Urdu Speech Recognition for Navigation Applications

By

Syed Meesam Raza Naqvi

00000117857

Supervisor

Dr. Muhammad Ali Tahir

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters in Computer Science (MS CS)

In

School of Electrical Engineering and Computer Science,

National University of Sciences and Technology (NUST),

Islamabad, Pakistan.

(December 2018)

Approval

It is certified that the contents and form of the thesis entitled "**Urdu Speech Recognition for Navigation Applications**" submitted by **Syed Meesam Raza Naqvi** have been found satisfactory for the requirement of the degree.

Supervisor: **Dr. Muhammad Ali Tahir**

Signature: _____

Date: _____

Committee Member 1: **Dr. Asad Waqar Malik**

Signature: _____

Date: _____

Committee Member 2: **Dr. Hassan Aqeel Khan**

Signature: _____

Date: _____

Committee Member 3: **Dr. Safdar Abbas Khan**

Signature: _____

Date: _____

Acceptance Certificate

Certified that final copy of MS/MPhil thesis written by Mr. **Syed Meesam Raza Naqvi**, Reg no. **00000117857**, of SEECS has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: _____

Name of Supervisor: _____

Date: _____

Signature (HOD): _____

Date: _____

Signature (Dean/Principal): _____

Date: _____

Abstract

Recently automatic speech recognition (ASR) has gained a lot of public attention due to spiking interest in the area by some major tech giants. From voice-activated digital assistants in our homes to voice recognition based search engines, speech recognition is being used everywhere these days. Modern voice recognition services support many languages but Urdu is usually not one of them. In Pakistan huge portion of population do not speak or understand English. Even some of the popular English voice recognition systems do not efficiently understand English in Pakistani accent. In this study we developed a mixed English-Urdu speech recognition system for TPL Maps Pakistan (a part of the TPL Corp) for their voice-enabled navigation service. Kaldi an open source speech recognition toolkit is used for development of speech recognition models. Two different ASR systems are developed and compared in this study using general Urdu data and mixed data (general Urdu + roman Urdu addresses). As a part of this study various GMM-HMM and DNN-HMM models are developed and evaluated for both ASR systems. In terms of Word Error Rate, ASR system developed using mixed data is found to achieve better performance as compared to the system trained using only general Urdu data.

Keywords — Urdu Speech Recognition, navigation, Kaldi, Gaussian Mixture Models, Hidden Markov Models, Deep Neural Network, LSTM

Certificate of Originality

I hereby declare that this submission is my own work and 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 acknowledgment 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.

Author's Name: **Syed Meesam Raza Naqvi**

Signature: _____

Dedication

I dedicate this work to my grandfather Syed Fida Hussain Naqvi who taught me the real meaning of courage and strength, and to my loving parent, Syed Hashim Raza Naqvi and Syeda Iffat Zahra Naqvi for their endless, love support and encouragement.

Acknowledgment

I would like to express my deepest gratitude to my supervisor Dr. Muhammad Ali Tahir and committee members (Dr. Asad Waqar Malik, Dr. Hassan Aqeel Khan and Dr. Safdar Abbas Khan) for their unwavering support and mentorship throughout my masters thesis.

I would like to extend my special thanks to Dr. Hassan Aqeel Khan for providing access to lab and GPUs for LSTM training.

Contents

List of Figures

List of Tables

Abbreviations

AM	Acoustic Model.
ASR	Automatic Speech recognition.
CMVN	Cepstral Mean and Variance Normalization.
DCT	Discrete Cosine Transform.
DFT	Discrete Fourier Transform.
DNN	Deep Neural Networks.
GMM	Gaussian Mixture Model.
HMM	Hidden Markov Models.
LDA	Linear Discriminant Analysis.
LM	Language Model.
LVCSR	Large Vocabulary Continuous Speech Recognition.
MFCC	Mel-Frequency Cepstral Coefficient.
MLLT	Maximum Likelihood Linear Transform.
SAT	Speaker Adaptive Training.
WER	Word Error Rate.
WFSA	Weighted Finite State Acceptors.
WFST	Weighted Finite State Transducer.

Chapter 1

Introduction

Over past few decades automatic speech recognition (ASR) has been active area of research as technology being considered as an efficient means of Human to Human and Human to machine communication. In past, however automatic speech recognition has not been used as a primary method for human to human or human to machine communication. This is partly because technology at that time was not mature enough to meet standards of users under real worlds sceneries and partly because other interaction methods were more preferable like keyboards, mouse, touch.

During recent years use of automatic speech recognition increased considerably compared to past as technology is now mature enough to be even integrated into smart devices. Mobile applications like Google assistant, Amazon's Alexa, Apple's Siri etc [5, 6] are redefining the way we interact with our smart devices. There are various reasons behind this trend, one of these reason is availability of better computational resources. Today we have more powerful computational resources including multi core processors, faster storage devices and general purpose graphical processing units (GPUs). With all these resources it is now possible to train more complex and pow-

erful models efficiently in shorter duration of time. These complex models have significantly reduced error rates of automatic speech recognition (ASR) systems leading to better performance and user friendliness. Secondly, with advancement in big data technologies we now have access to large databases that we can use to train better efficient models that are more generic in nature. By training models on these huge repositories of real world data we can avoid assumption that were made due to shortage of data and train more robust models. Thirdly, a major reason behind progress in speech system is increased usage of smart devices such as mobile phones, smart wearables, smart homes, infotainment systems in vehicles etc. On these devices unlike personal computers traditional input tools like mouse, keyboard are less convenient and interaction methods such as speech and touch are more preferred.

These were few of the important reasons behind rapid advancements in various speech technologies. In the following section some of the popular application of speech systems are briefly described.

1.1 Applications

Automatic speech recognition has many useful applications most of these application fall under the category of human to human communication or human to machine communication. Few of these applications are briefly discussed here.

1.1.1 Voice Search

Voice search [7, 8] is an important application of speech recognition. Using voice search user can search for anything using voice commands instead of

physically typing the search command. Voice search is one of the popular methods of search in modern smart phone devices and most of the modern mobile operating systems come with inbuilt voice search feature like Google assistant and Apple's Siri. There are various applications of voice search in different scenarios some of these applications include:

- Query search engines.
- Get driving directions.
- Search for hotels and restaurants.
- Search for products on e-commerce websites.
- Help user with accessibility issues in searching.
- Search map applications for destinations and places.

1.1.2 Speech-to-Speech (s2s) Translation

Usually when two people from different backgrounds don't understand language of each other they need a human interpreter who understands both languages to translate. Due to the language barrier speakers are unable to communicate freely or privately. Speech-to-speech [9, 10] translation is a technique in which one language can be automatically translated to another without the help of interpreter. Using this technique any language can be translated automatically to any other language thus removing language barrier between speakers of different backgrounds. Using speech translation speakers can communicate freely and privatively without any concerns. This technique can also be integrated into any online messengers like Skype, Whatsapp, Facebook etc. to make these applications more user friendly. Figure

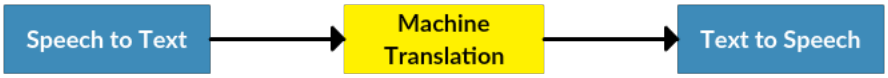

Figure 1.1: Typical speech to speech (s2s) translation system

1.1 shows main components of a typical s2s system. First step in s2s pipeline is automatic speech recognition of speaker's message/query. Obtained text from speech recognition is then translated into desired language using machine translation. Finally text obtained from translation is played on the listeners end using text to speech (TTS). There are various gadget available in the market that can perform both online and offline s2s translation.

1.1.3 Home Automation

Another important application of speech recognition is in home automation systems. Automation is becoming an important part of our daily life from smart homes to autonomous cars a lot of research is going on in this area. Home automation systems [11, 12] help residents to remotely control various home appliances. Most of these systems have integrated voice recognition system through which system is operated. These systems not only reduce human effort but also save time and are energy efficient. Home automation systems are also very useful for handicapped and old people with mobility issues and can also generate alerts in case of emergency and security breach. With voice recognition service like Alexa these systems are becoming perfect companion for a modern home.

1.1.4 Enhanced Gaming Experience

Removing the barriers between gaming experience and real life have always been target of gaming community. From photo-realistic high definition graphics to virtual reality experiences the ultimate goal of every advancement seems focused towards making artificial reality indistinguishable from the actual reality. Regardless of these advancements we still interact with most of these games using joystick, keyboard, mouse etc. Voice recognition is changing the way we interact with the games. Instead of using menu for selection of items in a game, modern games are using voice commands that are not only fast but more realistic.

Although modern speech recognition systems support multiple languages but Urdu is mostly not one of these languages. Most of these speech recognition systems are mainly focused on English speech recognition as English being most common language worldwide. In Pakistan a huge portion of population do not speak or understand English. Development of a Urdu speech recognition system to provide voice activated services to these Pakistani non-English and even English speakers can improve the way these people interact with their smart devices. This study is part of joint effort of national university of science and technology (NUST) and TPL maps (a part of the TPL Corp) to develop Pakistan's first voice-enabled navigation in Urdu. Developed Urdu speech based navigation system is integrated into existing TPL Maps application.

1.2 Problem Statement

Most of the speech recognition services do not support Urdu language and others do not recognize Pakistani accent efficiency. Also these services are difficult to be tailored for specific needs. Like addresses in Pakistan are most written in Roman Urdu containing code-mixed words from Urdu and English. For these type of scenarios we need a mixed English-Urdu model that can recognize both languages at the same time using a mixed language model. The goal of this thesis is development of voice-enabled navigation service for recognition of English-Urdu code-mixed addresses. In this study two different Urdu ASR systems are developed and compared using general Urdu data and mixed data (general Urdu + roman Urdu addresses). For both systems various models are developed using state of the art techniques such as Gaussian Mixture Models (GMMs) and Deep neural networks (DNNs).

1.3 Thesis Outline

Chapter 2, presents a brief background about a typical speech recognition system, different models developed during this study and Kaldi (an open source speech precognition toolkit) used to to develop the system. Chapter 3 is about the literature review of various automatic speech recognition systems developed so far, mainly focusing on Urdu. In chapter 4 design and methodology of developed system is discussed. In chapter 5 outcomes of different ASR systems developed during this study and different tests performed on these systems are discussed. Finally, chapter 6 concludes this thesis and presents possible future directions.

Chapter 2

Background

2.1 Automatic speech recognition (ASR)

Automatic speech recognition (ASR) is recognition and translation of spoken language into text. An ASR system is used to estimate most likely sequence of words for a given a speech input. Figure 2.1 shows the block diagram of a typical ASR system.

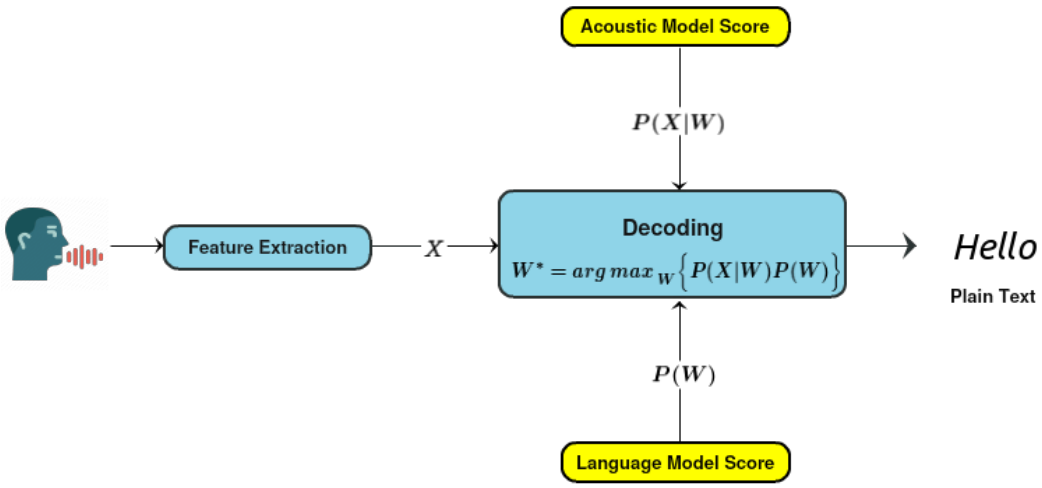

Figure 2.1: Architecture of an ASR system

In a typical speech recognition system, first step is features extraction from the input speech. Feature extraction involves applying various signal processing techniques to enhance the quality of input signal and transform input audio from time domain to frequency domain. Based on the features extracted a set of acoustic observations $X = \{x_1, x_2, x_3, ...x_k\}$ is generated given a sequence of words $W = \{w_1, w_2, w_3, ...w_n\}$. The speech recognizer then estimated *"the most likely word sequence W^* for given acoustic observations based on set of parameters $\boldsymbol{\Theta}$ of underlying model"*. This can be mathematically formulated as conditional probability as shown in equation 2.1. Equation 2.1 can be further simplified using Bayesian rule and represented as 2.2.

$$W^* = arg\,max_W\{P(W|X,\Theta)\} \tag{2.1}$$

$$W^* = arg\,max_W\Big\{\frac{P(X|W,\Theta)P(W|\Theta)}{P(X|\Theta)}\Big\} \tag{2.2}$$

The term in the denominator $P(X|\Theta)$ of equation 2.2 is the prior probability of given acoustic sequence X which is constant for all W^* so can be ignored. As we have two different models (language and acoustic model), thus there are two different sets of parameters $\boldsymbol{\Theta_{AM}}$ and $\boldsymbol{\Theta_{LM}}$. After assigning relevant parameters to respective terms, equation 2.2 can be further simplified.

$$W^* = arg\,max_W\Big\{P(X|W,\Theta_{AM})P(W|\Theta_{LM})\Big\} \tag{2.3}$$

Where $P(X|W,\Theta_{AM})$ also known as acoustic model (AM) score is the probability of set acoustic observation given parameters of acoustic model. $P(W|\Theta_{LM})$

also known as language model (LM) score is the probability of words given parameters of language model. Estimation of best acoustic and language model parameters is an active area of research in speech recognition. Following text will briefly explain various blocks in figure 2.1.

2.1.1 Feature Extraction

Feature extraction also known as speech parameterization is used to characterize spectral features of an input audio signal in order to facilitate speech decoding. *Mel-frequency cepstral coefficients (MFCC)* introduced by Davis and Mermelstein [13] is one of the most popular techniques for feature extraction in speech recognition systems. Reason behind the popularity of MFCC is its ability to mimic the behavior of human ear. Figure 2.2 shows key steps involved in calculation of MFCC features.

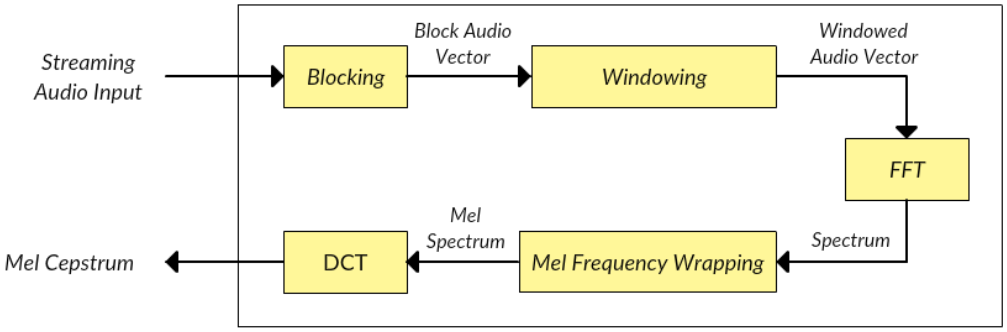

Figure 2.2: MFCC computation

- ***Frame Blocking:*** First step involved in the process is frame blocking in which streaming audio signal is blocked into frames of 25 *ms* shifted by 10 *ms*.
- ***Windowing:*** After blocking, each frame is multiplied by a window using a windowing function. There are many windowing functions

available in *Kaldi* but usually Hamming window is used which can be mathematically represented as equation 2.4.

$$w(n) = 0.54 - 0.46\, cos\Big(\frac{2\pi n}{N-1}\Big) \quad 0 \leq n \leq N-1 \tag{2.4}$$

Where $\boldsymbol{N}$ is the number of samples in a frame. Multiplying every frame by Hamming window reduces discontinuities at beginning and end of each frame. This step is also required because in order to do frequency analysis (FFT) of each frame it should be continuous.

- ***Fast Fourier Transform (FFT):*** Spectral analysis of speech signals shows that different timbres in a signal have different energy distribution over frequency. FFT is applied on each frame of N samples to obtain its magnitude frequency response. This process converts signals from time to frequency domain. FFT is fast implementation of Discrete Fourier Transform (DFT).

- ***Mel Frequency Wrapping:*** In this step magnitude frequency response resulting from FFT is multiplied by triangular bypass filters on Mel scale to get log energy of each bypass filter. Mel frequency that is more discriminative at lower frequencies and less discriminative at higher frequencies mimics the non-linear perception of sound by human ear. We can convert between Mel frequency (m) and frequency (f) in Hertz using following equations.

$$m = 2595\, log_{10}(1 + \frac{f}{700}) \tag{2.5}$$

$$f = 700(10^{m/2595} - 1) \tag{2.6}$$

Each filter in triangular filter bank has response 1 at the center frequency and decreases linearly towards 0 till it reaches center of adjacent frequency. In Kaldi default number of filters is 23 because it usually gives best results on 16 *khz* speech signals.

- ***Discrete cosine transform (DCT):*** After log energy computations, mel-frequency cepstrum is obtained by applying DCT on filtered results. Coefficients of mel-frequency cepstrum are called mel frequency cepstral coefficients or MFCC. Applying DCT after FFT transforms frequency domain into time like domain called quefrency domain. In Kaldi by default first 13 coefficients of cepstral are kept as features.

MFCC can be directly used as feature for speech recognition, but in order to get better performance various transforms are applied on results of MFCC. One of these transformations is *Cepstral Mean and Variance Normalization (CMVN)* [14]. CMVN is a computationally efficient normalization technique which reduces the effects of noise. Similarly in order to add dynamic information to MFCC features first and second order deltas can be calculated. Given a feature vector X first order deltas can be calculated as.

$$\Delta X_t = \frac{\sum_{i=1}^{n} w_i (X_{t+i} - X_{t-i})}{2 \sum_{i=1}^{n} w_i^2} \tag{2.7}$$

Where w_i is the regression coefficients and n is the window width. Second order deltas can be derived from first order deltas using equation 2.8.

$$\Delta X_t = \frac{\sum_{i=1}^{n} w_i(\Delta X_{t+i} - \Delta X_{t-i})}{2\sum_{i=1}^{n} w_i^2} \tag{2.8}$$

After first and second order delta calculation combined feature vector becomes

$$\Delta X_t = [X_t \quad \Delta X_t \quad \Delta^2 X_t] \tag{2.9}$$

Other feature transformation techniques used in Kaldi are Linear Discriminant Analysis (LDA) [15], Heteroscedastic Linear Discriminant Analysis (HLDA) [16] and Maximum Likelihood Linear Transform (MLLT) [17]. These transforms can be applied individually as well as in combination and can greatly enhance performance of speech recognition system. It is observed that applying diagonalizing MLLT after LDA improves effect of LDA (LDA+MLLT) [17].

2.1.2 Acoustic Modeling

Acoustic Modeling in ASR system estimates $P(X|W, \Theta_{AM})$. Acoustic model parameters (Θ_{AM}) are estimated by training the model. In speech data exact time of words in an utterance is not known so there is a level of uncertainty involved in training. Hidden Markov models (HMMs) are used to model this temporal variability of speech. HMMs model a frame or window of frames of coefficients as state machines. Gaussian mixture models (GMMs) or Deep neural networks (DNNs) are then used to determine how well each state of

each HMM fits frame/s of acoustic features to acoustic input. Nature of training in GMMs is usually generative while in DNNs its discriminative.

Basic unit of training in speech is phone instead of word. A *phone* is a smallest unit of speech. Each word consists of sequence of phones. Number of phone in a language are far less then the total unique words. If we use words as a training unit then model would have to know each unique word in a language making the dimensionality of the problem too high to handle. Words in the speech transcripts are converted to phones using a phonetic dictionary which contains phones against the words from the vocabulary. In addition to the dictionary, out of vocabulary (OOV) words a converted to phones using Grapheme-to-Phoneme (G2P) model which is training using manually converted words.

A monophone acoustic model is a model trained on individual phones. A better approach compared to monophone modeling is triphone modeling. A triphone is a sequence of three phones and it capture the context of the phone in middle very efficiently. If there are N base phone then there are N^3 possible triphones. Triphones are modeled using HMMs. Using triphones also increases the dimensionality of data to reduce this effect triphones that are acoustically similar are tied together using a technique know as state-tying discussed in section 2.2.1. In Kaldi state-tying is implemented using decision trees.

Urdu language has approximately 67 phones. Table 2.1 shows a full map of Urdu phones along with their International Phonetic Alphabet (IPA) and Case Insensitive Speech Assessment Method Phonetic Alphabet (CISAMPA).

Table 2.1: List of IPA and CISAMPA of different sounds in Urdu [4]

Sr.#	*Urdu Letter*	*IPA*	*CISAMPA*	*Sr.#*	*Urdu Letter*	*IPA*	*CISAMPA*
	Consonants						
1	پ	P	*P*	36	رھ	r^h	*R_H*
2	پھ	p^h	*P_H*	37	ڑ	ɽ	*R_R*
3	ب	B	*B*	38	ڑھ	$ɽ^h$	*R_R_H*
4	بھ	b^h	*B_H*	39	ی	J	*J*
5	م	M	*M*	40	یھ	j^h	*J_H*
6	مھ	m^h	*M_H*	41	چ	ʧ	*T_S*
7	ت،ط	t̪	*T_D*	42	چھ	$ʧ^h$	*T_S_H*
8	تھ	$t̪^h$	*T_D_H*	43	ج	ʤ	*D_Z*
9	د	d̪	*D_D*	44	جھ	ʤh	*D_Z_H*
10	دھ	$d̪^h$	*D_D_H*		**Vowels**		
11	ٹ	T	*T*	45	ُو	u:	*U_U*
12	ٹھ	t^h	*T_H*	46	ُوں	ũ:	*U_U_N*
13	ڈ	D	*D*	47	و	o:	*O_O*
14	ڈھ	d^h	*D_H*	48	وں	õ:	*O_O_N*
15	ن	N	*N*	49	وَ	ɔ:	*O*
16	نھ	n^h	*N_H*	50	وَں	ɔ̃ :	*O_N*
17	ک	K	*K*	51	ا،آ	ɑ:	*A_A*
18	کھ	k^h	*K_H*	52	اں،آں	ɑ̃:	*A_A_N*
19	گ	ɡ	*G*	53	ی	i:	*I_I*
20	گھ	$ɡ^h$	*G_H*	54	یں	ĩ:	*I_I_N*
21	نک,نگ,نگھ,نکھ in ن	ŋ	*N_G*	55	ے	e:	*A_Y*
22	ق	Q	*Q*	56	یں	ẽ:	*A_Y_N*
23	ع	ʔ	*Y*	57	ہ ِ	e	*A_Y_H*
24	ف	F	*F*	58		æ	*A_E_H*
25	و	V	*V*	59	ہ ُ	O	*O_O_H*
26	ث،ص،س	S	*S*	60	ے	æ:	*A_E*
27	ض،ظ،ز،ذ	Z	*Z*	61	یَں	æ̃:	*A_E_N*
28	ش	ʃ	*S_H*	62	ِ	ɪ	*I*
29	ژ	ʒ	*Z_Z*	63	ُ	ʊ	*U*
30	خ	X	*X*	64	ہ،ء َ	ə	*A*
31	غ	ɣ	*G_G*	65	ُ ں	ʊ̃	*U_N*
32	ہ،ح	H	*H*	66	َ ں	ə̃	*A_N*
33	ل	L	*L*	67	ِ ں	ɪ	*I_N*
34	لھ	l^h	*L_H*				
35	ر	R	*R*				

Table 2.2: Basic Urdu Alphabets

ا	ب	پ	ت	ٹ	ث	ج	چ	
ح	خ	د	ڈ	ذ	ر	ڑ	ز	
ژ	س	ش	ص	ض	ط	ظ	ع	
غ	ف	ق	ک	گ	ل	م	ن	
و	ہ	ء	ی	ے				

Table 2.3: Secondary Urdu Alphabets

آ	ں	ۃ	ھ	

Table 2.4: Urdu Diacritics

ٰ	ٓ	ٔ	ّ	ُ	َ
					ِ

Table 2.2 and 2.3 presented above show primary and secondary alphabets of Urdu. While table 2.4 shows different diacritics used in Urdu writing.

Hidden Markov models (HMMs)

Hidden Markov Model is statistical model used in speech recognition to represent acoustics of words. HMM model has chain of states in which current state is hidden and only output of each state can be observed. During acoustic training a_{ij} and $b_i(y_t)$ are estimated, where a_{ij} is state to state transition probability (*e.g* from state i to j) and $b_i(y_t)$ is the emitting function used to estimate output observations as shown in figure 2.3. An important feature

of HMMs is ability to self loop on a state which enables HMMs to model different phone lengths. First and the last state of the model are known as non-emitting states. These states are used for entry and exit in the model and help in concatenation of HMMs and phone models to generate words. There are various techniques used to estimate distribution of output observations. In this study two different techniques used for estimating emitting function $b_i(y_t)$ are:

- *Gaussian Mixture Model (GMM)*
- *Deep Neural Network (DNN)*

Both of these techniques are briefly described in the upcoming sections.

Gaussian Mixture Model (GMM)

While HMM is used to model temporal variability of speech, GMM a statistical generative model is common choice to estimate distribution of output observations. Combination of both models create a joint acoustic model capable of describing temporal as well as spectral dynamics of the speech. Figure 2.3 shows the architecture of an arbitrary GMM-HMM model for speech recognition. Where $\boldsymbol{Y}$ is output vector of observation sequence. The

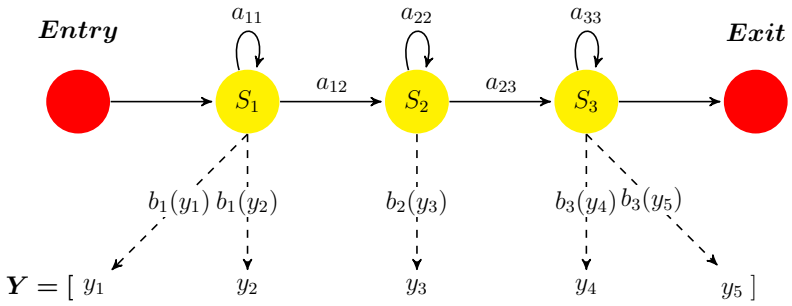

Figure 2.3: Architecture of left to right GMM-HMM based acoustic model with 3 states [1]

model consists of three emitting states S_1, S_2 and S_3 and two non-emitting states (entry and exit). Model can be fully described using matrix $A = [a_{ij}]$ containing probabilities of all possible transitions from one state to another and emitting functions for each state $b_i(y_t)$. Usually model only allows self loop but no backward transition. Emitting function for a given state used to estimate the probability of observation vector $\boldsymbol{Y}$ can be expressed by equation 2.10 below. Observation vector $\boldsymbol{Y}$ can be generated using assigned emitting function which estimates the probability of observations.

$$b_i(y_t) = \sum_{m=1}^{M} c_m \mathcal{N}(y_t; \mu_{im}, \Sigma_{im}) \tag{2.10}$$

$\mathcal{N}(\mu_{im}, \Sigma_{im})$ is the multivariate normal distribution. Parameters ($\Theta_{AM} = \{c_m, \mu_{im}, \Sigma_{im}\}$) of this distribution that need to be estimated are weights, mean and covariance matrix respectively. For a given frame y_t the observation probability depends only on emission probability $b_i(y_t)$ of respective state (i). Whereas the probability of state sequence ($S = S_1, S_2, ...S_k$) generation depends only on state to state transition probabilities [1].

$$P(Y|\Theta_{AM}) = \sum_{S_1, S_2, ...S_k} \prod_{t=1}^{T} a_{S_t|S_{t-1}} b_{S_t}(y_t) \tag{2.11}$$

Where $a_{S_t|S_{t-1}}$ is the state transition probability that can also be expressed as $P(S_t|S_{t-1})$. There are several schemes for learning acoustic model parameters (Θ_{AM}) from training data. The most common approach used to estimate the acoustic model parameters is based on Maximum Likelihood Estimation (MLE) [1]. Main drawbacks of using MLE are assumptions we make for GMM-HMM and MLE itself when modeling speech. Discriminative training methods on the other hand do not make any assumption about

the distribution of training data. It is one of the major reasons behind the success of discriminative training algorithms making them principle training algorithms in speech modeling. Kaldi uses Viterbi algorithm for updating acoustic model's parameters (Θ_{AM}) and Gaussian variables.

Deep Neural Network (DNN)

Deep neural network (DNN) is an alternative of Gaussian mixture model [18]. A DNN is a neural network with more than one hidden layers between input and output layer. Figure 2.4 illustrates the architecture of a typical DNN.

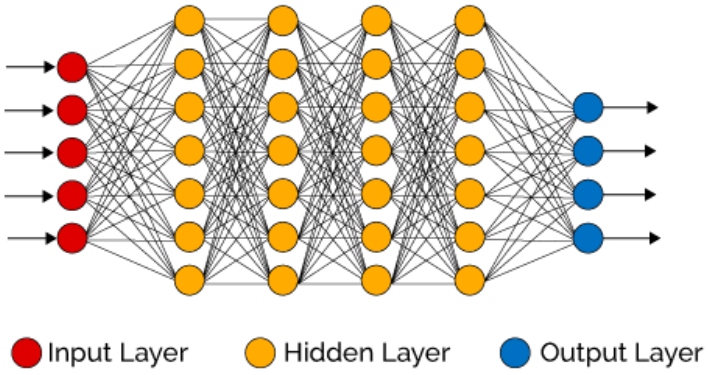

Figure 2.4: Architecture of a deep neural network

In DNN each hidden unit or neuron j uses a logistic function which could be closely related hyperbolic tangent or any other function with well behaved derivative. This function maps all inputs on that neuron from previous layer x_j to the scaler output y_j which acts as an input to the next layer. Equation 2.12 and 2.13 show the relation between x_j and y_j [18].

$$y_j = logistic(x_j) = \frac{1}{1 + e^{-xj}} \tag{2.12}$$

$$x_j = b_j + \sum_i y_j w_{ij} \tag{2.13}$$

Where b_j in equation 2.13 is bias of hidden unit j, i represents indexing of units in previous layer and w_{ij} is the weight of the connection from unit i in previous layer to unit j in current layer. Eventually y_j in the current unit becomes x_i for the units in the next layer. Figure 2.5 shows the structure of a single hidden unit in a feed-forward neural network.

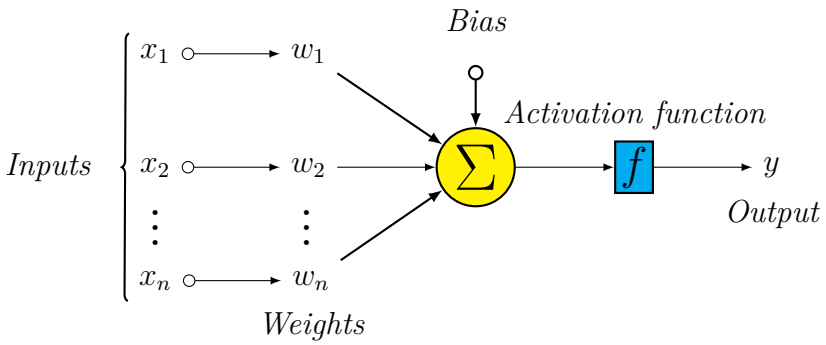

Figure 2.5: Architecture of a single hidden unit/neuron of a feed-forward neural network

In case of multiclass classification problems like speech recognition total input x_j is converted into class probabilities p_j using *softmax* activation function as shown in equation 2.14.

$$p_j = \frac{exp(x_j)}{\sum_k exp(x_k)} \tag{2.14}$$

Where k is the class index. An important feature and reason behind success of DNN in speech recognition is their ability to be trained discriminatively using back-propagation of cost function derivatives measuring the difference between actual and predicted output for each training case. Natural cost function for softmax output is the cross entropy between the actual proba-

bilities and output of softmax.

So far different DNN architectures are designed and tested on speech recognition tasks. In this study we will focus on time delay neural network (TDNN) and long short-term memory (LSTM). Both type of DNNs are briefly described in the following sections.

- ***Time Delay Neural Network (TDNN):*** As described in section 2.1.2 a typical neural network hidden unit computes weighted sum of all of its inputs and passes this weighted sum through nonlinear activation function (usually sigmoid). TDNN uses modified basic unit with delays d_1 to d_n as shown in figure 2.6.

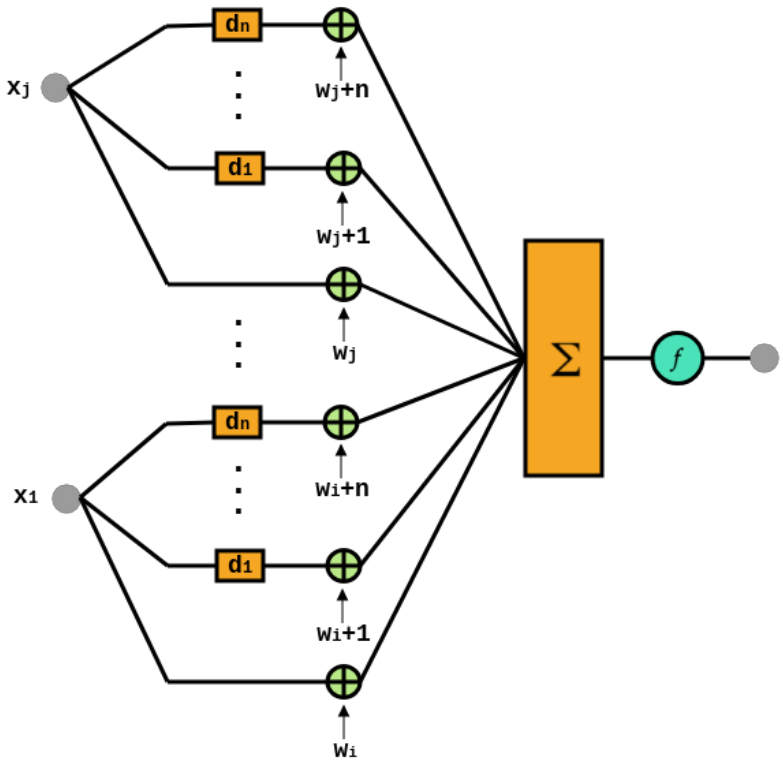

Figure 2.6: Architecture of a single hidden unit/neuron of a Time Delay Neural Network (TDNN) [2]

In TDNN each input from x_i to x_j is multiplied by various weights one

for each delay. In case of 14 inputs ($n = 14$) and delay of 3 ($d = 3$), 56 weights will be required to compute the weighted sum with each input being measured on four different points in time. In this way TDNNs can relate the current input with the context of past events. The activation function commonly used in TDNN is sigmoid. Common learning technique used in TDNN is backpropagation which is gradient decent of the mean squared error of actual and predicted outcomes [2].

- ***Long Short-term Memory (LSTM):*** LSTM is a variation of DNN in which special hidden unit called memory block is introduced. In memory block there are memory cells having recurrent connections that store the temporal state of the network. This block also contains special multiplicative units known as gates to control the flow of information through the unit. In original LSTM architecture [19] there was an input and output gate in each memory block. Function of input gate is to control the flow of input activations into the memory cell while output gate was designed to control the flow of activations from current memory cell to the rest of the neural network. There was a weakness in original LSTM architecture that hindered LSTMs to process continuous input stream if the stream is not subsequenced. To address this problem later a forget gate was introduced [20] to enable adoptive forgetting. Forget gate enabled forgetting or resetting capability in LSTM cell's memory. Modern LSTM architectures also contain peephole connection from memory cells to the gate in the same cell to learn precise output timing. Figure 2.7 shows the architecture of peephole LSTM memory cell [21]. It shows input and output to the memory cell and how different components are connected in LSTM

memory cell. LSTM networks are famous for sequence to sequence

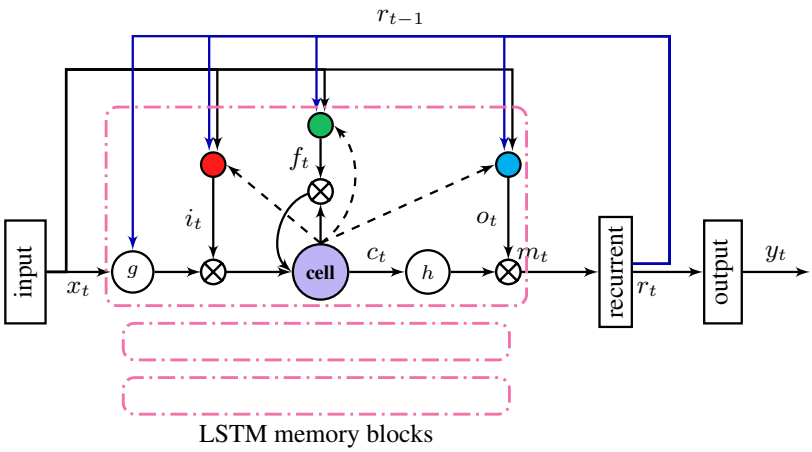

Figure 2.7: Architecture of LSTMP-RNN memory cell [3]

learning applications such as speech recognition. LSTM based ASR systems are among state of the art systems with the lowest word error rate (WER) regardless of underlying language.

2.1.3 Language Model

Language model contains likelihood of co-occurrence of words in the vocabulary. Language model is used to determine $P(\boldsymbol{w})$ which is hypothesized word sequence that can be further decomposed using chain rule.

$$P(\boldsymbol{w}) = \prod_{i=1}^{n} P(w_i|w_{i-1}, ..., w_i) \tag{2.15}$$

Where $P(w_i|w_1, ..., w_{i-1})$ in equation 2.15 is the probability of current word given previous history $(w_1, ..., w_{i-1})$. As creating a model given all possible word sequences is impracticable *n-gram* model is used in current state of the art approaches that limits length of history to $n-1$ words. Throughout

this study various experiment were performed using *3-gram* language models. Although language model optimization is hot area of research in speech recognition but it is not part of this research, that is why it is briefly discussed here.

2.1.4 Performance Measure

A common performance measure used to compare different ASR models is percentage Word Error Rate (WER). WER is calculated using Levenshtein distance between words [22]. It is calculated by counting number of insertions, substitutions and deletions performed to make two word sequences equal. Depending upon the problem, cost of insertions, substitutions and deletions can be set, by default this cost is equal for all operations and is set to 1. When reference (*ref*) transcript is matched with hypothesis (*hyp*) each word in hypothesis is assigned respective label based on whether it is insertion (I), substitution (S), deletion (D) or correct (C).

Tables 2.6 and 2.5 show sample reference and hypothesis with labels from decoding results of general Urdu and Roman Urdu addresses using tri3b model.

Table 2.5: Sample reference and hypothesis with labels from addresses decodings

ref	halar	cement	dealers	mirpur	mathelo
hyp	hilal	cement	dealers	mirpur	mathelo
label	S	C	C	C	C

Formula for calculating WER is presented in equation 2.16. WER is calculated after each alignment during decoding process.

$$WER = \frac{S_t + D_t + I_t}{N} \tag{2.16}$$

Table 2.6: Sample reference and hypothesis with labels from general Urdu decodings

ref	ایسی تجاویز سامنے لاءی جاءیں جنھیں قومی پالیسی میں سمو یا جا سکے مجھے یقین ہے کہ اے پی سی کا جذبہ یہاں بھی
hyp	ایسی تجاویز سامنے لاءی جاءیں جنھیں قومی پالیسی میں سمو یا جا سکے مجھے یقین ہے کہ *** ایک کسی کا جذبہ یہاں بھی
label	C C C C S S D C C C C C C C C C C C C C C C C

Where S_t, D_t, I_t, C_t are total substitutions, deletions, insertions and correct responses respectively. $N = (S_t + D_t + C_t)$ is the number of words in the reference transcript. Using this formula we can calculate WER of given samples. For sample in table 2.5 $N = 4$, where $C_t = 3$, $S_t = 1$, $D_t = 0$ and $I_t = 0$ so WER for this sample calculated below using equation 2.16 is 25%.

$$WER = \frac{1 + 0 + 0}{1 + 0 + 3} = 0.25$$

For sample in table 2.6 $N = 23$, where $C_t = 20$, $S_t = 2$, $D_t = 1$ and $I_t = 0$ so WER for this sample calculated below using equation 2.16 is 13%.

$$WER = \frac{2 + 1 + 0}{2 + 1 + 20} = 0.13$$

WER metric explained above is very intuitive and straight forward measure that can be used to compare different ASR models. In this study WER is used as performance measure to analyze and compare efficiency of various ASR models trained and tested during development of different ASR systems.

2.2 Kaldi ASR toolkit

Kaldi is an open source speech recognition toolkit released under Apache license [23]. Development of Kaldi started in 2009 and is now one of the most popular speech recognition toolkits. The basic idea behind Kaldi was the de-

velopment of an ASR toolkit that is flexible and extensible. The main reason behind success of Kaldi is availability of various deep neural network recipes that other toolkits lack. In the following sections some of the techniques used in Kaldi like triphone state tying and Weighted finite State Transducers (WFST) are briefly discussed.

2.2.1 Triphone state tying

A triphone is a sequence of three phones used to efficiently capture the context of the phones compared to monophone. Using triphones as training unit results into increase in dimensionality. To address this curse of dimensionality parameters tying technique described in [24] was introduced. Kaldi applies this technique at state level to map acoustically similar triphones to the same HMM state using decision tree. A binary decision tree is generated for each phone to cluster its associated triphones. Figure 2.8 shows an example of triphone state tying.

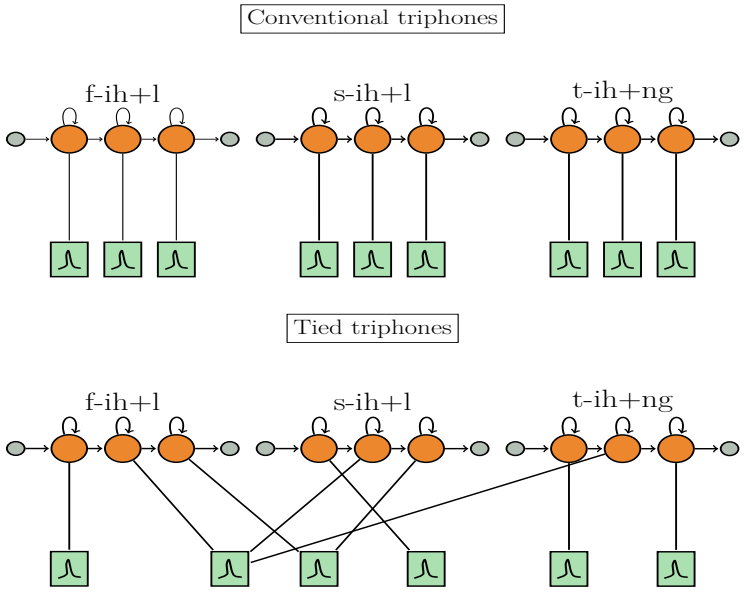

Figure 2.8: Triphone state tying

2.2.2 Weighted finite State Transducers (WFST)

A weighted finite state transducer (WFST) is used to map objects in input alphabet to the objects in output alphabet [25]. WFST is used in Kaldi to combine the information from language and acoustic models. Acoustic HMM and *n-gram* language model are special case of Weighted Finite State Acceptors (WFSAs). A WFSA model can have one state at a time that is know as current state. It can transition from one state to another in case of triggering event. If these transaction have cost associated then finite state acceptor is called weighted finite state acceptor (WFSA). In case of speech recognition this cost is associated probabilities. If we add information of final outcome in WFSA states it can then be interpreted as transducer. In this way all the information required for an ASR model can be integrated into one transducer.

Chapter 3

Literature Review

Earliest ASR system for single speaker digit recognition was developed at Bell Labs in 1950's but in the last two decades there have been huge advancement in this field mainly for English language [26]. Among a number of ASR systems developed lately, few systems were considered notable including; BYBLOS: BBN's recognition system for continuous speech [27], SUMMIT: MIT's speech recognition system [28] and Dragon: Nuance Communications' system [29] with an accuracy of 98.5%, 87% and 95% respectively. Although there is plenty of literature available that concern designing and development of Automatic Speech Recognition (ASR) systems but there is still a huge gap when it comes to Urdu ASR development. Similarly, improvements and research for the recognition of resource rich languages has been significant but there is still a huge research gap for under resourced languages. This section mainly focuses on the developments in Urdu ASRs developed using above mentioned architecture. Authors in [30] presented an Automated Learning of Accent and Articulation Mapping (SALAAM) system for under resource languages using the available resource rich languages such as English.

There are several open source speech recognition toolkits available. Some

of the famous tools include; Kaldi, CMUSphinx, Hidden Markov Model Toolkit (HTK), Simon, Julius etc. We have chosen Kaldi toolkit as it has proven to provide outstanding results and provides the most advance training recipes as compared to other tools [31]. It is called a "Low Development Cost, High Quality Speech Recognition for New Languages and Domains" [32] and thus can be incorporated for the Urdu language recognition.

Commonly two different architectures are used in ASR development. First architecture is the combination of Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM). Second architecture is based on combination of deep neural networks (DNN) and Hidden Markov Models (HMM).

For GMM-HMM models first major contribution was made about 4 decades ago when the expectation-maximization (EM) algorithm was used for training HMMs [33, 34]. In this way, using the effectiveness of GMMs it became possible to develop ASR systems for real world tasks [35]. Acoustic input in such systems is given as perceptual linear predictive coefficients (PLPs) or Mel-frequency cepstral coefficients (MFCCs) features that are computed from the first and second order temporal differences of the input signal [36, 37]. GMMs are very useful for probability distribution modeling over input data features associated with the state of HMM. There is plenty of literature available that focus on the optimization of GMMs flexibility, amount of training data required to avoid overfitting and increase their evaluation speed [38]. Accuracy can also be improved by combining the MFCC features with tandem features generated using NNs [39]. Performances of GMMs are difficult to outperform by using new approaches for acoustic modeling.

For NNs, there is plenty of literature available that concern designing and development of Automatic Speech Recognition (ASR) systems using Deep Neural Network (DNN) and Recurrent Neural Network (RNN). Hybrid ap-

proaches for Feed-forward neural network acoustic models were initially explored in 1990's [40, 41]. Similarly, RNN and CNN for speech recognition were initially explored around the same time [2, 42]. More recently, almost all state of the art research work on speech recognition and acoustic models includes some form of DNN [43, 18, 44]. CNN and RNN have also proved to be useful and are being deployed in current state of the art ASR systems [45, 46] as well as in feature extraction with convolutional layers [47].

Current major research in the development of such systems incorporates multilingual ASR systems comparing results in the context of scoring outputs of different DNN-HMM models [48, 45, 49, 50].

In the context of Urdu speech recognition, there are a number of challenges that are faced by the research community ranging from speech corpora and phonetic lexicon development, to the testing, training and improvements of speech recognition system. Two challenges in speech corpus development are photonic balance and photonic cover [51]. Photonic cover essentially means that a speech corpus for a specific language contains all the phones present in that language and if these phones occur in almost the same relative proportions it is termed as photonic balance [52, 53]. Speech corpus of a specific language can be developed from isolated words [54], continuous speech [55, 56] or spontaneous speech [57, 58], Moreover, adding more transcribed spontaneous speech data can always make the dataset richer.

For improving the performance of ASR system various techniques can be employed. As for the case of read speech, a number of techniques (including unsupervised approaches) are used to generate training data. Similarly, in the case of spontaneous speech, different techniques are employed that include; classifying and modeling of speech disfluencies/inconsistencies [59, 60], finding mostly mispronounced phones and words and modeling them

separately [61, 62] and recognizing pauses, word lengthening and correctly detecting filled pauses [63]. An additional limitation for languages that use similar scripts as Arabic language, e.g. Urdu, Pashto, Persian etc., is that they optionally use diacritics for vowels that are usually written in the text. This can be solved by first training a model that uses speech transcriptions with fully manual diacritics and then further using this model for unsupervised learning of un-transcribed data [64].

A spontaneous speech recognition system with single speaker and medium vocabulary for Urdu has been developed using the Sphinx toolkit [65]. Authors showed that by including read speech into the spontaneous speech training data, WER can be decreased. Similarly, the work in [66] presented a speaker independent Urdu ASR system using the Sphinx toolkit with a limited vocabulary of 52 isolated words. Another work [67] presented a digit recognizer with entirely Arabic environment using Sphinx toolkit, i.e., it did not include Romanized scripts. Research work presented in [68], developed a continuous Urdu speech recognizer that looks for pattern matching and acoustic phonetic modeling and provides 55 to 60% accuracy. Another isolated digit recognizer was presented in [69] and [70] uses a multilayer perceptron to develop a similar model that recognize Urdu digits. Authors in [71] discussed some approaches for improving the recognition rates for Urdu speech recognition and presented acoustic models for robust Urdu speech recognition using CMUSphinx. Authors in [72] presented Urdu speech recognition system specifically designed for district names of Pakistan. They have discussed development challenges and solutions and concluded that accent independent system performs better for isolated words.

For Hindi language (similar to Urdu), a system has been developed with 65000-word vocabulary and provides accuracy of 75 to 95% [73]. In terms of

best WER, a system developed for similar script language i.e. Arabic, authors in [74] showed WER of 14.9% for Arabic broadcast news transcriptions for spontaneous microphone-based system. Table 3.1 show the list of various Urdu speech corpora both public and propitiatory. Stats show thats there is no publicly available LVCSR corpus for Urdu.

Table 3.1: List of various Urdu Speech Corpora

Lang	**Duration**	**Speech type**	**Source**	**Speakers**	**Vocabulary**	**Public**	**Ref**
Urdu	3 hrs	Read	Mic	1	7k	Yes	[75]
Urdu	12 hrs	Isolated	Telephone	300	139	Yes	[76]
Urdu	41.9 hrs	Continuous	Radio	-	-	No	[77]
Urdu	45 hrs	Read & Continuos	Tel+Mic	82	14k	No	[78]
Urdu	200 utt	Isolated	Mic	10	20	No	[79]
Urdu	5200 utt	Isolated	Mic	10	52	No	[66]
Urdu	12000 utt	Isolated	Mic	50	250	No	[80]

Table 3.2 shows the summary of different Urdu ASR systems along with their stats. Most of these systems are trained on small vocabulary using GMM-HMM models.

Table 3.2: List of various Urdu ASR systems

Genre	**Vocabulary**	**Speakers**	**Public**	**Technique**	**WER**	**Ref**
District Names	139	Multiple	Yes	GMM-HMM	7.13 %	[72]
Frequent words	52	Multiple	No	GMM-HMM	10.60 %	[78]
Phonetically rich sentences	6k	Single	Partial	GMM-HMM	18.80 %	[65]
Interviews	14k	Multiple	No	GMM-HMM	68.80 %	[71]

In this study different GMM-HMM and DNN-HMM models are trained and tested on a large vocabulary Urdu speech corpus.

Chapter 4

Design and Methodology

In this chapter details about the development of datasets, phonetic dictionary and different ASR systems during this study are discussed. This chapter also explains training and evaluation sequence of different GMM-HMM and DNN-HMM models.

4.1 Dataset

To train an ASR system for efficient address recognition we used two different datasets. Both dataset are prepared by *Speech and Language Technology Group* [81] at *School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST).*

First dataset having transcriptions in general Urdu is prepared using recordings from various sources including Urdu news bulletins, talk shows, radio programs, recording of Urdu literature. This dataset has recordings from more than 144 different speakers. Almost half of this dataset is prepared by transcribing recordings from online sources. The other half is prepared by recording various Urdu manuscripts from different sources by members of

speech research group, students and volunteers using Urdu ASR recording portal developed by *Speech and Language Technology Group* [82]. Second dataset having transcriptions in roman Urdu consists of various addresses in Pakistan recorded by 20 different speakers. This dataset mainly contains mixed English-Urdu words and numbers used in addresses. All the transcripts present in both datasets are recorded at 16000 Hz. Statistical details of both datasets are described in following table:

Table 4.1: Statistics of datasets used in this study

Dataset	**Total speakers**	**Total Recordings**	**Vocabulary Size** *(words)*	**Size** *(GBs)*	**Duration** *(hours)*
General Urdu	144+	17855	28391	7.1	61.82
Roman Urdu Addresses	18	12918	6194	2.1	16.89

Using above described datasets two different ASR systems are developed during this study (system S_u and system S_m). System S_u is developed using only general Urdu data while system S_m is developed using mixed data (general Urdu + roman Urdu addresses). Table 4.2 shows the names of developed ASR systems along with the names of dataset/s used for its development.

Table 4.2: Different ASR systems developed during this study

ASR System	**Training Data**
S_u	General Urdu
S_m	General Urdu + Roman Urdu Addresses

In reset of the sections models trained for systems in table 4.2 will be referenced with respective system names (S_u or S_m).

4.2 Phonetic Dictionary

Phonetic dictionary contains mapping of words to respective phones. Phonetic dictionaries for both datasets are prepared separately. Around 20% of the vocabulary for both datasets was manually converted to phones and rest of the vocabulary was converted using sequitur grapheme-to-phoneme (g2p) [83]. Training grapheme-to-phoneme model makes conversion process very fast. Sequitur g2p is a data-driven technique used to solve monotonous sequence translation problems (like word to phones conversion). Sequitur g2p has no built in language specification and can be used for any language; provided example pronunciations for training g2p model. To train a g2p model manually converted examples (pronunciation dictionary) of words to phones is used. Each line in the training dictionary has one word followed by its pronunciation. A g2p model can be trained using this manually prepared pronunciation dictionary covering most of the phones in the language. After training g2p model it can be used to generate pronunciations of remaining words in the vocabulary. Sequitur g2p model can also be integrated with Kaldi tool kit directly for the conversion of new words in language that are not in dictionary. For this study two different g2p models were training one for general Urdu data and other for roman Urdu addresses. Accuracy of the conversion was manually inspected and minor correction were performed where required.

4.3 Training

Kaldi not only provide implementation for feature extraction algorithms and transforms but also includes various *recipes* for acoustic model training. Kaldi toolkit provides many examples of various speech datasets (free as well as

paid). For available examples in Kaldi usually a script (*run.sh*) is present containing step by step instructions for feature extraction, training different acoustic models and testing these models on given dataset. Kaldi includes different examples depending upon the language for which that example was developed. Since there is no built-in example for Urdu language in Kaldi, scripts of an English example *voxforge*[1] were modified to be used for Urdu language.

Before training acoustic model, language model is generated. A language model can be used to understand the structure of the language as discussed in section 2.1.3. Based on provided corpus generated language model can be used to predict next word in the sentence/transcript based on current set of words. To generate a language model in Kaldi, a corpus based on desired transcripts is prepared to estimate the structure of given language. For this study using Kaldi language model generation script various *n-gram* language models with n=3 were generated depending upon underlying language (General Urdu, Addresses or English).

4.3.1 Training Acoustic Models

Before training of acoustic model dataset is divided into training and test sets. In this study two types of acoustic models (GMM-HMM and DNN-HMM) are trained for both ASR systems (S_u and S_m). As described in section 2.1.2 in GMM-HMM model, Gaussian mixture model and in DNN-HMM model, deep neural network is used to estimate the correctness of HMM output observations. Stepwise description of training process for different GMM-HMM and DNN-HMM models for both ASR systems is provided below:

[1] `http://www.voxforge.org/`

1. In first step acoustic features from training and testing data were extracted. For this study MFCC features are used as acoustic feature. To calculate MFCC feature using Kaldi *make_mfcc* script is used. Detailed process for extraction of these features is describe in section 2.1.1

2. After feature extraction, cepstral mean variance normalization (CMVN) is applied on resulting features. For this purpose *compute_cmvn_stats* script is used in Kaldi.

3. After feature extraction and normalization a basic mono phone acoustic model called ***mono*** was trained and tested on the test set.

4. In the next step a basic triphone model ***tri1*** was trained using same features as monophone model. Number of leaves and Gaussian used to train the network are set to 2000 and 11000 respectively.

5. Next model trained was ***tri2a*** using delta transformed features. First and second order deltas $\Delta + \Delta\Delta$ are used as features for this model number of leaves and Gaussian used in training are set to 2000 and 11000 respectively.

6. Then ***tri2b*** triphone model was trained by applying LDA+MLLT transform on the acoustic features using train_lda_mllt script. Again same number of leaves (2000) and Gaussian (11000) were used to train the model.

7. Last triphone model ***tri3b*** was trained by applying LDA+MLLT+SAT feature transforms using train_lda_mllt script followed by train_sat script. Total leaves was set to 2000 total Gaussian were set to 11000 while training the model.

8. Using the alignments from tri3b two different DNN-HMM models were trained:

   - ***nnet2:*** A setup in Kaldi containing deep neural network (DNN) training recipes like time delay neural network (TDNN) 2.1.2. nnet2 recipe used for this study is based on TDNN[2]. Features used for DNN in Kaldi are 40 dimension MFCC(spliced) + LDA + MLLT + fMLLR transformed features. The neural network sees only a window of transformed features having 4 frames on either side of central frame at a time. Trained TDNN model is a 6 layer model with 4 hidden layers. Network was trained for 6 epochs with initial learning rate of 0.0017 and final learning rate of 0.00017. Implementation details of TDNN architecture can be found in the paper of this recipe [84]
   - ***nnet3:*** A setup in Kaldi containing more general kind of deep neural network (DNN) training recipes like RNN, LSTM 2.1.2 From this study a long short-term memory (LSTM) 2.1.2 based DNN recipe[3] was used. Trained LSTM model is 11 layer model with 9 hidden layers. Network was trained for 6 epochs with initial learning rate of 0.0003 and final learning rate of 0.00003. Complete description of the work can be found in the relevant paper for the recipe [85].

Above training steps were used while training acoustic models for both systems S_u and S_m. After completion of training process both systems (S_u

[2]TDNN recipe used for training nnet2 model `https://svn.code.sf.net/p/kaldi/code/trunk/egs/swbd/s5c/local/online/run_nnet2_ms.sh`

[3]LSTM recipe used for training nnet3 model `https://github.com/kaldi-asr/kaldi/blob/master/egs/wsj/s5/local/nnet3/tuning/run_tdnn_lstm_1a.sh`

and S_m) were extensively tested using different test sets and language models to analyze their performance.

4.4 Testing

After training different acoustic models for both ASR systems extensive testing was performed on the trained models to test the efficiency of these models. Models were tested on different datasets using various language models. Flow of testing performed on the trained models is described in this section.

At the start of this study first models trained were for system S_u based on general Urdu data. After training these models, general Urdu test set was used to test the accuracy of the model on general Urdu transcripts using general Urdu language model. But our final goal was to recognize code-mixed roman Urdu addresses using system S_u. For this purpose a new language model was developed using addresses dataset and models were tested on addresses. As acoustic model is trained on phones so we generated roman counter parts of all the general Urdu words and results were satisfactory. After testing best model was hosted on TPL server and real time voice queries from clients using TPL maps mobile app and decodings against those queries were observed for few months. Although results were satisfactory but English digits in addresses were not being properly recognized by ASR system S_u. To resolve this issue a new ASR system S_m was developed for which models were trained and tested thoroughly using mixed data (general Urdu + roman Urdu addresses). The resulting models were first tested on the test set containing mixed transcripts and mixed language model. After verifying results on mixed data, acoustic models were tested on addresses language model using addresses transcripts only. After this testing it was observed that system S_m

stated recognizing English digit properly.

After finalizing ASR system some additional test scenarios were also applied on the final system S_m. Its Acoustic models were tested against new speakers that were not included in training data. Following three different additional test scenarios were applied on system S_m:

- Random roman addresses were recorded by four different speakers that were not in training data. These new addresses were then tested on various acoustic models of ASR system S_m using roman language model. Results indicated similar outcomes for new speakers compared to speakers in training set.

- After testing on speaker not in training set we focused accent testing on our final system S_m. For this purpose librispeech which is a famous speech dataset recorded in American accent was used. We recoded 50 random transcripts from librispeech dataset in Pakistani accent in voice of four different Pakistani speakers. These same transcripts in American and Pakistani accent were then tested on our model as well as on pre-trained librispeech models using librispeech language model. Models tested during this test were tri3b and nnet2 from both librispeech and system S_m.

- This accent testing was further extended by testing system S_m and librispeech models on 363 Pakistani accent Youtube English test data from four different speakers. This data was new for both models. Models tested during this test were also tri3b and nnet2 from both librispeech and system S_m. Results indicated similar outcomes likes before with that acoustic models of ASR system S_m performed well on Pakistani accent as compared to librispeech models which indicates the need for

training model on Pakistani speech data.

Result of each test described above are discussed in details in chapter 5.

Chapter 5

Results and Discussion

During this study extensive testing was performed on developed ASR systems. In this chapter results of developed ASR systems on various testing scenarios discussed in section 4.4 are presented and discussed in detail.

5.1 Testing Results

Figure 5.1 shows the Word Error Rate (WER) for different acoustic models of system S_u on general Urdu Language Model (LM) trained using general Urdu speech data (i.e. recordings and their transcriptions). As different Kaldi recipes (acoustic models) were initially trained for system S_u using general Urdu transcriptions, thus initial results were obtained by testing these models only against general Urdu transcriptions. Results shown for different Kaldi recipes depicted expected behaviour, that is, the performance of acoustic models gradually improved as their complexity and richness increased. Thus WER (%) for mono, tri1, tri2a, tri3b, nnet2 and nnet3 was 54.22, 31.07, 31.34, 27.98, 23.39, 15.34 and 14.12 respectively. Though theses results were satisfactory for the general Urdu speech, but the goal of this ASR

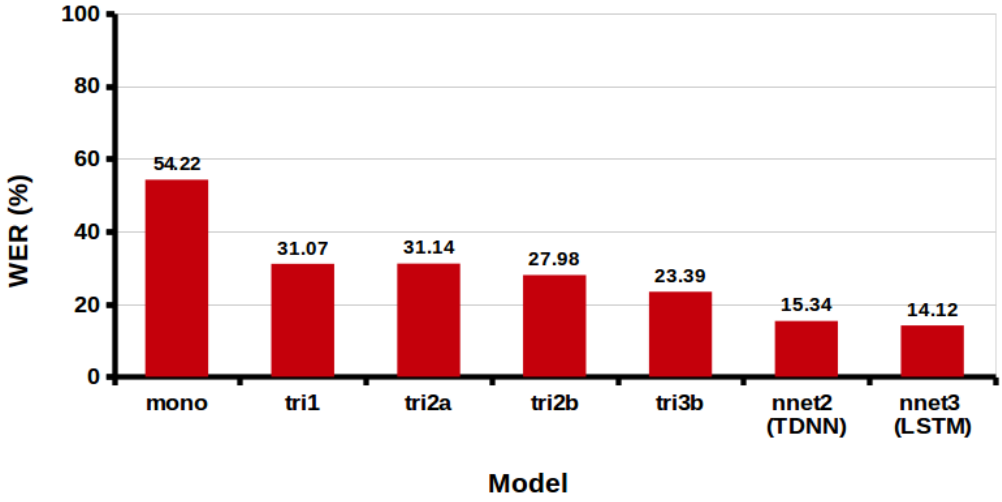

Figure 5.1: WER (%) of different acoustic models of system S_u on general Urdu data using general Urdu LM

system was to efficiently recognize street addresses in "Roman Urdu". For this purpose general Urdu LM was replaced by addresses LM prepared using roman Urdu addresses dataset. This approach works because the acoustic model is trained on phones and mapping of these phones is in phonetic dictionary, roman counterparts of general Urdu words can also be used with the acoustic model trained on general Urdu. Therefore, general Urdu language model was changed to address language model. Moreover, it was noticed that the street addresses in Pakistan are usually code-mixed i.e. they may include words from different languages, place's or person's names and digits in different languages. After development and testing ASR system S_u was hosted on an TPL Map's server for real time testing. Real time speech data was recorded and matched against resulting transcripts generated by the recognition system. The system was capable of recognizing the street addresses efficiently except the digits spoken in English. To cater this issue, roman addresses dataset was prepared (mentioned in section 4.1), as it consisted

only on street addresses and it also contained recordings and transcriptions of digits spoken in English. ASR System S_m was then developed by training acoustic models using both mixed data (roman Urdu addresses and general Urdu). Figure 5.2 shows the results of WER (%) of different acoustic models of system S_m when it was tested on mixed transcripts using mixed LM (general Urdu + roman addresses). It significantly improved the performance of the system as the WER (%) for mono, tri1, tri2a, tri3b, nnet2 and nnet3 reduced to $49.42, 30.4, 30.29, 28.34, 19.64, 16.54$ and 12.29 respectively.

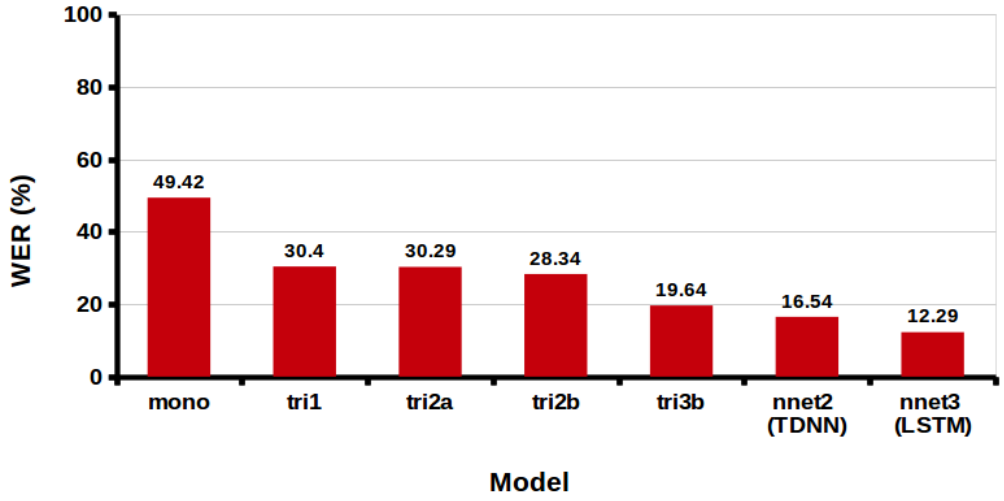

Figure 5.2: WER (%) of different acoustic models of system S_m on mixed data using mixed LM

Figure 5.3 shows the results of different acoustic models of system S_m when tested on 900 test addresses using only addresses LM to check how effectively it meets the requirements of street address recognition system. Because of large training data consisting of both datasets (general Urdu and roman Urdu addresses), system achieved high performance as WER (%) for mono, tri1, tri2a, tri3b, nnet2 and nnet3 reduced to $12.12, 7.2, 7.53, 7.8, 6.46, 4.34$ and 4.02 respectively. WER of 4.02% for nnet3 LSTM model is as good

as any state of the art recognition system for large vocabulary continuous speech.

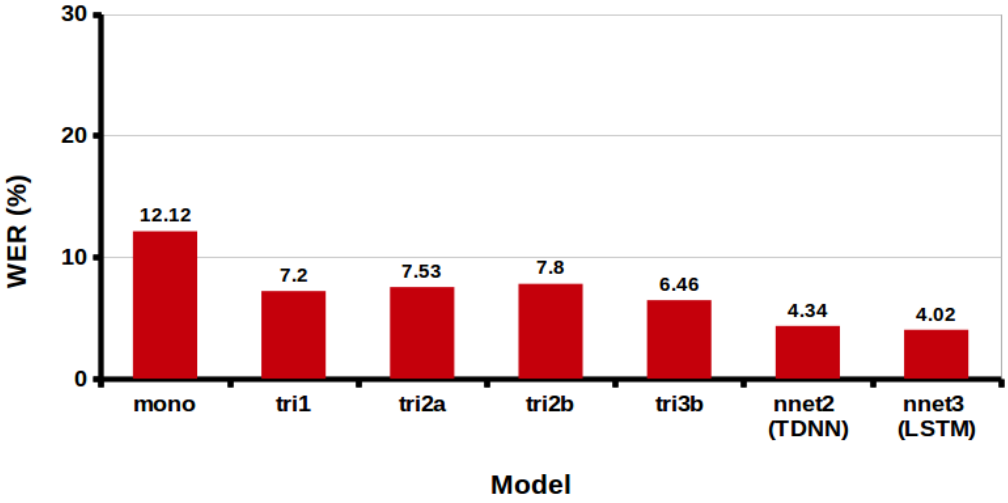

Figure 5.3: WER (%) of different acoustic models of system S_m on addresses test data using addresses LM

5.1.1 Additional Testing

Speaker Independent System Testing

So far, all the testing/decoding of system S_m's models was performed on the unseen testing data of speakers whom speech data was also included in the training data. To test the robustness of the system and see how it performs in case of speaker whose speech data is not included in the training, models of system S_m were tested on 50 random address transcripts of 4 different speakers not included in the training data. WER (%) for mono, tri1, tri2a, tri3b, nnet2 and nnet3 for these random speakers came out to be $10.15, 7.08, 8, 10.46, 4.62, 2.46$ and 2.13 respectively as shown in figure 5.4. Results indicate that acoustic models of system S_m are speaker independent

and there was no noticeable decrease in performance when tested on new speakers.

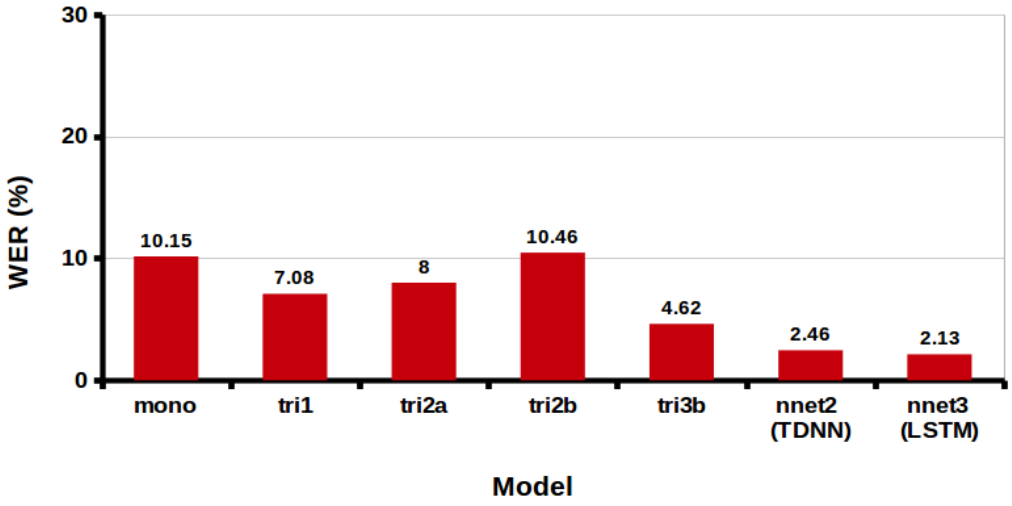

Figure 5.4: WER (%) of different acoustic models of system S_m on addresses test data from new speakers using addresses LM

Accent Based System Testing

Usually speech recognition systems trained for a language with a specific accent do not perform well with different accent of the same language. To show this, Librispeech dataset (for English Language)[1] was used for testing on Pakistani accent models. 50 random transcripts from Librispeech data were recorded in Pakistani accent from four different speakers. Using Librispeech language model, acoustic models of developed ASR system S_m and Librispeech's acoustic models were tested on Librispeech data in Pakistani and American accent. Figure 5.5 shows the comparison between WER (%) of Librispeech's tri3b[2] acoustic model and tri3b acoustic model of system S_m

[1] `http://www.openslr.org/12`

[2] `http://kaldi-asr.org/downloads/all/egs/librispeech/s5/exp/tri3b/`

that is 24.68 and 49.21 for American accent and 65.19 and 12.46 for Pakistani accent respectively. Figure 5.6 shows the comparison between WER (%) of

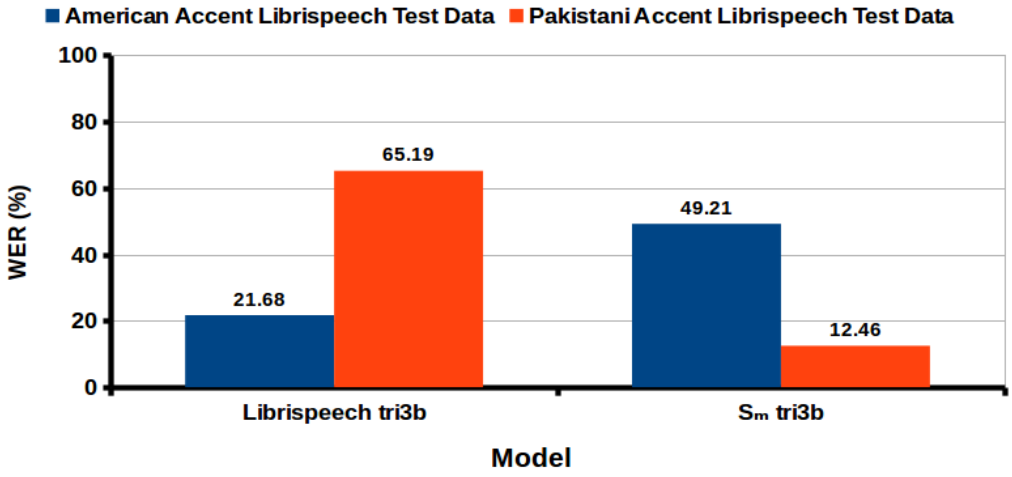

Figure 5.5: WER (%) of Librispeech tri3b vs S_m tri3b on Pakistani and American accent Librispeech recordings

Librispeech's nnet2[3] acoustic model and nnet2 acoustic model of system S_m that is 8.39 and 47 for American accent and 25.32 and 7.26 for Pakistani accent respectively.

Thus results suggest that acoustic models of system S_m, as compared to Librispeech's own model performs better for Librispeech data recorded in Pakistani accent.

Testing results presented in figures 5.5 and 5.6 were obtained using Librispeech dataset for which Librispeech's acoustic model were training. To further test both models, a separate dataset of 363 transcripts (duration 29.28 minutes) was prepared using Youtube English recordings in Pakistani accent. These transcripts were then tested on acoustic models of both systems i.e. Librispeech's and system S_m. Figure 5.7 shows the results of WER (%)

[3] `http://kaldi-asr.org/downloads/all/egs/librispeech/s5/exp/nnet2_online/`

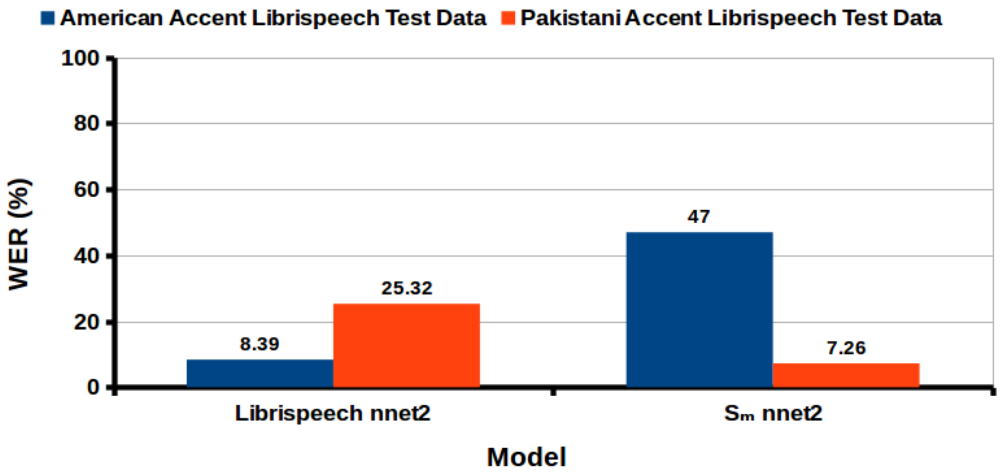

Figure 5.6: WER (%) of Librispeech nnet2 vs S_m nnet2 on Pakistani and American accent Librispeech recordings

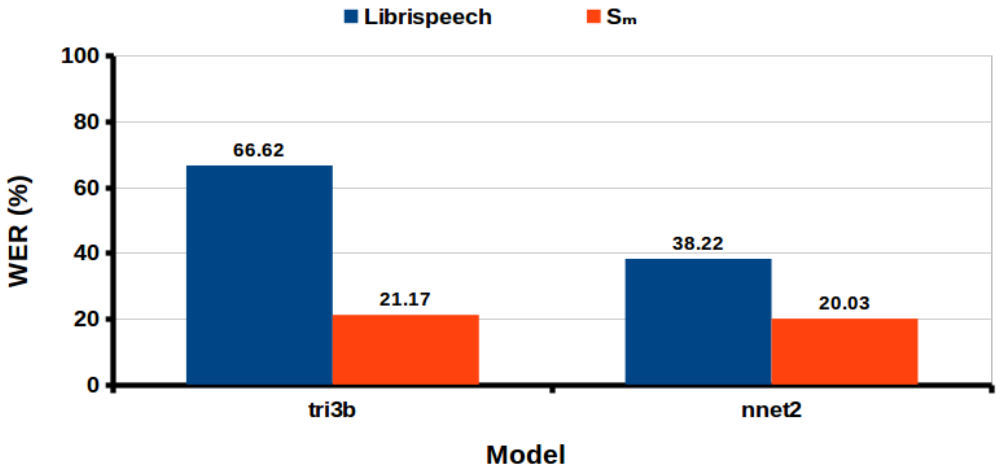

Figure 5.7: Average WER (%) of Librispeech models vs S_m models on Pakistani accent Youtube English test data

for Librispeech's acoustic models and S_m acoustic models respectively, that are 66.62 and 21.17 for tri3b and 38.22 and 20.03 for nnet2 on Pakistani accent Youtube English test data. Data contained in this test was collected from various Youtube videos of four different Pakistani speakers. Table 5.1

shows the detailed per speaker results for all speakers from whom these 363 transcripts were obtained.

Table 5.1: Detailed % WER of Librispeech vs system S_m models on Youtube English test data in Pakistani accent

Speaker	WER (%) *(Librispeech - tri3b)*	WER (%) *(Our - tri3b)*	WER (%) *(Librispeech - nnet2)*	WER (%) *(Our - nnet2)*
Speaker 1	65.09	22.86	32.15	19.31
Speaker 2	81.91	9.30	49.61	17.05
Speaker 3	60.91	23.40	40.17	23.83
Speaker 4	58.57	29.15	30.95	19.93
Average	**66.62**	**21.18**	**38.22**	**20.03**

Results in table 5.1 also indicate that system S_m models performed better than Librispeech models on English data in Pakistani accent. Which shows the need of ASR systems based on Pakistani/South Asian accent.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this study automatic speech recognition systems for general Urdu as well as roman Urdu addresses were successfully developed using Kaldi ASR toolkit. Two different acoustic modeling techniques, Gaussian mixture model and deep neural networks used to model the output accuracy of hidden markov models were compared. It was observed that models trained using deep neural network out performed conventional Gaussian mixture models in different testing scenarios. Along with regular testing using different language models, additional out of training speaker testing was performed on developed ASR system. Accent testing was also performed to demonstrate the need of training new models on Pakistani accent and how famous pre-trained models perform poorly on English in Pakistani accent. Results indicate that efficient Urdu speech recognition can be performed using Kaldi ASR toolkit and both GMM-HMM and DNN-HMM models were able to successfully transcribe general Urdu as well as roman Urdu addresses speech. Results in chapter 5 also show that DNN-HMM models performed better than GMM-HMM

models in all test cases.

6.2 Future Work

This study being first large vocabulary continuous Urdu speech implementation in Kaldi is a step towards improvements in Urdu speech recognition. To achieve a better performance from different models, focus should be put on more data collection to enrich the dataset. Future work of this study includes training and comparison of various DNN-HMM acoustic models on only general Urdu data to develop a best model for large vocabulary general Urdu speech recognition tasks. Also collection of more data to enrich this dataset further is one of the objectives of our research group. Another future work planned in the pipeline is identification of various applications on which Urdu speech recognition service can be applied like live Urdu speech transcription and mailing address transcription for automatic mailing address input.

Appendix

A1 - Frequently Asked Questions (FAQs)

- **Which toolkit was used for developing Urdu ASR system?** *Kaldi* a popular ASR toolkit is used to develop Urdu ASR system in this study.

- **Which dataset was used to develope the system and is it publicalty availble?**
  Dataset used to develope the system is collected by *Speech and Language Technology Group* [81] at *School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST)*. No, it is not a public dataset.

Bibliography

[1] Michal Borský. *Robust recognition of strongly distorted speech.* PhD dissertation, Czech Technical University, 2016.

[2] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin Lang. Phoneme recognition using time-delay neural networks. In *Readings in speech recognition*, pages 393–404. Elsevier, 1990.

[3] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014.

[4] Hasan Kabir and Abdul Mannan Saleem. Speech assessment methods phonetic alphabet (sampa): Analysis of urdu. In *CRULP Annual Student Report*. Akhbar-e-Urdu, 2002.

[5] Amanda Purington, Jessie Taft, Shruti Sannon, Natalya Bazarova, and Samuel Hardman Taylor. Alexa is my new bff: social roles, user satisfaction, and personification of the amazon echo. In *Proceedings of the CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 2853–2859. ACM, 2017.

[6] Gustavo López, Luis Quesada, and Luis Alfonso Guerrero. Alexa vs. siri vs. cortana vs. google assistant: a comparison of speech-based natural user interfaces. In *International Conference on Applied Human Factors and Ergonomics*, pages 241–250. Springer, 2017.

[7] Ye-Yi Wang, Dong Yu, Yun-Cheng Ju, and Alex Acero. An introduction to voice search. *IEEE Signal Processing Magazine*, 25(3), 2008.

[8] Geoffrey Zweig and Shuangyu Chang. Personalizing model m for voice-search. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[9] Yuqing Gao, Liang Gu, Bowen Zhou, Ruhi Sarikaya, Mohamed Afify, Hong-Kwang Kuo, Wei-zhong Zhu, Yonggang Deng, Charles Prosser, Wei Zhang, et al. Ibm mastor system: Multilingual automatic speech-to-speech translator. In *Proceedings of the Workshop on Medical Speech Translation*, pages 53–56. Association for Computational Linguistics, 2006.

[10] Farzad Ehsani, Demitrios Master, and Elaine Drom Zuber. Mobile speech-to-speech interpretation system, July 2 2013. US Patent 8,478,578.

[11] Kong Aik Lee, Anthony Larcher, Helen Thai, Bin Ma, and Haizhou Li. Joint application of speech and speaker recognition for automation and security in smart home. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[12] John Howard and Jean-claude Junqua. Automatic control of household activity using speech recognition and natural language, January 28 2003. US Patent 6,513,006.

[13] Steven 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, 1980.

[14] Olli Viikki and Kari Laurila. Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Communication*, 25(1-3):133–147, 1998.

[15] Reinhold Haeb-Umbach and Hermann Ney. Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-92.*, volume 1, pages 13–16. IEEE, 1992.

[16] Nagendra Kumar and Andreas Andreou. *Investigation of silicon auditory models and generalization of linear discriminant analysis for improved speech recognition.* PhD thesis, Johns Hopkins University, 1997.

[17] Ramesh Gopinath. Maximum likelihood modeling with gaussian distributions for classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 661–664. IEEE, 1998.

[18] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[20] Felix Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. In *9th International Conference on Artificial Neural Networks: ICANN '99*, pages 850–855. IET, 1999.

[21] Felix Gers, Nicol Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143, 2002.

[22] Gonzalo Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88, 2001.

[23] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[24] Steve Young, Julian Odell, and Philip Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the workshop on Human Language Technology*, pages 307–312. Association for Computational Linguistics, 1994.

[25] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

[26] KH Davis, R Biddulph, and Stephen Balashek. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, 24(6):637–642, 1952.

[27] Yen-Lu Chow, Mari Dunham, Owen Kimball, M Krasner, G Kubala, John Makhoul, Patti Price, Salim Roucos, and Richard Schwartz. Byblos: The bbn continuous speech recognition system. In *ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 12, pages 89–92. IEEE, 1987.

[28] Victor Zue, James Glass, Michael Phillips, and Stephanie Seneff. The mit summit speech recognition system: A progress report. In *Proceedings of the workshop on Speech and Natural Language*, pages 179–189. Association for Computational Linguistics, 1989.

[29] Nuance announces major new releases of dragon for windows and mac os x, powered by nuance deep learning technology. `https://www.nuance.com/about-us/newsroom/press-releases/dragon-new-releases-powered-by-deep-learning.html`, 2016. [Online; accessed 8-Dec-2019].

[30] Jahanzeb Sherwani, Nosheen Ali, Sarwat Mirza, Anjum Fatma, Yousuf Memon, Mehtab Karim, Rahul Tongia, and Roni Rosenfeld. Healthline: Speech-based access to health information by low-literate users. In *International Conference on Information and Communication Technologies and Development (ICTD)*, pages 1–9. IEEE, 2007.

[31] Christian Gaida, Patrick Lange, Rico Petrick, Patrick Proba, Ahmed Malatawy, and David Suendermann-Oeft. Comparing open-source speech recognition toolkits. *Tech. Rep., DHBW Stuttgart*, 2014.

[32] Daniel Povey, Nagendra Goel, Lukas Burget, Mohit Agarwal, Pinar Akyazi, Feng Kai, Arnab Ghoshal, Ondrej Glembek, Martin Karafiat, Ariya Rastrow, Richard Rose, Petr Schwarz, and Samuel Thomas. Low development cost, high quality speech recognition for new languages and domains, 2009.

[33] Janet Maciver Baker, Li Deng, James Glass, Sanjeev Khudanpur, Chin-Hui Lee, Nelson Morgan, and Douglas O'Shaughnessy. Developments and directions in speech recognition and understanding, part 1 [dsp education]. *IEEE Signal processing magazine*, 26(3), 2009.

[34] Sadaoki Furui. *Digital speech processing: synthesis, and recognition.* CRC Press, 2000.

[35] Bing-Hwang Juang, Stephene Levinson, and Man Mohan Sondhi. Maximum likelihood estimation for multivariate mixture observations of markov chains (corresp.). *IEEE Transactions on Information Theory*, 32(2):307–309, 1986.

[36] Hynek Hermansky. Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

[37] Sadaoki Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):254–272, 1981.

[38] Young Steve. Large vocabulary continuous speech recognition: a review. *IEEE Signal Processing Magazine*, 21:786–797, 1996.

[39] Hynek Hermansky, Daniel Ellis, and Sangita Sharma. Tandem connectionist feature extraction for conventional hmm systems. In *icassp*, pages 1635–1638. IEEE, 2000.

[40] Herve Bourlard and Nelson Morgan. *Connectionist speech recognition: a hybrid approach*, volume 247. Springer Science & Business Media, 2012.

[41] Steve Renals, Nelson Morgan, Hervé Bourlard, Michael Cohen, and Horacio Franco. Connectionist probability estimators in hmm speech recognition. *IEEE Transactions on Speech and Audio Processing*, 2(1):161–174, 1994.

[42] Tony Robinson, Mike Hochberg, and Steve Renals. The use of recurrent neural networks in continuous speech recognition. In *Automatic speech and speaker recognition*, pages 233–258. Springer, 1996.

[43] Abdel-rahman Mohamed, George Dahl, Geoffrey Hinton, et al. Acoustic modeling using deep belief networks. *IEEE Transaction on Audio, Speech & Language Processing*, 20(1):14–22, 2012.

[44] Frank Seide, Gang Li, and Dong Yu. Conversational speech transcription using context-dependent deep neural networks. In *Twelfth annual conference of the international speech communication association*, 2011.

[45] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pages 1764–1772, 2014.

[46] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays. Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*, 2015.

[47] Tara Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584. IEEE, 2015.

[48] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182, 2016.

[49] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 273–278. IEEE, 2013.

[50] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[51] Andrey Ronzhin, Rafael Yusupov, Izolda Li, and Anastasia Leontieva. Survey of russian speech recognition systems. In *Proceedings of 11th International Conference SPECOM*, pages 54–60, 2006.

[52] Solomon Teferra Abate, Wolfgang Menzel, and Bairu Tafila. An amharic speech corpus for large vocabulary continuous speech recognition. In *Ninth European Conference on Speech Communication and Technology*, 2005.

[53] Luis Villaseñor-Pineda, Manuel Montes-Y-Gómez, Dominique Vaufreydaz, and Jean-François Serignat. Experiments on the construction of a phonetically balanced corpus from the web. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 416–419. Springer, 2004.

[54] Gailius Raškinis and Danutė Raškinienė. Building medium-vocabulary isolated-word lithuanian hmm speech recognition system. *Informatica*, 14(1):75–84, 2003.

[55] Vassilios Digalakis, Dimitrios Oikonomidis, Dimitris Pratsolis, Nikos Tsourakis, Christos Vosnidis, Nikos Chatzichrisafis, and Vassilios Diakoloukas. Large vocabulary continuous speech recognition in greek: Corpus and an automatic dictation system. In *Eighth European Conference on Speech Communication and Technology*, 2003.

[56] Gopalakrishna Anumanchipalli, Rahul Chitturi, Sachin Joshi, Rohit Kumar, Satinder Pal Singh, RNV Sitaram, and SP Kishore. Development of indian language speech databases for large vocabulary speech recognition systems. In *Proceedings of SPECOM*, 2005.

[57] Aijun Li, Fang Zheng, William Byrne, Pascale Fung, Terri Kamm, Yi Liu, Zhanjiang Song, Umar Ruhi, Veera Venkataramani, and Xiaoxia Chen. Cass: A phonetically transcribed corpus of mandarin spontaneous speech. In *Sixth International Conference on Spoken Language Processing*, 2000.

[58] Diana Binnenpoorte, Catia Cucchiarini, Helmer Strik, and LWJ Boves. Improving automatic phonetic transcription of spontaneous speech through variant-based pronunciation variation modelling. In *Proceedings of LREC2004*, pages 681–684. Lisbon: ELRA, 2004.

[59] Jacques Duchateau, Tom Laureys, and Patrick Wambacq. Adding robustness to language models for spontaneous speech recognition. In *COST278 and ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*, 2004.

[60] Vivek Rangarajan and Shrikanth Narayanan. Analysis of disfluent repetitions in spontaneous speech recognition. In *14th European Signal Processing Conference*, pages 1–5. IEEE, 2006.

[61] Tilo Slobada and Alex Waibel. Dictionary learning for spontaneous speech recognition. In *Proceedings of Fourth International Conference on Spoken Language, ICSLP 96*, volume 4, pages 2328–2331. IEEE, 1996.

[62] Jon Nedel, Rita Singh, and Richard Stern. Automatic subword unit refinement for spontaneous speech recognition via phone splitting. In *Sixth International Conference on Spoken Language Processing*, 2000.

[63] Masataka Goto, Katunobu Itou, and Satoru Hayamizu. A real-time filled pause detection system for spontaneous speech recognition. In *Sixth European Conference on Speech Communication and Technology*, 1999.

[64] Hagen Soltau, George Saon, Brian Kingsbury, Jeff Kuo, Lidia Mangu, Daniel Povey, and Geoffrey Zweig. The ibm 2006 gale arabic asr system. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, volume 4, pages IV–349. IEEE, 2007.

[65] Agha Ali Raza, Sarmad Hussain, Huda Sarfraz, Inam Ullah, and Zahid Sarfraz. An asr system for spontaneous urdu speech. In *Proceedings of O-COCOSDA*, 2010.

[66] Javed Ashraf, Naveed Iqbal, Naveed Sarfraz Khattak, and Ather Mohsin Zaidi. Speaker independent urdu speech recognition using hmm. In *Proceedings of 7th International Conference on Informatics and Systems (INFOS)*, pages 1–5. IEEE, 2010.

[67] Hassan Satori, Hussein Hiyassat, Mostafa Haiti, and Noureddine Chenfour. Investigation arabic speech recognition using cmu sphinx system. *International Arab Journal of Information Technology (IAJIT)*, 6(2), 2009.

[68] Muhammad Usman Akram and Muhammad Arif. Design of an urdu speech recognizer based upon acoustic phonetic modeling approach. In *Proceedings of 8th International Multitopic Conference INMIC*, pages 91–96. IEEE, 2004.

[69] Sheikh Muhammad Azam, Zubair Ali Mansoor, Muhammad Shahzad Mughal, and Sajjad Mohsin. Urdu spoken digits recognition using classified mfcc and backpropgation neural network. In *Computer Graphics, Imaging and Visualisation, CGIV'07*, pages 414–418. IEEE, 2007.

[70] Abdul Ahad, Ahsan Fayyaz, and Tariq Mehmood. Speech recognition using multilayer perceptron. In *Proceedings of IEEE Students Conference, ISCON'02*, volume 1, pages 103–109. IEEE, 2002.

[71] Huda Sarfraz, Sarmad Hussain, Riffat Bokhari, Agha Ali Raza, Inam Ullah, Zahid Sarfraz, Sophia Pervez, Asad Mustafa, Iqra Javed, and Rahila Parveen. Large vocabulary continuous speech recognition for urdu. In *Proceedings of the 8th International Conference on Frontiers of Information Technology*, page 1. ACM, 2010.

[72] Muhammad Qasim, Sohaib Nawaz, Sarmad Hussain, and Tania Habib. Urdu speech recognition system for district names of pakistan: Development, challenges and solutions. In *Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 28–32. IEEE, 2016.

[73] Mohit Kumar, Nitendra Rajput, and Ashish Verma. A large-vocabulary continuous speech recognition system for hindi. *IBM journal of research and development*, 48(5.6):703–715, 2004.

[74] Mohammad Abushariah, Raja Ainon, Roziati Zainuddin, Moustafa Elshafei, and Othman Omran Khalifa. Natural speaker-independent arabic speech recognition sys-

tem based on hidden markov models using sphinx tools. In *International Conference on Computer and Communication Engineering (ICCCE)*, pages 1–6. IEEE, 2010.

[75] Agha Ali Raza, Sarmad Hussain, Huda Sarfraz, Inam Ullah, and Zahid Sarfraz. Design and development of phonetically rich urdu speech corpus. In *International Conference on Speech Database and Assessments, Oriental COCOSDA*, pages 38–43. IEEE, 2009.

[76] Sahar Rauf, Asima Hameed, Tania Habib, and Sarmad Hussain. District names speech corpus for pakistani languages. In *International Conference Oriental COCOSDA held jointly with Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pages 207–211. IEEE, 2015.

[77] Rats speech activity detection ldc2015s02. `https://catalog.ldc.upenn.edu/LDC2015S02`, 2015. [Online; accessed 8-Dec-2019].

[78] Huda Sarfraz, Sarmad Hussain, Riffat Bokhari, Agha Ali Raza, Inam Ullah, Zahid Sarfraz, Sophia Pervez, Asad Mustafa, Iqra Javed, and Rahila Parveen. Speech corpus development for a speaker independent spontaneous urdu speech recognition system. *Proceedings of the O-COCOSDA, Kathmandu, Nepal*, 2010.

[79] Bacha Rehmam, Zahid Halim, Ghulam Abbas, and Tufail Muhammad. Artificial neural network-based speech recognition using dwt analysis applied on isolated words from oriental languages. *Malaysian Journal of Computer Science*, 28(3):242–262, 2015.

[80] Hazrat Ali and Omar Farooq. A medium vocabulary urdu isolated words balanced corpus for automatic speech recognition. In *International Conference on Electronics Computer Technology (ICECT)*, 2012.

[81] Speech and language technology group. `http://speech.seecs.nust.edu.pk/`, 2018. [Online; accessed 5-Dec-2019].

[82] Urdu asr recording tool. `https://urduasr.com/audiorecording/`. [Online; accessed 9-Dec-2019].

[83] Maximilian Bisani and Hermann Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434–451, 2008.

[84] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[85] Vijayaditya Peddinti et al. *Low latency modeling of temporal contexts for speech recognition*. PhD thesis, Johns Hopkins University, 2017.

[86] Emelie Kullmann. Speech to text for swedish using kaldi. Master's thesis, KTH Royal Institute of Technology, School of Engineering Sciences, 2016.

[87] Muhammad Ali Basha Shaik, Zoltan Tuske, Muhammad Ali Tahir, Markus Nussbaum-Thom, Ralf Schluter, and Hermann Ney. Improvements in rwth lvcsr evaluation systems for polish, portuguese, english, urdu, and arabic. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[88] Agha Ali Raza, Awais Athar, Shan Randhawa, Zain Tariq, Muhammad Bilal Saleem, Haris Bin, Umar Saif Zia, and Roni Rosenfeld. Rapid collection of spontaneous speech corpora using telephonic community forums. *Proceedings of Interspeech Conference 2018*, pages 1021–1025, 2018.

[89] Tehseen Zia and Usman Zahid. Long short-term memory recurrent neural network architectures for urdu acoustic modeling. *International Journal of Speech Technology*, pages 1–10, 2018.

[90] Shubham Toshniwal, Tara Sainath, Ron Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao. Multilingual speech recognition with a single end-to-end model. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4904–4908. IEEE, 2018.