EPOCH PROBLEM STATEMENT- 02

END TO END SPEECH TRANSLATION

The aim is to translate speech from one language directly to another language using generative models without any text transcription in the latent space.

THEORY

INTRODUCTION-We are addressing s the task of speech-to-speech translation (S2ST): translating speech in one language into speech in another. This application is highly beneficial for breaking down communication barriers between people who do not share common language Specifically, we investigate whether it is possible to train model to accomplish this task directly, without relying on an intermediate text representation. This is in contrast to conventional S2ST systems which are often broken down into three components: automatic speech recognition (ASR), text-to-text machine translation (MT), and text-to-speech (TTS)

ADVANTAGES-Cascaded systems have the potential problem of errors compounding between components, e.g. recognition errors leading to larger translation errors. Direct S2ST models avoid this issue by training to solve the task end-to-end. They also have advantages over cascaded systems in terms of reduced computational requirements and lower inference latency since only one decoding step is necessary, instead of three. In addition, direct models are naturally capable of retaining paralinguistic and nonlinguistic information during translation, e.g. maintaining the source speaker's voice, emotion, and prosody, in the synthesized translated speech. Finally, directly conditioning on the input speech makes it easy to learn to generate fluent pronunciations of words which do not need to be translated, such as names

CHALLENGES-solving the direct S2ST task is especially challenging for several reasons. Fully-supervised end-to-end training requires collecting a large set of input/output speech pairs. Such data are more difficult to collect compared to parallel text pairs for MT, or speech-text pairs for ASR or TTS

SOLUTION

Speech translation is machine learning project. It uses deep neural networks to translate voice from one language to another in real time. It is desktop application that support windows ,linux and mac operations.

Following are the steps taken during the formation of the END TO END SPEECH TRANSLATION Project.

Step:1 import libraries

```
import os
import threading
import tkinter as tk
from gtts import gTTS
from tkinter import ttk
import speech_recognition as sr
from playsound import playsound
```

Step:2 create a dictionary of language name and codes(English=”en” ; Hindi= “hi”).

```
# Create a dictionary of language names and codes
language_codes = {
    "English": "en",
    "Hindi": "hi",
}

language_names = list(language_codes.keys())
```

Step:3 create a dropdown menu for the input and output languages.

```
# Create dropdown menus for the input and output languages

input_lang_label = tk.Label(win, text="Select Input Language:")
input_lang_label.pack()

input_lang = ttk.Combobox(win, values=language_names)
def update_input_lang_code(event):
    selected_language_name = event.widget.get()
    selected_language_code = language_codes[selected_language_name]
```

Step:4 define function update_translation(): & microphone as source.

```
def update_translation():
    global keep_running

    if keep_running:
        r = sr.Recognizer()

        with sr.Microphone() as source:
            print("Speak Now!\n")
            audio = r.listen(source)
```

Step:5 the microphone will record and save the voice & run it in translator

```
        voice = gTTS(translated_text, lang=output_lang.get())
        voice.save('voice.mp3')
        playsound('voice.mp3')
        os.remove('voice.mp3')

        output_text.insert(tk.END, translated_text + "\n")

    except sr.UnknownValueError:
        output_text.insert(tk.END, "Could not understand!\n")
```

Step:6 the translater will run it using function defined run _translator()

```python
def run_translator():
    global keep_running

    if not keep_running:
        keep_running = True
        update_translation_thread = threading.Thread(target=update_translation)
        update_translation_thread.start()
```

Step:7 Now run the program which lead us to the wedpage (url) ;