 Percent correct = 40.00% Error = 80.00%
Accuracy = 20.00%
Insertions: 1 Deletions: 3 Substitutions: 0
TOTAL Words: 79 Correct: 23 Errors: 71
TOTAL Percent correct = 29.11% Error = 89.87% Accuracy = 10.13%
TOTAL Insertions: 15 Deletions: 23 Substitutions: 33
```

Experiment (Single user) 5

Corpus:

```
<s>আমার কনটাক্ট তালিকায় যাও</s>
<s>ব্লুটুথ চালু কর</s>
<s>ব্লুটুথ বন্ধ কর</s>
<s>মেসেজ পাঠাতে চাই</s>
<s>ছবি তুলতে চাই</s>
<s>আমার অবস্থান জানতে চাই</s>
<s>আমার কল লগ দেখাও</s>
```

Dictionary:

আমার	amar
কনটাক্ট	kontakt
তালিকায়	<w>alikay
যাও	jao
ব্লুটুথ	blutu<h>
চালু	calu
কর	kor
বন্ধ	bondho
মেসেজ	mesej
পাঠাতে	pa<h>a<w>e
চাই	cai
ছবি	chobi
তুলতে	<w>ul<w>e
অবস্থান	obos<h>an
জানতে	jan<w>e
কল	kol
লগ	log
দেখাও	dakhao

Result:

```
আমার কনটাক্ট তালিকায় যাও  (SP_51-51_1)
আমার কনটাক্ট তালিকায় কর   (SP_51-51_1)
Words: 4 Correct: 3 Errors: 1 Percent correct = 75.00% Error =
25.00% Accuracy = 75.00%
Insertions: 0 Deletions: 0 Substitutions: 1
ব্লুটুথ চালু কর   (SP_51-51_2)
ব্লুটুথ চালু ***  (SP_51-51_2)
Words: 3 Correct: 2 Errors: 1 Percent correct = 66.67% Error =
33.33% Accuracy = 66.67%
Insertions: 0 Deletions: 1 Substitutions: 0
ব্লুটুথ বন্ধ কর  ***  (SP_51-51_3)
ব্লুটুথ বন্ধ কর চাই  (SP_51-51_3)
Words: 3 Correct: 3 Errors: 1 Percent correct = 100.00% Error =
33.33% Accuracy = 66.67%
Insertions: 1 Deletions: 0 Substitutions: 0
***     মেসেজ *** পাঠাতে *** চাই  (SP_51-51_4)
ব্লুটুথ মেসেজ ছবি পাঠাতে কর চাই  (SP_51-51_4)
Words: 3 Correct: 3 Errors: 3 Percent correct = 100.00% Error =
100.00% Accuracy = 0.00%
Insertions: 3 Deletions: 0 Substitutions: 0
ছবি তুলতে চাই     (SP_51-51_5)
চাই ছবি পাঠাতে  (SP_51-51_5)
Words: 3 Correct: 0 Errors: 3 Percent correct = 0.00% Error =
100.00% Accuracy = 0.00%
Insertions: 0 Deletions: 0 Substitutions: 3
আমার অবস্থান জানতে *** চাই  (SP_51-51_6)
আমার অবস্থান জানতে চাই চাই  (SP_51-51_6)
Words: 4 Correct: 4 Errors: 1 Percent correct = 100.00% Error =
25.00% Accuracy = 75.00%
Insertions: 1 Deletions: 0 Substitutions: 0
আমার কল লগ দেখাও  (SP_51-51_7)
আমার কল বন্ধ চাই    (SP_51-51_7)
Words: 4 Correct: 2 Errors: 2 Percent correct = 50.00% Error =
50.00% Accuracy = 50.00%
Insertions: 0 Deletions: 0 Substitutions: 2
আমার কনটাক্ট তালিকায় যাও  (SP_52-52_1)
আমার কনটাক্ট তালিকায় ***  (SP_52-52_1)
Words: 4 Correct: 3 Errors: 1 Percent correct = 75.00% Error =
25.00% Accuracy = 75.00%
Insertions: 0 Deletions: 1 Substitutions: 0
ব্লুটুথ চালু কর   (SP_52-52_2)
```

```
ব্লুটুথ চালু চাই  (SP_52-52_2)
Words: 3 Correct: 2 Errors: 1 Percent correct = 66.67% Error =
33.33% Accuracy = 66.67%
Insertions: 0 Deletions: 0 Substitutions: 1
ব্লুটুথ বন্ধ কর   (SP_52-52_3)
ব্লুটুথ বন্ধ ***  (SP_52-52_3)
Words: 3 Correct: 2 Errors: 1 Percent correct = 66.67% Error =
33.33% Accuracy = 66.67%
Insertions: 0 Deletions: 1 Substitutions: 0
মেসেজ পাঠাতে চাই    (SP_52-52_4)
দেখাও মেসেজ তুলতে  (SP_52-52_4)
Words: 3 Correct: 0 Errors: 3 Percent correct = 0.00% Error =
100.00% Accuracy = 0.00%
Insertions: 0 Deletions: 0 Substitutions: 3
*** ছবি ***    তুলতে চাই  (SP_52-52_5)
চাই ছবি পাঠাতে কর চাই  (SP_52-52_5)
Words: 3 Correct: 2 Errors: 3 Percent correct = 66.67% Error =
100.00% Accuracy = 0.00%
Insertions: 2 Deletions: 0 Substitutions: 1
আমার অবস্থান জানতে চাই     (SP_52-52_6)
আমার অবস্থান ***   পাঠাতে  (SP_52-52_6)
Words: 4 Correct: 2 Errors: 2 Percent correct = 50.00% Error =
50.00% Accuracy = 50.00%
Insertions: 0 Deletions: 1 Substitutions: 1
আমার কল লগ  *** দেখাও  (SP_52-52_7)
আমার কললগ যাও চাই    (SP_52-52_7)
Words: 4 Correct: 3 Errors: 2 Percent correct = 75.00% Error =
50.00% Accuracy = 50.00%
Insertions: 1 Deletions: 0 Substitutions: 1
আমার কনটাক্ট তালিকায় যাও  (SP_11-11_1)
আমার কনটাক্ট তালিকায় যাও  (SP_11-11_1)
Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error =
0.00% Accuracy = 100.00%
Insertions: 0 Deletions: 0 Substitutions: 0
ব্লুটুথ চালু *** কর   (SP_11-11_2)
ব্লুটুথ চালু কর কর   (SP_11-11_2)
Words: 3 Correct: 3 Errors: 1 Percent correct = 100.00% Error =
33.33% Accuracy = 66.67%
Insertions: 1 Deletions: 0 Substitutions: 0
*** ব্লুটুথ বন্ধ কর   (SP_11-11_3)
কর ব্লুটুথ লগ কর   (SP_11-11_3)
Words: 3 Correct: 2 Errors: 2 Percent correct = 66.67% Error =
66.67% Accuracy = 33.33%
```

```
Insertions: 1 Deletions: 0 Substitutions: 1
*** মেসেজ *** পাঠাতে ***   চাই  (SP_11-11_4)
চাই মেসেজ ছবি পাঠাতে দেখাও চাই  (SP_11-11_4)
Words: 3 Correct: 3 Errors: 3 Percent correct = 100.00% Error =
100.00% Accuracy = 0.00%
Insertions: 3 Deletions: 0 Substitutions: 0
*** ছবি ***   *** *** তুলতেচাই  (SP_11-11_5)
চাই ছবি জানতে লগ চাই চাই চাই  (SP_11-11_5)
Words: 3 Correct: 2 Errors: 5 Percent correct = 66.67% Error =
166.67% Accuracy = -66.67%
Insertions: 4 Deletions: 0 Substitutions: 1
আমার অবস্থান জানতে *** চাই  (SP_11-11_6)
আমার অবস্থান জানতে চাই চাই  (SP_11-11_6)
Words: 4 Correct: 4 Errors: 1 Percent correct = 100.00% Error =
25.00% Accuracy = 75.00%
Insertions: 1 Deletions: 0 Substitutions: 0
আমার কল লগ দেখাও  (SP_11-11_7)
আমার কল লগ দেখাও  (SP_11-11_7)
Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error =
0.00% Accuracy = 100.00%
Insertions: 0 Deletions: 0 Substitutions: 0
TOTAL Words: 72 Correct: 53 Errors: 37
TOTAL Percent correct = 73.61% Error = 51.39% Accuracy = 48.61%
TOTAL Insertions: 18 Deletions: 4 Substitutions: 15
```

List of Figures

List of Tables