Real Time Voice Cloning

Sakith Nalluri[1] A.Rohan Sai[1], M.Saraswati[3]

[1]*B. Tech (CSE), UG Scholar, Department of CSE, Sri Chandrasekarendra Saraswathi Viswa Maha Vidyalaya, Kanchipuram*

[2]*B. Tech (CSE), Department of CSE, Sri Chandrasekarendra Saraswathi Viswa Maha Vidyalaya, Kanchipuram*

[3]*Associate Professor, Department of CSE, Sri Chandrasekarendra Saraswathi Viswa Maha Vidyalaya, Kanchipuram*

***Abstract* - Recent progress in deep learning has shown impressive results in the area of speech-to-text. For this reason, a deep neural network is usually trained from a single speaker using a corpus of several hours of voice recorded professionally. Giving such a model a new voice is highly expensive, as it needs a new dataset to be collected and the model retrained. A recent research has developed a three-stage pipeline that allows you to clone an unseen voice from just a few seconds of reference speech during practice and without retraining the template. The researchers share strikingly natural-sounding findings. A Text-to-speech synthesizer is an application that converts text into spoken word, 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. Here, we developed a useful text-to-speech synthesizer in the form of a simple application that converts inputted text into synthesized speech and reads out to the user which can then be saved as an mp3. file. The development of a text to speech synthesizer will be of great help to people with visual impairment and make making through large volume of text easier.**

***Index Terms* - Text-to-speech synthesis, Natural Language Processing, Digital Signal Processing.**

I.INTRODUCTION

In many fields of computational machine learning, deep learning models have become prevalent. The method of synthesizing artificial speech from a text prompt, Text-to-speech (TTS), is no exception. Deep models that would create more natural-sounding speech than the conventional concatenative methods begun emerging in 2016 Low-dimensional embedding is derived from a speaker encoder model which takes reference speech as input. Typically, this approach is more data-efficient than training a separate TTS model for each speaker, as well as faster and less computationally expensive orders of magnitude. Interestingly, there is a broad disparity between the length of reference speech necessary to clone a voice. Text-to-speech synthesis -TTS - is the automatic conversion of a text into speech that resembles, as closely as possible, a native speaker of the language reading that text. Text-to speech synthesizer (TTS) is the technology which lets computer speak to you. The TTS system gets the text as the input and then a computer algorithm which called TTS engine analyses the text, pre-processes the text and synthesizes the speech with some mathematical models. The TTS engine usually generates sound data in an audio format as the output. The text-to-speech (TTS) synthesis procedure consists of two main phases. 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, where the output is produced from this phonetic and prosodic information. These two phases are usually called high and low-level synthesis [1]. A simplified version of this procedure is presented in project. The input text might be for example data from a word processor, standard ASCII from e-mail, a mobile text-message, or scanned text from a newspaper. The character string is then pre-processed and analyzed into phonetic representation which is usually a string of phonemes with some additional information for correct intonation, duration, and stress. Speech sound is finally generated with the low-level synthesizer by the information from high-level one. The artificial production of speech-like sounds has a long history, with documented mechanical attempts dating to the eighteenth century.

II. LITERATURER SURVEY

Speech synthesis can be described as artificial production of human speech [3]. A computer system used for this purpose is called a speech synthesizer and can be implemented in software or hardware. A text-to-speech (TTS) system converts normal language text into speech [4]. Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output [5]. The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer. A text-to-speech system (or "engine") is composed of two parts: [6] a front-end and a back end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, preprocessing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the synthesizer—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), [7] which is then imposed on the output speech. 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 Concatenative 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 [8]: 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. [9] 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. For example, in nonrhotic dialects of English the "r" in words like "clear" /ˈklɪə/ is usually only pronounced when the following word has a vowel as its first letter (e.g., "clear out" is realized as /ˌklɪərˈʌʊt/) [10]. Likewise in French, many final consonants become no longer silent if followed by a word that begins with a vowel, an effect called liaison. 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 if only the restricted volume of phrases and sentences is used and the variety of texts the system will output is limited to a particular domain e.g. a message in a train station, whether reports or checking a telephone subscriber’s account balance. 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. [11]. An index of the units in the speech database is then created based on the segmentation and acoustic parameters like the fundamental frequency (pitch), duration, position in the syllable, and neighboring phones. 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, because it applies only a small a90-mount of digital signals processing (DSP) to the recorded speech. 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. However, maximum naturalness typically requires unit selection speech databases to be very large, in some systems ranging into the gigabytes of recorded data, representing dozens of hours of speech. [12]. Also, unit selection algorithms have been known to select segments from a place that results in less-than-ideal synthesis (e.g., minor words become unclear) even when a better choice exists in the database. [13].

Diphone Synthesis: Diphone synthesis uses a minimal speech database containing all the diphones (sound-to-sound transitions) occurring in a language. The number of diphones depends on the phonotactics of the language: for example, Spanish has about 800 diphones, and German about 2500. In diphone synthesis, only one example of each diphone is contained in the speech database. 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 [12] or MBROLA. [14] 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. As such, its use in commercial applications is declining, although it continues to be used in research because there are a number of freely available software implementations [15].

III. STRUCTURE OF A TEXT-TO-SPEECH

Synthesizer System

Text-to-speech synthesis takes place in several steps. The TTS systems get a text as input, which it first must analyze and then transform into a phonetic description. Then in a further step it generates the prosody. From the information now available, it can produce a speech signal. The structure of the text-to-speech synthesizer can be broken down into major modules: • 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 phonemes ("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) [16] 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.

IV. PROPOSED WORK

Our software is called the TextToSpeech Robot, a simple application with the text to speech functionality. The system was developed using Java programming language. Java is used because it is 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 or the browser. This would make use of the open-source API called c#. 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 free TTS.

V METHODOLOGY

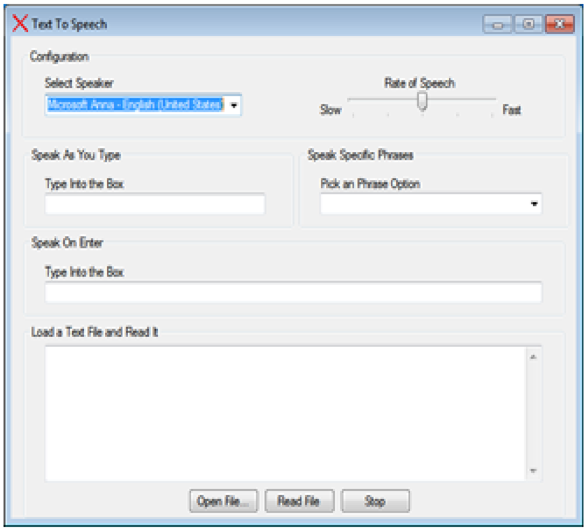
This article and code sample was intended to provide a very easy introduction into TTS based speech synthesis; there are a great many more things that you can do with the speech SDK than have been addressed in this document. A review of the contents of the speech SDK will provide greater details on the use of the speech libraries.

TextToSpeech Robot (TTSR) converts text to speech either by typing the text into the text field provided or by coping from an external document in the local machine and then pasting it in the text field provided in the application. It also provides a functionality that allows the user browse the World Wide Web (www) on the application. TextToSpeech Robot is capable of reading any portion of the web page the user browses. This can be achieved by the user highlighting the portion he wants to be read out loud by the TTSR and then clicking on the "Play" button. TTSR 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.

Configuration.

This control group contains two controls, the speaker combo box, and the speech rate track bar control. The speaker combo box is populated with the names of each of the TTS speaker voices, you may change the current speaker by selecting a different option form this combo box. The rate track bar control will speed up or reduce the cadence of the synthesized speech. It is set to contain five positions and whenever its value is changed, the rate of speech will be altered to execute at the newly set rate.

Text to Speech Application



Select Speaker: Microsoft Anna - English (United States)

Speak as You Type.

This control group contains a single text box which has been configured such that, whenever the user hits the space bar, the speaker will read the contents of the text box and once finished reading, it will clear the text box. The intent here was to see if you could type as you go and speak through TTS. It seemed like a nice idea and it seems like it would be worthwhile for someone lacking the capacity for speech to use a function like this to speak by typing. In reality, the action is a little choppy and the speech rendered is not too terrific. With the application running, you may key in a word and listen to the results for yourself. If you type slow enough, it is adequate, but it is not quite quick enough to use as a form of conversation.

Speak Specific Phrases.
This control group contains a single combo box; whenever a new value is selected from the box, it will immediately be read by the speaker.

Speak on Enter.
This appears to be a far more viable way to conduct a conversation using TTS as a voice medium. This control works in a manner very similar to the “Speak as You Type” option, however, it reads and clears the text box only after the user hits the “enter” key. You may try typing in a sentence and then hitting the enter key to get a feel for how that works.

Load a Text File and Read It.
This control group contains a single multi-line text box control and three buttons: “Open File”, “Stop”, and “Read File”. Click on the “Open File” button and use the open file dialog box to navigate to any text file. The text file will load into the text box and with a file loaded, you may click on the “Read File” button to have the speaker read the contents of the text box end to end. TTS does a fair job of this however I will point out that punctuation and abbreviations do not work out too well for the c#. You may also key text into the text box and evoke the “Read File” function to read the contents of the text box.

VI.CONCLUSION

The project successfully developed a framework for real-time voice cloning that had no public implementation. The results are found to be satisfying despite some unnatural prosody, and the voice cloning ability of the framework to be reasonably good but not on par with methods that make use of more reference speech time. There is still scoping to improve the given framework beyond the scope of this project, and possibly to implement some of the newer advances in the field that were made at the time of writing. It can be inferred that the aforementioned project and research is one of the latest approaches to audio processing (text-to-speech) and voice cloning using sophisticated deep learning networks and improving previously tried approaches that give us better similarity and naturalness of the generated speech. Therefore, it is found that the approach to be an attempt to understand, implement and innovate using the expertise that has been acquired in researching this venture. We believe that even more powerful forms of voice cloning will become available in a near future.

REFERENCE

[1] Lemmetty, S., 1999. Review of Speech Syn1thesis Technology. Masters Dissertation, Helsinki University of Technology.

[2] Dutoit, T., 1993. High quality text-to-speech synthesis of the French language. Doctoral dissertation, Faculte Polytechnique de Mons.

[3] Suendermann, D., Höge, H., and Black, A., 2010. Challenges in Speech Synthesis. Chen, F., Jokinen, K., (eds.), Speech Technology, Springer Science + Business Media LLC.

[4] Allen, J., Hunnicutt, M. S., Klatt D., 1987. From Text to Speech: The MITalk system. Cambridge University Press.

[5] Rubin, P., Baer, T., and Mermelstein, P., 1981. An articulatory synthesizer for perceptual research. Journal of the Acoustical Society of America 70: 321–328.

[6] van Santen, J.P.H., Sproat, R. W., Olive, J.P., and Hirschberg, J., 1997. Progress in Speech Synthesis. Springer.

[7] van Santen, J.P.H., 1994. Assignment of segmental duration in text-to-speech synthesis. Computer Speech & Language, Volume 8, Issue 2, Pages 95–128

[8] 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.

[9] 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.

[10] van Truc, T., Le Quang, P., van Thuyen, V., Hieu, L.T., Tuan, N.M., and Hung P.D., 2013.

Vietnamese Synthesis System, Capstone Project Document, FPT UNIVERSITY.

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

[12] 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.

[13] Zhang, J., 2004. Language Generation and Speech Synthesis in Dialogues for Language Learning. Masters Dissertation, Massachusetts Institute of Technology.

[14] Dutoit, T., Pagel, V., Pierret, N., Bataille, F., van der Vrecken, O., 1996. The MBROLA Project: Towards a set of high-quality speech synthesizers of use for noncommercial purposes. ICSLP Proceedings.

[15] Text-to-speech (TTS) Overview. In Voice RSS Website. Retrieved February 21, 2014, from http://www.voicerss.org/tts/

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

[17] Dutoit, T., 1997. High-Quality Text-to-Speech Synthesis:An Overview. Journal of Electrical and Electronics Engineering Australia 17, 25-36.

[18] Ngugi, K., Okelo-Odongo, W., and Wagacha, P. W., 2005. Swahili Text-To-Speech System. African Journal of Science and Technology (AJST) Science and Engineering Series Vol. 6, No. 1, pp. 80 – 89.

[19] Mohanan, S., Salkar, S., Naik, G., Dessai, N.B., and Naik, S., 2012. Text to Speech Synthesizer for Konkani Language. International Conference on Computing and Control Engineering (ICCCE 2012), 12 & 13 April, ISBN 978-1-4675-2248-9.

[20] Swathi, G., Mai, C. K., and Babu, B. R., 2013. Speech Synthesis System for Telugu Language. International Journal of Computer Applications (0975 – 8887), Volume 81 – No5.