BMAC DLA- A Novel Approach to SpeechH Recognition

Rahmathulla K.
Dept. of Computer Science and Engineering
MEA Engineering College
Perinthalmanna, Kerala, India

Jemsheer Ahmed.
Dept. of Computer Science and Engineering
MEA Engineering College
Perinthalmanna, Kerala, India

***Abstract*— Accessing a device through Speech recognition is a new research area. It provides users an easier way to search for information using voice. Recent advances in mobile technology have enabled users of voice recognition products to achieve desirable results. One of the most important reasons that led to the production a large number of voice input systems is the ability of the voice input systems to remove a lot of literacy barriers in communicating with the devices. Current work proposes multilayer Perceptron neural network based speech recognition method, the framework for controlling systems through Voice recognition approach. It uses several commands for recognition of the voices and trains the engine to the voice of different users. It can provide a foundation for improving the communication, learning and teaching environments. However, not all potential users have the capabilities that allow them to use the existing methodologies. This model provides the opportunity to operate the mobile or desktop devices without using the keypad. Experimental result provides better result when compare with the conventional methods.**

Keywords*— Speech recognition, multilayer perceptron neural network, MFCC.

I. INTRODUCTION

People with disabilities meet barriers of all types. However, technology is helping to lower many of these barriers. By using computing technology for tasks such as reading and writing documents, communicating with others, and searching for information on the Internet, students and employees with disabilities are capable of handling a wider range of activities independently. Still, people with disabilities face a variety of barriers to computer use. These barriers can be grouped into three functional categories: barriers to providing computer input, interpreting output, and reading supporting documentation. Hardware and software tools (known as adaptive or assistive technologies) have been developed to provide functional alternatives to these standard operations. Specific Mobile voice search is a new research area. It provides users an easier way to search for information using voice from mobile devices. Recent advances in mobile technology have enabled users of voice recognition products to achieve desirable results. One of the most important reasons that led to the production a large number of voice input systems is the ability of the voice input systems to remove a lot of literacy barriers in communicating with the mobile devices. Besides, the inventions of the technological innovations have suggested the emerging knowledge society, in which all our engagements in life will be focused on having knowledge of something. Such knowledge can be acquired in many ways and in different forms.

A. Speech Recognition

Speech Recognition (SR) is the translation of spoken words into text. It is also known as "automatic speech recognition", "ASR", "computer speech recognition", "speech to text", or just "STT".

Some SR systems use "speaker independent speech recognition" while others use "training" where an individual speaker reads sections of text into the SR system. These systems analyse the person's specific voice and use it to fine tune the recognition of that person's speech, resulting in more accurate transcription. Systems that do not use training are called "speaker independent" systems. Systems that use training are called "speaker dependent" systems.

Speech recognition applications include voice user interfaces such as voice dialling (e.g. "Call home"), call routing (e.g. "I would like to make a collect call"), domestic appliance control, search (e.g. find a podcast where particular words were spoken), simple data entry

(e.g., entering a credit card number), preparation of structured documents (e.g. a radiology report), speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed Direct Voice Input).

The term voice recognition refers to finding the identity of "who" is speaking, rather than what they are saying. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on specific person's voices or it can be used to authenticate or verify the identity of a speaker as part of a security process.

B. Voice Recognition

Voice recognition is an alternative to typing on a keyboard. Put simply, you talk to the computer and your words appear on the screen. The software has been developed to provide a fast method of writing on a computer and can help people with a variety of disabilities. It is useful for people with physical disabilities who often find typing difficult, painful or impossible. Voice-recognition software can also help those with spelling difficulties, including users with dyslexia, because recognised words are almost always correctly spelled.

Voice-recognition software programmes work by analysing sounds and converting them to text. They also use knowledge of how English is usually spoken to decide what the speaker most probably said. Once correctly set up, the systems should recognise around 95% of what is said if you speak clearly. Several programmes are available that provide voice recognition. These systems have mostly been designed for Windows operating systems; however programmes are also available for Mac OS X. In addition to third-party software, there are also voice-recognition programmes built in to the operating systems of Windows Vista and Windows 7. Most specialist voice applications include the software, a microphone headset, a manual and a quick reference card. You connect the microphone to the computer, either into the soundcard (sockets on the back of a computer) or via a USB or similar connection. Then you can begin talking using the following steps.

1. Enrolment

Everybody's voice sounds slightly different, so the first step in using a voice-recognition system involves reading an article displayed on the screen. This process, called enrolment, takes less than 10 minutes and results in a set of files being created which tell the software how you speak. Many of the newer voice-recognition programmes say this is not required, however it is still worth doing to get the best results. The enrolment only has to be done once, after which the software can be started as needed.

2. Dictating and Correcting

When talking, people often hesitates, mumble or slur their words. One of the key skills in using voice-recognition software is learning how to talk clearly so that the computer can recognise what you are saying. This means planning what to say and then speaking in complete phrases or sentences. The voice-recognition software will misunderstand some of the words spoken, so it is necessary to proofread and then correct any mistakes. Corrections can be made by using the mouse and keyboard or by using your voice.

When you make corrections, the voice-recognition software will adapt and learn, so that (hopefully) the same mistake will not occur again. Accuracy should improve with careful dictation and correction.

3. Editing and Formatting Text

Text can be edited very easily. You can highlight the text to be changed by using commands such as "Select line," or "Select paragraph," and then saying the changes you want to make to the computer. These will then replace the selected text.

Applying formatting is just as straightforward. For example, if a document has the phrase 'introductory thoughts', you can underline this phrase by saying "Select introductory thoughts," and then saying "Underline that."

4. Controlling the Computer

Many voice-recognition programmes offer the ability to start and control programmes through spoken commands.

a)Commands vary, depending on which voice-recognition program you are using. For example, with the program Dragon NaturallySpeaking you could say "Start Microsoft Word," then "Open letter to John."

b)Menus can be selected simply by pausing and then saying the menu item. For example, saying "File" would open the file menu.

c) On the Internet, you can dictate web addresses and browse websites simply by saying the text in the link.

d) *For tasks that require a mouse*, there are spoken commands to enable the mouse to be moved, dragged and clicked.

II. BACKGROUND STUDY

A. Automatic Speech Recognition for Generalized Time Based Media Retrieval and Indexing[1]

This report presents research undertaken which extends upon existing automatic speech recognition approaches, using appropriate string matching techniques. This approach demonstrated improved retrieval performance of speech data when compared with content based retrieval performance using automatic speech recognition techniques done. One possible means of overcoming this cost implement is to employ automatic speech recognition technology to support searching on audio media. This approach could potentially

eliminate the need for a human to listen to the entire audio tract to produce a transcript or an index. By Wowing the user of the media to search for index terms within an audio track (or having end users search for pertinent audio segments themselves), a more cost effective technique can be employed to retrieve and utilize media containing human voice.

The type of media the automatic speech recognition approach would be most effectively applied to is material containing voice data which conveys the primary semantic, or pragmatic content of the video. Typically, this could be news or documentaries where the audio track is the principle conduit of information. Videos or film with little or no speech contents such as music videos or silent films, would not be useable by the proposed solution. Audio recordings or radio publications would be constrained by the same restrictions.

1. Recognition phase

The recognition phase requires the designing and training of a set of models used to characterize a set of phonetic labels. Hidden markov models (HMM) a stochastic technique commonly used for speech recognition applications was used to model the forty-five phones. The HTK System was used to build the models. The training data used to train the forty-five continuous density HMMs consisted of eight hundred continuous speech sentences (4 males x 200), from the ANDOSL database, which had been manually segmented and labeled by trained phoneticians. The training set was limited by the amount of relevant labeled data available in the ANDOSL database. A speaker-dependent test was performed on all 800 sentences used for training. A speaker-independent test was conducted using speech data from a single male speaker not included in the training data.

2. Retrieval phase

Three minutes of the ten hours video was initially used to evaluate the produced recognition stream's accuracy. Six keyterms were selected from this audio segment "Australia, electronic, library, today, publishing and & development". Each of these terms occurred a minimum of three times in the chosen audio stream. These key terms (search term) were translated into an accepted correct phonetic sequence using the GTP program. Analysis of the keyterrns occurrences within the transcription data exposed very high errors level. For example, the key term "Am&&" occurred three times within the speech segment. All instances were spoken by one individual. None of the resulting transcription labels produced by the recognizer for these three occurrences matched the phonetic spelling of the key phrase.

3. Use of string matching technique to identify keyword

In the previous phase, the continuous speech recognizer produced a tie containing a string of labels representing the phonetic structure of the audio data. Errors within this string can be due to recognition errors where one phone is substituted for another, or phones are omitted or inserted. The string matching technique must manage these variances to find correct instances of search terms.

B. Subword Lexical Modelling for Speech Recognition[2]

1. Introduction

In this work, a novel framework of ANGIE is introduced for modeling sub word lexical phenomena in speech recognition. Our framework provides a flexible and powerful mechanism for capturing morphology, syllabification, phonology, and other subword effects in a hierarchical manner which maximizes sharing of subword structures. ANGIE models the subword structure within a context-free grammar and an accompanying probability model. We believe that our framework has several advantages: The sharing mechanism allows training data to be pooled amongst instances of the same word substructure even when they occur across different words in the lexicon.

In this work a probabilistic framework has been presented for sub lexical modeling which is named angie. The angie framework is a descendant of the framework described in Meng. The motivations behind the framework include creating a sublexical model which:

- captures various sublexical phenomena, including phonology, syllabification and morphology, in a unified framework;
- is probabilistic in nature;
- promotes sharing of common sublexical structures among different words in the vocabulary, words introduced into the vocabulary, and in principle, new out-of-vocabulary words;
- proceeds in a bottom-up manner, reflecting our bottom-up sublexical philosophy , our desire to share and our aim to model word-like structures for the background filler in word-spotting and new words;
- provides a single framework for multiple tasks, including recognition oriented tasks and letter-to-sound/sound-to-letter generation; and
- Shares a common context-free framework with many natural language understanding systems, permitting an integrated system for both phonological and linguistic modelling.

At the heart of angie are a context-free grammar, a parser, and a probability model associated with parses generated.

2. Context-free Grammar

The context-free grammar is hand written and is used to obtain a hierarchical representation of the sub lexical phenomena of interest. There has been some work in the literature which tries to automatically learn a grammar describing sub word structures. However, such an approach will migrate back in the direction of implicit sub word modeling, where the actual sub word relationships are hidden, not easily controllable, and not motivated by linguistic organizations. Therefore this has chosen to maintain explicit control over the structural organization.

The hierarchical representation has a very regular layered hierarchical structure. Here, we would also like to point out that, from a computational framework point of view, the choice of a hierarchical structure allows us access to

information available at each layer of the hierarchy. A set of transformations effected by linear rewrite rules, which is typically used in the linguistics literature, obscures this information. The root sentence node is currently realized as a sequence of word nodes. Presently, these two categories act as place holders, but could later be replaced with other alternatives such as topical units, syntactic units, or both.

Here introduced the angie framework for sublexical linguistic modelling. Angie is a unified computational framework capturing morphology, syllabification and phonology through a layered hierarchical structure. At the heart of angie are a context- free grammar, a bottom-up breadth-first left-to-right parser and a probability model. The probability model consists of two types of probabilities, advancement probabilities and bottom-up trigram probabilities. Our hope for the probability model is for it to capture both some of the context-dependencies commonly believed to govern phonological processes and also to account for variability. It believes that accounting for variability is crucial in a system that needs to handle errorfull inputs. Lexical units in angie are tracked via a listing of legal phonemic sequences or two listings, one of legal morphemic sequences and another of legal phonemic sequences for the morphemic units. This work has explored some of the implementational issues involved, including time and space complexity concerns.

C. Entropy based Pruning of Backoff Language Models[3]

1. Introduction

In this work the problem of N gram parameter selection is revised by deriving a criterion that satisfies the following desiderata.

a)Soundness: The criterion should optimize some well understood information theoretic measure of language model quality.

b)Efficiency: An N gram selection algorithm should be fast, i.e., take time proportional to the number of N grams under consideration.

c)Self containedness: As a practical consideration, the work needs to be able to prune Ngrams from existing language models. This means a pruning criterion should be based only on information contained in the model itself.

An N gram language model represents a probability distribution over words w, conditioned on (N-1) tuples of preceding words, or histories h. Only a finite set of N grams (w,h)have conditional probabilities explicitly represented in the model. The goal of N gram pruning is to remove explicit estimates p(w|h)from the model, thereby reducing the number of parameters, while minimizing the performance loss. Note that after pruning, the retained explicit N gram probabilities are unchanged, but backoff weights will have to be recomputed, thereby changing the values of implicit (backed off) probability estimates. Thus, the pruning approach chosen is conceptually independent of the estimator chosen to determine the explicit gram estimates. Since one of our goals is to prune N gram models without access to any statistics not contained in the model itself, a natural criterion is to minimize the 'distance' between the distribution embodied by the original model and that of the pruned model. A standard measure of divergence between distributions is relative entropy or Kullback-Leibler distance. Although not strictly a distance metric, it is a nonnegative, continuous function that is zero if and only if the two distributions are identical.

This suggests a simple thresholding algorithm for N gram pruning:

1. Select a threshold θ.

2. Compute the relative perplexity increase due to pruning each N gram individually.

3. Remove all N grams that raise the perplexity by less than θ, and recompute backoff weights.

An algorithm for N gram selection is developed for backoff N gram language models, based on minimizing the relative entropy between the full and the pruned model. Experiments show that the algorithm is highly effective, eliminating all but 26% of the parameters in a Hub4 four gram model without significantly affecting performance. The pruning criterion of Seymour and Rosenfeld is seen to be an approximate version of the relative entropy criterion; empirically, the two methods perform about the same.

III. PROBLEM DEFINITION

The problem of automatically recognizing speech for accessing a computer is a difficult problem, and the reason for this is the complexity of the human language.

Humans use more than their ears when listening; they use the knowledge they have about the speaker and the subject. Words are not arbitrarily sequenced together, there is a grammatical structure and redundancy that humans use to predict words not yet spoken.

In Speech recognition we only have the speech signal. With speech signal, a model for the grammatical structure has also been constructed and uses some kind of statistical model to improve prediction, but there are still the problem of how to model world knowledge, the knowledge of the speaker and encyclopedic knowledge.[4]

The following are the problems or issues arising during speech recognition are listed as follows:

A. Noises in Speech: Speech recognition often consists of speech signal with noises, Noises is defined as unwanted information in the speech signal such as environment of sounds, a clock ticking, a computer humming, a radio playing somewhere down the corridor, another human speaker in the background etc.Noises namely echo effect occurs with speech signal bounced on some surrounding object, and that arrives in the microphone a few milliseconds later. If the place in which the speech signal has been produced is strongly echoing, then this may give raise to a phenomenon called reverberation, which may last even as long as seconds.

B. Lexical problems in speech: Lexical problems in speech have to be identified is that the grammaticality of spoken language is quite different to written language at many different levels.

Some differences are pointed out:

• In spoken language, there is often a radical reduction of morphemes and words in pronunciation.

• The frequencies of words, collocations and grammatical constructions are highly different between spoken and written language.
• The grammar and semantics of spoken language is also significantly different from that of written language; 30-40% of all utterances consist of short utterances of 1-2-3 words with no predicative verb.

IV. PROPOSED SYSTEM

To improve the performance of the system and to overcome the problem identified in the existing system, the present work proposes Multilayer Perceptron Neural Network based speech recognition system which activates programs on desktop or panel by voice.
In order to start talking right away, the proposed system follows two steps.

1. The first thing to do is adjust the microphone
2. The second thing to do is training the engine to your voice by using neural network

After these changes, the program needs to be started by listening again by clicking the right mouse button and choosing "Start listen." The more training the engine, the better it will recognize the voice, although an improvement has been seen from the first training. After the program is started, it may be in several "states". In every state, it recognizes a list of specific commands. The list of the commands that the program can identify is shown below.

A. States or command for Processing

The program will show you what the available commands from here are. States (commands) available in the program:
The system architecture in fig 4.1 shows what happens to one single command. MFCC features from the normal and pathological speech samples are input to the NN for training. This is used for classifying the voice of different speaker.
The input samples are propagated in a forward direction on a layer-by-layer basis. The network computes its output pattern, and if there is an error - or in other words a difference between actual and desired output patterns - the weights are adjusted to reduce this error. In a back-propagation neural network, the learning algorithm has two phases. First, a training input pattern is presented to the network input layer. The network propagates the input pattern from layer to layer until the output pattern is generated by the output layer. If this pattern is different from the desired output, an error is calculated and then propagated backwards through the network from the output layer to the input layer

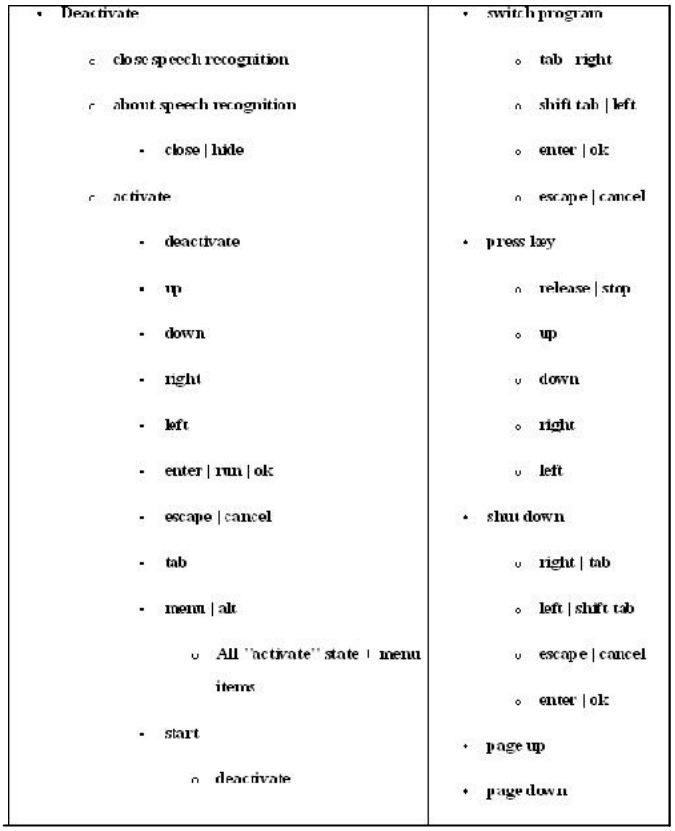

Fig. 1. command list

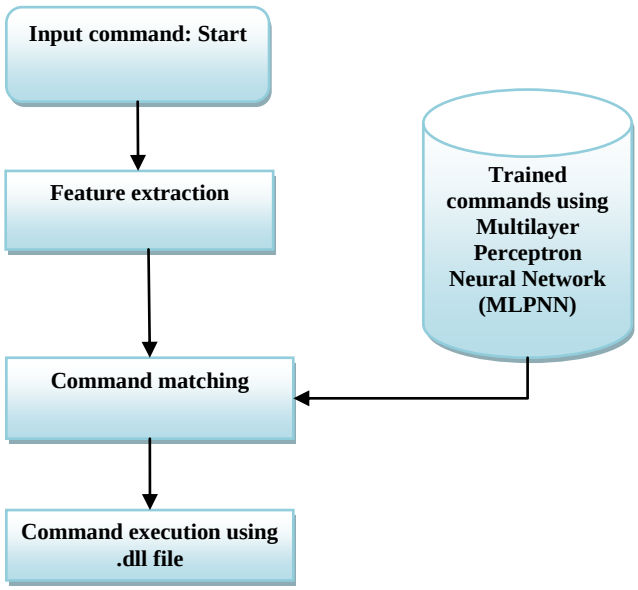

Fig. 2. System architecture

V IMPLEMENTATION DETAILS

A. Initialization step

The initial state is in the "deactivate" state, which means that the program is in a sleepy state. After the command "activate" you will wake up the program ("activate" state) and start recognizes other commands.

When the engine is activated the initialization step takes place. There are mainly 3 objects involved:

1. Recognition process
2. Adding events
3. Loading of grammar file

In the recognization phase **i**nitially the reference file is added and speech library can be used. To start the recognition process, a shared recognizer is created which will automatically start Windows Speech Recognition, which serves as the common UI for shared recognizers. Then events namely Audio Level and Recognition are added. Finally the static grammar file can be loaded from the Specified file.

1. Training the device using multilayer Perceptron neural network

The multilayer Perceptron Neural Network (MLPNN) is used to train the neural network for spoken words for each speaker. Ten speakers are trained using (you cannot say 10 speakers ,you have to mention generally not specifically) the multilayer perception with 240 input nodes, 2 hidden layers and 1 output node each for one word, with the nonlinear activation function sigmoid. The learning rate is taken as 0.1, momentum rate is taken as 0.5.Weights is initialized to random values between +0.1 and -0.1 and accepted error is chosen as 0.009

2. Multilayer Perceptron Neural Network (MLPNN)

MLPNN is composed of three layers consisting of an input layer, one or more hidden layers and an output layer. The input layer distributes the inputs to subsequent layers. Input nodes have linear activation functions and no thresholds. Each hidden unit node and each output node have thresholds associated with them in addition to the weights. The hidden unit nodes have nonlinear activation functions and the outputs have linear activation functions. The number of neurons in the hidden layer is dependent on the size of the input vector. The output layer has one neuron. MFCC features from the normal and pathological speech samples are input to the NN for training. This is used for classifying the voice of different speaker.[5]

The input samples are propagated in a forward direction on a layer-by-layer basis. The network computes its output pattern, and if there is an error - or in other words a difference between actual and desired output patterns - the weights are adjusted to reduce this error. In a back-propagation neural network, the learning algorithm has two phases.[6] First, a training input pattern is presented to the network input layer. The network propagates the input pattern from layer to layer until the output pattern is generated by the output layer. If this pattern is different from the desired output, an error is calculated and then propagated backwards through the network from the output layer to the input layer. The weights are modified as the error is propagated. This process is explained below.

Step 1: Initialization - Set all the weights and threshold levels of the network.

Step 2: Activation - Activate the back-propagation neural network by applying inputs and desired outputs. Calculate the actual outputs of the neurons in the hidden layer and the output layer.

Step 3: Weight training - Update the weights in the back-propagation network propagating backward the errors associated with output neurons. Calculate the error gradient for the neurons in the output layer and the hidden layer.

Step 4: Iteration - Increase iteration one by one, go back to Step 2 and repeat the process until the selected error criterion is satisfied.

B. Desktop control

In this section System desktop has been controlled by using following commands:

- start
    - deactivate
    - up
    - down
    - right
    - left
    - enter | run | ok
    - escape
    - tab
    - commands list
    - programs
    - documents
    - settings
    - search
    - help
    - run
- Shut down
    - right | tab
    - left | shift tab
    - escape | cancel
    - enter | ok

For example, use "start" to activate the start menu. Then say "programs" to enter the programs menu. From this point,

the navigation has been done by saying "down"," up", "right"... "OK" according the commands list. Then say "commands list" from any point to see a form with the list of the commands that can say.
One of the important states in the program is the "menu" state, meaning that if a program is running (and focused) say "menu" to hook all menu items and start using them. For example, if Notepad is running, then open a new file by saying "menu"->"File"->"new file". Every time menus are hooked, so that how many menus are hooked can be seen the program hooked then can start using them as commands.
The state of shutdown is used to shutdown the system by saying the commands such as right, left for navigation, escape to cancel the process and say enter to shutdown the system.

C. File control

In this section Files has been controlled by using following commands:

- page up
- page down
- Scroll

Page up command controls the file and displays the upper portion of the current file. The command Page down controls the file and displays the lower/bottom portion of the current file. By saying the command scroll, the file can be scrolled up and down respectively.

D. Numeric control

In this section the state of Numeric is controlled by the following commands

- enter numeric state
    - exit numeric state
    - back | back space
    - plus
    - minus
    - mul | multiply
    - div | divide
    - equal
    - Numbers from 0 – 9

For example, say the commands "favorites programs","calculator","enter numeric state", "one","plus","two","equal" and see the result. The result of “three” is displayed as a result in calculator. Similarly, mul command is for doing multiplication operation, div command is for division operation.

E. Alphabetic control

In this section the state of Alphabetic is controlled by the following commands

- enter alphabetic state
    - exit alphabetic state
    - back space
    - enter
    - at ("@")
    - underline ("_")
    - dash ("-")
    - dot (".")
    - back slash ("/")
    - Letters from A to Z

For example, say the commands "favorites programs","internet explorer","enter alphabetic state", "menu","down","down","O K", "enter alphabetic state","c","o","d","e",...,"dot","c","o","m" and see the results.

VI RESULTS AND OBSERVATIONS

A. Experimental Results`

The system is implemented using .NET programming language. For this purpose VISUAL STUDIO 2010 is installed.

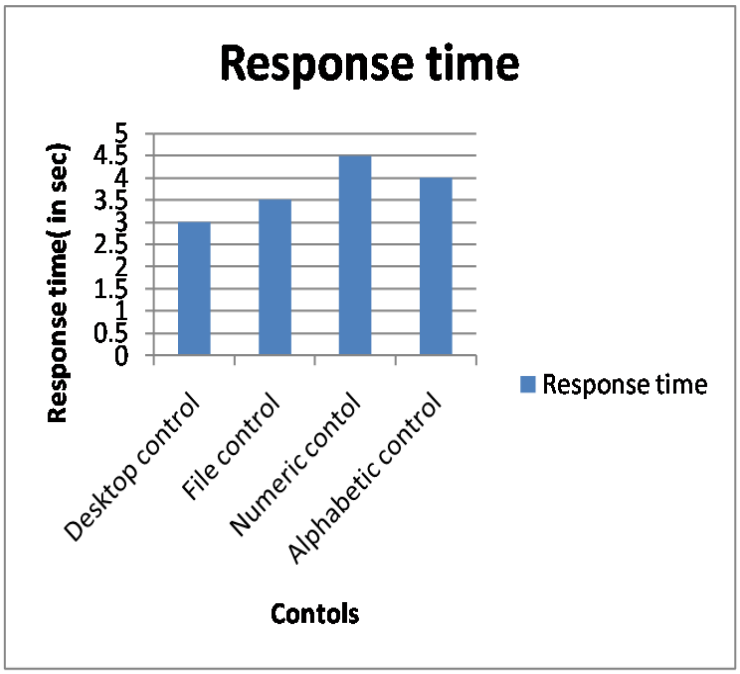

Fig. 3. response time graph

As user makes a call to the system the timer starts and when user says a command regarding the states the system match the command with its vocabulary if it finds match it follows the instruction. If the timer exceeds from certain limit the system discard the call or if system does not find the match it sends the message to preconfigured number "sorry could not understand your message".

The graphs of response time of system and Recognition Accuracy are given:

1. Response Time graph

The above graph in figure 6.1 shows the Response time for the controls such as Desktop control, File control, Numeric control and Alphabetic control respectively. In this graph the controls are depicted in X axis and Response time in seconds is represented in Y- axis respectively. When the user says the command, the system response time obtained for Desktop control is 3.0 sec, File control takes 3.5 seconds, Numeric

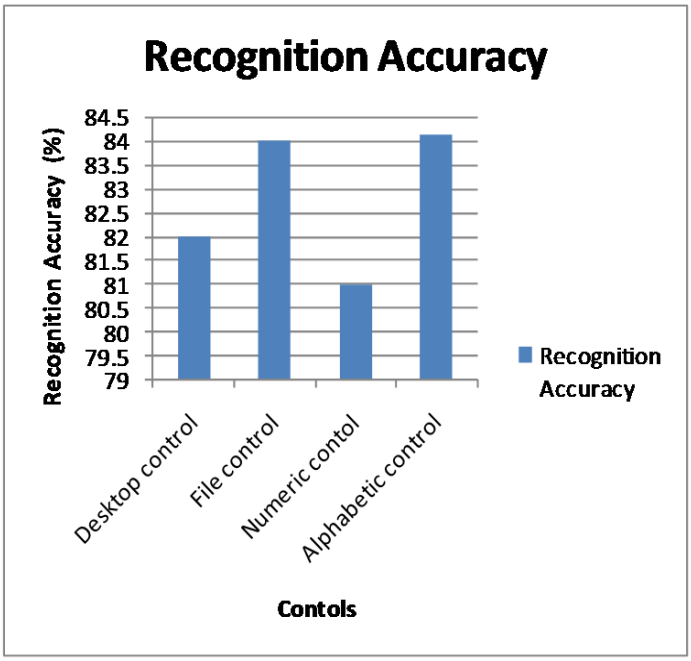

Fig. 4. Recognition Accuracy graph

control takes 4.5 seconds and Alphabetic control takes 4.0 seconds respectively.

2. Recognition Accuracy graph

The above graph in figure 6.2 shows the Recognition Accuracy for the controls such as Desktop control, File control, Numeric control and Alphabetic control respectively. In this graph the controls are depicted in X axis and Recognition Accuracy in % (percentage) is represented in Y-axis respectively. When the user says the command, the Recognition Accuracy obtained for Desktop control is 82 %, File control achieves 84%, Numeric control takes 81% and Alphabetic control achieves 84.12% respectively.

VII CONCLUSION AND FUTURE WORKS

The present work proposes multilayer Perceptron Neural Network[7] (MLPNN) based speech recognition method which was developed to activate programs on ones desktop or panel by voice. The proposed system trains the engine to ones voice by training the MLPNN. Multilayer Perceptron Neural Network architecture[8] has been shown to be suitable for the recognition of ambiguous words. Recognition of the words is carried out in speaker dependent mode. Prior to this process microphone has been adjusted. The proposed system recognizes voices efficiently and improvement has been seen in the first training process. The program has been started with the use of several states such as Desktop control, File control, Numeric control and Alphabetic control etc., and each state recognizes list of commands. The main advantage of the proposed system is there exist a performance metric such as Recognition accuracy and Respond time for each controls and the user can establish a threshold so that unsure recognitions have been filtered accordingly. Comparative result provides better recognition result when compare with the existing techniques.

Since the Smartphone's and mobile devices are in the middle of major innovations in technology to provide hands-free access to features and navigation, often called voice commands,[9] voice-enabled, voice actions or speech recognition. This technology has major implications for use by people who have disabilities as assistive technology. As long as a user has a strong, clear voice, these devices become easier to use and give increased access to use of the Internet, use of mobile devices and communication accessibility. In the future, the current work can be further explored and deployed for use on other handheld and mobile devices.

REFERENCES

[1] [1] John Robertson, Wai Yat Wong, Charles Chung, Dongki Kim," Automatic Speech Recognition for Generalised Time Based Media Retrieval and Indexing" In proceeding of: Proceedings of the 6th ACM International Conference on Multimedia '98

[2] [2] R. Lau. 1998. Subword Lexical Modelling for Speech Recognition. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

[3] [3] Stolcke. 1998. Entropy-based pruning of backoff language models. In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pages 270-274, Lansdowne, VA.

[4] [4] V. Gotlin, G. Riccardi C. Allauzen D. Hakkani A. Ljolje S.Parthasarathy M. Rahim, and M. Saraclar., "The at&t Watson speech recognizer," in ICASSP, 2005

[5] [5] B. T. F. Pengo Unsupervised query segmentation using generative language models and Wikipedia. In Proceedings of WWW-2008.

[6] [6] Junlan Feng. , AT&T Labs Research, Query Parsing in Mobile Voice Search, WWW 2010, April 26-30, 2010, Raleigh, North Carolina, USA.

[7] [7]http://en.wikipedia.org/wiki/Multilayer_perceptron

[8] [8] www.cs.bham.ac.uk/~jxb/INC/l7.pd

[9] [9] Chon, Yohan, et al. "Mobility prediction-based smartphone energy optimization for everyday location monitoring." *Proceedings of the 9th ACM conference on embedded networked sensor systems*. ACM, 2011.