COP Assignment 2(a)

ANAND SASIKUMAR

2022CS11087

This assignment consists of 3 parts

1. Transcribing data from a pdf file into a text file
2. Transcribing data from a audio file into a text file
3. Transcribing data from a video file into a text file

Part 1.

For this part we used PdfReader module which read through the entire pdf and prints the contents inside the text file. We have defined a function which takes two arguments pdf_path and txt_path which are the pdf file and text file destination respectively. Then inside a function it creates a PdfReader object with the provided pdf file. Then it loops over each page in the pdf file. For each page, it extracts the text and writes it to the text file followed by a newline character. In this program we also note the time taken to complete this process which is then printed to the console.

I also used another module named pdfplumber to read through the entire pdf but the program would terminate in between for large pdf(above 200 pages.) This is because the pdfplumber library loads the entire pdf document into memory when opened. This means the program would be overall quicker but it ends up taking lot of memory which causes the program to crash when it is dealing with large pdfs.

Part 2.

In this part we transcribe data from audio file to text file. For this I used Vosk API. This api is open sourced, free to use and allows unlimited use. In this program we had to import Model and KaldiRecognizer from vosk package which are used for speech recognition, wave module is used to read wav files, finally json modules is used to parse json data produces. For this we have defined a function transcribe_audio which has 2 aruguments named audio_file and model_part. audio_file is the path of audio file and the model_part is the path of the model used. In this program i used vosk-model-en-in-0.5 which can be downloaded from the vosk website. This model is based on Indian english.

In the function the audio file is first opened in read_binary mode using the wave.open function, the audio format should be strictly in wav format. Now the function creates a Model object using the model paath provided, this model is used to transcribe the audio, moreover the function also creates a KaldiRecognizer object which uses the model and audio file's framerate to recognize speech in the audio. Inside the loop data from the audio file is read in chunks(4000 frames at a time) and feeds each chunk to the recognizer. If the recognizer accepts the chunk then the transctibed text which is a json string is appended in the results list. This process continues until there is no more audio to transcribe.

The frame size of 4000 was decided because if the frame size is too large then the memory usage will be too much which might cause the program to crash or make the pc sluggish and if the frame is too small this might lead to lesser efficiency of the program.

In this program we also calculate the time taken the program to finish this process using the time module.

Other models and apis like speech_recognition were not used because only gave limited free uses, unlike vosk api which gives unlimited free use.

Part 3.

In this part we transcribe the audio present in the video and convert it to a text file. First the audio from the video file(.mp4) is extracted and converted to audio file(.wav). This is done inside extract_audio function. This function runs the command using subprocess.run to convert video file to audio file.

The transcribe_audio function is the same as part2. Here the data from audio is converted into a text file and printed inside the given text file.

Then we also calculate time it took to run this program and the end result is printed to the console.

Results

- The course of study pdf takes 101s to transcribe
- video1 takes 29s to transcribe
- video2 took 217s to transcribe

Problems

Since the audio transcribe model used is indian english model so indian english is better suited to it but still it can make certain errors due to various reasons.

- When there are multiple voices at the same time then it can cause inaccurate results at times.
- When the voice is not loud enough then the transcriber might not pick up those sentences.
- This transcriber often has problem with short forms and hence gives wrong result for these.(like iit, phd, etc.)

The model used by me is – vosk_model
Drive - drive