國 立 交 通 大 學

電機學院　電信學程

碩士論文

結合聲調辨認之中文關鍵詞辨認系統

A Mandarin Keyword Spotting System Assisted with Tone Recognition

研 究 生：鐘進竹

指導教授：王逸如 博士

中華民國 100 年七月

結合聲調辨認之中文關鍵詞辨認系統

A Mandarin Keyword Spotting System Assisted with Tone Recognition

研 究 生：鐘進竹　　Student：Chin-Chu Chung

指導教授：王逸如　　Advisor：Yih-Ru Wang

國 立 交 通 大 學

電機學院　電信學程

碩 士 論 文

A Thesis
Submitted to College of Electrical and Computer Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
in
Communication Engineering
July 2011

Hsinchu, Taiwan, Republic of China

中華民國 100 年七月

結合聲調辨認之中文關鍵詞辨認系統

研 究 生：鐘進竹　　指導教授：王逸如 博士

國 立 交 通 大 學　電 機 學 院　電 信 學 程　碩 士 班

摘要

在現今的國語語音辨認系統中，大多使用國語 411 音作為辨認單元，使用聲學模型後，大部分同音不同聲調的情況都能辨認出正確結果。但是在關鍵詞辨認系統中，多數情況關鍵詞都是命名實體(Named entity)，如人名、地名、公司名，常常都是二字詞且容易有混淆音，所以加上聲調辨認就十分重要了。

本論文中，使用兩階段式的 keyword spotting 系統，在原來關鍵詞辨認系統中之語音參數抽取部份，使用 RAPT(A Robust Algorithm for Pitch Tracking) 演算法[1]，求取基頻軌跡，在系統辨認出 Top-10 keyword 後的 likelihood 分數，對關鍵詞加上第二級的 MLP 聲調辨認器[2]所辨認出來的分數，與 Top-10 加總後的分數，再進行重排，以得到更正確的辨認答案。

論文中，對一特定關鍵詞組：新竹科學園區 341 公司名（若包含別名則有 1074 個關鍵詞），製作關鍵詞辨認系統；在未加入聲調辨認時，關鍵詞辨認率為 94.54%，加入第二級關鍵詞的聲調辨認器後，關鍵詞辨認率提昇至為 95.32%，錯誤下降率為 14.3%。

A Mandarin Keyword Spotting System Assisted with Tone Recognition

Student：Chin-Chu Chung　　Advisor：Dr. Yih-Ru Wang

Degree Program of Electrical and Computer Engineering
National Chiao Tung University

Abstract

Most of today's Mandarin speech recognition systems use 411 syllables (regardless of tone information) as recognition unit, and most of them could be recognized correctly with the help of language model. However, in the case of keyword spotting, keywords are always Named Entities, such as person names, location names, company names,…, etc. Those keywords are usually only two characters in length and easily confused with each other. So it is important to recognize words with tone information.

In this thesis, two-stage keyword spotting system is used. RAPT (A Robust Algorithm for Pitch Tracking) is applied to get the pitch contour in the feature extraction phase of the original system. The likelihood scores derived from Top-10 keyword recognition are added with the scores from the second stage MLP tone recognizer, and then the scores with Top-10 results are reordered to get better recognized answers.

In this thesis, keyword spotting system is made for a specific keyword phrases: 341 company names (1074 including the aliases) in Hsinchu Science Park. The keyword recognition rate is 94.54% without tone recognition, which increases to 95.32% with the second stage tone recognizer, and the error reducing rate is 14.3%.

誌謝

本論文得以完成，首先感謝我的指導教授王逸如老師與陳信宏老師，感謝兩位老師辛苦的指導與耐心的教誨，讓我得以學習到許多知識與研究的方法，尤其在系統的修改上，王逸如老師更是不遺餘力給於學生指導，令學生受益良多。

這段日子也要感謝實驗的學長、同學及學弟，大家都給我許多的幫助，尤其感謝振宇學長、經展學長及智合學長對我論文上的協助，也要謝謝輝哥學長生活上的開導；感謝銘傑、智誠、豆腐、大胖及小瞎，大家平時的互助與討論，讓我能更快的找到問題。這段日子也要感謝公司主管的支持，與同事的幫忙，讓我在做論文的最後衝刺期間，能無後顧之憂，雖然一邊工作一邊唸書很累，還好有堅持下來，終能完成學業。

當然，也要感謝父母的體諒與支持，由於求學期間，埋首於研究論文，少了許多時間可陪伴父母，今後就更有時間為家裡付出了。

目錄

圖目錄

表目錄

第一章 緒論

1.1 研究動機

隨著科技的進步，人與電腦或隨身裝置乃至於家電產品，希望可以建立一個人性化的溝通橋樑，首先想到的就是人與人之間口語對話，主要就是著重口語的方便性，不需要動用到手，也造就身障者的福音。那如何將具有語音輸入裝置，能夠準確無誤的辨認使用者所下達的聲音指令，就成了一個重要課題。指令、人名以及一些常用的名詞，就是本論文想要辨識的關鍵詞。在辨識關鍵詞上，除了近來已被普遍使用的 HMM (hidden Markov models) model 之外，為了提升辨識率，計畫加上中文語言所呈現的聲調資訊，來幫助關鍵詞辨認，考慮到連續語音中的連音(coarticulation)、下降(declination)及韻律(prosody)效應對中文連續語音中基頻及能量軌跡所造成的影響，以提升在中文連續語音的聲調辨認率。

1.2 研究方向

在中文語音辨認系統中，常使用不含聲調的 411 基本音節來做為辨認的基本單元，因為加上詞典及聲學模型後，大部份同音不同聲調的詞彙均可正確辨認，由[3]可以得到初步的認識。但在關鍵詞辨認系統中，尤其是關鍵詞是公司行號名稱，關鍵詞是相同或相似音必須使用聲調來鑑別的情況十分常見在本研究中，我們嘗試將聲調辨認加入中文關鍵詞語音辨認系統中，聲調辨認使用 MLP(Multi Layer perception)[2]，來進行辨認；為了進行聲調辨認，需使用到基頻(Fundamental frequency, f0)的資訊[3][4][5]，我們使用 RAPT(A Robust Algorithm for Pitch Tracking)[1]演算法來抽取基頻軌跡；另外從系統中抓取關鍵詞的音節切割位置資訊，以供聲調辨認器，辨認出每個音節的聲調。

即時中文鍵關詞辨認系統的建立，由於目前的系統平台存在音節切割位置不準現象，對我們即將加入聲調辨認，將不會有好的辨認結果，我們比對 HTK toolkit 所切出

來的音節位置，準確性相當高，所以將仿照 HTK toolkit 的做法，計畫將由 HTK toolkit 所訓練出來的 HMM(hidden Markov models) model 來替換，並將前級抽取語音特徵參數 MFCC(Mel-frequency cepstrum coefficients)程式，改成跟 HTK 上一樣，以求與 HMM model 的特徵參數算法一致性。修改完系統的前級處理與更換聲學模型(acoustic model)後，我們還會在前級加入自動抽取音高軌跡(pitch contour)程序；預計在後級加入聲調辨認，以聲調辨認的分數來提高中文關鍵詞辨認率。

1.3 章節概要

本論文共分為六章：

第一章 緒論：介紹本論文之研究動機與方向。

第二章 基本關鍵詞辨認系統：介紹辨認系統，如何修改系統，與使用那些資訊來幫助辨認。

第三章 基頻求取的方法：介紹使用RAPT演算法求取基頻軌跡(f0 contour)。

第四章 MLP中文連續語音聲調辨認：介紹聲調所用的特徵參數，與如何使用MLP來辨認音節的聲調。

第五章 實驗結果與分析：結合聲調辨認器與狀態長度模型的實驗結果。

第六章 結論與未來展望。

第二章 基本關鍵詞辨認系統

在本章中，將介紹我們所使用的基本關鍵詞辨認系統之架構，我們將使用HTK(Hidden Markov Model Toolkit)重新求取聲學參數及訓練聲學模型，以獲得更精確的聲學參數及辨認模型，以期獲得較佳之音節切割位置。並修改現有之即時關鍵詞辨認系統之聲學參數求取子系統與聲學模型處理部分，以期將來可以直接將HTK訓練獲得之聲學模型替換本關鍵詞辨認系統的聲學模型。同時將檢查基本關鍵詞辨認系統之音節切割位置，並對辨認錯誤的情況，做一分析。

2.1 關鍵詞辨認系統架構

首先介紹我們實驗室所用的關鍵詞辨認系統，如圖2.1所示，由一開始抽取語音特徵MFCC參數，由參數計算HMM model每個state的likelihood值，經由viteribi beam search，找到每個HMM state最佳的路徑，留下機率比較高的路徑，當語音結束後，我們的viterbi beam search 也跑完整個語音段HMM state，最後比對lexicon tree，找到機率最高的辨認答案，即是我們輸出的辨認結果。HMM model，我們使用中文411基本音節模型，辨認詞典為了節省記憶體，我們使用lexicon tree來展開，並為每個答案設定一個ID碼。

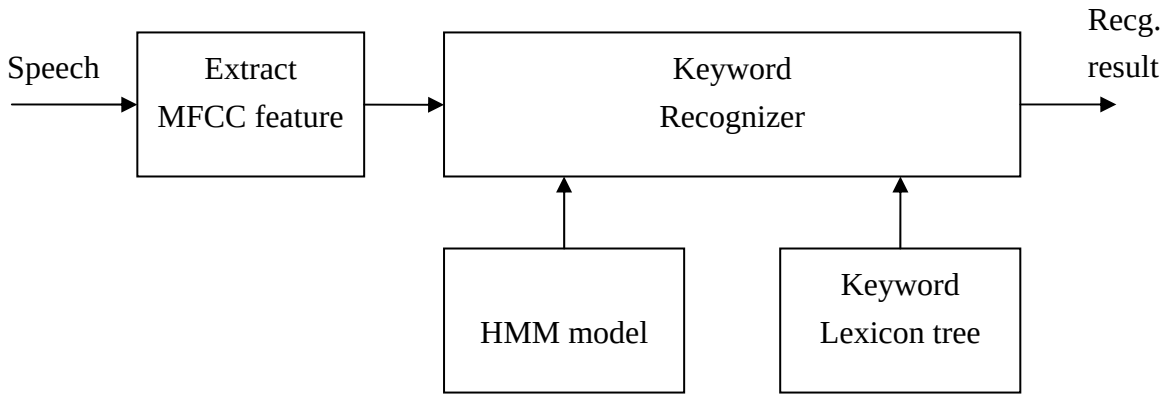

圖 2.1 關鍵詞辨認系統

由圖2.2所示，是我們關鍵詞辨認系統的語法架構，我們假定輸入語句除代辨認之關鍵詞外，允許前後還有一些非關鍵詞語音；我們將它們稱之為前填充字串(pre-filler)及後填充字串(post-filler)；於是我們使用圖2.2來表示，圖中keyword就是我們要辨認的關鍵詞，前後可接填充字串或可不接，前填充字串及後填充字串我們以411基本音節(base syllable)來表示，所以填充字串可跑任意的411基本音節，而關鍵詞的部份有字典(lexicon)路徑限制，所以我們會對跑到填充字串的部份，進行扣分(penalty)，當一句語音最後HMM state路徑有跑到關鍵詞或後填充字串時，才會做關鍵詞辨認，計算辨認分數如式子(2.1)所示，我們得出關鍵詞與填充字串的差值，若差值愈大者，代表愈像關鍵詞，而不像填充字串。

$$\text{score} = \log(\textit{keyword\_likelihood}) - \log(\textit{filler\_likelihood}) \qquad (2.1)$$

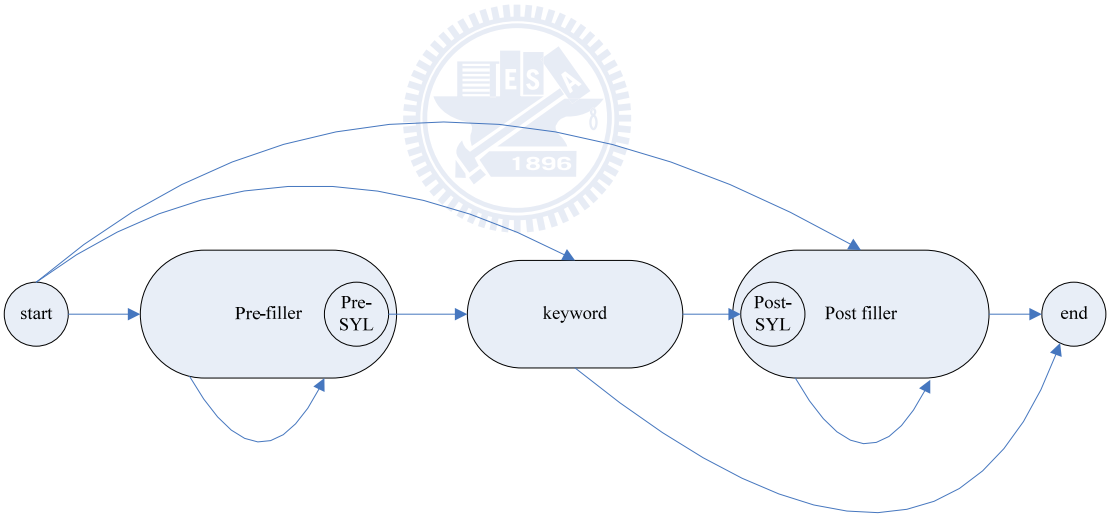

圖 2.2　keyword spotting之語法架構

在中文語音辨認系統中，常使用不含聲調的411基本音節來做為辨認的基本單元，因為加上詞典及語音模型後，大部份同音不同聲調的詞彙均可正確辨認。但在關鍵詞辨認系統中，尤其對關鍵詞是公司行號名稱，關鍵司是相同或相似音必須使用聲調來鑑別的情況十分常見。

在論文中，我們要加入第二階段的聲調辨認，我們要使用的是以音節(syllable)為單

元的聲調辨認器，所以第一階段的音節切割位置會嚴重影響第二階段聲調辨認的準確性，我們需要有一個好的音節切割位置，以供聲調辨認使用。所以我們在此先做了一些基本關鍵詞辨認系統音節切割位置的檢查，由圖2.3所示，可看出我們的基本關鍵詞辨認系統音節切割位置並不準確。

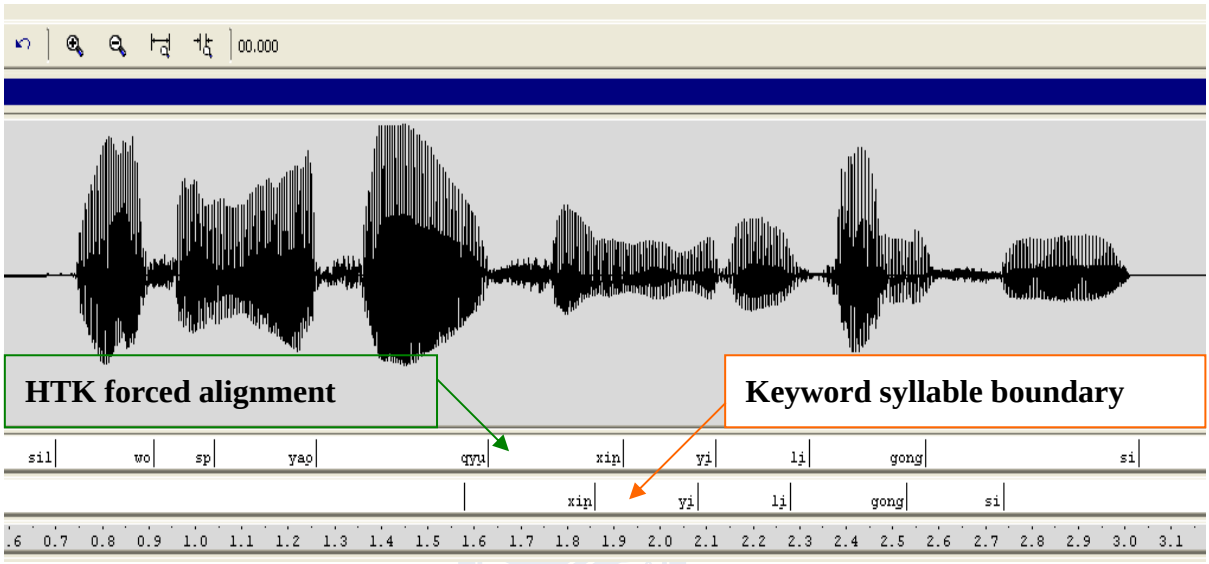

圖 2.3　基本關鍵詞系統的音節切割位置比較圖

由於系統存在音節切割位置不準現象，這會讓預期加入聲調辨認的方法，得不到好的辨認結果；反觀HTK toolkit [6]所切割的音節位置相對準確，所以計畫會將由HTK toolkit所訓練出來的HMM(Hidden Markov Models) model來替換原本系統的聲學模型(acoustic model)，並將前級抽取語音特徵參數MFCC(Mel-frequency cepstrum coefficients)程序，改成跟HTK上一樣，以期訓練與辨認，所抽取語音特徵參數程序的一致性；並且在系統上加入抽取基頻(F0)的資訊，以提供聲調辨認使用。

2.2 訓練聲學模型語料庫

我們要替換原本系統的聲學模型，就需利用HTK toolkit 訓練出新的聲學模型，訓練所需要的語料庫，以中文語料庫而言，字(Character)的類別約有12,000 餘種；若以發聲方式來區分，約有1,300 種帶聲調的音節(Tonal syllable)；若不考慮聲調的類別，則只

有411 種基本音節(Base syllable)。作為訓練聲學模型(acoustic model)的語料，就需具備基本音節的豐富性，如此才能統計出可靠的聲學模型。

TCC-300 語音資料庫是由國立交通大學、國立台灣大學、國立成功大學所共同錄製[7]，台灣大學語料庫主要包括詞以及短句，內容文字經過設計，考慮音節與其相連出現之機率，男女共 100 人錄製而成；交通大學及成功大學為長文語料，其語句內容由中研院提供之 500 萬詞詞類標示語料庫中選取，每篇文章包含數百個字，分別各 100 人朗讀錄製，且每人所朗讀之文章皆不相同，統計如表 2-1 所示。音檔的取樣頻率為 16k Hz，取樣位元解析度為 16bits。

表 2.1 TCC300語料統計表

學校名稱	文章屬性	語者總數	總音節數	檔案總數
台灣大學(NTU)	短文(平衡句)	女 50 男 50	女 24504 男 27309	女 3068 男 3404
交通大學(NCTU)	長文	女 50 男 50	女 73559 男 75058	女 616 男 622
成功大學(NCKU)	長文	女 50 男 50	女 68756 男 62549	女 582 男 583

2.3 聲學模型的建立

我們將聲母(initial)依後面接的韻母(final)關係，分成 100 類，韻母分成 40 類，將已經有音節切割資訊的音檔，利用 HTK(HMM Tool Kit) 訓練 HMM 模型。我們依據中文語音的特性，且子音長度通常比母音來得短，所以 HMM model 子音設定 3 state，母音設定 5 state，抽取 38 維的 MFCC 特徵參數，包括 12 維的 MFCC 特徵參數，12 維的 MFCC delta term 和 1 維 log energy delta term，與 12 維的 MFCC acceleration term 和 1

維 log energy acceleration term。依樣本個數訓練出平均 32 維 GMM (Gaussian Mixture Models) 的 HMM 模型，表 2-2 是我們抽取 MFCC 參數的設定值。

表 2.2 抽取MFCC特徵參數的設定值

NATURALWRITEORDER	TRUE
SOURCEFORMAT	ALIEN
HEADERSIZE	4096
SOURCERATE	625
TARGETKIND	MFCC_E_D_A_N_Z
TARGETRATE	100000
SAVECOMPRESSED	F
SAVEWITHCRC	T
WINDOWSIZE	320000
ZMEANSOURCE	T
USEHAMMING	T
PREEMCOEF	0.97
NUMCHANS	24
USEPOWER	F
CEPLIFTER	22
LOFREQ	0
HIFREQ	8000
NUMCEPS	12
ENORMALISE	T
DELATWINDOW	2
ACCWINDOW	2

為了確認我們統計的聲學模型，其樣本數量是否足夠，圖 2.3 及圖 2.4 分別是聲母與韻母資料數量統計圖，由圖中可以看出有些模型樣本數量非常少，事實上，就是我們的對話中比較少出現的用字，但為了避免因為資料太少，統計出來的模型，影響我們關鍵詞辨認系統的辨認率，我針對韻母模型中數量最少的三個，分別為“eh”、“yo”及“yai”，我們使用相似音替代，經實際對系統測試，只要替換掉“eh”(ㄝ)即可，替換前辨認率為 81%，替換後辨認率上升到 92%，為何一般對話中，較少出現的音，會影響這

麼大呢？我們檢視了測試音檔，其中音檔句子關鍵字真的會有出現“廿”相關的發音，如“悅達科技”及“凌越科技”等，其中的“悅”及“越”其韻母就是“廿”的發音。

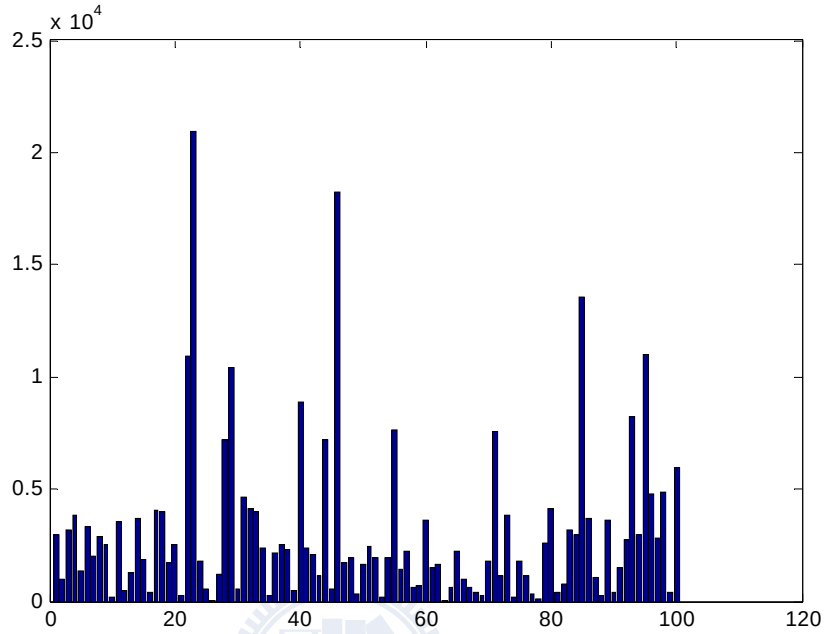

圖 2.3 聲母數量統計分佈圖

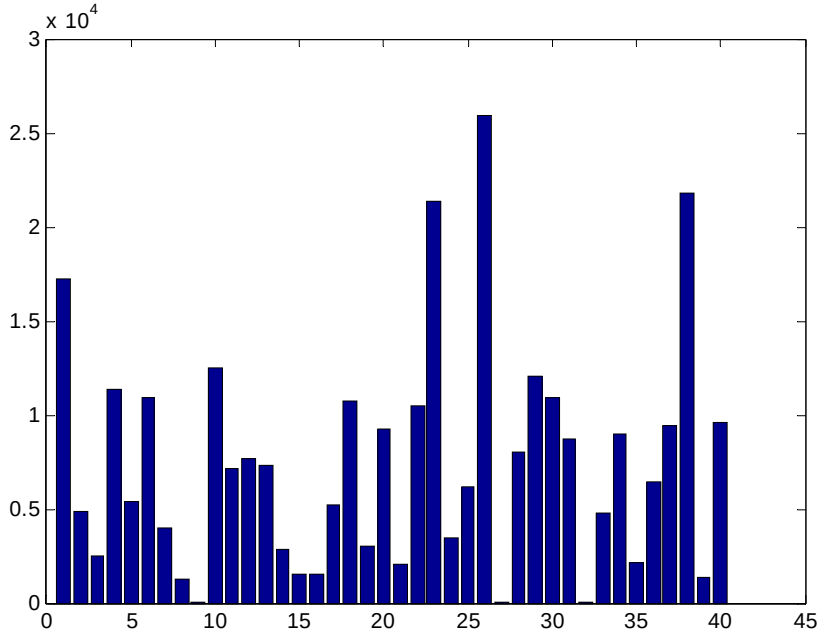

圖 2.4 韻母數量統計分佈圖

我們訓練模型使用 TCC300 所有語料，檔案總數 8032，音節總數 297659，語者總數 300，測試我們取 TCC300 其中的 108 檔案，音節總數 35905，語者總數 18 位。

以下是 HTK 測試結果：

WORD: %Corr=67.14, Acc=66.42 [H=17696, D=374, S=8287, I=189, N=26357]

N: total number of labels in the defining transcription files.

H: number of correct labels

D: number of deletions.

S: number of substitutions.

I: number of insertions.

$$\text{Correct} = \frac{H}{N} \times 100\%$$

$$\text{Accuracy} = \frac{H - I}{N} \times 100\%$$

2.4 系統改善檢查與辨認錯誤分析

將系統聲學模型換上我們由 HTK toolkit 所訓練出的聲學模型，並將原關鍵詞辨認系統中 MFCC 特徵參數之抽取方式，改成跟 HTK 一致，以求訓練模型與辨認的特徵參數的一致性，建立基本關鍵詞辨認系統。到此系統對音節切割位置準確性已大幅提昇。

在抽取 MFCC 參數的同時，我們另外加入了 Real-time CMN (Cepstral mean normalization)，來增加辨認多語者的強健性。

CMN 是用來移除倒頻譜平均值(cepstral mean)，定義如式子(2.2)所示。

$$\hat{Y}[n] = Y[n] - \bar{Y}[n] \qquad (2.2)$$

在 HTK 中，$\bar{Y}[n]$，倒頻譜平均值在非即時系統中通常是以句子為單位求取，而在實際系統中，為了達到 Real-time，所以我們在語音輸入訊號累積一段之後，就先計算這一段的倒頻譜平均值，來供前級計算 MFCC 使用，之後語音訊號進來，我們每隔一定時間會對倒頻譜平均值做更新，如此可達到當語音訊號輸入的同時，就可計算 MFCC，不

用等到句子結束時才做。

接下來，我們檢視系統對音節切割位置改善後的成果。我們與 HTK forced alignment 所切出來的位置做比較，只比較關鍵詞的音節切割位置部份。由圖 2.5 及圖 2.7 可看出，未修正系統前的音節切割位置，準確度並不高；圖 2.6 及圖 2.8 為改善之後的成果，由圖中可看出改善後，音節切割位置準確度相當高，已能符合供聲調辨認使用。我們對系統改善後的切割位置，統計改善成果，與 HTK 切割位置的落差：3.87%，平均 syllable length = 26.97 frame，比較音檔數量 769，比較 syllable 總數 3827，所以將落差換算成 frame 數的為 1.04frame = 26.97*0.0387。

只比較 keyword，計算式子：

1. 先計算落差佔一個 syllable 的比重。

   sum += (abs(key_ini – htk_ini) + abs(key_fin – htk_fin)) / (htk_fin – htk_ini)

   其中 abs 是取絕對值，key_ini 與 key_fin 分別是 speakerX 系統切割出來音節的開始位置與結束位置，htk_ini 與 htk_fin 分別是 HTK forced alignment 所切割出來音節的開始位置與結束位置。

2. sum / cnt, 最後統計總共的落差 ratio

   其中 cnt 是我們比較的音節總數。

雖然我們系統的音節切割位置資訊已相當準確了，不過我們還是對有落差的情況，做了分析，容易造成落差的情況：

1. 句首起始位置。
2. 句尾結束位置。
3. sp(short pause) keyword spotting system 不容易切出來。
4. 連音的地方。

上述 1 情況，我們系統是會將起始位置往前一些；情況 2，我們系統是會將結束位置往後延遲一些，因這兩種情況，前後都是無聲(silence)，也有沒有其它的基頻，所以並不影響聲調辨認；情況 3，音節間的停頓不容易被分辨出來，我們改成利用音節間基

頻軌跡斷開的長度，來代表 SP，也符合我們要抓取連音的影響程度的依據；情況 4，當音節彼此連音時，切割位置會與正確位置稍為有點落差，不過平均落差上述統計結果只有一個 frame，佔一個音節長度的比重很小，只有 3.87%，所以對我們要做聲調辨認，亦不影響。以上結論，系統的音節切割位置，已經足夠給聲調辨認使用了。

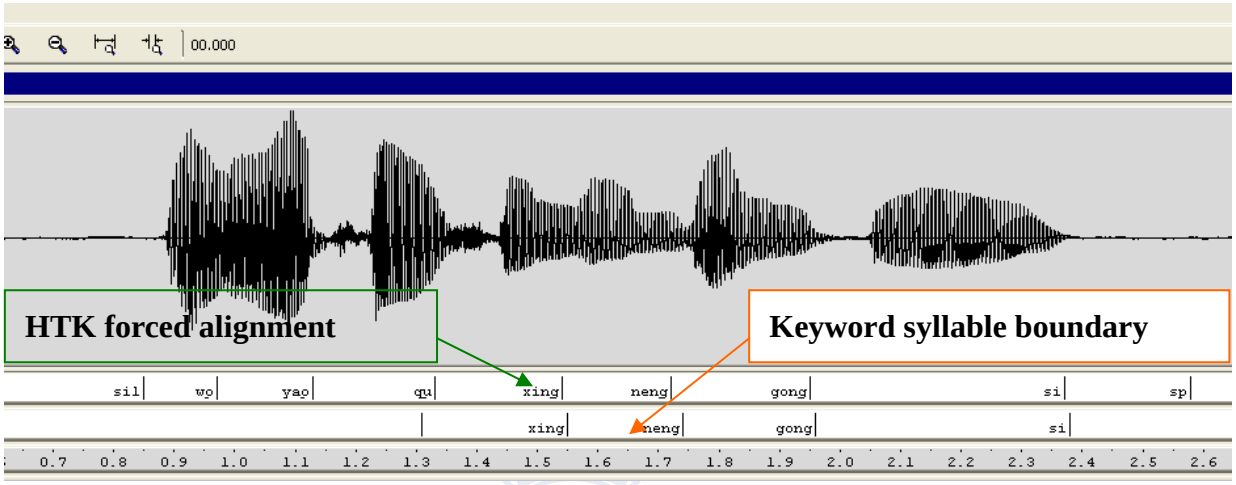

圖 2.5　系統原本的音節切割位置比較圖(一)

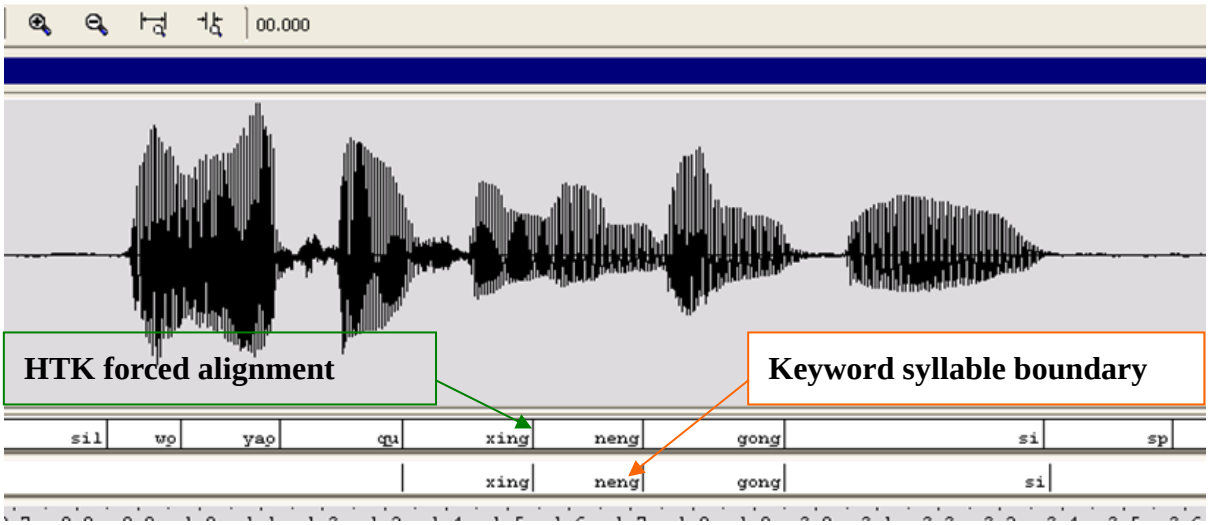

圖 2.6　系統修改後音節的切割位置比較圖(一)

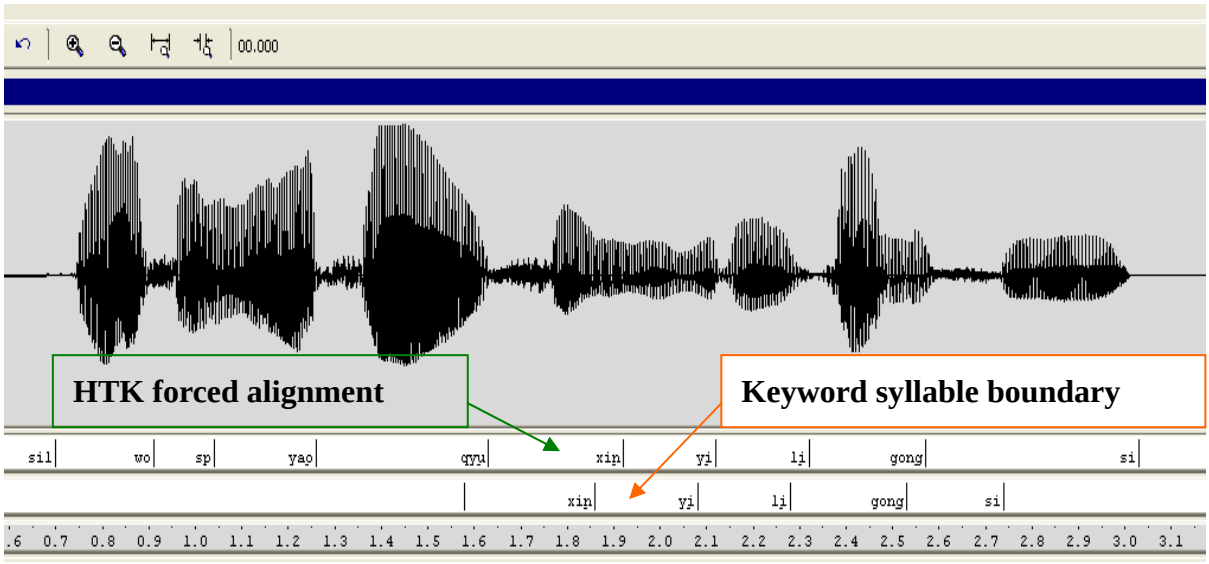

圖 2.7　系統原本的音節切割位置比較圖(二)

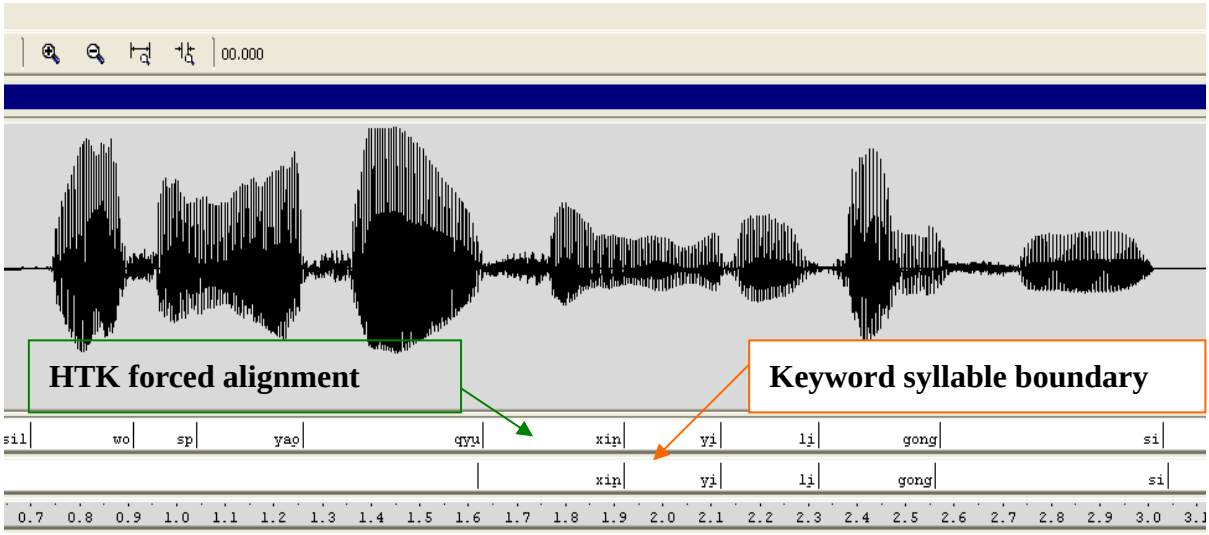

圖 2.8　系統修改後音節的切割位置比較圖(二)

當系統換上新的 HMM model 後，並修改前級抽取 MFCC 的程式，我們進行系統辨認測試，並對系統辨認錯誤的句子做分析，主要可以分三類，與答案音節切割位置幾乎一致、關鍵詞長度不一致、及音節切割位置不一致。

2.4.1 音節切割位置幾乎一致

分數前幾名，其音節切割位置大致一樣，此種錯誤情況，佔總錯誤率51.9%，例子如表2.3所示，由表2.4中可看出其辨認答案所對應到的聲調大多不同，所以我們可以利

用聲調辨認來提高正確答案的分數。

表 2.3　音節切割位置幾乎一致

排名	Likelihood 分數	關鍵詞中每一音節切割位置：(begin frame - end frame)				辨認結果
1	6.467	83 - 105	105 - 122	122 - 140	140 - 154	眾晶公司
2	5.844	85 - 104	104 - 122	122 - 140	140 - 154	永進公司
3	5.343	84 - 104	104 - 122	122 - 140	140 - 154	悠景公司
4	5.245	83 - 104	104 - 122	122 - 140	140 - 154	仲琦公司
5	5.145	84 - 105	105 - 122	122 - 140	140 - 154	榮群公司
6	4.565	93 - 105	105 - 122	122 - 140	140 - 154	鴻景公司
7	2.168	104 - 123	123 - 140			星通
8	1.816	106 - 118	118 - 140			義隆
9	0.81	83 - 104	104 - 119			仲琦
10	-1.496	84 - 93	93 - 105			友旺
正確答案：2 永進公司						

表 2.4　辨認結果與聲調對照表

辨認結果	聲調
眾晶公司	4_1_1_1
永進公司	3_4_1_1
悠景公司	1_3_1_1
仲琦公司	4_2_1_1
榮群公司	2_4_1_1
鴻景公司	2_3_1_1
星通	5_5
義隆	4_2
仲琦	4_2
友旺	3_4

2.4.2 辨認結果長度不一致

此類辨認結果其詞長大多都比正確答案來得長，此種錯誤情況，佔總錯誤率33.3%，例子如表2.5所示。由表中可觀察到，有些音節長度過短的情況，比如一個音節只有8個frame，等於平均一個state只有一個frame，不符合語速特性，所以我們計畫以狀態長度模型來訂正此類錯誤。

表 2.5 辨認結果長度不一致

排名	Likelihood分數	關鍵詞中每一音節切割位置：(begin frame - end frame)						辨認答案
1	16.399	166 - 202	202 - 233	233 - 263	263 - 294	325 - 333	341 - 349	捷誠科技公司
2	15.862	166 - 202	202 - 233	233 - 263	263 - 294			捷誠科技
3	10.417	201 - 213	213 - 233	233 - 263	263 - 294			啟亨科技
4	10.315	202 - 217	217 - 233	233 - 263	263 - 294			致遠科技
5	10.153	202 - 211	211 - 232	232 - 263	263 - 294			智森科技
6	9.936	201 - 225	225 - 233	233 - 263	263 - 294			前源科技
7	5.11	62 - 70	90 - 131					合勤
8	2.871	263 - 280	280 - 294					金麗
正確答案：2 捷誠科技								

2.4.3 音節切割位置不一致

此類辨認結果，分數前幾名，其音節切割位置差異較大，此種錯誤情況，佔總錯誤率14.8%，例子如表2.6所示。由表中可觀察到，有些音節長度過長的情況，所以我們計畫同樣以狀態長度模型來訂正此類錯誤。

表 2.6 音節切割位置不一致

排名	Likelihood分數	關鍵詞中每一音節切割位置:(begin frame - end frame)				辨認答案
1	16.689	124 - 149	149 - 191	191 - 221	221 - 273	漢基公司
2	15.869	149 - 168	168 - 191	191 - 221	221 - 273	晶宇公司

3	15.505	149 - 165	165 - 191	191 - 221	221 - 273	天鈺公司
4	15.388	149 - 168	168 - 191	191 - 221	221 - 273	信越公司
5	14.971	154 - 168	168 - 191	191 - 221	221 - 273	凌越公司
6	14.838	148 - 164	164 - 190	190 - 221	221 - 273	致遠公司
7	14.83	148 - 165	165 - 190	190 - 221	221 - 273	前源公司
8	14.764	154 - 165	165 - 191	191 - 221	221 - 273	瑞昱公司
9	14.757	149 - 171	171 - 191	191 - 221	221 - 273	勁取公司
10	14.654	149 - 177	177 - 190	190 - 221	221 - 273	吉聯公司
正確答案：2　晶宇公司						

2.5 狀態長度模型(state duration model)

由於辨認錯誤的狀況存在著音節長度過長或過短的現象，此原因為Viterbi search對HMM 模型計算每個狀態轉移最大相似機率，圖2.9呈現狀態轉移的變化，此方法並未對每個狀態停留的長度做限制，所以才會造成有音節長度太長或太短的現象。

我們統計狀態長度資料，發現其分佈的特性，資料集中，且狀態長度只會有大於零的正值，所以決定用伽瑪分佈（Gamma distribution）來描述[8]，用以限制每個狀態可停留長度的範圍，如此我們在每個狀態轉移的時候，加上伽瑪機率，當碰到較短或較長的狀態長度，呈現出來的伽瑪機率會較低，因此會扣較多分數，以達到去除錯誤的狀態轉移路徑，突顯出較好的狀態轉移路徑。

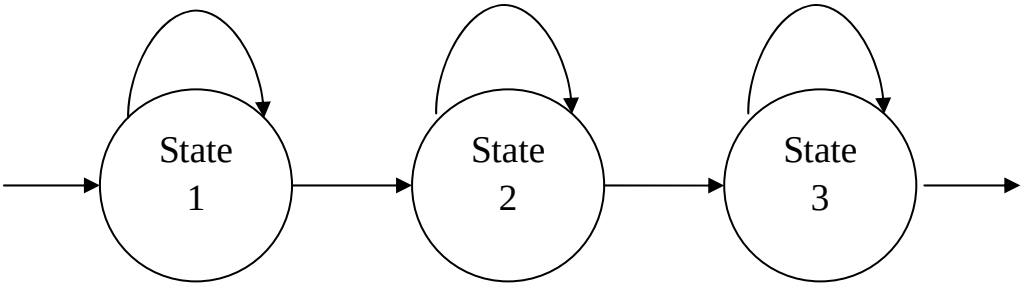

圖 2.9　句子的狀態轉移圖

為了得到狀態長度模型，我們將100個聲母與40個韻母，分別統計每個state的平均值與變異數，由式子(2.4)及(2.5)我們可取得(2.3)伽瑪機率密度函數所需的參數。

聯合伽瑪分佈機率密度函數(pdf):

$$p_i(d) = \frac{\eta_i^{v_i} d^{v_i - 1} e^{-\eta_i d}}{\Gamma(v_i)} \tag{2.3}$$

聯合伽瑪分佈其平均值與變異數定義：

$$E[d] = v_i \eta_i^{-1} \tag{2.4}$$

$$\text{var}[d] = v_i \eta_i^{-2} \tag{2.5}$$

$\log(\Gamma)$ 我們可以由查表得知。

第三章 基頻求取單元

我們在系統中要加入聲調辨認，而在聲調辨認時，最常使用的聲學參數是基頻軌跡的資訊，基頻軌跡是由每個單位時間，所對應到的基頻(fundamental frequency，F0)。而求取語音信號之基頻軌跡並不是一件容易的事，一般所求取之基頻軌跡常會有有聲及無聲音判斷錯誤及基頻軌跡不連續，會出現半頻或倍頻等基頻錯誤。這些錯誤將影響聲調辨認系統的正確率。在本章中，將介紹我們的結合聲調辨認之中文關鍵詞辨認系統中所使用的基頻求取單元，並檢查所求得基頻軌跡的正確性。

Pitch tracking 的技術有很多種，舉凡利用 Direct waveform processors，ACF (Autocorrelation Function)，AMDF (Average Magnitude Difference Function)，SIFT (Simple Inverse Filter Tracking)，Cepstrum 等都是，而我們系統採用 RAPT (A robust algorithm for pitch tracking) [14] 演算法，其架構在 NCCF (normalized cross-correlation) [13]方法的基礎上。

一段句子包括清楚(clear)和不確定(problematic)的語音聲音區段(voice speech segments)以及無聲區段(unvoiced segment)，如圖 3.1 所示。 所以我們希望在語音聲音區段能標示出 F0，在無聲區段則可以明確的將 F0 標示為 0。

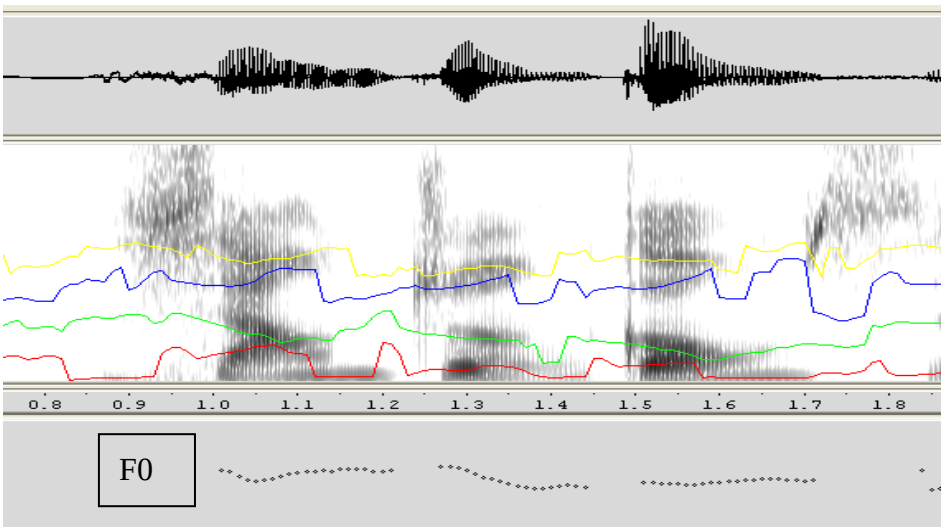

圖 3.1　音高軌跡(pitch contour)

為了能夠獲得穩健且準確的基頻 (F0)，和較小的運算複雜度，以及較小的延遲時間，所以我們選擇使用 RAPT 演算法，它能適用不同的取樣頻率和不同的幀(frame)大小，其架構在 NCCF 方法的基礎上。

NCCF 定義如下：

$$R_{i,k} = \frac{\sum_{j=m}^{m+n-1} s_j s_{j+k}}{\sqrt{e_m e_{m+k}}} \quad , k = 0 \sim K-1;\ m = iz;\ i = 0, M-1 \tag{3.1}$$

$$e_j = \sum_{l=j}^{j+n-1} s_l^2 \tag{3.2}$$

取樣週期 T=1/Fs, Fs 為取樣頻率，i 是幀(frame) 指標有 M frames, k 是延遲指標，每個 frame 間隔 t, 分析的視窗大小 w, 每個 frame 位移 $z=t/T$ 的取樣點，一個視窗 $n=w/T$ 的取樣點。

$-1 \le R \le 1$，若是真的基頻(F0)的話，R 的值會趨近於 1；若是白雜訊(white noise)的話，$R_{i,0}=1$，$R_{i,k}$ 會近似於零當 $k \neq 0$，NCCF 的值與訊號振幅大小無關。

典型的語音信號和 NCCF 的關係，有以下幾點特性：

- 在語音區段，最大的 *R* 對應到正確的 F0，且通常 *R* 值趨近於 1。
- 當多個趨近於 1 的 *R* 值存在，通常選擇對應到最短週期的 *R* 值，即頻率較高的 F0。
- 毗鄰 frame 的 F0 的差值不大。
- 一段語句 F0 狀態趨向於低頻，也就是說句首通常 F0 會較高，句尾會較低。
- 語音段與無聲段在短時間的頻譜其差異是相當大的。
- 聲音的振幅特性，增加在聲音的開始，下降在聲音的結束。

RAPT 演算法的系統流程圖，如圖 3.2 所示，我們先做一個概要說明：

- RAPT 的前處理，輸入訊號的每一幀減掉自己幀的平均值。
- 使用 NCCF 算法，得到自相關 *R* 值。
- 每個幀保留幾個較高的 *R* 當做候選者(candidate)。
- 在保留的候選名單中，利用拋物線插值法，找到區間內的最佳值，以提高其解析度。
- RAPT 的後處理，動態挑選最佳的 F0，使用 Viterbi 演算法來決定最佳的 F0 路徑。

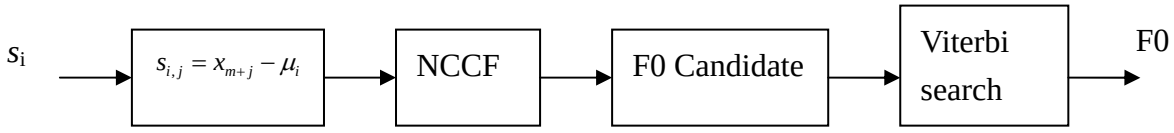

圖 3.2　RAPT流程圖

為了方便，我們將演算法相關的所有的常數與符號定義如下：

表 3.1　RAPT常數定義表

constant	meaning	value
F0_min	minimum F0 to search for (Hz)	50
F0_max	maximum F0 to search for (Hz)	400
t	analysis frame step size (sec)	0.01
w	correlation window size (sec)	0.0075
CAND_TR	minimum acceptable peak value in NCCF	0.3
LAG_WT	linear lag taper factor for NCCF	0.3
FREQ_WT	cost factor for F0 change	0.02
VTRAN_C	fixed voicing-state transition cost	0.005
VTR_A_C	delta amplitude modulated transition cost	0.5
VTR_S_C	delta spectrum modulated transition cost	0.5
VO_BIAS	bias to encourage voiced hyppotheses	0
DOUBL_C	cost of exact F0 doubling or halving	0.35
A_FACT	term to decrease R of weak signals	10000
N_CANDS	max. number of hypotheses at each frame	20

表 3.2　RAPT符號定義表

symbol	meaning
x_m	m-th sample of the input speech signal
F_s	sample rate of speech signal = 1/T
F_{ds}	reduced sample rate of speech for first-pss NCCF
round(v)	the integer that is closest to v
n	the number of samples correlated at each lag=round(wF_s)
z	the frame step size in samples = round($t\ F_s$)

i	the analysis frame index incrementing at a rate of 1/Tz
K	the longest lag at each frame = round(Fs/F0min)
$R_{i,k}$	normalized cross-correlation for frame I at lag k

3.1 計算 NCCF 與訊號的前處理

首先輸入訊號，我們先經過前處理，抓取每個框大小的訊號，減去每個框的平均值，用以去掉輸入訊號所存在的偏壓值，如式子(3.3)，若框的平均值是零的話，代表無語音訊號，此音框就可不用算 F0 了，直接設定為零。

$$s_{i,j} = x_{m+j} - \mu_i \ , \ m = iz; \ \ j = 0 \sim n + k - 1, \tag{3.3}$$

$$\mu_i = \frac{1}{n} \sum_{j=m}^{m+n-1} x_j \tag{3.4}$$

我們修改 NCCF 運算式，在分母加入一個常數值 A_FACT，如式子(3.5)所示，用意是降低無聲區段(silent)的 R 值，提高與聲音段的落差。因為無聲區段的能量會較低，所以加入了 A_FACT 常數值所佔的分母比例就相對變高，所以會壓低 R 值，若是聲音區段其能量會較大，故 A_FACT 較無影響。

$$R_{i,k} = \frac{\sum_{j=0}^{n-1} s_{i,j} s_{i,j+k}}{\sqrt{\text{A\_FACT} + \text{e}_0 e_k}} \tag{3.5}$$

$$e_j = \sum_{l=j}^{j+n-1} s_{i,l}^2 \tag{3.6}$$

3.2 求取基頻的後處理

為了減少運算量，我們訂定 F0 尋找範圍，將 F0 最小值定在 50Hz，F0 最大值定在 400Hz，符合一般男生或女生的 F0 會落在這個區間。

F0 的候選者，我們每個幀選取前 20 名 *R* 值較高者，然後針對每個幀，使用 Viterbi 演算法進行搜尋。在進行 Viterbi search 時，我們訂定幾個狀態轉移成本函數(cost function)。

每個幀的候選者本身的成本函數：

聲音框(voiced frame)候選者的成本函數：

$$d_{i,j} = 1 - C_{i,j}(1 - \beta L_{i,j}),\ 1 \le j < I_i \tag{3.7}$$

無聲框(unvoiced frame)候選者的成本函數：

$$d_{i,I_i} = \text{VO\_BIAS} + \max_j(C_{i,j}) \tag{3.8}$$

Ii 是每一幀的候選者數量，*i* 是幀的的指標，1≤ *Ii* < N_CANDS。*Ci,j* 是第 *i* 幀中的第 *j* 個最大的 R 值，-1≤ *Ci,j* ≤ 1，*Li,j* 是對應到 *Ci,j* 延遲長度。

$\beta = \text{LAG\_WT}/(F_s/\text{F0}_{\min})$，LAG_WT 調整的梯度，對於較長的延遲長度，以減少其價值，以便在聲音幀(voiced frame)選擇較短的延遲。在無聲幀(unvoiced frame)*Ci,j* 趨近於零，VO_BIAS 是為聲音似然度(likelihood)的調整偏壓值。

前後都是聲音幀的成本函數：

$$\delta_{i,j,k} = \text{FREQ\_WT} \times \min\{\xi_{j,k}, (\text{DOUBLE\_C} + |\xi_{j,k} - \ln(2.0)|)\} \tag{3.9}$$

$$\xi_{j,k} = |\ln\frac{L_{i,j}}{L_{i-1,k}}|,\ 1 \le j < I_i;\ 1 \le k < I_{i-1} \tag{3.10}$$

DOUBLE_C 是正的常數，這將使狀態轉移成本是個增加的函數，由式子(3.10)可看出，依照前後幀頻率的比例增加，通常毗鄰的音框，其頻率變化不大，故$\xi_{j,k}$會趨近於零。FREQ_WT 是一個正的常數，用來調整每一幀之間轉換成本的權重。

前後幀都是無聲音框(unvoiced frame)：

$$\delta_{i,I_i,I_{i-1}} = 0 \tag{3.11}$$

前後幀是不相同的定義：

聲音幀(voiced frame)接無聲幀(unvoiced frame)：

$$\delta_{i,I_i,k} = \text{VTRAN\_C+(VTR\_S\_C)}S_i + \text{(VTR\_A\_C)}rr_i,\ 1 \le k < I_{i-1} \tag{3.12}$$

無聲幀(unvoiced frame)接聲音幀(voiced frame)：

$$\delta_{i,I_i,k} = \text{VTRAN\_C+(VTR\_S\_C)}S_i + \text{(VTR\_A\_C)} / rr_i,\ 1 \le j < I_i \tag{3.13}$$

VTRAN_C,VTR_S_C和VTR_A_C都是正的常數，S_i是頻譜靜態函數。

$$rr_i = \frac{rms(i)}{rms(i-1)} \tag{3.14}$$

rms 為前後幀能量的斜率，如果語音訊號振幅是上升的話，$rr > 1$，若是下降的話 $0< rr < 1$。

$$S_i = \frac{0.2}{\text{mfcc\_delta(i,i-1)}} \tag{3.15}$$

Si 用來觀測訊號頻譜變化的函數，與頻譜變化快慢成反比，即變化快，*Si* 的值就小。

由式子(3.12)與(3.13)中，描述著聲音幀接無聲幀或無聲幀接聲音幀，其著重在能量斜率與前後幀頻率的差值，主要是因為當聲音剛開始或結束時，其能量的斜率較明顯，且無聲段與有聲段短時間的頻譜差異也比較大。

接下來我們決定每一幀最佳的路徑，其選擇的函數為式子(3.16)，我們選擇最小的狀態轉移成本：

$$D_{i,j} = d_{i,j} + \min_{k \in I_{i-1}} \{D_{i-1,k} + \delta_{i,j,k}\},\ 1 \le j \le I_i \tag{3.16}$$

初值設定：

$$D_{0,j} = 0,\ 1 \le j \le I_0;\ I_0 = 2 \tag{3.17}$$

最後將每一幀延遲的時間長度結果，轉換成頻率表示：

$$F0_i = \frac{F_s}{L_{i,j}} \tag{3.18}$$

3.3 求取基頻的結果

為了驗證 RAPT 演算法，所求到的 F0 的準確度，我們將使用 RAPT 演算法所求到的 F0，與 WaveSurfer 軟體[9]是使用 ESPS 演算法所求得的 F0 做比較，圖 3.3 我們是秀出一句話的波形與音高軌跡，由圖中可看出我們所求出的 F0 與 WaveSurfer 軟體相當近似，且在音節的開始與結束的地方，語音通常較不穩定，相較之下我們 RAPT 表現較好，F0 呈現較為平滑化與穩定。

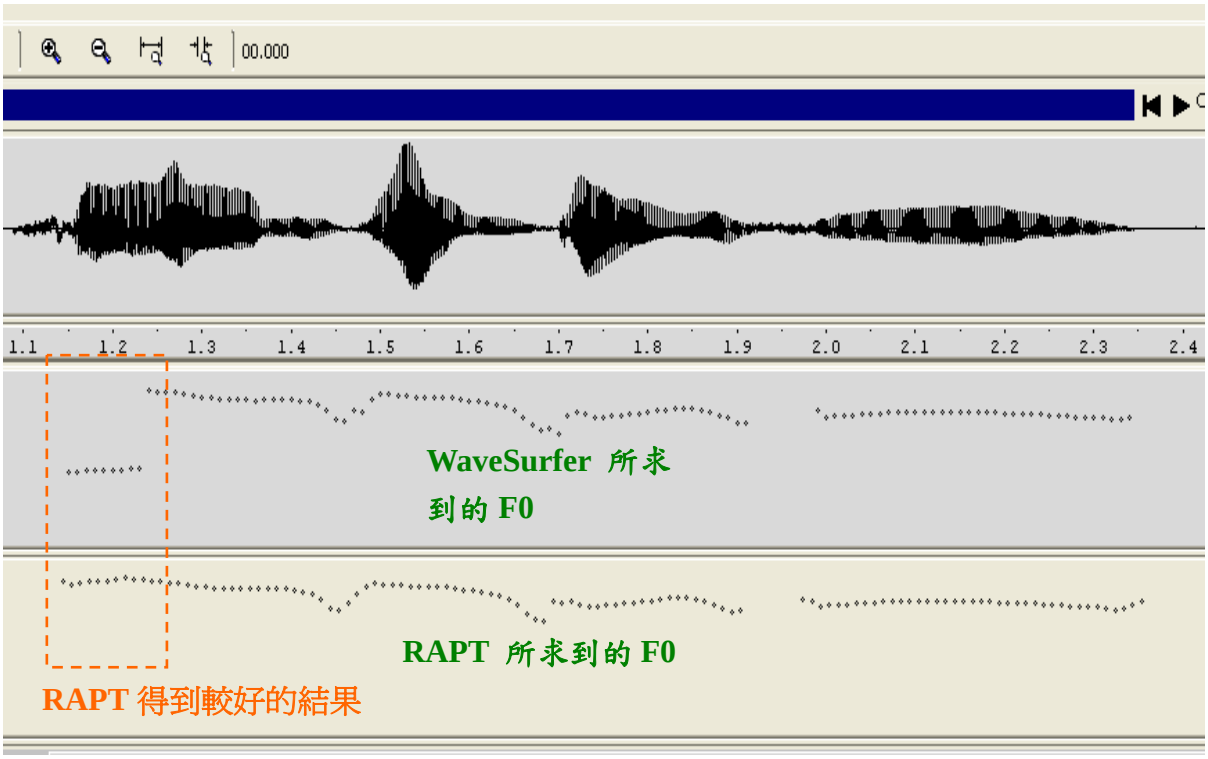

圖 3.3 做用RAPT求取的F0與用WaveSurfer求取的F0比較

第四章 MLP 中文連續語音聲調辨認

為了提升中文關鍵字辨認的辨認率，我們觀察基本關鍵詞辨認系統中每一個音檔辨認結果的前十名結果，因前十名結果對應到的答案其聲調大多數不同，所以加入關鍵詞的聲調辨認，就具有鑑別性。本章節中使用中文連續語音聲調辨認，是利用 MLP（multi-layer perceptron）類神經網路(neuron network)的方法來當辨認器，對單一音節作聲調辨認。本章中將介紹聲調的特徵，以及如何取得聲調的特徵參數，並介紹如何利用特徵參數來訓練 MLP 類神經網路，之後我們對 MLP 聲調辨認做測試，以得知我們聲調的辨認率。

4.1 中文聲調的特徵

英文發音強調重音與節奏感，而中文有別於英文，是一種強調聲調的語言，中文音節的結構主要由 411 個基本音節與五種聲調所組成，其中聲調表現在音高的軌跡(pitch contour)，音高(pitch)指的是聲帶振動的頻率，而振動的頻率就是發音的基頻(fundamental frequency，F0)。我們在發音時，音高會隨著時間，而做高低起伏的變化，因此產生了不同的聲調，所以音節的音高軌跡就是我們判斷聲調的重要依據。一般聲調區分五種聲調，分別為一聲(high-level)、二聲(mid-rising)、三聲(mid-falling-rising)、四聲(high-falling)，此外還有五聲(輕聲，neutral tone)，五聲的音高軌跡通常較不規則，五聲的音高輪廓是不固定的，容易受前後音節影響，且在能量的表現上，也比其它聲調來得底，同時在音節長度上，也是比其它聲調來得短。

一般而言，每一個人的音高平均值、音高範圍都不相同，而女生所發出的頻率都比男生來的高，就如同男生聲音大多比較低沉；而語者說話的速度，也會造成音節長度的不同；語者音量的大小，就表現在音節的能量上；若由一個句子來看，音高表現在句首

的音節平均高度通常會比句尾音節平均高度來得高；同樣的，音節的能量也是，如圖 4.1 所示。

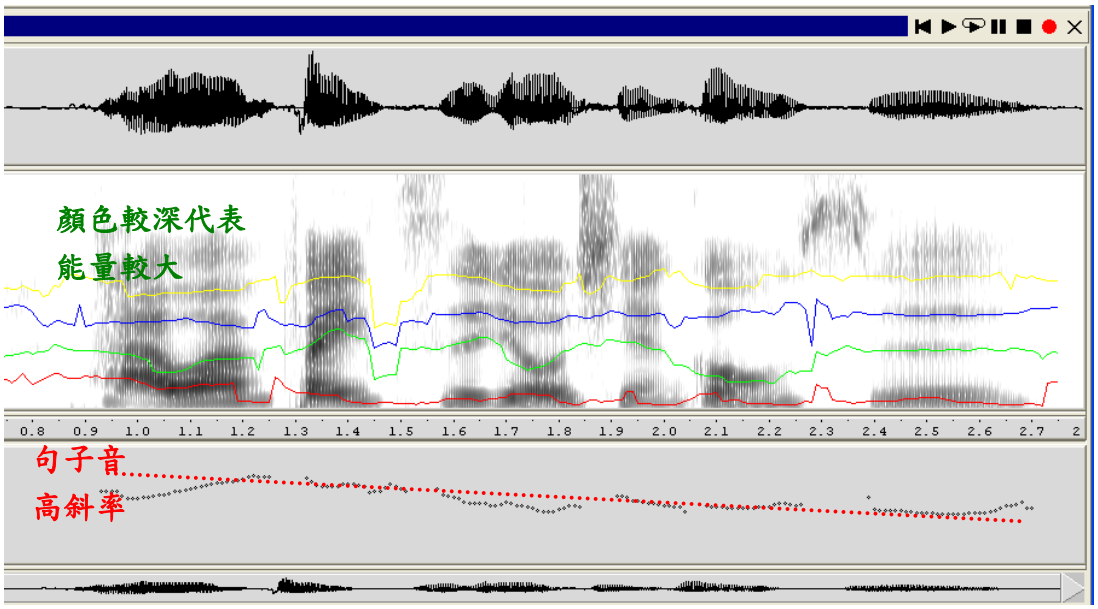

圖 4.1　句子的能量與音高分佈

我們由單一音節，所發出的聲調，看其音高軌跡的標準形式如圖 4.2 所示，各自具有獨特的基頻軌跡分佈。圖中沒有標示出第五聲，主要是因為容易受前後音節影響，音高軌跡較無規則性。

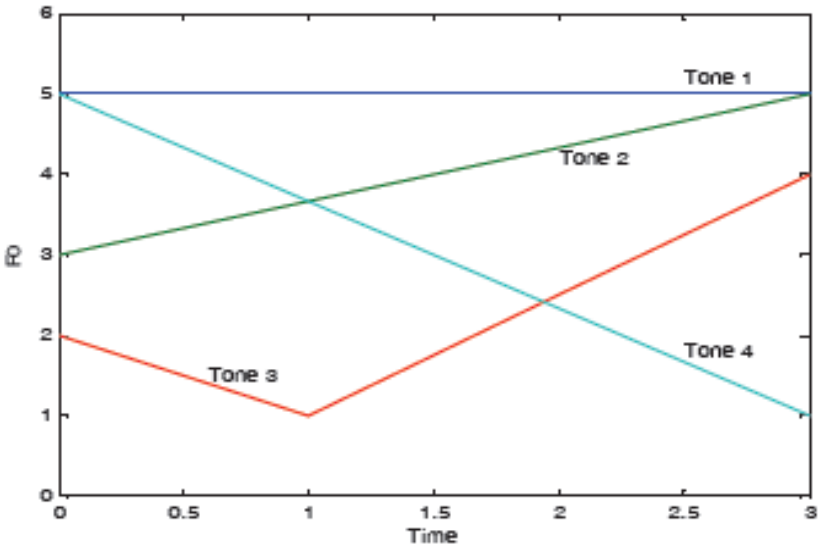

圖 4.2　音節聲調的基頻軌跡

我們由圖 4.2 中可看出，一聲與四聲其差異相當大，而三聲一般會有前半部下降軌跡沒有表現出來，而與二聲容易混淆；連續語音聲調的影響因素，聲調受後面音節影響比前面音節來得大，而三聲會有發生變調(tone-sandhi)的狀況，如三聲接三聲的情況下，前面大多會變成二聲的發音，若是連續的三聲出現，通常最後一個是三聲，前面的三聲一般會變成二聲的情況，此時音高軌跡就不會有先降後昇(falling-rising)的表現。

4.2 聲調辨認特徵參數

如何決定聲調辨認的特徵參數，首先就是要考量與聲調相關的影響因素，其最重要線索，就是前面所提及的音高軌跡；還有聲調受前後音節的影響，音節的影響強弱，表現在連音 (coarticulation)上，若連音愈強，影響程度愈高；音節的能量大小也與音節平均音高有正相關；音節的長度與音節位於句首、句尾或句中的位置，會有相對長短的差異。

抽取特徵參數的前處理，我們將基頻(F0)取 log，然後進行正規化(normalize)，接下來是對能量取 log，然後進行正規化，其中音框大小 30msec，位移大小 10msec。

能量的計算式子如下：

$$e_i = \frac{\sum_{i=j}^{j+n-1} s_j^2}{n}$$ ，n:一個音框的取樣個數，i 是幀的指標

我們將抽取出 20 個特徵參數，我們遂一列出如下，如圖 4.3 所示：

I. 一個旗標用來指示音節位於句首或不是。

II. 一個旗標用來指示音節位於句尾或不是。

III. 前一音節最後一段的 log(F0) 平均值、log(F0) 斜率和 log 能量平均值，共 3 個 feature。

IV. 本身音節三段各自的 log(F0) 平均值、log(F0) 斜率和 log 能量平均值，共 9 個

feature。

V. 後一音節最前面一段的 log(F0) 平均值、log(F0) 斜率和 log 能量平均值，共 3 個 feature。

VI. 音節間的停頓長度(pause duration)，分別與前一音節的停頓長度，及與後一音節之間的停頓長度，共 2 個 feature。

VII. 本身音節長度(syllable duration)，1 個 feature。

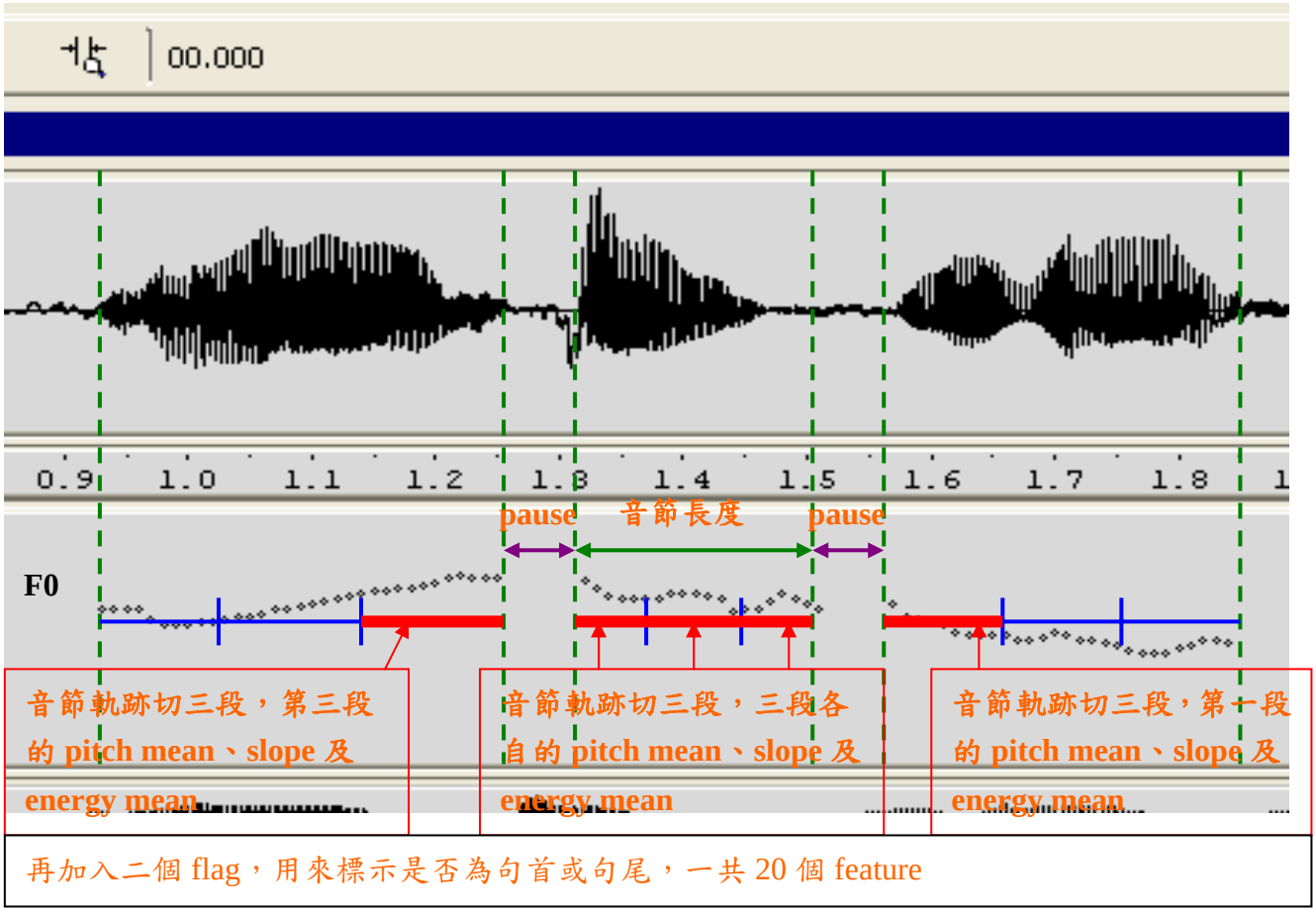

圖 4.3 前後音節相關特徵參數抽取示意圖

雖然我們音節的切割位置準確度相當高，但是音節間的停頓(short pause)不容易被切割出來，只有較長的停頓，會被切出來，如圖 4.4 所示，所以我們用音節間的基頻斷開的長度來做為音節間連音的依據，可以更貼切描述音高軌跡影響程度；圖 4.4 中可看出

音節前一小段，可看到音節能量較小，通常是音節聲音的開始，或是無聲子音，而這類的情況，通常都不會有基頻，這就表現在基頻斷開的長度。

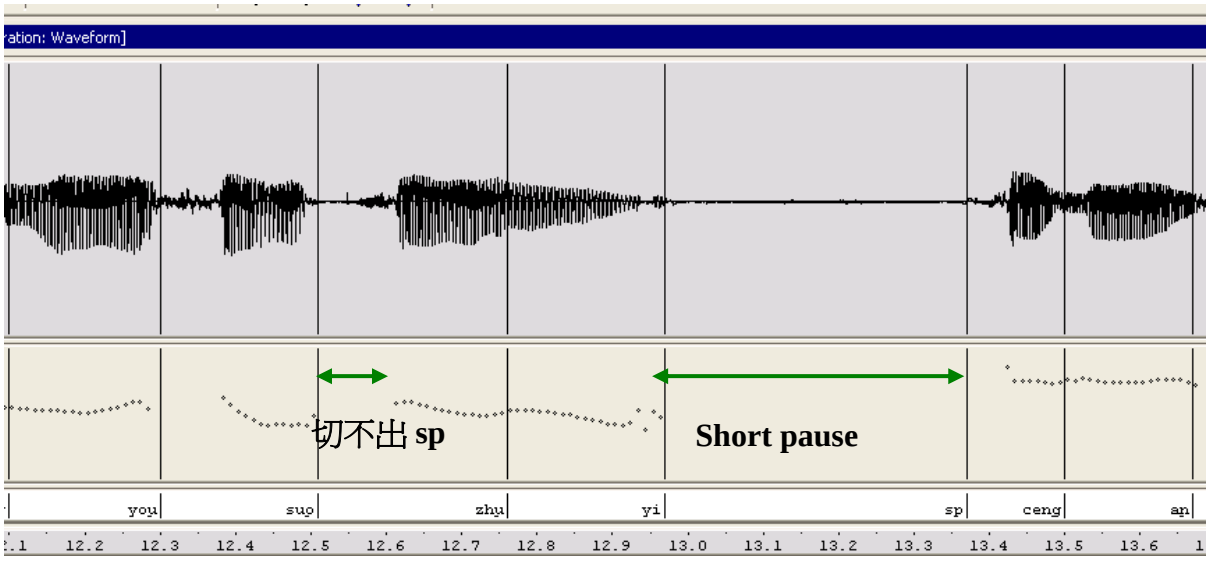

圖 4.4　音節的切割位置

我們將音節的音高軌跡，平均切成三段，然後求取每一段的平均音高、音高的斜率和平均能量，由於音節的聲母包括無聲(unvoiced)子音，所以會有前段沒有基頻的情況，或者音節的切割位置，會包涵到音節間的停頓(short pause)，所以就必須對音高軌跡找到開始與結束位置，如圖 4.5 所示；另一種情況，音節的切割位置會有些許誤差的情況發生時，如圖 4.5 所示，會包涵到前一音節的尾巴，而存在音節內有二段連續音高軌跡，所以我們會取最長的那一段來當做我們要的音高軌跡；同樣的，我們對於音節長度，也做了修改，改以求取音節內的音高軌跡長度。

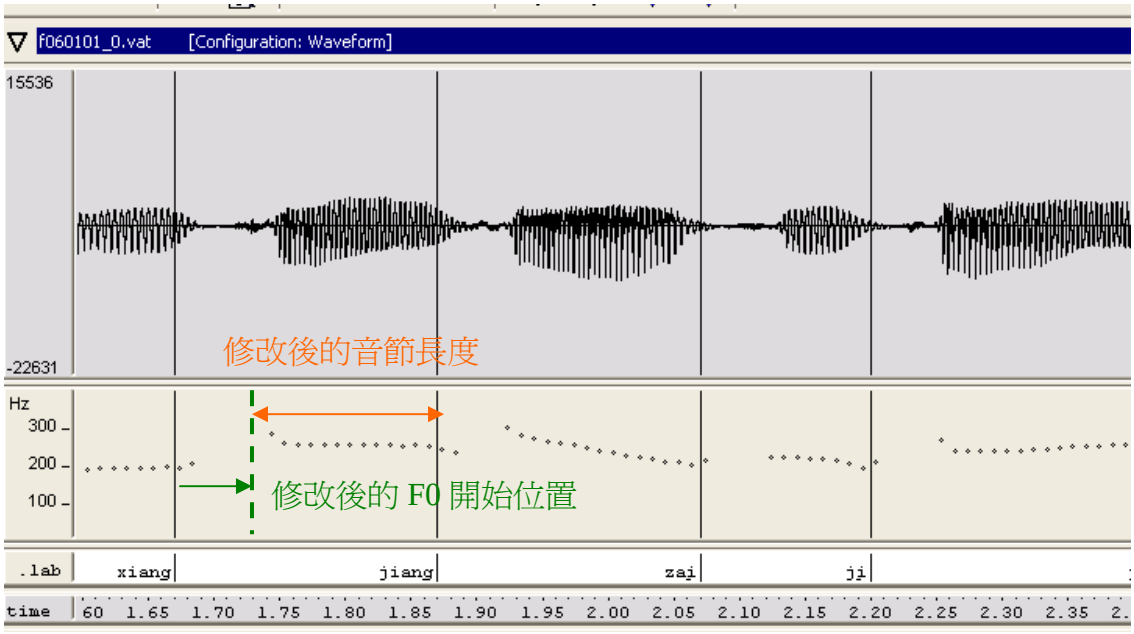

圖 4.5　定位音節音高軌跡開始與結束位置

在計算音節的音高平均值與斜率時，會遇到基頻有倍頻(double pitch)或半頻(half pitch)的情況，如圖 4.6 所示，為了不讓這些情況影響到統計特性，我們以音節的平均音高來取代這些基頻，這種現象較常發生在句子或音節剛開始或結束，聲音較不穩定的地方。

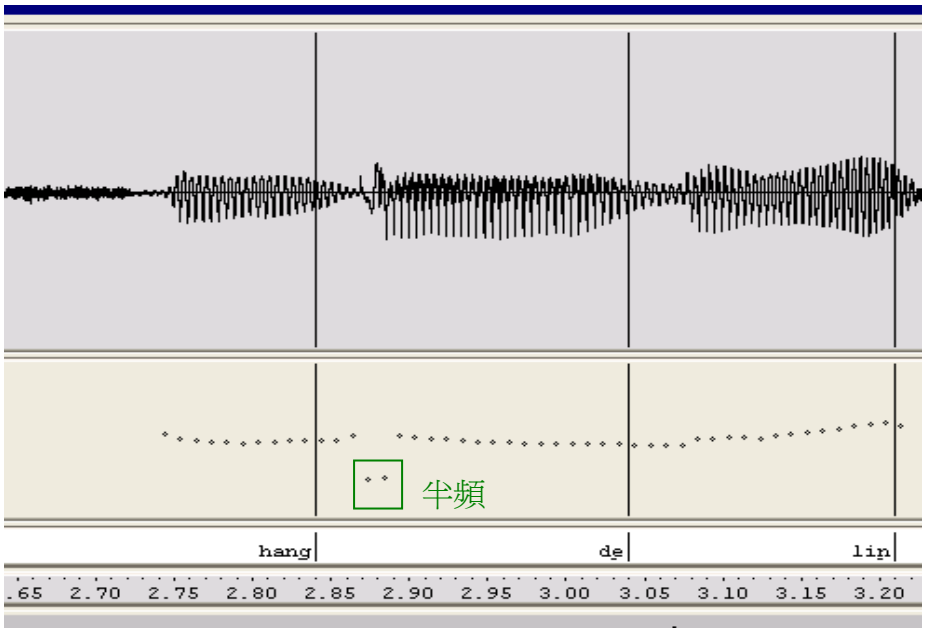

圖 4.6　發生半頻的情況

4.3 MLP 聲調辨認器

抽完聲調的特徵參數後，我們使用的聲調辨認器，是利用 MLP (multi-layer perception)類神經網路來當辨認器，其架構如圖 4.7，共有三層；第一層為輸入層(input layer)，輸入層代入我們抽取的 20 維特徵參數，第二層為隱藏層(hidden layer)，我們將隱藏層設定為 100 神經元，第三層為輸出層(output layer)，輸出 5 個 tone 的機率。

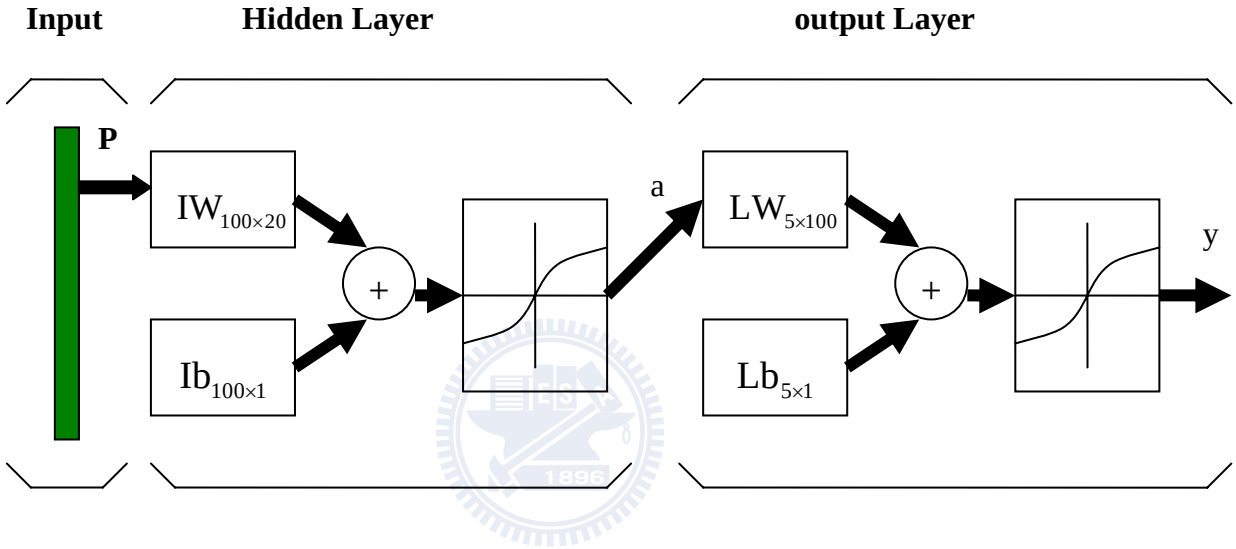

圖 4.7　多層前饋網路

在類神經網路裡神經元可使用任何可微分的轉移函數，我們選用 tansig 轉移函數來當神經元，tansig 是一個雙彎曲轉移函數，式子如(4.1)所示，其特性可將無限範圍的輸入 n → ±∞，壓縮成有限範圍的輸出 a → ±1，如圖 4.8 所示。

$$\text{tansig}(n) = \frac{2}{1+\exp(-2n)} - 1 \qquad (4.1)$$

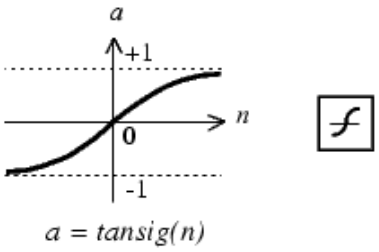

圖 4.8 Tan-Sigmoid 轉移函數

MLP 係為一正回饋 (feed forward) 網路， 每一個第 m+1 層的神經元的輸出，都是由第 m 層非線性函數輸出的加權總和，每個基本神經元加入一個適當的權重值來加權，如式子(4.2)代表 hidden layer 的輸出，式子(4.3)代表 output layer 的輸出。

$$a(j) = \text{tansig}(\sum_{i=1}^{20} iw(j,i)\,p(i) + ib(j)),\ \text{for}\ j = 1,2,...,100 \tag{4.2}$$

$$y(k) = \text{tansig}(\sum_{j=1}^{100} lw(k,j)\,a(j) + lb(k)),\ \text{for}\ k = 1,2,...,5 \tag{4.3}$$

訓練演算法是 back propagation algorithm，我們選用 MATLAB 裡提供的 trainrp 函數來訓練，這個函數是使用 Rporp(Resilient back propagation)演算法，這個函數與標準最陡坡降演算法相比，是可以更快的達到收斂，由於 Rporp 演算法不需要儲存每個權重值和偏壓值的更新值，所以對記憶體的需求量不大，訓練收斂條件為疊代次數 800 或 mse (mean square error)= 0.01，訓練結果如圖 4.9 所示。

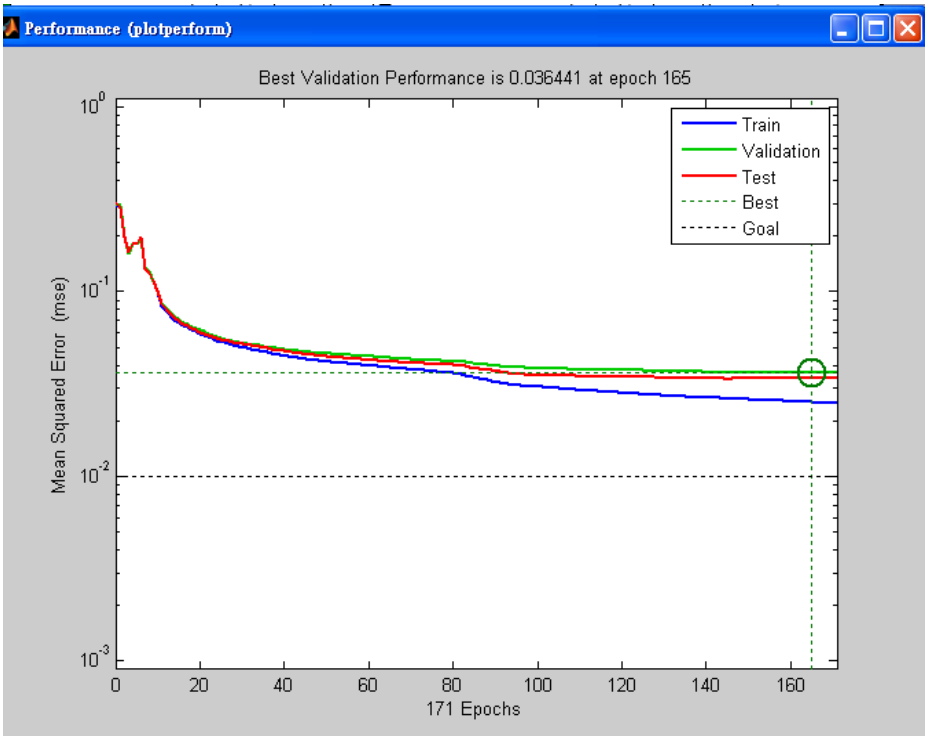

圖 4.9 訓練結果

4.4 MLP 聲調辨認結果

我們訓練的語料有兩個，分別來自 TCC300 及含關鍵詞之語句。TCC300 我們選用跟含關鍵詞之語料一樣的短句句型，是由國立台灣大學 NTU 所錄制朗讀短句的語料，如表 4.1 是其數量的統計表，表 4.2 是含有關鍵詞語句的數量統計表，以下就這兩種語料來源，來訓練 MLP 聲調辨認器。

表 4.1 NTU 語料資料統計表

學校名稱	文章屬性		語者總數		總音節數	檔案總數
台灣大學	短文	訓練	男	25	23326	2948
			女	25		
			總數	50		
		測試 1	男	5	2593	322
			女	5		
			總數	10		
keyword	keyword	測試 2	男	2	998	220
			女	2		
			總數	4		

表 4.2　含關鍵詞之語料資料統計表

學校名稱	文章屬性		語者總數		總音節數	檔案總數
keyword	keyword	訓練	男	8	9032	2000
			女	8		
			總數	16		
		測試	男	2	998	220
			女	2		
			總數	4		

接下來我們將呈現，由兩種語料，所訓練 MLP 聲調辨認的辨認率，與測試 MLP 聲調辨認的辨認率。

表 4.3 是使用 NTU 語料，所訓練出來的辨認率，平均辨認率是 80.08%，表 4.4 是使用表 4.3 所訓練出來的 MLP 聲調辨認器，來辨認同樣是 NTU 語料的測試辨認率, 平均辨認率是 74.67%；表 4.5 則是使用含有關鍵詞之語句來當測試語料，所呈現出來的辨認率，會變得很差，平均只有 65.88%的辨認率。

表 4.3　NTU訓練語料 Tone辨認統計表

Ans\Rec	tone 1	tone 2	tone 3	tone 4	tone 5	total
tone 1	0.828	0.072	0.021	0.077	0.003	5471
tone 2	0.070	0.813	0.073	0.034	0.010	5243
tone 3	0.041	0.198	0.655	0.083	0.022	4535
tone 4	0.055	0.032	0.039	0.867	0.008	7006
tone 5	0.039	0.171	0.336	0.214	0.239	1070
total syllable count						23325
average Rec. rate						**80.08%**

表 4.4　NTU 測試語料1 Tone辨認統計表

Ans\Rec	tone 1	tone 2	tone 3	tone 4	tone 5	total
tone 1	0.745	0.122	0.036	0.090	0.007	589
tone 2	0.075	0.796	0.080	0.040	0.009	574
tone 3	0.035	0.244	0.611	0.089	0.020	537
tone 4	0.097	0.035	0.039	0.815	0.015	751
tone 5	0.056	0.204	0.246	0.183	0.310	142
total syllable count						2593
average Rec. rate						**74.67%**

表 4.5　NTU 測試語料2 Tone辨認統計表

Ans\Rec	tone 1	tone 2	tone 3	tone 4	tone 5	total
tone 1	0.366	0.555	0.021	0.046	0.012	481
tone 2	0.028	0.900	0.056	0.017	0.000	180
tone 3	0.000	0.261	0.638	0.101	0.000	69
tone 4	0.027	0.059	0.160	0.730	0.023	256
tone 5	0.000	0.250	0.750	0.000	0.000	12
total syllable count						998
average Rec. rate						**65.88%**

表 4.6 是使用含有關鍵詞之語料，所訓練出來的辨認率，平均辨認率是 91.91%，表 4.7 是使用表 4.6 所訓練出來的 MLP 聲調辨認器，來辨認含有關鍵詞之語料測試辨認率，平均辨認率是 91.34%%。

表 4.6　含關鍵詞之語句訓練語料 Tone辨認統計表

Ans\Rec	tone 1	tone 2	tone 3	tone 4	tone 5	total
tone 1	0.960	0.013	0.002	0.024	0.001	4242
tone 2	0.087	0.858	0.027	0.024	0.003	1432
tone 3	0.042	0.126	0.677	0.146	0.009	740
tone 4	0.040	0.011	0.008	0.939	0.001	2494
tone 5	0.177	0.024	0.040	0.056	0.702	124
total syllable count						9032
average Rec. rate						**91.91%**

表 4.7　含關鍵詞之語句測試語料辨認統計表

Ans\Rec	tone 1	tone 2	tone 3	tone 4	tone 5	total
tone 1	0.946	0.033	0.002	0.017	0.002	481
tone 2	0.094	0.872	0.011	0.022	0.000	180
tone 3	0.058	0.116	0.696	0.130	0.000	69
tone 4	0.027	0.023	0.023	0.926	0.000	256
tone 5	0.000	0.000	0.000	0.167	0.833	12
total syllable count						998
average Rec. rate						**91.34%**

由以上的 tone 辨認結果，表 4.5 與表 4.7 比較可看出，使用含關鍵詞之語句訓練出來的 MLP 聲調辨認器，得到較好的辨認結果，這可由一般人在唸關鍵詞時，會特別唸清楚或強調，而略知一二；表 4.5 若將 NTU 句型訓練出來的 MLP 聲調辨認器，拿來測試含關鍵詞之語句的語料，其辨認率會明顯的下降，由此可看出，人們在唸 keyword 時的語氣不同於一般朗讀文章的唸法，所以會有訓練與測試不匹配(not match)的情況，而造成辨認率大幅度下降。所以我們將會拿表 4.6 的這一組訓練結果的 MLP 聲調辨認器，來當做我們系統第二階段，所要加入聲調辨認的辨認器。

第五章 實驗結果與分析

在論文中，我們要使用關鍵詞辨認系統建立一套新竹科學園的導覽或查詢系統。我們所使用的關鍵詞組為新竹科學園區裡的公司名，共 341 個公司名。但人們也常使用公司的別名，如”台灣積體電路”常被簡稱為”台積電”，若含其別名共有 1074 個。

在這組新竹科學園區公司名關鍵詞組有許多混淆詞組，如：“眾晶’’公司與“永進”公司或“思源科技”公司與“致遠科技”公司。這些混淆詞組在加入聲調資訊後較能分別。

我們設計測試關鍵詞辨認系統所需的句子結構，測試的句子都屬詢問或命令式的短句，填充字串的部份可為任意字的組合，如下所示：

例 1: 我要去興能科技。 關鍵詞: 興能科技，前面有接填充字串，後面沒接填充字串。

例 2: 興能科技怎麼走。關鍵詞: 興能科技，前面沒接填充字串，後面有接填充字串。

例 3：帶我去興能科技好嗎。關鍵詞: 興能科技，前後都有接填充字串。

例 4：興能科技。關鍵詞: 興能科技，前後都不接填充字串。

測試語句共 769 音檔，語者 10 男 10 女共 20 名，表 5.1 為基本系統的辨認率。

表 5.1 基本系統的辨認率

	Rec. rate
top 1	93.5
top 2	95.84
top 3	96.23
top 4	96.62
top 5	96.75
top 6	96.75
top 7	96.88
top 8	96.88
top 9	96.88
top 10	96.88

為了提升辨認率，在本論文第二章節，曾對系統辨認錯誤，做一分析，我們先針對辨認錯誤音節長度呈現過長或過短的情況，預計加入狀態長度模型(State duration model)做改善，狀態模型以 Gamma 分佈表示。當 Viterbi beam search 的路徑，發生狀態轉換(state transition)時，加入的狀態長度模型分數，如式子(5.1)所示，若狀態停留的長度，與我們的狀態模型差異很大時，其 Gamma 的機率就會較低，所以就會被扣較多的分數，Viterbi beam search 會將分數較低的路徑砍掉，就算沒有被砍掉，最後分數也不的進到 Top-10 的排名。

$$\text{path\_score} = \log(\text{likelihood}) + \text{dur\_wt} \times \log(\Gamma(x;\alpha,\beta)) \tag{5.1}$$

其中 dur_wt 是我們調整狀態長度模型分數的權重，值為 2.0。

表 5.2 呈現出加入狀態長度模型的辨認率，其中第一名由原本的 93.5%上升到 94.54%，錯誤減少率(error reduction rate) 16%。

表 5.2 系統加入了狀態長度模型的辨認率

	Rec. rate
top 1	94.54
top 2	96.23
top 3	96.75
top 4	96.88
top 5	96.88
top 6	96.88
top 7	96.88
top 8	96.88
top 9	96.88
top 10	96.88

我們在系統加入狀態長度模型後，辨認率獲得提升，接下來，我們將針對音節切割位置幾乎一致的錯誤情況，加入了第二階段之聲調辨認，利用聲調的分數，來提高正確答案的鑑別度。加入的方法是當系統最後辨認出 Top-10 的結果後，我們針對 Top-10 進行聲調辨認，將聲調辨認的分數與最後的 likelihood 值相加，相加之後的分數做重新排

名，其數學式子表示在式子(5.2)。

$$\text{score} = \log(\text{FL\_likelihood}) + \text{Tone\_wt} \times \sum_{i=0}^{n-1} (\log(t_{i,j}) - \log(t_{i,\max})) \qquad (5.2)$$

Tone_wt 是我們調整聲調分數的權重，值為 0.1，n 是辨認出關鍵字長度，$t_{i,j}$ 第 i 個音節對應到的聲調分數，$t_{i,\max}$ 第 i 個音節中最大的聲調分數。聲調辨認的分數，我們是以音節原本聲調的分數，減掉音節聲調辨認分數最高者，若是辨認到正確所加的分數為零，若是辨認到錯的分數，就會被加上一個負的值，如此就可拉開正確答案與錯誤答案的距離。

表 5.3 呈現出加入聲調辨認後的辨認率，其中第一名由原本的 94.54%上升到 95.32%，錯誤減少率(error reduction rate) 14.3%。

表 5.3 加入聲調辨認後系統的辨認率

	Rec. rate
top 1	95.32
top 2	96.1
top 3	96.62
top 4	96.75
top 5	96.88
top 6	96.88
top 7	96.88
top 8	96.88
top 9	96.88
top 10	96.88

第六章結論與未來展望

6.1 結論

本論文使用的 MLP 聲調辨認器，其聲調辨認率在關鍵詞上已很高了，達九成以上，所以可信度相當高，用以幫助關鍵詞辨認，辨認率從原本的 94.54% 提升到 95.32%，錯誤更正率(error reduction rate) 14.3%，由結果可看出，在原本關鍵詞辨認率已如此高下，仍然獲得提升，證明聲調辨認是有用的，且我們所加入的聲調辨認器，其本身運算量相當小，只需存放 MLP 聲調辨認器的參數與權重，所用到的記憶體 8820bytes，對於 pc base 少則好幾 Gigabytes 的記憶體來說，已算很小了，而且我們只對前 10 名做聲調辨認，所以運算的負擔也很小，整體來說所用到的資源少，辨認率又能得到提升，對於應用上，算是實用的。

除了加入聲調辨認外，在原本未加任何輔助模型的辨認率為 93.5%，當加入狀態長度模型，辨認率提升到 94.54%，錯誤更正率(error reduction rate) 16%。另外我們在系統的修改上，加入了基頻軌跡資訊的抽取；也將音節的切割位置取出，所以說語音的特徵資訊，幾乎在本系統都可取得，對於日後，若要加入其它模型，亦會顯得相當方便且快速。

6.2 未來展望

本論文的 MLP 聲調辨認器，是對單一音節做聲調辨認，未來可嘗試使用 tone pair 來做聲調辨認，由二個音節的資訊，可以學到聲調組合的規律，對於聲調辨認上，或許會更好，且更強健。

中文關鍵字辨認系統，我們已修改了更完備，加上了音高軌跡與音節切割位置資訊，對於未來要加入其它的模型，將更為方便，比如加入韻律模型(prosodic model)[10]，韻

律模型，相對於音節的關係有音節本身模型(intra-syllable)及音節之間(inter-syllable)的影響因素；以音節本身模型來說，有音節長度模型(syllable duration model)、音節音高模型(syllable pitch model)及音節能量模型(syllable energy model)；音節之間的影響因素來說，有能量低點(energy dip)、音節間的停頓長度(pause duration)、音節間的基頻跳躍(pitch jump)及音節之間長度差值(syllable duration difference)。

參考文獻

[1] David Talkin, “A Robust Algorithm for Pitch Tracking”.

[2] Yih-Ru Wang , Sin-Horng Chen ,“Tone Recognition of Continuous Mandarin Speech Based on Neural Networks,”IEEE Trans. On Speech and Audio Processing , Vol .3,No2,pp.146-150.March 1995.

[3] 王小川，“語音訊號處理”，全華科技圖書，中華民國九十三年三月。

[4] S.-H. Chen and Y.-R. Wang, “Vector quantization of pitch information in Mandarin speech,” IEEE Transactions on Communications, vol. 38, no. 9, pp. 1317-1320, September 1990.

[5] W.B. Kleijn and K.K. Paliwal, “A robust algorithm for pitch tracking”, Elsevier science B.V, 1995

[6] “HTK Web-Site”, http://htk.eng.cam.ac.uk. Accessed 2009

[7] Mandarin microphone speech corpus-TCC300 , http://www.aclclp.org.tw/use_mat.php#tcc300edu

[8] S.-H. Chen, W.-H. Lai, and Y.-R. Wang, “A new duration modeling approach for Mandarin speech,” IEEE Transactions on Audio, Speech and Language Processing, vol. 11, no. 4, pp. 308–320, July 2003.

[9] Wavesurfer Homepage：http：//www.speech.hth.se/wavesurfer/

[10] 施宏廣，“中文單詞之韻律模擬與其應用”，國立交通大學碩士論文，民國九十七年八月。

附錄一：子音 100 類

子音編號(100 類)	子音拼音(100 類)	注音
1	zh_NULL	ㄓ_NULL
2	ch_NULL	ㄔ_NULL
3	sh_NULL	ㄕ_NULL
4	r_NULL	ㄖ_NULL
5	z_NULL	ㄗ_NULL
6	c_NULL	ㄘ_NULL
7	s_NULL	ㄙ_NULL
8	zh_a	ㄓ_a
9	ch_a	ㄔ_a
10	sh_a	ㄕ_a
11	z_a	ㄗ_a
12	c_a	ㄘ_a
13	s_a	ㄙ_a
14	g_a	ㄍ_a
15	k_a	ㄎ_a
16	h_a	ㄏ_a
17	d_a	ㄉ_a
18	t_a	ㄊ_a
19	n_a	ㄋ_a
20	l_a	ㄌ_a
21	b_a	ㄅ_a
22	p_a	ㄆ_a
23	m_a	ㄇ_a
24	f_a	ㄈ_a
25	l_o	ㄌ_o
26	b_o	ㄅ_o
27	p_o	ㄆ_o
28	m_o	ㄇ_o
29	f_o	ㄈ_o
30	zh_e	ㄓ_e
31	ch_e	ㄔ_e
32	sh_e	ㄕ_e
33	r_e	ㄖ_e
34	z_e	ㄗ_e

35	c_e	ㄘ_e
36	s_e	ㄙ_e
37	g_e	ㄍ_e
38	k_e	ㄎ_e
39	h_e	ㄏ_e
40	d_e	ㄉ_e
41	t_e	ㄊ_e
42	n_e	ㄋ_e
43	l_e	ㄌ_e
44	b_e	ㄅ_e
45	p_e	ㄆ_e
46	m_e	ㄇ_e
47	f_e	ㄈ_e
48	r_a	ㄖ_a
49	zh_o	ㄓ_o
50	ch_o	ㄔ_o
51	sh_o	ㄕ_o
52	r_o	ㄖ_o
53	z_o	ㄗ_o
54	c_o	ㄘ_o
55	s_o	ㄙ_o
56	g_o	ㄍ_o
57	k_o	ㄎ_o
58	h_o	ㄏ_o
59	d_o	ㄉ_o
60	t_o	ㄊ_o
61	n_o	ㄋ_o
62	j_y	ㄐ_y
63	q_y	ㄑ_y
64	x_y	ㄒ_y
65	d_y	ㄉ_y
66	t_y	ㄊ_y
67	n_y	ㄋ_y
68	l_y	ㄌ_y
69	b_y	ㄅ_y
70	p_y	ㄆ_y
71	m_y	ㄇ_y

72	zh_w	ㄓ_w
73	ch_w	ㄔ_w
74	sh_w	ㄕ_w
75	r_w	ㄖ_w
76	z_w	ㄗ_w
77	c_w	ㄘ_w
78	s_w	ㄙ_w
79	g_w	ㄍ_w
80	k_w	ㄎ_w
81	h_w	ㄏ_w
82	d_w	ㄉ_w
83	t_w	ㄊ_w
84	n_w	ㄋ_w
85	l_w	ㄌ_w
86	b_w	ㄅ_w
87	p_w	ㄆ_w
88	m_w	ㄇ_w
89	f_w	ㄈ_w
90	j_yu	ㄐ_yu
91	q_yu	ㄑ_yu
92	x_yu	ㄒ_yu
93	n_yu	ㄋ_yu
94	l_yu	ㄌ_yu
95	INULL_a	Φ2
96	INULL_o	Φ3
97	INULL_e	Φ4
98	INULL_y	Φ5
99	INULL_w	Φ6
100	INULL_yu	Φ7

附錄二：母音 40 類

母音編號	母音符號(40 類)	注音
1	FNULL1	Φ1
2	a	ㄚ
3	o	ㄛ
4	e	ㄜ
5	eh	ㄝ
6	ai	ㄞ
7	ei	ㄟ
8	ao	ㄠ
9	ou	ㄡ
10	an	ㄢ
11	en	ㄣ
12	ang	ㄤ
13	eng	ㄥ
14	yi	ㄧ
15	wu	ㄨ
16	yu	ㄩ
17	ya	ㄧㄚ
18	ye	ㄧㄝ
19	yai	ㄧㄞ
20	yao	ㄧㄠ
21	you	ㄧㄡ
22	yan	ㄧㄢ
23	yin	ㄧㄣ
24	yang	ㄧㄤ
25	ying	ㄧㄥ
26	wa	ㄨㄚ
27	wo	ㄨㄛ
28	wai	ㄨㄞ
29	wei	ㄨㄟ
30	wan	ㄨㄢ
31	wen	ㄨㄣ
32	wang	ㄨㄤ
33	weng	ㄨㄥ
34	yue	ㄩㄝ

35	yuan	ㄩㄢ
36	yun	ㄩㄣ
37	yung	ㄩㄥ
38	er	ㄦ
39	yo	ㄧㄛ
40	FNULL2	Φ2