VOL. 29 NO. 6
ISSN: 2910-2080
JUNE, 2023

African Scholar Publications & Research International

African Scholar Journal of Arts and Sociological Research (JASR-6)

Development of a Text-To-Speech Sythesis System

Oladejo Rachel Adefunke
Ogun State Institute of Technology, Igbesa Ogun State

Abstract

Text-to-speech synthesis or recognition, sometimes known as TTS, is a technique that turns any input text into intelligent and natural-sounding speech. The conversion of text to speech is a very helpful hardware and software technique used in numerous applications, including voice monitoring systems for the blind, online browsers, mobile phones, Desktops, and laptops, among others. It facilitates the automatic transformation of a text into speech that most nearly matches a native speaker of the target language reading that text. The TTS system receives text as input, analyzes it using a computer algorithm known as the TTS engine, pre-processes it, and then uses some mathematical models to synthesize voice. Typically, the output of the TTS engine is sound data in an audio format. There are two primary stages to the text-to-speech (TTS) synthesis process. The first is text analysis, in which the source text is converted into a phonetic or other linguistic representation; the second is speech waveform generation, in which the output is created using this phonetic and prosodic data. Common names for these two stages are high-level synthesis and low-level synthesis.

Keyword: Text-to-Speech, synthesis, text analysis

Introduction

One area of computer technology that is fast evolving and taking on greater significance in how we interact with systems and user interfaces on multiple platforms is text-to-speech synthesis. Language is the ability to express one's thoughts by means of a set of signs (writing), gestures, and noises. Being the only creatures that use such a system, it is a distinguishing trait of humans. Speech is the oldest and most widely used means of communication between people.

Speech synthesis is the term for the synthetic generation of human speech (Pabasara et al, 2019). A voice synthesizer, which can be implemented in hardware or software, is a computer system used for this purpose. A text-to-speech (TTS) system converts normal language text into speech (Wesley & Werner, 2014). There are two basic stages to the text-to-speech (TTS) synthesis process. The first is text analysis, in which the source text is converted into a phonetic or other linguistic representation; the second is speech waveform creation, in which the output is created using this phonetic and prosodic data. Normally, these two stages are referred to as high and low-level synthesis. Figure 1 below shows a condensed version of this process. The input text could be, for instance, scanned text from a newspaper, standard ASCII from email, a mobile text message, or data from a word processor. The character string is then pre-processed and examined to produce a phonetic representation, which is typically a string of phonemes plus some extra information for correct intonation, length, and stress. The information from the high-level synthesizer is ultimately used to make speech sound using the low-level synthesizer.

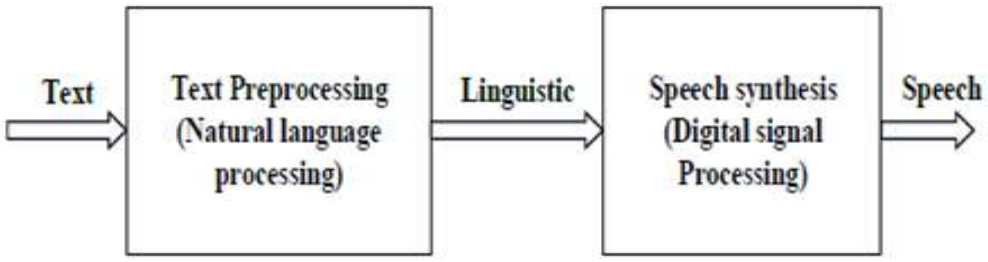

Figure 1: A Typical functional diagram of a Text-To-Speech system.

A database of audio recordings can be combined to create synthesized speech. The size of the speech units stored in different systems varies; a system that stores phonemes or diphones has the greatest output range but may not be as clear. The preservation of complete words or sentences enables high-quality output for particular usage domains. For an entirely "synthetic" voice output, on the other hand, a synthesizer can contain a model of the vocal tract and other aspects of the human voice (Rubin et al, 1981). Its resemblance to the human voice and its readability are used to assess a speech synthesizer's quality. Those who struggle with reading or vision problems can listen to written works on a home computer using an intelligible text-to-speech application.

The VODER system (1932) is the first reputable attempt to synthesis continuous acoustic-only speech sounds using electronics (Dudley et al, 2003).The emphasis in the field of acoustic-only speech synthesis has shifted over time from rule-based articulatory synthesizers (Dunn, 2005; Rosen, 2005) or formant synthesizers ((Fant, 1953; Lawrence, 1953)to data-driven synthesis, which includes diphone synthesizers (Dixon & Maxey, 1968), unit-selection synthesizers (Hunt & Black, 1996), and synthesizers based on statistical prediction (Zen, Tokuda, & Black,, 2009.) Several review articles and books, including those by (Schroeder, 1993; Dutoit, 1997 and (Taylor, 2009), provide a more thorough overview of the topic of acoustic-only speech synthesis.

Speech synthesis is utilized in a wide range of applications. It's crucial to remember that this technology was initially developed to assist people with impairments, particularly those who are visually impaired, in their daily lives. For instance, the well-known Stephen Hawking employed a speech synthesis device to communicate with those around him due to his severe impairment. Since then, numerous uses have emerged that are more or less consistent with TTS's original virtue. For instance, transportation providers employ this technology to speak messages to passengers, disabled or not. It is also noticeable in language translation tools engine. With the aid of speech synthesis technology, the textual translation can be completed by suggesting how the translated information should be pronounced.

The broad field of IoT is another area where voice synthesis is integrated into embedded devices or cloud applications and continues to revolutionize utilization. In fact, TTS is being incorporated by intelligent technologies more and more in our ever-growing cosmos. On the one hand, it makes it

possible to enhance user experience. Yet, it also enhances the interfaces' intelligence and usability. Household appliances (often known as "appliances" in English) are an excellent example of a field that is constantly progressing and increasingly incorporating speech technology into consumer goods and robots.

This paper gives an overview on different approaches and techniques used in related researches in developing a Text-to-Speech applications for different languages. In addition, the paper developed a typical text- to-speech application for English Language which works with an integrated Application Programming interface (API). This gives greater and better comprehensive access to population of those with literacy difficulties, learning disabilities, reduced vision and those learning a language. It also opens doors to anyone else looking for easier ways to access digital content.

RELATED WORKS

Many studies had carried out researches on text-to-speech conversion, however, some of the researches are conversion of text to languages other than English. Examples of such studies are highlighted below.

Lemma and Arabic pattern-based concatenative technique was described by (Oumaima & Abdelouafi, 2017). The use of a collection of sub-segments, where the consonant is regarded as the core of the acoustic unit and is therefore taken with its vocalic context, is also recommended as an alternate way for synthesizing diacritical Arabic texts. A speech corpus design for Arabic Text-To-Speech (ATTS) is also offered. It is based on pre-recorded audiobooks from the Masmoo3 Audiobooks website. The corpus includes more than 4 hours of nonstop Modern Standard Arabic (MSA) speech that was phonetically balanced and recorded with great intelligibility. The Diagnostic Rhyme Test (DRT), which gauges the word-level comprehensibility of the synthesized speech, was used to assess the suggested method. Moreover, the sentence-level test was run. Another study worked on An NLP based text-to-speech synthesizer for Moroccan Arabic was coducted by (Rajae et al, 2017), the purpose of the paper was to present a text-to-speech synthesizer for Moroccan Arabic based on NLP rule-based and probabilistic models. (Georg et al, 2017) presented a rule-based implementation for the 11 official South African languages that uses native

number expansion. It also discusses the architecture and performance of the implementation based on examples of cardinal and ordinal numbers, money, dates and times.

Tomoya et al(2023) Work on the development of neural incremental text-to-speech (iTTS), Prefix-to-prefix neural iTTS framework with look-ahead of 1-2 unit segments are used by the majority of modern state-of-the-art iTTS systems (i.e., phonemes or words). However, since the Japanese language is based on accent phrase units that are longer than words, using a prefix-to-prefix neural iTTS with a look-ahead approach increases latency. Hence, an alternative to the end-to-end neural iTTS architecture that does not apply look-ahead input when synthesizing speech chunks was proposed. A method to use information from the previous time step by connecting the synthesized vector and the model's internal state to the current time step was also proposed.

Research on Bridging the cross-modal gap using adversarial training for speech-to-text translation by Hao et al., (2022) suggests using adversarial training to relieve burden on the Speech Translation (ST) encoder by supplying internal supervision signals in order to close the cross-modal gap. This method significantly boosts performance by allowing the ST model's encoder to extract representations with rich semantics. The efficacy of this methodology has been tested using datasets from MuST-C English-German and Augmented Libri speech English-French. In comparison to strong baseline.

In this study (Pongsathon & Pusadee, 2018), a statistical parametric text-to-speech system for the regional Thai dialect of Isarn is presented. The Hidden Markov Model (HMM) was used to simulate the Mel-cepstrum and fundamental frequencies that make up speech characteristics. By converting the input text into context-specific phonemes, synthetic speech is produced. The context-dependent phonemes are used to generate speech parameters from the trained HMM models. The generated parameters are then used to synthesize speech using a vocoder.

Methodology

This research develops a dynamic Text-to-Speech application for English Language which works with an integrated Application Programming Language then converting it to its equivalent audio. To create the software, all possible

types of basic codes used for design principle based mainly on Javascript, CSS and HTML were used. It has a control system for speech and pitch voice.
TEXT, PHRASE, SENTENCE ⟶ API ⟶ AUDIO
CSS and HTML for were used to design the front end design structure While JavaScript for the interaction with the Application Programming Interface (API) which helps to translate any typed text to its speech equivalent. A brief description of the method deployed is giving in the following sections.

Method of Data collection

The method of data collection was through primary and secondary data collection method which involves the collection of English words, phrases and sentences from respondent as well as collection from Dictionary, the collated words were used in developing the framework for text to speech.

Area and Population of the Study

The area and population of the study is focused on English language words, phrases and sentences. Examples of word, sentences and phrase translated are described in Table 1

Design

Two different algorithms namely use case diagram and flowchart depicted in figure 2 and 3 respectively were used in showing the flow of text-to-speech process.

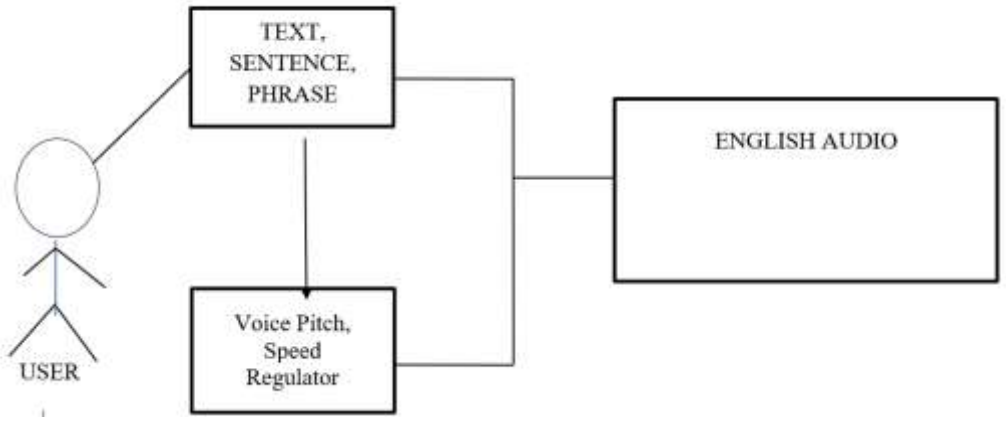

Figure 2: Use case diagram for the Text-to-Speech system

TABLE 1: Table Showing Words, Sentences Recognized and Translated To Speech

WORDS	ENGLISH VERSION
Come	Microsoft David- English (United State)
Here	Microsoft David- English (United State)
Hello	Microsoft David- English (United State)
Longest	Microsoft Mark- English (United State)
Time	Microsoft Mark - English (United State)
Going	Microsoft Zira- English (United State)
Where, etc	Microsoft Zira- English (United State)
SENTENCES	**ENGLISH VERSION**
Where are you	Microsoft David- English (United State)
Today is good	Microsoft MArk- English (United State)
Nice having you around me	Microsoft Zira- English (United State)
English is good, etc	Microsoft David- English (United State)

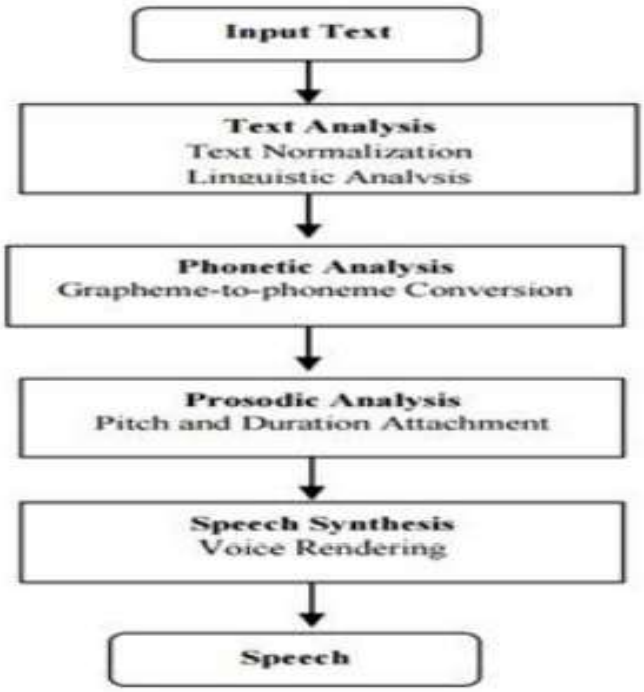

Figure 3: Text-to-speech system flow

Input Design

This involves the input value interface of the Text-To-Speech application and the generated text to speech displayed. This input interface is depicted in figure 4 below

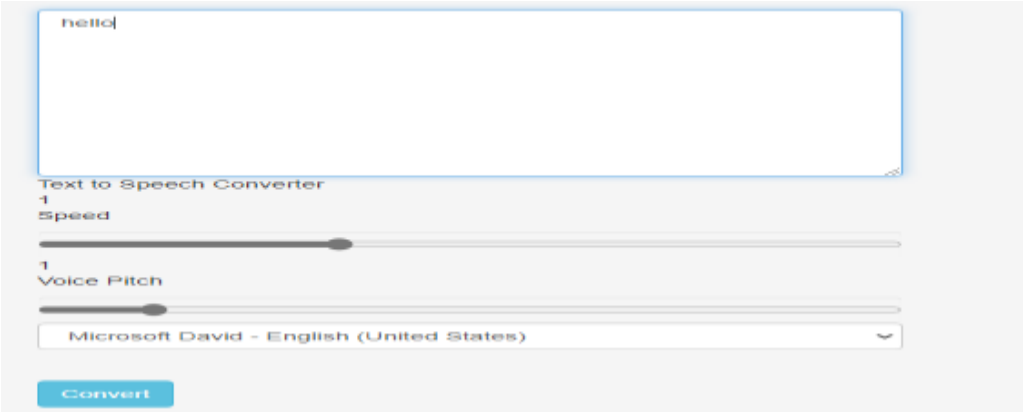

Fig 4: A text-to-speech System Interface

The Text Box: This will consist of the text that Word, sentence or phrase as input for recognition.

VOICE PITCH: This is used for controlling the voice

SPEED: This is used For regulating the speed of text conversion

System Testing, Implementation and Discussion

The **system testing** includes the following procedures;

(i) Unit Testing: The first is **unit testing**, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether the objectives have been met. Errors are noted down and corrected immediately. Unit testing is the important and major part of the research. So, errors are rectified easily in particular module and program clarity is increased. In this research, entire system is divided into several modules and is developed individually.

(ii)Integration testing: The second step includes **Integration testing**. It need not be the case, the software whose modules when run individually and showing perfect results, will also show perfect results when run as a whole. The individual modules are clipped under this major module and tested again and verified the results. A faulty module can have inadvertent, adverse effect on any other or on the global data structures, causing serious problems.

Implementation

(i)Choice of Programming

This research was designed using java script language for the text to speech recognition which works as the major functionality for the conversion. HTML and CSS for the front end. HTML was used for building the working UI and CSS for the styling of the UI. The system platform used was Window Operating System.

(Ii) System Requirements

It can run on any OS such as Window, Linux, or MAC, it requires at least 2Gb Ram with Hard disk Storage of 70Gb

Discussion

After a thorough design has been put in place, it was found that the system is only efficient for English language. It is not efficient for local languages like Yoruba, Igbo, and other languages. It has a speed and a voice pitch controller system.

Conclusion and Future Work

In this research work, an attempt has been made to develop a text to speech synthesis system that can convert text to speech easily. This application can only convert English Text to English Speech, future work in this field should involve conversion of indigenous languages like Yoruba, Ibo and Hausa to its equivalent speech respectively.

References

Dixon, N., & Maxey, H. (1968.). Terminal analog synthesis of continuous speech using the diphone method of segment assembly. *IEEE Trans.Audio Electroacoust., 16* (1), 40–50.

Dudley, H., Riesz, R., & Watkins, S. (2003). A synthetic speaker. *Journal of the Franklin Institute*(6), 739-764.

Dunn, H. K. (2005). The Calculation of Vowel Resonances, and an Electrical Vocal Tract. *The Journal of the Acoustical Society of America*.

Dutoit, T. (1997). *An Introduction to Text-to-Speech Synthesis.* . Kluwer Academics.

Fant, G. (1953.). Speech Communication Research,. *Tech. rep., Royal*.

Georg, I. S., Nkosikhona, D., Alfred, T., & Stan, R. (2017). Text normalisation in text-to-speech Synthesis for South African Languages: Native number expansion. *2017*

Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech). Bloemfontein, South Africa: IEEE.

Hao, Z., Xukui, Y., Dan, Q., & Zhen, L. (2022). Bridging the cross-modal gap using adversarial training for speech-to-text translation. *Digital Signal Processing, 131*.

Hunt, A., & Black, A. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. *International Conference on Acoustics, Speech, and Signal Processing,* (pp. 373–376.). IEEE.

Lawrence, W. (1953). . The synthesis of speech from signals which have a low information rate. . *Communication Theory. Butterworths, London,*.

Oumaima, Z., & Abdelouafi, M. (2017). Novel approach for quality enhancement of Arabic Text To Speech synthesis . *3rd International Conference on Advanced Technologiesfor Signal and Image Processing.* IEEE.

Pabasara, J., Achala, A., Naomi, K., & Amila, R. (2019). An Intelligent Approach of Text-To-Speech Synthesizers for English and Sinhala. *2nd International Conference on Information and Computer Technologies.* IEEE .

Pongsathon, J., & Pusadee, S. (2018). An Isarn dialect HMM-based text-to-speech system. *2017 2nd International Conference on Information Technology (INCIT).* Nakhonpathom, Thailand: IEEE.

Rajae, M., Raddouane, C., Rdouan, F., & Abdellatif, E. A. (2017). An NLP based text-to-speech synthesizer for Moroccan Arabic. *2017 3rd International Conference of Cloud Computing Technologies and Applications.* Rabat, Morocco: IEEE.

Rosen, G. .. (2005). Dynamic analog speech synthesizer. *The Journal of the Acoustical Society of America, 10*(6).

Rubin, P., Baer, T., & Mermelstein, P. (1981). An articulatory synthesizer for perceptual research. *Journal of the Acoustical Society of America*(70), 321–328.

Schroeder, M. (1993). A brief history of synthetic speech. (Elsevier, Ed.) *Speech Communication, 13*(1-2), 231-237.

Taylor, P. (2009). *Text-to-speech synthesis.* Cambrigde University Press.

Tomoya, Y., Sakriani, S., & Satoshi, N. (2023). Japanese Neural Incremental Text-to-Speech Synthesis Framework With an Accent Phrase Input. *IEEE*, 22355 - 22363.

Wesley, M., & Werner, V. (2014). Audiovisual speech synthesis: An overview of the state-of-the-art. *ELsevier*.

Zen, H., Tokuda, K., & B. A. (2009.). Statistical parametric speech synthesis. . *Speech Communication.*, 1039–1064.