DETECTING MALICIOUS ACTIVITIES OVER TELEPHONE NETWORK FOR URDU SPEAKER

by

Sehar Gul

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Master of Science

(Electrical Engineering)

at the

NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY, PAKISTAN

June 2020

To my affectionate mother, my father and my siblings...

Acknowledgments

This thesis describes the research that I have conducted from February 2018 till June 2020 during my Ms Studies at Pakistan Navy Engineering College, a constituent college of National University of Science Technology (NUST) at its campus Pakistan Navy Engineering College (PNEC), Karachi.

I am really grateful to Almighty Allah for giving me the strength to successfully accomplish my task regarding detecting malicious activities over telephone network for Urdu speaker and achieving the required results. This thesis report has been written to fulfill the requirement of MS Degree program. I am heartily thankful to my advisor, Dr. Arshad Aziz, whose support, supervision and encouragement from the initial to the final stage enabled me to develop an understanding of the objective. Finally my special thanks to the GEC committee members Dr. Dur-e-Shahwar Kundi and Dr. Lubna Moin.

Lastly I put forwad my consents and regards to my family who supported me throughout my research phase. A most special expression of gratitude goes to my Parents who have been very patient and supportive.

Table of Contents

Page

Page

Chapter Page

List of Tables

List of Figures

Figure Page

Abstract

Telephone is one of most important invention in the fields of communication it is because on this invention that we are able to connect with our friends and families without hassle of travelling and going to their places,but some people are also using it for negative purpose therefore to secure this medium of communication is one of the most important issue of today as many malicious activities are taking place on this channel.Humanely it is not possible to tap each and every phone call so that one could find malicious activities that are being done.In order to find such malicious activities we need an automatic system that can automatically detect malicious voice activities, for that we have decided to develop an automatic speech recognition system that will detect malicious sentences in Urdu Language from the telephonic conversation which will then be processed further.

Chapter 1

Introduction

In this modern world communication through telephones have become one of the easiest and cheap way of communication with one another due to which lot of voice activities are taking place on this medium, in order to secure this medium people use to tap telephone call but tapping every phone call is challenging so in order to secure this medium one may convert speech into text and detect malicious activities if found then further processing could be done. In order to convert speech into text we require an automatic speech recognition system this system is highly language dependent and it requires lot of data to train it for any desired language. In spite of tremendous work done in applied and theoretical technology of ASR it is limited to English and some other languages where as Automatic Speech Recognition Research and application in Urdu Language is limited therefore developing an Urdu ASR is a challenging task as there is no data available for training an ASR and if we are making an ASR specifically for detecting malicious sentences in Urdu Language then it become more challenging.

The purpose of our thesis is to develop an Urdu ASR that will detect malicious and suspicious sentences from the telephonic conversation as in Pakistan security agencies use to block mobile signal on special occasion so the only medium left for communication is telephone lines and to secure this medium we have proposed above solution.

1.1 Scope of thesis

Our thesis is mainly based on converting speech into text and then detecting malicious activities over telephonic conversion through this text. We will make our own model that model will be trained to detect malicious and suspicious words, phrase and sentences spoken in Urdu language. We will develop an automatic speech recognition system that will detect words and sentences in Urdu language. This system will help security agencies to keep track of such people who make use of telephone as a mean of spreading terrorist activities in Pakistan. This project will be a great help in controlling terrorist activities in Pakistan as this nation is being suffering from terrorist activities since long time so its high time we as a responsible citizen should make use of science and technology and develop such systems that should help our security agencies to catch red hand such people who are involve in crime and corruption and for that we have taken an initiative and develop such a system that will keep record telephonic call from a peasant farmer to the Prime Minister.

1.2 Chapter Structure

The remaining portion of the thesis is Systematized as follows: chapter 2 gives the brief description of traditional ASR system and an end to end ASR system.In Chapter 3, we have discussed what is machine learning and types of machine learning.

Chapter 4 presents previously work done in developing automatic speech recognition system.In Chapter 5, we have discussed the flow of our work in developing our ASR.

Implementation results and performance comparison between the two platform is given in Chapter 6.Finally, the proposed work has been summed up and some future directions are recommended in Chapter 7.

Chapter 2

Speech Recognition System

Speech is a primary means of communication between humans. Automatic speech recognition(ASR) is independent computer driven transcription of spoken language into readable text in real time. Automatic Speech Recognition is a technique that automatically translates incoming speech signals into their contextual information via sequence of words or phrases or other linguistic units by means of an algorithm implemented as a computer program. Based on major advances in statistical modeling of speech in the 1980s automatic speech recognition system today are frequently considered as a key technology for human machine communication and are incorporated in numerous useful application.

2.1 Traditional Speech Recognition System:

Traditional speech recognition system is a system that is probabilistic in nature, all the components that are present in this system evaluate the probabilities of a word given the audio which is then fed into the decoder to find the most probable sequence of words.
The figure 2.1 is showing the structure of traditional speech recognition system, it will provide a brief concept of how this recognition system used to work.

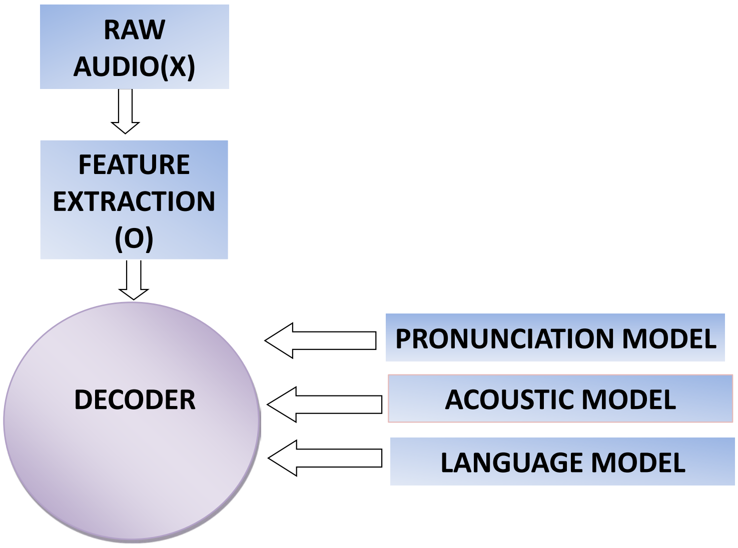


Figure 2.1 Traditional Speech Recognition System

2.1.1 Feature Extraction

In traditional ASR pipeline an audio X is given which is converted into some kind of signal in which some features are extracted so that further processing can easily be applied on that. Feature extraction is the heart of speech recognition system, it is one of the most important part in an ASR system. Following are the techniques for extraction of features [1]

1. Mel Frequency Cepstral Coefficient
2. Linear Predictive Coefficient
3. Discrete Wavelet Transform
4. Perceptual Linear Prediction

2.1.1.1 Mel Frequency Cepstral Coefficient

In MFCC first windowing is applied to the continuous speech and then Fast Fourier Transform is applied to change it in frequency domain after that Mel scale filter is applied to filter low frequency component then log of the Mel-scale is taken after that DCT(Discrete Cosine Transform) is applied to extract energy of the speech sample.[1]

2.1.1.2 Linear Predictive Coefficient

The linear predictive Coefficient technique help to extract features like pitch period, energy, formants by linearly predicting from the past sample to obtain the present sample using all zeros filter [1]

2.1.1.3 Discrete Wavelet Transform

In this signal is broken down into high and low frequency by passing it through a high and low pass filter to obtained the approximate and detail component for the next stage.[1]

2.1.1.4 Perceptual Linear Prediction

In PLP first preprocessing of audio is done after that FFT is applied then filtering is done after that IDFT is applied with Linear Prediction analysis and Cepstral analysis to obtain energy of the given signal.[1]

2.1.2 Acoustic Model

An acoustic model $P(O \mid W)$that has relationship between the features and the words that are being used.[2].It models the acoustic of the signal,it predicts the probability of each phoneme occurring in a short frame of audio.

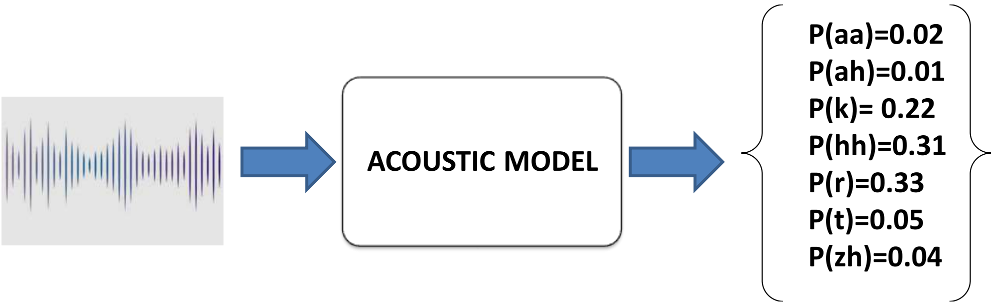


Figure 2.2 The Acoustic Model

2.1.3 Language Model

Language Model P(W) which have alphabets or words that are being used in that specific language.[3]It learns which words of sequence are most likely to be spoken and it predicts which words will follow on from the current word and with what probability.

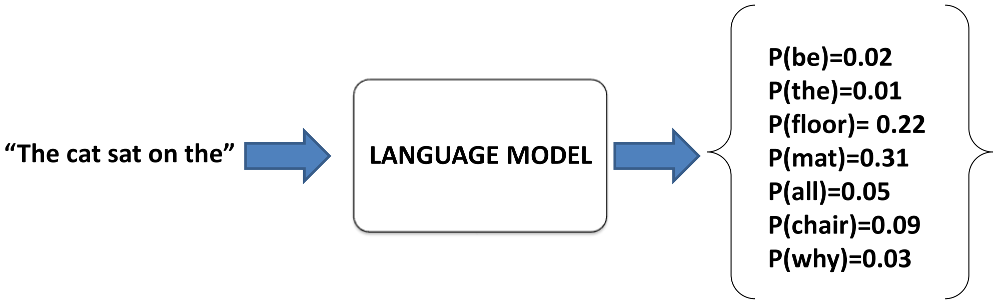


Figure 2.3 The Language Model

2.1.4 Pronunciation Model

If an audio is to be transcribed it cannot be directly converted into words because instead of predicting characters it will predict phonemes (phonemes are the perceptually distinct unit of sound). For example if I have a word Hello for that I will use phonemes like [HH AH L OW] so now the model is not only associated with words but phonemes as well therefore another component was added in the pipelines that is pronunciation model P(Q $|W)$.

2.1.5 Decoder

These language and acoustic model will be driven by some machine learning algorithm which we called as decoder and the job of the decoder is to find the sequence of words W that maximizes the probability.

$$W* = argmaxP(W|X) \quad [2.1]$$

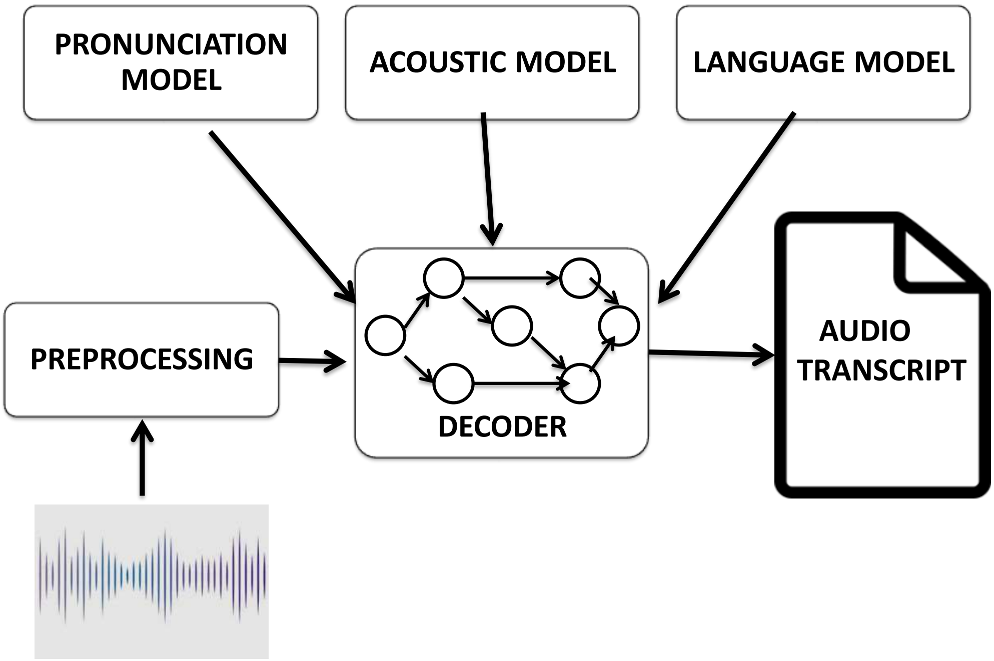


Figure 2.4 The Decoder

2.2 Drawback In Traditional ASR System

The drawback of traditional ASR was that it was very complicated to develop and If you were new then debugging and making such a complicated algorithm was difficult to develop and if you have to consider accent, heavy noise and speaker variability then it became more complicated.
To avoid these problems traditional ASR is replace by an end to end speech recognition system which is using deep learning technique to convert given audio into its actual transcript.[4]

2.3 End to End Speech Recognition System

In traditional ASR the machine learning techniques were implemented in the decoder therefore it was difficult to develop acoustic and language model that have probabilistic nature which made the system complicated on the other hand the new ASR system which is develop will be running on machine learning techniques from the extraction of audio features till the transcription of the audio which made it less complex. The latest ASR system is totally based on machine learning techniques in which the machine learns by its own without being programmed, it is an application of artificial intelligence that provides the system an ability to learn and improve from experience. An end to end deep learning speech recognition system employed machine learning techniques which were Recurrent Neural Networks (RNN) and Connectionist Temporal Coefficients (CTC).

2.3.1 Representation of Audio

An audio is a 1D signal that has just 1 dimension unlike others signals that have 2 or 3 dimensions. This 1D signal is sampled at 8 or 16 KHz that means each signal have 8000 or 16000 samples. These samples are considered as vector (X1, X2) that means if you have a 2ms wave it will be quantized on 8 or 16 bit and each value contains floating point number that is extracted from this 8 or 16 bit samples.[4]

2.3.2 Pre-Processing Of Audio Signal

During pre-processing the audio signal is converted into spectrogram this spectrogram which losses some of its information while converting into spectrogram is further used in acoustic model. Spectrogram gives a frequency domain representation of the signal in which a 20ms audio is being captured using a window this window consist of a different frequencies of sine wave. This audio signal is then converted into frequency domain by applying FFT. Taking the log of this frequency will give you the strength or power of each component of frequency these power or magnitude of signal represented by each frequency can be considered as vector. Now apply this bunch of windows on the whole audio signal these windows can be overlapping or disjoint windows. In this way we will get a bunch of vectors aligned which will be represented as spectrogram.[4]

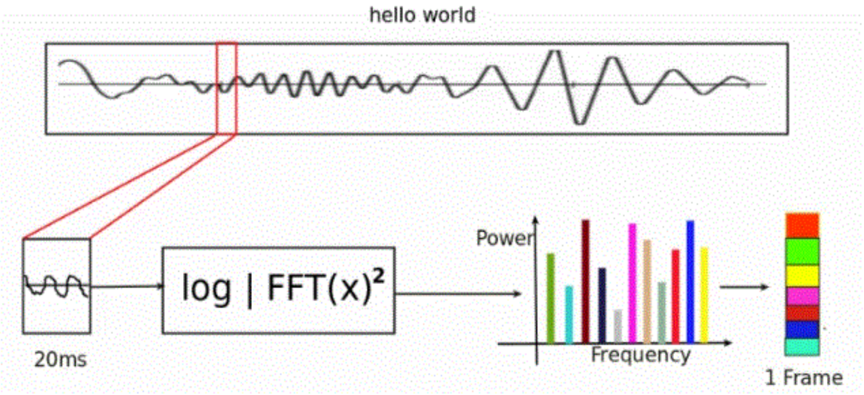


Figure 2.5 The feature extraction from raw audio

2.3.3 Acoustic Model

An acoustic model represents the whole ASR engine that is driven by recurrent neural networks and connectionist temporal classification (CTC).

The spectrogram feeds in the recurrent neural networks which is train to produce some output that is represented by the character C that will produce the correct transcription of the given audio. For example I have said a word Hello so the first thing that the ASR will do is pre-processing that means it will convert the audio signal into spectrogram which contain the vector of every 20ms frame the output of this spectrogram is feed into the neural network to produce an output that is represented by C. The problem here is that the length of the output signal is not equal to the length of our transcription for example if I say hello very slowly then we have a long audio signal whose length is not equal to the length of my transcription and if I say very quickly then my length of the signal will be short as compared to the transcription that means the neural network is changing length.

This problem can be resolve using CTC (Connectionist Temporal Classification).The output of the neural network is represented by C called softmax neurons and the job of this output neurons is to encode a distribution over the output symbols, since these softmax neurons are the output of recurrent neural networks therefore the length of the sequence C is the same as the audio signal. Once RNN gives a distribution over these symbols C, now consider C itself as a probabilistic creature there will be distribution over the choices of C according to the given audio. Now train the network to maximize the probability of the correct transcription given the audio[5] Following are the steps to make CTC work

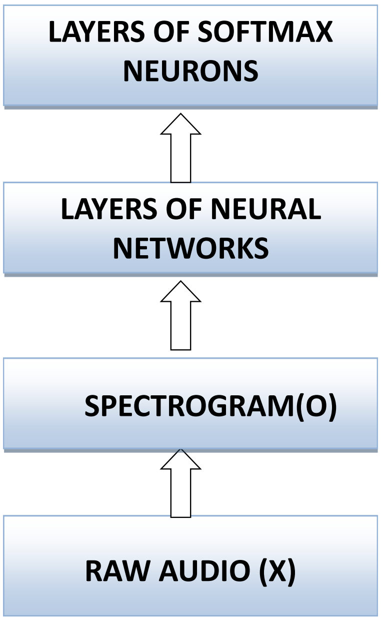


Figure 2.6 The Acoustic Model

2.3.3.1 ENCODE DISTRIBUTION OVER SYMBOLS

We have the output C that gives the distribution over these symbols according to the given audio X. The softmax neurons give the probability of each symbol represented according to the audio.[5]

For example C1,7 represents that in row 1 and column 7 the probability of the occurrence of B given the audio. Now applying this on not just one character but on the whole characters represented by the symbol C. Suppose I have a word WWW O RR LD this is a string in this alphabet C.

$$P(C|X) = \Pi_{i=1}^{N} P(Ci|X) \qquad [2.2]$$

$$P(C = WWW\_O\_\_RR\_LD\_\_|X) = P(C1 = W|X)P(C2 = W|X)P(C10 = blank|X) \qquad [2.3]$$

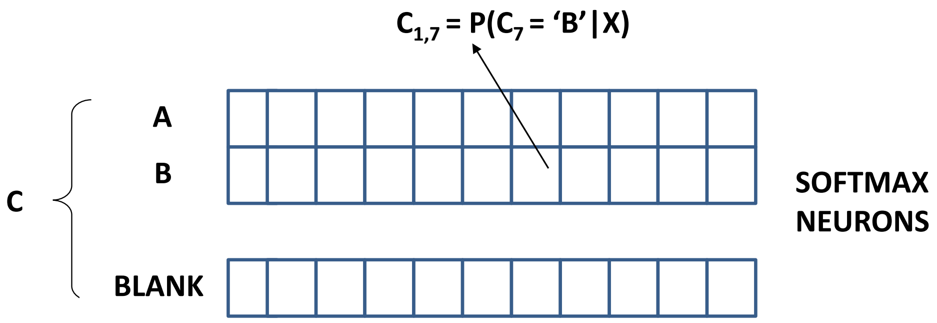


Figure 2.7 The probability assign to each alphabet for a given audio signal

I can use the above formula to compute the probability of the specific sequence of character. In this way we compute the probability of the sequence of character which have the same length as our audio signal.[6]

2.3.3.2 Define Mapping $\beta(C) = Y$

This operator squeeze the given symbols into the actual transcribe that we have predicted it will remove the duplicates by taking just one of them remove spaces and if the two character are different next to each other it will keep both of them.

$$Y = \beta(C) = \beta(WWW\_O\_RR\_\_LD\_\_) = WORLD \qquad [2.4]$$

There are other ways of sequence to write the same transcription so what you can do is to sum up the probabilities of all the possible alignments as this will produce according to the way the speaker has uttered it. You will sum up all the possible choices of C that could give the transcription in the end.

P(C$|X) = \{0.1 WWW__O_RR__LD__ = WORLD$

0.02 WW__OOO__R_LD =WORLD

0.01W__OO__RR_L_D =WORLD

0.001 WW__O_OR_LD__ =WOORLD }

$$P(Y|X) = \sum_{C:\beta(C)} = Y P(C|X) \qquad [2.5]$$

P(WORLD)=0.1+0.02+0.01.

2.3.3.3 Maximize Likelihood

The job is to tune the neural network to maximize the probability of that transcription using the model defines.

$$\Theta * = arg_{\theta} Max_{i} log P(Y *^{(\ i)} | X^{(i)}) \qquad [2.6]$$

2.3.4 Language Model

It is impossible that our system correctly predicts all the transcription as sometimes the transcription contain some words that it had never seen before in their training and for that we need to train our model with huge amount of data which is impractical so we could employ n- gram model these n-gram models[6] is trained from a corpus of million phrases supporting a vocabulary of thousands of words [5].

2.3.5 Beam search Decoder

After going through the language model which determines the probability of sequence of words by considering the corpus that is provided to it.The beam search decoder will then look for the most likelihood words from the acoustic and language model and will finally decode the sequence of words.

Chapter 3

Machine Learning

Machine learning is an application of artificial intelligence that has the ability to learn and improve from experience; it has the ability to learn by its own without being explicitly programmed. [7] There are two categorized of machine learning supervised and unsupervised learning.

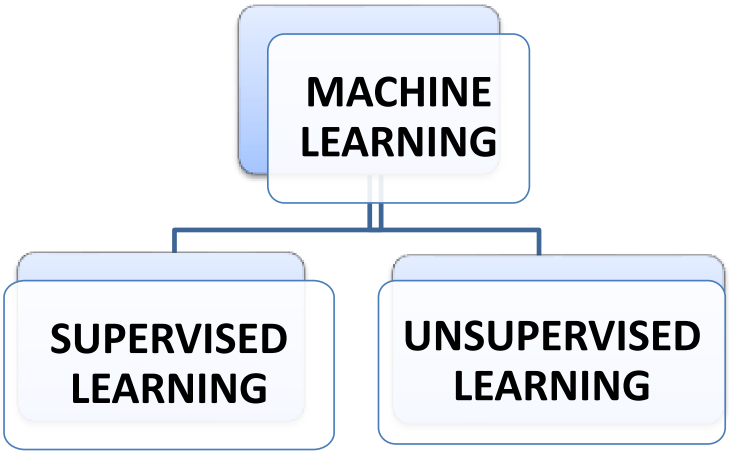


Figure 3.1 Types of Machine Learning

3.1 Supervised Learning

In supervised learning you used an algorithm to learn the mapping function from the input to the output. All the data is labeled and the technique learns to predict the output from the input data.[7]

3.2 Unsupervised Lerning

In unsupervised learning the input is present and no corresponding output variable is present. All the data is unlabeled and the techniques learn by similarities, pattern recognition from the input data.[7]

3.3 Methods Of Machine Learning

There are 10 methods for machine learning[8].

1-Regression

2-classification

3-clustering

4-Dimensional Reduction

5-Ensemble Method

6-Neural net

7-transfer learning

8-Reinforcement Learning

9-Natural Language processing

10-Word Embedding

From the ten methods of machine learning we have employed the neural networks for our project. The detail structure of neural network is discuss below.

3.3.1 Neural Net Our Employed Method

Neural networks which is also known as connectionist computational system provides the best solutions in different fields such as security and health. Artificial neural networks derive the concept of complex computation from human brain. The human brain can be described as biological neural network an interconnected web of more than 100 billion neurons transmitting elaborate pattern of minute power of electrical signal. The overall structure contains three main layers which include the input layer, hidden layer and the output layer. Input layer also known as dendrites where input signal are received and after performing all the computation in the hidden layers it provides us the desired output through output layer known as axon. In the same way artificial neural networks are designed which contains a number of neurons divided into three main layers (input layer, hidden layer and output layer) they are interconnected with each other to perform computational task which is often referred to as pattern recognition.

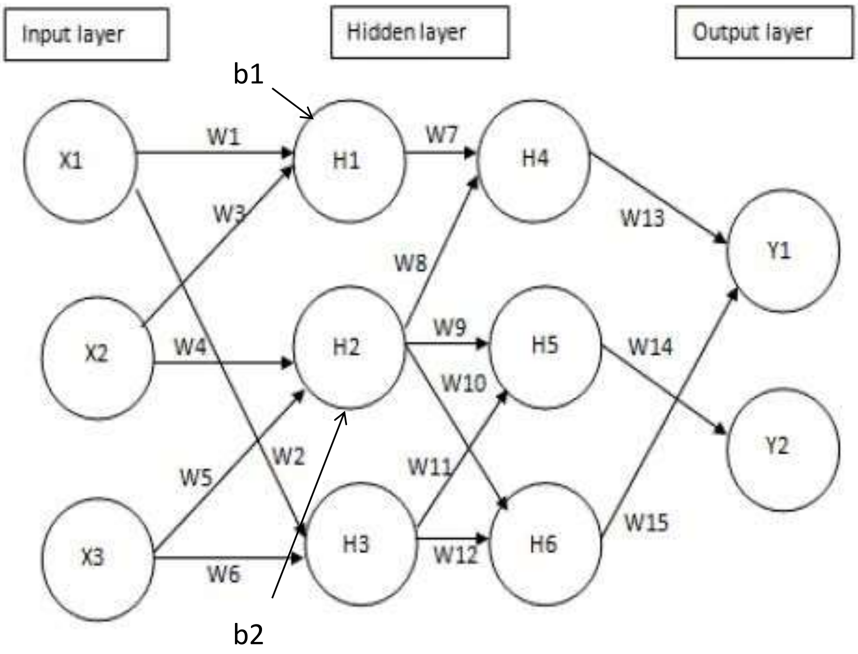


Figure 3.2 Shows architecture of neural networks that contain input layer, hidden layers and output layers

3.3.2 Working Of Neural Network

According to the figure 3.1 the neural network contain three inputs X1, X2 and X3, hidden layer that contain Relu(Rectified Linear unit) activation function which will give zero for any negative value and same value for the positive values the hidden units are H1, H2, H3, H4, H5, and H6 then it has the output layer that will be Y1 and Y2. The neural network also contains biases in our case there are two biases b1 and b2.

3.3.2.1 Forward Propagation

In order to achieve are target values we will first perform forward propagation in forward propagation we will first compute H1 from the following equation;

$$H1 = X1 * W1 + X2 * W3 + b1 \tag{3.1}$$

where H1 is an activation function, X1 is the input,W1 is the weight and b1 is the bias
In this way we will calculate H2 and H3
Next we will calculate H4 from the following equation:

$$H4 = H1 * W7 + H2 * W8 + b2 \tag{3.2}$$

In the same way we will calculate H5, and H6
After calculating all the activation function using input, weights and biases we will then calculate the output Y1 and Y2 using the following equation respectively

$$Y1 = H4 * W13 + H6 * W15 \tag{3.3}$$

where Y1 is the first output

$$Y2 = H5 * W14 \tag{3.4}$$

Once the output is calculated we will then calculate the error as we have set our output target.
To compute the error we have the following equation:

$$\Xi ETotal = \Xi(1/2)(target - output)^2 \tag{3.5}$$

In order to reduce this loss we will we will update our weights and for that we will have to perform backward propagation.

3.3.2.2 Backward Propagation

In back Propagation weights are updated so as to achieve the target values and to minimized the loss that is calculated by subtracting the output target to the actual target. We will first update the weights of W13 for that we have the following equations

$$\partial ETotal/\partial W13 = \partial ETotal/\partial outY1 * \partial outY1/\partial Y1 * \partial Y1/\partial W13 \quad [3.6]$$

After calculating we will now calculate the updated value of W13

$$UpdateW13 = W13 - \eta * \partial ETotal/\partial W13 \quad [3.7]$$

In this way we will update the weights W14 and W15
Now we will move to the second hidden layer and will update the weights of this layer

$$\partial ETotal/\partial W7 = \partial ETotal/\partial outH4 * \partial outH4/\partial H4 * \partial H4/W7 \quad [3.8]$$

Now we will update the weights

$$UpdateW7 = W7 - \eta * \partial ETotal/\partial W7 \quad [3.9]$$

In this way we will update the weights of W8, W9, W10, W11 and W12
Now we will move to the next layer and will update the weights

$$\partial ETotal/\partial W1 = \partial ETotal/\partial outH1 * \partial outH1/\partial H1 * \partial H1/\partial W1 \quad [3.10]$$

Updating the weights in the following equation:

$$UpdateW1 = W1 - \eta * \partial ETotal/\partial W1 \quad [3.11]$$

In this way we will update the weights of W2, W3, W4 and W5. Once all the weights are updated we will then perform forward propagation with the newly updated weights this will reduce our error as we will reach near our targeted value the forward and backward propagation will be repeated again and again until we will achieve our target value.

3.3.3 Application Of Neural Network In Our ASR

The speech signal is split into frames these frames are then fed into neural network as input. Our ASR consist of first three non recurrent layers which extract features from the frames at each time step, then it is fed into the fourth layer of neural network which is our bidirectional recurrent neural network where it explore the content of the speech. It add the results from forward and backward direction together for each time step then it applies the fifth layer which is a non recurrent layer to transform the results, finally the probability for each character at each time step is computed by softmax neurons. These softmax neurons convert numeric value that is obtained from the neural network into probability of each character at each time step so that the entire output vector add upto one.

Chapter 4

Literature Review

In past lot of research have been done in the field of human interfacing with machine, due to its application in various fields such as in mobiles phones, military equipment , automobiles , content captioning and in many other application have made the researcher to make it more effective and robust, for that lot of research and development is made in the improvement of automatic speech recognition system. Initially it was only made for isolated words in different languages but now it has been developed for continuous speech as well in different languages.

R.sandanalakshmi et al have proposed an efficient speech to text converter for mobile application using VAD(voice activity detection) and extracting feature using Mel frequency Cepstral Coefficient(MFCC) and then training General Regression Neural Networks(GRNN), the system was made fo specfic speaker.[9]. Nuzhat Atique Nafis and Md.Safaet Hossain proposed speech to text conversion in real time for the handicapped it uses two softwares namely visual studio and Matlab to develop the proposed system, it was made for American accent so if someone is speaking with any other accent it will write it incorrectly. [10] Neha Sharma and Shipra Sardana proposed speech to text converter by extracting features using MFCC and eliminating noise using Kalman Filter words were distinguished according to feature matching of each sample words.The system that contain filter take large time to filter out noise.[11] Hazrat Ali et al proposed a speech recognition system for Urdu Language which was done using using linear discriminant analysis in which 52 MFCC was calculated for isolated Urdu words, the number of correct or incorrect matches can be found using confusion matrix.The system was developed to recognize Urdu words.[12] Syed Abbas Ali et al in which a language translator was developed to convert Urdu speech into English Speech for that an effective ASR(Automatic Speech recognition) was developed to convert urdu speech into text using deep neural network.During testing it was observed that it worked well for recorded speech but gave poor performance while tested in real time. [13] Thiang, et al. (2011) developed speech recognition system using Linear Predictive Coding (LPC) and Artificial Neural Network (ANN) for controlling the movement of mobile robot. Input signals were sampled directly from the microphone and then the feature extraction was done by LPC and classification was done using ANN.The system was trained for seven words for commanding the robot.[14] Ms.Vimala.C and Dr.V.Radha (2012) proposed speaker independent isolated speech recognition system for Tamil language. Hidden Markov Model was implemented for Feature extraction, pronunciation dictionary, acoustic and language model which produced 88% of accuracy for 2500 words.The model was developed only for isolated words.[15] Cini Kurian and Kannan Balakrishnan (2012) found evaluation and development of different acoustic models for Malayalam continuous speech recognition. In this paper Hidden Markov model is used to compare and evaluate the Context Dependent (CD), Context Independent (CI) models and Context Dependent tied (CD tied) models from this CI model 21%. The database consists of 21 speakers including 11 females and 10 males. [16] Suma Swamy et al. (2013) developed an efficient speech recognition system which employed Mel Frequency Cepstrum Coefficients (MFCC), Vector Quantization (VQ) and Hidden Markov model which recognize the speech by 98% accuracy. The database consists of five words spoken 10 times by 4 speakers.The model was developed for isolated words in English[17]. Annu Choudhary et al. (2013) presented an automatic speech recognition system for connected and isolated words of Hindi language by using Hidden Markov Model Toolkit (HTK). Hindi words feature extraction was done using MFCC the recognition system achieved 95% accuracy in isolated words and 90% in connected words.[18] Preeti Saini et al. (2013) presented Hindi automatic speech recognition using Hidden Markov Model Toolkit. Isolated words were used to recognize the speech with 10 states in HMM topology which gave 96.61% accuracy.[19] Md. Akkas Ali et al. (2013) developed automatic speech recognition technique for Bangla words. Feature extraction was done by, Linear Predictive Coding (LPC) and classification was done using Gaussian Mixture Model (GMM). Total100 words were recorded 1000 times which gave 84% accuracy.The problem with this system was that it was developed for isolated words.[20] Maya Moneykumar, et al. (2014) presented

Malayalam word identification for speech recognition system. The proposed work was done with syllable based segmentation using HMM and Feature extraction using MFCC.The system would detected only words in Malayalam[21] Jitendra Singh Pokhariya and Dr. Sanjay Mathur (2014) proposed Sanskrit speech recognition using Hidden Markov Toolkit MFCC and HMM two states were used for extraction which produces 95.2% to 97.2% accuracy respectively.[22] In 2014, Geeta Nijhawan et al. presented real time speaker recognition system for Hindi words. Feature extraction done with MFCC using Quantization Linde, Buzo and Gray (VQLBG) algorithm. Voice Activity Detector (VAC) was proposed to remove silence part.[23]

Chapter 5

Our Work

We have developed an automatic speech recognition system which is detecting malicious words, phrases or sentences in Urdu language. Our speech to text conversion is an end to end speech recognition system in this system from capturing of input speech to the conversion into text is done using machine learning algorithm therefore our automatic speech recognition system is termed as an end to end speech recognition system . The figure 5.1 is showing how our speech will be converted into transcript. In figure 5.1 we can see an ASR block that takes the input from a microphone and convert it into text how it will convert it into text will be discuss below.

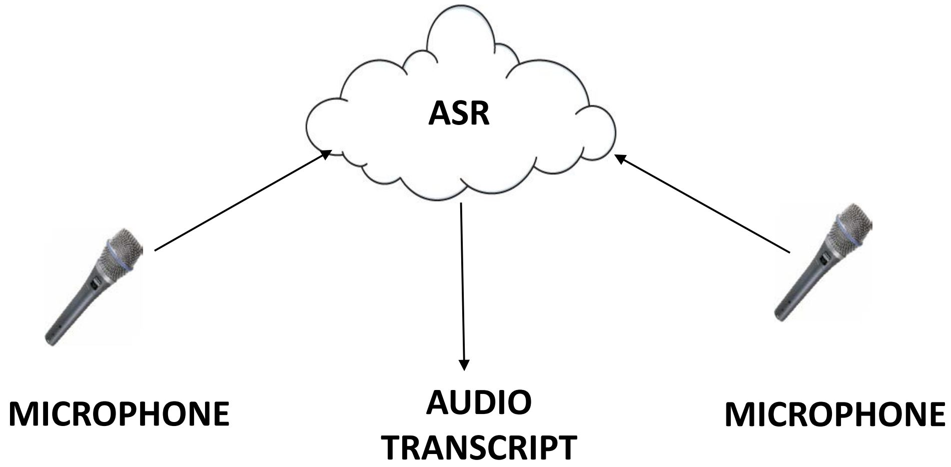


Figure 5.1 Showing the flow of an end to end speech Recognition system

5.1 Internal Architecture Of Our Speech Recognition System

The figure below is showing the internal architecture of automatic speech recognition system.

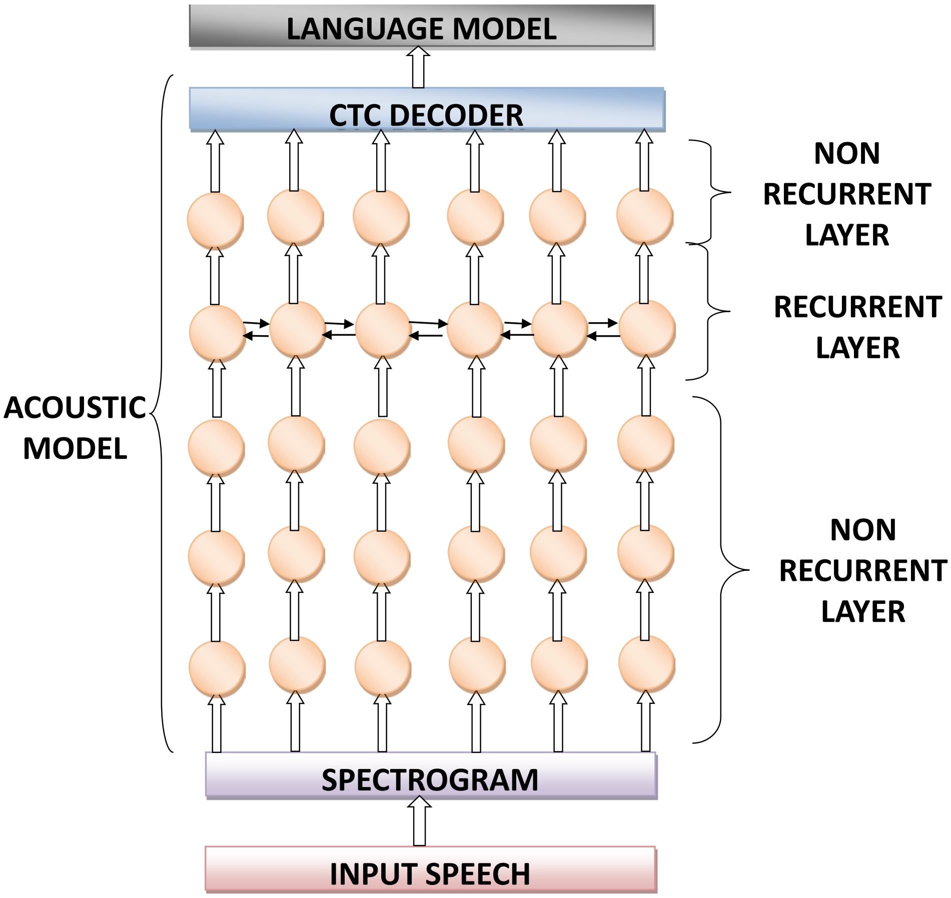


Figure 5.2 Showing internal architecture of our ASR system

5.1.1 Input Speech

The raw input speech is taken from a microphone or from a receiver of a telephone.In order to acquire useful information from the raw input speech we first need to extract features from that raw input speech that is done by making speech spectrogram in which first the raw input is splitted into 20ms windows on each window is then applied Fast Fourier Transform to acquire it in frequency domain after that the log of this frequency domains is taken so that we can compute the power of each frequency that power of the frequency is then converted into a vectors in this way all the vectors are arranged making a band of different frequencies called Spectrogram.[5]

5.1.2 Acoustic Model

The acoustic model is created in an end to end speech recognition system using neural networks and Connectionist temporal classification(CTC), in our case we have five layer of neural networks the first three layer of neural networks are non recurrent and provide the output using the following equation:

$$h_t^{(l)} = g(W^{(l)}h_t^{(l-1)} + b^{(l)}) \qquad [5.1]$$

where g(z) is an Relu activation function and W(l) and b(l) are the weights and biases respectively. Then after the three non recurrent layers we have the fourth layer that is called the recurrent layer having forward and backward propagation. The fourth layer is computed using the following equation:[5] For forward Propagation:

$$h_t^{(f)} = g(W^{(4)}h_t^{(3)} + Wr^{(f)}h^{(f)}{}_{(t-1)} + b^{(4)}) \qquad [5.2]$$

For backward Propagation:

$$h_t^{(b)} = g(W^{(4)}h_t^{(3)} + Wr^{(b)}h^{(b)}{}_{(t-1)} + b^{(4)}) \qquad [5.3]$$

then we have the fifth layer which takes both the backwards and forward units as input the output of fifth layer can be computed using the following equation

$$h_t^{(5)} = g(W^{(l)}h_t^{(4)} + b^{(5)}) \qquad [5.4]$$

where

$$h_t^{(4)} = h_t^{(f)} + h_t^{(b)} \qquad [5.5]$$

the output of the fifth layer is a series of softmax function these can be computed using the following equation:

$$h^{(6)}{}_{t.k} = y_{t,k} = P(c_t = k|x) = exp(W_k^{(6)}h_t^{(5)} + b_k^{(6)})/\sum jexp(W_j^{(6)}h_t^{(5)} + b_j^{(6)}) \qquad [5.6]$$

where Wk and bk denote the kth column of the weight matrix and kth bias respectively.[5]

5.1.3 Language Model

When training with large amount of data our RNN will now be capable of providing correct transcript of the given input speech. Most of the transcriptions are predicted correctly by our RNN. Many of the errors occur on words that are rarely or never occurred for that we have to develop a language model that is integrated by N-gram language modeling [6] since these models are easily trained from huge unlabeled text corpora.

5.2 Flow Of Our Work

For developing our malicious detecting speech recognition system we will go through the following process.

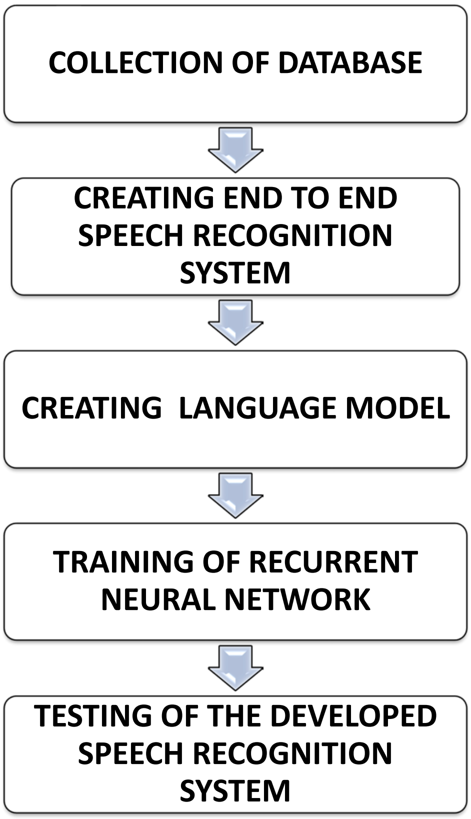

COLLECTION OF DATABASE
CREATING END TO END SPEECH RECOGNITION SYSTEM
CREATING LANGUAGE MODEL
TRAINING OF RECURRENT NEURAL NETWORK
TESTING OF THE DEVELOPED SPEECH RECOGNITION SYSTEM

Figure 5.3 Showing Flow of work

5.2.1 Collection Of Database

The most important task for making an automatic speech recognition system is to first develop a meaningful dataset as we are working on Urdu language which is spoken mostly is subcontinent and specially in Pakistan so we could easily have a lot of dataset but unfortunately there is a lack of data available in Urdu Language we tried to search for Urdu dataset on different websites but unfortunately we were lacking for having an Urdu Language dataset and to be very specific we were unable to get dataset for malicious and suspicious Urdu sentences which was our goal, so the main challenging task was to develop a malicious and suspicious Urdu sentences uttered by different people with different accent.

We have acquired a dataset from Urdu Asr website [24] but they were just isolated randomly uttered words while we wanted to develop such an ASR system that will detect malicious sentences. So our first important task was to develop a dataset that contain sentences having suspicious and malicious sentences.

We actually developed two datasets first with malicious sentences and then with some positively uttered sentences. So in order to develop an end to end speech recognition system we need to have the speech with the following parameter along with the comma separated files that contain wav file name, wav file size and their transcript.

The parameters for input speech were:

Sampling Rate	16000Hz
Channel	Mono
File Format	wav

Table 5.1 Wav Parameters

5.2.1.1 First Dataset

The first dataset that we have obtained from the Urdu Asr website contain 250 randomly spoken words that were spoken by 10 different people, in this way there were 2500 wav files of the randomly spoken word.

5.2.1.2 Second Dataset

The Second dataset was developed by us with the help of Post graduate Students Enrolled in NUST PNEC under the supervision of Dr. Arshad Aziz. The dataset contained 86 malicious and suspicious Urdu sentences spoken by 25 students which contain 2150 wav files.

5.2.1.3 Third Dataset

The third dataset was made with the help of currently enrolled master students of NUST PNEC under the supervision of Dr. Arshad Aziz. The data set contained 100 Urdu sentences spoken by 25 Post graduate students which contain 2500 wav files.

5.2.2 Creating An End To End Speech Recognition System

Once our first task that was collection of dataset has been completed, now our second task was to create an environment to develop an end to end speech recognition system for that we will install few libraries and site packages.

5.2.3 Creating Language Model

The language model helps to predict words that have occurred rarely or never using N- gram modeling [6] for that we took few news, speeches and dramas converted that into text and form a language model for Urdu using Kenlm.

5.2.4 Training Of Recurrent Neural Network

Once the dataset is received and language model is created our next task was to give hyper parameters to our recurrent neural networks in order to start training our model. These hyper parameters are:

5.2.4.1 Epoch

An epoch is when an entire data set is passed through forward and backward through the recurrent neural network for just once.

5.2.4.2 Batch size

It is the total number of training samples present in a single batch.

5.2.4.3 Iterations

It is the number of batches needed to complete one epoch.

5.2.4.4 Dropouts

By dropouts we drop activation functions temporarily with its incoming and outgoing connections.

5.2.4.5 Learning Rate

Determines the step size at each iterations while moving towards a minimum loss functions.

5.2.5 Testing Of Developred Speech Recognition System

For testing the accuracy of our speech recognition system we have to consider two parameters:

5.2.5.1 Word Error Rate (WER)

The factor that is used to check the accuracy of the model is word error rate (WER)[25] to compute the word error rate following formula is provided

$$WER = (S + D + I)/N \qquad [5.7]$$

Where S= Number of substitutions D=Number of Deletions I=Number of Insertions N= Number of words in a single sentence

5.2.5.2 Character Error Rate (CER)

The factor that is used to check the accuracy of the model is Character error rate (CER) to compute the character error rate following formula is provided

$$CER = (S + D + I)/N \qquad [5.8]$$

Where S= Number of substitutions D=Number of Deletions I=Number of Insertions N= Number of letter in a single word

5.3 Test Bed

In order to develop an end to end speech recognition system we will develop a test bed for that we have to go through the following process.

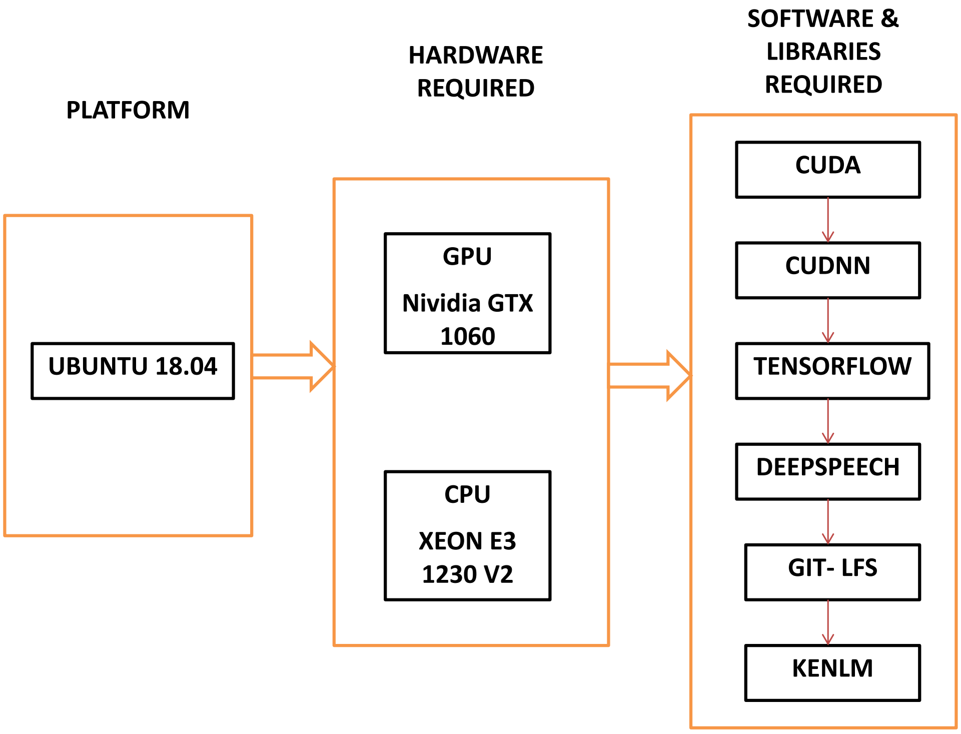


Figure 5.4 Process Of Developing Speech Recognition System

5.3.1 Selection Of Platform

We had two platform one was window and the other one was ubuntu, working on ubuntu was quite easier as compared to windows therefore we decided to install ubuntu since most of the packages and libraries supports ubuntu, although the model that we will acquire will be platform independent.

5.3.2 Prerequities Of Driver

We will train our model on two platforms namely GPU and CPU. For GPU we need to install the required driver according to the model of GPU. Our GPU is GTX 1060 and by finding manually we came to know that we have to install driver version 430.50.

5.3.3 Required Library For The Driver

As CUDA is a standard feature in all Nvidia Geforce, so installation of Cuda was necessary. CUDA stands for Compute Unified Device Architecture is an extension of C programming language created by Nvidia using Cuda will allow programmers to take advantage of huge parallel computing power of GPU. On the other hand Deep Neural Network library CUDNN is a GPU accelerated library of primitives of Deep Neural Networks so we need to install CUDNN as well.[26]

5.3.4 Prerequities For The Speech Recognition system

In order to install pre-build binaries and libraries we first need to install python 3.6 before installing any other packages.
We also need to install Git LFS (large file storage) which replaces large files such as audio samples, datasets, video and graphics with their text pointer inside Git while storing the file content on a remote server but as we have to work with binary files these we will save space by saving these files at different locations and keeping the pointer in Git.[27]

5.3.5 Creating Virtual Environment

Before Installing Deep Speech it is necessary to create a virtual environment we have created two virtual environment one for CPU and the other one for the GPU both environment are independent from each other that means if we install any package or library using python in one environment it will not affect the other environment.

5.3.6 Activating The Virtual Environment

After creating the virtual environment it is necessary to activate the environment in order to install different packages.

5.3.7 Installing Deepspeech Python Binding

Once the environment is set up use pip to install deep speech if you are using GPU platform then you need to install deep speech-gpu

5.3.8 Required DS-CTC Decoder

We need to install the ds-ctcdecoder python package.ds-ctcdecoder is required for decoding the output of deep speech acoustic model into text.[28][29]

5.3.9 Installation Of Tensorflow

Tensorflow is an open source software library released by Google to make it easier for the developer to design, build, and train deep learning modules. Tensorflow is a python library that allows user to express aribitary computation as a graph of data flows.[30]

5.3.10 Making Comma Separated Value Files(CSV)

In Deep speech the dataset of the train, dev and test files should be in the form of CSV files which allow data to be in the form of tabular format the first column of the CSV file is of wave file path second column is for the wave file size and the last column is for the wave transcript.
Following are the explanation of the three files
Train file: This file contains the data the model is trained with.
Dev file: This file contains the data for validation
Test file:Test files contain the data that is tested at the end to check how well the model is trained.

5.3.11 Creating LM.binary, LM.arpa and Trie files Using KENLM

For developing and training our models we need to have lm.binary,lm.arpa alphabet.txt and trie files for that we need to install Kenlm.
Kenlm is an implementation of kneaser-Ney smoothing [6] in which it smoothes the probability on n-gram model, it does this smoothing by considering the frequency of the n-gram in relation to possible words preceding it.
After the installation of kenlm we need to create lm.binary file that helps in the fast loading of large model files ,lm.arpa file which is used to represent all possible word sequence from a corpus of text data next we need to create trie file which is designed to save memory.
The data should be separated by the ratio of 70:20:10 for train, dev and test comma separated files.

5.3.12 Creating Run File

The run file is created for training a model, that run file contains following hyper parameters: Path of the three CSV files that is train, dev and test then the path of three model files that is lm.binary, trie and alphabet.txt files .After that the parameters provided are number of epochs, batch size, dropouts and learning rate.

5.3.13 Checkpoint Directory

Checkpoints are the model internal state (its current learning, weights etc) so whenever the training is stop due to any reason, it could start from the same instance using these checkpoints. Its just a way to save current state of your training.

5.3.14 Obtaining Our Model

Once the training is done completely and successfully you will get the output_graph.pb in the directory where you want to export it. This file including the other three files that is lm.binary, trie and alphabet.txt (which contain the alphabet of the language that you want to train) will make up your model. Now we have obtained our model once we have these four files.

Chapter 6

Results

Our task was to develop an automatic speech recognition system which would detect malicious and suspicious urdu words, phrases and sentences.We have successfully developed our ASR system that will perform the above mentioned task.Below are the results achieved and the comparison of our results with previous work done.

6.1 Dataset

We have made three dataset first dataset was obtained from Urdu Asr website and the remaining two were made by us.Following table shows the total number of wavs and their division in train, dev and test files.

Dataset	Total No.of wavs	No. of training wavs	No. of dev wavs	No. of test wavs
First Dataset	2500	1825	450	225
Second Dataset	2150	1505	430	215
Second+Third Dataset	4650	3255	930	465

Table 6.1 Showing total number of wavs division for train,dev and test files

6.2 Implementation Platform

During training we have used two platforms for training our dataset.The first platform was our CPU with 12 GB RAM and GPU ATI ATOMBIOSAnd the second platform was our GPU GeForce

GTX 1060 6GB/PCIe.

From the below table it is easily observed that training on GPU is far more efficient than CPU.

Dataset	CPU training time for 1 epoch (min)	GPU training time for 1 epoch (min)
First Dataset	15.01	1.08
Second Dataset	37.14	1.46
Second+Third Dataset	50.01	2.50

Table 6.2 Comparison between CPU and GPU for training one epoch

6.3 Training Hyper Parameters

From table 6.3 you can observed that we have provided dropouts in second and second plus third dataset this is because we were facing overfitting during training that is our model started learning minute detail including noise from training dataset in a way that it negatively impact the performance of our model.

Dataset	Epoch	Learning rate	Dropout
First Dataset	200	0.0001	0
Second Dataset	200	0.0001	0.5
Second+Third Dataset	200	0.0001	0.5

Table 6.3 Showing hyper parameters for the three datasets

6.4 Training Results

We obtained the minimum WER of 40%from our first dataset, for our second data set which was the model that contained 86 malicious sentences have the WER of 43.30% and finally the WER of 100 randomly taken sentences along with 86 malicious sentences was 48.39%.

Our objective was to develop an automatic Speech recognition system that would detect malicious

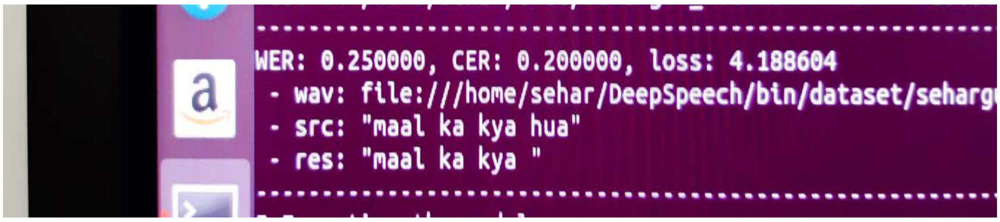
Dataset	No. of Epoch	WER	CER	Accuracy
First Dataset	200	40%	10.3%	60%
Second Dataset	200	43.30%	10.73%	56.7%
Second+Third Dataset	200	48.39%	12.59%	51.61%

Table 6.4 Training Evaluations

and suspicious words or phrases uttered in Urdu language by a speaker on telephone. We have achieved our goal by obtaining 43.30% WER for developing our automatic speech recognition system that will detect malicious and suspicious Urdu sentences.

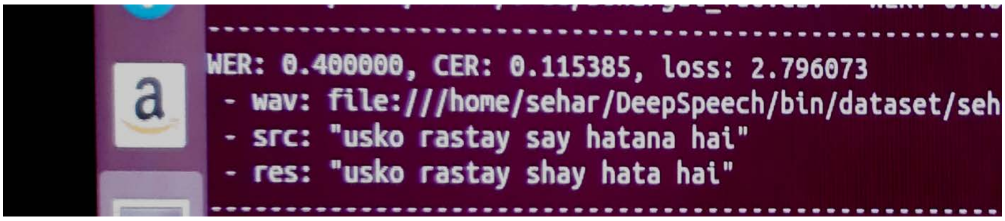
6.5 Best Results Achieved By Training Our Model

Following are some of the best results that we have achieved after training our model for detecting malicious and suspicious Urdu sentences.



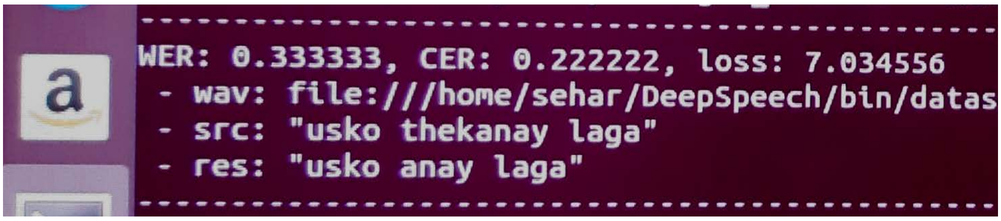
```
---------------------------------------------------------
WER: 0.250000, CER: 0.200000, loss: 4.188604
 - wav: file:///home/sehar/DeepSpeech/bin/dataset/seharg
 - src: "maal ka kya hua"
 - res: "maal ka kya "
---------------------------------------------------------
```

Figure 6.1 Showing WER=25% and CER=20%



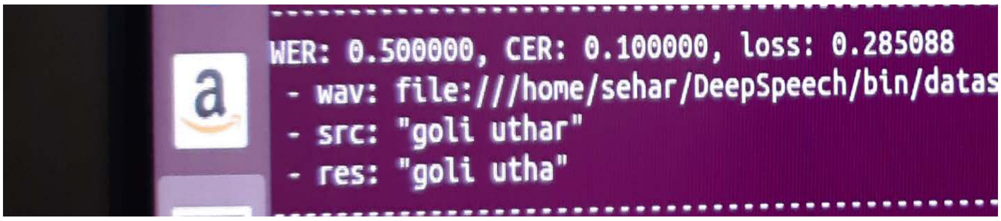
```
---------------------------------------------------------
WER: 0.400000, CER: 0.115385, loss: 2.796073
 - wav: file:///home/sehar/DeepSpeech/bin/dataset/seh
 - src: "usko rastay say hatana hai"
 - res: "usko rastay shay hata hai"
---------------------------------------------------------
```

Figure 6.2 Showing WER=40% and CER=11.53%



```
WER: 0.333333, CER: 0.222222, loss: 7.034556
 - wav: file:///home/sehar/DeepSpeech/bin/datas
 - src: "usko thekanay laga"
 - res: "usko anay laga"
```

Figure 6.3 Showing WER=33.33% and CER=22.22%



```
WER: 0.500000, CER: 0.100000, loss: 0.285088
 - wav: file:///home/sehar/DeepSpeech/bin/datas
 - src: "goli uthar"
 - res: "goli utha"
```

Figure 6.4 Showing WER=50% and CER=10%

6.6 Comparison With Previous Work Done

We have compared our results with previously done work in the development of Speech recognition system in Urdu Language.Muhammad Qasim et al developed a speech recognition system that was accent dependent and accent independent using CMU Sphinx Tool.This system can recognize only 139 names of district[31].Shaik Riyaz et al developed a speech recognition system in which MFCC was used to extract features, Vector quantization was applied to reduce feature vector size and Hidden Markov Model was applied to classify the feature vectors using probability estimation[32].This system will detect only words in Urdu.Hazrat Ali et al proposed a speech recognition system for Urdu Language which was done using linear discriminant analysis in which 52 MFCC was calculated for isolated Urdu words, the number of correct or incorrect matches can be found using confusion matrix.The system was developed to recognize Urdu words.[12] The table clearly indicate that our speech recognition system is performing better for the sentences in Urdu language.

From the below table it is clear that with few amount of data our speech recognition system work well is comparison with other developed model of speech recognition system.

Authors	Dataset	Results
Muhammad Qasim et al[31]	139 District names of Pakistan spoken by 300 people	Accent Dependent 60.06% Accent Independent 75.25%
Shaik Riyaz et al[32]	The dataset contain 250 isolated words uttered by 20 speakers	96.4%
Hazrat Ali et al[12]	the dataset contain 250 Isolated words uttered by 10 speakers	66.66%
First dataset	250 isolated words uttered by 10 speakers	60%
Second dataset	86 sentences spoken by 25 people	56.7%
Second + Third dataset	86 malicious sentences + 100 randomly spoken sentences	51.61%

Table 6.5 Comparison With Previous Work Done

Chapter 7

Conclusion And Future Work

In this chapter we will provide concluding remarks related to our research work, future directions which may be taken into considerations in continuation of this research.

7.1 Conclusion

We have successfully developed an ASR system that will detect malicious and suspicious sentences in Urdu Language with word error rate of 43.30% unlike other ASR which were developed for the detection of randomly spoken isolated words but not sentences.
We have also developed a model that is train to detect 250 randomly spoked isolated words with the word error rate of 40%.Next we have developed another model that contain 100 positive sentences with 86 malicious sentences with the word error rate of 48.39%.
We have studied detail Neural Network and Connectionist Temporal Classification which were the main algorithm used in our ASR system these algorithms are latest and provide excellent results if meaningful dataset is provided.

7.2 Future Work

The future work includes collection of more dataset for our ASR system the more the dataset the more accuracy will be achieved using an end to end speech recognition system. So our future work is to develop more Urdu sentences that are malicious and suspicious uttered by different people with different accent in order to decrease our WER for the develop ASR.

List of References

[1] S. A. Alim and N. K. A. Rashid, “Some commonly used speech feature extraction algorithms,” in *From Natural to Artificial Intelligence* (R. Lopez-Ruiz, ed.), ch. 1, Rijeka: IntechOpen, 2018. 2.1.1, 2.1.1.1, 2.1.1.2, 2.1.1.3, 2.1.1.4

[2] A. Waris and R. Aggarwal, “Acoustic modeling in automatic speech recognition a survey,” in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, October 2018. 2.1.2

[3] E. Arisoy, M. Kurimo, M. Saraclar, T. Hirsimaki, J. Pylkkonen, T. Alumae, and H. Sak., “Statistical language modeling for automatic speech recognition of agglutinative languages, speech recognition,” Master’s thesis, Vienna, Austria, 2008. 2.1.3

[4] “Deep learning in speech recognition https://androidkt.com/recognition-encoding/,” 2.2, 2.3.1, 2.3.2

[5] A. Hannun, C. Cas, J. Caspera, B. Catanzar, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, “Deep speech: Scaling up end-to-end speech recognition,” *Baidu Research Silicon Valley AI Lab*. 2.3.3, 2.3.3.1, 2.3.4, 5.1.1, 5.1.2, 5.1.2

[6] . S.F.Chen and J.Goodman, “An empirical study of smoothing techniques for language modeling, computer speech and language modeling,” *Computer Speech and Language*, vol. 13, no. 4, pp. 359–394, 1999. 2.3.3.1, 2.3.4, 5.1.3, 5.2.3, 5.3.11

[7] S. V. Amina Simon, Mahima Singh Deo and D. R. Babu, “An overview of machine learning and its applications,” *Dept of Computer Science and Engineering. International Journal of Electrical Science Engineering (IJESE)*, vol. 1, pp. 22–24, 2015. 3, 3.1, 3.2

[8] J. Castanon, “10 machine learning methods that every data scientist should know,” *https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9*. 3.3

[9] M. M. R.Sandanalakshmi, P.Abinaya viji and A.Sharina, “Speaker independent continuous speech to text converter for mobile application,” 4

[10] N. A. Nafis and M. S. Hossain, “Speech to text conversion in real-time,” *Innovative Space of Scientific Research Journals*, vol. 17, no. 2, pp. 271–277, August 2015. 4

[11] N. Sharma and S. Sardana, “A real time speech to text conversion system using bidirectional kalman filter in matlab,” september 2016. 4

[12] N. A. Hazrat Ali and X. Zhou, “Automatic speech recognition of urdu wordsusing linear discriminant analysis,” *Journal of Intelligent and Fuzzy Systems*, vol. 28, no. 5, pp. 2369–2375, June 2015. 4, 6.6

[13] S. A. Ali, S. Khan, H. Perveen, R. Muzzamil, M. Malik, and F. Khalid, “Urdu language translator using deep neural network,” *Indian Journal of Science and Technology*, vol. 40, no. 10, October 2017. 4

[14] Thiang and S. Wijoyo, “Speech recognition using linear predictive coding and artificial neural network for controlling movement of mobile robot,” *Proceedings of International Conference on Information and Electronics Engineering (IPCSIT), Singapore, IACSIT Press*, vol. 6, pp. 179–183, 2011. 4

[15] Ms.Vimala.C and Dr.V.Radha, “Speaker independent isolated speech recognition system for tamil language using hmm,” *Proceedings International Conference on Communication Technology and System Design*, p. 1097 1102, 2012. 4

[16] K. B. Cini Kuriana, “Development evaluation of different acoustic models for malayalam continuous speech recognition,” *Proceedings of International Conference on Communication Technology and System Design 2011 Published by Elsevier Ltd*, pp. 1081–1088, 2011. 4

[17] S. Swamy and K. Ramakrishnan, “An efficient speech recognition system,” *Computer Science Engineering: An InternationalJournal(CSEIJ)*, vol. 3, no. 4, pp. 21–27, 2013. 4

[18] A. Choudhary, M. R. Chauhan, and M. G. Gupta, “Automatic speech recognition system for isolated connected words of hindi language by using hidden markov model toolkit (htk),” *International Conference on Emerging Trends in Engineering and Technology*, pp. 2444–252, 2012. 4

[19] P. Saini, P. Kaur, and M. Dua, “Hindi automatic speech recognition using htk,,” *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, pp. 2223–2229, 2013. 4

[20] M. A. Ali, M. Hossain, and M. N. Bhuiyan, “Automatic speech recognition technique for bangla words,” *International Journal of Advanced Science and Technology*, vol. 50, pp. 51–60, 2013. 4

[21] M. Moneykumar, E. Sherly, and W. S. Varghese, “Malayalam word identification for speech recognition system,” *An International Journal of Engineering Sciences*, vol. 15, pp. 22–26, 2014. 4

[22] J. S. Pokhariya and D. S. Mathur, “Sanskrit speech recognition using hidden markov model toolkit,” *International Journal of Engineering Research Technology (IJERT)*, vol. 3, pp. 93–98, 2014. 4

[23] G. Nijhawan and D. M. Soni, “Real time speaker recognition system for hindi words,” *International Journal of Information Engineering and Electronic Business*, vol. 6, pp. 35–40, 2014. 4

[24] *http://csalt.itu.edu.pk/PRUSCorpus/index.html*. 5.2.1

[25] S. Seljan and I. Duner, “Combined automatic speech recognition and machine translation in business correspondence domain for english-croatian,” *World Academy of Science, Engineering and Technology International Journal of Industrial and Systems Engineering*, vol. 8, no. 11, 2014. 5.2.5.1

[26] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, “cudnn: Efficient primitives for deep learning,” *Baidu ResearchSunnyvale*, 18 December 2015. 5.3.3

[27] *https://towardsdatascience.com/why-git-and-git-lfs-is-not-enough-to-solve-the-machine-learning-reproducibility-crisis-f733b49e96e8*. 5.3.4

[28] T. Zenkel1, R. Sanabria1, F. Metze1, J. Niehues2, M. Sperber2, S. S. 2, and A. Waibel1, “Comparison of decoding strategies for ctc acoustic models,” *1Carnegie Mellon University; Pittsburgh, PA; U.S.A. 2Karlsruhe Institute of Technology; Karlsruhe, Germany*, 15 August 2015. 5.3.8

[29] *https://github.com/mozilla/DeepSpeech*. 5.3.8

[30] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. Software available from tensorflow.org. 5.3.9

[31] M. Qasim, S. Nawaz, S. Hussain, and T. Habib, “Urdu speech recognition system for district namesof pakistan: Development, challenges and solutions,” *The Oriental Chapter of International Committee*, 2016. 6.6

[32] S. Riyaz, B. L. Bhavani, and S. P. Kumar, “Automatic speaker recognition system in urdu using mfcc hmm,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, 2019. 6.6