The 2015 Conference on Computational Linguistics and Speech Processing
ROCLING 2015, pp. 289-305

運用 Python 結合語音辨識及合成技術於自動化音文同步之實作

A Python Implementation of Automatic Speech-text Synchronization Using Speech Recognition and Text-to-Speech Technology

賴俊翰　Chun-Han Lai
長庚大學資訊工程學系
Department of Computer Science and Information Engineering
Chang Gung University
j79916@hotmail.com

張朝凱　Chao-Kai Chang
長庚大學資訊工程學系
Department of Computer Science and Information Engineering
Chang Gung University
aw.81761109@gmail.com

呂仁園　Renyuan Lyu
長庚大學資訊工程學系
Department of Computer Science and Information Engineering
Chang Gung University
renyuan.lyu@gmail.com

摘要

本研究設計一個方便處理有聲書音文同步的技術，利用雲端的文字轉語音(Text-to-speech)技術，結合語音辨識(Speech Recognition)技術，讓使用者能夠使用自行準備的文章來製作自己的『跟述練習』(Shadowing technique)的學習素材，製作達到詞層級(Word-level)的音文同步有聲書。此音文同步有聲書是藉由『帶時間點的文字』 (Timed-text)檔案所製作，而帶時間點的文字則是由使用者所提供的文章連同對應的語音聲波檔案，經由一套名為 CGUAlign 的音文同步技術之處理所產生的。CGUAlign 是運用 Python 將一有名的語音辨識技術—HTK(Hidden Markov Model Toolkit) 包裝，只要提供文字檔及其朗讀的語音檔，其中語音檔是經由雲端語音合成技術而得來的，即能製作出音文同步的帶時間點的文字檔案，隨後，我們也建立一個簡易的以 JavaScript 製作的網站，能夠運用這個檔案做電腦輔助語言學習(Computer-assisted language learning, CALL)之用，此網站能夠閱讀音文同步有聲書，讓使用者能夠較輕鬆的做跟述練習，最後我們也提供即時翻譯的功能來達到電腦輔助語言學習的目標。

Abstract

In this study, we establish a method to create speech and text synchronized audiobooks with “speech recognition” and “cloud text-to-speech” technology. The user can prepare his own arbitrary articles to create the learning materials for "Shadowing technique" with this method. Besides, the materials are made by "word-level" speech and text synchronized audiobooks. These audiobooks are created by "timed-text" files, and the files are produced from the user's articles and corresponding speech files. By synchronization for speech and text technology, named "CGUAlign", user can easily make the "Timed-text" files. CGUAlign, uses Python to wrap the well-known speech recognition technology—HTK(Hidden Markov Model Toolkit). Just providing text file and the corresponding speech file, obtained from cloud text-to-speech technology, CGUAlign can create the timed-text file to achieve the synchronization of speech and text. Subsequently, we also build a simple website created with JavaScript. This website can use the timed-text file as CALL(Computer-assisted Language Learning) purposes. Using the website, user can browse the synchronized audiobooks to easily do Shadowing technique. Finally this website also provides dictionary function to achieve the goal of CALL.

關鍵字：語音辨識、文字轉語音、雲端語音合成、隱藏式馬可夫模型工具程式庫、電腦輔助語言學習、音文同步

Keywords: Speech Recognition、Text-to-speech、HTK、Computer-assisted Language Leanring、Speech-text Syncronization

一、緒論

隨著地球村的趨勢來臨,「語言學習」是現今社會普羅大眾所需要面臨的一項課題，也是一種趨勢，因此培養良好的多國語言能力，已成為當今社會不可或缺的目標。針對於台灣人而言，英語學習的需求更是顯得比其他語言來得更為重要，事實上我們知道,「語言學習」並非只是如同一般課程的學習，又分為「聽」、「說」、「讀」、「寫」，其需要經過「自我內化」、「練習」、「演繹」等過程才能根深蒂固的記憶在我們腦海中，而在舊有的自我學習中，又缺乏獨特的語言學習環境，缺乏練習的對象，如果要請他人來指導教學，往往又所費不貲，而數位化雲端學習在當今的世代是一個熱門的趨勢，如「視訊教學」、「線上學習」，這些都是網路普及與資訊發展下的重要產物，若我們可以利用適度的電腦回饋結合數位化學習，也許能為更多使用者造就一個新形態的自我學習方式。

本研究設計一個方便處理有聲書音文同步的技術，利用雲端的文字轉語音(Text-to-speech)技術，結合語音辨識(Speech Recognition)技術，讓使用者能夠使用自行準備的文本來製作自己的跟述練習的學習素材，製作達到詞層級(Word-level)的音文同步有聲書，其不僅可以提供音文同步的電子書供使用者閱讀文章，也可以讓使用者藉由朗誦文章的方式，並透過跟述練習的實作和即時翻譯的效果，以達到

自我內化學習及增進語言能力。

二、相關研究

(一) 跟述練習(Shadowing Technique)

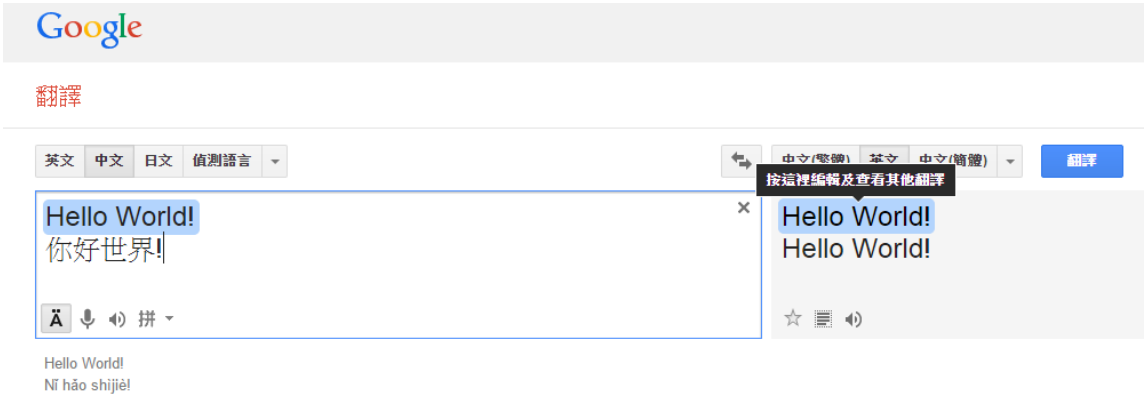
Shadowing technique 是一種語言學習技巧，一般我們稱之為跟述練習或者是影子練習，與目前台灣所常見的講述式教學法不同，它比較相似於所謂的聽說式教學法，但其又與聽說式教學法有一點點不同，跟述練習與聽說教學法較為不同的地方在於聽說教學是教師以自身的演繹口說內容來促使學生反覆練習語言內容而跟述練習比較傾向於學習者自主訓練的方式。

語言學習者跟述的學習對象不一定為真人，也可能僅是一個語音或影像檔，在跟述的過程中，跟述者以自我所能的發音技巧以及閱讀能力去盡可能地模仿所要學習的語言對象或者是影音內容，這種學習方式有如鸚鵡學舌，是一種反覆練習以及自我內化的過程，在其他的研究中[1]我們也可以發現到利用這樣的語言學習技巧是一種快速內化方式去學習一種語言的方法。

(二) Google Translate

現在 Google 在許多方面廣泛地被使用，不只是在搜尋引擎的功能上，許多人在遇到語言上問題的時候，往往會藉由 Google 所提供的翻譯功能— Google Translate 幫忙。 Google Translate 所提供的翻譯功能非常強大，提供近百種的語言相互的翻譯，而且在取得此功能的便利性上，也是無與倫比，據 Google 統計，至 2015 年 6 月 Google Translate 每天需要處理超過 1000 億筆字詞。

圖一是 Google Translate 的使用介面，其提供的即時翻譯功能，讓使用者可以在左邊的輸入欄位輸入文字，翻譯結果會即時在右邊的結果框顯示，將滑鼠鼠標移到翻譯結果文字上可以看到其對應的原文，圖一範例為將中文翻譯成英文的範例。



圖一、Google Translate 網頁介面

除此之外，Google Translate 也提供朗讀的功能，即文字轉語音(Text-to-speech)的人工語音合成朗讀的功能，另外也提供查詢文字拼音的功能，即能夠提供非拼音語言的羅馬拼音查詢。

Google Translate 所提供的三種功能—翻譯、朗讀、拼音已經很適合做初步的語言學習，但是其頂多只能製作出句層級(Sentence-level)的效果，其句層級是指在音文同步播放時當下語音內容是以句子為單位的顯示於畫面中。由而本研究則是進一步利用這三種功能，利用拼音和人工語音合成的功能能夠製作出 Google Translate 所無法達到的詞層級(Word-level)的音文同步有聲書，其詞層級是指在音文同步播放時當下的語音內容除了以句子為單位的顯示於畫面中並附加句子中每個詞的顯示效果，而詞層級的音文同步有聲書能夠讓學習者更容易的耳聽、眼看來做跟述技巧的練習。

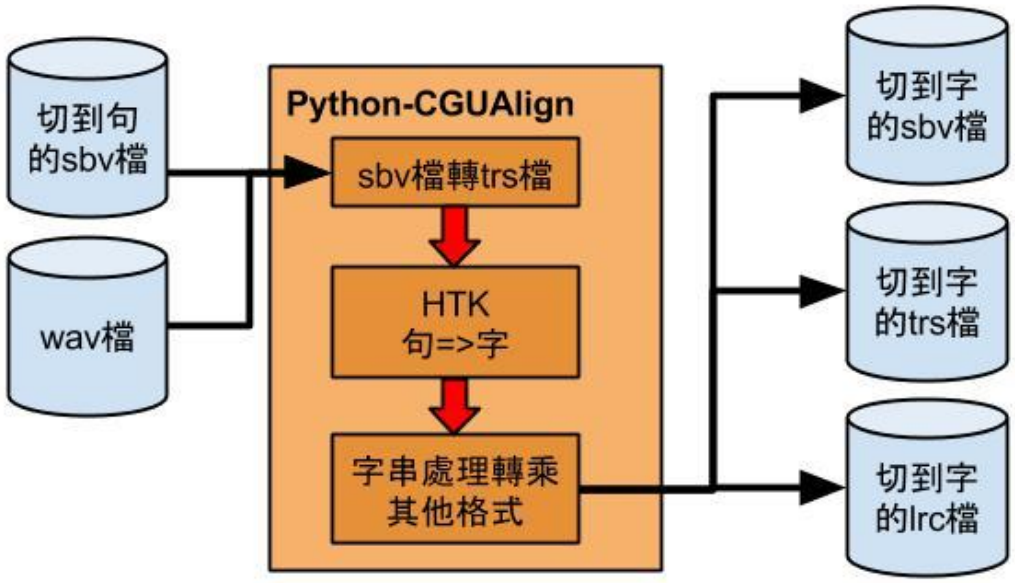
(三) HTK (Hidden Markov Model Toolkit)[2]

HTK 的全名為 Hidden Markov Model Toolkit，是一套應用於語音訓練與辨識的免費軟體。HTK 於 1989 年開始由英國劍橋大學工程系 (Cambridge University Engineering Department, CUED)的機器智能實驗室 (Machine Intelligence Lab，或是大眾較為熟悉的 Speech Vision and Robotics Group)進行開發，該團隊利用隱藏式馬可夫模型 (HMM)建造出一套 HMM-based 的語音辨識系統。1999 年十一月，微軟購入擁有此軟體的 Entropic 公司，並於翌年將 HTK 定位為免費軟體，期望 HTK 作為語音辨識的共同平台，便能豐富 HTK 的功能性，以及提升語音辨識等相關技術。為了達到這個目標，HTK 建置官方網站，以提供開放的完整功能原始碼及說明書。

由於語音辨識的原理包含相當高深的數學，相對地使得程式碼也不易撰寫，造成進入門檻高，複雜度不易掌控的情況產生。但自從 HTK 在 2000 年定位成開放原始碼的免費軟體後，大幅降低了進入門檻，並加速提昇語音技術的發展，綜觀目前國內外語音技術相關的實驗工具和系統開發，絕大部分都以 HTK 為主流；由此可知，HTK 在語音技術的研究領域占了不可或缺的地位。

(四) CGUAlign[3]

CguAlign 是模仿[4]以 Perl 包裝 HTK 的方法，CguAlign 改用 Python 將 HTK 包裝、運用的一套技術，為本實驗室一個方便處理音文同步有聲書的技術，原本是為了將從 Youtube 上所取得的句層級 Timed-text sbv 檔，以程式自動切音取代傳統人工手動的方式切音成詞層級 Timed-text 檔的方法，此方法除了可以減少人力資源，還能夠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。圖二為 CGUAlign 之流程圖。



圖二、CGUAlign 流程圖

```
0:0:0.000000,0:0:1.619000
Thank you very much,

0:0:1.619000,0:0:3.022000
Gertrude Mongella,

0:0:3.022000,0:0:6.549000
for your dedicated work that has brought us to this point,

0:0:6.549000,0:0:8.276000
distinguished delegates,

0:0:8.276000,0:0:9.391000
and guests:

0:0:9.391000,0:0:13.494000
I would like to thank the Secretary General for inviting me to

0:0:13.494000,0:0:18.389000
be part of this important United Nations Fourth World Conference on Women.

0:0:18.389000,0:0:20.548000
This is truly a celebration,

0:0:20.548000,0:0:25.407000
a celebration of the contributions women make in every aspect of life:
```

圖二.a、切到句的 sbv 檔

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Trans SYSTEM "trans-14.dtd">
<Trans scribe="wjLabtoTrs" audio_filename="Input/Hillary_Womens_Rights-1.wav">
<Episode>
<Section type="report" startTime="0" endTime="26.676">
<Turn startTime="0" endTime="26.676">
<Sync time="0.000"/>
I would like to thank the Secretary General for inviting me to  //I would like to thank the Secretary General for inviting me to
<Sync time="4.104"/>
be part of this important United Nations Fourth World Conference on Women.  //be part of this important United Nations Fourth World Conference on Women
<Sync time="9.000"/>
This is truly a celebration,  //This is truly a celebration
<Sync time="11.160"/>
a celebration of the contributions women make in every aspect of life: //a celebration of the contributions women make in every aspect of life
<Sync time="16.020"/>
in the home,  //in the home
<Sync time="16.920"/>
on the job,  //on the job
<Sync time="17.892"/>
in the community,  //in the community
<Sync time="19.152"/>
as mothers,  //as mothers
<Sync time="20.232"/>
wives,  //wives
<Sync time="21.024"/>
sisters,  //sisters
<Sync time="21.996"/>
daughters,  //daughters
<Sync time="22.860"/>
learners,  //learners
<Sync time="23.760"/>
```

圖二.b、trs 檔

```
[0.050]I
[0.080]would
[0.650]like
[0.760]to
[0.850]thank
[1.530]the
[1.920]Secretary
[2.430]General
[2.940]for
[3.060]inviting
[3.720]me
[3.960]to
[4.224]be
[4.254]part
[4.624]of
[4.784]this
[4.974]important
[5.614]United
[6.214]Nations
[6.794]Fourth
[7.144]World
[7.494]Conference
[8.224]on
[8.374]Women.<br/><br/>
[9.100]This
[9.290]is
[9.610]truly
[9.970]a
[10.000]celebration,
[11.250]a
[11.310]celebration
```

圖二.c、lrc 檔

三、研究方法

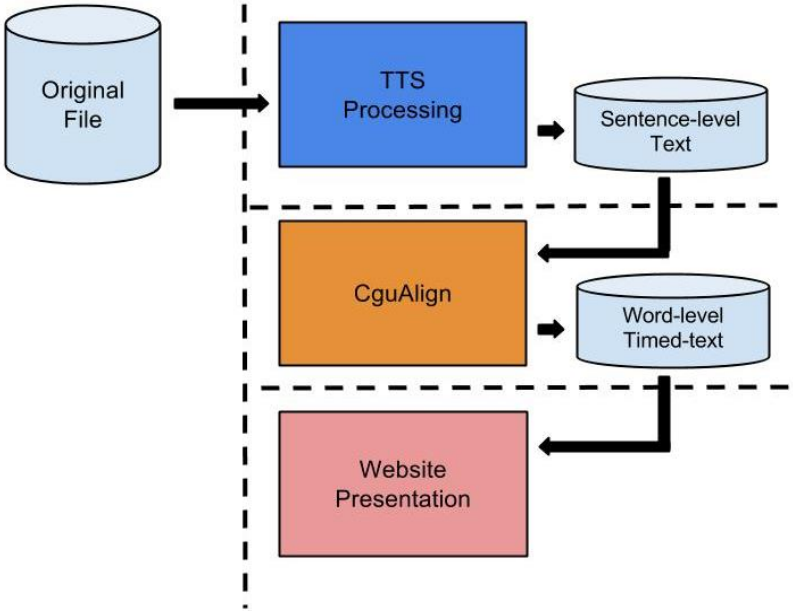
本章節內容旨在介紹整體的研究方法，全章分為三節，

(一) 雲端語音合成(Text-to-speech,TTS)

(二) CGUAlign 語音辨識-ForceAlignment

(三) 網站呈現(Website presentation)

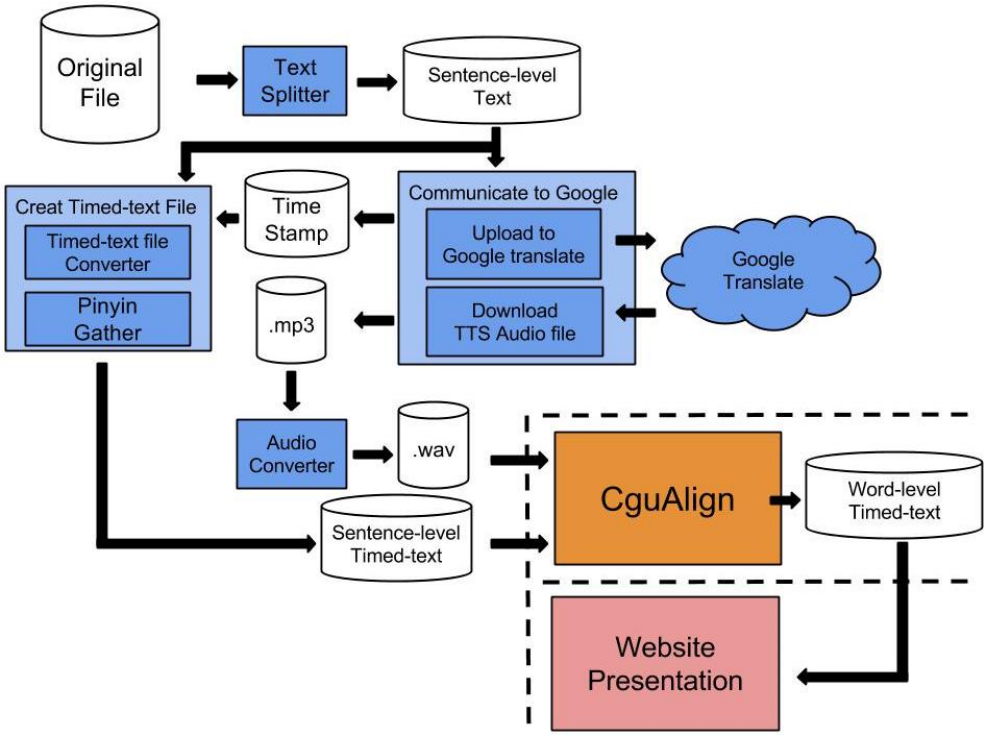
圖三為系統整體流程圖，原始文字檔經由(一)會得到句層級的帶有時間點的文字的檔案，再經由(二)會得到詞層級的帶有時間點的文字的檔案，最後經由(三)能夠以音文同步有聲書的方式瀏覽此帶有時間點的文字的檔案，並以此做一個電腦輔助語言學習的動作。



圖三、系統流程圖

(一) 雲端語音合成(Text-to-speech,TTS)

本節將說明如何將純文本經由文字的預先處理，透過 Google Translate 的雲端文字轉語音(Text-to-speech,TTS)的服務，取得 TTS 的語音檔，並將此語音檔和文字檔結合產生句層級的 Timed-text 檔案以供下一階段 CGUAlign 使用。



圖四、雲端語音合成流程圖

1. 文字切割

因 Google Translate 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，並稱此為句層級的純文字檔，基本的切割方法只先按照標點符號作切割。

Step1:按照標點符號做切割例如:

"句號"(“.”,”。”)、"問號"(“?”,”？”)、" 驚嘆號"(“!”,”！”)、"破折號"(“-”,”—”)、"冒號"(“:”,”：”)、"逗號"(“,”,”，”)。

Step2: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。

2. 連結 Google 發出請求

此節將討論如何藉由 Google Translate 的 TTS 服務將純文字轉成 TTS 的語音檔案，利用 Python 的 Standard Library—” urllib.request"和"urllib.parse"，傳送 HTTP GET Request 至 Google Translate 的 URL，其 URL 為：

http://translate.google.com/translate_tts

其 URL 的 parameters 如表一。

表一、Google Translate TTS Parameters

parameters	意義
tl	Target Language，目標語言，表示要文字 TTS 的語言種類。
q	Query，欲 TTS 的文字。
total	Total number of text segments，文章分段的個數。
idx	Index of text segments，文章分段的指標。
textlen	String length in this segment，此 Query 的字串長度。

```
1  import urllib.request
2     import urllib.parse
3     savefile="./TTS.mp3"
4     f= open(savefile, 'wb+')
5     文字= "Chung Gung University Student"
6     GOOGLE_TTS_URL= 'https://translate.google.com.tw/translate_tts?'
7     payload = { 'ie': 'utf-8',
8                 'tk': '308912',
9                 'client': 't',
10                'tl': 'en',
11                'q':  文字,
12                'total': 1,
13                'idx':   0,
14                'textlen': len(text) }
15    try:
16        hdr = {'User-Agent':'Mozilla/5.0'}
17        data = urllib.parse.urlencode(payload)
18        req = urllib.request.Request(GOOGLE_TTS_URL+data,headers=hdr)
19        r = urllib.request.urlopen(req)
20
21
22        byte= r.read()
23        f.write(byte)
24        byteNum= len(byte)
25    except Exception as e:
26          raise
27    f.close()
```

圖五、Communicate to Google 範例程式碼

在此範例程式碼中，先對設定 GOOGLE_TTS_URL 輸入網址，用 payload 輸入對 Google Translate 的 TTS 之參數如上述表一所示，最後運用 urllib.request.urlopen() 發出 request 取得句子內容的 mp3 語音，然後用 read()讀取 mp3 語音的內容並用 len( )計算出句子 mp3 音訊內容的長度。

3. Creat Timed-text File

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其公式如下，SegmentLength(i)為第 i 段語音的長度，ByteNum(i)為第 i 段語音的檔案大小，Sum(ByteNum)為總共的檔案大小，TotalLength 為總語音長度。

$$SegmentLength(i) = \frac{ByteNum(i)}{Sum(ByteNum)} \times TotalLength$$

計算出時間之後即可使之與句層級的文字檔黏合，製成句層級的 Timed-text 檔案。

```
Thank you very much,
Gertrude Mongella,
for your dedicated work that has brought us to this point,
distinguished delegates,
and guests:

0:0:0.000000,0:0:1.619000
Thank you very much,


0:0:1.619000,0:0:3.022000
Gertrude Mongella,


0:0:3.022000,0:0:6.549000
for your dedicated work that has brought us to this point,


0:0:6.549000,0:0:8.276000
distinguished delegates,


0:0:8.276000,0:0:9.391000
and guests:
```

圖六、句層級的文字檔轉成 Timed-text 檔案範例

若輸入文本不為英文時，需要從 Google Translate 取得其原文的羅馬拼音，並且將此羅馬拼音取代原文，將句層級的原文文字檔轉成句層級的羅馬拼音檔，而若利用 Google Translate 取得拼音，其也會幫我們做斷詞的動作。

利用同 Communicate to Google 的方法，只要將 URL 改成，

http://translate.google.com.tw/translate_a/single

以及其 parameter 改成如表二，即可得到此原文的羅馬拼音。

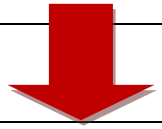
表二、取得羅馬拼音的 parameters(以中文為例)

parameters	值	parameters	值
ie	UTF-8	kc	1
inputm	1	tk	520254|125262
oe	UTF-8	dt	bd
otf	1	dt	ex
trs	1	dt	ld
client	T	dt	md
sl	Zh-CN	dt	qca
hl	Zh-TW	dt	rw
rom	1	dt	rm
srcrom	1	dt	ss
ssel	0	dt	t
tsel	0	dt	at
tl	目標語言(zh-TW)	q	欲取得拼音的文字

```
0:0:0.000000,0:0:5.760000
話說山東登州府東門外有一座大山，名叫蓬萊山。

0:0:5.760000,0:0:9.144000
山上有個閣子，名叫蓬萊閣。

0:0:9.144000,0:0:13.968000
這閣造得畫棟飛雲，珠簾捲雨，十分壯麗。
```



```
0:0:0.000000,0:0:5.760000
Huàshuō shāndōng dēng zhōu fǔ dōngmén wài yǒu yīzuò dàshān, míng jiào pénglái
shān.

0:0:5.760000,0:0:9.144000
Shānshàng yǒu gè gé zi, míng jiào pénglái gé.

0:0:9.144000,0:0:13.968000
Zhè gé zào dé huà dòng fēi yún, zhū lián juǎn yǔ, shífēn zhuànglì.
```

圖七、非英文文字 sbv 檔轉羅馬拼音 sbv 檔

4. Audio Converter

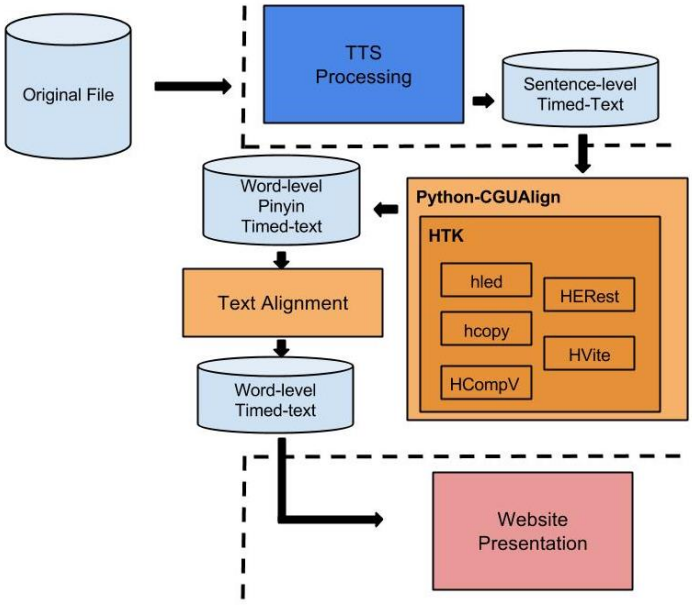
本節旨在說明如何將 Google Translate TTS 所得的 mp3 檔案轉成 CguAlign 能接受的 wav 檔案，使用自由軟體—FFmpeg 來幫助轉檔，FFmpeg 可以執行音訊和視訊多種格式的錄影、轉檔、串流功能，因需借助 FFmpeg 的幫助，而 FFmpeg 屬於外部程式，在 Python 中若需要呼叫外部的程式，需要 import os 模組，並且使用 os.system() 函式來呼叫 FFmpeg。

```
def ffmpeg AudioDuration(filename):
    os.system("ffmpeg -report -y -i ./TTS-MP3/{0}.mp3" +
        "./FFmpeg-WAV/{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+" .\\FFmpeg-WAV\\"+report name)
    os.system("del "+report_name)
    return total_ms
```

圖八、Audio Converter 範例程式碼

(二) CGUAlign 語音辨識-Force Alignment

本章節將說明如何將(一)雲端語音合成(Text-to-speech,TTS)得到的句層級 Timed-text 檔案經由 CGUAlign 的語音辨識，對齊成詞層級的 Timed-text 檔案。其流程圖如圖八所示。

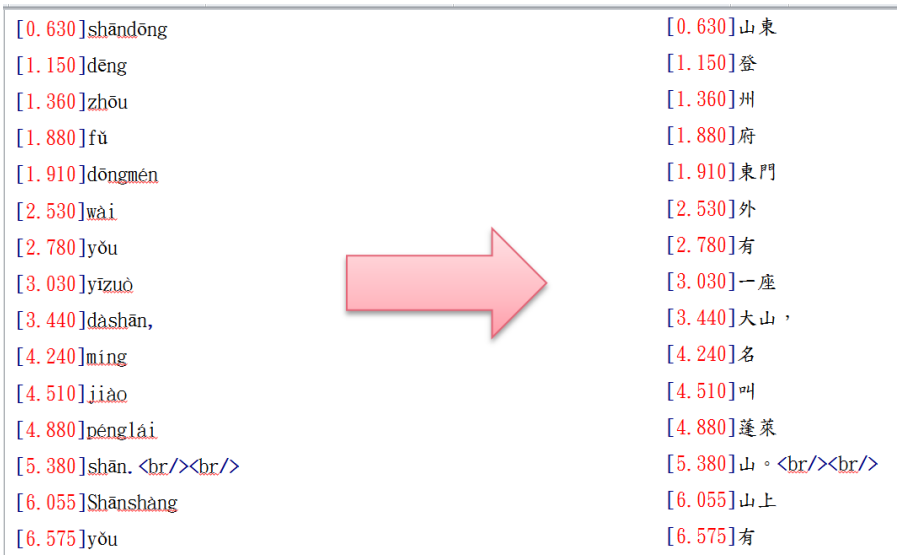


圖九、CGUAlign 語音辨識-Force Alignment 流程圖

將經由(一)所得到的句層級帶有時間點的文字檔案經由 CGUAlign 所包裹的 5 個 HTK 工具—

1. Hled：語音標籤及詞典處理
2. Hcopy：語音特徵擷取
3. HCompV：語音模型訓練
4. HERest：語音模型反覆、精緻化的訓練
5. HVite：語音文字做對齊

就會得到詞層級帶有時間點的文字檔案，而若是處理非英文時，則還需經過 Text Alignment 的動作才能夠得到詞層級帶有時間點的文字檔案，如圖十所示。



圖十、Text Alignment(以中文為範例)

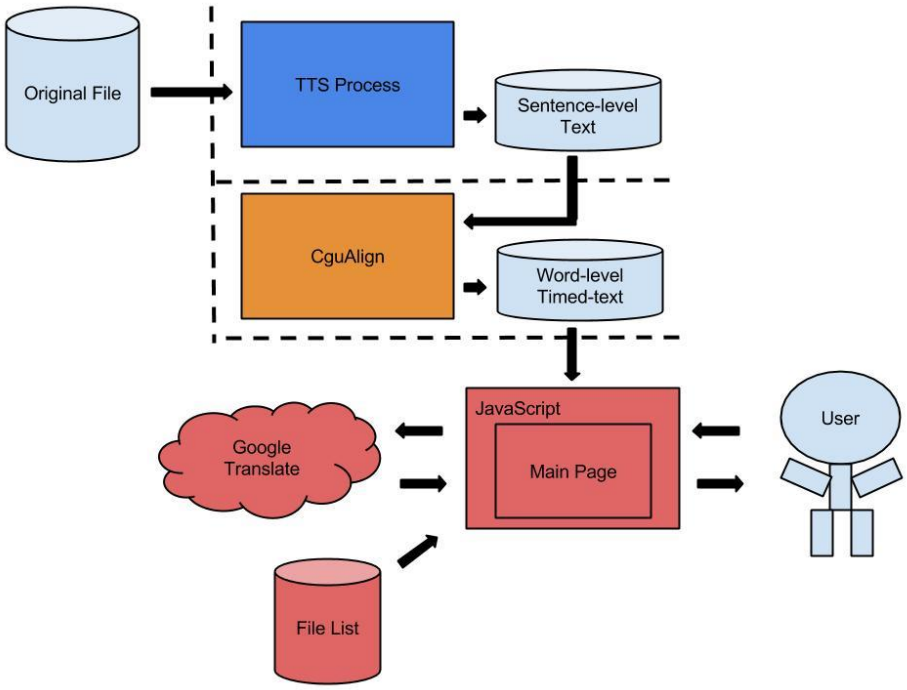
運用中文的特性，每個字只有一個音節，而每個音節母音必定相連的規則下，利用聲調母音表，能夠計算出每段拼音的中文字數，將原文依計算出來的字數做對齊。

表三、聲調母音表

ā	ē	ī	ō	ū
á	é	í	ó	ú
ǎ	ě	ǐ	ǒ	ǔ
à	è	ì	ò	ù
a	e	i	o	u

(三) 網站呈現

利用 JavaScript 製作一個簡單的能夠讀取 Timed-text 檔案—lrc 檔的網頁，其流程圖如圖十一所示。



圖十一、網站呈現流程圖

主頁面會讀取書籍清單並呈現給使用者選擇，而書籍清單是以 txt 檔做儲存，讀取的方法是用 XMLHttpRequest()。主頁面會根據使用者選取書籍清單的書籍，會傳送 book 參數去撈取資料庫的 lrc 檔和 wav 檔。

```
var bookFileName="YourTextFile.txt";
var Textfile=new XMLHttpRequest();
```

```
Textfile.open("GET",bookFileName,false);
Textfile.send(null);
var BookData=Textfile.responseText;
```

圖十二、XMLHttpRequest()讀取文字檔範例程式碼

將 lrc 檔以 XMLHttpRequest()讀取之後，將 lrc 檔轉化成如下圖所示，每個單字都會有它的 id 編號、高亮記號、頁數、起始時間點…等等資訊。

```
</span><span id="152" class="normalLrcClass" page="2" time="103.410" entence="0" style="background: yellow;">our
</span><span id="153" class="normalLrcClass" page="2" time="103.640" entence="0" style="background: yellow;">talk
</span><span id="154" class="normalLrcClass" page="2" time="104.480" entence="0" style="background: yellow;">turns
</span><span id="155" class="normalLrcClass" page="2" time="104.900" entence="0" style="background: yellow;">to
</span><span id="156" class="normalLrcClass" page="2" time="105.080" entence="0" style="background: yellow;">our
</span><span id="157" class="normalLrcClass" page="2" time="105.270" entence="0" style="background: yellow;">children
</span><span id="158" class="normalLrcClass" page="2" time="106.040" entence="0" style="background: yellow;">and
</span><span id="159" class="normalLrcClass" page="2" time="106.230" entence="0" style="background: yellow;">our
</span><span id="160" class="normalLrcClass" page="2" time="106.410" entence="0" style="background: yellow;">families
</span><span id="161" class="normalLrcClass" page="2" time="108.320" entence="0" style="background: yellow;">however
</span><span id="162" class="normalLrcClass" page="2" time="108.770" entence="0" style="background: yellow;">different
</span><span id="163" class="normalLrcClass" page="2" time="109.460" entence="0" style="background: yellow;">we
```

圖十三、將 lrc 檔轉成網頁上的標籤資訊

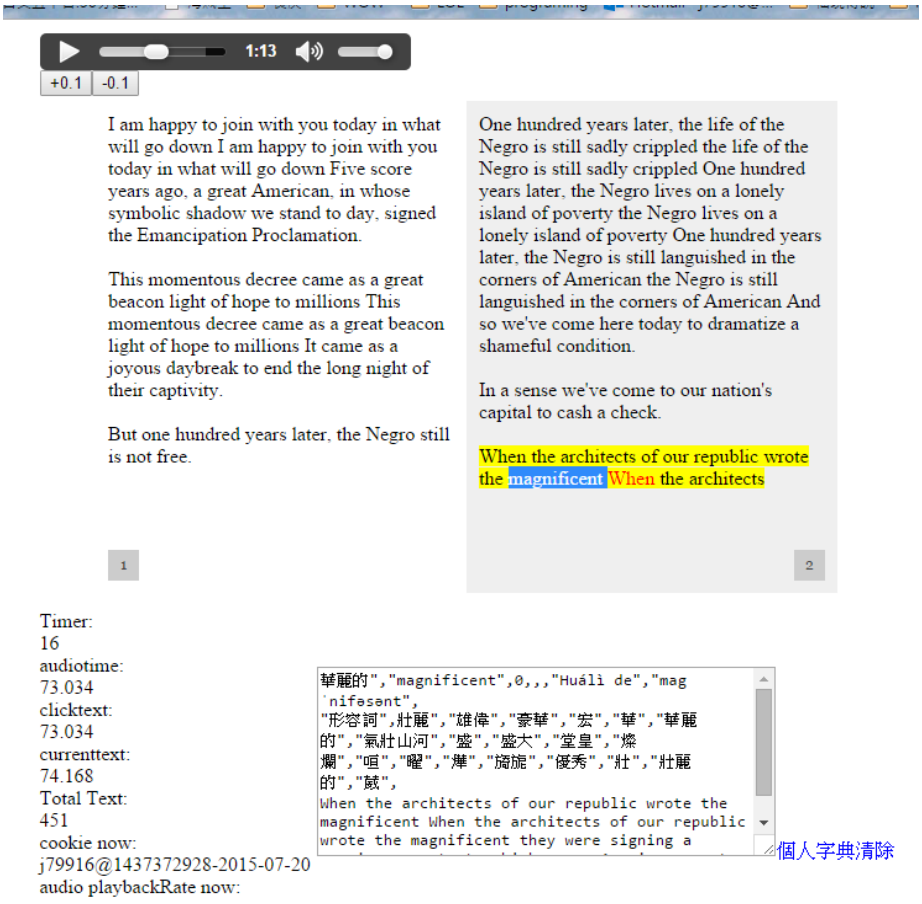
使用 HTML5 新增的 audio 標籤，能夠先創建一個播放音訊的物件，賦予此物件一個獨特的 ID—mainAudio，再利用 HTML DOM 的物件，能夠指定 mainAudio 要讀取的音訊檔案以及此音訊的一些資訊。

```
1 var audioFileName="YourAudioFile.wav"
2 document.write("<div><audio id='mainAudio' src='' controls=controls /></div>");
3 document.getElementById("mainAudio").src=audioFileName;
4 var playrate= mainAudio.playbackRate
5 document.getElementById("playrate").innerHTML =playrate;
6 var time= mainAudio.currentTime;
7 document.getElementById("audiotime").innerHTML =time;
```

圖十四、讀取音訊的範例程式碼

在圖十四的範例中，在第 2 行先創立一個 audio 物件，並且在第 3 行指定此物件所要讀取的音訊檔案，第 5 行、第 7 行能夠得到此音訊的播放速度和目前所撥放的音訊時間點，此音訊時間點是用來做音文同步非常重要的資訊。

以類同前述取得中文拼音的方法，不僅能夠取得單字的拼音，同樣也能夠取得單字的翻譯，我們可以用此功能來實作線上查詢字典的功能，與拼音取得的方法不同的是，翻譯功能必須指定好 sl 和 tl 兩個參數，其意義代表 source language 來源語言和 target language 目標語言。



圖十五、線上即時翻譯範例

四、結論

本研究的目的是利用純文字檔轉成語音檔的技術(Text-to-speech)結合語音辨識(Speech-recognition)中的音文對齊技術(Speech-text Synchronization)製作能夠以電腦輔助語言學習(Computer-assisted Language Learning)為目標，幫助語言學習者能夠借助此系統較輕鬆地實現跟述學習法(Shadowing technique)的一個系統。

在此系統中，使用者能夠自由地取得任何想跟述的素材的文本，以此文本，借助本系統，能夠從 Google Translate 取得此文本的 TTS 語音檔及其已對齊的帶有時間點的文本(Timed-text)，不同於以往較常見的句層級(Sentence-level)的文本，本系統運用語音辨識技術能夠製作出詞層級(Word-level)的文本，以此帶有時間點的文本藉由我們的網站瀏覽，即是一本音文同步的電子有聲書，在此網站上，不僅可以進行跟述學習法學習，也可以做一個線上查詢字典的功能，其不僅可以提供音文同步的電子書供使用者閱讀文章，也可以讓使用者藉由朗誦文章的方式，並透過跟述學習法的實作和即時翻譯的效果，以達到自我內化學習及增進語言能力。

參考文獻

[1] 戴安娜,*跟述練習對口譯課學生的聽力之影響*,台灣科技大學,2013.

[2] Steve Young, *The HTK Book version 3*, Microsoft Corporation, 2000. http://htk.eng.cam.ac.uk/

[3] 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，2012.

[4] A. Katsamanis, M. P. Black, P. G. Georgiou, L. Goldstein, S. Narayanan"*SailAlign: Robust long speech-text alignment*" University of Southern California, Los Angeles, CA, USA , Jan. 28-31, 2011.