國立臺灣師範大學
資訊工程研究所碩士論文

指導教授：陳柏琳博士

使用鑑別式語言模型於語音辨識結果重新排序

Applying Discriminative Language Models to Reranking of *M*-best Speech Recognition Results

研究生：劉 鳳 萍　撰

摘要

語言模型代表語言的規律性，在語音辨識中，它可用以減輕聲學特徵混淆所造成的問題，引導辨識器在多個候選字串中作搜尋，並量化辨識器產生的最終辨識結果字串的可接受度高低。然而，隨著時空及領域的不同，語言產生差異，固定不變的語言模型無法符合實際需求。語言模型調適提供了一個解決之道，使用少量同時期或同領域的調適語料對語言模型進行調整，以增進效能。鑑別式語言模型為語言模型調適方法之一，它首先取得一些特徵(Feature)，每一個特徵各有其對應之權重(Feature Weight)，以代表語言中的句子或字串，並以這些特徵及其相關權重為基礎，構建出一套評分機制，用以對基礎辨識器(Baseline Recognizer)所產生的多個辨識結果進行重新排序(Reranking)，以期最正確的詞序列可以成為最終辨識結果。本文提出以關鍵詞自動擷取方法所得結果，增加鑑別式語言模型之特徵。關鍵詞自動擷取方法是透過計算字或詞在語料庫中同時重複出現的次數以擷取出關鍵詞，其優點為可以在不依賴詞典(Lexicon)的情況下，擷取出新生詞彙或不存在詞典裡的語彙，這樣的特性也許會對鑑別式訓練有所助益，但實驗結果顯示未有顯著之改善效果。

Abstract

A language model (LM) is designed to represent the regularity of a given language. When applied to speech recognition, it can be used to constrain the acoustic analysis, guide the search through multiple candidate word strings, and quantify the acceptability of the final word string output from a recognizer. However, the regularity of a language would change along with time and cross domains, such that a static or invariable language model cannot meet the realistic demand. Language model adaptation seems to provide a solution, by using a small amount of contemporaneous or in-domain data to adapt the original language model, for better performance. The discriminative model is one of the representative approaches for language model adaptation in speech recognition. It first derives a set of indicative features, where each feature has a different weight, to characterize sentences or word strings in a language, and then build a sentence scoring mechanism on the basis of these features and the associated weights. This mechanism is used to re-rank the *M*-best recognition results such that the most correct candidate word string is expected to be on the top of the rank. This paper proposes an approach which takes the results of a fast keyword extraction method as additional features for the discriminative model. This method extracts keywords by counting the repetition of co-occurrences of characters or words in the speech corpus, such that these keywords may capture the regularity of language being used. A nice property is that it extracts keyword without the need of a lexicon, so it can extract new keywords and the keywords which do not exist in, or contain words of the lexicon. This property may be useful for discriminative language modeling, but, however, empirical experiments show it only provides insignificant improvements.

誌謝

感謝父母與弟弟陪伴我從小到大一路走來，始終關懷有加。感謝我的先生安立，你的支持與鼓勵，使我能安心致力於學業，同時品味生活的豐富喜樂。

感謝指導教授陳柏琳博士在碩士班求學過程中的諄諄教誨，老師做研究的認真態度與嚴謹的治學方法，是學生們應努力學習的榜樣。

感謝口試委員古鴻炎博士、王新民博士與洪志偉博士對學生論文的指正，使得這本論文能更趨完全。

感謝炫盛學長、鴻欣學長及冠宇在學業上的幫助，向你們請教或討論問題的過程中，讓我學習到如何從不同的角度去思索與分析問題。

感謝永典、韋豪、鈺玫、家妏為實驗室帶來的溫暖與歡笑，使得小小的實驗室裡充滿了活力與希望。期待著新進學弟妹敏軒、紋儀與珮寧的加入，為實驗室開展新的氣象。

目錄

圖目錄

表目錄

第一章 緒論

1.1 研究背景

語音是人類最自然、歷史也最悠久的溝通媒介之一。

從古至今，人類日常生活中多半倚賴「聽說讀寫」作為溝通方式。隨著電子設備的發展，「讀」與「寫」的部分，在紙筆書寫之外，又多了螢幕、鍵盤與滑鼠等輸出入設備作為選項；至於「聽」與「說」的部分，則有錄音與播放設備可以保存與輸出聲音訊號。

但是「聽、說」與「讀、寫」兩類型媒體之間的訊息交換，需要人類作為中介；人類與電子設備之間的溝通，仍然大量依賴使用者與機器之間透過鍵盤滑鼠的直接接觸，方能進行。

近年來，語音處理技術的發展，使得「聽、說」與「讀、寫」兩個系統之間有了轉換的機會，例如自動語音辨識(Automatic Speech Recognition, ASR)可以代替人類將「聽」到的語音轉而「寫」為文字紀錄；而語音合成(Speech Synthesis)則可將「讀」到的文字紀錄轉換成聲音訊號，代替人類「說」話。

也正因如此，語音成為人機互動的媒介之一。機器可以透過自動語音辨識系統取得人類欲輸入之指令，再透過語音合成以回應人類。這樣的功能，在以往也許只是電影中令人炫目的高科技象徵，然而在今日，已透過科技的發展逐漸落實，成為生活中的一部分。例如目前在手機上已有語音撥號的功能，可以直接要求手機撥號給特定連絡人，不需手動按鍵撥號。這樣的功能，未來有機會可以運用到更多領域，為人類生活帶來便利。

1.2 語音辨識

人類在聽取一段語音之時，由耳朵接收了能量，傳遞至大腦，由大腦接收訊號，得知「聽」到了什麼樣的聲音，再根據此人對於一個語言長時間受到的訓練所得到的瞭解，可判斷出這段語音的內容。

在設計語音辨識系統時，人類試圖將造物者運用在人類身上的設計，利用電子設備加以重現。人類腦部對一段聲音訊號的接收，是由特徵擷取(Feature Extraction)這個模組所負責處理。它將聲音訊號轉化為可量測之數據，並擷取對語音辨識有意義的資訊，以供後續其他模組對此段聲音訊號作出相應之判斷。

至於人類對特定語言所受到的訓練與理解，則由聲學模型(Acoustic Model)與語言模型(Language Model)作為代表。人類在學習語言的過程中，首先學習的就是聲音與語意之間的關係，以及聲音與用來記錄語意的文字之間的關係。在語音辨識系統中，這個部分是由聲學模型來負責。聲學模型是由聲音語料訓練而成，其訓練目的是為了讓辨識系統記住聲音與文字之間的對應關係。

在習得聲音與文字之間的對應關係後，人類進一步學習的，就是語言中的規則性(Regularity)。語言的規則性是語言長期發展下約定俗成的結果，保留人們普遍可以接受的語言使用習慣，其規則性有助於人類用來敘述見聞、表達看法，並因此而可以進行人與人之間的相互瞭解與溝通。在語音辨識系統中，這個部分是由語言模型來負責。語言模型是由文字語料訓練而成，其訓練目的是為了擷取語言中的規則性，使得語言模型可在測試階段引導辨識器選擇正確詞序列作為辨識結果。

人類在接收一段語音後，判斷自己聽到的聲音內容，並將它轉換成相對應的文字。這段過程被重現於自動語音辨識系統中。如圖 1-1 所示，自動語音辨識系統包括特徵擷取、聲學模型、語言模型與語言解碼等四個主要部分，我們使用

文字語料訓練出語言模型，代表語言使用的規律，並使用語音語料訓練出聲學模型，代表示語音與文字之間的對應關係。在給定一段語音訊號的情況下，透過特徵擷取，可以從一段語音訊號中取得其特徵向量，再根據聲音語料訓練而成的聲學模型，以及文字語料訓練而成的語言模型，對特徵向量作語言解碼(Decode)，最後產生辨識結果。

在語言解碼階段，判斷並決定辨識結果的過程是根據以下公式：

$$\begin{aligned} & W^* \\ & = \arg\max_W P(W \mid X) \\ & = \arg\max_W \frac{P(W)P(X \mid W)}{P(X)} \\ & \approx \arg\max_w P(W)P(X \mid W) \end{aligned} \tag{1.2.1}$$

X 表示一段語音，而 W 表示詞序列，即一段文字。$W^* = \arg\max_W P(W \mid X)$ 代表語音辨識的功能在於給定一段語音訊號的情況下，求出最有可能之對應詞序列。

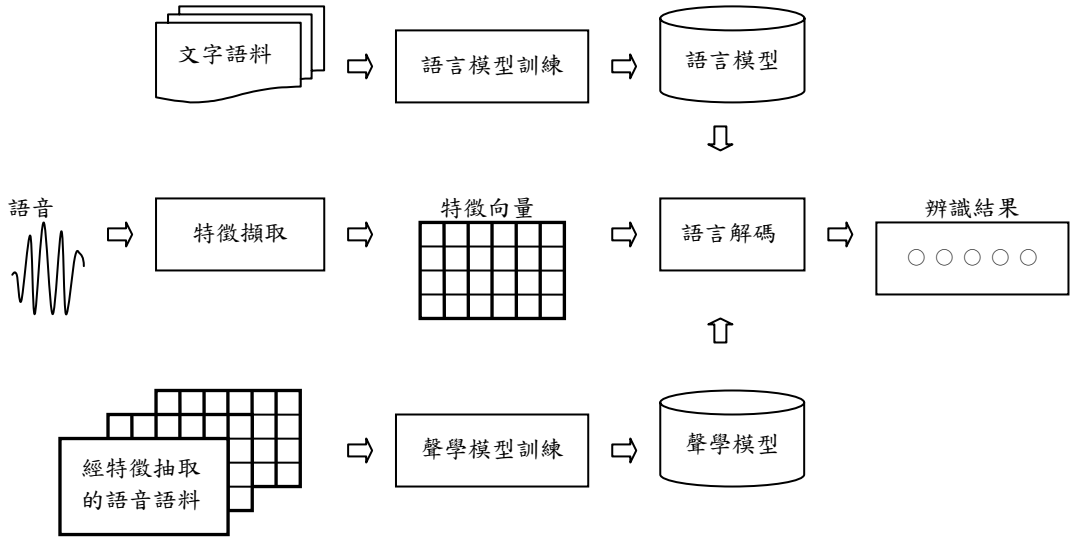


圖 1-1 自動語音辨識流程圖

由於 $P(W \mid X)$ 無法直接估算，因此先用貝式定理(Bayes' Theorem)展開為 $\frac{P(W)P(X \mid W)}{P(X)}$，接著由於對所有候選詞序列$W$來說，其分母項皆相同，故可省略，僅需根據 $P(W)P(X \mid W)$，便能找出能使 $P(W \mid X)$ 之值最大的詞序列W^*，作為辨識結果。

$P(X \mid W)$ 為聲學模型機率，其意義為在一個訓練好的辨識系統中，某一個詞序列W對應到某一段語音 X 的機率。$P(W)$ 則為語言模型機率，代表的是在一個訓練好的辨識系統中，某一個詞序列W產生的機率。

以下將分別概述特徵擷取、聲學模型、語言模型與語言解碼等部分。

(一) 特徵擷取

特徵擷取的主要目的，是從一段語音訊號中取出其特徵，量化為一組數據——例如特徵向量(Feature Vector) ——以作為參數，供語音辨識系統對此段語音訊號作出估測與判斷。在取得特徵之後，便需考慮環境或噪音對語音特徵的影響，對特徵向量作進一步修正，以增加語音的強健性(Speech Robustness)。

(二) 聲學模型

一個中文音節(Syllable)由兩個次音節(Sub-syllable)組成，分別為聲母(Consonant)與韻母(Vowel)。聲母與韻母各自對應一個聲學模型。聲母的聲學模型稱為 INITIAL，韻母的聲學模型則稱為 FINAL。

目前的聲學模型是透過馬可夫模型(Markov Model)來表示，由於語音具有時序性，因此是利用由左至右的隱藏式馬可夫模型(Left-to-right Hidden Markov Model)來模擬語音的產生。

隱藏式馬可夫模型是用來模擬特定環境下，某個事件發生的機率。在一個

隱藏式馬可夫模型中事先預設了數個狀態(State)，再根據訓練資料作計算，以設定此隱藏式馬可夫模型中個狀態之初始機率(Initial Probability)、狀態轉移機率(State Transition Probability)，以及各狀態產生各事件之機率(Observation Probability)。由於語音具有時序性，因此採用由左至右的隱藏式馬可夫模型，先預設數個狀態，再根據經過特徵擷取的語音語料作訓練，以估算出此模型中之各項機率，此後便根據這些機率，來判斷在此語音環境下，某段語音發生的機率 $P(X \mid W)$ 。

(三) 語言模型

目前的語言模型是統計式的基於歷史資訊的模型(History-based Model)。基於歷史資訊的模型的設計理念是根據經驗法則，統計先前已出現的一連串事件與下一個出現的事件之間的關係。

在訓練階段，先統計每一個詞 w_i 與其歷史詞序列 $w_1 w_2 \cdots w_{i-1}$ 之間的關係，將它視為語言的規律性，若以機率作為準則，即為 $P(w_i) = P(w_i \mid w_1 w_2 w_3 \cdots w_{i-1})$ 。統計之目的是為了在測試階段，可以根據一段已辨識出的詞序列，去推算下一個最有機會出現的詞為何。

例如在訓練語料中，就歷史詞序列 h_i 而言，既有 h_i 與詞 w_x 組成的詞序列 $h_i + w_x$ ，亦有 h_i 與詞 w_y 組成的詞序列 $h_i + w_y$ ，則在訓練階段，需根據 $h_i + w_x$ 與 $h_i + w_y$ 在訓練語料中出現的次數分別計算其機率。當測試階段出現詞序列 h_i 時，欲判斷下一個詞應為 w_x 或者是 w_y ，除了依據聲學模型所提供的資訊外，還要根據語言模型中 $h_i + w_x$ 與 $h_i + w_y$ 的機率作判斷，選擇出機率較大者，即可判斷出下一個出現的詞應為 w_x 或者是 w_y 。

(四) 語言解碼

對一段語音作語言解碼，是根據特徵擷取時所取得此段語音之特徵，建立這段語音所對應的多條可能詞序列，並根據事先訓練好的聲學模型與語音模型，賦予每一條候選詞序列W對應之機率，從中選取可能性最大之詞序列W^*作為辨識結果。

1.3 語言模型調適簡介

語言的規律性是人類長久使用下所保留的特定習慣，然而，因人事時地物之差異，語言亦隨之產生變化。

《語法與修辭》一書中提到：「語言是隨著時代的變化而發展的，在語言的發展過程中，語音、詞彙的發展變化很快，語法發展變化比較緩慢。」就語法來看，古代漢語中存在「主語—賓語—述語」結構，例如「卿欲何言？」這是現代漢語所沒有的。現代漢語中使用的是「主語—述語—賓語」結構，例如「你想說什麼？」雖然變化緩慢，但變動性仍是確實存在的。

近年來，由於電視、廣播與網路等傳播媒介的迅速發展，人們能夠發抒己見的管道日益增多，訊息傳遞也越廣越快，使得語言的使用習慣有更多機會產生變化。例如「宅」這個詞彙，原本是指的是「房舍」，近年來卻被用於形容一個人喜歡經常待在家中，如「宅男」；又如「粉絲」一詞，原本指的是一種豆粉製成的食品，如今時常被用來代表熱衷於特定人事物的追隨者(Fans)。

諸如此類新興的語言使用習慣，也許並不存在於原本用來訓練語言模型的文字語料之中。若語音辨識的應用對象是含有這些新興語言使用習慣的測試語料，就很可能因為訓練語料與測試語料的不匹配(Mismatch)，影響語言模型協助辨識系統選擇出正確辨識結果的能力，降低語音辨識系統的效能。

為解決語言模型與測試對象不匹配的問題，需要對語言模型進行調適(Adaptation)，使其適用於特定測試對象。語言模型是由文字語料訓練而得，為使其適用於特定對象，需另以一份調適語料對語言模型進行調適。

最大事後機率法(Maximum a Posteriori, MAP)是最常被用來調整N連語言模型參數的語言模型調適方法，它根據調適語料，對語言模型中 N 連詞的機率值進行調整，使其更符合測試對象之需要。

近年來，以最小化辨識錯誤率為目標的鑑別式訓練(Discriminative Training)方法被應用於語言模型調適，成為語言模型調適方法之一。此方法訓練分類器(Classifier)能夠區分最接近正確辨識結果的詞序列與其他候選詞序列，使得最接近正確辨識結果的詞序列能順利成為辨識結果，降低辨識錯誤率。

1.4 研究內容與貢獻

將透過鑑別式語言模型訓練方法進行的語言模型調適應用於中文大詞彙語音辨識，進行辨識結果的重新排序，並作一系列實驗與討論。

此外，對鑑別式語言模型中特徵(Feature)的使用作改進，以關鍵詞自動擷取方法所得詞彙，增加鑑別式訓練之特徵，以期能擷取語言規則特性與語意資訊，並取得詞典中未列舉之詞彙。

1.5 論文章節安排

本文後續幾個章節分述如下：

第二章 介紹語言模型以及語言模型調適的內容和方法。

第三章 介紹鑑別式語言模型的數種訓練方法，例如 Boosting 演算法、Perceptron 演算法與 Minimum Sample Risk 方法等。

第四章 描述筆者使用之改良方法，透過關鍵詞自動擷取方法取得關鍵詞以增加鑑別式訓練之特徵，期望能夠進一步掌握語料中的語言規律以及詞典中未列舉之詞彙。

第五章 說明實驗語料與實驗結果。

第六章 總結本論文研究內容。

第二章 文獻回顧

2.1 語言模型

2.1.1 統計式語言模型

語言模型在自動語音辨識系統中的主要的作用，是代表語言使用習慣與規律。當前使用的語言模型為統計式語言模型(Statistical Language Model, SLM)，它統計一個詞在訓練語料中的出現情形，給予該詞一個機率，以代表該詞於某種語言使用環境下的重要性。至於語言模型在辨識的過程中對辨識器產生的引導作用，則是在給定一個歷史詞序列的情況下，判斷下一個最有可能出現的詞為何。

此設計理念與 Claude Elwood Shannon 在資訊理論(Information Theory)研究中所提出的問題有關：「給定一個字母序列，下一個字母最有可能會是什麼？」在這個觀念之下，統計式語言模型成為一個基於歷史資訊之模型(History-based Model)：每一個詞 w_i 的出現，都與其歷史詞序列有關，亦即 $P(w_i \mid w_1 w_2 \cdots w_{i-1})$。因此辨識的過程，即是在給定一個歷史詞序列的條件下，判斷下一個最有可能出現的詞為何，藉由從訓練語料中計算出給定歷史詞序列的條件下所有詞的機率分布，並取其機率最高者，即可選出下一個最有可能出現的詞。

一段詞序列 $W = w_1 w_2 \cdots w_k$ 在訓練語料中的事前機率 $P(W)$，可使用連鎖律(Chain Rule)分解成：

$$P(W) = P(w_1)P(w_2 \mid w_1)\cdots P(w_k \mid w_1, w_2, \cdots, w_{k-1}) = \prod_{i=1}^{k} P(w_i \mid h_i) \quad (2.1.1.1)$$

當詞序列長度很大時，難以直接估算其條件機率 $P(w_i \mid w_1, w_2, \cdots, w_{i-1})$，且考慮各種詞的排列組合有其困難性，因此目前語言模型多半採取 N 連語言模型，利用

$N-1$ 階馬可夫假設(N-1 Order Markov Assumption)，主張每一個詞的出現，只與前 $N-1$ 個詞相關：

$$P(w_i \mid h_i) \approx P(w_i \mid w_{i-N+1}, \cdots, w_{i-2,} w_{i-1}) \tag{2.1.1.2}$$

也就是說，一個詞序列 W 的機率可寫為：

$$P(W) = \prod_{i=1}^{n} P(w_i \mid h_i) = \prod_{i=1}^{n} P(w_i \mid w_{i-N+1}, \cdots, w_{i-1}) \tag{2.1.1.3}$$

這樣的模型稱為 N 連語言模型。目前常用的 N 連模型有單連(Unigram)、雙連(Bigram)與三連詞(Trigram)模型。例如常見的三連語言模型，可寫為：

$$P(W) = P(w_1)P(w_2 \mid w_1)\prod_{i=3}^{n} P(w_i \mid w_{i-2}, w_{i-1}) \tag{2.1.1.4}$$

N 連詞模型的訓練是透過最大相似度估測(Maximum-likelihood Estimation)來作估測，以 $P(w_i \mid w_{i-2}, w_{i-1})$ 為例，其估測值為：

$$P(w_i \mid w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})} \tag{2.1.1.5}$$

$C(x)$ 表示詞序列 x 在訓練語料中的出現次數，即詞頻(Word Count)。

如前所述，統計式語言模型是一種基於歷史資訊之模型，它根據一段歷史詞序列來決定下一個詞的機率，正因如此，當訓練語料中的資料不足，也就是資料稀疏(Data Sparseness)的情形出現時，在測試階段就無法提供必要的歷史詞序列資訊以判斷下一個詞出現的機率，造成辨識結果和實際語音內容有所誤差。

為解決此問題，前人就各個角度發展方法來因應資料稀疏情形之對策。首先，就模型面而言，發展了類別語言模型(Class-based Language Model) [Brown *et al.* 1990]，將語意(Semantic)或語法(Grammatical Behavior)相近的詞歸為一類，以歷史類別序列代替歷史詞序列作為判斷一個詞出現機率的根據。

其次，就資料面而言，快取(Caching)模型和語言模型調適(Language Model

Adaptation)都利用額外的資料以補訓練語料之不足。

快取語言模型[Kuhn and Mori 1990]本身也是個基於歷史資訊之模型，只是它除了根據訓練語料建立一個基於歷史資訊之模型外，另外在測試階段動態地根據當下已判讀所得之測試語料歷史，對原有的模型做插補(Interpolate)。快取語言模型的設計理念是：如果你說了什麼，你可能很快又會再說一次。因此，它在判讀一部分測試語料後，立即計算這部分測試語料的 N 連詞出現次數，建立一個動態的 N 連模型。

語言模型調適則是在原有的訓練語料之外，另加入一份資料量較小的調適語料作為訓練之用。這份調適語料可能時空與領域皆與原有訓練語料相近，只是用來增加訓練語料的份量以避免資料稀疏，也可能與原有訓練語料之時空或領域並不相近，而是希望能使訓練結果偏向某一特定時空或領域，卻不具備該時空或領域之充足資料，因此改為在原有訓練語料中加入一部分特定時空或領域的語料，以改變語言模型的傾向。

另外，各種平滑化(Smoothing)技術也用來因應資料稀疏問題，如 Good-Turing Estimate [Goodman 2001]、Katz Back-off Smooting [Katz 1987]與 Kneser-Ney Back-off Smoothing [Kneser *et al.* 1995] 等技術，將訓練語料中存在之詞機率分出一部分給其他未曾出現(Unseen)在訓練語料中的詞。這是一種捐獻的觀念，由於詞機率和必須為 1，若要使某些詞序列的機率不再是 0，也就是欲增加它們的機率，則勢必要減少其他詞序列之機率，因此，便從原本機率不為 0 之詞序列處收集一些機率，將它分給原本機率為 0 的詞序列。例如後向(Back-off)方法中，就三連語言模型而言，訓練完成之語言模型中包含了多種單連、雙連及三連詞機率，在測試階段針對特定三連詞而言，若語言模型中並無此三連詞之資訊，則改以雙連詞與單連詞的組合，來提供此三連詞之機率，使此三連詞之機率不至為 0。

2.1.2 語言模型評估

為了評估(Evaluate)語言模型是否能在辨識過程中順利引導辨識器，使其選擇最接近正確參照轉寫的候選詞序列作為辨識結果，以及衡量語言模型實際運用效益多寡，需進行語言模型評估。語言模型評估的主要標準有錯誤率與複雜度二者，錯誤率評估的是辨識結果，語言複雜度則是單就語言模型進行評量。

(一) 錯誤率(Error Rate)

錯誤率是將一段語音之正確參照轉寫與語音辨識結果作字串比對所得到之數據。依比對之單位(Unit)不同，可分為字錯誤率(Character Error Rate, CER)與詞錯誤率(Word Error Rate, WER)。

字串比對可透過動態規劃(Dynamic Programming)方法進行。若以字為對齊(Align)單位來看，若正確字串中的某個字在辨識結果中被取代為錯誤的另一字，稱為替代(Substitution)；若正確字串中存在的某一字並不存在於辨識結果中，而是被移除了，則稱為刪除(Deletion)；與刪除相反，若辨識結果中多出了並不存在於正確字串中的字，則稱為插入(Insertion)。

錯誤率之計算方式如下：

$$Error\ Rate = \frac{S + D + I}{N} * 100\% \qquad (2.1.2.1)$$

其中 S 代表替代數，D 代表刪除數，I 代表插入數，而 N 則為正確參照轉寫的字串長度。

(二) 語言複雜度(Perplexity)

語言複雜度是用來評估一個語言模型的方式，其幾何意義為語言模型測試語料正確語句 $w_1, w_2, \cdots, w_n$ 之機率倒數的幾何平均數：

$$PP(w_1, w_2, \cdots, w_m) = \sqrt[m]{\frac{1}{P(w_1)} * \prod_{i=2}^{m} \frac{1}{P(w_i \mid w_1, w_2, \cdots, w_{i-1})}} \quad (2.1.2.2)$$

例如就三連詞而言則為：

$$PP(w_1, w_2, \cdots, w_n) = \sqrt[m]{\frac{1}{P(w_1) \cdot P(w_1 \mid w_2) \cdot \prod_{i=3}^{m} P(w_i \mid w_{i-2}, w_{i-1})}} \quad (2.1.2.3)$$

n 為測試語料之總詞數。語言複雜度也可視為語言模型中詞的平均分支係數 (Geometric Mean of the Branching Factor)。

辨識錯誤率的計算是針對辨識結果作出衡量，由於辨識結果是由辨識系統產生，包括特徵擷取、聲學模型與語言模型等模組的共同合作結果，因此辨識錯誤率適用於上述各模組的評估。至於語言複雜度則不然，它評估的對象是語言模型的內容本身，考量的是在出現一段歷史詞序列的情況下，下一個出現的字或詞的選擇共有多少種可能。複雜度愈高，表示給定一個歷史詞序列，我們所要考慮的下一個字或詞選擇愈多。

2.2 語言模型調適

2.2.1 語言模型調適的意義與架構

人類的語言並非恆常不變，它會因時間、領域的差異而有所不同，且隨著文化的發展而不斷地有新語彙產生。為因應不同時間或領域中，人們對語法、語意(Semantic)及詞彙(Lexicon)的使用習慣差異，避免傳統統計式 *N* 連語言模型內容與應用領域之間有不匹配(Mismatch)的情形發生，所以需要進行語言模型調適(Language Model Adaptation) ，使得統計式 *N* 連語言模型內容能夠符合特定應用

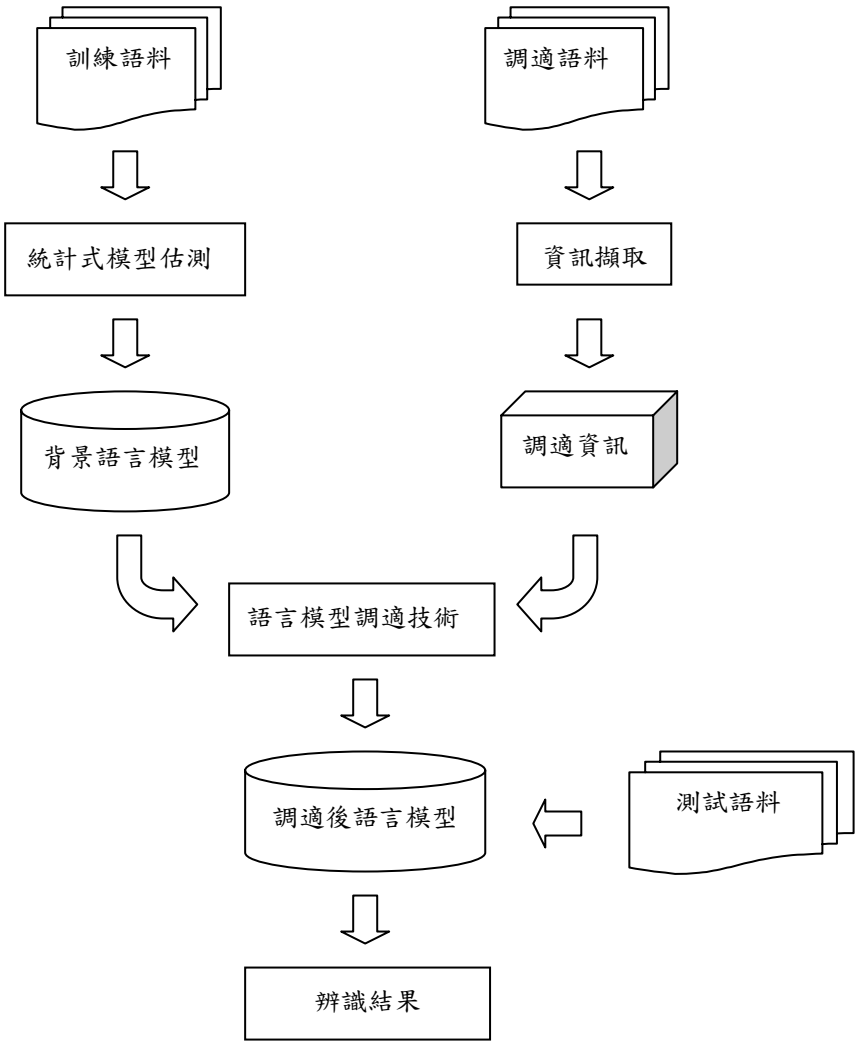


圖 2-1 語言模型調適架構

領域之語音辨識需求。語言模型調適的主要架構如圖 2-1 所示：從圖中可見語言模型調適架構中有訓練語料、調適語料與測試語料三種語料，其中訓練語料與調適語料為我們在發展語言模型、估算並調整機率分佈的過程中所使用到的語料，而測試語料則為我們欲應用語音辨識技術之目標對象語料。

訓練語料與調適語料之比較如下表所示：

比較項目 \ 語料類型	訓練語料	調適語料
資料量	較多	較少
與測試語料關係	較不密切	較密切

表 2-1 訓練語料與調適語料之比較

訓練語料用來統計並估算機率分佈，所產生的語言模型稱為背景語言模型(Background Language Model)；至於調適語料則用來調整背景語言模型中的機率分佈，使其有機會更接近於我們欲應用語音辨識技術之目標對象的語言環境，其所產生的語言模型為調適後的語言模型(Adapted Language Model)。

根據語料特性不同，語言模型調適可分為三種類型：同領域語言模型調適、跨領域語言模型調適與同時期語言模型調適。如表 2-2 所示，其語料選擇雖然不同，但其目的都是希望能夠得到更好的辨識結果。若調適目標為調整語言模型之「領域」，則調適語料的領域必與測試語料領域一致，即使訓練語料領域與測試語料領域未必相同；若調適目標為語言模型之「時期」，則調適語料之時期必與測試語料之時期一致，即使訓練語料時期與測試語料時期未必相同。表 2-1 與表 2-2 皆反映出調適語料與測試語料之間關係密切，這是因為調適語料的存在，本就是為了調整原有語言模型，使其更能代表某種特定的語言使用環境，有效引導辨識系統選擇最正確的詞序列作為辨識結果。

比較項目 語言模型調適類型	時期	領域
同領域語言模型調適 (Within-Domain LM Adaptation)	訓練語料與測試語料時期不同，而調適語料與測試語料時期相同。	三種語料來源皆相同。
跨領域語言模型調適 (Cross-Domain LM Adaptation)	三種語料時期皆相同。	訓練語料與測試語料來源不同，而調適語料與測試語料來源相同。
同時期語言模型調適 (Contemporaneous LM Adaptation)	調適語料與測試語料時期相同。	調適語料與測試語料來源未必相同。

表 2-2 訓練語料、調適語料與測試語料三者關係

2.2.2 語言模型調適方法

語言模型調適方法試圖運用各種語言相關資訊調整語言模型，使其更符合於特定語言使用環境之需求。例如最大事後機率(Maximum a Posteriori, MAP)法[Bacchiani *et al.* 2003] 根據調適語料中各詞彙之詞頻以調整語言模型機率，詞主題混合模型(Word Topical Mixture Model, WTMM) [Chiu and Chen 2007]透過隱藏的詞主題預測另一個詞的機率，位置相關語言模型(Position-Dependent Language Models) [邱炫盛 2007] 將詞彙在文件中的位置資訊加入原有的模型中，而鑑別式語言模型(Discriminative Language Model)則是引進全域線性模型(Global Linear Model)架構以對辨識結果進行重新排序。

最大事後機率法是最常用於語言模型調適的方法之一，它根據調適語料調整 N 連語言模型中 N 連詞的機率，以進行語言模型調適。最大事後機率法之目標在

於給定調適語料 Π 的情形下，求得最符合調適語料內容的語言模型參數估測 θ^*，以最大化 Π 之事後機率：

$$\theta^* = \arg\max_{\theta} P(\theta \mid \Pi) = \arg\max_{\theta} P(\theta)P(\Pi \mid \theta) \tag{2.2.2.1}$$

亦即其目標在於找出最符合調適語料中語言使用習慣之語言模型。$P(\Pi \mid \theta)$ 代表 N 連語言模型，$P(\theta)$ 代表此模型的事前機率。

$P(\theta)$ 的參數為各種詞序列組合：$\phi_{h_1,w_1},\cdots,\phi_{h_1,w_{|V|}},\cdots,\phi_{h_K,w_1},\cdots,\phi_{h_K,w_{|V|}}$，其中 N 連模型裡共有 K 種歷史詞序列，而 h_i 代表其中某一個歷史詞序列。w_j 代表某個詞，$|V|$ 代表詞典大小，則 $\sum_{j=1}^{|V|}\phi_{h_i,w_j}=1$。令 $P(\theta)$ 為一 Dirichlet 分布(Dirichlet Distribution)，則：

$$\begin{aligned} &P(\theta) \\ &= P(\phi_{h_1,w_1},\cdots,\phi_{h_K,w_{|V|}} \mid v_{h_1,w_1},\cdots,v_{h_K,w_{|V|}}) \\ &\propto \prod_{i=1}^{K}\prod_{j=1}^{|V|}\phi_{h_i,w_j}^{v_{h_i,w_j}-1} \end{aligned} \tag{2.2.2.2}$$

$v_{h_i,w_j} > 0$，為 Dirichlet 分布參數。至於 N 連語言模型 $P(\Pi \mid \theta)$ 則為一個多項式分布（Multinomial Distribution）：

$$\begin{aligned} &P(\Pi \mid \theta) \\ &= P(w_1,\cdots,w_{|V|} \mid \phi_{h_1,w_1},\cdots,\phi_{h_1,w_{|V|}},\cdots,\phi_{h_K,w_1},\cdots,\phi_{h_K,w_{|V|}}) \\ &\propto \prod_{i=1}^{K}\prod_{j=1}^{|V|}\phi_{h_i,w_j}^{c_{h_i,w_j}} \end{aligned} \tag{2.2.2.3}$$

c_{h_i,w_j} 為詞 w_j 追隨在歷史詞序列 h_i 之後出現的次數。將方程式 2.2.2.2 與 2.2.2.3 代入方程式 2.2.2.1 可得：

$$
\begin{aligned}
&P(\theta \mid \Pi) \\
&= P(\theta)P(\Pi \mid \theta) \\
&= \prod_{i=1}^{K}\prod_{j=1}^{|V|}\phi_{h_i,w_j}^{v_{h_i,w_j}-1+c_{h_i,w_j}}
\end{aligned}
\tag{2.2.2.4}
$$

上式經由對ϕ_{h_i,w_j}作偏微分求極值並加以推算後，可得最大事後機率法的解：

$$
\phi_{h_i,w_j} = \frac{v_{h_i,w_j}-1+c_{h_i,w_j}}{\sum_{p=1}^{|V|}(v_{h_i,w_p}-1+c_{h_i,w_p})} \tag{2.2.2.5}
$$

若賦予v_{h_i,w_j}不同的值，則會得到不一樣的結果，例如最大事後機率法中的詞頻數合併法(Count Merging)與模型插補法(Model Interpolation)這兩種方法，其差別即在於它們賦予v_{h_i,w_j}之值相異。

若令$v_{h_i,w_j} = C_B(h_i)\frac{\alpha}{\beta}P_B(w_j \mid h_i)+1$，可得：

$$
\begin{aligned}
&P(w_j \mid h_i) \\
&= \frac{C_B(h_i)\frac{\alpha}{\beta}P_B(w_j \mid h_i)+C_A(h_i w_j)}{\sum_{p=1}^{|V|}C_B(h_i)\frac{\alpha}{\beta}P_B(w_p \mid h_i)+C_A(h_i w_p)} \\
&= \frac{\alpha C_B(h_i w_j)+\beta C_A(h_i w_j)}{\sum_{p=1}^{|V|}\alpha C_B(h_i w_p)+\sum_{p=1}^{|V|}\beta C_A(h_i w_p)} \\
&= \frac{\alpha C_B(h_i w_j)+\beta C_A(h_i w_j)}{\alpha C_B(h_i)+\beta C_A(h_i)}
\end{aligned}
\tag{2.2.2.6}
$$

此為詞頻數合併法，其中$C_B(.)$代表背景語言模型中某一個詞的詞頻(Word Count)，$C_A(.)$代表調適語言模型中某一個詞的詞頻，而$\frac{\alpha}{\beta}$則為一個經由最大化期望值(Expectation-Maximization, EM)演算法 [Dempster *et al.* 1977] 估算後所求得之常數。

若令$v_{h_i,w_j} = C_A(h_i)\frac{\lambda}{1-\lambda}P_B(w_j \mid h_i)+1$，則可得：

$$
\begin{aligned}
&P(w_j \mid h_i) \\
&= \frac{C_A(h_i)\frac{\lambda}{1-\lambda}P_B(w_j \mid h_i) + C_A(h_i w_j)}{\sum_{p=1}^{|V|}\left(C_A(h_i)\frac{\alpha}{\beta}P_B(w_p \mid h_i) + C_A(h_i w_p)\right)} \\
&= \frac{C_A(h_i)\left(\frac{\lambda}{1-\lambda}P_B(w_j \mid h_i) + P_A(w_j \mid h_i)\right)}{C_A(h_i)\frac{\lambda}{1-\lambda}\sum_{p=1}^{|V|}P_B(w_p \mid h_i) + \sum_{p=1}^{|V|}C_A(h_i w_p)} \\
&= \frac{C_A(h_i)\left(\frac{\lambda}{1-\lambda}P_B(w_j \mid h_i) + P_A(w_j \mid h_i)\right)}{C_A(h_i)\frac{\lambda}{1-\lambda} + C_A(h_i)} \\
&= \frac{\frac{\lambda}{1-\lambda}P_B(w_j \mid h_i) + P_A(w_j \mid h_i)}{\frac{\lambda}{1-\lambda}+1} \\
&= \lambda P_B(w_j \mid h_i) + (1-\lambda)P_A(w_j \mid h_i)
\end{aligned}
\qquad (2.2.2.7)
$$

此為模型插補法，其中 λ 為一權重(Weight)，其功能在於支配背景語言模型與調適語言模型之間孰輕孰重的傾向。

另外，前面曾提及的快取模型，也可視為一種模型插補法：

$$
\begin{aligned}
&P_{cache}(w_j \mid w_{j-N+1}, \cdots, w_{j-1}) \\
&= \lambda_c P_s(w_j \mid w_{j-N+1}, \cdots, w_{j-1}) + (1-\lambda_c)P_{cache}(w_j \mid w_{j-2}, w_{j-1})
\end{aligned}
\qquad (2.2.2.8)
$$

其中 $P_s(w_j \mid w_{j-N+1}, \cdots, w_{j-1})$ 為原本語言模型機率，$P_{cache}(w_j \mid w_{j-2}, w_{j-1})$ 為根據當前已辨識出之詞序列所得之語言模型機率，λ_c 則為用以支配二者重要性高低的權重。

由方程式 2.2.2.6 與 2.2.2.7 可看出，詞頻數合併法與模型插補法這兩種調適方法都需要先統計調適語料中詞彙之詞頻，用以衡量調適語料中詞彙出現情形，並據此計算其機率。就這一點而言，最大事後機率法可說是順應了統計式語言模型的設計理念。統計式語言模型的機率是根據詞頻計算而得，同樣地，詞頻數合

併法是將背景語言模型與調適語言模型的機率先還原成詞頻，再計算其機率。而模型插補法則是將根據訓練語料詞頻所求得之機率 $P_B(w_j \mid h_i)$ 與根據調適語料詞頻所求得之機率 $P_A(w_j \mid h_i)$ 作線性插補。這兩種語言模型調適方法皆如同統計式語言模型，試圖以計算語料中各詞彙機率值的方式，呈現出最接近特定語言使用環境的語言模型。

近年來，鑑別式訓練方法被應用於語言模型調適中。有別於最大事後機率法旨在最大化事後機率以找出最符合語料特性的語言模型，鑑別式訓練的目標是在追求辨識錯誤率的降低，透過訓練使得分類器能找出最接近正確參照轉寫的候選詞序列，以作為辨識結果，希望能因此降低辨識錯誤率。以機器代替人類將語音轉譯成文字，是語音辨識的最終目標，若能降低辨識錯誤率，便距此目標更近一些。鑑別式訓練的觀念，被應用於語言模型調適中，成為以降低辨識錯誤率為目標進行語言模型調適的方法，這部分將在下一節中作說明。

2.3 鑑別式語言模型訓練與調適

鑑別式訓練是以最小化分類錯誤(Minimum Classification Error)為目標，運用各種訓練方法，以訓練分類器有能力作出最正確的辨識。鑑別式訓練應用於聲學模型與語言模型中，都有一定的成效。

就語言模型而言，鑑別式訓練的目標在於調整語言模型中的參數，使得語言模型可以引導辨識器找出字/詞錯誤率最低的候選詞序列，以作為辨識結果。

例如 1998 年 Rigazio 等人 [Rigazio *et al.* 1998] 以最小化分類錯誤為目標，對語言模型機率及語言權重作鑑別式的訓練及調適，其目標在於訓練分類器(Classifier)，使其能從 *M* 個最佳辨識結果中擇其預期錯誤率(Expected Error Rate)最小者。語言模型機率代表詞序列在特定語言使用環境下的重要性，而語言權重(Language Weight)則是代表在辨識系統中，語言模型與聲學模型二者相較之下的可信賴度(Relative Reliability)。

隔年，Warnke 等人 [Warnke *et al.* 1999] 提出以最大相互資訊(Maximum Mutual Information Estimation, MMIE)與最小化分類錯誤(Minimum Classification Error, MCE)來訓練語言模型插補(Language Model Interpolation)的權重。

2002 年，Kuo 等人 [Kuo *et al.* 2002] 提出以最小分類錯誤為基礎的鑑別式語言模型訓練，目的在於區分最接近正確辨識結果的候選詞序列與其他候選詞序列。其方法為比較 *N* 連詞在正確辨識結果與候選詞序列中的出現情形，以決定如何增減該候選詞序列之機率值。若一個雙連詞(Bigram)出現在正確辨識結果中，但並未出現在候選詞序列裡，則增加語言模型中此雙連詞的機率值；反之，若該雙連詞並未出現在正確辨識結果中，但卻出現在候選詞序列裡，則降低此雙連詞在語言模型中的機率值。

2005 年，Kuo 與 Chen 等人 [Kuo & Chen 2005] 則是以估測語言模型機率的

方式，最大化訓練語料中詞圖的期望正確率，以期達到最小化詞錯誤 (Minimum Word Error, MWE)的目標。其方法為以求得最佳詞正確率為目標，透過延伸波式(Extended Baum-Welch)演算法推得語言模型參數估測之更新公式，透過一次次修正語言模型機率，以期能夠最大化詞正確率之期望值。

2007 年，Kuo 等人 [Kuo *et al.* 2007] 將有線狀態機(Finite-state Machine)的觀念用於鑑別式訓練，其方法為調整有線狀態圖(Finite-state Decoding Graph)中狀態之間的轉移權重(Transition-weight)，以達到最小化詞錯誤率的目標。

除了運用鑑別式訓練直接調整語言模型參數，近年來亦興起另一種鑑別式訓練模式，其方法為引進全域線性模型(Global Linear Model)架構以重新衡量基礎辨識器產生的 M 個最佳辨識結果，並以鑑別式訓練方法調整模型中的參數，使分類器能對基礎辨識器產生的 M 個最佳辨識結果進行重新排序，以期最接近正確辨識結果的詞序列能成為最終辨識結果，達到最小化辨識錯誤率的目標。此類型方法雖與其他鑑別式訓練方法同樣以最小化分類錯誤為目標對分類器進行訓練，但不像傳統 N 連語言模型是以機率來衡量詞序列之重要性，而是改以全域線性模型重新衡量各候選詞序列之間的差異，期待能夠成功訓練分類器從中選取最接近正確轉寫的詞序列。

此類型方法一開始被應用於自然語言處理領域，以全域線性模型重新衡量詞序列，並利用鑑別式訓練方法對分類器進行訓練，使分類器有能力選取最正確的詞序列作為輸出結果。例如以 Boosting 演算法訓練分類器在語法剖析(Natural Language Parsing) [Collins *et al.* 2000]的過程中找出最正確詞序列，或以 Perceptron 演算法訓練分類器進行詞性標示(Part-of-speech Tagging) [Collins 2002]。其中，Boosting 演算法以資料選取方法(Data Selection)選取全域模型參數並賦予參數適當估測值，而 Perceptron 演算法則是以最小平方誤差(Minimum Square Error, MSE)為前提進行全域線性模型的參數估測。

其後，Roark 等人[Roark *et al.* 2004a]採用 Perceptron 演算法進行鑑別式語言模型訓練。此外[Roark *et al.* 2004b]，又另採用條件隨機域(Conditional Random Field, CRF)方法進行鑑別式訓練，並以之與 Perceptron 演算法的實驗成果作比較。

隔年，Gao 等人[Gao *et al*. 2005a]提出以 Minimum Sample Risk 演算法求取全域線性模型參數，並將其與 Boosting 演算法和 Perceptron 演算法在語言模型調適上的效果作比較[Gao *et al*. 2005b]。Minimum Sample Risk 演算法則是將正確辨識結果與候選詞序列之間的編輯距離(Edit Distance)視為樣本風險(Sample Risk)，試圖搜尋出可使訓練語料之樣本風險降至最低的參數，並以此參數對測試語料作評估。

2006 年，Zhou 等人 [Zhou *et al.* 2006] 將 Ranking SVM 方法應用於以全域線性模型進行鑑別式語言模型訓練中，以進行語言模型調適，並將其與 Perceptron 演算法、Boosting 演算法與 Minimum Sample Risk 演算法的效果作比較。

2007 年，Gao 等人 [Gao *et al*. 2007] 提出以最大化熵值(Maximum Entropy, ME)搭配回歸(Logistic Regularization)方法，選擇最佳候選詞序列。同時，還提出 Bossted Lasso (BLasso)演算法，其意在採用 Boosting 演算法搭配回歸(Logistic Regularization)方法，以訓練全域線性模型。此外，亦將上述方法與 Perceptron 演算法、Boosting 演算法的實驗成果作比較。

同年，Roark 等人[Roark *et al.* 2007] 以有線狀態機(Weighted Finite-state Automata)實作全域條件式對數線性模型(Global Conditional Log-linear Models, GCLM)方法，並以之與 Perceptron 演算法的實驗結果作比較。

其後，Zhou 等人[Zhou *et al.* 2008]則是利用全域線性模型訓練所得之參數，對 *N* 連語言模型之機率值進行調整，形成一個擬傳統的 *N* 連模型(Pseudo-conventional *N*-gram Model)，為傳統 *N* 連語言模型與全域線性模型建立起一種新的合作關係。

圖 2-2 為使用鑑別式語言模型進行語言模型調適之架構。下一章將介紹這種以全域線性模型架構重新衡量基礎辨識結果的鑑別式語言模型，並介紹數種以最小化分類錯誤為目標進行全域線性模型參數估測之演算法。

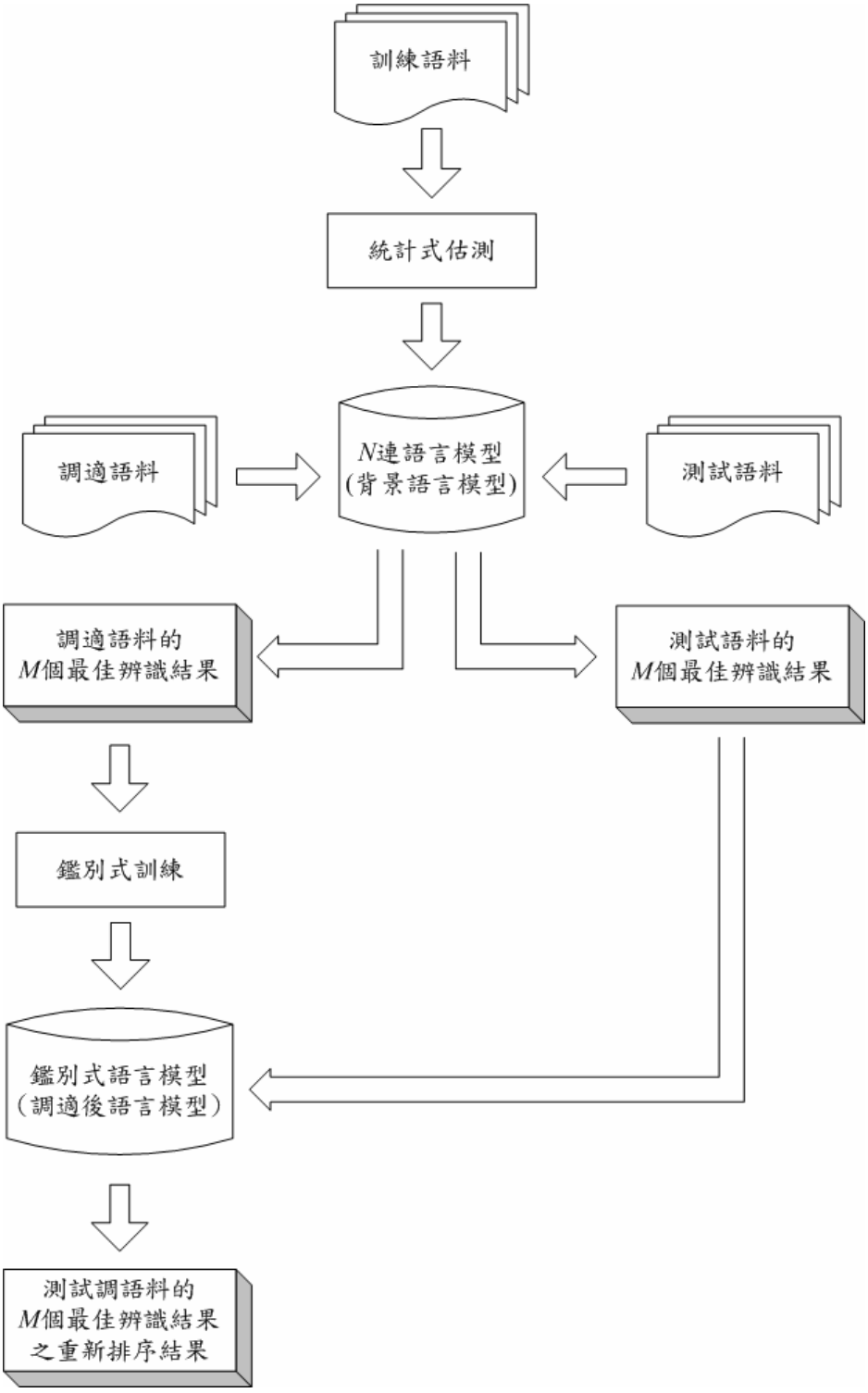


圖 2-2 使用鑑別式語言模型進行語言模型調適之架構

第三章 使用鑑別式語言模型重新排序辨識結果

3.1 基於歷史資訊之模型與全域線性模型

目前常見的 N 連語言模型是一種基於歷史資訊之模型(History-based Model)，它假設一個詞的出現與先此詞之前出現的歷史詞序列有關。近年來，全域線性模型(Global Linear Model)提供另一種看待訓練語料的觀點，透過鑑別式訓練調整全域線性模型之參數值，以此對基於歷史資訊之模型的排序結果作進一步的調整(Reranking)，以期得到更正確的辨識結果。

3.1.1 基於歷史資訊之模型

基於歷史資訊之模型是以一連串決策的過程來賦予當前事件(Event)一個意義，每下一個決定，都是根據先前的歷史來作判斷(Decision)，每個判斷都有一個相關的條件機率(Conditional Probability)。

統計式語言模型即是一個基於歷史資訊之模型，它將一個詞序列的機率視為一連串條件機率的乘積： $P(W) = P(w_1)P(w_2 \mid w_1)\cdots P(w_n \mid w_1, w_2, \cdots, w_{n-1})$ ，每一個詞都根據它的歷史詞序列作為條件(Condition)，用以決定目前這個詞的機率，這是一連串的決策過程。而找出對應於一段語音訊號 X 的最佳詞序列 W^* 的方法，便是找出機率值最高的之詞序列。

基於歷史資訊之模型在語言模型中有相當大的重要性，它嘗試去取得語言的規律性，統計各種歷史詞序列出現的機率，這個方法在實際運用上得到很不錯的辨識結果。

然而，基於歷史資訊之模型的限制，也正因為它根據歷史以作出決策，早

先對某段歷史的估量會影響接下來根據這段歷史所做出的判斷。就語言模型而言，亦即對一個詞 w_k 判斷，會影響其後以 w_k 作為歷史詞序列元素的詞之機率如 $P(w_i \mid w_1,\cdots,w_k,\cdots,w_{i-1})$。假設 w_k 本應為「口試」一詞，若統計式語言模型判斷它是「可是」一詞的機率高於它是「口試」一詞的機率，則接下來原本判斷應 P(如何 | 今天的口試) 的問題，可能就變成了判斷 P(如何 | 今天的可是) 的問題，影響到其後的辨識結果。

因此，全域線性模型架構被引進，它保留基礎辨識器產生的多條機率值較高的候選詞序列，以最小化辨識錯誤為目標對這些候選詞序列進行重新排序，以修正基於歷史資訊之模型可能產生的排序錯誤。

3.1.2 全域線性模型

全域線性模型 [Collins 2003] 是以基於歷史資訊之模型的辨識結果作為初始值，以使用者所定義的特徵(Feature)作為根據，將基於歷史資訊之模型的辨識結果利用這些特徵的線性組合(Linear Combination)作重新呈現(Representation)，估算出新的分數，以對基於歷史資訊之模型的辨識結果作重新排序。

線性模型的訓練資料為一個輸入/輸出組合所形成的集合 $\{X, Y^R\}$，其中 X 為欲辨識之輸入內容，Y^R 為正確辨識結果，線性模型的訓練目的在於找出 X 與 Y^R 之間的對應(Mapping)關係。其作法為利用歷史模型產生一個集合 $GEN(X)$，此集合中每一個元素，皆為 X 所可能對應之 Y，即 $Y \in GEN(X)$，而線性模型訓練，就是為了訓練辨識器將 $GEN(X)$ 中的所有元素重新排序，從 $GEN(X)$ 找出一個最接近正確的 Y^*，作為辨識結果。

全域線性模型的主角為特徵(Feature)，在全域線性模型中有一組全域的特徵(Global Features)，用來描述我們想要從訓練或測試資料中獲取的資訊。特徵是可以自由定義的，例如 N 連詞、詞性…等，皆可定義為特徵。每一個 Y 的內容都根

據其特徵，被重新表示為一個特徵向量(Feature Vector)。

另外，全域線性模型定義了一個誤差方程式(Loss Function)，用以估計訓練過程中，Y 與正確辨識結果 Y^R 之間的誤差程度，以最小化誤差為目標，找出一個能使誤差最小的 Y^*，作為辨識結果。

對於線性模型的訓練，主要是調整對應於特徵向量 $f(W)$ 的特徵權重(Weight)向量 λ，此權重根據訓練階段分數最高的候選辨識結果詞序列與實際上最正確的辨識結果詞序列之間特徵向量的差距來作出調整。特徵向量與特徵權重向量將每一個候選詞序列對應至一個實數值分數，以此分數對 $GEN(X)$ 作重新排序。

調整權重的目的是為了使實際上最正確的辨識結果能夠順利在測試階段得到最高的分數，成為最終產出的辨識結果，降低因基於歷史資訊之模型的限制而形成的排序誤差。

特徵向量與特徵權重向量相對應(Mapping)，意即特徵向量的第 d 維，對應至特徵權重向量的第 d 維。以下用 $f_d(W)$ 代表特徵向量的第 d 維，以 λ_d 代表特徵權重向量 λ 的第 d 維元素。$f_d(W)$ 記錄第 d 維特徵之值，而相對應之特徵權重 λ_d 表示該特徵之重要性。

特徵向量 $f(W)$ 與特徵權重向量 λ 由線性鑑別式(Linear Discriminant Function)作結合。一個線性鑑別式可以表示如下 [Duda et al. 2001]：

$$g(Y) = \lambda_o f_0(Y) + \sum_{d=1}^{D} \lambda_d f_d(Y) \qquad (3.1.2.1)$$

其中 $f_0(Y)$ 為一偏差值(Bias)，為模型提供給候選辨識結果 Y 的初始分數，$f_i(Y)$ 為特徵向量的某一維，λ_i 則為特徵向量所對應之特徵權重，其中 i 的值介於 0 到 D 之間。

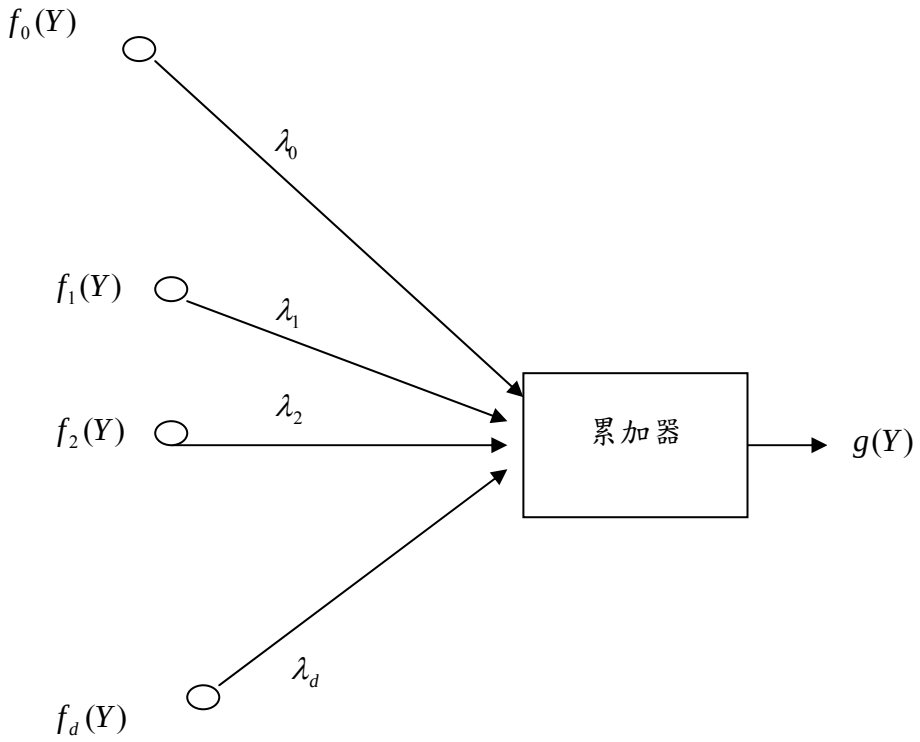


圖 3-1 線性鑑別式示意圖

根據方程式 3.1.2.1，線性鑑別式 $g(Y)$ 將候選辨識結果 Y 之特徵向量 $f(W)$ 與特徵權重向量 λ 這兩個向量作內積(Inner Product)運算，得到一個實數值，圖 3-1 為其示意圖。此實數值代表全域線性模型賦予候選辨識結果 Y 的分數，線性模型便是根據此實數分數對候選詞序列集合 $GEN(X)$ 中所有的候選詞序列 Y 作重新排序。在訓練階段，訓練此模型有能力提供一個正確的評分環境，在輸入訓練語料 X_{train} 的情況下，希望模型可以賦予最接近正確答案 Y_{train}^{R} 的詞序列 Y_{train}^{*} 最高分數；在測試階段，則是依此訓練好的模型，對測試語料 X_{test} 所對應的候選詞序列集合 $GEN(X_{test})$ 中所有候選詞序列 Y_{test} 作出評分，得分最高之詞序列 Y_{test}^{*} 便成為輸出結果。

3.2 鑑別式語言模型訓練之定義

鑑別式 *N* 連語言模型(Discriminative *N*-gram Language Modeling)針對基礎辨識器(Baseline Recognizer)所產生的多個分數較高之詞序列作訓練。有別於最大相似度估計(Maximum Likelihood Estimation, MLE)方法旨在找出與訓練資料最相近的語言模型，鑑別式訓練方法著眼於如何降低辨識的錯誤率。

鑑別式訓練對基礎辨識器(Baseline Recognizer)所產生的 *M* 個最佳辨識結果(*M*-best Recognition Hypotheses)作重新排序，以期較正確的辨識結果可有較高的排序，其對問題的定義如下[Gao *et al*. 2005]：

- 將訓練語料視為 $\{A_i, W_i^R\}$ 的組合, 其中 i 的值在 1 到 L 之間，其中 A_i 代表一段語音訊號，W_i^R 為此段語音訊號所對應之正確詞序列，L 為訓練資料筆數。
- 定義 $GEN(A)$ 為基礎辨識器對於一段語音訊號 A 所產生的 M 條候選詞序列之集合。
- 定義一組 $D+1$ 維的特徵向量 $f_d(W)$，其中 d 的值在 0 到 D 之間，其中 $f_0(W)$ 為三連(Trigram)語言模型所賦予詞序列 W 的機率之對數，即 $\log P_{Trigram}(\overline{W})$；而其他每一維度 d，則是記錄 W 中特定 N 連詞的出現次數，即 $f_d(\overline{W})$。
- 此外，定義一個 $D+1$ 維的參數向量 $\lambda = [\lambda_0, \lambda_1, \cdots, \lambda_D]$，參數向量的第 d 維對應至特徵向量之第 d 維，其中 d 的值在 0 到 D 之間。

詞序列 W 所得到的分數可寫成：

$$Score(W,\lambda) = \lambda f(W) = \sum_{d=0}^{D} \lambda_d f_d(W) \tag{3.2.1}$$

而辨識的目標則在於找出得分最高的詞序列W^*，亦即：

$$W^*(A,\lambda) = \arg\max_{W \in GEN(A)} Score(W,\lambda) \tag{3.2.2}$$

若要順利得到最正確的辨識結果，則必須先為鑑別式訓練求得一組最合適之參數。鑑別式訓練求得最正確辨識結果的方法，就是取得最佳參數解，使得辨識錯誤最小。以$Er(W^R,W)$表示某一候選詞序列與正確辨識答案之間的誤差，而SR(Sample Risk)表示所有訓練語料的辨識誤差總和，則鑑別式訓練的目的在於求得參數之最佳解：

$$\lambda^* = \arg\min_{\lambda} SR(\lambda) = \arg\min_{\lambda} \sum_{i=1\cdots M} Er(W_i^R, W_i(A_i,\lambda)) \tag{3.2.3}$$

能夠使得辨識結果W與正確辨識答案W^R之間的誤差最小之參數，即為鑑別式訓練所冀望求得之最佳參數解。

特徵向量與特徵權重向量如圖 3-2 所示。每一維特徵可能是一個單連詞，或是一個雙連詞，特徵向量記錄此項特徵在某一個候選詞序列中的出現次數，其相對應的特徵權重向量則是模型經過訓練後所得到的數據，其更新(Update)權重的依據是該候選詞序列W與正確參照轉寫W^R之間的差距，例如 Boosting 演算法是以二者間 N 連語言模型所賦予該詞序列的機率對數的差距(Margin)為依據，計算出現在該詞序列之特徵對辨識誤差所產生的影響，Perceptron 演算法是以二者特徵向量的差距作為更新特徵權重的依據，而 Minimum Sample Risk 則是透過線性搜尋找出最佳特徵權重，為使模型能從候選詞序列集合$GEN(A_{train})$中找出與正確答案W_{train}^R編輯距離(Edit Distance)最小的詞序列W_{train}^*。

在訓練階段中，模型的訓練目標在於使得最接近正確答案W_{train}^R的詞序列

W^*_{train} 能夠在 $Score(W,\lambda)=\sum_{d=0}^{D}\lambda_d f_d(W)$ 的評分下得到比候選詞序列集合 $GEN(A_{train})$ 中其他候選詞序列都還要高的分數；在測試階段，則是根據訓練階段建立起由特徵權重與特徵向量組成的評分機制，從候選詞序列集合 $GEN(A_{test})$ 中找出得分最高之詞序列 W^*_{test}，視之為與正確答案 W^R_{test} 最接近的候選詞序列，作為辨識的輸出結果。

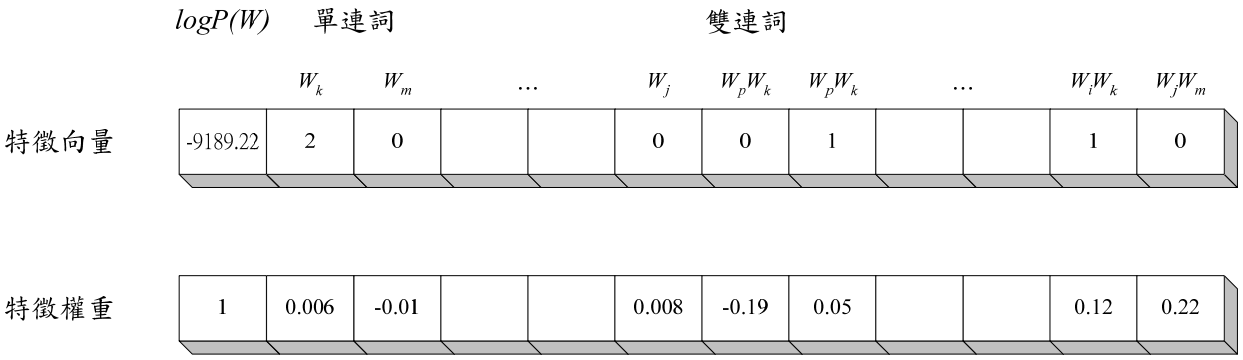


圖 3-2 特徵向量與特徵權重向量

3.3 Boosting 演算法

Boosting 演算法為一種機器學習方法，它以多個分類器(Classifier)組成的投票(Voting)機制來決定分類結果，以期最小化分類錯誤。近年來，Boosting 演算法被應用於鑑別式語言模型訓練。

3.3.1 Boosting 演算法

Boosting 演算法的設計理念，是結合數個弱勢分類器(Weak Classifier)成為一個功能強大的分類機制，以減少單一弱勢分類器所易造成的分類錯誤。

其後，Boosting 演算法被應用在重新排序(Reranking)的議題上[Freund *et al.* 1998]。像其他 Boosting 方法一樣，RankBoost 是結合數個對於當前訓練資料的弱勢排序，成為一個強勢排序。其排序方法是將分類問題設計成將分類對象分為「喜歡(Prefer)/較不喜歡(Less Prefer)」二類的問題，而受「喜歡」程度較高者，將得到較高的分數，並取得較領先的排名。

此學習演算法根據答案，試圖找出一個排序方式，使得排序錯誤越少越好，其目標在於確認排序的相對性，而非分數的差距大小。例如 2 個個體(Instance) A 與 B，此學習者(Learner)要學的是 A 必須排在 B 之前，以及這個證據在排序機制中應給予它多少的重要性，而非 A 與 B 之間的分數差距有多少。

此外，線性對數模型(Log-linear Model)與 Boosting 演算法在分類問題上的關聯性亦被提出[Friedman *et al.* 1998]，有助於後繼者將 Boosting 演算法運用於語言模型或自然語言處理之領域[Collins *et al.* 2000]。

3.3.2 Boosting 演算法於鑑別式語言模型之應用

Boosting 演算法主要意涵，在於認為在一個合理的評分環境下，正確答案 W^R 之得分應高於候選詞序列集合 $GEN(A)$ 中任一個候選詞序列 W 之得分 $(W \neq W^R)$，否則此評分環境是不合理的，亦即產生排序錯誤(Ranking Error)。

Boosting 演算法[Gao *et al*. 2005b] 將 (W^R, W) 二者的差數(Margin)定義為：

$$M(W^R, W) = Score(W^R, \lambda) - Score(W, \lambda) \tag{3.3.2.1}$$

而減損函數(Rank Loss Function)則定義為：

$$RLoss(\lambda) = \sum_{i=1\cdots L} \sum_{W_i \in GEN(A_i)} I[M(W_i^R, W_i)] \tag{3.3.2.2}$$

其中 $I[\pi] = 1$ if $\pi \leq 0$, and 0 otherwise.

Boosting 演算法試圖找出一組最合適之參數向量 λ，使得訓練/調適語料的減損函數值最小，亦即辨識結果最正確。然而，*RLoss* 為一階躍函數(Step Function)，無法直接求得其最佳 λ 解，因此 Boosting 演算法將 *RLoss* 取指數，以作為 *RLoss* 之上限(Upper Bound)，即：

$$ExpLoss(\lambda) = \sum_{i=1\cdots L} \sum_{W_i \in GEN(A_i)} \exp(-M(W_i^R, W_i)) \tag{3.3.2.3}$$

由於 *ExpLoss* 函數為一凸函數(Convex Function) ，因此可對其求得最佳解，亦即可使辨識錯誤率降至最低之 λ 解。

Boosting 演算法如圖 3-3，它進行的是一個資料選取的過程。除了第 0 維特徵外，其餘特徵權重初始值皆預設為 0。在每一回合中，選取一個可使誤差最小的特徵，更新其特徵權重，可重複選取，因此在進行 T 個回合後，只有小於或等於 T 維的特徵權重被更新為非 0 值。

1	Set $\lambda_0 = 1$ and $\lambda_d = 0$ *for* $d = 1 \cdots D$
2	For $t = 1 \cdots T$
3	Select a feature f_d which has largest estimated impact on reducing ExpLoss
4	Update $\lambda_d = \lambda_d + \delta_d$

圖 3-3　Boosting 演算法

3.4 Perceptron 演算法

感知機(Perceptron)為機器學習中的重要議題，以下將簡介感知機並說明它在鑑別式訓練中的應用。

3.4.1 感知機

人類發明電子設備的目的，是為了讓電子設備代替人類處理生活中的一些事務。事實上，人類解決問題最好的工具，就是人類的大腦。因此，在設計電子設備時，如何使電子設備可以像人類的大腦一樣自動且順利地運作，處理各類外界事物，就成為一門重要的學問。

隨著醫學發展，人類腦部的運作方式逐漸被瞭解，學者陸續提出類神經元的運算模型 [McCulloch *et al.* 1943]，以及主張學習現象的發生，乃在於神經元間的突觸產生某種變化所引起 [Hebb 1949]。

大腦的運作機制隨之逐漸被應用於資訊領域，產生感知機 [Lippman 1987]的設計，試圖以機器代替人類大腦對某些事件作出判斷與處理。感知機為單一類神經元(Neuron)，具有可調整的鍵結值(Synaptic Weight)以及閥值(Threshold)。這些參數的設定，使得感知機可以處理分類問題。

3.4.2 Perceptron 演算法於鑑別式語言模型訓練之應用

Perceptron 演算法則是以最小平方誤差(Minimum Square Error, MSE)的形式去估算樣本風險(sample risk) [Mitchell 1997]，以求得特徵權重向量λ可能的最佳解。其 *MSELoss* 方程式定義[Gao *et al*. 2005b]如下：

$$MSELoss(\lambda) = \frac{1}{2}\sum_{i=1\cdots M}(Score(W_i^R,\lambda) - Score(W_i,\lambda))^2 \quad (3.4.2.1)$$

若 *MSELoss* 函數對 λ_d 做偏微分，則為：

$$\begin{aligned} & G(\lambda_d) \\ & = \frac{\partial MSELoss(\lambda)}{\partial \lambda_d} \\ & = \sum_{i=1\cdots K}(Score(W_i^R,\lambda) - Score(W_i,\lambda))(f_d(W_i^R) - f_d(W_i)) \end{aligned} \quad (3.4.2.2)$$

上式是對全部 K 句訓練語料而言，換言之，對單一一句訓練語料而言，可寫成：

$$G(\lambda_d) = (Score(W_i^R,\lambda) - Score(W_i,\lambda))(f_d(W_i^R) - f_d(W_i)) \quad (3.4.2.3)$$

因為 *MSELoss* 有許多局部最佳解，因此 Perceptron 演算法採取隨機(Stochastic)策略，不再一次對 L 個訓練語句求參數之最佳解，而是改以對一個個單一訓練語料求最佳解，以用來更新參數：

$$\lambda_d = \lambda_d + \eta * G(\lambda_d) \quad (3.4.2.4)$$

其中 η 為學習步調大小(Learning Step Size)。更新特徵權重的方程式為 $\lambda_d = \lambda_d + \eta * (Score(W_i^R,\lambda) - Score(W_i,\lambda))((f_d(W_i^R) - f_d(W_i))$，亦有學者直接以 $\lambda_d = \lambda_d + \eta * (f_d(W_i^R) - f_d(W_i))$ 方式更新特徵權重。更新動作進行 T 個回合。

在 Perceptron 演算法中，特徵權重分為區域(Local)特徵權重與全域(Global)特徵權重兩組，區域特徵權重記錄的是 L 個訓練語句各自獨立計算之權重值，全域特徵權重則是在更新動作進行 T 個回合後，將 L 個訓練語句的區域特徵權重統整起來之結果，其式為：

$$(\lambda_d)_{Global} = \sum_{i=1}^{L}(\lambda_d^i)_{Local} \quad (3.4.2.5)$$

有學者指出[M. Collins 2002]，使用 Averaged Perceptron 演算法會得到較佳

結果，其方法為記錄 T 個回合中 L 個語料各自更新區域特徵權重之結果，最後再將 T 個回合中 L 個語料的參數全部加總起來取平均數，成為一個平均後的全域特徵權重：

$$(\lambda_d)_{avg} = \frac{(\sum_{t=1}^{T}\sum_{i=1}^{L}(\lambda_d^{t,i})_{Local})}{T*L} \qquad (3.4.2.6)$$

此參數即為 Averaged Perceptron 演算法訓練所得之最佳參數解。

本文採用的 Perceptron 演算法如圖 3-4 所示，就第 i 個訓練語料而言，選擇 $GEN(A_i)$ 中能得到最高 $Score(W_i,\lambda)$ 值的詞序列 W_i，用它與最接近正確參照轉寫的詞序列 W_i^R 之間每一維特徵值的差距 $f_d(W_i^R) - f_d(W_i)$ 來更新對應之特徵權重 λ_d，希望能以此調整評分機制，使最接近正確參照轉寫的詞序列 W_i^R 能成為最終辨識結果。

1	Initialize all parameters in the model, i.e. $\lambda_0 = 1$ and $\lambda_d = 0$ for $d = 1 \cdots D$
2	For $t = 1 \cdots T$, where T is Total number of iterations
3	For each training sample $(A_i, W_i^R), i = 1 \cdots L$
4	Use current model λ to choose the W_i from $GEN(A_i)$ with the largest $Score(W_i,\lambda)$ value
5	For $d = 1 \cdots D$
6	$\lambda_d = \lambda_d + \eta * (f_d(W_i^R) - f_d(W_i))$, where η is the size of the learning step

圖 3-4　Perceptron 演算法

3.5 Minimum Sample Risk 演算法

Minimum Sample Risk 演算法[Gao *et al*. 2005a]的基本精神與 Boosting 演算法相近，都是採用資料選取(Data Selection)方法來選出一個特徵的子集合(Subset)，這個子集合中的特徵，都對辨識錯誤率的下降有所幫助。

樣本風險(Sample Risk, SR)指的是對 L 句訓練語料而言，辨識結果詞序列W與正確詞序列W_R之間的編輯距離(Edit Distance)總和。鑑別式訓練的目的，在於最小化字錯誤率(Character Error Rate)，而字錯誤率的計算與編輯距離正相關，若能使編輯距離變小，則字錯誤率自然會降低，因此，Minimum Sample Risk 演算法將焦點放在如何使 L 句訓練語料的編輯距離降至最小，也就是使整個訓練語料的樣本風險降到最低。

格狀線性搜尋(Grid Line Search)在此演算法中扮演重要的角色，如圖 3-5 所

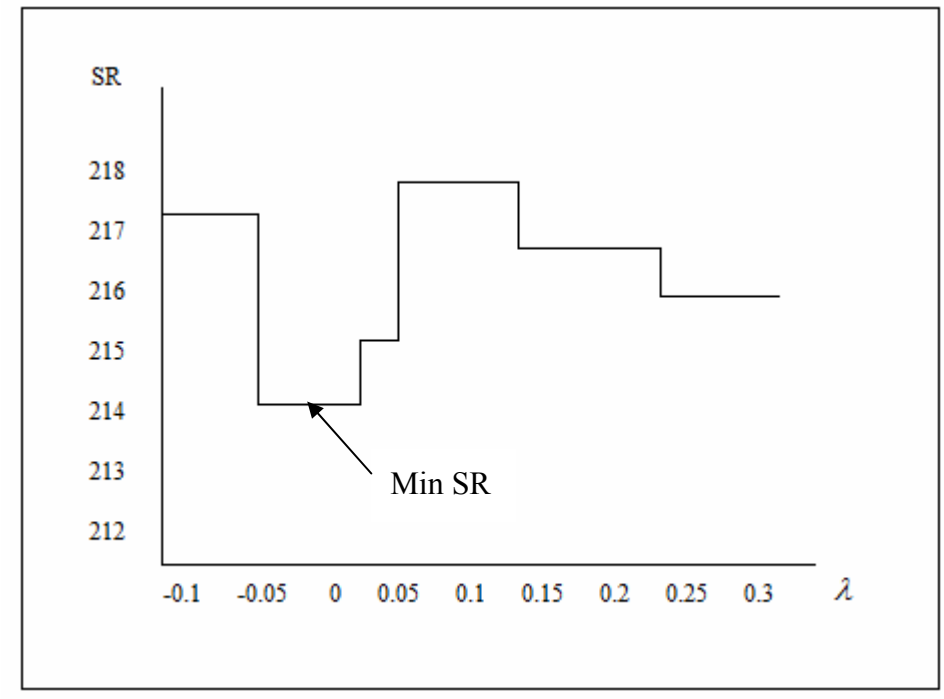


圖 3-5 格狀線性搜尋示意圖

示，格狀搜尋之對象為特徵權重λ_d之值，格狀指的是λ_d值之調整量。而搜尋的目標則是能使樣本風險最低的特徵權重。格狀線性搜尋的作法，是一次調整特徵向量中某一維特徵所對應之特徵權重。就某一維特徵權重而言，必須於某個區段內，以一個短距離作為單位，一次又一次調整特徵權重之值，並且逐一試算並記錄其對應之樣本風險。最後，再從所有試算過的特徵權重值中找出能使樣本風險降至最低者。

1	Set $\lambda_0 = 1$ and $\lambda_d = 0$ for $d = 1 \cdots D$
2	Rank all features and select the top K features, using the feature subset selection method.
3	For $t = 1 \cdots T$ (T = total number of iterations)
4	For each $k = 1 \cdots K$
5	Update the parameter of f_k using line search.

圖 3-6　Minimum Sample Risk 演算法

Minimum Sample Risk 演算法的流程如上圖所示，需先對 D 維特徵做資料選取，只選取 K 個特徵形成特徵子集合，再針對這 K 個特徵所對應之特徵權重做格狀線性搜尋，以為這 K 個特徵分別找出最佳權重值。

實作步驟依序為：

先以格狀線性搜尋為 D 維特徵分別找出最佳特徵權重，並以此計算每一維特徵的影響力(Effect)，影響程度的計算如下：

$$E(f_d) = \frac{SR(f_0) - SR(f_0 + \lambda_d f_d)}{\max\limits_{f_i, i=1 \cdots D} (SR(f_0) - SR(f_0 + \lambda_i f_i))} \tag{3.5.1}$$

$SR(f_0)$指的是只用第 0 維特徵所取得之辨識結果的樣本風險，$SR(f_0 + \lambda_d f_d)$指的是使用第 0 維與第 d 維特徵所取得之辨識結果的樣本風險。此式的意義的是加入第 d 維特徵後，樣本風險的下降幅度。D 維特徵影響程度需由大至小排序，影

響力最大者，被選為特徵子集合中第 1 個特徵 f_1。

在選擇特徵子集合中第 2 個特徵之前，需依據下式計算剩餘 D-1 維特徵各自與特徵子集合中第 1 個特徵 f_1 之間的交互關係(Cross-correlation)：

$$C(i,j)=\frac{\sum_{m=1}^{M}x_{mi}x_{mj}}{\sqrt{\sum_{m=1}^{M}x_{mi}^2\sum_{m=1}^{M}x_{mj}^2}} \tag{3.5.2}$$

x_{md} 的值為布林(Boolean)值，若 x_{md} 為 1，表示加入第 d 維特徵可使第 m 句訓練語料的編輯距離下降，若 x_{md} 為 0，表示並未下降

根據下式，選擇特徵子集合之第 2 個特徵：

$$j^*=\arg\max_{j=2\cdots D}\{\alpha E(f_j)-(1-\alpha)C(1,j)\} \tag{3.5.3}$$

其中 α 為一權重。根據待選特徵 f_j 的影響力 $E(f_j)$，以及 f_j 與 f_1 之間的交互關係 $C(1,j)$，可選出特徵子集合之第 2 個特徵。

至於特徵子集合之第 k 個特徵，$k=3\cdots K$，則依下式選出：

$$j^*=\arg\max_{j}\left\{\alpha E(f_j)-\frac{1-\alpha}{k-1}\sum_{i=1}^{k-1}C(i,j)\right\} \tag{3.5.4}$$

此式與選擇特徵子集合之第 2 個特徵的式子不同之處在於，需計算待選特徵 f_j 與特徵子集合中所有已被選取的特徵之間的交互關係。

根據上述四個步驟，可產生一個特徵子集合。最後，再針對特徵子集合中全部特徵所對應之特徵權重做格狀線性搜尋，便可得到 Minimum Sample Risk 演算法的訓練結果。

為降低 Minimum Sample Risk 演算法的計算量，因此有三個減少實際計算量的方法：

(1) 在做資料選取時，只考慮影響力較大的特徵，例如影響力排名前 S 名的特徵，以減少計算量。

(2) 定義一個反向的清單(Inverted List)，此清單記錄每一維特徵出現在哪些訓練語句中。有了這份清單，在計算某一特徵的相關資訊時，便無需考慮 L 句訓練語料，只需考慮確有該特徵存在的句子即可。

(3) 定義一個有效候選詞序列集合(Active Candidate Set)。如前所述，在鑑別式語言模型訓練中，某一詞序列的分數計算方式為 $Score(W,\lambda)=\sum_{d=0}^{D}\lambda_d f_d(W)$，由於最佳化特徵權重的過程中，一次只對一維特徵的權重值做最佳化，因此，計分式可表示為：

$$Score(W,\lambda)=\lambda f(W)=\sum_{d'=0\vee d'\neq d}^{D}\lambda_{d'} f_{d'}(W)+\lambda_d f_d(W) \qquad (3.5.5)$$

在對第 d 維特徵做最佳化時，$\sum_{d'=0\vee d'\neq d}^{D}\lambda_{d'} f_{d'}(W)$ 項的值不會改變。因此，可依此將某一訓練語句之所有候選詞序列分群，f_d 相同者分成一群，每一群中只保留 $\sum_{d'=0\vee d'\neq d}^{D}\lambda_{d'} f_{d'}(W)$ 值最高者。

第四章 以關鍵詞作為鑑別式語言模型之特徵

4.1 關鍵詞自動擷取方法

在電子設備迅速發展、資訊流通速度日益加快的時代，資訊的來源日趨多樣化，且訊息量之龐雜已到了可以用「知識爆炸」來形容。

在這樣的情況下，處理各式訊息的工作若交由人工處理，將需耗費大量人力與時間，且由於每個人的觀念或想法的差異，也未必能使資料得到具有一致性的分析，甚至進一步處理資料成為有用的資訊。

因此，自動化處理各種資訊的方法應時而生，例如資訊檢索系統、摘要系統，索引系統等等，可以對資料內容先行剖析，並依其主題或資料內容的相似度進行分類。但在此之前，必須對資料內容進行一項前處理的動作，那就是斷詞(Word Segmentation)。

斷詞的目的，是從資料中爬梳出承載訊息的最小單位，以供各項自動化系統可以對資料內容作出分析，以決定後續的分類或處理。斷詞的正確性，將影響其後各項自動化處理結果的正確性。以中文而言，其斷詞的難度較英文來得高，這是由於英文的詞與詞之間以空白(Space)字元作為間隔，而中文則無。此外，中文詞典(Lexicon)中詞彙之長度較短，詞意較不明顯。

關鍵詞擷取則是在斷詞的同時，擷取其中重要性可能較高的詞彙。目前關鍵詞擷取方法主要有幾種：

(1) 詞庫比對：對照事先建立的詞典(Lexicon)擷取關鍵詞。

(2) 文法剖析：利用文法規則，對照事先建立的詞典擷取關鍵詞。

(3) 統計式分析：計算詞頻以作為判斷。

(4) 關鍵詞自動擷取：根據字的排列規則，選擇其中重複出現次數較多的排列方式。

第 1、2 種方法都需依賴事先建立的詞典，無法擷取出詞典中未定義的詞彙，因此無法因應當前資料的內容擷取關鍵詞；第 3、4 種方法則不需仰賴詞典，因此可以因應當前資料的內容擷取關鍵詞，即使資料中出現新生詞彙，也不受影響。

關鍵詞自動擷取方法可以不受詞典限制，依文本(Context)內容擷取關鍵詞，較切合文本本身特性。在鑑別式語言模型訓練中，特徵對辨識結果的排序有決定性的影響，若能找出最切合文本內容的特徵，應對排序的正確性有所助益。因此，筆者認為利用文本本身詞彙使用模式的重複性所設計的關鍵詞自動擷取方法，其所擷取出的關鍵詞，可作為鑑別式語言模型的特徵。

關鍵詞自動擷取方法的目標在於找出最長的重複出現字串(Maximum Repeated Pattern)，其方法為[Tseng 1997]：

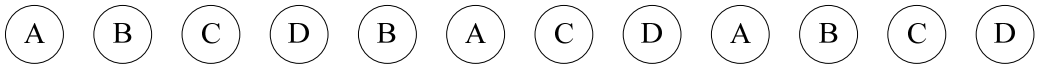
將一個中文字(Character)視為一個標記(Token)，而 *n* 個連續的字所形成的詞則視為 *n* 個連續標記(*n*-token)。其演算法如下：

- 步驟一：將輸入字串轉換為兩個連續標記(2-token)之串列
- 步驟二：合併 *n* 個連續標記(*n*-token)成為 *n*+1 個連續標記(*n*+1-token)，直到無法合併為止
- 步驟三：過濾不合規則的辭彙，其餘留存辭彙即為擷取結果

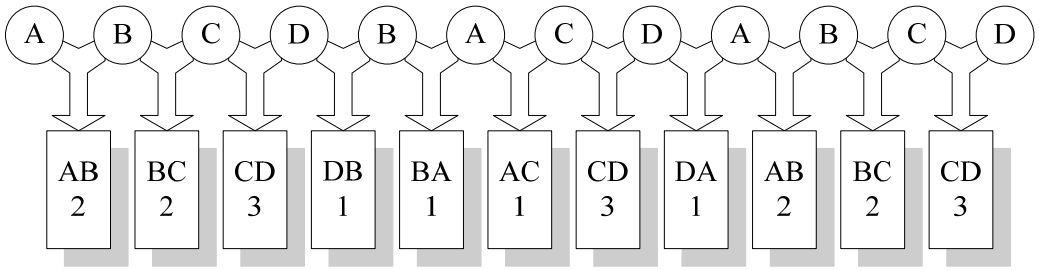
關鑑詞自動擷取過程如圖 4-1 中之範例所示，以此方法擷取關鍵詞，可以自動取得最長重複字串，除了不需依賴詞典，也可以避免中文詞典中詞彙長度較短、詞意較不明顯之問題。

輸入字串:ABCDBACDABCD
設threshold為2

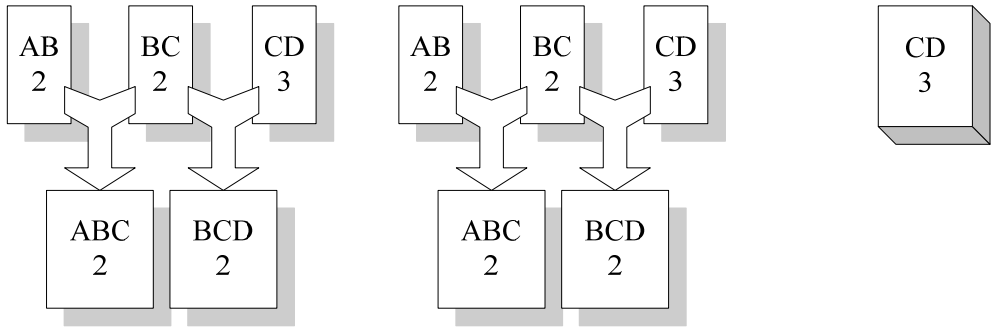
將輸入字串中的每一個字視為一個標記(token):



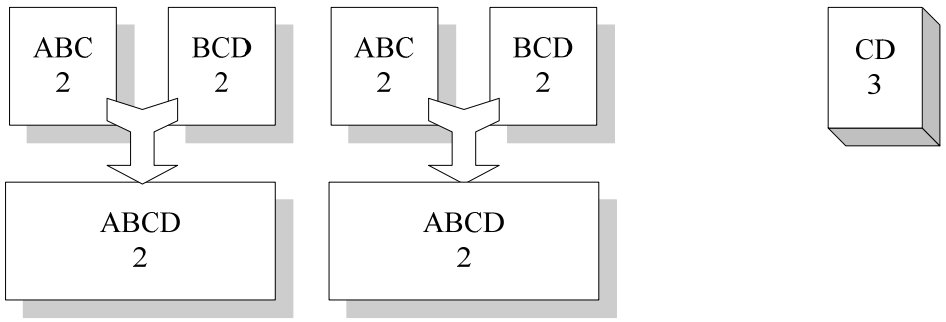
步驟一：將輸入字串轉換為兩個連續標記(2-token)之串列
捨棄出現次數小於threshold者，留存高於threshold者



步驟二：合併2-token成為3-token



步驟二：合併3-token成為4-token



結果：

圖 4-1 關鍵詞自動擷取範例

一開始先將輸入字串拆解成一個個以字為單位(Unit)的個體，接著將相鄰的兩個個體合併，並記錄它們在輸入字串中的出現次數。

其次，過濾掉出現次數小於預設閥值(Threshold)之個體。這些個體的出現次數小於閥值，且即使未來將其與相鄰個體合併，其出現次數也不會高於預設閥值，例如「DB」的出現次數為 1，它的前一個相鄰個體「CD」的出現次數為 3，在它們合併為「CDB」後，其出現次數僅僅為 1，小於預設閥值，故可捨棄。

在捨棄出現次數小於預設閥值的個體，並保留出現次數高於或等於預設閥值的個體後，便繼續回到合併的步驟。合併與過濾的過程不斷重複遞迴，直至沒有相鄰的個體可以合併為止。例如圖 4-1 中，就字串「ABCDBACDABCD」而言，「CD」這個標記出現三次，其中第二個「CD」在形成二個連續標記(2-token)後，雖出現次數高於預設閥值，但它已無法再與相鄰個體合併為三個連續標記(3-token) ，因此，「CD」立即被視為一個擷取出的關鍵詞；與此同時，第一個「CD」與第三個「CD」因為尚能繼續與相鄰個體合併，所以都與相鄰之「BC」合併為「BCD」，進入三個連續標記(3-token)階段繼續受到檢視。

最後的擷取結果，得到「ABCD」與「CD」這兩個關鍵詞。使用這個方法擷取關鍵詞的優點，在於關鍵詞是經由文本內容字串不斷遞迴所產生，不必依賴外來詞典資訊，一方面可以取得訓練語料之語言習慣，一方面則是可以掌握到詞典中不存在之新生詞彙或流行用語。

4.2 增加關鍵詞自動擷取所得長詞作為鑑別式訓練之特徵

鑑別式 N 連語言模型將基礎辨識器所產生的 M 個最佳辨識結果中出現之單連詞與二連詞作為特徵，計算在每個候選詞序列中，各維特徵出現之次數。筆者擬透過關鍵詞自動擷取方法，增加語句中擷取出之最長重複出現字串作為鑑別式訓練之特徵。

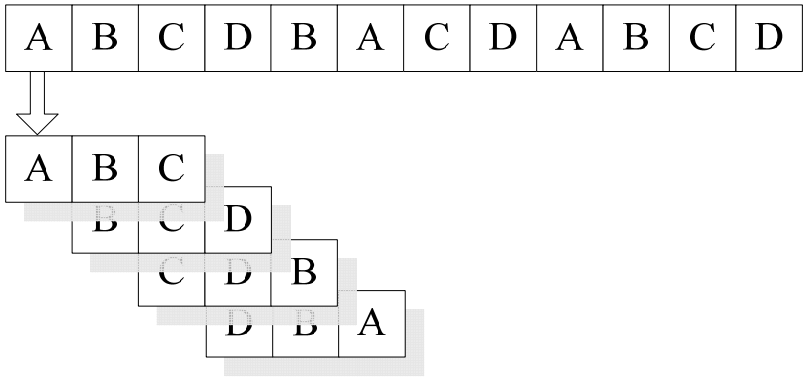
統計式 N 連語言模型是根據歷史來估算一個詞的機率 $P(w_i \mid w_1 w_2 \cdots w_m)$，其中基於 N-1 階馬可夫假設設計的 N 連模型 $P(w_i \mid w_{i-N+1} \cdots w_{i-1})$ 是根據一個詞的前 N-1 個歷史詞形成的詞序列來決定其機率。若 N 太小，就長詞而言，若非事先建立在詞典中的長詞，則容易被截斷，視為多個詞；若 N 太大，則不易取得足夠的長詞資料，以計算相對應的機率，此時必須仰賴後向(Back-off)機制，利用較短的 N 連詞來補其不足。因此，使用 N 連模型時，需要決定 N 的適當大小為何。

關鍵詞自動擷取所得到的長詞與歷史模型中的 N 連詞使用有其共同點：它們都試圖取得語言中的規律性。至於二者的相異處，則是關鍵詞的長度則是依語言實際運用情形找出的慣用詞，其詞長不是固定的預設長度，而是依訓練語料中語言實際使用習慣擷取而出，不需依賴詞典。

圖 4-2 中的關鍵詞自動擷取方法為圖 4-1 簡化後的示意圖。由圖 4-2 中可看出三連語言模型與關鍵詞自動擷取方法在處理同一段輸入字串時所採取的不同方法。三連語言模型是以固定為 3 的長度去檢視輸入字串，它先將輸入字串視為一個大單元，再以固定長度將此大單元分割為較小的單元來檢視；而關鍵詞自動擷取方法則是以合併(Merge)小單元成為較大單元的模式遞迴擷取詞彙。由於看待輸入字串的觀點不同，造成結果上的差異，就「ABCDBACDABCD」這個例子而言，三連語言模型並不會檢視到「ABCD」這個關鍵詞，因為這個詞的長度為 4，超過了三連語言模型所預設的詞長。

輸入字串:ABCDBACDABCD

三連語言模型處理輸入字串的方式:



關鍵詞自動擷取處理輸入字串方式(預設閥值2):

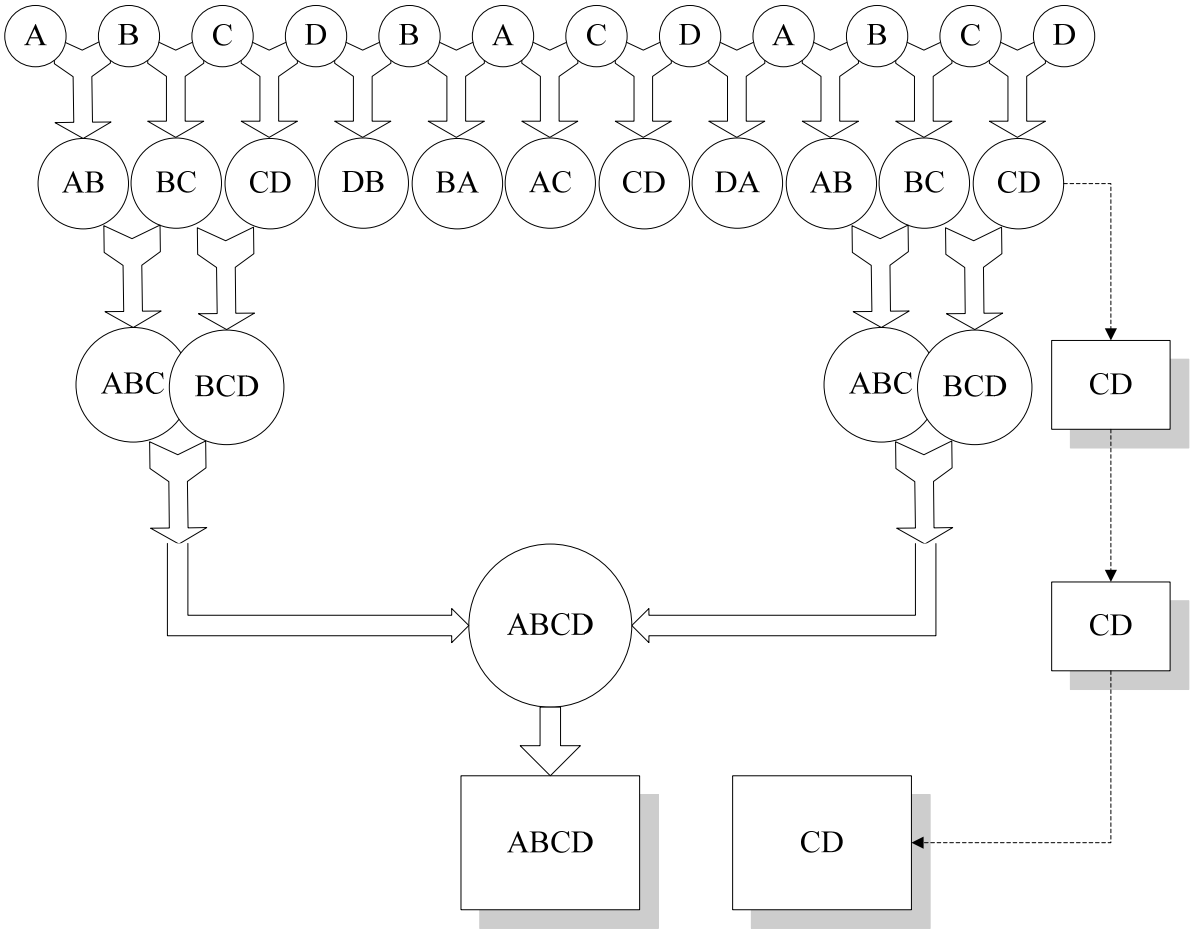


圖 4-2　*N* 連詞與關鍵詞自動擷取在處理相同字串時所採取的不同方法

第五章 實驗架構與結果

本章包括幾個部分，首先將介紹實驗架構，包括台師大之大詞彙連續語音辨識系統，並說明聲學模型訓練語料、語言模型訓練語料、語言模型調適語料，以及測試語料的來源與特性。

其次將以鑑別式語言模型對基礎辨識器產生的 M 個最佳辨識結果作重新排序，說明在前述實驗架構下，採用前人鑑別式語言模型理論的實作結果，以及使用筆者提出之理論所得到的結果。

5.1 實驗架構

5.1.1 台師大之大詞彙連續語音辨識系統

(一) 特徵擷取

台師大之大詞彙連續語音辨識系統使用兩種方法進行特徵擷取：一是使用梅爾倒頻譜係數(MFCC)作為語音訊號的特徵參數，二是異質性線性鑑別分析(Heteroscedastic Linear Discriminant Analysis, HLDA)配合最大相似度線性轉換(Maximum Likelihood Linear Transformation, MLLT)取得特徵參數。

本論文實驗架構中，前端處理處理的部分是採用異質性線性鑑別分析(HLDA) 配合最大相似度線性轉換(MLLT)以取得特徵參數。

(二) 聲學模型

中文的音節由聲母與韻母這兩個次音節組成。在聲學模型中，我們分別為聲母和韻母各建立一個模型，代表聲母的是 INITIAL 模型，代表韻母的是 FINAL

模型，共有 22 個 INITIAL 模型，38 個 FINAL 模型，以及 1 個代表靜音的 SIL 模型。由於聲母會受搭配之韻母影響而有所變化，因此使用右相關聯模型 (Right-Context-Dependent Model, RCD)，將聲母細分為 112 個 INITIAL 模型。加上韻母與靜音模型，共有 151 個聲學模型。每一個隱藏式馬可夫模型中有 3~6 個狀態(State)，其中每一個狀態為 1~128 個高斯分布所組成的高斯混合分布。

(三) 詞典建立

中文約有 7000 個單字詞，藉由合併單字詞的方式，可以產生一些新詞。本系統建立新詞的方式，是根據語料中的單字詞，以統計式方法自動建立新的複合詞(Compound Words)。

對於語料中連續兩個單字詞，例如 $w_i w_j$ ，分別以正向和逆向觀點建立此二連詞機率，再取幾何平均。正向觀點建立的是前二連(Forward Bigram)機率 $P_f(w_j \mid w_i)$ ，逆向觀點建立的是後二連(Backward Bigram)機率 $P_b(w_i \mid w_j)$ ，幾何平均為 $FB(w_i, w_j) = \sqrt{P_f(w_j \mid w_i) P_b(w_i \mid w_j)}$ 。前後二連(Forward and Backward Bigram)的幾何平均需高於一個基準值(Threshold)，才會被合併為一個新的複合詞。

在本系統的原始詞典中包含六萬八千個詞，每一個詞的長度為一至四字。文字語料先根據原始詞典作斷詞，再根據前後二連的幾何平均判斷是否合併為新詞，經由數次迭代與不同基準值設定，產生約五千個新的複合詞，詞長在二至四字之間。將新的複合詞加入原始詞典後，可得到一個含有七萬二千個詞的新詞典。

(四) 詞彙樹複製搜尋

本系統是採取由左至右(Left-to-right)、音框同步方式(Frame-synchronous)進行詞彙樹複製搜尋[Aubert 2002]。詞彙樹中每一個分枝(Arc)代表一個 INITIAL 或 FINAL 模型，由根節點(Root)到葉節點(Leaf)的路徑代表一個詞，若有多條路徑，

則代表一些發音相同的詞。在每一個音框中，若有片段路徑(Partial Path)到達葉節點時，代表產生了一個完整的詞。

在搜尋每一個音框時，因為存在多個不同歷史詞序列，因此每一個音框會同時存有多棵複製的詞彙樹(Tree Copies)以代表不同歷史詞序列，若搜尋時產生的片段路徑具有相同的歷史詞序列，則會儲存在同一棵詞彙樹中，進行隱藏式馬可夫模型狀態層次的維特比(Viterbi)動態規劃搜尋。

由於留存的隱藏式馬可夫模型的狀態節點會隨著音框數呈指數倍增加，因此採取光束搜尋(Beam Search)方法適當裁剪分數較低的路徑或節點。分數的計算包括詞彙樹內部節點(Internal Node)中儲存搜尋過程累計的解碼分數(Decoding Score)、聲學模型向前看分數(Acoustic Model Look-ahead Score)與語言模型向前看分數(Language Model Look-ahead Score)，以此三種分數作為是否裁剪節點的根據。

本系統中語言模型向前看分數是採用單連(Unigram)語言模型向前看技術，就詞彙樹中某一內部節點而言，其語言模型向前看分數為經由此節點所能到達之所有葉節點中單連語言模型機率最高者。

詞彙樹中留存的葉節點代表可能的候選詞，每一個音框中皆儲存分數較高的葉節點資訊，包括開始音框與結束音框、聲學模型解碼分數、歷史詞序列等，以此建立詞圖(Word Graph)。

(五) 詞圖搜尋

詞圖為詞彙樹複製搜尋過後留存的詞段所建立的圖，詞圖中每一個詞段皆有對應的開始與結束音框、聲學模型解碼分數與歷史詞資訊。

詞圖搜尋是根據詞段對應的開始與結束音框與歷史詞資訊對詞圖進行搜尋，以建立多條歷史詞序列，並根據搜尋過程中累計的解碼分數、聲學模型解碼

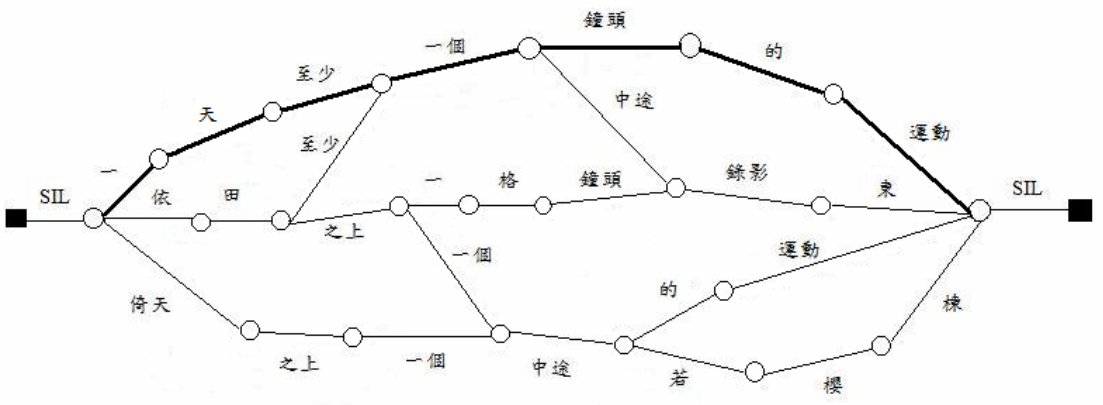


圖 5-1 詞圖範例

分數與語言模型分數計算出各歷史詞序列之分數。詞圖搜尋可得到多個詞序列及其對應分數，可取分數最高者直接作為辨識結果，亦可保留分數較高的數個詞序列進一步做鑑別式訓練，以期得到更正確的辨識結果，本文的研究屬於後者。

5.1.2 實驗語料

實驗語料皆來自公視新聞(Mandarian Across Taiwan—Broadcast News, MATBN)。公視新聞語料是 2001 年至 2003 年間由中央研究院資訊所口語小組[SLG]與公共電視台[PTS]合作錄製的，包含內場與外場兩個部分，內場為主播語料(Studio Anchors)，外場則有採訪記者(Field Reporters)語料與受訪者(Interviewees)語料。

論文的實驗採外場採訪記者語料。這是因為內場主播語料大部分來自同一主播，為避免語者相依(Speaker Dependent)現象造成實驗偏差，故不用之；至於外場受訪者語料，則有過多語助詞，亦不適用。外場記者語料中，有 25.5 小時的語料(5774 個句子)被挑選出來作為聲學模型訓練語料，這些語料之中都不包含語助詞。

(一) 聲學模型訓練語料

聲學模型語料取自 2001 至 2002 年間的公視新聞語料，男女性語者語料各半，其中男性語者語料有 766.68 分鐘，女性語者語料 766.78 分鐘，總字數 477098 字，總詞數 289513 詞。

(二) 語言模型語料

背景語言模型的訓練語料來自 2001 至 2002 年中央通訊社(Central News Agency, CNA)，大約包含一億五千萬(150M)個中文字。背景語言模型為一個三連(Trigram Language Model)語言模型，且採用 Katz Back-off Smoothing 方法因應資料稀疏(Data Sparseness)問題。此語言模型是透過 SRI Language Modeling Toolkit (SRILM)訓練所得結果。

語言模型調適語料取自 2003 年公視新聞，選擇外場記者語料，共 292 句，26219 字，約 1.5 小時。測試語料與調適語料屬於同時期語料，亦取自 2003 年公視新聞，選擇外場記者語料，共 230 句，18461 字，約 1 小時。

5.1.3 語言模型評估與基礎實驗結果

線性鑑別式訓練並不會改變語言模型的複雜度(Perplexity)，因此主要評估方式為字或詞的錯誤率。此處以字錯誤率(CER)作為評估線性模型的依據。

表 5-1 中資料為訓練語料與測試語料經由基礎辨識器所得辨識結果的字錯誤率。本文中實驗的基礎(Baseline)辨識率即為測試語料之字錯誤率 18.12%，如前所述，鑑別式訓練的目的在於降低辨識錯誤率，因此，在本文實驗中以降低字錯誤率為目標，訓練鑑別式語言模型以進行基礎辨識結果的重新排序，而所謂以降低字錯誤率為目標，指的是在訓練階段，以每一訓練語句之 M 個最佳辨識結果中與正確參照轉寫相較之下字錯誤率最低的那條詞序列作為分類器的學習對

象，調整模型參數，以使字錯誤率最低的那條詞序列能在多個辨識結果之中得到最高的排名，目的是為了在測試階段，分類器能對每一測試語句的 M 個最佳辨識結果進行重新排序，從中選出字錯誤率最低者作為最終辨識結果。若是實驗結果字錯誤率低於 18.12%，就表示鑑別式訓練確實能有效地降低字錯誤率。

語料庫 ＼ 錯誤率	字錯誤率(%)
調適語料（292 句）	20.77
測試語料（230 句）	18.12

表 5-1 基礎實驗結果

5.2 前人理論實驗結果

本論文實驗是以線性鑑別式訓練對基礎辨識器(Baseline Recognizer)所產生的 100 個最佳辨識結果(100-best Recognition Hypotheses)重新排序。

5.2.1 Boosting 演算法實驗結果

本實驗中的特徵取自語料中的單連詞與雙連詞，且以訓練語料中字錯誤率最小的候選詞序列代表正確參照轉寫的詞序列W^R。

本實驗依據[Collins *et al.* 2000]中提及的演算法實作，如圖 5-2 所示。其中$W_{i,j}$代表第 i 個訓練語句的第 j 個候選詞序列，$L(W_{i,j})$為基礎辨識器賦予$W_{i,j}$的機率對數。

A_k^+, A_k^-用來統計特徵 k 在某一辨識假設中的出現情形，若特徵 k 出現在第 i 個訓練語句之字錯誤率最小的候選詞序列W_i^R中，但並未出現在第 i 個訓練語句的第 j 個候選詞序列$W_{i,j}$中，則視$W_{i,j}$為A_k^+之成員；反之，若特徵 k 出現在第 i 個訓練語句的第 j 個候選詞序列$W_{i,j}$中，但並未出現在第 i 個訓練語句之字錯誤率最小的候選詞序列W_i^R中，則視$W_{i,j}$為A_k^-之成員。其式如下：

$$\begin{aligned} A_k^+ &= \left\{(i,j) : \left[h_k(W_i^R) - h_k(W_{i,j})\right]\right\} = 1 \\ A_k^- &= \left\{(i,j) : \left[h_k(W_i^R) - h_k(W_{i,j})\right]\right\} = -1 \end{aligned} \qquad (5.2.1.1)$$

其中 $h(.)$ 為一指標函數(Indicator Function)，代表某特徵是否出現在某一個候選詞序列中，若是，則其值為 1，反之則為 0。

Input Examples $W_{i,j}$ with initial model scores $L(W_{i,j})$, set A_k^+, A_k^- for each feature $h_k, k = 1 \cdots m$

Initialize $\bar{\alpha}^0 = \{\alpha_0, 0, 0, \cdots, 0\}$, *for all* $i, 2 \le j \le n_i$, set margins

$$M_{i,j} = \alpha_0 [L(W_i^R) - L(W_{i,j})]$$

Repeat for t = 1 to N

- for $k = 1$ to m
  - Set $E_k^+ = E_k^- = 0$
  - for $(i, j) \in A_k^+, E_k^+ = E_k^+ + e^{-M_{i,j}}$
  - for $(i, j) \in A_k^-, E_k^- = E_k^- + e^{-M_{i,j}}$
  - $BestWt(k) = \frac{1}{2} \log \frac{E_k^+}{E_k^-}$
  - $BestLoss(k) = 2\sqrt{E_k^+ E_k^-} - E_k^+ - E_k^-$
- Choose $k^* = \arg\min_k BestLoss(k),$ $\delta^* = BestWt(k^*)$
- For $(i, j) \in A_k^+, M_{i,j} = M_{i,j} + \delta^*$
- For $(i, j) \in A_k^-, M_{i,j} = M_{i,j} - \delta^*$

圖 5-2　Naïve Boosting 演算法

E_k^+ 代表所有屬於 A_k^+ 之成員其與字錯誤率最小的候選詞序列 W_i^R 之差距之負數取指數結果，E_k^- 代表所有屬於 A_k^- 之成員其與字錯誤率最小的候選詞序列 W_i^R 之差距之負數取指數結果。

在每一回合中，選出能使誤差(Loss)最小的特徵 k^*，及其對應之最佳權重調整量 δ^*，並以此權重調整量去更新 k^* 的特徵權重。雖權重非零特徵數大致上與訓練回合數成正比，但由於特徵可以重複選取，因此並非程式進行多少回合，特徵權重非零之特徵就有多少個，而是小於或等於回合數，如圖 5-3 所示。

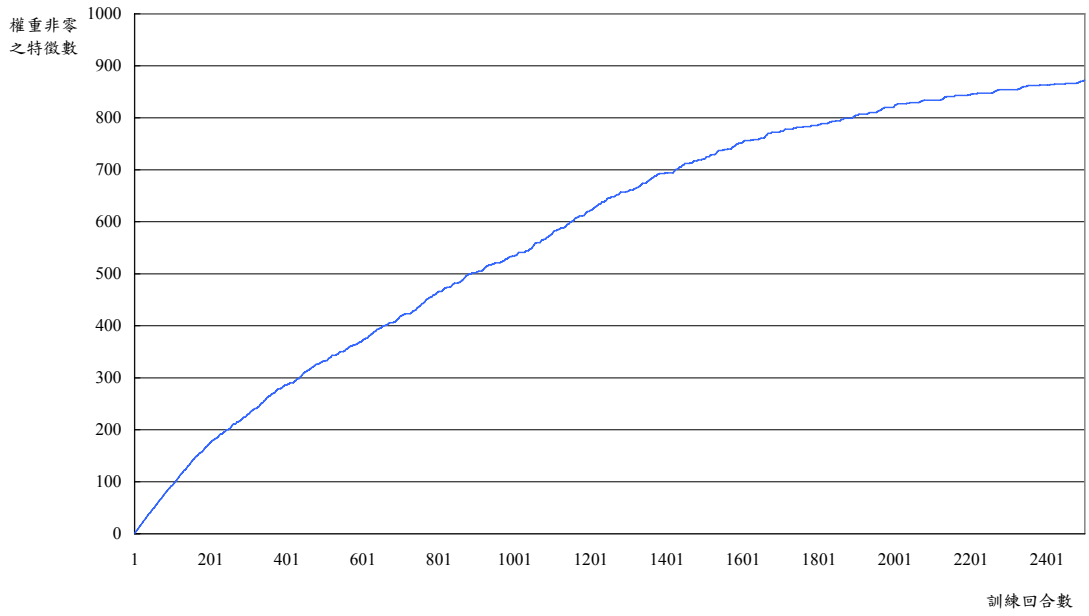


圖 5-3 Boosting 演算法實驗中訓練回合數與權重非零特徵數之關係

Boosting 演算法實驗數據較佳者出現在 64 回合左右，獲得之最低字錯誤率為 18.09%。表 5-2 中列舉字錯誤率較低之若干回合數據，由數據中可見，特徵數與訓練回合數成正比。然而與此同時，在某些連續回合中，雖特徵數增加，但字錯誤率並無變化，表示該回合所選取之特徵對字錯誤率的降低並無太大幫助。例如第 413～433 回合間，雖然非零特徵權重數，也就是曾在資料選取時被選中作為更新權重對象的特徵權重數是持續增加的，但字錯誤率仍維持在 18.10%，並未有所升降。

訓練回合	字錯誤率 (%)	非零特徵權重數		訓練回合	字錯誤率 (%)	非零特徵權重數
1	18.11	1		418	18.10	290
2	18.11	2		419	18.10	291
3	18.11	3		420	18.10	292
63	18.10	59		421	18.10	293
64	18.09	60		422	18.10	294
65	18.09	61		423	18.10	294
66	18.11	62		424	18.10	294
248	18.10	201		425	18.10	295
249	18.10	201		426	18.10	296
250	18.10	201		427	18.10	297
251	18.10	202		428	18.10	297
413	18.10	290		429	18.10	298
414	18.10	290		430	18.10	298
415	18.10	290		431	18.10	298
416	18.10	290		432	18.10	298
417	18.10	290		433	18.10	298

表 5-2 Boosting 演算法實驗數據

圖 5-4 顯示 Boosting 演算法實驗結果中，字錯誤率與訓練回合數之關係。由圖中可見，字錯誤率於某些訓練回合中略為下降。

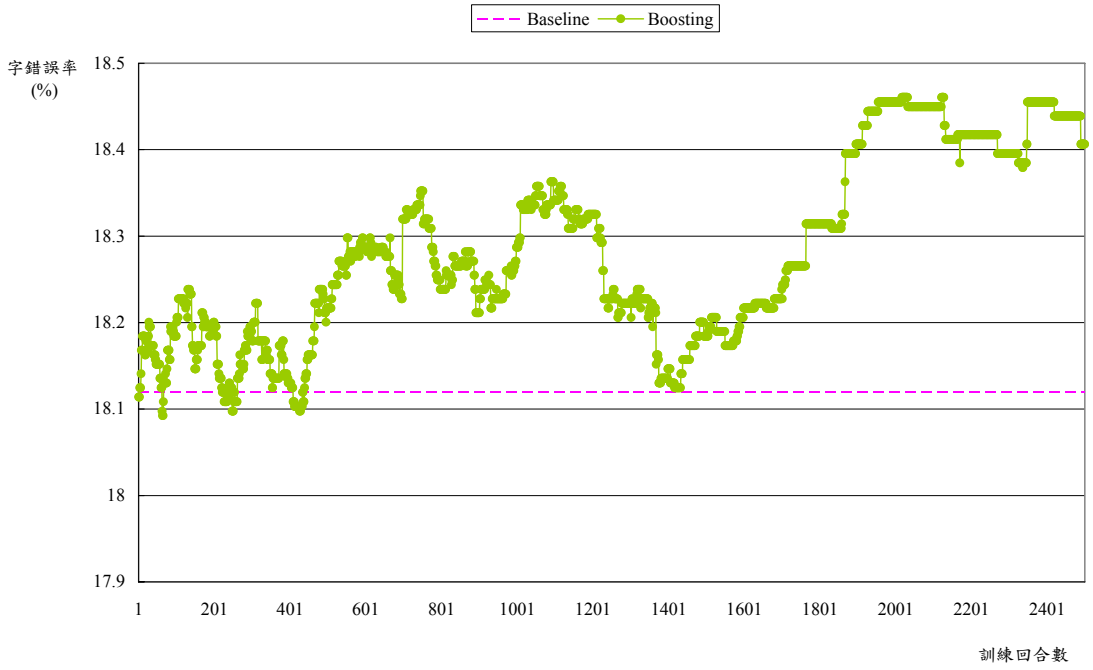


圖 5-4　Boosting 演算法實驗結果

5.2.2 Perceptron 演算法實驗結果

本實驗主要依據[Gao *et al*. 2005b]上描述之 Averaged Perceptron 演算法實作。表 5-3 為實驗相關設定。

項目	定義內容
特徵	語料中的單連詞與雙連詞
代表正確參照轉寫的詞序列	訓練語料中字錯誤率最小的候選詞序列
學習步調大小(Learning Step Size)初始值	0.01
學習步調大小更新方式	每一回合將學習步調大小乘以 0.95

表 5-3 實作 Perceptron 演算法之若干項目定義

關於特徵權重的部分，實驗中是以 $\lambda_d = \lambda_d + \eta * (f_d(W_i^R) - f_d(W_i))$ 方式更新區域特徵權重。此外，由於實驗中每一回合特徵權重之更新量不多，若像方程式 3.4.2.6 中所述方法為全域特徵權重取平均數，則更新量過小，不足以對實驗結果產生影響，因此改用下列方式計算全域特徵權重：

$$(\lambda_d)_{Global} = \sum_{t=1}^{T}\sum_{i=1}^{L}(\lambda_d^{t,i})_{Local} \qquad (5.2.2.1)$$

也就是同樣累計每一回合每一訓練語句之區域特徵權重，但最後並不除以訓練語句數 L 與訓練回合數 T。

此外，經過試驗，累加多回合之區域特徵權重以計算計算全域特徵權重的 Averaged Perceptron 演算法，其效果較直接採用最後一個回合之區域特徵權重加總以計算出全域特徵權重的 Perceptron 演算法來得好。

Averaged Perceptron 演算法實驗結果如表 5-4 所示，較好的數據在 1~50 回合之間，最低字錯誤率在第 14 回合時出現，其值為 17.94%。

訓練回合	字錯誤率(%)		訓練回合	字錯誤率(%)
1	18.09		26	18.09
2	18.09		27	18.04
3	18.10		28	18.05
4	18.11		29	18.04
5	18.11		30	18.04
6	18.14		31	18.01
7	18.12		32	18.01
8	18.08		33	18.00
9	18.09		34	17.98
10	18.05		35	17.98
11	17.99		36	17.96
12	17.99		37	17.95
13	17.95		38	17.96
14	17.94		39	17.99
15	17.95		40	17.97
16	17.96		41	17.99
17	18.01		42	18.01
18	17.97		43	18.00
19	18.01		44	18.02
20	18.03		45	18.03
21	18.03		46	18.03
22	17.96		47	18.04
23	18.02		48	18.03
24	18.03		49	18.03
25	18.09		50	18.03

表 5-4　Averaged Perceptron 演算法實驗數據

圖 5-5 中，Averaged Perceptron 代表前述以方程式 5.2.2.1 為特徵權重取平均數的實驗結果，Perceptron 代表直接使用第 T 個訓練回合的全域特徵權重的實驗結果。明顯可見，Averaged Perceptron 演算法的實驗成效較佳，能夠得到較低的辨識字錯誤率。

此外，在 Perceptron 演算法實驗中，字錯誤率的下降速度較 Boosting 演算法來得快，這是由於 Perceptron 演算法從一開始即採用所有特徵之權重進行鑑別式訓練的計分，而 Boosting 演算法一開始所選取的特徵數尚且不足之故。

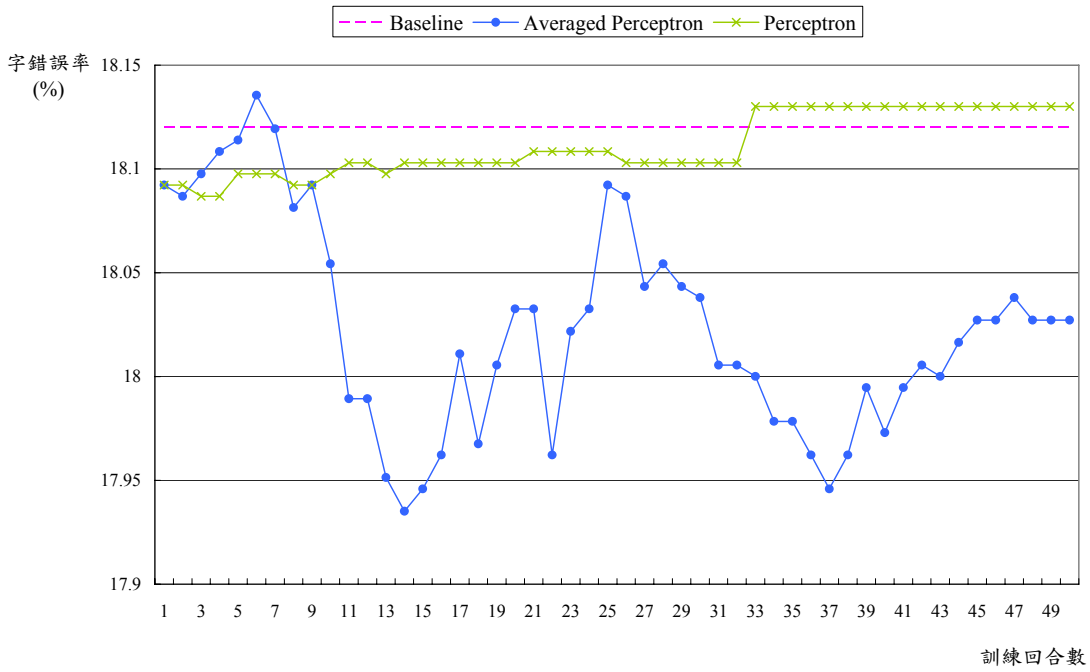


圖 5-5　Perceptron 演算法實驗結果

圖 5-6 顯示在 Averaged Perceptron 演算法實驗中訓練回合數與全域特徵權重非零特徵數之關係。在最初的 10 個訓練回合中，全域特徵權重非零之特徵數上升的幅度較大，在第 10 回合後逐漸減緩，至 25 回合後，其成長速度已近乎停滯。這應是因為學習步調之值在經過多回合調降後，已成為一個不足以控制特徵權重更新量的小數，因此，少有原本權重為 0 的新特徵可以有機會能夠更新其權重。

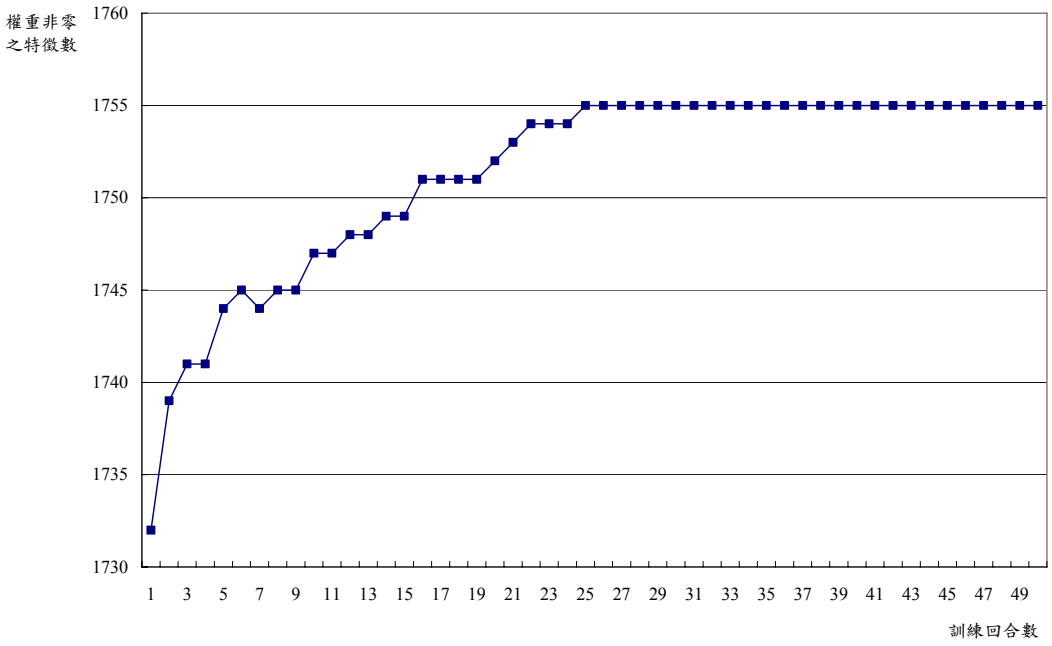


圖 5-6 Averaged Perceptron 演算法實驗中訓練回合數與權重非零特徵數之關係

5.2.3 鑑別式訓練與模型插補法實驗結果

如本文 2.2.2 節所述，最大化事後機率法(MAP)為一種語言模型調適方法。本節將比較最大化事後機率法中的模型插補法(Model Interpolation)、Boosting 演算法與 Perceptron 演算法這三種方法以同樣的調適語料與測試語料進行語言模型調適之結果。

模型插補法透過插補(Interpolation)方式法結合背景語言模型機率 P_B 與調適語言模型機率 P_A，其方程式為 $P(w_i \mid h_k) = \lambda P_B(w_i \mid h_k) + (1-\lambda)P_A(w_i \mid h_k)$，以插補結果所得機率作為調適後語言模型之機率。

以下將分兩部分討論鑑別式訓練與模型插補法之實驗：第一部份中，比較鑑別式訓練與模型插補法之效果(Effect)，在此實驗中，鑑別式訓練與模型插補法為競爭(Competition)關係；第二部分以模型插補法結果所得 M 個最佳辨識結果進一步作鑑別式訓練，觀察鑑別式訓練是否提供了對字錯誤率降低有益的額外資訊，在這部分實驗中，鑑別式訓練與模型插補法為合作(Cooperation)關係。

(一) 鑑別式訓練與模型插補法效果比較

表 5-5 中列舉以模型插補法作語言模型調適實驗數據，在這個實驗中，使用了完整詞圖中所有候選詞序列進行模型插補法調適。為使模型插補法與鑑別式訓練有較公平、一致的競爭，因此另作一個實驗，改以基礎辨識器提供之 M 個最佳辨識結果作模型插補法實驗，同於前述鑑別式訓練實驗，此處 M 亦為 100，數據顯示於表 5-6 中。

在完整詞圖的模型插補法調適結果中，在 $\lambda = 0.7$ 時可得最低字錯誤率為 17.23%，比 Boosting 演算法最低字錯誤率 17.92% 略低 0.69%，比 Averaged Perceptron 演算法最低字錯誤率 17.88%略低 0.65%；至於在 100 個最佳辨識結果

的模型插補法實驗中，在$\lambda = 0.7$時則可得最低字錯誤率為 17.59%，與鑑別式訓練之數據差距較小，其與 Boosting 演算法最低字錯誤率相差 0.33%，與 Averaged Perceptron 演算法最低字錯誤率相差 0.29%。

λ	字錯誤率(%)		λ	字錯誤率(%)
1	18.12		0.5	18.34
0.95	17.90		0.45	18.66
0.9	17.52		0.4	19.02
0.85	17.33		0.35	19.46
0.8	17.34		0.3	20.03
0.75	17.32		0.25	20.77
0.7	17.23		0.2	21.73
0.65	17.32		0.15	23.10
0.6	17.62		0.1	25.04
0.55	17.96		0.05	27.07

表 5-5 完整詞圖經模型插補法進行調適之實驗數據

λ	字錯誤率(%)		λ	字錯誤率(%)
1	18.12		0.5	17.69
0.95	18.03		0.45	17.72
0.9	17.75		0.4	17.76
0.85	17.60		0.35	17.75
0.8	17.70		0.3	17.76
0.75	17.66		0.25	17.75
0.7	17.59		0.2	17.76
0.65	17.61		0.15	17.78
0.6	17.62		0.1	17.79
0.55	17.68		0.05	17.78

表 5-6　100 個最佳辨識結果經模型插補法調適所得實驗數據

圖 5-7 顯示使用完整詞圖作模型插補法(ALL_WG_MAP)實驗之結果，以及只使用 100 個最佳辨識結果作模型插補法實驗之結果（Top100MAP）。實驗結果顯示，使用完整詞圖的模型插補法調適結果字錯誤率下降較多，而只使用 100 個最佳辨識結果作模型插補法調適之實驗結果字錯誤率下降幅度較小，且與鑑別式訓練之調適結果較接近。

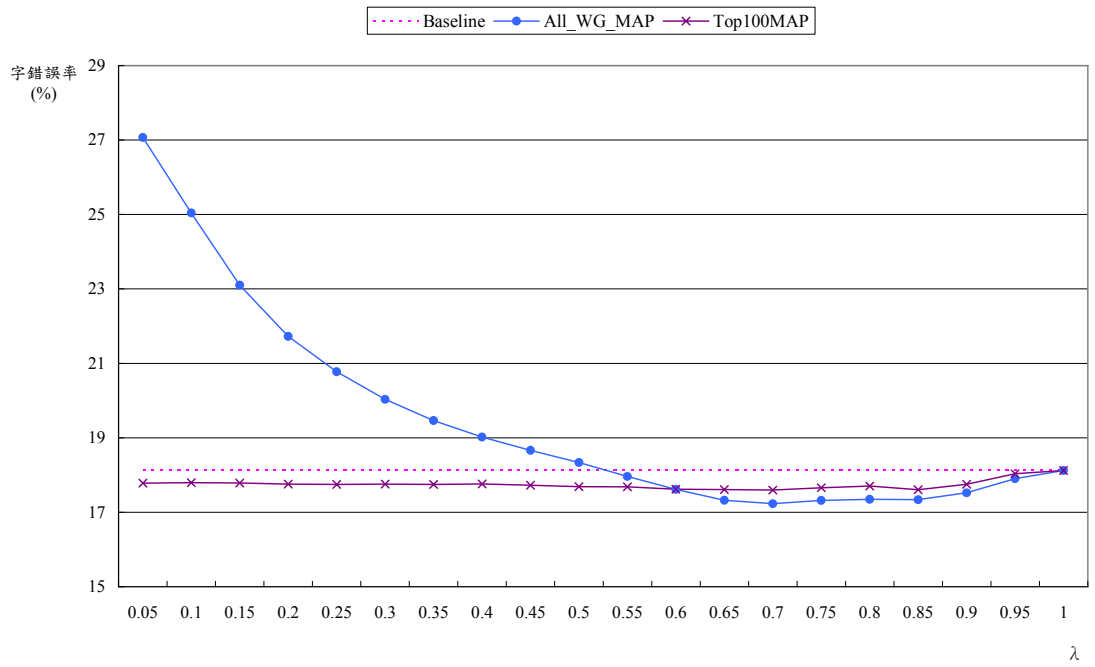


圖 5-7 模型插補法實驗結果

(二) 以模型插補法調適結果所得 M 個最佳辨識結果進一步作鑑別式訓練

為觀察鑑別式訓練是否提供了對字錯誤率降低有益的額外資訊，因此以模型插補法調適結果所得 M 個最佳辨識結果進一步作鑑別式訓練。如先前所述，以完整詞圖進行模型插補法調適，在 $\lambda = 0.7$ 時可得最低字錯誤率為 17.23%，本實驗以模型插補法於 $\lambda = 0.7$ 時調適所得 100 個最佳辨識結果，進一步作鑑別式訓練，觀察是否可以得到低於 17.23%之字錯誤率。實驗結果如圖 5-8 與圖 5-9 所示，不論以模型插補法調適結果所得 100 個最佳辨識結果進行 Boosting 演算法訓練或 Averaged Perceptron 演算法訓練，都可以進一步得到低於 17.23%之字錯誤率，顯示鑑別式訓練確實提供了足以使字錯誤率進一步降低的額外資訊。

表 5-7 列出以模型插補法調適所得 100 個最佳辨識結果進一步作 Boosting 演算法訓練所得實驗結果中較好的數據，圖 5-8 顯示其字錯誤率的變化，在第 496 個訓練回合可得最低辨識字錯誤率 17.08%。

訓練回合	字錯誤率(%)		訓練回合	字錯誤率(%)
496	17.08		562	17.10
497	17.10		563	17.10
498	17.10		564	17.10
499	17.10		565	17.10
500	17.10		566	17.10
501	17.10		567	17.10
502	17.10		568	17.10
503	17.10		569	17.10
507	17.11		570	17.10
508	17.11		571	17.10
561	17.10		572	17.10

表 5-7 以模型插補法調適所得 100 個最佳辨識結果進一步作 Boosting 演算法訓練所得部分較佳實驗結果

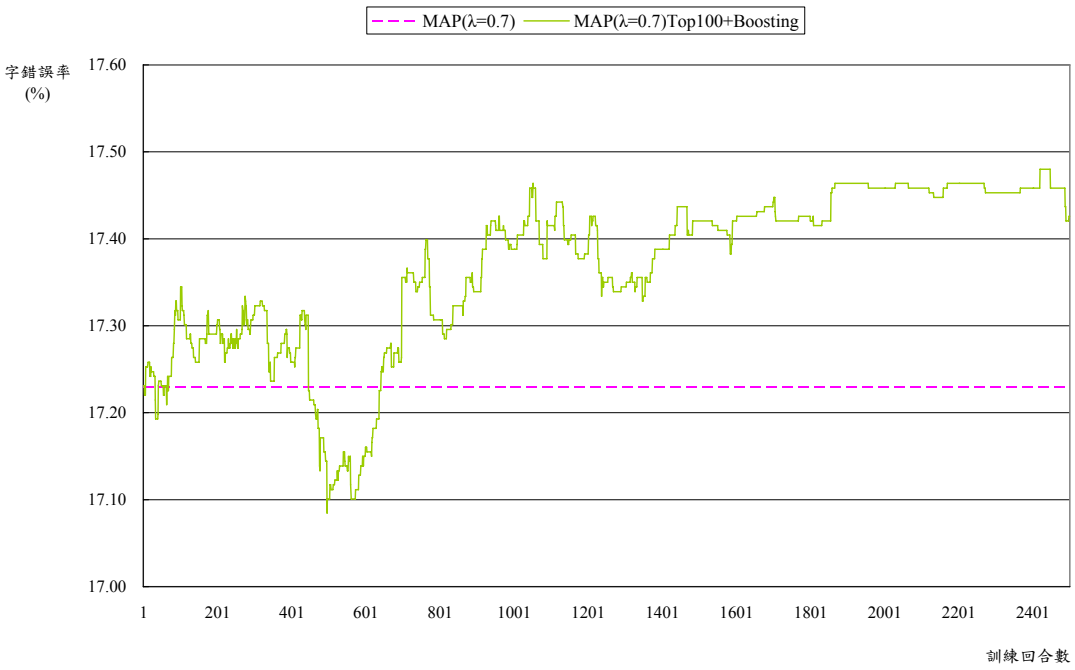


圖 5-8 以模型插補法調適所得 100 個最佳辨識結果進一步作 Boosting 演算法訓練所得實驗結果

表 5-8 中列出以模型插補法調適所得 100 個最佳辨識結果進一步作 Averaged Perceptron 演算法訓練所得實驗結果，在第 13 個訓練回合可得最低字錯誤率 17.08%，與前述以模型插補法調適所得 100 個最佳辨識結果進一步作 Boosting 演算法訓練實驗結果中所得最低辨識字錯誤率相同，只是 Averaged Perceptron 演算法以較快的速度獲得最低辨識字錯誤率。

訓練回合	字錯誤率(%)		訓練回合	字錯誤率(%)
1	17.23		26	17.22
2	17.23		27	17.24
3	17.20		28	17.25
4	17.18		29	17.24
5	17.18		30	17.25
6	17.18		31	17.30
7	17.17		32	17.30
8	17.18		33	17.31
9	17.16		34	17.36
10	17.14		35	17.38
11	17.12		36	17.37
12	17.10		37	17.40
13	17.08		38	17.43
14	17.11		39	17.46
15	17.14		40	17.42
16	17.15		41	17.43
17	17.15		42	17.45
18	17.12		43	17.47
19	17.11		44	17.47
20	17.10		45	17.50
21	17.12		46	17.50
22	17.12		47	17.52
23	17.13		48	17.55
24	17.14		49	17.55
25	17.17		50	17.55

表 5-8 以模型插補法調適所得 100 個最佳辨識結果進一步作 Averaged Perceptron 演算法訓練所得實驗結果

Averaged Perceptron 演算法實驗之所以能較 Boosting 演算法實驗用較快的速度獲得最低辨識字錯誤率，如前所述，這是由於 Perceptron 演算法從一開始即採用所有特徵之權重進行鑑別式訓練的計分，而 Boosting 演算法則是在每一訓練回合進行資料選取，只針對單一特徵更新特徵權重，因此在最初的訓練回合中，僅使用少數特徵進行鑑別式訓練的計分，造成字錯誤率的下降速度較 Perceptron 演算法來得慢。

以模型插補法調適所得 100 個最佳辨識結果進一步作鑑別式訓練的方法，不論是以 Boosting 演算法或 Averaged Perceptron 演算法訓練鑑別式語言模型的參數，都可以在辨識階段進一步得到比單獨使用模型插補法進行語言模型調適之辨識錯誤率更好的結果。這顯示結合模型插補法與鑑別式語言模型這兩種語言模型調適方法，可以得到更多的調適資訊，足以進一步降低辨識錯誤率。

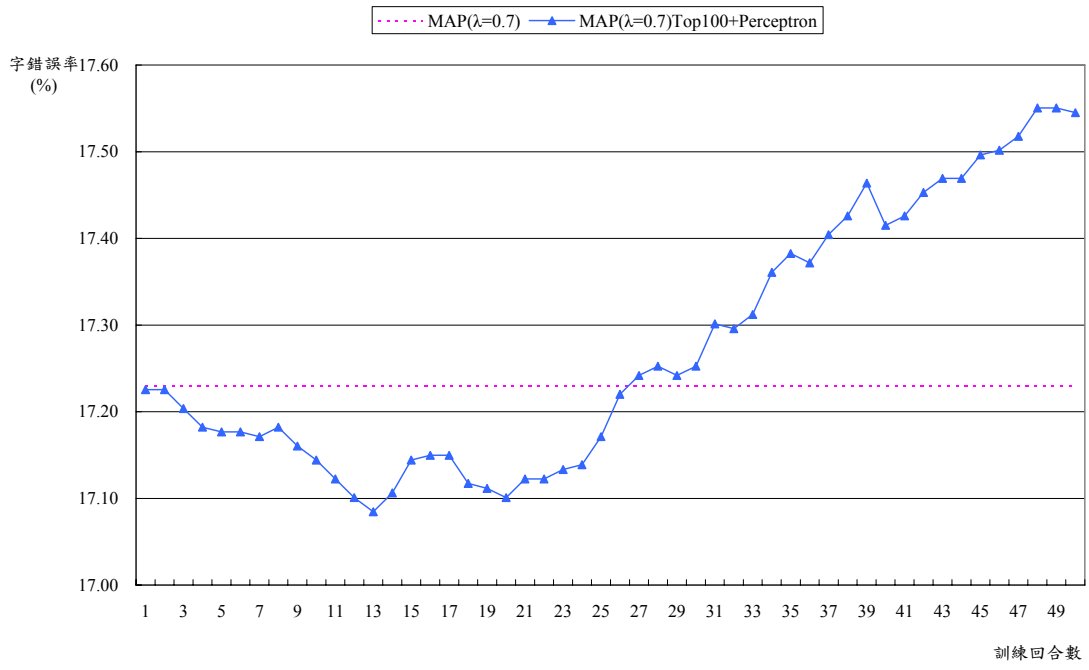


圖 5-9 以模型插補法調適所得 100 個最佳辨識結果進一步作 Averaged Perceptron 演算法訓練所得實驗結果

5.3 本文理論實驗結果

本文提出以關鍵詞自動擷取方法所得之長詞，作為鑑別式語言模型訓練之特徵。實作過程中，是以每一句訓練語料之字錯誤率最低的候選詞序列W_i^R連結成一個長字串，進行關鍵詞自動擷取，其中 i 值介於 1 至 L 之間，L 為訓練語料句數。預設閥值為 2，表示出現 2 次以上的詞彙，才會被擷取為關鍵詞。

本文實驗採取兩種關鍵詞擷取標準，一是按照前述關鍵詞自動擷取方法所得長詞(以 LongKeyword 表示)，另外我們嘗試將合併後出現次數超過預設閥值但未必合併至無法繼續合併的詞彙，亦視為關鍵詞(以 AllKeyword 表示)。在增加關鍵詞前，實驗中有 32627 個特徵，在 LongKeyword 實驗中，增加了 116 個特徵(共 32743 個特徵)。至於 AllKeyword 實驗中，則增加 369 個特徵(共 32996 個特徵)。

一_年_的
九_十_一
二_零_零
公視_新聞_中
公視_新聞_張
公視_新聞_陳
公視_新聞_陳_娟娟_陳_保羅
公視_新聞_綜合報導
兩性_工作_平等_法
所_作_的
個_水庫_的
這_一_次
陳_柏_宇
陳_柏_宇_採訪_報導

圖 5-10 透過關鍵詞自動擷取方法所得長詞(LongKeyword)範例

一_年_的
九_十_一
二_零_零
八_色_鳥
千_多_萬
千_兩_百
不_好_的
兩_百_多
工作_平等_法
兩性_工作_平等
兩性_工作_平等_法
所_作_的
的_採訪_報導
是_西方_採訪_報導
重要_棲息_環境
這_一_次
這_也_是

圖 5-11 未必是長詞的關鍵詞(AllKeyword)範例

圖 5-10 中列出透過關鍵詞自動擷取方法所得長詞(LongKeyword)部分結果，圖 5-11 中列舉的則是未必一定要是長詞的關鍵詞(AllKeyword)部分結果。

在擷取出的關鍵詞中，有多個關鍵詞是命名實體(Named Entity)，例如「八_色_鳥」、「兩性_工作_平等_法」…等，這些詞彙也許並未在詞典中被定義為一個詞，但仍能透過關鍵詞自動擷取系統擷取出來，表示這個方法確實可以針對文本擷取出新生詞彙，或是詞典中尚未定義的詞語。

不過，擷取結果中也有一些字串本身並不是很合理的詞彙，如「個_水庫_的」，依前述關鍵詞自動擷取方法，應以人工去除之，不過在本實驗中，並未在訓練階段與測試階段之間以人力介入處理。在未必是長詞的關鍵詞(AllKeyword)擷取結果中，由於其擷取關鍵詞的過程中並不以是完整長詞為目標，因此採用了未必合併至無法繼續合併的詞彙作為關鍵詞，使得這樣的情形較為明顯。

5.3.1 Boosting 演算法與關鍵字擷取

以下說明 Boosting 演算法增加關鍵詞特徵的實驗結果。表 5-9 中列舉出一些較好的數據，在第 1400 個訓練回合後，增加關鍵詞 (包括 LongKeyword 與

回合	Boosting (%)	LongKeyword (%)	AllKeyword (%)		回合	Boosting (%)	LongKeyword (%)	AllKeyword (%)
1410	18.13	17.98	17.99		1450	18.16	17.97	18.07
1411	18.13	17.98	17.99		1451	18.16	17.97	18.07
1412	18.13	17.98	17.99		1452	18.16	17.97	18.07
1413	18.13	17.98	17.99		1453	18.16	17.97	18.07
1414	18.13	17.98	17.99		1454	18.16	17.97	18.07
1415	18.13	17.98	17.99		1455	18.16	17.97	18.07
1416	18.13	17.99	17.99		1456	18.16	17.96	18.06
1417	18.12	18.00	17.99		1457	18.17	17.96	18.06
1418	18.12	18.00	17.99		1458	18.17	17.96	18.06
1419	18.12	18.00	17.99		1459	18.17	17.96	18.06
1435	18.14	17.97	18.07		1460	18.17	17.96	18.05
1436	18.14	17.97	18.07		1461	18.17	17.96	18.05
1437	18.14	17.97	18.07		1462	18.17	17.96	18.05
1438	18.16	17.97	18.07		1463	18.17	17.96	18.05
1439	18.16	17.97	18.07		1464	18.17	17.96	18.04
1440	18.16	17.97	18.07		1465	18.17	17.95	18.04
1441	18.16	17.97	18.07		1466	18.17	17.95	18.04
1442	18.16	17.97	18.07		1467	18.17	17.95	18.04
1443	18.16	17.97	18.07		1468	18.17	17.95	18.04
1444	18.16	17.97	18.07		1469	18.17	17.95	18.04
1445	18.16	17.97	18.07		1470	18.17	17.95	18.04
1446	18.16	17.97	18.07		1471	18.17	17.95	18.04
1447	18.16	17.97	18.07		1472	18.18	17.95	18.04
1448	18.16	17.97	18.07		1473	18.18	17.95	18.04
1449	18.16	17.97	18.07		1474	18.18	17.95	18.04

表 5-9　Boosting 演算法增加關鍵詞特徵實驗數據

AllKeyword) 的實驗結果都比單用 Boosting 演算法的實驗結果略佳。

圖 5-12 將單用 Boosting 演算法以及 Boosting 演算法增加關鍵詞特徵的方法之數據依回合數並列作觀察。在前 500 個訓練回合中，兩種方法所得字錯誤率不分軒輊，不過在約 500 回合以後，增加關鍵詞特徵的方法其實驗結果有比單用 Boosting 演算法效果來得好。

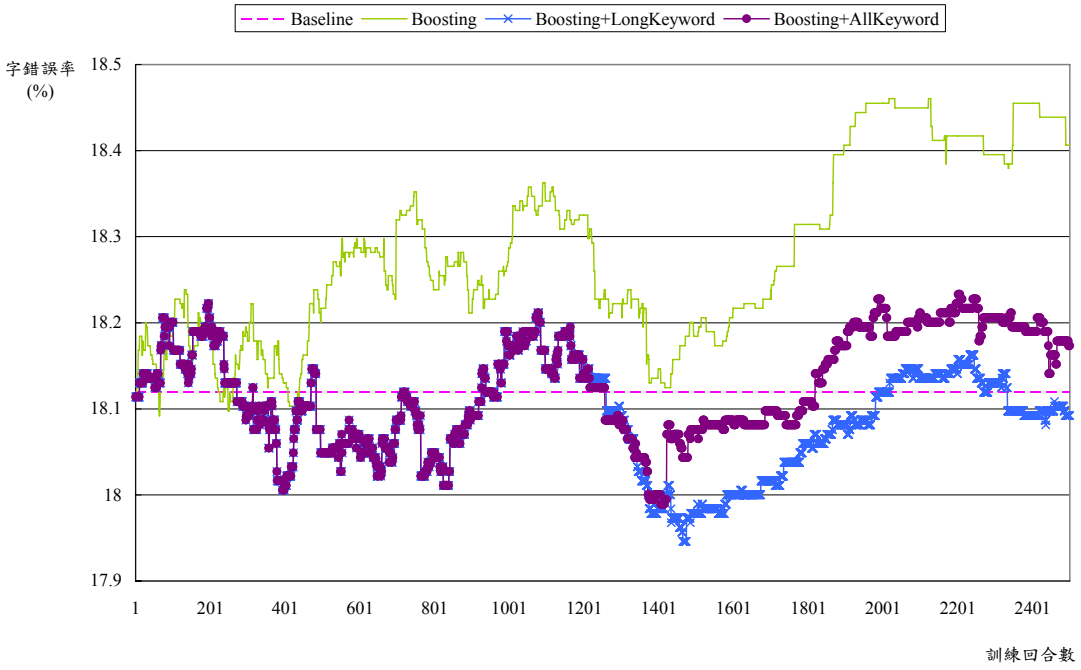


圖 5-12　Boosting 演算法增加關鍵詞特徵實驗結果

圖 5-13 為此實驗中訓練回合數與非零權重關鍵詞數的關係，顯示未必是長詞的關鍵詞(AllKeyword)比長詞(LongKeyword)容易受到特徵選取機制的青睞，這可能是因為未必是長詞的關鍵詞通常長度較短，而訓練語料中較容易存在多個這類關鍵詞。雖然未必是長詞的關鍵詞(AllKeyword)較受特徵選取機制青睞，但就前述圖 5-12 的實驗結果來看，未必是長詞的關鍵詞(AllKeyword)的實驗結果並沒有比長詞(LongKeyword)實驗結果來得好。

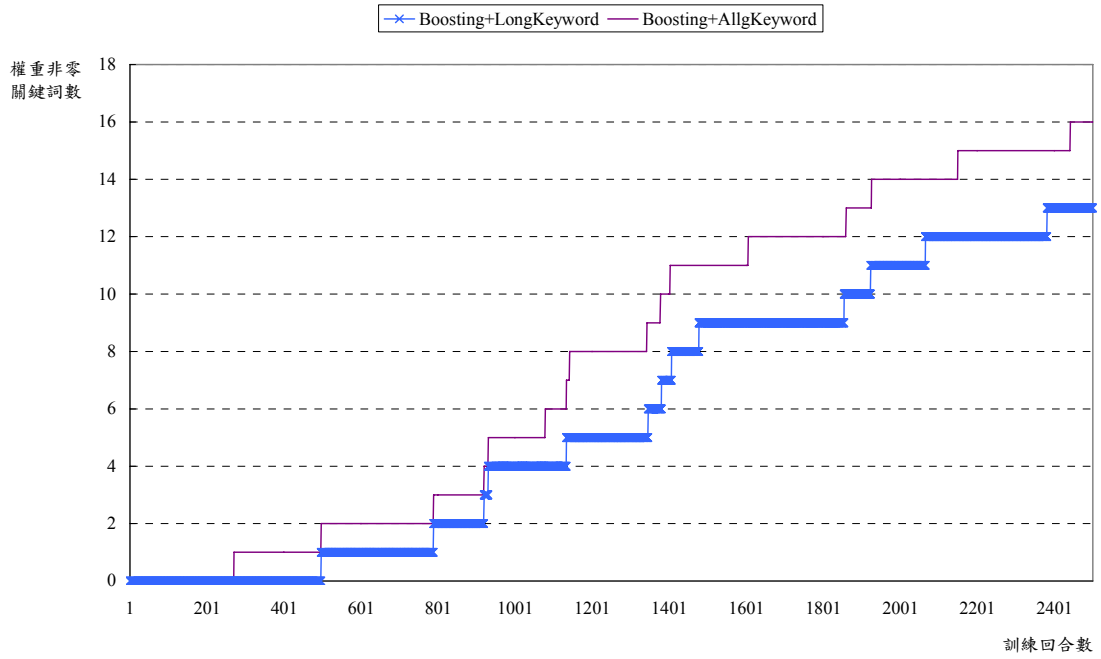


圖 5-13 Boosting 演算法增加關鍵詞特徵實驗中
訓練回合數與非零權重關鍵詞數之關係

5.3.2 Averaged Perceptron 演算法與關鍵字擷取

表 5-10 列舉 Averaged Perceptron 演算法增加關鍵詞特徵實驗前 50 個訓練回合的數據。最低辨識字錯誤率出現在增加未必為長詞的關鍵詞(AllKeyword)實驗中，在第 18 個訓練回合可得最低字錯誤率 17.92%。

回合	Perceptron (%)	LongKeyword (%)	AllKeyword (%)		回合	Perceptron (%)	LongKeyword (%)	AllKeyword (%)
1	18.09	18.09	18.09		26	18.09	18.09	18.07
2	18.09	18.09	18.09		27	18.04	18.05	18.04
3	18.10	18.10	18.10		28	18.05	18.04	18.03
4	18.11	18.11	18.11		29	18.04	18.04	18.03
5	18.11	18.11	18.11		30	18.04	18.02	18.01
6	18.14	18.14	18.14		31	18.01	17.99	17.98
7	18.12	18.12	18.12		32	18.01	17.99	17.98
8	18.08	18.08	18.08		33	18.00	18.00	17.98
9	18.09	18.09	18.09		34	17.98	17.98	17.96
10	18.05	18.05	18.05		35	17.98	17.97	17.96
11	17.99	17.99	17.99		36	17.96	17.96	17.96
12	17.99	17.99	17.99		37	17.95	17.95	17.94
13	17.95	17.95	17.95		38	17.96	17.95	17.94
14	17.94	17.94	17.94		39	17.99	17.97	17.96
15	17.95	17.95	17.94		40	17.97	17.97	17.95
16	17.96	17.97	17.96		41	17.99	17.99	17.98
17	18.01	17.97	17.96		42	18.01	18.01	18.00
18	17.97	17.95	17.92		43	18.00	18.00	17.99
19	18.01	17.97	17.95		44	18.02	18.01	18.01
20	18.03	18.02	18.01		45	18.03	18.01	18.01
21	18.03	18.03	17.99		46	18.03	18.02	18.01
22	17.96	17.99	17.97		47	18.04	18.03	18.02
23	18.02	18.04	18.03		48	18.03	18.03	18.01
24	18.03	18.02	18.01		49	18.03	18.03	18.03
25	18.09	18.09	18.08		50	18.03	18.03	18.03

表 5-10 Averaged Perceptron 演算法增加關鍵詞特徵實驗數據

圖 5-14 將單用 Averaged Perceptron 演算法以及 Averaged Perceptron 演算法增加關鍵詞特徵方法之數據依回合數並列作觀察。在第 13 個訓練回合後，增加關鍵詞特徵的方法無論是長詞或未必是長詞的關鍵詞，都較原本只用單連詞與雙連詞作特徵的 Averaged Perceptron 演算法實驗數據稍佳，這表示關鍵詞特徵於 Averaged Perceptron 演算法來說，可能對字錯誤率的降低產生一定的影響。

與前述 Boosting 演算法增加關鍵詞特徵實驗相反，在 Averaged Perceptron 演算法增加關鍵詞特徵實驗中，未必是長詞(AllKeyword)的關鍵詞實驗結果較長詞(LongKeyword) 實驗結果來得好。這也許是因為在本文實驗中，Boosting 演算法採用所有候選詞序列以更新特徵權重(圖 5-2)，而 Perceptron 演算法僅採用得分最高的一條候選詞序列以更新特徵權重(圖 3-4)，造成長詞(LongKeyword)特徵較不容易有機會更新其特徵權重，難以發揮效果。

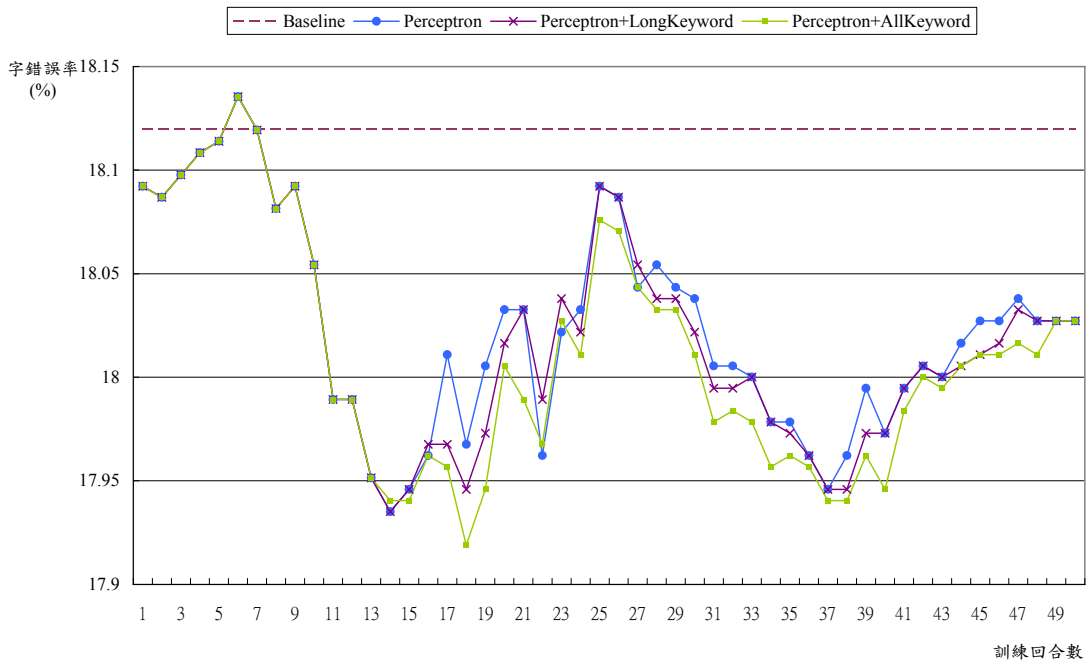


圖 5-14　Averaged Perceptron 演算法增加關鍵詞特徵實驗結果

第六章 結語

語言模型在語音辨識中扮演重要角色，它代表的是人類長久以來使用語言的規律性，用來判斷辨識器中哪一個詞序列較符合語言實際運用情形。然而，它可能面臨兩種問題。

其一，因時間或領域的差異造成這些訓練語料與測試目標的不一致，需透過語言模型調適以同時期或同領域之調適資料對語言模型進行調適；其二，現行語言模型為一個基於歷史資訊之模型，它根據歷史詞序列判斷一個詞的機率高低，若辨識歷史中有誤差，便會影響各詞序列之機率值，造成排序不盡然正確而影響辨識結果。

透過線性模型作鑑別式訓練以進行語言模型調適，可以同時針對上述兩方面作調整，一方面可以選擇合乎所需時間或領域之調適語料進行訓練，以改變辨識系統對詞序列的偏好高低，另一方面，鑑別式訓練根據調適語料中的正確參照轉寫，調整線性模型中的特徵權重，構成一個合乎調適語料辨識傾向的評分環境，在測試階段可對基於歷史資訊之模型產生的多個辨識結果進行重新排序，減少排序錯誤的發生。

首先，本文將透過鑑別式語言模型訓練方法進行的語言模型調適應用於中文大詞彙語音辨識，進行辨識結果的重新排序。

其次，將上述方法與模型插補法作互動：一則比較這兩種語言模型調適方法之效能高低，一則結合這兩種方法以期進一步降低辨識錯誤率。在比較效能的實驗中，模型插補法較鑑別式語言模型來得好；而在結合這兩種語言模型調適方法的實驗中，則可得到本文實驗中最低辨識字錯誤率 17.08%，相較於基礎辨識率(Baseline)有 5.74%的相對進步率，可見這兩種調適方法的結合對辨識錯誤率的下降具有加乘效果。

此外，本文中提出以關鍵詞自動擷取所得之關鍵詞作為鑑別式訓練的特徵。關鍵詞自動擷取系統可以在不需仰賴詞典的情況下，就文本本身內容特性，也就是語言使用習慣擷取出關鍵詞，因此即使是新生詞彙或是詞典中並未列舉之詞語，只要其出現次數超過預設閥值，便可以透過此系統擷取出來。

以關鍵詞自動擷取方法所得之關鍵詞，因直接透過文本的使用習慣篩選出來，應更能掌握調適語料之語言規律，以及詞典中並未列舉之詞彙，對實驗中字錯誤率之降低有所幫助。在 Boosting 演算法增加關鍵詞特徵實驗與 Averaged Perceptron 演算法增加關鍵詞特徵實驗中，增加關鍵詞作為特徵，都對辨識錯誤率的降低有所幫助。若能將其應用在句長較長的語料庫中，或存在多個新生詞彙的訓練環境下，也許會對辨識結果產生更大的助益。

參考文獻

[Aubert 2002] X. Aubert, “An Overview of Decoding Techniques for Large Vocabulary Continuous Speech Recognition,” Computer Speech and Language, Vol. 16, pp. 89-114, 2002.

[Bacchiani *et al*. 2003] M. Bacchiani and B. Roark.,” Unsupervised Language Model Adaptation”, ICASSP , 2003.

[Brown *et al.* 1992] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, JennifeC. Lai, and Robert L. Mercer. "Class-based *N*-gram Models of Natural Language", Computational Linguistics, 18(4):467–479, December, 1992.

[Chen *et al*. 2002] Z. Chen, K. F. Lee and M. J. Li, “Discriminative Training on Language Model”, ICSLP, 2002.

[Cherry *et al*. 2008.] C. Cherry and C. Quirk, “Discriminative, Syntactic Language Modeling through Latent SVMs”, ATMA, 2008.

[Collins *et al*. 2000] M. Collins, T. Koo, “Discriminative Reranking for Natural Language Parsing”, ICML, 2000.

[Collins 2002] M. Collins, “Discriminative Training Methods for Hidden Markov Models : Theorey and Experiments with Perceptron Algorithms”, EMNLP, 2002.

[Collins 2003] Machine Learning Approaches for Natural Language Processing , Lecture Slide 14, “Global Linear Models”,

http://www.ai.mit.edu/courses/6.891-nlp/l14.pdf

[Collins *et al*. 2005] M. Collins, B. Roark and M. Saraclar, "Discriminative Syntactic Language Modeling for Speech Recognition", ACL 2005.

[Dempster *et al.* 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", Journal of the Royal Statistical Society, Series B, Vol.39, no. 1, pages 1-38, 1977.

[Duda *et al.* 2001] R. O. Duda, P.E. Hart and D. G. Stork, "Pattern Classification", Wiley, New York, 2001.

[Freund *et al*. 1996] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm ", ICML 1996.

[Freund *et al*. 1998] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer, "An Efficient Boosting Algorithm for Combining Preferences", In Machine Learning: Proceedings of the Fifteenth International Conference, 1998.

[Friedman *et al.* 1998] J, Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting", Dept. of Statistics, Stanford University, Stanford, CA, 1998.

[Gao *et al* 2005a] J. Gao, H. Yu, W. Yuan and P. Xu, "Minimum Sample Risk Methods for Language Modeling," HLT/EMNLP, 2005.

[Gao *et al*. 2005b] J. Gao, H. Suzuki, W. Yuan, "An Empirical Study on Language Model Adaptation", ACM Transactions on Asian Language Information

Processing, Vol. 5, No. 3, September 2005, pp. 209-227

[Gao *et al*. 2005c] J. Gao, H. Suzuki, B. Yu, "Approximation Lasso Methods for Language Modeling", ACL, 2006.

[Gao *et al*. 2007] J. Gao, G. Andrew, M. Johnson and K. Toutanova ,"A Comparative Study of Parameter Estimation Methods for Statistical Natural Language Processing", ACL, 2007.

[Hebb 1949] D.O. Hebb, "The Organization of Behavior : A Neuropsychological Theorey", Wiley, 1949.

[Katz 1987] S. M. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of A Speech Recognizer. IEEE Trans. On Acoustics,Speech and Signal Processing, Volume 35 (3), pp. 400-401, March 1987.

[Kneser *et al.* 1995] R. Kneser and H. Ney, "Improved Backing-off for *M*-gram Language Modeling", ICASSP, 1995.

[Kuhn *et al.* 1990] R. Kuhn and R. De Mori., "A Cache-based Natural Language Model for Speech Reproduction", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990.

[Kuo *et al.* 2002] H.-K. J. Kuo, E. Fosler-Lussier, H. Jiang and C. H. Lee, "Discriminative Training of Language Models for Speech Recognition", ICASSP, 2002.

[Kuo *et al.* 2007] H.-K. J. Kuo, B. Kingsbury, G. Zweig, “Discriminative Training of Decoding Graphs for Large Vocabulary Continuous Speech Recognition”, ICASSP, 2007.

[Kuo *et al.* 2005] J. W. Kuo and B. Chen, “Minimum Word Error Based Discriminative Training of Language Models”, Eurospeech, 2005.

[Lin *et al.* 2005] S. S. Lin and F. Yvon, “Discriminative training of finite-state decoding graphs,” Proc. InterSpeech, 2005.

[Lippman 1987] R.P. Lippman, “An Introduction to Computing With Neural Nets”, IEEE ASSP Magazine, vol. 4, pp.4-22, 1987.

[McCulloch *et al.* 1943] W. S. McCulloch, and W. Pitts, “A logical calculus of the ideas immanent in nervous activity”, Bulletin of Mathematical Biophysics, Vol. 5, pp.115-133, 1943.

[Mitchell 1997] T. Mitchell, “Machine Learning”, New York, 1997

[Okanohara et al. 2007] D. Okanohara, J. Tsujii, “A Discriminative Language Model with Pseudo-Negative Samples”, ACL, 2007.

[Rigazio *et al.* 1998] L. Rigazio, J.-C. Junqua, M. Galler, “Multilevel Discriminative Training for Spelled Word Recognition”, ICASSP, 1998.

[Roark *et al.* 2004a] B. Roark, M. Saraclar and M. Collins, “Corrective Language Modeling for Large Vocabulary ASR with the Perceptron Algorithm”, ICASSP, 2004.

[Roark *et al.* 2004b] B. Roark, M. Saraclar, M. Collins, M. Johnson, “Discriminative Language Modeling with Conditional Random Fields and the Perceptron Algorithm”, ACL 2004.

[Roark *et al.* 2007] B. Roark, M. Saraclar and M. Collins,“Discriminative *N*-gram Language Modeling”, Computer Speech and Language, 2007.

[Tseng 1997] Y. H. Tseng, “Fast Keyword Extraction of Chinese Documents in a Web Environment”, International Workshop on Information Retrieval with Asian Languages , pp.81-87, 1997.

[Warnke et al. 1999] V. Warnke, S. Harbeck, E. Noth, H. Niemann and M. Levit., “Discriminative Estimation of Interpolation Parameters for Language Model Classifiers”, ICASSP, 1999.

[Woodland *et al.* 2000] P. C. Woodland and D Povey, “Large Scale Discriminative Training for Speech Recognition,” ASR-Speech Recognition: Challenges for the Millenium, pp. 7-16, 2000.

[Zhao *et al*. 2004] P. Zhao, B. Yu, “Boosted Lasso”, Tech Report, Statistic Department, U. C. Berkeley.

[Zhou *et al.* 2006] Z. Zhou, J. Gao, F. K. Soong and H. Meng, “A Comparative Study of Discriminative Methods for Reranking LVCSR *N*-Best Hypotheses in Domain Adaptation and Generalization”, ICASSP, 2006.

[Zhou *et al.* 2008] Z. Zhou and H. Meng, “Recasting the Discriminative *N*-gram Model as a Pseudo-conventional *N*-gram Model for LVCSR”, ICASSP ,2008.

[邱炫盛 2007] 邱炫盛，《利用主題與位置相關語言模型於中文連續語音辨識》，國立台灣師範大學資訊工程所碩士論文, 2007.

《語法與修辭》聯編組，《語法與修辭》，新學識文教出版中心，台北，1998。