

Virtual dubber (improve ASR result)

Student : Sian-Yi Chen

Advisor : Tay-Jyi Lin and Chingwei Yeh

Outline

虛擬配音員

● Action item

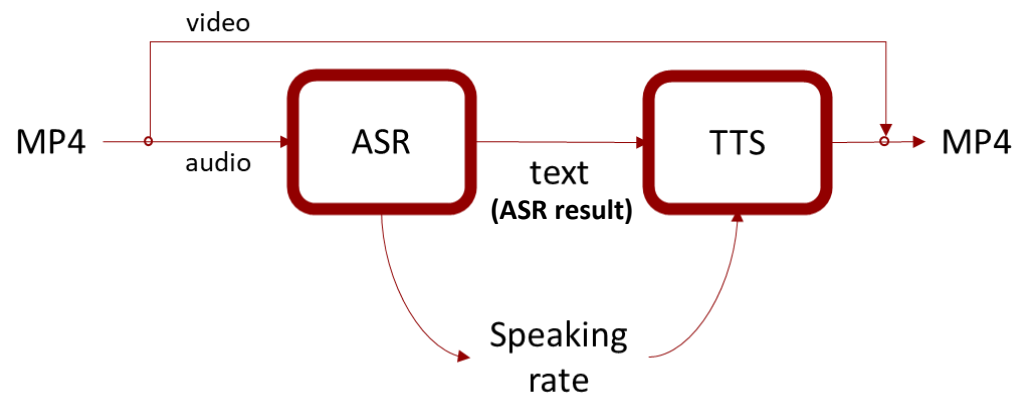
- ❑ Speaking rate control
- ❑ Improve ASR result

● Recap

- ❑ Virtual dubber 架構如 (圖一)
- ❑ 主要針對音訊做處理，以下為兩個重要參數
 1. 文字，涉及到 ASR 辨識率
 2. 語速控制

● Status report

- ① 語速控制 (上周)
 - demo link : <https://youtu.be/PR23ZwADHeQ>
- ② 音訊前處理 (本周)
 - 嘗試對輸入音檔做降噪、增幅，使用等化器 (對特定頻段作增強或減弱) 等功能，但無法提升辨識率
- ③ 提出針對 Google ASR API 既有功能的改善方法
 - Google ASR API speech adaptation 功能為輸入專有名詞以提升辨識率
 - 先前是針對影片辨識錯誤的部分手動加入，現在希望做成自動化，PPT 下一頁使用範例做技術說明



(圖一) Virtual dubber 架構



(圖二) 影片 demo 方式

Google ASR API

既有功能的改善方法介紹

- Google ASR API speech adaptation 功能：提高特定名詞的辨識率

```
"config": {  
  "encoding": "LINEAR16",  
  "sampleRateHertz": 8000,  
  "languageCode": "en-US"  
  "speechContexts": [{  
    "phrases": ["weather"]  
  }]  
}
```

針對想要提高的名詞直接輸入進 phrases 欄位

Speech adaptation : <https://cloud.google.com/speech-to-text/docs/speech-adaptation>

- ASR 辨識改善方法流程 (圖三)

▣ 情境為今天想要配音一部講述關於印度疫情的新聞影片

Step 1：使用爬蟲去收集關鍵字“印度”、“疫情”、“新聞”的新聞稿

Step 2：利用斷詞系統將新聞稿切割成數組名詞

Step 3：取出出現機率高的名詞作為的 Google ASR API 專有名詞功能的 input

Step 4：計算 ASR 結果是否提升

印度疫情失控至今竟快達成群體免疫？醫護：代價台灣無法承受



印度疫情至今仍相當嚴峻，疫苗接種率也低，但該國竟有接近7成人口已有武漢肺炎病毒抗體。（路透）

2021/07/21 23:58

內文：

〔即時新聞／綜合報導〕印度武漢肺炎（新型冠狀病毒病，COVID-19）疫情失控至今仍未見趨緩...

↓ 透過爬蟲抓取新聞稿內文

內文：

〔即時新聞／綜合報導〕印度武漢肺炎（新型冠狀病毒病，COVID-19）疫情失控至今仍未見趨緩...

↓ 利用斷詞系統切字

即時／新聞／綜合／報導／印度／武漢肺炎／新型／冠狀／病毒病／疫情／失控／至今／仍未／見／趨緩

↓ 計算出現機率高的詞語

作為特定名詞輸入 phrases 欄位

(圖三) 示意流程

附錄

Outline

虛擬配音員

● Action item

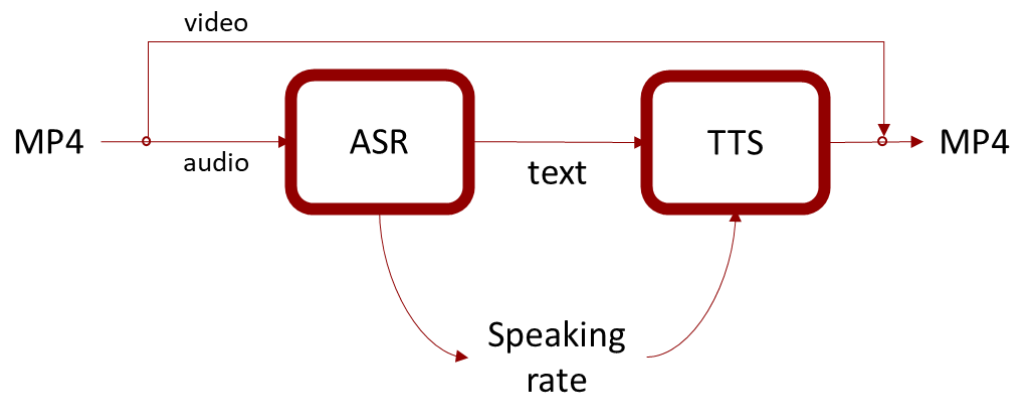
- ▣ Speaking rate control

● Demo link

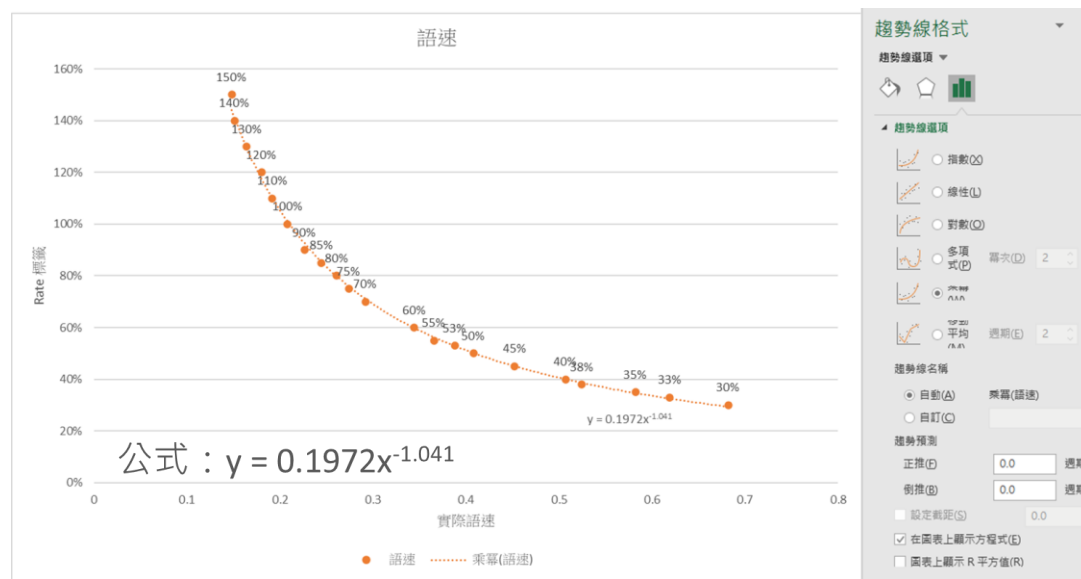
- ▣ Test sequence 王進賢教授：<https://youtu.be/PR23ZwADHeQ>

● Speaking rate control method

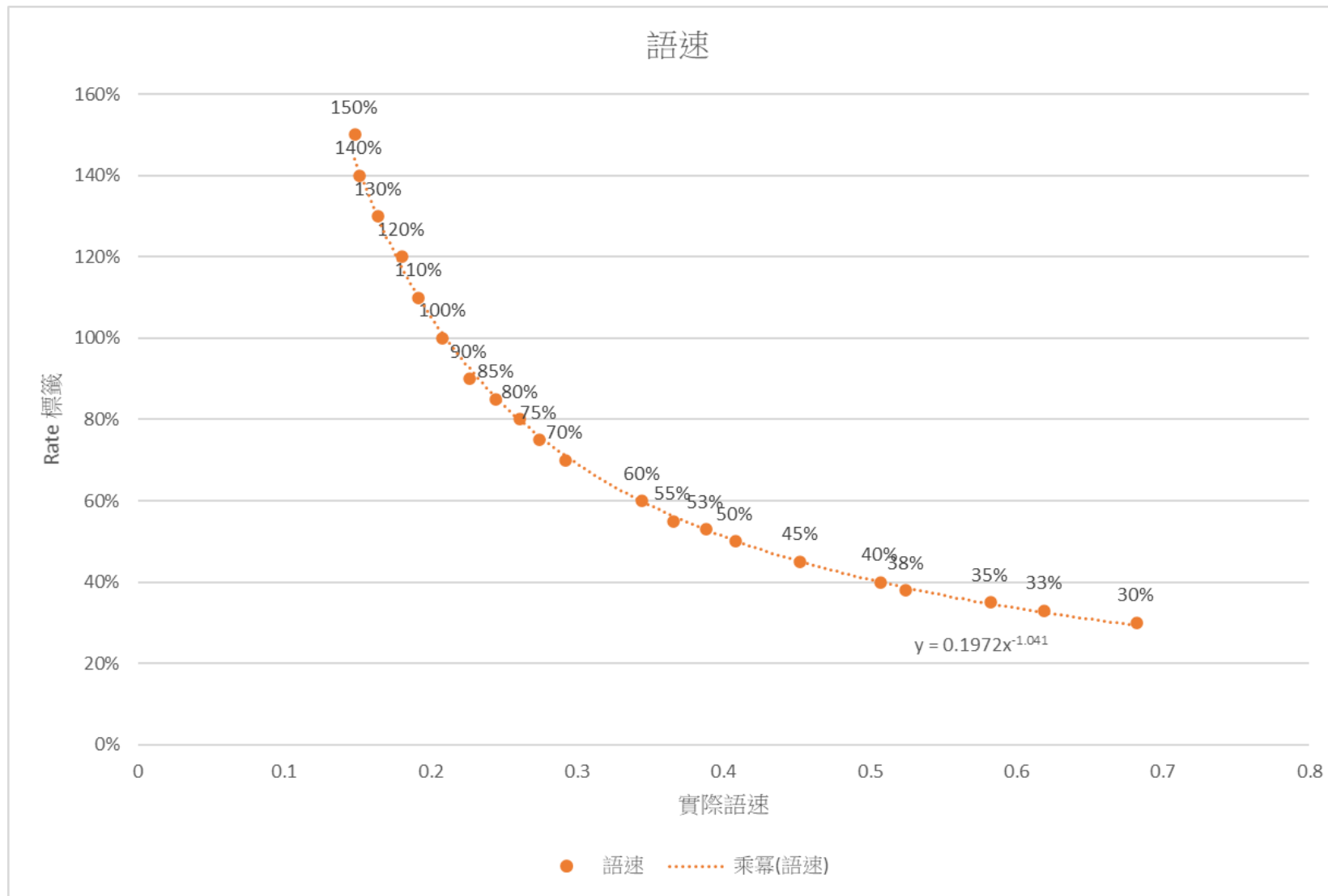
1. 利用 TTS 提供的百分比語速 (e.g., 90%, 100%, 110%) 合成語音用於模擬實際音檔
2. 利用 ASR 提供 “計算每個字聲音持續的時間” 的功能計算 TTS 合成出來的語音速度
3. 將合成語音語速與提供的百分比製成表格
4. 利用 excel 將表格內容繪製成散佈圖
5. 選擇乘冪趨勢線並算出曲線方程式
 - 公式： $y = 0.1972x^{-1.041}$
6. 利用此公式將文字標上速度標籤



(圖一) Virtual dubber 架構



(圖二) 實際語速與 TTS 提供語速的曲線



趨勢線格式

趨勢線選項 ▾

趨勢線選項

- ☐ 指數(X)
- ☐ 線性(L)
- ☐ 對數(O)
- ☐ 多項式(P) 冪次(D) 2
- ☒ 乘幂(M)
- ☐ 平均 (A) 週期(E) 2

趨勢線名稱

☒ 自動(A) 乘幂(語速)

☐ 自訂(C)

趨勢預測

正推(F) 0.0 週期

倒推(B) 0.0 週期

☐ 設定截距(S) 0.0

☒ 在圖表上顯示方程式(E)

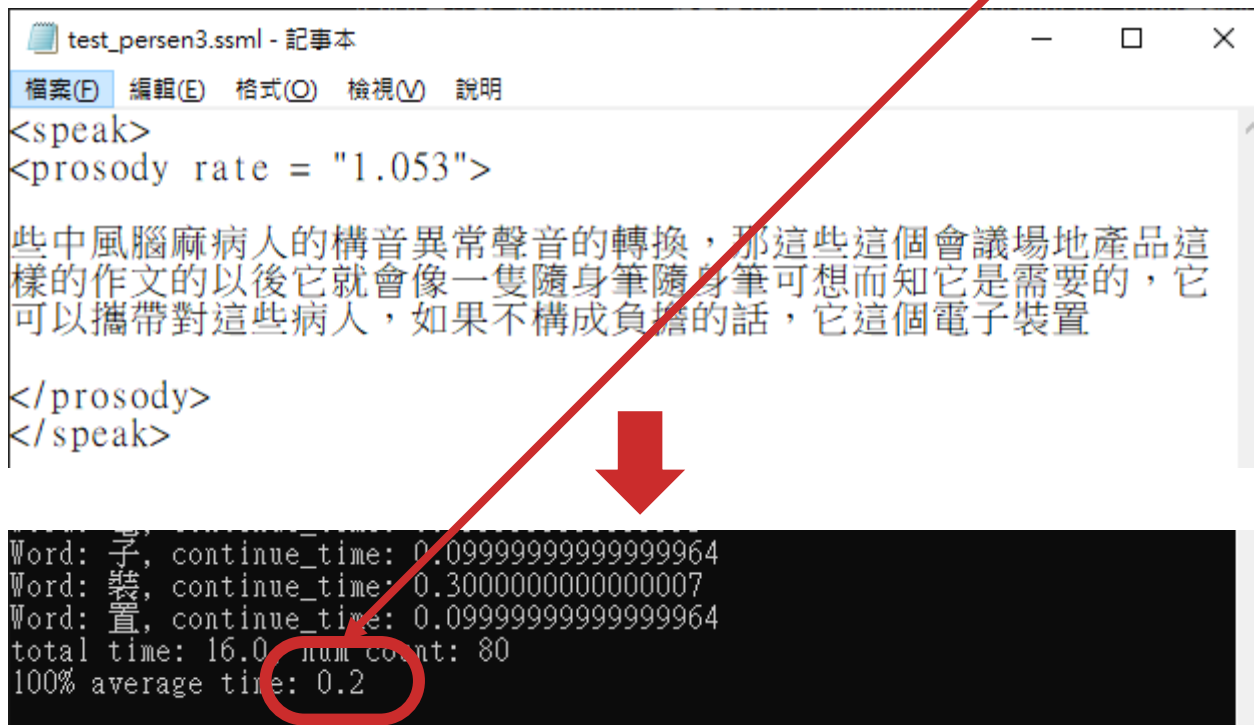
☐ 圖表上顯示 R 平方值(R)

公式： $y = 0.1972x^{-1.041}$

```
print ("0.1972 * math.pow(x, -1.041) : ", 0.1972 * math.pow(0.19999999, -1.041))
```

公式： $y = 0.1972x^{-1.041}$

0.1972 * math.pow(x, -1.041) : 1.0532578574884441



The image shows a Notepad window titled "test_persen3.ssml - 記事本" with the following SSML code:

```
<speak>
<prosody rate = "1.053">
些中風腦麻病人的構音異常聲音的轉換，那這些這個會議場地產品這
樣的作文的以後它就會像一隻隨身筆隨身筆可想而知它是需要的，它
可以攜帶對這些病人，如果不構成負擔的話，它這個電子裝置
</prosody>
</speak>
```

Below the Notepad window is a terminal window showing the following output:

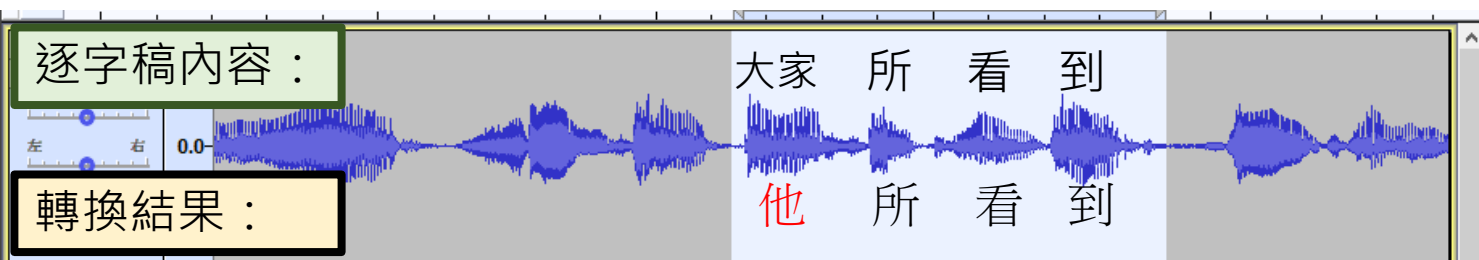
```
Word: 子, continue_time: 0.099999999999999964
Word: 裝, continue_time: 0.30000000000000007
Word: 置, continue_time: 0.099999999999999964
total time: 16.0 num count: 80
100% average time: 0.2
```

得證，此公式合理且準確
(已完成) 30sec_In_addtag_optimize_syn.py



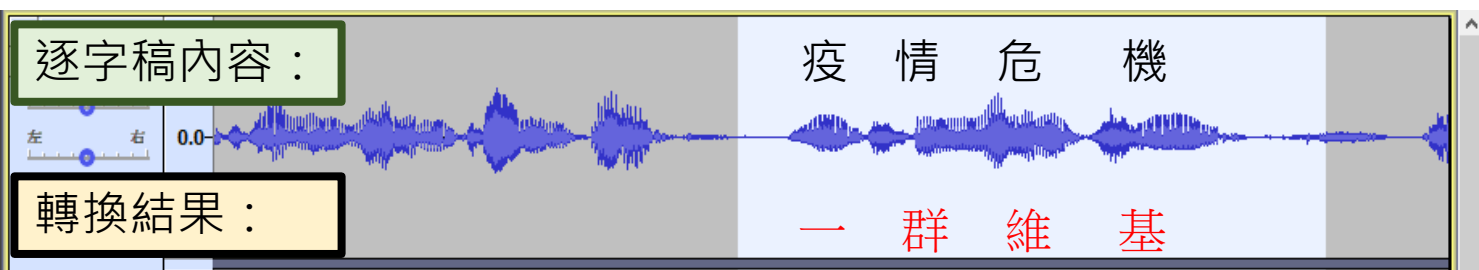
做了各式各樣的語音前處理，感覺對於辨識沒有什麼幫助

目前想到的方法是利用一些方式，取得特定名詞加入



在辨識之前先預測音檔裡面會有什麼名詞容易辨識錯誤

像是疫苗、疫情
這些比較像是時事會出現的詞彙



所以在輸入音檔前要求輸入 3~5 個關鍵字

e.g. 印度、疫情、新聞、骨氣、防疫
然後就會利用爬蟲去抓相關的新聞

