國立彰化師範大學資訊工程學系專題期末報告

施明毅/陳宥廷/廖祐興/徐浩瑋/蔡秉達/林映彤

國立彰化師範大學資訊工程學系

500 彰化市進德路一號

Email: myshih@cc.ncue.edu.tw

一、摘要

近年來出現的語音助理為人們帶來更智慧的生活，但是大部分的傳統網頁仍需要手動操作，不管是滑鼠控制頁面或是鍵盤輸入，來進行註冊、登入、搜索、瀏覽等功能。所以我們為了讓傳統的網站也能夠使用語音操作，我們以傳統的租房網站為範本，運用語音辨識 API 將使用者說話的內容轉換為文字，接著利用結巴斷詞對語音辨識的結果進行處理，讓使用者只要運用聲音的獨特性和話語即可操作網頁功能。此外，配合趨勢，區別於過去常見的內容推薦方式，我們架構出協同過濾推薦系統，推薦的每一個項目是基於是否有其他相似的用戶也同樣喜歡這個項目，讓不知道要選擇看哪些房子的使用者，也能經由合適的推薦快速的找到符合需求的房子。

Abstract

For the past few years, the voice assistants have made our life been more and more convenient. However, users interact with most conventional websites using a mouse and a keyboard to sign up ,log in, searching and browsing. To make a difference, that is, making an ordinary website capable of being browsed through via voice. We made one general house-renting website as a paradigm, then we used Google Web Speech API to record the request said by the user, and convert it from speech into text. After that, we took advantage of Jieba, a Chinese word segmentation module, to cut the speech and extract the keyword for further search. In this way, the user can browse our website with not only keyboard and mouse, but his/her tongue and voice, to operate this websites. Additionally, Different from the past, collaborative filtering recommend items based on how similar users liked the item. Because of recommender systems, user can find the ideal house for rental.

二、相關技術介紹

2.1 Jieba ：

自然語言處理的其中一個重要環節就是中文斷詞的處理，比起英文斷詞，中文斷詞在先天上就比較難處理，然而使用 Jieba 結巴就可以輕易達成中文斷詞的目的，Jieba 結巴是 Python Based 的開源中文斷詞程式，所使用的演算法是基於 Trie Tree 結構去生成句子中中文字所有可能成詞的情況，然後使用動態規劃（Dynamic programming）算法來找出最大機率的路徑，這個路徑就是基於詞頻的最大斷詞結果。對於辨識新詞（字典詞庫中不存在的詞）則使用了 HMM 模型（Hidden Markov Model）及 Viterbi 算法來辨識出來。基本上這樣就可以完成具有斷詞功能的程式了。

2.2 Web Speech API ：

Web Speech API 是支援使用者可以在所屬網站上利用語音去達成其各式功能的系統，而我們所使用的 Web Speech API 是一套由 Google Chrome 支援的語音套件，由於是 Chrome 內建的功能，所以不需要額外載入其他資源就可以運行，算是相當的方便使用，但也只能在 Chrome 瀏覽器中執行。此語音套件可以利用麥克風擷取環境音訊，並將擷取之語音轉換成中文字串。當套件接收到語音訊息時，系統會先將其即時轉換成中文字串並存入「中間結果」變數中，停止語音輸入後，系統會依據輸入之語音訊息的上下文意搭配中間結果做比對，最終轉換成為「最終結果」，而最終結果將會是我們網頁執行的判斷關鍵。

2.3 Xampp：

Xampp 是已經有了十多年的歷史、且相當知名、相當多人愛用的快速架站工具，因為他完全免費且易於安裝，而且其中包含了 MariaDB、PHP 與 Perl。Xampp 的 X 就是代表跨平台、支援多種作業系統，A 代表 Apache 網頁伺服器，M 代表 MariaDB，兩個 P 分別代表 PHP 和 Perl，也因為它的便利性，一次性的把架站所需之程式結合在一起，以及可以快速的啟動伺服器與資料庫程式，再加上整潔乾淨的操作環境，沒有過多複雜且艱深的功能，使我們最終決定選用 Xampp 當我們主要的架站工具。

2.4 MySQL：

MySQL 原本是一個開放原始碼的關聯式資料庫管理系統，原開發者為瑞典的 MySQL AB 公司，因其開發公司陸續被昇陽微系統、甲骨文公司併購，MySQL 成為甲骨文公司旗下產品。MySQL 在過去由於效能高、成本低、可靠性好，已經成為最流行的開源資料庫，因此被廣泛地應

用在 Internet 上的中小型網站中，所以我們選擇最方便亦最為常見的 MySQL 作為我們的資料庫系統。

2.5 Numpy：

NumPy 是 Python 在進行科學運算時，一個非常基礎的 Package，同時也是非常核心的 library，提供非常高效能的多維陣列數學函式庫，可定義任意的數據型態，使得能輕易及無縫的與多種資料庫整合。

2.6 Json：

將結構化資料呈現為 JavaScript 物件的標準格式，常用於網站上的資料呈現、傳輸，雖然 JSON 是以 JavaScript 語法為基礎，但可獨立使用，且許多程式設計環境亦可讀取並產生 JSON。

三、系統架構

3.1 系統架構：

本系統主要包含三大部分，如圖 3.3.1 所示(圖片較長故放置於章節末)，大致說明如下：

- 語音系統：

包含使用者的個人音檔訓練，以及使用語音操作網頁時的字詞處理。

- 推薦系統：

包含登入首頁部分的協同過濾推薦，以及未登入狀態的首頁、房屋資訊內頁下方的熱門推薦。

- 網頁操作：

含註冊、登入、搜索等功能。

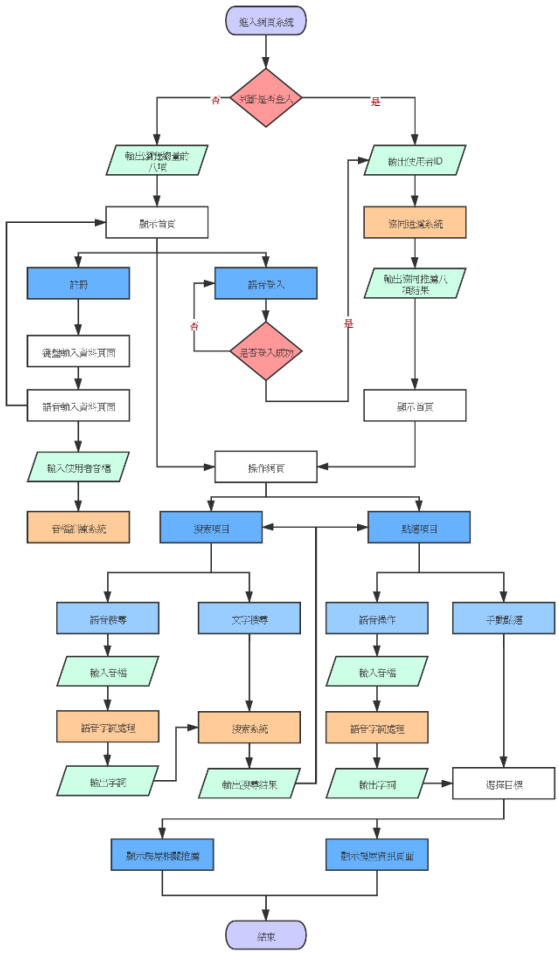

圖 3.1.1 網頁架構流程圖

3.2 語音系統架構：

語音辨識技術，其主要目的是將人類所發出的聲音中的類比訊號，透過數位轉換器，轉換成數位訊號，並經由電腦做解析、運算，然後建立語音模型以供辨識。欲產生語音模型，需經過三個階段的處理方可產生，分別為語音訊號的前處

理(Speech Signal Processing)、語音訊號的特徵萃取(Feature Extraction)和語者辨識模型(Speaker Recognition Model)。

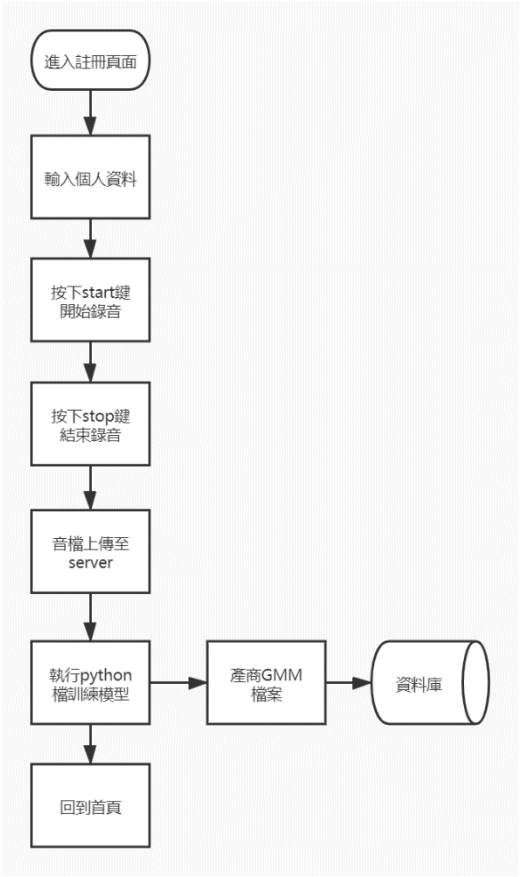

圖 3.2.1 語音註冊流程圖

3.2.1 語音訊號的前處理：

語音訊號的前處理又可以依序分為四個步驟：音框取樣、預強調、端點偵測、視窗化。

- 音框取樣：

當人類發聲使聲帶產生振動時，聲帶端之結構可以看成如同一串脈衝訊號通過一個由聲帶 所形成的濾波器(Glottal Shaping Filter) ，其中產生的空氣流量速度波形即有-12dB/oct 的高頻衰減。而嘴唇部分對於氣流產生的阻擋效果則可視為一個輻射阻抗，若是當聲音訊號通過時則會產生一個高通濾波的效果，有著+6dB/oct 的高頻增強。若發聲腔道模型為 0dB/oct 的高頻衰減，則最後所產生的訊號有-6dB/oct 高頻衰減。因此若是要補償此衰減，則必須將語音信號通過一階高頻濾波器來做處理。

- 端點偵測：

在龐大的音訊處理量中，本身包含著許多不帶音訊或是次要音訊的內容在其中，這將會大大拖慢計算機的處理時間，因此為了更進一步精簡音訊處理量，而有了端點偵測這項技術的產生，通常是根據能量曲線來設定一個門檻值，把不必要的聲音做初步的篩選。

- 視窗化：

通常在求取音框內數值的同時，音框兩邊界所產生的不連續現象，在聽覺感受上會有著明顯的落差，若觀察在於頻域上的語音頻譜也會發現其連續性受到破壞。因此必須使用可以將邊界誤差降低的方法。透過視窗化即可大幅減少音框之間邊界的大幅誤差。

3.2.2 語音訊號的特徵萃取：

完成語音前置處理之後的資料量，雖說以省去不少次要資訊，但整體來說，剩餘的資料仍是

非常龐大，若是直接交由計算機來做交互比對，那將會非常沒效率，因此還必須更進一步的將語音資料，尋其特性來取出當中具有代表性的特徵值來代表語音訊號，而此一過程即稱為特徵萃取。常見的有線性預測倒頻譜參數(Linear Prediction Cepstrum Coefficient, LPCC)、梅爾頻率倒頻譜參數(Mel Frequency Cepstrum Coefficient, MFCC)等。

3.2.3 語音辨識的模型

做完了前面的一連串處理之後，便可開始建立每一筆聲音獨一無二的語音模型，語音辨識的部分分為兩種：語者辨識(聲音的主人是誰) 與詞彙辨識(講了什麼內容)，兩種的辨別分成兩種不同的模型來當作依據判斷，最常使用高斯混和模型來辨別語者以及隱藏式馬可夫模型來辨別詞彙。

- 高斯混和模型(Gaussian Mixture Model, GMM)：

高斯混合模型作為高斯機率密度函數的一個線性組合，是語音訊號處理上常用的統計模型，其基本理論之前提是當只要有足夠多數目的混合分量，則就可以逼近任意一種密度函數，而語音特徵通常有著平滑的機率密度函數，因此在於有限數目的高斯密度函數 就足以對語音特徵的密度函數形成平滑逼近。會使用高斯混合模型來代表語者模型主要有的原因為，高斯混合模型的每一個基本密度皆可以模擬出一些發聲狀態的特徵。因此我們使用高斯混合模型中第 i 個平均值來代表第 i 個聲音特徵的頻譜形狀，並且使用共變異矩陣來代表頻譜形狀的變化。

- 隱藏式馬可夫模型(Hidden 8 Markov Model, HMM)：

隱藏式馬可夫模型屬於一種雙重的隨機程序(Double Stochastic Process) ，是以一種無法觀察(Hidden)且為有限可能值(Finite Number)的隨機程序做為基礎，在透過另一個隨機程序，可從中觀察到隱藏式馬可夫模型所產生的一連串觀測值，去判別最有可能的數值。

3.3 推薦系統架構：

推薦系統的部分，如圖 3.3.1 所示(圖片較長故放置於下一頁)，根據使用者登入情況的不同，在不同頁面中採取不同的推薦形式。

推薦系統已成為相當普遍的資料科學應用，最常被使用的就是協同過濾(Collaborative Filtering)以及以內容為基礎的推薦方法(Content-Based Recommendations)，這裡我們選擇使用的是協同過濾。對於每一個用戶，推薦系統所推薦的每一個項目是基於是否有其他相似的用戶也同樣喜歡這個項目。

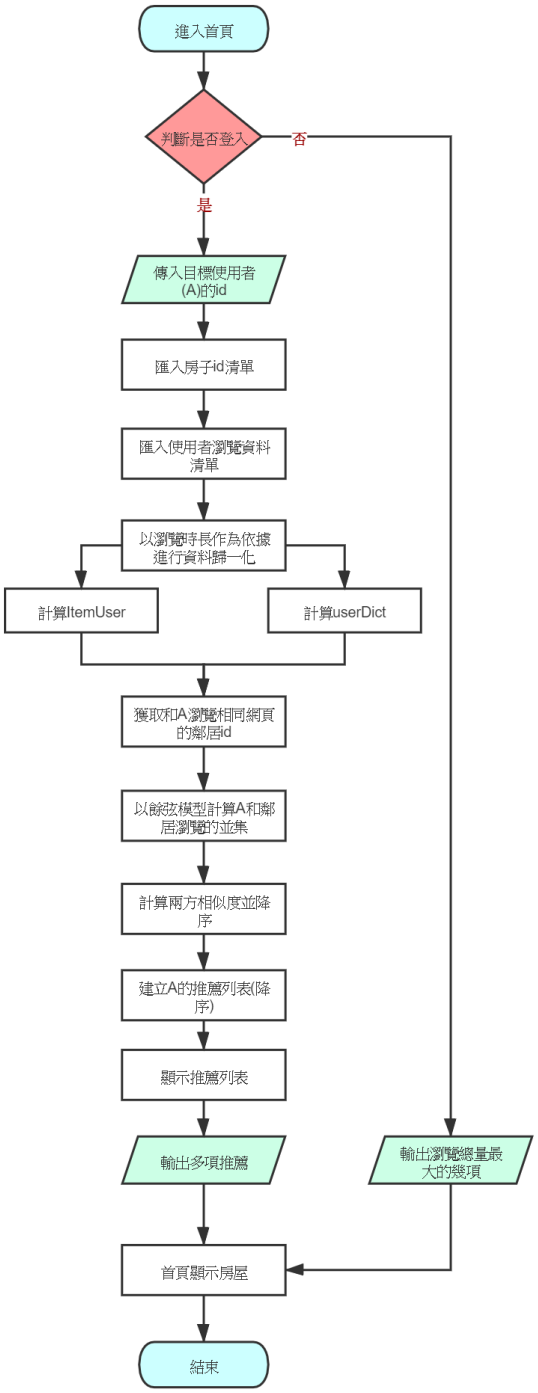

圖 3.3.1 推薦系統流程圖

3.3.1 衡量相似性：

將當前使用者 id 傳入由 python 製作的推薦系統後，當我們用於計算相似度時，我們將會計算歐式距離(Euclidean distance)，意即兩個物件之間的距離越高，則代表越分開，相似度越小。反過來說，當兩個物件之間的相似度越高，則距離則越接近。通常相似度的指標會用 0、1 來代表，0 表示不相似(完全不一樣)，1 表示相似(完全一樣)。

3.3.2 餘弦相似性：

餘弦相似性(Cosine Similarity)為最普遍被使用的相似度測量法，透過餘弦的相似性，我們將能評估在兩個向量角度之間的關係及相似性，角度越小則兩個之間的相似度則越高。藉由餘弦相似性，我們可以知道當兩個向量角為 0°時，其為最大相似度(兩個向量指向同一個方向)；當兩向量間的角度為 90°時，則相似度為零(兩向量相互正交)；而當兩向量角為 180°，則它們的相似度為-1(意指它們兩個向量指向相反的方向)。如果我們限制向量為非負值，然後在兩向量之中的角度拆開在 0°到 90°之間，相對應的餘弦相似度即會分別落在 0 與 1 之間。因此，對於正值的向量來說，在一個"理想"的相似度測量狀況下，餘弦相似度就只會在 0 與 1 兩個值之間。

3.4 資料庫架構：

本系統之資料庫可大致區分為五個部份，如圖 3.4.1 所示，

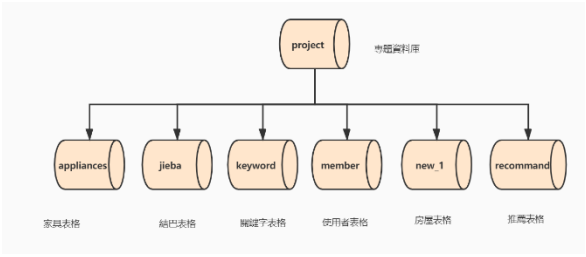

圖 3.4.1 資料庫架構圖

以下為各部份說明：

- appliances：

紀錄每筆房屋中，房東提供的屋內家具項目。

- jieba：

存放經結巴斷詞處理後取出之關鍵字，用於搜尋與關鍵字相符條件的房屋結果，並在搜尋出結果後刪除，以利再次搜尋。

- keyword：

存放與關鍵字比對的資料，以及每項條件的分類，以便多重條件查詢。

- member：

存放已註冊的使用者相關資料。

- new_1：

紀錄於租屋網站中，每筆出租房屋之個別詳細資料。

- recommand：

紀錄所有使用者瀏覽網站相關資料，提供給協同推薦系統使用。

四、系統實作畫面

圖 4.1 首頁畫面

圖 4.2 註冊-語音註冊畫面

圖 4.3 語音登入畫面

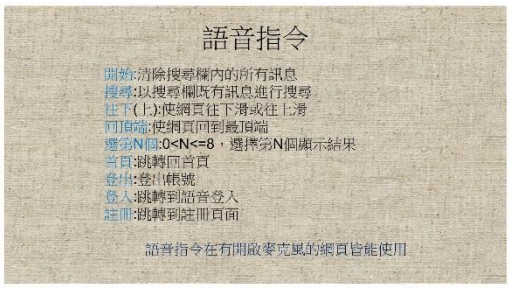

圖 4.4 語音操作相關指令

圖 4.5 房屋總覽畫面

圖 4.6 房屋資訊內頁畫面

圖 4.7 房屋資訊內頁下方推薦畫面

五、結論

本專題以語音與網頁結合，透過語音指令取代部分輸入，如鍵盤與滑鼠，以達到越發先進、講求便利的世代之需求，並透過語者辨識嘗試取代帳號密碼輸入的動作，對於帳戶的安全性相對較高，在達到便利性之際，亦不流失其可靠性，同時依據用戶的身分(遊客/會員)、瀏覽的項目來推薦相關項目。

在語音指令部分，我們使用 google 之套件以供辨識內容，再將套件辨識結果交由後台程式處理，在二階段的處理中，處理速度可說是無法避免的挑戰。

在語者辨識部分，透過用戶在註冊時所提供的音檔，交由後台程式訓練語音模型以供之後使用，需要足夠的訊息量以支持辨識率，如何在訊息量和用戶便利性之間平衡亦是一道難題。

在推薦部分，使用的是協同過濾推薦系統，此系統透過以使用者為基礎，抑或是以項目為基礎來決定推薦項目，相關性的計算與推薦的可靠性，以及進階使用 Hybrid method 解決 CF 的 cold-start problem 等同樣是值得鑽研之處。

本次專題專注於上述三點進行探討與實踐，以提供使用者良好的使用體驗，但仍然有能完善及改進之處，希冀往後能更進一步。

六、參考文獻

[1] 協同過濾(Collaborative Filtering)模型 https://blog.dominodatalab.com/recommender-systems-collaborative-filtering/

[2] 吳肇銘、金志聿、林怡秀「協同過濾技術在商品推薦系統上之應用與成效評估」，中原大學資訊管理學系研究所

[3] 林熙禎、許益誠「以網頁探勘技術為基礎之電子目錄上推薦系統之研究」，中央大學資訊管理學系研究所

[4] 協同系統與推薦系統
https://www.getit01.com/p20171225819971859/

[5] python 實現協同過濾
https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/362063/

[6] 推薦系統之協同過濾演算法
https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/568697/

[7] Speaker-identification-using-GMMs
https://github.com/abhijeet3922/Speaker-identification-using-GMMs

[8] Google 的語音辨識 API 之使用 (作者：陳鍾誠)
http://programmermagazine.github.io/201310/htm/article2.html?fbclid=IwAR2sTF1qz9pobklQhlb-kZVAajVznmu2yGYMtm10bNrJmuNxep7X3wpJqcU

[9] Capturing Audio & Video in HTML5 By Eric Bidelman
https://www.html5rocks.com/zh/tutorials/getusermedia/intro/

[10] HTML5 Speech Recognition API by Kai Wedekind
https://codeburst.io/html5-speech-recognition-api-670846a50e92

[11] Navigator.getUserMedia()-web API
https://developer.mozilla.org/en-US/docs/Web/API/Navigator/getUserMedia

[12] MediaRecoder() -web API
https://developer.mozilla.org/en-US/docs/Web/API/MediaRecorder

[13] Recording Audio from the User
https://developers.google.com/web/fundamentals/media/recording-audio

[14] Record to an Audio File using HTML5 and JS
https://air.ghost.io/recording-to-an-audio-file-using-html5-and-js/