International Journal of Recent Trends in Engineering & Research
ISSN(ONLINE) : 2455-1457
IMPACT FACTOR : 3.344

SPOKEN DIALOGUE SYSTEM USING SPEECH RECOGNITION TECHNOLOGY

S.Azhagu vediappan[1],J.Baskaran[2],M.Gopalakrishnan[3],G.senthil kumar[4]
[1,2,3]*Information Technology,S.K.P Engineering College*
[4]*Assisstant Proffesor,Information Technology,S.K.P Engineering College*

Abstract: This project is used to create a educational tools for speech technology. The demand for such tools is increasing with the advent of speech as a medium for man–machine communication. The world wide web was chosen as our platform in order to increase the usability and accessibility of our computer exercises. The aim was to provide dedicated educational software instead of exercises based on complex research tools. Currently, the set of exercises comprises basic speech analysis, multi-modal speech synthesis and spoken dialogue system.Students access web pages in which the exercises have been embedded as applets. This makes it possible to use them in a classroom setting, as well as from the students' home computers.
Keywords: *Mel Frequency Cepstral Coefficients (MFCC), DSP Starter Kit (DSK), Code Composer Studio (CCS), Gaussian Mixture Model (GMM), Euclidian Distance (ED*

I. INTRODUCTION

Speech recognition is an important field of digital signal processing. There are various objectives for the development of Automatic Speech Recognition (ASR). Main objective of ASR is to extract features, characterize and recognize speaker. The application can be aimed at recognition to be performed either on isolated words or utterances or on continuous speech. There are various languages spoken in this world that makes to consider the one of the language for the recognition system. There are also situations, when recognition system should be speaker dependent or independent. The most difficult class of recognition system is to develop speaker independent recognition on continuous speech.

II. RELATED WORKS

These SR technologies have been applied to automati- cally transcribe instructor's lecture and process the tran- scription to acquire near verbatim lecture transcripts for students [1], [2], [3]. The benefits of producing lecture transcripts have shown to enhance both learning and teaching. Students could make up for missed lectures as well as to corroborate the accuracy of their own notes during the lectures they attended. Coupled with a recorded audio/video lecture track and copies of the lecture slides, students could recreate the lecture material for replicating the lecture at their own learning pace. These lecture transcripts and additional multimedia recordings also enable instructors to review their own teaching performance and lecture content to assist them to improve individual pedagogy [1]. Likewise, SR has been used for quickly searching certain keywords to retrieve specific text or video lecture content [4],
KEYWORDS: *Mel Frequency Cepstral Coefficients (MFCC), DSP Starter Kit (DSK), Code Composer Studio (CCS), Gaussian Mixture Model (GMM), Euclidian Distance (ED)*

III. PROPOSED SYSTEM

A speech technology toolkit that serves as a basis in the creation spoken language systems has been developed.Our courses on speech technology includes an introductory section on basic phonetics and speech analysis.

IV. MODULE

4.1)VOICE INPUT:

With the help of microphone, audio is input to the system, the pc sound card produces the equivalent digital representation of received audio.

4.2)DIGITIZATION:

The process of converting the analog signal into a digital form is known as digitization , it involves the both sampling and quantization processes. Sampling is converting a continuous signal into discrete signal, while the process of approximating a continuous range of values is known as quantization.

4.3)ACOUSTIC MODEL:

An acoustic model is created by taking audio recordings of speech, and their text transcriptions, and using software to create statistical representations of the sounds that make up each word. It is used by a speech recognition engine to recognize speech. The software acoustic model breaks the words into the phonemes.

4.4)LANGUAGE MODEL:

Language modeling is used in many natural language processing applications such as speech recognition tries to capture the properties of a language and to predict the next word in the speech sequence. The software language model compares the phonemes to words in its built in dictionary.

4.5)SPEECH ENGINE :

The job of speech recognition engine is to convert the input audio into text , to accomplish this it uses all sorts of data, software algorithms and statistics. Its first operation is digitization as discussed earlier, that is to convert it into a suitable format for further processing. Once audio signal is in proper format it then searches the best match for it. It does this by considering the words it knows, once the signal is recognized it returns its corresponding text string.

V. SPEECH SYNTHESIS

Speech synthesizer converts written text into spoken language. Speech synthesis is also referred to as *text-to speech (TTS) conversion.*
The major steps in producing speech from text are as follows:

VI. STRUCTURE ANALYSIS

Process the input text to determine where paragraphs, sentences and other structures start and end. For most languages, punctuation and formatting data are used in this stage.

VII. TEXT PRE-PROCESSING

analyze the input text for special constructs of the language. In English, special treatment is required for abbreviations, acronyms, dates, times, numbers, currency amounts, email addresses and many other forms. Other languages need special processing for these forms and most languages have other specialized requirements.

VIII. SYNTHESIZER AS AN ENGINE

The basic functionality provided by a Synthesizer is speaking text, management of a queue of text to be spoken and producing events as these functions proceed. The Synthesizer interface extends the Engine interface to provide this functionality.

The following is a list of the functionality that the javax.speech.synthesis package inherits from the javax.speech package and outlines some of the ways in which that functionality is specialized.
The properties of a speech engine defined by the Engine Mode Desc class apply to synthesizers. The SynthesizerModeDesc class adds information about synthesizer voices.
Synthesizers are searched, selected and created through the Central class in the javax.speech package.

IX. SYNTHESIZER AS AN ENGINE

The basic functionality provided by a Synthesizer is speaking text, management of a queue of text to be spoken and producing events as these functions proceed. The Synthesizer interface extends the Engine interface to provide this functionality.
The following is a list of the functionality that the javax.speech.synthesis package inherits from the javax.speech package and outlines some of the ways in which that functionality is specialized.
The properties of a speech engine defined by the Engine Mode Desc class apply to synthesizers. The SynthesizerModeDesc class adds information about synthesizer voices.
Synthesizers are searched, selected and created through the Central class in the javax.speech package.

X. SELECTING VOICES

Most speech synthesizers are able to produce a number of voices. In most cases voices attempt to sound natural and human, but some voices may be deliberately mechanical or robotic.
The Voice class is used to encapsulate the four features that describe each voice: voice name, gender, age and speaking style. The voice name and speaking style are both String objects and the contents of those strings are determined by the synthesizer. Typical voice names might be "Victor", "Monica", "Ahmed", "Jose", "My Robot" or something completely different. Speaking styles might include "casual", "business", "robotic" or "happy" (or similar words in other languages) but the API does not impose any restrictions upon the speaking styles. For both voice name and speaking style, synthesizers are encouraged to use strings that are meaningful to users so that they can make sensible judgments when selecting voices.

XI. SPEECH RECOGNITION

Speech recognition we mean that converting speech into text for this purpose we have use the ocvolume speech recognition engine which is entirely built in java and has provide good results. For using this engine firstly we need to make the object of ocvolume through the constructor.

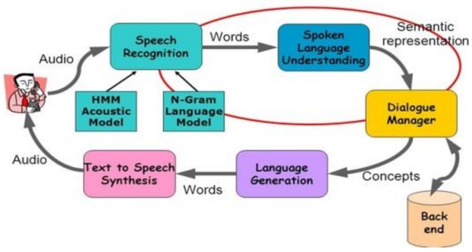
ocvolume(java.lang.Stringdict,
java.lang.String folder)

constructor to create a speech recognition engine using VQ for recognition

PARAMETERS:

dict - file path of the dictionary file that contains all the words that the engine can recognize folder - path of the folder where *.vq are located.

XII. ARCHITECTURE



XIII. CONSLUTION

In this paper we depicted the real time hardware implementation of speech recognition using DSP processor software development kit, DSK-TMS320C6713 with Code Composer Studio (CCS). MFCC algorithm calculates cepstral coefficients of Mel frequency scale. After feature extraction from recorded speech, each Euclidian Distance (ED) from all training vectors is calculated using Gaussian Mixture Model (GMM). The command/voice having minimum ED is applied as similarity criteria. The timing analysis is done for various individual blocks of algorithm. The time required for processing in DSP and PC processors are compared. Timing analysis in MATLAB is taking less time this is due to more Clock speed of CPU and more memory.

REFERENCE

1. D.A. Raynolds, “Speaker Identification and verification using Gaussian mixture speaker models”, Speech Communication, Vol.17, No. 1-2, pp. 91-108, 1995.
2. Spectrum Digital Incorporated, TMS320C6713 DSK Technical Reference, 2004.
3. Texas Instruments, Code Composer Studio User’ Guide, 2005.
4. Molau S., Pitz M., Schluter R. and Ney H., “Computing Mel-frequency Coefficients on Power Spectrum”, Proceeding of IEEE ICASSP-2001, Vol. 1, pp.73-76, 2001.