SpeechRecognition for Python

What is it?

SpeechRecognition is a python library for performing speech-to-text operations. The library is compatible with several different APIs such as Google Cloud Speech API, IBM Speech to Text, and Microsoft Bing Voice Recognition. The SpeechRecognition library provides functions for processing audio files that make getting started much easier than other libraries.

The easiest API to use right away is the Google Web Speech API, since the default API key is hard-coded into the library. While it is the easiest to use, the Google Web Speech API does have some limitations. The API limits requests to approximately 10MB, so depending on the quality of the given audio file, the length may be an issue. The API may also raise an error if it encounters too much unintelligible noise. For files with a slight amount of noise, SpeechRecognition provides a function for adjusting accordingly. However, for audio files that have a lot of complex background noise, such as music, some form of external preprocessing is necessary before using this library.

This library will be most useful for transcribing speech from high-quality, low-noise recordings. It also tends to be more accurate for shorter durations of audio input. If the intended task is to transcribe lyrics from music, for example, which typically has a multiple-minute duration and a lot of complex background noise, SpeechRecognition will likely not be a sufficient tool on its own

Setup

- Windows setup
    - Install required modules
        - pip3 install SpeechRecognition pydub
    - If you get ffmpeg not found warning
        - Download ffmpeg from https://ffmpeg.org/
        - Move the three .exe files into the same directory as transcribe.py
    - If you get FLAC conversion utility not available error
        - Download flack from https://xiph.org/flac/download.html
        - Download the windows .zip
        - Copy flac.exe from win64 and paste it into C:\Windows\System32
        - Remove .exe from the file name
        - https://stackoverflow.com/questions/65939571/installing-flac-command-line-tool-on-windows
    - Acquire an mp3 file
        - You'll need the file path in your call to the 'AudioSegment.from_mp3' function below

Quick-start Tutorial

- In a new directory, create a file ‘transcribe.py’
- Add import statements

  ```
  import speech_recognition as sr
  from os import path
  from pydub import AudioSegment
  ```
- Prepare .wav file

  ```
  sound=AudioSegment.from_mp3(r"c:/path/to/your/file/here.mp3")
  sound.export("transcript.wav",format="wav")
  AUDIO_FILE="transcript.wav"
  ```
- Use the Recognizer

  ```
  with sr.AudioFile(AUDIO_FILE) as source:
      audio=r.record(source)
      print(r.recognize_google(audio))
  ```
- The transcribed words should be printed to the console

Additional Items

- If you want to see all of the alternative transcriptions, use the ‘show_all’ parameter

  ```
  print(r.recognize_google(audio, show_all=True))
  ```
- If your audio has some background noise, the recognizer can be adjusted to account for it with the ‘adjust_for_ambient_noise’ function before recording

  ```
  r.adjust_for_ambient_noise(source, duration=5)
  audio=r.record(source)
  ```
  - The function will consume the first second of the audio to measure the background noise level by default, but that amount of time can be changed using the ‘duration’ parameter
- If your audio is too long, or you only need to transcribe a certain part of the audio
  - Use the ‘offset’ parameter to choose how many seconds into the audio the transcription will begin

  ```
  audio=r.record(source, offset=10)
  ```
  - Use the ‘duration’ parameter to choose how many seconds of the audio you want transcribed (starting from 0:00 unless an offset is specified)

  ```
  audio=r.record(source, duration=15)
  ```
- If your audio has too much background noise, or complex background noise, you may need to pre-process your audio or choose a different solution
  - librosa is a Python package for music and audio analysis
    - It may be possible to first extract the vocals from a song using librosa’s functions and then transcribe the vocals using SpeechRecognition

More Resources

- General introductory guide to SpeechRecognition
  - https://pdf.co/blog/transcribe-speech-recordings-to-text-python#:~:text=Transcribing%20audio%20recordings%20to%20texts,into%20humanly%20readable%20textual%20format.
- Another guide to SpeechRecognition
  - https://realpython.com/python-speech-recognition/
- SpeechRecognition list on PyPI
  - https://pypi.org/project/SpeechRecognition/
- Example code for transcribing an audio file with any of the API options:
  - https://github.com/Uberi/speech_recognition/blob/master/examples/audio_transcribe.py
- Vocal separation using librosa
  - https://librosa.org/librosa_gallery/auto_examples/plot_vocal_separation.html