International Journal of Pure and Applied Mathematics
Volume 117 No. 21 2017, 119-125
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

Text Reader for Blind: Text-To-Speech

L. Mary Gladence [1], Shubham Melvin Felix[2], Aatisha Cyrill[3]
[1]*Assistant Professor/IT,Sathyabama University,Chennai, India*
[2]*UG student,IT,Sathyabama University,Chennai,India*

***Abstract*-With humans moving towards higher standards of living and to a more digitalised and interconnected world , computers prove to play an eminent role by providing the most efficient and optimal ways in achieving the required goals. Human resource and the computer system give the perfect paradigm of a trouble shooter. Such systems need to be user friendly, accurate, and multitasking as they are needed by every section of people. But when it comes to visually impaired people they (the software's/systems) pose a great deal of struggle and difficulty and the complete utilization of the facilities is hampered while using the visual interface.This can be solved by using the hearing capability. Keeping this in mind the software will be able to read the text present in the screen, webpage, document or a text entered in a text box using FreeTTS text-to-speech synthesizer. The text will be converted into a speech by analyzing and processing the text using Natural Language Processing (NLP) and then using Digital Signal Processing (DSP) technology to convert this processed text into synthesized speech representation of the text. Through the speech or voice visually impaired people can be able to hear large volume of text easier. Other than just the text to speech facility the software will have a facility to extract the text into an audio file like *.mp3,*.wav etc.It will be an efficient way in which blind people can also interact with the computer and utilize the facilities of the computer.**

Keywords: **FreeTTS, Text-to-speech synthesis, Natural Language Processing, Digital Signal Processing.**

I. INTRODUCTION

Artificial speech has been a dream of the humankind for centuries. The computer is a silent teacher for most. Often computer instructions are transmitted visually through textual presentation--analogous to conducting a lesson using the chalkboard without speaking. The majority of currently available educational software provides feedback through pictures, written words or electronic beeps and tunes. Special-education teachers are well aware of the problems created when students with learning problems are forced to use only written material--computer-based text has a similar potential for causing difficulty among poor readers and most significantly the visually impaired. Hence the idea of incorporating computer-generated voice in all types of software's has revolutionised people's lives and there is way more to go.

IBM's Writing to Read, most idea of the talking software produced for the education market prior to 1986 had a special-education focus. In the last few years, however, most speech-based instructional programs target general education. The three most common ways of adding speech to educational software are compressed, digitized human speech, linear predictive coding (LPC); and text-to-speech. The text-to-speech gives access to putting in lesser efforts in everyday chores but for the visually impaired people it can be a great tool to lead a more normal life.

II. RELATED WORK

Itunuoluwa Isewon, Jelili Oyelade and Olufunke Oladipupo from Covenant University, Nigeria have proposed in their paper "Design and Implementation of Text To Speech Conversion for Visually Impaired People" [1] the idea of text to speech synthesis. In their work we see that their model is divided into these following structures:

- Natural Language Processing (NLP) module: It produces a phonetic transcription of the text read, together with prosody.

- **Digital Signal Processing (DSP) module:** It transforms the symbolic information it receives from NLP into audible and intelligible speech.

The major operations of the NLP module are as follows:

- **Text Analysis:** First the text is segmented into tokens. The token-to-word conversion creates the orthographic form of the token. For the token "Mr" the orthographic form

"Mister" is formed by expansion, the token "12" gets the orthographic form "twelve" and "1997" is transformed to "nineteen ninety seven".

- **Application of Pronunciation Rules:** After the text analysis has been completed, pronunciation rules can be applied. Letters cannot be transformed 1:1 into phonemes because correspondence is not always parallel. In certain environments, a single letter can correspond to either no phoneme (for example, "h" in "caught") or several phoneme ("m" in "Maximum"). In addition, several letters can correspond to a single phoneme ("ch" in "rich"). There are two strategies to determine pronunciation:

- In dictionary-based solution with morphological components, as many morphemes (words) as possible are stored in a dictionary. Full forms are generated by means of inflection, derivation and composition rules. Alternatively, a full form dictionary is used in which all possible word forms are stored. Pronunciation rules determine the pronunciation of words not found in the dictionary.

- In a rule based solution, pronunciation rules are generated from the phonological knowledge of dictionaries. Only words whose pronunciation is a complete exception are included in the dictionary.

The two applications differ significantly in the size of their dictionaries. The dictionary-based solution is many times larger than the rules-based solution's dictionary of exception. However, dictionary-based solutions can be more exact than rule-based solution if they have a large enough phonetic dictionary available.

- **Prosody Generation:** after the pronunciation has been determined, the prosody is generated. The degree of naturalness of a TTS system is dependent on prosodic factors like intonation modelling (phrasing and accentuation), amplitude modelling and duration modelling (including the duration of sound and the duration of pauses, which determines the length of the syllable and the tempos of the speech).[2]

The output of the NLP module is passed to the DSP module. This is where the actual synthesis of the speech signal happens. In concatenative synthesis the selection and linking of speech segments take place. For individual sounds the best option (where several appropriate options are available) are selected from a database and concatenated.

III. PROPOSED WORK

Speech synthesis can be described as artificial production of human speech and Text-to-speech synthesizer (TTS) is the technology which lets computer speak to you.

The text-to-speech (TTS) synthesis procedure consists of two main phases which is shown in "Fig. 1". The first is text analysis, where the input text is transcribed into a phonetic or some other linguistic representation, and the second one is the generation of speech waveforms.

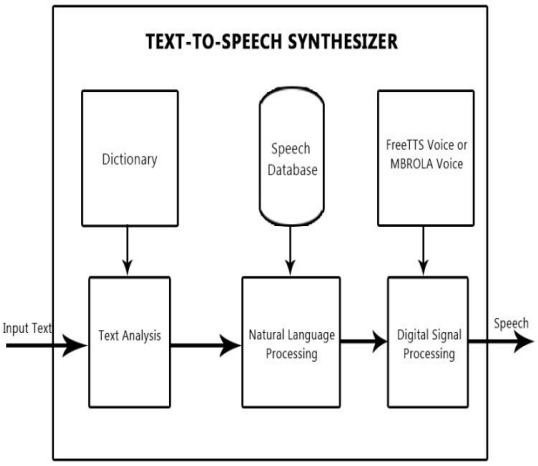

Fig 1: Process Diagram

In the first phase of the software the raw input text is entered by the user or the text/document file is imported to the software which goes under the text analysis. The text analysis is nothing but the process in which it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words using the english dictionary words. This process is often called text normalization, pre-processing, or tokenization.

The second phase of the software can be sub-divided into two parts. The first part of the second phase is for the Natural Language Processing (NLP). The Natural Language Processing produces a phonetic transcription of the text read, together with prosody where the speech database is referred for the processing of words in a correct way. The other part is for the Digital Signal Processing (DSP). The Digital Signal Processing transforms the symbolic information it receives from NLP into audible and intelligible speech. For the digital speech of information FreeTTS and Mbrola voices are used, which are the API for the producing the voice for text-to-speech.

These phases can also be divided on the two parts as Front-end and Back-end. The Front-end has the two major task. First for the text analysis and secondly the natural language processing. On the other hand the back-end part often referred to as the synthesizer—that converts the symbolic linguistic representation into sound. Which is also referred as the digital signal processing.

There are different ways to perform speech synthesis. The choice depends on the task they are used for, but the most widely used method is Concatentive Synthesis, because it generally produces the most natural-sounding synthesized speech. Concatenative synthesis is based on the concatenation (or stringing together) of segments of recorded speech. There are three major sub-types of concatenative synthesis [3] :

Domain-specific Synthesis: Domain-specific synthesis concatenates pre-recorded words and phrases to create complete utterances. It is used in applications where the variety of texts the system will output is limited to a particular domain, like transit schedule announcements or weather reports. [4] The technology is very simple to implement, and has been in commercial use for a long time, in devices like talking clocks and calculators. The level of naturalness of these systems can be very high because the variety of sentence types is limited, and they closely match the prosody and intonation of the original recordings. Because these systems are limited by the words and phrases in their databases, they are not general-purpose and can only synthesize the combinations of words and phrases with which they have been pre-programmed. The blending of words within naturally spoken language however can still cause problems unless many variations are taken into account.This alternation cannot be reproduced by a simple word-concatenation system, which would require additional complexity to be context-sensitive. This involves recording the voice of a person speaking the desired words and phrases. This is useful in limited ways like checking the train status, weather reports etc.

Unit Selection Synthesis: Unit selection synthesis uses large databases of recorded speech. During database creation, each recorded utterance is segmented into some or all of the following: individual phones, diphones, half-phones, syllables, morphemes, words, phrases, and sentences. Typically, the division into segments is done using a specially modified speech recognizer set to a "forced alignment" mode with some manual correction afterward, using visual representations such as the waveform and spectrogram. [5]. An index of the units in the speech database is then created based on the segmentation and acoustic parameters. At runtime, the desired target utterance is created by determining the best chain of candidate units from the database (unit selection). This process is typically achieved using a specially weighted decision tree.

Unit selection provides the greatest naturalness. DSP often makes recorded speech sound less natural, although some systems use a small amount of signal processing at the point of concatenation to smooth the waveform. The output from the best unit-selection systems is often indistinguishable from real human voices, especially in contexts for which the TTS system has been tuned.

Diphone Synthesis: Diphone synthesis uses a minimal speech database containing all the diphones (sound-to-sound transitions) occurring in a language. At runtime, the target prosody of a sentence is superimposed on these minimal units by means of digital signal processing techniques such as linear predictive coding, PSOLA[6] or MBROLA. [7] The quality of the resulting speech is generally worse than that of unit-selection systems, but more natural-sounding than the output of formant synthesizers. Diphone synthesis suffers from the sonic glitches of concatenative synthesis and the robotic-sounding nature of formant synthesis, and has few of the advantages of either approach other than small size.

A. Analysis

- **Identification of the need:**

Language technologies can provide solutions in the form natural interfaces so that digital content can reach to the masses and facilitate the exchange of information across different kind of people. But for the blind people the scenario is not the same. As we know that blind people use braille for studying and reading books and it is a very efficient way to learn but what if the person is not good at it, what if he/she does not know braille and there can be many other possibilities. This traditional system can be even more efficiently solved when the computer technology will contribute towards the blind people. This made us to select this topic for the project to do something for the visually impaired people. There cannot be anything better than the use of text synthesis in this field. As this will not only help then to read the books which are only in braille but this will make them read every e-books in the world and nowadays every single book is available as a soft copy. So nothing can be much better than doing something for the disabled people in our society.

- **Preliminary Investigation:**

In current scenario there are plenty of such softwares already existing. But most of them are complicated and are not free to public. Now a days there are lots of screen reader applications which

can read the screen content like a website by the help of browsers. This is a very interesting thing and for further update can be integrated with the existing application.

B. *Design*

Our software is called the Text Reader for Blind, a simple application with the text to speech functionality. The system is developed using Java programming language. Java is used because it's robust and independent platform. The application is divided into two main modules - the main application module which includes the basic GUI components which handles the basic operations of the application such as input of parameters for conversion either via file or direct keyboard input. The second module, the main conversion engine which integrated into the main module is for the acceptance of data hence the conversion. This would implement the API called FreeTTS. Text Reader for Blind converts text to speech either by typing the text into the text area provided or by importing an external document file to the text area from the local machine. Text Reader for Blind is capable of reading *.txt, *.doc, *.docx document file as an external source. This can be achieved by the user importing the document file by clicking on the Open button in the application and choose the desired file using the JFileChooser from the local machine. By pressing the Speak button the textual data will be converted into the speech of the corresponding text. Not only this, there are 4 different voices of different accents to select from and the user can select which ever voice they are comfortable with. Text Reader for Blind contains an exceptional function that gives the user the choice of saving its already converted text to any part of the local machine in an audio format; this allows the user to copy the audio format to any of his/her audio devices and use them for later use. The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood. Thus by this we can convert any e-book into an audio book which is a very great feature.

I. Implementation

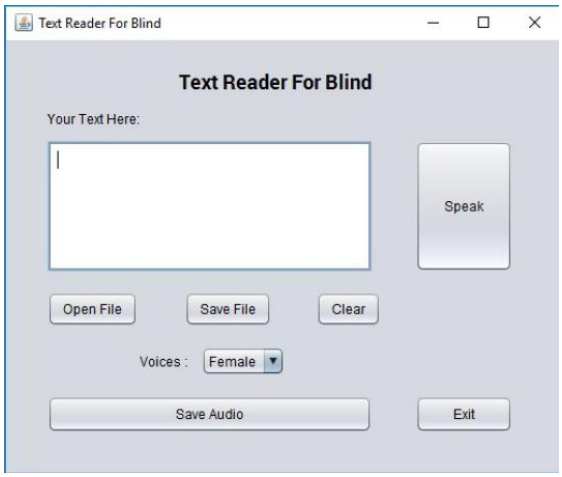

Fig. 2: Text Reader for Blind

The above figure "Fig. 2" shows the UI design of the software where there are several buttons to perform tasks. As it is seen that the UI is very simple, user friendly and can be used in a very easy way. It is made friendly so that the every person can use it. It can be even be used by the visually impaird people. The working of the software is also very simple, just the user need to type some text in the JTextArea and click Speak button and the software will do the rest of the work. When the Speak is clicked the text content will be converted to the speech and will be audible. The Open File button is used to import the external document file into the software which can be made into speech. The Save File button and Save Audio button can save the text content into *.txt format and save the speech into *.wav format respectively at desired location with desired name.

IV. RESULTS AND DISCUSSIONS

In "fig. 3" it shows the screenshot of the list of voices accent functionality. It contains 4 different voices namely Female, Male 1, Male 2, Male 3. These voices are of FreeTTS which are the external jar files and are included as the external library in the project. The voice names are mbrola_us1, mbrola_us2, mbrola_us3, kevin 16.

The reason of adding multiple voice functionality is to make sure that the people can choose the accent based on their comfortability level.

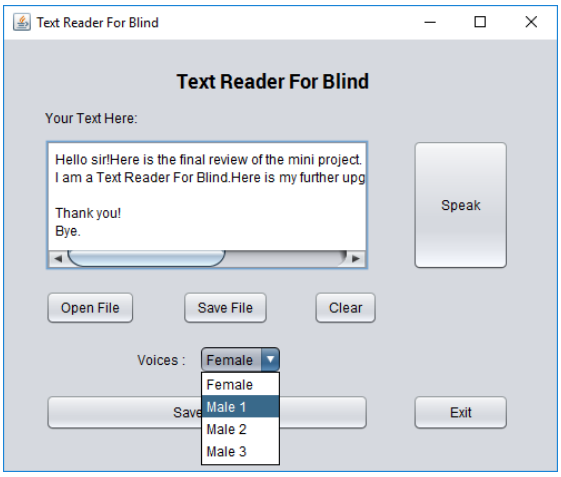

Fig. 3: Multiple voice functionality

In “fig. 4” it shows the screenshot while browsing external document file. With the addition of this feature people can easily browse and import the desired file into the software. This will help visually impaired people listen to their favourite novels, books etc.

The formats of document which can be imported to the software are *.doc, *.docx, *.txt. It uses JFileChooser to navigate the user to the desired location to select the document file.

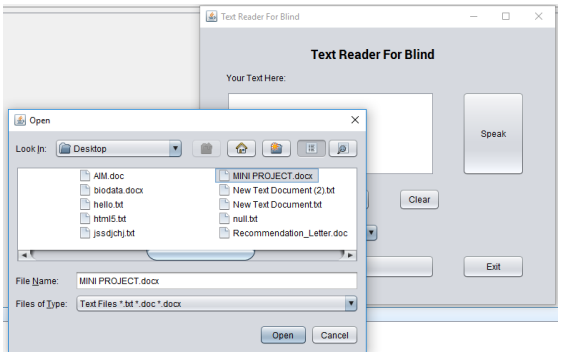

Fig. 4: Browsing file functionality

In “fig. 5” it shows the screenshot while saving the text which was directly used for speech synthesis to an external location. The text will be saved to the *.txt format to any desired location in the computer.

This feature assures that the data which was directly typed in the given space is safe and can be used for future reference.

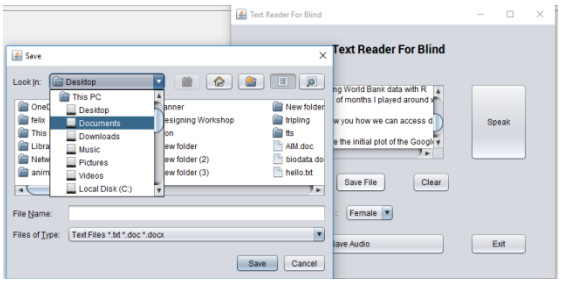

Fig. 5: Saving text functionality

In “fig. 6” it shows the screenshot while saving the text-to-speech converted audio to an external location in the computer. This feature is used to save the speech of the textual data into an audio file to the desired location with desired name in *.wav format. This also opens the door for the user to copy the audio format to any of his/her audio devices and use them for later use. Thus by this we can convert any e-book into an audio book which is a very great feature.

This is a very exceptional feature of the software to save the text into audio file for future use.

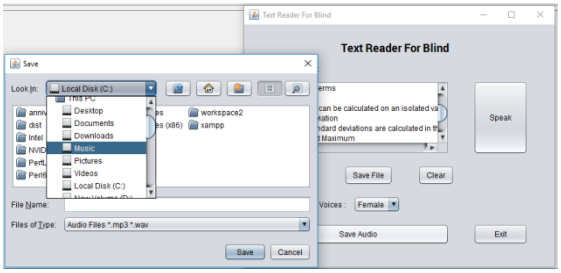

Fig. 6: Saving audio functionality

V. CONCLUSION

Text to speech synthesis is a rapidly growing aspect of computer technology and is increasingly playing a more important role in the way we interact with the system and interfaces across a variety of platforms. We have identified the various operations and processes involved in text to speech synthesis. We have also developed a very simple and attractive graphical user interface which allows the user to type in his/her text provided in the text field or import external document in the application. Our system interfaces with a text to speech engine developed for English which can also save the audio to an external audio file which can be used anytime, anywhere without the software. In future, we plan to make efforts to create engines for regional language so as to make text to speech technology more accessible to a

wider range of people. We have also planned to add more features in the software to pause-resume, recorder, voice command, manipulating the tempo of the voice and much more. Another area of further work is the implementation of a text to speech system on other platforms, such as telephony systems, ATM machines, video games and any other platforms where text to speech technology would be an added advantage and increase functionality.

REFERENCES

[1] Itunuoluwa Isewon, Jelili Oyelade and Olufunke Oladipupo, 2014. "Design and Implementation of Text To Speech Conversion for Visually Impaired People". Department of Computer and Information Sciences, Covenant University, Nigeria.

[2] Text-to-speech technology: In Linguatec Language Technology Website. Retrieved February 21, 2014, from http://www.linguatec.net/products/tts/information/technology.

[3] Wasala, A., Weerasinghe R. , and Gamage, K., 2006, Sinhala Grapheme-to-Phoneme Conversion and Rules for Schwaepenthesis. Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, Sydney, Australia, pp. 890-897.

[4] Lamel, L.F., Gauvain, J.L., Prouts, B., Bouhier, C., and Boesch, R., 1993. Generation and Synthesis of Broadcast Messages, Proceedings ESCA-NATO Workshop and Applications of Speech Technology.

[5] Black, A.W., 2002. Perfect synthesis for all of the people all of the time. IEEE TTS Workshop.

[6] Jeyanthi V., V.Maria Anu "Secured Data Storage Design using Cryptography" ARPN Journal of Engineering and Applied Sciences ISSN: 1819-6608 in Vol 10,No 14 August2015, indexed in SCOPUS.

[7] Kominek, J., and Black, A.W., 2003. CMU ARCTIC databases for speech synthesis. CMU-LTI-03-177. Language Technologies Institute, School of Computer Science, Carnegie Mellon University.

[8] Language Translator and Platform Comparator,is Presented in Springer International conference on Communication, Cloud and Big Data (CCB2016),Organized by Department of Information Technology,Sikkim Manipal Institute of Technology, Sikkim manipal Institute on November 2016,and Proceedings will be Published in Springer Lecture Notes on Networks & Systems(Indexed by EI and SCOPUS).