Construction of an Online Application Platform for Chinese Singing Voice Synthesis

中文歌聲合成之線上應用平台架設

組別：A212　指導教授：劉奕汶　組員姓名：王曉涵、王威智

Abstract

In our experiment, the goal is to promote the lab former research in the field of Chinese singing voice synthesis [1] to the open web. After familiarizing with the operating rules and limitations of the model of this research, the overall operation will be simplified and automated through the online platform. As long as the user upload the document conforming to the rules, the back-end can do related pre-processing in the pre-built environment, and use the Chinese singing voice synthesis model to generate the sound files corresponding to the documents written by the user, without having to set a complex operating environment and various input and output path. Therefore, ordinary users who are not familiar with related technical content can easily use this technology.

We mainly use Python-Flask, HTML, CSS, JavaScript and other languages, and use the Bootstrap package as the main basis for the layout of the webpage. This package has advantages in adapting to different window sizes, so mobile devices can also use this webpage smoothly.

Because the web page production of this group is a process of continuous refinement towards the goal of "allowing users to operate intuitively", we first achieve the most important input, synthesis, and output functions, and then optimize various functions that assist users. We have found 6-7 trial users to test, we have successfully allowed most users to synthesize their own audio files through this website, and understand the advantages and disadvantages of this Chinese singing voice synthesis model in the process.

We also hope to use this as the beginning to the online demonstration of laboratory's research. Other research can also be promoted to the online platform through the webpage of this special implementation result, and the functions will continue to be optimized.

摘要

使用實驗室學姊在中文歌聲合成領域的研究[1]作為欲推廣之技術，在熟悉該研究之模型的操作規則及限制後，將其整理簡化並透過網路平台連接，只要上傳符合規則的文檔，後端即可在已經建置好的環境中做相關的前處理，並且使用中文歌聲合成模型生成與使用者自己編寫的文檔對應之音檔，不必設定複雜的運作環境以及各式輸入輸出的路徑，讓不熟悉相關技術內容的一般使用者也能輕鬆操作。

我們主要使用 Python-Flask、HTML、CSS、JavaScript 等語言，並且使用 Bootstrap 套件作為網頁 layout 主要的根據，此套件在不同視窗大小的適應上具有優勢，因此行動裝置也可順暢的使用此網頁。

因本組實作專題之網頁製作是一個不斷往「讓使用者能直覺操作」的目標不斷細化的過程，首先達成最重要的輸入、合成、輸出功能，接著優化各項協助使用者的功能，我們尋找了 6-7 位的試用者進行測試，我們已成功讓大部分的使用者能夠透過此網站合成自己專屬的音檔，並且在過程中了解此中文歌聲合成模型的優缺點。

我們也希望能以此做為線上 demo 實驗室研究成果的開端，後續其他的研究成果也可藉由這次專題實作成果之網頁上架到網路平台，並且繼續優化功能。

內容

一、 前言

在歌聲合成實驗室中有非常多學長姊留下的優秀研究，但這些研究技術沒有一個合適的管道可以直接的讓不熟悉相關技術的人操作，和教授討論之後，我們決定設計一個線上的網路平台，引用學長姊留下的研究成果，讓這些成果能更簡單的推廣給更多人了解、使用。在這個實作專題中，我們編寫的網路平台是基於實驗室過往在中文歌聲合成方面的研究成果，這項技術經過多位學長姊的努力，至今已可以透過輸入文字與數字化的旋律，產生出聽感自然的中文歌聲。

編寫一個機器學習模型的網路平台，我們事先須熟悉此模型的使用方式。經過多次的測試，我們慢慢的了解這個模型的運作原理，知道使用者在甚麼樣的使用條件才能使模型正常運作，也會發現這個模型有一些需要解決的缺陷。因此我們便會開始評估要從模型本身做修正、另外的編寫可以補償此缺陷的程式，甚至如果我們最後無法修正，是否應改以給予使用者規範的方式實現。

接著我們開始進行網站的架設，網站的後端是負責定義使用者操作時所能使用的功能，在此我們是使用 python 的 Flask 網頁框架，基於此框架，我們可以有效率的使用 python 直接定義網頁前端和後端模型的溝通，之後再利用 HTML、CSS、JavaScript 語法搭配 Bootstrap 網頁框架製作響應式網頁，希望經過使用者介面的視覺優化與研究使用者體驗，讓大眾可以在各自的電腦或行動裝置上，體驗中文歌聲的合成技術。

二、 主架構

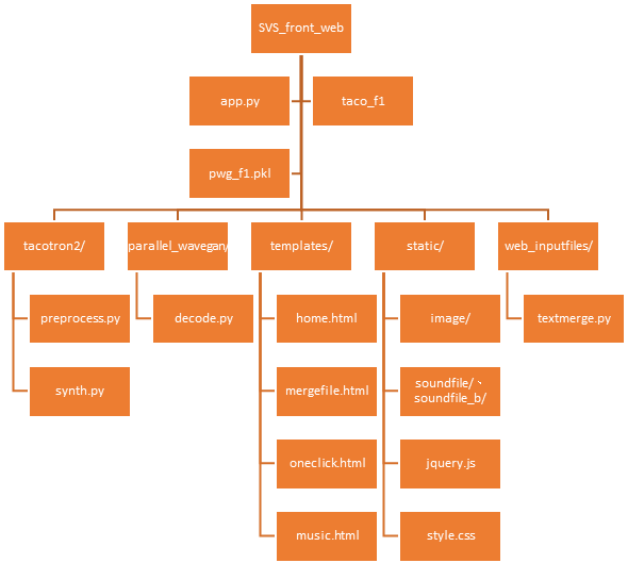

圖 一、網站運行架構圖

架設此網站的架構主要是利用 python 的輕量級網頁框架「Flask」，利用此框架，將既有的歌聲合成之機器學習模型（Tacotron2、Parallel WaveGAN），以模組化的方式，引入以 Flask 框架所寫成的 app.py 進行整合，作為後端 API，並在前端以 HTML, JavaScript, CSS 等語言撰寫。透過調校與設計可讀性、實用性較高的使用者介面，使原本須使用終端機操作的程式，變為可讓大眾使用的響應式網站。以下將依照此架構圖詳細介紹本平台之運作原理：

I. app.py、templates 資料夾、static 資料夾

app.py 為此線上歌聲合成平台的主程式，主掌整體網頁的運作，其總管了使用者可以在這個線上平台上執行的所有功能。在此程式當中，我們規範每個網址所對應的頁面，各個網址底下則以 python 函式與 html 頁面端進行對應。

templates、static 兩資料夾代表的是頁面端的資料，依照 Flask 套件的官方文件規範，app 要讀取的資料來源，必須為這兩個資料夾。所有撰寫好的網頁都會被放在 templates 資料夾，在這之中包含使用者正常使用時的頁面，以及網頁發生錯誤的頁面。其餘要給 app 讀取的內容都放置於 static 資料夾，包括：JavaScript 函式庫(jquery.js)、CSS 程式(style.css)、網頁中的圖片(images)以及生成供使用者試聽的音檔(暫存區：soundfile 資料夾、永久存放區：soundfiles_b 資料夾)。

II. web_inputfiles 資料夾

在此資料夾中放置的是使用者合成歌聲所需輸入的文字文件，在這之中為使用者分別輸入之歌詞、音高與音長之文字檔。使用者可透過網頁點擊呼叫 textmerge.py 之函式，此函式會將使用者的個別文字檔轉換成可以給歌聲合成模型讀取之格式。

III. tacotron2 資料夾

在 Tacotron2 資料夾之下，為文字轉頻譜之模型 taco_f1。事先編寫好的歌詞與旋律（文字文件），會先進入 preprocess.py 中根據進行中文到拼音與節奏至音長的轉換，轉換過後的文字檔會由 synth.py 將文字轉換成對應之 Mel-spectrogram 頻譜。

IV. parallel_wavegan 資料夾

在 parallel_wavegan 資料夾中，為神經網路聲碼器，藉由過往學長姐所訓練而成的模型 pwg_f1.pkl，可將在 tacotron2 所生成之頻譜，解碼成歌聲。一開始使用者輸入的文字經過上述過程至此最後一步，便會變成歌聲。

上述步驟雖然複雜，但使用者不需要理解 I.~IV.之相關原理，只會需要上傳所需檔案並點擊一個按鈕就可以自動化的執行所有程序。

三、 引用技術

我們製作的網站背後所應用的技術為由碩士班學長姐所研究開發完成之歌聲合成模型[1]。進行完整的中文歌聲合成流程，須結合以下兩項技術：

I. Tacotron 2

Tacotron 2為一個長短期記憶模型（long short-term memory，LSTM）架構的的遞歸神經網路（recurrent neural networks，RNN），屬於文字轉語音（Text-to-Speech）的相關應用。考量到歌唱比起說話，若要聽起來自然會有更嚴格的音韻學觀念，經過先前的實驗室學長姐的編寫，將此 Tacotron 2轉換成更適合用於歌聲合成（Singing-Voice-Synthesis）的架構，Tacotron2能夠利用訓練好的 taco_f1模型，將包含歌詞與旋律的文字文件，轉換成 Mel-spectrogram。

II. Parallel WaveGAN

Parallel WaveGAN 為神經網路聲碼器（Neural Vocoder），用途為將 Mel-spectrogram 解碼成時域的波型，也就是人耳所能夠聽到的聲音檔。藉由實驗室畢業學長過往蒐集之 Mpop600 Dataset 中的女聲進行訓練而成的 pwg_f1.pkl 模型，Tacotron 2所產生之 Mel-spectrogram 經過此聲碼器即可讓轉換成相當擬真且自然的歌聲。

四、 使用套件選擇

在確定實作專題題目之後，我們便開始進行相關學習資源的蒐集，由於需要用到的程式語言、簡單的計算機網路概論、網頁設計方法等幾乎都需要從頭自學，大概花了一個月才將所需之知識補齊，並且決定出實作歌聲合成平台的架設方向。透過學習的過程，我們認知到要完全從頭實作出可以正常使用、外觀精美，並且親自架設一個可接收外部連結的伺服器，將稍嫌困難，因此我們決定選擇能達到相同目的且較易於學習的框架，來實作我們的平台。以下將分成後端、前端來介紹：

I. Flask（v2.0.2，後端框架）

一般來說，網站架設的後端架構並非使用 python，application server（含 API）本身可以直接與 Web Server 進行響應。但若後端要架設一個以 python 為正常運行的網站，則需透過 WSGI Server（Web Server Gateway Interface Server），這是由於一般 Web Server 無法判讀 python 語法，WSGI Server 定義了 HTTP Request 和 python 相互溝通的規範，使開發人員可以專注於應用程式的開發。

本實驗室的機器學習演算法皆以 python 程式語言撰寫，網路上也有將機器模型使用 Flask 開發應用程式發布於網路的先例，故我們選擇使用 python 之 Flask 套件架設後端 API。使用 Flask 框架好處是我們可以直接以同樣的語法執行固有模型並且進行網頁開發，尤其它還內建豐富的函式，可與前端 html 網頁端進行良好的整合，這個套件同時具備簡易的（同一時間只能接收單一指令）WSGI Server，因此在網站的測試階段，搭配開發時所用的電腦（充當 Web Server），只要在 Flask 應用程式運行當中，就可以完成簡易的網站供使用者遠端測試，接收並處理個別使用者的需求。由此可見這非常符合我們的開發需求。

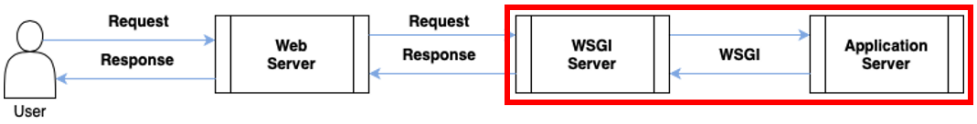

圖 二、Flask 網頁框架可負責 Application Server 和 WSGI Server 之任務[1]

II. Bootstrap（v5.1.2，前端網頁框架）

Bootstrap 為知名網頁框架，利用簡單的引用就可以使網頁有好看的外觀，這個網頁框架的便利性也受到市場上許多網頁開發人員的喜愛，經常被使用於製作響應式網頁。舉例來說，若我們需要製作一個按鈕，可以到其官網瀏覽官方文件[2]並複製引用至欲設計的 HTML 網頁中，如此一來就可以很快速地做好一個按鈕。

Bootstrap 可以大大的省去學習 CSS 與 JavaScript 所需的時間，因此我們可以更專注於開發應用程式之功能。

[1] https://minglunwu.github.io/notes/2021/flask_plus_wsgi.html#%E5%89%8D%E8%A8%80
[2] https://getbootstrap.com/docs/5.0/components/buttons/

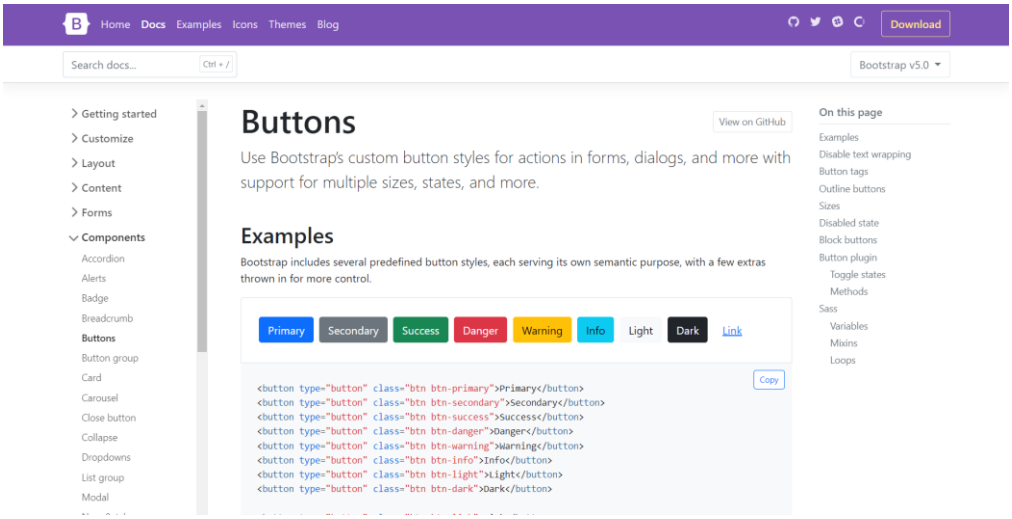

圖 三、多種網頁元件皆可直接於 Bootstrap 官網選用（以按鈕為例）

III. Gunicorn WSGI

如同前面所述，Flask 框架所提供的 WSGI Server 並不適合用於一般用途，因為該伺服器只具有執行單一指令的能力，實務上，一個網頁同一時間會收到來自四面八方的 HTTP Request，因此比須尋找支援多工處理的 WSGI Server。最多人使用的 WSGI Server 分別為 Gunicorn 和 uWSGI，Gunicorn 之操作較為簡易，可以在 Linux 系統快速的建立，可設定多工處理，但礙於在其他方面花了大部分的研究時間，對於此 WSGI Server 的架構尚未有深入的認識，但可以確定在未來這會是網頁正式上線時重要的一環。

五、 結論

為測試我們建置的網頁是否達到將中文歌聲合成模型更簡單的推廣給更多人了解、使用的目的，我找了 6 位測試者試用網頁，並統整出以下幾點回饋意見:

- 使用檔案上傳的部分容易搞錯，不知道有沒有上傳。
- 較長句子的合成會出現趕拍或吃音。
- 不清楚當前合成結果單句的檔案是什麼意思。
- 使用說明的下拉選單如果第一個是默認展開容易讓使用者忽略後面音高和音長的說明。
- 有些字合成不出來，換掉就可以了
- 檔案上傳的地方操作不直覺，容易搞混
- 樂理背景知識較少的人計算音長、音高要花較多時間。
- 在使用過程中對此技術感到新奇也延伸詢問了許多相關知識。
- 檔名若包含特殊字元會造成後續合成錯誤。
- 錯誤頁面鍵一跳回合成畫面而非首頁。

對於使用者給予的回饋，大致可分為中文歌聲合成模型本身以及網頁使用介面，使用者在操作過程中會發現模型本身的限制（部分字句不在字庫、趕拍、吃音等）以及體驗到模型實際合成出擬真且自然的歌聲，雖然合成限制會讓網頁回

報錯誤給使用者，但這也符合我們希望使用者能夠了解此模型的初衷，故對此我們不做過多調整。

而網頁使用介面我們發現了一個統一且能夠改善問題，即檔案上傳的網頁介面操作不直覺，容易讓使用者混淆，因時間限制我們將其視為未來改善的目標，希望能在專題競賽後將其解決。

即使在做這個專題的一開始，我們甚麼都看不懂，但經過這段時間尋找、規劃並自主學習所有相關知識，以及向實驗室其他人請教，不只學習到如何架設網站，也見習了許多在機器學習上的課題，感謝一年來在這個實驗室所帶給我們的成長。

心得感想

在此次專題實作前兩位組員都對網頁建置沒有任何基礎，從奠定目標到專題競賽這段期間有了非常大的收穫，一開始連前端後端如何實現都不曉得，在閱讀大量資料後才從中找到我們需要的內容，好不容易選定使用的套件後又要面對與程式語言不熟悉的困境，在不斷的學習、嘗試、失敗的過程中，慢慢將經驗累積成現在的成果，不但對前端後端彼此交流的方式有基礎概念，也更加熟悉 Python、HTML、CSS、JavaScript 等語言，但隨著相關知識的增加，也讓我們更加意識到網頁建置這們學問的水非常深，還有更多的內容還未接觸熟悉，此網頁在功能上尚有許多可以改進的空間。