DOI: http://dx.doi.org/10.26483/ijarcs.v9i1.5155

ISSN No. 0976-5697

Volume 9, No. 1, January-February 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

DICTIONARY APPLICATION WITH SPEECH RECOGNITION AND SPEECH SYNTHESIS

Gaikwad Vijayendra Sanjay,
Assistant Professor,
Dept. of Computer Engineering,
ABMSP's Anantrao Pawar College of Engineering & Research, Savitribai Phule Pune University,
Pune, Maharashtra, India

Kamble Sanket Mohan
Graduate Student,
ABMSP's Anantrao Pawar College of Engineering & Research
Savitribai Phule Pune University,
Pune, Maharashtra, India

Kundur Ajinkya Sham,
Graduate Student,
ABMSP's Anantrao Pawar College of Engineering & Research,
Savitribai Phule Pune University,
Pune, Maharashtra, India

Hulbutti Akash Huchcheshwar
Graduate Student,
ABMSP's Anantrao Pawar College of Engineering & Research,
Savitribai Phule Pune University,
Pune, Maharashtra, India

Thorve Shubham Prakash
Graduate Student,
ABMSP's Anantrao Pawar College of Engineering & Research,
Savitribai Phule Pune University,
Pune, Maharashtra, India

Abstract: In today's environment while reading documents (online/offline) or browsing through many websites, we face many unfamiliar words, as we cannot get the meaning of those words we lack in complete understanding of the concepts or situations. So, to overcome this problem we are developing a speech to text and text to speech input output method for our Voice Based Dictionary system. The system will be implemented using Java Sphinx-4 API for speech synthesis and speech recognition. The application or the system uses Java concepts like Swing API, JavaFX for graphical user interface (GUI) the system will be deployed in the background process using Java multi-threading support.

Keywords: Speech Recognition, Searching and Mapping, Speech Synthesis, Sphinx, Hidden Markov Model, MARY TTS.

I. INTRODUCTION

The fastest growing technologies such as Speech Recognition and Speech Synthesis have gained a lot new netizen [1]. Speech based research is the widest growing research nowadays. One demerit/limitation of speech based recognition is the sharpness of the voice quality that is to be captured or recognized [2]. In Speech Recognizer module the input data which needs to be recognized has to be provided slowly and clearly [3]. The Dictionary consists of many words and meanings so, it is being stored as same as the operating system's file system database. The microphone quality also matters because of noise issues. Speech Recognition has many complex applications [4]. Also, since it is used as a replacement for typing. Speech Recognition also lacks in voice sharpness, due to which user has to speak slowly and clearly [3].

The system consists of these speech domains in a well-defined way. Domains can be easily interpreted. System recognizes voice and synthesizes for displaying and dictating the word and meaning. The Sphinx API internally consists of noise removal techniques which has impact on Performance metrics of the system [2]. System manipulates the input data/text, obtained by converting the speech-to-text, then it attempts to look for the word from database and it searches the particular meaning with the same method, when a specific match is found it is shown on the console window and dictated to the user.

To use this Speech Based Domains in system, two methods are used: The Hidden Markov Model(HMM), and a slight different and not used much called time warping method. Due to its high Accuracy and low computational need the HMM method is known to be DE-facto standard [3]. Sphinx uses HMM method.

In this paper we use following methodologies:

(1) Real time Speech Recognition.

(2) Searching Algorithms.

(3) Visualized Speech Synthesizer.

II. EXISTING RELATED WORK

A. CMU Sphinx IV

CMU Sphinx 4 is an open source speech recognition tool. The Sphinx 4 API is developed in Java, which consists of different modules. Architecture of Sphinx 4 is shown in Figure 1. Modules in Sphinx 4 consists of Front End, Decoder,

Knowledge Base [2]. Front end of the Sphinx 4 is responsible taking input wave-forms and converts them into an appropriate form. Second module, Decoder is responsible for decoding/recognizing but before recognizing, knowledge base must be loaded. Knowledge base is nothing but the information required for speech decoder/recognition phase. Knowledge base can vary for different languages. Knowledge base consists of language model, Acoustic model and Dictionary contains information which will be used by decoder for recognizing purpose.

1) Front End: Front end has the responsibility of transforming the input waveform into the sequence of feature. Cepstral audio signals are extracted in this block/section of Sphinx 4. This signals then forwarded to decoder block.

2) Decoder: Decoder consists of 3 components as search manager, linguist, acoustic scorer.

a. Search Manager:

Search manager constructs the tree for better hypothesis comparison and search [2]. The tree is generated using the information available, in obtained linguist. On top of that the search manager also communicates with the acoustic scorer for incoming data.

b. Linguist:

The linguist block is responsible for translating the linguist constraints provided through the grammar [2]. Which is also used by search manager. Linguistic constraints are generally provided as context free grammar (CFG) through grammar file (. gram file).

c. Acoustic Scorer:

The acoustic scorer generates state output probability for various states. Acoustic scorer provides these probabilistic scores on demand of search manager [2]. For computing these scores, the scorer must share the probabilities with the front end.

3) Knowledge Base:

The knowledge base provides whatever resources needed by the decoder in order to carry out its functionality [2]. Knowledge base provide/consists of language model (. lm file), Dictionary (. dic file), Acoustic model (included in Sphinx 4 API jar file).

Dictionary consists of words with their phoneme sequence. Language model consists of grammar required for understanding of sequences [2]. The acoustic model consists of acoustic data for every phoneme in dictionary.

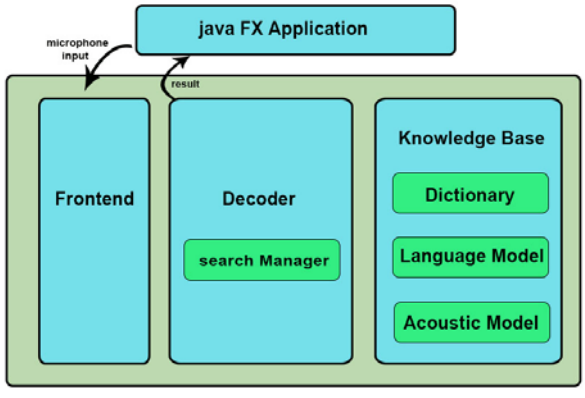

Figure 1 – Sphinx 4 Architecture

B. Hidden Markov Model

A HMM is a statistical model for sequences of discrete symbols [5]. From many years HMM is being used, for speech recognition and for perfect gene finding task. HMM provides effective framework which is used for modeling time-varying spectral vector sequences. HMM is applied efficiently on well-known biological problems, such as [5],

1. Protein Secondary Structure Recognition.
2. Multiple Sequence Alignment.
3. Gene Finding.

HMM is a heart of all modern Speech Recognition Systems [5]. Well known applications of HMM are in Reinforcement Learning, Temporal Pattern Recognition such as Speech-Gesture Recognition.

C. MARY TTS

MARY TTS is an open source voice creation toolkit of unit selection [6]. It generates the voice based on HMM [6]. MARY stands for Modular Architecture for Research on Speech Synthesis. The MARY TTS is successfully implemented in British English, Turkish, Telugu and Mandarin Chinese languages. The voice can be easily generated for the supported languages. MARY TTS also gives support for new language support/implementation [6].

The MARY TTS is developed in Java. It consists of following modules.

A. The Preprocessing

The preprocessing module consist tokenizer, abbreviation expansion and numeral expansion [7].

B. Natural Language Processing

To calculate the speech-relevant data the input text natural language processing is done. Initially in NLP part of speech labeling is done and shallow parsing (chunking) [7].

III. PROPOSED SYSTEM

The current applications, which uses such functions does not uses the off-line speech recognition and off-line database for storage purpose. The proposed system is overcoming this drawback of existing desktop applications. On top of that the proposed system application will be running in background, in the form of process. Which makes it completely hidden from the user but, will be functioning in background.

The Application will be totally functioning on user's voice/speech which improves the interaction of user with system. The application initially will be running in background. To invoke the application a specific command is predefined by developers, the application will be continuously listening for the invocation command. Whenever the application detects the specific command application will start listening again but now it will be listening for the word from the user voice, for which the meaning needs to be searched. This phase of application will be running for a specific period of time, if it couldn't locate any word then it will again go in listening mode for invocation command.

After listening the word, the interpreted word will be searched in the off-line dictionary database, stored in the system. If the word is not present in the dictionary then application will show appropriate error messages through pop-ups. If word is found in dictionary then it will collect all the

variations of the word and returns an appropriate meaning and also shows the variations.

The word and meaning will be shown in a pop-up window at a bottom-right corner of the screen, also it will dictate the meaning to the user.

Figure 2 shows the architecture of application with speech recognition and speech synthesis.

A. Speech Recognizer

In this application the user is going to interact with his/her voice. So, for detecting and recognizing the voiced input we are using CMU Sphinx 4 API [2]. We are providing the language model (. lm file), Dictionary (. dic file). The acoustic model of CMU Sphinx 4 API is used for speech recognition purpose. The dictionary file contains the English dictionary words and phonemes for those words and the language model contains the context free grammar (CFG) for constraint matching [2].

B. Searching and Mapping

The main functionality of synthesizer module is to convert the textual data into the artificial human voice.

Searching and mapping module consists of searching the recognized word in the dictionary file and then mapping the meaning of that word. Dictionary file consisting the word and meaning is stored in Operating System File System in a text file (.txt file). Whenever the input word is recognized it will be searched in this text file if word found then the meaning for that word will be mapped and further it will be forwarded to the synthesizer module.

C. *Speech Synthesizer*

This module converts the word, meaning retrieved from searching and mapping module. This converted information is in the form of audio which is delivered to the user through speakers/headphones and also displayed through the pop-up messages.

IV. ACKNOWLEDGMENT

We would like to thank Prof. Manoj Mulik, Head of Computer Department at ABMSP's A.P.C.O.E.R, Prof. Rama Gaikwad, Project Co-ordinator at ABMSP's A.P.C.O.E.R, Prof. Vijayendra Gaikwad, Project Guide at ABMSP's A.P.C.O.E.R, for making this team possible. We also thank github.com, for their open source contents/resources.

We would also like to thank and acknowledge P. Lamere, P. Kwok, W. Walker, E. Gouva, R. Singh, B. Raj, P. Wolf for their useful resources and information [2].

V. CONCLUSION

The application being developed is dictionary application which works/operates on Human speech. The CMU Sphinx 4 API for speech recognition purpose. The application delivers the meaning of the word by searching and mapping the meaning from English dictionary (Database). The word and meaning are synthesized and converted to audio format using open source voice generation toolkit MARY TTS.

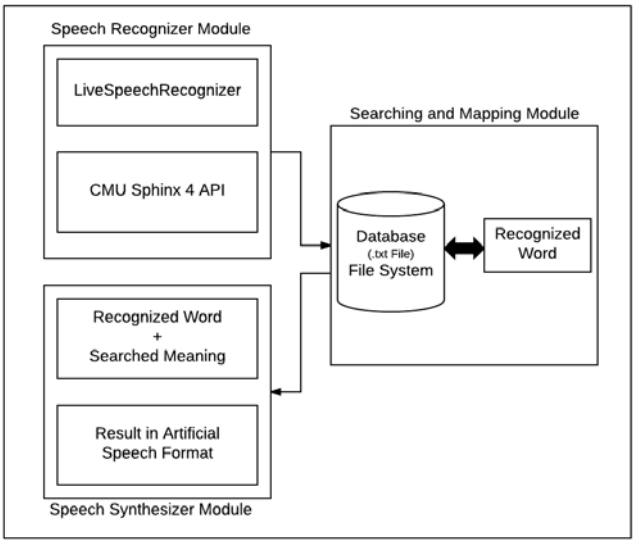

Figure 2 – System Architecture

VI. REFERENCES

[1] J. Twiefel, T. Baumann, S. Heinrich and S. Wermter, “Improving Domain-Independent Cloud-Based Speech Recognition With Domain-Independent Phonetic Post-processing”, Proceedings of the Twenty-Eight AAAI Conference on Artificial Intelligence, 2014.

[2] P. Lamere,P.Kwok,W. Walker,E. Gouva,R. Singh,B .Raj,P. Wolf and J. Woelfel. ” Sphinx-4: A flexible open source framework for speech recognition” ,Sun Micro-system Inc. 2004.

[3] S. Batlouni, H. Karaki, F. Zaraket, F. Karameh, “Mathifier – Speech Recognition of Math Equation”,18th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2011, Beirut, Lebanon, December 11-14, p. 301, 2011.

[4] Information retrieved on November 10th, 2017, from http://www.guidogybels.eu/asrp4.html

[5] J. Watada,IEEE ,Hanayuki. “Speech Recognition in a Multi-Speaker Environment by using Hidden Markov Model and Mel-frequency Approach”, Third international conference on computing measurement control and sensor network.

[6] S. Pammi,M. Charfuelen,M. Schröder, ”Multilingual Voice Creation Toolkit for MARY TTS Platform”.

[7] Documentation retrieved on October 20th, 2017 from, mary.dfki.de/documentation/index.html.