Virtual dubber

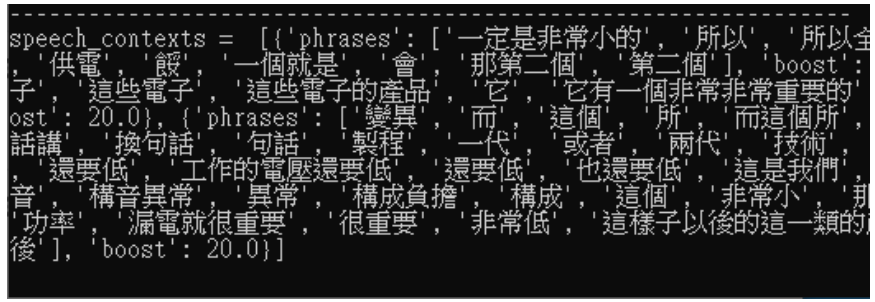
Student : Sian-Yi Chen

Advisor : Tay-Jyi Lin and Chingwei Yeh

Outline
虛擬配音員

- Action item
  1. 改善 ASR (Auto speech recognition) 結果
  2. 手動調整生成語速，使語速對應上嘴型

- Status report
  1. ASR 辨識率提升
     - Test sequence 為王志郁主播 (未調整語速)：
       - 利用 STT (Speech-to-Text) API 提供的 speech adaptation 功能，將原本 91% 辨識率提升至 95%
       - https://youtu.be/XFpMkJpLvg0
     - Test sequence 為王進賢教授 (未調整語速)：
       - 利用 STT (Speech-to-Text) API 提供的 speech adaptation 功能，將原本 89% 辨識率提升至 95%
       - https://youtu.be/vvZasxUwqnM
  2. 手動對齊語速
     - Test sequence 為王進賢教授：
       - 利用 Google API (Text-to-Speech) 提供：SSML (Speech Synthesis Markup Language) 語法手動對齊影片語速
       - https://youtu.be/OtrP2q7AXz8



STT API 提供的 speech adaptation 功能，給予提示字元，用於提升專有名詞的辨識率

ssml (Speech Synthesis Markup Language) 語法

附錄

Google 文件上的 JSON 格式

```
"speechContexts": [
        { "phrases" : [ "fair" ], "boost" : 15 },
        { "phrases" : [ "fare" ], "boost" : 2 }
]
```

```python
phrases1 = ["一二三四"]
phrases2 = ["whether"]
phrases3 = ["$OPERAND"]

# Hint Boost. This value increases the probability that a specific
# phrase will be recognized over other similar sounding phrases.
# The higher the boost, the higher the chance of false positive
# recognition as well. Can accept wide range of positive values.
# Most use cases are best served with values between 0 and 20.
# Using a binary search happroach may help you find the optimal value.
boost1 = 20.0
boost2 = 20.0
boost3 = 20.0
# speech_contexts_element =
speech_contexts = [{"phrases": phrases1, "boost": boost1},
                   {"phrases": phrases2, "boost": boost2},
                   {"phrases": phrases3, "boost": boost3}]
print("------------------------------------------------------------")
print("speech_contexts = ", speech_contexts)
print("\n")
# Sample rate in Hertz of the audio data sent
sample_rate_hertz = 44100
```

左邊的程式碼會輸出與下面相同的格式，與 Google 文件一致

```
speech_contexts =  [
        {'phrases': ['一二三四'], 'boost': 20.0 },
        {'phrases': ['whether'], 'boost': 20.0 },
        {'phrases': ['$OPERAND'], 'boost': 5.0 }
]
```

Python 字串比對工具：difflib：
https://peilee-98185.medium.com/python-%E5%AD%97%E4%B8%B2%E6%AF%94%E5%B0%8D%E5%B7%A5%E5%85%B7-difflib-8efcf99dacfd

Python difflib.SequenceMatcher方法代碼示例：
https://vimsky.com/zh-tw/examples/detail/python-method-difflib.SequenceMatcher.html

解決 Python 中 UnicodeDecodeError: 'cp950' codec can't decode:
https://oxygentw.net/blog/computer/python-file-utf8-encoding/

因為上次 demo 王老師影片，所以調整王老師的影片，並查看原始錯路率

目前使用了 `SequenceMatcher` 來比較兩字串的相似度，範圍從 0~1，越接近 1 就越相近

比較了原始音檔(有背景音樂) 與去背景音樂的音檔，兩者相似度為 0.8881839809674861，所以還是有差，那現在就來看看哪一個與原始音檔比較像
沒想到去掉背影音樂辨識效果更差，不知道為什麼

有背景音樂的：
0.8925348646431501 ➔ 0.877968877968878 ➔ 0.9010629599345871 ➔ 0.9168026101141925 ➔ 0.9213300892133008
沒： 0.8573784006595219

若有編碼問題，需要加上 `,encoding="utf-8"`