SpeechRecognition Python library

SPOKEN LANGUAGE PROCESSING IN PYTHON

Daniel Bourke

Machine Learning Engineer/YouTube Creator

datacamp

Why the SpeechRecognition library?

Some existing python libraries

- CMU Sphinx
- Kaldi
- **SpeechRecognition**
- Wav2letter++ by Facebook

Getting started with SpeechRecognition

Install from PyPi:

```
$ pip install SpeechRecognition
```

- Compatible with Python 2 and 3
- We'll use Python 3

Using the Recognizer class

```
# Import the SpeechRecognition library
import speech_recognition as sr
# Create an instance of Recognizer
recognizer = sr.Recognizer()
# Set the energy threshold
recognizer.energy_threshold = 300
```

Using the Recognizer class to recognize speech

- `Recognizer` class has built-in functions which interact with speech APIs
  - `recognize_bing()`
  - `recognize_google()`
  - `recognize_google_cloud()`
  - `recognize_wit()`

Input: `audio_file`

Output: transcribed speech from `audio_file`

SpeechRecognition Example

- Focus on `recognize_google()`
- Recognize speech from an audio file with SpeechRecognition:

```
# Import SpeechRecognition library
import speech_recognition as sr
# Instantiate Recognizer class
recognizer = sr.Recognizer()
# Transcribe speech using Goole web API
recognizer.recognize_google(audio_data=audio_file
                            language="en-US")
```

```
Learning speech recognition on DataCamp is awesome!
```

Your turn!

SPOKEN LANGUAGE PROCESSING IN PYTHON

Reading audio files with SpeechRecognition

SPOKEN LANGUAGE PROCESSING IN PYTHON

Daniel Bourke
Machine Learning Engineer/YouTube Creator

datacamp

The AudioFile class

```
import speech_recognition as sr
# Setup recognizer instance
recognizer = sr.Recognizer()
# Read in audio file
clean_support_call = sr.AudioFile("clean-support-call.wav")
# Check type of clean_support_call
type(clean_support_call)
```

```
<class 'speech_recognition.AudioFile'>
```

From AudioFile to AudioData

```
recognizer.recognize_google(audio_data=clean_support_call)
```

```
AssertionError: ``audio_data`` must be audio data
```

```
# Convert from AudioFile to AudioData
with clean_support_call as source:
    # Record the audio
    clean_support_call_audio = recognizer.record(source)
# Check the type
type(clean_support_call_audio)
```

```
<class 'speech_recognition.AudioData'>
```

Transcribing our AudioData

```
# Transcribe clean support call
recognizer.recognize_google(audio_data=clean_support_call_audio)
```

```
hello I'd like to get some help setting up my account please
```

Duration and offset

- `duration` and `offset` both `None` by default

```
# Leave duration and offset as default
with clean_support_call as source:
    clean_support_call_audio = recognizer.record(source,
                                                 duration=None,
                                                 offset=None)
```

```
# Get first 2-seconds of clean support call
with clean_support_call as source:
    clean_support_call_audio = recognizer.record(source,
                                                 duration=2.0)
```

```
hello I'd like to get
```

Let's practice!

SPOKEN LANGUAGE PROCESSING IN PYTHON

Dealing with different kinds of audio

SPOKEN LANGUAGE PROCESSING IN PYTHON

Daniel Bourke
Machine Learning Engineer/YouTube Creator

datacamp

What language?

```
# Create a recognizer class
recognizer = sr.Recognizer()
# Pass the Japanese audio to recognize_google
text = recognizer.recognize_google(japanese_good_morning,
                                   language="en-US")
# Print the text
print(text)
```

```
Ohio gozaimasu
```

What language?

```
# Create a recognizer class
recognizer = sr.Recognizer()
# Pass the Japanese audio to recognize_google
text = recognizer.recognize_google(japanese_good_morning,
                                   language="ja")
# Print the text
print(text)
```

```
?????????
```

Non-speech audio

```
# Import the leopard roar audio file
leopard_roar = sr.AudioFile("leopard_roar.wav")
# Convert the AudioFile to AudioData
with leopard_roar as source:
    leopard_roar_audio = recognizer.record(source)
# Recognize the AudioData
recognizer.recognize_google(leopard_roar_audio)
```

```
UnknownValueError:
```

Non-speech audio

```
# Import the leopard roar audio file
leopard_roar = sr.AudioFile("leopard_roar.wav")
# Convert the AudioFile to AudioData
with leopard_roar as source:
    leopard_roar_audio = recognizer.record(source)
# Recognize the AudioData with show_all turned on
recognizer.recognize_google(leopard_roar_audio,
                            show_all=True)
```

```
[]
```

Showing all

```
# Recognizing Japanese audio with show_all=True
text = recognizer.recognize_google(japanese_good_morning,
                                   language="en-US",
                                   show_all=True)
# Print the text
print(text)
```

```
{'alternative': [{'transcript': 'Ohio gozaimasu', 'confidence': 0.89041114},
  {'transcript': 'all hail gozaimasu'},
  {'transcript': 'ohayo gozaimasu'},
  {'transcript': 'olho gozaimasu'},
  {'transcript': 'all Hale gozaimasu'}],
 'final': True}
```

Multiple speakers

```
# Import an audio file with multiple speakers
multiple_speakers = sr.AudioFile("multiple-speakers.wav")
# Convert AudioFile to AudioData
with multiple_speakers as source:
    multiple_speakers_audio = recognizer.record(source)
# Recognize the AudioData
recognizer.recognize_google(multiple_speakers_audio)
```

```
one of the limitations of the speech recognition library is that it doesn't
recognise different speakers and voices it will just return it all as one block
of text
```

Multiple speakers

```python
# Import audio files separately
speakers = [sr.AudioFile("s0.wav"), sr.AudioFile("s1.wav"), sr.AudioFile("s2.wav")]
# Transcribe each speaker individually
for i, speaker in enumerate(speakers):
    with speaker as source:
        speaker_audio = recognizer.record(source)
    print(f"Text from speaker {i}: {recognizer.recognize_google(speaker_audio)}")
```

```
Text from speaker 0: one of the limitations of the speech recognition library
Text from speaker 1: is that it doesn't recognise different speakers and voices
Text from speaker 2: it will just return it all as one block a text
```

Noisy audio

- If you have trouble hearing the speech, so will the APIs

```
# Import audio file with background nosie
noisy_support_call = sr.AudioFile(noisy_support_call.wav)
with noisy_support_call as source:
    # Adjust for ambient noise and record
    recognizer.adjust_for_ambient_noise(source,
                                        duration=0.5)
    noisy_support_call_audio = recognizer.record(source)
# Recognize the audio
recognizer.recognize_google(noisy_support_call_audio)
```

```
hello ID like to get some help setting up my calories
```

Let's practice!

SPOKEN LANGUAGE PROCESSING IN PYTHON