A Seminar on

Neural Network-Based Voice Dialogue System for Email Management

Team Details
1. G. Sri Latha(20EG105647)
2. K. Nithin(20EG105653)

Project Supervisor
Name: : Mr. B.V. Srikanth
Designation

Introduction

This E-mail architecture that will help the Blind people to access the services easily and efficiently for communication without previous training. This system designed for drawback that motivates targeted solution focus on effectively usage by both handicapped and illiterate persons. By utilizing TTS for voice interaction, Gmail API for email retrieval, composition, and organization, and Cloud Storage for storing user preferences and data securely, this seeks to revolutionize email management for the visually impaired.

Problem Statement

ANURAG UNIVERSITY

The internet new advancement has been implemented very efficiently the visually challenged users find it very difficult to use the technologies as normal users. This project aims in developing an E-mail architecture that will help the Blind people to access the services easily and efficiently for communication without previous training. This system designed for drawback that motivates targeted solution focus on effectively usage by both handicapped and illiterate persons.

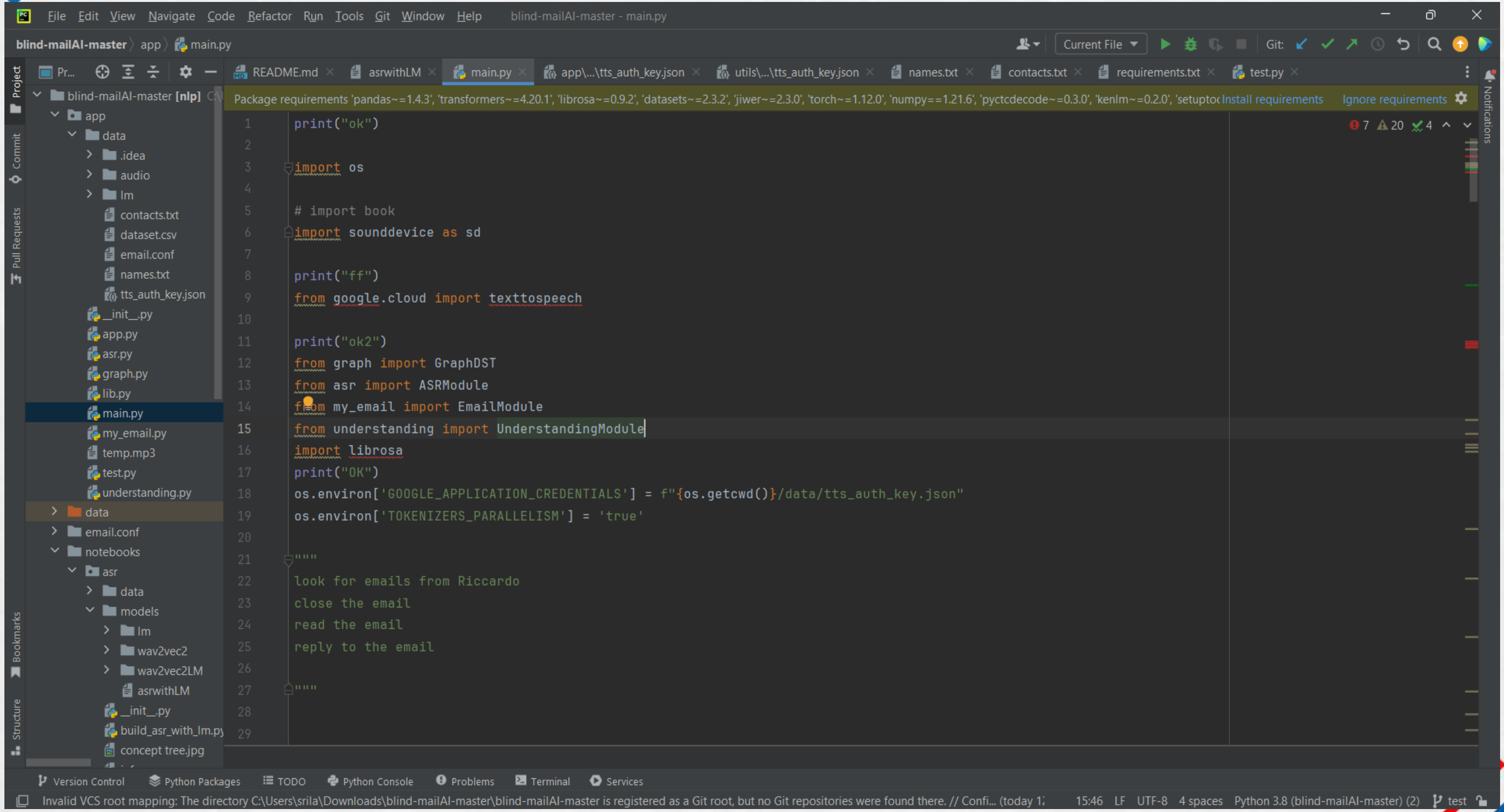
A voice email manager ... with Neural Networks. It use a wav2vec2 + LM for ASR, google cloud TTS for voice and BENT for understanding (intent classification and token classification). The system is trained with syntetic data. It use rdf for dialogue state tracking and smtp/pop3 to comunicating with email server.

Proposed Method

Email Management systems for visually impaired individuals faced several challenges. These systems relied heavily on visual interfaces, making them inaccessible for blind users. In addition, they lacked efficient accessibility features, such as screen readers and basic keyboard shortcuts, hindering effective communication as:

1. **User Input**: The user interacts with the system using voice commands.
2. **Text-to-Speech (TTS):** The system converts text to speech using Google's TTS API.
3. **Gmail API**: The system interacts with Gmail API to retrieve, compose, and organize emails.
4. **Voice Interaction**: The system prompts the user with voice commands to perform certain actions.
5. **Customization**: Users can customize the system to their preferences.
6. **Efficient Navigation:** The system eliminates the need for keyboard input, allowing users to navigate emails using voice commands and mouse clicks.
7. **Email Management:** Users can read, compose, and organize emails efficiently using the voice-based interface

Experiment Environment(1/4)

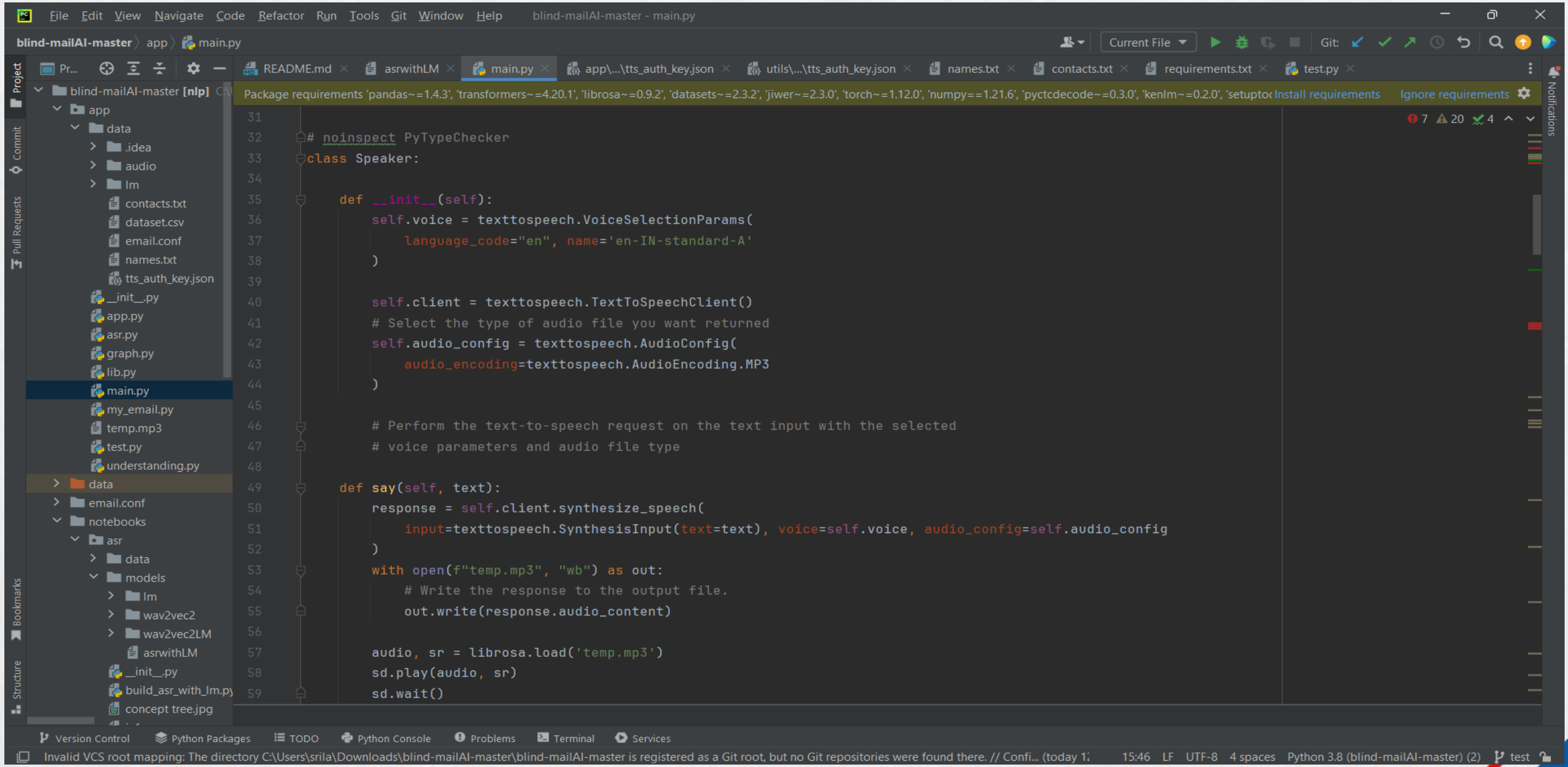


Package requirements 'pandas~=1.4.3', 'transformers~=4.20.1', 'librosa~=0.9.2', 'datasets~=2.3.2', 'jiwer~=2.3.0', 'torch~=1.12.0', 'numpy==1.21.6', 'pyctcdecode~=0.3.0', 'kenlm~=0.2.0', 'setuptoc Install requirements Ignore requirements

```python
print("ok")

import os

# import book
import sounddevice as sd

print("ff")
from google.cloud import texttospeech

print("ok2")
from graph import GraphDST
from asr import ASRModule
from my_email import EmailModule
from understanding import UnderstandingModule
import librosa
print("OK")
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = f"{os.getcwd()}/data/tts_auth_key.json"
os.environ['TOKENIZERS_PARALLELISM'] = 'true'

"""
look for emails from Riccardo
close the email
read the email
reply to the email

"""
```

Invalid VCS root mapping: The directory C:\Users\srila\Downloads\blind-mailAI-master\blind-mailAI-master is registered as a Git root, but no Git repositories were found there. // Confi... (today 1: 15:46 LF UTF-8 4 spaces Python 3.8 (blind-mailAI-master) (2) test

Experiment Environment(2/4)

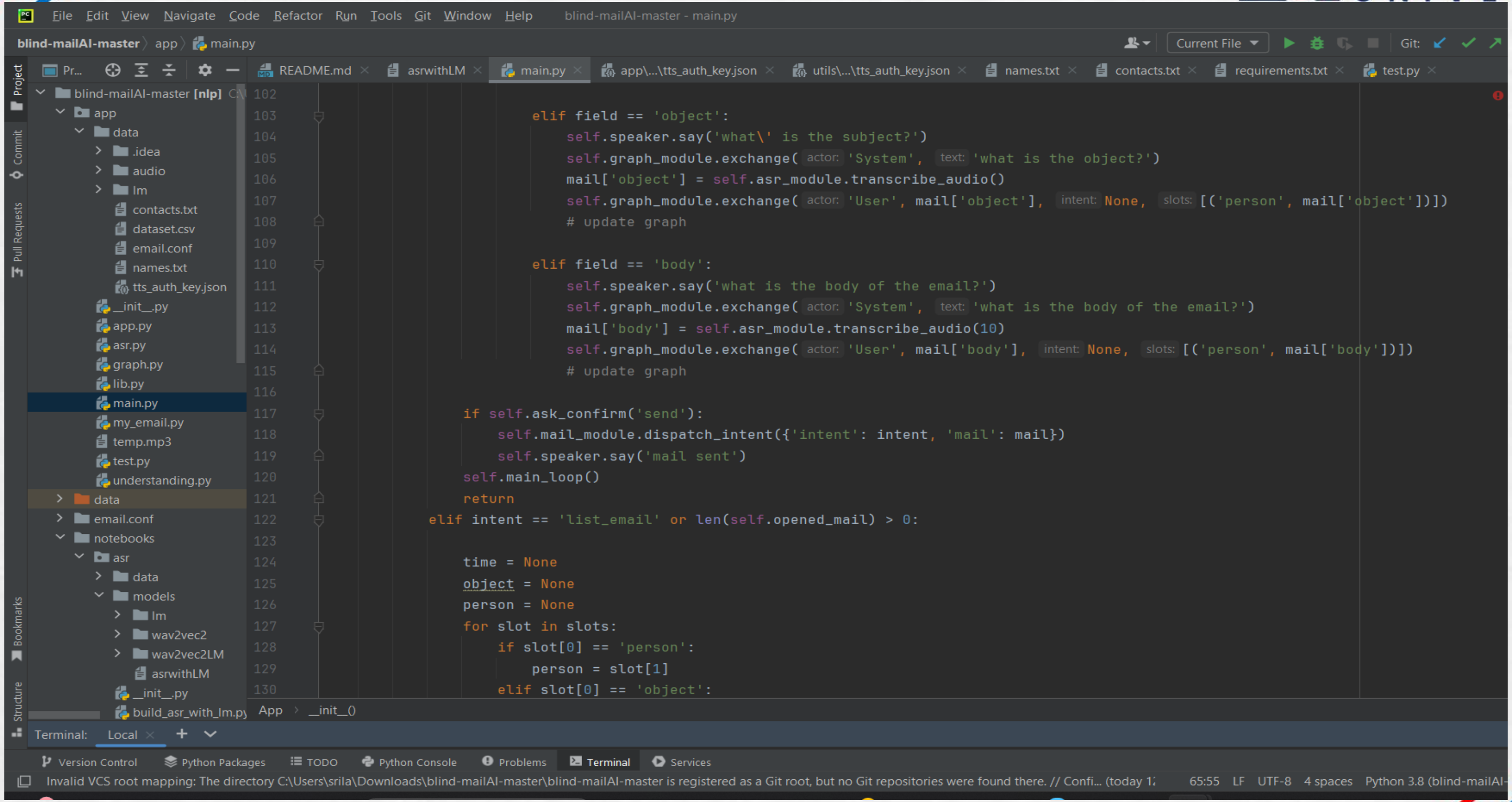


Package requirements 'pandas~=1.4.3', 'transformers~=4.20.1', 'librosa~=0.9.2', 'datasets~=2.3.2', 'jiwer~=2.3.0', 'torch~=1.12.0', 'numpy==1.21.6', 'pyctcdecode~=0.3.0', 'kenlm~=0.2.0', 'setuptoc Install requirements Ignore requirements

```python

# noinspect PyTypeChecker
class Speaker:

    def __init__(self):
        self.voice = texttospeech.VoiceSelectionParams(
            language_code="en", name='en-IN-standard-A'
        )

        self.client = texttospeech.TextToSpeechClient()
        # Select the type of audio file you want returned
        self.audio_config = texttospeech.AudioConfig(
            audio_encoding=texttospeech.AudioEncoding.MP3
        )

        # Perform the text-to-speech request on the text input with the selected
        # voice parameters and audio file type

    def say(self, text):
        response = self.client.synthesize_speech(
            input=texttospeech.SynthesisInput(text=text), voice=self.voice, audio_config=self.audio_config
        )
        with open(f"temp.mp3", "wb") as out:
            # Write the response to the output file.
            out.write(response.audio_content)

        audio, sr = librosa.load('temp.mp3')
        sd.play(audio, sr)
        sd.wait()
```

Invalid VCS root mapping: The directory C:\Users\srila\Downloads\blind-mailAI-master\blind-mailAI-master is registered as a Git root, but no Git repositories were found there. // Confi... (today 1: 15:46 LF UTF-8 4 spaces Python 3.8 (blind-mailAI-master) (2) test

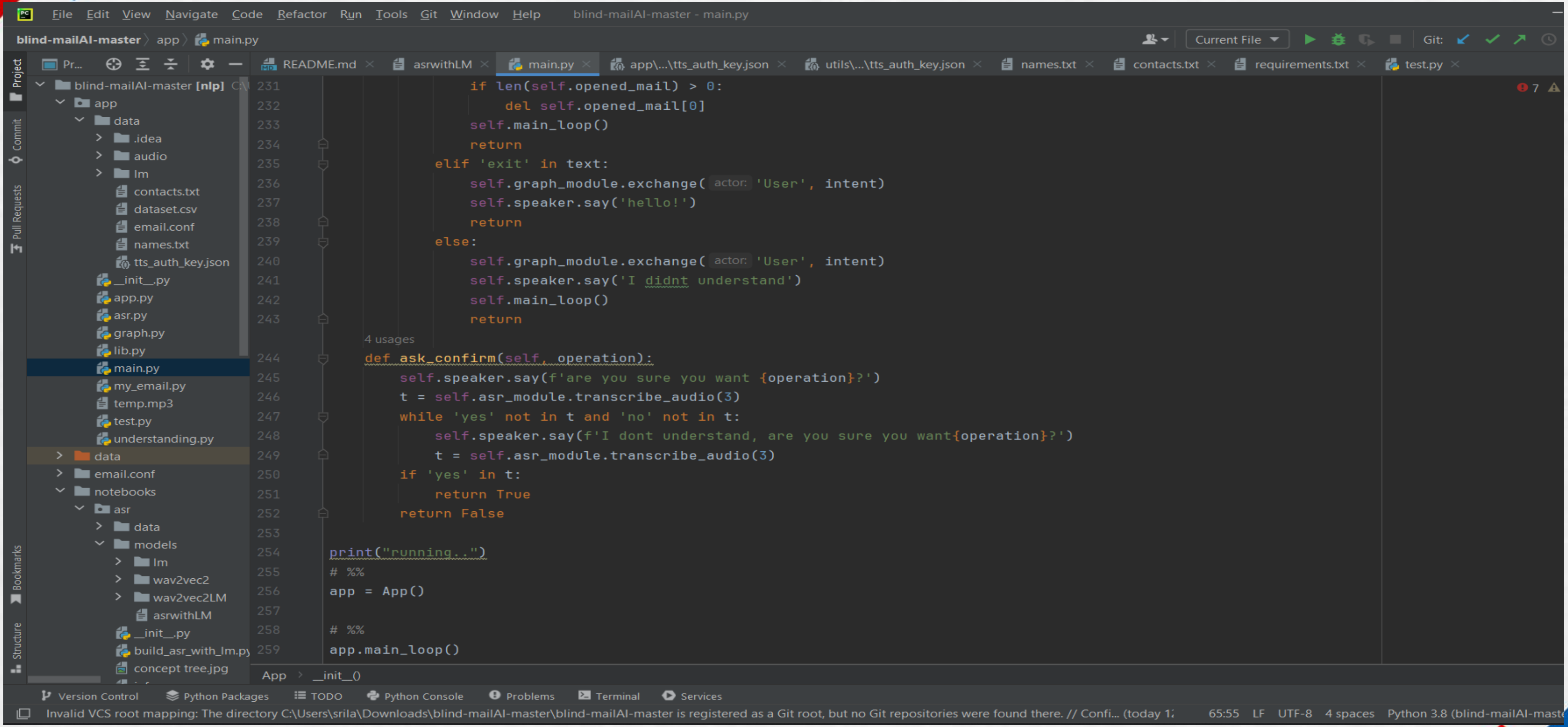
Experiment Environment(3/4)



```
                    elif field == 'object':
                        self.speaker.say('what\' is the subject?')
                        self.graph_module.exchange( actor: 'System',  text: 'what is the object?')
                        mail['object'] = self.asr_module.transcribe_audio()
                        self.graph_module.exchange( actor: 'User', mail['object'],  intent: None,  slots: [('person', mail['object'])])
                        # update graph

                    elif field == 'body':
                        self.speaker.say('what is the body of the email?')
                        self.graph_module.exchange( actor: 'System',  text: 'what is the body of the email?')
                        mail['body'] = self.asr_module.transcribe_audio(10)
                        self.graph_module.exchange( actor: 'User', mail['body'],  intent: None,  slots: [('person', mail['body'])])
                        # update graph

            if self.ask_confirm('send'):
                self.mail_module.dispatch_intent({'intent': intent, 'mail': mail})
                self.speaker.say('mail sent')
            self.main_loop()
            return
        elif intent == 'list_email' or len(self.opened_mail) > 0:

            time = None
            object = None
            person = None
            for slot in slots:
                if slot[0] == 'person':
                    person = slot[1]
                elif slot[0] == 'object':
```

```
Invalid VCS root mapping: The directory C:\Users\srila\Downloads\blind-mailAI-master\blind-mailAI-master is registered as a Git root, but no Git repositories were found there. // Confi... (today 1:   65:55  LF  UTF-8  4 spaces  Python 3.8 (blind-mailAI-
```

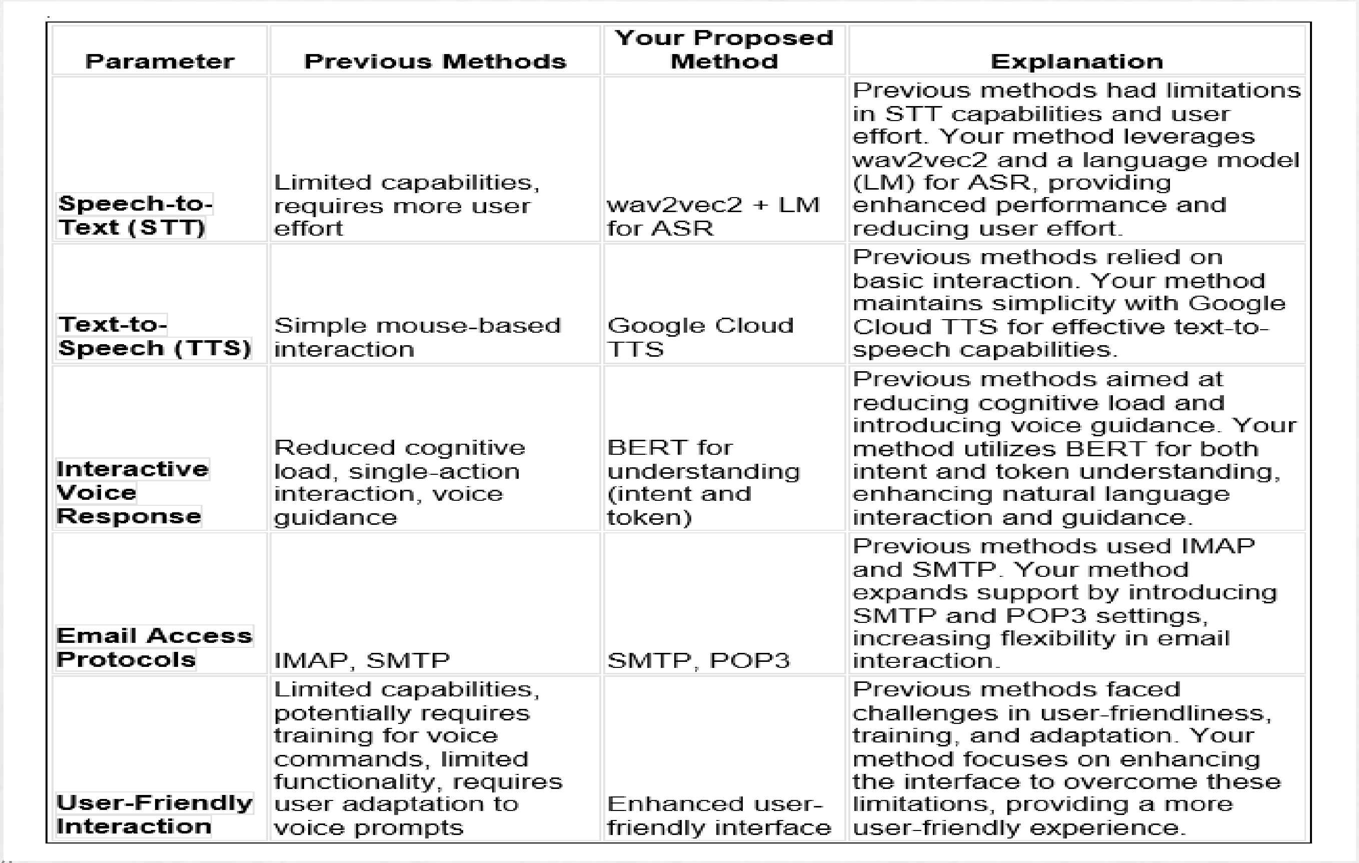
Experiment Environment(4/4)



```
                if len(self.opened_mail) > 0:
                    del self.opened_mail[0]
                self.main_loop()
                return
            elif 'exit' in text:
                self.graph_module.exchange( actor: 'User', intent)
                self.speaker.say('hello!')
                return
            else:
                self.graph_module.exchange( actor: 'User', intent)
                self.speaker.say('I didnt understand')
                self.main_loop()
                return
    4 usages
    def ask_confirm(self, operation):
        self.speaker.say(f'are you sure you want {operation}?')
        t = self.asr_module.transcribe_audio(3)
        while 'yes' not in t and 'no' not in t:
            self.speaker.say(f'I dont understand, are you sure you want{operation}?')
            t = self.asr_module.transcribe_audio(3)
        if 'yes' in t:
            return True
        return False

print("running..")
# %%
app = App()

# %%
app.main_loop()
```

Invalid VCS root mapping: The directory C:\Users\srila\Downloads\blind-mailAI-master\blind-mailAI-master is registered as a Git root, but no Git repositories were found there. // Confi... (today 1:

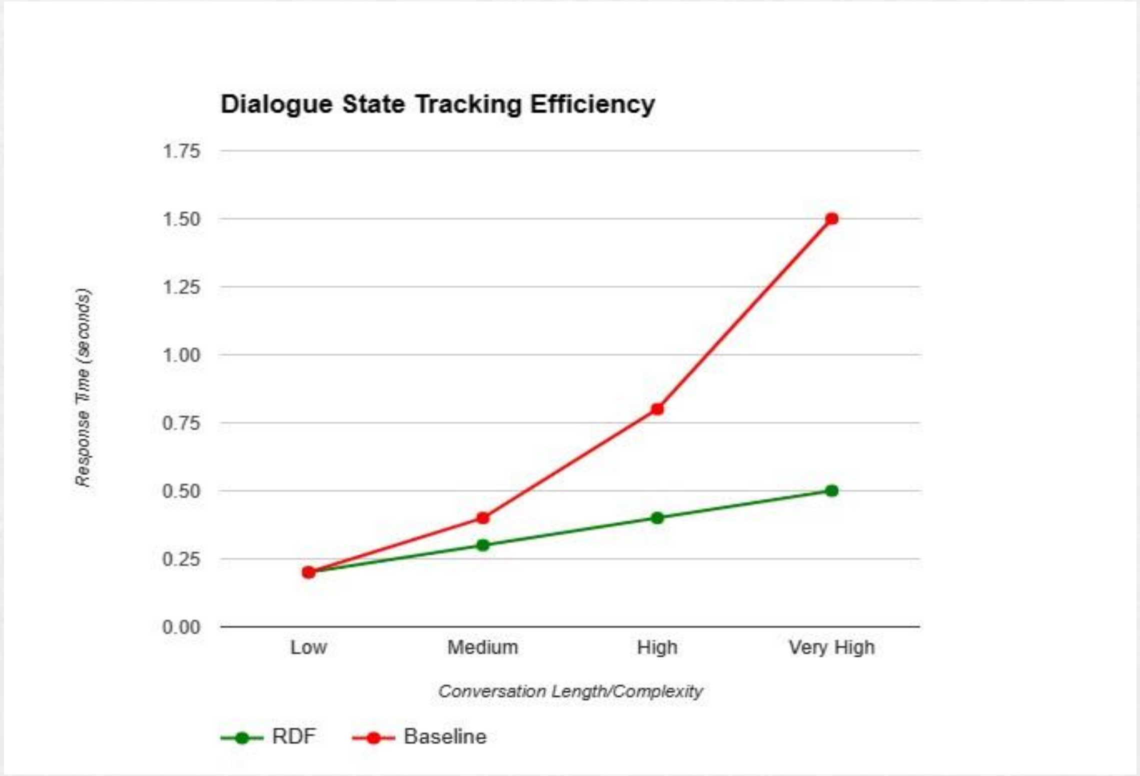
Parameters



Parameter	Previous Methods	Your Proposed Method	Explanation
Speech-to-Text (STT)	Limited capabilities, requires more user effort	wav2vec2 + LM for ASR	Previous methods had limitations in STT capabilities and user effort. Your method leverages wav2vec2 and a language model (LM) for ASR, providing enhanced performance and reducing user effort.
Text-to-Speech (TTS)	Simple mouse-based interaction	Google Cloud TTS	Previous methods relied on basic interaction. Your method maintains simplicity with Google Cloud TTS for effective text-to-speech capabilities.
Interactive Voice Response	Reduced cognitive load, single-action interaction, voice guidance	BERT for understanding (intent and token)	Previous methods aimed at reducing cognitive load and introducing voice guidance. Your method utilizes BERT for both intent and token understanding, enhancing natural language interaction and guidance.
Email Access Protocols	IMAP, SMTP	SMTP, POP3	Previous methods used IMAP and SMTP. Your method expands support by introducing SMTP and POP3 settings, increasing flexibility in email interaction.
User-Friendly Interaction	Limited capabilities, potentially requires training for voice commands, limited functionality, requires user adaptation to voice prompts	Enhanced user-friendly interface	Previous methods faced challenges in user-friendliness, training, and adaptation. Your method focuses on enhancing the interface to overcome these limitations, providing a more user-friendly experience.

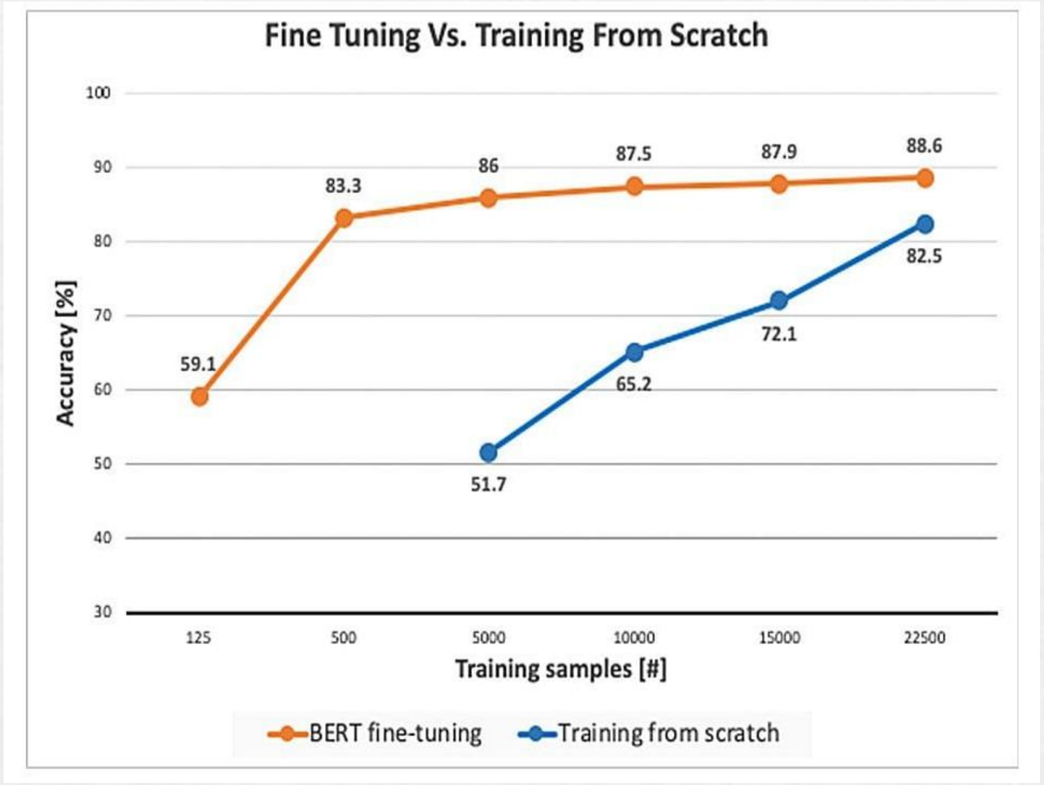
Experiments

Experiment 2:Dialogue State Tracking Efficiency with RDF



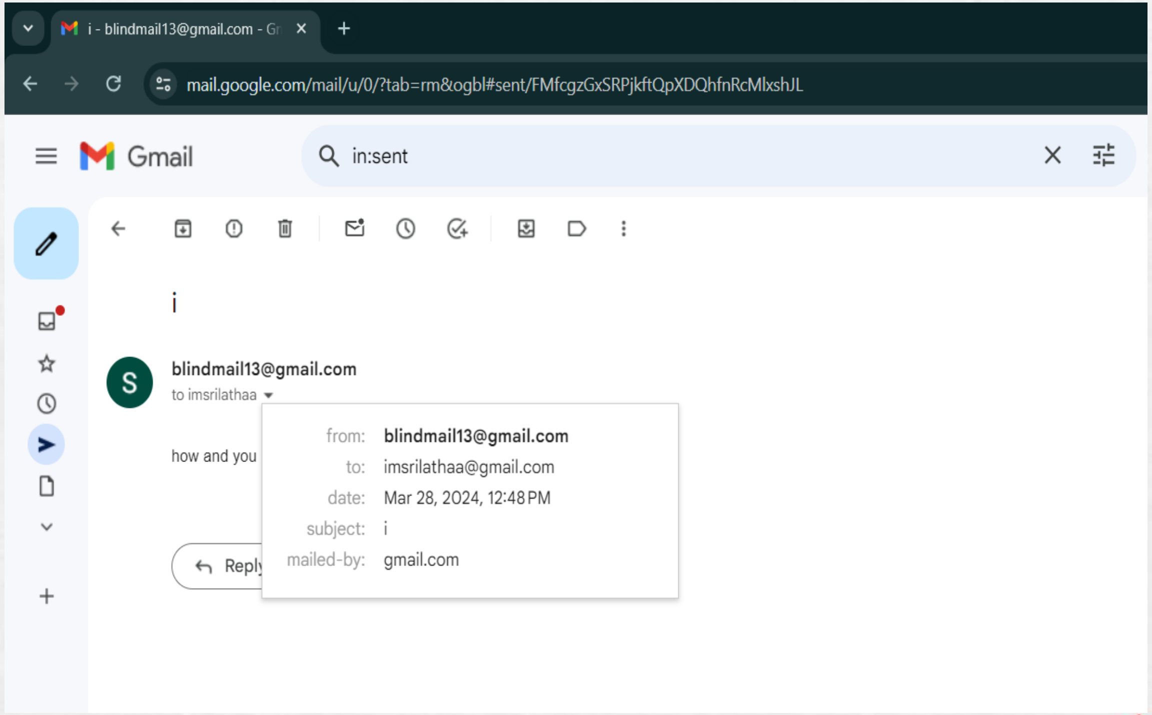
Experiments

Experiment 1: Synthetic Data Size on Speech Recognition Accuracy



Experiment Results

Mail sent from blindmail13@gmail.com to imsrilathaa@gmail.com



mail.google.com/mail/u/0/?tab=rm&ogbl#sent/FMfcgzGxSRPjkftQpXDQhfnRcMlxshJL

i

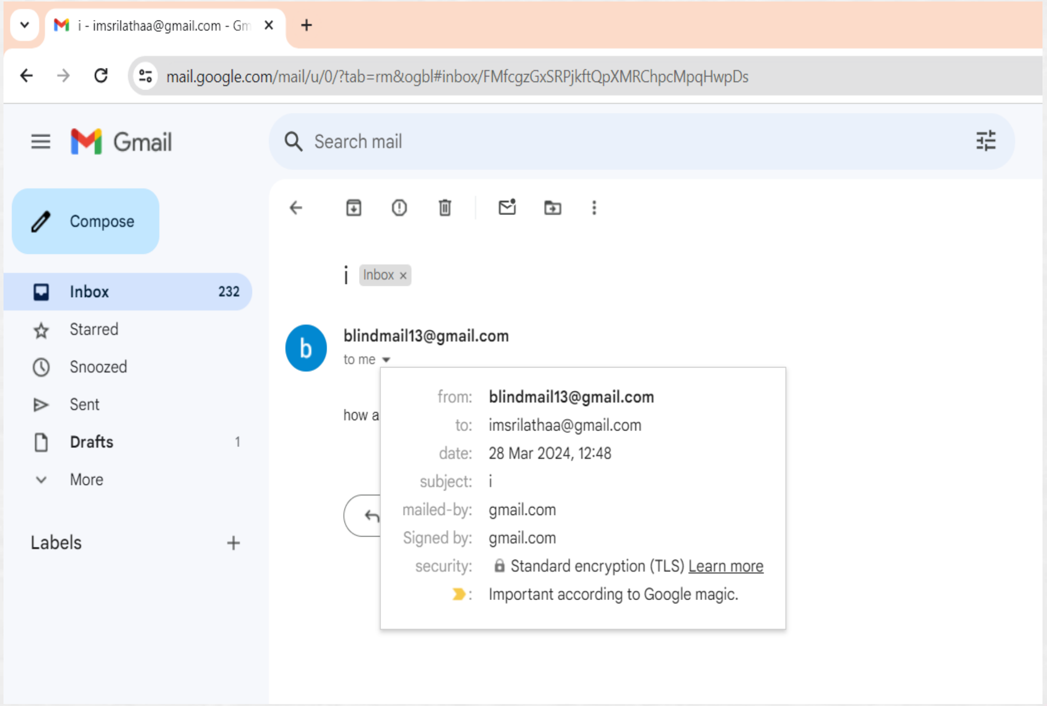
blindmail13@gmail.com
to imsrilathaa

how and you

from:	blindmail13@gmail.com
to:	imsrilathaa@gmail.com
date:	Mar 28, 2024, 12:48 PM
subject:	i
mailed-by:	gmail.com

Experiment Results

Mail received from blindmail13@gmail.com to imsrilathaa@gmail.com



from: blindmail13@gmail.com
to: imsrilathaa@gmail.com
date: 28 Mar 2024, 12:48
subject: i
mailed-by: gmail.com
Signed by: gmail.com
security: Standard encryption (TLS) Learn more
: Important according to Google magic.

References

[1] Ingle, P., Kanade, H., Lanke, A., & Choche, M. (2016). An email architecture for visually impaired people using TTS, STT conversions and IVR technologies. International Journal of Computer Applications, 97(1), 32-38.

[2] Sharma, B., Roshan, B., Gaur, M., & De, P. (2020). Voice-activated email management system with single action at a time. International Journal of Advanced Research in Computer Science, 11(8), 52-58.

[3] Kumar, S., Yogitha, R., & Aishwarya, R. (2021). An accessible email management system for all users using interactive voice response. International Journal of Innovative Technology and Exploring Engineering, 10(7), 521-525.

[4] S Tripathi, Nidhi Kushwaha and Puneet Shukla, "Voice based email system for visually impaired and differently abled", International Journal of Engineering Research & Technology (IJERT), vol. 8, no. 07, July 2019.

[5] Mullapudi Harshasri, Manvam Durga Bhavani and Misra Ravikanth, "Voice Based Email for Blind", International Journal of Innovative Research in Computer Science & Technology (IJIRCST), vol. 9, no. 04, pp. 10-13, July 2021.

[6] Pallavi Tyagi, Tanishka Sharma, Mayank Mittal and Ankit Kumar, "Voice based Email for Physically Challenged", International Research Journal of Engineering and Technology (IRJET), vol. 7, no. 05, May 2020.

Thank you