國立交通大學

電信工程研究所

碩士論文

使用加權有限狀態轉換器之漢語語音辨認系統

A Mandarin Speech Recognition System Using Weighted Finite-State Transducer

研究生：林昂星

指導教授：王逸如　教授

中華民國 一百零一 年 七 月

使用加權有限狀態轉換器之漢語語音辨認系統

A Mandarin Speech Recognition System Using Weighted Finite-State Transducer

研 究 生： 林昂星　　Student：Ang-Hsing Lin

指導教授： 王逸如 博士　　Advisor：Dr. Yih-Ru Wang

國 立 交 通 大 學

電信工程研究所

碩 士 論 文

A Thesis

Submitted to Institute of Communication Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in Communication Engineering

July 2012

Hsinchu, Republic of China

中 華 民 國　一百零一 年　七 月

使用加權有限狀態轉換器之漢語語音辨認系統

研 究 生：林昂星　　指導教授：王逸如 博士

國立交通大學電信工程研究所碩士班

中文摘要

本論文主要探討如何使用加權有限狀態轉換器來建構漢語語音辨認系統。首先介紹加權有限狀態轉換器的相關演算法，以及不同層級的語音模型如何以有限狀態機圖形來表示，並整合成漢語語音辨認系統。從語音辨認實驗結果中提出詞辨認錯誤約 55%因 OOV Words 引起，經統計，一個 OOV 詞平均造成 2.4 個詞辨認錯誤。為了降低 OOV words 在漢語語音辨認系統中所造成詞辨認錯誤的影響，經統計結果顯示 OOV words 中，人名約佔了 30%，其中三字中文人名(姓氏+名字)約佔了 23%，故我們引入階層式語言模型的概念，訓練人名模型來輔助降低詞錯誤率。

測試語料採用包含朗讀式長句之 TCC300 語料庫實驗。使用 HTK 兩階段辨識，詞錯誤率為 13.76%，使用加權有限狀態機 RT 為 13 可達到相同的錯誤率，辨認速度比傳統 HTK 辨認快約 15 倍。另一方面，在語言模型層建構出 OOVs 人名模型置入語音辨識系統，並有效地降低詞錯誤率約 0.12%。

A Mandarin Speech Recognition System Using Weighted Finite-State Transducer

Student：Ang-Hsing Lin Advisor：Dr. Yih-Ru Wang

Institute of Communication Engineering
National Chiao Tung University

Abstract

This study focuses on how to use a Weighted Finite-State Transducer (WFST) to construct a Mandarin Speech Recognition System (MSRS). It first introduces algorithms for WFST, as well as different levels of speech model to represent the Finite-State Machine (FSM) graph, and integrates into the MSRS. The experimental results indentify the Word Error Rate (WER) at about 55% is related to the appearance of OOV words, and statistics shows that one OOV word results in 2.4 words error averagely in MSRS. According to the statistical results, it shows that the names OOV words accounts for about 30%, and in which three words Chinese names accounts for about 23%, In order to reduce the negative impact of the OOV words results in the MSRS, we introduce a hierarchical language model, training name model to assist lower WER.

The test corpus uses for the read-type long sentences TCC300 corpus. The 13.76% WER is obtained by using HTK two-stage recognition, while use of WFST RT=13 can achieve the same WER, the recognition speed is about 15 times faster than the traditional HTK. Besides, we construct OOVs names model in the language model layer and placed in the MSRS, this effectively reduces the WER at about 0.12%.

致謝

本編論文順利完成，必須感謝陳信宏老師與王逸如老師的教誨與指導。兩位老師以各自的觀點帶給自己兩年來充實的研究所生涯，也讓這篇論文有了不同的生命。感謝兩位老師親切且不失嚴謹的態度讓我從懵懂無知的大學生轉變成具有面對未知領域不懼挑戰的企圖心，讓我成為一個好的研究生。

其次感謝希群學長讓我學習如何去攥寫程式，並感謝性獸學長給我不同的觀點，感謝隨時帶有微笑的合哥，感謝阿德學長讓我學會如何去看第一篇論文，因為你們的指導造就了現在的我，雖然這兩年的過程中遇到了大風大浪，但很開心自己能浪子回頭，一路走來。感謝這兩年一起奮鬥的好夥伴們，你們都是最優秀的，一切盡在不言中，也希望下一屆學弟妹們可以好好加油，707 實驗室靠你們了。

最後要謝謝我的家人，一路上給予支持，讓我無後顧之憂可以做自己，我很愛妳們。

目錄

表目錄

圖目錄

第一章 緒論

1.1 研究動機

今日科技發展日新月異，因應人類對科技產品實用性的追求，網路的快速發展推動人與人之間的資訊交流，大幅影響了現今人類的生活模式。聲音是人類最直接的溝通方式，語音辨識的發展逐漸成為人類與機器之間溝通的媒介，且語音的應用領域相當廣泛，因此發展語音辨識技術成為相當重要的課題。

大詞彙連續語音辨識(Large Vocabulary Continuous Speech Recognition, LVCSR)已發展數十年，對中文語音辨識而言，由於中文的“詞” (Word)的邊界相當模糊，造成詞彙的定義困難，受限於詞典的詞條數限制，詞典無法收錄所有的詞條數，產生詞典外詞彙(out-of-vocabulary, OOV)過多，影響了辨識結果。由於某些類別的詞可任意組合，如:人名、數量複合詞、詞綴構詞等，考慮這些類別的特性，進行更深一步的分析探討，可能改進上述缺點。

近年來，有限狀態機(Finite State Machine, FSM)廣泛地應用在語音辨識領域中，藉由有限狀態機的各式演算法，我們將各個語音模型整合，並對辨識網路進行最佳化而得到良好的辨識速度。在本研究中，我們針對規則性的詞類進行分解，提高詞典的涵蓋率，以及定義詞彙的模糊邊界，提升語言模型的效能，並提出使用階層式的語言模型對中文人名進行處理，希望藉此提升辨識成效。

1.2 文獻回顧

有限狀態機(Finite State Machine)是一種簡單而有效率的數學模型，近年來廣泛的使用在語音辨識中。最早由 AT&T 實驗室的莫氏(M. Mohri)等人[1]-[3]提出使用加權有限狀態機(Weighted Finite State Machine, WFSM)，在狀態間的轉移上賦予一個加權值，利用加權值將語音辨認中最重要的機率分數整合至有限狀態機

之中。我們將語音辨認系統獨立的三個部份:聲學模型、發音辭典、語言模型，透過有限狀態機整合成一個單一的有限狀態機，並且運用了莫氏等人提出的確定化（determinization）以及最小化（minimization）演算法[1]來除去辨認網路中的冗贅路徑。經過確定化的有限狀態轉換器可以在搜尋時減少存活的狀態數，最小化演算能求得狀態數最少的等價有限狀態轉換器；加權推移（weight pushing）演算法[4]的實現，則讓語言模型的分數可以提早利用，修剪掉不必要的辨認路徑。台灣大學的余氏[5]與交通大學的姜氏[6]曾以有限狀態自動機實作過中文大詞彙連續語音辨認，他們的論文核心在敘述有限狀態機的基本定義、建構流程，如何以有限狀態機建構一套大詞彙連續中文語音辨認系統。並在實驗證明：有限狀態機比起傳統演算法，在相同辨識率下可以減少數倍辨識時間。

在大詞彙語音辨認系統中，N 連語言模型(*n*-gram language model)[7]最常被使用到，此模型以統計的方式來描述詞與詞之間相接的機率，但隨著 N 值提升，我們無法收集到所有 N 連詞彙的組合。而後有許多學者提出方法來加強語言模型。1992 年 Brown 等人提出類別式 N 連語言模型(class-based *n*-gram language model)[8]，加入了類別資訊來訓練語言模型，將詞彙依照特性分群，則資料的預估由詞彙組合數降低為類別的組合數，能夠改善資料稀疏的問題。

傳統在中文大詞彙辨認中所用的詞典，大多以語料中詞的詞頻排序取詞頻較高的詞納入詞典中。由於中文構詞的多元與彈性，詞典無法收錄所有詞彙，使得不在詞典裡的詞無法辨認出。周氏[9]在其論文中提出階層式的辨認系統，針對中文構詞最為彈性的人名、定量複合詞與詞綴三個類別，以構詞學的角度出發，依照各種詞類的特性將之拆解，以較少數量的構詞單元收錄以提升詞的涵蓋率。許氏[10]在其論文中提出將 OOV 人名視為單一類別建立人名語言模型並整合到辨認系統中，並以兩階段式辨認系統架構實現。先以第一級 LM 辨認產生混合 word 與 sub-word 構詞單元之 word lattice，在第二級加入不同詞類的語言模型重新配置其語言模型的分數加強 lattice 上的路徑而得到最後的辨認結果。

1.3 研究方向

本論文針對中文詞彙中的定量複合詞、綴詞進行拆解，藉以提升詞典涵蓋率；由於詞彙的邊界定義模糊，使得訓練語料的斷詞結果不一致，大幅下降語言模型的精確度，因此本研究對這部分加重處理，從選詞到訓練語料的修正，進行了若干的處理，希望藉此改良語言模型，以提升辨識率；另一方面針對 OOV 中的中文人名進行相關研究，利用有限狀態機將不同模型整合在一起，希望能建構出一個具實用性的大詞彙辨認系統。

1.4 章節概要說明

本論文一共分為五章，其各章節內容分配如下：

第一章：緒論

第二章：使用加權有限狀態轉換器之語音辨認系統

第三章：使用加權有限狀態轉換器實現語音辨認與分析

第四章：階層式語言模型實驗結果與分析

第五章：結論與未來展望

第二章　使用加權有限狀態轉換器之語音辨認系統

傳統的語音辨認系統使用 HTK tool 實現語音辨認，其辨認速度非常慢，且受限於 HTK tool 詞典大小，導致在大詞彙語音辨認的相關研究無法順利進行。近年來加權有限狀態轉換器(Weighted Finite-State Transducer, WFST)廣泛應用於大詞彙語音辨認系統，突破了眾多的限制及加快了語音辨認速度。

本章介紹使用加權有限狀態轉換器於大詞彙語音辨認系統中，語音辨認系統包含聲學模型、語言模型與發音詞典三部份。本研究的聲學模型是使用 TCC-300 語料庫建立，以隱藏式馬可夫模型呈現，用以描述發音過程的狀態轉移現象和輸出結果；語言模型則由大量文字語料庫先經過前級文字處理再以 *n*-gram 方式訓練來得到詞與詞之間相接的機率，由此來幫助語音辨識；在詞典挑選時，我們引入資料檢索的 TF-IDF 概念從較大詞典中剔除不常被使用的詞，再以詞頻排序選取本研究所使用的發音詞典。

本章節共分為五個部份：2.1 介紹有限狀態機相關演算法簡介；2.2 介紹聲學模型之建立；2.3 介紹語言模型之建立；2.4 介紹 TF-IDF 詞典選詞；2.5 介紹有限狀態機辨認系統整合。

2.1　加權有限狀態轉換器之相關演算法簡介

本小節簡單介紹加權有限狀態轉換器，及其演算法使用於語音辨認系統，包括組合演算法與取代演算法。

2.1.1　加權有限狀態轉換器

加權有限狀態轉換器可視為一個邊上帶有輸入輸出字元的有向圖，在 WFST

的圖形中，我們將點(node)稱為狀態(state)，邊(arc)稱為轉移(transition)，邊上帶有該轉移的輸入字元(input symbol)、輸出字元(output symbol)與權重(weight)。參照圖 2.1 所示，以粗線圈代表初始狀態(initial state)，雙線圈表示終止狀態(final state)。假如同時為初始與終止狀態則以雙粗線圈表示。

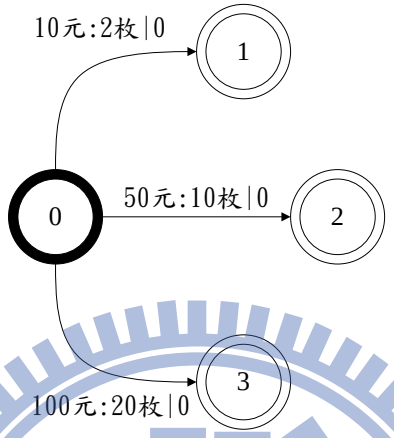

圖 2.1: 兌幣機之有限狀態機

每個有限狀態機，基本上皆由六個元素(Q, I, F,Σ,Δ,δ)所構成，其中：

1.) Q：所有狀態的集合，在此範例中含初始狀態與結束狀態，共有 4 個狀態。在圖 2.1 的例子中，Q={0, 1, 2, 3}

2.) I：初始狀態，表示有限狀態機的唯一初始狀態。I={0}

3.) F：終止狀態，表示有限狀態機結束的狀態，至少要含有一個以上的終止狀態。

F={1, 2, 3}

4.) Σ：所有可接受的輸入字元。Σ={10 元, 50 元, 100 元}

5.) Δ：輸出字元集。Δ={2 枚, 10 枚, 20 枚}

6.) δ：轉移函式。表示某來源狀態(source state)接受輸入字元後，會轉移到哪一個目標狀態(destination state)。

除上述之基本定義之外，還有一些專有名詞的解釋也一併在此論述：

1.) 狀態：

WFST 含有有限數量個狀態，這些狀態中必須有一個初始狀態與一個以上的

結束狀態。一開始由初始狀態出發，接受輸入字元序列後，經過一連串的狀態轉移，當最後一個轉移完成後，若停留在終止狀態，表示此條路徑是可接受(accept)的；反之則拒絕輸出(reject)。

2.) 轉移：

狀態與狀態間的轉移由轉移函式 δ 所定義，每個轉移需帶有來源狀態 s[t]、目的狀態 d[t]、輸入字元 i[t]、輸出字元 o[t]與該轉移的權重 w[t]。描述一個轉移時寫作(I:O|W)，I 表示 input symbol、O 表示 output symbol、斜線後的數值表示 weight。

3.) 空轉移：

我們允許轉移上的輸入與輸出字元為 ε(epsilon)。當輸入字元為 ε 時，表示不需要輸入就可以轉移到下一個狀態；當輸出字元為 ε 時，表示經過此轉移不會輸出字元。在設計 WFST 的 graph 時，會藉由空轉移來表示圖形上的特性。

4.) 路徑：

路徑(path)由一連串相連的轉移所組成，令 $P = p_1 p_2 \ldots p_n$ 為一條路徑，p_i 表示路徑上第 i 個轉移(i=1…n)，又 $s[p_{i-1}]=d[p_i]$，一條被接受的路徑 P 之結束狀態為 $d[p_n]$。

5.) 加權值：

在描述語音辨識所用之 WFST 的圖形時，利用加權值來表示各種模型的分數，除了在轉移上會帶有權重之外，每個結束狀態也可以再賦予加權值。在設計 WFST 的圖形時，一般採用 log semi-ring 的數學模型。此時對機率的轉移取 negative nature log，則尋找最佳路徑時為搜尋累積加權值最小的路徑。

2.1.2 組合演算法

給定兩個有限狀態機 *A* 與 *B*，將 *A* 的輸出字元作為 *B* 的輸入字元，進而將 *A*、*B* 整合成一個新的有限狀態機 *C*，寫作 *C*=*A*。*B*，每個 C 的狀態、轉移都是由 A 跟 B 的狀態與轉移所組成，並且只留下可成功走完的路徑。

範例如下：

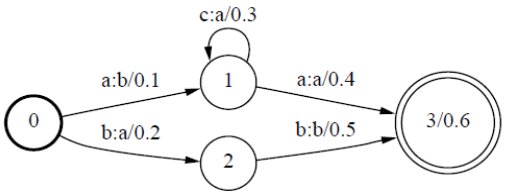

圖 2.2: 有限狀態機 A

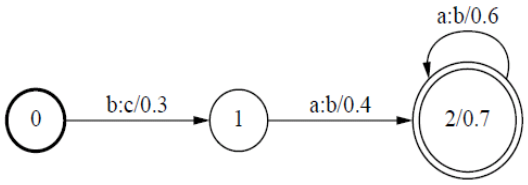

圖 2.3: 有限狀態機 B

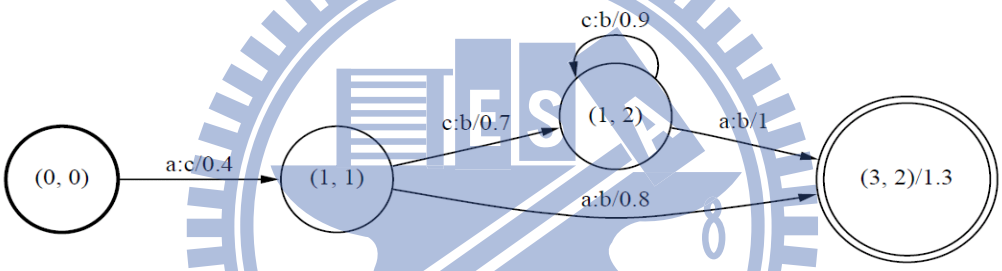

圖 2.4: 有限狀態機 C=A。B

執行組合演算法時，有限狀態機 A 下方的路徑因為其輸出字元 a 無法與狀態機 B 的輸入字元 b 相接，因此不會出現在有限狀態機 C 之中。

利用組合演算法的特性，就可以把不同層級(AM、Lexicon、LM)的有限狀態機全部整合成一個網路，由於不同層級的有限狀態機是獨立製作，想要隨意更換哪一層的架構都很方便。

2.1.3 取代演算法

取代演算法用來將一個有限狀態轉換器的轉移取代為另一個有限狀態轉換器。精確地說：一個從狀態 s 到狀態 d 的轉移上帶有輸出符號 n，我們欲用有限狀態轉換器 F 取代該轉移。取代演算法會先將此輸出符號 n 換為 ε，接上這個有

限狀態轉換器 F，再把 F 的結束狀態(Final state)接到原先的狀態 d。此演算法可以很好地應用在混合語言模型上，這點會在後述的研究提到。

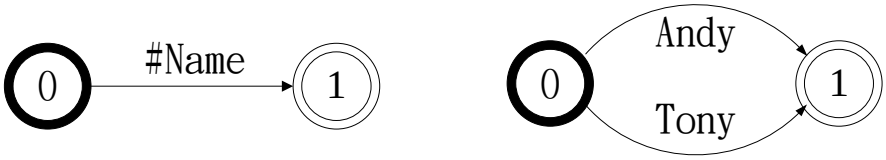

圖 2.5: 有限狀態機 A(左)與 B(右)

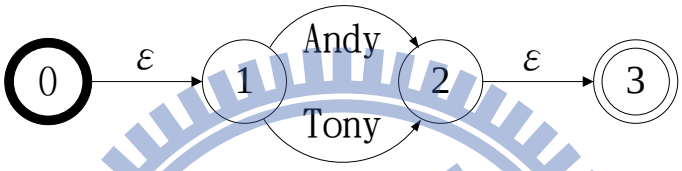

圖 2.6: 以 B 取代 A 上含有#Name label 之 arc

2.2 聲學模型之建立

本小節先介紹訓練聲學模型的語料庫內容，接著介紹建立聲學模型的相關參數設定。

2.2.1 語料庫簡介

在本研究中使用 TCC-300 麥克風語音資料庫，此資料庫由國立台灣大學、國立成功大學及國立交通大學的 300 位同學共同錄製，屬於麥克風朗讀語音，檔案統計資料如表 2.1 所示。語句取樣頻率皆為 16000 赫茲（Hertz），取樣位元數為 16 位元。音檔檔頭為 4096 位元組（byte），副檔名為*.vat。將此語料庫再區分為訓練語料及測試語料，訓練語料的部分約占 90%，共 274 位語者，長度共約 23 小時，測試語料的部分約 10%，共 29 位語者，長度約 2.43 小時。在進行辨識時，所使用的測試語料為交通大學與成功大學的長句音檔，共 19 位語者

226 句長句音檔，長度約 2 小時，詞總數量為 15493，每個句子平均含有 117.2 個音節。

表 2.1: TCC-300 語料庫統計表

學校名稱	文章屬性	語者總數		總音節數		音檔總數	
台灣大學	短文	男	50	男	27541	男	3425
		女	50	女	24677	女	3084
		總數	100	總數	52218	總數	6590
交通大學	長文	男	50	男	75059	男	622
		女	50	女	73555	女	616
		總數	100	總數	148614	總數	1238
成功大學	長文	男	50	男	63127	男	588
		女	50	女	68749	女	582
		總數	100	總數	131876	總數	1170

2.2.2 聲學模型的建立

在語音辨識中，需對輸入語音抽取出語音參數，考量到人耳聽覺效應的補償作用與短時間穩定特性，在本研究中使用 MFCC 參數(Mel-Frequency Ceptral Coefficients，梅爾倒頻譜參數)進行抽取與訓練。它的成分包含 12 維 MFCC 加上 1 維能量共 13 維，並取其 Delta 和 Delta-Delta term 用以描述參數變化訊息，最後可得共 39 維參數。本次實驗訓練的模型為中文單音節(mono-syllable)模型一共 411 個音節，每音節使用 8 個狀態(state)的隱藏式馬可夫模型(HMM)表示之，並使用 HTK 中之 MMI 鑑別性訓練得到。

訓練相關設定如下表：

表 2.2: MFCC 參數抽取設定檔

Frame size	32ms
Frame shift	10ms
Filter bank number	24
Sampling frequency	16kHz
Pre-emphasis Filter	First order with coefficient 0.97

2.3 語言模型之建立

本小節首先介紹訓練語言模型所使用的文字語料庫；接著介紹針對語料庫分析與前處理的步驟；再介紹部份歧異的詞彙針對形、音、義三部份的處理方式；最後介紹本研究使用 *n*-gram 計算詞與詞之間的機率建立語言模型。

在訓練語言模型之前，我們針對語料庫的文本進行若干處理，大致上分為：斷詞、文字正規化...等，之後再以統計方式得到訓練用的詞典，利用處理過後的語料進行語言模型訓練。

訓練流程如下：

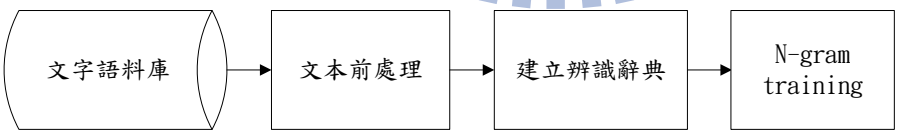

圖 2.7: 訓練語言模型流程示意圖

2.3.1 文字語料庫簡介

用於訓練語言模型的文字資料庫共有以下來源：

1.) 光華雜誌(Sinorama)：內容為一般雜誌的文章，蒐集的資料年代範圍介於 1976 年到 2000 年之間。

2.) NTCIR：為一個建立資訊檢索系統的標竿測試集，其內容由數種不同學科領域文章構成。

3.) 中研院平衡語料庫(Sinica)：它是一套由中研院收集，內容包含多種主題，以語言分析研究為目的的資料庫。

4.) Chinese Gigaword：由 Linguistic Data Consortium (LDC)整合發行，內容包含台灣中央社、北京新華社等國際新聞。

將文字資料庫進行文本前處理後，可得到詞數約 $3.82*10^8$，經由 TF-IDF 選詞挑選出六萬詞以供語言模型訓練所用，佔文字資料庫約 97.38%，OOV 詞條約佔 2.62%，平均詞長為 2.43 個字。以下各小節會陸續介紹相關的處理步驟。

2.3.2 文本前處理

文本前處理的步驟細分為以下幾個步驟：

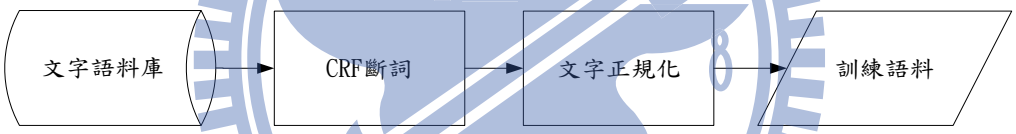

圖 2.8: 文本前處理示意圖

1.) CRF 斷詞:

以條件隨機域(conditional random field, CRF)[11]方法進行斷詞，此法主要藉由標記詞性與學習句法結構來進行斷詞，相較於傳統以詞典為基礎之長詞優先的斷詞規則，使用 CRF 斷詞可產生較正確的辨認結果，也能將詞典未收錄的詞正確斷出，一方面減少 OOV 所帶來的連續短詞串問題，另一方面也能擴充人名詞、詞綴詞等清單。

本研究使用由王逸如老師所撰寫的斷詞器，訓練語料庫為中研院平衡語料庫[12], 約 1.1×10^7 words，F-measure 為 97.14%。

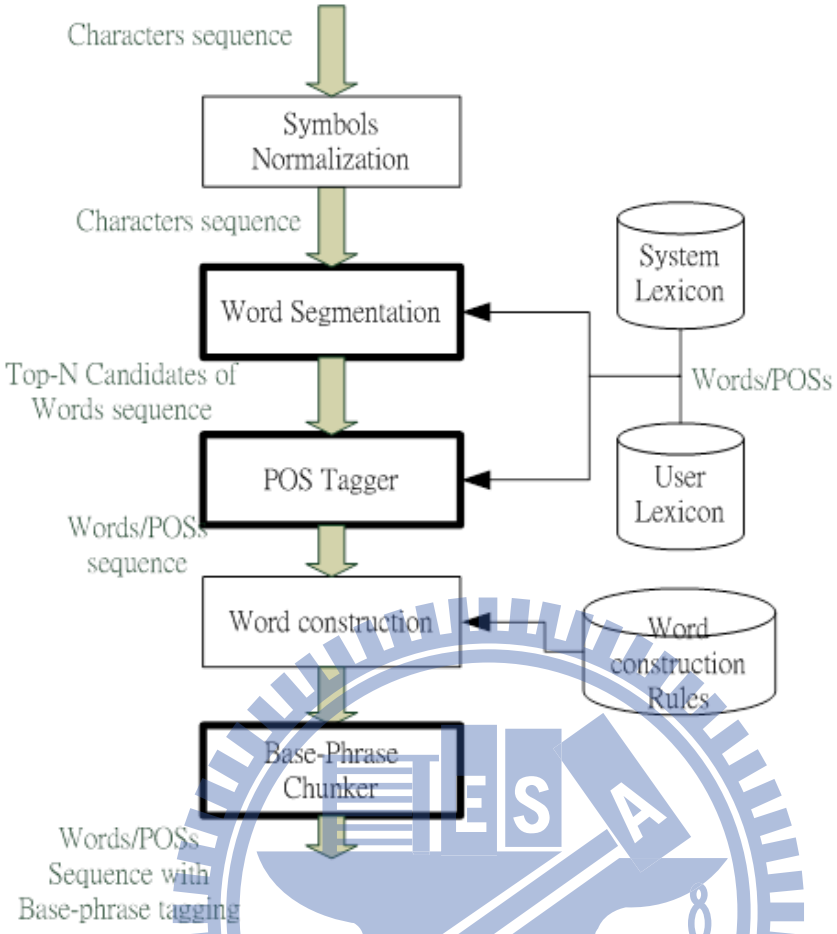

圖 2.9: 斷詞器訓練流程示意圖

2.) 文字正規化處理:

2.1) 文字格式正規化:

由於文字語料庫所收納的文字相當廣泛，在 CRF 斷詞結果中部份數字以及量詞單位等詞必須轉換成同一格式，符合大眾口語化的發音文字，例如:廿(ㄋㄧㄚˋ)，大多數人都會念成(二十)，在訓練語言模型前對文字語料庫做了數步驟的前處理。

2.2) 部份數量詞切短:

在 CRF 斷詞結果中屬於 Neqa、Neu 詞性的詞皆為長詞，由於詞典的容量有限，無法將其全部收錄於詞典中，我們將若干長詞切短，以短詞的形式收錄於詞典中，如此也能增加詞典的涵蓋率。

例如:

表 2.3: 文本前處理部分範例

未處理前的詞	處理後的詞
50%_Neqa	百分之_Neqa 五十_Neu
X 分之 Y_Neqa	X_Neu 分之_Neqa Y_Neu
一兆兩千三百四十五億_Neu	一兆_Neu 兩千_Neu 三百_Neu 四十_Neu 五億_Neu

2.3) 形音義分合詞處理

中文辨識因應形音義的異別，有些詞類其實是可以合併訓練，例如異體字在發音上與語意上皆相同，僅有字形不同，若視為不同詞彙看待會在辨認實造成混淆，在下面的章節會介紹對於各式漢字特質的處理方法。

2.4) 標點符號、POS 標記、英文詞串處理

中文共有十六種標點符號(PM)，其中又可分為標號與點號兩類，而點號與說話的停頓有較大的關聯性。我們藉由點號中的句號、驚嘆號、問號、分號將文章分段，並將除此之外所有的標點符號移除，並一併將 POS 標記也移除。

本實驗的辨認目標為中文詞彙，故將文章中的英文詞以「FW」標記為同一個類別，FW 類別並沒有收錄進訓練詞典中，而是視為一個 OOV 對待。

2.3.3 形音義分合詞處理

在大詞彙中文辨認的課題上，會因為字形、字音、字義三者的關係與分合情況而影響到訓練與辨識，以下將簡介漢字的特質，並就各式特質提出我們在處理語料時所應對的方法。

漢字具有的三大要素為：「形、音、義」，其中**字義**為我們語文的核心，字形、字音都是為字義而存在。在文化的演進中，有些字形變得不一致、或因沒有創製而借用，各種複雜的因素使得漢字形成了「多形、歧音、異義」的狀況，所以目前

所使用的「漢字」呈現出字形不一、字音分歧、字義寬廣的特質。

1.)　字形不一

歷史上漢字有甲骨、金文、篆、隸、楷、行、草等不同形體，如今使用者也有簡體／繁體的差別。也存在有結構上同字但異形的差別，例如足夠的「夠」一字也有人寫作「够」、人群的「群」寫作「羣」。在字形不一的情況下，影響到的是斷詞器的訓練、斷詞詞頻統計、詞典收錄、語言模型統計...等。

2.)　字音分歧

一字多音一向是漢語的特色，當中音變而意思不同者俗稱破音字。例如「便（ㄆㄧㄢˊ）宜」、「方便（ㄅㄧㄢˋ）」。字音的分歧所影響的是詞典收錄，就破音字意義不同的層面來看，也影響了語言模型的訓練（尤其指單字詞的情況）。

3.)　字義寬廣

在漢字中有一字多義的情況存在，相同的字形可同時代表不同意義，如「稀少（不多）」、「少（年輕）年」、「少（丟失）了東西」。

表 2.4: 漢字形音義異同表

形	音	義	現象	範例	處理
同	同	異	多義字	挨（靠近、順著、擠...）	—
同	異	同	又讀字	角（ㄐㄧㄠˇ／ㄐㄩㄝˊ）色	O
異	同	同	異體字	群／羣、夠／够	O
異	異	同	同義字	足／腳、頭／首	※
異	同	異	同音字	ㄍㄨㄥ（工、公、...）	—
同	異	異	破音字	藏（ㄗㄤˋ／ㄘㄤˊ）	O

在語音辨認的課題中，「多義字」、「同音字」的異別可以經由 *n*-gram 語言模型學習到，而「又讀字」可用 multi-pronunciation 形式收錄在發音辭典中，額外

處理的三個項目為：

A.) 異體字：

由於僅有字形不同，若將異體字雙雙收錄在辭典中，僅會瓜分掉原本該有的機率，我們將其轉寫為同一字後才進行語言模型的訓練。延伸的狀況是，異體字落在一個詞內時（如：人蔘、人參），在轉寫文字時應以詞為單位進行。

B.) 同義字：

針對單字詞的情況，同義字是不該被合併訓練的，儘管「足」跟「腳」係屬同義字，但前後文通常存有差異，故不對單字詞同義字進行處理。延伸的情況為同義詞(variant word)，所指為語義相同但字形不盡相同的詞，我們希望同義詞能在語言模型中共享相同的分數。

例如：在訓練語料中有 [跑得] [越來越／愈來愈] [快] 這兩類詞出現，而「越來越」與「愈來愈」是一組 variant word，在訓練 n-gram LM 模型的時候應將他們視為一樣的 word 進行訓練。但由於「越來越」與「愈來愈」的發音不同，在我們將 Grammar (Language model)層向下展開到 Lexicon 層之前，必須將這些 variant word 的資訊補回 Language model 上。

由於在訓練語言模型前，已將每組同義詞合併訓練，故在訓練好的 Language Model 上只有被合併的詞，所以在 Grammar 層找出被合併的詞的 Arc，這條 Arc 上帶有輸入字元、輸出字元以及 Weight，只要將這條 Arc 複製並將輸入輸出的資訊取代成另一個同義詞補回，如此這組同義詞就能享有相同的語言模型分數。

表 2.5: variant word 對照表(部分節錄)

Variant Word Pair		Variant Word Pair	
身臨其境	身歷其境	飛短流長	蜚短流長
來歷不明	來路不明	鼎鼎有名	赫赫有名
危在旦夕	命在旦夕	憤恨不平	憤憤不平
…	…	…	…

C.) 破音字：

同形異音異義的「破音字」以及同形異音同義的「又讀字」，這些同形歧音(multi-pronunciation)的問題，傳統上是在詞典中收錄兩種發音來進行，但儘管我們可以將破音字都收入辭典中，卻無從得知文本中該詞正確發音，因為文本的斷詞結果是沒有標記上音節的，在訓練語言模型時等同將所有的破音字合併在一起訓練。

例如:「供給」的「給」是一個又讀字，可以念成ㄍㄟˇ或ㄐㄧˇ。

而我們收錄字典時並未有 Syllable 的資訊，在訓練語言模型時，我們將其視為同一個詞，但在辨認時必須將這組破音字的資訊補回。處理這部分的方式跟同義詞雷同，同義詞是在 Grammar 層補回資訊，而又讀字的處理方式是在 Lexicon 層補回。

2.3.4 TF-IDF 選詞

TF-IDF (term frequency–inverse document frequency) 是一種用於資訊檢索(IR - Information Retrieval)的常用加權技術。它是一種統計方法，用於評估一個詞對於一個文件集或一個語料庫中的其中一份文件的重要程度。例如我們在搜尋引擎輸入"*Wiki*"，那麼會出現一堆包含" *Wiki* "四字的網頁，在這麼多網頁中，如何排列出那些最能代表" *Wiki* "四字的網頁？直覺的作法，不外乎先把全世界所有的網頁掃描一次，然後計算" *Wiki* "這四個字在每一個網頁出現的次數。出現頻率愈高，表示該網頁和" *Wiki* "這四個字愈有相關性，這個頻率參數便為 TF。

從另一個角度來看，如果有一串字在每個網頁都出現很多次，是不是表示這個字串沒什麼重要性？例如搜尋" *a book*"，照理第一個網頁應該是某個含有" *a* "字的英文網頁，因為" *a* "必定比" *book* "更易出現。實則不然，因為每個英文網頁大概都有" *a* "這個字，如此反而讓它的重要性降低。

針對我們訓練的語料庫，可以使用下列算式算出每個 word 對應的 IDF 值：

$$idf_i = \log \frac{|D|}{|\{d : d \ni t_i\}|}$$

其中 D 表所有文件的集合，分子 $|D|$ 表示語料庫中的文件總數，d 表文件，t_i 表正在處理的詞，分母則表示包含該詞 t_i 的文件數目。

進行大詞彙辨認時，我們欲收錄的詞是一般常見的詞語，也就是說：必須找到廣泛出現在各個文章中的詞。由於“出現的文章數”在分母項，因此我們將挑選 IDF 分數低的詞收錄進詞典中。

我們將訓練所用的文字語料庫取出所有詞條數共有 $1.76*10^6$，因詞典收錄的詞條數有限無法全數收錄，所以會造成大量的 OOV 詞出現，故選詞的方法會影響語言模型的效能。

本研究首先以詞頻排序選取十萬詞為候選詞，藉由 IDF 的方法對候選詞重新計算一組 IDF 值，再對其重新排序剔除只有在特定主題文章才會出現的詞，將剩下的詞再以詞頻重新排序選取六萬詞為我們所用的字典，以六萬詞的大小限制來看，以這樣的方式選詞比僅以詞頻選詞約更動了 3300 個詞。

表 2.6: IDF 排序剔除的詞(部分節錄)

Word	詞頻	文件數	IDF 值
駱明慧	890	63	2.124162517
李文秀	885	6	3.145351817
黃兆能	822	42	2.300253777
洪曉慧	774	61	2.138173232
楊斌	681	61	2.138173232
尼奧	657	21	2.601283772

由表 2.5 我們可以觀察出若以詞頻排序這些詞會被納入詞典中，而這些詞大部分都是人名以及特定的文章才會出現的詞，所以將這些詞剔除不納入詞典已經

達到我們選取詞典的目的。

2.3.5 語言模型之訓練

在本研究中使用了日前運用廣泛的 *n*-gram 語言模型，語言模型是用來預估一個詞串的出現機率，此模型假設任一個詞在詞串中只受到前 *n-1* 個詞的影響。令 $W = w_1 w_2 ... w_N$ 為一個 N 詞長的詞串，藉由前述的假設，第 k 個詞所出現的機率表示為 $P(w_k \mid w_{k-n+1} w_{k-n+2} ... w_{k-1})$，這個 N 詞長的 W 詞串之出現機率可展開為：

$$P(W) = P(w_1) \cdot P(w_2 \mid w_1) ... P(w_i \mid w_{i-n+1} ... w_{i-1}) ... P(w_N \mid w_{N-n+1} ... w_{N-1}) \quad (2.1)，$$

$$P(w_i \mid w_{i-n+1}, ..., w_{i-1}) = \frac{Count(w_{i-n+1}, ..., w_i)}{Count(w_{i-n+1}, ..., w_{i-1})} \quad (2.2)，$$

由於 *n*-gram 語言模型是統計式的模型，如果訓練語料中沒出現該詞語組合，就無法預估其機率，且隨著 n 值上升，所需的訓練語料也呈指數成長。為了解決這些問題，我們以後撤平滑化(back-off smoothing)來調整模型的機率分佈。當詞串 $w_{i-n+1} ... w_{i-1}$ 不存在時，我們丟棄距離最遠的詞的資訊，以低一階的 $w_{i-n+2} ... w_{i-1}$ 機率乘上後撤加權值 α 預估之，寫為 $\alpha(w_{i-n+1} w_{i-n+2} ... w_{i-1}) \cdot P(w_i \mid w_{i-n+2} w_{i-n+3} ... w_{i-1})$。若也沒有 $P(w_i \mid w_{i-n+2} w_{i-n+3} ... w_{i-1})$ 的資訊，繼續後撤並逐一乘上後撤加權值。改寫機率預估式如下：

$$P(w_i \mid w_{i-n+1}, ..., w_{i-1})$$

$$= \begin{cases} a(w_{i-n+1}, ..., w_{i-1}) P(w_i \mid w_{i-n+2}, ..., w_{i-1}) & , Count(w_{i-n+1}, ..., w_i) = 0 \\ d_a \cdot \dfrac{Count(w_{i-n+1}, ..., w_i)}{Count(w_{i-n+1}, ..., w_{i-1})} & , 1 \le Count(w_{i-n+1}, ..., w_i) \le k \\ \dfrac{Count(w_{i-n+1}, ..., w_i)}{Count(w_{i-n+1}, ..., w_{i-1})} & , Count(w_{i-n+1}, ..., w_i) > k \end{cases} \quad (2.3)，$$

後撤加權值 $a(w_{i-n+1}, ..., w_{i-1})$ 需經過正規化(normalization)處理，並滿足條件式：

$$\sum_{w \in V} P(w_i = w \mid w_{i-n+1}, ..., w_{i-1}) = 1 \quad (2.4)，$$

另外，當 *Count*(·) 的數值很小時，可能造成預估的不準確性，因此不信任此預估機率，而是以 d_a (Discount Coefficient Factor)來進行平滑化。當一詞串組合的出現次數小於某設定的次數時，我們將原始預估的 *n*-gram 機率乘上 d_a 值，d_a 依據 Good-Turning discounting 計算得出，並會將 discounting 扣除的機率值再平分給詞串沒有出現的 *n*-gram 機率使用。

2.4 加權有限狀態轉換器之整合

我們首先使用有限狀態機建立出目前已知的語音模型系統，其中包含：聲學模型、詞典以及語言模型，並透過有限狀態轉換器將此三部分 Compose 成一個巨大的搜尋網路，最後再以確定化、最小化等演算法對這個搜尋網路進行最佳化的動作。

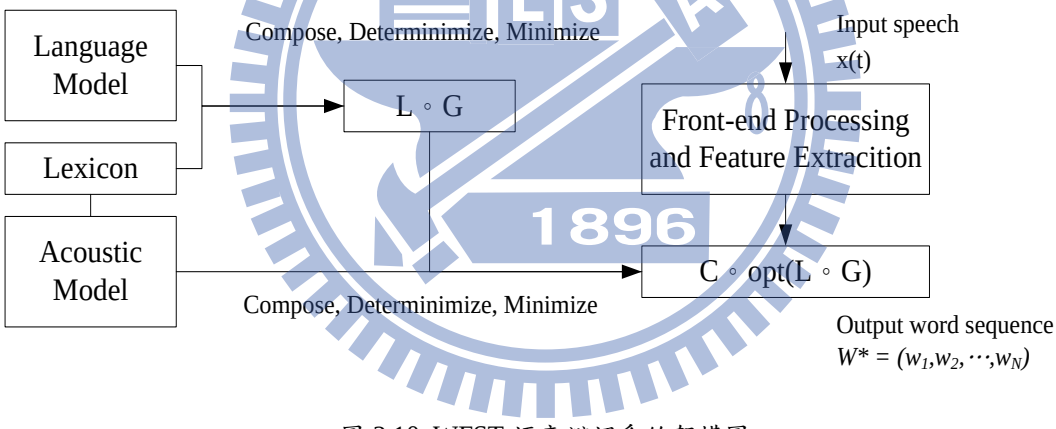

圖 2.10: WFST 語音辨認系統架構圖

在本實驗中使用的辨認器為 Idiap Research Institute 所開發之 Weighted Finite State Transducer Decoder – Juicer[13]。處理 WFST 圖形等相關演算法則採用 Google Research and NYU's Courant Institute 發展之 OpenFst library[14]進行。

以下小節簡單介紹各語音模型 WFST 之建立。

2.4.1 聲學模型之 WFST 建立

在語音辨識中，聲學模型是展開辨認網路時的最後一個層級，本實驗中採用目前應用最廣的隱藏式馬可夫模型(hidden Markov model)描述之，細節可以參考雷氏(L. Rabiner)的著作。Juicer 辨認器將聲學模型的分數與語言模型的分數分開計算，並有一獨立計算 HMM 分數之程式，因此在製作聲學模型層級的有限狀態機時，我們不將 HMM 模型使用 WFST 表示，而是製作前後文相關(context-dependent)的 WFST 圖形。由於在本實驗中採用與前後文無關之單音節模型，因此僅使用單一狀態表示。

圖 2.11: 聲學模型之 WFST

2.4.2 發音詞典之 WFST 建立

藉由聲學模型解碼出聲音序列後，必須藉由發音詞典來將序列對應到有意義的詞語。每個詞對應到一連串的 HMM 序列，故我們可以使用線性的方式簡單製造出詞典的有限狀態機。在這裡不考慮建構樹狀詞典，即使對詞典的有限狀態機進行優化，在整合語言模型層時仍會將該詞條的完整路徑展開，因此最佳化網路的步驟會在組合運算的演算結束後再實現。

由於中文有許多發音相同的同音異義詞，加上我們的聲學模型並無聲調資訊，因此在序列的結尾加上一個輔助符號(auxiliary symbol)來標註這些不同的詞，使得

此圖形符合 functional 特性(functional：每組 input string 對應到唯一的 output string)用以進行確定性演算法。

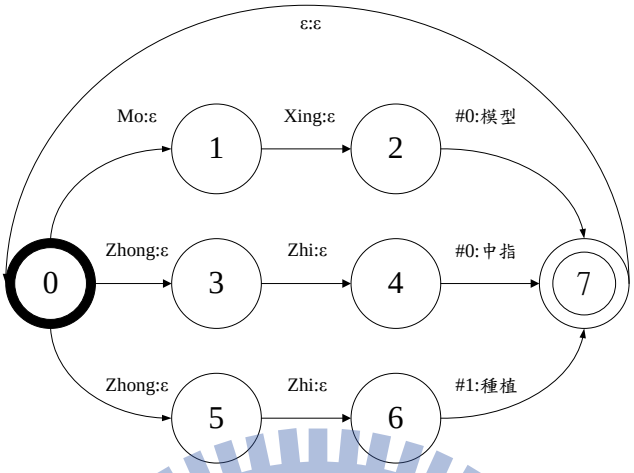

圖 2.12: 發音詞典之 WFST

2.4.3 語言模型之 WFST 建立

在本研究中使用 *n*-gram 語言模型來描述語言模型，其中後撤平滑化可用有限狀態機中的空轉移來表示。我們以 bi-gram 模型為例，狀態內的文字代表其走過的 history，從圖中可以觀察到：當沒有有效的輸入時，就藉由空轉移走到狀態 α，同時也帶上了一個後撤的分數，而由狀態 α 走到其他狀態所帶上的分數，就是已經後撤到 uni-gram 的 *n*-gram 分數。

以下提供一個 bi-gram 語言模型轉換為 WFST 圖形的範例：

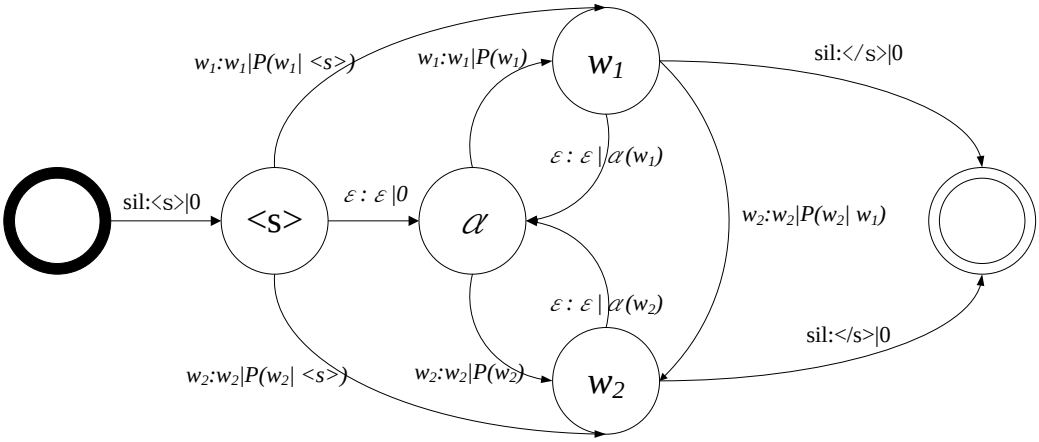

圖 2.13: 雙連語言模型圖

第三章　使用加權有限狀態轉換器實現語音辨認與分析

本章主要介紹傳統 HTK 辨認實驗結果與 WFST 實現大詞彙語音辨認實驗結果，並分析由於中文詞彙數量繁多，詞與詞間也可再構成新的詞彙，其中許多詞類為 open set 無法完整的收錄在詞典中，如:數詞(Neu)、專有名詞(Nb)、詞綴詞…等，而在辨認詞典詞條數的限制下，使得詞典的涵蓋率過低，語音辨識效果有限，經過討論 OOV 詞與詞錯誤率之間的影響提出降低 OOV 詞的方法。

本章節共分為四部份：3.1 節介紹以傳統 HTK 辨認系統之實驗結果；3.2 節介紹以 WFST 實現語音辨認及分析辨認速度與辨識率；3.3 介紹 OOV 造成辨識結果詞錯誤率之計算與分析；3.4 介紹 OOV 詞類分析並提出建立人名階層式模型降低 OOV 詞，並在下一章節實現之。

3.1　使用 HTK 之語音辨認

圖 3.1 就是基本語音辨認系統架構，從輸入音檔中抽取聲學特徵參數序列 $\mathbf{X}_a$，經辨認過程後輸出辨認詞串 $\mathbf{W}^*$，其基本原理數學式如下：

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{X}_a) = \arg\max_{\mathbf{W}} P(\mathbf{W})P(\mathbf{X}_a \mid \mathbf{W}) \qquad (3.1)，$$

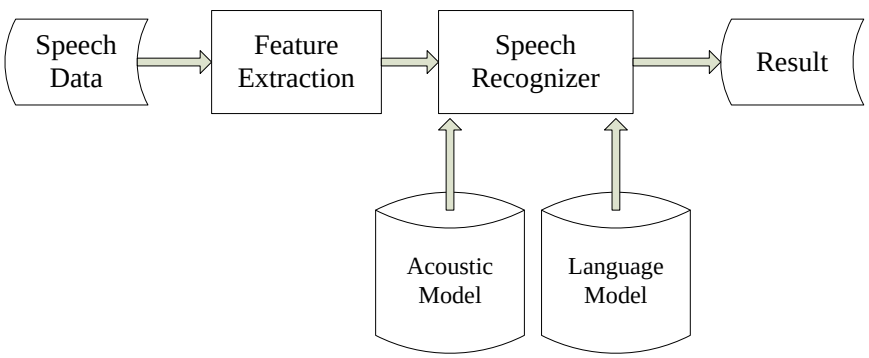

圖 3.1: 基本語音辨認流程圖

3.1.1 使用 HTK 之實驗結果與分析

首先我們加入 bigram 語言模型，同時產生詞辨認實驗結果及 word lattice，然後將產生的 word lattice 利用 trigram 語言模型及 4-gram 語言模型展開再重新評分產生新的辨認結果，各項實驗結果如下表所示。

表 3.1: 搭配語言模型之詞錯誤率

Bigram LM	27.04%
Trigram LM	13.76%
4-gram LM	13.20%

表 3.2: 搭配語言模型之字元錯誤率

Bigram LM	20.02%
Trigram LM	10.54%
4-gram LM	10.20%

從實驗結果我們可以發現，加入語言模型從 bigram 到 trigram 辨識率提升幅度很大，而加入 4-gram 辨識率提升幅度很小，所以實驗只做到 4-gram 為止，經過觀察的結果認為訓練資料量不足是可能的主要原因。

我們將 *n*-gram 辨識率加入 PPL 曲線進行觀察，PPL 會隨著 *n* 越大而越低，所以在足夠資料量的情況下，若可以再提升語言模型的性能，辨識結果的改進是可以預期的，如下圖所示。

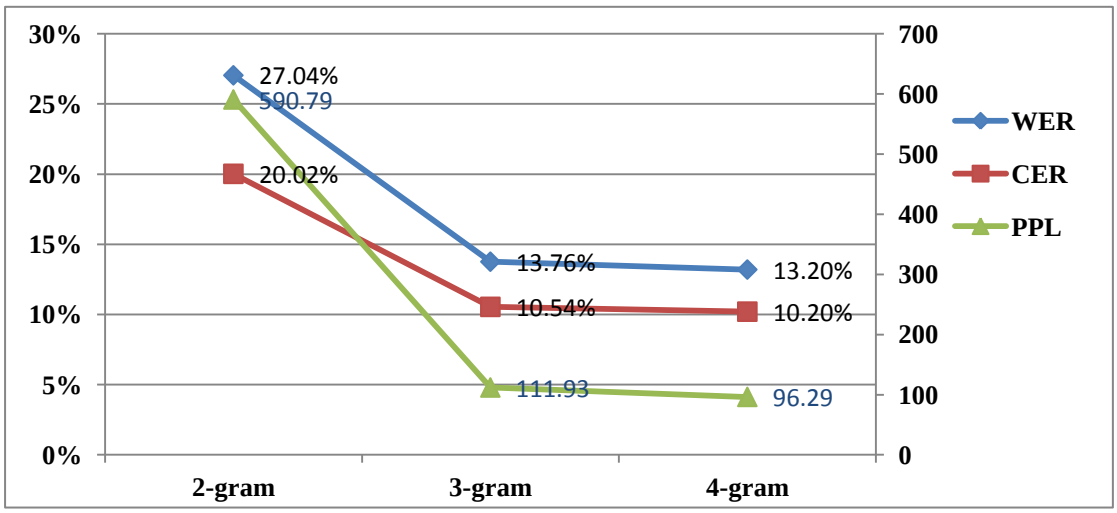

圖 3.2: 詞辨識率、字辨識率與 PPL 的關係圖

3.2 使用 WFST 之語音辨認

語音模型包含: 聲學模型、詞典以及語言模型，並透過有限狀態轉換器將此三部分整合成一個巨大的搜尋網路，最後再以確定化、最小化等演算法對這個搜尋網路進行最佳化的動作。

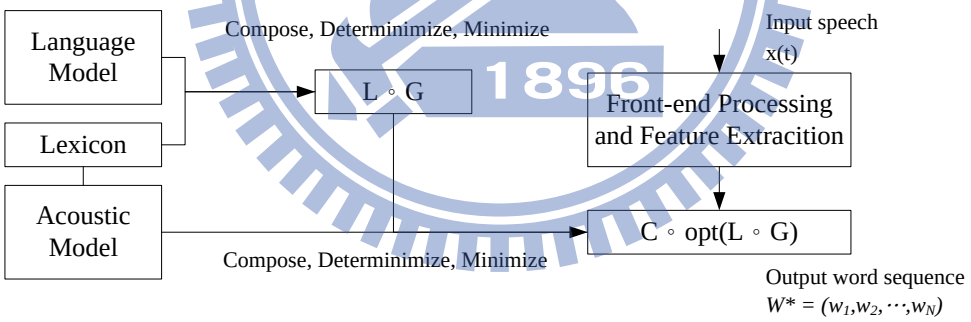

圖 3.3: 有限狀態轉換器語音辨認架構

由於語音模型將三部份整合成巨大的搜尋網路，若語言模型太過龐大，會導致記憶體不足的情況產生，下圖描述使用 SRILM 製作語言模型的參數設定與 PPL 之間的關係。

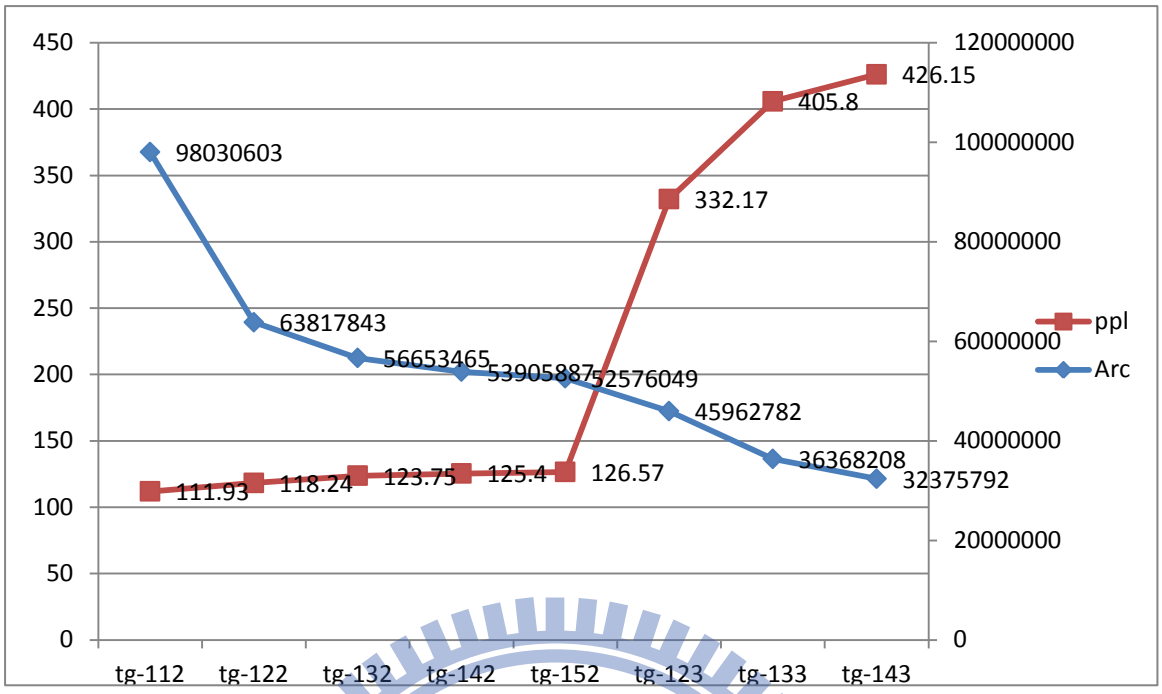

圖 3.4: 語言模型 Arc 數與 PPL 的關係圖

以 trigram 語言模型為例，若 gt1min 設為 1、gt2min 設為 1、gt3min 設為 2，則製作出來的 grammar arc 為 98,030,603，若再對下層 lexicon 展開會變成一個非常大的 FST，雖然較精細的語言模型對語音辨識可以達到更好的辨識結果，但對硬體部分造成很大的負擔，如上圖所示，本實驗以 gt1min 設為 1、gt2min 設為 2、gt3min 設為 2 建立 trigram 語言模型。

3.2.1 使用 WFST 之實驗結果與分析

我們將三部份語音模型整合成一個 FST，並對 TCC300 測試語料 226 個音檔總長度約 2 小時進行語音辨認。

FST 大小如下:

Number of states : 46,834,912

Number of arcs : 96,852,004

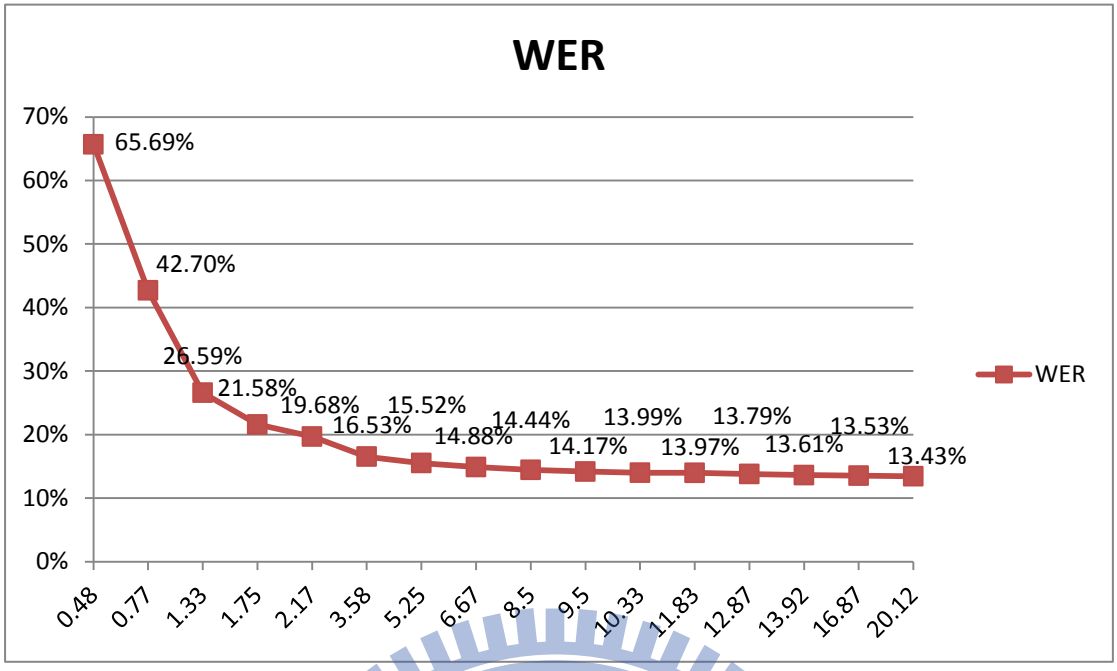

圖 3.5: WER 與 RT 的關係圖

如上圖所示，我們將辨認時的 Maximum hypotheses 提高，則辨認所需的時間越長，相對詞錯誤率也可以越低，但降低的幅度非常緩慢。在同樣的辨識率情況下，WFST 辨認所需的時間比 HTK 快了約 15 倍左右。

3.3 OOV 詞對辨識率的影響

當輸入語音中含有 OOV 詞除了 OOV 詞無法辨識出來，連帶會影響到附近的詞辨認錯誤，導致詞辨識率下降，下圖為擷取部份音檔的辨識結果，我們可以發現當句子中出現 OOV 詞，整句的詞辨識率會大幅下降。

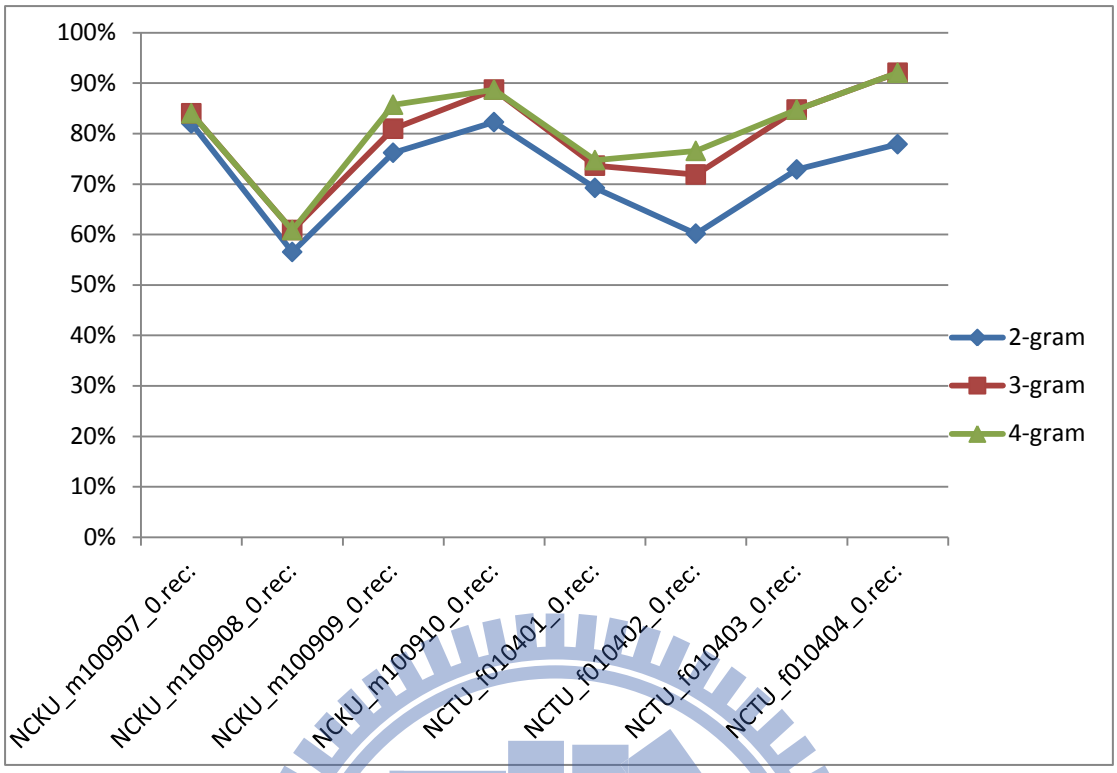

圖 3.6: 音檔辨識結果之辨識率分析(部分節錄)

從上圖我們可以發現，部分音檔的辨識結果不管是 bi-gram、tri-gram 或是 4-gram 的辨識率都非常低，而大部份的音檔都會因為 n 提高而達到辨識率提高的效果，我們對這些音檔的辨識結果取出並加以分析。

表 3.3: 正確解答與辨識結果對照表(部分節錄)

正確解答	辨識結果
跟 崔蓉芝_OOV 目前 一同	證券 融資 目前 一同
使得 崔蓉芝_OOV 也 不得不	使得 全球 紙業 不得不

董事長 周音喜_OOV 共同 前往	董事長 就 應 是 共同 前往

從上表節錄的部分正確結果與辨識結果的對照，我們可以發現 OOV 詞除了辨識錯誤以外，亦影響了前後詞的搶詞狀況，是導致辨識率下降的主因。

下一小節我們針對 OOV 詞對辨識結果的影響進行了分析與統計，以此作為評估詞辨識率的依據。

3.3.1 OOV 詞對辨識率之影響分析

OOV 詞對語音辨識不只是單一詞彙無法辨認出，而會連帶影響附近的詞辨認錯誤，在 M. Gales & P. Woodland [15]提到在英文的語音辨認中一個 OOV 詞平均造成約 1.6 個詞辨認錯誤，本實驗提出在中文大詞彙語音辨認中一個 OOV 詞平均造成約 2.4 個詞辨認錯誤。

表 3.4: OOV 詞對辨識的影響

正確結果	辨識結果
公所	公所
秘書	秘書
NULL	現實
謝石定_OOV	並
認為	認為

如上表範例所示，「謝石定」是一個 OOV 人名，因詞典中並未收錄，所以在辨認時將其辨識為兩個詞，造成一個 Substitution 錯誤、一個 Insertion 錯誤，以此方法將所有辨識結果進行計算得出在中文語音辨識中 OOV 平均所造成的影響。

本實驗所使用的 TCC300 測試語料共有 15,479 個詞，OOV 詞有 482 個，以上所述的方法計算 OOV 詞所造成的影響，以詞錯誤率為 13.76%的辨識結果來計算，約有 55%的錯誤由 OOV 詞所造成，所以如何降低 OOV 詞是接下來研究所面臨的課題。

在某些需要大詞彙的語言中，如俄羅斯語言詞典詞條數擴張至 800K 仍有 1%的 OOV 詞，阿拉伯語言詞典詞條數擴張至 400K 仍有 1%的 OOV 詞，而中文經統計詞典詞條數擴張至 180K 有 1%的 OOV 詞。所以擴張詞典詞條數來提高詞典的涵蓋率不是一個好方法，下一節對 OOV 詞進行分析，並提出解決 OOV 詞的方法。

3.4 OOV 詞類分析

以詞典六萬詞而言有 2.7%為 OOV 詞，約 1.0 x 10^7 個 OOV 詞，各詞類詞典涵蓋率如下圖所示。我們可以觀察出某些詞類如 Nb、Na、Nc，詞典涵蓋率較低，大部分的 OOV 詞由這三詞類所組成，也就是所謂的 Named Entity，這些詞類是一個 open set，隨著時代的改變新增或者減少。

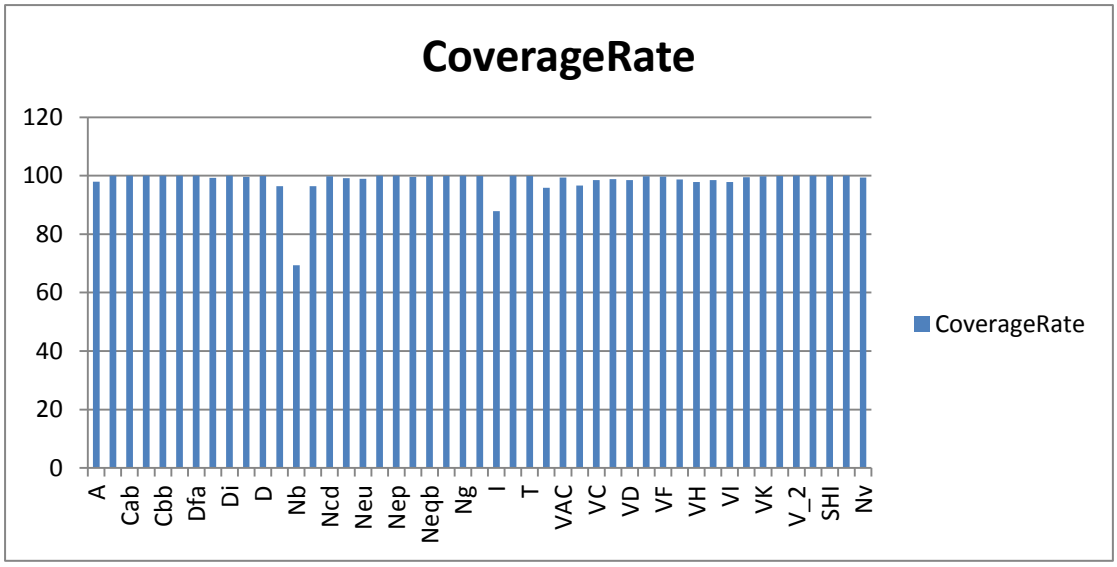

圖 3.7: 各詞類詞典涵蓋率

我們將這些 OOV 詞再依詞類進行統計，詞類為 Na 佔 OOV 詞 30.59%，詞類為 Nb 佔 OOV 詞 40.07%，詞類為 Nc 佔 OOV 詞 11.69%，如下圖所示。

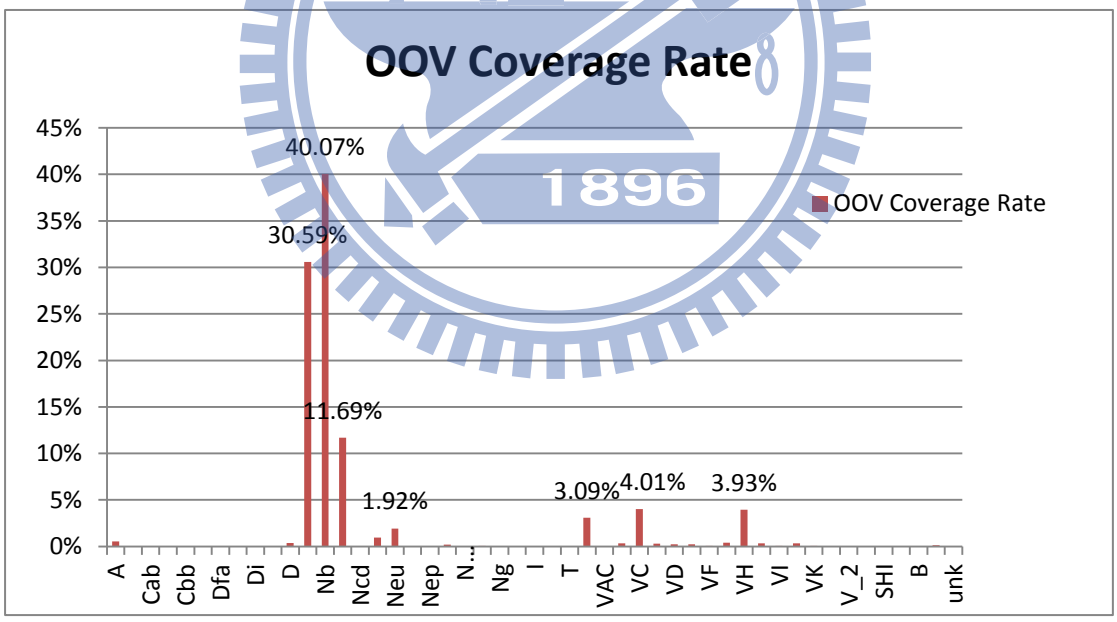

圖 3.8: OOV 詞各詞類涵蓋率

從上列的圖表觀察，詞類標記為 Nb 的 OOV 詞為我們首要處理的對象，詞類標記為 Nb 的詞中約有 75%為人名，而三字詞中文人名(姓+名)佔 58%，所以下一章節我們提出將這些三字中文人名訓練成一個人名語言模型加入辨識，希望藉此降低語音辨識詞錯誤率。

第四章　階層式語言模型實驗結果與分析

前一章我們提到OOV詞類中標記為Nb的詞中有58%為三字詞的中文人名，佔 OOV 詞類約 23%，本章將 OOV 人名視為一個類別處理，利用人名與前後詞的關連性以 *n*-gram 模型訓練之。我們將此類別的人名切短，以較少的單元來涵蓋無法收錄的詞彙，將此類別分為姓氏和名字分別建立語言模型，最後在以取代演算法將兩個語言模型整合在一起。

本章共分為三部分：4.1 節介紹建立人名語言模型之方法；4.2 節介紹將人名語言模型與原語言模型之整合；4.3 節介紹實驗結果與討論。

4.1　建立人名語言模型

4.1.1　人名抽取

圖 4.1: 文字語料庫處理流程

在第二章的文字處理流程中，我們先以 CRF 斷詞器進行斷詞並依照 TF-IDF 來決定收錄詞典，但此斷詞結果並無包含命名實體(Named Entity, 文本中具有特定意義的實體，如人名、地名、組織名等專有名詞)之標記結果。由於詞性標記為 Nb 大多為人名，所以我們針對斷詞後結果詞性標記為 Nb 的 OOV 詞並對人名姓氏(522 個人名姓氏)進行偵測取出符合的的資料，一共取得 3,033,630 個詞條，約佔 OOV 詞

30.28%，則各姓氏詞條分布如下圖所示。

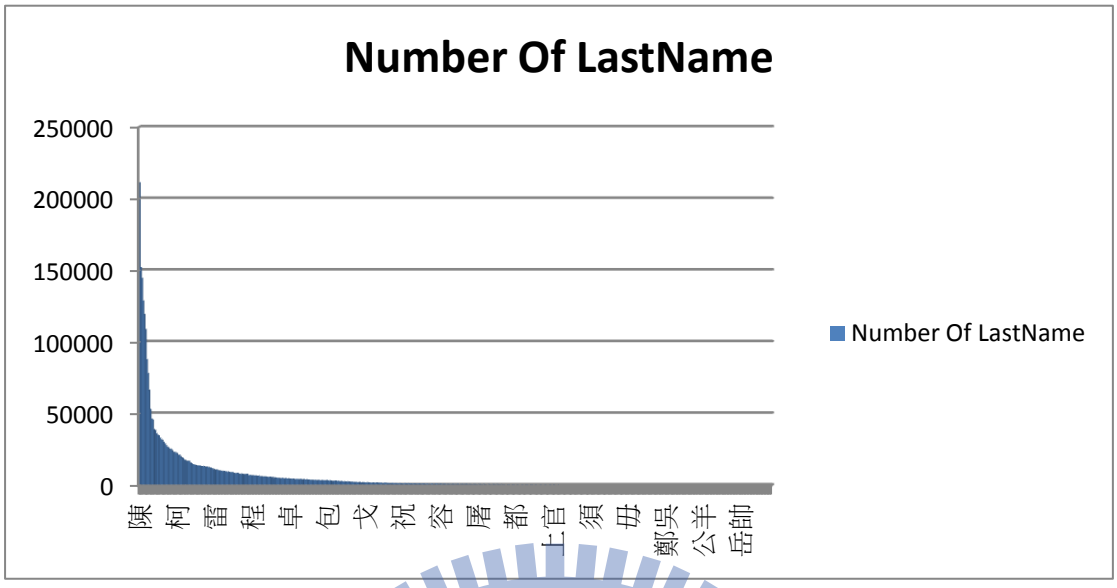

圖 4.2: OOV 詞條中符合人名姓氏之分布圖

4.1.2 OOV 中文人名之選擇與拆解

從統計結果我們可以知道雖然中文人名姓氏有 522 個，但大部分姓氏並無法從文字語料庫中找到匹配的詞條，如下圖所示，我們進一步統計姓氏的涵蓋率。

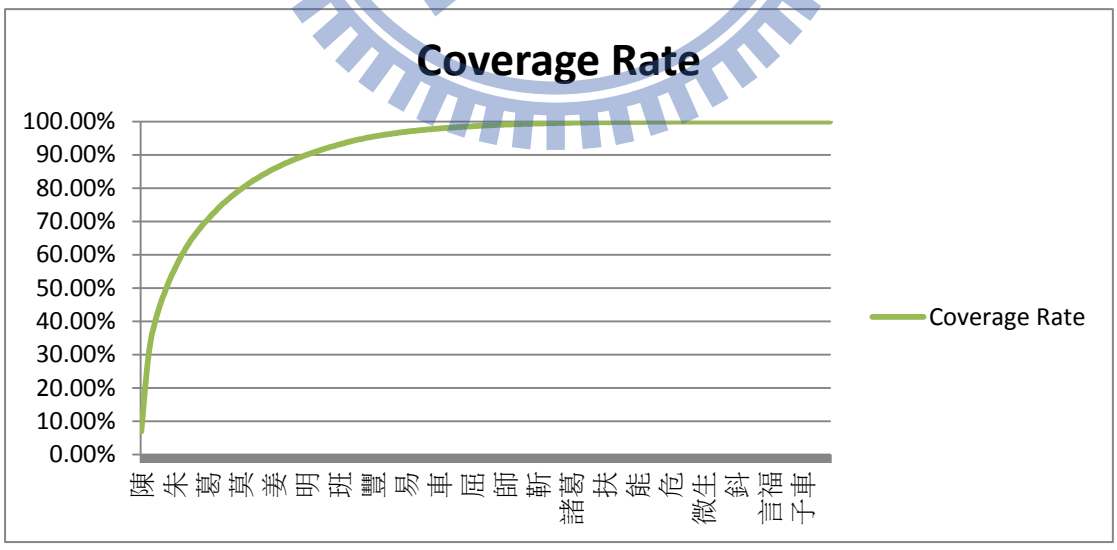

圖 4.3: 符合人名姓氏詞條之涵蓋率

從圖 4.3 中觀察，我們選取以詞頻排序前 200 個姓氏做為人名模型的訓練之用，涵蓋率約為 96.86%。

考慮到將約 300 萬個人名視為一個 Class，所建出來的人名模型前後詞的關連性相當混淆，若直接當作訓練語料建立人名模型勢必會讓整個語言模型瓦解，所以我們進一步得做了以下的處理。

1.) 首先我們選出來的人名詞條數共有 514,869，所以我們針對這些候選詞將詞頻 10 以下的詞條去除，取得 48,058 個符合的詞。

2.) 接著選取姓氏詞頻前兩百名的候選詞並將音節符合原六萬詞詞典的詞剔除。共取得詞條數 47,428，1,519,344 筆資料作為訓練之用。

表 4.1: 候選詞與原六萬詞音節相同對照表

原六萬詞詞典的詞	訓練人名的候選詞
朱家崎	朱家琦
曹竣揚	曹峻揚
彭紹瑾	彭紹謹
張震嶽	張震岳
…	…

3.) 將姓氏與名字拆解分別建立語言模型，由於辨認時所使用的聲學模型不含聲調(tone)資訊，因此我們將訓練文本轉寫成僅有音節(syllable)的形式，保留同樣的音節資訊並取頻率較高者為代表字元。

例如:

江、蔣、姜三者音節相同，頻率最高者為江，則把蔣、姜轉寫成江來訓練。

姓氏部分:

原收錄200個姓氏，將同音節合併後剩153個。

名字部分:

處理方式與姓氏部分相同，並將名字(兩個字元)視為一個詞，共取得 17,213 個詞條，然後以詞頻排序取前 8000 個詞條收錄在詞典中作為訓練之用，其涵蓋率為 87.20%。

4.1.3 建立人名語言模型

將 OOV 人名視為一個類別，並藉由 *n*-gram 模型訓練詞與類別之間的機率，則預估人名的機率是可拆解為 Root LM 與 PersonName LM。Tri-gram 的機率預估如下，bi-gram 與 uni-gram 類推之:

$$P(W^*) = P(w_1) * P(w_2 \mid w_1) * \prod_{i=3}^{N} P(w_i \mid w_{i-1}, w_{i-2}) \quad (3.1)，$$

則人名的機率預估可寫成:

$$P(w_i = w_{i-1}, w_{i-2}) = P(PN \mid w_{i-1}, w_{i-2}) \cdot P(PN) \quad (3.2)，$$

1.) 外部機率(Inter-word probability)的預估:

我們將 OOV 人名與前後詞的關連性以 *n*-gram 模型來預估，並將內部機率 weight push 至外部機率上，則數學式可寫成:

$$P(w_i = w_{i-1}, w_{i-2}) = P(PN \mid w_{i-1}, w_{i-2}) \cdot P_{\max}(LN) \cdot P_{\max}(FN) \cdot \left(\frac{P(LN)}{P_{\max}(LN)} \cdot \frac{P(FN)}{P_{\max}(FN)}\right) \quad (3.3)，$$

其中 $P(LN)$ 為 Last Name 的機率，$P(FN)$ 為 First Name 的機率。

將 $P_{\max}(LN)$ 與 $P_{\max}(FN)$ 機率 Push 至進入這類別的 Arc 上，在辨認時可降低進入這類別的機率，避免 PN Model 不必要之展開，發生嚴重的搶詞狀況。

2.) 內部機率(Intra-word probability)的預估:

由於中文人名由「姓氏」、「名字」所組成，我們假設這兩個類別為不相關，故將這兩個類別分別以 uni-gram 預估之，其機率為:

$$P(LN_i, FN_i \mid PN) = P(LN_i) \cdot P(FN_i) \quad (3.4)，$$

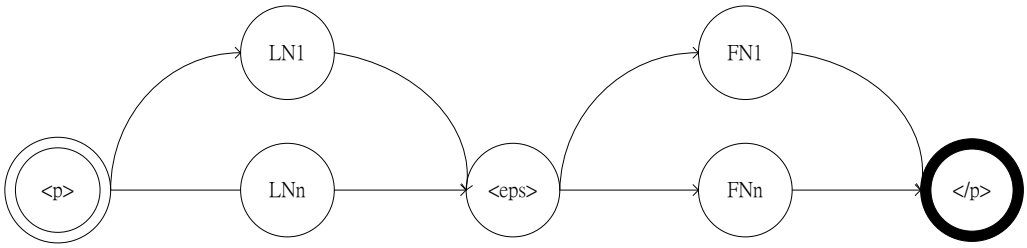

圖 4.4: 內部人名語言模型示意圖

4.2 階層式語言模型之整合

訓練完語言模型並轉換成 WFST，要將 PN WFST 構回 Root WFST 中，在此我們使用 Open FST 的取代演算法實現之。PN WFST 是特別針對 OOV 人名類別所訓練的語言模型，故要將 Root WFST 中 Arc 的輸入輸出符號為 PersonName 以 PN WFST 取代之。

圖 4.5: Root FST 示意圖

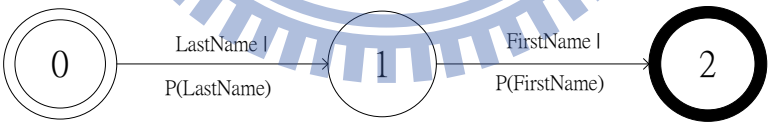

圖 4.6: PN FST 示意圖

以改寫之文本進行 *n*-gram 語言模型訓練，可得 hierarchical language model 中的外部機率(inter-word probability)，另外建立出 PNLM 模型則來給定其內部機率(intra-word probability)，再以 replace 演算法重構回 Root WFST 上。由圖 4.5 可看出 Inter-word 的分數原被放置在帶有 PersonName 非終結符號之轉移上，經過取代演算法後，被一條進入 PN WFST 的空轉移所繼承了，如圖 4.7 所示。

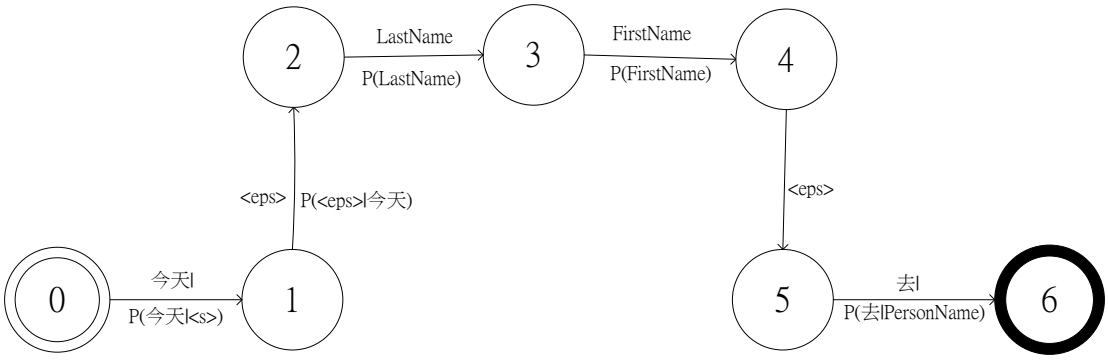

圖 4.7: 整合 PN WFST 與 Root WFST 示意圖

4.2.1 取代演算法展開之數量級

我們將選取出來的 OOV 人名全部作為一個 class 來訓練 tri-gram 語言模型，產生轉移的數量級為 6.5×10^7 條，帶有 PersonName non-terminal label 之轉移的數量級為 1.2×10^4 條，人名語言模型轉移數的數量級為 8×10^3 條。

若直接以取代演算法將 PN model 置入 Root LM 中，則會造成取代後的 graph 過大且向下對發音詞典展開會導致記憶體不足無法進行運算。再者若建立的 Root LM 對 PN class 前後詞的語言結構不夠可信，就算硬體的問題可以解決，辨認結果也是可以預期的，會使得進入 PN model 的機率過高造成嚴重的搶詞狀況。

本實驗提出在語言模型層就對帶有 PersonName non-terminal label 之轉移 arc 進行 pruning，將語言結構機率較低的部分移除，除了可以得到辨認時進入 PN model 的可信度之外，亦可以降低硬體部份的需求並可以實現之。

下表為對語言模型層進行 pruning 之後保留下來的部分節錄之 trigram 語言模型結構，我們可以從中看出 PersonName 的前詞通常為頭銜或者是介詞，後詞通常為動詞或者是介詞，而這樣的語言結構具有相當的可信度，很符合中文的語言特性。

表 4.2: trigram 語言模型含 PersonName 之語言結構(部分節錄)

Wn-2	Wn-1	Wn
公司	董事長	PersonName
管理	課長	PersonName
教授	PersonName	認為
分局長	PersonName	表示
<s>	PersonName	進行
PersonName	也	指出
PersonName	以	五百

若未進行 pruning 在 grammar 層原帶 PersonName non-terminal label 之轉移數為 12,438 條，進行 pruning 後留下 2,908 條帶有 PersonName non-terminal label 之轉移數，下表為各層模型加入 PN model 前與加入 PN model 後之狀態數與轉移數。

表 4.3: 各層模型之狀態數與轉移數

	Number of States	Number of Arcs
PN WFST	7	8,157
Root LM WFST	14,171,500	63,881,664
PNLM WFST + Root LM WFST	14,176,291	70,108,019
未加 PNLM 前 Lexicon WFST	288,080	348,250
已加 PNLM 後 Lexicon WFST	335,479	409,579
未加 PNLM 前 Final WFST	46,834,912	96,852,004
已加 PNLM 後 Final WFST	47,067,354	101,879,857

4.3 實驗結果與分析

我們將建立的 PM LM 置入 Root tri-gram LM，然後將其對 Lexicon 展開進行 determinize、minimize 演算法將 graph 簡化，再對 Hmmlist 展開，之後加入 AM 並對輸入語音經過抽參數之資訊進行 decode，找出一條最佳的辨識結果。以下為整個語音辨認的流程圖:

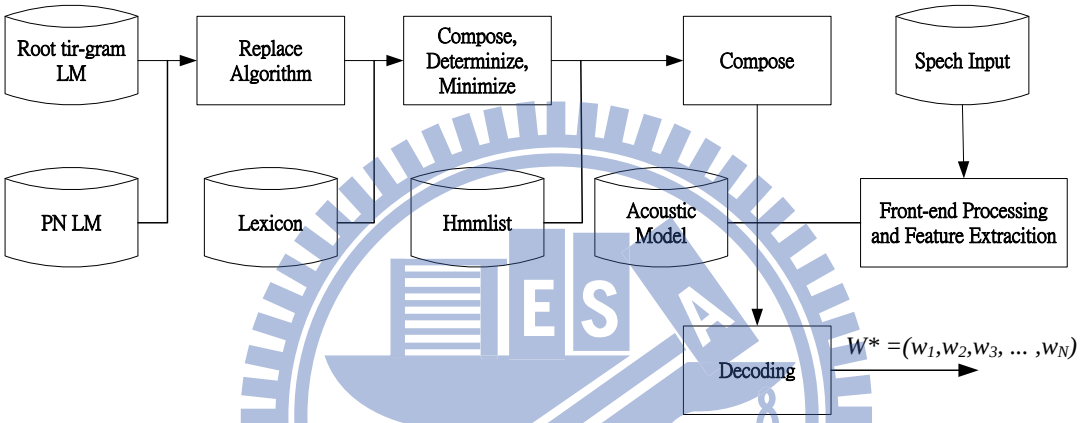

圖 4.8: 階層式模型語音辨認流程圖

4.3.1 實驗結果

TCC-300 測試語料共有 15,493 個詞，其中 OOV 詞共有 437 個，佔測試語料總詞數約為 2.8%，OOV 詞中符合三字中文人名(姓氏+二字名字)共有 126 個，佔 OOV 詞總數 28.83%，若這些三字中文人名全數辨認正確，且每個 OOV 詞平均影響 2.4 個詞辨認錯誤，則辨認上限預估可提升 1.92%。測試語料中 126 個人名可以從 PN model 構回的人名共有 97 個，在實際辨認時，部分人名雖然在 PN model 中無法組合出，若前後詞的語言結構機率夠高，則會以相近的音組合出人名。在語言模型層進行 pruning 後，例如移除 OOV 人名與 OOV 人名相接的機率，而在測試語料庫中存在著這樣的語言結構，若將未建立語言模型的這部分不列入

統計，則欲構回人名數之辨識結果統計如下表所示。

表 4.4: 辨識結果構出人名統計表

	欲構回人名數	辨認出人名數	正確辨認出人名數
修正統計前	126	21	20
修正統計後	110	21	20

從上表統計顯示本實驗欲辨識的三字中文人名共有 126 個，而將測試語料中未建立語言模型結構的部分不列入計算，如移除人名接人名的機率，則重新統計後欲構回的人名數為 110 個，從辨識結果來看共構出 21 個人名，其中有 16 個人名正確辨認出相同音節人名，1 個辨認錯誤，4 個構出不同音節的人名，修正後辨識出人名之詞錯誤率約降低 0.12%，如下圖所示。

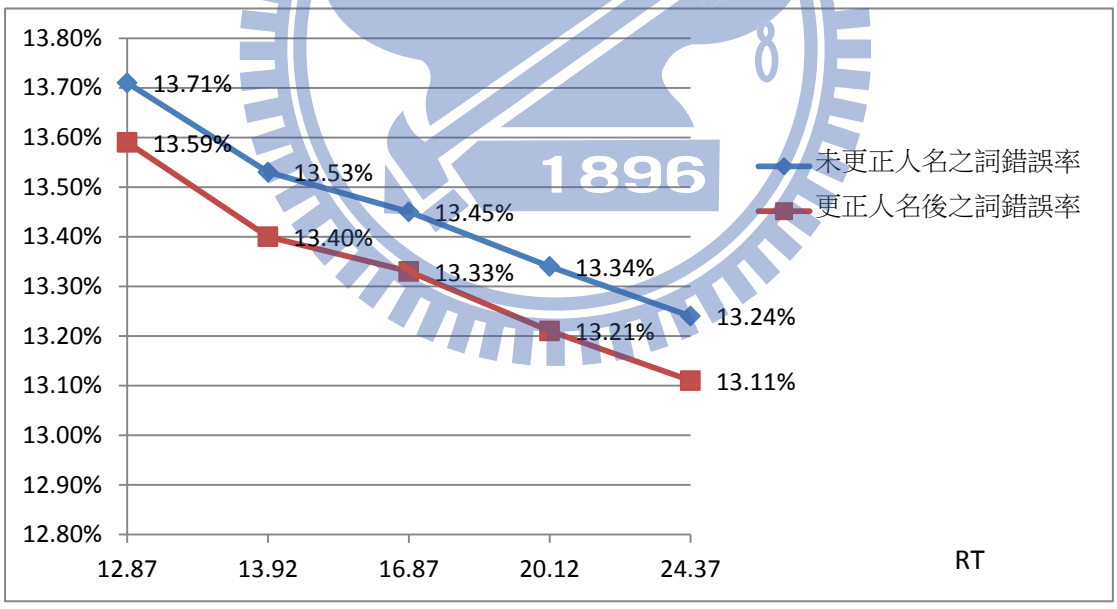

圖 4.9: 加入 PNLM 之詞錯誤率

4.3.2 實驗結果之分析

本小節主要呈現部份的辨識結果並加以分析，以下列出了四種不同情形的辨

識結果，並加以詳細分析。

下表結果我們可以觀察到當 OOV 人名的前詞為稱謂以及後詞為動詞這樣的機率在加入 PNLM 後可以成功地將 OOV 人名構出，從此可以推論出我們將 OOV 人名在語言模型中的前後詞機率嚴謹的限制，使得進入人名模型的機率降低，並以相同發音(無 Tone 資訊)的詞(姓氏+人名)取代，若只針對聽寫部份，人都無法正確地寫出人名何況是電腦以無聲調資訊的語音模型辨認，故下表的例子在計算辨識率時，應將其列為辨認正確。

表 4.5: 正確人名辨識結果

正確答案	未加入 PNLM 辨認結果	加入 PNLM 辨認結果
NCKU_f070301_0		
管理處	管理處	管理處
經理	經理	經理
楊政雄_OOV	楊鎮雄	楊正雄_PersonName
表示	表示	表示
NCKU_f070303_0		
水源	水源	水源
里長	裡	里長
陳枝福_OOV	長成	程智富_PersonName
表示	支付	表示
	表示	
NCKU_f070307_0		
公會	公會	公會
理事長	理事長	理事長
郭振興_OOV	國	郭振興_PersonName
針對	振興	針對

	針對	

下表的辨識結果可以看出，當 OOV 詞前後的語言結構可信度不足，在辨認時並未從 PN Model 中構出人名，而找出未加入 PNLM 前的辨認路徑產生辨認結果。

表 4.6: 並未構回人名的辨識結果

正確答案	未加入 PNLM 辨認結果	加入 PNLM 辨認結果
NCKU_f070307_0		
特殊	特殊	特殊
族群	處	處
郭振興_OOV	尋獲	尋獲
指出	真心	真心
	指出	指出
NCKU_f080110_0		
<s>	<s>	<s>
董俊斐_OOV	東芝	東芝
大臺北	在	在
	大臺北	大臺北

下表的辨識結果雖為構出錯誤的詞，但不影響辨識率，兩個辨識錯誤人名的結果都是同一類型， OOV 詞在句首且從下一個詞可以推測出前一個 OOV 詞為人名的語言結構。

表 4.7: 構出錯誤人名結果

正確答案	未加入 PNLM 辨認結果	加入 PNLM 辨認結果
NCTU_f020413_0		

<s>	<s>	<s>
賭迷_OOV	杜林	杜明義_PersonName
指出	指出	指出
目前	目前	目前

下表的辨識結果雖為構出不同音節的人名，但因為前後詞語言結構的關係，而降低了原本因為 OOV 詞造成的搶詞情況修正了辨識結果。

表 4.8: 構出非同音節人名的辨識結果

正確答案	未加入 PNLM 辨認結果	加入 PNLM 辨認結果
NCKU_m061009_0		
建設局長	建設局長	建設局長
邱傳榮_OOV	從	邱昌榮_PersonName
則	傳統	則
說	則	說
	說	
NCKU_m080906_1		
得主	得主	得主
吳珊貞_OOV	湖南省	吳山生_PersonName
等	衡山	等
三	人	三
人		人

第五章　結論與未來展望

5.1　結論

本研究採用加權有限狀態轉換器實現一階段大詞彙語音辨認，藉由不同的演算法整合語音模型，主要研究內容為語音模型層進行改良，並以 TF-IDF 進行選詞；另一方面注重 OOV 詞對辨識結果影響之分析，並提出從 OOV 詞中建立類別之語言模型，將其與原語言模型整合成一個階層式語音辨認系統，並以一階段式進行大詞彙語音辨認，實驗結果顯示加入階層式語言模型確實可以改善辨識效能。

從本研究中可以發現語言模型的結構如果可以更可信，則可以有效地幫助語音辨認，本研究實現 OOVs 人名模型，將 OOV 詞中人名部份視為 class，並適當地對進入 PN model 機率較低的 arc 進行 pruning，實驗結果有效地降低了詞錯誤率，若將 pruning 的設定繼續放大，或許可以構回更多的 OOV 人名，但也有可能造成搶詞狀況影響辨識結果。

5.2　未來展望

未來可以針對不同語言結構的詞彙建立類別，如數量複合詞(DM)與其他 Named entity 資訊如地名、組織名等，在語言模型層進行整合，進而提升語音辨識效能。另一方面可以從 WFST 辨識結果產生的 lattice 上加入語速、韻律等模型實現二階段式語音辨認，並達到進一步提升語音的辨認效能。

參考文獻

【1】 Mehryar Mohri, “Finite-State Transducers in Language and Speech Processing,” AT&T Labs – Research, 1997

【2】 M. Mohri, F. Pereira, M. Riley, “Weighted finite-state transducers in speech recognition,” Proc. of ASR2000, pp. 97–106, 2000.

【3】 Mohri, M., Pereira, F., Riley, M.l.: Weighted finite-state transducers in speech recognition. Computer Speech and Language 16(1), 69–88 (2002)

【4】 Mehryar Mohri, Michael Riley, “A Weight Pushing Algorithm for Large Vocabulary Speech Recognition,” AT&T Labs – Research

【5】 Chia-Hsing Yu, “Large Vocabulary Continuous Mandarin Speech Recognition Using Finite-State Machine,“ NTU Digital Speech Signal Processing Lab, 2004

【6】 Shang-Yao Chang, “Large Vocabulary Continuous Mandarin Speech Recognition Using Finite-State Machine,” NCTU Speech Processing Lab, 2008

【7】 Daniel Jurafsky and James H. Martin, ”SPEECH and LANGUAGE PROCESSING,”2008

【8】 Slava M. Katz, “Estimation of probabilities from sparse data for the language model component of a speech recognizer,” *IEEE Transactions on Acoustic, Speech and Signal Processing*

【9】 Chien-Pang Chou, “Improvement on Language Modeling for Large-Vocabulary Mandarin Speech Recognition,” NCTU Speech Processing Lab, 2009

【10】 Yu-Chao Hsu, “Large Vocabulary Continuous Mandarin Speech Recognition Using Weighted Finite-State Transducer,” NCTU Speech Processing Lab, 2011

【11】 J. Lafferty, A. McCallum, F. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,” In proc. ICML01, 2001.

【12】 陳克健, 黃居仁, “中央研究院漢語平衡語料庫(簡稱Sinica Corpus)第4.0版,” 中央研究院資訊所, 2010.

【13】 D. Moore, J. Dines, M. Magimai Doss, J. Vepa, O. Cheng, and T. Hain, “Juicer: A weighted finite state transducer speech decoder,” in Proc. MLMI (to appear), Washington DC, May 2006.

【14】 C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri. OpenFst: A general and efficient weighted finite-state transducer library. In Proceedings of the 12th International Conference on Implementation and Application of Automata (CIAA 2007), Prague, Czech Republic, July 2007, volume 4783 of Lecture Notes in Computer Science, pages 11–23. Springer, Heidelberg, 2007.

【15】 M. Gales & P. Woodland, “Recent Progress in Large Vocabulary Continuous Speech Recognition: An HTK Perspective,” ICASSP Tutorial, 2006.

附錄一:實驗所用 Variant Word Pair 表

Variant Word Pair		Variant Word Pair	
愈來愈	越來越	鄭重其事	慎重其事
巴金森氏症	帕金森氏症	鼎鼎有名	赫赫有名
無人問津	乏人問津	除此以外	除此之外
身臨其境	身歷其境	勢所難免	在所難免
非同尋常	非比尋常	努力以赴	全力以赴
意興風發	意氣風發	生氣勃勃	生氣蓬勃
同聚一堂	齊聚一堂	言之成理	言之有理
共聚一堂	齊聚一堂	自嘆不如	自嘆弗如
不亢不卑	不卑不亢	從頭至尾	從頭到尾
成竹在胸	胸有成竹	無怨無尤	無怨無悔
精疲力盡	精疲力竭	耶誕	聖誕
垂手可得	唾手可得	耶誕節	聖誕節
難分軒輊	不分軒輊	耶誕老人	聖誕老人
無分軒輊	不分軒輊	聖誕老公公	聖誕老人
百折不撓	不屈不撓	損毀	毀損
堅毅不撓	不屈不撓	星期一	週一
縛手縛腳	綁手綁腳	禮拜一	週一
指手劃腳	比手畫腳	星期二	週二
粥少僧多	僧多粥少	禮拜二	週二
餐風宿露	餐風露宿	星期三	週三
心驚膽顫	膽顫心驚	禮拜三	週三
聞名遐邇	名聞遐邇	星期四	週四

縮衣節食	節衣縮食	禮拜四	週四
豐功偉績	豐功偉業	星期五	週五
臨機應變	隨機應變	禮拜五	週五
如醉如癡	如癡如醉	星期六	週六
煞有介事	煞有其事	禮拜六	週六
洋洋得意	得意洋洋	星期天	週日
命在旦夕	危在旦夕	星期日	週日
峻工	完工	禮拜天	週日
一脈相承	一脈相傳	禮拜日	週日
猶疑不決	猶豫不決	醫師	醫生
和衷共濟	同舟共濟	耶誕夜	聖誕夜
一新耳目	耳目一新	耶誕樹	聖誕樹
一言不發	不發一語	耶誕卡	聖誕卡
鼎鼎大名	大名鼎鼎	教部	教育部
來歷不明	來路不明	教局	教育局
僕僕風塵	風塵僕僕	市銀行	市銀
聳人聽聞	駭人聽聞	昨天	昨日
譬如說	比如說	日昨	昨日
天淵之別	天壤之別	明天	明日
迫在眉梢	迫在眉睫	今天	今日
前所未聞	前所未見		
萬不得已	逼不得已		
迫不得已	逼不得已		