International Journal of Electrical and Electronics Engineers
Vol. No. 9, Issue No. 01, January-June 2017
ISSN (O) 2321-2055
ISSN (P) 2321-2045

ADVANCED SPEECH SYNTHESIZER OVER LINUX OPERATING SYSTEM AND ARM PROCESSOR- A REVIEW

Meenal M. Garad[1], Sunita P. Aware[2]

Error! Bookmark not defined.Error! Bookmark not defined.*Dept of E&TC, M.S.S. College of Engineering & Technology, Jalna, (India).*

ABSTRACT

Nowadays, listening to audio books on mobile devices is quite common. Therefore, in the future we will obtain ever-increasing amounts of information through speech instead of conventional printed materials. So instead of Reading the books we can hear the speech in the form of the headphones or in the form of the speakers and also whatever the User will Write it will be read in the form of the speech. In this project we have used Open source Speech Synthesis Algorithm for converting the files from Text to Speech So People read books at various levels of detail from close reading to skimming. By using this concept whatever we entered in the Text file will be converted in the form of the Speech in the form of the computer generated voice. This opens up a lot of opportunities in the future by their own generated voice.

Key words- TTS, Festival, Speech synthesis, Raspberry

I. INTRODUCTION

Speech is one of the most vital forms of communication in our everyday life. So it is natural for people to expect to be able to carry out spoken dialogue with computers. This involves the integration of speech technology and language technology. A freely available (and hopefully open-source) TTS system for English language can greatly aid the human computer interaction: the possibilities are endless – such a system can help overcome the literacy barrier of the common masses, empower the visually impaired population, increase the possibilities of improved man-machine interaction through on-line newspaper reading from the internet and enhancing other information systems. Festival[1] is a complete TTS synthesis system, with language modeling, and speech synthesis engine. The language model supports all language processing tasks. For example document analysis, text analysis, and phonological processing. We used festival to develop Text to Speech for English language by providing language processing parameter in language model part and recorded speech in speech engine. Here we described the methodology and the implementation of a Text to Speech system for English based on the Festival TTS engine. At the end of the paper, assessment results of the TTS system are given and some promising directions for future work are mentioned. Organization of the paper are as follows. Section 2 discusses about related works. Section 3 discusses about implementation of system. Section4 discusses

methodology. Then section 5 discusses about implementation result. After that in section 6 we discusses advantages application and in section 7 we discusses conclusion.

II. SYSTEM ARCHITECTURE

2.1 Related work

Following are the related works for the English Text To Speech. Several attempts were made in the past, where different aspects of a English TTS system were covered [2][3][4][5]. In [2] authors described about different modules (optimal text selection, G2P conversion, automatic segmentation tools) in detail and experiment results of the different module have shown. In [3] significant amount of work done for developing English TTS. Phoneme and partname (similar to diphone) are used to develop voice database and ESOLA technique used for concatenation. But quty may suffer for lack of smoothness. In [4] authors showed some practical applications with English TTS system using ESNOLA technique. But performance of the output not described. In [5] author showed the pronunciation rule and phoneme to speech synthesizer using formant synthesis technique. None of them have shown the naturalness and intelligibility of the system. This work is done with multi syncronize unit selection and unit selection technique within festival framework and performance of the intelligibility and naturalness of the system has shown.

2. 2 System Algorithm

In Linux environment any application is able to run at a quite faster rate. Linux based raspberry pi module as shown in Figure 1 will be very efficient to overcome the requirement. There are many Linux based OS and among them Raspbian is sufficient for my research work and has some advantage to use this. Raspbian is user-friendly. For a beginner it is an absolute test. It is both official and user contributed. It has wealth of documentation and also has many more features. The software is used to design a speech browser is QT creator which is used to make efficient GUI application. Qt Creator is a good example of an application t hat mixes different user interface technologies. In fact, it uses all of the three different approaches described below. Qt Creator uses the traditional Qt Widgets such as menus and dialogs as a basis of the user interface, Qt Quick amongst others for the welcome screen, and Qt WebKit for presenting the Qt reference documentation. Qt Creator includes a project manager that uses a cross platform project file format (.pro). A project file can contain information such as what files are included into the project, custom build steps and settings for running the applications. Qt Creator includes a code editor and integrates Qt Designer for designing and building graphical user interfaces (GUIs) from Qt widgets. The code editor can parse code in C++ and QML languages... It is possible to compose and customize the widgets or dialogs and test those using different styles and resolutions directly in the editor. Widgets and forms created with Qt Designer are integrated with programmed code, using the Qt signals and slots mechanism.

III.WORKING

3.1 HARDWARE: RASPBERRY Pi (ARM 11)

The Raspberry Pi is a credit-card sized computer that plugs into your TV and keyboard. It is a capable for little projects, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games. It

International Journal of Electrical and Electronics Engineers
Vol. No. 9, Issue No. 01, January-June 2017
ISSN (O) 2321-2055
ISSN (P) 2321-2045

also plays high-definition videos. We want to see it being used by kids all over the world to learn how computers work, how to manipulate the electronic world around them and, how to program. The original Raspberry Pi is based on the Broadcom BCM2835 system on a chip (SoC), which includes, Video Core IV GPU, RAM of 512 MB. The system has Secure Digital (SD) socket for boot media and persistent storage. A SoC consists of the hardware, described above, and the software controlling the microcontroller, microprocessor or DSP cores, peripherals and interfaces. The design flow for Soc aims to develop this hardware and software in parallel

Fig 1: Raspberry Pi Board

3.1 SOFTWARE

The software is used to design and developed is QT creator which is used to make efficient GUI application. Qt Creator is a good example of an application that mixes different user interface technologies. In fact, it uses all of the three different approaches described below. Qt Creator uses the traditional Qt Widgets such as menus and dialogs as a basis of the user interface, Qt Quick amongst others for the welcome screen, and Qt WebKit for presenting the Qt reference documentation. Qt Creator includes a project manager that uses a cross platform project file format (.pro). A project file can contain information such as what files are included into the project, custom build steps and settings for running the applications. Qt Creator includes a code editor and integrates Qt Designer for designing and building graphical user interfaces (GUIs) from Qt widgets. The code editor can parse code in C++ and QML languages... It is possible to compose and customize the widgets or dialogs and test those using different styles and resolutions directly in the editor. Widgets and forms created with Qt Designer are integrated with programmed code, using the Qt signals and slots mechanism.

3.1.1Raspbian Operating System:

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.The initial build of over 35,000 Raspbian packages, optimized for best performance on the Raspberry Pi, was completed in June of 2012. However, Raspbian is still under active development with an emphasis on improving the stability and performance of as many Debian packages as

possible. The Raspberry Pi primarily uses Linux kernel-based operating systems Raspbian (recommended) – Maintained independently of the Foundation; based on ARM hard-float (armhf)-Debian 7 'Wheezy' architecture port, that was designed for a newer ARMv7 processor (or one with Jazelle RCT/ThumbEE, VFPv3 and NEON SIMD extensions built-in) whose binaries would not work on the Rapberry Pi, but Raspbian is compiled for the ARMv6 instruction set of the Raspberry Pi making it work but run more slowly. It provides some available deb software packages, pre-compiled software bundles. A minimum size of 2 GB SD card is required, but a 4 GB SD card or above is recommended. There is a Pi Store for exchanging programs. The Raspbian Server Edition is a stripped version with other software packages bundled as compared to the usual desktop computer oriented Raspbian and Reversal protocol.

IV METHODOLOGY

The TTS for English is developed by widely used third party tool Festival. The different phases of the synthesis task are performed by several modules as shown in Figure 2. The text analysis part converts all non standard words to standard words. A grapheme- to-phoneme module produces strings of phonemic symbols based on information in the written text. The problems it addresses are thus typically language dependent. So is the prosodic generator, which assigns pitch and duration values to individual phonemes. Final speech synthesis is performed by concatenative unit selection technique and multisyncronize unit selection technique. We implemented all of modules by festival tools.

4.1 Text analysis

The first step of Text to Speech system is text analysis [6] that means analysis of raw text into pronounceable word. It involves the work on the real text, where many Non-Standard Word (NSW) [7] representations appear, for e.g., numbers (year, time, ordinal, cardinal, floating point), abbreviations, acronyms, currency, dates, URLs. All of these non-standard representations should normalize, or in other words convert to standard words. These NSW should normalize using text normalization and ambiguous token should disambiguate using rules.

4.1.1 Text analysis part in Festival

Festival does not support Unicode directly, so in the first step we transliterated our Unicode text to ASCII code according English phone set [8]. The transliteration table is given in table-1. In our system of text analysis parts we worked on standard words. We identified more than 10 types of NSW in English Language, which in not implemented yet. Some example of NSW in English Language is given in table 2 that can be Implemented in future. Now our system only supports Unicode, not ASCII coded English text.

4.1.2 Steps of Text Analysis in Festival

Step 1: Split the token: We can split our token based on white-space and punctuation.

- White-space can be viewed as separators.
- Punctuation can separate the raw tokens.
- Festival converts text into: Ordered list of tokens, each with features of white-space, and punctuation.

White-space is the most commonly used delimiter between words and is extensively used for tokenization. But using white-space as the only delimiter have some limitation: a token type which allows the occurrence of

International Journal of Electrical and Electronics Engineers
Vol. No. 9, Issue No. 01, January-June 2017
ISSN (O) 2321-2055
ISSN (P) 2321-2045

white-space within the token will not recognize as a single token, but split up into two or more tokens.

Step 2: Type identifier: As we explained English Language have more than 10 types of NSW, so each NSW can identify as separate token by token identifier rules. To identify the token we can use scheme regular expression in festival, which is not implemented yet. There is also an ambiguity in abbreviation, and number in English language.

Step 3: Token expander: After identification of all NSW we can convert these to standard word by pronunciation lexicon or (letter to sound) LTS rule.

4.2 Text analysis

The second step of TTS system is to convert the text to its pronunciation form. For finding pronunciation of a word we need large list of lexicon and LTS rule.

Steps of Phonetic Analysis within festival:

- Building large amount of lexicon.
- Building letter-to-sound rules.

4.2.1 Building large amount of lexicon by hand: We included 900 lexicons by scheme programming. Developing LTS rule for a language is too much difficult and much more computation is needed in run time of TTS. So lexicon dictionary is important in TTS system. We implemented our pronunciation lexicon by scheme within festival. The basic assumption in festival is that we have a large set of lexicon that is a used as a standard part of an implementation of a voice. A pronunciation in festival requires not just a list of phones but also a syllabic structure. The lexicon structure that is basically available in festival takes word, part of speech (and arbitrary token) and stress marker to find the given pronunciation of a given word. We implemented our large set of lexicon based on English syllabic structure.

4.2.2 Building letter-to-sound rules: English language always borrows words from other languages like computer. To find pronunciation of new arrival words that is not found in the lexicon we have to use LTS rule. In festival there is a letter to sound rule or Grapheme to Phoneme (G2P) system that allows rules to be written, but festival also provided a method for building rule sets automatically, which will often be more useful. An explicit lexicon isn't necessary in festival and it may be possible to do much of the work in letter- to-sound rules. But in this case we have to identify proper LTS rule and we have to consider performance issue because it may take lots of computation. We used some of the LTS rule in our implementation based on our syllabification rule.

4.3 Speech Database / Waveform Synthesis

This is one of the major parts in TTS. The general-purpose concatenative synthesis [10][11] translates incoming text onto phoneme labels, stress and emphasis tags, and phrase break tags. This information is used to compute a target prosodic pattern (i.e., phoneme durations and pitch contour) . Finally, signal processing methods retrieve acoustic units (fragments of speech corresponding to short phoneme sequences such as diphones) from a stored inventory, modify the units so that they match the target prosody, and smooth (concatenate) them together to form an output utterance. Concatenative synthesis techniques give the most natural sound in speech synthesis. Three techniques are available in concatenative synthesis: diphone, unit selection and multisyn-cronize unit selection. Diphone based speech synthesis systems can produce very intelligible synthetic speech,

International Journal of Electrical and Electronics Engineers
Vol. No. 9, Issue No. 01, January-June 2017
ISSN (O) 2321-2055
ISSN (P) 2321-2045

but less natural than unit selection technique. Unit selection [12] database can be created by automatically clustering units of the same phone class based on their phonetic features and prosodic context. The appropriate cluster is then selected for a target unit offering a small set of candidate units. We used unit selection and multisyn cronize unit selection technique [13] for waveform synthesis. To implement speech database using festival at first we have to identify all the features of the phonemes and total number of phones. It can be done by articulatory technique or acoustic technique. Acoustic technique is the best way to identify all the phoneme of a language. phones excluding 31 diphthongs with their features [14] based on articulatory analysis. To build diphone database we have to include diphthong as well. In our implementation we excluded the diphthongs. As we explained earlier we added lexicon for pronunciation of words. Also duration of the each phone is added to implement the TTS for English. The duration we added is taken from Kiswahili [15] TTS system. This is not exact duration for the phone set of English language. Using acoustic analysis procedure we can measure exact duration of the phone set.

V. IMPLEMENTATION

The Implementation process for converting speech to text by using Raspberry pi and on the terminal of Raspbian image installed on raspberry pi. This is the basic step to understand that how can we convert any text data into voice and this will be helpful to understand the commands of linux. The Raspberry Pi is an ultra-low-cost, deck-of-cards sized Linux computer. It is controlled by a modified version of Debian Linux optimized for the ARM architecture. It has two models model A and model B. The Model B has 512 MB RAM, BCM2385 ARM11, 700 MHz System on chip processor. It has 2 USB ports, HDMI out, audio output jack and Ethernet port for internet access.

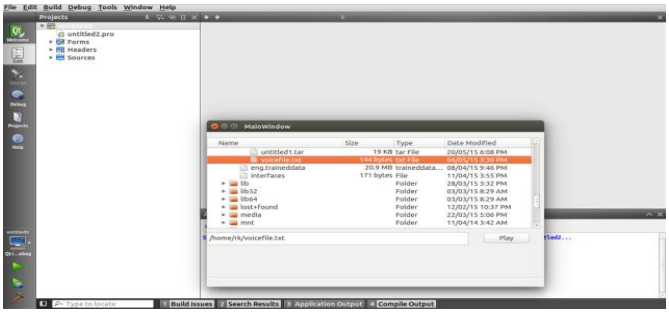

Fig 2: Text to Speech GUI on Qt Creator tool

International Journal of Electrical and Electronics Engineers
Vol. No. 9, Issue No. 01, January-June 2017
ISSN (O) 2321-2055
ISSN (P) 2321-2045

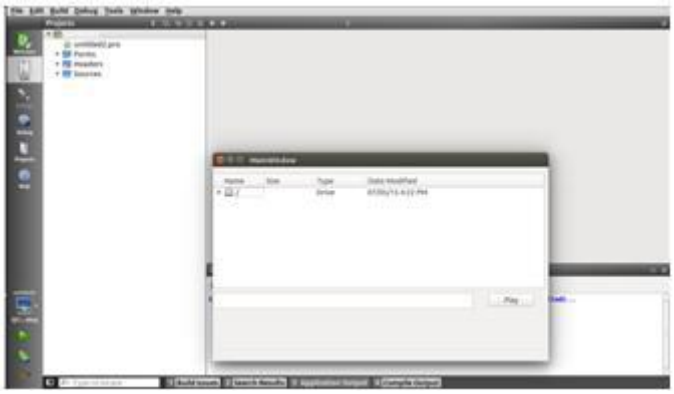

Fig 3: Text File selection for Voice conversion

VI. ADVANTAGES & APPLICATIONS

- Advantage of this project is only hardware it occupies is one GPIO Pin for data acquisition.
- It can be modified & can be applied to other working environments.

Applications

- Reading aid for the visually disabled persons.
- Announcement or warning systems.
- Electronic mail readers.

VII. CONCLUSION

The described speech synthesis system is the open source and freely distributable TTS system for English language. This is the complete process to develop commarcial TTS system which includes most of the complexity of English language. Besides the obvious uses of a TTS system, from listening to computerized books to one's email, it also allows the visually impaired and those who cannot read English access to English electronic content such as the World Wide Web. We have described a proof-of-principle implementation of a English TTS, and there is much work to be done before we have a complete and commercial quty TTS system such as those available for many other languages. We have a plan to continue developing the English festival voice to improve the quty of the synthesized speech. The synthetic speech produced by the system is intelligible, but lacks of naturalness. Improvement of intelligibility and naturalness depend on significant amount of work in each phase.

REFERENCES

[1] Black A., Taylor P., "The Festival Speech Synthesis System", Technical Report HCRC/TR-83, University of Edinburgh, Scotland, (1997), http://www.cstr.ed.ac.uk/projects/festival.html.

[2] Tanuja Sarkar, Venkatesh Keri,. Santhosh M and Kishore Prahallad, "Building English Voice Using Festvox", ICLSI 2005

[3] Asok Bandyopadhyay, "Some Important Aspects of English Speech Synthesis System" IEMCT Pune, June 24-25 2002.

[4]Shyamal Kr. DasMandal, Barn Pal "English text to speech synthesis system a novel approach for crossing literacy barrier" . CSI-YITPA(E)2002.

[5]Aniruddha Sen, "English Pronunciation Rules and a Text-to-Speech System", Symposium on Indian Morphology, Phonology & Language Engineering, 2004, pp. 39.