國 立 交 通 大 學

電信工程研究所

碩 士 論 文

以加權有限狀態轉換器實現中文連續語音辨認

Large Vocabulary Continuous Mandarin Speech Recognition Using Weighted Finite-State Transducer

研究生：許昱超

指導教授：陳信宏 博士

中 華 民 國 一百 年 八 月

以加權有限狀態轉換器實現中文連續語音辨認

Large Vocabulary Continuous Mandarin Speech Recognition Using Weighted Finite-State Transducer

研 究 生：許昱超　　Student：Yu-Chao Hsu

指導教授 ：陳信宏 博士　　Advisor：Dr. Sin-Horng Chen

國 立 交 通 大 學

電 信 工 程 學 系

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

August 2011

Hsinchu, Taiwan, Republic of China

中 華 民 國 一 百 年 八 月

以加權有限狀態轉換器實現中文連續語音辨認

研 究 生：許昱超　　指導教授：陳信宏 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中文大詞彙連續語音辨認系統。首先介紹加權有限狀態轉換器的相關演算法，以及不同層級之語音模型如何以有限狀態機圖形來表示。接著加入階層式語言模型的概念，使用 NER 標記來訓練人名模型以解決 OOV words 中屬於人名的問題，並提出一階段式與兩階段式的架構來進行辨識的方法。在一階段式的辨認中，我們實現一個即時展開有限狀態機圖形的演算法，使得在一階段式架構下也能使用較複雜的階層式模型來提升辨識效能。實驗使用 TCC-300 的朗讀式語音進行辨識，在加入階層式的語言模型後，一階段辨識對於詞錯誤率為 26.26%，而採用兩階段式重計分之辨認則最多可以將錯誤率降低至 23.73%。

Large Vocabulary Continuous Mandarin Speech Recognition Using Weighted Finite-State Transducer

Student：Yu-Chao Hsu　　Advisor：Dr. Sin-Horng Chen

Institute of Communication Engineering
National Chiao Tung University

Abstract

This thesis presents an ASR system based on Weighted Finite-State Transducer(WFST). In the first we will introduce some algorithms that used to construct WFST graph, and how we express different models using WFST format. Then, we described a hierarchical language model training by NER labels to deal with the problem of chinese person name, which often detected as OOV words. We incorporating the hierachical langugage model into one-stage and two-stage ASR system. In the one-stage ASR system, an on-the-fly replace algoritm was implemented to reduce the memory's allocation, so we can use a complex hierachical model to calculate the probability of chinese person name. We evaluate our approach on the TCC-300 corpus, which consists of long paragraphic utterances, obtained a 0.38% absolute improvement in word error rate in one-stage ASR system; and at most 2.91% when using two-stage ASR system.

致謝

最後的學生生活要結束了，由衷感謝指導教授陳信宏老師以及王逸如老師這兩年碩士生活的教誨。兩位老師在研究上著眼的觀點不同，親切而不失嚴謹的態度，無論是研究的大方向或是實驗的實作上都給予我很多的幫助。

再來要感謝實驗室的博班學長們：第一個是甚麼都會的性獸，總會關心我的進度並且跟我討論研究與程式上的缺失；然後是正在當兵中的合哥，不管是 linux 的問題或是訓練模型的問題都給我很多指導；感謝希群學長跟我討論階層式語言模型的實作與細節、講話很幽默而且不知道我是碩二的輝哥、跑去開公司的 Barking 學長、去中研院才看得到的阿德學長。還有就是跟我們一起奮戰的戰友們：街頭詩人的筆格胖、舞藝高超的智障、常常空震的文良、神一樣的銘傑、帥氣漂撇的小瞎、已經變成聰明人的佳緯、修一堆課的啟全、假日也很拼的勁竹、最後加入我們的冠驛，這兩年辛苦了！當然也不能忘了碩一的學弟妹們：小邱、企鵝、Kiwi、睿詮、俊翰、昂星、雅婷，預祝你們明年也能順利完成學業！此外我要感謝兩年來跟我一起重訓的小豬、雄哥、榕榕、熊大、基諾，讓我即使很少打球也能保持運動的好習慣。感謝我的好室友們：歐拉、謙和、咪哭、文中、大郭、柏益，不管是玩樂或是唸書都有人可以互相支援，還有嘟嘟、華山…族繁不及備載，能夠認識各位真是太好了。

最後，感謝我的父母與老妹給我在精神與經濟上的支持，使我在外地讀書也能感受到家庭的溫暖，祝大家心想事成、美夢成真！

目錄

圖表目錄

第一章 緒論

1.1 研究動機

傳統使用隱藏式馬可夫模型的一階段式語音辨認，是將聲學模型依照發音辭典串接為一個詞的大型隱藏式馬可夫模型，在語言模型的層級每個詞會再串接所有的詞，形成一個廣大的辨認網路。在辨認時將網路展開至聲學模型層級，整個搜尋空間並沒有經過任何優化的處理，在搜尋時使用維特比光束搜尋來提昇搜尋速度。

另外，針對中文語音辨識而言，中文在構成"詞"（word）的規則相當彈性，每個中文"字"（character）都代表其各自的意思，不像拼音語系在詞間有著分隔符號（delimiter）作為區隔。欲進行大詞彙語音辨認時，這些多元的變化使得詞典的收錄變得很困難，諸如：人名、數量複合詞、詞綴構詞等，此類別的詞可任意組合，故中文語音辨識存在著辨認詞典無法完整收錄詞彙而阻滯辨識效能成長的情形。

近年來，有限狀態機(Finite-State Machine)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，我們可以將不同層級的語音模型整合在一起，並對辨識網路進行最佳化而得到良好的辨認速度。在本研究中，並提出使用階層式的語言模型對中文人名進行處理，希望藉此提高詞涵蓋率而提升辨識成效。

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，近年來廣泛地運用在語音辨認中，如：用來表示前後文相關(Context dependency)的發音模型、表示大詞彙的字典…等。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)，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。依照有限狀態機的特性，將傳統語音辨認中各自獨立的三個部份：聲學模型、發音辭典以及語言模型，利用組合運

算(composition)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辭典，大多將語料中出現的詞按照詞頻排序，取其順位高者納入詞典中。考慮中文構詞的多元與彈性，無法收入所有的詞彙，但不在詞典中的詞彙便無法辨認得出。這些不在詞典中(Out-of-Vocabulary, OOV)的詞(OOV words)會影響辨識效能，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。辨認時採用二階段式的辨認系統，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 上的路徑而最後的辨認結果。

針對 OOV 處理的議題，前人提出過不少研究的成果，多以 class n-gram 預估詞的機

率，一方面也可以減少 data-sparseness 的問題，而預估 word bi-gram 的機率算式即改寫為：$P(w_i \mid w_{i-1}) = P(c_i \mid c_{i-1})P(w_i \mid c_i)$

在 2003 年 Hiroaki[10]等人提出的研究中，預估 C_{OOV} (OOV words 類別)的 class n-gram 機率時，將 OOV words 視為名詞的一種，由 C_{noun}(名詞的類別)discount 出一定比例的機率以供 C_{OOV} 使用，這個比例 λ 即為 OOV rate。而 OOV 類別內部的 Intra-word constraints 則拆解為 sub-word sequences，以 bi-gram 機率預估之。另又針對 C_{OOV} 可能具有不同的 intra-word constraints ，細分為數類 $C_{OOV,i}$，這些不同的 multiple intra-word models 標示為 PM_j，最後把機率的預估寫為：

$$\begin{aligned} P(OOV \mid C_A) &= P(C_{OOV} \mid C_A)P(OOV) \\ &= \sum_{i,j} P(C_{OOV,i} \mid C_A)P(OOV \mid PM_j) \\ &\approx \max_{i,j} P(C_{OOV,i} \mid C_A)P(OOV \mid PM_j) \end{aligned}$$

在所有詞的類別中，named entity words 諸如人名、地名、組織名等詞類會與訓練語料息息相關，又因為中文的特性，造成人名/組織名為一個 open set，使得很多字/詞在訓練時是 unseen 的，而這些詞也經常落入 OOV words 類別之中。若將 named entity 的資訊使用在辨識中，可幫助取得斷詞邊界等語法上的資訊。在以往的研究中，named entity 多用於兩階段式辨認來強健辨識結果，如 Lufeng ZHAI 等人[11]提出採用 N-best list 在第一級結果進行標記，再輔以不同參數做 re-scoring(在 L ZHAI 的研究中使用了六個分數進行：acoustic, language model, number of words, number of phones, number of silence, NER score)。

由於 two-statg 的 re-scoring 方式受限於第一級辨識的結果，爾後在 NTT 所提出的研究中[12]，他們對 named entity 的標記於前級完成，並將這些 named entity words 視為一個類別收錄，讓辭典收錄詞數高達 1.8M；該篇論文中並使用了 on-the-fly composition 的技術，不將 named entity 類別如此大量的詞在 language model 上全展開，而在 decoding 時才進行，由於整張 graph 不是 fully-composed，因此在記憶體的使用上節省許多。

1.3 研究方向

在本論文中，承襲了有限狀態機的建構方式，進而改以有限狀態轉移器實作系統。我們在語言模型層進行改良，從選詞方法到訓練語料的挑選都有更動，並依照中文的特性提出若干改進方式；另一方面也注重處理 OOV words 相關的研究，利用有限狀態機的特性將不同的模型整合在一起，希望能建構出一個泛用的大詞彙辨認系統。

以往為解決傳統辨認無法收錄大量詞彙的問題，多採用兩階段式的辨認方式，先由第一級模組辨識出 word/subword 單元之 word lattice ；第二級模組中，對於 word lattice 上的 subword 單元重新構詞;最後於第三級模組重新給予 word lattice 上語言模型的分數。但此法受限於第一級辨識出 word lattice 之涵蓋率,如何將 subword 重構也是一個重要的問題。在本研究中，針對 OOV 問題中人名的部份進行處理，並提出以一階段方式進行辨識的演算法。

1.4 章節概要說明

本論文一共分為六章，其各章節內容分配如下：

第一章：緒論

第二章：有限狀態機辨認系統之建立

第三章：語言模型改進與階層式語言模型

第四章：使用階層式語言模型於有限狀態機語音辨認系統

第五章：實驗結果及分析

第六章：結論與未來展望

第二章 有限狀態機辨認系統之建立

本章將介紹本研究中所使用的有限狀態機語音辨認系統，包含所使用之聲學模型、選詞方式及語言模型。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；在挑選詞典的時候，引入資料檢索的 TF-IDF 概念來收錄較泛用的詞彙，語言模型則由大量文字語料庫以 n-gram 方式訓練來得到詞與詞之間相接的機率，由此來提升語音辨識率。

2.1 有限狀態機與相關演算法簡介

2.1.1 加權有限狀態轉換器

加權有限狀態轉換器(Weighted Finite-State Machine, WFST)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。

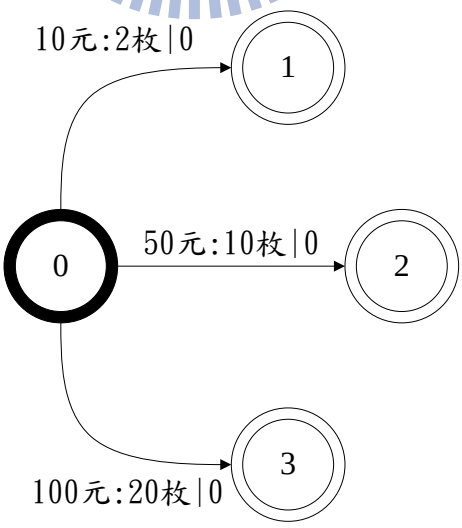


圖 2.1: 兌幣機之有限狀態機

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 $P = p_1p_2...p_n$ 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：

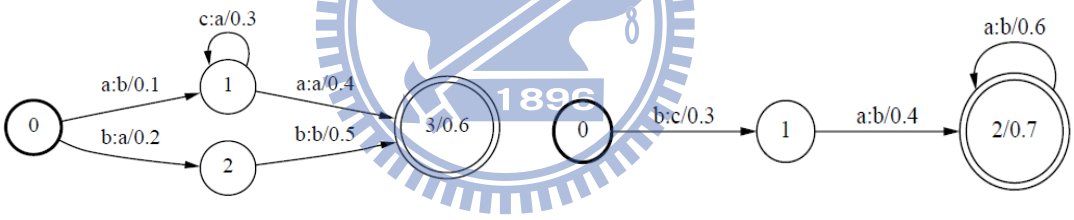


圖 2.2: 有限狀態機 A(左)與 B(右)

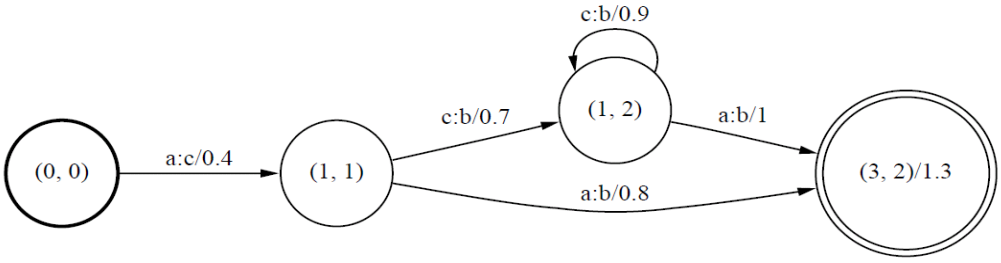


圖 2.3: 有限狀態機 C=A。B

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 2.4: 有限狀態機 A(左)與 B(右)

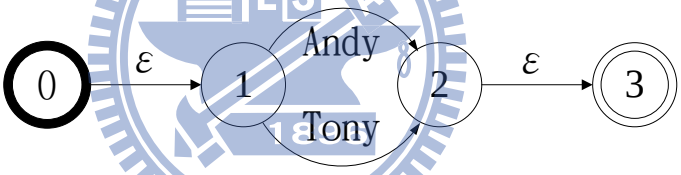


圖 2.5: 以 B 取代 A 上含有#Name label 之 arc

2.2 語音辨識系統之建立

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。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

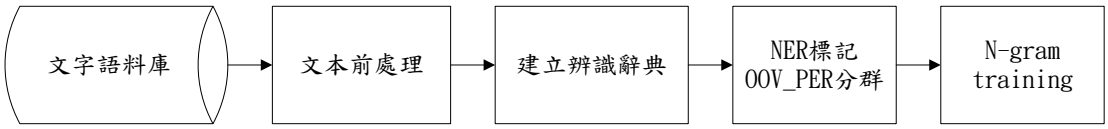
的測試語料為交通大學與成功大學的長句音檔，共 19 位語者 226 句長句音檔，長度約 2 小時，詞總數量為 15497，每個句子平均含有 117.2 個音節。

表 1.1: TCC-300 語料庫統計表

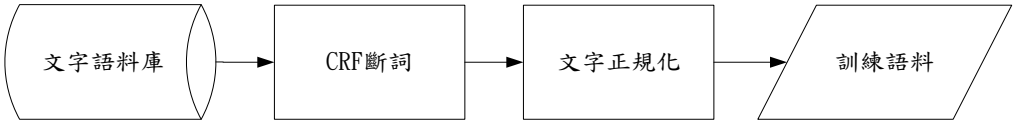
學校名稱	文章屬性	語者總數		總音節數		音檔總數	
台灣大學	短文	男	50	男	27,541	男	3,425
		女	50	女	24,677	女	3,084
		總數	100	總數	52,218	總數	6,590
交通大學	長文	男	50	男	75,059	男	622
		女	50	女	73,555	女	616
		總數	100	總數	148,614	總數	1,238
成功大學	長文	男	50	男	63,127	男	588
		女	50	女	68,749	女	582
		總數	100	總數	131,876	總數	1,170

2.2.2 文本前處理

在訓練語言模型前，我們針對語料庫的文本進行若干前處理，大致上分為：斷詞、文字正規化、NER 標記...等。之後再以統計方式得到訓練用詞典並且區別出不收錄於詞典中的 OOV words，利用處理過後的語料進行語言模型的訓練。訓練流程如下：



其中文本前處理的步驟又可以再細分為以下數個步驟：



1.) CRF 斷詞：

以條件隨機域(conditional random field, CRF)[13]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。

2.) 文字正規化處理：

2.1) 部分數量詞切短：

在 CRF 斷詞結果中屬於 Neqa、Nf 詞性的數詞皆是構成長詞，但將所有的長串數詞收錄在詞典中並不合理，我們將若干數詞切短，以短詞形式收錄於詞典中，如此也能增加詞典的涵蓋率。

例如：[X 分之 Y_Neqa]切短為[X 分之_Neqa] [Y_Neu]

例如：[XX 學年度_Neqa]切短為[XX_Neu] [學年度_Neqa]

2.2) 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。

2.3) 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。

2.2.3 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 2.2: 漢字形音義異同表

形	音	義	現象	範例	處理
同	同	異	多義字	挨（靠進、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。

補回同義詞資訊的方式採用取代演算法進行，取代演算法的作用規則是將一個 WFST 上的某些轉移(Arcs)取代為另外一個 WFST。我們需定義出：

1.) 欲取代該非終結符(non-terminal label)的 WFST

2.) 非終結符在 root WFST 上的對應編號

而輸出的 WFST 便是將 root WFST 所有帶有該**非終結符**的轉移取代成與其編號對應的 WFST，此外，在該轉移上的分數也能被保留下來。

較精確地說，一個從 state **s** 到 state **d**，帶有 non-terminal output label **n** 的轉移，欲用 WFST **F** 來取代之。進行演算時，這條轉移會將 state **s** 接至 **F** 的 initial state，並以空轉移符取代原本輸出的**非終結符**，而原本落在該轉移上的分數便不做更動，接著以一條空轉移將 **F** 的 final state 連至 state **d**。

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，在這裡可以將被合併的詞視為一個 non-terminal label，而我們欲使用帶有完整資訊同義詞的小型 WFST 取代它，如此這組同義詞就能享有相同的語言模型分數。

以下提供一個 n-gram root LM WFST、variant word WFST 與執行完 Replace 演算法後整合在一起的 WFST 結果。此 variant word 的對照表由左至右為訓練 n-gram 時的 non-terminal label 與其分別對應到的 word label。

表 2.3: variant word 對照表(部分節錄)

# in WFST	Word	# in WFST	Word	# in WFST	Word
1693	力有未逮(SW)	80100	力有未逮	80300	力有不逮
25507	帕金森氏(SW)	80101	帕金森氏	80301	巴金森氏
25506	乏人問津(SW)	80102	乏人問津	80302	無人問津
…	…	…	…	…	…

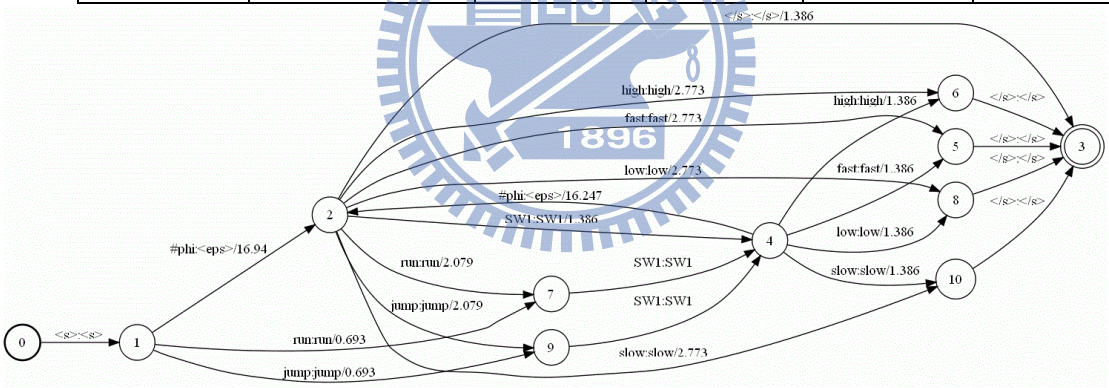


圖 2.6: n-gram Root LM WFST

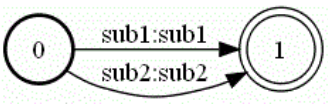


圖 2.7: Variant Word WFST

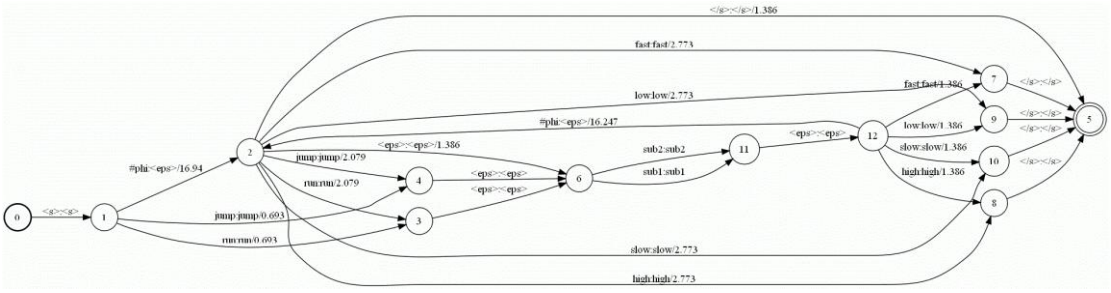


圖 2.8: Replaced Variant Word WFST in Root LM WFST

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“gong1 gei3”或“ gong1 ji3”。

而我們收錄字典的形式是以[Big5 碼_拼音]的形式收錄，於是“供給”的兩個異音詞寫為 A8D1B5B9_gong1gei3 與 A8D1B5B9_gong1ji3。在訓練語言模型時，只使用其中一種寫法訓練。因此我們要在語言模型層的 WFST 補回這個資訊，與 variant word 的處理方式一樣，建立小型的 multi-pronunciation WFST，再使用取代演算法重構回 n-gram root LM 上。

在異體字、同義詞的尋找上，目前實驗室是採用人工標記的方式進行，一方面檢查收錄的詞典是否有問題，另一方面將異體字與同義詞建檔整理。截至目前為止修正的詞共 1752 個，修正詞數共 108495 個。

針對異體字、同義字進行修正後，我們訓練出新的語言模型並對其計算 PPL，計算的對象為 TCC-300 的測試語料。

表 2.4: Baseline PPL

TCC - Testing	Order	ppl	ppl1
	3	365.862	487.08

表 2.5: 異體字、同義字置換後的 PPL

TCC - Testing	Order	ppl	ppl1
	3	364.588	485.23

2.2.4 IDF 選詞方法

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。

藉由 IDF 方法對詞頻統計出的候選詞表，重新計算收錄的優先順序，並統計兩種選

詞方法差異的詞數。以六萬詞的大小限制來看，使用 IDF 選詞而更動的詞數約 3700 個詞（詳細的表格收錄於附件檔案: IDF_diff.xlsx）。

表 2.6: 詞典 IDF 表(部分節錄)

Word	出現總數	文件數	IDF	Word	出現總數	文件數	IDF
光晞	204	1	3.913707914	大炒家	149	66	2.094163978
花羽露	181	1	3.913707914	長跑隊	149	66	2.094163978
浩天	225	2	3.612677918	范光陵	148	66	2.094163978
李文秀	445	5	3.21473791	西拉雅族	146	66	2.094163978
小樂	412	5	3.21473791	梁建銘	145	66	2.094163978
多麗	193	5	3.21473791	老花眼	144	66	2.094163978
娜克莉	169	7	3.068609874	白金卡	268	67	2.087633111
巫招明	193	10	2.913707914	省博館	257	67	2.087633111
林信元	157	10	2.913707914	哈定	245	67	2.087633111

原先收錄詞頻的出現總數落在 144 次，可看到許多人名因為出現次數高而被收錄進了詞典中，但僅出現在少數的文件中；而像是"老花眼"一詞卻落在了收錄詞典的邊緣，這表示 IDF 選詞方式能使詞典收錄到較泛用的詞語。

以下分別以詞頻挑選詞典與 IDF 方法挑選詞典來計算其 PPL，計算的對象為 TCC-300 的測試語料。

表 2.7: Baseline PPL

TCC - Testing	Order	ppl	ppl1
	3	365.862	487.08

表 2.8: 使用 IDF 法挑選詞典之 PPL

TCC - Testing	Order	ppl	ppl1
	3	364.372	484.683

使用 IDF 方法挑選詞典，能使得 PPL 下降。而被剔除的人名等資訊，若都被歸類

為 OOV words 似乎有些可惜，在本研究中，希望能將這些資訊進行有效的利用，在關於 Hierarchical Language Model 的章節將一併介紹之。

2.2.5 語言模型的建立

在本研究中使用了目前運用廣泛的 *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：

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

另外，當 $Count(\cdot)$ 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。

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)：它是一套由中研院錄製，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。

將文字資料庫共進行文本前處理後，可得到詞數共 386,939,797 個，再對其計算出挑選出 IDF 值最高的前 8 萬詞以供語言模型訓練所用，平均詞長為 2.44 個字。

2.2.6 聲學模型的訓練

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 2.9: 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

2.2.7 有限狀態機的整合

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（Acoustic Model）、詞典（Lexicon）以及語言模型（Language Model），並透過有限狀態轉換器將此三部分整合（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。

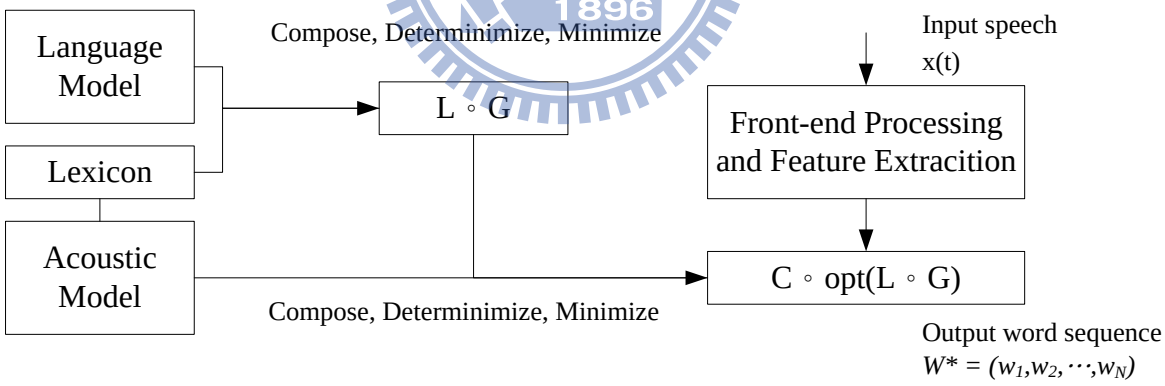


圖 2.9: 語音辨認系統架構圖

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[15]。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[16]進行。

1.) 聲學模型：

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。

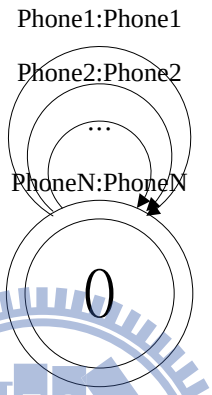


圖 2.10: 聲學模型的 WFST

2.) 發音詞典：

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。

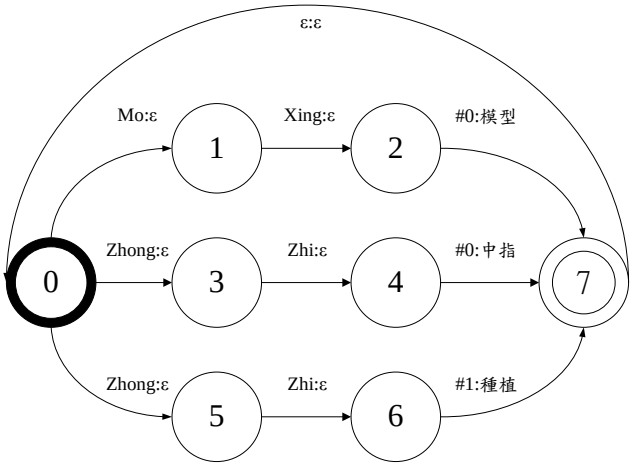


圖 2.11: 發音詞典的 WFST

3.) 語言模型：

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：

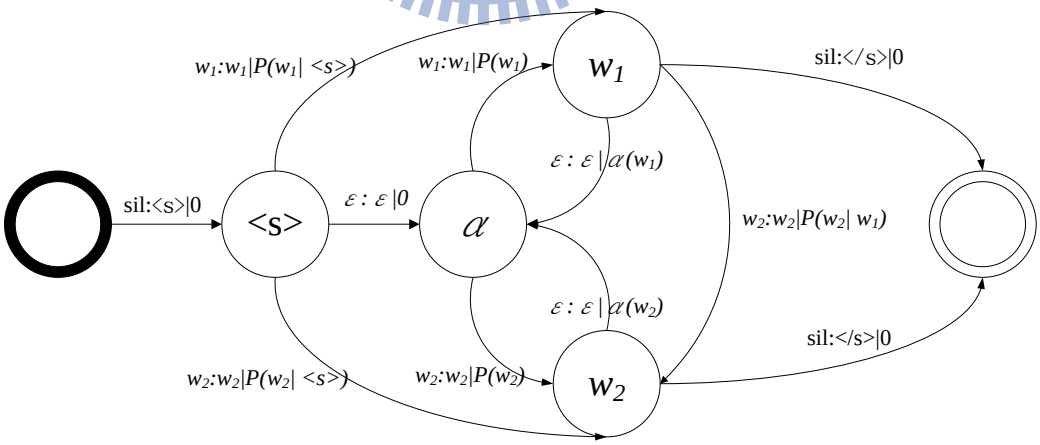


圖 2.12: 雙連語言模型圖

第三章 階層式語言模型

在傳統辨識上，辨識率一直受限於詞典大小，而中文構詞彈性的特性使得某些詞類為 open set，諸如數詞(Neu)、專有名詞(Nb)…等，考慮到辭典的大小並無法收錄所有詞可能的組合，但如此一來落在詞典外的詞(OOV, out-of-vocabularary)就無法被辨識出來。這些 open set 中的若干詞彙具有較明顯的構詞規則，例如：定量複合詞(DM)、中文人名(PN)與綴詞(MD)…等。

本研究針對中文人名詞彙進行討論，中文人名可視為「姓氏」與「名字」的組合，若將無法收錄的人名皆視為 OOV 處理過於可惜。在此我們將 OOV 人名視為一個類別處理，利用人名與前後詞的關聯性以 n-gram 模型訓練之，並將此類別中的人名切短，以較少的單元來涵蓋無法收錄的詞彙，另外將類別內的單元組合訓練出語言模型，最後以取代演算法將兩個語言模型整合在一起。

3.1 人名語言模型的訓練

3.1.1 Named Entity Recognition

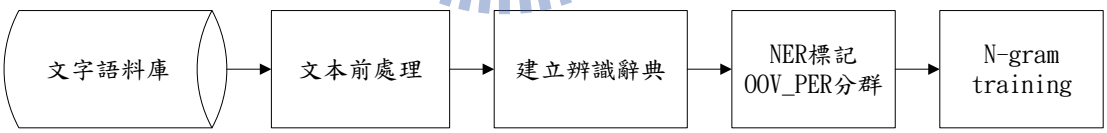


圖 3.1: 文字資料庫處理流程

在第二章的文字處理流程中，我們以 CRF 斷詞器進行斷詞並依照每個詞的 IDF 值來決定收錄詞典，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。為了偵測出文本中屬於人名的詞，採用王逸如老師撰寫之 NER 斷詞器，對 CRF 斷詞後的結果進行 Named Entity Recogntion，一共可得到 PER 標記數量 7,320,020 個。

在這份 PER 標記中，除中文人名外，尚有音譯之外文人名（如：席維斯史特龍、戈巴契夫…等）、常見古代人名、特殊人名（如：上帝、聖誕老人、文昌公、孔子…等）。

3.1.2 OOV 中文人名的選擇與拆解

考慮到某些重要人名對前後文的關聯性高，若將所有人名都當作一個 class 看待，在訓練語言模型時會丟失這個特性。因此我們將 PER 標記中 IV 與 OOV 的人名分開，經 NER 斷詞器取得人名標記後，若被標記的人名沒有收錄於辭典中，則將該詞轉寫為 OOV_PER，當作一類 class 進行訓練，並將此人名收集起來另外訓練人名模型。

< 例句.1 > [… 台北 市長 陳水扁_PER 將 …]

由於 ”陳水扁” 已收錄在詞典中，我們不對這個句子更動。

< 例句.2 > [… 女 童軍 楊惠敏_PER 現 尚 在 台灣 …]

“楊惠敏” 不在收錄詞典中，此人名屬於 OOV_PER，重新轉寫文本為：

[… 女 童軍 PER_3 現 尚 在 台灣 …]

以 IDF 挑選之詞典為基準，除去在詞典中的人名後，尚餘 441,813 類人名，總計出現 2,345,468 次。又，剩餘的人名標記中尚含有外文人名、特殊人名…等，不在我們處理的範疇中，因此依照中文常見姓氏與詞長篩選出可能為中文人名的候選詞，詳細的處理可分為：

1.) 詞長判斷：

NER 斷出的人名標記有可能是「姓氏」、「姓氏+名字」或「名字」，而中文姓氏多為單姓，偶有複姓；而單名的情況也較少。

綜合以上結果，將 OOV 人名詞長>5 的結果去除。

2.) 姓氏斷詞：

以百家姓判斷此詞首是否為姓氏，若為姓氏則再將人名斷開成「姓氏+名字」；

3.) 同音合併：

在語音辨識中，同音異形的人名將會造成混淆，例如「江」與「姜」，無從得知輸入之語音到底是念哪個「ㄐㄧㄤ」。由於聲學模型的分數一樣，則除了在語言模型分數最高的該條路徑，其餘皆不會勝出，這會造成許多冗贅的路徑。

針對斷開的「姓氏」、「名字」字元，將發音相同的字元組合併，取出現次數較高者

顯示。在文本前處理時，原收錄姓氏共 128 個，去除同聲調剩下 117 種，再將同音節合併後剩下 104 種字元。名字部分合併後則剩下 379 個字元。

NER 標記中文人名的 F-measure 統計如下，測試組使用的是 TCC-300 測試語料。

表 3.1: NER 中文人名標記之 F-measure 統計

Precision	Recall	F-measure
94.64%	89.52%	92.01%

3.1.3 人名語言模型的建立

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。則預估人名的機率式可拆解為原始的 n-gram root LM 與 PNLM 這兩個模型。tri-gram 的機率預估如下，bi-gram 與 uni-gram 類推之：

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

而人名的機率預估則寫作：

$$P(w_i \mid w_{i-1}, w_{i-2}) = \begin{cases} P(PN \mid w_{i-1}, w_{i-2}) \cdot P(c_1, c_2, c_3 \mid PN_{len}), & w_i \in PN_{len},\ len = 2 \sim 4 \\ P(w_i \mid w_{i-1}, w_{i-2}), & \text{otherwise} \end{cases} \quad (3.2)，$$

1.) 外部機率(Inter-word probability)的預估：

我們將 OOV 人名與前後詞的關聯性使用 n-gram 模型來預估，但將所有的 OOV 人名視為一類不夠妥當，現依據詞長將 OOV 人名分為二字、三字、四字共三群，分開訓練其與前後詞的關聯性。

表 3.2: OOV 中文人名的長度分佈

Word	Counts	Percentage
中文人名總數	6,677,902	
IV_PER	4,706,169	70.47%
OOV_PER_len3	1,663,575	24.91%

OOV_PER_len2	195,542	2.93%
OOV_PER_len4	112,616	1.69%

2.) 內部機率(Intra-word probability)的預估：

由於中文人名可視為「姓氏」、「名字」的組合，我們假設名字與姓氏的組合是無相關的，而名字的機率則使用 n-gram 預估之。以三字詞人名為例，其機率寫作：

$$P(c_1, c_2, c_3 \mid PN) = P(c_1 \mid LastName) * P(c_2 \mid FirstName) * P(c_3 \mid c_2) \quad (3.3)，$$

以 n-gram 訓練人名的 WFST 圖形時，我們不放入 back-off state 以避免在 decoding 時走出多餘的字元而造成搶詞的情況。

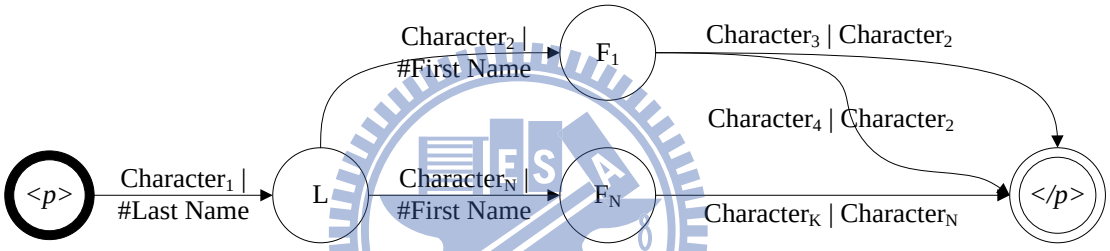


圖 3.2: 三字詞 PN model 之 WFST

3.2 階層式語言模型之整合

訓練完語言模型後，要將此 PN model 重構回原本的 root LM 之中，在此使用了有限狀態機的取代演算法實現之。PNLM 是針對 OOV 的人名類別所訓練的，要將原始 LM 上帶有輸出符號為 OOV_PER 的轉移取代為 PNLM 模型。

如下所示：

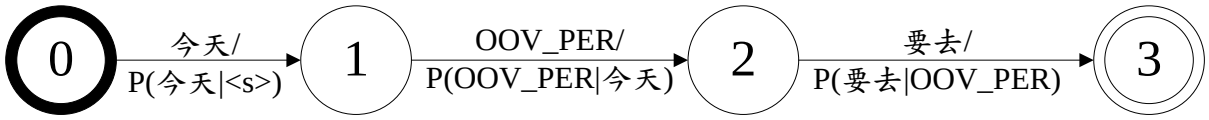


圖 3.3: Root LM 之 FST

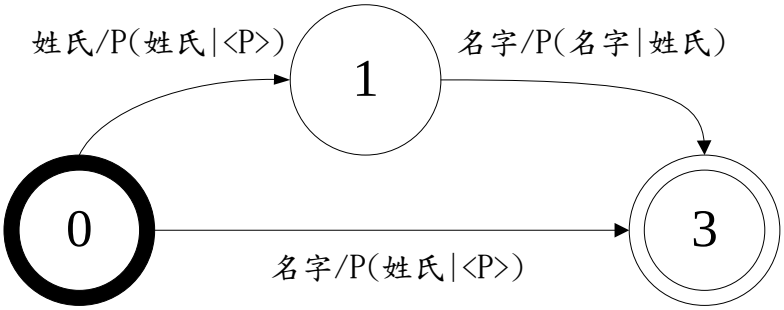


圖 3.4: PNLM 之 WFST

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)，再依 replace 演算法重構回 n-gram word graph 上。由圖一可看出 Inter-word 的分數原被放置在帶有 OOV_PER 非終結符號之轉移上，經過取代演算法後，被一條進入 PN model graph 的空轉移所繼承了(圖三)。

另外需注意到 word insertion penalty 應以整個 PN model 為單位加入，不要在每個 intra-word model 的輸出字元上都給予懲罰。

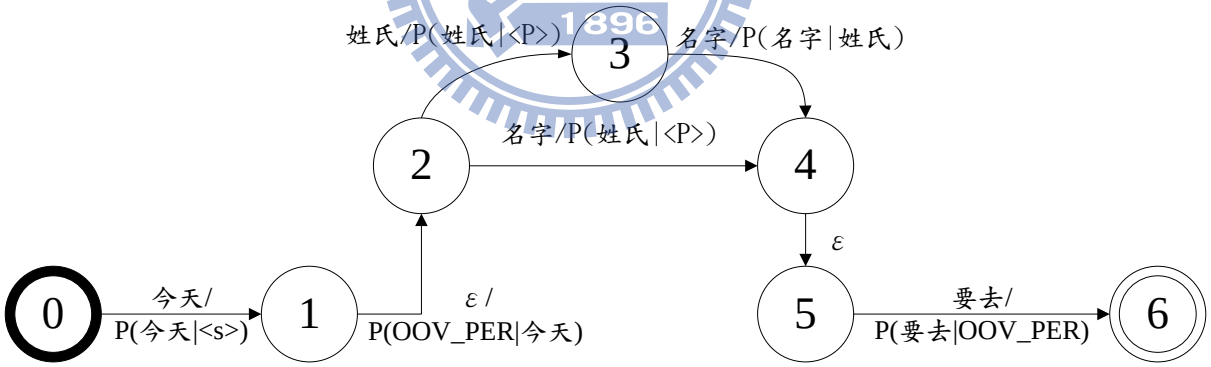


圖 3.5: Replaced PNLM in Root LM FST

3.2.1 取代演算展開之數量級

為了預估取代演算法將 root LM graph 展開後的數量級，我們將這些 OOV 的人名全部作為一個 class 來訓練 tri-gram LM，共可得總轉移數的數量級為 10^7 條，而帶有 OOV_PER non-terminal label 之轉移的數量級為 10^5 條。

由於我們欲使用取代演算法將 PN model 置入 Root Language Model 中，若 PN model 過於複雜，將使得演算過後的 graph 過大而無法向下進行詞典層的展開，這也影響我們在預估內部機率時可用的模型。在實驗時我們用了三組不同的 PN model 設計來觀察取代演算法展開後的影響：

1.) 直接以 Tri-gram 方式訓練 PN model：

取代演算法執行後展開過大的 graph(10^10)，向下與詞典層執行組合演算時耗費過大的記憶體而無法進行。(Root LM graph 的 arc 數僅為 10^7 條)

2.) 將姓氏與名字分開作為兩個類別，名字部分以 bi-gram 訓練之：

即我們原本預想的人名模型，在前述以 bi-gram 模型訓練人名模型時，共可得狀態數約 700 個，轉移數 30,000 條，取代完之轉移數有 1,345,016,885、狀態數 88,811,484。取代演算法可以成功實現，但取代完成之 graph 仍過大，導致讀入 decoder 時發生記憶體不足的情況。

3.) 將姓氏與名字分為兩個類別，皆以 uni-gram 訓練之：

最後我們退而求其次，選擇以 uni-gram 模型來製作人名模型，以三字詞人名為例，轉移數量共 104(姓) + 379(名字 A) + 379(名字 B)，狀態數量 5 個。取代完之轉移數量為 14,898,762 條、狀態數 76,731,662 個。

在記憶體為 24G 下的情況，此數量級可以實現全部的 WFST 演算法，因此採取這個設計進行辨識。

3.2.2 一階段式辨認之實驗結果

原先在辨識率的計算上，必須是完全相同的詞才算一個 hit，在觀察人名的辨識結果時，我們在乎的不外乎是：

1.) 是否能辨認出一句話中人名的正確位置

2.) 是否能辨識出人名的正確發音。

由於發音相近的音節容易發生混淆，在計算人名的辨識率時，若辨識出之人名詞落

在詞串中正確的位置時，視為正確的 hit 來計算，而不考慮其字元的正確性，若將 IV 人名辨識為 OOV 人名則不計入 hit。

與 OOV 人名相關之數據如下：

1.) 辨識所用的 Testing data 共 226 句

2.) 出現人名的句子共 129 句，共計出現 369 次人名

3.) 369 次中，屬於 OOV_PER 的人名共計 161 次

4.) 161 次中，三字中文人名共 109 次(共 64 位)、二字人名 16 次(5 位)、四字人名 3 次(1 位)，其餘 33 次為外國人名。因此我們目標要辨識出的中文人名共 128 個。

若 PN model 能將 OOV 中文人名共 128 次完全辨識出，辨識率的提升將由 73.36% 到 74.07%(不含前後詞更正)。

表 3.3: OOV_PER 一階段式辨識結果

	tri-gram root LM with uni-gram PN model	Tri-gram LM (without PN)
Word Acc	73.47%	73.36%
Syl. Acc	87.00%	86.07%

實際觀察辨認答案中屬於 OOV_PER 但沒有被 PN model 抓出之人名，可以發現有進入 PN model 的人名大多都有稱謂詞(title)支撐，另外也有被 PN model 救回的稱謂詞，或是將非人名的詞錯誤辨識出之結果，如下所示：

表 3.4: 正確辨識人名之範例

正確答案	未加入 PN model	加入 PN model
NCKU_f070307_0		
理事長	理事長	理事長
郭振興	果真	郭振信

針對	新	針對
	針對	

表 3.5: 無法辨識出人名之範例

NCKU_f070308_0		
演講	演講	演講
中	中國	中國
郭振興	振興	振興
深入淺出	深入淺出	深入淺出

表 3.6: 人名模型救回前後詞之範例

NCTU_f010407_0		
銀行	銀行	銀行
總經理	重心	總經理
王昭慶(OOV_PER)	移往	王昭欽
在	較	在
	輕	
	在	

我們並以 F-measure 分數來評估人名模型的效能，數據如下：

表 3.7: OOV_PER 一階段式辨識之 F-measure

All find	Golden hits	IV hits	Wrong hits	Precision	Recall	F-measure
34	28	0	6	82.35%	21.88%	34.57%

Golden hits 代表抓取到的結果完全正確的 hits 數，IV hits 為抓出的結果屬於 inside

vocabulary 的 hits。在此實驗中顯示 Precision 數值高但 Recall 數值低，表示大部分的 OOV_PER 皆是 missing detection，此問題來自於 PN model 中使用的 uni-gram model 過於簡化，導致分數太低而無法走出 PN model。

由於受限於記憶體的限制，無法採用過於複雜的 PN model，否則會造成組合演算與最佳化演算無法進行，抑或是無法讀入 decoder 之中，展開的問題源自 tri-gram language model graph 的特性，由於每個 state 需要記憶前兩個 word 的資訊，表示在 tri-gram 層級展開的 PN model 無法共用，否則將失去更前一個 word 的資訊，但計算 intra-word 分數所用的 sub-net 與 inter-word 分數並無關聯，這些額外展開來計算 intra-word 分數的 PN model 佔據了相當的記憶體空間。為了能使用一個比較可靠的 PN model，下面以 2-stage 的做法來進行整個語音辨識。

3.3 兩階段式辨認系統

承襲前述 n-gram language model 的特性，在 tri-gram LM 中為了保留住前兩層 word 的資訊，無法在 graph 上將 PN model 整合在一起。以前述所使用的 tri-gram LM 為例，需要展開的 PN model 共有 34,578 個；但在採用 bi-gram LM 的情況下，僅需展開一個 PN model 用於計算 intra-word 分數。因此我們在 1-st stage 採用 bi-gram LM 與 bi-gram PN model 產生 lattice，再以 tri-gram LM 對其進行 rescoring。如下所示：

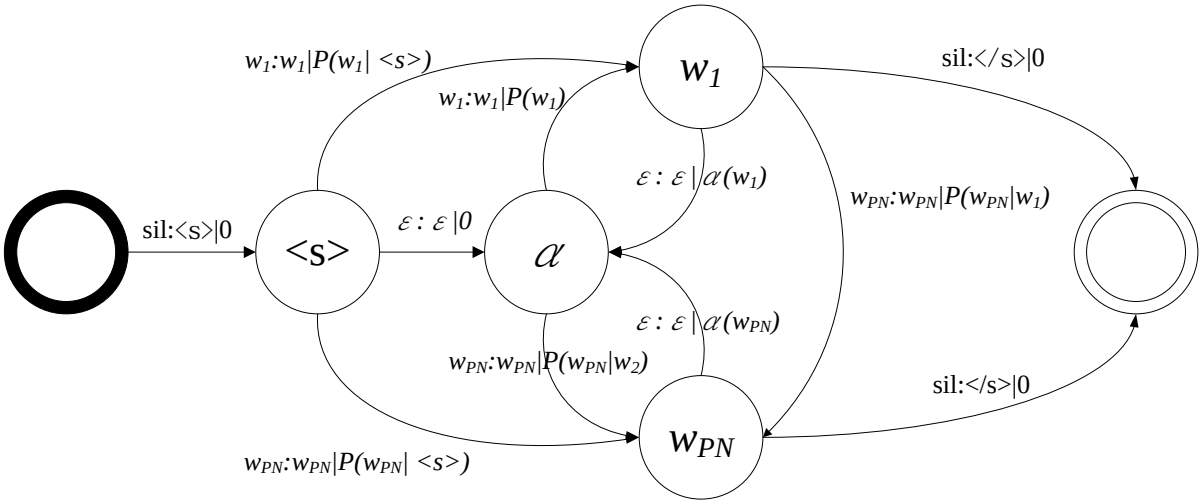


圖 3.6: Root bi-gram LM graph

利用 bi-gram graph 僅記憶前一個 word 的特性，只需一個狀態來表示前一個詞，因此需要被展開的 non-terminal PN label 就可以共用同一個 sub-net 來計算 intra-word 分數，如此一來我們就可以將一個較複雜的 PN model 模型置入 root LM graph 上。

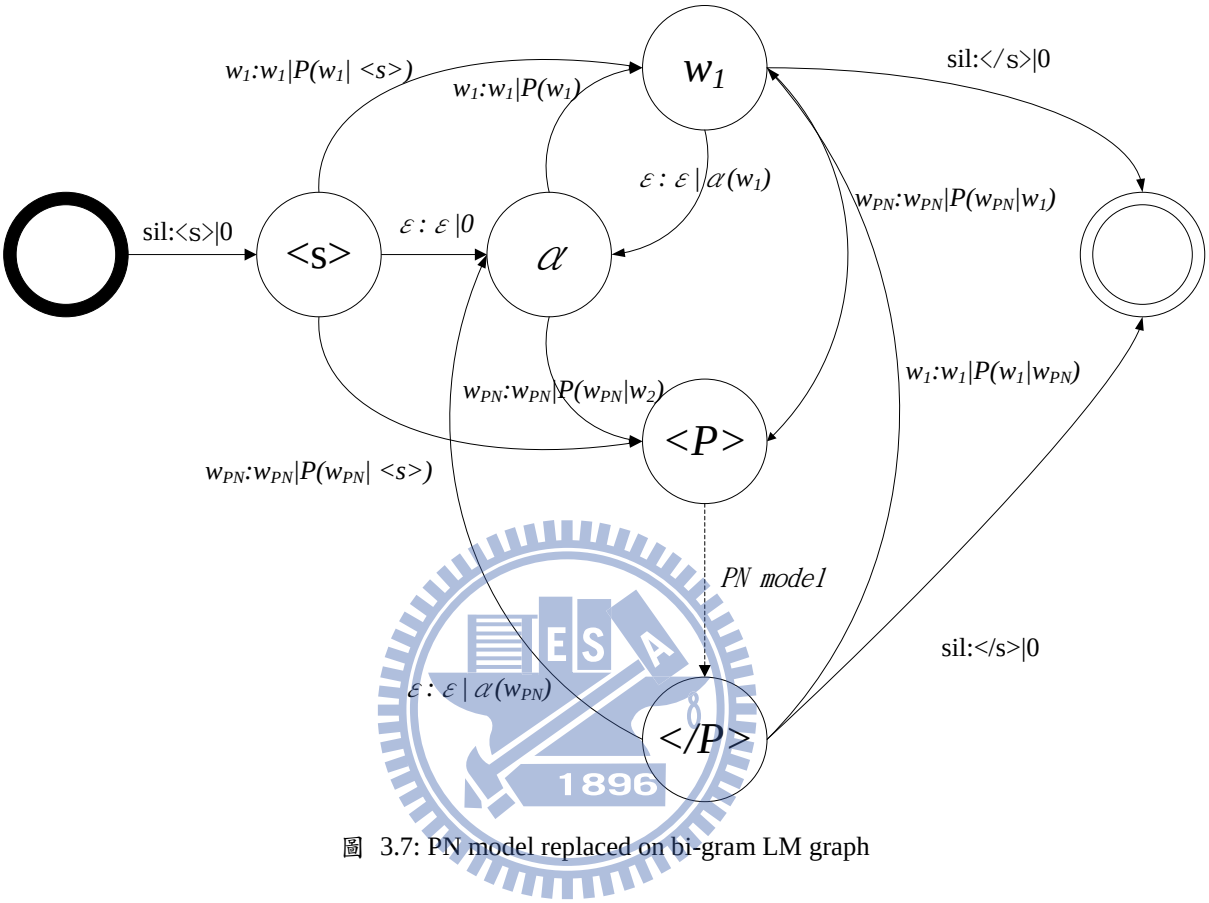


圖 3.7: PN model replaced on bi-gram LM graph

3.3.1 遭遇問題

在採用 Juicer decoder 進行辨識時，遭遇問題如下：

1.) 辨認所用的 FST graph 經過最佳化處理，若我們僅希望對人名部分進行 re-scoring，必須回查沒有經過 optimize 的 graph 以確保分佈在人名相關之 arc 上的分數是僅和人名有關的。

2.) 輸出的 FSM lattice 上每個 arc 的分數為 AM 與 LM 分數相加的結果。

事實上這兩個問題是相同的，因為 decoder 產生出的 lattice 沒有將 AM 與 LM 的分數分開，無法僅針對 LM 分數進行 Rescoring。因此對於 decoder 進行修改，使其僅輸出

AM socre，得到一個僅有 AM 分數的 FSM lattice 後，我們再以組合演算法將 LM 分數進行 rescoring。

3.3.2 兩階段式辨認系統架構

整個實驗可以分為兩個 stage 來看，1-st stage 產生出僅有 AM 分數的 word lattice，2-nd stage 再對 lattice 上的 word 進行 rescoring，以下為整個語音辨識的流程圖：

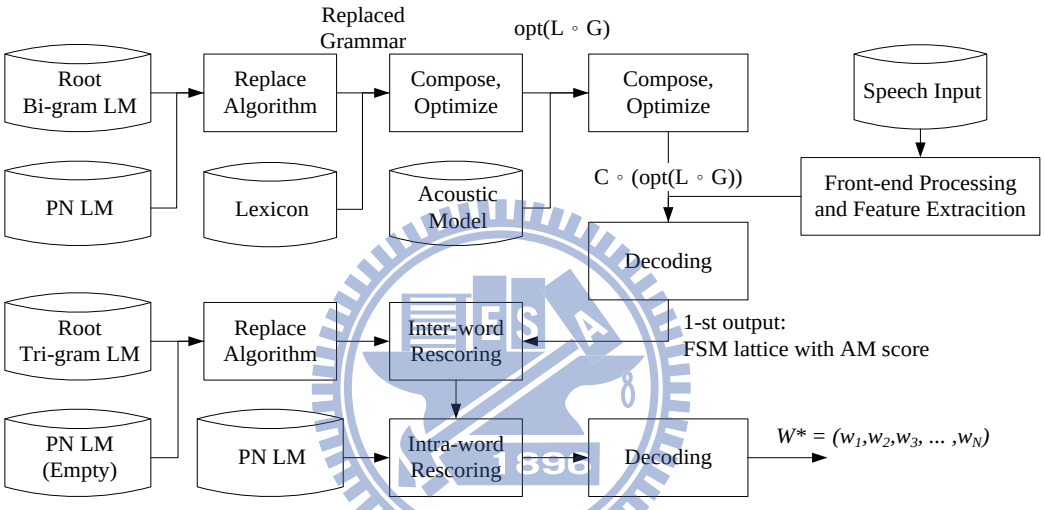


圖 3.8: 兩階段式辨識流程圖

1.) First stage – Lattice generate

a.) root LM model:

承前述理由，在製作第一級辨識的 grammar graph 時，採用 bi-gram 做為 root LM。

b.) PN model:

在 root LM 採用 bi-gram 的情況下，可採用較複雜之 sub-net，因此把計算「名字」分數的 PN model 改回 bi-gram 的方式來預估：

$$P(c_1, c_2, c_3 \mid PN) = P(c_1 \mid LastName) * P(c_2 \mid FirstName) * P(c_3 \mid c_2)$$

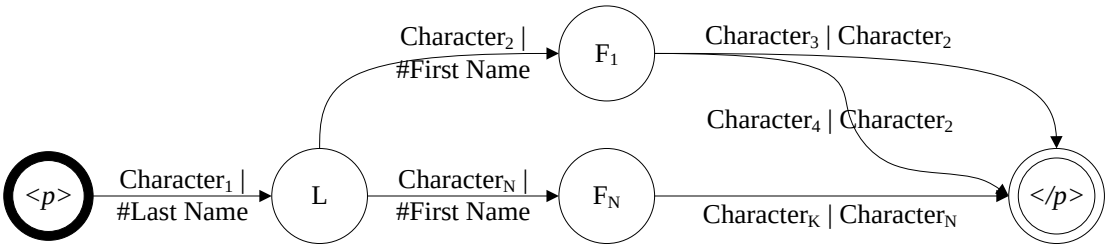


圖 3.9: Bi-gram PN 之 WFST

2.) Second stage – LM rescoring & PN rescoring

a.) LM (inter-word) rescoring:

採用 tri-gram LM 製作 grammar 層的 graph，由於沒有要向下 compose 到 lexicon 層級，在製作時不用加入輔助展開用的 auxiliary label。與 PN model 相關的 non-terminal label 則以一個沒有分數的 graph 取代之，因為結構簡單，因此取代演算法也可順利進行。將 word lattice 與這個 tri-gram graph 進行 compose 後，便可將 inter-word 的 LM 分數配置到 arc 上。如下所示，PN 內部不給定分數：

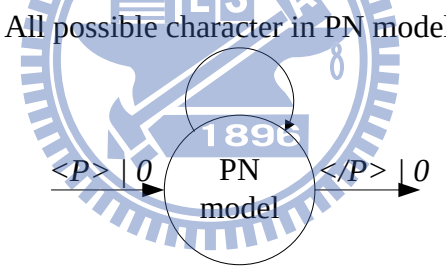


圖 3.10: empty PN model

b.) PN (intra-word) rescoring:

在設計第一級的 grammar graph 時，我們加入了不佔時間的 auxiliary label 來觀察辨識結果是否有進入 PN model 中，在進行 2-nd stage 的 rescoring 時，也可依靠 auxiliary label 來決定分數要配置在 lattice 的哪條 arc 上。如下圖，與 PN model 無關的 word 將不會配置分數。

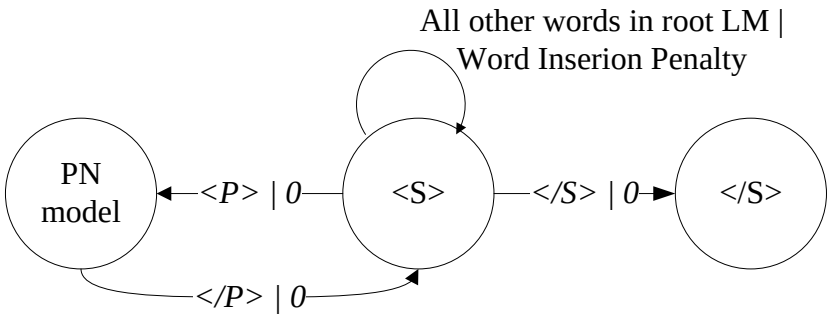


圖 3.11: intra-word rescoring

3.3.3 兩階段式辨認之實驗結果

首先列出的是 1-st stage 產生之 word lattice 上分數最高的 best path 辨識結果，以及 one-pass recognition 的 baseline 實驗數據，最後是以 tri-gram LM 對 inter-word 進行 rescoring 的結果。

在進行第二階段的 rescoring 時，我們採用兩組不同的 LM 設定進行，設定(A)的 tri-gram LM 與一階段辨識所用的 root tri-gram LM 相同；設定(B)的 tri-gram LM 在進行訓練時設定之 discount 值小於第一組的 discount 值，意即此設定較(A)組更為精細。我們之所以不在一階段辨識時使用(B)組之 tri-gram LM，是因為其訓練出之狀態數與轉移數過於龐大，使得無法向下展開至詞典層。但使用在兩階段的 rescoring 時，我們只需要詞與詞的相接機率而不用對此語言模型做展開與優化的動作，因此可採用較精細的語言模型給予 inter-word 之間的分數。

表 3.8: One-pass recognition results

Models	Word Accuracy
Tri-gram LM	73.36%
Tri-gram LM with uni-gram PN model	73.47%

Bi-gram LM	71.76%
Bi-gram LM with bi-gram PN model (Lattice generation)	71.82%
Rescoring with tri-gram LM (A)	72.32%
Rescoring with tri-gram LM (B)	76.27%

表 3.9: 人名模型標記之 hit 數

Models	All find	Golden hits	IV hits
Tri-gram LM with uni-gram PN model	34	28	0
Bi-gram LM with bi-gram PN model	67	43	3
- Rescoring with tri-gram LM (A)	72	42	6
- Rescoring with tri-gram LM (B)	62	40	2

表 3.10: F-measure

Models	Precision	Recall	F-measure
Tri-gram LM with uni-gram PN model	82.35%	21.88%	34.57%
Bi-gram LM with bi-gram PN model	64.18%	33.60%	44.11%
- Rescoring with tri-gram LM (A)	58.33%	32.81%	42.00%
- Rescoring with tri-gram LM (B)	64.51%	31.25%	42.10%

在產生 WFST 的 lattice graph 時，無法將全部的狀態都留下，我們觀察實驗結果就可發現：在相同的語言模型設定下，使用兩階段式的方式進行辨認，將會使得詞辨識率較一階段式辨認來得較差；除非使用較精細的語言模型重新進行給分才能取得較好的 inter-word 分數進而提升詞辨識率。針對 OOV 人名的部份來看，使用不同的設定去估算 inter-word 時，對 F1 分數並無太大的影響，但相較於 one-pass 辨識僅能採用較簡單之 uni-gram PN model，顯然使用兩階段的做法可以偵測到較多人名。

第四章 即時展開取代演算法

在第三章我們提出了以兩階段式的做法進行語音辨識，可使用較精細的模型來完成辨識而得到較一階段辨識較好的詞辨識率與 F1 score。同時也發現了 WFST 在整合不同模型時會展開一些可化簡的路徑，因此在本章提出一個在 run-time 階段才進行展開的 on-the-fly replace 演算法。

4.1 取代演算法與 n-gram 模型

n-gram language model 假設任一個詞在詞串中只受到前 *n-1* 個詞的影響，在使用 WFST 圖形表示 n-gram 模型時，我們以每個狀態代表目前詞的歷史資訊(history)，如以 bi-gram LM 為例：

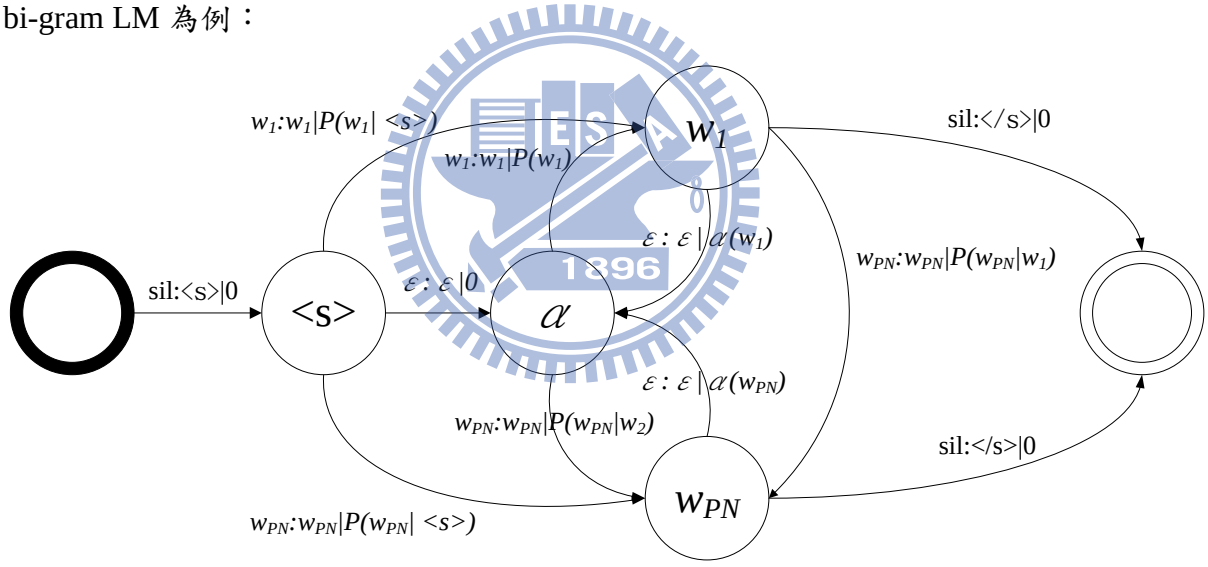


圖 4.1: Root bi-gram LM graph

可以發現從每個轉移上帶有的分數皆是按照其來源狀態來配置，因此當 n-gram 的 n 值上升時，就需要更多的狀態數來描述詞的歷史資訊。

而 PNLM 本身的設計是用來計算人名內部的 intra-word 分數，與 inter-word 分數無關，但以取代演算法在 root LM 上展開 PNLM 模型時，是對每條帶有非終結符號的轉移進行，每個轉移又會走到不同的目標狀態以表示現在的歷史資訊，等於以 PNLM 的 graph 來代表目前的 history，造成了許多重覆的狀態與轉移而導致取代後的圖形過於龐大。

4.2 即時展開取代演算法

在此提出一個在 run-time 階段才以動態方式去展開階層式模型的演算法，藉此降低 decoder 所需耗費的記憶體空間，進而實現以一階段方式完成搜尋，如此一來也可避免受限於 two-stage 做法時第一階段產生之 word lattice 涵蓋率不足的情況。

4.2.1 Juicer decoder

本次實驗中使用的辨認器之原始碼是由 Idiap research institute 發展的辨認器 Juicer，此辨認器採用 token passing 的演算法進行搜尋，並以 Viterbi algorithm 保留同一個狀態內機率分數較高的 hypothesis。

存活的 token 隨著每個音框送入時進行分數的更新與轉移，處理順序如下：

1.) 根據目前 token 所在的狀態內計算其 HMM model 的分數

2.) 依照 HMM model 分數與 graph 上的 LM 分數決定是否進行下一個轉移

3.) 如果該轉移的輸出字元不為 epsilon，則存入存活的路徑清單

4.) 對於所有存活的 token，展開下一組轉移路徑並加入待計算的 HMM 清單

我們針對步驟 4.)展開轉移時的進行改寫，在此加入 on-the-fly replace 演算法。

4.2.2 即時展開取代演算法

由於取代演算法會展開許多重複的狀態與轉移用來計算 intra-word 之間的分數，我們希望使用一個可共用的圖形來計算之，以降低記憶體的使用量。

如果將每條帶有非終結符的轉移都連接至同一個 PNLM 的 WFST graph 之中，等於失去了更前一層的歷史資訊，只知道前一個詞的資訊是 OOV_PER。因此我們仍需在圖形上保留住正確的歷史訊息。

在這裡提出的作法是：預先建構數個計算 intra-word 分數所用的 PNLM 之 WFST 做為 pool 備用。當進行 token 的轉移時，若發現該轉移 t 之輸出為 OOV PER 之非終結符，則動態地將此轉移 t 的目標狀態重導向 pool 中未啟動之 PNLM WFST 的初始狀態，並將該 PNLM 的終止狀態連接至轉移 t 之目標狀態的所有目標狀態上。一但 token 離開此

PNLM 或因分數太低而 pruning 掉時，再將此 PNLM deactive，如此一來此 PNLM 就可被重複的利用。

以下圖為例，我們以重新導向的 PN WFST 繼承詞的歷史資訊，進而得到一個等效在圖形上對全部的 PN 非終結符做展開的結果。

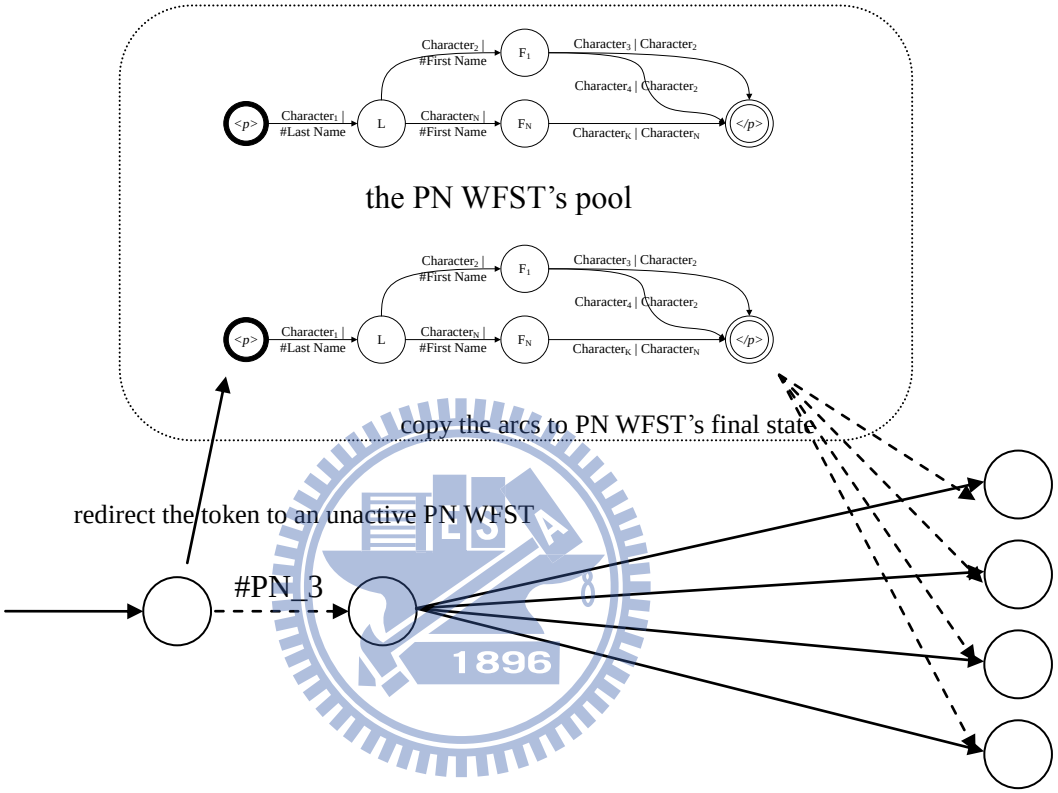


圖 4.2: 即時展開之示意圖

由前述實驗使用於 one-pass 辨識之 root tri-gram LM 而言，使用取代演算法共需要展開 PN model 數目為 34,578 個，而使用即時展開演算法僅需展開約 1/10 的數量就能在辨識時取得等效的圖形。

辨識的流程圖如下，原始的 decoder 即為左半邊的路徑，虛線代表我們處理階層式語言模型所加入之 on-the-fly replace 演算法。

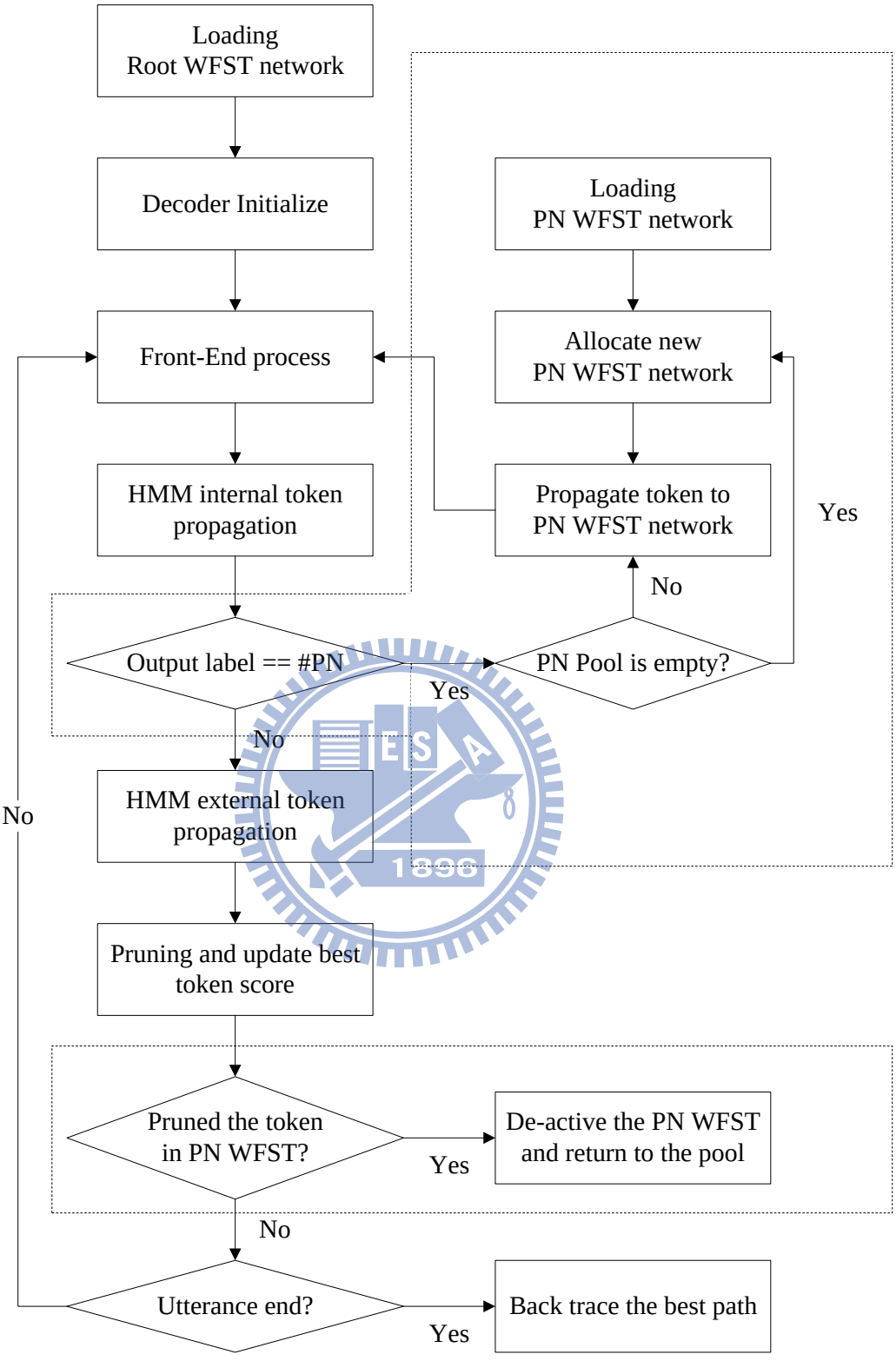


圖 4.3: The block diagram of on-the-fly replace algorithm

4.2.3 實驗結果與分析

實驗環境採用之 CPU 為 Intel Core i7 960 四核心處理器 3.20GHz，記憶體共 24G。

以 one-pass recognition 進行辨識時，每個音框內 token 的最大存活數(max hyps)定為 6000，而使用於 lattice generate 之 Max Hyps 則設定為 4000。

以 on-the-fly replace 展開 PNLM 時，不再受限於較低的 intra-word 分數，即使在人名前後沒有稱謂詞也能夠抓取到人名的位置。另一方面，加入 PNLM 也可能將原本不是人名的詞錯誤地辨識成人名，這點在採用 uni-gram PNLM 時幾乎沒有發生，因為找到的人名個數原先就較少；而無論以兩階段式或者是 on-the-fly replace 演算進行辨識時，約有 30%辨識出的人名是錯誤的。

表 4.1: 正確標記出人名的辨識結果

正確答案	Off-line replace (uni-gram PN model)	On-the-fly replace (bi-gram PN model)
NCKU_f070303_0		
水源里長(OOV)	雖	隨員
陳枝福(OOV_PER)	遠離	里長
表示	章程	陳志福
	支付	表示
	表示	
NCKU_f070308_0		
演講	演講	演講
中	中國	中
郭振興(OOV_PER)	振興	郭振興
深入淺出	深入淺出	深入淺出

表 4.2: 錯誤標記出人名的辨識結果

NCKU_f090409_1

<s>	<s>	<s>
王	王達瑪	王達瑪
大媽(OOV)	的	的
的		
NCTU_f020402_0		
據瞭解	瞭解	據瞭解
澂清樓(OOV)	陳景樓	陳清老
於	與	與
先總統	先總統	先總統
NCKU_f100305_0		
讚不絕口	讚不絕口	讚不絕口
兩	將軍	江仁俊
人	認為	認為
均		
認為		

原先就屬 OOV 之詞很容易因為分數輸給 PNLM 而造成錯誤的辨識，而短詞相接由於詞的插入懲罰也會造成被搶詞的情況。如“澂清樓(OOV)”、“兩 人 均”等錯誤可望日後加入不同之 NER 模型或是 DM 模型來進行改善。同時我們可以發現辨識錯誤的人名中也有將外文人名錯誤地被中文 PNLM 抓取出，如：

表 4.3: PNLM 對 OOVs 外文人名的影響

NCTU_m030902_0		
不料	不料	不料
葛洛瑞索(OOV_外文)	河洛	柯洛偉

卻	萎縮	所
出奇不意	卻	卻
	出其不意	出其不意
NCTU_m030909_0		
打電話	打電話	打電話
督促	度	杜珠茂
馬佐維奇(OOV_外文)	出馬	朱偉啟
不要	作為	表示
使	其	

由於外文人名與中文人名在部分前後詞的關聯性享有共同的特性，而目前我們並無加入外文人名所用的 model 因此造成此情況。除外文人名外，實驗結果也有一些原先收錄在詞典內的人名又被 PNLM 的模型所辨識出的情形，表示該詞與前後詞的 n-gram 關係不足以支持而輸給了 PNLM 模型，舉例如下：

表 4.4: PNLM 對於 IV 人名搶詞之影響

NCKU_f100303_0		
郭朝武(IV_PER)	郭朝武	郭朝武
邱文明(IV_PER)	邱文明	邱文英
蘇徐瓊枝(OOV_PER)	所	所
和	需求	需求
	時	時
NCTU_f030801_0		
奧會	奧會	奧會
副秘書長	副秘書長	副秘書長
李慶華(IV_PER)	李慶華	李信華

昨天	昨天	昨天
召開	召開	召開

在實驗中訓練 root n-gram LM 時為了保留重要人名與前後的 n-gram 關係，並無將所有人名整合為一個類別訓練，而是以 IDF 法保留下常見的人名。日後可以將重要人名與其高關連性之稱謂詞結合成為一個人名片語(如：“陳水扁 總統”)，在片語中再計算該人名與稱謂詞的關聯性，如此可學習到片語非單獨人名與前後詞的關聯性，進而降低 somebody(IV person)被辨識為 nobody(OOV person)的可能性。

第三章的辨識結果中，採用 uni-gram PNLM 於一階段辨識時，大多仍須依靠稱謂詞支撐才能解碼出人名的位置；而以兩階段式做法進行辨識時，又受限於第一級 bi-gram root LM 的涵蓋率，需要採用更為精細的語言模型進行重計分才能取得較一階段辨識更好的辨認率。而使用同樣的 root LM 與 PNLM 模型之條件下，相較於在 off-line 採用取代演算法的辨識結果，on-the-fly replace 演算法能取得更好的 F-measure 分數與辨識率的提升。我們將前述之實驗的數據在此做一個整理：

表 4.5: Recognition results

Models	Word Accuracy
Tri-gram LM	73.36%
Tri-gram LM with uni-gram PN model	73.47%
Bi-gram LM	71.76%
Bi-gram LM with bi-gram PN model (Lattice generation)	71.82%
- Rescoring with tri-gram LM (A)	72.32%
- Rescoring with tri-gram LM (B)	76.27%
Tri-gram LM with bi-gram PN model (on-the-fly replace)	73.74%

表 4.6: Tri-gram LM with bi-gram PN model (on-the-fly replace)之 F-measure

All find	Golden hits	IV hits	Wrong hits	Precision	Recall	F-measure

76	48	7	21	63.16%	37.50%	47.06%

表 4.7: F-measure scores 比較

Models	F-measure
Tri-gram LM with uni-gram PN model	34.57%
Bi-gram LM with bi-gram PN model (Lattice generation)	44.11%
- Rescoring with tri-gram LM (A)	42.00%
- Rescoring with tri-gram LM (B)	42.10%
Tri-gram LM with bi-gram PN model (on-the-fly replace)	47.06%

第五章 結論與未來展望

本研究採用加權有限狀態轉換器進行實驗，可藉由不同的演算法整合語音模型，我主要為在語言模型層進行改良，更動辭典選詞方法並依照中文的特性提出若干改進方式；另一方面也注重處理 OOV words 相關的研究，建立出一套整合階層式語言模型的系統，能在進行辨識時再動態展開較複雜的模型以實現一階段式的辨認，也可經由兩階段式方式產生 WFST 格式之 lattice 來進行重計分，實驗結果指出加入階層式的語言模型確實可改善辨識效能，並能解碼出階層式的結構資訊。

從本研究可以延伸出各種不同的議題與應用方式：第一，除了本研究所採用之 OOVs 人名模型，未來可以加入不同的階層式模型如數量複合詞(DM)與其他 Named entity 資訊如地名、組織名等來提升系統的辨識效能。二，在進行兩階段式辨識產生 WFST 之 lattice graph 時，我們一併輸出了時間標記來取得音節間的切割資訊，可再利用這些資訊整合韻律模型，進而解決成詞邊界不準確而造成搶詞等錯誤的解碼結果。第三，優化後的加權有限狀態機圖形可以提供一個快速且可靠的辨識結果，可以依此建立一個即時辨認的系統。第四，目前使用於本研究的辨認音檔為朗讀式的語音，日後若考慮自發性語音的特性改良模型，相信可以將語音辨識更廣泛地應用在生活之中。

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】 Hiroaki, “Out-of-vocabulary word recognition with a hierarchical doubly markov language model,” Graduate School of Global Information and Telecommunication Studies, 2003

【11】 Lufeng Zhai, Pascale Fung, Richard Schwartz, Marine Carpuat, and Dekai Wu, “Using n-best lists for named entity recognition from chinese speech.” In HLT-NAACL

2004: Short Papers, pages 37--40, Boston, Massachusetts, USA. Association for Computational Linguistics, 2004

【12】 Hori, T., Hori, C., Minami, Y.: Fast on-the-fly composition for weighted finite-state transducers in 1.8 million-word vocabulary continuous speech recognition. In: Proc. Interspeech (ICSLP), vol. 1, pp. 289–292 (October 2004)

【13】 H. J. G. A. Dolfing, I. L. Hetherington, “Incremental language models for speech recognition using finite-state transducers,”Proc. of ASRU2001, 2001.

【14】 J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. ICML01, 2001.

【15】 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.

【16】 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.

【17】 Young, S., et al.: The HTK Book. Cambridge University Engineering Department, For HTK Version 3.2.1 (December 2002)

附錄一：實驗所用之 Variant Word 相關表格

Variant Word Pair		Variant Word Pair	
一石兩鳥	一石二鳥	按耐	按捺
一脈相承	一脈相傳	迫在眉梢	迫在眉睫
一蹴可及	一蹴可幾	僕僕風塵	風塵僕僕
入境問俗	入境隨俗	根柢	根基
力有不逮	力有未逮	桀傲不馴	桀驁不馴
萬馬千軍	千軍萬馬	成竹在胸	胸有成竹
千錘百鍛	千錘百鍊	掀然大波	軒然大波
鼎鼎大名	大名鼎鼎	除此以外	除此之外
不亢不卑	不卑不亢	洋洋得意	得意洋洋
天淵之別	天壤之別	從頭至尾	從頭到尾
引以自豪	引以為豪	源遠流長	淵遠流長
心腹之患	心腹大患	趾高氣揚	趾高氣昂
指手劃腳	比手畫腳	普天下	普天之下
譬如說	比如說	青天霹靂	晴天霹靂
無人問津	乏人問津	無怨無尤	無怨無悔
生氣勃勃	生氣蓬勃	猶疑不決	猶豫不決
努力以赴	全力以赴	短小精幹	短小精悍
和衷共濟	同舟共濟	醜態畢露	窘態畢露
聞名遐邇	名聞遐邇	委靡	萎靡
勢所難免	在所難免	開開玩笑	開玩笑
如夢如幻	如夢似幻	坍陷	塌陷
如醉如癡	如癡如醉	意興風發	意氣風發

米開蘭基羅	米開朗基羅	鄭重其事	慎重其事
羊腸小道	羊腸小徑	煞有介事	煞有其事
一新耳目	耳目一新	縛手縛腳	綁手綁腳
自嘆不如	自嘆弗如	杯觥交錯	觥籌交錯
兵分二路	兵分兩路	粥少僧多	僧多粥少
峻工	完工	春風滿面	滿面春風
言之成理	言之有理	精疲力盡	精疲力竭
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
巴金森氏	帕金森氏	發奮圖強	奮發圖強
巴金森氏病	帕金森氏病	隨機應變	臨機應變
巴金森氏症	帕金森氏症	餐風露宿	餐風宿露
戴月披星	披星戴月	聳人聽聞	駭人聽聞
明察暗訪	明查暗訪	盪氣選腸	盪氣迴腸
細微末節	枝微末節	節衣縮食	縮衣節食
送舊迎新	迎新送舊	心驚膽戰	膽顫心驚
雨過天晴	雨過天青	心驚膽顫	膽顫心驚
非同尋常	非比尋常	膽戰心驚	膽顫心驚
前所未聞	前所未見	豐功偉績	豐功偉業
唾手可得	垂手可得	鍛羽而歸	鎩羽而歸
觸目所見	觸目所及	耶誕卡	聖誕卡
耶誕	聖誕	耶誕夜	聖誕夜
耶誕節	聖誕節	耶誕紅	聖誕紅

耶誕樹	聖誕樹	耶誕老人	聖誕老人
耶誕老公公	聖誕老公公		

Variant Word Pair				
難分軒輊		無分軒輊		不分軒輊
百折不撓		堅毅不撓		不屈不撓
厥功至偉		厥功其偉		厥功甚偉
萬不得已		迫不得已		逼不得已
同聚一堂		共聚一堂		齊聚一堂
禮拜一		星期一		週一
禮拜二		星期二		週二
禮拜三		星期三		週三
禮拜四		星期四		週四
禮拜五		星期五		週五
禮拜六		星期六		週六
禮拜天	禮拜日	星期天	星期日	週日