Research Paper Engineering

E-Examination For Handicapped People

*N. G. Aaras **B. R. Gawhale ***S. S Kharade
****P. B. Makasare *****L. J. Sankpal

*

**

***

****

*****

ABSTRACT

The "E-Examination For Handicapped People" is basically an online examination system. It reflects the justification and objectivity of examination. It helps to release the workload of teachers and students. The current online examination systems do not allow handicapped people to appear for the examination. Hence, the proposed application is designed for the blind people so that they can appear for examination; it is also useful for the other handicapped people like upper limb disability.
This research work provides the speech user interface for blind people to interact with the application. The proposed application will dictate the questions to the candidate with the help of Speech Synthesis. The application will accept the answers from candidates through voice commands using Speech Recognition. The proposed system also allows users (examiners/teachers) to add their questions to conduct the examination. The questions can be added easily by uploading a file. As the proposed system is online; result generation will be quick.

Keywords :

I. INTRODUCTION

The traditional way for conducting examination for the handicapped people requires scribe/writer. These candidates face difficulties in using the scribes for examinations. It is hard for them to dictate the answers to the scribes. Because, the scribes might be of lower qualification, they can't interpret their words and write down them as it is.

"E-Examination For Handicapped People" is a software solution, which allows a particular company or institute to arrange, conduct and manage examinations via online environment. This can be done through the Internet and Local Area Network Environment.

Candidate can answer his/her examination paper on the computer and submit their answers. The Examination Software evaluates the submitted answers and the results will be available immediately after completion of the examination.

The proposed application avoids main drawback of the current online examination system is that, it provide a facility to handicapped people to interact with the system.

II. SYSTEM ARCHITECTURE

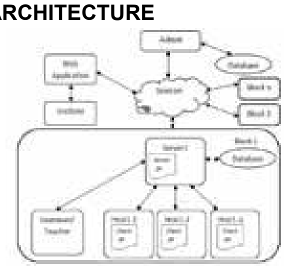

Fig. 1 SYSTEM ARCHITECTURE

Explanation:
System Architecture basically contains web application and java application.

A) Web Application:
The institute which wants to conduct examination will register through web application. Registering institute's administrator will fill basic information of the institute at the time of registration process. After Registration he will get the java application (.zip file) which can be used for examination purpose.

B) Java application:
Zip file contains two .jar files, one for server and other for client. Server.jar file will reside on server side (college's local server) and by running this file teacher or examiner uploads the questions for conducting examination. Client.jar file will reside on local host of the college. It provides functionality to candidates to appear for the examination. Application will dictate questions to handicapped candidates and accept answers through voice commands. Normal candidate can give examination through Graphical User Interface.

C) Databases:
There are two databases first at the admin server and other at the college server. Admin server will store all the details about the institutes who are using the proposed system. And the college server will store all the accounts of teachers, all questions, results, etc. Because of this type of architecture the databases will be secured. The institute need not reveal its private data like teachers' information and questions.

The main feature provided in the proposed system is the Speech User Interface for the handicapped people. This is achieved by the series of steps like Speech Recognition (i.e. speech to text conversion) and Speech Synthesis (i.e. text to speech conversion). The description of these steps is explained in following sections.

III. ACCEPTING QUESTIONS

Examiner can upload questions by using Microsoft Excel (.xls) file. The questions will be accepted in specific format.

Apache POI provides pure Java libraries for reading and writing files in Microsoft Office formats, such as Word, PowerPoint and Excel. Proposed system uses Apache POI to read the excel file which will be used to upload questions to the database. The questions will be read and stored with the options and the correct answer in the database. These questions and answers will be used for conducting the examination and for generating result.

The advantage of this is that the examiner will be able to set question paper easily. He can add the questions in the database simply by uploading a file.

IV. SPEECH SYNTHESIS

The speech synthesis process has a series of steps [2]. They are shown in Fig. 4 below.

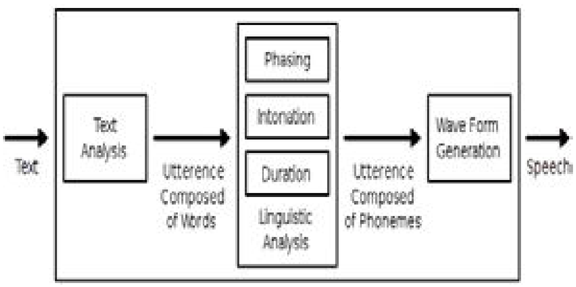

Fig. 2 Steps of the speech synthesis process

A) Text Analysis:

This contains string tokenization and normalization. It breaks a string into tokens which could be words, phrases, or symbols. [2].

Considering that punctuation marks are removed from words, there will be an issue with tokenization. In the normalization phase [3], text is transformed into a more reliable way to be processed means it will become consistent so that it will be easier to deal with. An example of normalization would be “don’t” and “do not”. Furthermore, an important step in normalizing text is expanding abbreviations. For example, if the string “WWW” is encountered, it will be normalized to “world wide web.” [2]

B) Linguistic Analysis:

The normalized tokens are assigned with phonetic transcriptions. A phonetic transcription of a word is the visual representation of speech sounds. For example, the word “elephant” has the phonetic transcription “el uh fuh nt.” This process is called text-to-phoneme or grapheme-to-phoneme conversion [2].

A grapheme is a unit in a written language such as a letter, a number, or a punctuation mark. On the other hand, a phoneme is a unit of sound to represent the utterance of a grapheme, such as the letter “x” which is uttered “eks”. When the process of text-to-phoneme or grapheme-to-phoneme is finished, the text is divided into prosodic units [4]. These units determine whether the text is a clause, phrase, or sentence. To elaborate more, prosody is the rhythm, stress, and intonation of speech. Therefore, when the text is divided into these kinds of units, the form of utterance of the text can be determined whether it is a statement, question, command, emphasis, and so on [4].

C) Waveform Generation:

After completing the previous steps, the text will be in a Symbolic Linguistic Representation form, which is phonetic transcriptions and prosody merged together. The main purpose now is converting this text into sound [2]. A phone is a speech sound and a diaphone is an adjacent pair of speech sounds. Based on this there are four synthesizer technologies that are mostly used.

1) Unit Selection Synthesis:
2) Diaphone Synthesis:
3) Domain-Specific Synthesis:
4) HMM-based Synthesis:

The application uses an open source speech synthesizer FreeTTS [8] for the implementation of all above steps related to speech synthesis. It is written entirely in the Java programming language and developed by Speech Integration Group of Sun Microsystems Laboratories. FreeTTS adopts diaphone synthesis in the wave generation step.[4]

The questions stored in the database by the examiner will be retrieved and shown to the user. The question and the options will be dictated using TTS for handicapped people.

V. SPEECH RECOGNITION

The speech recognition process has a series of steps [2] as shown in Fig. 2 below.

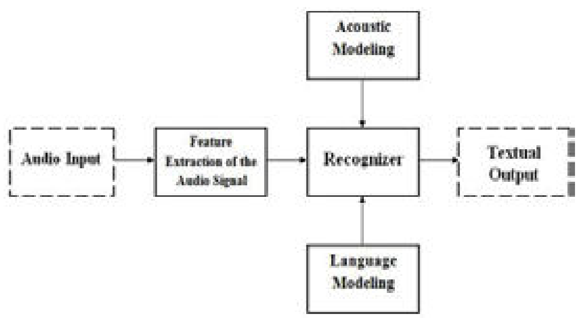

Fig. 3 Steps of the speech recognition process

A) Audio Input:

User will give audio input through microphones. He will give answers to respective questions and also interact with the application via speech commands.

B) Feature Extraction of the Audio Signal:

After obtaining the input audio signal, it has to be processed and its features have to be extracted [1]. This step takes place when the signal is divided into small intervals (in terms of milliseconds), and a feature vector is generated for each interval. The input utterance is represented as a sequence of these feature vectors, where each vector consists of coefficients, called cepstral coefficients,

C) Acoustic Modeling:

Hidden Markov Models (HMM) are used for acoustic modeling process. Such models are probabilistic models used for pattern recognition, a field in which Speech Recognition is.

Every known phoneme has its own HMM, represented by a graph with different states (nodes) and transition probabilities [6] from one state to another. These HMMs are obtained after extracting the features of the recorded audios that were used to prepare and train the acoustic model.

Every word is made up of a number of phonemes, and the HMM of a certain word is made up of the concatenation of the HMMs of the phonemes that constitute the word. These HMMs are present in the acoustic model [6] which the recognition procedure relies on. The obtained features vectors are then matched to the available HMM phonemes, to be assigned to the one that most matches it (with the highest probability).

Then, the combined assigned HMMs are used to find the HMM of the whole word to be recognized. Combining phonemes' HMMs together is also a probabilistic process. Fig. 3 shows an example of HMMs matching. Matching HMMs, on both phoneme and word levels, is done by the Viterbi algorithm [1][5].

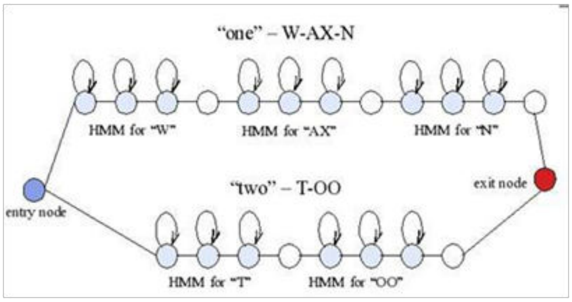

Fig. 4 An example of HMMs matching

D) Language Modeling:
In a similar manner to the one above, the HMM of a meaningful sequence of words is a concatenation of the HMMs of the words included in this sequence. To handle this phrase/ sentence HMM formation, a language model is needed that knows how to group different words together to provide the desired output. [1].

E) Textual Output:
After the recognizer completes the acoustic modeling and the language modeling, the obtained textual output is returned.

The application uses sphinx-4 [6][7] to implement all above steps, which is an open source speech recognizer.

VI. JSGF GRAMMAR
The proposed application uses Sphinx-4 for speech recognition which uses the grammar format as java speech grammar format (JSGF) to specify the grammar rules[9].

For Example,
grammar answer;
<options>=one|two|three|four;
This will accept any one option as an answer from user.

This grammar will be defined in a file with ".GRAM" extension and will be used by sphinx-4 for recognition of words.

VII. CONCLUSION
Therefore keeping in mind the flaws of the existing system, the proposed system seems to be far better and efficient in terms of technology and integration point of view.

The accuracy of the speech recognition system was among the top challenges. Also, studying new codes and trying to fix them was a problem as well.

The proposed system will provide a better option for handicapped people to appear for examination.

REFERENCES

[1] L. R. Rabiner, (1989). "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. of the IEEE, vol. 77, no. 2, pp. 257–286. | [2] Hussein Mohsen, Karim Jahed, Marwan Fawaz, (2011). "Multi-Purpose Speech Recognition and Speech Synthesis System",IEEE MULTIDISCIPLINARY ENGINEERING EDUCATION MAGAZINE, VOL. 6, NO. 4, DECEMBER 2011. | [3] Suresh Kumar Thakur et al., "Study of Various kinds of Speech Synthesizer Technologies and Expression for Expressive Text To Speech Conversion System", (IJAEST) INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES Vol No. 8, Issue No. 2, 301 – 305. | [4] Book: "NATURAL SOUNDING TEXT-TO-SPEECH SYNTHESIS" by SAMUEL THOMAS | [5] Book: "Speech Recognition using Hidden Markov Model performance evaluation in noisy environment" by Mikael Nilsson and Marcus Ejnarsson. | [6] Sphinx-4: "A Flexible Open Source Framework for Speech Recognition", Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, Joe Woelfel. | [7] (2008). Sphinx-4, a speech recognizer written entirely in Java Programming Language [Online]. Available: http://cmusphinx.sourceforge.net/sphinx4. | [8] (2005). FreeTTS, a speech synthesizer written entirely in Java Programming Language [Online]. Available: http://freetts.sourceforge.net/docs/index.php | [9] "A Graphical Tool for Handling Rule Grammars in Java Speech Grammar Format", Kallirroi Georgila, Nikos Fakotakis, George Kokkinakis, Wire Communications Laboratory, Electrical and Computer Engineering Dept. University of Patras,265 00 Rion, Patras, Greece.