題目：雲端語音合成技術應用於長篇文章音文同步有聲書之建立
An Application of Speech-Text Alignment in Text-to-speech to Creation of Audio-Books with Speech-Text Synchronization

摘要：

運用雲端語音合成技術(Text-to-speech Technology,TTS)結合音文同步技術(Speech-Text Synchronization)，建立一套能夠從一段長篇文章取得音文同步有聲書的Timed-text檔案的方法，並基於電腦輔助語言學習(Computer-assisted Language Learning ,CALL)之目的，希望以此製作一套輔助語言學習的系統。大概有以下步驟(1)Text Spliter ─ 將原始文章切割成Sentence-level Text，使其長度能夠讓Google TTS接受(2)Upload to Google translate ─ 利用Python Standard Library 的urllib模組，將字串交給Google TTS(3)Download TTS Audio file ─ 用binary的方式將Google TTS回傳的內容儲存為MP3(4)Timed-text file Converter ─ 將Sentence-level Text進一步黏上時間標籤，製作成SBV檔(5)Audio Converter ─ 利用FFmpeg將MP3檔轉成WAV檔(6)CguAlign ─ 將WAV檔和SBV檔作自動切音，使產生Word-level Text的SBV檔(7)Website Presentation ─ 用javascript製作簡單的網頁能夠流覽Word-level音文同步的文章。

介紹：

運用雲端語音合成技術(Text-to-speech Technology,TTS)，諸如 Google Translate、iSpeech等等，建立一套能夠從一段長篇文章取得音文同步(Speech-Text Synchronization)的技術，並且運用此技術取得的音文同步有聲書的Timed-text檔案，基於電腦輔助語言學習(Computer-assisted Language Learning ,CALL)之目的，以此Timed-text檔案製作一套基礎的有聲書學習系統。

研究背景與動機：

隨著地球村趨勢的來臨，擁有多種的語言能力成為競爭力的一種指標，在聽力(listening)及口說(speaking)上的表達能力猶是，因此如何培養語言的能力，漸漸的受到重視。

數位教材在電腦與網路普遍並大量使用之下，已逐漸形成一種趨勢，運用電腦及網路科技輔助的語言教學廣受注意，「電腦輔助語言學習」(Computer-assisted language learning , CALL)，成為一項熱門的研究主題。

目的：

為達CALL之目的，希望能將雲端語音合成技術和音文同步有聲書結合，製作一套輔助語言學習的系統。

相關研究與文獻探討：

1. **Timed-text檔案：**

市面上有許多 Timed-text檔案，用途包括電影字幕、音樂歌詞.....等等，諸如LRC、SBV、SRT.....等等格式，而這些格式不外乎是"時間標籤"+"內容"的不同的組合方法。

```
0:00:08.660,0:00:13.830
THE ADVENTURES OF TOM SAWYER

0:00:13.830,0:00:15.969
MARK TWAIN (Samuel Langhorne

0:00:15.969,0:00:22.040
P R E F A C E
```

SBV檔格式

```
[00:06.30]She was more like a beauty queen from a mo
[00:11.10]I said don't mind, but what do you mean I
[00:18.10]Who will dance on the floor in the round
[00:24.29]She said I am the one will dance on the fl
[00:36.39]She told me her name was Billie Jean, as s
```

LRC檔格式

```
1
00:00:20.109-->00:00:21.849
thank you very much

2
00:00:21.849-->00:00:23.869
gertrude mondello

3
00:00:23.869-->00:00:28.369
for your dedicated work that has brought us

4
00:00:28.369-->00:00:31.419
distinguished delegates and guests
```

SRT檔格式

2. Google Translate -TTS：

Google Translate TTS API的機制是基於HTTP GET Request，其URL格式為：

http://translate.google.com/translate_tts?ie=utf-8&tl={lang}&q={query}&total={total}&idx={index}&textlen={textlen}

其意義為：

- tl={lang} ：tl，即target language，欲以何種語言對文章做TTS，常用的有
  - en：英文
  - zh-tw：中文
  - ja：日文
  - 其種類繁多，可至Google Translate API查看。
    - https://cloud.google.com/translate/v2/using_rest#language-params
- q={query}：欲做TTS的字串，若是直接從 Google Translate的網頁上輸入長度過長的字串，其會自動切割，其上限長度為100。
- total={total}：欲做TTS的長字串經過 Google Translate的切割，所切割出來的份數。
- idx={index}：經過 Google Translate的切割，標示目前所處理的為第幾份經過切割 的字串，其標記從0開始。
- textlen={textlen}：此query的字串長度，其上限長度為100。

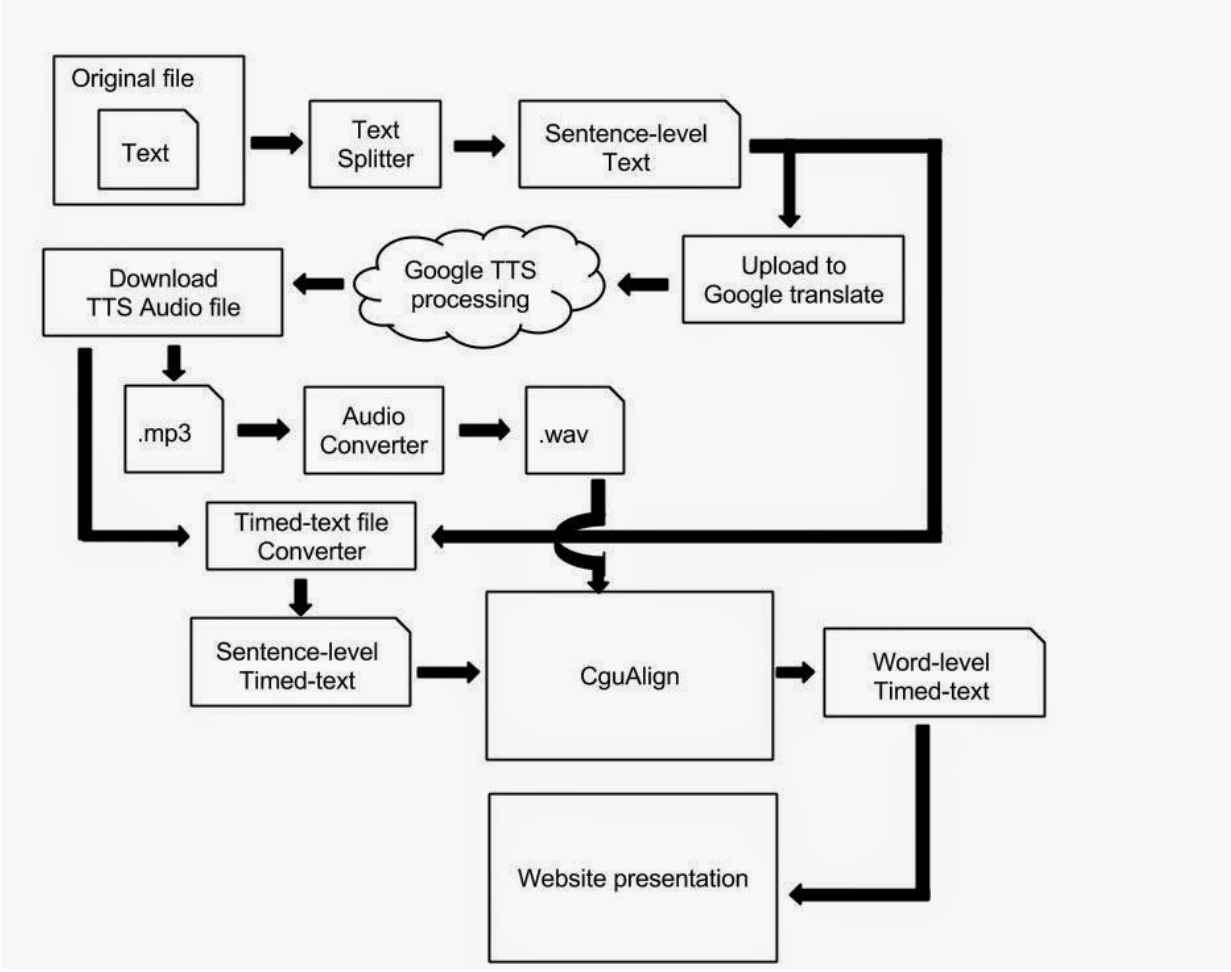
Example：

- I am a Chang Gung University Student
- 其URL格式為：http://translate.google.com.tw/translate_tts?ie=UTF-8&q=I%20am%20a%20Chang%20Gung%20University%20Student&tl=en&total=1&idx=0&textlen=36
- tl = en，即將此字串以英文做TTS。
- q = I%20am%20a%20Chang%20Gung%20University%20Student，即欲TTS的字串"I am a Chang Gung University Student"。
- total = 1，因此字串長度不超過100，因此不須經過切割，而此字串只有一份，因此為1。
- idx = 0，因此字串未經過切割，且只有本句，因此設為0。
- textlen = 36，此字串的長度為36。

3. CguAlign - 音文同步：

CguAlign是本實驗室一個方便處理音文同步有聲書的技術，以程式自動切音取代傳統人工手動的方式，除了可以減少人力資源，還大幅減少人工手動切音所浪費的時間。只需輸入文字檔以及聲音檔，經過音文對齊的處理，即可得到帶有時間點的Timed-text文字檔案。但此技術無法處理過長的聲音檔，因此希望站在雲端語音合成技術上，將以"句"為層級的TTS改良，使其能夠達到"字"的層級。CguAlign是用Python將HTK包裝、運用的一套技術，何謂HTK？HTK為Hidden Markov Toolkit的縮寫，是一套用於語音辨識以及語音訓練的免費軟體，是語音學術界已沿用20多年的軟體。

研究方法及步驟：

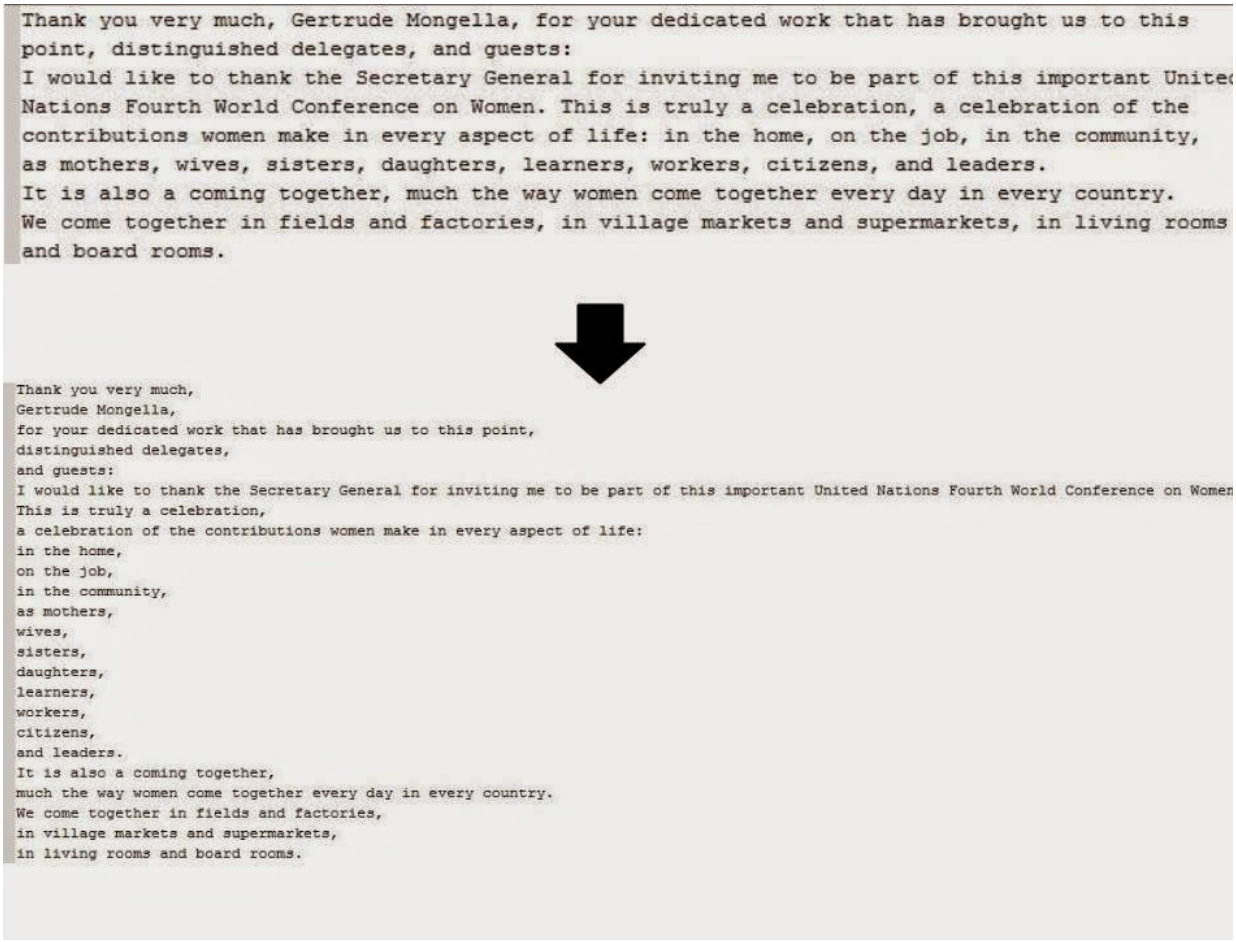


流程圖

1. **Text Spliter ：**

因Google TTS無法直接輸入長度大於100的字串，因此需要先做文字分割，將其長度降低於小於100，並稱此為Sentence-level的Text檔，基本的切割方法只先按照標點符號的切割。

(1)按照"句號"做切割，若字串長度皆小於100，則切割結束，否則繼續切割。
(2)按照"問號"做切割，若字串長度皆小於100，則切割結束，否則繼續切割。
(3)按照"驚嘆號"做切割，若字串長度皆小於100，則切割結束，否則繼續切割。
(4)按照"破折號"做切割，若字串長度皆小於100，則切割結束，否則繼續切割。
(5)按照"冒號"做切割，若字串長度皆小於100，則切割結束，否則繼續切割。
(6)按照"逗號"做切割，若字串長度皆小於100，則切割結束，否則繼續切割。
(7)若最終字串長度還是有超過100的，則會從超過100的字串以中間的"空白"切割。



OriginalText to Sentence-level Text

2. **Upload to Google translate：**

利用Python內建的Standard Library---"urllib.request"和"urllib.parse"，傳送HTTP GET Request。Google TTS的URL為"http://translate.google.com/translate_tts"，其格式須包含

- tl - target language
- q - query
- total - total number of text segments
- idx - index of text segments
- textlen - string length in this segment

```
GOOGLE_TTS_URL= 'https://translate.google.com.tw/

payload = { 'ie': 'utf-8',
            'tl': lang,
            'q':  text,
            'total': len(textlangL),
            'idx':   idx,
            'textlen': len(text) }

hdr = {'User-Agent':'Mozilla/5.0'}
data = urllib.parse.urlencode(payload)
req = urllib.request.Request(GOOGLE_TTS_URL+data
r = urllib.request.urlopen(req)
```

Upload to Google translate

3. **Download TTS Audio file：**

上圖Python URL Request的最後一行：

r = urllib.request.urlopen(req)

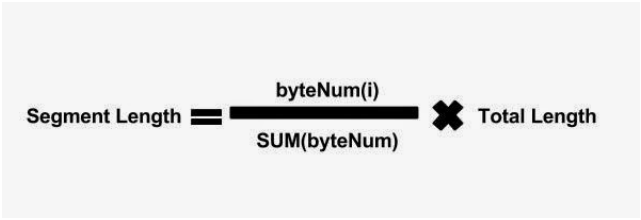
就是Google TTS處理後回傳的資料，因已知其資料格式是MP3檔，因此已binary的方式將此回傳的資料寫入savefile內，並將每一段segment的大小都用byteNum記下來，留以下階段**Timed-text file Converter**做使用。

```
f= open(savefile, 'wb+')
for idx, textlang in enumerate(textlangL):
    GOOGLE_TTS_URL= 'https://translate.google.com.tw/
    payload = { 'ie': 'utf-8',
                'tl': lang,
                'q':  text,
                'total': len(textlangL),
                'idx':   idx,
                'textlen': len(text) }
    hdr = {'User-Agent':'Mozilla/5.0'}
    data = urllib.parse.urlencode(payload)
    req = urllib.request.Request(GOOGLE_TTS_URL+data,
    r = urllib.request.urlopen(req)

    byte= r.read()
    byteNum= len(byte)
    f.write(byte)
f.close()
```

Download TTS Audio file

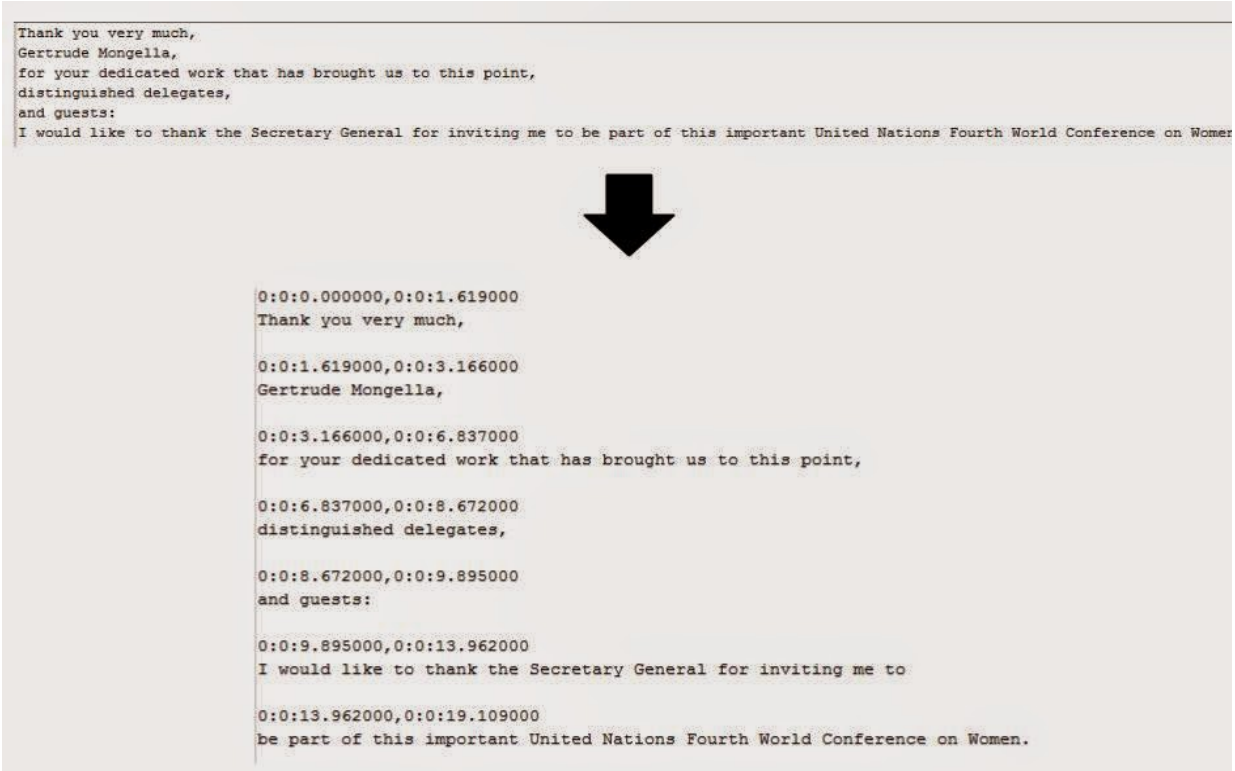
4. **Timed-text file Converter：**

利用上一步驟所蒐集的每一個segment的byteNum大小，並計算byteNum的總和，能夠計算出每一段segment在總語音長度中的時間長度。



Segment Length

並且依此與Sentence-level Text黏合，製作Sentence-level的Timed-text file，在此我是製作成SBV格式的Timed-text file。



Sentence-level Text to Timed-text file

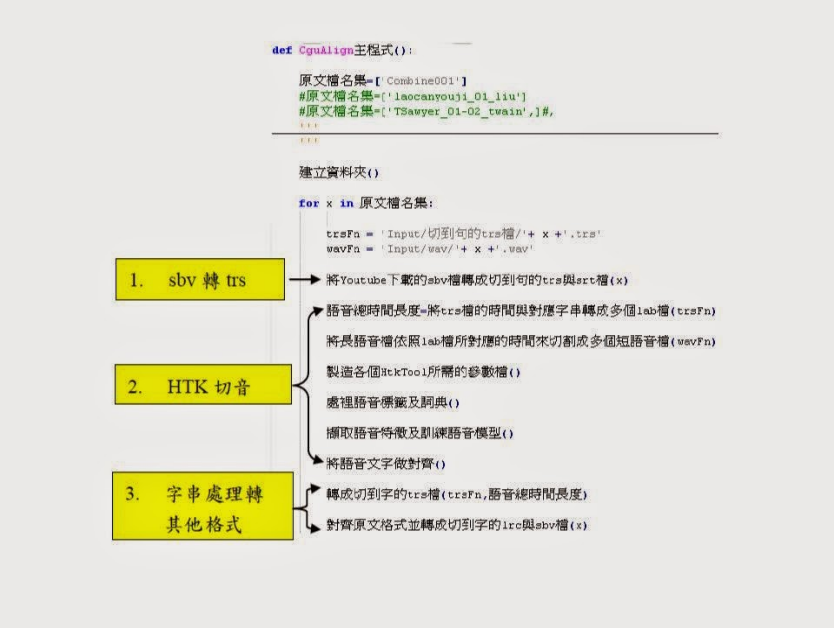
5. **Audio Converter：**

因CguAlign只能夠使用WAV檔，因此需要將Google TTS所下載到的MP3檔轉換成WAV檔。Google TTS的MP3檔格式，取樣頻率為16kHz，bitrate為32kbps，單聲道。使用FFmpeg來幫助轉檔，FFmpeg是一個自由軟體，可以執行音訊和視訊多種格式的錄影、轉檔、串流功能。在Windows環境Python呼叫FFmpeg的方法就是import os，呼叫os.system()，並且利用FFmpeg的"report"option，可以得到聲音檔的長度。

```
def ffmpeg_AudioDuration(filename):
    os.system("ffmpeg -report -y -i ./input/{0}.mp3 ./input/{1}.wav".format(filename,filename))

    dirlist= os.listdir()
    for i in dirlist :
        if i.find('ffmpeg')!=-1 and i.find('.log') !=-1 :
            report_name= i
            break

    f=open(report_name,"r")
    for i in f:
        if i.find("Duration:") != -1:
            duration= i.split("  Duration: ")[1].split(",")[0]
            hour= int(duration.split(":")[0])
            min = int(duration.split(":")[1])
            sec = float(duration.split(":")[2])
            total_ms= int(hour* 3600000 + min*60000 + sec*1000)
            print(total_ms)
    f.close()
    os.system("copy "+report_name+" .\\ffmpeglog\\"+report_name)
    os.system("del "+report_name)

    return total_ms
```

呼叫FFmpeg

6. CguAlign：



CguAlign主程式

CguAlign是將HTK用Python包裝的一隻程式，其功能是要將Sentence-level Text轉成Word-level Text，分成三個部分：

a. **SBV轉TRS－將SBV檔轉成TRS檔**

```
00:00:20.109,00:00:21.849
thank you very much

00:00:21.849,00:00:23.869
gertrude mondello

00:00:23.869,00:00:28.369
for your dedicated work that has brought
us to this point

00:00:28.369,00:00:31.419
distinguished delegates and guests

00:00:31.419,00:00:34.800
i would like to thank the secretary
general

00:00:34.800,00:00:36.700
for inviting me
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Trans SYSTEM "trans-14.dtd">
<Trans scribe="wjLabtoTrs" audio_filename="Input/Hillary_Womens_Rights.wav">
<Episode>
<Section type="report" startTime="0" endTime="1219.4481875">
<Turn startTime="0" endTime="1219.4481875">
<Sync time="20.109"/>
thank you very much //thank you very much
<Sync time="21.849"/>
gertrude mondello //gertrude mondello
<Sync time="23.869"/>
for your dedicated work that has brought us to this point //for your dedicated work that has brought us to this point
<Sync time="28.369"/>
distinguished delegates and guests //distinguished delegates and guests
<Sync time="31.419"/>
i would like to thank the secretary general //i would like to thank the secretary general
<Sync time="34.800"/>
for inviting me //for inviting me
```

SBV轉TRS

b. HTK切音 －

step1.將trs檔的時間與對應字串轉成多個lab檔

抓取切到句層級trs中的時間與對應字串，在時間部分轉換時間的格式；在相同時間的字串部分，將字串頭尾加上sil，並在字與字中間空白處替換成底線來連接成句。

將trs檔的時間與對應字串轉成多個lab檔

step2.將長語音檔依照lab檔所對應的時間來切割成多個短語音檔

將長語音檔依照lab檔中每行的時間進行切音，並轉存成多個短語音檔。

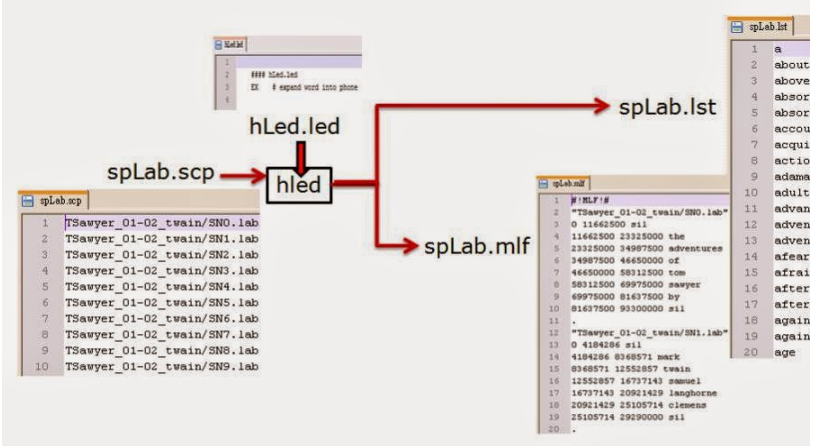
長語音檔依照lab檔所對應的時間來切割成多個短語音檔

step3.製造各個HtkTool所需的參數檔

這裡製造出7個參數檔，分別為hLed.led、hLed00.led、hCopy.conf、hInit.conf、hRest.conf、hErest.conf、hVite.conf

step4.處裡語音標籤及詞典

這邊主要在製作mlf檔，使用HTK 的hled工具。

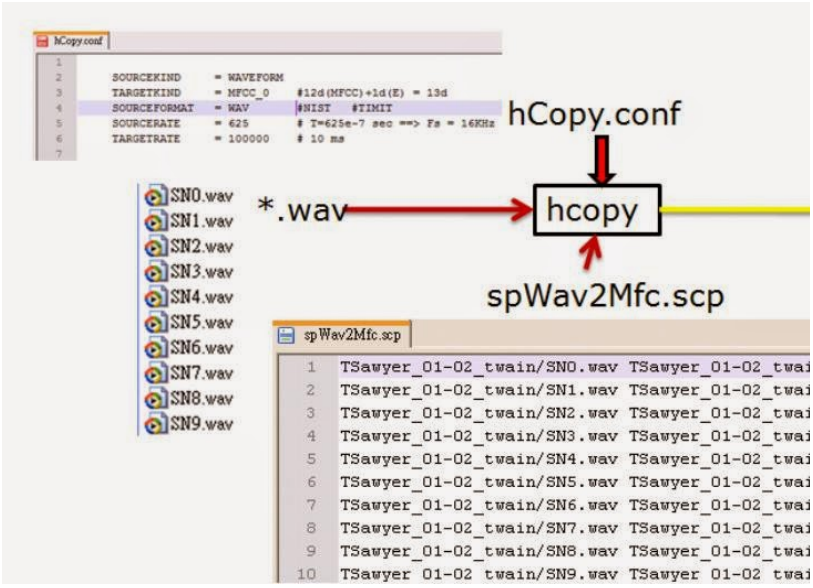


hled 製作標籤檔mlf

step5.擷取語音特徵及訓練語音模型

這裡分為聲音處理與模型訓練兩個部分。

1).聲音處理 – 對WAV檔做特徵擷取，而使用HTK的hcpoy工具。



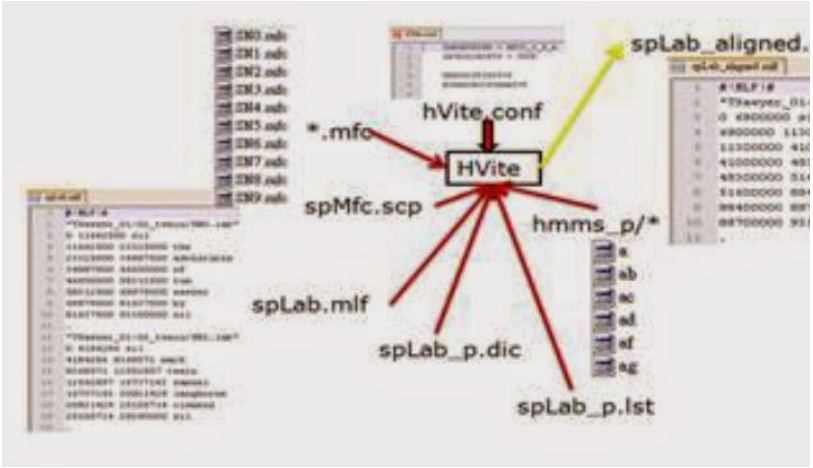
hcopy 製作特徵檔mfc

2).模型訓練 ─ 取得標籤檔mlf和特徵檔mfc後， 用HCompV做模型的初始訓練， 並用hEReset做精緻化訓練。

HCompV模型初始訓練

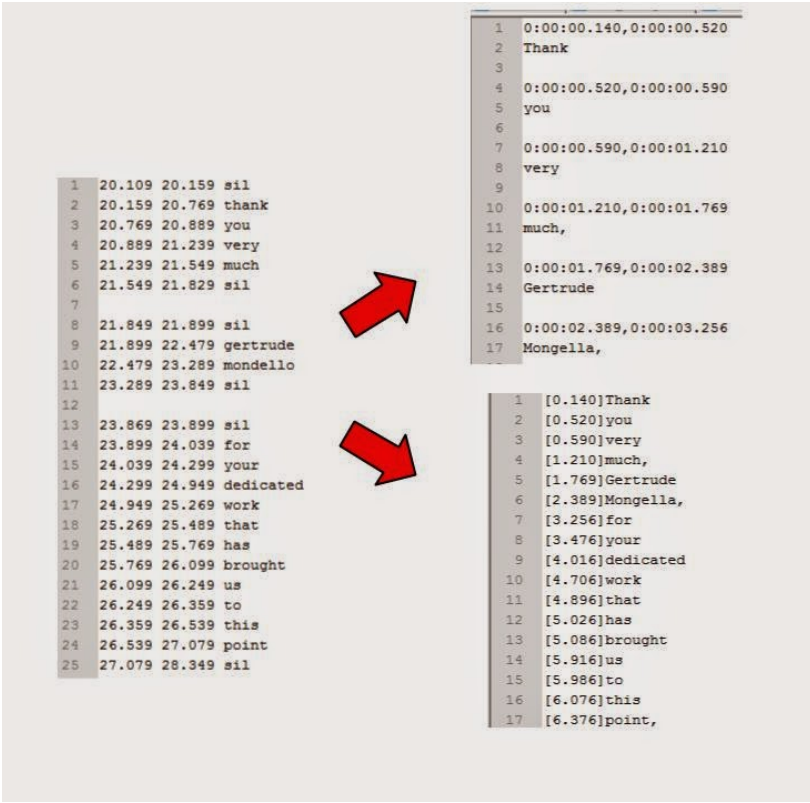
step6.將語音文字做對齊

這邊為語音切割的部分， 使用維特比演算法─HVite強制對齊(Forced alignment)的功能。



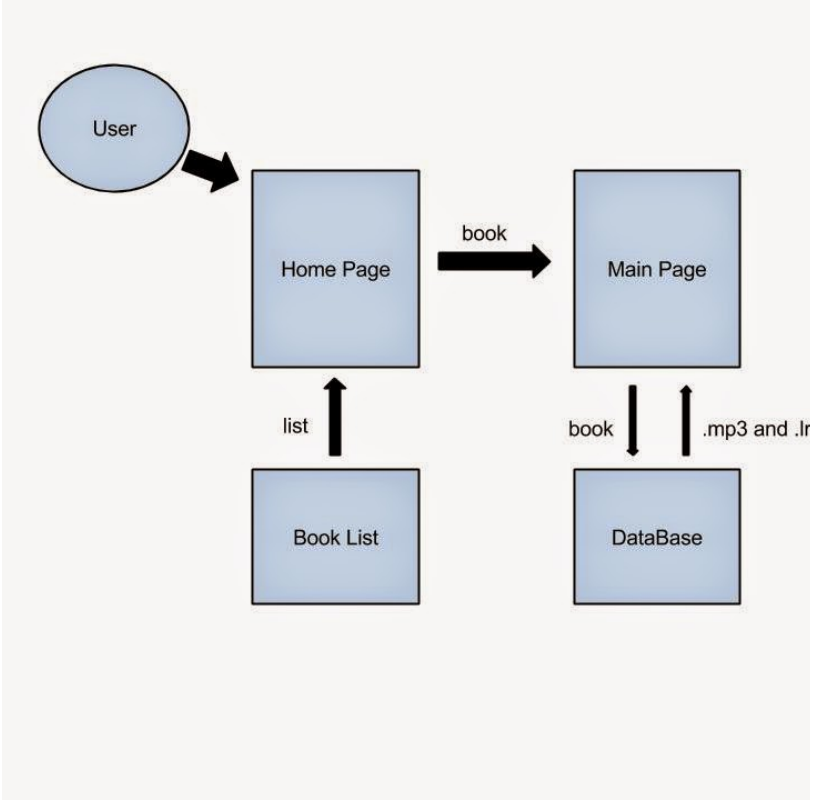
Hvite處理過程

c. **字串處理轉其他格式** – 在這部分主要將帶有切到Word-level的標籤檔lab與原文單字做對齊，並轉成一般較常見的Timed-text 格式，諸如LRC、SBV..... 等等。



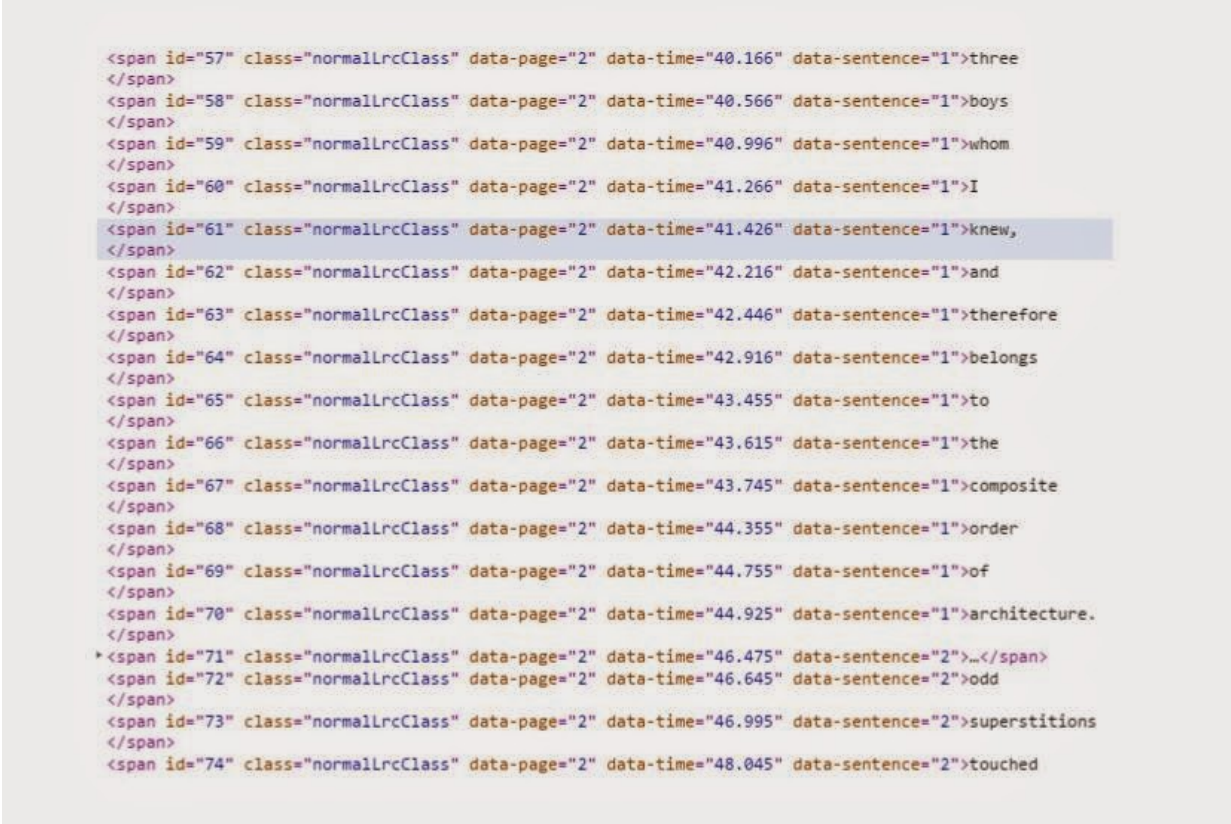
Forced alignment 後的lab檔轉成其他格式

7. **Website Presentation：**
   利用javascript製作一個簡單的能夠讀取Timed-text檔案─LRC檔的網頁。



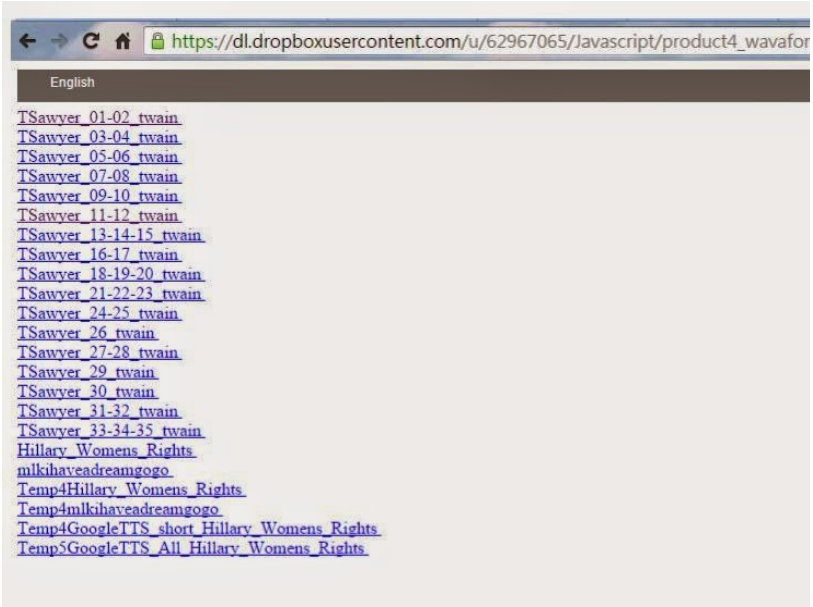
流覽網站流程圖

1.首頁會讀取書籍清單並呈現給使用者選擇，讀取的方法是用XMLHttpRequest()
2.使用者選擇書籍之後，回傳送book的參數並導向到主頁面。
3.主頁面會根據所收到的book參數，決定跟資料庫索取所需要的檔案。
4.主頁面讀取lrc檔上的時間標籤和文字，並使用HTML5上的Audio Tag，利用每個文字上的時間標籤，達到音文同步的效果，並且有Random Access的功能。
5.每個文字都有自己的id，利用此id能夠知道每次點擊的文字，並且將此文字以同Google TTS的方法，傳送至Google Translate做即時的翻譯。

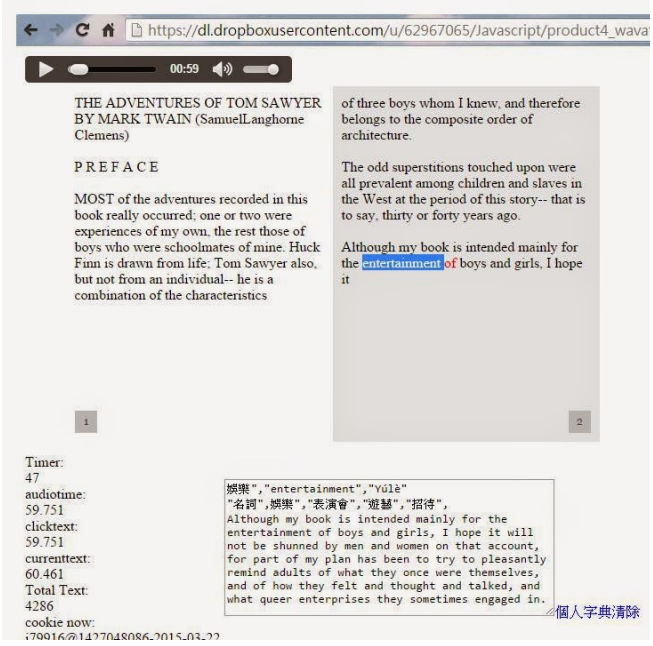


擁有時間標籤等資訊的文字

實驗結果與後端系統呈現：



Home Page



Main Page

Reference：

[1] 黃偉杰，語音辨識之音文對齊技術應用於音文同步有聲音之建立，長庚大學，民國101年
https://drive.google.com/file/d/0ByPRx5aruSFcRnVMc3E0aTljZWM/view?usp=sharing
[2]Google Translate https://cloud.google.com/translate/docs
[3]Google 的語音合成 API 之使用 (作者：陳鍾誠)
http://programmermagazine.github.io/201309/htm/article2.html
[4]FFmpeg https://www.ffmpeg.org/
[5]Python http://www.python.org/