國 立 交 通 大 學

電信工程學系

碩士論文

基於類神經網路之中文語音屬性偵測器

A Neural Network based Mandarin Speech Attribute Detection

研 究 生：張友駿

指導教授：王逸如　博士

中華民國九十六年八月

基於類神經網路之中文語音屬性偵測器

A Neural Network based Mandarin Speech Attribute Detection

研 究 生：張友駿　　Student：Yio -Jun Zhang

指導教授：王逸如　　Advisor：Dr. Yi-Ru Wang

國 立 交 通 大 學

電 信 工 程 學 系

碩 士 論 文

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

August 2007
Hsinchu, Taiwan

中華民國九十六年八月

基於類神經網路之中文語音屬性偵測器

研究生：張友駿　　指導教授：王逸如博士

國立交通大學電信工程學系碩士班

中文摘要

新世代的自動語音辨識技術架構是一個以知識為基礎（knowledge-based），加上資料驅動（data-driven）的模式，其前端為語音屬性與事件偵測器群，藉由抽取不同的語音特徵參數去偵測某一時段中語音的屬性及事件，尋找任何可以提供語音辨識的線索，提供給後級作語音事件及知識整合後，作證據確認及決策，以其能夠突破目前語音辨識的能力與技術。

本論文基於此概念，首先由於中文語料庫並無精確的音素切割位置，因此我們從中文音節的切割位置起始對語料庫作自動切割以求得音素的初始切割位置，接著以 Segmental Kmeans Segmentation Algorithm 自動調整音素的切割位置，並以此切割位置製作中文發音方法偵測器。首先訓練線性的混合高斯模型偵測器，接著訓練非線性的多層感知機模型偵測器，之後以 segment-based 的概念在偵測過程中加入狀態轉移機率(State transition probability)來對於中文發音方法進行偵測實驗，最後引入信任度量測(Confidence measure)來對偵測結果可靠的程度作量化的評比，提供語音資訊傳給後級辨識器當參考依據。最後再對各架構語音屬性偵測器以及信任度量測作效能與錯誤分析。

A Neural Network based Mandarin Speech Attribute Detection

Student：Yio-Jun Zhang Advisor：Dr. Yi-Ru Wang

Institute of Communication Engineering

National Chiao Tung University

Abstract

Next generation ASR system is a knowledge-based and data-driven paradigm. It’s front-end is the bank of speech attribute and event detectors, and it’s function is to detect the speech attributes and events in the speech signal. By organizing the outputs of front-end and knowledge, it would be sent to next stage to make evidence verified and decision. It would be expected to exceed the current state-of-the-art HMM-based ASR.

Based on the concept, firstly, because there is no manual labeling for Mandarin corpus ,we start with syllable labeling and then forced-align the corpus to get initial phone labeling. Then we use Segmental Kmeans Segmentation Algorithm to automatically refine phone labeling and use this phone labeling to train Mandarin attribute detector. First, we train linear GMM based detector and then train nonlinear MLP based detector. Then based on concept of segment-based ,we add state transition probability to MLP based detector to examine Mandarin speech detection. Secondly, we use confidence measure to evaluate the result of attribute detection, providing confident speech information to recognizer for reference. Finally, we would make error analysis and performance evaluation of different Mandarin speech attribute detectors and confidence measure.

誌謝

這篇論文的完成，首先感謝指導教授王逸如老師以及陳信宏老師，這兩年期間在兩位老師的身旁，學習作研究的方法及態度。跟著老師出席研討會，更讓我增廣見聞。

這兩年在實驗室的期間，特別要感謝振宇學長以及見徨學長，給予我研究上的建議、將研究成果完整的交接給我繼續接續的研究、教我使用工作站上的軟體、偶爾還找振宇學長討論軍武跟歷史...等，對我而言等於是"助教"與"知己"。感謝認真負責的智合學長，跟我成為親戚的阿德學長，斯文具有藝術氣質的希群學長，有許多神奇經歷的輝哥學長，總是知道哪裡好吃好玩的巴金叔叔，還有上一屆畢業的其他學長們。

感謝這兩年來，一起為研究打拼的戰友們，地上卡丁車速最快的獻文大大，籃球很強的小傅，讓老師滿意的宏宇大大，貌似忠厚的銘彥，Very good 的啟風，神圖的大師柏蒼，不折不扣的神手小鄧，長的像暴龍的胤賢，有各位兩年的陪伴，一起同甘共苦完成碩士的學業，讓我除了研究的領域之外這兩年不曾留白。

最後感謝辛苦的父母及兄長，支持我完成碩士的學業。

目錄

表目錄

圖目錄

第一章　緒論

1.1 研究動機

回顧現今的語音辨識技術，由早期的學者研究聲學與語言學的規則建立一個以規則為基礎的（rule-based）語音辨識系統，此種辨識系統可以說是以知識驅動（knowledge-driven）的解決方式，但此種系統無法應付複雜的語音變化，因此語音辨識技術繼續進展至以資料驅動（data-driven）的模式，機器由資料中學習，再進展至大詞彙的連續語音辨識（large vocabulary continuous speech recognition，LVCSR）技術，其所依賴的就是大量的語音資料與語言資料。然而這些方法雖然大大改進機器語音辨識的能力，但漸漸的可以發覺到與人類辨識語音的能力相比，仍然有一段不小的差距。因此為了使語音辨識技術有所突破，近年來國際上不斷有學者主張，應該回頭將語音與語言的知識結合進現今的辨識技術，建立一個以知識為基礎（knowledge-based）加上資料驅動的（data-driven）模式，開放測試平台，共享一個合作的設計與評量機制（如圖 1.1 所示），邁向下一代自動語音辨認技術。

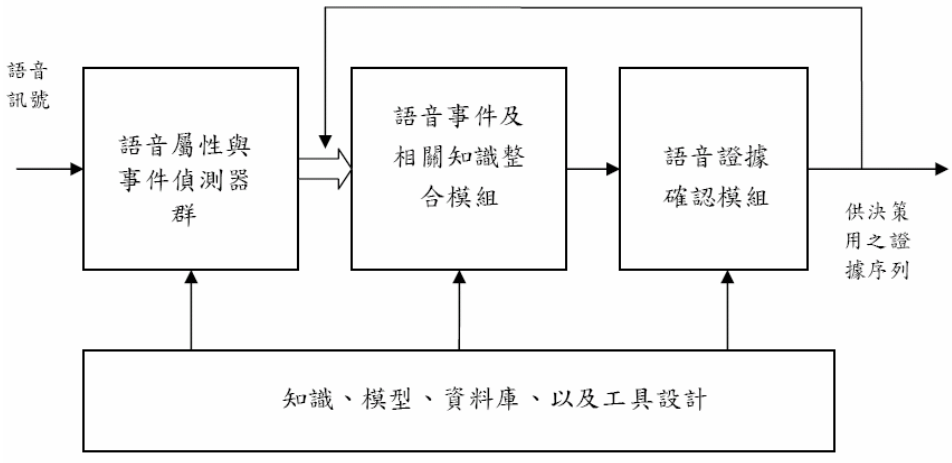


圖 1.1：新世代自動語音辨識技術架構圖

1.2 研究方向

由於新世代的語音辨識系統架構前端為一個偵測器群，而語音屬性偵測器-如發音方法（manner of articulation）及發音位置（place of articulation）等偵測器，為最基本的偵測器，它們可以提供語音特性之資訊，因此我們將致力於製作語音屬性偵測器。而由於國語 TCC300 語料庫並無人工切割位置，而製作偵測器首先需要的是精確的音素切割位置，因此我們首先將先對中文語料庫作自動切割(forced alignment)以取得初始的音素切割位置[2]，接著運用 Segmental Kmeans Segmentation Algorithm 自動調整切割位置，並以此切割位置製作語音屬性偵測器，並且對所製作出來的語音屬性偵測器作效能及錯誤分析，最後引入信任度量測(Confidence measure)來對於屬性偵測器的結果作可靠程度的評比。

1.3 章節概要

本論文共分為五章：

第一章　緒論：介紹本論文之研究動機與研究方向。

第二章　以音框為基礎的中文發音方法貝氏偵測器之初步建立

第三章　進階中文語音屬性偵測器之建立

第四章　中文發音方法偵測器的效能的分析與討論

第五章　結論與未來展望

第二章　以音框為基礎(frame-based)的中文語音屬性貝氏偵測器之初步建立

本章內容首先介紹使用中文語料 TCC300 來製作中文語音發音方法的屬性偵測器，製作語音屬性偵測器的第一步，就是需要有語音資料庫及詳細的標示資料，而此標示資料必須標示到語音屬性階層並包含其時間訊息，不像從前以隱藏式馬可夫模型為基礎(HMM-Based)所作的語音辨認研究，其所需之語料庫都只需標示到音節階層即可，也不需要詳細的時間資訊。由於 TCC300 中文語料庫並沒有人工標記的音素切割位置，最基本取得音素標記位置的方法是訓練音素的隱藏式馬可夫模型(HMM)對 TCC300 語料進行強迫切割(force-alignment)，然而 HMM 強迫切割的結果並不是很精準，這可能對於我們以音框為基礎的屬性偵測實驗是一大致命傷，因此我們必須在訓練屬性偵測器之前就取得比較可靠的切割位置來當作訓練偵測器模型的依據，為了取得比較可靠的切割位置，首先對於資料庫做音節標記檔標記錯誤的檢查，排除大部分因人為因素而造成標記錯誤之後，接著由音節切割位置開始，訓練音素的 HMM 模型之後對語料庫進行強迫切割得到初步的切割位置，之後用自動的方式去調整得到比較可靠的切割位置以利於往後的屬性偵測實驗使用。在本論文中使用的語料為中文語料 TCC300 長句的部份(由交通大學以及成功大學所錄製)，訓練語料量約 13.03 個小時，而測試語料量約 1.96 個小時。

2.1 中文音節標記檔的訂正

TCC300 語料庫原始的音節 transcription 檔可能存在某部份人為標記的錯誤，這個現象直接會污染我們訓練的音節模型，使得 HMM 強迫切割的結果變

差，進而造成偵測器模型參雜了不正確的訓練資料影響到偵測器的效能，因此我們構想一個機制如下圖所示，首先能夠自動的找出大部分可能發生錯誤的標記錯誤，接著再用人工去聽音檔確認並且改正錯誤的標記，首先我們先訓練 411 音節的 HMM，接著對於語料做辨認，如果標示答案未出現於音節辨認結果的 Top-N 當中，我們將用人工檢查是否為標示錯誤。在我們所使用的辨認器之辨識率已經很高的情況下，這樣將可以有效的找出語料當中的標示錯誤。

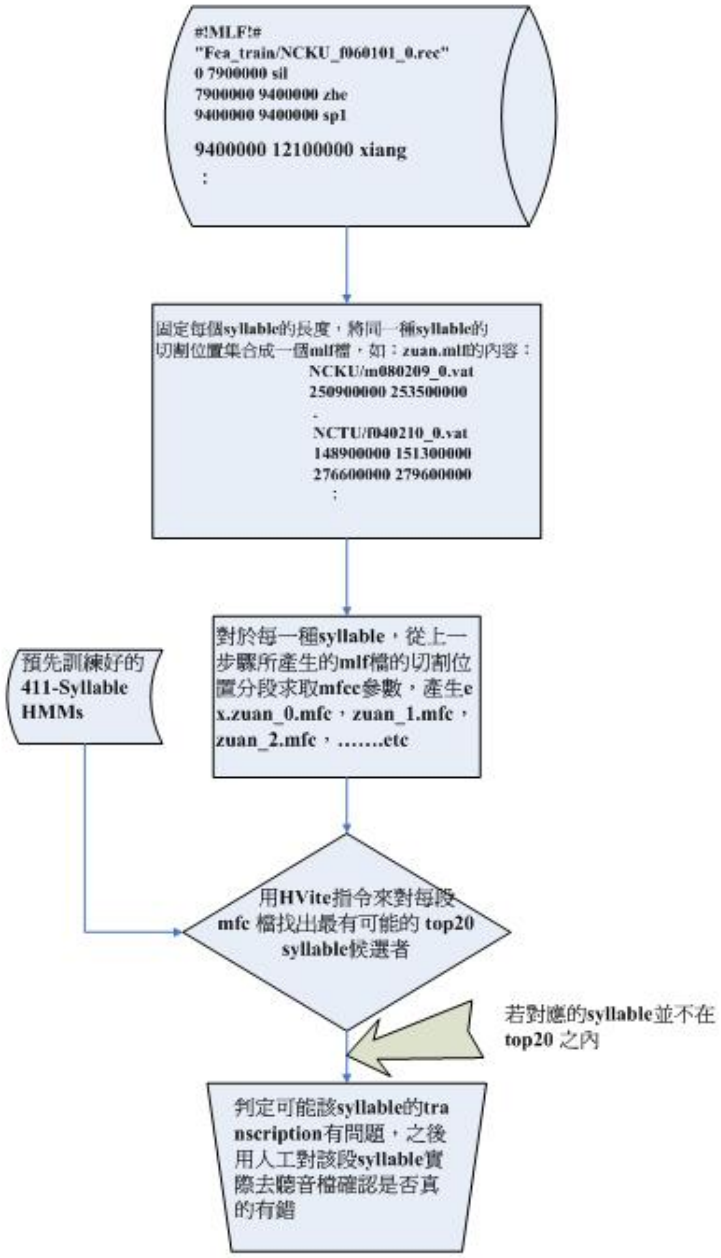


圖 2.1：半自動改正標記檔錯誤流程圖

根據上述的流程，我們將 top 20 自動篩選出來可能有問題的音節用人工去聽音檔對照檢查，確認該 syllable 是否有音節標記錯誤或是其他類型的錯誤，下表是錯誤類別的統計：

單位： syllable　　表 2.1：語料庫音節 top20 篩選結果分布

語料	錯誤分類	單一音節標錯 or 唸錯(包括破音字)	多音節或少音節	一連串音節標記錯誤(在同一檔案中)	單一音節切的稍前或稍後	單一音節唸的不標準或是連音
NCKU_train	約 120000	59 (0.049%)	4 (0.004 %)	204(因為該音檔壞掉) (0.77%)	0	92
NCKU_test	約 12000	13 (0.11 %)	0	0	3 (0.024 %)	39
NCTU_train	約 130000	24 (0.018%)	0	0	0	66
NCTU_test	約 13000	9 (0.069%)	0	0	0	43

將以上統計結果中實際上音節標記錯誤的資料量佔所有語料的比例統計:

表 2.2：音節標記錯誤佔所有語料庫的資料量比例

語料	總音節數	實際上標記錯誤的音節數
TCC300_train	約 250000	287(約 0.1%)
TCC300_test	約 25000	22(約 0.09%)

由上述統計結果可以看出，音節標記錯誤佔所有語料庫資料量中約 0.1%，這些標記的錯誤不但會使訓練語料拿來訓練偵測器模型時由於錯誤的標記位置學習到不正確的資料影響偵測器的效能，同時被拿當來當作偵測實驗當中的參考答案的測試語料中的標記錯誤，更是可能直接使得錯誤率升高的原因之一。

2.2 中文音素切割位置的取得

在前一小節當中我們已經半自動的訂正音節 transcription 檔當中的錯誤，因此接下來我們首先訓練音節的 HMM 模型進行強迫切割[7]取得音節的切割位置，然後以音節切割位置開始訓練狀態數為 3 的音素的 HMM 模型，同時訓練 short pause (sp) 與 silence 模型並且將 non-speech signal(如 breath , noise…等)模型隱藏在 short pause 以及 silence 的 HMM 狀態當中，允許狀態跳躍來切出更合理的非語音段，下圖 2.2 為非語音模型當中的狀態轉移示意圖:

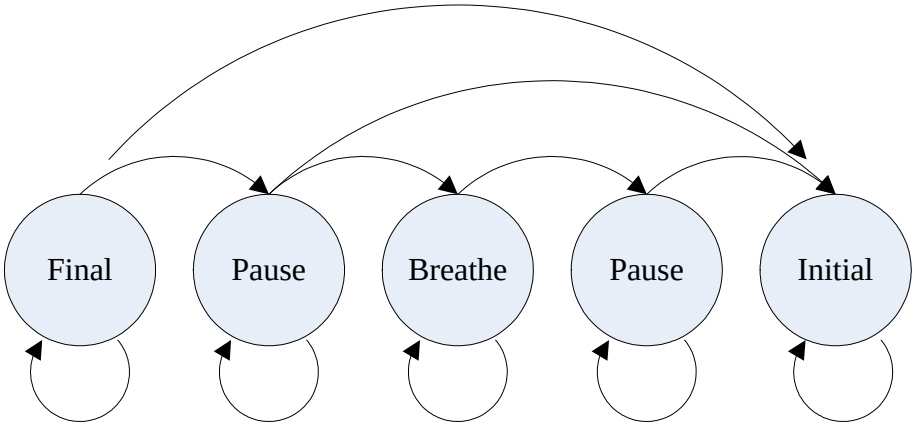


圖 2.2：非語音段模型狀態轉移圖

有了音素的 HMM 模型之後緊接著便對語料庫進行強迫切割，雖然把呼吸聲切割出來能夠切出比較乾淨的 short pause 以及 silence，然而以切割的角度來看 HMM 的切割位置仍舊不夠精準，底下是一個簡單的例子:

語者呼吸聲

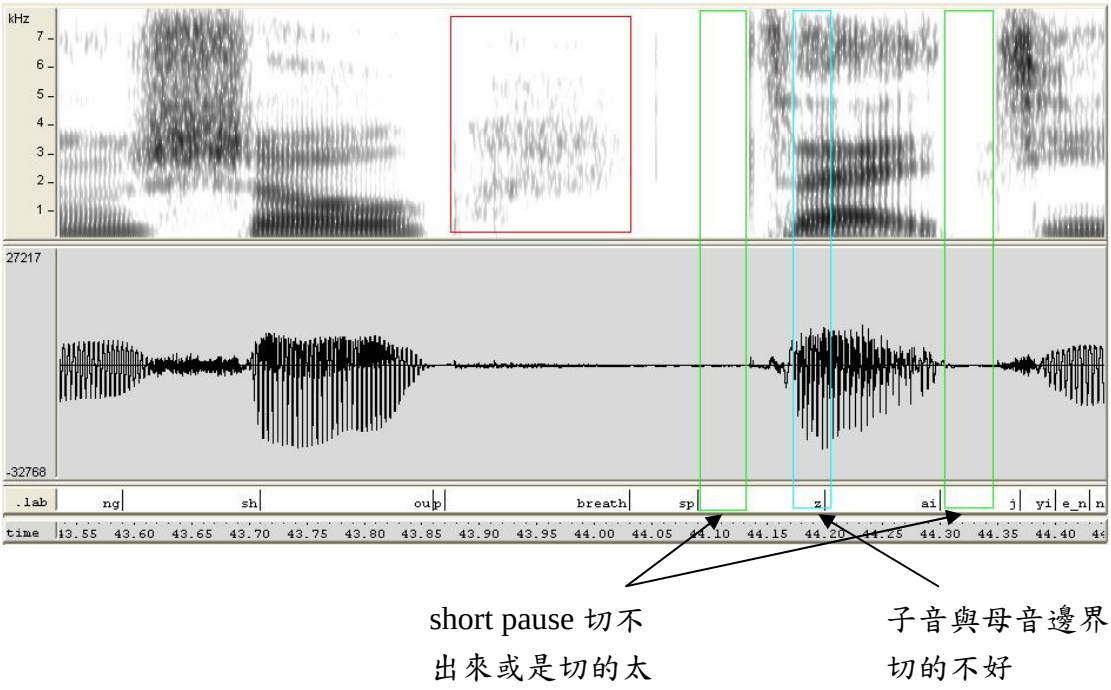


圖 2.3：音素 HMM 強迫切割不準確的例子

我們可以看到 HMM 自動強迫切割的結果對於音素的邊界常常有不小的誤差，尤其是子音之前的短暫 short pause 往往都切不出來或者是切的太短，造成子音的長度普遍過長，同時子音與其後母音的交界切割也不甚理想，以下是 HMM 強迫切割的各發音方法平均音長的統計:

表 2.3：強迫切割各發音方法平均音長

單位:音框 發音方法	HMM 強迫切割 平均音長
Vowel	9.62
affricate	9.62
Liquid	6.69
fricative	10.99
Nasal	6.59
Stop	7.78
Silence	15.74

我們拿音長切的過長最明顯的 Stop 音出來更細部的觀察音素的平均音長狀況，

其中音長特別短的ㄅ、ㄉ、ㄍ三個音的音長做統計:

表 2.4：ㄅㄉㄍ三種音素的平均音長

單位：frame

	平均長度
ㄅ(b)	6.36
ㄉ(d)	5.86
ㄍ(g)	6.85

這三個音的音長實際上大約在 3 個音框長以下，但 HMM 自動切割的平均長度竟然整整多了一倍，代表說以 HMM 強迫切割取得的切割位置確實將子音的長度切的過長，其中又以音長較短的子音特別明顯。

因此底下我們提出以使用局部樣本(local sample)之 Segmental K-means segmentation algorithm[3]的方法來調整音素的切割位置，它是一種廣為人知拿來對於資料分群的 K-means iterative procedure，它能夠藉著 Viterbi algorithm 找到最佳的分段序列，重新將樣本點的資料分類，因此我們將這方法應用來對我們音節之間的 sp 以及子音母音的邊界進行切割位置調整。首先我們固定呼吸聲的切割位置，假設 Observation sequence $O = (o_1 o_2 ... o_N)$ 用來代表由一音節之 final 起始點至下一音節 initial 之終止點間語音信號參數，並且使用 HMM 之音節切割位置並且將之分成 I = 3 段落 (final， short pause， initial)， observation vector o_j 由 13 維 MFCC 參數組成，而 i th ($1 \leqq i \leqq$ I) segment；S_i；的音框訓練一個高斯模型 $\Phi_i = N(\mu_i, \sum_i)$，其中 μ_i 為 mean-vector，$\sum_i$ 為 covariance matrix，這些參數都可由第 i th 段落當中 n_i 個音框求得：

$$\hat{\mu} = \frac{1}{n_i} \sum_{k=1}^{n_i} o_k \tag{2.1}$$

$$\hat{\Sigma}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} (o_k - \hat{\mu}_i)\ (o_k - \hat{\mu}_i)^t \tag{2.2}$$

而在調整音節間 short pause 切割位置的步驟當中，likelihood 方程式可以寫成：

$$\prod_{S} p(o_j \mid \Phi_{S_i}) = \prod \frac{1}{2\pi \mid \hat{\Sigma}_i \mid^{1/2}} \exp\left[-\frac{1}{2}(o_j - \hat{\mu}_i)^t \times \hat{\Sigma}_i^{-1}(o_j - \hat{\mu}_i) \right] \qquad (2.3)$$

使用 maximum likelihood 的要求利用 Viterbi search 來找到最佳之切割位置 S。

我們針對每個句子，收集該句當中較可靠的 short pause，silence，breath 音框(該非語音段落長度至少長於 5 ms)，用 VQ 將這些資料依據 energy 大小分為兩群，能量較大的一群定為"non-speech signal"，能量較低的一群定為"silence"，將能量較低的一群抽取 13 維的 MFCC 參數拿來訓練 Gaussian 模型當做該句中 short pause 模型，之後從句子的開頭開始循序往後處理每個音節之間的 short pause，處理的方式如下：拿 sp 之前的 final 音段當中所有的音框拿來訓練一個 13 維 MFCC 的 final 高斯模型，同時拿 short pause 之後的 initial 音段當中較可靠的音框(一般來說是所有音框，但是針對某些音的特性而有些限制，比如說爆破音當中ㄅ，ㄉ，ㄍ這三個音特別的短，因此僅取該音結束點往前的 3 個音框)拿來訓練 13 維 MFCC 的 initial 模型，然後從 short pause 前的 final 起始點開始往後逐個音框對於 final， short pause，non-speech signal， initial 這幾個模型如下圖 2.3 進行 Viterbi Search 的比對，不過比對結束之後決定每個 state 的區段位置時必須保留呼吸聲保留原始的切割位置，因此實際上調整的有兩部分：final 與 short pause 之間的切割位置以及 initial 與 short pause 之間的切割位置。

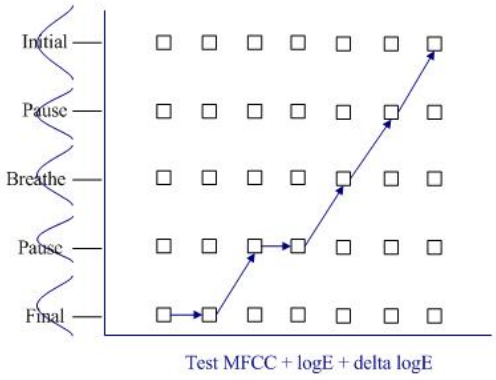


圖 2.4：音節間進行 Viterbi search 比對

每執行完一次 Viterbi search，規定結束點狀態必須是 initial，起始點狀態必須是 final 之後用 back trace 決定各個狀態的音框數，獲得新的切割位置之後，取新的 initial 與 final 音段音框分別更新 initial 與 final 的高斯模型，再次執行上述的流程，直到各個狀態音框段落都收斂之後，就算處理完一個音節之間的 short pause，之後以此類推處理完整個音檔，至於調整 initial 與 final 之間切割位置，所不同的是僅有兩個狀態轉換之間去做 Viterbi search，下頁圖 2.4 為取得音素切割位置的流程:

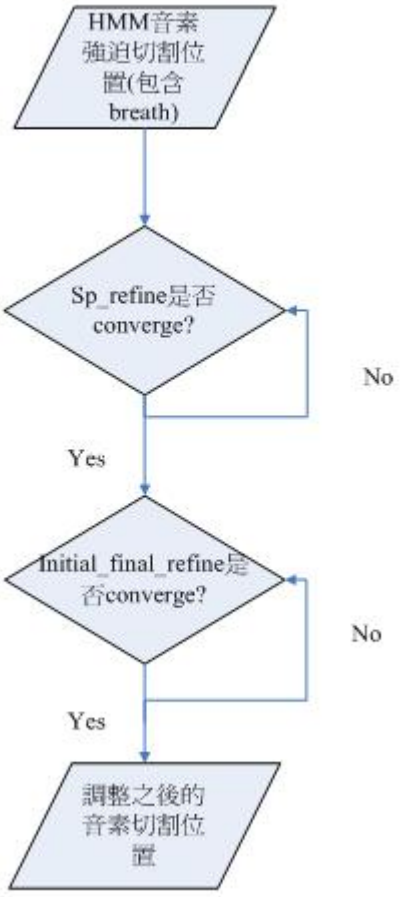
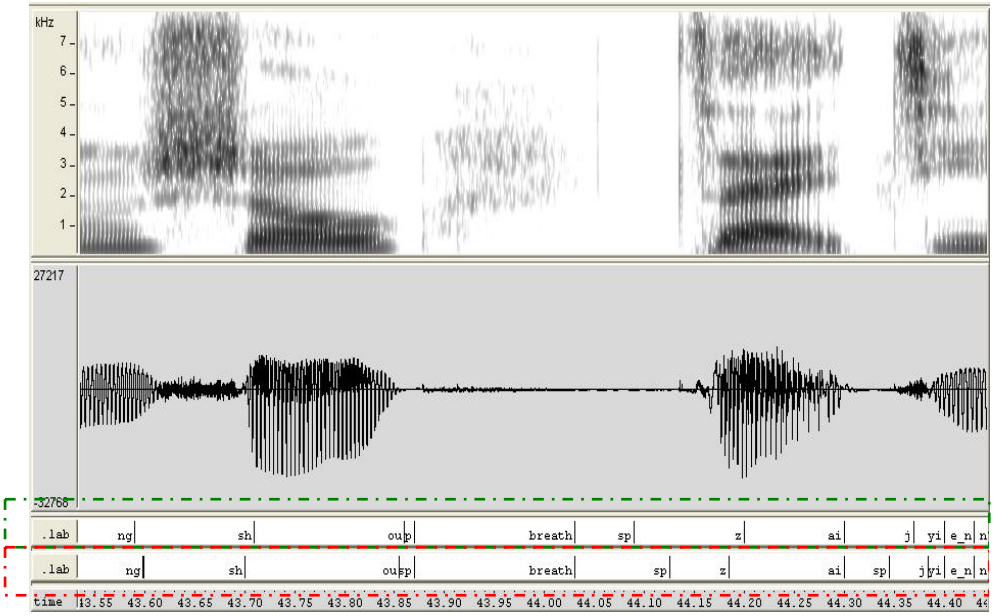


圖 2.5：取得音素切割位置的流程

舉個例子來觀察經過調整之後的切割位置更趨近於人工標記的切割位置:



未經調整的切割位置

經過調整的切割位置

圖 2.6：音素切割位置調整前後之比較

由上圖我們可以觀察到不管是音節與音節之間的 sp 或者是子音與母音之間的邊界切割位置都切的更準確，底下我們更細部的去觀察統計切割位置改進的情形，首先統計各發音方法調整之後的平均音長:

表 2.5：調整前後的發音方法平均音長統計

發音方法	調整前平均音長	調整後平均音長
Vowel	9.62	9.49
affricate	9.62	7.95
Liquid	6.69	6.13
fricative	10.99	10.88
Nasal	6.59	6.90
Stop	7.78	4.89
Silence	15.74	4.83
Breath	16.67	16.67

子音的音長除了 nasal 稍微變長以外都有或多或少的下降，這是由於子音前的 sp 能夠被有效的切出來因此使得子音的平均音長下降，特別是在發音的時候必須先緊閉聲道的 stop 以及 affricate 特別明顯，也因為音節間短暫的 sp 被有效

的還原出來因此造成 silence 的平均長度被拉低了，這時我們在將音長特別短的ㄅ、ㄉ、ㄍ這三個音的音長做比較:

表 2.6：ㄅㄉㄍ三種音素調整前後平均音長比較

	原始平均長度	調整之後平均長度
ㄅ(b)	6.36	3.41
ㄉ(d)	5.86	3.46
ㄍ(g)	6.85	3.58

很明顯的看到，ㄅ、ㄉ、ㄍ經過調整之後的平均長度都大幅的降低約 3 個音框長，明顯的比調整之前的音長要更趨近於合理的長度，接著我們進一步定量的分析實際上經過調整之後的 initial 之前的 sp 能夠大量的被還原出來使得 initial 的長度變的比較合理，我們針對此一現象特別明顯的 stop 音來抽樣，我們抽樣取 100 個 stop 音事件當中的前三個音框，而這些 stop 音事件分別平均分布於 10 個不同的音檔，我們用人工去實際比對音檔與切割位置，觀察比較未經過調整之前 stop 音的前三個音框有很大的成分實際上是沒被切出來的 sp，底下用直方圖來表示會很清楚：

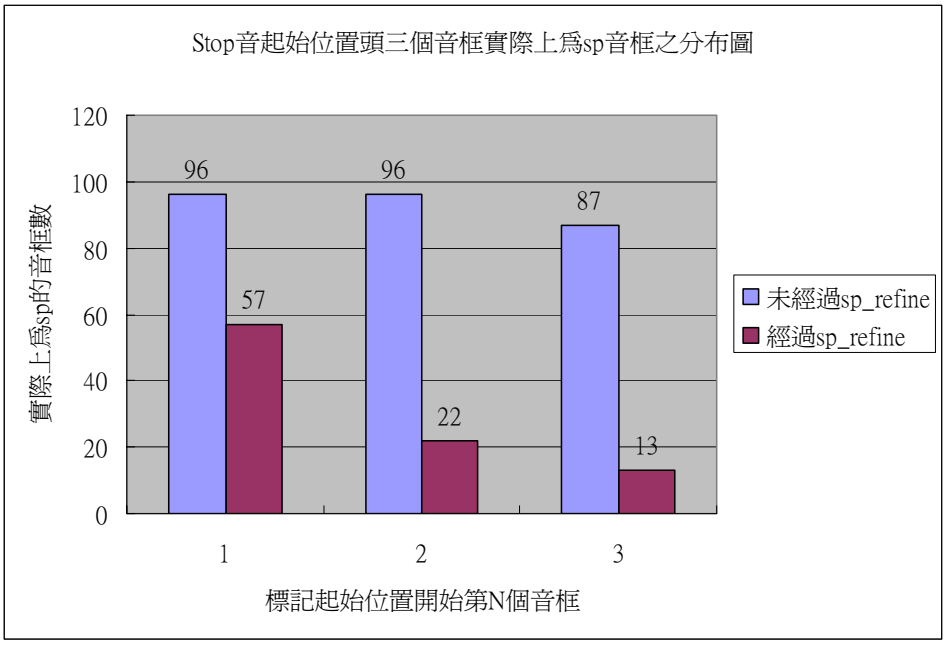


圖 2.7：Stop 音改善切割位置的比較統計

我們可以看到藉由抽樣觀察 100 個 stop 音事件當中，未經過調整的 stop 音切割位置幾乎前三個音框實際上都是 sp，而經過自動調整之後的切割位置大幅度的將 stop 音起始的頭三個音框(實際上是 sp)還原為 sp 音框。接下來為了分析調整子音與母音之後切割位置的改進，同樣取來自不同語者的 10 個音檔，其中每一句平均取若干個 initial 的事件來做統計，用直方圖來表示經過調整前後切割位置與實際人工標記比較的改進情況:

Total 樣本: 100 samples: fricative，affricate

50 samples: stop，nasal

40 samples: liquid

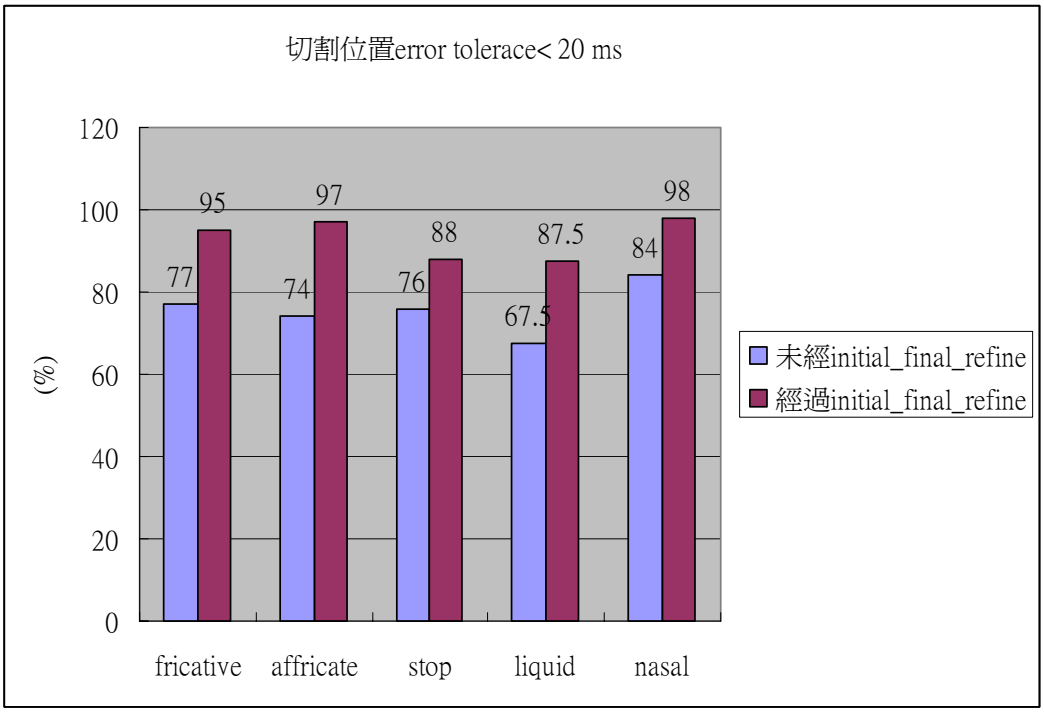


圖 2.8：子音與母音邊界切割位置改善的統計

由上圖統計資料可以看出，調整之前的子音母音交界切割位置誤差在 20 ms 以內的比例大約是還算不錯的 75%左右，但是經過自動調整之後的子音母音交界切割位置誤差在 20 ms 以內幾乎可以達到約 90%以上，由以上的切割位置改善統計結果可以看出，雖然 TCC300 語料庫沒有人工標記的音素切割位置，但是經由 Segmental K-means segmentation algorithm 自動的調整切割位置之後，已經能

夠得到趨近於人工標記的音素切割位置,因此往後章節的偵側器的訓練以及實驗都將以此調整後的音素切割位置來當作訓練各個發音方法模型後製作偵測器的依據。

2.3 中文語音屬性偵測器之初步建立

在前一節當中我們已經自動的調整 HMM 強迫切割的結果得到可靠的音素切割位置，接著將利用中文音素的發音方法分類表(表 2.7)[5]，將訓練語料以及測試語料的音素切割位置轉為發音方法的切割位置,在發音方法分類當中直得注意的是，原本在參考資料[5]當中ㄖ這個音的分類是屬於摩擦音，但是參考資料當中同樣有統計ㄖ這個音被 Liquid 偵測器偵測為 Liquid 的比例高達 76%，這是因為如果單獨念ㄖ這個音聲學特徵確實是屬於摩擦音,但是在中文連續語音當中語者往往因為連音的現象因此只有唸出ㄖ這個音的前半捲舌音(類似於 r 系音)因此也符合於參考文獻[4]當中對於 Liquid 這類音素的定義，因此在本論文中我們將ㄖ這個音素由摩擦音移至 Liquid 音的分類當中。

p.s.：括弧中為 IPA 表示　　表 2.7：中文發音方法分類表

1	爆破音（Stop）	ㄅ (p)	ㄆ (p□)	ㄉ (t)	ㄊ (t□)	ㄍ(k)	ㄎ (k□)
2	鼻音　（Nasal）	ㄇ (m)	ㄋ (n)	n_n , ng			
3	摩擦音（Fricative）	ㄈ (f)	ㄙ (s)	ㄒ (□)	ㄏ (x)	ㄕ (□)	
4	塞擦音（Affricate）	ㄓ (t)	ㄔ (t□□)	ㄑ (t□□)	ㄐ (t□)	ㄘ （t□s）	ㄗ（ts）
5	流音　（Liquid）	ㄌ (l)	ㄖ (□)				
6	母音　（Vowel）	others					

P.S. n_n , ng 為ㄤㄣㄢㄥ的鼻音韻尾

而此訓練語料的發音方法切割位置便作為我們在製作中文發音方法高斯混合模型貝氏偵測器的切割位置，我們使用 38 維 MFCC 參數當作我們訓練模型的特徵參數，最後再將製作出來的中文發音方法偵測器對測試語料作偵測求取偵測效能。

2.3.1 高斯混合模型

高斯混合模型是以高斯機率分佈為主體，包含多個高斯機率分佈，因此模型參數包含平均值向量（mean vector）、變異數向量（variance vector）以及混合數權重（mixture weight）。

下列式子為 n 個基本高斯機率分佈加權和(weighted summation)之高斯混合模型。

$$p(x|\theta) = \sum_{i=1}^{n} C_i \cdot N(\mu_i, \Sigma_i) \tag{2-4}$$

$$N(\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp[-\frac{1}{2}(x-\mu_i)^T (\Sigma_i)^{-1}(x-\mu_i)] \tag{2-5}$$

$$\theta = \{(C_i, \mu_i, \Sigma_i), 1 \le i \le n\} \tag{2-6}$$

其中 x 為一 D 維度大小之特徵參數向量，θ 為高斯混合模型，$N(\mu_i, \Sigma_i)$ 為高斯混合模型中各高斯分佈之機率密度函數，μ_i 為平均向量，Σ_i 為共變異矩陣(Covariance Matrix)，C_i 為混合權重，且須滿足 $\sum_{i=1}^{N} C_i = 1$。而在此實驗，我們假設共變異矩陣為一對角矩陣(Diagonal Matrix)。

在高斯混合模型的訓練中，可以利用最大似然度法則(Maximum Likelihood Criterion)來求得最佳模型，假設 $\bar{\theta}$ 為更新之模型、θ 為初始模型，使用預估最大值演算法（EM algorithm）去重新估算模型參數，使其滿足 $p(X|\bar{\theta}) \ge p(X|\theta)$ 之條件。亦即根據所有資料來估計統計特性，因此我們可以估算所有的平均向量，共變異矩陣，及各混合高斯模型之混合加權值，並將該統計出來的資料結果，根據

最大似然度方法達到最大化 $p(X|\bar{\theta})$ 的要求，如此即可找到模型參數，重估公式如下[10]：

$$C_i = \frac{1}{K}\sum_{k=1}^{K} p(i \mid x_k, \theta) = \frac{1}{K}\sum_{k=1}^{K} \frac{C_i\ p(x_k \mid i, \theta)}{\sum_{i=1}^{n} C_i\ p(x_k \mid i, \theta)} \tag{2-7}$$

$$\mu_i = \frac{1}{K}\sum_{k=1}^{K} \frac{C_i\ p(x_k \mid i, \theta)\, x_k}{\sum_{i=1}^{n} C_i\ p(x_k \mid i, \theta)} \tag{2-8}$$

$$[\Sigma_i]_{dd} = \frac{1}{K}\sum_{k=1}^{K} \frac{C_i\ p(x_k \mid i, \theta)\, [x_k - \mu_i]_d^2}{\sum_{i=1}^{n} C_i\ p(x_k \mid i, \theta)} \quad ; \quad 1 \le d \le D \tag{2-9}$$

其中 $[x]_d$ 是指向量中的第 d 個元素，$[\Sigma]_{ij}$ 則是指矩陣 Σ 的第 i 行第 j 列之元素。

2.3.2 貝氏偵測器架構

我們將建立以音框為基礎的中文語音屬性偵測器，而採用的偵測器架構為以高斯混合模型為基礎的貝氏偵測器架構。此貝氏偵測器架構為製作每一種發音方法以及發音位置偵測器時，我們將訓練兩種高斯混合模型[8]：一個為 target model，另一個為 anti-model。再藉由計算每個音框在此兩種模型上的似然度分數及考慮事前機率，採用最大似然度法則來決定每個音框是否屬於所要偵測的類別:

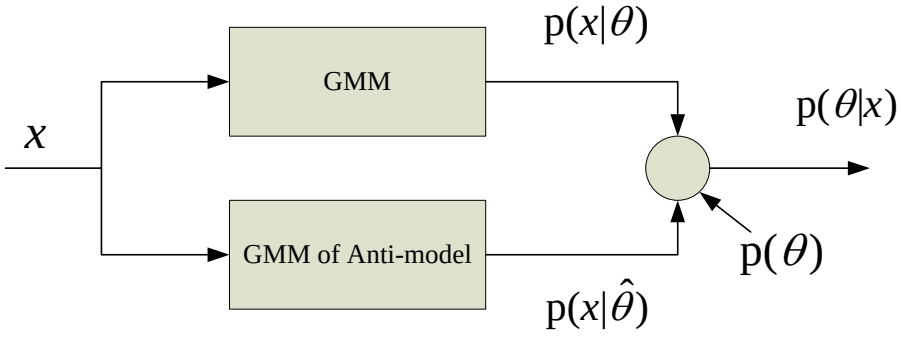


圖 2.9：貝氏偵測器架構圖

其中 X 語料庫中每一個音框的特徵參數向量，θ 為 target model、$\hat{\theta}$ 為 anti-model，$p(x|\theta)$ 為每一個音框在 target model 的近似度(likelihood)，$p(x|\hat{\theta})$ 為每一個音框在 anti-model 的近似度，$p(\theta)$ 為 target model 的事前機率，$p(\theta|x)$ 為每一音框屬於 target model 的事後機率(a posterior probability)。

接著藉著微調臨界值（threshold），可以得到偵測器對測試語料的錯誤警戒率（false alarm rate，FA）以及錯誤拒絕率（false reject rate，FR）的值，以下為錯誤警戒率、錯誤拒絕率、音框錯誤率的定義：

FA Rate = # of FAs / total # of non-target (2.10)

FR Rate = # of FRs / total # of targets (2.11)

Frame Error Rate = (# of FAs + # of FRs)/ total # of labels (2.12)

最後將所有錯誤警戒率與錯誤拒絕率的值畫出一個 FA-FR 的曲線圖。將可得到當錯誤警戒率等於錯誤拒絕率時的等錯誤率(Equal Error Rate，EER)。

2.3.3 中文發音方法偵測器之偵測效能

由 2.2 節當中，我們從音節切割位置起始著手訓練音素的馬可夫模型後，對語料庫作切割的方法，同時切出非語音的呼吸聲，接著再半自動的調整音素切割位置，而我們也將以此較可靠的切割位置當作是中文 TCC300 訓練語料的切割位置，訓練各個發音方法的高斯混合模型偵測器，以下將用此結果與使用音素 HMM 對語料進行強迫切割的切割位置所訓練的高斯模型製作的偵測器偵測效能做比較。

1. 以 HMM 強迫切割的音素切割位置訓練的高斯模型製作偵測器。
2. 以 HMM 強迫切割之後的音素切割位置再經過自動調整的切割位置訓練

的高斯模型偵測器。

表 2.8：以調整前後的切割位置訓練高斯混合模型偵測器偵測實驗

訓練語料切割位置 / 發音方法偵測	HMM 強迫切割位置 (EER%)	HMM 強迫切割經過調整後的切割位置 (EER%)
Stop	12.17	11.12
Nasal	11.90	11.57
Vowel	12.33	11.05
Affricate	11.91	10.98
Fricative	12.47	11.38
Liquid	9.73	9.16
Silence	11.98	7.25

上表顯示出經過調整切割位置之後的切割位置訓練偵測器，各種發音方法的偵測錯誤率都有明顯的下降，除了 Nasal 與 Liquid 之外，其餘發音方法偵測器的等錯誤率都有約 1%以上的下降，特別是 Silence 偵測器等錯誤率大幅的降低了 4%以上，這主要也是因為經過自動調整之後還原了許多音節間的 sp 的緣故，所以對於發音方法屬性偵測而言，經過調整之後的切割位置確實是比 HMM 強迫切割的切割位置要好，因此往後的章節當中將以不同方法訓練發音方法偵側器以及偵測實驗都將以此調整後的音素切割位置來當作訓練各個發音方法模型後製作偵測器以及測試語料的依據。

第三章　進階中文語音屬性偵測器之建立

在前一章當中我們已經取得了相當可靠的 TCC300 中文語料庫的音素切割位置之後，接著建立最基礎的 frame-based 高斯模型中文發音方法偵測器。由於非線性的類神經網路架構已經證明在資料類別分類上有優於線性高斯混合模型的效能，因此在本章當中首先建立屬於類神經網路的多層感知機 (Multi-layer perceptrons, MLP)模型為基礎的中文發音方法偵測器。然而在連續語音的語音屬性偵測當中單純的只考慮每個音框本身的資訊其實是不大合理的，因為即使是以音框為偵測的基本單元，每個音框仍舊會受到前後音框以及一些語言特性的影響，因此本章接著會加入類似以音段為基礎(segment-based)的概念，在原本的 MLP 模型為基礎的偵測器上加入 target 與 anti-model 這兩個狀態轉換的機率(transition probability)分數改善偵測器的效能，最後我們將由 MLP 發音方法偵測器為基礎建立階層式的語音屬性信任度量測(Confidence Measure)，如此一來便能夠評量偵測器偵測結果的可靠性，提供給自動語音辨識架構後級辨識器更可靠的語音資訊。

3.1 以 MLP 模型為基礎發音方法貝氏偵測器之製作

3.1.1 MLP 模型偵測器架構

MLP 模型廣泛的運用在各個領域當中作為資料分類的架構，同時因為其簡單、快速、容易收斂的特性因此在本章的一開始我們就運用此架構來訓練發音方法的偵測器。

我們採用的發音方法 MLP 模型分為三層如圖 3.1，包含一個輸入層、一個隱藏層、一個輸出層，根據我們輸入每個音框的 38 維 MFCC 參數因此輸入層點

數設定為 38，而隱藏層點數設定為 50，而輸出層由於我們同樣要訓練 target model 以及 anti model 因此設定點數為 2。

MLP 網路是一種正向饋入(feed-forward)網路，每一個第 i 層神經元的輸出 $O_k^{(i)}$ 都是第 i-1 層輸出加權總合的非線性函數，其數學式如下:

$$O_k^{(l)} = f(\sum_{i=1}^{N_\Delta} w_{ki}^{(l)} O_i^{(\Delta)} + \theta_k^{(l)}) \tag{3.1}$$

其中

$$f(x) = \frac{1.0}{1+e^{-x}} \tag{3.2}$$

為 sigmoid function，N_Δ 為第 Δ 層的神經元數目，$\theta_k^{(l)}$ 為 bias，而符號 $w_{ki}^{(l)}$ 則表示由 Δ 層的第 i 個神經元到 l 層的第 k 個神經元的加權值。

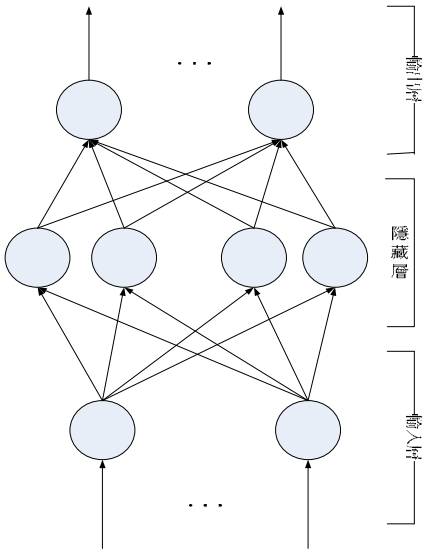


圖 3.1： MLP 網路架構

3.1.2 MLP 模型偵測器之偵測效能

在前一章當中我們已經得到相當可靠的音素切割位置，有了切割位置之後我們便直接拿來訓練各個中文發音方法的偵測器，然後與高斯混合模型為基礎的偵測實驗結果相比較:

1. 以 TCC300 中文語料庫經過調整後的音素切割位置訓練 38 維 MFCC 參數的

高斯混合模型發音方法偵測器。

2. 以 TCC300 中文語料庫經過調整後的音素切割位置訓練 38 維 MFCC 參數的 MLP 模型發音方法偵測器。

表 3.1：GMM 及 MLP 為基礎的發音方法偵測效能比較

EER(%) / manner	GMM	MLP
Vowel	11.05	8.29
Stop	11.12	9.98
Fricative	11.38	10.06
Affricate	10.98	9.17
Nasal	11.57	9.25
Liquid	9.15	9.16
Silence	7.25	5.72

由結果可以得知，除了 Liquid 的等錯誤率幾無變化之外，其餘的發音方法偵測等錯誤率都有約 1~2.5%的下降，特別是 Vowel 以及 Nasal 還有 Silence 這三類偵測器，error reduction 都有超過 20%的下降，而這三類的發音方法資料量約佔語料庫總資料量的 70%，因此這三類偵測器錯誤率明顯的下降對於整體偵測器的效能有顯著的提升。

在得到了 frame-based MLP 中文發音方法偵測器的結果之後，我們將以 segment 的角度來觀察 frame-based 的 MLP 偵測器的偵測錯誤情形類別做分析，首先是錯誤拒絕的部份:

(I) False-reject:

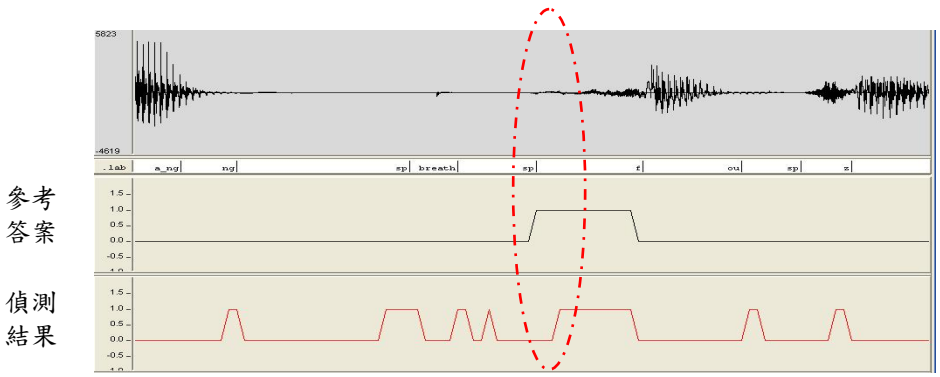


圖 3.2：偵測的結果稍微向內縮

這類型的錯誤多半是因為邊界附近的聲學特徵還不是很穩定因此發生錯誤拒絕的偵測錯誤，但是以 segment 的角度來看這類型偵測錯誤的情形並不算嚴重，而這類型的錯誤對於 frame-based 偵測器的偵測錯誤影響我們將在後面的章節再作較深入的分析。

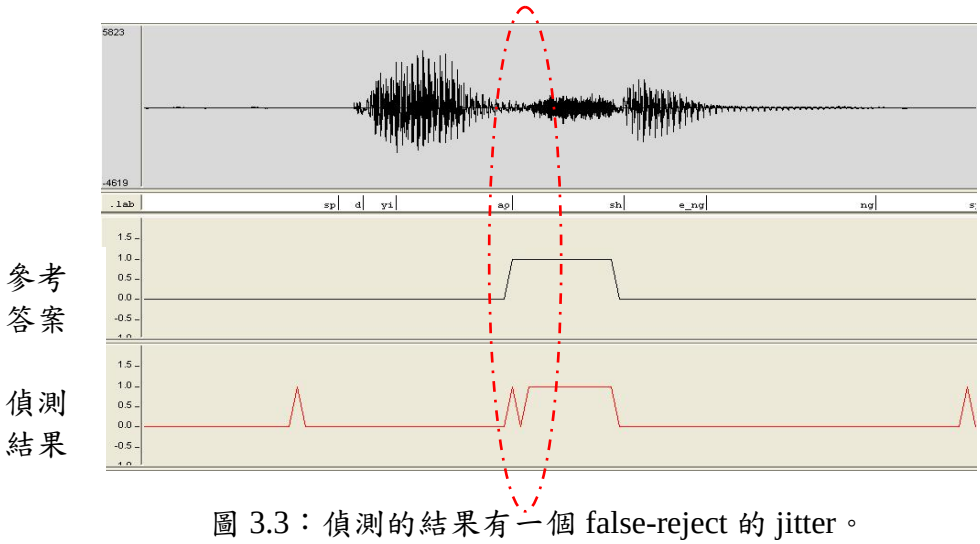


圖 3.3：偵測的結果有一個 false-reject 的 jitter。

這類型的錯誤非常常見但是實際上這些短暫的 jitter 是造成偵測錯誤的主要來源之一，並且也不能提供後級的辨識器可靠的資訊，如果能夠加入一些 segment 概念的資訊應該就能夠有效抑制這類型的錯誤，因此在下一節當中我們將會加入 target 與 anti-model 這兩種狀態轉移的機率分數來試圖克服此類型的偵測錯誤。

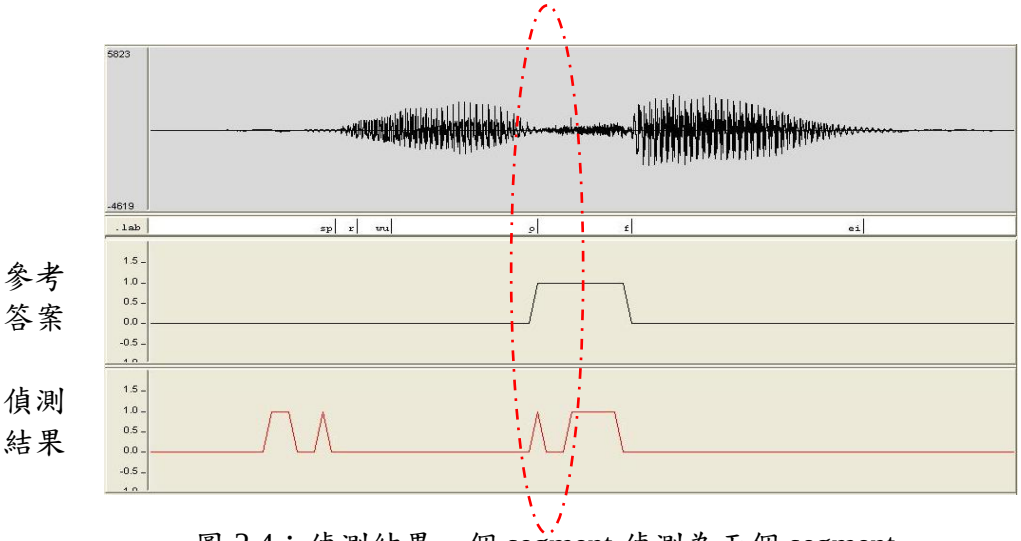


圖 3.4：偵測結果一個 segment 偵測為兩個 segment。

這類型的錯誤時常出現於各種發音方法中能量相對比較小的摩擦音段當中音頭以及音尾部分以及 Silence 段中可能夾雜些微背景雜訊時。

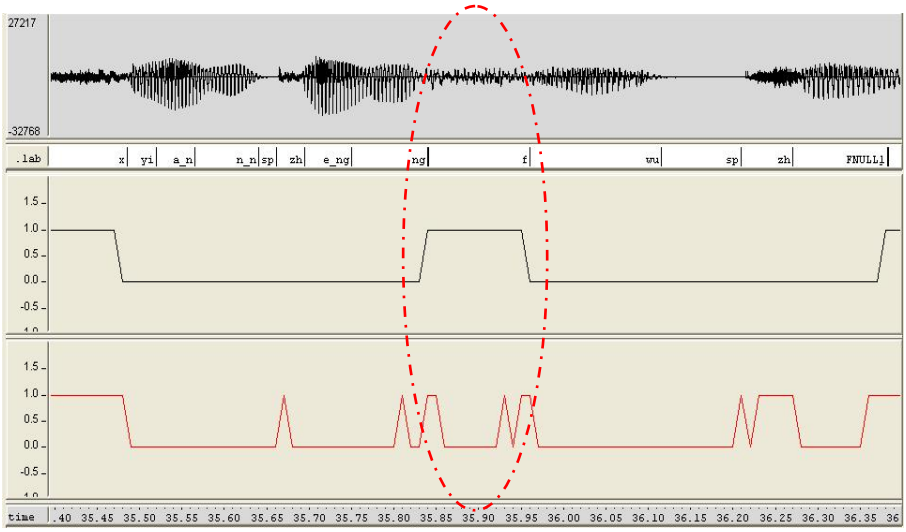


圖 3.5：偵測結果一個 segmnet 偵測為數個 segment。

這類型的偵測錯誤通常發生在聲學特徵較不穩定的音素當中，比如說摩擦音當中的ㄈ這個音，在隨機抽樣觀察的幾個句子當中發現到ㄈ這個音常常有錯誤拒絕很嚴重的情形，因此我們特別針對這個音去對摩擦音偵測器作偵測，同樣發現到ㄈ這個音雖然屬於摩擦音但是對於摩擦音偵測器的錯誤拒絕率卻高達 32%，並且對於 Silence 偵測器的錯誤警戒率(也就是被偵測為 Silence)將近有 50%，這是因為ㄈ這個音的聲學特徵其實有些類似於語者呼吸聲，因此也許此音素在發音方法屬性偵測的分類上因其特殊的聲學特性而有需要獨立出來自成一類。在分析完了錯誤拒絕類型的錯誤之後我們接著對於錯誤警戒類型的偵測錯誤同樣用 segment 的角度來做各種偵測警戒錯誤的類型分析:

(II) False_alarm:

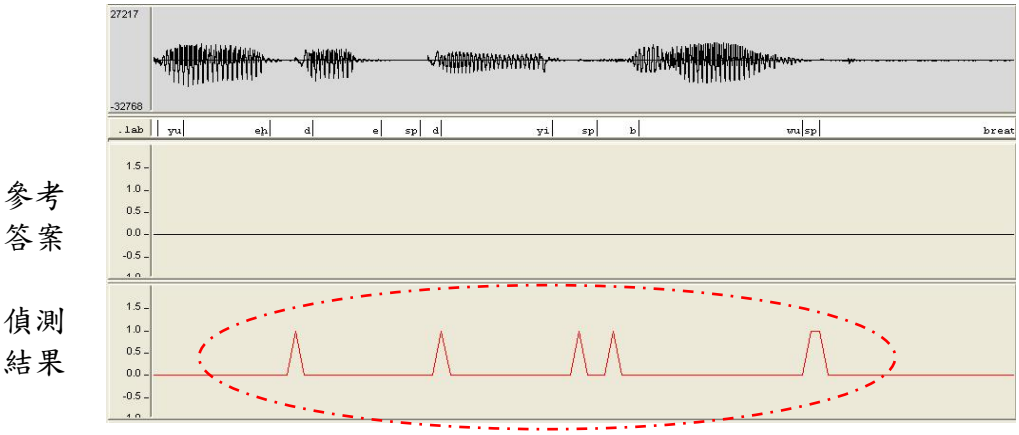


圖 3.6：false-alarm jitter

同偵測錯誤拒絕當中的第 b 類型錯誤，事實上除了一部分 Stop 音以及 Silence 以外其他類發音方法幾乎不可能單獨出現這麼短的 target segment(1~2 個音框)，因此這類十分明類的錯誤便是我們下一節當中提出狀態轉移機率概念最主要要解決的偵測錯誤類型。

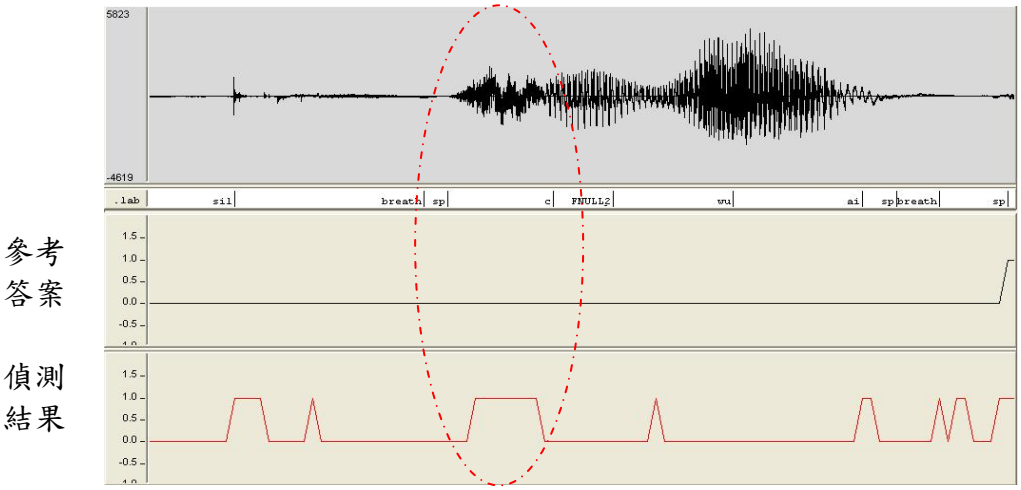


圖 3.7：一長段連續 false-alarm

這類型的錯誤通常發生在該種發音方法與某種聲學特徵相近的發音方法之間互相混淆情形非常嚴重時，最明顯的情形就是如上圖當中的例子，該音素ㄘ(c)是屬於 Affricate，但是由於 Affricate 與 Fricative 混淆的情形十分嚴重因此整段被 Fricative 偵測為錯誤警戒的錯誤，同樣的錯誤類型也常見於聲學特徵類似的 Nasal 與 Liquid 之間以及 Nasal 與 Vowel 之間。

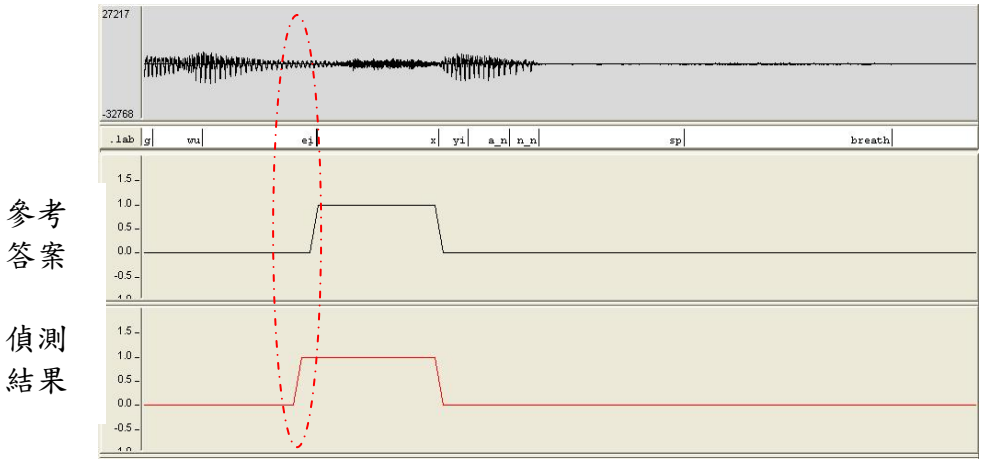


圖 3.8：segmnet 邊緣向外稍微延伸

此類型的錯誤類似於錯誤拒絕當中的 a 類型錯誤，因此同樣以 segment 的角度來看這類型的錯誤時這類型並不算是嚴重的偵測錯誤。

以上是對於 frame-based MLP 屬性偵測的錯誤類型的分析，之後的章節我們將針對考量 frame-based MLP 屬性偵測器偵測錯誤的缺失為基礎，提出加入提供偵測器更多資訊的方式，期望能夠降低偵測器一部分的偵測錯誤。

3.2 MLP 模型為基礎加上狀態轉移機率的發音方法偵測

在前一節當中我們利用類神經網路當中 MLP 模型架構所訓練的偵測器雖然已經能夠得到偵測器效能明顯的提升，但是如果提供更多的聲學資訊加入偵測器的訓練當中應該能再進一步的提升偵測器效能，因此接下來我們將 MLP 的偵測器的輸出分數加入狀態轉移機率的分數作整合。

3.2.1 整合狀態轉移機率的偵測架構

由於在之前 frame-based 的 MLP 發音方法偵測實驗中，我們已經得到每一個音框在偵測器的分數以及偵測結果，然而偵測的結果純粹是用等錯誤率的狀況下該音框在 target model 上的事後機率分數是否大於 anti model 上的事後機率分

數來判定是 target 還是 anti-model，接下來我們取出偵測器的分數，加入 duration model 限制的概念，在每一句當中考慮 target，anti-model 這兩類狀態轉換的機率，讓句子當中的每一個音框進行 Viterbi Search，找到最佳的偵測結果。假定經過 Viterbi Search 後的最佳化 utterance score 為 Q^*，其數學表示式如下:

$$Q^* = \arg\max_s Q(S,O)$$

$$= \sum_{t=1}^{N} Q_{MLP}(S_t,O_t) + \sum_{t=1}^{N} Q_A(S_t,S_{t-1}) \quad (3.3)$$

其中 S =1、2； O 為 observation；當 t=1 的時候，若偵測的 target 為 silence，則 S_1 為 2，若偵測的 target 為其餘發音方法，則 S_1 為 1，因此原式可寫成:

$$Q^* = \sum_{t=1}^{N} Q_{MLP}(S_t,O_t) + \sum_{t=2}^{N} Q_A(S_t,S_{t-1}) \quad (3.4)$$

而 $Q_{MLP}(S_t,O_t)$ 即為原本 frame-based 偵測器輸出每個音框在 target model 上的分數取對數，而狀態轉移分數 $Q_A(S_t,S_{t-1})$ 為狀態轉移機率取對數，也就是 $\log(P(S_t \mid S_{t-1}))$，狀態轉移機率用訓練語料當中 target-segment，anti-model segment 的平均長度求得，假設 target-segment 平均長度為 L2,anti_segment 平均長度為 L1 則狀態轉移機率為:

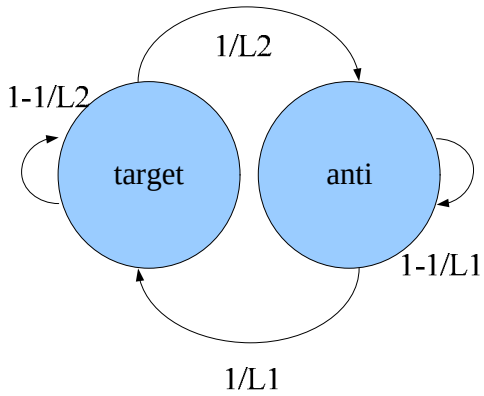


圖 3.9：發音方法音長模型狀態轉移圖

3.2.2 整合狀態轉移機率的偵測效能

以下我們將加入狀態轉移機率分數之後的偵測結果與之前 frame-based 的

MLP 偵測器之偵測結果比較:

表 3.2：加入音長模型之後的錯誤率統計

	MLP	MLP + duration model		
manner	EER(%)	False alarm rate(%)	False reject rate(%)	Frame error rate(%)
Vowel	8.29	8.71	7.05	7.92
Stop	9.98	6.99	10.53	7.13
Fricative	10.06	7.08	9.68	7.31
Affricate	9.17	7.74	8.47	7.80
Nasal	9.25	7.03	9.67	7.30
Liquid	9.16	6.91	9.26	6.95
Silence	5.72	3.59	8.21	4.52

雖然這裡求得的錯誤率並不是等錯誤率，但是我們仍舊可以從統計的結果看出偵測錯誤率下降的現象，如上表當中用紅色數字標記錯誤率的 Fricative 以及 Affricate，其偵測的結果無論是錯誤警戒率(FA rate)或者是錯誤拒絕率(FR rate)都較原本的等錯誤率(錯誤警誡律=錯誤拒絕率)低的多，因此很明顯的這兩類的偵測器效能獲得很明顯的提升，至於 Vowel，Stop，Nasal，Liquid 這四類發音方法的偵測結果，其中一種錯誤率大幅的下降但是同一時間另外錯誤率卻小幅的上昇，雖然錯誤警戒以及錯誤拒絕率沒有同時下降，但是依照比例以及音框錯誤率(frame error rate)來看整體來說偵測器的效能依然是比原本沒有加上音長模型的效能要好，至於 Silence 偵測器由於錯誤警戒率以及錯誤拒絕率的變動以及差距較大，因此比較不能判斷偵測器的好壞。

下面我們觀察加上狀態轉移機率之後的偵測情形來分析，舉個 Nasal 的偵測情形觀察是否 jitter 類型的偵測錯誤能夠有效的被抑制:

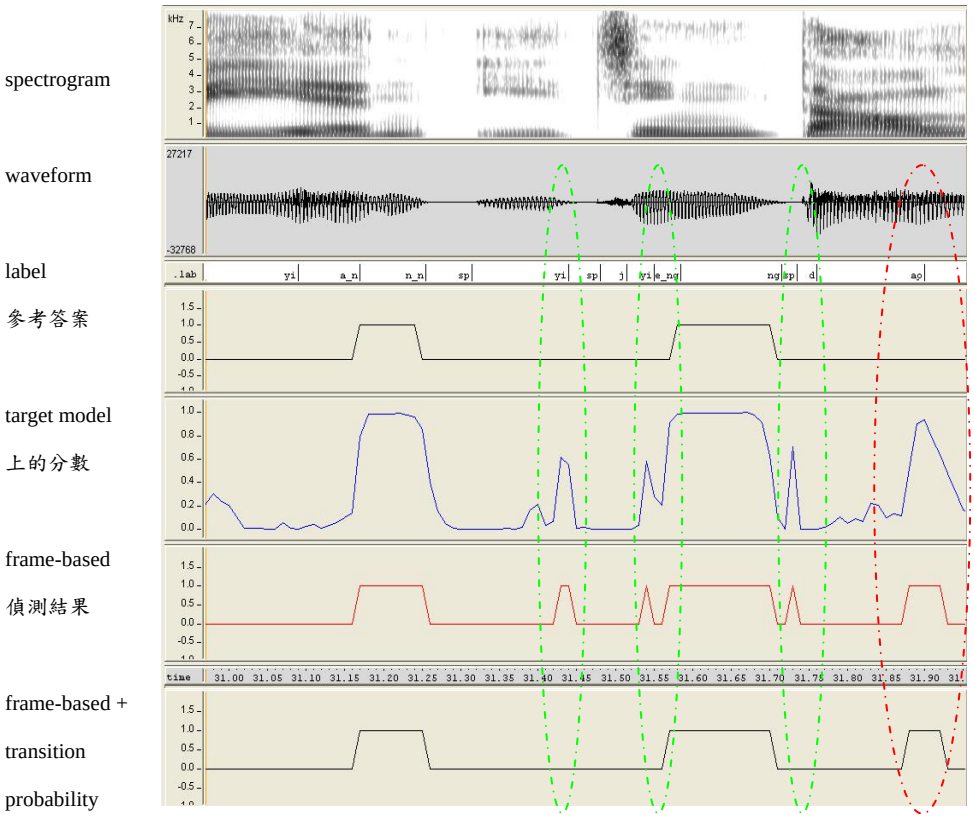


圖 3.10： Nasal 發音方法偵測實例

從上面的例子可以看到，雖然一整個音段聲學特徵都非常類似 Nasal 而造成的整段偵測錯誤警戒(如紅色虛線所示)這類型的偵測錯誤沒有明顯改善，但是在 frame-based 偵測錯誤當中十分常見的 jitter 偵測錯誤類型(綠色虛線標示)的偵測錯誤幾乎都被排除了，代表說加入狀態轉移機率分數確實能夠有效的排除不合理的 jitter 類型偵測錯誤。

接著圖 3.11~3.17 我們統計出原本各發音方法音長段落的分布、frame-based MLP 偵測結果段落長度分布以及加入狀態轉移機率的偵測結果段落長度分布:

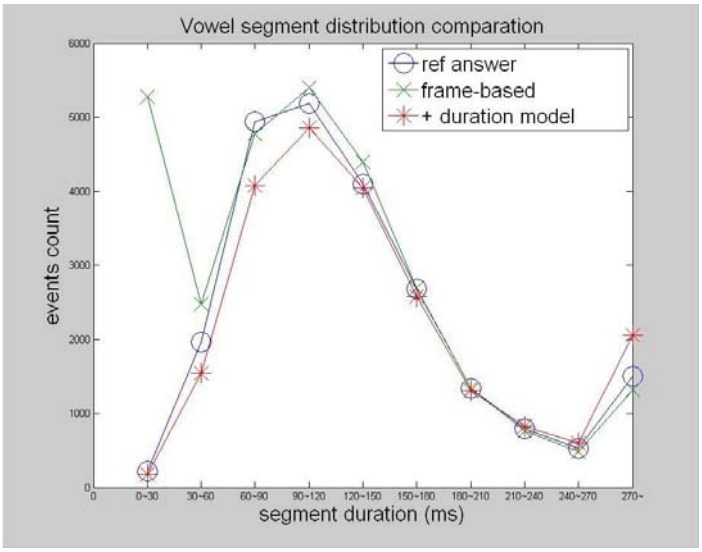


圖 3.11：Vowel 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)

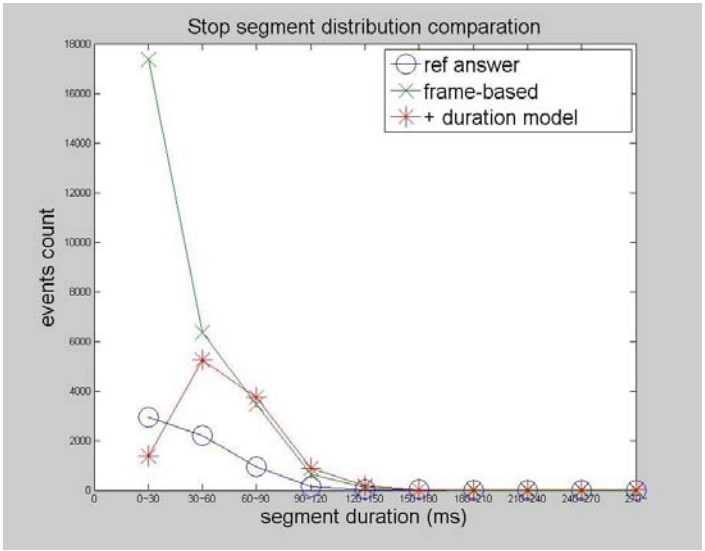


圖 3.12：Stop 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)

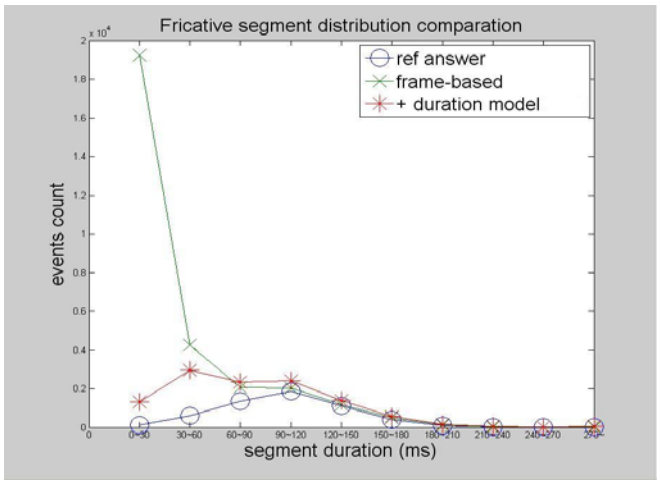


圖 3.13：Fricative 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)

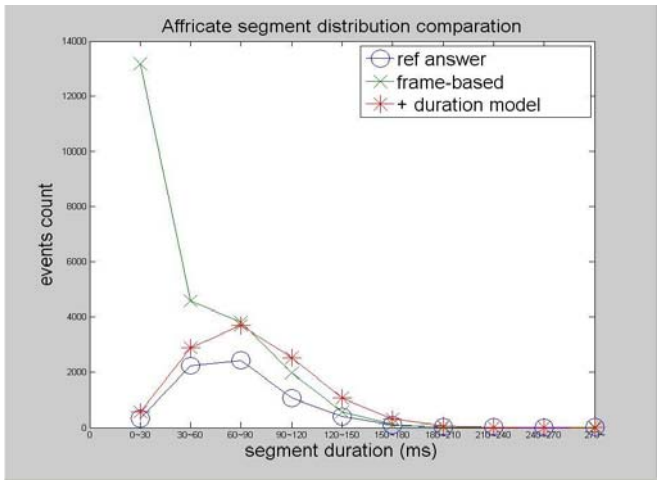


圖 3.14：Affricate 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)

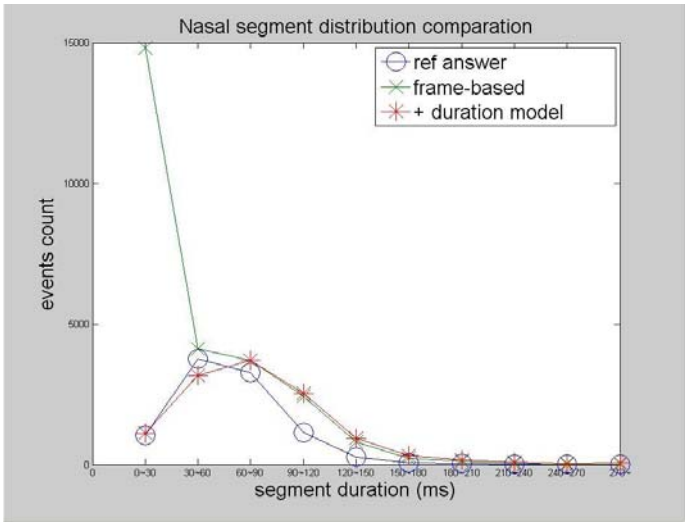


圖 3.15：Nasal 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)

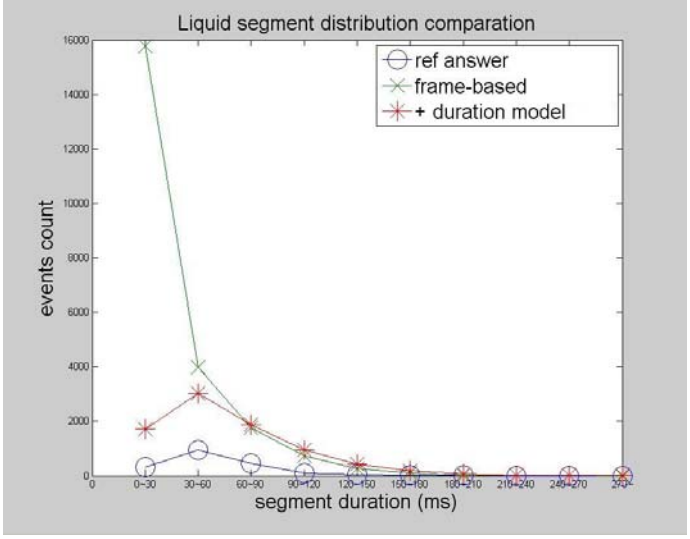


圖 3.16：Liquid 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)

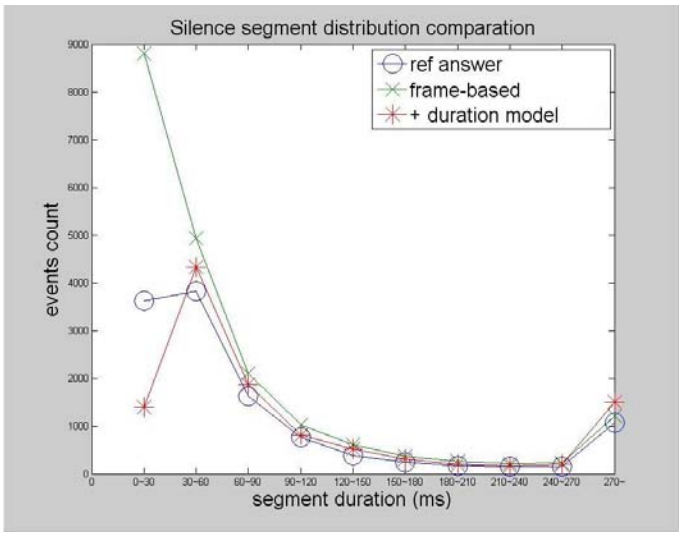


圖 3.17：Silence 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)

由各種發音方法偵測結果音長分布的統計我們可以清楚的看到，原始的 MLP 偵測結果偵測出太多 jitter 型態的段落，造成了音長分布大多偏向音長很短的段落，而加入狀態轉移機率之後的偵測結果，等於說是加入各種發音方法在 segment 音段長度的資訊，使得偵測的結果 segment 長度分佈明顯較趨近於實際上的音長分佈。

不過一般來說偵測器的效能仍舊是以求得等錯誤率來評斷，因為等錯誤率考量到 target 資料量與 anti 資料量不一定相當的問題，因此接下來我們將導入以

下的數學式加入一個可以調整的權重值，取得偵測器錯誤拒絕率與錯誤警戒率相等的等錯誤率。

$$AP'(S=1)=\frac{AP(S=1)\times C}{AP(S=1)\times C+AP(S=0)} \tag{3.5}$$

$$AP'(S=0)=\frac{AP(S=0)}{AP(S=1)\times C+AP(S=0)} \tag{3.6}$$

$AP(S=1)$，$AP(S=0)$ 分別為音框在 target model 以及 anti model 上的事後機率分數，C 是用來將錯誤率調適成等錯誤率的權重參數，$AP'(S=1)$ 為調適之後新的 target model 分數，$AP'(S=0)$ 為調適之後新的 anti model 分數。底下是調整權重使得錯誤率為等錯誤率的結果比較:

表 3.3：加入狀態轉移機率前後的等錯誤率偵測結果比較

EER(%) / manner	frame-based MLP	frame-based MLP+ transition probability
Vowel	8.29	7.93
Stop	9.98	8.58
Fricative	10.06	8.32
Affricate	9.17	8.07
Nasal	9.25	8.30
Liquid	9.16	8.01
Silence	5.72	5.39

由上表的偵測結果可以看到 Vowel 與 Silence 的偵測器錯誤率僅降低了 0.3%，不過其餘各類的發音方法偵測結果等錯誤率均有 1%以上的下降，證明在等錯誤率的情形下加入segment 資訊的偵測器效能比單純僅有音框資訊的偵測器要明顯提升許多。

統計完了等錯誤率的比較之後，我們將等錯誤率情形下的偵測結果與之前沒有調至等錯誤率的偵測結果(如圖 3.10)做比較觀察偵測結果段落的差異:

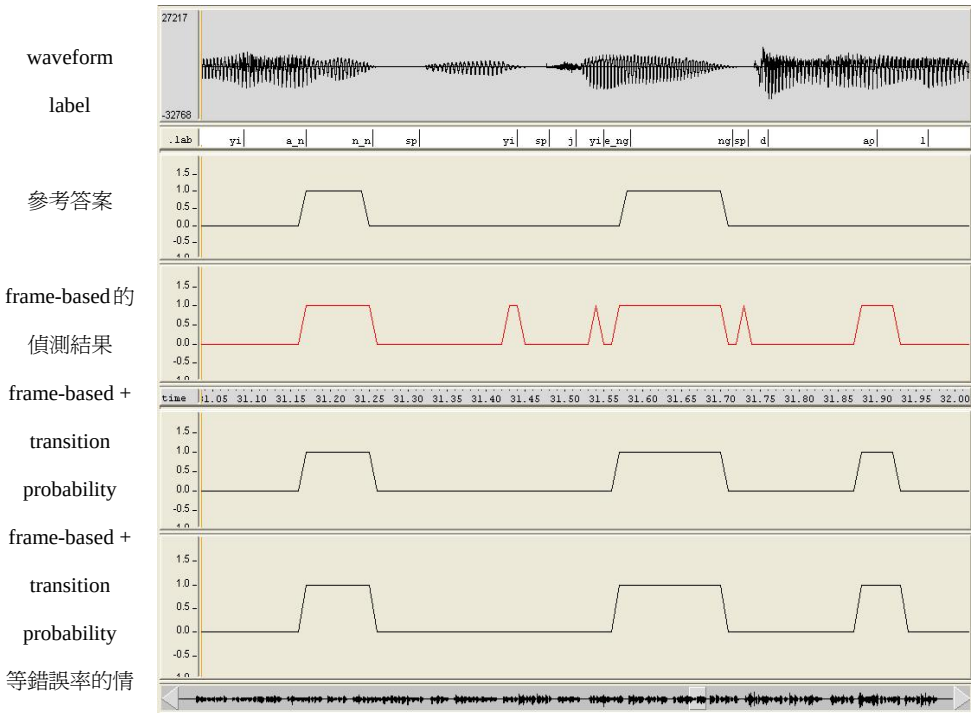


圖 3.18：調整成等錯誤率前後的偵測結果實例(I)

由這個例子可以看到偵測結果其實沒有什麼差異，同樣保留了去除 jitter 類型錯誤的優點，底下我們再找另外一個差異較明顯的例子:

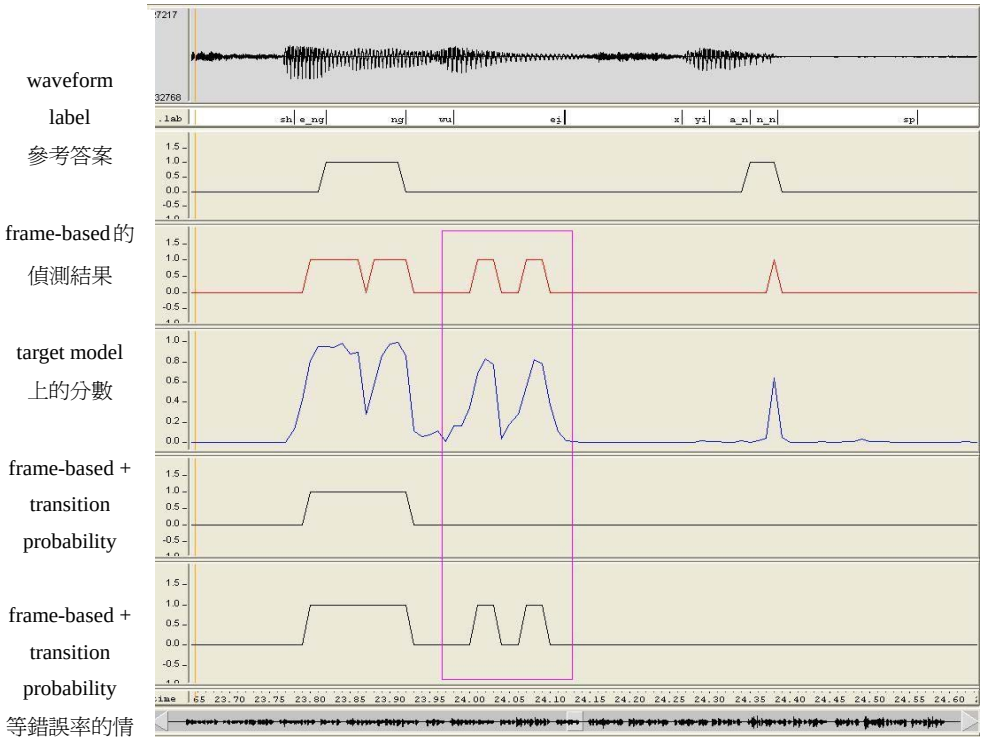


圖 3.19：調整成等錯誤率前後的偵測結果實例(II)

由圖 3.18、圖 3.19 觀察得知，調整成等錯誤率雖然會稍微增加一部分的錯

誤率，不過加入轉移機率最主要優點是抑制 jitter 這一點仍然保留了下來，在附錄當中有統計等錯誤率狀況下偵測結果的segment長度分布與原本未調至等錯誤率的分布差不多同樣證明了這一點，因此我們還是可以肯定偵測器加入 segment 的資訊能夠有效提升偵測器效能。

3.3 以 frame-based MLP 偵測器為基礎之階層式信任度量測

3.3.1 階層式信任度量測架構

在本章的第一小節當中我們已經得到 frame-based MLP 發音方法的等錯誤率偵測結果，但是屬性偵測的目的是當作自動語音辨識系統的前端，提供可靠的語音資訊提供給後端的辨識器使用，因此得到偵測器的結果之後我們必須對於偵測的結果進行信任度的量測(Confidence Measure)，信任度較高的偵測結果才能提供有效的語音資訊給後級辨識器。

我們提出階層式(hierarchical)的信任度量測架構，最底層為七種發音方法加上呼吸聲(Breath)共八類偵測器的偵測結果進行信任度量測，而第二層將聲學特徵極為類似的 Fricative 與 Affricate 以及 Vowel 與 Nasal 還有 Vowel 與 Liquid 分別合併，再上一層便將非響音(non-sonorant)包含 Vowel、Nasal、Liquid 以及響音(sonorant)的 Fricative、Stop、Affricate 合併在一起，而最上層便是將語音(speech)的部份包括 Vowel、Nasal、Liquid、Fricative、Affricate、Stop 合併以及非語音(non-speech)的部份包括 Silence、Breath 合併。架構運作的方式為欲確認偵測結果可靠程度之音框假使在最底層的偵測信任度就超過門檻值，我們就確認該音框的屬性偵測結果，假如說該音框在最底層的信任度低於門檻值，我們便對該音框進行第二層語音屬性分類偵測的信任度量測，假如該音框在第二層的偵測信任度量測高於門檻值，我們便確認該音框的偵測結果為第二層當中對應的屬性分類，假使該音框在第二層的信任度仍舊低於門檻，便將該音框進行第三層的信任度量

測，以此類推，而下圖 3.20 是架構圖:

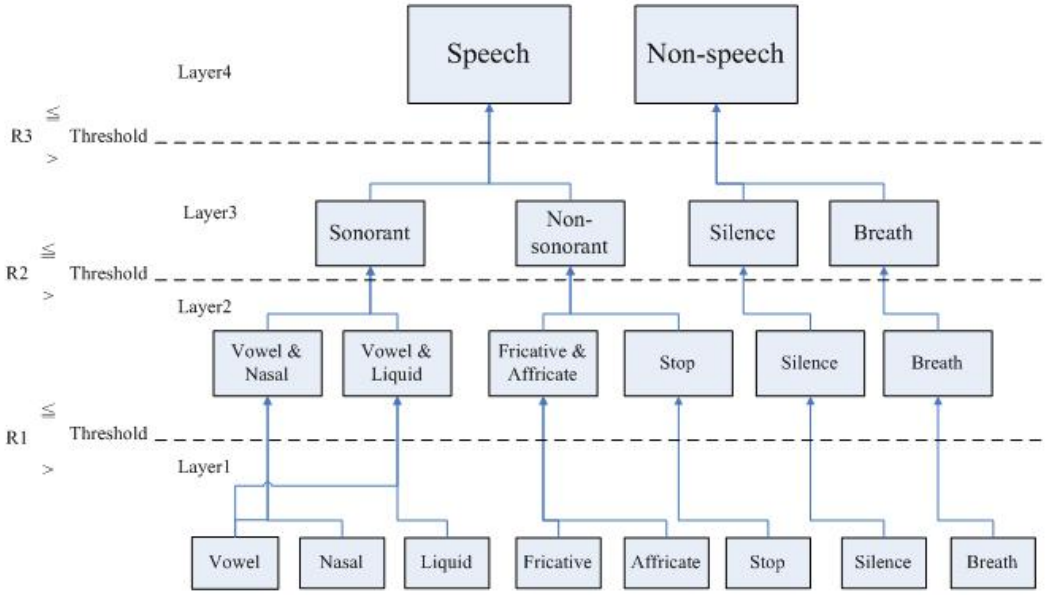


圖 3.20：階層式的屬性偵測器信任度量測架構圖

為了計算屬性偵測結果的信任度，首先將八類偵測器的偵測結果 normalize 使其總和值為 1:

$$AP_{total} = \sum_{i=1}^{N} AP_i \qquad N=8 \tag{3.7}$$

$$AP_i' = \frac{AP_i}{AP_{total}} \qquad 1 \leqq i \leqq 8 \tag{3.8}$$

AP_i 為各種偵測器在 target model 上的分數，接著將上式 normalized 後的結果 AP_i' 運算每個音框偵測結果的亂度(entropy)[9]:

$$H = \sum_{i=1}^{N} AP_i' \log(\frac{1}{AP_i'}) \qquad \text{N=8} \tag{3.9}$$

計算出亂度 H 之後便將 H 代入 sigmoid function 得到信任度 R，亂度 H 若是越低會得到較高的 R 值，也就是說該音框的偵測結果較為可信:

$$R = \frac{1}{1+\exp(\lambda(H-\beta))} \tag{3.10}$$

其中 λ、β 的值目前調整為適當的值使得 R 值的分布在 0~1 之間。得到信任度之

後我們將信任度必須超過一個門檻值的音框偵測結果才認定為可靠的資訊。

3.3.2 階層式信任度量測效能

以下是對於各個階層以不同的信任度門檻值分別求得的可靠資訊正確率:

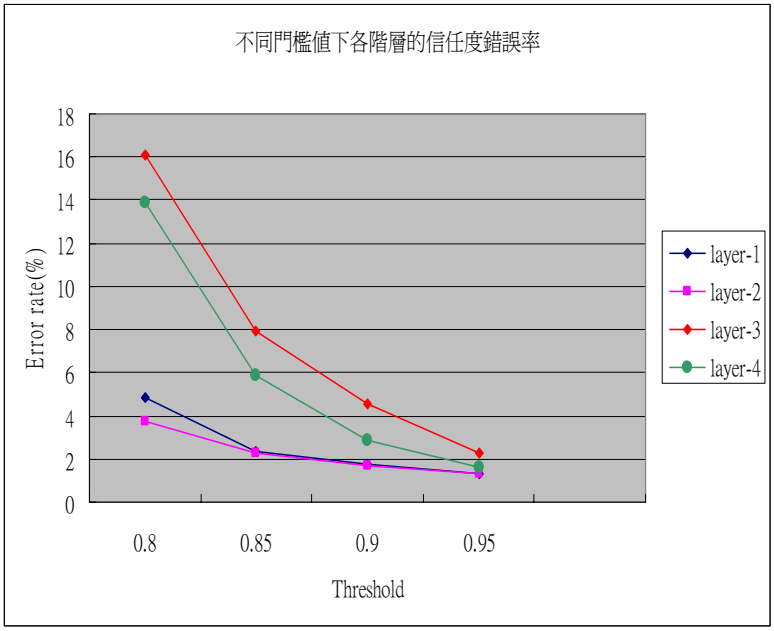


圖 3.21：不同門檻值下各階層的信任度錯誤率

我們可以從統計的結果看到，如果將可靠度的門檻訂的越高，則提供給後級辨識器的資料的錯誤率就越低，底下我們統計各個門檻值之下，各階層信任度高於門檻值的資料涵蓋所有資料量的比例(inclusion rate)再與圖 3.21 一起做分析:

表 3.4：各門檻值下各階層信任度高於門檻值的資料比例

門檻值 階層	0.8	0.85	0.9	0.95
1-layer	28.6%	21.8%	16.7%	10.3%
2-layer	19.6%	21.8%	21.0%	16.7%
3-layer	24.9%	21.0%	20.0%	21.0%
4-layer	26.9%	27.1%	27.0%	28.0%
Total	100%	91.7%	84.7%	76.0%
Correct rate	90.1%	95.8%	97.2%	98.7%
Undetect	0%	8.3%	15.3%	24.0%
Undetect correct rate (speech / nonspeech)	----	61.1%	70.8%	77.6%
No hierarchy correct rate	81.0%			

對於第一層而言，若是信任度量測要高於門檻值，必須要僅有一類發音方法偵測器的分數比其餘發音方法偵測器分數高很多，因此隨著門檻值提高，涵蓋率明顯下降，同時由於分類較細，因此在高門檻值之下較不容易發生信任度量測錯誤的現象。而第二層將容易混淆的發音方法合併，從圖 3.21 上可以得到此一合併的結果能夠得到不錯的錯誤率，不過由於合併之後的分類涵蓋了一種以上的發音方法，因此同一分類當中個別發音方法的偵測錯誤可能在分類合併時加成在一起，造成信任度錯誤率稍有提高。而第三層將響音(sonorant)及非響音合併，由統計的結果看來此一合併的信任度量測結果由於多種發音方法的個別偵測錯誤加成在一起，因此在較低的門檻值下信任度量測的效能會大幅的降低。而最上層的語音(speech)與非語音(non-speech)的信任度量測同樣有與第三層類似的問題，在較低的門檻值下錯誤率升高的很快，不過整體來說錯誤率要比第三層為低。綜合圖 3.21 以及表 3.4 的結果，可以看到隨著門檻值提高，信任度量測的涵

蓋率便隨之降低，但是即使將門檻值提高到 0.9，信任度量測的涵蓋率依然有大約 85%，同時信任度的正確率達到 97%以上，不過值得注意的是，每個階層的分類不同，因此各個階層不一定要用相同的門檻值，由統計中可以看出，即使在門檻值為 0.8 時涵蓋率僅有 28.6%，代表說在最底層其實將很多的資料認定為”不可靠”而進行更上層的信任度量測。接著在下圖當中我們對於信任度量測的結果以最底層的信任度量測結果與各偵測器的偵測結果之間的關係做初步的分析:

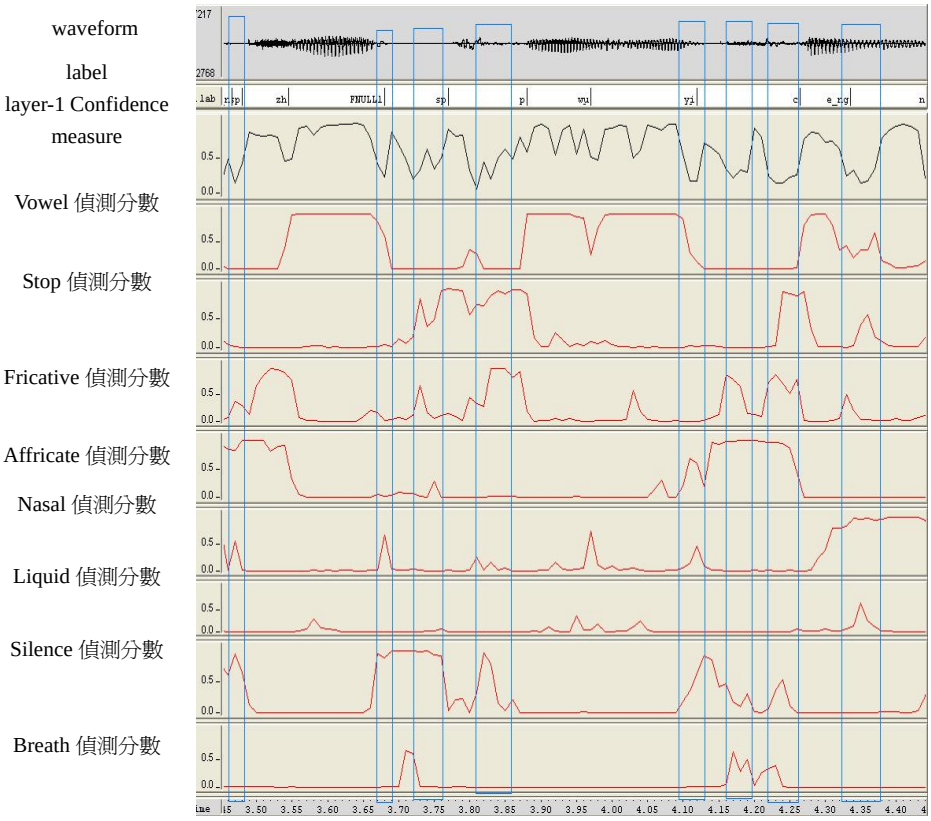


圖 3.22：信任度量測結果與各發音方法偵測結果比較

我們可以從上圖中看到，大部分信任度不足的情形(如藍色框框標示)發生在音素間的交界附近以及發音方法容易混淆的地方，因為音素間交界的部份往往因為前後音素不同兩類發音方法偵測分數都很高，因此該音素交界區域偵測結果的亂度較高造成信任度的降低，同時容易混淆的音框往往有兩類以上的發音方法偵測分數均頗高，因此造成偵測結果同樣不可靠，而在較不容易混淆的音框由於大部分僅有一類發音方法偵測的分數很高，因此該區段偵測結果的亂度較低所以信任度也就相對的提高，代表說該區段的偵測結果是較為可信的。

第四章　中文發音方法偵測器效能的分析與討論

在前兩章當中已經有對於不同偵測器架構當中各發音方法偵測等錯誤率的結果做統計並且初步比較偵測器效能的改進,在本章當中我們將更進一步去分析哪些發音方法以及哪些音素容易造成特定發音方法的偵測錯誤造成混淆、中文連續語音當中語者時常未把鼻音韻尾唸出來之現象對於屬性偵測的影響以及討論偵測錯誤發生在音素的邊界對整體偵測錯誤率的影響,最後將統計哪些特定發音方法或是哪些屬性分類最容易造成信任度量測的錯誤並且分析發生錯誤的原因。

4.1 中文發音方法偵測器對於各發音方法偵測之錯誤分析

4.1.1 MLP 偵測器容易偵測錯誤的發音方法類別

在 3.1 節當中統計完了 frame-based MLP 偵測器各種發音方法本身的偵測等錯誤率之後，我們接著對於造成各發音方法偵測錯誤的原因作分析,首先是觀察各種發音方法偵測器是否容易對特定某幾類發音方法發生偵測錯誤，下表 4.1 為各種發音方法之間彼此互相偵測的混淆矩陣。

說明: 表 4.1 中每一個比率數值為橫軸的發音方法類型音框(測試語料)全部拿來對於縱軸對齊的發音方法屬性偵測器作偵測而又被偵測為 target 的偵測錯誤比例(以表 4.1 中加了底線的數值 6.72 為例,這代表說所有 Fricative 音框當中有 6.72% 的音框會被 Vowel 偵測器偵測為 target):

$$\text{error rate} = \frac{\text{detect "fricative" as "vowel"}}{\text{total "fricative" frame}}$$

表 4.1：各發音方法之間互相偵測的混淆矩陣

Desired (%) Detector detected as target	vowel	fricative	Stop	nasal	liquid	affricate	silence
Vowel	91.71	6.72	11.71	20.10	36.34	4.83	1.63
Fricative	3.59	89.94	21.28	4.70	3.63	48.19	12.24
Stop	6.46	15.35	90.02	10.73	33.96	10.20	13.57
Nasal	12.92	4.31	7.14	90.75	28.94	1.84	4.08
Liquid	12.60	2.06	14.53	18.12	90.84	1.96	0.96
Affricate	2.68	50.80	16.46	2.11	3.40	90.84	9.03
Silence	1.85	12.74	22.02	4.81	1.45	15.65	94.28

由表 4.1 可以看出，Fricative 偵測器以及 Affricate 偵測器在互相偵測對方時，所造成的偵測錯誤率均在 50%左右，這是因為 Affricate 與 Fricative 這兩種發音方法之間聲學特徵非常的近似，這點我們可以如下圖畫出這兩類發音方法在 MFCC 參數當中 C1，C2 的分布[5]看出來:

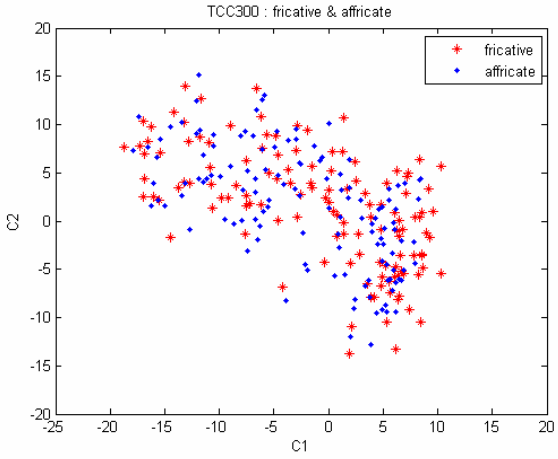


圖 4.1：Fricative 與 Affricate C1,C2 平均值分佈

從上圖中我們可以看到這兩類發音方法的特徵參數分佈幾乎是重疊在一起，因此有如此高的混淆偵測錯誤率，另外 Liquid 也有不小的比例會被 Nasal 或者 Vowel 偵測器偵測為 target，這可能是受到 Liquid 與 Nasal 之間有時會因為語者將ㄋㄌ這兩個音唸的含混不清[5]而造成的混淆，另外 Liquid 又容易與其所接的 Vowel 發生連音現象的影響，因此也造成了混淆的現象。

4.1.2 MLP 偵測器加入狀態轉移機率容易偵測錯誤的發音方法類別

得到 frame-based MLP 中文發音方法偵測器的結果以及各種發音方法之間互相混淆偵測的情形之後，另一方面同樣統計加上狀態轉移機率的發音方法偵測器與各個發音方法資料之間的偵測混淆情況:

表 4.2：各發音方法之間互相偵測的混淆矩陣

Desired (%) / Detector detected as target	Vowel	Fricative	Stop	Nasal	Liquid	Affricate	Silence
Vowel	92.07	7.38	11.86	17.84	*39.99*	5.97	1.14
Fricative	2.51	91.68	16.21	3.46	2.15	45.20	8.96
Stop	5.02	15.14	91.42	8.84	28.75	8.56	*14.93*
Nasal	12.21	3.84	6.30	91.70	26.44	1.65	2.86
Liquid	11.11	1.38	12.38	16.70	91.99	1.43	0.57
Affricate	1.89	*51.46*	9.83	1.74	1.45	91.93	7.76
Silence	2.18	11.15	18.46	*5.79*	0.61	12.42	94.61

由上表 4.1 與表 4.2 做比較，上圖中錯誤率明顯下降的以藍色及底線顯示，而錯誤率上升的較明顯的以斜體紅字顯示，其餘錯誤率變動不大的以黑體字顯示，我們可以清楚看到加上轉移機率之後偵測器大部分混淆的情況都獲得改善而

僅有四類的混淆情形稍微變差，因此整體而言偵測器的效能是向上提升的，不過最容易混淆的 Affricate 與 Fricative 之間混淆的程度依然非常嚴重，均有超過 45% 以上的混淆錯誤率。

4.2 中文發音方法偵測器各音素偵測之錯誤分析

4.2.1 MLP 偵測器偵測各個中文音素的錯誤警戒分析

前一節 frame-based MLP 偵測器架構中各個發音方法偵測器對語料庫中的各個發音方法作偵測後，分析各個發音方法偵測器對何種發音方法容易偵測錯誤，在本節中，我們將細分為各個發音方法偵測器對各個中文音素作偵測，觀察發音方法偵測器是否對某一發音方法的偵測錯誤是來自於所偵測的發音方法其所屬的某一或某群音素所造成。下表為各個中文發音方法 MLP 偵測器偵測各個中文音素的錯誤率排序。

說明：

下頁表 4.3 當中的每一個數值為對測試語料中為橫軸的音素類型音框全部拿來對於縱軸對齊的發音方法屬性偵測器作偵測而又被偵測為 target 的偵測錯誤比例(以表中加了底線的數值 64.97 為例，這代表所有的ㄑ /q/ 作偵測，Fricative 偵測器將其偵測為 target 的比例為 64.97%)。

$$\text{error rate} = \frac{\text{detect "ㄑ/q/ " as "fricative"}}{\text{total "ㄑ/q/" frame}}$$

表 4.3：中文發音方法 MLP 偵測器容易偵測錯誤的音素類別統計

Rank / Detector	1	2	3	4	5	6
vowel	ㄖ /r/ 42.84	ㄌ/l/ 31.78	/ng/ 22.02	/n_n/ 20.05	ㄅ /b/ 18.79	ㄋ /n/ 16.94
stop	ㄏ /h/ 57.18	ㄈ /f/ 48.32	ㄋ /n/ 37.49	ㄇ /m/ 37.31	ㄌ /l/ 35.81	ㄖ /r/ 31.32
fricative	ㄑ /q/ 64.97	ㄘ /c/ 51.59	ㄔ /ch/ 46.96	ㄗ /z/ 45.71	ㄓ/zh/ 42.77	ㄐ /j/ 41.85
affricate	ㄒ /x/ 66.50	ㄙ /s/ 65.80	ㄕ /sh/ 60.25	ㄊ /t/ 39.45	ㄈ /f/ 25.56	ㄎ /k/ 17.15
nasal	ㄌ /l/ 37.31	ㄥ /e_ng/ 29.63	FNULL2 24.96	ㄣ /e_n/ 24.75	ㄖ /r/ 22.23	ㄤ /a_ng/ 18.28
liquid	ㄋ /n/ 71.47	ㄇ /m/ 63.29	ㄩ /yu/ 30.49	ㄉ /d/ 29.94	ㄟ/ei / 16.35	ㄧ/yi / 15.14
silence	ㄈ /f/ 49.43	ㄊ /t/ 30.27	ㄆ /p/ 30.10	ㄘ /c/ 25.16	ㄎ /k/ 23.81	ㄅ /b/ 18.07

註: ng、n_n 為鼻音韻尾、FNULL2 為空韻母

從表 4.3 可以看出，以 Vowel 偵測器偵測中文音素錯誤率排序中，對於有聲的子音 Liquid 當中的/l/以及/r/音具有高偵測錯誤率外，其餘後四名中有三名容易偵測錯誤的音素都是屬於 nasal，符合在表 4.1 所得到結果，且對於 nasal 的高偵測錯誤率分佈在 /n_n/、/ng/ 此兩鼻音韻尾的音素，這兩類鼻音韻尾的音框數佔測試語料中 nasal 的音框數的 80%以上，除了已知 vowel 偵測器偵測 nasal 語料容易發生偵測錯誤的地方在於 vowel 與鼻音韻尾的交接處，從 nasal 偵測器來看也是如此，其偵測錯誤率排序的前幾項中有三項是屬於有鼻音韻尾的 vowel，然而還有一種類型的錯誤是由於連續語音當中鼻音韻尾時常沒被念出來的效應，我們將在下一節當中做討論，至於/b/這個音也有 18%的偵測錯誤，我認為是由於/b/這個音的音長普遍非常短(平均音長在第二章表 2.4 有統計)而 HMM 強迫切割對於音長太短的切割本來就有缺陷，會將一部分的母音切進來，造成在自動調整子音與母音交界的切割位置時子音的模型訓練的不好因此沒辦法調整到理想的位置。

對於 Stop 偵測器而言，偵測錯誤率最高的是對屬於舌葉不上提的 Fricative：ㄏ/h/與ㄈ/f/音素，錯誤率為 57.18%及 48.32%。而對於屬於 Nasal 的子音ㄋ/n/，ㄇ/m/以及 Liquid 的ㄖ/r/與ㄌ/l/，偵測錯誤率為 31~37%，偵測效果亦差。

至於 Liquid 的部份，Nasal 的ㄋ/n/與ㄇ/m/的偵測錯誤率都超過 60%，推測可能是因為語者往往唸的不清楚的原因所致，另外對於 Silence 偵測結果而言，Fricative 當中的ㄈ/f/偵測錯誤率最高的原因在之前的第三章的觀察當中已經提過因此不再贅述，另外無聲的 Stop 音ㄊ/t/、ㄆ/p/、ㄎ/k/偵測錯誤率也有 20~30%。

4.2.2 MLP + transition probability 偵測器偵測各中文音素的錯誤分析

底下我們將對於偵測器加上狀態轉移機率之後的偵測結果，同樣去分析觀察發音方法偵測器是否對某一發音方法的偵測錯誤是來自於所偵測的發音方法其所屬的某一或某群音素所造成，各種發音方法偵測器偵測錯誤的前幾名音素統計排行如下:

表 4.4：MLP 偵測器加上轉移機率後容易偵測錯誤音素類別統計

Rank / Detector	1	2	3	4	5	6
vowel	ㄖ /r/ 45.85	ㄌ/l/ 36.02	/ng/ 19.31	ㄅ /b/ 18.94	/n_n/ 18.94	ㄋ /n/ 18.17
stop	ㄏ /h/ 58.50	ㄈ /f/ 49.08	ㄇ /m/ 32.83	ㄋ /n/ 32.46	ㄌ /l/ 29.77	ㄖ /r/ 27.25
fricative	ㄑ /q/ 66.87	ㄘ /c/ 50.11	ㄔ /ch/ 45.23	ㄗ /z/ 42.14	ㄓ/zh/ 36.84	ㄐ /j/ 35.40
affricate	ㄒ /x/ 70.40	ㄙ /s/ 67.89	ㄕ /sh/ 63.31	ㄊ /t/ 30.29	ㄈ /f/ 16.18	ㄎ /k/ 9.39
nasal	ㄥ /e_ng/ 34.68	ㄣ /e_n/ 30.22	ㄌ /l/ 30.18	FNULL2 23.84	ㄤ/a_ng/ 21.61	ㄖ /r/ 20.93
liquid	ㄋ /n/ 73.45	ㄇ /m/ 65.12	ㄩ /yu/ 30.27	ㄉ /d/ 27.90	ㄟ/ei / 14.18	ㄧ/yi / 13.73
silence	ㄈ /f/ 47.75	ㄆ /p/ 28.42	ㄊ /t/ 26.64	ㄘ /c/ 21.94	ㄎ /k/ 19.50	ㄅ /b/ 15.02

由表 4.3，4.4 的統計結果相比較，我們觀察到幾個現象，基本上兩種架構的偵測結果對於非常容易混淆的音素偵測錯誤率差異不大(如表中紅色數值)，但是對於加上狀態轉移矩陣的的偵測結果而言，錯誤率是有些微的上昇，這個原因是因為在容易混淆的音素類別音框在 MLP 偵測器 target model 上的分數大多呈現一整個區段的分數平均甚高，因此加上狀態轉移機率之後反而有些許錯誤擴大的現象，以下是一個實際的例子:

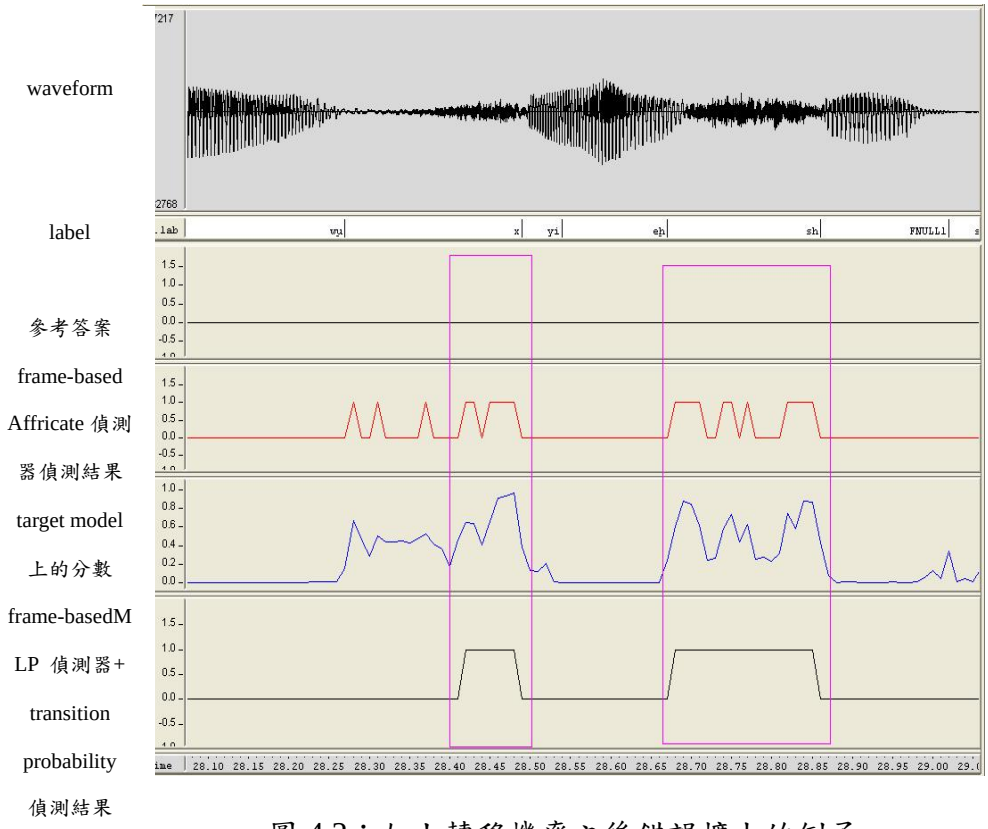


圖 4.2：加上轉移機率之後錯誤擴大的例子

上圖中用粉紅色框起來的區域就是 Fricative 音段容易造成 Affricate 偵測器偵測錯誤擴大的例子，我們可以看到由於整段區塊當中的平均分數甚高，因此再加上類似於 segment 概念的狀態轉移機率分數之後會造成整段都被偵測為 Affricate 的結果，造成容易混淆的音素偵測錯誤率稍有上升。

不過同時由於之前提到過的MLP偵測器加上轉移機率分數之後能夠移除大部分 jitter 類型的偵測錯誤，因此偵測器較不容易混淆的音素類別普遍偵測錯誤率都有下降，因此以下我們統計出各類發音方法偵測器音素偵測錯誤的 7~11 名:

表 4.5：加入轉移機率前後較不易混淆的音素偵測錯誤率變化比較

Detector \ Rank	7	8	9	10	11
Vowel	Frame-based MLP	ㄉ/d/ 13.18	ㄍ/g/ 12.84	ㄆ/p/ 12.80	ㄇ/m/ 11.56	ㄊ/t/ 7.72
	MLP + transition probability	ㄉ/d/ 13.69	ㄍ/g/ 11.49	ㄆ/p/ 11.32	ㄇ/m/ 11.60	ㄊ/t/ 8.28
Stop	Frame-based MLP	ㄔ/ch/ 19.95	ㄘ/c/ 15.33	ㄗ/z/ 12.11	ㄨ/wu/ 11.37	ㄓ/zh/ 9.11
	MLP + transition probability	ㄔ/ch/ 19.53	ㄘ/c/ 15.81	ㄗ/z/ 10.12	ㄨ/wu/ 10.32	ㄓ/zh/ 6.93
Fricative	Frame-based MLP	ㄎ/k/ 36.19	ㄆ/p/ 35.38	ㄊ/t/ 32.99	ㄍ/g/ 20.25	ㄅ/b/ 19.50
	MLP + transition probability	ㄆ/p/ 31.59	ㄎ/k/ 31.39	ㄊ/t/ 26.29	ㄍ/g/ 11.94	ㄅ/b/ 14.97
Affricate	Frame-based MLP	ㄆ /p/ 14.93	ㄉ /d/ 11.94	ㄏ /h/ 6.63	ㄅ /b/ 5.75	ㄧ /yi/ 4.80
	MLP + transition probability	ㄆ /p/ 7.42	ㄧ /yi/ 4.15	ㄉ /d/ 4.08	ㄩ /yu/ 3.37	ㄏ /h/ 1.37
Nasal	Frame-based MLP	ㄩ/yu/ 17.56	ㄟ/ei/ 17.24	FNULL1 16.81	ㄢ/a_n/ 13.63	ㄨ/wu/ 13.48
	MLP + transition probability	ㄢ/a_n/ 17.14	ㄩ/yu/ 16.19	ㄟ/ei/ 14.89	FNULL1 14.07	ㄨ/wu/ 11.79
Liquid	Frame-based MLP	ㄅ/b/ 13.23	n_n 12.58	ㄜ/e/ 12.55	ㄨ/wu/ 11.52	ㄢ/a_n/ 10.99
	MLP + transition probability	ㄜ/e/ 11.05	ㄅ/b/ 10.43	n_n 10.30	ㄢ/a_n/ 9.41	ㄨ/wu/ 8.90
Silence	Frame-based MLP	ㄍ/g/ 17.89	ㄗ/z/ 17.80	ㄉ/d/ 17.69	ㄓ/zh/ 15.81	ㄙ/s/ 14.50
	MLP + transition probability	ㄗ/z/ 19.44	ㄓ/zh/ 16.69	ㄍ/g/ 14.27	ㄉ/d/ 13.29	ㄙ/s/ 12.65

由上表可以看到，以 Affricate 偵測器為例子，原本偵測錯誤率還有 10%以上的ㄆ/p/與ㄉ/d/ 這兩種音素的 error reduction 超過 50%，而ㄏ/h/與ㄅ/b/的錯誤率更是從原本的 6%左右降到剩下不到 2%比 Affricate 之後常接的母音ㄧ/yi/跟ㄩ/yu/還低，因此整體而言 Affricate 偵測器的效能是向上提升的。其餘偵測器的效能除了ㄢ/a_n/對於鼻音偵測器以及ㄗ/z/對於 Silence 偵測器這兩種音素的偵測錯誤率稍有上升(如表中紅色數值標示)之外，除了一部分的音素偵測錯誤率變動不大(如表中黑色數值標示)之外，大部分的音素偵測錯誤率都有明顯的下降(如表中藍色數值標示)，由於偵測錯誤率降低的音素資料量明顯超過偵測錯誤率升高的音素資料量，因此整體而言加入狀態轉移機率後發音方法偵測器的混淆錯誤率可以說有顯著的下降。

4.3 中文連續語音當中連音現象造成屬性偵測錯誤的分析

前一小節當中我們已經對於容易偵測錯誤的音素作了統計以及分析，其中我們注意到了其實某一部份偵測的錯誤是由於鼻音化的母音與其後的鼻音韻尾之間由於在連續語音當中語者時常沒將鼻音韻尾唸出來造成偵測錯誤，此類錯誤直接造成 Vowel 的錯誤警戒率以及 Nasal 錯誤拒絕率上升，以下是個實際的例子:

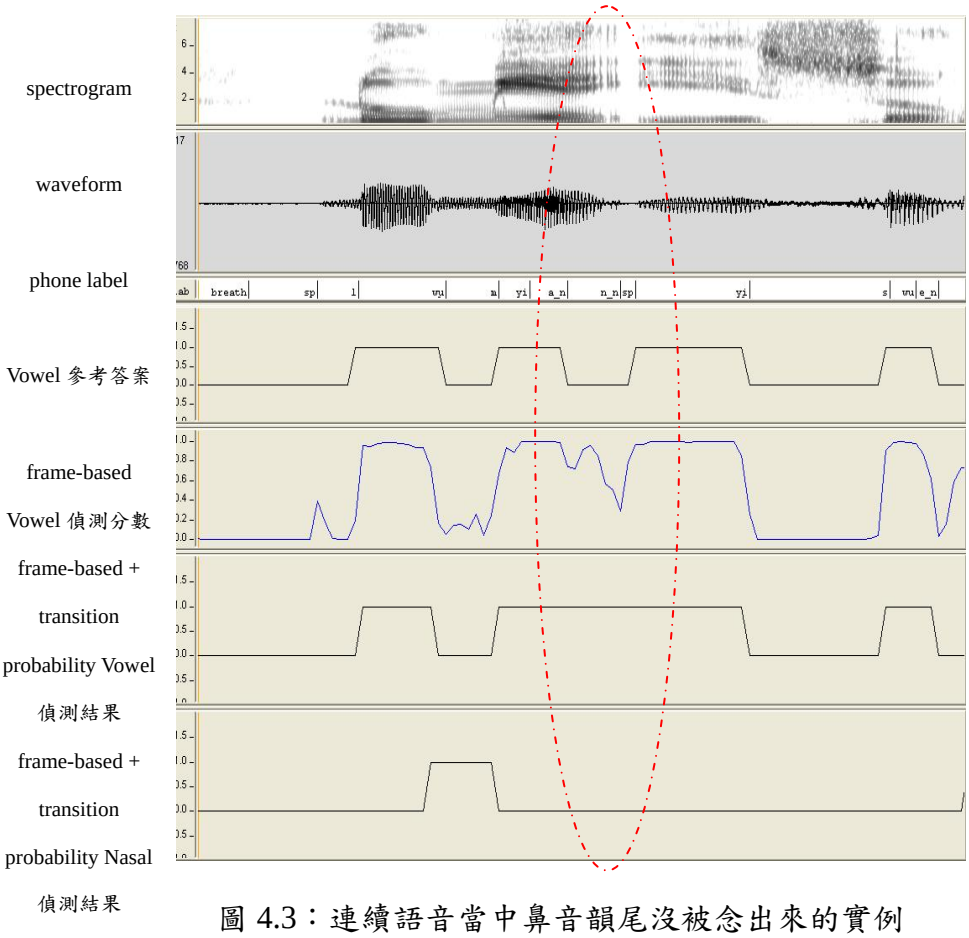


圖 4.3：連續語音當中鼻音韻尾沒被念出來的實例

由上圖的例子可以看出，在紅色虛線所示的位置標記檔標記是 Nasal，但是實際上語者並沒有完整的將鼻音韻尾唸出來，因此這段音框在 Vowel 的 target model 分數很高，而從頻譜上來看也極為相似 Vowel，如此一來，這個段落造成了對 Vowel 而言是錯誤警戒的偵測錯誤，而對於 Nasal 而言則是錯誤拒絕的偵測錯誤，底下將統計測試語料中整段鼻音都沒有被偵測出來的資料量:

表 4.6：測試語料整段鼻音韻尾 missing detection 的統計

Nasal events(counts)	9914
Nasal missing detecttion count(count)	642(6.48%)
Nasal frame(frames)	65639
Nasal missing detection rate(frame)	2563(3.90%)

由上面的統計資料可以看出，整段 Nasal 都沒偵測出來的音框數佔了所有 Nasal 音框的 3.90%，直接大大影響了 Nasal 的偵測錯誤拒絕率，底下我們將抽樣整段 miss detection 的 Nasal 音段約十分之一的資料量來檢查是否確實因為語者沒有完整的唸出鼻音韻尾而造成偵測器偵測錯誤:

Nasal 整段 missing detection events count:59

表 4.7：抽樣觀察整段鼻音韻尾 missing detection 的分佈統計

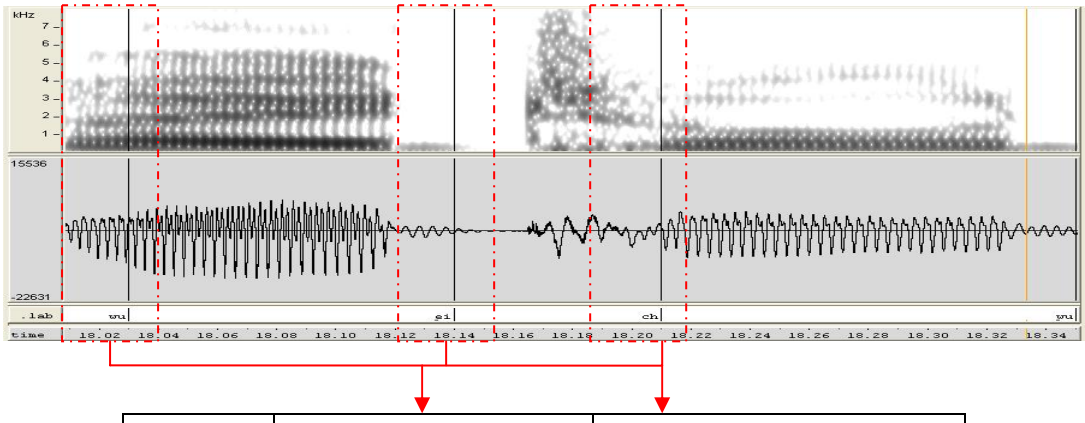
	counts	備註
Tatal events	59	
鼻音韻尾有唸出來	12(20%)	
鼻音韻尾沒唸出來	47(80%)	ㄢ(a_n+n_n):28 ㄥ(e_ng+ng):6 ㄣ(e_n+n_n):4 ㄤ(a_ng+ng):9

由上表統計中可以看出，未被偵測出來的鼻音韻尾取樣數當中約有 80%是確實語者未將鼻音韻尾清楚的唸出來，其中又以ㄢ/a_n+n_n/這類的鼻音化母音其後的鼻音韻尾在連續語音當中沒唸出來的情形最為常見，因此如果從切割一開始便考量到對於連音現象部份音素標記而做修正，對於屬性偵測結果必定能夠再進一步的提升。

4.4 音素邊界附近屬性偵測錯誤對整體偵測錯誤率的影響

邊界附近的音框由於處於音素轉變的過渡區間，因此容易同時被前後兩類發音方法所偵測出來，應該是造成整體偵測錯誤的主要來源之一，因此在本節當中我們將分別統計不同偵測器架構在音素邊界附近的音框偵測錯誤率以及音素邊界以外的音框偵測錯誤率。

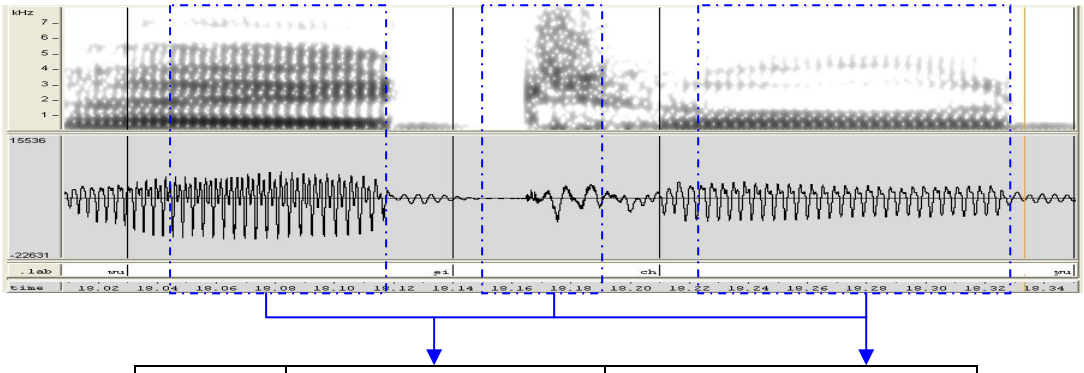
在下表 4.8 當中我們首先對於音素邊界前後音框對於 frame-based MLP 偵測器的偵測結果以及加上轉移機率之後的偵測結果做統計比較:



	Frame-based MLP	Frame-based MLP + transition probability
Vowel	FA：33.78% FR：15.24%	FA：32.95% FR：15.69%
Stop	FA：48.92% FR：10.03%	FA：52.75% FR：9.13%
Fricative	FA：24.31% FR：21.85%	FA：27.06% FR：19.95%
Affricate	FA：31.70% FR：14.53%	FA：32.48% FR：14.03%
Nasal	FA：43.29% FR：13.43%	FA：46.67% FR：12.88%
Liquid	FA：51.51% FR：13.51%	FA：55.40% FR：12.54%
Silence	FA：18.45% FR：23.95%	FA：20.35% FR：20.41%

表 4.8：音素邊界前後音框的偵測錯誤率

由上表的統計可以看到，邊界前後音框的錯誤警戒率或是錯誤拒絕率均明顯高於所有資料的等錯誤率甚多，證明邊界附近由於聲學特徵不分明容易造成偵測的錯誤，另外，加入狀態轉移機率之後的偵測結果在音素邊界前後音框普遍有錯誤警戒率稍高的現象，這是因為加入轉移機率之後偵測結果 segment 的長度通常會比實際的音長要長，這一點呼應了 3.2 節當中統計的結果，至於加入轉移機率在邊界錯誤拒絕率較低的原因是因為在音素邊界附近由於正處於過度區間聲學特徵較不分明，因此 MLP 偵測器在此區間的偵測結果時有 jitter 錯誤拒絕發生，而加入狀態轉移機率之後能夠有效的排除 jitter 類型的錯誤，因此造成音素邊界的錯誤拒絕率稍低。接著我們統計邊界附近以外的音框發生偵測錯誤的情形:



	Frame-based MLP	Frame-based MLP + transition probability
Vowel	FA：4.44% FR：5.36%	FA：4.16% FR：4.66%
Stop	FA：9.00% FR：9.85%	FA：7.30% FR：7.31%
Fricative	FA：9.35% FR：5.95%	FA：7.52% FR：4.26%
Affricate	FA：8.41% FR：5.86%	FA：7.25% FR：4.39%
Nasal	FA：7.59% FR：5.52%	FA：6.43% FR：4.21%
Liquid	FA：8.79% FR：4.75%	FA：7.59% FR：3.41%
Silence	FA：3.32% FR：1.80%	FA：2.58% FR：2.16%

表 4.9：音素邊界以外的音框偵測錯誤率

由表 4.9 統計的結果可以看出音素邊界以外的音框偵測錯誤警戒率及錯誤拒絕率均明顯低於等錯誤率，因此搭配上表 4.8 的統計結果比較之後我們可以得知，事實上發音方法偵測器在音素邊界的偵測效能非常不穩定，很容易發生偵測錯誤，而在音素邊界以外音框的偵測情形除了部分聲學特徵類似的發音方法之間容易混淆而造成偵測效能較差之外，大致上而言偵測的結果相對比較可靠，而此一分析的結果呼應了 3.3 節屬性偵測器信任度量測的分析當中，音素邊界附近的信任度普遍較低代表該區段偵測結果較不可靠，而大多數能夠提供給後級辨識器的有效偵測結果多半分佈在音素邊界以外的區域。

4.5 信任度量測錯誤的統計與分析

在前一章的第三節當中我們初步建立起屬性偵測的信任度量測，對於語音屬性偵測有了評量偵測結果可靠度的指標，也統計了在各種門檻值下，各階層偵測的可靠度以及偵測錯誤率的情形，並且對於可靠度高低的分布情形做初步的分析，在本節當中我們將會對於偵測錯誤的原因作較詳細的分析。以下我們首先統計在各階層下各屬性分類偵測錯誤的分佈情形(門檻值為 0.9):

表 4.10：各階層屬性分類信任度偵測錯誤分佈

Layer-1		Layer-2		Layer-3		Layer-4	
Class	Error rate(%)	Class	Error rate(%)	Class	Error rate(%)	Class	Error rate(%)
Vowel	1.21	Vowel & Nasal	0.70	Sonorant	2.20	Speech	10.48
Nasal	6.63						
Liquid	17.95	Vowel & Liquid	1.52				
Fricative	4.20	Fricative & Affricate	4.40	Non-sonorant	23.93		
Affricate	5.73						
Stop	14.43	Stop	15.48				
Silence	7.67	Silence	9.38	Silence	10.29	Non-Speech	1.94
Breath	3.99	Breath	7.38	Breath	4.86		

由上表的統計我們可以看到，Liquid 以及 Stop 這兩類的屬性信任度偵測錯誤率偏高，因此我們先針對這兩類屬性信任度錯誤的情形作分析，首先如果是一整段都發生偵測錯誤的情形:

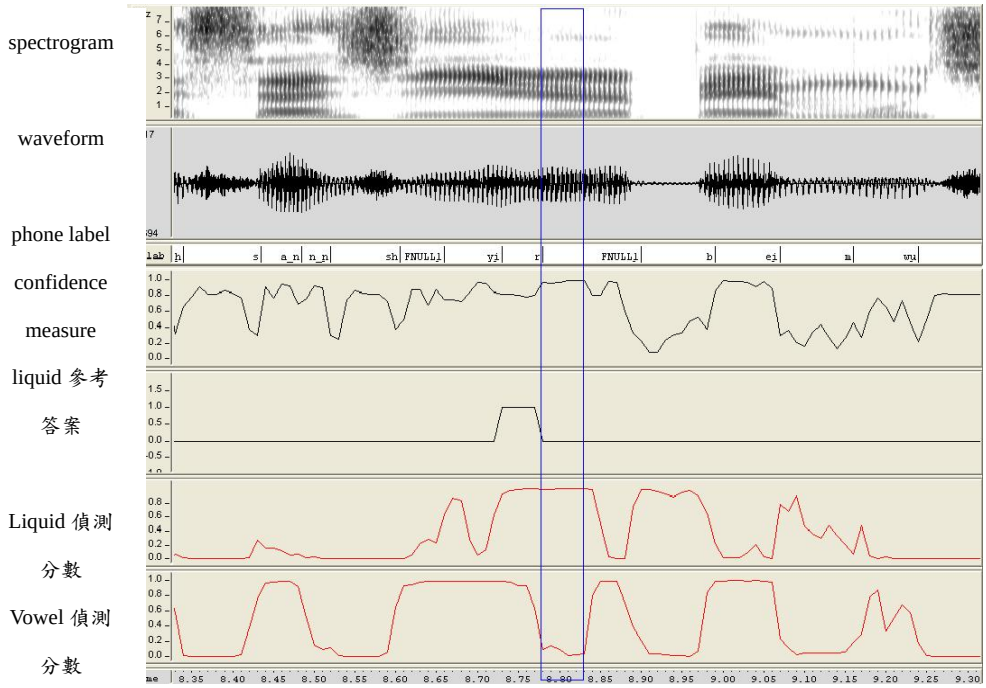


圖 4.4：一整段偵測錯誤的實例

由上圖我們可以看到藍色區塊標示一整段的信任度都很高，同時 Liquid 偵測器的分數也很高，但是觀察波形以及頻譜上來看，事實上ㄖ/r/與其後的空母音的聲學特性非常的類似沒有明顯的分界，造成了切割位置不理想，因此造成了偵測的錯誤。另一種錯誤情況是由於連音的現象，造成ㄉ/d/唸成類似ㄌ/l/的音而造成少許音框偵測錯誤:

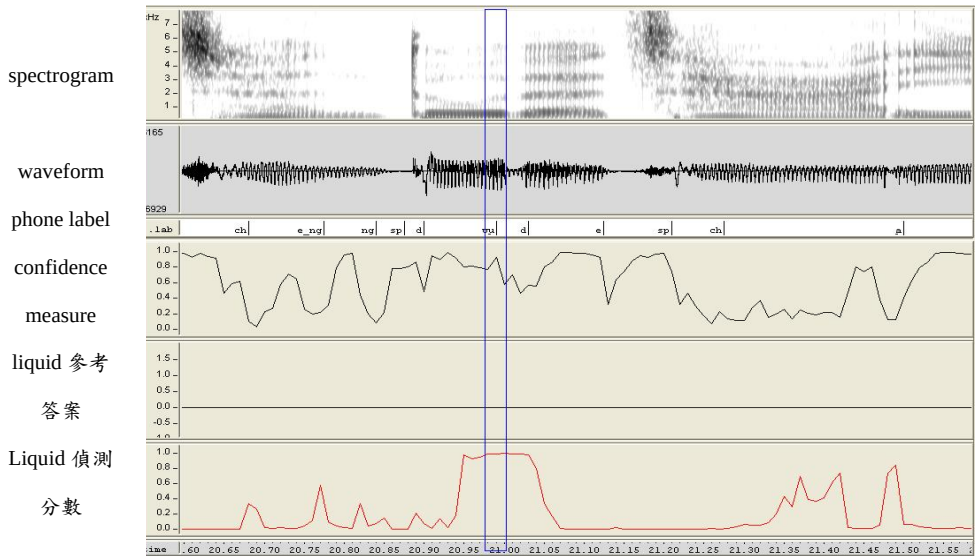


圖 4.5：連音現象造成偵測錯誤的實例

此外對於音長非常短的 Stop 音段，邊界附近也有極少數音框的偵測錯誤情

形:

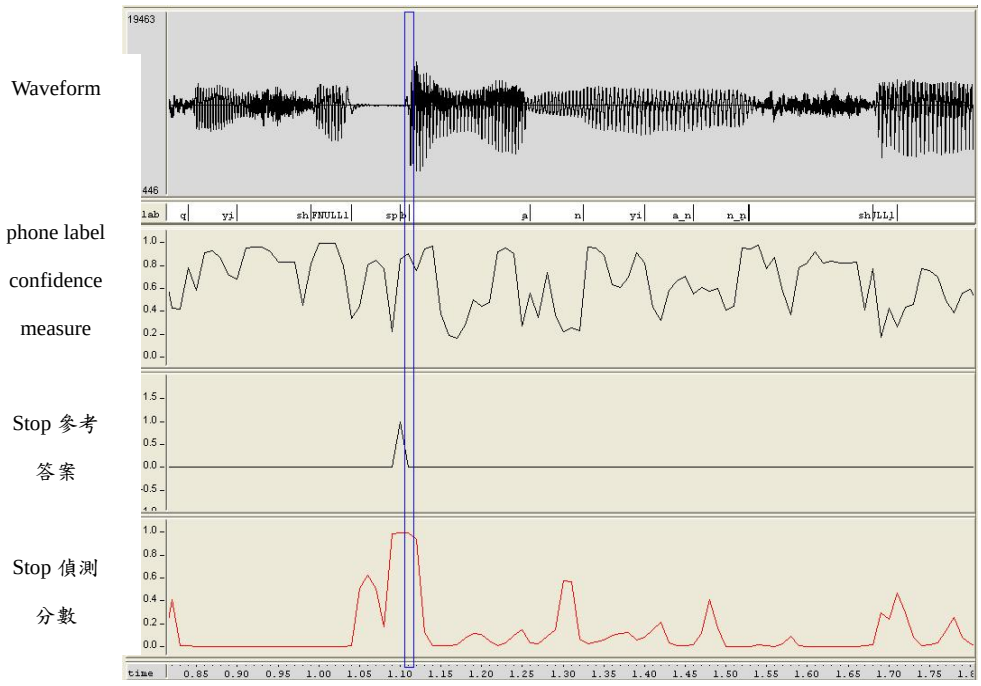


圖 4.6：Stop 音邊界單一音框偵測錯誤的實例

不過這幾類型的錯誤次數對於整體資料量來說非常少，因此對於整體錯誤率影響不大。另外 Silence 的信任度錯誤率也稍高，主要是因為仍舊有部份音節間的 short pause 沒有被切出來或是切的太短，以下是個實際的例子:

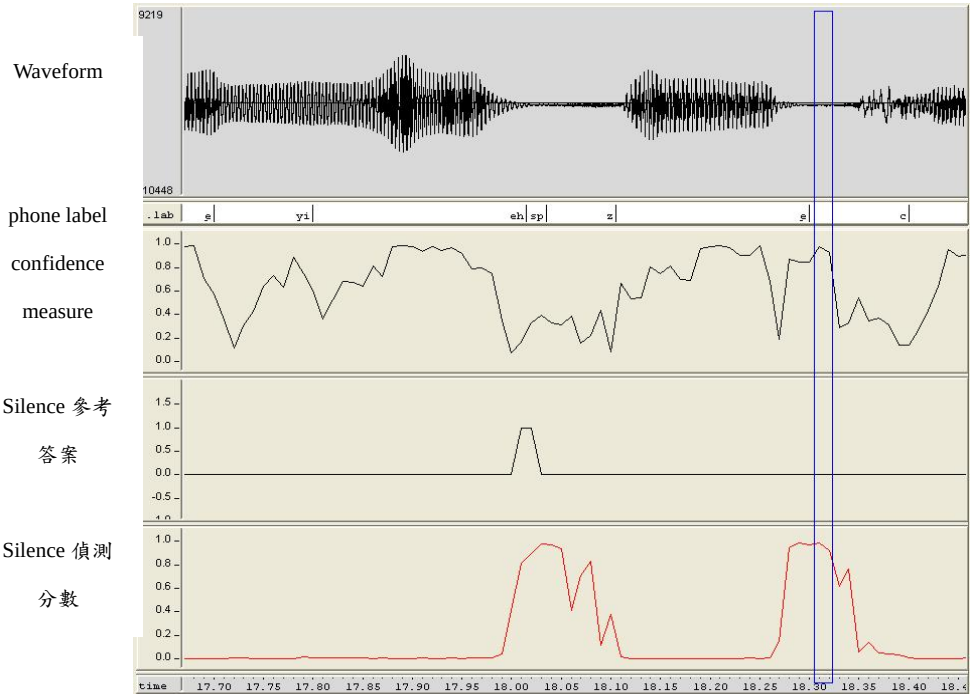


圖 4.7：short pause 沒切出來造成偵測錯誤的實例

綜合以上的結果，我們認為主要造成信任度的偵測錯誤的原因來自沒有人工音素切割位置，雖然已經用自動調整的方式將語料庫中大部分自動切割明顯不理想的切割位置作修正，但是仍舊有少部份的切割偏差情形影響了信任度量測的錯誤率，另外一個影響信任度量測錯誤率的原因就是在錄製語料時語者講話速度過快或是唸的不清楚也同樣會使信任度量測的效能降低。

第五章 結論與未來展望

5.1 結論

在本論文中，首先在中文語料庫沒有良好的切割位置情況下，我們考慮了以中文音節的切割位置起始，在音節切割位置內 flat-start 訓練音素的隱藏式馬可夫模型來對語料庫求取音素的初始切割位置，接著使用 Segmental Kmeans Segmentation Algorithm 來自動調整音素切割位置以製作中文語音屬性偵測器。我們以高斯混合模型為基礎製作中文語音屬性貝氏偵測器，並將其效能當作是後面所要製作的中文發音方法偵測器的 baseline，接著訓練非線性 MLP 模型為基礎的發音方法偵測器，以此方法所製作出來的中文語音屬性偵測器其偵測效能在與基本系統相比後的確有明顯的改善，之後在 MLP 偵測器當中加入狀態轉移機率，便能更進一步提升偵測器的效能，最後引入信任度量測的概念對於偵測器的結果作可靠程度的評比。

在中文語音屬性偵測器的效能與錯誤分析中，發音方法的摩擦音(fricative)與塞擦音(affricate)，因為其兩者聲學特性相近，而導致在互相偵測時非常容易產生混淆，而同屬於響音(Sonorant)的母音(Vowel)也與鼻音(Nasal)及流音(Liquid)這兩類有部份混淆的情況。另外，音素交界為一個模糊的地帶，容易造成前後兩種偵測器的錯誤偵測。除此之外，雖然調整過後的切割位置已經相當近似於人工切割，但是仍舊有少部分的切割差距造成了偵測器效能的降低。

5.2 未來展望

在中文語音屬性偵測器的部份，仍然有很多路線可以去探索及改進，無論是針對所要偵測的語者來做調適，或者是針對某些特定發音方法求取具有鑑別性的

語音特徵參數，抑或是採用不同的偵測架構，比如說對前後音框資訊有記憶特性的遞迴式類神經網路(RNN)為基礎的偵測器。本論文除了訓練最基本的以音框為基礎(frame-based)的中文語音屬性偵測器之外，也加入以段落為基礎(segment-based)的資訊與音框為基礎的偵測器做整合，同時對於音框為基礎的偵測結果進行信任度量測的評比，以期提供後級辨識器更可靠的語音屬性資訊。並且以實驗分析的方式，獲得在偵測中文語音中音素、發音方法或發音位置之間交互的影響，以提供後人在對中文語音偵測時的參考。希望藉由這些經驗、知識的累積，建立一個以知識為基礎(knowledge-based)加上資料驅動(dara-driven)的新一代語音辨識系統架構，以推進語音辨識能力的突破。

參考文獻

【1】 C.-H. Lee, “From knowledge-ignorant to knowledge-rich modeling：A new speech research paradigm for next generation automatic speech recognition” Proc. ICSLP2004, Keynote speech, 2004

【2】 Sérgio Paulo，Luís C. Oliveira，“Automatic Phonetic Alignment and Its Confidence Measures”, Advances in Natural Language Processing,Vol.3230, pages 36-44,2004.

【3】 Jinsong Zhang, Keikichi Hirose “Tone nucleus modeling for Chinese lexical tone recognition” , Speech Communication 42(2004) pages447-466.

【4】 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。

【5】 許見徨，“中文語音屬性偵測之研究”，交通大學電信工程所，中華民國九十六年八月。

【6】 C.-H. Lee, “A Study on Separation between Acoustic Models and Its Applications,” Proc. ICASSP2005

【7】 S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey,V. Valtchev, P. Woodland, ”The HTK Book (for HTK Version 3.3)”, Cambridge University, 2005

【8】 R. P. Lippmann, L C. Kukolich, and E. Singer, “LNKnet: Neural Network, Machine Learning, and Statistical Software for Pattern Classification”, Lincoln Laboratory Journal, vol. 6, pp. 249-268, 1993.

【9】 Erhan Mengusoglu, Christophe Ris,” Use of Acoustic Prior Information for Confidence Measure in ASR application ”,TCT Lab , Mons , Belgium , Eurospeech 2001-Scandinavia.

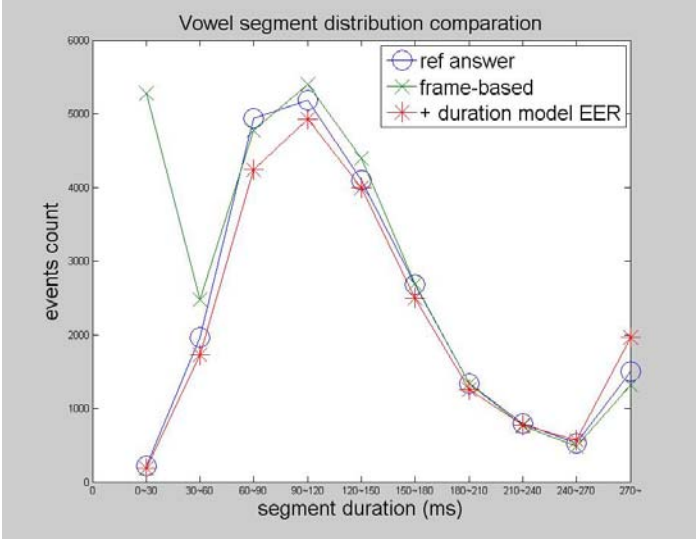
【10】Bilmes J.A., "A Gentle Tutorial of the EM algorithm and its application to

Parameter Estimation for Gaussian Mixture and Hidden Markov Models", ICSI-Technical Report-97-021, 1997.

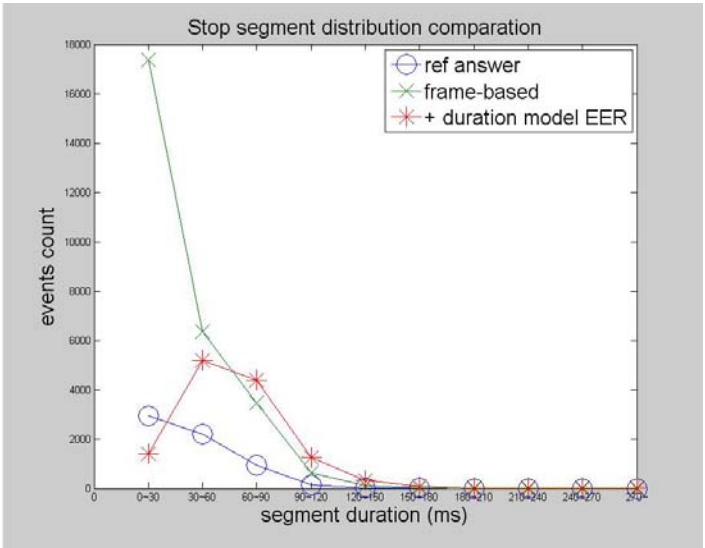
附錄一

加入轉移機率 MLP 偵測器等錯誤率下音段長度分佈

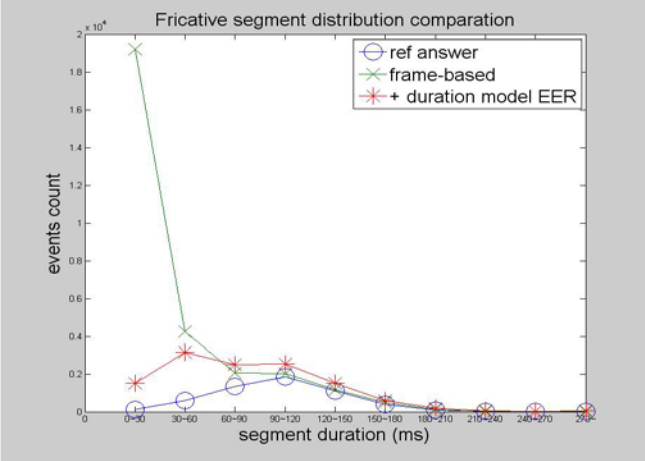
1. Vowel 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)



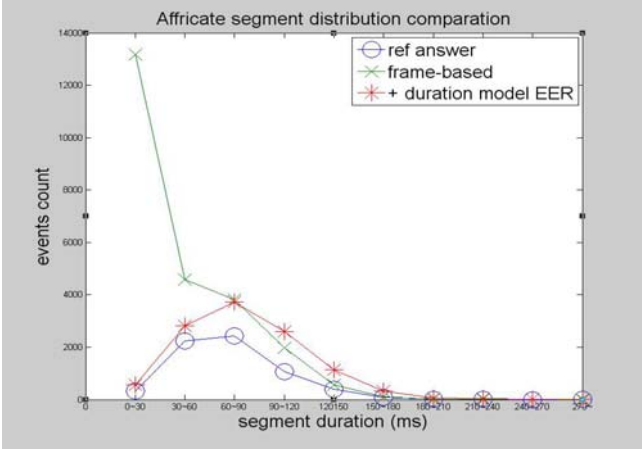
2. Stop 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)



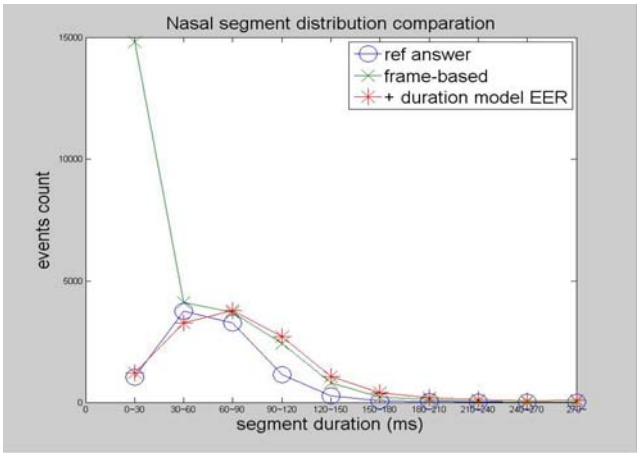
3. Fricative 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)



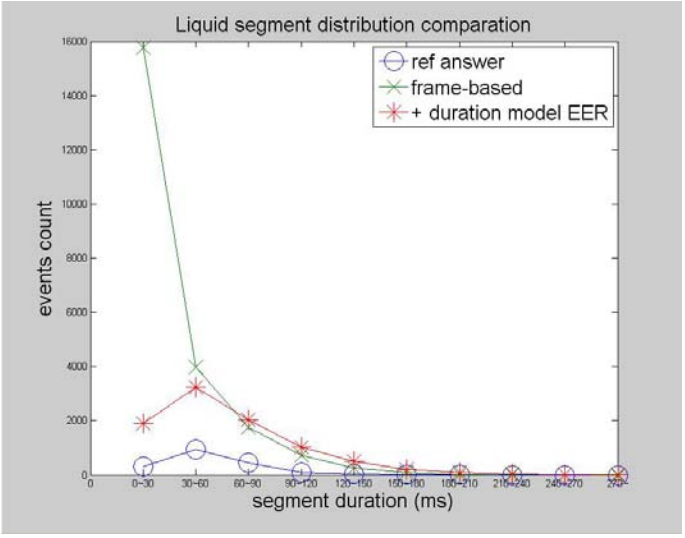
4. Affricate 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)



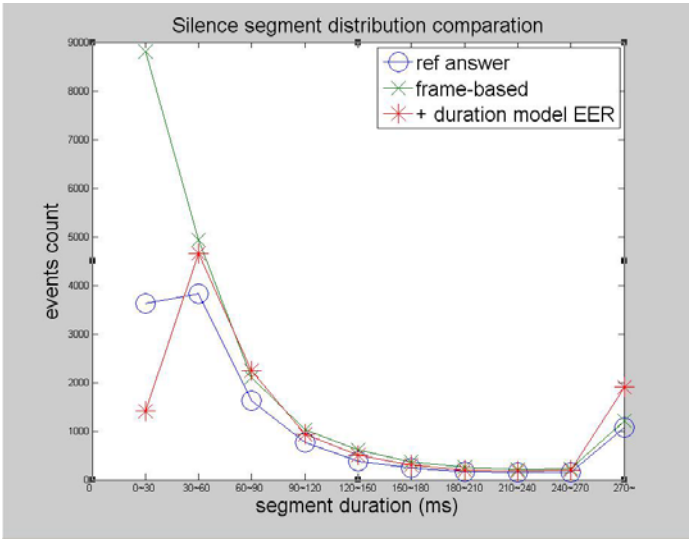
5. Nasal 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)



6. Liquid 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)



7. Silence 段落音長分佈比較(藍色為參考答案音長分布，綠色為 frame-based 偵測結果，紅色為加上轉移機率的偵測結果)



附錄二

中文音素分類及漢拼、注音對照表

表一　21 類聲母表

編號	注音	漢拼		編號	注音	漢拼		編號	注音	漢拼
1	ㄅ	b		9	ㄍ	g		17	ㄕ	sh
2	ㄆ	p		10	ㄎ	k		18	ㄖ	r
3	ㄇ	m		11	ㄏ	h		19	ㄗ	z
4	ㄈ	f		12	ㄐ	j		20	ㄘ	c
5	ㄉ	d		13	ㄑ	q		21	ㄙ	s
6	ㄊ	t		14	ㄒ	x				
7	ㄋ	n		15	ㄓ	zh				
8	ㄌ	l		16	ㄔ	ch				

表二　16 類韻母表

編號	注音	漢拼		編號	注音	漢拼
1	ㄚ	a		9	ㄢ	a_n
2	ㄛ	o		10	ㄣ	e_n
3	ㄜ	e		11	ㄤ	a_ng
4	ㄝ	eh		12	ㄥ	e_ng
5	ㄞ	ai		13	ㄧ	yi
6	ㄟ	ei		14	ㄨ	wu
7	ㄠ	ao		15	ㄩ	yu
8	ㄡ	ou		16	ㄦ	er

Ps.實際“ㄢ”　“ㄣ”　“ㄤ”　“ㄥ”的漢拼分別為

“an”　“en”　“ang”　“eng”

在此我們將細分至鼻音韻尾，因此做些改變

表三　　空母音　　與　　鼻音韻尾

編號	符號		編號	符號
1	FNULL1		1	n_n
2	FNULL2		2	ng

Ps.其中 n_n 為”ㄢ”與”ㄣ”的鼻音韻尾，ng 為”ㄤ”與”ㄥ”的鼻音韻尾