```python
#*******************************************
# etri+google 한글,영어 음성인식 및 STT 테스트
#*******************************************
# pydub API 문서 - https://github.com/jiaaro/pydub/blob/master/API.markdown
# 설치할 패키지들
# pip install SpeechRecognition gTTS
# pip install playsound==1.2.2        # version 1.3.0은 오류 발생
# pip install pydub librosa
# conda install pyAudio matplotlib ffmpeg

# STT관련
import speech_recognition as sr
from pydub import AudioSegment, silence
import librosa, librosa.display
import matplotlib.pyplot as plt
import numpy as np

# TTS 관련
import playsound
from gtts import gTTS

# web browser 구동 관련
import webbrowser

import os, sys, random
from time import ctime

import urllib3
import json
import base64

# ETRI 서버 및 개인 accesskey 관련 정보
accessKey = "3ab9d53f-cce1-4d82-b207-5593e15407e0"
audioFilePath = "KOR_M_RM0276MKDH0135.pcm"

#*********************************************************
# ETRI 한글 음성인식과 발음 평가
#*********************************************************
def sentence_from_etri(wavFile) :
    make_melspectrogram(wavFile,'original_voice')

    # 16khz mono 오디오 데이터로 읽기
    wavAudio = AudioSegment.from_file(wavFile, format="wav").set_frame_rate(16000)

    print('앞쪽 조용한 구간 시작점: ',silence.detect_silence(wavAudio.silent(10000),
                                                  silence_thresh=-50))

    print('frame_rate,channels,sample_width:',wavAudio.frame_rate,wavAudio.channels,wavAudio.sample_width)
    print('loudness(rms), peak값:',wavAudio.rms,wavAudio.max)

    # 헤더를 제외한 실제 음성 데이터 부분만 추출
    rawData = wavAudio.raw_data
    audioContents = base64.b64encode(rawData).decode("utf8")
    # .wav 파일로 부터 실행시 사용
    '''
    audioFile = open(audioFilePath, "rb")
    audioContents = base64.b64encode(audioFile.read()).decode("utf8")
    audioFile.close()
    '''

    # 음성인식
    http = urllib3.PoolManager()
    openApiURL = "http://aiopen.etri.re.kr:8000/WiseASR/Recognition"
    requestJson = {
        "access_key": accessKey,
```

```
66          "argument": {
67              "language_code": 'korean',      # languageCode
68              "audio": audioContents
69          }
70      }
71
72      response = http.request(
73          "POST",
74          openApiURL,
75          headers={"Content-Type": "application/json; charset=UTF-8"},
76          body=json.dumps(requestJson)
77      )
78
79      print("[responseCode_음성인식] " + str(response.status))
80      print("[responseData_음성인식]" + str(response.data,"utf-8"))
81
82      sentence_str = str(response.data,"utf-8")
83      sentence_start_pos = sentence_str.find('recognized')+len('recognized: ')+2
84      sentence_end_pos = sentence_str.find('"',sentence_start_pos+1)
85      sentence = sentence_str[sentence_start_pos:sentence_end_pos]
86
87      # 발음 평가
88      openApiURL = "http://aiopen.etri.re.kr:8000/WiseASR/PronunciationKor"   # 한국어
89      #openApiURL = "http://aiopen.etri.re.kr:8000/WiseASR/Pronunciation"      # 영어
90      script = "이름이 뭐에요?"
91      requestJson = {
92          "access_key": accessKey,
93          "argument": {
94              "language_code": 'korean',      # languageCode
95              "script": sentence,         # script,
96              "audio": audioContents
97          }
98      }
99
100     response2 = http.request(
101         "POST",
102         openApiURL,
103         headers={"Content-Type": "application/json; charset=UTF-8"},
104         body=json.dumps(requestJson)
105     )
106
107     print("[responseCode_발음평가] " + str(response2.status))
108     print("[responseData_발음평가]" + str(response2.data,"utf-8"),"Wn")
109
110     score_str = str(response2.data,"utf-8")
111     score_pos = score_str.find('score')+len('score:')+1
112     score = score_str[score_pos:score_pos+6]
113
114     return sentence,score
115
116 #************************************************************
117 # 한글,영어음성 문장을 음성으로 입력받아 Text로 출력하는 음성인식
118 #************************************************************
119 def sentence_from_audio( lang, direction='How can I help You? [say "goodbye" to finish]',
    soundFrom='MIC') :
120     COMPARE_RECOGNIZERS = False
121
122     print("언어: ",lang)
123
124     recog = sr.Recognizer()                     # Recognizer 인스턴스 recog 생성
125     sentence = []                               # 인식기별로 인식결과 문장들 저장
126
127     with sr.Microphone() as source:
128         recog.adjust_for_ambient_noise(source, duration=1)
129         while True:
130             print("Wn",direction)
```

```
131             audio = recog.listen(source) # 오디오 데이터 추출
132 
133                 # 사용자가 말한 음성을 wave파일로 저장
134             with open("spokenWav.wav","wb") as fpWave:
135                 fpWave.write(audio.get_wav_data())
136 
137             #print("audio type: ",type(audio)," data: ",audio)
138             try :
139                 if soundFrom == 'MIC': # 마이크로 입력한 음성 인식
140                     if lang == 'en' :
141                         result = recog.recognize_google(audio)   # 구글 WebSpeech 음성인식 이용, 인식
142                     elif lang == 'ko' :
143                         result, score = sentence_from_etri("spokenWav.wav")
144 
145                     sentence.append( result )
146                     print("===> you spoke: ", sentence[0], "["+score+" 점]")
147                 else : # 음성 파일로 부터 구글 Webspeech 이용 인식
148                     audio_file = sr.AudioFile('spokenWav.wav')
149                     with audio_file as source:
150                         audio = recog.record(source)
151                         sentence.append( recog.recognize_google(audio) ) # 구글 WebSpeech 음성인식 이용, 인
    식
152                         print("The FILE says: ", sentence[0])
153 
154                 '''
155                 if COMPARE_RECOGNIZERS == True:
156                     sentence.append( recog.recognize_bing(audio, key) )         # Microsoft 음성인식 이용
157                     sentence.append( recog.recognize_ibm(audio, username, password) ) # IBM 음성인식 이용
158                     sentence.append( recog.recognize_houndify(audio client_id, client_key) ) # Houndify 음
    성인식 이용
159                     sentence.append( recog.recognize_wit(audio, key) ) # Wit.ai 음성인식 이용
160                         # 아래 인식엔진들은 SpeechRecognition외에 별도 패키지 추가가 필요
161                     sentence.append( recog.recognize_google_cloud(audio, ...) ) # Google Cloud 이용
162                     sentence.append( recog.recognize_sphinx(audio, ...) ) # CMYU Sphinx 이용, Offline 이용
    가능
163                     print('* 다른 인식기 인식 결과 *')
164                     for ndx in range(1,len(sentence)):
165                         print(ndx, "번째 인식기 결과: ", sentence[ndx])
166                 '''
167                 return sentence[0]
168 
169             except sr.UnknownValueError: # 인식결과 오류 발생시
170                 print('Please say again')
171 
172             except sr.RequestError as e: # 서버접속, 사용한도 초과, 인터넷연결 등의 문제시 발생
173                 print('오류'.format(e))
174 
175 #************************************************************
176 # 구글 TTS를 이용하여, TEXT를 음성파일로 저장후 소리 출력
177 #************************************************************
178 def googleTTS(sentence, language='en') :
179     tts = gTTS(text=sentence, lang=language) # language로 설정한 언어와 문장으로 tts 인스턴스 생성
180 
181     r = random.randint(1,1000000)   # 파일명 난수로 임의로 설정
182     audioFile = './audio_' +str(r) + '.mp3'
183 
184     #print('audio_file_name: ',audioFile)
185     tts.save(audioFile)             # 문장을 음성파일로 저장
186     playsound.playsound(audioFile)  # 음성파일을 불러서 음성 출력
187 
188     print(sentence)                 # 문장 내용 출력 후 음성 파일은 지움
189     #os.remove(audioFile)
190 
191 #************************************************************
192 # 사용자가 입력한 문장에 따라 몇가지만 반응하는 함수
193 #************************************************************
```

```
def respond(sentence=''):
    searchWords = ['search', 'find', 'i want to know', 'open web browser']
    nameWords = ['your name', 'should i call']
    positiveAnswerWords = ['you free', 'can i call you', 'is it okay']

    if 'time' in sentence: # time 단어가 들어있는 경우
        print (ctime())
        googleTTS(ctime(), 'en')
        return 'ok'

    for phrase in nameWords :
        if phrase in sentence: # 이름을 묻는 질문인 경우
            txt = 'My name is AI Kim'
            txtKor = '저는 AI 로봇입니다'
            print(txt)
            googleTTS(txtKor, 'ko')
            return 'ok'

    for phrase in positiveAnswerWords :
        if phrase in sentence: # 긍정적 답을 하는게 좋은 질문인 경우
            googleTTS('Sure, Feel free to say anything', 'en')
            return 'ok'

    for phrase in searchWords :
        if phrase in sentence: # search 단어가 들어있는 경우, 웹브라우저로 검색결과 실행
            search_data = sentence_from_audio('en','What do you want to search for?', 'MIC')
            url = 'https://google.com/search?q=' + search_data
            webbrowser.get().open(url)
            str = 'Search Result for '+ search_data
            print(str)
            return 'ok'

    return 'NO_MATCH'

#*********************************************************
# ETRI 한글 질의 응답
#*********************************************************
def answer_from_etri(question) :
    # ETRI 서버로 전송할 데이터를 json형태로 준비
    openApiURL_QnA = "http://aiopen.etri.re.kr:8000/WiseQAnal"

    requestJson = {
        "access_key": accessKey,
        "argument": {
            "text": question
        }
    }

    http = urllib3.PoolManager()
    response = http.request(
        "POST",
        openApiURL_QnA,
        headers={"Content-Type": "application/json; charset=UTF-8"},
        body=json.dumps(requestJson)
    )

    #print("[responseCode] " + str(response.status))

    return response.status,str(response.data,"utf-8")

#*********************************************************
# Make MelSpectrogram
#*********************************************************
def make_melspectrogram(wavFilename,filename):
    spokenWaveData, srVal = librosa.load(wavFilename, sr=None)  # 음성파일을 읽어서 spokenWaveData에 저장
    spokenWaveData *= 1.0
```

```
    plt.interactive(False)
    plt.figure()
    S = librosa.feature.melspectrogram(y=spokenWaveData, sr=srVal, n_mels=64)
    S_dB = librosa.power_to_db(S, ref=np.max)
    fig = plt.figure(figsize = [0.8, 0.8])
    ax = fig.add_subplot(1,1,1)
    ax.axes.get_xaxis().set_visible(False)
    ax.axes.get_yaxis().set_visible(False)
    ax.set_frame_on(False)
    librosa.display.specshow(S_dB)
    melSavePath = './'
    melSavePath += filename+'.jpg'
    print(melSavePath)
    plt.savefig(melSavePath, dpi=400, bbox_inches='tight',pad_inches=0)    # 이미지파일로 출력
    plt.close('all')

#************************************************************
# 직접 실행시 프로그램 시작 부분
#************************************************************
if __name__ == '__main__':

    lang = 'en'
    question = ''
    if len(sys.argv) > 1:
        if sys.argv[1] == 'ko' or sys.argv[1] == 'en':
            lang = sys.argv[1]
            googleTTS('문장을 말하세요','ko')
        else :
            question = sys.argv[1]
    else :
        print('[1]한영 음성인식 사용법: Python etri.py ko/en')
        print('[2]한글 질의응답 사용법: Python etri.py 질문문장')

    while True:

        if question == '' : # 음성인식
            result_sentence = sentence_from_audio(lang) # 음성인식하여 TEXT(문자) 값을 가져옴

            if 'goodbye' in result_sentence:
                googleTTS('bye bye','en')
                break
            else : # 종료명령이 아니면
                if respond(result_sentence) == 'NO_MATCH' and sys.argv[1]=='en':
                    print("Sorry, I didn't get it. Would you say again?")
        else : # 질의 응답
            result_status, result_sentence = answer_from_etri(question)
            if result_status == 200 :   # 답 문장을 받은 경우
                print('ETRI 리턴 값: Wn',result_sentence,'Wn')

                start_pos = result_sentence.find('vTitles')    # 답 위치 항목 찾기
                last_pos = result_sentence.find('vQTopic',start_pos)    # 답 위치 항목 찾기

                if start_pos == -1 :            # 문장을 받았으나 답이 포함되지 않은 경우
                    print('받은 답이 없습니다 !')
                else :
                    answer_no = 1;
                    while True :
                        start_pos = result_sentence.find('strExplain',start_pos+1)
                        if start_pos == -1 or start_pos > last_pos:
                            break
                        start_pos += len('strExplain: ')
                        end_pos = result_sentence.find('dWeightEn',start_pos)
                        answer =result_sentence[start_pos:end_pos-2]
                        print('ETRI 답 [',answer_no,']Wn',answer,'Wn')
                        answer_no += 1
```

```
326             question = input('*** 질문을 입력하세요[종료시엔 end 라고 입력하세요] : ')
327             if question == 'end' :
328                 break
```