

Design and development of Doctor's dictation kit Using Raspberry PI



Inspiring Excellence

Kazi Injamamul Haque 13101103*

Ullash Saha 13101156*

Sudipto Biswas 13101159*

Md. Muhtasim Billah 13101167*

Abu Saleh Al Momin 13101220*

Department of Computer Science and Engineering

BRAC University

This dissertation is submitted for the degree of

Degree of Bachelor in Science

We would like to dedicate this thesis to our loving parents ...

Declaration

We hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of our own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains fewer than 15000 words including appendices, bibliography, footnotes, tables and equations and has less than 150 figures.

We hereby declare that this report is our own work and effort and that it has not been submitted anywhere for any award.

All the contents provided here is totally based on our own labor dedicated for the completion of the thesis. Where other sources of information have been used, they have been acknowledged and the sources of information have been provided in there reference section.

Kazi Injamamul Haque 13101103*

Ullash Saha 13101156*

Sudipto Biswas 13101159*

Md. Muhtasim Billah 13101167*

Abu Saleh Al Momin 13101220*

2016

Acknowledgements

We write this acknowledgment with great honor, pride and pleasure to pay our respects to all who enabled us either directly or indirectly in completing this thesis. We would like to show our gratitude to our supervisor Professor Dr. Mohammad Zahidur Rahman for being a constant source of inspiration, valuable guidance and constant encouragement to us especially for solving the problems that we have encountered while working on this thesis. And we also like to thank our Co-Supervisor Mr. Samiul Islam.

Abstract

Natural language processing and speech to text can make a significant improve in medical dictation (transcription, radiology report, prescription etc) in a developing country like Bangladesh. In the field of telemedicine it can play a very crucial part in the absence of qualified doctors and specialists to prescribe medicine and provide with medical support in remote and rural places. This paper is based on a real time speech detection with a standalone system to implement it in a single board computer Raspberry PI that can also work in crowded place. The recognition engine used for the system is JULIUS along with the toolkit HTK to manipulate HMM(Hidden Markov Model). The acoustic model is set to such a way that it can detect selected medicine names those are widely used in Bangladesh. The accuracy rate of our trained dictionary is 84% but a silent environment and longer string prodeces 94% accuraccy which can also be imroved with more accurate training with advanced directional microphone. The intention of implementing the system in Raspberry PI was to have a future innovation of a standalone device for medical dictation and telepharmacy.

Contents

Contents	xi
List of Figures	xv
List of Tables	xvii
1 Overview	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Problem definition	2
2 Literature review	5
2.1 Telemedicine in Bangladesh	5
2.2 JULIUS	6
2.2.1 N gram model	8
2.3 HTK	8
2.4 Hidden Markov Model (HMM)	9
2.5 Previous works on voice recognition with HMM and HTK	10
3 System overview	13
3.1 Use case	13
3.2 RASPBERRY PI	14
3.3 implementation of JULIUS	14
3.4 Installation of HTK	15
3.5 JULIA	15
4 Processing training data	17
4.1 Language model	17
4.1.1 Grammar file	17

4.1.2	Voca file	17
4.1.3	Prompts file	19
4.1.4	Dictionary file	19
4.2	Audio training	20
4.2.1	WAV files	20
4.2.2	MFCC	21
4.3	Configuration	21
4.3.1	JCONF File	21
4.3.2	Global option	22
4.3.2.1	Miscellaneous	22
4.3.3	Grammar	22
4.3.4	Audio input	23
4.3.4.1	-input {miclawfilelmfccfileladinnetlstdinlnetaudio}	23
4.3.4.2	Speech segment detection by level and zero-cross	23
4.3.4.3	acoustic HMM and parameters	24
4.3.4.4	Speech analysis parameters	25
4.3.5	Recognizer and search (-SR)	25
4.3.5.1	General parameters	25
4.3.5.2	1st pass parameters	25
4.3.5.3	2nd pass parameters	26
5	Tests and results	27
5.1	Table of different user who trained data	28
5.1.1	Trained Speaker:	28
5.1.1.1	Speaker 1 and Speaker 2:(MALE)	28
5.1.1.2	Speaker 3 and Speaker 4:(FEMALE)	30
5.1.2	Speakers who did not train data	32
5.1.2.1	Speaker 5 and Speaker 6:(MALE)	32
5.1.2.2	Speaker 7 and Speaker 8:(FEMALE)	34
5.2	Charts of various accuracy and analysis	36
5.2.1	Male vs. Female (Who trained data)	36
5.2.2	Male vs. Female (Who did not trained data)	38
5.2.3	Overall accuracy in silent and normal environment	39
5.2.4	Overall accuracy for longer strings in silent and normal environment	41
5.2.5	Male vs. female comparison in overall accuracy	42
5.2.6	Overall accuracy with complete dataset in different environment	43
5.2.7	Length vs. Accuracy	44

5.3	Result analysis	44
5.3.1	Microphone issue	44
5.3.2	Longer strings	44
5.3.3	Environment	45
6	Conclusion	47
6.1	Conclusion	47
6.2	Further work	47
	Refernces	49
A	Appendix	53
B	A sample output	55

List of Figures

2.1	System architecture of JULIUS	6
2.2	Overview of JULIUS	7
2.3	HTK architecture	9
2.4	Hidden Markov Model	9
3.1	Use case diagram of the system	13
3.2	JULIUS on the system	14
3.3	JULIA(installed in the system)	15
4.1	Grammar file	17
4.2	Voca file	18
4.3	Prompts file	19
4.4	Glimpse of the dictionary file	20
4.5	Audio wave sample	21
5.1	Male speakers(Who trained data)	36
5.2	Female speakers(Who trained data)	36
5.3	Male speakers(Who did not trained data)	38
5.4	Female speakers(Who did not trained data)	38
5.5	Overall accuracy (normal environment)	39
5.6	Overall accuracy (silent environment)	39
5.7	Overall accuracy for longer strings (normal environment)	41
5.8	Overall accuracy for longer strings (silent environment)	41
5.9	Male vs. female speaker accuracy(who trained data)	42
5.10	Male vs. female speaker accuracy(who did not train data)	42
5.11	Overall accuracy(silent vs. normal)	43
5.12	Length vs.accuracy	44
B.1	Sample Output	55

List of Tables

4.1	Audio files parameters	20
4.2	Audio conversion parameters	21
5.1	Male speakers who trained data(continued)	28
5.2	Male speakers who trained data(end)	29
5.3	Female speakers who trained data(continued)	30
5.4	Female speakers who trained data(end)	31
5.5	Male speakers who did not train data(continued)	32
5.6	Male speakers who did not train data(end)	33
5.7	Female speakers who did not train data(continued)	34
5.8	Female speakers who did not train data(end)	35
A.1	Appendix 1	53

Chapter 1

Overview

1.1 Introduction

Telepharmacy plays one of the pivotal roles in telemedicine and its services. In rural and remote places where medical treatment is not sufficient enough to serve dense population, telepharmacy can provide an arsenal of medical help and solution for the people. It's a form of healthcare where the patient does not have direct communication with a pharmacist but he can have pharmaceutical care. For this case speech recognition can play a major role for faster and efficient transcription. Up to this point of development it has been proved as an efficient way to provide medical services. Such as GrameenPhone, a leading telecom operator in Bangladesh introduced an emergency medical consultation service through dialing the number "789" from 4th November 2006.([6])

Application of CSR is not new for medical purpose; since the inception of telemedicine and its sub-sectors, scientists and developers continuously tried to materialize NLP in order to make it more time efficient. For t. Drastic change of implementing speech recognition came in the decade of 1990-2000. Research from that time showed usage NLP can be efficient, user friendly and also can reduce workload in medical environment([18]). Medical dictation can be also used in extracting data as much correctly as a human being(medical handbook). For that purpose it needs a knowledge base of medical terminology(such as medicine name, disease name, organ name) [15]). It is clear that instead of typing or writing medical prescription, a speech recognition based transcription can be useful not only for the doctor who are prescribing those but also for the patient who are at distance from the doctor. Furthermore if speech recognition is used in such a way so that the end device can transform the described medication from spoken words to text format, the error rate of prescribed medicine can be subsequently reduced. It will also improve telepharmacy by audio conferencing/video conferencing significantly.

1.2 Motivation

In Bangladesh, shortage of HCP is not an uncommon issue. Survey says that there are only 12 unqualified village doctors and 11 salespeople at drug retail outlets per 10 000 people in rural area([4]). It is proved that HRH shortage is a serious issue in case of national health concern. But this is not end, inappropriate mixture of skills (more doctor than nurse and technologists) is also playing a hazardous role in providing proper medical care in remote places.

When it comes to take a look into this issue from the point of view of urban area things change a little bit but the outcome is same. Government medical facilities is not enough to cope up with this huge number of people. So time efficient system can play a vital character in improving health care management in urban area.

Also prescription error causes some severe damage in the health issue. Interpreting wrong medication and dosage, review error by nurse and staffs should be reduced to upbringing more improved and compact systems for medical sectors.

Now analyzing the current market medical dictation kit is hard to find. Dragon naturally speaking provides medical kits but its price is USD1599 (per user)([3]). It is out of reach for most of the medical facilities in Bangladesh. So considering cost effectiveness was also a part for our work.

1.3 Problem definition

Speech recognition for writing transcription purpose can be very effective in reducing turnaround time per transcription (3.65 minutes versus a turnaround time of 39.6 minutes) ([27]). On various occasion saving time may be proved very crucial (emergency department, prescribing medication to a long queue, radiology). For that reason several approach to build up medical dictation has been pursued.

Medical dictionary dedicated to medicine names is not a common field of work. The existing softwares are hugely priced and not also very suitable for work environment of medical in perspective of Bangladesh. Not many medical facilities uses stand alone systems or digitally recorded data.

Another fact is that the softwares found in today's market is not only remarkably costly, it does not also support the language barrier exists in the different parts of the world. English may be the most widely used language but people from different geology and culture utters it with their very own specific accent.

In light of these issues(cost, working standalone, time efficient, and dedicated for medicine and medical prescription for Bangladesh) we intended to build up such a speech recognition

system that can be

1. Integrated with other system
2. Can work alone
3. Reduce workload of the doctors(writing prescription and store them)
4. Real time dictation speech-to-text
5. Cost efficient

Chapter 2

Literature review

2.1 Telemedicine in Bangladesh

As an emerging innovation for the benefit of human healthcare; telemedicine has been proved to bring a tremendous change in the developing country's health issue over the past decade. Evidently, rapid evolution in ICT sector has developed a prominent and remarkable effect on developing an improved telemedicine for people residing in rural or remote places. For its long distance and improved health care capabilities governments of developing countries like Bangladesh are putting much more emphasis on telemedicine and its branches. Its quite clear that adopting e-health can definitely provide a better solution for all the stakeholders involving in healthcare system like doctor, patient, nurse, physician etc.([?])

Apparently only large hospitals like Apollo, Square Hospital, United hospital, Medinova hospital and popular diagnostic center uses their own database system for keeping records of their patient records, diagnostic reports, previous medication, preferred doctor etc.([?])

According to various survey 76% people of Bangladesh lives in rural area. Facilitating this huge number of population with a better healthcare system ; an approach to build an online EHR is now on progress. The ultimate goal of this operation is to maintain an integrated and compact health record for all the patients to improve better service quality in healthcare system. The project is aimed connect all existing systems, clinics, and other health related project by internet with internet by 2016.([8])

Bangladesh's largest telecom operator Grameenphone has successfully devised tele-healthcare starting from 2007. Its wide known people-centric project named "healthline" provides 24/7 doctor's care for medical service and consultation. It has minimal call rate for providing service in terms of encouraging more people to get under the project.([6])

A telemedicine center is occupied with trained personnel who provides the expert doctor with patient health report and information such as: BP, sugar level, weight, age, any previous

report etc. The doctor can communicate with the patient directly with the help of audio/video conferencing. After consultation, the doctor prescribes medication for the patient and send the file to the center. the operator receives the file and relay it to the patient. On several occasion the doctor can also use stethoscope for further observation on the patient. ([21])

Using voice recognition software a drastic reduction of turnaround time has been documented. ([9])

According to Chin-Feng Lin, wireless telecommunication service along with mobile telecommunication service be the best viable option in the field of relaying medical data over long distance and remote places.([13]) Speech recognition can improve existing system by adopting advance concepts, equipments and technique to overcome conventional boundaries and also enrich the system with better quality.

Telemedicine in Bangladesh can significantly improve the healthcare system for the large population inhabiting in rural places. DATA ABOUT RECENT SURVEY OF MEDCAL FACILITIES NUMBER. Arrival of 3g and further advanced network can be a blessing for the implementation of a mobile based telemedicine service in rural and remote places. ([5])

2.2 JULIUS

The free software toolkit for japanese LVSCR, JULIUS is a 2 pass high performance speech recognition software([20]). It is a language independent decoder that can work with any given word dictionary, language model, acoustic model([11]). Upon given these three component julius performs its searching on a given inputs. It's response time is almost real-time and it can process up to 60k word but its accuracy depends on the language models it has been given. The internal structure of JULIUS is like the figure 2.1

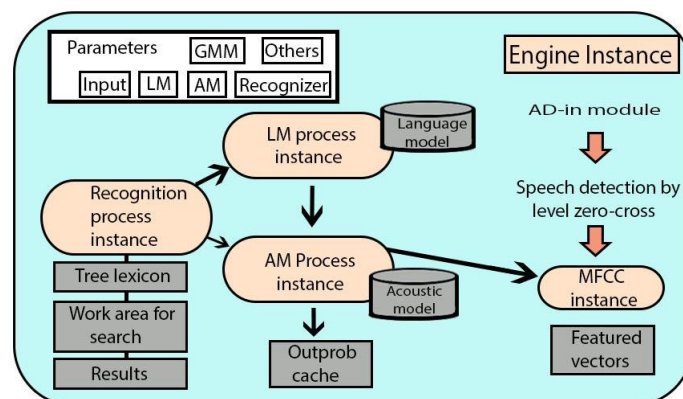


Figure 2.1: System architecture of JULIUS

In these processes JULIUS detects the speech first. When the recognition process starts it creates the lexicon tree with the given word dictionary for the phonemes that could be used in the input words. The LM and AM process instances are created from the given language model and acoustic model it has been given before. These instances leads to the creation of MFCC instances which extracts the feature vectors from the given input. From the language model given it finds out the likelihood of the input with certain word(s) and compute the result by analyzing the acoustic model and the feature vector.

As JULIUS can work with multiple model to compute and recognize a spoken word a input waveform of stochastic model. Generally it uses HMM which is a stochastic and partially observable model used when the system is autonomous. As JULIUS uses HMM ASCII format acoustic model([11]) it runs alongside of HTK to provide a HMM system with a platform of benchmark evaluation([25]). An overview of JULIUS looks like the following figure2.2

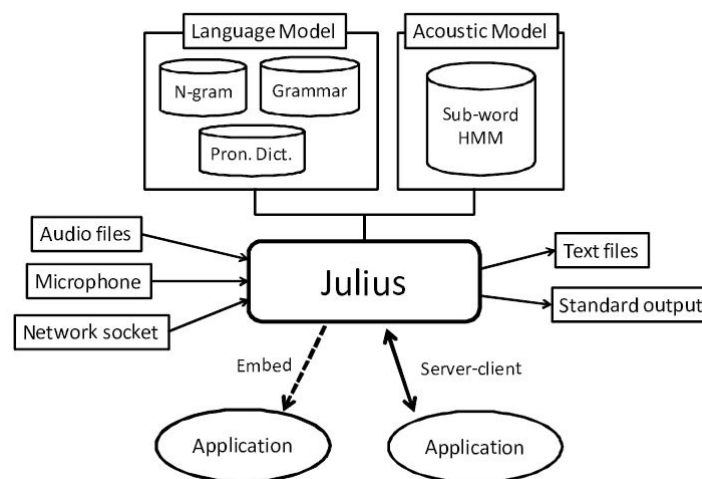


Figure 2.2: Overview of JULIUS

JULIUS also works with GMM; another model to instantiate a probabilistic field of state. The latest JULIUS stable version can be integrated with multi stream HMM and MSD-HMM trained with HTS. For this paper we have used JULIUS 4.3.1 (Stable version).

Figure2.2 shows the components of the language model that JULIUS uses. The pronunciation dictionary consists of the words sorted according to their alphabets. Each of the line consists of the words and their corresponding phonemes needed to make up the words. The dictionary we used in our system is a open source one developed by the website VOXFORGE.ORG([23]). We updated the dictionary with our own medicine name(25 names).

2.2.1 N gram model

An n-gram model is a contiguous sequence of n-items from a given series ([2]). It can be used in speech recognizing and pattern classifying. In the case of items phonemes, letters and words can be used.

N-gram model is used here as part of predicting next item in the HMM. The n in the n-gram model represents integers for 1(mono), 2(bi), 3(tri) and so on. The outcome of n-gram model is a probabilistic output by analyzing the previous output.

In this system in the place of elements we are using a set of phonemes those are needed to utter a word. For example PANTONIX(medicine name used in the system) contains the following phonemes:

PANTONIX p ae n t ow n ih k s

As we set the acoustic model to interpret the phonemes sequentially to recognize the whole word it finds out what is the probability of the phoneme “ae” after the phoneme “p”

$$P("ae"|"p") = \frac{C("p"ae")}{C("p")}$$

Now this kind of assumption is called “Maximum Likelihood Estimation” or MLE ([14]). This is the key factors or making Markov assumption

2.3 HTK

HTK is exclusively designed for manipulating HMM and has a wide work range. It can train isolated words as well as continuously spoken words with connectivity([24]). Developed at Cambridge University it supports both continuous density Gaussian mixture and discrete distribution for training HMM systems. The modules of the library and tools are available as open source material in C form.

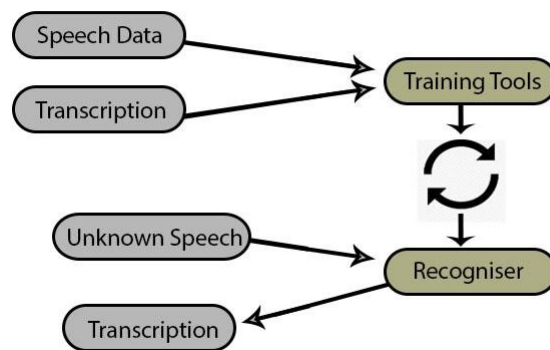


Figure 2.3: HTK architecture

Figure 2.3 shows the internal architecture of the toolkit HTK. This kit is extremely hand-ful for the purpose of speech recognition and pattern matching. As the figure shows it has two sets of major processing state. One is the estimation of HMM parameters by training instances using HTK training tools and second is the transcription of unknown speeches by the recognizer([19]).

2.4 Hidden Markov Model (HMM)

Among four Markov model HMM is used when the system is autonomous and system states are partially observable([1]). It is a stochastic model that predicts the next state on the basis of current state.

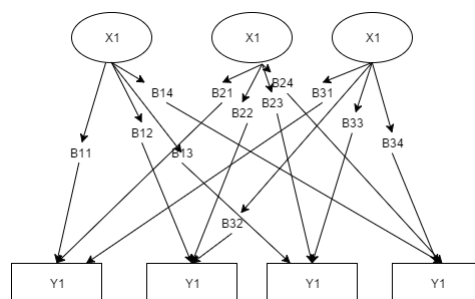


Figure 2.4: Hidden Markov Model

Figure 2.4 is a simple demonstration of Markov model processing. X being the state and y is observable state it shows each possible combination of transition that can exists in the system. After knowing all these values it creates a matrix containing each of the values.

These transition models can then be used to predict what is the best possible transition based on its current position. This quantity is the probability density function for time duration. It grows exponentially and is the characteristic of the HMM ([17]).

An HMM model needs five things to characterize:

1. The number of states
2. Number of distinct observation
3. Probability distribution
4. Observation symbol probability distribution
5. Initial probability

Upon given these five elements a Markov model generates an observation sequence which uses 3 different algorithms to run:

1. Viterbi
2. Forward
3. Baum-Welch

These three provides with different output for the model to analyze

1. Most probable corresponding state
2. Probability of the observation sequence
3. The estimation of starting probabilities.

2.5 Previous works on voice recognition with HMM and HTK

1. ASR has been developed for Arabic, Hindi, Punjabi language using HTK.
2. Cross-accent experiments show that the accent problem is very dominant in speech recognition([7]).
3. The open source speech recognition engine JULIUS has been developed with Japanese and English(American Accent).
4. Multi-level voice activity detection based on power/Gaussian mixture model (GMM)/decoder statistics has been implemented on Julius by 2009[[12]].

5. Tarun Pruthi et al. (2000) describe a speaker-dependent, real-time, isolated word recognizer for Hindi. System uses a standard implementation. Features are extracted using LPC and recognition is carried out using HMM[[14]].
6. An Isolated word speech recognition tool for Hindi language is designed by Gupta (2006) using continuous HMM. [[10]]
7. The Lincoln robust hidden Markov model speech recognizer currently provides state of the art performance for both speaker-dependent and speaker-independent large vocabulary continuous-speech recognition. [[16]]

Chapter 3

System overview

3.1 Use case



Figure 3.1: Use case diagram of the system

The above figure(3.1) shows the system overview and functionality in a simpler way. The main actors of the system are the user, Julius, HTK, Raspberry PI. Also it shows the functional workflow of the system. The user input the speech, microphone receives it and send to the

system. Then the system process is also shown in the above picture.

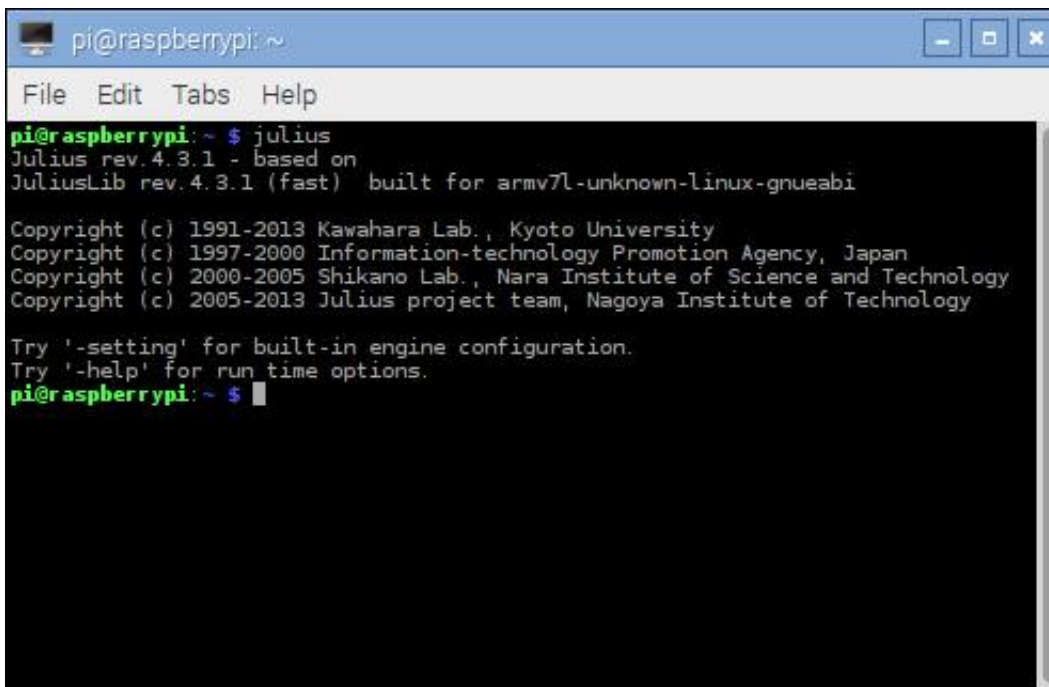
3.2 RASPBERRY PI

Being a credit card sized minicomputer, RASPBERRY PI is the core of this system in the sense of its compatibility, low cost, low power and multi functionality. We have implemented our system into two different version of PI. One is model 2B another one is 3B former one being a 32 bit system while the latter is 64 bit.

The microphone we used are both USB (one is a USB microphone another one is a A4TECH web cam). As RASPBERRY PI have only audio output in its 3.5 mm jack; USB is the better option.

3.3 implementation of JULIUS

For this system we using RASPBERRY PI model 2B. The operating system we used to run in it is RASPBIAN JESSIE (kernel version 4.3). As JULIUS is an open source material we used its current stable version 4.3.1.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi: ~ $ julius  
Julius rev.4.3.1 - based on  
JuliusLib rev.4.3.1 (fast) built for armv7l-unknown-linux-gnueabi  
  
Copyright (c) 1991-2013 Kawahara Lab., Kyoto University  
Copyright (c) 1997-2000 Information-technology Promotion Agency, Japan  
Copyright (c) 2000-2005 Shikano Lab., Nara Institute of Science and Technology  
Copyright (c) 2005-2013 Julius project team, Nagoya Institute of Technology  
  
Try '-setting' for built-in engine configuration.  
Try '-help' for run time options.  
pi@raspberrypi: ~ $
```

Figure 3.2: JULIUS on the system

3.4 Installation of HTK

To set up the HMM we had to use a dedicated toolkit for making HMM supported acoustic and language model. The HTK is designed to adopt any given HMM and transform the models accordingly. The version of HTK used in our system is 3.4.1.

But in order to compile HTK the GCC need to be integrated with version 3.4 (our default one was 4.9.2) as the latest version is not compatible with the HTK.

3.5 JULIA

JULIA is a scripting language. JULIUS can not compile the wav file directly. JULIA provides a sophisticated compiler so that JULIUS can decode the speech([22]).

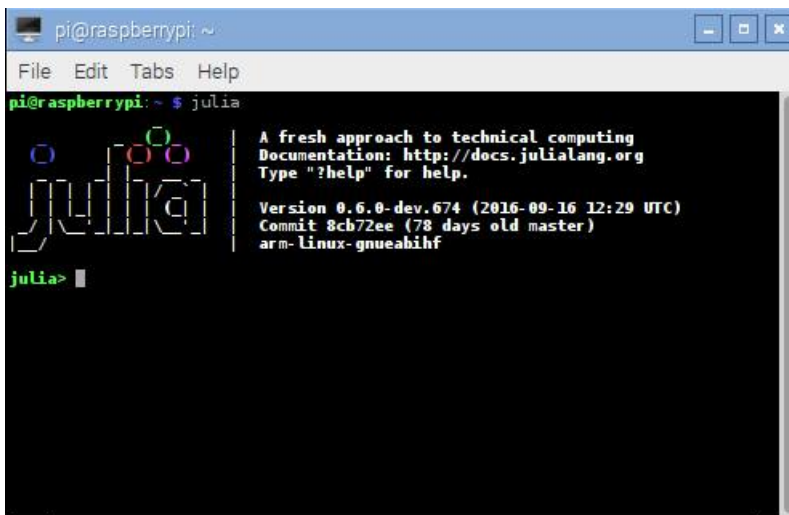
A screenshot of a terminal window on a Raspberry Pi. The window title is 'pi@raspberrypi: ~'. The terminal shows the command 'julia' being executed, which displays the Julia logo (a stylized 'julia' with colored dots) and the following text: 'A fresh approach to technical computing', 'Documentation: <http://docs.julialang.org>', 'Type "?help" for help.', 'Version 0.6.0-dev.674 (2016-09-16 12:29 UTC)', 'Commit 8cb72ee (78 days old master)', and 'arm-linux-gnueabihf'. The prompt 'julia>' is visible at the bottom left of the terminal.

Figure 3.3: JULIA(installed in the system)

Chapter 4

Processing training data

4.1 Language model

Our target was to recognize live speech input using Julius in Raspberry PI. Julius can recognize isolated word as well as a complete sentence. so for a complete sentence there must be a grammar or rule. As a result there is a “.grammar” file and “.voca” file. In “.grammar” file contains sets of predefined combinations of words. On the other hand “.voca” file contains the word with specific phonemes.

4.1.1 Grammar file

A grammar file contains the following lines ..

```
S : NS_B SENTENCE NS_E
SENTENCE: ACTION MEDICINE
```

Figure 4.1: Grammar file

Here “S” is the initial sentence symbol. “NS_B” and “NS_E” correspond to the silence that occurs just before the utterance we want to recognize and after. “S”, “NS_B” and “NS_E” are required in all Julius grammars. “NS_B”, “NS_E”, “ACTION”, and “MEDICINE” are terminals, and represent Word Categories that are defined in the “.voca” file. Here only “SENTENCE” is non terminal which is replaced by two terminals (“ACTION” and “MEDICINE”).

4.1.2 Voca file

The voca file contains the words of different category inscribed in the grammar file so that the system can practically expect the probable inputs from a user

```

% NS_B
<s> sil
% NS_E
</s> sil
% ACTION
WRITE r ay t
% MEDICINE
ACE          ey s
ADOVAS       ae d ow v aa s
AMODIS       ae m ow d ih s
NAPA         n ah p ah
SECLO        s eh k l ow
IMOTIL       ih m ow t iy l
KOP          k ow p
LIDO         l iy d ow
MELIXOL      m eh l ih k z ow l
NEBANOL      n eh b aa n ow l
NORVIS       n ow r v ih s
ONI          ow n ih
DEXPOTEN     d eh k s p ow t eh n
PARACETAMOL p er ae s ih t ah m ow l
PANTONIX     p ae n t ow n ih k s
ANTACID      ae n t ae s ih d
BUTAFEN      b uw t aa f eh n
BUTAFEN      b uw t aa f ih n
MOTIGUT      m ow t iy g ah t
FLAGYL       f l ah zh ih l

```

Figure 4.2: Voca file

In this .voca file each terminal is defined clearly with their phonemes. "NS_B" and "NS_E" for 'silence begin' and 'silence end'. So "NS_B" and "NS_E" has one word definition with silence model(sil).

"ACTION" is broken into one word :

write (r ay t)

Here 'r ay t' is the combination of the phonemes when 'write' is spoken. There are 44 phonemes in English. We have to define each word with its corresponding combination of

phonemes according to Asian peoples' pronunciation.

“MEDICINE” is broken into 27 separate words (medicine names). pronunciation may vary a little bit from one person to another Such as

INDEVER (iy n d eh v aa r)

INDEVER (iy n d ih v aa r)

For that reason we included multiple possible way to utter that word in the voca file.

4.1.3 Prompts file

A sample file is needed to make a documentation of data (words) so that while training words through a audio file it can learn which audio wave represent rich word. The figure below is a sample of our prompts file

```
*/sample1 PRESCRIBE SECLO NAPA ONI ACE MELIXOL IMOTIL KOP
*/sample2 NAPA SECLO ACE ONI KOP LIDO WRITE DEXPOTEN NORVIS
*/sample3 ACE ACE NAPA NAPA PARACETAMOL NORVIS NORVIS PARACETAMOL
*/sample4 WRITE WRITE PRESCRIBE PRESCRIBE PARACETAMOL AMODIS
ADOVAS
*/sample5 PRESCRIBE NAPA WRITE PARACETAMOL DEXPOTEN ACE MELIXOL
KOP
*/sample6 BOOKENDS KENNEL KENNETH KENYA WEEKEND
*/sample7 BELT BELOW BEND AEROBIC DASHBOARD DATABASE
*/sample8 GATEWAY GATORADE GAZEBO AFGHAN AGAINST AGATHA
*/sample9 ABALON ABDOMINALS BODY ABOLISH
*/sample10 ABOUNDING ABOUT ACCOUNT ALLENTOWN
```

Figure 4.3: Prompts file

Each line of this represents the corresponding voice training input from different users. There are 2 different columns; first one represents the name of sample file while second one represents the line expected from the speaker who will train the system.

4.1.4 Dictionary file

The dictionary file we used to compile our system contains 268982 words with each one's phonemes. All these words used English phonemes with proper accents. We have included our own 27 medicine names in it with our own modulated phoneme series. we did not change the

phonemes but we created the words with certain phonemes so that it can be trained with people having south Asian accent. The following figure contains a some lines from the dictionary file

ABACUS	[ABACUS]	ae b ah k ah s
ABANDON	[ABANDON]	ah b ae n d ah n
JOB	[JOB]	jh aa b
SCHOOL	[SCHOOL]	s k uw l
THESIS	[THESIS]	th iy s ih s
VOICE	[VOICE]	v oy s
RECOGNITION	[RECOGNITION]	r eh k ah g n ih sh ah n

Figure 4.4: Glimpse of the dictionary file

Noting that the dictionary is taken from www.voxforge.org([23]). It is an open source website contributing to develop a dictionary for English words.

4.2 Audio training

Audio training from different speakers is a key part of this system. It trains the system with different audio wave patterns to learn the words as it meant to be uttered.

4.2.1 WAV files

Audio data needs to be recorded in wav format. An WAV file is used generally for its lossless, uncompromisable quality and uncompressed audio bit stream storage.

SL	Parameters	Value
1	Training input audio format	WAV
2	Sampling rate	16000 Hz
3	Sample format	16 bit
4	Channel	mono

Table 4.1: Audio files parameters

The audio file is recorded between the range of -0.3 to 0.3 in recording software. The following figure shows the wave change and range

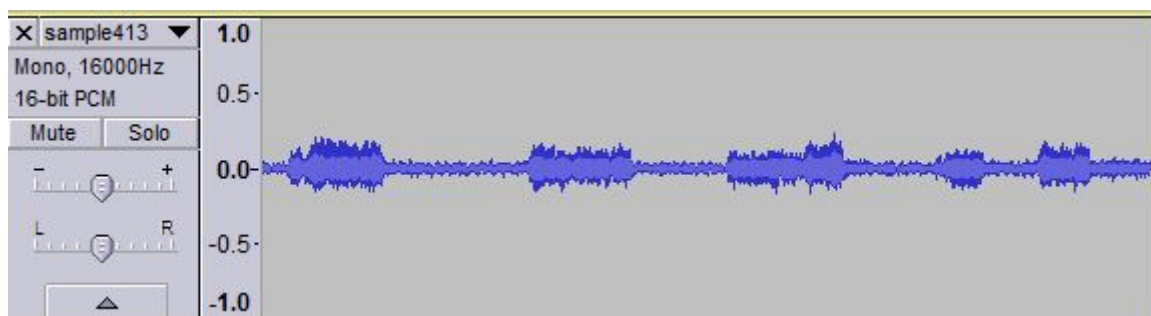


Figure 4.5: Audio wave sample

4.2.2 MFCC

HTK is not able to process WAV format files efficiently in its original form. For that, it is needed to be transformed into subsequent MFCC format files. The parameters used in the conversion process is in the following table

SL	Parameters	Value	Purpose
1	Sourceformat	WAV	Identify source format
2	Targetkind	MFCC	The format that the file needs to be transformed into
3	Targetrate	100 μ sec	Sample rate of target
4	Savedformat	True	Save output in compressed form

Table 4.2: Audio conversion parameters

All these parameters are documented in the HTKbook ([19]). MFCC formats allows better audio recognition, modulation of subjective pitch with expected frequency content of the audio signal.([26])

4.3 Configuration

4.3.1 JCONF File

The variables that can be written in Jconf file are organized as follows.

- Global options
- Instance declaration
- Language model instance

- Acoustic model and speech analysis instance
- Recognizer and search instance

4.3.2 Global option

4.3.2.1 Miscellaneous

1. **-C jconffile**

The **jconf** file is loaded and the options are extended here which other jconf files can use.

2. **-version**

Provides information about the version.

3. **-settings**

Gives information about engine settings.

4. **-quiet**

Makes the log output less and prints only the best answer.

5. **-debug**

(For debug) Sends extended internal information to log.

6. **-check {wchmm|trellis|triphone}**

For debug, goes into interactive check mode.

4.3.3 Grammar

Multiple grammars can be specified by using **-gram** and **-gramlist**. When you specify grammars using these options multiple times, all of them will be read at startup. Note that this is unusual behavior from other options (in normal Julius option, last one override previous ones). You can use **-nogram** to reset the already specified grammars at that point.

1. **-dfa dfa_file**

Finite state automaton grammar file.

2. **-v dict_file**

Pronunciation dictionary of the language model.

4.3.4 Audio input

4.3.4.1 `-input {micrawfile|mfcfile|adinnet|stdin|netaudio}`

Mic command is used to get live input from the speaker. we used mic to get live speech in this case.

Only WAV (no compression) and RAW (noheader, 16bit, big endian) are supported for wave input as default.

4.3.4.2 Speech segment detection by level and zero-cross

- **-cutsilence**
- **-nocutsilence**

The speech detection is turned on/off by level and zero cross.

- **-lv thresh**

Threshold level for speech input detection which values from 0 to 32767.

- **-zc thresh**

Determines how many times the wave has to cross the zero crossing per second to be counted. (default: 60)

- **-headmargin msec**

The amount of time it waits to take the input at the beginning. (default: 300)

- **-tailmargin msec**

The amount of time it waits after the input was received to make sure the input was completed. (default: 400)

- **-rejectshort msec**

Shorter input than the specified length will not be accepted and the search will be terminated with no result.

4.3.4.3 acoustic HMM and parameters

- **-h hmmdef_file**

Acoustic HMM definition file should be in HTK ASCII format, or Julius binary format.

- **-hlist hmmlist_file**

HMMList file for phone mapping. This options is required when using a triphone model. This file provides a mapping between logical triphone names generated from the dictionary and defined HMM names in hmmdefs.

- **-spmodel name**

Specify an HMM name that corresponds to short-pause model in HMM. This option will affect various aspects in recognition: short-pause skipping process on grammar recognition, word-end short-pause model insertion with -iwsp on N-gram recognition, or short-pause segmentation (-spsegment). (default: "sp").

- **-multipath**

Enable multi-path mode. Multi-path mode expand state transition availability to allow model-skipping, or multiple output/input transitions in HMMs. However, since defining additional word begin / end node and perform extra transition check on decoding, the beam width may be required to set larger and recognition becomes a bit slower.

- **-iwcd1 {max|avg|best number}**

Select method to approximate inter-word triphone on the head and tail of a word in the first pass. "max" will apply the maximum likelihood of the same context triphones. "avg" will apply the average likelihood of the same context triphones. "best number" will apply the average of top N-best likelihoods of the same context triphone. Default is "best 3" for use with N-gram, and "avg" for grammar and word. When this AM is shared by LMs of both type, latter one will be chosen.

- **-iwspenalty float**

Short pause insertion penalty for appended short pauses by -iwsp.

4.3.4.4 Speech analysis parameters

- **-smpFreq Hz**

Set sampling frequency of input speech in Hz. Sampling rate can also be specified using

- **-smpPeriod**

This frequency should be the same as the trained conditions of acoustic model that is used. (default: 16000). When using multiple AM, this value should be the same among all AMs.

4.3.5 Recognizer and search (-SR)

4.3.5.1 General parameters

- **-inactive**

The recognition process is started with inactive state. (Rev.4.0).

- **-1pass**

Perform only the first pass. This mode is automatically set at isolated word recognition.

- **-iwsp**

(Multi-path mode only) Enable inter-word context-free short pause handling. This option appends a short pause model for every word end. The added model will be skipped on inter-word context handling. The HMM model to be appended can be specified by -spmmodel.

4.3.5.2 1st pass parameters

- **-lmp weight penalty**

(N-gram) Language model weights and word insertion penalties for the first pass.

- **-penalty1 penalty**

(Grammar) word insertion penalty for the first pass. (default: 0.0)

- **-b width**

Beam width for rank beam in number of HMM nodes on the first pass. This value defines search width on the 1st pass, and has great effect on the total processing time. Smaller width will speed up the decoding, but too small value will result in a substantial increase of recognition errors due to search failure. Larger value will make the search stable and will lead to failure-free search, but processing time and memory usage will grow in proportion to the width. The default value is dependent on acoustic model type: 400 (monophone), 800 (triphone), or 1000 (triphone, setup=v2.1)

4.3.5.3 2nd pass parameters

- **-lmp2 weight penalty**

(N-gram) Language model weights and word insertion penalties for the second pass.

- **-penalty2 penalty**

(Grammar) word insertion penalty for the second pass. (default: 0.0)

- **-b2 width**

Envelope beam width (number of hypothesis) in second pass. If the count of word expansion at a certain length of hypothesis reaches this limit while search, shorter hypotheses are not expanded further. This prevents search to fall in breadth-first-like status stacking on the same position, and improve search failure.

Chapter 5

Tests and results

5.1 Table of different user who trained data

5.1.1 Trained Speaker:

5.1.1.1 Speaker 1 and Speaker 2:(MALE)

Medicine	Speaker 1 normal env. right output	Speaker 1 normal env. search failed	Speaker 1 normal env. right output	Speaker 2 normal env. search failed	Speaker 2 normal env. right output	Speaker 2 normal env. search failed	Speaker 2 silent env. right output	Speaker 2 silent env. search failed
Napa	7	3	10	0	9	1	9	1
Ace	6	4	8	2	8	2	10	0
Adovas	5	5	10	0	10	0	10	0
Amodis	10	0	10	0	10	0	10	0
Seclo	10	0	9	1	10	0	10	0
Imotil	10	0	10	0	10	0	10	0
KOP	0	10	3	7	0	10	0	10
Lido	9	1	10	0	10	0	10	0
Melixol	8	2	10	0	10	0	10	0
Nebanol	10	0	10	0	10	0	10	0
Norvis	6	4	10	0	10	0	10	0
Oni	2	8	8	2	2	8	3	7
Dexpotin	7	3	8	2	10	0	10	0
Pantonix	10	0	10	0	10	0	10	0
Antacid	10	0	10	0	10	0	10	0

Table 5.1: Male speakers who trained data(continued)

Medicine	Speaker 1 normal env. right output	Speaker 1 normal env. search failed	Speaker 1 normal env. right output	Speaker 2 normal env. search failed	Speaker 2 normal env. right output	Speaker 2 normal env. search failed	Speaker 2 silent env. right output	Speaker 2 silent env. search failed
Revert	8	2	8	2	2	8	2	8
Emistat	10	0	10	0	10	0	10	0
Sedil	10	0	10	0	9	1	9	1
Rolac	10	0	10	0	5	5	5	5
Indever	10	0	10	0	10	0	10	0
Ventolin	10	0	10	0	10	0	10	0
Parisol	10	0	10	0	10	0	10	0
Butafen	10	0	10	0	10	0	10	0
Parkinil	10	0	10	0	10	0	10	0
Motigut	10	0	10	0	10	0	10	0
Flagyl	10	0	10	0	10	0	10	0
Metril	7	3	9	1	10	0	10	0
Paracetamol	10	0	10	0	10	0	10	0
Total	235	45	263	17	245	35	248	32

Table 5.2: Male speakers who trained data(end)

Table (5.1) and (5.2) is the tabular data collection of test speakers with our total list of medicine names that we trained in our system, in this case the number is 28. These two tables contains the data collection from female speakers who trained the system. It is quite clear that except for the medicine name consisting of two short syllables and phoneme the system can detect the medicine name quite flawlessly.

5.1.1.2 Speaker 3 and Speaker 4:(FEMALE)

Medicine	Speaker 3 normal env. right output	Speaker 3 normal env. search failed	Speaker 3 silent env. right output	Speaker 3 silent env. search failed	Speaker 4 normal env. right output	Speaker 4 normal env. search failed	Speaker 4 silent env. right output	Speaker 4 silent env. search failed
Napa	5	5	9	1	7	3	9	1
Ace	2	8	9	1	6	4	8	2
Adovas	10	0	10	0	10	0	10	0
Amodis	10	0	10	0	10	0	10	0
Seclo	10	0	10	0	10	0	10	0
Imotil	10	0	10	0	10	0	10	0
KOP	10	0	0	10	0	10	0	10
Lido	8	2	10	0	10	0	10	0
Melixol	10	0	10	0	10	0	10	0
Nebanol	10	0	10	0	10	0	10	0
Norvis	10	0	10	0	10	0	10	0
Oni	0	10	4	6	0	10	1	9
Dexpotin	10	0	10	0	10	0	10	0
Pantonix	10	0	10	0	10	0	10	0
Antacid	10	0	10	0	10	0	10	0

Table 5.3: Female speakers who trained data(continued)

Table (5.3) and (5.4) is the tabular data collection of test speakers with our total list of medicine names that we trained in our system, in this case the number is 28. These two tables contains the data collection from female speakers who trained the system. It is quite clear that except for the medicine name consisting of two short syllables and phoneme the system can detect the medicine name quite flawlessly.

Medicine	Speaker 3 normal env. right output	Speaker 3 normal env. search failed	Speaker 3 silent env. right output	Speaker 3 silent env. search failed	Speaker 4 normal env. right output	Speaker 4 normal env. search failed	Speaker 4 silent env. right output	Speaker 4 silent env. search failed
Revert	0	10	0	10	0	3	7	3
Emistat	10	0	10	0	0	10	0	10
Sedil	10	0	10	0	0	9	1	9
Rolac	2	8	6	4	8	5	5	5
Indever	3	7	6	4	0	6	4	6
Ventolin	5	5	5	5	0	10	0	10
Parisol	10	0	10	0	0	10	0	10
Butafen	10	0	10	0	0	10	0	10
Parkinil	10	0	10	0	0	10	0	10
Motigut	10	0	10	0	3	10	0	10
Flagyl	10	0	10	0	0	10	0	10
Metril	10	0	10	0	0	10	0	10
Paracetamol	10	0	10	0	0	10	0	10
Total	225	55	239	41	236	44	241	39

Table 5.4: Female speakers who trained data(end)

5.1.2 Speakers who did not train data

5.1.2.1 Speaker 5 and Speaker 6:(MALE)

Medicine	Speaker 5	Speaker 5	Speaker 5	Speaker 5	Speaker 6	Speaker 6	Speaker 6	Speaker 6
	normal env. right output	normal env. search failed	normal env. right output	normal env. search failed	normal env. right output	normal env. search failed	silent env. right output	silent env. search failed
Napa	10	0	10	0	0	10	0	10
Ace	10	0	10	0	0	10	0	10
Adovas	10	0	10	0	10	0	10	0
Amodis	10	0	10	0	10	0	10	0
Seclo	10	0	10	0	10	0	10	0
Imotil	10	0	10	0	10	0	10	0
KOP	0	10	0	10	0	10	0	0
Lido	10	0	10	0	6	4	7	3
Melixol	10	0	10	0	10	0	10	0
Nebanol	10	0	10	0	10	0	10	0
Norvis	10	0	10	0	10	0	10	0
Oni	0	10	0	10	0	10	0	10
Dexpotin	10	0	10	0	10	0	10	0
Pantonix	10	0	10	0	10	0	10	0
Antacid	10	0	10	0	10	0	10	0

Table 5.5: Male speakers who did not train data(continued)

The above(5.5) and following(5.6) table is the test data collection from male speakers who were new into the system. We did not take training data from them. They were asked to say the medicine name and we collected the right output and “search failed” output from them. The number of medicine name is 28 here also. Also noting that the output is quite similar to the results collected from the speakers who trained the system rejecting words with shorter syllables.

Medicine	Speaker 5 normal env. right output	Speaker 5 normal env. search failed	Speaker 5 normal env. right output	Speaker 5 normal env. search failed	Speaker 6 normal env. right output	Speaker 6 normal env. search failed	Speaker 6 silent env. right output	Speaker 6 silent env. search failed
Revert	10	0	10	1	2	8	5	5
Emistat	10	0	10	0	10	0	10	0
Sedil	10	0	10	0	10	0	10	0
Rolac	9	1	10	7	3	7	6	4
Indever	10	0	10	0	6	4	6	4
Ventolin	10	0	10	0	8	2	8	2
Parisol	10	0	10	0	10	0	10	0
Butafen	10	0	10	0	9	1	9	1
Parkinil	10	0	10	0	9	1	9	1
Motigut	10	0	10	0	7	3	7	3
Flagyl	10	0	10	0	9	1	9	1
Metril	10	0	10	0	9	1	9	1
Paracetamol	10	0	10	0	10	0	10	0
Total	259	21	260	32	208	72	215	55

Table 5.6: Male speakers who did not train data(end)

5.1.2.2 Speaker 7 and Speaker 8:(FEMALE)

Medicine	Speaker 7	Speaker 7	Speaker 7	Speaker 7	Speaker 8	Speaker 8	Speaker 8	Speaker 8
	normal env. right output	normal env. search failed	normal env. right output	normal env. search failed	normal env. right output	normal env. search failed	silent env. right output	silent env. search failed
Napa	0	10	0	10	9	1	10	0
Ace	0	10	0	10	4	6	6	4
Adovas	10	0	10	0	9	1	9	1
Amodis	7	3	8	2	10	0	10	0
Seclo	9	1	9	1	10	0	10	0
Imotil	7	3	8	2	10	0	10	0
KOP	0	10	0	10	0	10	0	10
Lido	6	4	8	2	10	0	10	0
Melixol	9	1	10	0	10	0	10	0
Nebanol	9	1	10	0	10	0	10	0
Norvis	10	0	10	0	10	0	10	0
Oni	0	10	0	10	1	9	1	9
Dexpotin	9	1	10	0	10	0	10	0
Pantonix	10	0	10	0	10	0	10	0
Antacid	10	0	9	1	10	0	10	0

Table 5.7: Female speakers who did not train data(continued)

The above(5.7) and following(5.8) table is the test data collection from female speakers who were new into the system. We did not take training data from them. They were asked to say the medicine name and we collected the right output and “search failed” output from them. The number of medicine name is 28 here also. Also noting that the output is quite similar to the results collected from the speakers who trained the system rejecting words with shorter syllables.

Medicine	Speaker 7 normal env. right output	Speaker 7 normal env. search failed	Speaker 7 normal env. right output	Speaker 7 normal env. search failed	Speaker 8 normal env. right output	Speaker 8 normal env. search failed	Speaker 8 silent env. right output	Speaker 8 silent env. search failed
Revert	2	8	6	4	10	0	10	0
Emistat	10	0	10	0	10	0	10	0
Sedil	8	2	9	1	10	0	10	0
Rolac	2	8	3	7	1	9	2	8
Indever	5	5	6	4	8	2	10	0
Ventolin	10	0	10	0	10	0	10	0
Parisol	10	0	10	0	10	0	10	0
Butafen	7	3	8	2	10	0	10	0
Parkinil	10	0	10	0	10	0	10	0
Motigut	10	0	10	0	7	3	7	3
Flagyl	9	1	9	1	10	0	10	0
Metril	10	0	10	0	10	0	10	0
Paracetamol	10	0	10	0	10	0	10	0
Total	199	81	213	67	239	41	245	35

Table 5.8: Female speakers who did not train data(end)

5.2 Charts of various accuracy and analysis

5.2.1 Male vs. Female (Who trained data)

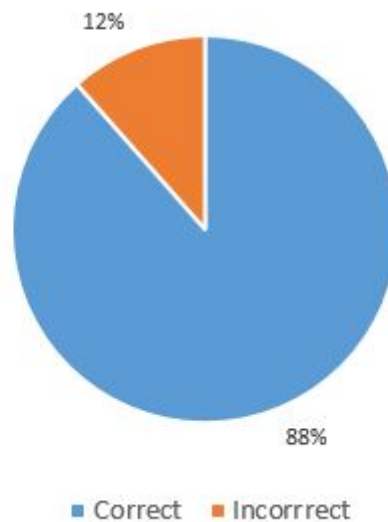


Figure 5.1: Male speakers(Who trained data)

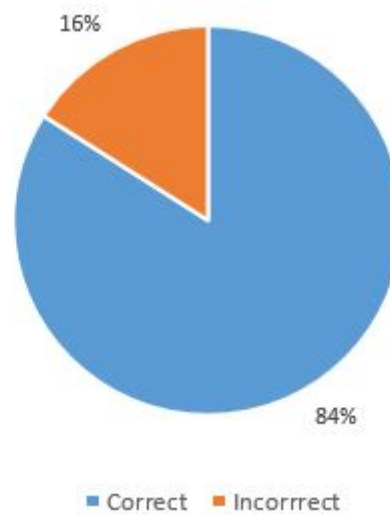


Figure 5.2: Female speakers(Who trained data)

In the above Figure (5.1) we can see the amount of correct outputs against the number of incorrect outputs when the testing person was a male who has trained the system. We can see that the results has an accuracy of 88% which is satisfactory. In this case, the incorrect outputs

were a mix of “Search Failed” and Incorrect outputs of which almost all of them where search failed. Figure (5.2) we see the correct outputs against the number of incorrect outputs when the system was being tested by a female who had trained the system previously. Here we got an accuracy of 84% which is lower than the accuracy the system achieved in case of a male tester because 71% of the total training given to the system was given by male trainers. In this case as well almost all of the incorrect outputs were search fails.

5.2.2 Male vs. Female (Who did not trained data)

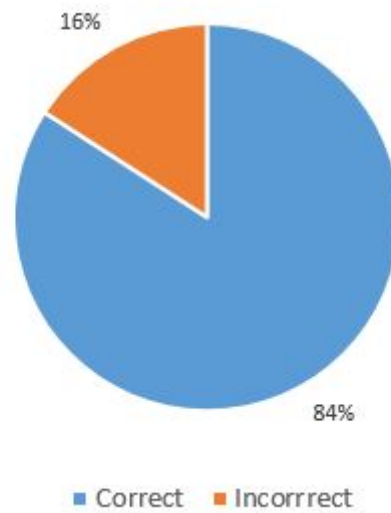


Figure 5.3: Male speakers(Who did not trained data)

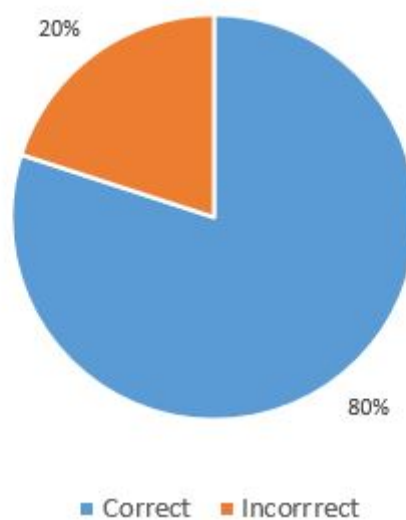


Figure 5.4: Female speakers(Who did not trained data)

In the above figure (5.3) we see the correct outputs Vs the Incorrect outputs when the testing person has not trained the system previously. In this case the system gained an accuracy of 84%. It was slightly less than the accuracy we got when the testing was being by a person who trained the system, which was expected.

In the above figure (5.4) we can see the amount correct outputs Vs the number of incor-

rect outputs when the system was being tested by a woman who had not tested the system previously. Here we also acquired an accuracy of 80% which was a little lower than what we achieved when the system was being tested by a female who had trained previously.

5.2.3 Overall accuracy in silent and normal environment

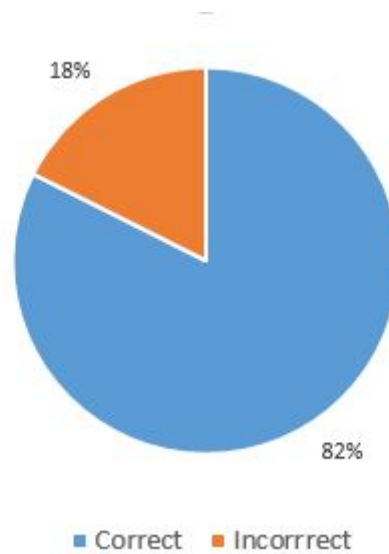


Figure 5.5: Overall accuracy (normal environment)

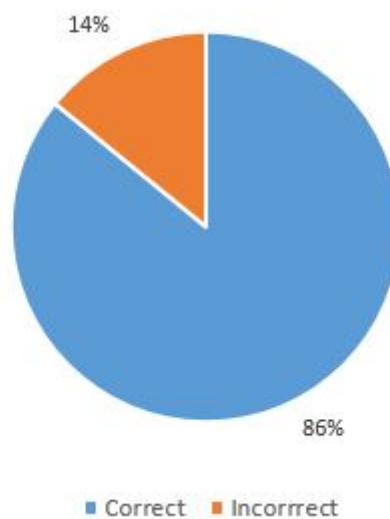


Figure 5.6: Overall accuracy (silent environment)

In the above Figure (5.5) we compared the number of correct outputs against the number of incorrect outputs when we tested the system in normal environment. The gained accuracy was 82% , which was slightly lower than expected. The main reason behind it was the microphone not being one directional and capturing noises from all the directions which derailed the results in most cases. figure (5.6) we compare the system's accuracy when it was being tested in a silent environment. Here we obtained an accuracy of 86% which was better than the results we had in a normal environment because, in spite of the microphone not being one directional there was no noise for the microphone to catch which led us to a better result.

5.2.4 Overall accuracy for longer strings in silent and normal environment

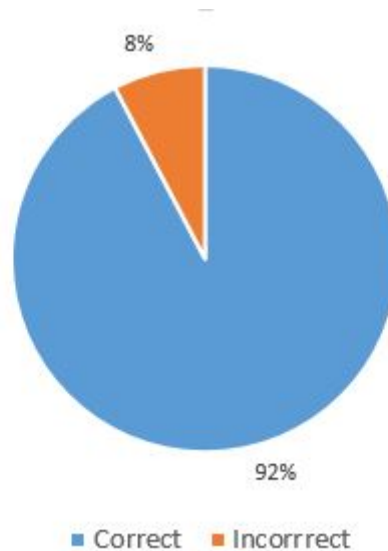


Figure 5.7: Overall accuracy for longer strings (normal environment)

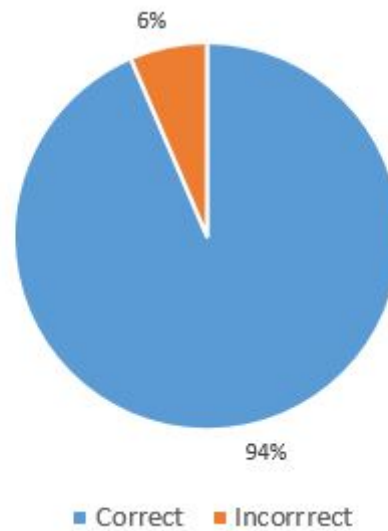


Figure 5.8: Overall accuracy for longer strings (silent environment)

IN the above Figure (5.7) we observe the total system's accuracy if we disregard the results we got from the strings shorter than 4 lengths. In this case we get an excellent result of 92%.

Which tells us there was issue in processing the words with one or two syllables.

In the above figure (5.8) we see the results when we test the system in a silent environment excluding the short strings. In this case the system obtained an accuracy of 94%. This was the highest accuracy the system achieved in any condition.

5.2.5 Male vs. female comparison in overall accuracy

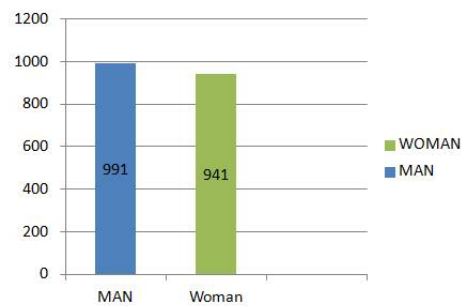


Figure 5.9: Male vs. female speaker accuracy(who trained data)

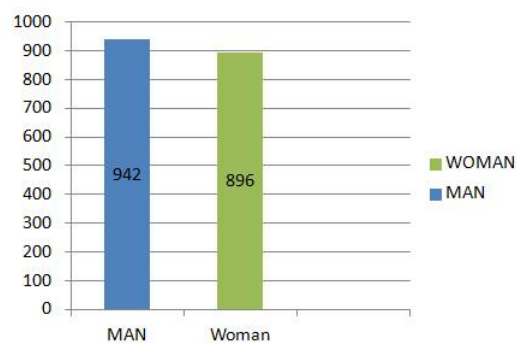


Figure 5.10: Male vs. female speaker accuracy(who did not train data)

In the above figure(5.9) we can see the correct number of accurate results the male testers had against the number the female testers achieved, given they had trained the system previously. In this case the male testers had achieved more correct answers than the female testers had. The difference between the number of male and female training files given to the system might be the reason in this case.

In the above figure(5.10) we can see the accurate outputs the male testers had against the number of correct outputs the female testers had who had not trained the system. Here we see a slight difference in the accuracy which might have been caused by the fact that there was a difference between the male training and the female training the system had received.

5.2.6 Overall accuracy with complete dataset in different environment

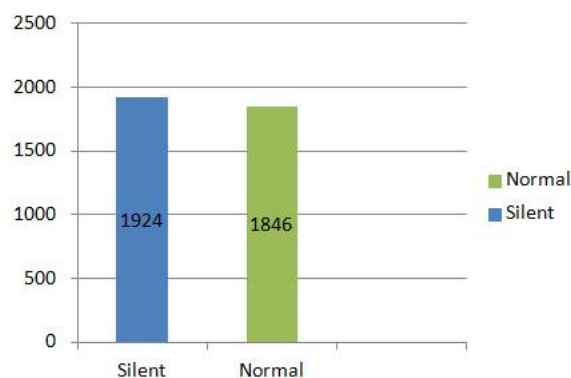


Figure 5.11: Overall accuracy(silent vs. normal)

In the above figure(5.11) we see the system's accuracy when the system was tested in a normal environment against the accuracy the system achieved in a silent environment. This of course had a difference in accuracy. The microphone not being one directional and catching noise from every direction reduced normal environment's accuracy in this case.

5.2.7 Length vs. Accuracy

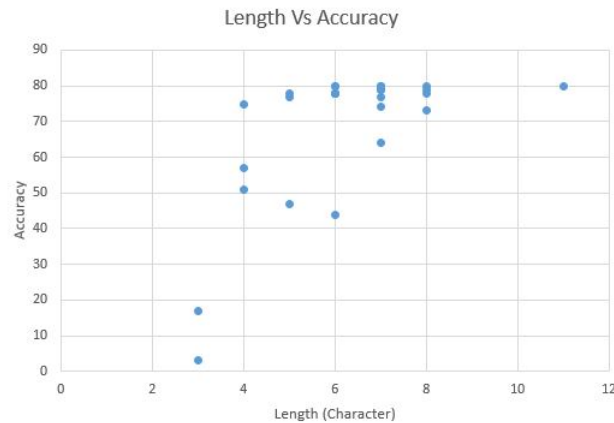


Figure 5.12: Length vs. accuracy

In the above figure (5.12) we showed the comparison of a medicine's name against their accuracy in overall. We can clearly see from the above graph that the higher the name's length the higher accuracy they achieved.

5.3 Result analysis

5.3.1 Microphone issue

From our test we noticed some factors of our result. First of all it is about microphone. We have to use a better microphone which is directional. At the very beginning we tested our system using a omni directional microphone which was catching sound around the system. So sometimes by getting human speech of others rather than the speaker the system started to process of detection the speech and it was giving search failed or a wrong output. On contrary if the microphone is directional it will decrease the noise and will avoid the speech of the people around the speaker.

5.3.2 Longer strings

Secondly we noticed that it was giving perfect output all most all the time for the longer medicine name like 'Amodis', 'Norvis', 'Melixol'. On the other side if the name of the medicine name is too small like 'Kop', 'Ace', 'Oni' it was giving a bad result (most of the time 'search failed').

5.3.3 Environment

The next factor is about the environment surrounding the system. In silent environment the system works very fine. In normal environment like sound of fan or vehicles the output is also satisfactory. However if the environment is crowded It does not give a satisfactory result. It can be reduce by using directional microphone.

Chapter 6

Conclusion

6.1 Conclusion

Lack of medical health care in the rural area is a concerning issue from a nationwide perspective. For the largest portion of our population living in rural areas; ensuring them with proper health is one of the biggest challenge for the authority. We strongly believe telemedicine can play a vital factor for this issue and adopting a system with speech recognition can make the process a lot more efficient, user friendly and time conserving for both doctor and patient. We tried to compile our system with keeping the accent concern in our mind. We also tried to make the system less complicated in the outside so that people not familiar with this kind of technology can also use the system as per their own capability. Also we think speech recognition can really be the newer, more compact and effortless input system for all kind of electronic device. It will really ease the patient live if they don't have to worry about their medication if they can have a system generated prescription in their hand. On the other hand from the perspective of doctor this can reduce their workload and also their unwillingness to write transcription and store them in database. This paper intends to encourage further works regarding speech recognition concerning the people of south east Asia specially Bangladesh.

6.2 Further work

- **Implementing network capabilities**

Our system is offline. However we want to implement the system online. So people can buy the device and connect online to get the service.

- **Integrating with existing telepharmacy services**

For instance Grameen Phone is providing telepharmacy services through mobile calls. Our system can be integrated in such a way that will improve the whole system.

- **Introducing learning agent**

A learning agent can be introduced to the system so that it can train the system with new medicine names from the doctors.

- **Maintaining dedicated database to store/retrieve records**

A database can be there so that the system can store and retrieve data whenever it needs to.

- **Single centralized system.**

There will be one centralized system so that if one doctor trains the system for a certain medicine nobody else has to.

Refernces

- [1] Hidden markov model. Internet: https://en.wikipedia.org/wiki/Hidden_Markov_model.
- [2] N-gram model. Internet: <https://en.wikipedia.org/wiki/N-gram>.
- [3] Nuance dragon medical practice, edition 2. <https://www.amazon.com/Nuance-Dragon-Medical-Practice-2/dp/B00DGCHDBS>. Accessed: 2016-11-28.
- [4] Syed Masud Ahmed, Md Awlad Hossain, Ahmed Mushtaque RajaChowdhury, and Abbas Uddin Bhuiya. The health workforce crisis in bangladesh: shortage, inappropriate skill-mix and inequitable distribution. *Human Resources for Health*, 9(1):1, 2011.
- [5] M Sanaullah Chowdhury, Humaun Kabir, Kazi Ashrafuzzaman, and Kyung Sup Kwak. A telecommunication network architecture for telemedicine in bangladesh and its applicability. *JDCTA*, 3(3):156–166, 2009.
- [6] GrameenPhone. Grameenphone launches health information & service. <https://m.grameenphone.com/bn/node/2143>.
- [7] Chao Huang, Tao Chen, and Eric Chang. Accent issues in large vocabulary continuous speech recognition. *International Journal of Speech Technology*, 7(2-3):141–153, 2004.
- [8] Sana Z Khan, Zahraa Shahid, Karin Hedstrom, and Annika Andersson. Hopes and fears in implementation of electronic health records in bangladesh. *The Electronic Journal of Information Systems in Developing Countries*, 54, 2012.
- [9] Arun Krishnaraj, Joseph KT Lee, Sandra A Laws, and T Jay Crawford. Voice recognition software: effect on radiology report turnaround time at an academic medical center. *American Journal of Roentgenology*, 195(1):194–197, 2010.
- [10] Kuldeep Kumar, RK Aggarwal, and Ankita Jain. A hindi speech recognition system for connected words using htk. *International Journal of Computational Systems Engineering*, 1(1):25–32, 2012.

- [11] Akinobu Lee. The julius book. Internet: <https://julius.osdn.jp/juliusbook/en>, 2008.
- [12] Akinobu Lee and Tatsuya Kawahara. Recent development of open-source speech recognition engine julius. In *Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference*, pages 131–137. Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, International Organizing Committee, 2009.
- [13] Chin-Feng Lin, Cheng-Hsing Chung, Zhi-Lu Chen, Chang-Jin Song, and Zhi-Xiang Wang. A chaos-based unequal encryption mechanism in wireless telemedicine with error decryption. *WSEAS Transactions on Systems*, 7(2):49–55, 2008.
- [14] James H Martin and Daniel Jurafsky. Speech and language processing. *International Edition*, 710, 2000.
- [15] Mark A Musen and Jan H van Bemmelen. *Handbook of medical informatics*. Bohn Stafleu Van Loghum Houten, 1997.
- [16] DB Paul. The lincoln robust continuous speech recognizer. In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, pages 449–452. IEEE, 1989.
- [17] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [18] DI Rosenthal, FS Chew, DE Dupuy, SV Kattapuram, WE Palmer, RM Yap, and LA Levine. Computer-based speech recognition as a replacement for medical transcription. *AJR. American journal of roentgenology*, 170(1):23–25, 1998.
- [19] Phil Steve Young, Julian Woodland, and Valtchev Odell. The htk book. Internet: <http://htk.eng.cam.ac.uk/docs/docs.shtml>, December 2015.
- [20] Julius Team. Julius. <http://julius.osdn.jp/en-index.php>.
- [21] Results Management Team. A tcv+ study on field trial of telemedicine using locally developed pc based diagnostic equipment. <http://www.a2i.pmo.gov.bd/resources/studies/>, pages 8–10, February 2016.
- [22] Voxforge.org. Julia. <http://www.voxforge.org/home/dev/acousticmodels/linux/create/htkjulius/tutorial/d>
- [23] Voxforge.org. Phonetics dictionary. <http://www.voxforge.org/home/dev/acousticmodels/linux/create/htkprep/step-2>.

-
- [24] Phil C Woodland, CJ Leggetter, JJ Odell, V Valtchev, and SJ Young. The development of the 1994 htk large vocabulary speech recognition system. In *Proceedings ARPA workshop on spoken language systems technology*, pages 104–109, 1995.
- [25] Philip C Woodland, Julian J Odell, Valtcho Valtchev, and Steve J Young. Large vocabulary continuous speech recognition using htk. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume 2, pages II–125. Ieee, 1994.
- [26] Min Xu, Ling-Yu Duan, Jianfei Cai, Liang-Tien Chia, Changsheng Xu, and Qi Tian. Hmm-based audio keyword generation. In *Pacific-Rim Conference on Multimedia*, pages 566–574. Springer, 2004.
- [27] Robert G Zick and Jon Olsen. Voice recognition software versus a traditional transcription service for physician charting in the ed. *The American journal of emergency medicine*, 19(4):295–298, 2001.

Appendix A

Appendix

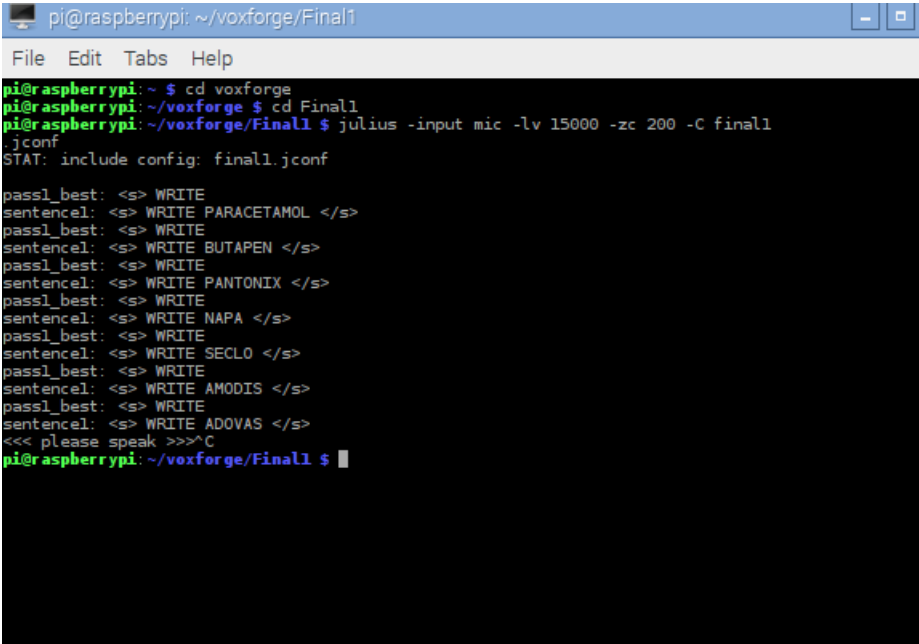
SL	Acronym	Full form
1	CSR	Continuous Speech Recognition
2	HMM	Hidden Markov Model
3	HTK	Hidden markov model Tool Kit
4	LVCSR	Large Vocabulary Continuous Speech Recognition
5	SR	Speech Recognition
6	NLP	Natural Language Processing
7	HCP	Health Care Professional
8	LM	Language Model
9	AM	Acoustic Model
10	MFCC	Mel Frequency Cepstral Coefficients
11	GMM	Gaussian Mixture Model
12	MLE	Maximum Likelihood Estimation
13	ASR	Automatic Speech Recognition
14	LPC	Linear Predictive Coding
15	GCC	GNU Compiler Collection

Table A.1: Appendix 1

Appendix B

A sample output

The following picture is a sample output screen taken from the system



```
pi@raspberrypi: ~/voxforge/Final1
File Edit Tabs Help
pi@raspberrypi: ~ $ cd voxforge
pi@raspberrypi: ~/voxforge $ cd Final1
pi@raspberrypi: ~/voxforge/Final1 $ julius -input mic -lv 15000 -zc 200 -C final1
.jconf
STAT: include config: final1.jconf

pass1_best: <s> WRITE
sentencel: <s> WRITE PARACETAMOL </s>
pass1_best: <s> WRITE
sentencel: <s> WRITE BUTAPEN </s>
pass1_best: <s> WRITE
sentencel: <s> WRITE PANTONIX </s>
pass1_best: <s> WRITE
sentencel: <s> WRITE NAPA </s>
pass1_best: <s> WRITE
sentencel: <s> WRITE SECL0 </s>
pass1_best: <s> WRITE
sentencel: <s> WRITE AMODIS </s>
pass1_best: <s> WRITE
sentencel: <s> WRITE ADOVAS </s>
<<< please speak >>>^C
pi@raspberrypi: ~/voxforge/Final1 $
```

Figure B.1: Sample Output