Technical Test - USC ICT Student AI Programmer position

(Submitted by Luv Kumar Aidasani)

Approach and Research:

I searched for various open source libraries for this task. For the purpose of figuring out the number of silences in the audio, I chose to go with Pydub. My main idea to figure out the number of silences was to break the input into various chunks of audio, separated by silence. I assumed several factors related to silence which are mentioned in Assumptions. Once I had the individual chunks with me, I processed those chunks to convert speech to text. I had quite a few options to choose from for Speech Recognition, namely apiai, assemblyai, pocketsphinx, wit, SpeechRecognition. I went on with SpeechRecognition for the task. I processed each audio chunk to calculate the number of words and word frequency.

Libraries Used:

1. ***Pydub (https://github.com/jiaaro/pydub):*** I used this library of Python to break the input audio into various chunks based on silences in the audio (Definition of silence cleared in Assumptions). This was necessary to figure out the number of silences in the audio. I also used this library to get the length of the input audio (needed to calculate words per minute).

2. ***Speech Recognition (https://github.com/Uberi/speech_recognition):*** I used this library of Python to convert each chunk of the audio to speech. This was necessary to find the total number of words (and hence the words per minute). This was necessary to calculate the word frequency as well.

Assumptions:

1. ***Definition of Silence*****:** I have assumed a silence in an audio to be considered as silence if it lasts 1.5 seconds or more, with audio frequency less than or equal to -30dbFS (https://en.wikipedia.org/wiki/DBFS) throughout that duration.
   I used the Pydub library's split_on_silence function for this, in which I provided the following parameters min_silence_len = 1500 (in ms), silence_thresh = -30 (in dbFS).

2. ***Printing the results*****:** I have printed all the results of the task to the console, where the program is run.

3. ***Start and End Silence:*** I ignored the silence at the start and end of the audio in the count for number of silences. I counted only those silences that appeared between spoken words or phrases.

4. In case the Speech Recognition fails to understand the speech of a particular chunk, it prints to the console, "Could not understand audio for chunk number" (followed by the chunk number). Also, an active internet connection is required by Speech Recognition and any such error is handled accordingly and printed to the console.

5. Words per Minute and Word Frequency only involves those chunks whose speech was clear and could be converted to text by the library used.

6. Currently the program takes in input an audio file with **.wav** extension. This can be easily extended to other audio formats as well. But for now, I chose to go with .wav audio files only.

Options to run the Program:

Method 1: GitHub Repository Clone

I have even included various sample audios that I used for testing the code in the SampleAudios folder in the Github Repository.

1. Clone the repository: https://github.com/laidasani/ICTTask.git
2. Switch to ICTTask folder.
3. Follow the steps below method 2 to run the program.

Method 2: Library Installation

To run the program, you'll need to install the libraries using the method specified below:

Library Installation:

1. ***Pydub***: pip install pydub
2. ***Speech Recognition***: pip install SpeechRecognition

Running the Program from Console:

The program logic is in the Python file ICTTask.py. The program takes in input an audio file which should be in .wav format. To run the program, use command line from your Python File's folder and type:

python ICTTask.py PATH_OF_INPUT_AUDIO

- for eg: If there is an audio file named input.wav in the same folder as that of the Python File ICTTask.py. Run command line from that folder and run:
  python ICTTask.py input.wav

- If the .wav audio file is in a different folder, please provide the entire path of the file in PATH_OF_INPUT_AUDIO
  For eg:
  Windows: python ICTTask.py C:/Users/laida/Desktop/ICTTask/New/input1.wav
  Mac and Linux: python ICTTask.py /Users/laida/Desktop/ICTTask/New/input1.wav

On running the program, a runtime warning would be shown, which is related to ffmpeg. **Even with the warning the program runs smoothly.**

To remove the warning:

- For Mac, please write the following command to console-> brew install ffmpeg
- For Linux, please write the following command to console-> apt-get install ffmpeg
- For Windows, please install ffmpeg from https://ffmpeg.zeranoe.com/builds/, and set System Path Variable to the bin folder of ffmpeg.