

# Virtual dubber (speaking rate control)

---

Student : Sian-Yi Chen

Advisor : Tay-Jyi Lin and Chingwei Yeh

# Outline

虛擬配音員

## ● Action item

1. 找控制語速的 API function
2. 提出 TTS speaking rate control 自動化的方法並執行

## ● Status report

1. ASR API speaking rate function
  - Legal values are: a non-negative percentage or "x-slow", "slow", "medium", "fast", "x-fast", or "default". Labels "x-slow" through "x-fast" represent a sequence of monotonically non-decreasing speaking rates.

(<https://www.w3.org/TR/speech-synthesis11/#S3.2.4>)



Test sequence : 你好 我的名字 叫做陳憲億 請多多指教

Speaking rate : 

2. Speaking rate control 自動化的方法 (共 7 點)

### 一. 加標籤

- 在 Speech-to-text API 中找到有提供“計算每個字聲音持續的時間”如 (圖一) 的函數
- 讀入音檔、利用上一步驟的函數設定閾值、標上 Speech Synthesis Markup Language (SSML) 標籤
- 將連續擁有同樣標籤的字合併成同一個標籤如 (圖二)

```
Transcript: 這個專利主要指的是靜態隨機存  
Word: 這, start_time: 0.0, end_time: 0.5  
Word: 個, start_time: 0.5, end_time: 0.6  
Word: 專, start_time: 0.6, end_time: 0.8  
Word: 利, start_time: 0.8, end_time: 0.9
```

(圖一) 計算每個字聲音持續的時間

```
<prosody rate = "fast">我</prosody>  
<prosody rate = "fast">的</prosody>  
<prosody rate = "fast">名</prosody>  
<prosody rate = "fast">字</prosody>
```



```
<prosody rate = "fast">我的名字</prosody>
```

(圖二) 合併標籤

(下一頁)

## 2. Speaking rate control 自動化的方法

### 一. 加標籤

- 在 Speech-to-text API 中找到有提供 “計算每個字聲音持續的時間” 如 (圖一) 的函數
- 讀入音檔、利用上一步驟的函數設定閾值、標上 Speech Synthesis Markup Language (SSML) 標籤
- 將連續擁有同樣標籤的字合併成同一個標籤如 (圖二)

### 二. 拆音檔

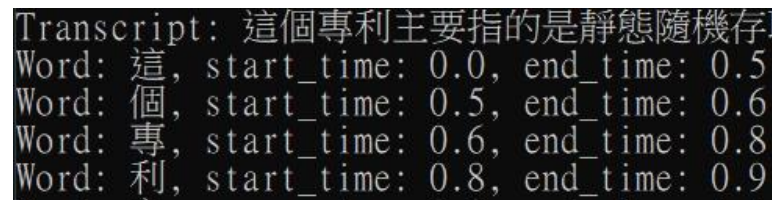
- 將 SSML 文字檔合成語音有字數限制，須將讀入的音檔拆成數個小音檔，音檔拆成數段，想到兩種實作方式
  - A. 每 x 秒切一段 (目前使用 30 秒) **(目前使用方式)**
    - 優點：較容易實作
    - 缺點：有可能從句與句中切斷，造成 ASR 辨識時，辨識率下降
  - B. 每一句話切一段
    - 優點：切割成片段不影響語意、ASR 辨識
    - 缺點：如果一口氣念一大串文字、中斷語氣短至無法判別時不易切割、實作困難
- 將數段音檔分別轉換為文字 **(完成至此)**
- 標上標籤，輸出成 .ssml 檔

### 三. 合併

- 將數個 .ssml 檔分別合成為語音並合併在一起

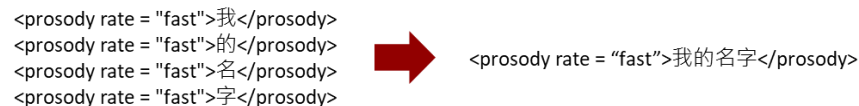
## 3. 下周規劃

- 完成 Speaking rate control 自動化方法的最後兩步驟 (灰字)
- ASR API speaking rate function 目前使用 slow、medium、fast...等預設值，更改成百分比使語速更精準
- 將 Speaking rate control 自動化方法與虛擬配音員主程式合併



Transcript: 這個專利主要指的是靜態隨機存  
Word: 這, start\_time: 0.0, end\_time: 0.5  
Word: 個, start\_time: 0.5, end\_time: 0.6  
Word: 專, start\_time: 0.6, end\_time: 0.8  
Word: 利, start\_time: 0.8, end\_time: 0.9

(圖一) 計算每個字聲音持續的時間



```
<prosody rate = "fast">我</prosody>  
<prosody rate = "fast">的</prosody>  
<prosody rate = "fast">名</prosody>  
<prosody rate = "fast">字</prosody> → <prosody rate = "fast">我的名字</prosody>
```

(圖二) 合併標籤

# 附錄

Speech Synthesis Markup Language (SSML) :  
<https://cloud.google.com/text-to-speech/docs/ssml>

語音合成標記語言 (SSML) 1.1 版 :  
<https://www.w3.org/TR/speech-synthesis11/#S3.2.4>

3.2.4 韻律元素

所述韻律元素允許控制音高的，語速和語音輸出的音量。屬性，所有 *optional*，是：

- **pitch**：包含文本的基線間距。儘管“基線音高”的確切含義因合成處理器而異，但增加/減少此值通常會增加/減少輸出的近似音高。合法值是：一號碼後跟“Hz”的，一個相對變化或“X-低”，“低”，“中”，“高”，“X-高”，或“默認”。標籤“x-low”到“x-high”代表一系列單調非遞減的音高電平。
- **contour**：設置包含文本的實際音高輪廓。格式在下面的音高輪廓中指定。
- **range**：所包含文本的音高範圍（可變性）。儘管“音高範圍”的確切含義因合成處理器而異，但增加/減少此值通常會增加/減少輸出音高的動態範圍。合法值是：一號碼後跟“Hz”的，一個相對變化或“X-低”，“低”，“中”，“高”，“X-高”，或“默認”。標籤“x-low”到“x-high”代表一系列單調非遞減的音高範圍。
- **rate**：包含文本的語速變化。合法值為：[非負百分比](#)或“x-slow”、“slow”、“medium”、“fast”、“x-fast”或“default”。標籤“x-slow”到“x-fast”代表一系列單調非遞減的語速。當值為[非負百分比](#)時它充當違約率的乘數。例如，值 100% 表示語速不變，值 200% 表示語速是默認語速的兩倍，值 50% 表示語速是默認語速的一半。語音的默認速率取決於語言和方言以及語音的個性。語音的默認速率應該是這樣的，當朗讀文本時，它被視為正常的語音語速。由於語音是特定於處理器的，因此默認速率也是如此。
- **duration**：以秒或毫秒為單位的值，用於讀取包含的文本所需的時間。遵循級聯樣式表第 2 級建議 [CSS2] 中的時間值格式，例如“250ms”、“3s”。
- **volume**：包含文本的體積。合法值是：一號碼前面加“+”或“-”和緊跟“dB”；或“silent”、“x-soft”、“soft”、“medium”、“loud”、“x-loud”或“default”。默認值為 +0.0dB。指定“靜音”值 相當於指定負無窮分貝 (dB)。標籤“silent”到“x-loud”代表一系列單調不降低的音量級別。當值為有符號數(dB) 時，) 和電流幅度 (a<sub>0</sub>)，並以 dB 為單位定義：

$$\text{volume}_{\text{(dB)}} = 20 \log_{10} (a_1 / a_0)$$

請注意，所有數字音量級別（以 dB 為單位）都與當前級別相關，並且它們始終帶有符號（包括零）。另請注意，一旦當前音量級別設置為“靜音”，所有子級相關更改也會導致靜音。子韻律元素 可以使用標籤“默認”來重置當前音量級別。

所以對於一個值：

- "silent"，默讀所包含的文本；
- '-6.0dB'，在當前信號幅度的大約一半處讀取包含的文本；
- '-0dB'，讀取包含的文本，音量沒有相對變化；
- '+6.0dB'，所包含的文本以當前信號幅度的大約兩倍的幅度被讀取。

請注意，標籤值的此屬性的行為可能與數值的行為不同。使用數值會導致波形的直接修改，而使用標籤值可能會導致更準確地反映人類如何增加或減少其語音感知響度的韻律修改，例如，不同地調整頻率和功率不同的聲音單位。

雖然每個屬性單獨是可選的，但如果在[使用prosody](#)元素時沒有指定屬性，則會出錯。“x-foo”屬性值名稱旨在作為“額外foo”的助記符。所有單位（“Hz”、“st”）都區分大小寫。另請注意，習慣音高級別和標準音高範圍可能因語言而異，音高目標和範圍的標記值的含義也可能會有所不同。

語速：

[https://github.com/googleapis/python-speech/blob/HEAD/samples/snippets/transcribe\\_word\\_time\\_offsets.py](https://github.com/googleapis/python-speech/blob/HEAD/samples/snippets/transcribe_word_time_offsets.py)

```
script: 你好我的名字叫做陳憲一請多多指教
: 你, start_time: 0.0, end_time: 0.2, continued_time: 0.2
: 好, start_time: 0.2, end_time: 0.5, continued_time: 0.3
: 我, start_time: 0.5, end_time: 0.7, continued_time: 0.19999999999999996
: 的, start_time: 0.7, end_time: 0.8, continued_time: 0.10000000000000009
: 名, start_time: 0.8, end_time: 0.8, continued_time: 0.0
: 字, start_time: 0.8, end_time: 1.0, continued_time: 0.19999999999999996
: 叫, start_time: 1.0, end_time: 1.3, continued_time: 0.30000000000000004
: 做, start_time: 1.3, end_time: 1.5, continued_time: 0.19999999999999996
: 陳, start_time: 1.5, end_time: 1.8, continued_time: 0.30000000000000004
: 憲, start_time: 1.8, end_time: 2.1, continued_time: 0.30000000000000004
: 一, start_time: 2.1, end_time: 2.2, continued_time: 0.10000000000000009
: 請, start_time: 2.2, end_time: 2.5, continued_time: 0.29999999999999998
: 多, start_time: 2.5, end_time: 2.7, continued_time: 0.200000000000000018
: 多, start_time: 2.7, end_time: 2.9, continued_time: 0.19999999999999973
: 指, start_time: 2.9, end_time: 3.1, continued_time: 0.200000000000000018
: 教, start_time: 3.1, end_time: 3.3, continued_time: 0.19999999999999973
```

所以 slow 的語速差不多在	0.2~0.3	之間
fast 的語速在	0~0.2	之間
medium 的語速在	0.19~0.29	之間
x-slow 的語速大概在	0.2~0.4	之間

```
<prosody rate="slow" pitch="-2st">
    你好
</prosody>
```

```
<prosody rate="fast" pitch="+0st">
    我的名字
</prosody>
```

```
<prosody rate="slow" pitch="+2st">
    叫做
</prosody>
```

```
<prosody rate="fast" pitch="+0st">
    陳憲億
</prosody>
```

```
<prosody rate="medium" pitch="+0st">
    請多多指教
</prosody>
```

STTResult\_2021-07-08\_03-50-56.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
<speaK>
你好我的名字叫做陳憲一請多多指教
</speaK>
```



STTResult\_2021-07-08\_04-17-55.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
<speaK>
<prosody rate = "medium">你</prosody>
<prosody rate = "slow">好</prosody>
<prosody rate = "slow">我</prosody>
<prosody rate = "fast">的</prosody>
<prosody rate = "fast">名</prosody>
<prosody rate = "fast">字</prosody>
<prosody rate = "slow">叫</prosody>
<prosody rate = "fast">做</prosody>
<prosody rate = "fast">陳</prosody>
<prosody rate = "slow">憲</prosody>
<prosody rate = "fast">一</prosody>
<prosody rate = "medium">請</prosody>
<prosody rate = "medium">多</prosody>
<prosody rate = "fast">多</prosody>
<prosody rate = "fast">指</prosody>
<prosody rate = "slow">教</prosody>
</speaK>
```

第 15 列，第 35 行



\*STTResult\_2021-07-08\_04-17-55

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
<speaK>
<prosody rate = "medium">你</prosody>
<prosody rate = "slow">好我</prosody>
<prosody rate = "fast">的名字</prosody>
<prosody rate = "slow">叫</prosody>
<prosody rate = "fast">做陳</prosody>
<prosody rate = "slow">憲</prosody>
<prosody rate = "fast">一</prosody>
<prosody rate = "medium">請多</prosody>
<prosody rate = "fast">多指</prosody>
<prosody rate = "slow">教</prosody>
</speaK>
```

第 12 列，第 9 行 100% Windows (CRLF) ANSI



研究：

報告事項：Virtual dubber (speaking rate control)

(已完成)

- 在 Speech Synthesis Markup Language (SSML) 中有找到控制語速的標籤，將 xxx.ssm1 檔放入 TTS 中可轉換出有快有慢的人聲
- 在 Speech-to-text 中找到有提供 "計算每個字的聲音持續的時間" 的函數
- 讀入音檔，根據每個字持續的時間加入 speak rate 標籤
- 將檔案優化，因為上一步驟是每個字都打標籤，因此會有連續好幾個字都是同一個標籤，將之合併

(目前遇到的問題)

- 在 text-to-speech API 中輸入的 ssm1 檔案大小一次不能超過 5000 characters，大約為 50 秒的音檔大小限制

(待完成事項)

- 將輸入的音檔切成眾多小音檔，然後轉換後，再將所有生成的人聲合併

將音訊切成片段，目前想到兩種方法：

假設一段音訊為 2 分鐘 (120秒)

1. 每 30 秒切一段音訊，總共切成 4 等分

- 優：較簡單
- 缺：一句話可能會被從中間切斷，造成語句不完整性，降低 ASR 辨識率

2. 每一句話切一段，從波形 (找音訊停頓的時候) 判斷哪裡到哪裡是一句話

- 優：切成片段再組合不會影響語意
- 缺：如果一口氣念一大串文字，中斷語氣又幾乎沒有就無法切割