CONTENTS

CONTENTS

1.INTRODUCTION:

. The Text-to-Speech Synthesis Project aims to develop a sophisticated system capable of converting written text into natural-sounding speech. This innovative technology holds immense potential for various applications, including accessibility tools for visually impaired individuals, voice assistants, interactive educational platforms, and more. In this project, we will leverage cutting-edge natural language processing (NLP) and machine learning techniques to create a robust and adaptable text-to-speech engine. By combining deep learning algorithms with linguistic analysis, we aim to produce high-quality, human-like speech output

In an era dominated by digital communication, the importance of natural and intelligible speech synthesis cannot be overstated. The Text-to-Speech (TTS) system represents a pivotal technology in bridging the gap between human language and machines. This project aims to develop a robust and versatile TTS system that converts written text into lifelike spoken language.The core objective of this endeavor is to create a TTS model that not only exhibits exceptional linguistic accuracy but also offers a diverse range of voice styles and tones, allowing for personalized user experiences. By employing cutting-edge deep learning techniques and leveraging a vast corpus of textual data, we aim to surpass the current benchmarks in TTS technology.Moreover, this project will explore avenues for optimizing the computational efficiency of the TTS system, ensuring it is capable of real-time performance across various platforms. Accessibility will be a key focus, with the goal of making the synthesize

In recent years, the most popular acoustic model in automatic speech recognition (ASR) and text-to-speech synthesis (TTS) is a hidden Markov model (HMM), due to its ease of implementation and modeling flexibility. However, a number of limitations for modeling sequences of speech spectra using the HMM have been pointed out, such as i) piece-wise constant statistics within a state and ii) conditional independence assumption of state output probabilities. To overcome these shortcomings, a variety of alternative acoustic models have been proposed. Although these models can improve model accuracy and speech recognition performance, they generally require an increase in the number of model parameters. In contrast, dynamic features can also enhance performances of HMM-based speech recognizers and has been widely adopted. It can be viewed as a simple mechanism to capture time dependencies in the HMM. However, this approach is mathematically improper in the sense of statistical modeling. Generally, the dynamic features are calculated as regression coefficients from their neighboring static features. Therefore, relationships between the static and dynamic features are deterministic. However, these relationships are ignored and the static and dynamic features are modeled as independent statistical variables in the HMM framework. Ignoring these interdependencies allows inconsistency between the static and dynamic features when the HMM is used as a generative model in the obvious way. In the present dissertation, a novel acoustic model, named a trajectory HMM, is described. This model is derived from the HMM whose state output vector includes both static and dynamic features. By imposing explicit relationships between the static and dynamic features, the HMM is naturally translated into a trajectory model. The above inconsistency and limitations of the HMM can be alleviated by the trajectory HMM.

Furthermore, parameterization of the trajectory HMM is completely the same as that of the HMM with the same model topology. Therefore, any additional parameters are not required. In the present dissertation, model training algorithms based on a Viterbi approximation and a Markov chain Monte Carlo (MCMC) method and a search algorithm based on a delayed decision strategy are also derived. Results of continuous speech recognition and speech synthesis experiments show that the trajectory HMM can improve the performance both of speech recognizers and synthesizers.

1.1 PROBLEM DEFINITION.

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 ares to red in a database. Systems differ in the size of the stored speech units; a system that stores phones or dip hones 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.

2.System Specifications.

2.1 HARDWARE SPECIFICATION

PROCESSOR	Intel core i5 11th gen
RAM	8 GB
SSD	512 GB
MONITOR	19.5” LED Moniter

OS	windows 11

2.2 Software SPECIFICATION

Environment	python 3.7.2 IDE
Operating system	Windows 10/11
coding Language	python

3.PROJECT DESCRIPTION

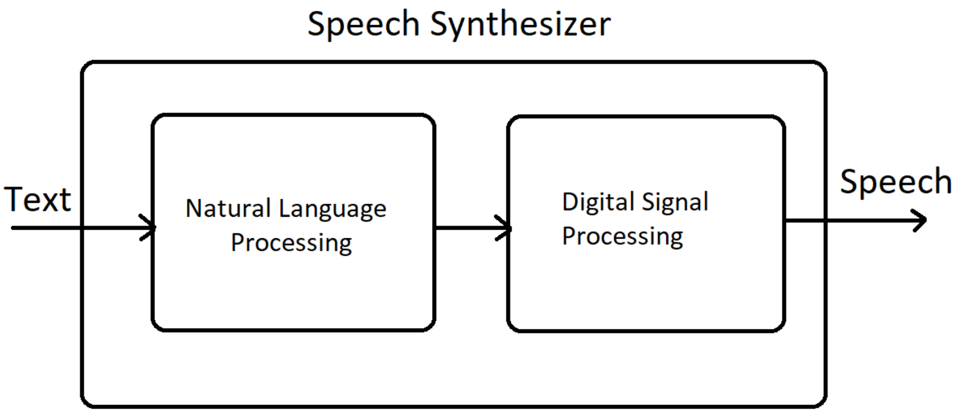
3.1 INTRODUCTION

Text to speech is a process to convert any text into voice. Text to speech project takes words on digital devices and convert them into audio with a button click or finger touch. Text to speech python project is very helpful for people who are struggling with reading.

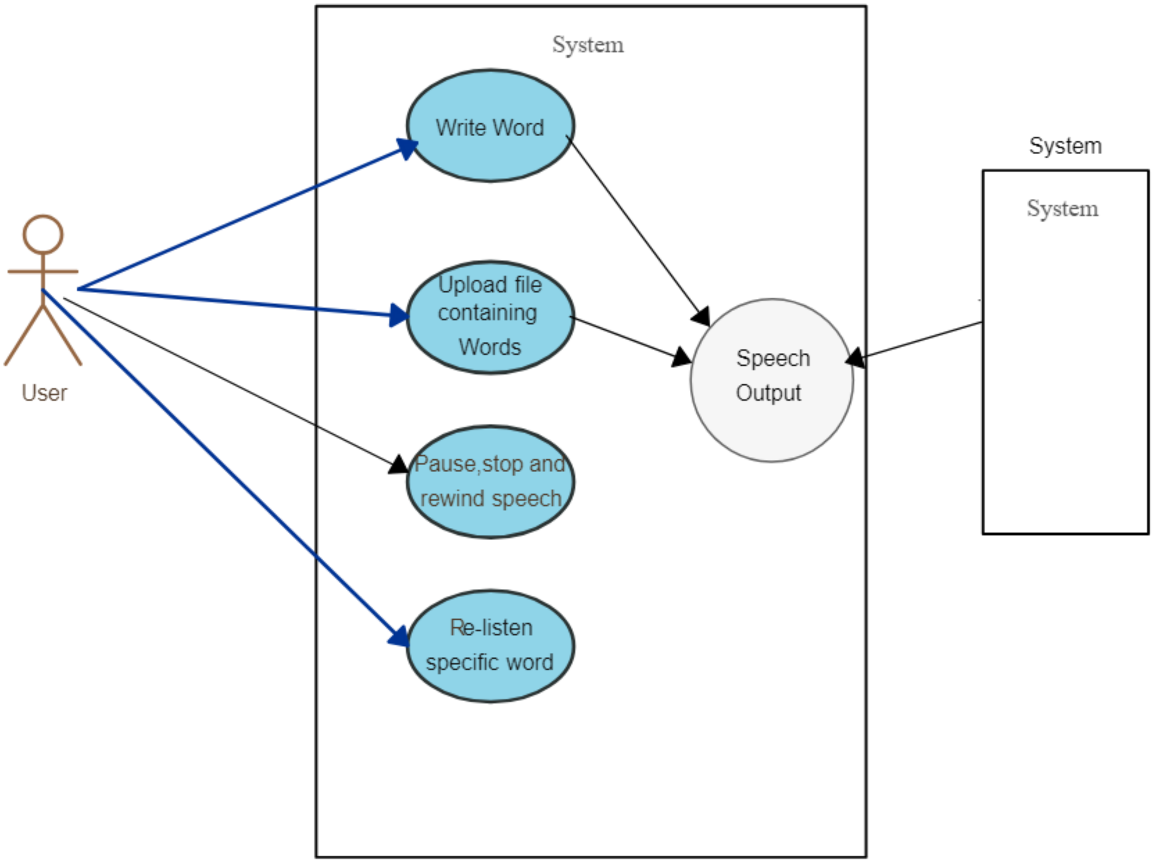
Text-to-Speech (TTS) is a useful technology that converts any text into a speech signal. It can be utilized for various purposes, e.g. car navigation, announcements in railway stations, response services in telecommunications, and e-mail reading. Corpus-based TTS makes it possible to dramatically improve the naturalness of synthetic speech compared with the early TTS. However, no general-purpose TTS has been developed that can consistently synthesize sufficiently natural speech. Furthermore, there is not yet enough flexibility in corpus- based TTS. This thesis addresses two problems in speech synthesis. One is how to improve the naturalness of synthetic speech in corpus-based TTS. The other is how to improve control of speaker individuality in order to achieve more flexible speech synthesis. To deal with the former problem, we focus on two factors: (1) an algorithm for selecting the most appropriate synthesis units from a speech corpus, and (2) an evaluation measure for selecting the synthesis units. Moreover, we focus on a voice conversion technique to control speaker individuality to deal with the latter problem. Since various vowel sequences appear frequently in Japanese, it is not realistic to prepare long units that include all possible vowel sequences to avoid vowel-to-vowel concatenation, which often produces auditory discontinuity. In order to address this problem, we propose a novel segment selection algorithm based on both phoneme and diphone units that does not avoid concatenation of vowel sequences but alleviates the resulting discontinuity. Experiments testing concatenation of vowel sequences clarify that better segments can be selected by considering concatenations not only at phoneme boundaries but also at vowel centers. Moreover, the results of perceptual experiments show that speech synthesized using the proposed algorithm has better naturalness than that using the conventional algorithms. A cost is established as a measure for selecting the optimum waveform segments from a speech corpus. In order to achieve high-quality segment selection for concatenative TTS, it is important to utilize a cost that corresponds to perceptual characteristics. We first clarify the correspondence of the cost to the perceptual scores and then evaluate various functions to integrate local costs capturing the degradation of naturalness in individual segments. From the results of perceptual experiments, we find a novel cost that takes into account not only the degradation of naturalness over the entire synthetic speech but also the local degradation. We also clarify that the naturalness of synthetic speech can be slightly improved by utilizing this cost and investigate the effect of using this cost for segment selection. We improve the voice conversion algorithm based on the Gaussian Mixture Model (GMM), which is a conventional statistical voice conversion algorithm. The GMM-based algorithm can convert speech features continuously using the correlations between source and target features. However, the quality of the converted speech is degraded because the converted spectrum is excessively smoothed by the statistical averaging operation. To overcome this problem, we propose a novel voice conversion algorithm that incorporates Dynamic Frequency Warping (DFW) technique. The experimental results reveal that the proposed algorithm can synthesize speech with a higher quality while maintaining equal conversion- accuracy for speaker individuality compared with the GMM-based algorithm.

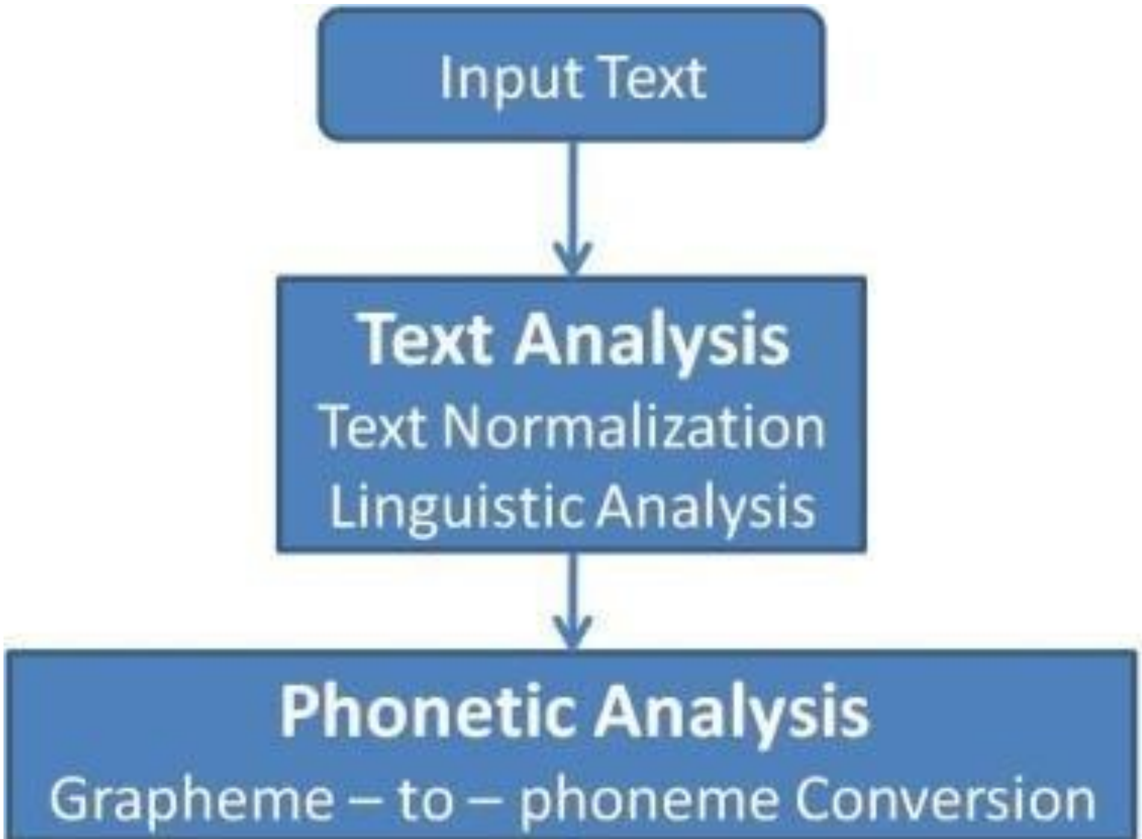
3.2 DATAFLOW DIAGRAM

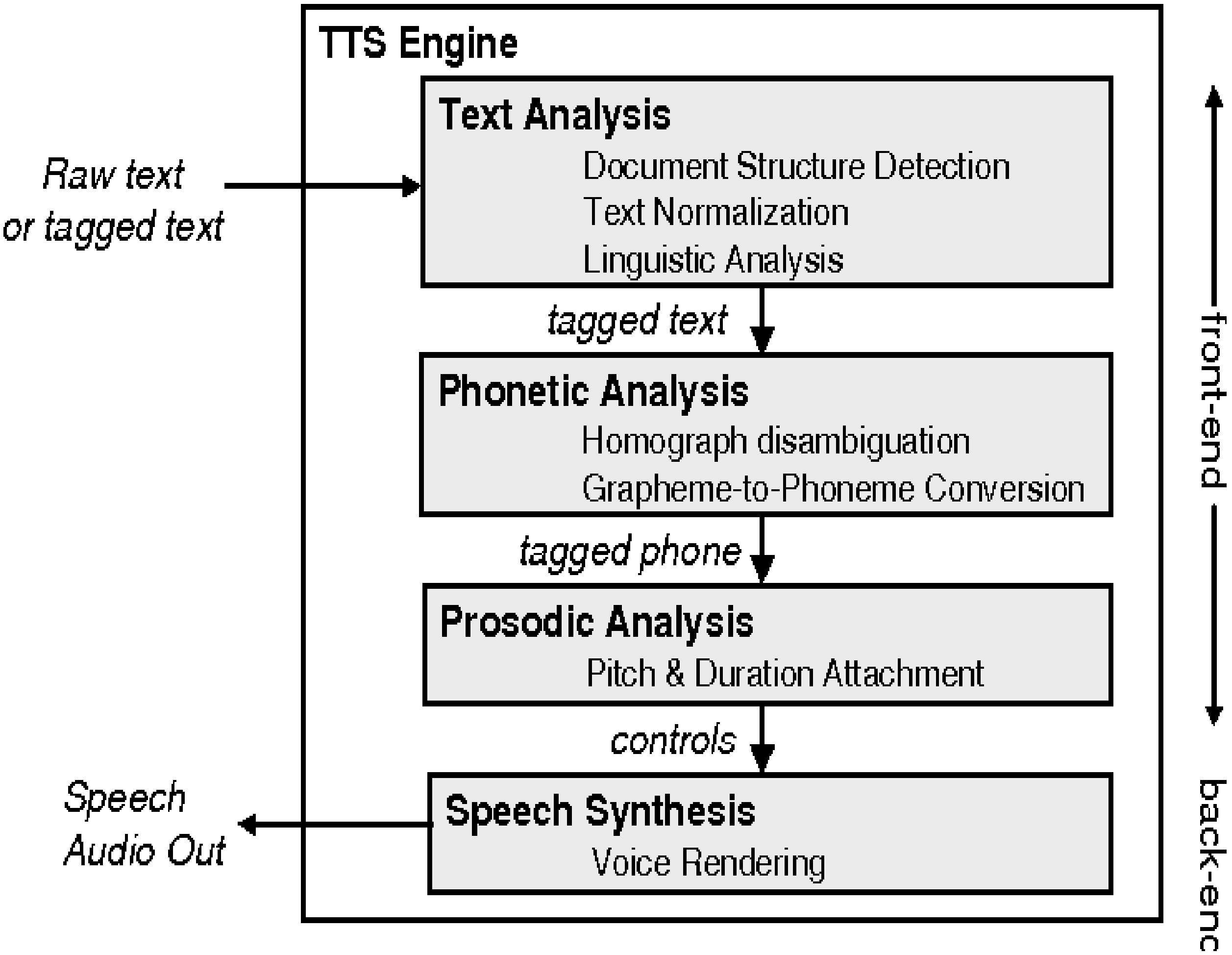
STEPS INVOLVED

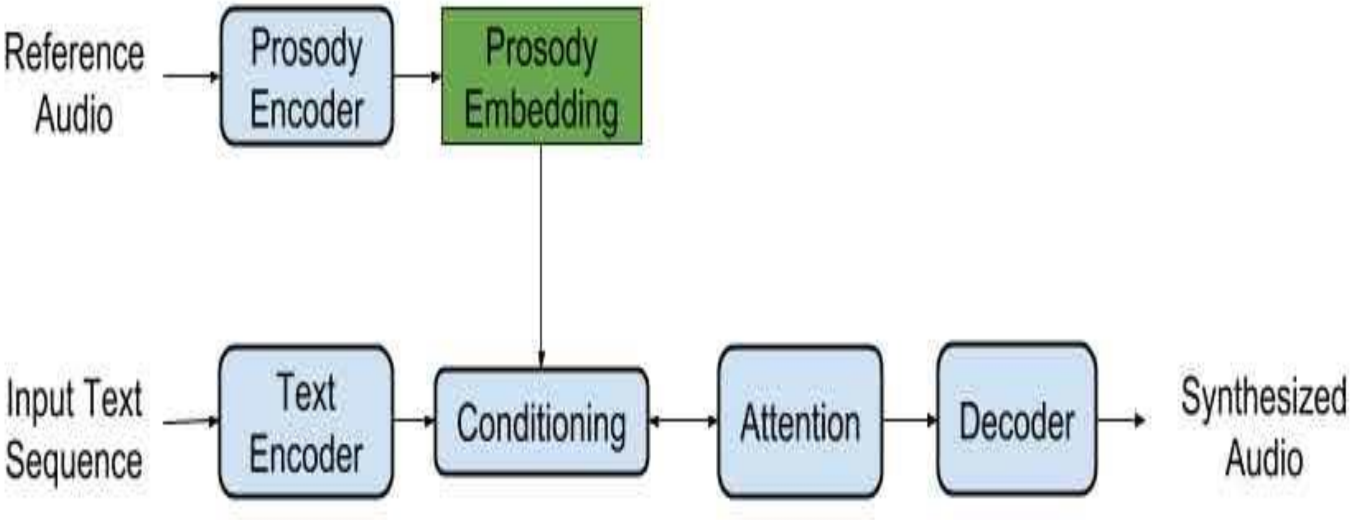


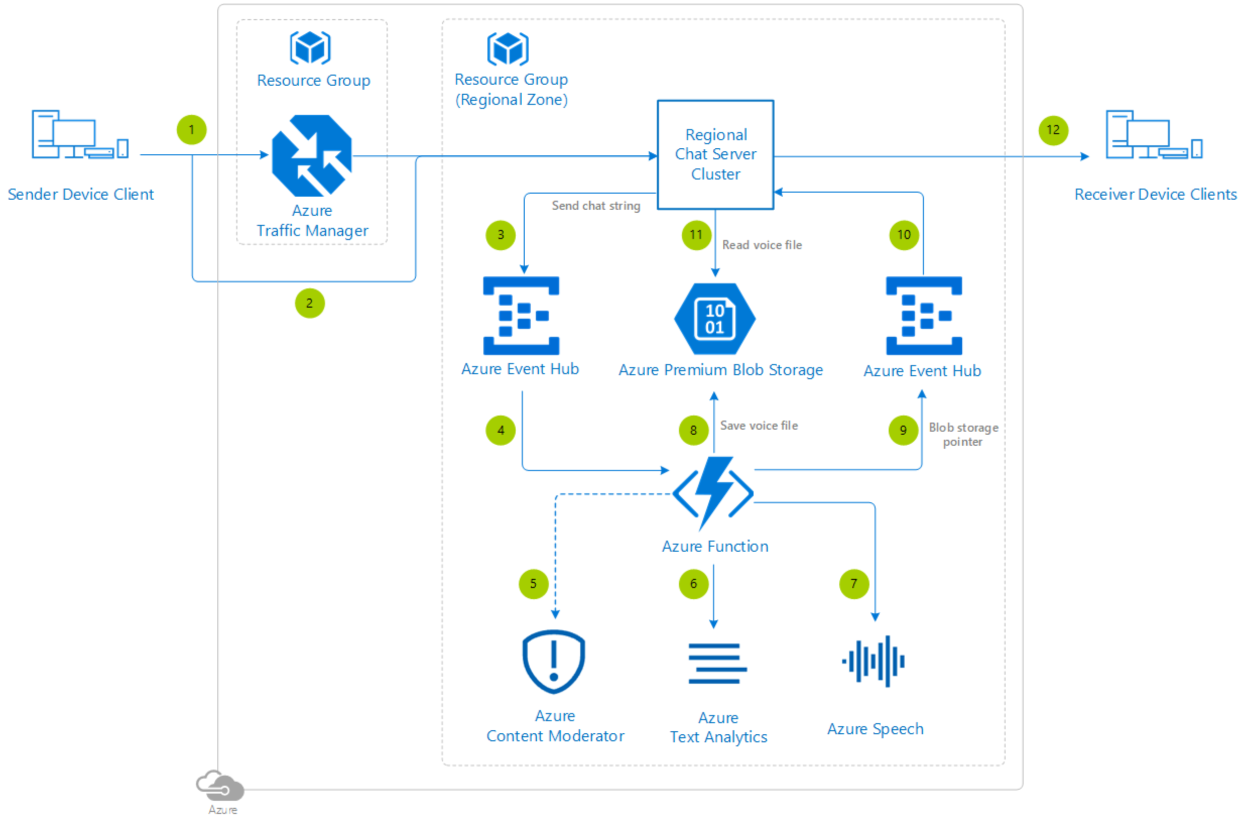
Text To speech Use Case Diagram

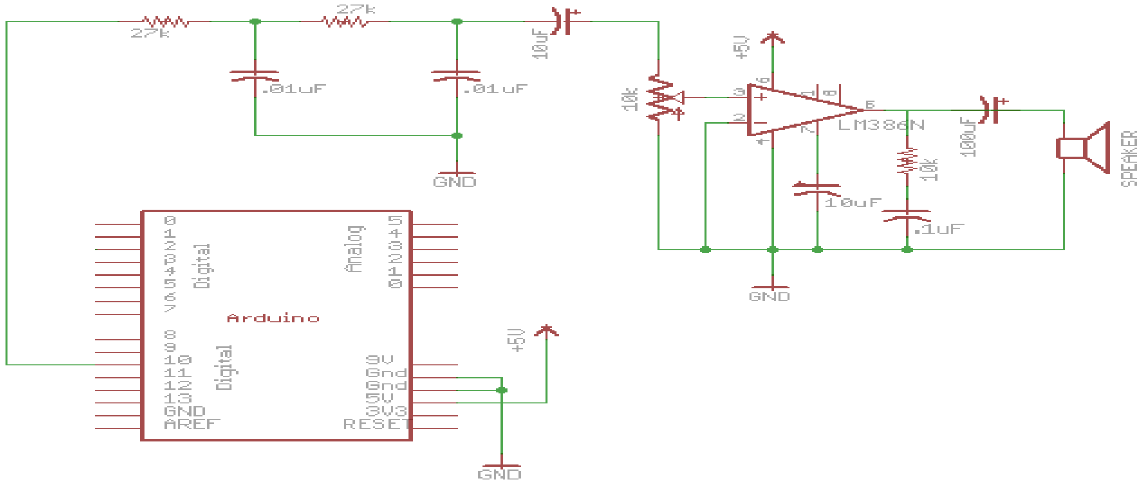


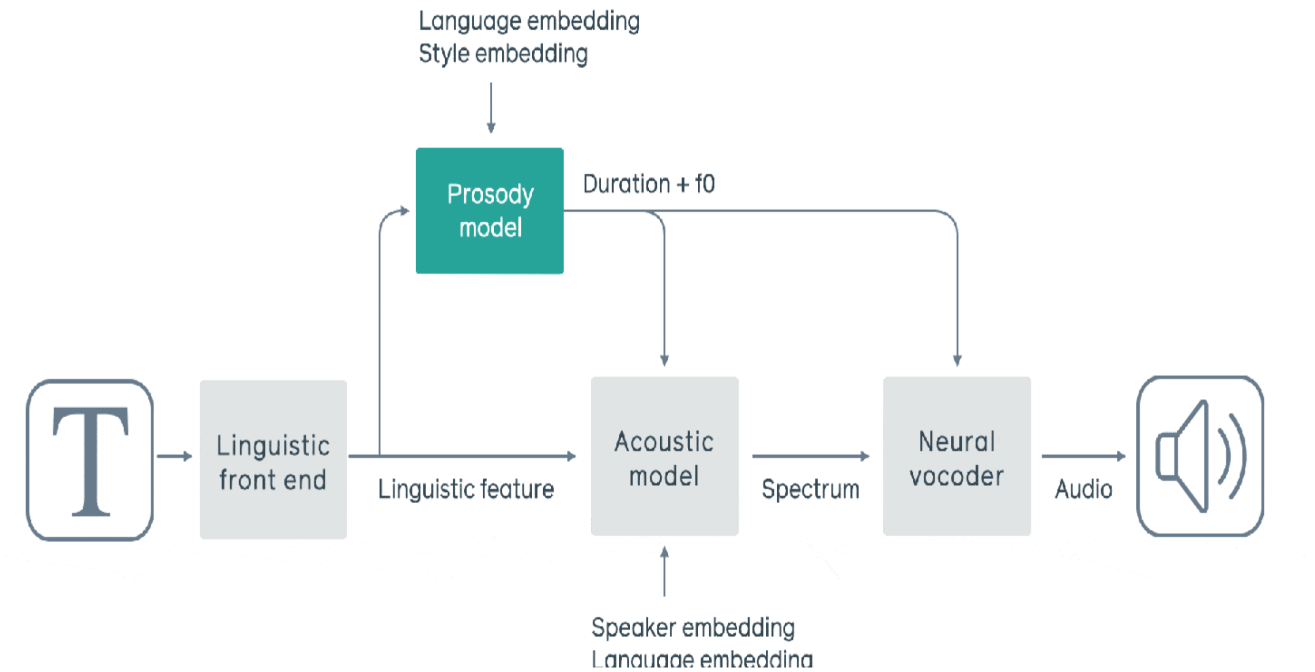


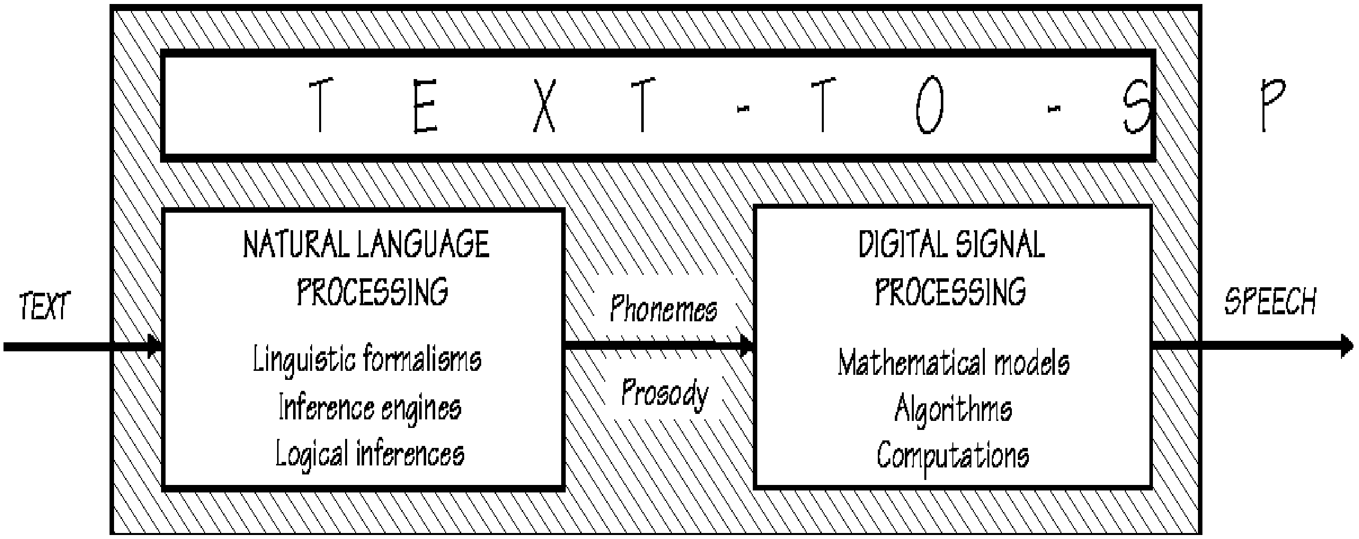

TTS Engine
Text Analysis
Document Structure Detection
Text Normalization
Linguistic Analysis
Raw text
or tagged text
tagged text
Phonetic Analysis
Homograph disambiguation
Grapheme-to-Phoneme Conversion
tagged phone
Prosodic Analysis
Pitch & Duration Attachment
controls
Speech Synthesis
Voice Rendering
Speech
Audio Out
front-end
back-enc


Reference
Audio
Prosody
Encoder
Prosody
Embedding
Input Text
Sequence
Text
Encoder
Conditioning
Attention
Decoder
Synthesized
Audio


Resource Group
Resource Group
(Regional Zone)
Sender Device Client
1
Azure
Traffic Manager
2
Regional
Chat Server
Cluster
12
Receiver Device Clients
Send chat string
3
11
Read voice file
10
Azure Event Hub
Azure Premium Blob Storage
Azure Event Hub
4
8
Save voice file
9
Blob storage
pointer
Azure Function
5
6
7
Azure
Content Moderator
Azure
Text Analytics
Azure Speech
Azure


27k
27k
.01uF
.01uF
GND
10uF
10k
+5V
LM386N
100uF
10k
10uF
.1uF
SPEAKER
GND
Arduino
Digital
Analog
Digital
9V
Gnd
Gnd
5V
3V3
RESET
GND
AREF
+5V
GND





3.3 Module description

Module that allows the use of TTS (Text-To-Speech).
The conversion of text into speech or speech synthesis (TTS or Text-to-Speech) is the technology that allows you to convert, automatically, a written text into a natural voice, naturally

and with maximum intelligibility. The system provided with this engine can “read aloud” a written text.

The TTS module works similar to an audio player but is much more flexible. It can be used to read out a text to the customer. The text does not have to be uploaded but can be inserted directly into the module.

The conversion of text into speech or speech synthesis (TTS or Text-to-Speech) is the technology that allows you to convert, automatically, a written text into a natural voice, naturally and with maximum intelligibility. The system provided with this engine can “read aloud” a written text.

Thanks to this module, SIME software can be integrated with the high quality Verbio TTS motor, and the user can write the desired message and it will be converted to voice. This message can be routed to any desired zone.

This text-to-speech module gives the installation a way to communicate with its users in a natural and intelligible way –expressively and dynamically. The integrated voices can be modulated for many applications in large venues such as convention centres and, at the same time, can express feelings –making them very realistic.

The LDA SIME’s TTS module is integrated with the Verbio’s SDK, expert in TTS applications, and can generate dynamic text. Paired with LVSR (Large Vocabulary Speech Recognition) and Natural Language Understanding, you can achieve optimal compression levels.

To create a text-to-speech project, you'll need a suitable programming language and a TTS (Text-to-Speech) module or library. Here are some popular choices for different languages:Python:pyttsx3: This is a text-to-speech conversion library in Python. It works offline and supports multiple TTS engines.gTTS (Google Text-to-Speech): This is a Python library and CLI tool to extract the spoken text from videos and audios.SpeechRecognition: While not a TTS module, it allows you to use various online TTS services through an API.

4.SAMPLE CODING

```
from time import sleep

from tkinter import*

from tkinter import simpledialog

from tkinter import messagebox as msg

import pyttsx3

import speech_recognition as sr

import os

import shutil
import datetime

import socket


def listen(duration):

    t= sr.Recognizer()

    with sr.Microphone() as source:

        text = t.record(source, duration=duration)

        try:

            return t.recognize_google(text)

        except:

            return "Could not understand audio"


def ssk():

        text = listen(5)

        e.insert(END, text)

        sh.place(x=2332,y=2322)

        e.place(x=10,y=10)


def write_text():

    if (socket.gethostbyname(socket.gethostname()) == "127.0.0.1"):

        msg.showerror("App","Your device is not connected to internet")

    else:

        e.place(x=10000,y=10000)

        sh.place(x=30,y=20)

        t.after(1000, ssk)


def speak():

    pyttsx3.speak(e.get("1.0",END).replace("\n",""))
```

```
def save():
        p = simpledialog.askstring("Save","Enter filename.")
        if (p+".txt" in os.listdir()):
                pyttsx3.speak("File with this name already exists")
                msg.showerror("Error","File with this name already exists")
        else:
                open(p+".txt","a").write(e.get("1.0",END))
                pyttsx3.speak("File saved successfully.")
                msg.showinfo("Success","File saved successfully")


t= Tk()
t.geometry("300x300")
t.title("Speech recognition system")
Label(background="grey", width=100 , height=1000).place(x=0,y=0)
Button(text="Activate Microphone", command =write_text).place(x=10,y=180)
Button(text="Speak", width=5, command=speak).place(x=10,y=220)
Button(text="Save", width=5, command=save).place(x=140,y=180)
sh = Label(text='say something!', font=["Arial", 20], background="orange")
e=Text(bd=4, height=8, width=32)
e.place(x=10,y=10)
t.mainloop()
```

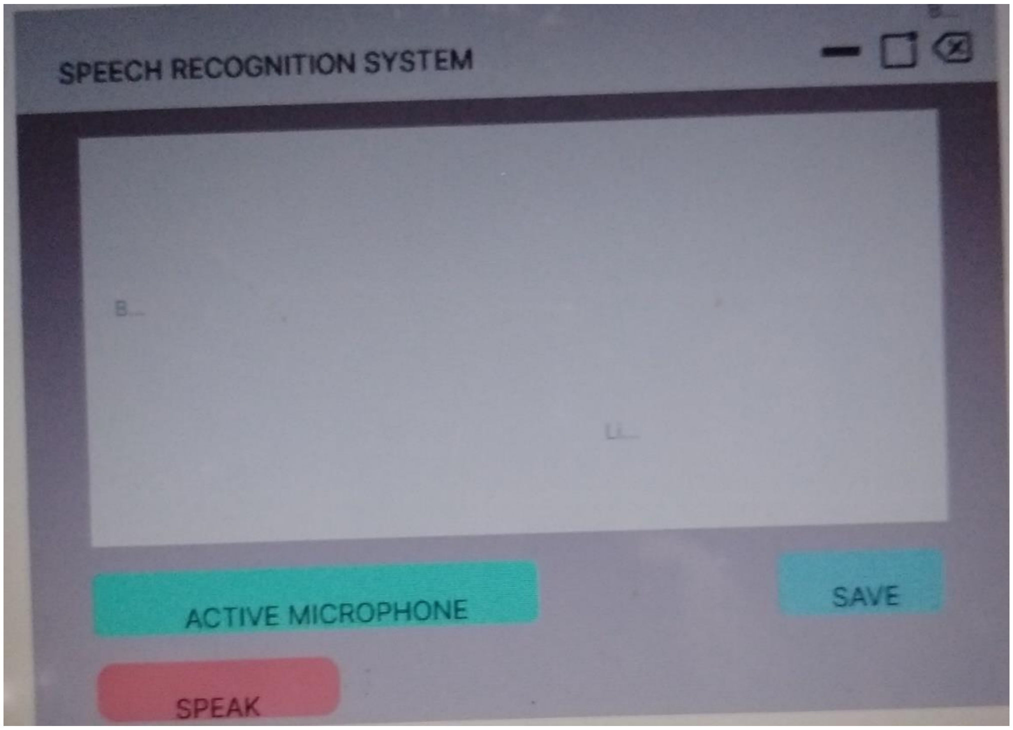
5.SCREENSHOT

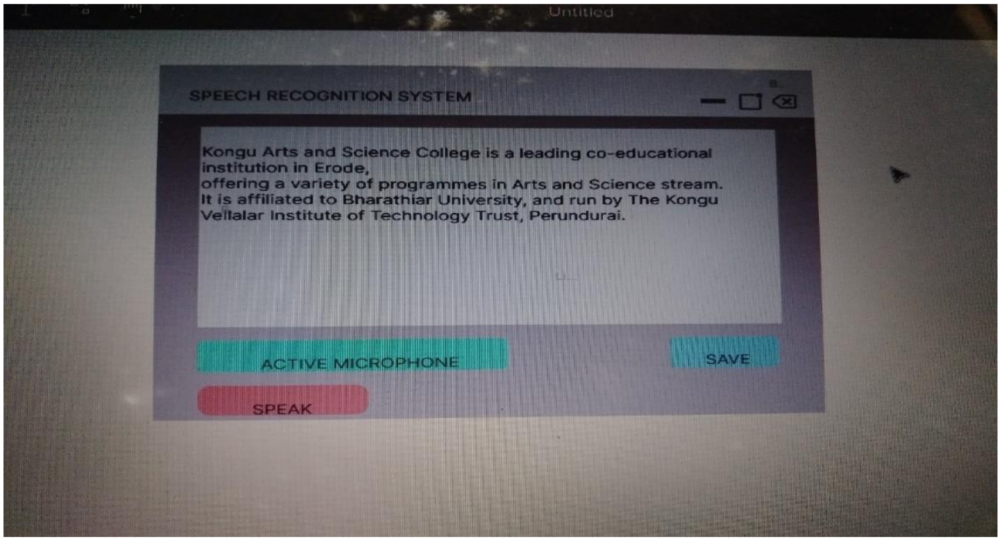
INPUT FORMS

SAMPLE SCREENS FOR EXTRACTING TEXT TO SPEECH

5.SCREEN SHORT

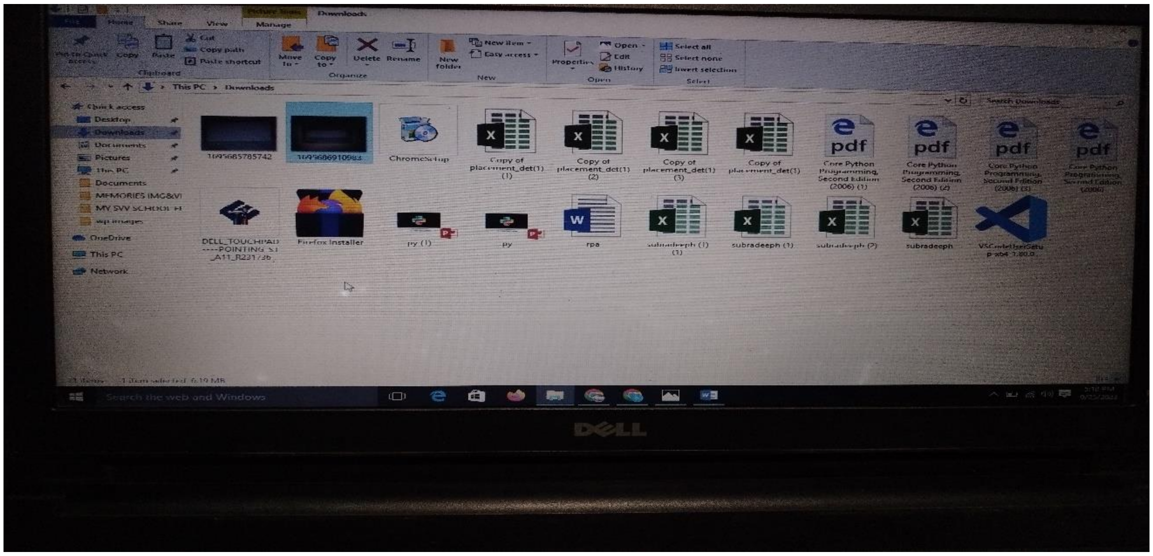
5.1 INPUT FORMS





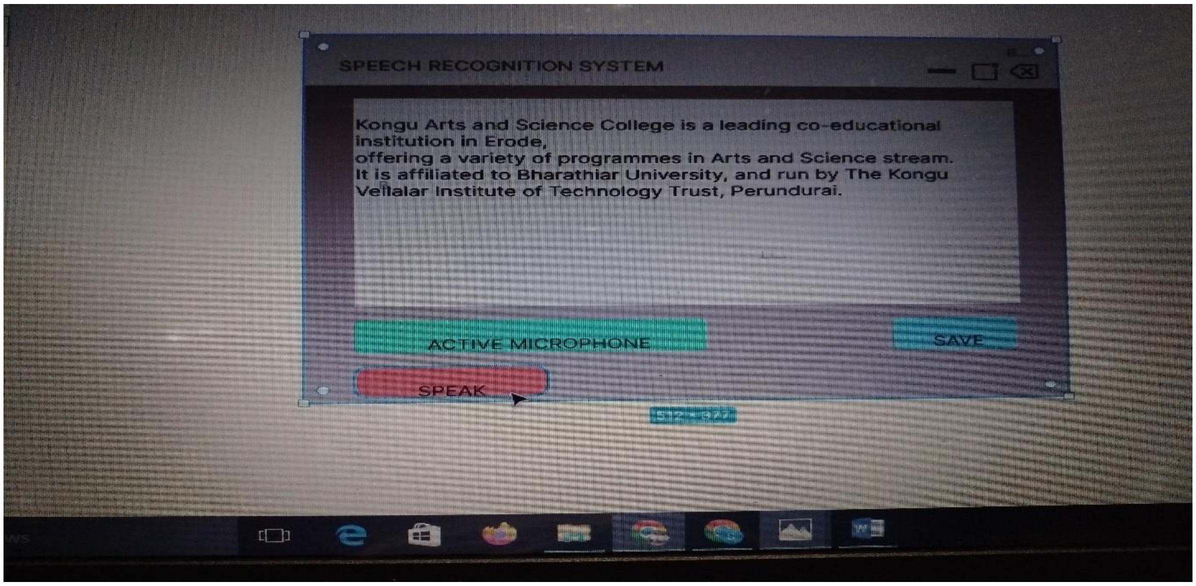
Kongu Arts and Science College is a leading co-educational
institution in Erode,
offering a variety of programmes in Arts and Science stream.
It is affiliated to Bharathiar University, and run by The Kongu
Vellalar Institute of Technology Trust, Perundurai.

THE RECOGNIZERD TEXT WILL BE STORED IN THE NOTEPAD DOCUMENT NAMED TEXT.DOC CURRENT WORKING DIRECTORY



5.1 OUTPUT FORMS

FOLLOWING SCREEN ARE SHOWN THE TEXT STORED IN NOTEPAD READY TO SPEAK



Kongu Arts and Science College is a leading co-educational institution in Erode,
offering a variety of programmes in Arts and Science stream.
It is affiliated to Bharathiar University, and run by The Kongu Vellalar Institute of Technology Trust, Perundurai.

6. CONCLUSION &FUTURE WORKS

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 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 in the application. Our system interfaces with a text to speech engine developed for American English. In future, we plan to make efforts to create engines for localized Nigerian language so as to make text to speech technology more accessible to a wider range of Nigerians

Although speech-to-text conversion (STT) machines aim at providing benefits for the deaf or people who can't speak, it is difficult to review, retrieve and reuse speech transcripts. Hence, when the speech to text conversion module is combined with the summarization, the applications further increase in educational fields as well. This chapter discussed the need for speech summarization, various issues in the summarization of a spoken document, supervised, and unsupervised summarization algorithms. Isolated Tamil speech recognition was performed using a sample set of Tamil spoken words. In addition, state-of-the-art recognition techniques were used, and analysis was performed. Also, the summarization of speech data in Tamil language is explored, along with related work on text summarization. The features used in the summarization of a spoken document are analyzed and compared, based on the various forms of input into the spoken document.

Text-to-speech (TTS) conversion can provide status information to an eyes-busy CR user. TTS technology automatically speaks textual information. Textual information could originate from text-based communications or equipment display readouts. Text-based examples of communications that could be spoken via TTS include email, news, Web, rich site summary (RSS), Web logs (blogs), instant messaging (IM), Internet relay chat (IRC), and short message service (SMS). Traditional equipment display readouts could also be spoken via TTS, such as radio frequency, battery power, signal strength, network data rate, time, velocity, location, and bearing.

By providing status information to an eyes-busy user, TTS enables soldiers to focus on their mission while hearing an explanation of their battle space and status. Different synthesized voice types (e.g., male and female) are usually employed to convey various types of information. For example, routine or urgent information could be conveyed in male or female voices, respectively.

The current state of TTS technology produces mostly reasonable-sounding speech; however, it does not yet sound quite human. Future research directions in TTS are focusing on improving the quality of voice synthesis, pronunciation of named entities, conveyance of expression, and integration with machine translation and speech-to-text conversion**.**

Text-to-speech (TTS) coding is a powerful ultra-low-bit-rate (as low as a few dozen bits per second) method for transmission of speech when the precise timbre and diction of the speaker's voice are not important. MPEG-4 standardizes an interface to TTS synthesis systems so that a single bitstream format can be synthesized with a plug-in TTS module. This interface allows transmission of language-independent phoneme representations, prosody information in the form of pitch and timing, and certain aspects of control over vocal timbre. The particular method of speech synthesis is not standardized; only the parameter stream and its meaning is standardized in MPEG-4. Any desired method for mapping from parameters to audio may be used.

The TTS interface in MPEG-4 can also be used to automatically derive parameters for the Face Animation visual tool. By using this functionality, an audiovisually synchronized “talking head” can be created at very low bandwidth

7.BIBLIOGRAPHY

FOR PYTHON IN YOUTUB

https://youtu.be/u6cmiBp1obc

FOR PYTHON IN GEEKS FOR GEEKS

https://www.geeksforgeeks.org/machine-learning/?ref=shm