

Comparison of deep learning techniques for Sindhi language speech recognition



By

Muhammad Nawaz

00000203443

Supervisor

Dr. Muhammad Ali Tahir

Department of Computing

School of Electrical Engineering and Computer Science (SEECS)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

August 2021

Approval

It is certified that the contents and form of the thesis entitled "Comparison of deep learning techniques for Sindhi language speech recognition" submitted by MUHAMMAD NAWAZ have been found satisfactory for the requirement of the degree

Advisor : Dr. Muhammad Ali Tahir

Signature: 


Date: 12-Aug-2021

Committee Member 1:Dr. Asad Waqar Malik

Signature: 

12-Aug-2021

Committee Member 2:Dr. Yasir Faheem

Signature: 

Date: 12-Aug-2021

Signature: _____

Date: _____

Dedication

This thesis is dedicated to *my beloved parents, my beloved wife and my dearest newborn son who is the center of my universe*

Certificate of Originality

I hereby declare that this submission titled "Comparison of deep learning techniques for Sindhi language speech recognition" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEecs or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEecs or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: MUHAMMAD NAWAZ

Student Signature:  _____

Acknowledgments

I would like to express my sincerest gratitude to my supervisor Dr. Muhammad Ali Tahir and committee members (Dr. Asad Waqar Malik and Dr. Yasir Faheem) for their continuous support and guidance throughout my masters thesis.

Contents

1	Introduction	1
1.1	Motivation	2
1.1.1	Sindhi Speech Recognition in Keyboards	2
1.1.2	Sindhi Speech Recognition in Searches	3
1.1.3	Sindhi Speech Recognition in Home Assistants	4
1.1.4	Sindhi Speech Recognition in Speech Translation	4
1.2	Research Objective	4
1.3	Thesis Organization	5
2	Literature Review	7
2.1	Speech Recognition	7
2.1.1	Tools and techniques in Speech Recognition	7
2.1.2	Speech Recognition in Arabic language	8
2.1.3	Speech Recognition in South Asian languages	10
2.1.4	Limitations in Speech Recognition in Sindhi language and proposed solution	11
3	Methodology	12
3.1	Automatic Speech Recognition system	12
3.2	Feature Extraction	14
3.2.1	Mel Frequency Cespstral Coefficients	14

CONTENTS

3.3	Acoustic Model	15
3.3.1	Hidden Markov Models	16
3.3.2	Guassian Mixture Models	16
3.3.3	Deep Neural Networks	20
3.4	Language Model	22
3.5	Performance measure	23
3.6	Summary of Methodology	24
4	System Design	25
4.1	Kaldi	25
4.2	Dataset	26
4.3	Phonetic Dictionary	28
4.4	Training	28
4.5	Summary of System Design	29
5	Experiments and Results	30
5.1	Speaker dependent system	30
5.2	Speaker independent system	32
5.3	Training and testing data	34
5.3.1	Noise and speech speed in data	34
5.3.2	Training data size	36
5.3.3	Phonetic dictionary size	37
5.3.4	Extra language model data in training	38
5.3.5	Testing DNN parameters	40
5.4	Summary of Experiments and Results	41
6	Conclusion and Future work	42
6.1	Conclusion	42
6.2	Future work	42

List of Abbreviations and Symbols

Abbreviations

ASR	Automatic Speech Recognition
DL	Deep Learning
WER	Word Error Rate
HCI	Human Computer Interaction
DNN	Deep Neural Network
NNET	Neural Net
MFCC	Mel Frequency Cepstral Coefficients
PLP	Perceptual Linear Prediction
DFT	Discrete Fourier Transform

List of Tables

3.1	The Sindhi Alphabet	16
3.2	Diacritics in the Sindhi Alphabet	18
3.3	Phonemes in the Sindhi Alphabet	19
3.4	Word Error Rate Example	23
4.1	Sindhi ASR dataset	26
4.2	Kaldi audio file properties	27
5.1	WER on 4 test speakers in a speaker dependent system	31
5.2	WER on 15 test speakers in a speaker dependent system	31
5.3	WER on 15 test speakers in a speaker independent system	33
5.4	WER on ASR with noise in test data and ASR with noise in test and train data	35
5.5	WER on DNN with noise in test data and DNN with noise in test and train data	36
5.6	WER on GMM-HMM and DNN with 1 hour of test data and 11.5 hours of training data	36
5.7	WER on ASR with extra language model data in training data and ASR without extra language model data in training data	39
5.8	WER on DNN with corpus in training data and without corpus in training data	40
5.9	DNN word error rates with different number of hidden layers	40

5.10 DNN word error rates with different values of p in p-norm non linearity	40
--	----

List of Figures

1.1	Distribution of Pakistanis speaking Sindhi as a first language [1]	2
1.2	Google’s keyboard in Sindhi doing speech to text recognition in Roman Hindi [2]	3
1.3	Google keyboard in Sindhi doing speech to text recognition in English instead of Sindhi [2]	3
1.4	Speech to Speech translation with speech recognition in first phase.	5
2.1	A Hidden Markov Model for weather [3] [4]	8
2.2	Effects of dialect, gender and training data size in Arabic speech recognition [5][6]	9
3.1	ASR Architecture	13
3.2	Feature Extraction	14
3.3	HMM Model [7]	17
3.4	Guassian Mixture Model (GMM) distributions [8]	17
3.5	A GMM-HMM example [9]	18
3.6	An example of a Deep Neural Network [10]	20
3.7	A Neural Network with one hidden layer [11]	20
3.8	An example of a TDNN [12]	22
3.9	An example of an LSTM [13]	22
4.1	Example of a Kaldi directory structure for a recipe [14].	27

LIST OF FIGURES

5.1	Word error rates with 4 test speakers and 15 test speakers in a speaker dependent system	32
5.2	Comparison between speaker dependent and speaker independent system with 15 test speakers	33
5.3	Comparison between ASR with noisy data in test and ASR with noisy data in test and train	35
5.4	Word error rates of ASRs different train and test data size	37
5.5	Word error rates of ASRs with different phonetic dictionary size	38
5.6	Comparison between ASR with corpus in training data and ASR without corpus in training data	39

Abstract

With great technological advancements made in computational powers, Automatic Speech Recognition (ASR) systems have seen a surge in interest and usage. Much research has been done in ASR systems in languages like Chinese, English, Spanish, Korean or even in our national language Urdu, resulting in a better Human Computer Interaction (HCI). But there is a dearth of speech recognition systems done in regional and local languages like Sindhi. Over 30 million speakers of Sindhi Language in Pakistan are unable to communicate with a machine in Sindhi which is a great hurdle in utilizing the best of what technology has to offer. Automatic Speech Recognition (ASR) systems specifically built for local languages can help in overcoming these hurdles. In this study a speech recognition system for Sindhi language has been built with **Kaldi** toolkit. Hidden Markov Models (HMM) have been used along with Guassian Mixture Models (GMM) and Deep Neural Networks (DNN). Experiments have been conducted on GMM-HMM and DNN-HMM techniques regarding noise, training size, phonetic dictionary size and DNN parameters. DNNs were tested and compared using parameters such as value of p in p -norm non-linearity, number of hidden layers and learning rates. DNN with 6 hidden layers and $p=2$ gave best results. Accuracy of our speech recognition system is measured in Word Error Rate (WER). Experiments have been carried out on various speech recognition models and recipes for improved WER and results. These results could then be utilized in different areas like navigation, home automation etc. to increase HCI and usage of technology by Sindhi speakers.

Keywords: *Sindhi Speech Recognition, Kaldi, HCI, WER*

Introduction

An ASR system enables us to convert spoken utterances to easily readable text [15]. It acts as a communications bridge between computers and people. Earlier, due to limited power in computation, limited availability of data and limited research, there was not much interest in speech recognition. Preferable methods to communicate with machines were mouse etc. But recently with great technological advancements made in computational powers in various devices, Automatic Speech Recognition (ASR) systems have seen a surge in interest and usage [16][17]. Now a days we are completely relying on these in various ways. From instructing our devices to send out messages, connecting us with people through calls, providing us navigation in maps, playing music for us, to lighting bulbs, adjusting temperature and automating our home on a regular basis, we are using ASR systems everywhere.

There are over 30 million Sindhi speaking people living in different parts of Pakistan (See Fig. 1.1) But for these 30 million Sindhi speakers there is a communications gap with computers and technology. These people can not communicate with machines in their language. Beside the literacy rate in province of Sindh, where these people reside is very low[18]. So the only way to communicate with machines for these people is to either learn English or Urdu or be provided with a system enabling them interaction with machines in their own language. For this very purpose, we have developed a speech recognition system in Sindhi for these people to overcome this gap in communication. This will result in a better Human Computer Interaction for Sindhi speaking people with machines and they will able to utilize technology similarly like people of other languages.

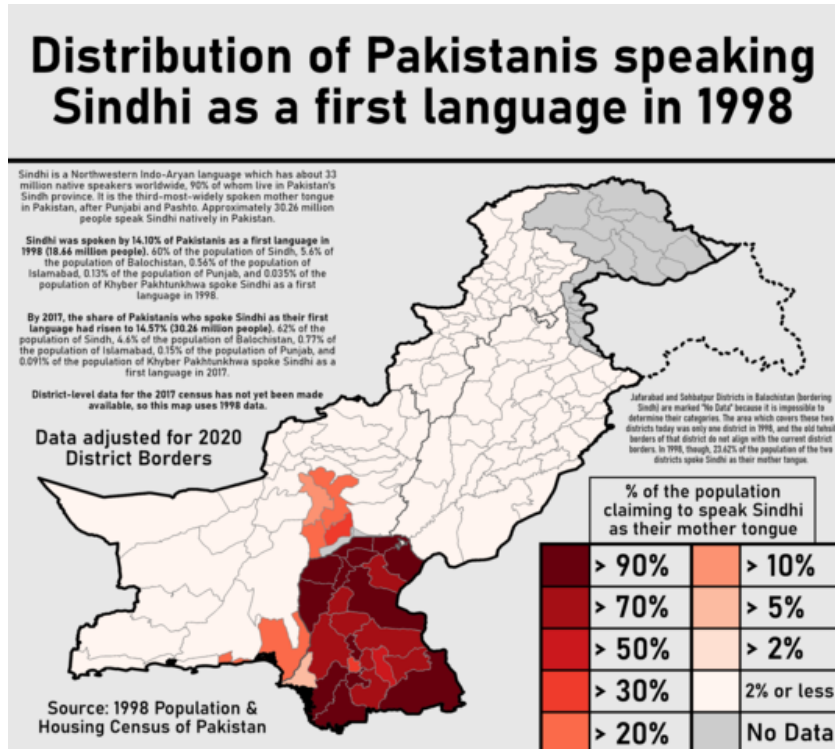


Figure 1.1: Distribution of Pakistanis speaking Sindhi as a first language [1]

1.1 Motivation

There are a number of useful applications of an automatic speech recognition system in Sindhi. It mostly helps in lessening the gap in human to machine interaction for over 30 million of people. Some of these useful applications are discussed below.

1.1.1 Sindhi Speech Recognition in Keyboards

There are various keyboards for writing in Sindhi. Google's Gboard is one of the popular and easy choices for this [2]. Gboard has built in facility to do speech to text conversion in English or in Roman Hindi (see Fig. 1.2).

But Google's Gboard in Sindhi does not convert speech to text when Sindhi words are uttered. It would convert familiar words either in English, Roman Hindi etc., if detected, but not in Sindhi (see Fig. 1.3).

Therefore, it would be really useful if Sindhi speech recognition is introduced in key-

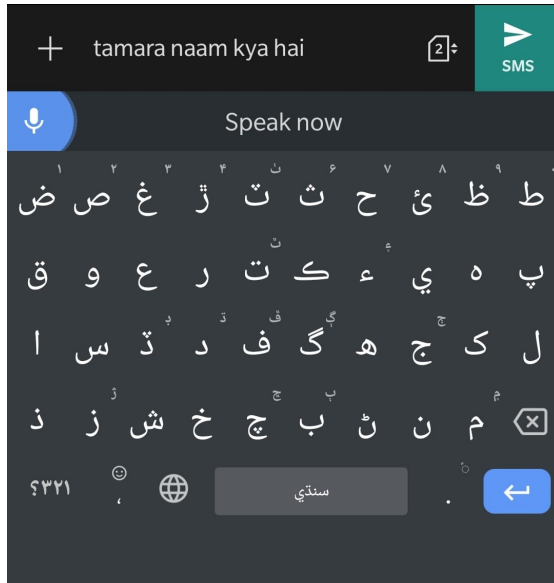


Figure 1.2: Google’s keyboard in Sindhi doing speech to text recognition in Roman Hindi [2]

boards. This would allow Sindhi speaking Diaspora to better utilize phones if they are not familiar in other languages.

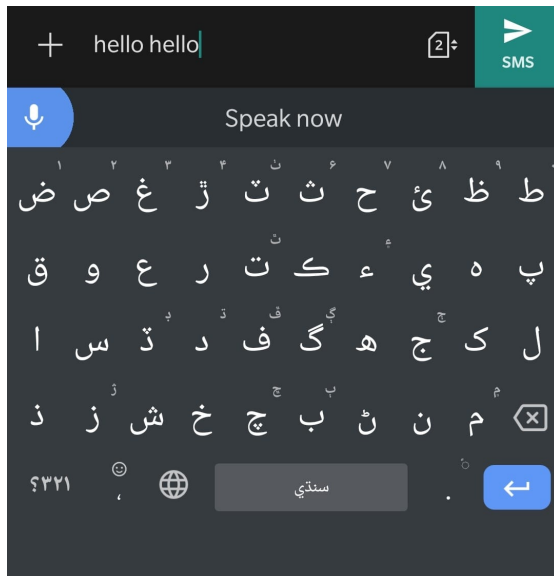


Figure 1.3: Google keyboard in Sindhi doing speech to text recognition in English instead of Sindhi [2]

1.1.2 Sindhi Speech Recognition in Searches

Another useful application of a Sindhi speech recognition system would be in performing various searches. Some of them are described below.

- Search by voice on search engines.
- Search music or videos by voice to listen to on streaming services
- Search shopping items by voice on e-commerce platforms
- Search locations by voice to navigate to with directions on map providing platforms
- Search hotels or places by voice to stay or rent
- Search food or food providers by voice on food-delivery platforms

1.1.3 Sindhi Speech Recognition in Home Assistants

Speech recognition has seen a boost in usage for home assistants and automation controls. Some of the applications of it in home assistants are shown below.

- Integration in voice assistants like Alexa, Google Home, Siri etc. [19]
- Controlling various home appliances like fans or adjusting their controls [20][21]
- Adjusting temperatures of enclosed spaces
- Turning on appliances like lights, microwave, washing machines etc.

Our Sindhi speech recognition system can also provide above benefits to Sindhi speaking people if integrated in home automation systems.

1.1.4 Sindhi Speech Recognition in Speech Translation

Another useful application of speech recognition is speech translation, it involves translating utterances of one spoken language, using machine learning, to another language between two speakers speaking different languages (see Fig. 1.4). Many corpus exist for multiple languages [22][23] for speech translation. But none exist incorporating Sindhi, our speech recognition system can help in this regard.

1.2 Research Objective

There is a lot of research being carried out in English in the field of speech Recognition. Recently, even local Pakistani languages like Urdu [24][25][26] and Pashto [27], have also

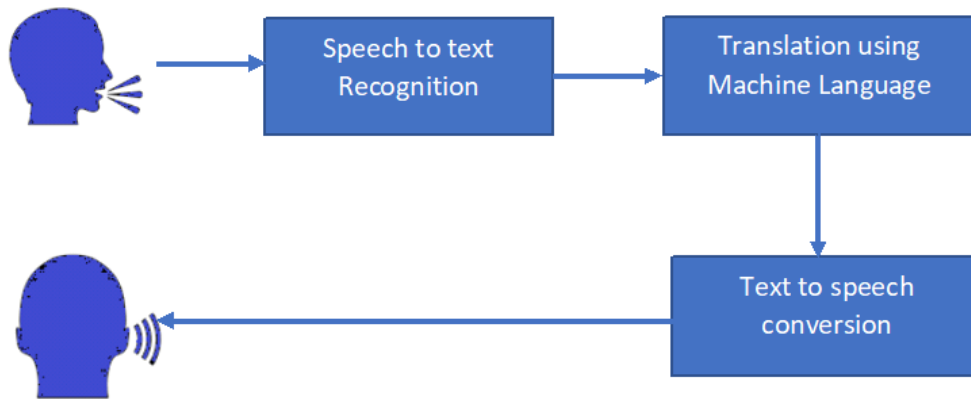


Figure 1.4: Speech to Speech translation with speech recognition in first phase.

seen an increase in research. But there is a dearth of speech recognition research being made in Sindhi.

This research aims to fill that gap by providing an automatic speech recognition system in Sindhi language using Kaldi toolkit. This study further aims to compare experimentation and results of models like Guassian Mixture Model - Hidden Markov Model (GMM-HMM) and Deep Neural Network (DNN).

1.3 Thesis Organization

This thesis is organized in following chapters.

- Chapter 2 provides a literature review of research carried out in speech recognition mainly in regional languages like Urdu, Sindhi, Pujabi, Saraiki, Gujrati, Pashto etc.
- Chapter 3 describes the methodology followed in speech recognition system built for Sindhi language.
- Chapter 4 lays out the system, design and architecture.
- Chapter 5 presents the results of various experiments and tests carried out on multiple models for this speech recognition system.

CHAPTER 1: INTRODUCTION

- And lastly, chapter 6 presents concluding remarks and any possible future work that can take the work of this study forward.

Literature Review

2.1 Speech Recognition

Automatic Speech Recognition has always been an area of interest, but due to limited computing power and resources, major breakthroughs had not been achieved. That changed after when Bell labs introduced Audrey [28][29] and IBM developed Shoebox [30], devices that could detect only spoken digits [31][32]. The Harpy [33] in 1970s could detect over a thousand words. But it was the rising popularity of Hidden Markov Models (HMM) [3] in early 1980s, that caused a shift to statistical processing from pattern recognition processing [32] causing improvements in accuracies achieved.

Since then, lately, due to advancements in processing powers [34] automatic speech recognition has seen a surge in research and usage. Now more and more people are communicating and interacting with machines using voice. Around 30 percent [35][36] of user interaction happens through some powerful speech recognition systems like Google Home, Alexa and Siri etc. These systems correctly recognize the spoken utterances around 95 percent of the time. [37][35].

2.1.1 Tools and techniques in Speech Recognition

There are various popular tools and softwares being used for the purpose of recognizing speech. Some popular ones are Dragon [38], Sphinx-4 [39] and Kaldi [40], which is an open source toolkit. Most of these perform excellent speech recognition in English and other major international languages. On local and regional languages research needs to be done to get remarkable results like those of English.

There are various techniques used in speech recognition. One such popular technique is of Hidden Markov Models (HMM) [3]. HMM shows a number of hidden states and the observations for those states. Given a probability a transition from one state to another or itself occurs and the result is the observation. Jason Eisner (2002) in his paper [4] gave an example (See Fig. 2.1) of a Hidden Markov Model to find the state of the weather (Hot or Cold) given the number of observations (ice creams eaten on a particular date).

HMM are used with combination of Gaussian Mixture Models (GMM) (See Fig. 3.4) and Deep Neural Networks (DNN) (See Fig 3.6). Input to the acoustic model in GMM-HMM technique is given through features extracted by Mel frequency cepstral coefficients (MFCC) or Perceptual linear predictive (PLP) coefficients [41]. Kaldi [40] offers various recipes and algorithms to use DNNs on top of GMM-HMM models for improving word error rates (WER). It uses p-normalized vectors along with Time delay neural networks (TDNN) and Long short-term memory (LSTM) to build DNNs.

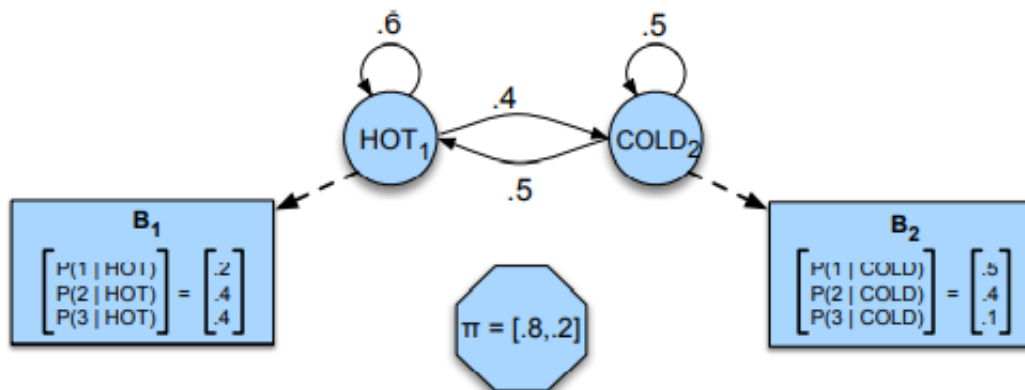


Figure 2.1: A Hidden Markov Model for weather [3] [4]

2.1.2 Speech Recognition in Arabic language

In automatic speech recognition, same words spoken by different people have different features due to various number of reasons. Bezoui et al. (2016) extracted features on same speeches by a number of speakers in Arabic language [42]. They extracted features of recitations of verses of the Holy Quran from multiple reciters. Even though the verses are same, each reciter will have different features in the sound. Differences would emerge because of the dialects, speed, pronunciation and other factors. They

stated that extracting features is a vital step for data preparation for classification. They used MFCC technique for that purpose.

An automatic speech recognition system is affected by multiple factors. Eiman et al (2020) explored a number of reasons that influence the quality and performance of a speech recognition system [6]. They discovered that gender, dialects, amount of training data, recording quality and variations in pronunciation all affect the performance of an ASR. Diacritics also play a role in speech recognition. Abed et al. (2019) found out that adding diacritics increased WER by 3.29% [43]. Even if they cause an uptick in WER, they recommended to add them in ASR because they could be helpful if the results of ASR system are fed to speech to text systems for translation.

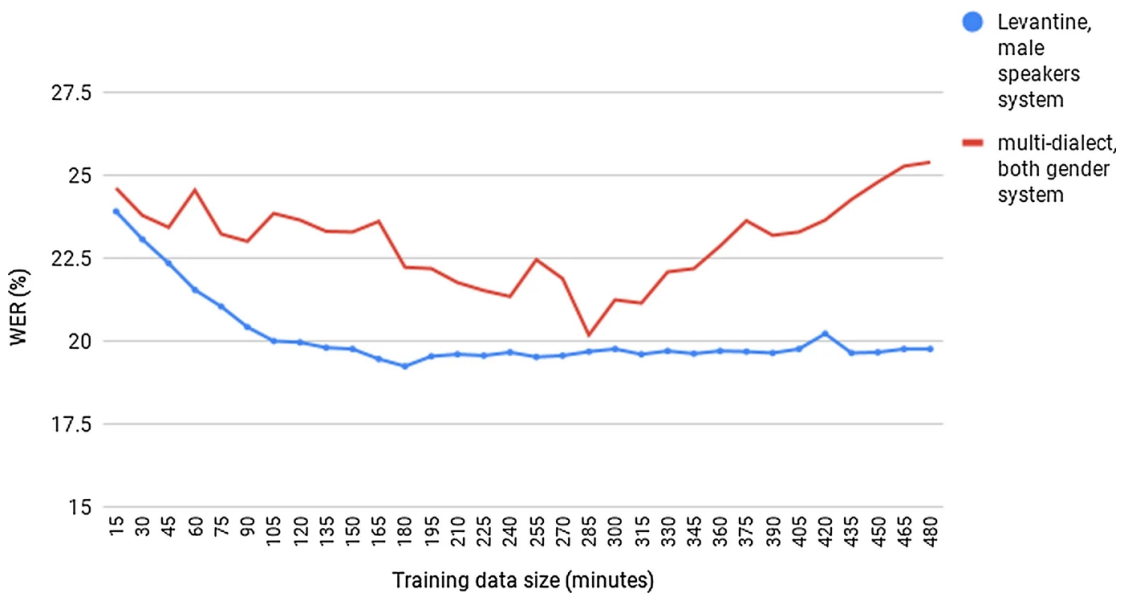


Figure 2.2: Effects of dialect, gender and training data size in Arabic speech recognition [5][6]

Noise also affects the quality and performance of a speech recognition systems. Ouisaadane et al. (2021) compared the effects of noisy environments on an Arabic speech recognition system [44]. They used HMM with DNN and Gaussian Mixture Models (GMM) separately using CMU Sphinx and Kaldi on twenty isolated words and extracted features through MFCC.

2.1.3 Speech Recognition in South Asian languages

On Hindi, an ASR on a corpus of around 1000 sentences was developed by Upadhyaya et al. (2017) [45]. The authors found out that performance of a model trained on triphone (tri) n-grams produced better word error rate than monograms (mono). A similar research [46] was conducted by Guglani et al. (2018) who presented a speech recognition system in Punjabi using Kaldi toolkit. They extracted features from Punjabi speech and tested the performance on n-grams. They reported a noteworthy decrease in word error rate (WER) when triphones2 (tri2) n-grams were used over monograms. Consequently WER was further reduced if triphone3 (tri3) n-grams were used over monograms. They used two techniques for feature extraction. The first technique Mel frequency cepstral coefficients (MFCC) produced much better accuracy than second technique Perceptual linear prediction (PLP).

Kumar et al. (2017) built an ASR system for Punjabi [47]. They trained their model on over 6000 words and over 1400 sentences. They achieved an accuracy of 93% on Punjabi words. Bhardwaj et al. (2020) developed a corpus of children speakers in Punjabi and build an ASR on it using Deep Neural Networks (DNN) in Kaldi toolkit [48]. They achieved an accuracy of 87% in the system. Tailor et al. (2018) built an ASR for Gujrati language using Hidden Markov Models (HMM) [49]. They trained the system on 40 speakers on 650 utterances and achieved 12.7% word error rate (WER).

Humayun et al (2019) trained an Urdu speech recognition system using Kaldi on 100 hours of data with test data WER of 9% [24]. Farooq et al. (2019) developed a speech recognition system for Urdu [26]. They trained the model with 1671 Punjabi and Urdu speakers on 300 hours of data. Using various techniques for acoustic modeling, they achieved 13.5% WER.

Syed et al. (2020) developed a speech emotion recognition system on Urdu and Sindhi [50]. Their corpus consisted of 1435 utterances in Urdu and Sindhi. They measured the performance of the system in unadjusted average recall (UAR). On test partitions in Urdu and Sindhi, their UAR was 56.96% and 55.29% respectively.

2.1.4 Limitations in Speech Recognition in Sindhi language and proposed solution

There is only a small amount of research done in speech recognition for Sindhi. Hashmi et al. (2019) created an ASR using HMM and Artificial Neural Networks (ANN) [51]. They trained 100 words on 10 speakers, evenly distributed between male and female speakers. They used CMU Sphinx as ASR toolkit and MFCC for extracting features. Accuracies achieved were in range of 81-97 percent for each speaker. Speakers with higher training data size, in their findings, had higher accuracies. Even though their accuracies were high, the dataset had a limited vocabulary of 100 words. The lack of good datasets is a very big limitation in speech recognition for Sindhi.

Sindhi language has seen very little research in the area of speech recognition due to a number of reasons. The population is declining with children not speaking their mother tongue. The urbanization has led to children speaking English or Urdu language. On top of it Sindhi is a very difficult language to write as it contains 52 letters and every letter can be written differently depending on the position in word and most of the population prefers to write Roman Sindhi. All these factors have contributed to the lack of datasets and research for Sindhi ASR. This thesis aims to reduce this limitation by providing an automated speech recognition in Sindhi with a dataset of 13 hours of speech data with their transcriptions, more than 50 speakers and a phonetic dictionary of over 10,000 words.

In this chapter the literature review and the progress made in area of automatic speech recognition has been discussed, followed by the problem and motivations that led to creation of an ASR for Sindhi. In the following chapter technical methodologies of our ASR, given contributions and a general overview of proposed approach to creating this ASR will be briefly discussed.

Methodology

In Sindh, the literacy rate is very low, therefore interaction with machines in English or Urdu is not possible by majority of population. Interaction with machines by these people can only happen via voice based interfaces in Sindhi through an automated speech recognition system. But the educated population in the province prefers to communicate with machines in English or Urdu, this has led to a serious lack of research in Sindhi speech recognition. The small research that has been carried out has been on very small datasets with severely limited vocabulary set and few speakers resulting in poor results in real life scenarios.

These problems have motivated us to creating an ASR specifically designed for Sindhi language. In this chapter we present the methodologies, technicalities and the approach we have used in creating our automated speech recognition system for Sindhi language. ASR for Sindhi language in this study has been developed by creating an acoustic and language model specific to Sindhi language. This thesis has explored and tuned multiple acoustic parameters like HMM-states or leaves, number of hidden layers and dimensions in a DNN, value of p in p-norm non-linearity along with TDNN and LSTMs as well as tuning language model to get good results in the form of word error rate. These details are discussed in below sections.

3.1 Automatic Speech Recognition system

An automatic speech recognition system (See Fig 3.1) takes as input the speech signal received, performs acoustic analysis by extracting features, decodes the spoken signals

in recognition system using acoustic and language model parameters and produces text of spoken utterances as output.

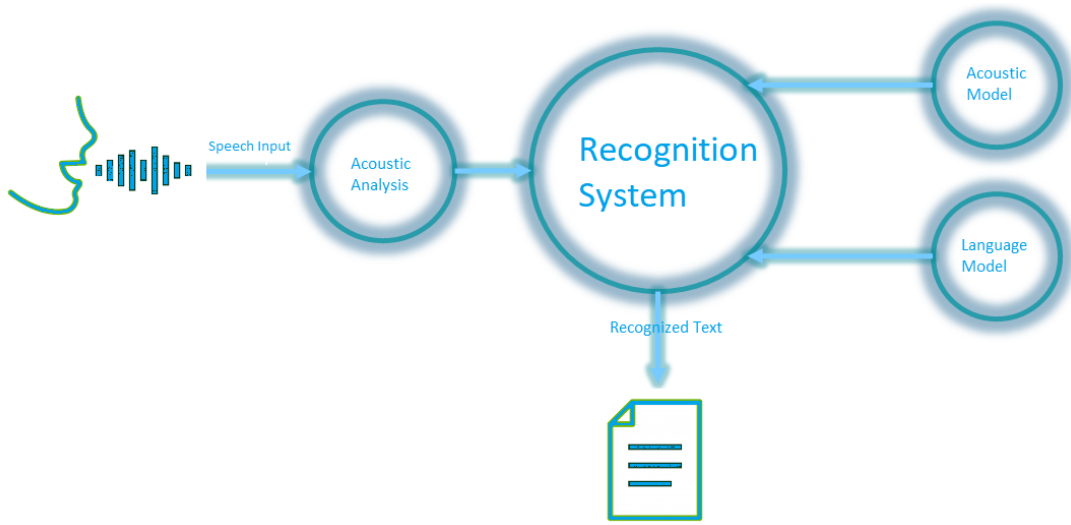


Figure 3.1: ASR Architecture

The main objective of an ASR is to deduce the probability of a sequence of words given a features sequence [7]. The given feature vector $X = \{x_1, x_2, x_3, \dots, x_n\}$ produces a sequence of words $W = \{w_1, w_2, w_3, \dots, w_n\}$. This is mathematically shown in Equation 3.1.1.

$$W^* = \arg_W \max P(W|X) \quad (3.1.1)$$

Equation 3.1.2 is the result when Naive Bayes is applied on Equation 3.1.1.

$$W^* = \arg_W \max P(X|W) P(W) / P(X) \quad (3.1.2)$$

We can discard probability $P(X)$, because it does not change w.r.t to given sequence of words W . This is mathematically shown in Equation 3.1.3.

$$W^* = \arg_W \max P(X|W) P(W) \quad (3.1.3)$$

The expression $P(X|W)$ in equation 3.1.3 is the acoustic model and expression $P(W)$ is

the language model. Equation 3.1.3 is the maximum likelihood of sequence of words W , given features sequence X .

An acoustic model for Sindhi ASR requires features as input extracted in the process of feature extraction which is discussed below.

3.2 Feature Extraction

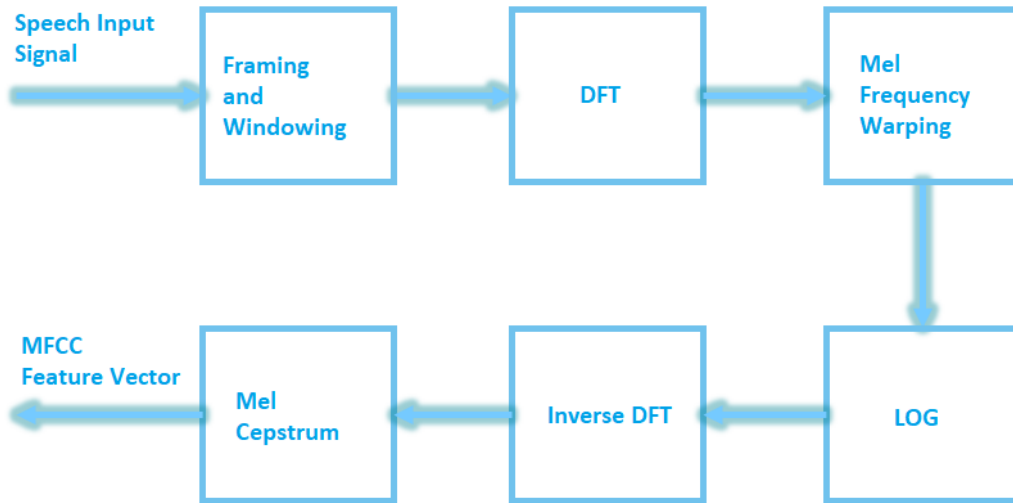


Figure 3.2: Feature Extraction

For Feature extraction, Mel Frequency Cespstral Coefficients (MFCC) [52] are used. MFCC is one of the most popular feature extraction technique currently in use. Mimicking the behaviour of a human ear is one of the reasons for the popularity of MFCC. The steps involved MFCC feature extraction are shown in 3.2.

3.2.1 Mel Frequency Cespstral Coefficients

- In the first step, number of frames in the incoming speech signal are worked out. Each frame is 25 millisecond in length and is then shifted by 10 milliseconds. Afterwards, a windowing function is multiplied with every frame. There are various windowing functions but commonly used one is Hamming window (see Equation 3.2.1).

$$w(n) = (1 - \alpha) \alpha \cos \frac{2\pi n}{L - 1} \quad 0 \leq n \leq L - 1, \quad \alpha = 0.46164, \quad L = \text{window width} \quad (3.2.1)$$

- In next step Discrete Fourier Transform (DFT) is applied
- Afterwards incoming speech signal's frequencies re changed to Mel frequencies.
- Then the log is taken of the output.
- After which Inverse DFT is performed.
- In the last step Mel cepstrals are produced.

3.3 Acoustic Model

In automatic speech recognition, an acoustic model predicts $P(X|W)$. It is the probability of the sequence of the acoustic features given the words sequence. There is a large data set of vocabulary of words involved in speech recognition, therefore a smaller unit of words, phenome, is considered. N-phone modeling, which is mono phone and triphones, is commonly used in speech recognition. Every word contains a phones' sequence. Phones are used instead of words in speech recognition because there are fewer unique phones in a language than unique words. Training on triphone models gives better results than training on mono phone models [46], because of triphones taking into account the context.

Acoustic vector of same phenome even by same speaker is not always same [42]. Further variability is added because time of when words will be uttered is not known. To address this variability in acoustic features Hidden Markov Models (HMM) are used. After HMM, GMM or DNN is applied to find performance of HMM states in acoustic features.

The Sindhi language contains 52 letters in its alphabet (see Tab 3.1) and 62 phenomes (see Tab 3.3). Table 3.2) shows the diacritics in Sindhi. The phenomes can have 2 to 4 shapes depending on the position of it in the word.

ا	ب	پ	پ	ت	ت	ت
ن	ث	پ	ج	ج	جھ	چ
چ	چ	ح	خ	د	د	ڈ
ب	د	ذ	ر	ر	ز	س
ش	ص	ض	ط	ظ	ع	غ
ف	ق	ق	ک	ک	گ	گ
گھ	نگ	ل	م	ن	ٹ	و
				ھ	ء	ي

Table 3.1: The Sindhi Alphabet

3.3.1 Hidden Markov Models

A model of HMM (See Fig. 3.3) consists of a chain of acoustic observables and states that are hidden. Only the observable of each underlying state can be seen. A transition probability in Hidden Markov Model is the probability to switch from one state to other. An emission probability in HMM is one which is used to estimate the observation.

For estimation of observation distributions in speech recognition, two techniques have been used in this research.

- GMM
- DNN

Both of the above techniques are described in sections below.

3.3.2 Gaussian Mixture Models

Gaussian Mixture Model (GMM) is used to find distribution of observables in a HMM states model. Jointly with HMM, Gaussian Mixture Model produces an acoustic model that produces good results. A **GMM-HMM** model is also used later on, if a DNN is to be trained.

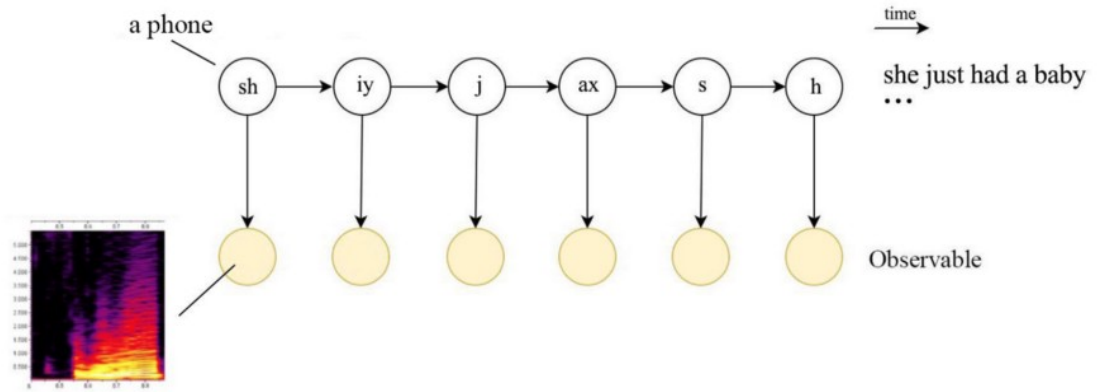


Figure 3.3: HMM Model [7]

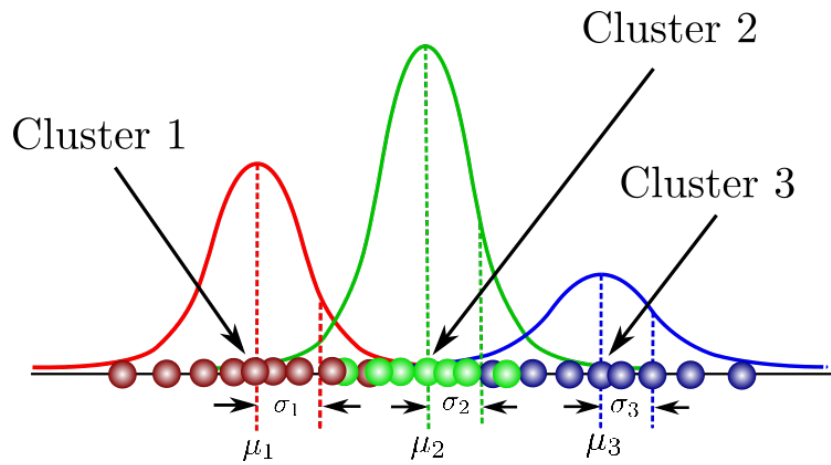


Figure 3.4: Gaussian Mixture Model (GMM) distributions [8]

a	اَ	e	اِ	i	اِي	o	اُو
u	اُ	A	آَ	y	اِيٓ	I	اِيٓٓ
0	و	U	وُ	ay	اِيٓٓ	ao	وِو
i	آِ	_A	اِيٓٓ	'A	آِٓ	'a	آِٓٓ
'i	آِٓ	'y	اِيٓٓٓ	'w	آِٓٓ	'a	آِٓٓٓ

Table 3.2: Diacritics in the Sindhi Alphabet

In a GMM-HMM model (see Fig. 3.5), each emitting function $b(o)$ of observation o is a gaussian distribution mathematically represented as.

$$b_i(o) = \sum_{m=1}^M c_{im} \mathcal{N}(o; \mu_{im}, \Sigma_{im}) \quad (3.3.1)$$

$\mathcal{N}(\mu_{im}, \Sigma_{im})$ is the normal distribution for multivariate o , c_{im} is the weight for gaussian distribution m and state i . The equation 3.3.1 is the likelihood of observation o for state i in a GMM-HMM model.

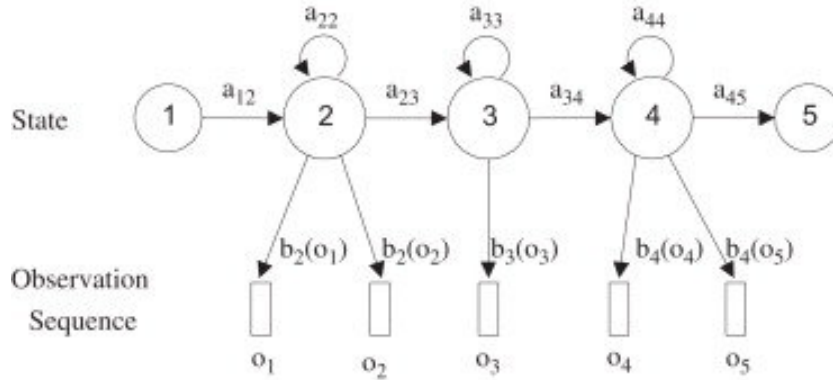


Figure 3.5: A GMM-HMM example [9]

In our thesis we have used 2000 HMM-states or leaves. Each state represents a phenome. Since a phenome could be in start, middle or end of a word it can represent 3 states. Also a phenome can sound differently in different words therefore 2000 HMM-states have been used. Beside HMM-states, 10000 GMM distributions have been used.

a	اَ	â	ج	z	ز	kh	ک
b	ب	^c	چ	s	س	g	گ
:b	بَ	^ch	چ	^s	ش	:g	گپ
bh	پ	.h	ح	.s	ص	gh	گھ
t	ت	_h	خ	.d	ض	:n	نگ
th	ث	d	د	.t	ط	l	ل
,t	تَ	dh	ڈ	.z	ظ	m	م
,th	ثَ	:d	ڈ	'	ع	n	ن
_s	ث	,d	د	.g	غ	,n	ن
p	پ	,dh	د	f	ف	w	و
j	ج	_d	ذ	ph	ڦ	h	ھ
:j	جَ	r	ر	q	ق	'a	ء
jh	جھ	,r	ڙ	k	ڪ	y	ي
mh	مھ	,rh	ڙھ	,nh	ڻھ	nh	نھ
lh	لھ	a	اَ	i	اِ	u	اُ
ay	اِي	ao	او				

Table 3.3: Phonemes in the Sindhi Alphabet

3.3.3 Deep Neural Networks

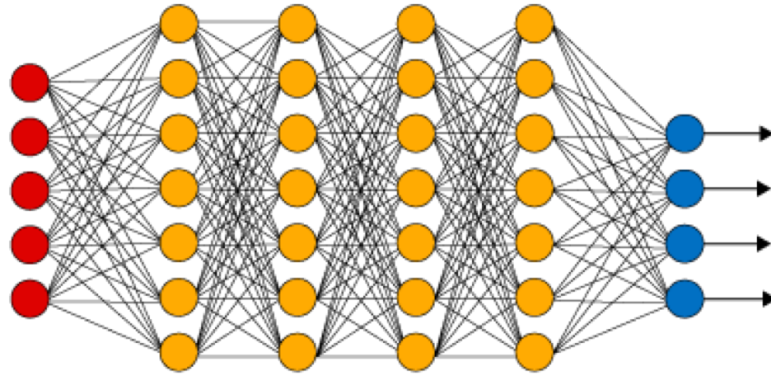


Figure 3.6: An example of a Deep Neural Network [10]

Deep Neural Networks, **in this study**, are built on top of GMM-HMM. A typical DNN (See Fig. 3.6) consists of an input and output layer along with two or more hidden layers in between them. One of the main reasons about the better result produced by a DNN model, is the ability of it to self learn through back propagation.

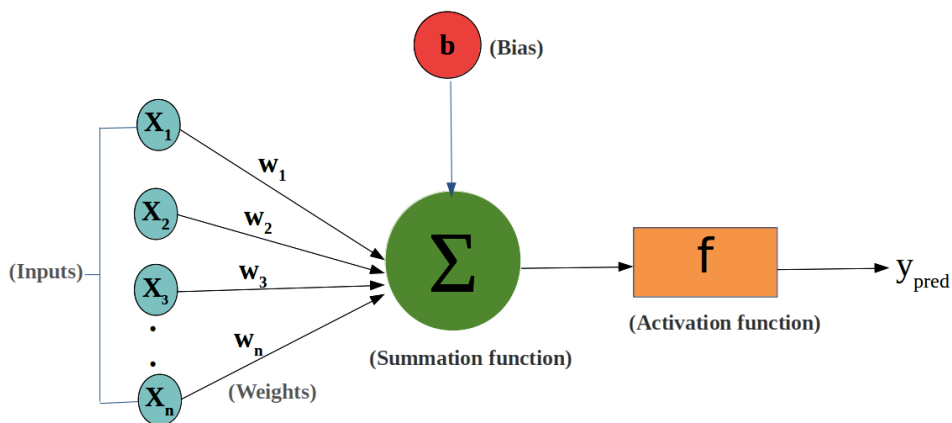


Figure 3.7: A Neural Network with one hidden layer [11]

A typical DNN hidden unit is called a neuron. It applies an activation function which could either be a logistic sigmoid function (see Eq. 3.3.2), a hyperbolic tangent function (see Eq. 3.3.3) or any other function.

$$y_j = \sigma(x) = \frac{1}{1 + e^{x_j}} \quad (3.3.2)$$

$$y_j = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.3.3)$$

Figure 3.7, is one hidden layer feed forward neural network called perceptron. Each input layer value x_j is multiplied by a weight w_{ij} and the result is added with a bias b_j . This is mathematically represented in Equation 3.3.4

$$x_j = b_j + \sum_i y_i w_{ij} \quad (3.3.4)$$

$$\|x\|_p = \left(\sum_{p=1}^n |x|^p \right)^{\frac{1}{p}} \quad (3.3.5)$$

In our thesis we tested and compared the results of multiple hidden layers on which DNNs were trained. We tested 2, 3, 4, 5 and 6 number of hidden layers and found a DNN with 6 hidden layers performed best. We have used ***p-norm*** non-linearity (see Eq. 3.3.5) as an activation function. We test 2, 3 and 5 values for p and found $p=2$ gave the best results. These have been used along with Time delay neural networks (TDNN) and Long short-term memory (LSTM) to build DNN. Both of these are discussed below

- ***Time delay neural networks (TDNN)***: Context in a speech recognition system is very important. A TDNN models the context of past events. It follows a feed forward model and encompasses delays as shown in figure 3.9.

If there are n number of inputs and m number of delays, then a total of $n(m + 1)$ computations are needed for a weighted sum. So a TDNN with 5 inputs with 3 delays would require 20 computations. Mostly *sigmoid* is used for the purpose of activation function.

- ***Long short-term memory (LSTM)***: LSTM is a type of recurrent neural network (RNN), which solves the vanishing gradient problem of RNN. The hidden

unit in LSTM is called a memory block. It contains multiple gates for different purposes.

Input gate is responsible for maintaining the flow of input activations and output gate is responsible for maintaining the activations of output of memory cell to the rest of system. The forget gate either forgets the value or refreshes it. Figure ?? shows an example of an LSTM memory block.

3.4 Language Model

Expression $P(W)$ in equation 3.1.3 is the language model. It is the maximum likelihood that the sequence of words W will occur, given features sequence X . N-gram model finds

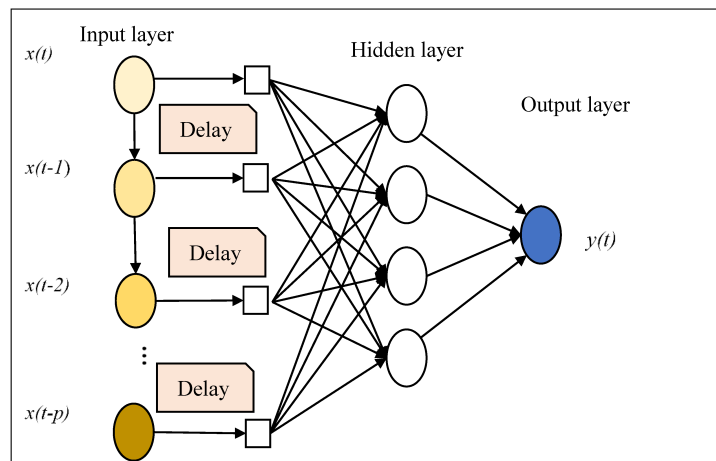


Figure 3.8: An example of a TDNN [12]

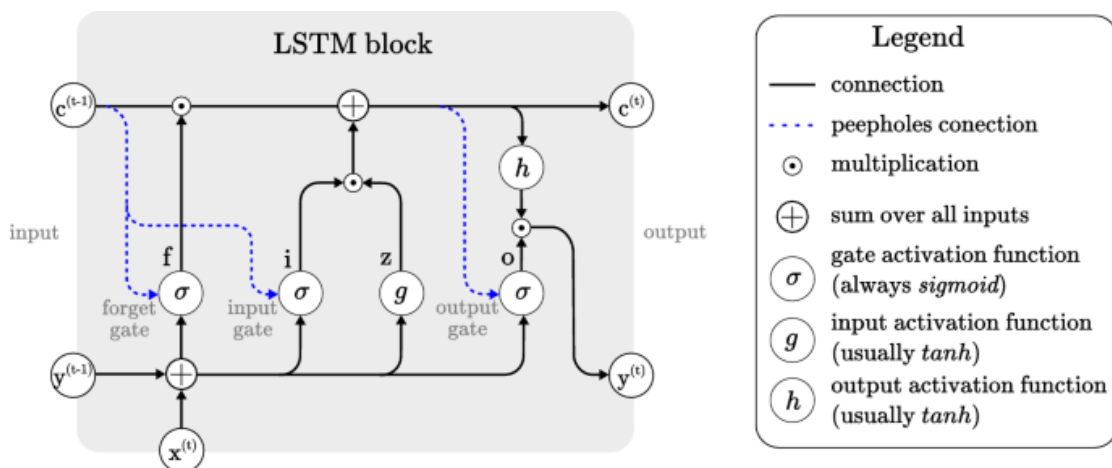


Figure 3.9: An example of an LSTM [13]

the probability of occurrence of a word given previous N-1 words. In an N-gram model, unigrams, digrams and trigrams are used. Unigram is not context dependent as it does not take into account the probability of current word given previous words. Diagram finds likelihood of occurrence of a pair of words and takes into account probability of one previous word to find the probability of current word. Trigram finds likelihood of occurrence of a tuple of words and takes into account probability of two previous words to find the probability of current word.

For a good ASR a good language model is necessary. In this thesis we tuned our language model by manually transcribing the speech audio. Phonetic dictionary was tested with various sizes and improved to its final shape containing over 10,000 words with their phenomes. In our thesis we also added a large extra language model data containing Sindhi text in our language model for training our ASR if an unknown word is encountered which is not present in phonetic dictionary.

3.5 Performance measure

Word Error Rate (WER) is the unit measurement of performance in this study. It is calculated by dividing total errors with all the words.

$$WER = \frac{Substitutions + Deletions + Insertions}{N} \quad (3.5.1)$$

Where $N = Substitutions + Deletions + Correct\ words$.

Reference	تُهَنجُو نَالُو چَا آهي
Hypothesis	تُهَنجُو نَالُو * آهي
Labels	C D C C

Table 3.4: Word Error Rate Example

In the example above there are 3 corrections and one deletion. Therefore WER produced is $(0 + 1 + 0) / (0 + 1 + 3) = 1 / 4 = 0.25\%$.

3.6 Summary of Methodology

The performance of an ASR depends on various factors. A good acoustic model with a badly transcribed language model is always going to perform poorly. Similarly good acoustic and language models with an uneven train-test split or too little training may overfit or underfit the results.

In this chapter, technicalities, methodologies and approaches used were discussed in creating models for our ASR. Further discussion was carried out on tuning parameters like HMM-states, number of hidden layers, value of p in p-norm non-linearity activation function for DNN and the phonetic dictionary size used in this thesis. The effects of these parameters are discussed in chapter 5. In the next chapter the overall design and architecture of proposed system and various algorithms used for training the ASR will be discussed.

System Design

In previous chapter the methodologies and technicalities of our proposed ASR were discussed. In this chapter, a brief overview of the system architecture and design of the ASR built is given. The architecture and design consists of toolkit and dataset used and the algorithms used in training. A brief overview of Kaldi toolkit, dataset and training and testing procedures is given in sections below.

4.1 Kaldi

Kaldi, a popular open source automatic speech recognition toolkit, is used in this research for speech recognition. The reasons for which it has been used is its features, flexibility, support, extensibility and a vast pool of ready made recipes. Some of the benefits Kaldi offers are mentioned below.

- ***Finite State Transducers (FST)***: FST allow to work on two tapes simultaneously. It can produce an output on a tape while simultaneously reads an input tape as well. FST gives state sequence probabilities in Kaldi using language model or lexicon. FST support is provided by ***OpenFST***.
- Matrix library support for linear algebraic calculations.
- Complete recipes available than can be modified or extended according to the needs of an ASR.
- Extensive support of Graphical Processing Units (GPU).

A typical Kaldi recipe lies in *egs* directory. An ASR is built either from scratch or on top of an already built recipe. Figure 4.1 shows the directory structure of a Kaldi recipe. Kaldi uses of Hidden Markov Models (HMM) [3] to estimate the probability of occurrence of an state. HMM shows a number of hidden states, transitions between those states and the observations when a transition occurs. Given a probability a transition from one state to another or itself occurs and the result is the observation.

Hidden Markov Models (HMM) are used with combination of **Gaussian Mixture Models (GMM)** (See Fig. 3.4) and **Deep Neural Networks (DNN)** (See Fig 3.6). Acoustic features are extracted through Mel frequency cepstral coefficients (MFCC) or Perceptual linear predictive (PLP) coefficients [41] and given as input to the acoustic models for GMM-HMM models.

Kaldi [40] has many built in examples, recipes and algorithms to design Deep Neural Networks (DNN) on top of GMM-HMM models for improving word error rates (WER). It uses p-normalized vectors along with Time delay neural networks (TDNN) and Long short-term memory (LSTM) to build DNNs with learning rates usually between 0.001 and 0.0001. The parameters can be tuned to improve results.

4.2 Dataset

To train an ASR for Sindhi speech recognition, a baseline dataset [53] by Kalhoru et al. (2020) has been used. More speakers and recordings on top of the baseline dataset have been added. The final dataset consists of around 13 hours of speech data with 61 speakers.

TOTAL SPEAKERS	MALE SPEAKERS	FEMALE SPEAKERS	DURATION	SIZE
53	41	12	13 Hours	4 GB

Table 4.1: Sindhi ASR dataset

In the dataset, the audio is generated by multiple speakers by recording text of newspaper archives. Afterwards corresponding transcript files are generated of each recording. Audio files are then converted to specific format with specific properties required by Kaldi (see Tab).

Each recording is then split into multiple smaller files of length around 10 seconds to a

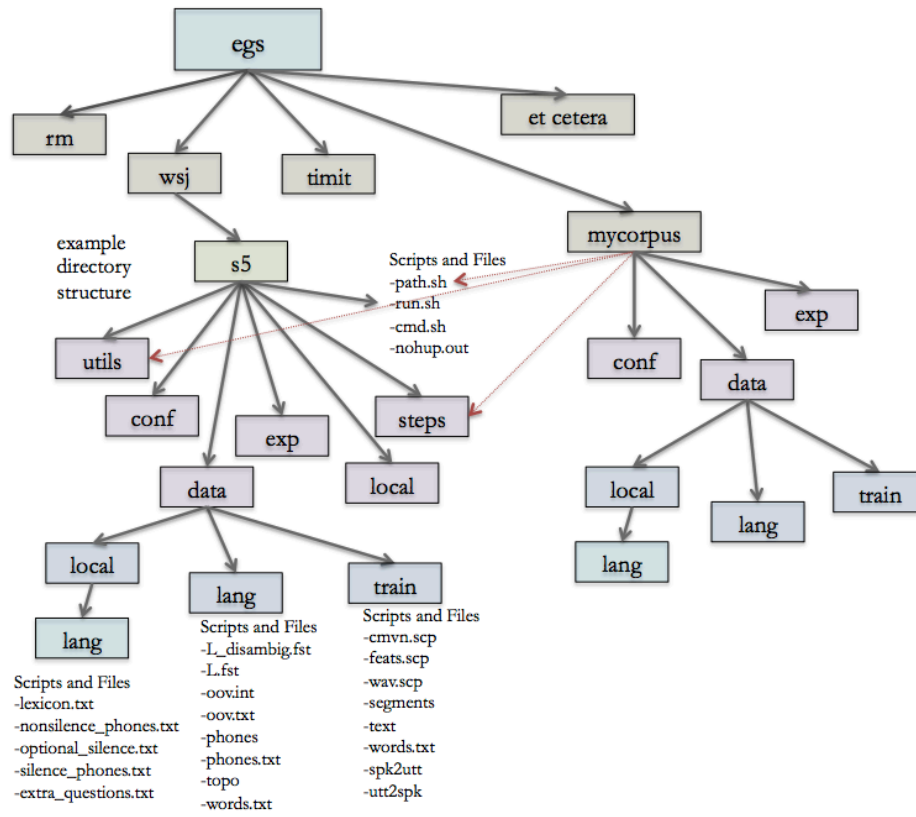


Figure 4.1: Example of a Kaldi directory structure for a recipe [14].

PROPERTY NAME	PROPERTY VALUE
Format	Wave
Channel	Mono
Bit depth	16 its
Bit rate	256 kb/sec
Sampling rate	16.0 kHz

Table 4.2: Kaldi audio file properties

minute. For every small audio file a separate transcript file is then created.

4.3 Phonetic Dictionary

The phonetic or pronunciation dictionary consists of phonetics mapped against a word in Sindhi. Every pronunciation for a word is written on a separate line in the dictionary. Initially one fifth of the pronunciation of words is manually created. These phonetics are later used to train a *grapheme-to-phoneme* (*g2p*) model which predicts the phonemes of the rest of the words. If the *g2p* model is trained again and again, a significant reduction in WER is observed.

In this thesis a *g2p* model of around 1860 words was initially created and used to train GMM-HMM and DNN models. Later a model of around 5000 words was created giving a significant better performance. Finally a model of 10,000 words was created which produced a much better WER than previous models.

4.4 Training

To train an ASR on a particular language, Kaldi provides various built-in recipes which can be used. Most of the recipes have a *run.sh* bash script that can be used as is or modified to use as per the requirements. Different recipes work on different languages. Since there is no Sindhi language specific recipe in Kaldi, therefore the *voxforge* recipe on English language is used for our ASR for Sindhi language. It was modified to work on Sindhi language by providing Kaldi specific Sindhi recordings, a Sindhi phonetic dictionary and a *g2p* model built on the dictionary.

For training the acoustic model two techniques, GMM-HMM and DNN on GMM-HMM, have been used in this research. These techniques are then used for estimation of observation distributions in HMM. Following are the steps in training.

- MFCC features are extracted using *make_mfcc.sh* and *compute_cmvn_stats.sh* scripts in the `steps` directory in *wsj* recipe 4.1.
- A **mono** phone model is trained using *train_mono.sh* in the `steps` directory in *wsj* recipe.

- Train first triphone model **tri1** using *train_deltas.sh* script in steps directory in *wsj* recipe.
- Train $\Delta + \Delta \Delta$ triphone model **tri2a** using *train_deltas.sh* script in steps directory in *wsj* recipe.
- Train LDA+MLTT **tri2b** triphone model using *train_lda_mllt.sh* script in steps directory in *wsj* recipe.
- Train LDA+MLTT+SAT **tri3b** triphone model using *train_sat.sh* script in steps directory in *wsj* recipe.
- Test results, in the form of WERs, are extracted using *decode.sh* script in steps directory in *wsj* recipe.

Above steps were for training and testing a GMM-HMM model. A DNN is built on top of this using **tri3b** triphone model. Steps involved are mentioned in below.

- Train **nnet2** neural network using *run_nnet2.sh* in *local/online* directory in *rm* recipe.
- **p-norm** non-linearity has been used as an activation function. Tangent hyperbolic functiona was also used, but p-norm with $p=2$ performed with better results.

4.5 Summary of System Design

The result for testing, in the form of WERs, are extracted using *decode.sh* script in steps directory in *wsj* recipe. In this study it was observed that increasing the training dataset along with increasing and improving phonetic dictionary and transcriptions gives better results. The value of number of hidden layers and p in p-norm non-linearity for DNNs was also tested with different parameters. It was tested that a DNN on top of a good trained **tri3b** with *LDA+MLTT+SAT* GMM-HMM model performs better than other triphone models. The various experiments conducted are discussed in the next chapter.

Experiments and Results

In this thesis, experiments on GMM-HMM and DNNs were conducted and their results were compared. Experiments performed included training and testing results regarding speaker dependent and independent systems, number of test and train speakers, phonetic dictionary size, ideal data in controlled environment, real time data with faster speeds and noise on GMM-HMM models and tuning and changing parameters in DNN models. Experiments conducted and their results are discussed in sections below.

5.1 Speaker dependent system

In a speaker dependent system, the test speakers are prefixed and chosen. The data in predefined test speakers could be clean and lab controlled, ideal with slow speech and containing minimal noise or noisy and real world data. The reason for choosing predefined test speakers could be to get good results on clean data or test results on how noise and speed behave.

A very basic GMM-HMM ASR was trained with 4 predefined test speakers in which data was of ideally controlled environment with no noise and normal or slower speeds to produce good results. It was trained on 41 speakers on a phonetic dictionary of around 1860 words. The highest WER achieved in this experiment was **23.74%** on **tri3b_fmml_c** triphone model and **27.6%** on **tri3b** triphone model. Table 5.1 shows all the best WERs of each model in this experiment.

The reason for good WER in above experiment was the small amount of test data size and data recorded in a controlled environment with minimal noise and clear speech.

MODEL	WER
mono	42.73%
tri1	27.15%
tri2a	26.56%
tri2b	25.96%
tri3b	26.7%
tri3b_mmi	26.11%
tri3b_mmi_c	23.74%

Table 5.1: WER on 4 test speakers in a speaker dependent system

To test an ASR containing a mix of ideally controlled environment with no noise and normal or slower speeds as well as some noisy data with faster speeds another experiment was conducted. The GMM-HMM ASR was tested on 15 predefined test speakers and trained on 30 speakers with a phonetic dictionary of around 1860 words. The highest WER achieved in this experiment was **40.35%** on **sgmm2_4a** and **44.43%** on **tri3b** triphone model. The reason for high word error rates in this ASR was the inclusion of some noisy data.

MODEL	WER
mono	73.28%
tri1	54.54%
tri2a	52.79%
tri2b	50.64%
tri3b	44.43%
tri3b_mmi	44.32%
tri3b_mmi_c	44.20%
sgmm2_4a	40.35%

Table 5.2: WER on 15 test speakers in a speaker dependent system

Word error rates in above experiment are high due to a number of reasons. Training data was reduced in this experiment from 41 speakers to 30 speakers while testing data was increased to 15 speakers from 4 speakers. The data in test contained noise and faster speeds as well while data in train lacked that.

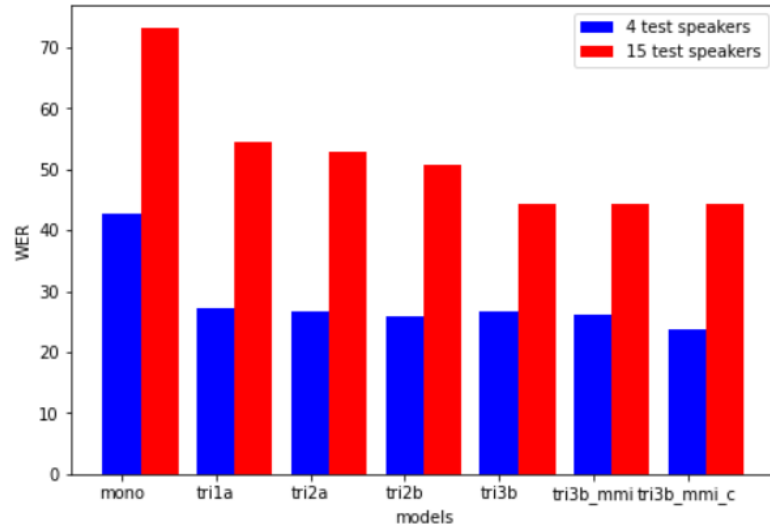


Figure 5.1: Word error rates with 4 test speakers and 15 test speakers in a speaker dependent system

5.2 Speaker independent system

In a speaker independent system, the test speakers are not chosen and prefixed. They are taken from dataset by random by ASR. The randomness introduced reflects a real world scenario but can reduce performance. Below are some experiments conducted on speaker independent systems.

With 15 test speakers in an independent speaker system, a GMM-HMM ASR was trained with 30 speakers on a phonetic dictionary of around 5000 words. The highest WER achieved in this experiment was an improved **37.57%** on **sgmm2_4a** and again an improved **42.8%** on **tri3b** triphone model. The reason for better word error rates in this ASR was the larger size of phonetic dictionary.

Figure 5.2 shows comparison between speaker dependent and speaker independent system with 15 test speakers. *Tri3b* model gives improved result in speaker independent system and the reason is random test speakers chosen by ASR in speaker independent system had lesser mix of noise in test data than data of speaker dependent system.

MODEL	WER
mono	77.82%
tri1	58.36%
tri2a	52.63%
tri2b	53.92%
tri3b	42.80%
tri3b_mmi	43.51%
sgmm2_4a	37.57%

Table 5.3: WER on 15 test speakers in a speaker independent system

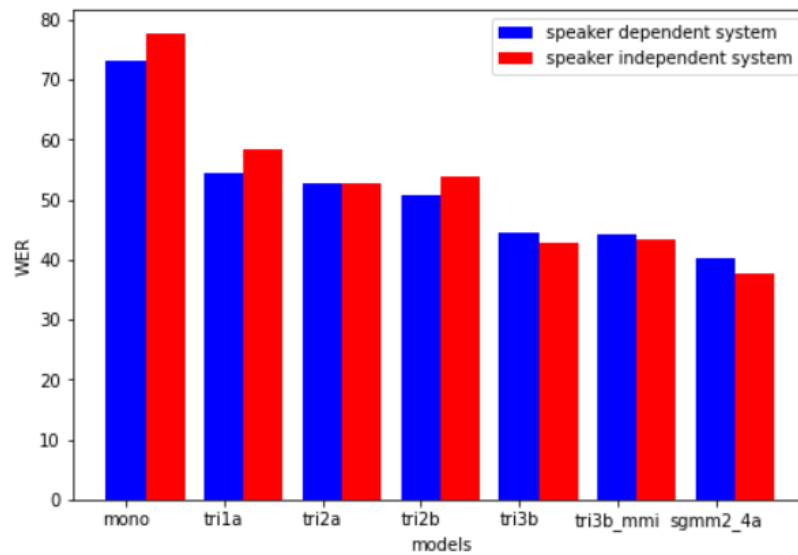


Figure 5.2: Comparison between speaker dependent and speaker independent system with 15 test speakers

5.3 Training and testing data

Performance of a speech recognition system is affected due to various factors in training and testing data. Noise and speed, training data size, male female ratio, phonetic dictionary and corpus in training are some important aspects impacting the overall result of an ASR. Besides there are a few other important things that need to be kept in mind such as if a model is trained on a particular data and tested on a completely different type of data, the WER produced would not be good. Below sub sections describe experiments conducted keep in mind such aspects.

5.3.1 Noise and speech speed in data

A normal speech of an speaker in a real word will contain various types of noise. The speaking speed of the person will also vary. An ASR trained with noised will produce better results when noise is introduced in test data. If training data contains minimal noise while test data contains noisy data, the WER produced tends to be higher.

A GMM-HMM ASR was trained on 40 speakers. The ASR was tested with 8 test speakers. The test data contained real world data with faster speeds and noise (buzzing sounds, fan and vehicle noise, background noise of people speaking), meanwhile training data contained minimal noise.

The highest WER achieved in this experiment was **36.28%** on **sgmm2_4a** and **40.86%** on **tri3b** triphone model. The reason for larger word error rates in this ASR was inclusion of real world data with faster speed and noise.

A p-norm DNN on top of above GMM-HMM ASR was trained on 2 hidden layers. Word error rate achieved was **55.91%**. The reason for a higher WER in DNN was lack of noisy data in training data set. If training data had contained some noisy data and faster speeds then better results would have been achieved. Figure ?? shows all the WERs achieved in in this experiment.

To check the effects of training on noisy an non-noisy data while testing on the same, a GMM-HMM ASR was trained with 45 speakers in training data. The ASR was tested with 8 test speakers. Testing data had 4 speakers containing real world data with faster speed and noise and 4 speakers containing ideal data. Similarly training data also contained at least 4 speakers containing real world data with faster speed and noise.

The highest WER achieved in this experiment was **29.63%** on **tri3b_fmmi_c** and **31.8%** on **tri3b** triphone model. Word error rates in this ASR are better than previous GMM-HMM ASR because training data now also contains noisy data.

MODEL	NOISY DATA IN TEST	NOISY DATA IN TEST AND TRAIN
mono	74.63%	58.29%
tri1	63.65%	45.53%
tri2a	63.40%	44.18%
tri2b	60.88%	42.19%
tri3b	40.86%	31.80%
tri3b_mmi	40.81%	31.0%

Table 5.4: WER on ASR with noise in test data and ASR with noise in test and train data

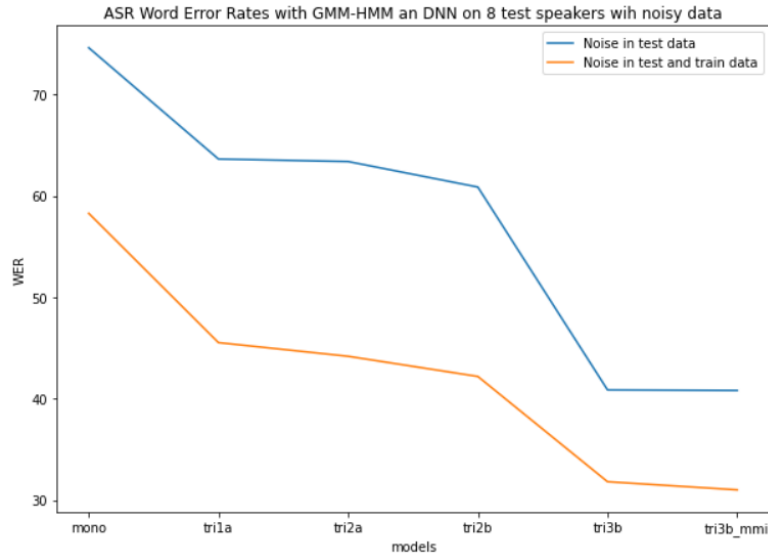


Figure 5.3: Comparison between ASR with noisy data in test and ASR with noisy data in test and train

Similarly, for DNN, to check above effects of training on noisy and non-noisy data while testing on the same, a p-norm DNN was built with 2 hidden layers and trained on tri3b of above ASR. Word error rate was improved to **41.35%** from **55.91%**. This is an improvement of 14%. The reason for a better WER in this experiment than DNN is the inclusion of noisy data in training data.

To further test, it was not just noise reducing the performance of ASR with 8 test speakers containing noisy data only, another ASR was built in which training data

MODEL	NOISY DATA IN TEST	NOISY DATA IN TEST AND TRAIN
tri3b	55.91%	41.35%

Table 5.5: WER on DNN with noise in test data and DNN with noise in test and train data

contained test data as well. Resultant ASR contained 53 train speakers. The speech data in test speakers contained real world data with faster speed and noise and that same data was copied in training as well. The highest WER achieved in this experiment was **21.63%** on **tri3b_fmml_d** and **23.98%** on **tri3b** triphone model. A p-normalized vector DNN with 2 hidden layers was trained on tri3b of the this GMM-HMM ASR. Word error rate was improved to **23.17%**. The reason for better word error rates in these ASRs was inclusion of test speakers data in training data.

5.3.2 Training data size

Increasing training data size, greatly improves the performance the ASR. In this study, various experiments were conducted that saw impressive improvements in WER when training data was increased.

To show the effect of training data size, the first ASR with 1 hour of test data and 5.5 hours of training data can be used as a baseline. The highest WER achieved in that experiment was **21.81%** on **tri3b** triphone model.

To compare, a GMM-HMM ASR with 1 hour of test data and 11.5 hours of training data was trained. The speech data of test speakers was same as that of baseline ASR. The highest WER achieved in this experiment was an impressive **20.45%** on **tri3b** triphone model (an improvement of 1.5%). The reason for better word error rates in this ASR was the inclusion of more training data.

Building a p-norm DNN with 6 hidden layers trained on tri3b of above GMM-HMM ASR, word error rate was improved by 4% to **15.99%**.

MODEL	GMM-HMM WER	DNN WER
tri3b	20.45%	15.99%

Table 5.6: WER on GMM-HMM and DNN with 1 hour of test data and 11.5 hours of training data

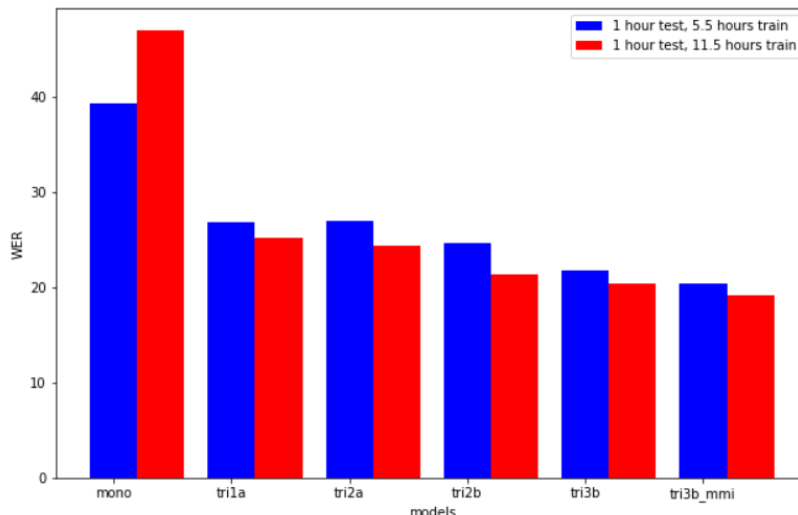


Figure 5.4: Word error rates of ASRs different train and test data size

5.3.3 Phonetic dictionary size

Phonetic dictionary in an ASR contains the mapping of words to their phonetic pronunciation with every word and its mapped pronunciation written on a separate line. Increasing the size of phonetic dictionary decreases WER hence improving the results.

To test the results of an increased phonetic dictionary, a previously created ASR with smaller phonetic dictionary is used as baseline. In section 5.2, an ASR was tested on a phonetic dictionary of around 1860 words with 15 test speakers and 30 train speakers. The highest WER achieved in that experiment was 40.35% on `sgmm2_4a` and 44.43% on `tri3b` triphone model.

Now for comparison, a similar ASR with a larger phonetic dictionary of around 10,000 words was trained with 30 speakers. The ASR was tested with 15 test speakers. The speech data in test speakers was a mix of ideally controlled environment with no noise and normal or slower speeds as well as some noisy data with faster speeds. The highest WER achieved in this experiment was an improved **25.01%** on `sgmm2_4a` (an improvement of 15%) and **29.30%** on `tri3b` triphone model (an improvement of 15%). The reason for better word error rates in this ASR was the larger size of phonetic dictionary than baseline model.

A p-norm DNN with 2 hidden layers was trained on `tri3b` above GMM-HMM ASR. Word error rate achieved was **36.03**. A lower WER for DNN could be a sign that model

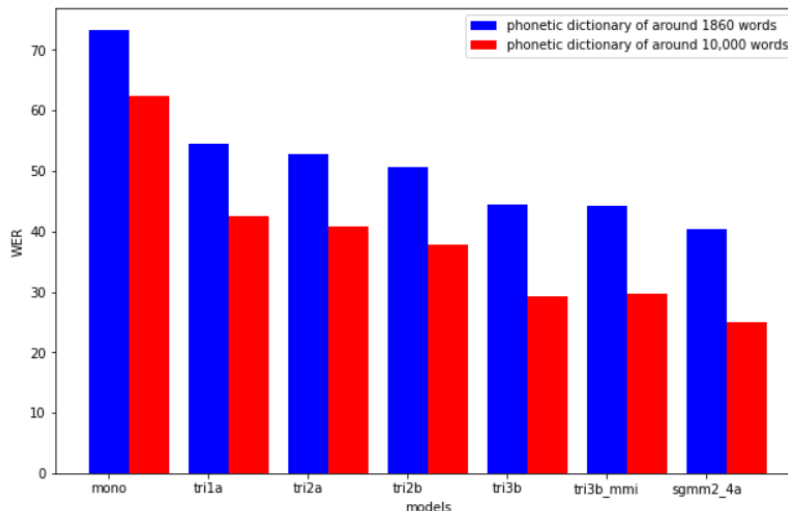


Figure 5.5: Word error rates of ASRs with different phonetic dictionary size

was over-fitted or the presence of noisy data in test and lack of it in training data.

5.3.4 Extra language model data in training

Including extra language model data corpus or text data in training improves the performance of an ASR.

To test the results of addition of corpus in training, a previously created ASR without extra language model text data in training is used as baseline. The first ASR in subsection 5.3.1 had 8 test speakers, 45 train speakers and no corpus in training. The highest WER achieved in that experiment was 29.63% on tri3b_fmml and 31.8% on tri3b triphone model. While p-norm DNN with 2 hidden layers and trained on tri3b of baseline ASR gave word error rate of 41.35%.

Now for comparison, a similar ASR containing the extra language model text data corpus was trained with 45 speakers on a phonetic dictionary of around 10,000 words. The ASR was tested with 8 test speakers. Similar to baseline ASR, testing data had 4 speakers containing real world data with faster speed and noise and 4 speakers containing ideal data. Training data also contained at least 4 speakers containing real world data with faster speed and noise.

The highest WER achieved in this experiment was improved by 4% to **25.15%** on **tri3b_fmml** and **26.81%** on **tri3b** triphone model (an improvement of 5% than baseline ASR).

A p-norm DNN with 2 hidden layers was trained on tri3b of ASR with corpus and word error rate was improved by 7% to **33.01%**.

Word error rates in this experiment for GMM-HMM and DNN-HMM ASR are better than WER in baseline ASR because now training data contains extra language model text data or transcriptions which improves the language model thus giving improved results.

MODEL	WITH EXTRA LANGUAGE MODEL	WITHOUT EXTRA LANGUAGE MODEL
mono	50.47%	58.29%
tri1	37.72%	45.53%
tri2a	36.19%	44.18%
tri2b	34.10%	42.19%
tri3b	26.81%	31.80%
tri3b_mmi	25.68%	31.0%

Table 5.7: WER on ASR with extra language model data in training data and ASR without extra language model data in training data

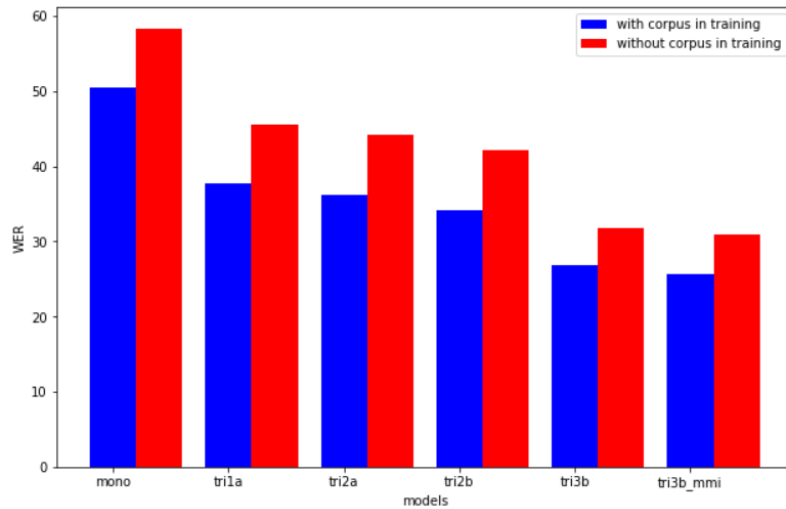


Figure 5.6: Comparison between ASR with corpus in training data and ASR without corpus in training data

MODEL	WITH CORPUS WER	WITHOUT CORPUS WER
tri3b	33.01%	41.35%

Table 5.8: WER on DNN with corpus in training data and without corpus in training data

5.3.5 Testing DNN parameters

Experiments on DNN with different number of hidden layers, learning rates and values of p for p-norm non-linearity were conducted. For testing, DNN was trained on GMM-HMM tri3b model with 1 hour of test data and 11.5 hours of training data.

DNNs were tested with 2, 3, 4, 5 and 6 number of hidden layers with $p=2$ and it was observed that DNN with 6 hidden layers gave the best word error rate of **15.99%** outperforming all other models.

DNNs were also tested with 2, 3 and 5 values for p in p-norm non-linearity with 6 hidden layers and it was found that $p=2$ gave best WER of **15.99%**.

NUMBER OF HIDDEN LAYERS	WER
2	17.35%
3	16.70%
4	16.18%
5	16.02%
6	15.99%

Table 5.9: DNN word error rates with different number of hidden layers

VALUES OF P	WER
2	15.99%
3	16.07%
5	16.13%

Table 5.10: DNN word error rates with different values of p in p-norm non linearity

5.4 Summary of Experiments and Results

After conducting tests and experiments, it is found that Sindhi ASR performs better when more training data is introduced. If there is noise in data, the ASR needs to contain enough of that in training phase to produce good results in testing. It is further found that increasing phonetic dictionary size and incorporating a large corpus in language model for training also greatly improves the performance.

It is concluded that for Sindhi ASR, a DNN with 6 hidden layers and p-norm non-linearity activation function with $p=2$ built on top of tri3b GMM-HMM model with 2000 HMM-states and 10000 Gaussian distributions with enough training data outperforms a GMM-HMM model and produces a smaller and better word error rate (See Fig 5.4 and Tab. 5.6). In the next chapter concluding remarks and suggestions are presented for any work to carry this ASR forward in future.

Conclusion and Future work

6.1 Conclusion

Various automatic speech recognition systems were created and compared in this thesis for recognizing utterances of Sindhi language using Kaldi. GMM-HMM and deep neural networks were used to build multiple ASR systems. The performances in word error rates of these various systems was compared to recommend one final ASR which will give satisfactory results on test data. We found that a DNN with 6 hidden layers with $p=2$ for p -norm non-linearity gave the best WER of 15.99% outperforming all other models. We observed that an ASR system which is trained and tested on data containing both, real time data with noise and normal speed of speech along with speech data of a controlled environment, would give better results. We further observed that including more speech data, corpus text and larger phonetic dictionary tends to improve the performance of the system. This study contained most of the speakers living in upper Sindh region.

6.2 Future work

For future work, it is suggested to increase the speech data to 100 hours and include more female speakers. Since this study had speakers speaking an specific accent in upper Sindh, more speakers speaking other accents of Sindhi should be added to make this an accent independent system. The system should be trained on more noisy data. Another suggestion is to build an ASR system in Roman Sindhi, as it is easier to write and is commonly used now by most of the youth.

References

- [1] Distribution of pakistanis speaking sindhi as a first language. https://commons.wikimedia.org/wiki/File:Distribution_of_Pakistanis_speaking_Sindhi_as_a_first_language_in_1998.png. Accessed: 2021-07-14.
- [2] Gboard - the google keyboard. <https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin>. Accessed: 2021-07-14.
- [3] Hidden markov models. <https://web.stanford.edu/~jurafsky/slp3/A.pdf>. Accessed: 2021-07-14.
- [4] Jason Eisner. An interactive spreadsheet for teaching the forward-backward algorithm. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, pages 10–18, 2002.
- [5] Investigating the effects of gender, dialect, and training size on the performance of arabic speech recognition. <https://link.springer.com/article/10.1007/s10579-020-09505-5/figures/3>. Accessed: 2021-07-14.
- [6] Eiman Alsharhan and Allan Ramsay. Investigating the effects of gender, dialect, and training size on the performance of arabic speech recognition. *Language Resources and Evaluation*, 54(4):975–998, 2020. doi: 10.1007/s10579-020-09505-5.
- [7] Speech Recognition - GMM, HMM. <https://jonathan-hui.medium.com/speech-recognition-gmm-hmm-8bb5eff8b196/>, . Accessed: 2021-07-14.
- [8] Gmm explained. <https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95>, . Accessed: 2021-07-14.

REFERENCES

- [9] Pradeep Rangan and Ramaswamy Kumaraswamy. Comparison of conventional methods and deep belief networks for isolated word recognition. 01 2015. doi: 10.1109/NCCSN.2014.7001147.
- [10] Introduction to deep learning (i2dl) (in2346). <https://niessner.github.io/I2DL/img/introdl.png>, . Accessed: 2021-07-14.
- [11] A dnn with one hidden layer. <https://towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f>, . Accessed: 2021-07-14.
- [12] Yuehjen E. Shao and Shih-Chieh Lin. Using a time delay neural network approach to diagnose the out-of-control signals for a multivariate normal process with variance shifts. *Mathematics*, 7(10), 2019. ISSN 2227-7390. URL <https://www.mdpi.com/2227-7390/7/10/959>.
- [13] Greg Van Houdt, Carlos Mosquera, and Gonzalo Nápoles. A review on the long short-term memory model. *Artificial Intelligence Review*, 53, 12 2020. doi: 10.1007/s10462-020-09838-1.
- [14] Kaldi directory structure. <https://www.eleanorchodroff.com/tutorial/kaldi/images/directorystructure2.png>. Accessed: 2021-07-14.
- [15] Michael Saxon, Samridhi Choudhary, Joseph P. McKenna, and Athanasios Mouchtaris. End-to-end spoken language understanding for generalized voice assistants. *CoRR*, abs/2106.09009, 2021. URL <https://arxiv.org/abs/2106.09009>.
- [16] Zhaofeng Wu, Ding Zhao, Qiao Liang, Jiahui Yu, Anmol Gulati, and Ruoming Pang. Dynamic sparsity neural networks for automatic speech recognition, 2021.
- [17] Ashutosh Pandey, Chunxi Liu, Yun Wang, and Yatharth Saraf. Dual application of speech enhancement for automatic speech recognition, 2020.
- [18] Naseer Ahmad Huma Akram, Yingxiu Yang and Sarfraz Aslam. Factors contributing low english language literacy in rural primary schools of karachi, pakistan. *International Journal of English Linguistics*, 10(6):335–346, 2020.
- [19] Matthew B. Hoy. Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical Reference Services Quarterly*, 37(1):81–88, 2018. doi: 10.1080/02763869.

- 2018.1404391. URL <https://doi.org/10.1080/02763869.2018.1404391>. PMID: 29327988.
- [20] Jesusimo L. Dioses Jr. Androiduino-fan: A speech recognition fan-speed control system utilizing filipino voice commands. *International Journal of Advanced Trends in Computer Science and Engineering*, 9:3042–3047, 06 2020. doi: 10.30534/ijatcse/2020/84932020.
- [21] Bhagath Parabattina, Savinay Parihar, and Pradip Das. Speech recognition for indian spoken languages towards automated home appliances. pages 1–5, 05 2021. doi: 10.1109/INCET51464.2021.9456267.
- [22] Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. Must-c: A multilingual corpus for end-to-end speech translation. *Computer Speech & Language*, 66:101155, 2021. ISSN 0885-2308. doi: <https://doi.org/10.1016/j.csl.2020.101155>. URL <https://www.sciencedirect.com/science/article/pii/S0885230820300887>.
- [23] Ruiqing Zhang, Xiyang Wang, Chuanqiang Zhang, Zhongjun He, Hua Wu, Zhi Li, Haifeng Wang, Ying Chen, and Qinfei Li. Bstc: A large-scale chinese-english speech translation dataset, 2021.
- [24] Saad Naeem, Majid Iqbal, Muhammad Saqib, Muhammad Saad, Muhammad Soban Raza, Zaid Ali, Naveed Akhtar, Mirza Omer Beg, Waseem Shahzad, and Muhhamad Umair Arshad. Subspace gaussian mixture model for continuous urdu speech recognition using kaldi. In *2020 14th International Conference on Open Source Systems and Technologies (ICOSST)*, pages 1–7, 2020. doi: 10.1109/ICOSST51357.2020.9333026.
- [25] Mohammad Ali Humayun, Ibrahim A. Hameed, Syed Muslim Shah, Sohaib Hassan Khan, Irfan Zafar, Saad Bin Ahmed, and Junaid Shuja. Regularized urdu speech recognition with semi-supervised deep learning. *Applied Sciences*, 9(9), 2019. ISSN 2076-3417. doi: 10.3390/app9091956. URL <https://www.mdpi.com/2076-3417/9/9/1956>.
- [26] Muhammad Farooq, Farah Adeeba, Sahar Rauf, and Sarmad Hussain. Improving large vocabulary urdu speech recognition system using deep neural networks. pages 2978–2982, 09 2019. doi: 10.21437/Interspeech.2019-2629.

REFERENCES

- [27] Shibli Nisar and Muhammad Asadullah. Home automation using spoken pashto digits recognition. In *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, pages 1–4, 2017. doi: 10.1109/ICIEECT.2017.7916545.
- [28] History of voice recognition: from audrey to siri. <https://www.itbusiness.ca/news/history-of-voice-recognition-from-audrey-to-siri/15008>. Accessed: 2021-07-14.
- [29] From audrey to siri. is speech recognition a solved problem? <https://www.icsi.berkeley.edu/pubs/speech/audreytosiri12.pdf>. Accessed: 2021-07-14.
- [30] Ibm archives: Ibm shoebox. https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html. Accessed: 2021-07-14.
- [31] B. Juang and Lawrence Rabiner. Automatic speech recognition - a brief history of the technology development. 01 2005.
- [32] A brief history of asr: Automatic speech recognition. <https://medium.com/descript/a-brief-history-of-asr-automatic-speech-recognition-b8f338d4c0e5>. Accessed: 2021-07-14.
- [33] B. T. Lowerre. *The Harpy speech recognition system*. PhD thesis, Carnegie-Mellon Univ., Pittsburgh, PA., April 1976.
- [34] MIT develops a speech recognition chip that uses a fraction of the power of existing technologies. <https://techcrunch.com/2017/02/13/mit-speech-chip/>. Accessed: 2021-07-14.
- [35] 10 best voice recognition software (speech recognition in 2021). <https://www.softwaretestinghelp.com/voice-recognition-software/>. Accessed: 2021-07-14.
- [36] Market trends: Voice as a ui on consumer devices what do users want? <https://www.gartner.com/en/documents/3021226>. Accessed: 2021-07-14.
- [37] The machines that learned to listen. <https://www.bbc.com/future/article/20170214-the-machines-that-learned-to-listen>. Accessed: 2021-07-14.

- [38] Dragon speech recognition - get more done by voice | nuance. <https://www.nuance.com/dragon.html>. Accessed: 2021-07-14.
- [39] Cmusphinx open source speech recognition. <https://cmusphinx.github.io/>. Accessed: 2021-07-14.
- [40] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Luká Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíek, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Vesel. The kaldi speech recognition toolkit. *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 01 2011.
- [41] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *The Journal of the Acoustical Society of America*, 87 4:1738–52, 1990.
- [42] Mouaz Bezoui, A. Elmoutaouakkil, and A. B. Hssane. Feature extraction of some quranic recitation using mel-frequency cepstral coefficients (mfcc). *2016 5th International Conference on Multimedia Computing and Systems (ICMCS)*, pages 127–131, 2016.
- [43] Abed S., Alshayeji M., and Sultan S. Diacritics effect on arabic speech recognition. *Arabian Journal for Science and Engineering*, 44(11):9043–9056, 2019. doi: 10.1007/s13369-019-04024-0.
- [44] A. Ouisaadane and Said Safi. A comparative study for arabic speech recognition system in noisy environments. *International Journal of Speech Technology*, pages 1–10, 2021.
- [45] Prashant Upadhyaya, Omar Farooq, Musiur Raza Abidi, and Yash Vardhan Varshney. Continuous hindi speech recognition model based on kaldi asr toolkit. *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 786–789, 2017.
- [46] Guglani J. and Mishra A.N. Continuous punjabi speech recognition model based on kaldi asr toolkit. *International Journal of Speech Technology*, 21:211–216, 2018. doi: <https://doi.org/10.1007/s10772-018-9497-6>.
- [47] Yogesh Kumar and Navdeep Singh. An automatic speech recognition system for spontaneous punjabi speech corpus. *International Journal of Speech Technology*, 20

- (2):297–303, 2017. doi: 10.1007/s10772-017-9408-2. URL <https://doi.org/10.1007/s10772-017-9408-2>.
- [48] Taniya, Vivek Bhardwaj, and Virender Kadyan. Deep neural network trained punjabi children speech recognition system using kalditoolkit. In *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, pages 374–378, 2020. doi: 10.1109/ICCCA49541.2020.9250780.
- [49] Jinal H. Tailor and Dipti B. Shah. Hmm-based lightweight speech recognition system for gujarati language. In Durgesh Kumar Mishra, Malaya Kumar Nayak, and Amit Joshi, editors, *Information and Communication Technology for Sustainable Development*, pages 451–461, Singapore, 2018. Springer Singapore. ISBN 978-981-10-3920-1.
- [50] Zafi Syed, Sajjad Ali, Muhammad Shehram Shah Syed, and Abbas Shah. Introducing the urdu-sindhi speech emotion corpus: A novel dataset of speech recordings for emotion recognition for two low-resource languages. *International Journal of Advanced Computer Science and Applications*, 11, 04 2020. doi: 10.14569/IJACSA.2020.01104104.
- [51] Muhammad Saim Younus Hashmi, Dil Nawaz Hakro, and Anjali Mandhan. Offline sindhi speech recognition. In *2019 International Conference on Digitization (ICD)*, pages 32–37, 2019. doi: 10.1109/ICD47981.2019.9105891.
- [52] Steven B. Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, August 1980. ISSN 0096-3518. doi: 10.1109/TASSP.1980.1163420.
- [53] A. G. Kalhor, M. R. Rabbani and H. Nasir. Speech recognition in sindhi language for telephonic security surveillance. *School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan*, 2020.