國立台北大學 資訊工程學系 113學年專題報告海報

中文語音大師
Chinese Pronunciation Master

呂冠廷，呂詠儒，曹禕中

簡介

隨著全球化的發展，國際交流日益頻繁。在這個時代，不僅國人需要學習英語，許多外國人同樣開始學習中文。而在學習中文的過程中，口語練習是一個至關重要的環節。許多外國人在說中文時常出現奇怪的口音，而對於初學者來說，若缺乏教師的指導，往往難以察覺自己的發音錯誤，即便察覺了，也未必能明白錯誤所在或知道如何修正。現實中，大部分學習者都無法隨時獲得真人指導，且**市面上並無類似功能之軟體，即便是最類似的Google翻譯對於不熟悉中文的使用者辨識效果也不佳**。因此，我們計劃開發一套基於深度學習的中文發音矯正系統，**能辨識使用者所說的任意語句，而非僅限特定句子**。系統將判斷使用者的語意，並協助其糾正發音，促進更有效的中文學習。

本計畫預計開發的系統主要流程如下：首先，我們訓練了兩套自動語音辨識(ASR)模型，將語音輸入以字元和單字這兩種形式轉換為對應的拼音序列。接著，透過 n-gram 和 beam search 等演算法，初步修正辨識出的語音結果。隨後，我們使用 OpenAI 的 API，將拼音轉回中文。同時，在聲調辨識方面，我們利用 CTC Greedy 解碼將音訊切割為單字單位，再擷取每個音節的梅爾頻譜圖 (Mel-spectrogram) 作為特徵，並利用 CNN 模型判斷其對應的聲調，以分析使用者所發出的每個音節的聲調。最後，透過對比使用者的發音與正確句子的發音（聲調與念法），找出發音錯誤。系統將以直觀的方式顯示使用者的發音及聲調錯誤所在。**本系統不僅能處理任意語句，亦能結合語意理解與聲調分析提供多層回饋，在現有中文學習工具中展現出高度創新性。**我們相信，此系統可以更好的幫助外國學習者理解和掌握中文發音。

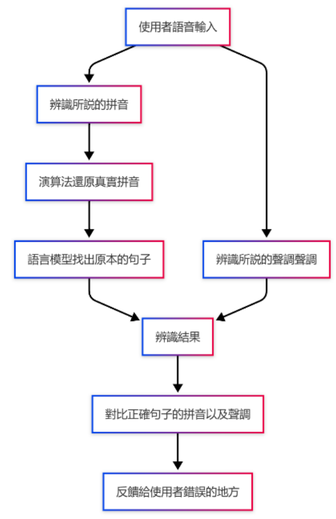


圖1:系統流程總覽圖

實作方法

演算法第一部分：使用者口語語音識別出拼音字串

我們訓練了兩個model,「字元層級 ASR 模型 (Char-based ASR)」及「詞彙層級 ASR 模型 (Word-based ASR)」，來進行拼音辨識。其中兩個model皆使用 Connectionist Temporal Classification (CTC) Loss進行訓練，以對齊長度不匹配的音訊序列與目標文字序列。

CTC架構圖

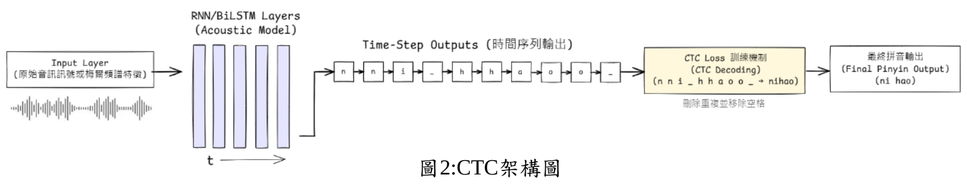


圖2:CTC架構圖

字母和單字層級的識別差異如下：

1. 字母層級：利用包括AIShell 中文語音，NexData模糊中文，FAD等資料集來達成把說出的中文以字母對字母的方式轉換成拼音。CTC預測的輸出空間為27，也就是26個英文字母和1個空格。
2. 單字層級：模型為Wav2Vec2-Large-XLSR-53 對單字拼音做微調訓練，CTC預測的輸出空間為自建的拼音對應表(word map)，共包含404種拼音類別。Word map計算出現頻率的公式：$f(w)=\sum_{i=1}^{M}\delta(w_i,w)$

在這兩種ASR模組，均會產出10種字串輸出，所以共輸出 20種候選字串。

演算法第二部分：拼音字串錯誤分析與修正

1.使用 Levenshtein Distance 修正字詞

ASR 模型輸出後，我們先以 Levenshtein Distance 修正無效拼音，將其替換為 word map 中最相近的合法拼音。該距離透過計算「插入（Insertion）」、「刪除（Deletion）」與「替換（Substitution）」所需的最少步驟數定義，如下：

$$D(i,j)=\begin{cases}\max(i,j) & \text{若 } \min(i,j)=0\\ \min\begin{cases}D(i-1,j)+1\\ D(i,j-1)+1\\ D(i-1,j-1)+1_{[w_i\neq v_j]}\end{cases}\end{cases}$$

2. Beam Search 解碼

隨後，我們利用下面兩個指標：

- Levenshtein Distance(由近到遠)
- 詞頻 (我們透過上百萬字的中文資料集所建立的word map決定)

去延伸出7個可能的候選單字字串。由此字串與其後單字字串每次共可延伸出最多7^3個可能的候選單字陣列，再利用n-gram 語言模型(設n=3)，為每個候選單字陣列評分，最後保留前 5 條分數最高之候選序列。篩選過程如圖3所示。其中Beam Search的評分方式：$S(y)=\sum_{t=1}^{T}\log P_{ASR}(y_t|x)+\lambda\cdot\log P_{LM}(y)$

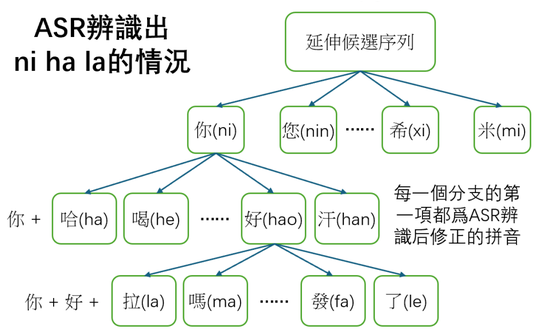


圖3:Beam Search + Ngram 示意圖

和用來評估候選序列的語言合理性之三元語言模型 (Trigram)定義如下：

$$P(w_1,w_2,...,w_T)=\prod_{t=1}^{T}P(w_t|w_{t-2},w_{t-1})$$

每個條件機率由以下公式計算：

$$P(w_t|w_{t-2},w_{t-1})=\frac{C(w_{t-2},w_{t-1},w_t)+1}{C(w_{t-2},w_{t-1})+|V|}$$

演算法第三部分：將篩選出的發音序列還原成中文字串

我們已經利用了語意分析出了最有可能的五個發音序列。在此步驟，我們會利用提示工程，把已經還原到正確或者非常接近正確的拼音輸出給OpenAI的API, 讓其返還正確的中文語句。該方案經過微調提示詞。

演算法第四部分：聲調

我們先通過單獨訓練好的 Wav2Vec2 + CTC 模型，取得每個時間點的 token 機率分佈，並以 CTC Greedy 解碼法得到每個單字的獨立音頻，如圖4。

其中，中文的聲調主要可以分爲五種聲調，各自有明顯頻率走向差異。我們從梅爾頻譜圖提取特徵來產生特徵向量x： $\mathbf{x}=[\bar{F}0, F0_{max}, F0_{min}, \text{slope}, \text{variance}]$

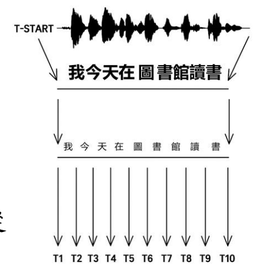


圖4:單字切割示意圖

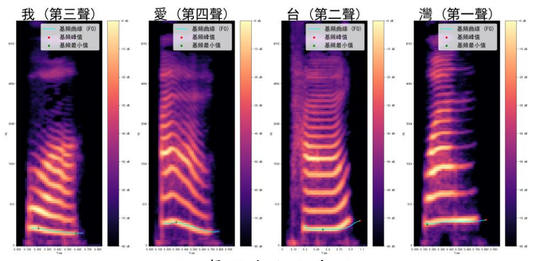
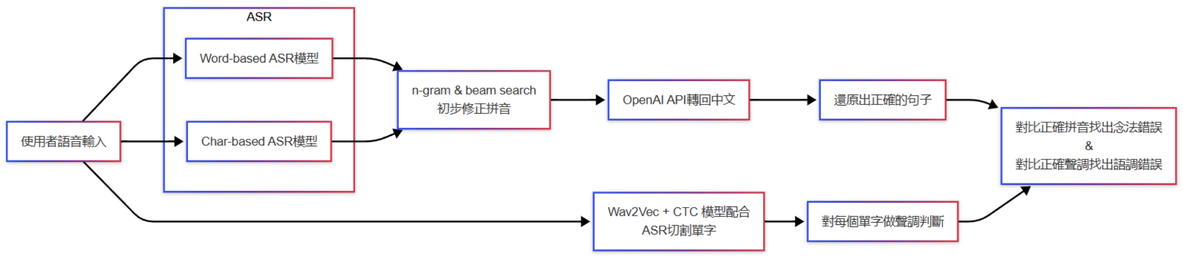


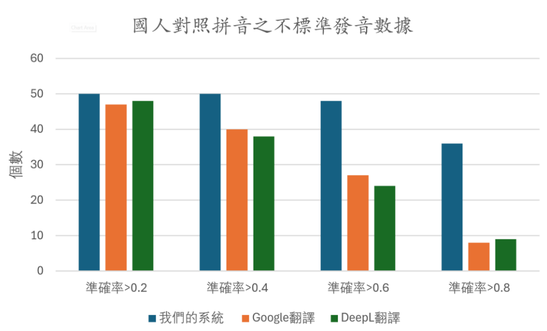
圖5:聲調走向示意圖

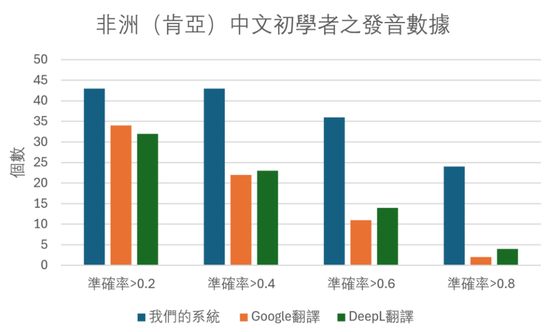
最後，我們利用 TonePerfect資料集(包含每種單字拼音）訓練了一個 CNN 音訊分類模型，辨識每個單字的聲調類型。輸入為單字音頻，模型輸出為四類分類結果，分別對應中文的一聲至四聲。其中，在標準發音中，各聲調在頻率走向上展現出顯著差異，其變化特性如圖5。

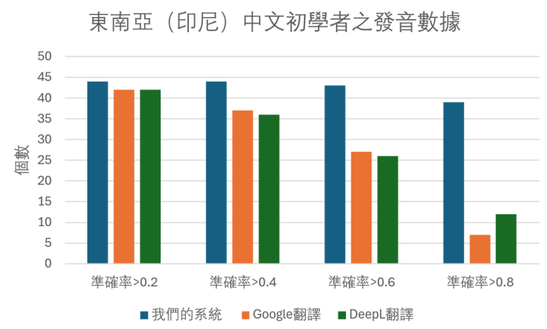
演算法整體流程圖

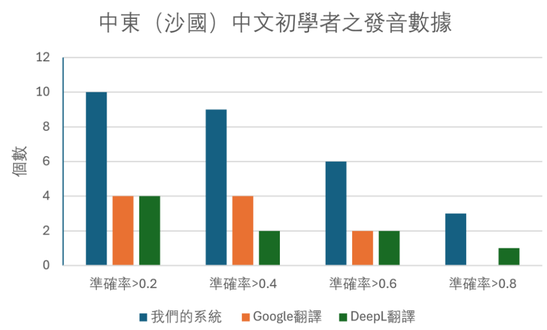


成果與評估









*分數為用edit distance計算識別語句對比正確語句

發音辨識準確率總體平均分數			
發音地區	我們的系統	Google翻譯	DeepL翻譯
國人發音	0.90	0.57	0.56
非洲發音	0.79	0.40	0.42
東南亞發音	0.91	0.61	0.61
中東發音	0.67	0.27	0.22

我們收集了來自世界各地中文初學者的語音資料，方式為提供拼音文字，並請受試者依據拼音朗讀多組中文句子。根據測試結果顯示，**本系統在準確率高於 0.8 的高標準區間中表現明顯領先，同時在整體平均準確率方面，也超越市面主流中文辨識系統（如 Google 翻譯與 DeepL）30% 以上**，展現出本系統在處理非標準中文發音時的穩定性與辨識精度。**右下角的二維碼可以下載我們所收集的測試集**，歡迎隨時來測試！

結論

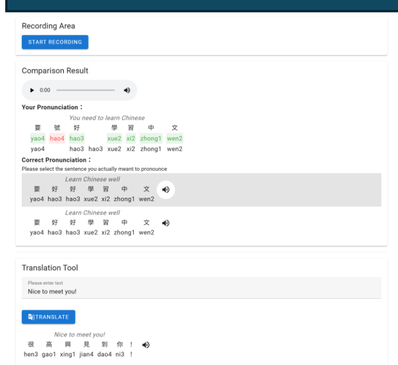


圖6:系統介面

本專題成功設計並實作了一套結合 ASR、聲調分類與語言模型的中文發音矯正系統。透過聲音切割與語調分析技術，**系統能有效辨識使用者任意語句的發音偏誤，並提供拼音與聲調兩層級的即時回饋，且**在拼音辨識與發音糾錯上，相較於市面翻譯工具更具發音識別準確度。此外，我們亦整合了多項輔助學習功能，例如翻譯查詢、歷史記錄、標準發音的 TTS 播放，多種語言界面來輔助學習者進步。圖6便展現了我們系統的主頁面。我們相信通過使用改軟體，可以讓學習中文發音變得更加簡單，也更加有趣。