

Speech Recognition In Python

Project members:

1.190638

2.190625

3.190616

4.190640

5.190632

Name Of The Project: Speech Recognition In Python

Abstract:

This speech processing project aims to develop a system that can transcribe spoken language and perform various language processing tasks on the transcribed text. The system will use advanced speech recognition technology and natural language processing techniques to extract useful information from transcribed text, such as sentiment analysis and entity recognition. The ultimate goal is to create an accurate and efficient system that can be applied to a range of real-world applications, such as voice-based search and speech-to-text transcription for those with hearing impairments.

Objectives:

Some potential objectives for a speech recognition project in Python could include:

- Developing a proof-of-concept application that can recognize and transcribe simple voice commands, such as "turn on the lights" or "play music"
- Building a more complex speech recognition system that can recognize natural language input and respond with appropriate actions or responses
- Optimizing a speech recognition system to work well in noisy environments or with non-standard accents and dialects
- Integrating speech recognition capabilities into an existing Python application or workflow, such as a chatbot or virtual assistant.

Theory:

Speech processing is a field of study that encompasses a range of techniques for analyzing and processing speech signals.

The goal of this speech processing project is to develop a system that can automatically transcribe spoken language and perform various language processing tasks on the transcribed text. The system will use state-of-the-art speech recognition technology to transcribe audio recordings of human speech into text, and will then apply natural language processing techniques to extract meaningful information from the transcribed text.

The project will involve several stages, including

Some build in python library for making some Speech processing technique which maintains some model like

acoustic modeling, language modeling, and decoding.

These are the library:

1. speech recognition
2. Google Translator
3. pyttsx3
4. pywhatkit
5. web browser
6. ctime

Speech recognition:

Speech Recognition incorporates computer science and linguistics to identify spoken words and converts them into text. It allows computers to understand human language.

Speech recognition is a machine's ability to listen to spoken words and identify them. You can then use speech recognition in Python to convert the spoken words into text, make a query or give a reply.

Google Translator:

This uses the Google Translate Ajax API to make calls to such methods as detect and translate.

pyttsx3:

This is a text-to-speech conversion library in Python

PyWhatKit: PyWhatKit is a Python library with various helpful features. It's easy-to-use and does not require you to do any additional setup. Currently, it is one of the most popular library for WhatsApp and YouTube automation. New updates are released frequently with new features and bug fixes.

Webbrowser:

The webbrowser module provides a high-level interface to allow displaying web-based documents to users. Under most circumstances, simply calling the `open()` function from this module will do the right thing.

Ctime: `ctime()` method converts a time in seconds since the epoch to a string in local time.

The acoustic modeling stage will involve training a deep neural network to recognize speech sounds and patterns

We can applying various language processing techniques such as part-of-speech tagging, sentiment analysis, and entity recognition to extract useful information from the transcribed text.

The ultimate goal is to develop a system that can transcribe and process spoken language with a high degree of accuracy and efficiency, and that can be applied to a range of real-world applications, such as voice-based search, language learning, and speech-to-text transcription for people with hearing impairments.

Source Code:

```
# library import
import time
from time import ctime
from deep_translator import GoogleTranslator
import speech_recognition as sr
import pyttsx3
import pywhatkit
import webbrowser

listener = sr.Recognizer()
engine = pyttsx3.init()
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)

def talk(text):
    engine.say(text)
    engine.runAndWait()

def take_command(ask =False):
    try:
        with sr.Microphone() as source:
            print('listening....')
            if ask:
                print(ask)

            voice = listener.listen(source)
            command = listener.recognize_google(voice)
            command = command.lower()
```

```
except sr.UnknownValueError:
    print("Google Speech Recognition could not understand audio")
except sr.RequestError as e:
    print("Could not request results from Google Speech Recognition service; {0}".format(e))
return command
```

```
def run_assistant(command):
    if 'music' in command:
        music = take_command('what kind of music do you like?')
        song = command.replace('music', music)
        talk(song)
        pywhatkit.playonyt(song)
    if 'send a message' in command:
        pywhatkit.sendwhatmsg("+8801749034060", "Hi", 11, 12, True, 2)
    if 'what is your name' in command:
        talk('my name is sakib')
        print('my name is sakib')
    if 'what time is it' in command:
        talk(ctime())
        print(ctime())
    if 'who is your project manager' in command:
        talk('shoaib Sakib')
        print('Shoaib Sakib')
    if 'google find' in command:
        google = take_command('what do you want to search for?')
```

```
#
```

```

        url = 'https://google.com/search?q=' + google
        webbrowser.get().open(url)
        print('here is search' + google)

    if 'wikipedia search' in command:
        wiki = take_command('please select something?')

        url = 'https://en.wikipedia.org/wiki/' + wiki
        webbrowser.get().open(url)
        print('here is search' + wiki)

    if 'translation' in command:
        translet = take_command('please say something')
        translated = GoogleTranslator(source='auto',
target='bn').translate(translet)
        print(translated)

    if 'find map' in command:
        location = take_command('what is the location?')
        url = 'https://google.nl/maps/place/' + location +
'/' + '&'
        webbrowser.get().open(url)
        print('here is the location' + ' ' + location)
    if 'close' in command:
        exit()

time.sleep(1)
talk('listening')

while 1:
    command = take_command()
    run_assistant(command)

```

Result and Discussion:

Example:

listening....

please say something

result2:

```
{ 'alternative': [{ 'confidence': 0.88687539, 'transcript': 'how are you' }],  
  'final': True}
```

আপনি কেমন আছেন