**User’s Manual for Speech Analysis program, Ver. 1.0**

2020.9.

Bum Joon Park ([qkrskaqja@snu.ac.kr](mailto:qkrskaqja@snu.ac.kr))

**Manual Outline**

1. Overview of Speech Analysis code
2. Requirement installation
3. Code execution

**1. Overview**

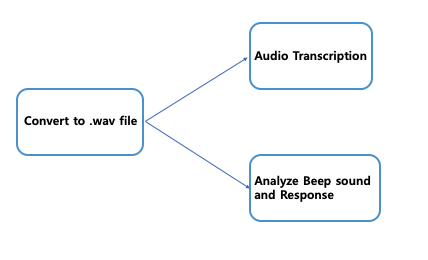


Figure 1. Oveall structure

위 프로젝트는 한국어 음성으로 이루어진 파일을 파이썬으로 분석하여 관련 실험에 활용하는 것을 목적으로 한다. 전체적인 구성은 다음과 같다. 3gp, mp4 등 다양한 포맷으로 저장된 음성파일들을 기본 wav파일로 변환한 후, 이를 분석한다. 크게 분석은 두 가지로 이루어지며, 첫 번째는 음성을 듣고 전사하는 것을 목적으로 하며, 대부분의 전사는 한국어로 이루어진다. 두 번쨰 분석은 실험 과정에서 사용된 ‘비프’ 음성과 그 후에 나오는 학생의 응답을 자동으로 인식해 각각 시간대 및 음성과 응답간의 시간대를 측정해 저장한다.

**2. Requirement Installation**

* Environment

To run this program, you should install the following environments first:

* + anaconda
  + python 3.8
  + numpy
  + scipy
  + crepe
  + resampy
  + pydub
  + google cloud speech API
  + tensorflow
  + ffmpeg
* How to install

설치 및 실행 순서는 다음과 같다.

1. Anaconda 설치 : Python을 사용하기 위한 환경 설정 프로그램
2. 필요한 Library 설치 : 각종 필요한 Python 보조 함수들이라고 생각하면 되고 Requirements.txt를 이용해 설치한다.
3. Google Cloud Speech Api 설치

구글의 음성인식 Api를 사용하기 위해 관련 프로그램들을 설치해야한다.

1. FFmpeg 설치

각종 음성 파일들을 일괄적으로 wav파일로 변환하기 위해 설치한다. 실제 코드에서는 FFmpeg가 있으면 자동으로 모든 3gp, mp4파일들을 변환할 수 있는 스크립트 파일을 구성해놓았으며, 이게 불편하다면 다른 편한 방식으로 진행하여도 무방하다.

1. 프로젝트 폴더 다운로드

프로젝트 폴더를 다운로드 받은 후 압축을 해제한다. 장기적인 코드 저장을 위해 온라인 코드 저장소인 Github에 해당 폴더를 업로드 해놓았으며, Github 사용에 익숙하지 않다면 편한 방식으로 폴더를 저장하고 활용해도 무방하다.

1. 음성 파일 저장

실험 내용을 기록한 음성파일들은 프로젝트 폴더 아래에 audio\_raw 폴더에 저장한다. 이후에 FFmpeg를 설치하고 제공된 convert.py 파일을 이용하면 일괄적으로 해당 폴더에 있는 .3gp, .mp4 파일들은 변환되어 audio\_wav 폴더에 저장된다. 이 방식 대신 직접 audio\_wav에 wav 파일로 변환하여 저장해도 상관없다. 단, audio\_wav 에는 wav 파일만 저장해야하며, Speech\_Analysis 코드들은 audio\_wav안에 있는 wav파일들에만 적용된다.

1. 설치 가이드라인
   * Anaconda

다운로드: <https://www.anaconda.com/download/>

참조: <https://wonderbout.tistory.com/22> (첫 번째 add path~ 체크 해제, vscode 설치 불필요, 단, 개인별로 익숙한 기존 개발 환경이 있다면 자율 선택)

* + Requirements.txt

가상환경 생성 (https://niceman.tistory.com/85)

conda create –name myenv python=3.8 (myenv : 가상환경 이름, 자율 변경)

conda activate myenv (가상환경 활성화)

pip install requirements.txt

* + - Anaconda를 통해 필요한 python library들을 쉽게 설치 및 사용할 수 있음
    - 가상환경을 설정함으로써 library들을 매 순간 상황에 맞게 깔끔하게 사용 가능
  + Google Cloud Speech API

참조:

<https://webnautes.tistory.com/1247> (Windows)

<https://jungwoon.github.io/google%20cloud/2018/01/17/Speech-Api/>

<https://jungwoon.github.io/google%20cloud/2017/10/26/install-gcloud/> (Mac)

* + FFmpeg

<https://ko.wikihow.com/%EC%9C%88%EB%8F%84%EC%9A%B0%EC%97%90-FFmpeg-%EC%84%A4%EC%B9%98%ED%95%98%EB%8A%94-%EB%B0%A9%EB%B2%95> (Windows)

<https://m.blog.naver.com/PostView.nhn?blogId=erm00&logNo=221263403808&proxyReferer=https:%2F%2Fwww.google.com%2F> (Mac)

**3. Code** **Execution**

모든 코드 실행은 Windows 기준으로 명령 프롬프트 (cmd)를 열어서 실행하는 것을 기준으로 설명한다. 개발환경에 익숙하다면 다른 에디터를 사용해도 무방하다.

1. 모든 파이썬 코드를 실행하기 전에는 미리 설정해둔 anaconda 가상환경을 activate하고 사용한다.

**conda activate myenv** (가상환경 활성화, 명령어의 마지막에 myenv는 내가 설정한 가상환경의 이름을 말한다. 개인이 미리 설정해둔 가상환경 이름으로 대체한다.)

1. **python convert.py :** audio\_raw(-> directory) 안에 있는 모든 3gp, mp4 파일들을 wav파일로 바꿔 audio\_wav(-> directory) 안에 저장
2. **python transcription.py test.wav :** audio\_wav 폴더 안에 있는 wav 파일을 읽고(명령어 중 test.wav는 원하는 wav 파일 이름으로 대체해서 사용하면 된다) 전사해서 프로젝트 폴더 아래에 Transcripts 폴더 안에 txt파일로 저장한다.
   1. **파일 길이가 대략 10분이 넘어갈 때**

**Description**: 간혹 파일 크기가 10분이 넘어가는 경우, 자동으로 Google cloud platform에 업로드가 안되는 경우가 있다 (업로드가 되어야만 Google cloud speech API를 사용할 수 있다). 이럴 경우, Google cloud sdk에 해당 파일을 수작업으로 업로드를 하면 된다. 직접 크롬을 켜고 Google cloud platform에 로그인해서 들어가 직접 업로드하면 된다 (일반적인 업로드와 매우 유사)

1. **python Timeline.py test.wav :** audio\_wav 폴더 안에 있는 wav 파일을 읽고 (명령어 중 test.wav는 원하는 wav 파일 이름으로 대체해서 사용하면 된다) 분석하여 beep 사운드와 응답을 분석해 타임라인을 csv파일 형태로 프로젝트 폴더 아래에Timelines 폴더 안에 저장한다.
   1. **파일 길이가 대략 10분이 넘어갈 때**

**Description**: 간혹 파일 크기가 10분이 넘어가는 경우, 자동으로 Google cloud platform에 업로드가 안되는 경우가 있다 (업로드가 되어야만 Google cloud speech API를 사용할 수 있다). 이럴 경우, Google cloud sdk에 해당 파일을 수작업으로 업로드를 하면 된다. 직접 크롬을 켜고 Google cloud platform에 로그인해서 들어가 직접 업로드하면 된다 (일반적인 업로드와 매우 유사)