Karaoke-style Read-aloud System Using Speech Recognition and Text-to-Speech Technology

Chun-Han Lai, Renyuan Lyu, ,Chang Gung University ,Taiwan.

Abstract

This demonstrate a system, which can synchronize arbitrary text with its corresponding speech, just like a karaoke machine, but now the contents are not song, instead they are normal text-based article, which are very useful for language learning purpose. The Technology embedded in such a system include, Text-to-speech, Speech-recognition, and a web browsing demo program. By the Text-to-speech, the Google Translate is adopted as a kernel technology, we use Python standard library to access this technology. By the Speech-text alignment, a Speech-recognition technology, called HTK (Hidden Markov Model Toolkit), which is adopted by using Python to wrap the HTK executable programs. By the web-based browsing, a JavaScript program is used to show time-aligned high-light text on the popular browser.

Description

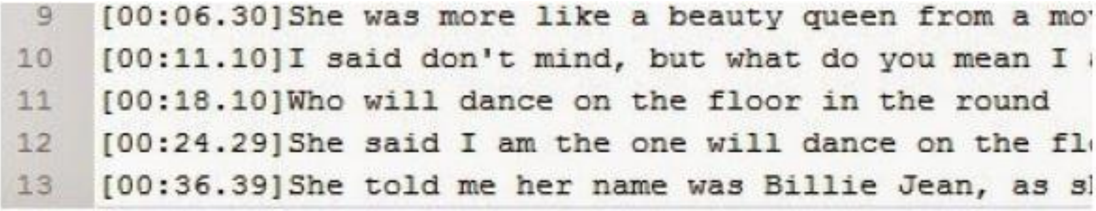
Using Cloud TTS (Text-to-speech) technology, such as Google Translate, iSpeech ... etc, and simple Speech-recognition technology, can be made to establish a method to create a Speech-text Synchronization file from original text-only file, that file can be used to show time-aligned high-light text like karaoke, which are very useful for language learning purpose.

Related Work

1. Timed-text File :

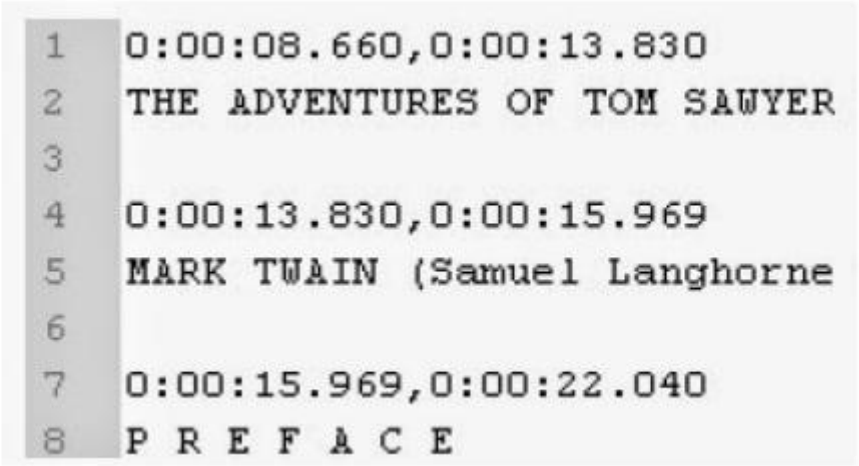
For task on speech-text synchronization, the first thing is to create some Timed-text file, The following are listed some example of Timed-text file, where you see some time tag are inserted into the original text. Please refer to Figure.1 ~ Figure.3 .

Python has excellent string processing ability, so we can use Python to help create timed-text file which can also be converted between various formats. Generally speaking, there are some popular timed-text format, like .lrc file, .srt file…etc.These timed-text formats are often used on song lyrics, movie subtitle.



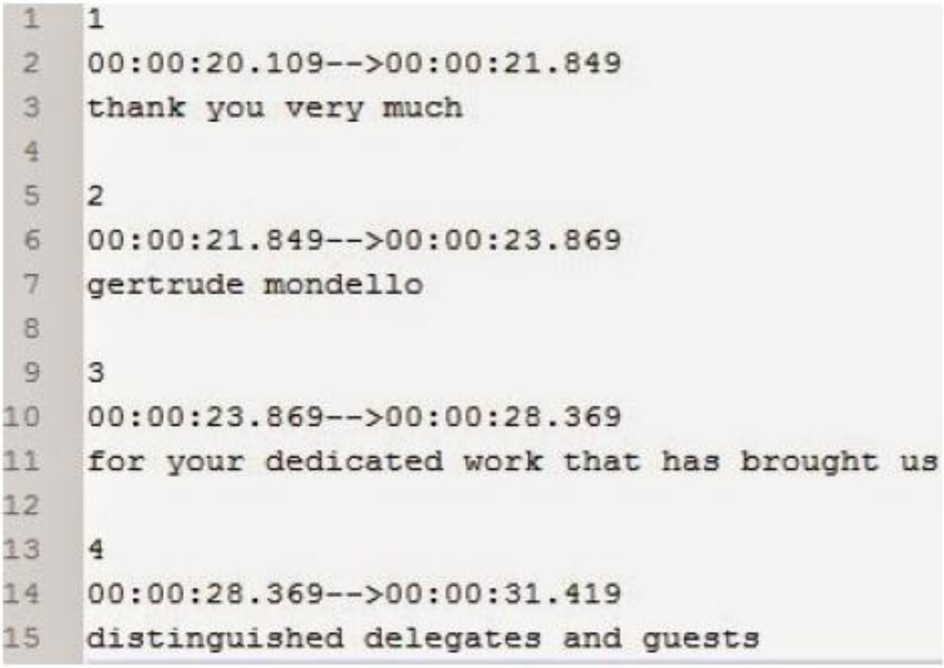
```
[00:06.30]She was more like a beauty queen from a mo
[00:11.10]I said don't mind, but what do you mean I
[00:18.10]Who will dance on the floor in the round
[00:24.29]She said I am the one will dance on the fl
[00:36.39]She told me her name was Billie Jean, as s
```

Figure 1. Lrc File Format



```
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
```

Figure 2. Sbv File Format



```
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
```

Figure 3. Srt File Format

2. TTS base on Google Translate:

Text-to-Speech (TTS) technology has made great progress during this years, which is not only good enough for language learning, but also easily accessible for web users. The TTS technology embedded in Google Translate is a good example. In this project, we adopt it as our kernel technology to make arbitrary text being read aloud.

By using Google Translate TTS, all we need do is using Python built-in library, 'urllib.request' and 'urllib.parse', that let us can communicate to Google Translate with sending HTTP Request.

The URL of Google Translate TTS is that -- http://translate.google.com/translate_tts

The parameters for this URL are also listed in the Figure.4 .

parameters	Meaning
tl	Target Language, The language you want to convert
q	Query, Query you want to Text to Speech
total	Total number of text segments.
idx	Index of text segments.
textlen	String length in this segment.

Figure 4. Google Translate TTS Parameters

We can also provide some example as follows :

Example :

- Query= I am a Chang Gung University Student
- tl = en (English), total = 1, idx = 0, textlen = 36.
- URL :

  http://translate.google.com.tw/translate_tts?q=I%20am%20a%20Chang%20Gung%20University%20Student&tl=en&total=1&idx=0&textlen=36

3. Speech-text Synchronization base on HTK :

Speech-text Synchronization is usually achieved through the Speech-recognition technology, which is not very popular in Python world. Here, we adopt a well-known C-based Speech-recognition open source software, called Hidden Markov Model Toolkit,

Speech-recognition involves quite profound mathematical, the code is not easy to write, high barriers to entry, the complexity is not easy to control. But since HTK from 2000 into freeware open source, significantly reducing barriers to entry, rapidly enhance the development of Speech technology.

There is a previous task done as similar purpose, called CGUAlign (https://drive.google.com/open?id=0ByPRx5aruSFcRnVMc3E0aTljZWM), which will be incorporated in this project.

System Functional Block

The whole system is split into three parts:

1. TTS Processing
2. CGUALIGN
3. Website Presentation

Which can be shown Figure 5.

Original files through the first layer, TTS Processing, will become the sentence-level text, and then after the second layer, CGUAlign processing, will become Word-level Timed-text file, and finally through the Website Presentation, enables the user to browse.

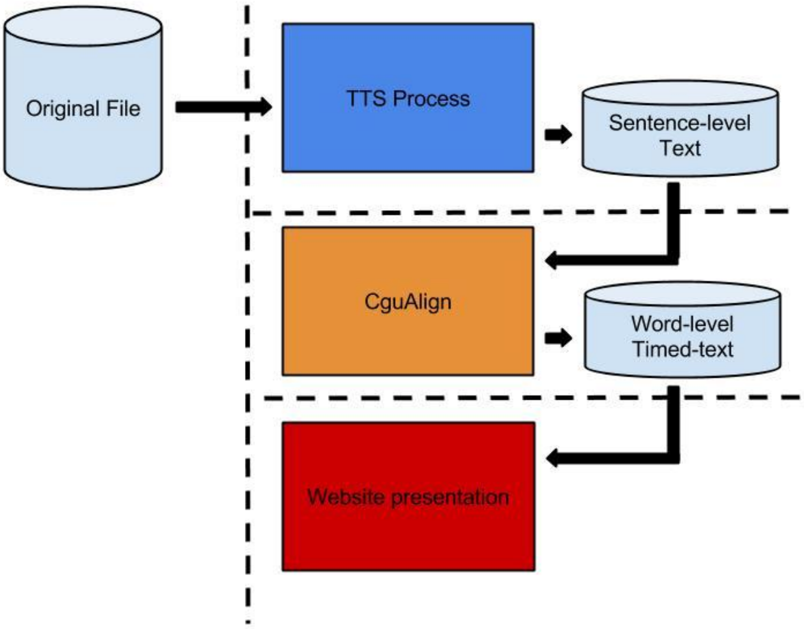


Figure 5 . Flow Diagram

1. TTS Processing

Here is split into five parts:

1.1. Text Splitter

1.2. Google Translate Uploader

1.3. TTS Audio file Downloader

1.4. Audio Converter

1.5. Timed-text file Converter

As shown below Figure 6.

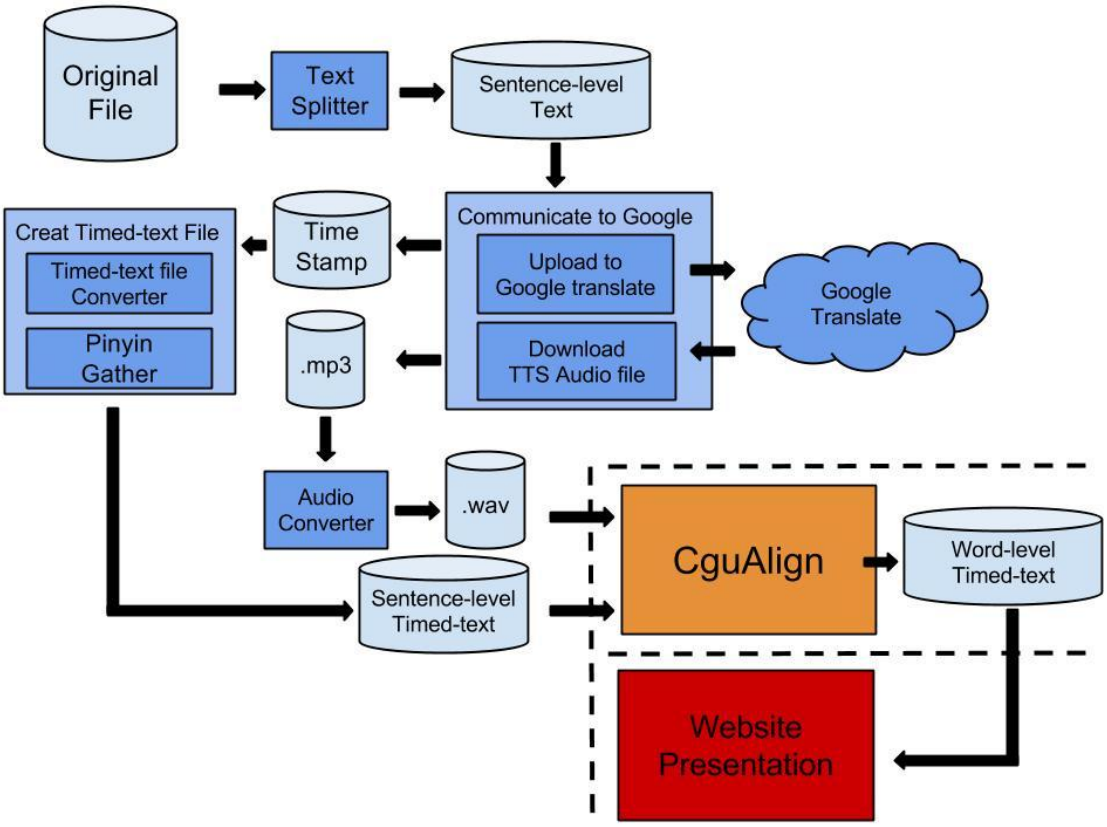
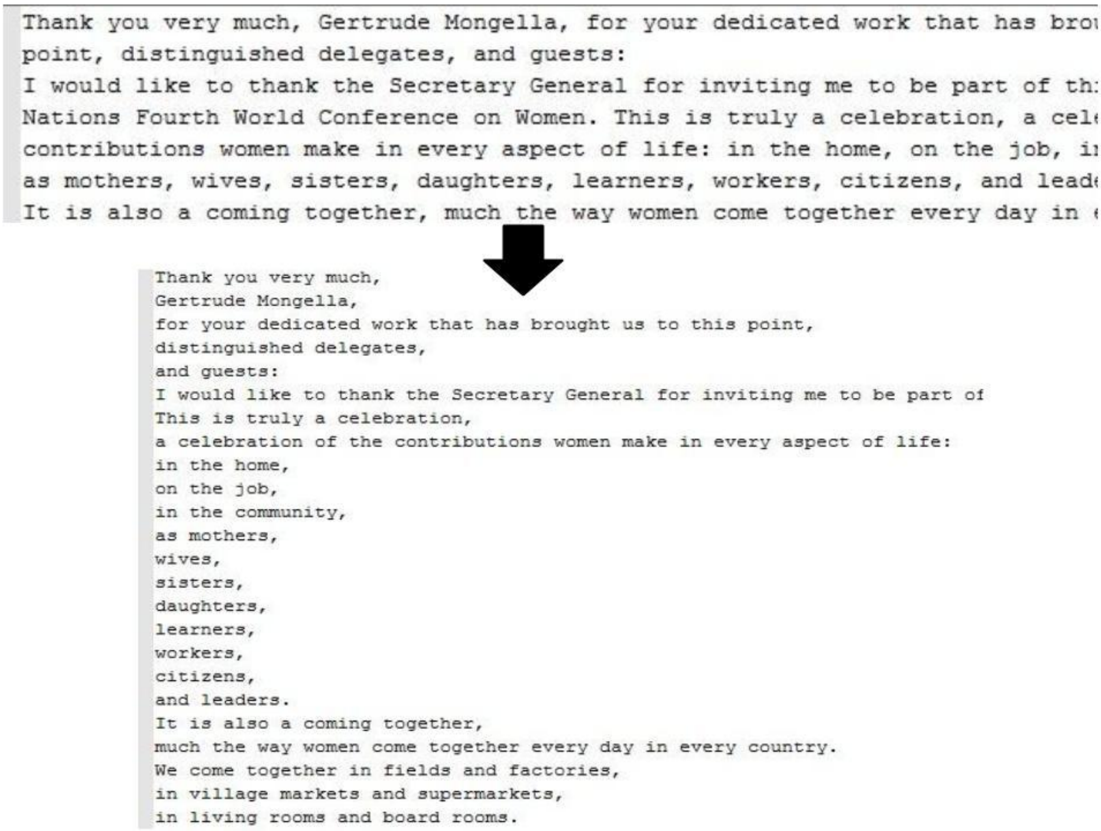


Figure 6. TTS Process Flow Diagram

1.1. Text Splitter:

The purpose of the Text Splitter is to split the whole article into smaller pieces, according to the punctuations and the other obvious cues of a written text.

It can be shown such as following Figure.7 .



```
Thank you very much, Gertrude Mongella, for your dedicated work that has brou
point, distinguished delegates, and guests:
I would like to thank the Secretary General for inviting me to be part of th
Nations Fourth World Conference on Women. This is truly a celebration, a cel
contributions women make in every aspect of life: in the home, on the job, i
as mothers, wives, sisters, daughters, learners, workers, citizens, and lead
It is also a coming together, much the way women come together every day in
```

```
Thank you very much,
Gertrude Mongella,
for your dedicated work that has brought us to this point,
distinguished delegates,
and guests:
I would like to thank the Secretary General for inviting me to be part of
This is truly a celebration,
a celebration of the contributions women make in every aspect of life:
in the home,
on the job,
in the community,
as mothers,
wives,
sisters,
daughters,
learners,
workers,
citizens,
and leaders.
It is also a coming together,
much the way women come together every day in every country.
We come together in fields and factories,
in village markets and supermarkets,
in living rooms and board rooms.
```

Figure 7. Text Splitter Example

In this project, we use the following steps to achieve such a goal.Here we use Python built-in String Function – **String.replace()**.

(1) In accordance with "full stop" to do the cutting, if the string length are less than 100, the cut end, or continue to cut.

(2) The "question mark" do the cutting, if the string length are less than 100, the cut end, or continue to cut.

(3) In accordance with "exclamation point" to do the cutting, if the string length are less than 100, the cut end, or continue to cut.

(4) In accordance with "dash" to do the cutting, if the string length are less than 100, the cut end, or continue to cut.

(5) In accordance with "colon" do the cutting, if the string length are less than 100, the cut end, or continue to cut.

(6) The "comma" to do the cutting, if the string length are less than 100, the cut end, or continue to cut.

(7) If the final string length still has more than 100 from more than 100 the string will be in the middle of the "space" cut.

1.2. Google Translate Uploader

Here, we use Python built-in library, **‘urllib.request’** and **‘urllib.parse’**, to communicate with Google Translate. That can send HTTP Request to Google Translate. The URL of Google Translate TTS is that --

http://translate.google.com/translate_tts

The Example Code is following :

```
import urllib.request
import urllib.parse
savefile="./TTS.mp3"
f= open(savefile, 'wb+')
文字= "Chung Gung University Student"
GOOGLE_TTS_URL= 'https://translate.google.com.tw/translate_tts?'
payload = { 'ie': 'utf-8',
          'tl': 'en',
          'q':  文字,
          'total': 1,
          'idx':   0,
          'textlen': len(text) }
try:
    hdr = {'User-Agent':'Mozilla/5.0'}
    data = urllib.parse.urlencode(payload)
    req = urllib.request.Request(GOOGLE_TTS_URL+data,headers=hdr)
    r = urllib.request.urlopen(req)


    byte= r.read()
    f.write(byte)
    byteNum= len(byte)
except Exception as e:
        raise
f.close()
```

Figure 8. Google Translate Uploader Example Code

1.3. TTS Audio file Downloader

Follow Figuare 9. line 4,20,21,22.

Line 4 and 21 are open a file and save the TTS file to mp3 by binary write.

Line 20 is decode the HTTP Response from Google Translate.

Line 21 is record every TTS file capacity by byteNum. This can be used for the next step ‘Timed-text file Converter’.

```
 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              'tl': 'en',
 9              'q':  文字,
10              'total': 1,
11              'idx':   0,
12              'textlen': len(text) }
13    try:
14        hdr = {'User-Agent':'Mozilla/5.0'}
15        data = urllib.parse.urlencode(payload)
16        req = urllib.request.Request(GOOGLE_TTS_URL+data,headers=hdr)
17        r = urllib.request.urlopen(req)
18
19
20        byte= r.read()
21        f.write(byte)
22        byteNum= len(byte)
23    except Exception as e:
24            raise
25    f.close()
```

Figure 9. Download TTS Audio File Example

1.4. Timed-text file Converter

The use of the previous step collected byteNum size of each segment, and the sum of the calculated byteNum, it is possible to calculate the length of each segment in the total period of the speech length of time formula is as follows,

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

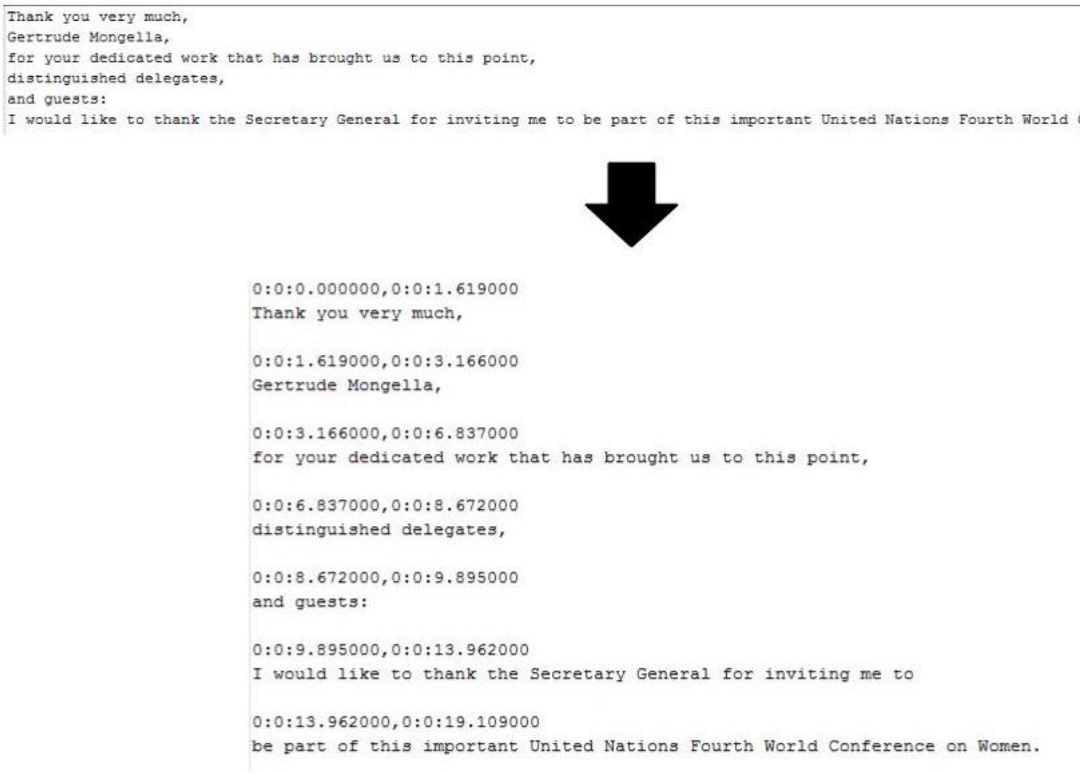
SegmentLength (i) : the length of segment i.

ByteNum (i) : the size of segment i.

Sum (ByteNum) : total file size.

TotalLength : total speech length.

When calculated the length of every segment, we can make the Sentence-level of Timed-text files by Sentence-level text file and every segment's length.



```
Thank you very much,
Gertrude Mongella,
for your dedicated work that has brought us to this point,
distinguished delegates,
and guests:
I would like to thank the Secretary General for inviting me to be part of this important United Nations Fourth World
```

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

0:0:1.619000,0:0:3.166000
Gertrude Mongella,

0:0:3.166000,0:0:6.837000
for your dedicated work that has brought us to this point,

0:0:6.837000,0:0:8.672000
distinguished delegates,

0:0:8.672000,0:0:9.895000
and guests:

0:0:9.895000,0:0:13.962000
I would like to thank the Secretary General for inviting me to

0:0:13.962000,0:0:19.109000
be part of this important United Nations Fourth World Conference on Women.
```

Figure 10. Timed-text file Converter Example

1.5. Audio Converter

In order to use HTK, we need to convert TTS mp3 file to wav file. Here we use another free software – ‘FFmpeg ’, it produces libraries and programs for handling multimedia data. To use FFmpeg in Python, we import Python built-in module, named ‘os’, which can help us to call External program, like FFmpeg, HTK…etc. The function we need in os is os.system().

```
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_m
```

Figure 11. Audio Converter Example Code

2. CGUAlign

Here is split into five parts:

- Hled
- Hcopy
- HCompV
- HERest
- HVite

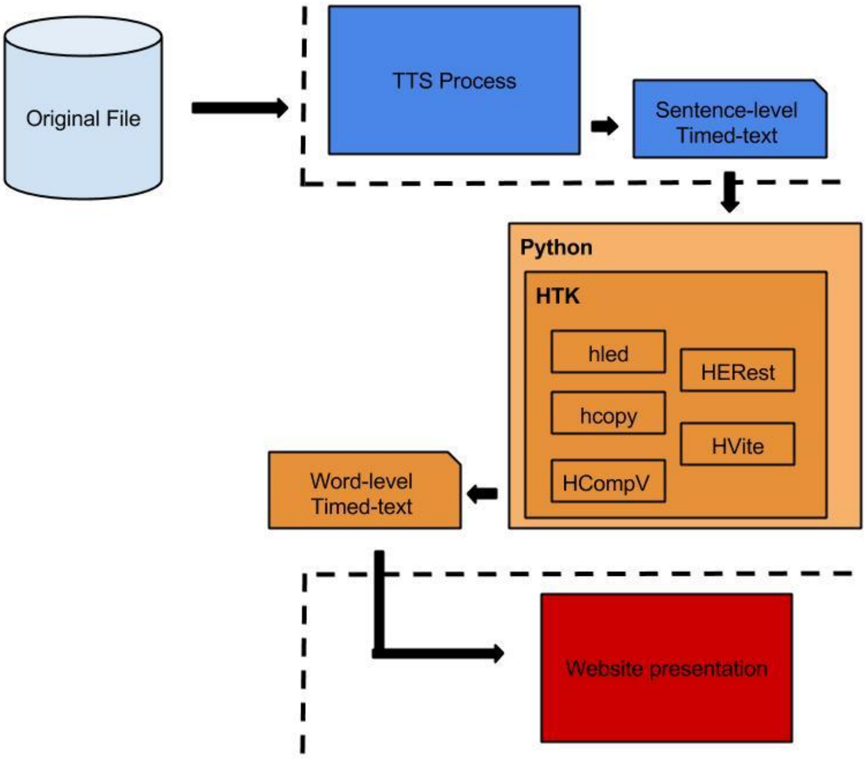


Figure 12. CGUAlign Flow Diagram

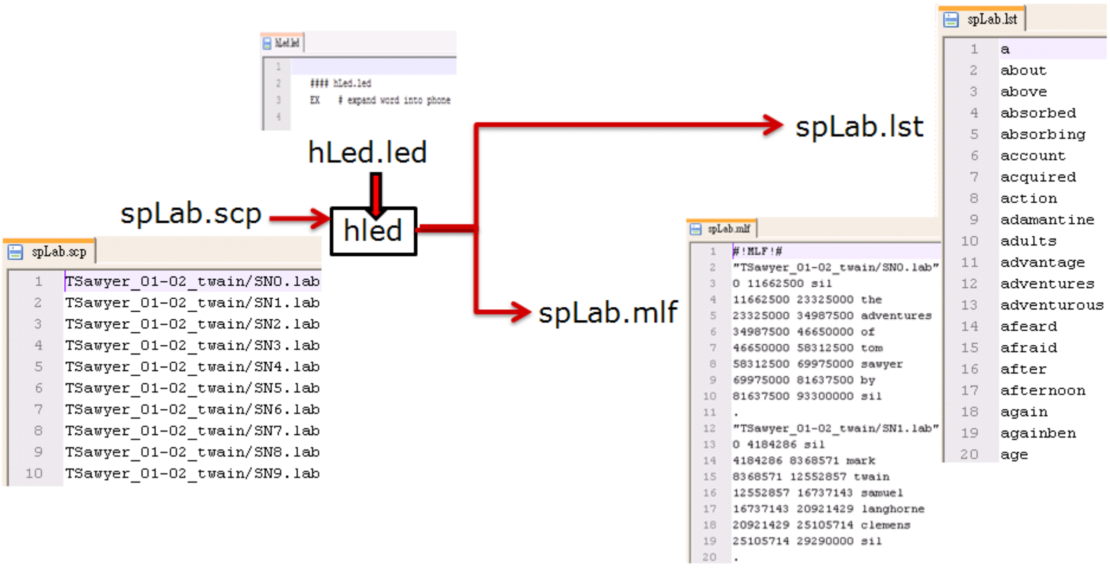
- Hled

The main purpose of Hled is to deal with the linguistic labels.

Here we call **os.system()** function to call Hled.exe .

```
def htk01_處裡語音標籤及詞典():

        os.system('hled -A -i spLab00.mlf -n spLab00.lst -S spLab.scp  hLed00.led')

        os.system('hled -A -i spLab.mlf -n spLab.lst -S spLab.scp  hLed.led')

        lst2dic()

        os.system('hled -A -i spLab_p.mlf -n spLab_p.lst -S spLab.scp -I spLab.mlf -d spLab_p.dic hLed.led'
```

Figure 13. Hled Example Code



```

#### hLed.led
EX    # expand word into phone

```

```
TSawyer_01-02_twain/SN0.lab
TSawyer_01-02_twain/SN1.lab
TSawyer_01-02_twain/SN2.lab
TSawyer_01-02_twain/SN3.lab
TSawyer_01-02_twain/SN4.lab
TSawyer_01-02_twain/SN5.lab
TSawyer_01-02_twain/SN6.lab
TSawyer_01-02_twain/SN7.lab
TSawyer_01-02_twain/SN8.lab
TSawyer_01-02_twain/SN9.lab
```

```
#!MLF!#
"TSawyer_01-02_twain/SN0.lab"
0 11662500 sil
11662500 23325000 the
23325000 34987500 adventures
34987500 46650000 of
46650000 58312500 tom
58312500 69975000 sawyer
69975000 81637500 by
81637500 93300000 sil
.
"TSawyer_01-02_twain/SN1.lab"
0 4184286 sil
4184286 8368571 mark
8368571 12552857 twain
12552857 16737143 samuel
16737143 20921429 langhorne
20921429 25105714 clemens
25105714 29290000 sil
.
```

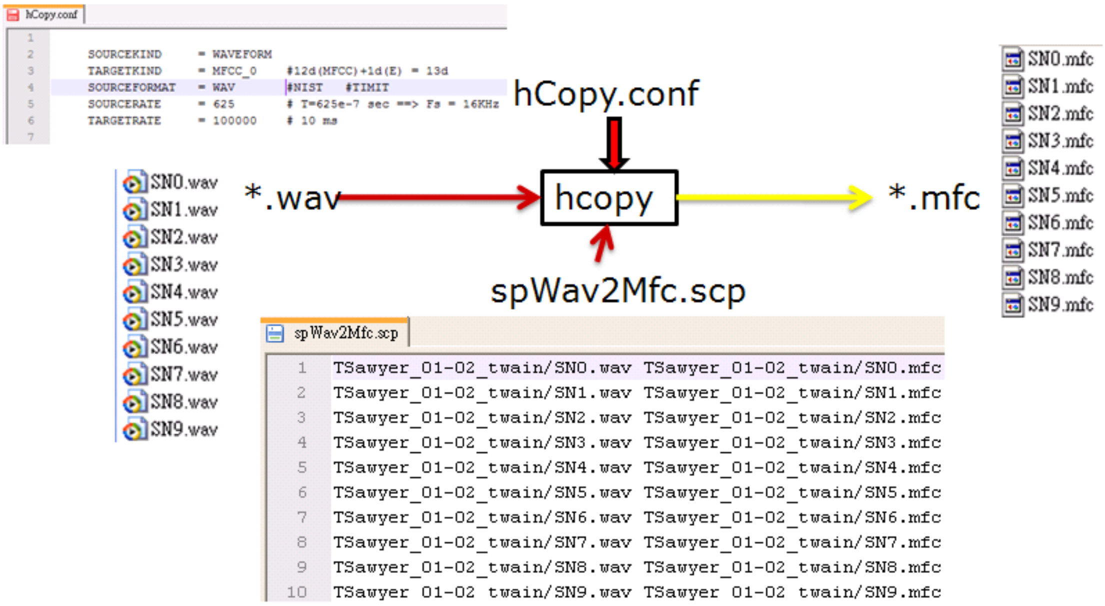
```
a
about
above
absorbed
absorbing
account
acquired
action
adamantine
adults
advantage
adventures
adventurous
afeard
afraid
after
afternoon
again
againben
age
```

Figure 13. Hled Processing Example

- Hcopy

The main purpose of Hcopy is to extract speech feature.

Here we also call the function, **os.system()** , to call Hcopy.exe .



hCopy.conf

```

SOURCEKIND      = WAVEFORM
TARGETKIND      = MFCC_0      #12d(MFCC)+1d(E) = 13d
SOURCEFORMAT    = WAV         #NIST   #TIMIT
SOURCERATE      = 625         # T=625e-7 sec ==> Fs = 16KHz
TARGETRATE      = 100000      # 10 ms

```

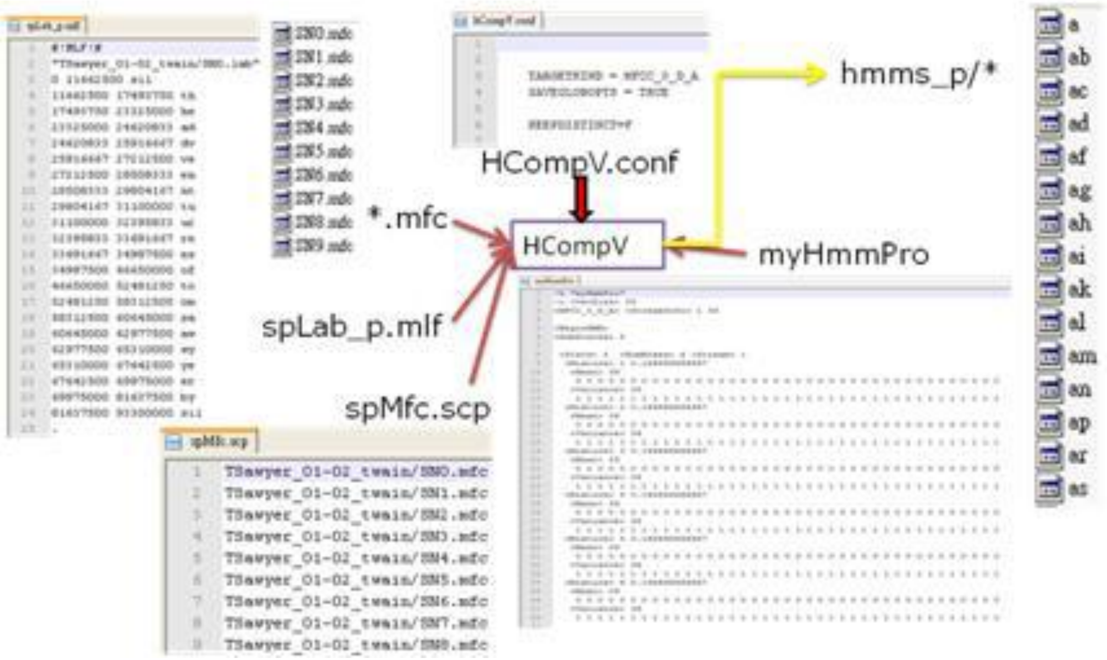
```
TSawyer_01-02_twain/SN0.wav TSawyer_01-02_twain/SN0.mfc
TSawyer_01-02_twain/SN1.wav TSawyer_01-02_twain/SN1.mfc
TSawyer_01-02_twain/SN2.wav TSawyer_01-02_twain/SN2.mfc
TSawyer_01-02_twain/SN3.wav TSawyer_01-02_twain/SN3.mfc
TSawyer_01-02_twain/SN4.wav TSawyer_01-02_twain/SN4.mfc
TSawyer_01-02_twain/SN5.wav TSawyer_01-02_twain/SN5.mfc
TSawyer_01-02_twain/SN6.wav TSawyer_01-02_twain/SN6.mfc
TSawyer_01-02_twain/SN7.wav TSawyer_01-02_twain/SN7.mfc
TSawyer_01-02_twain/SN8.wav TSawyer_01-02_twain/SN8.mfc
TSawyer 01-02 twain/SN9.wav TSawyer 01-02 twain/SN9.mfc
```

Figure 14. Hcopy Processing Example

- HCompV

The main purpose of Hcopy is to train proto language model.

Here we also call the function, **os.system()** , to call HCompV.exe .



```
TARGETKIND = MFCC_0_D_A
SAVEGLOBOPTS = TRUE

#KEEPDISTINCT=F
```

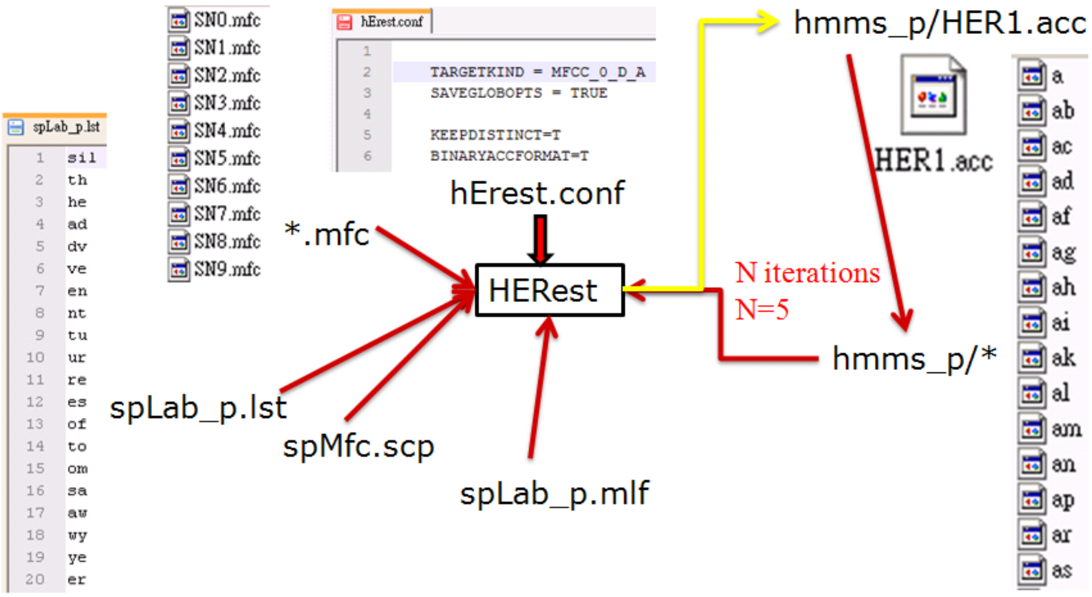
```
TSawyer_01-02_twain/SN0.mfc
TSawyer_01-02_twain/SN1.mfc
TSawyer_01-02_twain/SN2.mfc
TSawyer_01-02_twain/SN3.mfc
TSawyer_01-02_twain/SN4.mfc
TSawyer_01-02_twain/SN5.mfc
TSawyer_01-02_twain/SN6.mfc
TSawyer_01-02_twain/SN7.mfc
TSawyer_01-02_twain/SN8.mfc
```

Figure 15. HCompV Processing Example

- HERest

The main purpose of HERest is to embedded train language model.

Here we also call the function, os.system() , to call HERest.exe .



```
sil
th
he
ad
dv
ve
en
nt
tu
ur
re
es
of
to
om
sa
aw
wy
ye
er
```

```

TARGETKIND = MFCC_0_D_A
SAVEGLOBOPTS = TRUE

KEEPDISTINCT=T
BINARYACCFORMAT=T
```

Figure 16. HERest Processing Example

```
def htk02_擷取語音特徵及訓練語音模型():

    os.system('hcopy -A -C hCopy.conf -S spWav2Mfc.scp')


    os.system('mkdir hmms p')


    f=open('spLab p.lst',encoding='utf-8')
    lines=f.readlines()
    f.close()

    mList=[]
    for l in lines:
        m = l.strip('\n')
        mList.append(m)
    m = 'myHCompV'
    os.system('HCompV -A -C hInit.conf  -S spMfc.scp -m -I spLab_p.mlf -M hmms_p/ -o '+m+' myHmmPro')

    f=open('hmms p/'+m)
    myHCompV=f.read()
    f.close()

    for m in mList:
        myModel=myHCompV.replace('myHCompV',m)
        f=open('hmms p/'+m,'w',encoding='utf-8')
        f.write(myModel)
        f.close()

    for i in range(5):
        print('[%d]HERest '%i)
        os.system('HERest -A -C hErest.conf  -S spMfc.scp -p 1 -t 2000.0 -w 3 -v 0.05 -I
spLab p.mlf -M hmms p -d hmms p/ spLab p.lst')
        os.system('HERest -A -C hErest.conf -p 0 -t 2000.0 -w 3 -v 0.05 -I spLab p.mlf -M
hmms_p/ -d hmms_p/ spLab_p.lst hmms_p/HER1.acc')
```

Figure 17. HCopy, HCompV, HERest Example Code

- HVite

The main purpose of HVite is to forced alignment.

Here we also call the function, os.system() , to call HVite.exe .

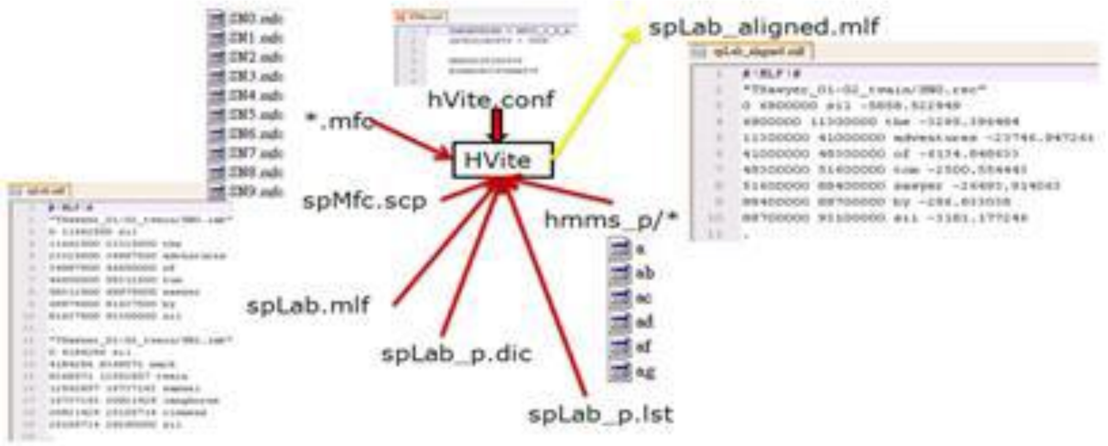


Figure 18. HVite Processing Example

```
def htk03_語音文字對齊():

    os.system('HVite -A -C hVite.conf  -S spMfc.scp  -a -d hmms_p/ -i
spLab_aligned.mlf -I spLab.mlf spLab_p.dic spLab_p.lst')
```

Figure 19. HVite Example Code

3. Website presentation

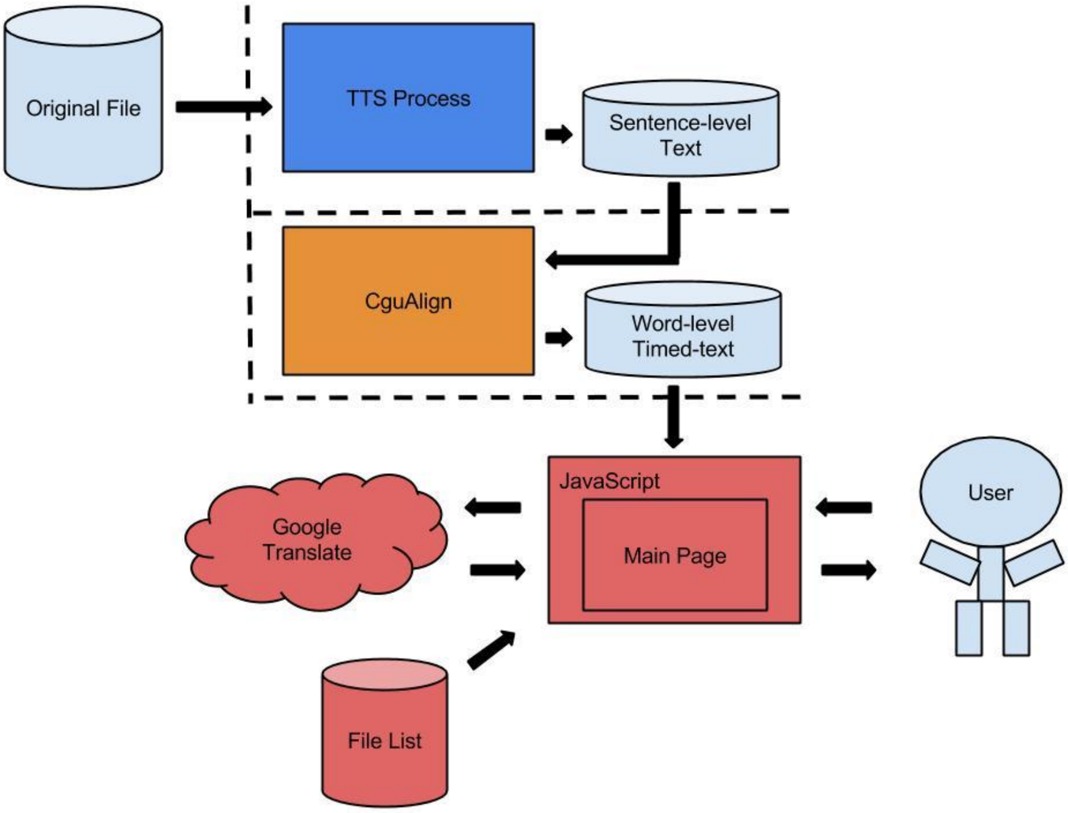
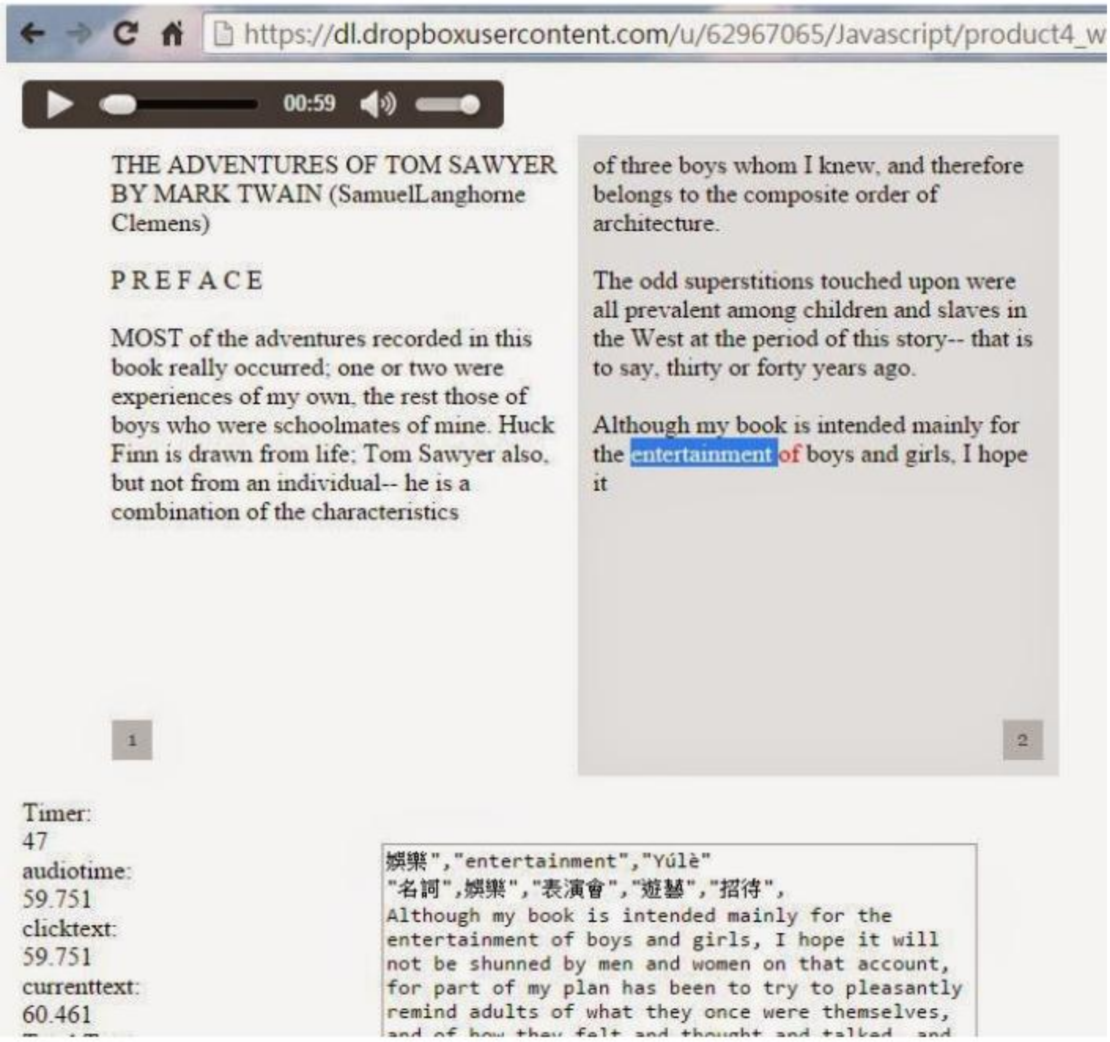


Figure 18. Website Presentation Flow Diagram

Here we use simple JavaScript to build a website to browse the word-level timed-text file.


https://dl.dropboxusercontent.com/u/62967065/Javascript/product4_w
00:59

THE ADVENTURES OF TOM SAWYER BY MARK TWAIN (SamuelLanghorne Clemens)

PREFACE

MOST of the adventures recorded in this book really occurred; one or two were experiences of my own, the rest those of boys who were schoolmates of mine. Huck Finn is drawn from life; Tom Sawyer also, but not from an individual-- he is a combination of the characteristics of three boys whom I knew, and therefore belongs to the composite order of architecture.

The odd superstitions touched upon were all prevalent among children and slaves in the West at the period of this story-- that is to say, thirty or forty years ago.

Although my book is intended mainly for the entertainment of boys and girls, I hope it

Timer:
47
audiotime:
59.751
clicktext:
59.751
currenttext:
60.461

```
娛樂","entertainment","Yúlè"
"名詞",娛樂","表演會","遊藝","招待",
Although my book is intended mainly for the
entertainment of boys and girls, I hope it will
not be shunned by men and women on that account,
for part of my plan has been to try to pleasantly
remind adults of what they once were themselves,
and of how they felt and thought and talked, and
```

Here we can take easy shadowing technique to language learning. In bottom of the page, show the meaning of the word you choose.

Conclusion

The Demo Video link on YouTube

https://www.youtube.com/my_videos_annotate?v=xgpgX1X24Ro