CONVERSION OF VOICE-TO-TEXT USING ANDROID

R.Ajith Kumar
Sri Vidya College of engineering &technology, Virudhunagar.
ajithkum21@gmail.com

R.Ramachandran
Sri vidya College of engineering &technology, Virudhunagar.
ramachandran0824@gmail.com

S.Gururagavendran AP/CSE
Sri vidya college of engineering &technology, Virudhunagar.
guruboy30@gmail.com

Abstract— The truth of our modern times is that you probably won't find a person who does not use innovative gadgets today. It is also true that people use different mobile applications not only for entertainment, but for communication as well. Every person probably knows plenty of applications that fit his or her needs most, but there are still some mobile "helpers" no person can live without expressing there thoughts.Speech is the first important primary need, and the most convenient means of communication between people. Speech recognition reduces the overhead caused by alternate communication methods. In this project, we are going to develop an off-line speech-to-text engine. With some algorithms we can process speech signals easily and recognize the text. The system acquires speech through microphone and processes the sampled speech to recognized uttered text. A speech to text system can be used for the communication of the deaf people. Speech recognition for voice uses a technique based on google speech recognition api. An added feature is that it also offers offline access as any commands given by a user are processed and executed locally on the device.

Keywords: hmm, android os, dvm, speech recognition, intents

I. INTRODUCTION

Android is an operating system for smartphones,tablets and laptops from the Google-sponsored Open Handset Alliance.Android is the leading mobile platform worldwide. Android is a Linux OS, and Android apps are programmed in Java.An Android app is a software application running on the Android platform. Because the Android platform is built for mobile devices, a typical Android app is designed for a smartphone or a tablet PC running on the Android OS.Initially developed by Android that is Google backed financially and later bought in 2005. Android was unveiled in 2007, with the founding of the Open Handset Alliance; a package of hardware, software and telecommunication companies devoted to advancing open standards for mobile devices.

Android is open source and this open source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers and enthusiast developers (Google releases the code under the Apache License) [1].And Android has a large community of developers for writing applications (apps) and these extend the functionality of devices, written primarily in customized version of the Java Programming language.There were approximately 700,000 apps available for Android, and the number of applications downloaded from Google Play - Android's primary store, was 25 billion in October 2012. A developer survey found that Android is the most popular platform for developers, used by 71% of the mobile developer population in April–May 2013.These factors have contributed making Android, platform, and the software of choice for technology companies. Hence, despite being primarily designed for phones and tablets, Android has additional applications with televisions, games, digital cameras and other electronics. The open nature of Android has further encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, and this add new features for advanced users or to bring Android devices officially released running on other operating systems.

Android's share of the global smart phone market, led by Samsung products, was 64% in March 2013. There were 11,868 different models of Android device, crores of screen sizes and eight OS versions simultaneously in March 2013 for use. The Android's success has made it a target as part of the so-called "smart phone wars" between technology companies. By May 2013, 48 billion apps have been installed from the Google Play store, and at September 3, 2013, there have been 1 billion Android devices activated.Since when Smart Phone was commercialized and distributed in 2009, which may commonly used the mobile App and also, under which all Internet services are expected to be modified and provided to App when the computing market moves from PC to Smart phone. Anyhow each request on developing and distributing mobile App occurs additionally. And students being interested with such applications and smart phones [2]. Android provide dozens of programs and apps for college students that can be very informative and useful if you want to keep up with the times and boost your studying process. The proposed system was implemented on the smart phones.It helps the deaf people to communicate with others to express their thougts.

2.RELATED WORKS

To start with we use a set of tools that are included in the Android SDK. Once we have downloaded and installed the SDK, we can access these tools right from our Android SDK, through the ADT plug-in, or from the command line. how speaker recognition followed by speech recognition is used to recognize the speech faster, efficiently and accurately. mfcc is used to extract the characteristics from the input speech signal with respect to a particular word uttered by a particular speaker. then hmm is used on quantized feature vectors to identify the word by evaluating the maximum log likelihood values for the spoken word[4]. after years of research and development the accuracy of automatic speech recognition (asr) remains one of the most important research challenges e.g. speaker and language variability, vocabulary size and domain, noise. the design of speech recognition system require careful attentions to the challenges or issue such as various types of speech classes, speech representation, feature extraction techniques, database and performance evaluation. this paper presents a study of basic approaches to speech recognition and their results shows better accuracy. this paper also presents what research has been done around for dealing with the problem of asr[5].

3.SPEECH-TO-TEXT CONVERSION

For the past several decades, designers have processed speech for a wide variety of applications ranging from mobile communications to automatic reading machines. Speech recognition reduces the overhead caused by alternate communication methods. Speech has not been used much in the field of electronics and computers due to the complexity and variety of speech signals and sounds. However, with modern processes, algorithms, and methods we can process speech signals easily and recognize the text. In this project, we are going to develop an on-line speech-to-text engine.

The system acquires speech at run time through a microphone and processes the sampled speech to recognize the uttered text. The recognized text can be stored in a file. We are developing this on android platform using eclipse workbench. Our speech-to-text system directly acquires and converts speech to text. It can supplement other larger systems, giving users a different choice for data entry. A speech-to-text system can also improve system accessibility by providing data entry options for blind, deaf, or physically handicapped users. Voice SMS is an application developed in this work that allows a user to record and convert spoken messages into SMS text message. User can send messages to the entered phone number.

Speech recognition is done via the Internet, connecting to Google's server. The application is adapted to input messages in English. Speech recognition for Voice uses a technique based on hidden Markov models (HMM - Hidden Markov

Model). It is currently the most successful and most flexible approach to speech recognition.

So we propose the system in mobile application (Android) to convert speech-to-text conversion engine in offline. There is no need of internet for the conversion of speech-to-text. Which is helpful for deaf people to use the smart phones for their communication.

4.BLOCK DIAGRAM AND ITS MODULE

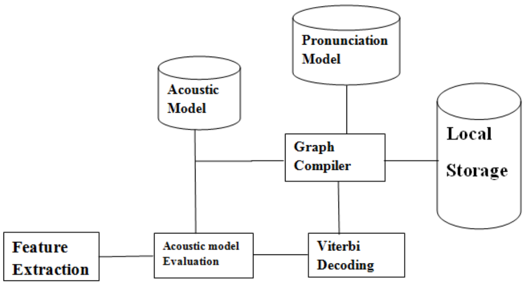

Figure 1: System Architecture

4.1MODULES

- Feature Extraction
- Acoustic Modeling
- Language Modeling
- Implementation

4.1.1 FEATURE EXTRACTION

First of all, recording of various speech samples of each word of the vocabulary is done by different speakers. After the speech samples are collected, they are converted from analog to digital form by sampling at a frequency of 16 kHz. Sampling means recording the speech signals at a regular interval. The collected data is now quantized if required to eliminate noise in speech samples. The collected speech samples are then passed through the feature extraction, feature training & feature testing stages. Feature extraction transformsthe incoming sound into an internal representation such that it is possible to reconstruct the original signal from it. There are various techniques to extract features like MFCC, PLP,
RAST, LPCC, but mostly used is MFCC.

Mel Frequency Cepstral Coefficients (MFCC)

MFCCs are used because it is designed using the knowledge of human auditory system and is used in every state of speech recognition system or art speech. MFCC is a standard method for feature extraction in speech recognition tasks. MFCC include certain steps applied on an input speech signal. These computational steps of MFCC include: - Framing, Windowing, DFTH, Mel filter bank algorithm, computing the inverse of DFT.

4.1.2 ACOUSTIC MODELING

An acoustic model is implemented using different approaches such as HMM, ANNs, dynamic Bayesian networks (DBN), support vector machines (SVM). HMM is used in some form or the other in every state of the art speech and speech recognition system.

Hidden Markov Model (HMM)

HMMs are used for acoustic modelling. There are two stochastic processes which are inter-related which are same as Markov Chain except that the output symbol and well as the transitions are probabilistic. Each HMM state may have a set of output symbols known as output probabilities and having a finite number of states Q= {q1, q2, ..., qn}. One process is related to the transitions among the states which are controlled by a set of probabilities called transition probabilities to model the temporal variability of speech. Other process is concerned with the state output observations O= {o1, o2, ..., on} regulated by Gaussian mixture distributions bj(ot) where 1<=j<=N, to simulate the spectral variability of speech. Any and every sequence of states that has the same length as the symbol sequence is possible, each with a different probability. The sequence of states is sad to be "hidden" from the observer who only sees the output symbol sequence, and that is why this model is known as Hidden Markov Model. The Markov nature of the HMM i.e. the probability of being in a state is dependent only on the previous sate, admits use of the Viterbi algorithm to generate the given sequence symbols, without having to search all possible sequences. At each distinct instance of time, one process is assumed to be in some state and an observation is produced by the other process representing the current state. The underlying Markov chain then changes states according to its transition from state i to state j denoted as: aij= P[Qt+1 = j| Qt = i].

4.1.3 LANGUAGE MODELING

Language models are used to guide the search correct word sequence by predicting the likelihood of nth word using (n-1) preceding words.

Pronunciation Modelling

In pronunciation modelling, during recognition, the sequence of symbols generated by acoustic model HMM is compared with the set of words present in dictionary to produce sequence of words that is the system's final output contains information about which words are known to the system and how these words are pronounced i.e. what is their phonetic representation.

4.1.4 IMPLEMENTAION

Voice Recognition Activity class

Voice Recognition Activity is startup activity defined as launcher in AndroidManifest.xml file. REQUEST_CODE is static integer variable, declared on the beginning of activity and used to confirm response when engine for speech recognition is started. REQUEST_CODE has positive value. Results of recognition are saved in variable declared as *ListView* type. Method onCreate is called when activity is initiated. This is where the most initialization goes: setContentView (R.layout.voice_recognition) is used to inflate the user interface defined in res > layout > voice_recognition.xml, and findViewById(int) to programmatically interact with widgets in the user interface. In this method there is also a check whether mobile phone, on which application is installed, has speech recognition possibility. Package Manager is class for retrieving various kinds of information related to the application packages that are currently installed on the device. FunctiongetPackageManager() returns PackageManager instance to find global package information. Using this class, we can detect if the phone has a possibility for speech recognition. If a mobile device doesn’t have one of many Google’s applications which integrate speech recognition, further work of this application Voice SMS will be disabled and message on the screen will be “Recognizer not present”. Recognition process is done trough one of Google’s speech recognition applications. If recognition activity is present user can start the speech recognition by pressing on the button and thus launching startActivityForResult (Intent intent, int requestCode). The application uses startActivityForResult() to broadcast an intent that requests voice recognition, including an extra parameter that specifies one of two language models. Intent is defined with intent.putExtra (RecognizerIntent).

XML FILES

Application consists of two different interfaces. When the user runs application screen is defined in voice_recognition.xml. The linear arrangement of elements allows adding widget one below another. Width and height are defined with *fill_parent* attribute, which means to be equal as parent (in this case the screen). The second interface, defined within sms.xml file, is displayed when the user chooses one of offered messages. AndroidManifest.xml realizes installing and launching applications on the mobile device. Every application must have an AndroidManifest.xml file (with precisely that name) in its root directory. The manifest presents essential information about the application to the Android system, information the system must have before it can run any of the application's code. It defines the activities and permissions required for the application.

CONCLUSION AND FUTURE WORK

This project developed an android application for real-time speech-to-text conversion system. That will be helpful for deaf people to use smartphones for communicating with others. In this proposed system, it has much faster speech-to-text conversion rate than online conversion system. The proposed system has many added advantage that it is cost effective process which consumes less processing power than the online conversion system. It is an automated process with the help of Google backend speech recognition database.

As for the future work, instead of opening the application manually we are going to trigger the application automatically when the call is established. We will develop the application like an interactive call based messaging application.

REFERENCE

[1] Lemos, Robert: Open source vulnerabilities paint a target on Android March 25, 2011, accessed on December 27, 2012.

[2] R.Klevansand R.Rodman, “Voice Recognition, Artech House, Boston, London 1997.

[3] Zahi N.Karam,William M.Campbell “A new Kernel for SVM MIIR based Speaker recognition “MIT Lincoln Laboratory, Lexington, MA, USA.

[4] The Application of Real-Time Voice Recognition to Control Critical Mobile Device Operations by Omyonga Kevin, Kasamani Bernard Shibwabo at IJETT Jan 2014.

[5] M.A.Anusuya and S.K.Katti, “Speech Recognition by Machine: A Review”, (IJCSIS) International Journal of Computer Science and Information Security, vol. 6, no. 3, pp.181-205, 2009.

[6] Anusuya, M. A., & Katti, S. K.. Front end analysis of speech recognition: A review. International Journal of Speech Technology,Springer, vol.14, pp. 99–145, 2011.

[7] Samudravijaya K. Speech and Speaker recognition tutorial TIFR Mumbai 400005.

[8] Dr.Ravi Sankar, Tanmoy Islam, Srikanth Mangayyagari, “Robust Speech/Speaker Recognition Systems”.