國立臺灣大學電機資訊學院電機工程學研究所

碩士論文

Graduate Institute of Electrical Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

具備語音功能的雲端應用: 個人化語言模型與互動式語音文件檢索

Voice Access of Cloud Applications : Language Model Personalization and Interactive Spoken Content Retrieval

溫宗憲
Tsung-Hsien Wen

指導教授：李琳山博士
Advisor: Lin-shan Lee, Ph.D.

中華民國 102 年 7 月
July, 2013

國立臺灣大學碩士學位論文
口試委員會審定書

具備語音功能的雲端應用: 個人化語言模型與互動式語音文件檢索

Voice Access of Cloud Applications : Language Model Personalization and Interactive Spoken Content Retrieval

本論文係溫宗憲君 (R00921033) 在國立臺灣大學電機工程學系、所完成之碩士學位論文 , 於民國 102 年 6 月 24 日承下列考試委員審查通過及口試及格 , 特此證明

口試委員：

李琳山 (簽名)

(指導教授)

王小川　　鄭秋豫

陳信宏

李仁貴

系主任、所長　　(簽名)

致謝

兩年來的碩士生涯，終於跟著這本碩士論文的完成，而到了終點。或許這樣說來有些浮誇，但確實這兩年的碩士歲月對我而言，不管是在學術研究、人生歷練、思想塑造等等，都有舉足輕重的影響。碩士論文只是一篇對這段研究生活的總結，但太多無法放進去這短短十來頁文章內容的，是許多人的陪伴、指導、切磋、與關懷。而這些生活的一切一切，對我來說才是無價珍寶。

首先，謝謝我的指導教授，李琳山老師，提供了一個很好的研究環境，讓我能夠在充滿了內部討論及外部交流的實驗室裡發展自己的可能性。從老師身上學到的不只是學術技能，更是做學問的嚴謹態度、教學的熱忱、以及對於凡事兢兢業業的行事風格。

感謝我的爸媽，願意理解並接受我為了追求自己的夢想而常常四外奔波。雖然說，許多專業術語以及研究內容看似築起了一道溝通的高牆，但心裡對待彼此的心意卻是不言自明的。真的非常感謝你們，一路來的支持與陪伴。

感謝實驗室的各位同學們，一路來的互相扶持與切磋琢磨。特別要感謝的是宏毅哥 (你好你好)，帶我入門做研究、指導我寫人生第一份論文、及出國一起開會的互相照應。可以說有宏毅哥在前方幫我領路，才讓我這條研究之路一路走來非常順遂。也要謝謝實驗室的死黨們，培豪、庭曜、博智、宥宇，一起修課、一起作實驗、一起吃飯、然後一起沒梗。這一段日子有大家陪伴很好，雖然從此之後大家各奔西東，但也希望能持續聯絡，做一輩子好友。另外還要感謝許多人，馬雅、Aaron、小安、青峰、Popo、威爺、向姐、魯蛇、道哥・・・等等，無法盡數。要走的祝福大家都有個美好的前程，會留下的則希望能持續開發這個知識寶庫，讓台大語音實驗室的名聲能越打越響。

另外要感謝在 StorySense 一起工作的大家，在這段期間提供我一個亦師亦友亦家的陪伴。感謝 Edward 找我進來，雖然一開始的 office 空空蕩蕩，但心在一路上 always 是滿的：在產品實作面上不斷挑戰我的研究所學、在留學創業經驗上的分享、以及在日常生活的放鬆 hang out。謝謝鼎翔、挺正、松松，咱們五人一起走到現在人丁興旺的狀態。也感謝後來加入的大家，Steven、Jason、湘庭、圈圈、玫樺、芳瑜・・・等等，不可勝數。只是想讓你們知道，平常調皮的 Shawn，真

的很喜歡你們。

最後要謝的，是各個海內海外的摯友們。感謝 Rick、Hans、竟彰、彥昇、志軒、冠彰、文翔、Chimat···等，在我這一年來去美國的種種照顧，希望可以很快再跟大家相見。還有家安、哲銓、昕芳、古竹、怡安···等，同樣難以窮舉。不管大家選擇是什麼，未來在何方，相信大家都可以走出一條很不一樣的路。

一本碩士論文，代表著一個人生階段的結束，不免帶著一些揮別的傷感。慶幸的是，科技發達的現在，魚雁往返只是彈指之間的事情。願大家在畢業之後仍能保持聯絡，並且持續用紅色炸彈與人生捷報塞爆我的信箱。最後，再讓我誠摯地說一次，謝謝你們。

中文摘要

本論文探討具備語音功能的兩項雲端應用技術：個人化的語言模型及互動式語音文件檢索。

在語音辨識中，模型的不匹配對辨識率一向有很大的損害。由於個人化手機普及，個人化辨識系統成為可行，而考量到每個個人語言使用習慣的差異，語言模型的個人化有其必要。過去個人語料庫建立不易，但如今，越來越多人習慣性地在社群網站上留下大量的文章與留言，故個人化語料庫較以前容易取得許多。但資料稀疏的問題仍然不易解決。在本論文中，我們提出以各種方式估計社群網站上不同使用者間的用語相似度，並據以加入不同使用者的語料庫來幫助估計更強健的個人化語言模型。我們並比較了用 N 連文法語言模型及遞迴式類神經網路語言模型來實做的效能表現，並驗證了新提出的方法確實提升了對個人語言的預測能力。

在第二部分裡，我們探討互動式語音文件檢索。由於語音文件很難呈現且瀏覽耗時，而過差的辨識率更可能使檢索結果不如人意，因此藉由與使用者互動使系統對使用者想找的資訊有更多瞭解，是一個有效改善此問題的方法。在本論文中，我們用馬可夫決策模型 (Markov Decision Process, MDP) 來模擬互動式檢索的問題，並採用強化學習 (Reinforcement Learning) 演算法學習出最佳系統決策，亦採用不同的檢索模型來實作檢索系統。實驗顯示，我們提出的方法確實能夠輔助檢索進行，幫助使用者更有效的找到所要找的資訊。

Abstract

This thesis considers voice access of cloud applications with two parts: (1) Personalized Language Model and (2) Interactive spoken document retrieval.

Model mismatch has been a major problem in speech recognition. With hand-held devices widely used today, personalized models become possible. A huge quantities of posts and comments with known owners emerged on social network websites, personal corpora become practically available but with data sparseness problem unsolved. In the first part of this thesis, we proposed personalized language modeling approaches by estimating the language similarities between different social network users and integrating the corresponding personal corpora accordingly. We studied both N-gram language models as well as recurrent neural network language models, and the experimental results support the concept.

In the second part of this thesis, we studied interactive spoken document retrieval. Interactive retrieval is helpful to spoken content retrieval because retrieved spoken items are difficult to be shown on screen and browsed by the user, in addition to the speech recognition uncertainty. We model the interaction process by a Markov Decision Process and train the policy with Reinforcement Learning. Experimental results demonstrate the retrieval performance can be improved with the interactions.

Contents

List of Figures

List of Tables

Chapter 1

導論

1.1 具備語音界面的雲端應用平台

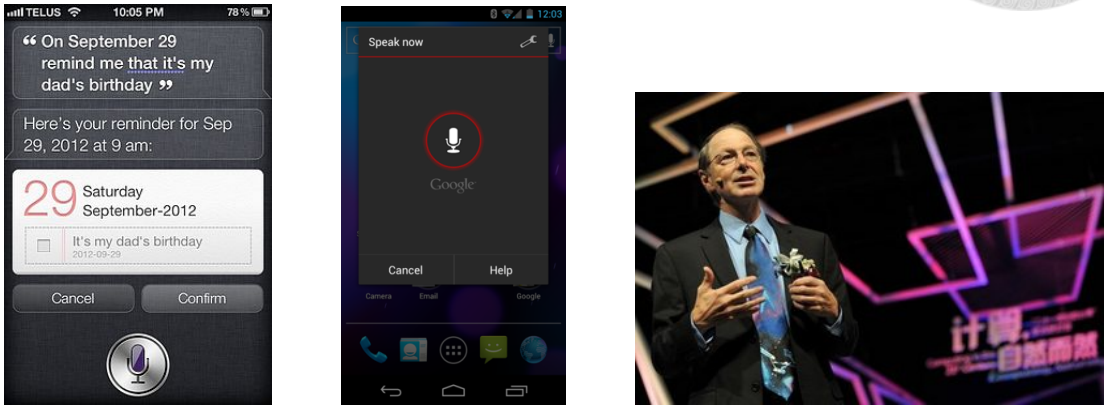
在科技越來越發達的今日，人們的生活漸漸開始產生許多的改變，如：人們普遍接受的媒體已經漸漸被網際網路所取代，傳統的媒體如電視、廣播等等其重要性地位已經不如以往；大量的多媒體數位內容在網路上廣泛傳播，資訊影音化勢不可擋；智慧型手機的發展帶動了手機應用程式的興起，在許多平台例如蘋果應用程式商城 (App Store) 與安卓商城 (Android Market) 蔚為一股風潮。在這樣的大環境裡，語音技術 (speech technology) 的應用越趨成熟，以下為三個近年來最為明顯的例子 (圖 1.1)：

- Apple 個人語音助理 (Siri)：蘋果公司在 2011 年發佈的 iPhone 4S 智慧型手機結合了語音助理 Siri，可以利用語音指令 (voice command) 作簡單問答並幫忙安排個人行程與發送簡訊等功能。被認為是近年來語音與人工智慧技術結合的商業產品新典範。

- Google 語音搜尋 (Google Voice Search)：科技巨人 Google 長年以來都投入了許多人力致力於發展語音技術的研究，也是最早將語音辨識應用在搜尋技術的公司之一。尤以其搜尋關鍵字辨識的準確率十分可靠，支援的語言數目也相當多。在許多現行的語音產品中，Google 語音服務無疑仍是最可靠的。

- Microsoft 即時口譯：2012 年 11 月微軟展示了最新即時口譯 (realtime translation) 系統：該系統能在近乎接近及時的時間裡，將一個人所說的語音由英文

轉譯至中文。這項結合了語音辨識 (speech recognition)、機器翻譯 (machine translation)、及語音合成 (speech synthesis) 的技術，在語音科技的研究領域俱有劃時代的意義。



(a) Siri 語音助理　(b) Google 語音搜尋　(c) Microsoft 即時口譯

Figure 1.1: 語音科技的現代例子

我們仔細審視這波語音科技的發展浪潮，歸結出以下三項推動的主要因子：(1) 智慧型手機 (smartphone) 的快速普及，由於有限的手機螢幕大小，使得人機界面的設計變成一個很大的挑戰。儘管觸控面板相當程度地解決了這個問題，但許多的使用者仍然期待著更自然的互動界面 - 語音界面的整合 [1]。(2) 語音科技本身技術的成熟也是一個很大的關鍵：最基礎的語音辨識部分由於深度學習 (Deep Learning) [2--4] 的發展在辨識率上有了顯著的突破；而語音技術的應用上諸如語音文件搜尋 (Spoken Document Retrieval) [5--7]、對話系統 (Dialogue System) [8--10]、語音摘要 (Spoken Summarization) [11, 12] 等等的發展，也使得我們越來越能對語音內容甚或多媒體內容進行分析。(3) 雲端運算與硬體的進步，使得更大量的資料可以被儲存與處理，這不但使得原先複雜度過高的應用可以做到及時，也使得更複雜、更強健的模型可以被用來對原本差強人意的正確率作改量。在這樣的背景下，可以預料的是在不久的將來，語音相關的應用程式將會如雨後春筍般不斷地出現。但由於語音科技本身是一個高度依賴訓練語料與模型計算的技術，倘若未經適當調整，對於不同使用者的表現會有極顯著的不同。即使是同一個使用者，隨著時間演進，新的內容、新的資訊、甚或新的使用習慣也可能使原來的模型過時 (out-of-date)。因此，若不妥善設計，任何的現行技術在不久的未來都可能不再適用。

考量到以上多個面向，本篇論文旨在設計一個具備語音界面的雲端應用平台：使用者透過手機上的語音界面向此雲端應用平台要求服務，服務可以是搜尋檢索 (retrieval)、影音串流 (video streaming)、甚或是網路社交 (social networking) 等等。而這個雲端平台除了提供使用者相對應的服務之外，亦必須從跟使用者的互動過程中，搜集分析使用者所產生的資料，並從這些採集到的資料中，學習整個使用的趨勢甚或是對特定使用者個人化 (personalization) 以期提升其系統表現或使用者的滿意程度。這樣一個動態演化 (dynamic evolving) 的系統就像是一個自給自足的生態系 (ecosystem)，我們可以簡易的以圖 1.2來表達這樣的概念。 在本論文中，

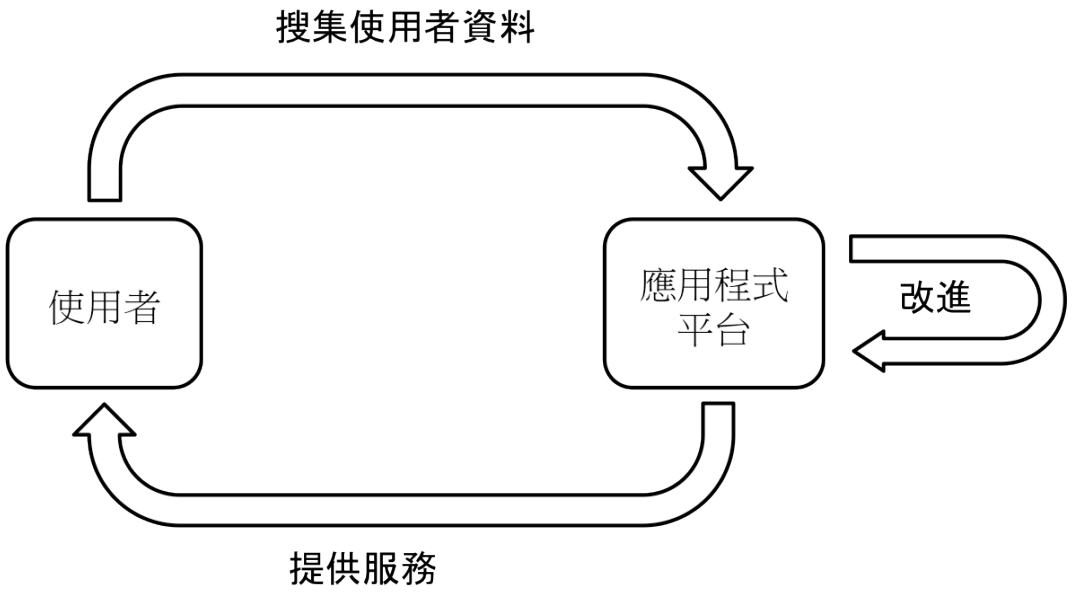


Figure 1.2: 動態演化的系統生態系概念圖

我們將分別從語音基礎技術及應用兩個層面切入，探討具備語音界面的雲端應用平台所需的兩項技術：個人化辨識系統及互動式搜尋系統。

1.1.1 個人化語音辨識系統

語音辨識系統 (speech recognition system) 可以說是所有語音技術的根本，降低辨識錯誤率一直是這個部分研究的主要重點。儘管某些現行的辨識系統在乾淨的聲音環境 (clean) 可以達到極準確的辨識率，但許多都是在高度控制的實驗環境下的結果，很多因子一旦改變辨識錯誤便會非常急速地增加。例如：背景雜訊的類型大小、不同麥克風收音的效果、離麥克風遠近、說話的快慢、聲調、大小聲、發音方式、不同使用者的聲音不同、所說語言是否為母語 (native speaker)、使用

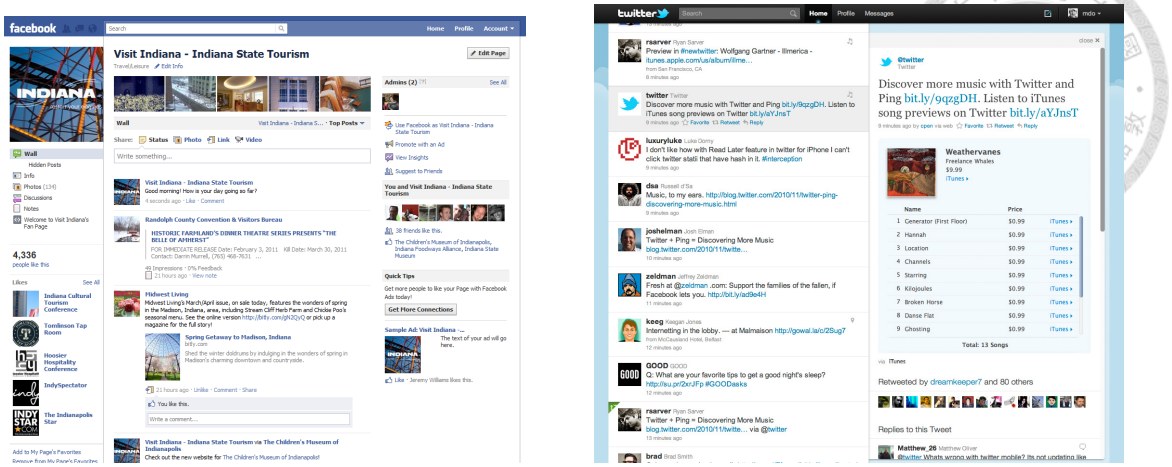
語言的方式、與用的文字···等等都有關係。如果我們細看，其實可以把上述的因子，大致歸類為三個方向：(1) 雜訊及外部環境干擾 (2) 語者聲音特徵不匹配 (3) 語者語言特徵不匹配。其中 (2)(3) 便是本文的個人化語音辨識系統部分希望探討的主題。

語音辨識系統依賴聲學模型 (Acoustic Model) 及語言模型 (Language Model) 兩者分別模擬語者的聲音特性與語言特性。聲學模型的個人化在傳統上被稱作為聲學模型調適 (Acoustic Model Adaptation) 或者語者調適 (Speaker Adaptation)，利用使用者使用後產生的聲音訊號微調背景聲學模型的參數，使其儘量能夠切合該語者的發音形式。相對上，語言模型在過去尚未有關於個人化的研究，主要原因在於欠缺大量的個人化語料庫 (corpus) 可以提供模型學習。因此許多研究學者退而求其次，將許多由不同個人產生出來的小語料庫按照領域分群，做領域導向 (domain-oriented) 的語言模型調適 (Language Model Adaptation) [13, 14]。這樣子的語言模型調適在巨觀上雖然模擬了不同的領域不同的字詞特性而在辨識上獲得相當大的成就，但在微觀上卻無法對每個個人的用詞特徵做精細地調適。

慶幸的是，這個缺乏個人語料庫的難題在社群網路風行的今天被打破。在社群網路上的每個人都會註冊一組帳號密碼，並且利用這個帳戶身份與朋友們在網路上進行社交活動，例如邀約、聊天、分享新聞或抒發己見等，如圖 1.3。在這些過程中，使用者便會不斷的留下新的語料在這些社群網路上，這些語料在相當程度上，就可以代表這個使用者所使用的語言特性。雖然個人語料庫可能不大，但社交網路提供的不止個人語料，不同族群中的共享語言特徵也可以被捕捉 [15]。

1.1.2 互動式搜尋系統

在這個部分我們想要深入探討的，是結合資訊檢索 (Information Retrieval) 及對話系統 (Dialogue System) 兩項技術的一個新的應用，我們稱之為互動式搜尋系統 (Interactive Retrieval System)。為何需要建立互動式的資訊檢索系統？主要原因在考量我們後端需要被檢索的內容是語音內容 (spoken content) 的時候。語音內容，如上課錄音、電視新聞、或者演講內容等等，我們往往很難對其做檢索，其主要原因在於：(1) 語音內容由於是訊號內容，它很難顯示在螢幕上給使用者看。儘管秀出了排序的清單 (ranking list)，使用者仍然必須一筆一筆點進去聽才有辦法判斷



(a) 臉書 (Facebook)　　(b) 推特 (Twitter)

Figure 1.3: 兩大社群網站：臉書與推特

這筆內容究竟是否是他想要的。因此它很難瀏覽也很難選擇。(2) 雖然這些語音內容有少數有專人幫忙轉譯成文字 (如 TED talk)，但大部份的內容是不會有的，尤其是在數位內容產生地如此快速的大環境之下。因此，儘管電腦自動轉譯仍包含許多錯誤，但似乎仍是個比較可行的做法。但這樣一來，錯誤的辨識結果將對搜尋技術造成另一層挑戰，使其更難做到精確的字詞比對。(3) 為了克服語音辨識錯誤造成的檢索上的問題，語音檢索大量仰賴次詞單元 (subword unit) 才輔助檢索，但如此一來雖然增進了召回率 (recall)，卻造成了極低的精確率 (precision)，使用者因此更難在前幾個排序結果找到其想要的資訊。(4) 即使在文字檢索的案例上，當使用者輸入過度模糊的關鍵字時，系統仍然必須仰賴互動來獲得更多使用者真正想要的資訊。雖然有許多現行的研究試著解決這個問題 [16--18]，但所有的實驗還是顯示辨識錯誤仍不可避免的影響到系統效能。而這層影響，將會導致回傳給使用者的排序清單效果較差，使用者就必須花更多的時間去聽更多的內容才能找到他所想要的內容。

考量到這些層面，我們希望設計一個系統可以根據當下的狀態做出判斷：倘若判斷第一遍搜尋結果 (first-pass) 太差或者查詢指令 (query) 太過模糊，系統會自動跟使用者要求更多的資訊；反之，倘若系統判斷結果已經夠好了則會將結果回傳給使用者 [19, 20]。一個可能的使用情境如圖 1.4所示。

這樣的系統學著在效能與使用者多餘的工作間做取捨，希望以最高的效率讓使用者找到他想要的內容。

U1: US President, please.
S1: Your query is ambiguous. Anything more?
U2: Diplomatic issue.
S2: Persian Gulf?
U3: No.
S3: Please view this list and select one item relevant to your need.
U4: (Pick the doc at rank 6th, "Obama: We Welcome China's Rise, January 19, 2011 3:47 PM, CBSNEWS")
S4: (Show the list) Here is the result.

Figure 1.4: 互動式搜尋系統可能的使用情境

1.2 章節安排

本論文再來的各個章節安排如下：

1. 第二章主要會根據我們想要討論的兩個層面，個人化語音辨識系統及互動式搜尋系統，做現有技術的回顧：包含聲學模型、各種語言模型、模型調適、檢索模型、對話管理員、及馬可夫決策模型等等。

2. 第三章至第五章屬於第一部分 (Part I)，主要探討個人化語音辨識系統：第三章主要解釋群眾外包 (crowdsourcing) 的概念及如何利用這樣的概念來設計前述提到的自給自足的系統生態系，並以結合臉書 (facebook) 作為一個例子。第四章探討語言模型的個人化，包含 N 連文法語言模型 (N-gram Language Model) 及遞迴式類神經網路語言模型 (Recurrent Neural Network Language Model)，並比較兩者的優劣。第五章談如何將這些個人化語言模型與現有的個人化聲學模型作整合，並建構一個完整的個人化辨識系統。

3. 第六章與第七章則是第二部分 (Part II)，探討互動式搜尋系統這項應用：第六章首先探討系統組成，並進入搜尋系統本身，探討在不同的檢索模型如何做相關性回饋 (Relevance Feedback)。第七章進入對話管理員本身，主要是探討馬可夫決策模型，在不同的狀態空間設定下該如何模擬，並給出實驗結果並比較不同狀態設定對結果的影響。

4. 第八章則是對整篇論文的總結，並給出在這個架構下未來可以繼續研究發展的幾個方向。

Chapter 2

基礎背景簡介

2.1 個人化語音辨識系統

一個語音辨識系統，主要是三個模型：聲學模型 (Acoustic Model)、語言模型 (Language Model)、以及詞典模型 (Lexicon) 的互相結合輔助。在給定聲音訊號後，套用搜尋演算法 (Search Algorithm) 在其所有可能的候選字串建成的網路中搜索機率最高的一條路徑 (如圖 2.1所示)。以下將簡單介紹一下建構一個語音辨識系統的架構，並且點出哪些是主要我們強調可以個人化的部分，並大略了解在本文這一部份我們想做的事及採用的實作方法。

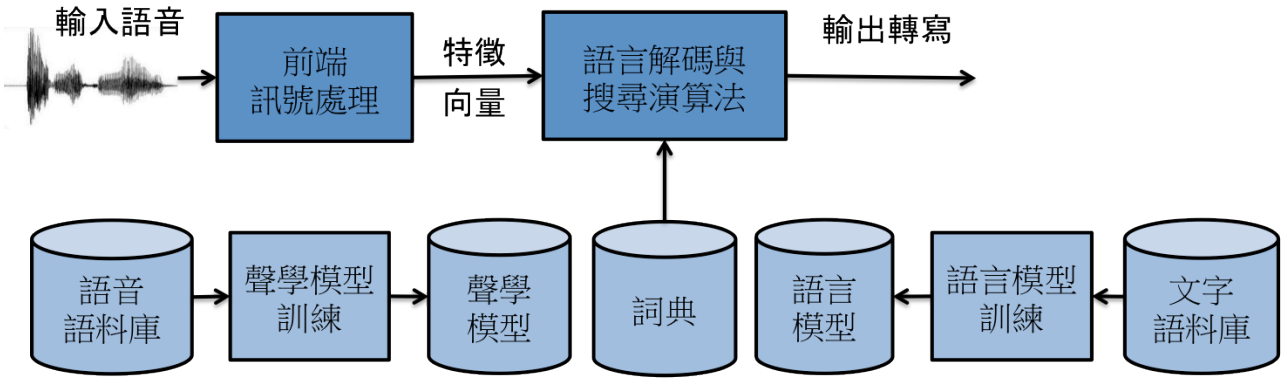


Figure 2.1: 大字彙語音辨識系統.

2.1.1 聲學模型與調適

聲學模型主要基於圖像辨識技術 (Pattern Recognition) 這門學問。圖樣辨識為一門建立模型來分辨類別的學問，將一個實際的分類問題分成三個步驟：分段 (Segmentation)、抽取特徵 (Feature Extraction)、分類 (Classification)。在大字彙連續語音辨識當中，圖樣辨識技術提供的是：給定一段時間的語音訊號，計算音素模型的機率，進而以最小風險 (Mimimum Risk) 的方式判定為哪一個音素模型。在分段這一步驟中可利用無語音部分的偵測，以便之後的處理。抽取特徵方面，目的是要將訊號轉為特徵向量，成為分類器 (Classifier) 數學模型的輸入。抽取特徵有著許多不同的方法，一般採用梅爾倒頻譜係數 (MFCC：Mel-frequency Cepstral Coefficient) 的方法。分類器則是圖樣辨識的核心部分，應用在語音上，較常用的模型為隱藏式馬可夫模型 (HMM：Hidden Markov Model)，因其利用機率來表示狀態 (State) 之間的轉移，以及觀測向量的不確定性，很適合用來表示語音可長可短，且同一音素的訊號不會都一樣的問題。

語者調適技術的目的在於利用語者所提供的有限語料，來改善辨識系統對於個別使用者聲音的辨識能力。當系統要進行語者調適工作時，須先獲得語者提供的語料，此稱為調適語料 (adaptation data)。從語料的角度來看，如果系統事先得知語料的內容，也就是清楚知道語料每一句的轉寫 (transcription)，系統及可以找出語音訊號和相對應的語音參數做精確的調整，這稱為監督式調適法 (supervised adaptation)；相反的，若不知道語料的內容，須先對語料辨認過後，才將辨認結果當作語料的內容來調適，則稱為非監督式調適法 (unsupervised adaptation)。就以個人化聲學模型調試的角度來看，不需要人工轉寫的非監督式調試法是較為實際的做法，因此在我們系統內被採用並實作。在許多的語者調適法當中，最大相似度線性迴歸 (MLLR：Maximum Likelihood Linear Regression) [21--23] 的方法被廣泛的應用在快速語者調適，例如它只需要些許的語料便可以對模型參數做調適。在最大相似度線性迴歸中，非語者獨立的模型參數可以根據一個或多個仿射轉換方式 (affine transformations) 來達到調適的目的，該轉換則是藉由最大相似度的估計準則求得。因此考慮系統效率，我們採用最大相似度線性迴歸作為我們聲學模型個人化的主要手段。

2.1.2 N 連文法語言模型 (N-gram)

語言模型主要目的在於模擬句子的組成，在語音應用上主要在於給定解碼過後的音素序列，利用語言模型進一步解碼成字串序列。雖然在自然語言的領域中，有文法模型等等較複雜的模型來模擬句子構成。但在語音應用上，較常用 N 連文法語言模型 (N-gram Language Model) [**?**, 13, 24] 來模擬句子的機率。N 連文法語言模型運用了 N-1 馬可夫假設 (N-1 Order Markovian Assumption)，將超過 N-1 之外的詞都假設是沒有影響的。套用這個假設，我們可以將句子的機率展開：

$$P(w_1^N) = \prod_{q=1}^{N} P(w_q|w_1^{q-1}) = \prod_{q=1}^{N} P(w_q|w_{q-N+1}^{q-1}) \tag{2.1}$$

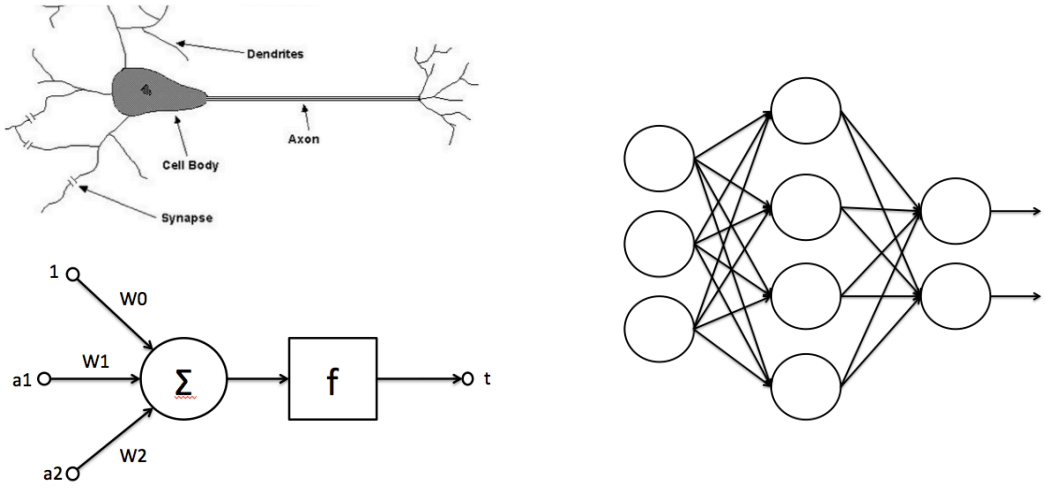
從 (2.1) 式中先用條件機率展開，代表現在這個詞出現的機率，會受到詞的歷史訊息影響有不一樣的分佈。之後應用了 N-1 的馬可夫假設，相關性就只有到前 N-1 個詞，因此機率的條件部分就只剩下了前 N-1 個詞的部分。N 連文法語言模型的好處是只要留有前 N-1 個詞的歷史紀錄就好，在這之前的詞都假設不會影響到接下來詞的機率分佈。而在文法模型中，要形成文法樹，最後一個詞如果跟第一個詞是在同一層，則要將它們都留著紀錄；並且，N 連文法語言模型可以從一個語料中計算得到，文法則需要一定程度的人工修訂。因此，在時間、記憶體都很要求的大字彙連續語音辨識系統中，N 連文法語言模型是比較廣泛使用的。較常使用的 N 連文法語言模型是雙連文法 (Bigram) 語言模型或三連文法 (Trigram) 語言模型，而完全不看詞的歷史紀錄則稱為單連文法 (Unigram) 語言模型。一般 N 連文法語言模型訓練過程，首先會先找一個足夠大且平衡、覆蓋率高的語料 (Corpus)，在正規化 (Normalize) 之後，計算這個語料中出現的詞頻 (Word Count)，再利用最大相似度估測法 (MLE：Maximum Likelihood Estimation)，及一些平滑化 (Smoothing) 的技術猜測無法由語料中估計的部分，最後獲得離散的機率分佈。

平滑化 (Smoothing) 技術是 N 連文法語言模型中很重要的一環。由於訓練語料庫只是在一個特定時間裡提供某個特定語言的語料子集，因此並無法涵蓋所有未來的語料資料。因此，這些未曾在訓練語料庫中出現的語言行為其所被分配到的觀察機率將是零。這是一個非常嚴重的後果。故此，平滑化技術的主要精神在於，即使某些事件在過去的觀察中未曾出現過，我們也必須有個方法給予這些未

知事件一個合理的估測機率。方法通常是調低高頻率的事件機率挪出一部份機率質量，再將其分配給未曾出現甚或出現次數過少的事件。這樣子調整雖然無法完美匹配訓練語料庫的特徵，但通常在未知的測試語料上能達成較高的估測率。這方面，最常用的方法屬古德圖靈 (Good-Turing) 平滑法 [25] 及強化聶氏 (Modified Kneser-Ney Smoothing) 平滑法 [26]。考量到強化聶氏法在許多文獻中不斷被使用且漸漸成一個標準，本論文中的 N 連文法語言模型都採用此平滑法。

2.1.3 類神經網路語言模型 (NNLM)

人工類神經網路 (Artificial Neural Networks, ANN)，簡稱類神經網路 (Neural Networks)，發源於人工智慧領域 (Artificial Intelligence)，是一種模仿生物神經網路的結構與功能的數學模型，如圖 2.2(b)所示。它由大量類神經元 (neuron) 互相聯接所構成。每個神經元通常有多個輸入訊號 (input) 及一個輸出訊號 (output)，而該神經元內部代表一個將輸入映射到輸出的激勵函數 (activation function)。而兩兩神經元間的鏈結都代表一個對於通過該鏈結訊號的一個加權值，稱作權重 (weight)，亦即該神經網路的記憶。而網路的最終輸出則與每層的激勵函數及網路結構不同而不同。類神經網路本身往往代表對一個自然界函數的逼近或是一種邏輯策略的表達。其學習過程乃基於對所觀測到的訓練樣本，對各個層的權重進行校正而建立模型，具體方法則因網路結構不同而不同。一般而言常用反向傳播演算法 (Backpropagation) 來學習。 類神經網路被認為俱有以下優點：(1) 俱有高度平行化



(a) 類神經元構造　　(b) 類神經網路結構

Figure 2.2: 類神經網路網路結構與類神經元構造

特性 (2) 錯誤容忍的能力 (3) 自我調試與聯想式記憶 (associative memory) (4) 解決一般演算法及最佳化難以直接解決的問題 (5) 可以直接以硬體實作 (6) 輸入到輸出為非線性映射且數學推導嚴謹。但其最令人詬病的兩大缺點：(1) 計算複雜度過高及 (2) 訓練不當極容易遇到過度貼合 (over-fitting) 或過低趨近 (under-fitting) 問題。卻一直是類神經網路發展以來的兩大障礙。過度貼合是因為過度訓練而導致將訓練資料中的雜訊也當成是資訊學習進去，導致對未看過的資料做出了過度自信的誤判。這種現象往往是用了高度複雜的網路結構 (例如過多的隱藏層或過多的神經元數目)，而導致訓練收斂到局部最佳解 (local optimal) 而非全域最佳解 (global optimal)。但過於簡單的網路結構卻可能反向造成過低趨近的問題，導致訓練難以收斂或網路預測能力下降。因此，在現代的類神經網路訓練中，常常採用多種正規化 (regularization) 方法與加速學習方法並用，以期快速地得到強健又表現良好的模型。這些方法包含：(1) 迷你批次梯度下降法 (mini-batch gradient decent) (2) 動量方法 (momentum method) (3) 自適應學習速率 (adaptive learning rate) (4)L1, L2 正規化 (L1 or L2 regularization) (5) 在深度學習 (Deep learning) 中常用的受限波茲曼機器初始化 (Restricted Boltzman Machine initialization) [4, 27]。

類神經網路語言模型 (NNLM) 是一種建立在類神經網路上的語言模型，透過將字詞投射到連續的特徵空間中 (continuous feature space) 表示，以克服 N 連文法語言模型的維度詛咒 (Curse of dimensionality)，以期獲得更高的未知詞估計能力。在語言模型方面，主要以兩種結構的類神經網路模型結構為主：

(1) 前饋式類神經網路語言模型 (Feed-forward Neural Network Language Model) [28--30]，如圖 2.3 所示，共有四層網路結構：輸入層 (input layer)，投射層 (projection layer)，隱藏層 (hidden layer)，及輸出層 (output layer) 所構成。前饋式類神經網路語言模型只利用有限長度的上下文關係估測下一個未知詞的機率，其上下文長度由串接的輸入層數目決定。輸入層的大小即為設定的詞典 (lexicon) 的大小，表示方式為 1-of-N 編碼 (1-of-N encoding) 形式，意即對應到該字詞的索引 (word index) 之值為 1，其餘為 0。每個字詞對應到的輸入層還必須經過一層線性轉換投射到一個較低維度的向量空間，稱作投射層 (projection layer)。這層線性轉換不管時間資訊，對各個時間點的輸入層來說是共享的 (shared)，主要目的是在進入隱藏層以前對每個字詞有更簡練的表示方式 (compact representation)。隱藏層 (hidden layer) 則

是總結了該網路對過去前 n 個字詞所構成的上下文歷史 (context history)，並將這個資訊往後傳遞產生最後的輸出層 (output layer)。輸出層與輸入層大小同樣為設定的詞典大小，唯一不同的是每個輸出層的類神經元代表了對下個未知字詞的預測機率。

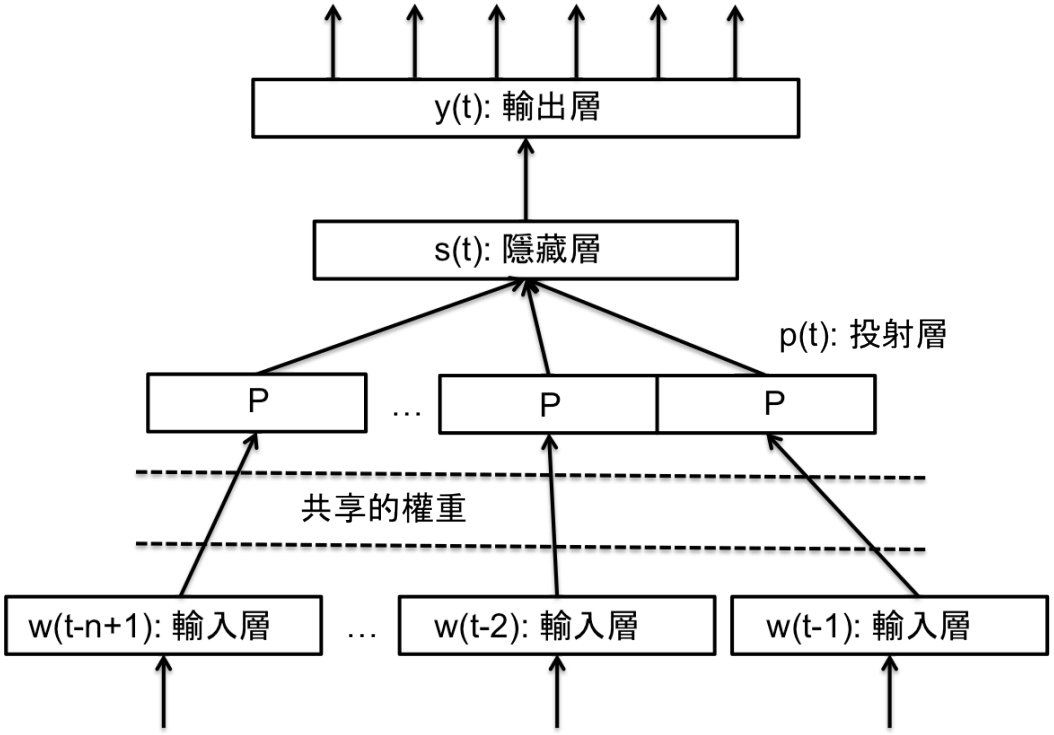


Figure 2.3: 前饋式類神經網路語言模型

(2) 而遞迴式類神經網路語言模型 (Recurrent Neural Network Language Model) [31--33]，如圖 2.4 所示，其輸入層與輸出層設定都與前饋式網路類似，而捨棄了映射層及多個輸入層只用了單一輸入層。 其主要賣點在於隱藏層裡的遞迴式結構 (recurrent structure)，實際做法是將隱藏層的輸出當作下一次時間點隱藏層的輸入，而根據不同的需求也有許多不同的網路形成。這樣子的遞迴式結構，一方面可以免去人為明確模擬本質上模棱兩可的上下文長度的缺點；另一方面藉由不斷的遞迴資訊，可以模擬任意長度的上下文關係。有關於更詳細的遞迴式類神經網路語言模型探討，包含他的結構組成及數學推導，將在 4.3.1 做更深一層的闡述。

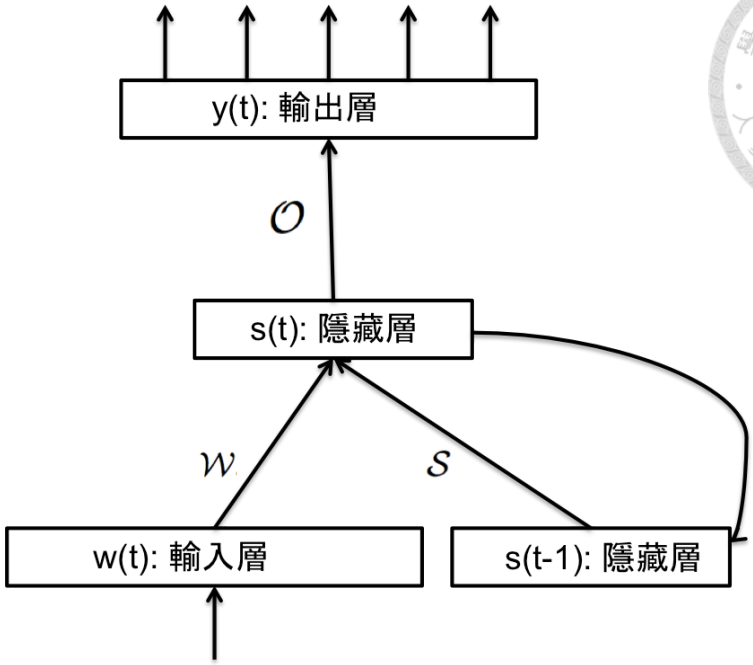


Figure 2.4: 遞迴式類神經網路語言模型

2.1.4 語言模型調適

人類語言俱有一定的規律性，這是隨著人們長時間慢慢演變且保存下來的特定習慣。然而語言在許多層面來說，並不是定性而是隨著許多因素而動態改變著：首先，我們可以很容易想像語言的詞典是隨著時間不斷在改變的。在十八世紀時的語言很理所當然的絕對不會出現" 電腦"、" 電視機"、與" 核融合" 這些隨著歷史演進在十九世紀甚或二十世紀才被發展演變出來的概念而成為新的詞彙。又如" 核四" 與" 粉絲" (fans) 這些詞更是近代才出現的。除此之外，當下正在討論的主題內容也是一個很大的影響關鍵。舉例來說，討論的主題是科普知識與討論主題是政治新聞其使用的詞彙，甚或同一詞彙的使用方式將會完全不同。再者，即使人們討論的主題是相同的，在不同的使用情境下也會有不同的遣詞用字。假設同樣在討論科學知識，論文的內容與寄給同儕的電子郵件也將有截然不同的機率特性。最後，不可忽略的是每個人在談話內容的當下所在的社會地會與背景。最明顯的例子便是在下對上與上對下這兩種不同角色地位時呈現出來的語言差異。

諸如上述，由於語言本身存在如此多的變異性，加上這類統計或機器學習的方法很容易被訓練語料與測試語料中的不匹配 (mismatch) 影響測試結果，這對於語言模型來說無疑是一個很糟糕的消息。為了解決語料不匹配的問題，語言模型必

須隨著時間與使用情境來更新似乎是一個很直觀的做法。但是不斷重新訓練模型不實際的點在於，從訓練語料到語言模型往往需要大量的計算，如此一來將導致整個系統缺乏效率。因此，為了有效率地動態調整語言模型，若干語言模型調適 (Language Model Adaptation) [13] 的技術在過去的幾年來不斷地推陳出新。

當前廣泛應用在語言模型調適的架構如下：我們考慮兩組訓練語料：(1) 一組小的調適語料 A (adaptation corpus) 非常貼近我們要辨識的目標測試語料；(2) 一組大且健全的背景語料 B (background corpus)，但可能是過時的 (out-of-date) 或者沒有那麼直接相似於目標測試語料。給予一句長度為 N 的句子，$\{w_q : 1 \leq q \leq N\}$，該句子的機率可以被寫為

$$P(w_1, ..., w_N) = \prod_{q=1}^{N} P(w_q|h_q) \tag{2.2}$$

$h_q = \{w_{q-n+1}, ..., w_{q-1}\}$ 是在時間 q 時的有限長度上下文 (limited-length context, N 連文法語言模型) 甚或任意長度的上下文資訊 (arbitrary-length context, 遞迴式類神經網路語言模型)。在沒有調適語料的狀況下，$P(w_q|h_q)$ 的值便是完全由背景語料所訓練的背景語言模型 (background language model) 估計所得。在統計式語言模型調適 (Statistical Language Model Adaptation) 的架構下，調適後的機率估計是由兩個模型的線性內插所得

$$P(w_q|h_q) = \alpha P_A(w_q|h_q) + (1 - \alpha) P_B(w_q|h_q) \tag{2.3}$$

其中，$P_A(w_q|h_q)$ 與 $P_B(w_q|h_q)$ 分別是調適語料與背景語料的語言模型對該測試語料的估計機率，α 是他們之間內插的權重。權重估計的方法有很多 [34--37]，常見的兩種做法一是直接用某種標準計算或是切出另一組發展語料組 (development set) 並調整權重使得模型對該發展語料的混淆度 (perplexity) 最低。在章節 4.1 中我們會針對如何計算各個語料間的權重有更完整地介紹。另外，由於類神經網路語言模型調適非常不同於傳統 N 連文法語言模型，它有自己的優點及缺點，且目前也尚未有另外一套廣泛有效的做法，因此我們將獨立出來在章節 4.3 內深入探討。

2.2 互動式搜尋系統

誠如章節 1.1.2所提到，由於辨識錯誤對文件搜尋技術的負面影響使得搜尋結果不一定盡如人意，再加上語音文件本身難以瀏覽的特性，系統必須更明確地過濾不相關的文件以期能幫助使用者快速地找到目標文件。這樣一來，當使用者一開始輸入的搜尋指令 (query) 過度模糊時，為了做更精確地檢索，系統與使用者互動 (human-machine interaction) 從使用者端獲取更多資訊是在所難免的。但這樣子的互動倘若過多將會對使用者造成不快，倘若過少又無法回傳精準的搜尋結果，因此一個良好的互動式搜尋系統應該要在這兩者之間做一個權衡 (trade-off)，在任意一個可能所處的狀態裡都能做出最聰明的決定進而最佳化使用者的滿意程度。

建構一個互動式搜尋系統主要的兩個核心技術在於：(1) 資訊檢索技術 (Information Retrieval)，根據使用者輸入的查詢指令 (query) 找出對應的相關文件集合 (relevant document set)。(2) 對話管理員 (dialogue manager)，決策核心，根據現在系統所處的狀態採取最佳的回應。在本章節的後半段我們將簡單介紹在上述兩個領域裡的重要背景知識，以期在更深入探討核心內容前打下學理基礎。

2.2.1 語音文件搜尋

資訊檢索 (Information Retrieval)，一般被定義成為從未結構化的資料庫 (unstructured database) 中，擷取使用者所要的資訊的技術。在眾多檢索技術裡面，本篇論文想要討論的重點在於語音文件檢索技術 (spoken document retrieval)。語音文件檢索與一般文件檢索 (document retrieval) 方法上大致相同，唯一不同在於必需在後端外接一個辨識系統將語音文件先轉譯成文字文件，再套用文字檢索的相關技術。其基本架構大致如圖 2.5 所示。 從圖中我們可以得知，文字檢索的工作主要是由使用者輸入查詢指令 (query) 後，利用資訊檢索模型 (retrieval model) 去計算文件集中每一篇文件與查詢指令之間的相關度 (relevance)，最後再將檢索出來的文件依相關性程度高低做排序後呈現給使用者看。若以較正式的定義方式我們可以表示如下：

- 詞典 (lexicon)， $V = \{v_1, v_2, ..., v_{||V||}\}$，其中每一個 v_i 都是一個索引詞 (indexing word)。

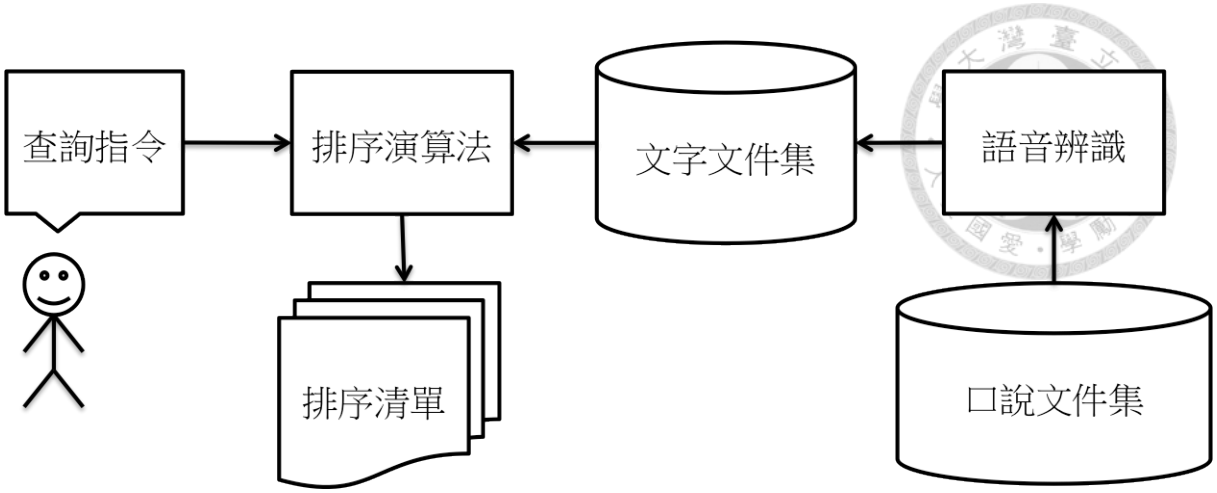


Figure 2.5: 語音文件檢索系統基本架構

- 查詢指令 (query)， $Q = \{q_1, q_2, ..., q_{||Q||}\}$， $q_i \in V$，亦即 q_i 為詞典裡的詞。
- 文件 (document)， $D_j = \{w_{1,j}, w_{2,j}, ..., w_{||D_j||,j}\}$， $w_{i,j} \in V$， $w_{i,j}$ 都在詞典裡。
- 文件集 (document archive)， $C = \{D_1, D_2, ..., D_{||C||}\}$。

使用者以查詢指令 Q 作為給系統的提示，希望能從系統的文件集 C 中找到與該查詢指令 Q 相關的文件集合 $R(Q) \subseteq C$。可惜的是儘管 Q 相同 $R(Q)$ 仍因每個使用者而異，因此對系統來言是無法求得真正的 $R(Q)$。因此，文件檢索的目的便是希望求得 $R'(Q)$ 盡量逼近 $R(Q)$，而這部分便是由匹配演算法 (matching algorithm) 來做計算。下面我們將先進行語音文件索引 (Indexing) 方法的介紹，而後在介紹當前最常見的兩種不同匹配演算法：章節 2.2.3，向量空間檢索模型 (Vector Space Model Retrieval Model) 及章節 2.2.4，語言模型檢索模型 (Language Model Retrieval Model)。

2.2.2 語音文件索引

索引 (Indexing) 是一種將目標文件轉換成特定資料結構 (Data structure) 的技術，常用在具有靜態 (static) 或半靜態性質 (semi-static) 之文件集的文件檢索工作上。建立索引跟搜尋本身是密不可分的，不同的索引方式會大大影響到搜尋的準確性與速度。因此，針對文件的索引設計，除了要考量文件本身的多種資料格式與查詢指令間可能存在的不匹配問題以外，我們也必須同時考量檢索速度、檢索效果、以及儲存空間等等不同的因素，才能針對該特定的文件集，選出真正適合的

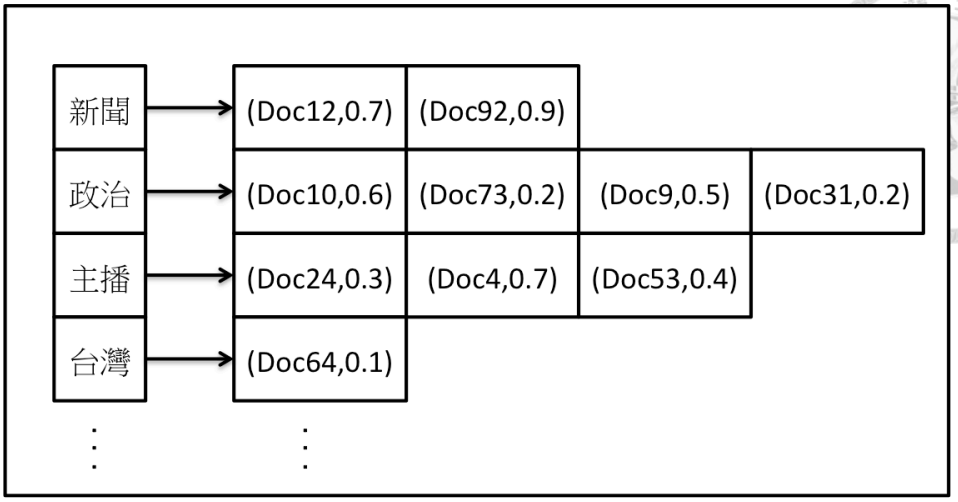


Figure 2.6: 反向索引法的例子

一套索引方式。由此可知，檢索的使用決定了系統的效能，因此針對我們語音文件檢索而言，索引必然須針對語音的特性做適當的調整，才有辦法增進檢索的功效。以下介紹了目前最典型常用的兩種語音文件索引方式：

- 最佳序列反向索引法 (1-best Inverted Indexing)：最佳序列 (1-best) 即是經過語音辨識對語音文件自動轉譯之後，在所有轉譯的可能序列中，挑選事後機率 (Posterior probability) 最高的那串文字序列，$W^* = w_1 w_2, ...w_n$ 作為索引目標。由於最佳序列只保留了辨識過程中機率最高的該條路徑而捨棄了其它機率較低的路徑，因此雖然它對於儲存空間的使用及計算複雜度上較有效率，但在面對辨識錯誤時將完全沒有補救的辦法。我們將整個文件集都每一篇語音文件都進行語音自動轉譯並取其最佳序列，而後針對每一個在詞典裡的字詞 $w \in V$，我們計算他在每篇文章裡面的重要性分數 (可以是權重或者機率)，並同時記錄該字詞有出現的每篇文章代碼 (ID)。而後，在資料庫內的結構我們便只要記錄每個字詞出現在哪些文章中，以及它在該篇文章中的重要性分數，如圖 2.6所示。 這樣的反向索引法的好處在於，給定使用者的查詢指令 (Query) 後，我們能夠很快地找到對應有出現該查詢指令的文件，並依照其重要性分數總和做排序，有利於檢索的速度。

- 詞圖反向索引法 (Lattice Inverted Indexing)：在語音辨識的搜尋過程，如果我們為每個音框 (Frame)，將目前存活且分數較高的詞彙樹 (lexicon tree) 葉端節點 (leaf node) 中的聲學解碼分數、語言模型歷史或是候選詞開始與結束的

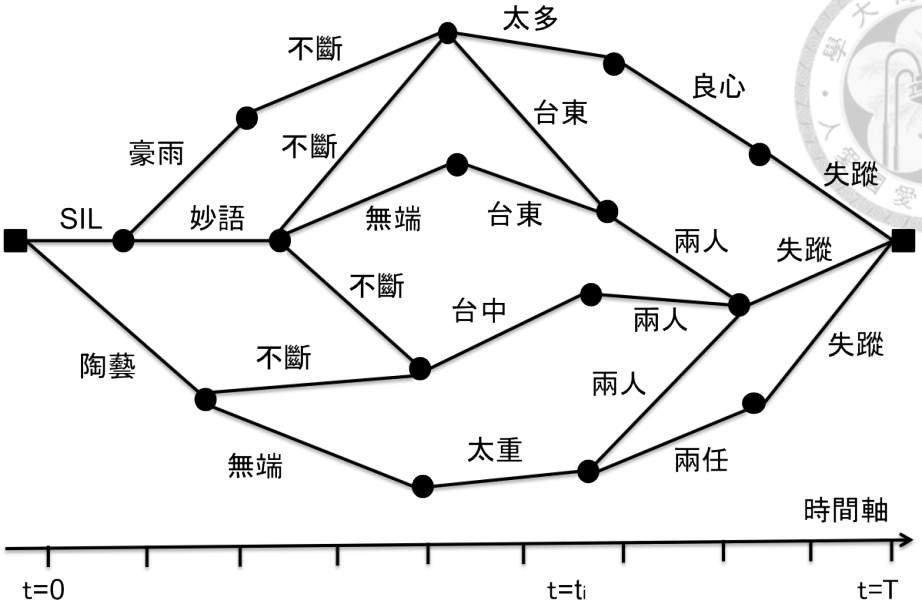


Figure 2.7: 詞圖範例

音框等資訊記錄下來，在最後進行回溯 (back track) 的動作後，便能藉由這些資訊展開一張高分候選詞所組成的詞圖 (word lattice)，如圖 2.7。詞圖的每個節點記錄了該節點的時間，每個詞弧則代表一個字詞，同時記錄了其開始結束時間、及辨識的信心分數。這樣的詞圖包含了許多資訊，允許我們利用許多不同的方法 [16, 38, 39]，來更精確地估計每個字詞在該語音文件中的期望出現次數 (expected count)，進而更精確地計算該字詞的重要性分數。再者，由於詞圖同時納入了許多前 N 最佳序列，因此它對於對抗辨識錯誤有一定的成效。而後，我們同樣可以套用反向索引法，來加速搜尋的速度。

2.2.3 向量空間檢索模型 (VSM)

向量空間檢索模型可以說是文字檢索系統中最早被廣為使用的模型。在這個模型的架構下，每篇文件 D_j 中的每個在索引詞典內的詞 $v_i \in V$ 都有一個權重值，我們稱為 $t_{i,j}$。$t_{i,j}$ 與該詞 v_i 在文件 D_j 及文件集 C 中的統計特性有關，最常見的屬詞頻反文件頻 (TF-IDF：Term Frequency/Inverse Document Frequency) 權重：

$$t_{i,j} = (1 + \ln c_{i,j}) \cdot \ln \frac{N}{N_i} \tag{2.4}$$

$c_{i,j}$ 是索引詞 v_i 在文件 D_j 中出現的次數，N 是整個文件集的文件數目，N_i 是出現該索引詞 v_i 的文件總數 (又稱文件頻，document frequency)。如此一來，文件 D_j 在向量空間中的表示法便可定義為一組特徵向量 (feature vector)：

$$\hat{d_j} = (t_{1,j}, t_{2,j}, ..., t_{||V||,j}) \tag{2.5}$$

而同樣的，使用者的查詢指令 Q 也可依法炮製表示成一組在該向量空間中的特徵向量：

$$\hat{q} = (t_{1,q}, t_{2,q}, ..., t_{||V||,q}) \tag{2.6}$$

最後，衡量文件向量 $\hat{d_j}$ 及查詢指令向量 $\hat{q}$ 之間的相關程度我們可以用三角函數中的餘弦函數 (cosine) 來估計：

$$R(\hat{d_j}, \hat{q}) = \cos(\hat{d_j}, \hat{q}) = \frac{\hat{d_j} \cdot \hat{q}}{||\hat{d_j}|| \cdot ||\hat{q}||} = \frac{\sum_{i=1}^{||V||} t_{i,j} \cdot t_{i,q}}{\sqrt{\sum_{i=1}^{||V||} t_{i,j}^2} \cdot \sqrt{\sum_{i=1}^{||V||} t_{i,q}^2}} \tag{2.7}$$

2.2.4 語言模型檢索模型 (LM)

在向量檢索模型在資訊檢索領域風行了一段時間之後，語言模型檢索模型近幾年來越來越受到許多研究學者的歡迎。主要原因在於它背後的數學理論可以套用許多機率演算，比起向量模型而言較為完整。相較於向量檢索模型將文件與查詢指令都表示為向量，語言模型檢索模型則是將兩者都表示成一個由索引詞典詞彙展開的空間中的一個多項機率分佈 (multinomial distribution)，這個機率分佈亦可以看作是該文件或查詢指令的語言模型。其中，查詢指令語言模型的多項機率分佈我們可以表示為：

$$P(v_i|\theta_Q) = \frac{c_{i,q}}{\sum_{i=1}^{||V||} c_{i,q}} \tag{2.8}$$

其中 θ_Q 表示查詢指令語言模型，而 $c_{i,q}$ 是索引詞 v_i 在查詢指令 Q 中出現的次數。較為複雜的是文件語言模型。我們通常不直接利用文件語言模型的多項分佈 $P(v_i|\theta_{D_j})$ 如式 2.8。相反的，我們會計算一個平滑化 (smoothing) 過後的文件語言模型 θ'_{D_j} 來作為未來比對的依據，其多項分佈可以寫成：

$$P(v_i|\theta'_{D_j}) = \alpha_{D_j} P(v_i|\theta_{D_j}) + (1 - \alpha_{D_j}) P(v_i|\theta_C) \tag{2.9}$$

由式 2.9 我們可以看出平滑化的文件語言模型基本上是原來的文件語言模型 θ_{D_j} 與背景文件集語言模型 (background language model) θ_C 的線性內插。其中，$P(v_i|\theta_{D_j}) = \frac{c_{i,j}}{\sum_{i=1}^{||V||} c_{i,j}}$ 而 $P(v_i|\theta_C) = \frac{\sum_{j=1}^{||C||} c_{i,j}}{\sum_{i=1}^{||V||}\sum_{j=1}^{||C||} c_{i,j}}$。$\alpha_{D_j} = \frac{||D_j||}{||D_j||+L}$ 是一個文件相關 (document-dependent) 的內插權重，L 則是需要調整的自由參數 (free parameter)。最後，有了文件與查詢指令的語言模型之後，我們可以利用 KL 散度 (KL-divergence) 衡量兩個模型的相似程度 [5, 40]：

$$R(\theta_Q|\theta_{D_j}) = -KL(\theta_Q|\theta_{D_j}) = -\sum_{i=1}^{||V||} P(v_i|\theta_{D_j}) \ln \frac{P(v_i|\theta_{D_j})}{P(v_i|\theta_Q)} \tag{2.10}$$

2.2.5 對話管理者 - 馬可夫決策模型 (MDP)

欲提及對話管理者 (dialogue manager)，就不得不提及對話系統 (dialogue system)，如圖 2.8。對話系統主要包含三個部分：(1) 語音辨識 (Speech Recognition)

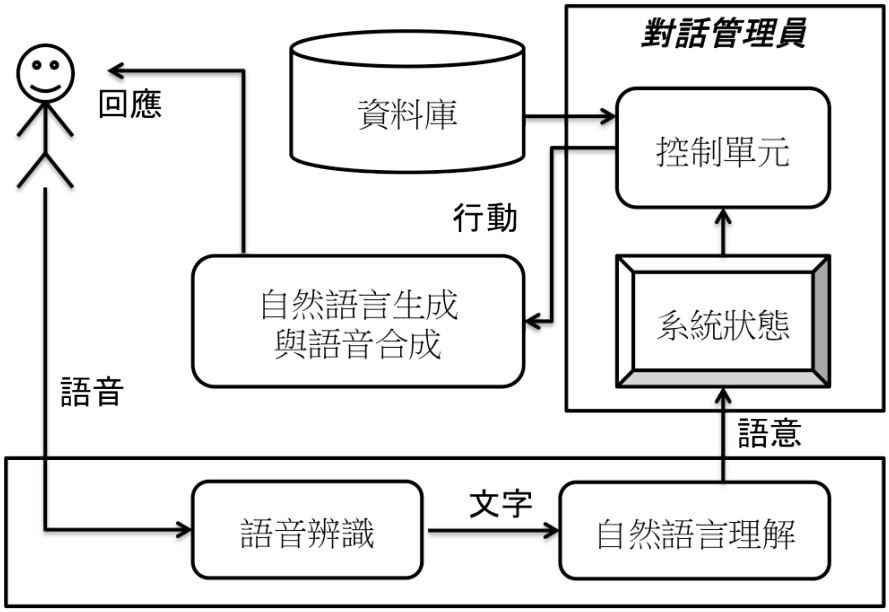


Figure 2.8: 傳統對話系統架構

與自然語言理解 (NLU：Natural Language Understanding) (2) 語音合成 (Speech Synthesis) 與自然語言生成 (NLG：Natural Language Generation)。(3) 對話管理員 (Dialogue Manager) 我們其實可以簡單地將前兩個部分當成該系統的兩個轉譯機制：第一部分將使用者的語音輸入轉譯成系統可以了解的語意片段 (semantic slots)，而後進一步可以將其映射到一個系統內部所定義好的狀態 (state)。第二部分則是將系統的決策動作 (action) 生成自然語言後再合成語音回應的過程。而對話

管理員的部分則是負責根據當下系統的狀態作出最適當的決策，而這個系統狀態主要取決於互動歷史 (interaction history) 與使用者當下的輸入 (input)。故對話管理員可說是一個對話系統最核心的部分，而這當中最有名也最行之有年的模型，便是再來要介紹的馬可夫決策模型 (MDP：Markov Decision Process)。

馬可夫決策模型 (MDP) [41, 42] 可以用五個參數表示之：$\{S, A, T, R, \gamma\}$。S 代表是系統狀態 (state) 的集合，包含系統任何可能所處的狀態，是由人預先定義好的，可以是離散空間 (discrete space) 亦可以是連續空間 (continuous space)。空間狀態集合的定義可以說對整個系統的表現影響最大，因為它代表了系統對整個對話過程所獲得資訊的總整理，而未來的決策也都取決於所處的系統狀態。一般對話系統制定狀態集合的方式是窮舉 (enumerate) 所有狀態的組合，這樣往往演伸出來的問題就是狀態數目指數遞增，導致某些不常走到的狀態很難被訓練演算法 (training algorithm) 訓練的好，這個問題亦被稱為維度詛咒 (Curse of dimensionality)。故後來很多在馬可夫決策模型或部分可觀測馬可夫模型 (POMDP：Partially Observable Markov Decision Process) 的研究目標都放在如何對狀態空間作降維 (dimension reduction) 但同時又不影響其系統表現。A 是系統可以採取的動作 (action) 集合，同樣是人為預先定義好的。動作的定義通常是抽象的 (artbitrary)，例如詢問某個特定問題或回應，而實際的回應語句則是由外部的自然語言生成 (NLG) 單元產生。R 是系統的獎勵函數 (reward function)，一般嚴謹寫法應被寫作 $R_a(s, s')$，亦即該系統在狀態 s 採取動作 a 後轉移入狀態 s' 後所獲得的獎勵 (reward) 或代價 (cost)。T 則是系統的轉移機率函數 (transition probability function)，同樣的，嚴謹寫法為 $T_a(s, s')$，即該系統在狀態 s 採取動作 a 後轉移入狀態 s' 的機率。因此轉移機率函數又被稱作系統動態 (system dynamics)，對很多實際應用的系統如對話系統都是未知的，因此必須用採集來的數據對之建模，這被稱作以模型為主的方法 (model-based method)；或是不建模而直接模擬最後的系統表現，稱作無模型方法 (model-free method)。γ 是一個折扣係數 (discount coefficient)，通常用來表示系統往後考量的輕重程度。

再來，我們定義系統的決策 (policy)，為一個從狀態空間 $s \in S$ 映射到動作空間 $a \in A$ 的函數，稱它為 π。這個決策便是系統最終要學習的目標。給定決策之後，為了輔助決策的學習，我們再進一步定義 Q-函數 (Q-function) 為 $Q^\pi : S \times A \to \mathbb{R}$，

為一個從狀態與動作的聯合空間映射到一個實數 (real number) 空間的函數。其數學形式我們可以寫為：

$$Q^{\pi}(s,a) = E[\sum_{k=0}^{\infty} \gamma^k r_k | s_k = s, a_k = a, k = 0, 1, 2, ...] \tag{2.11}$$

k 為時間序列上的索引，而 r_k、 s_k、 a_k 則分別代表在該時間點系統獲得的獎勵、所處的狀態、及採取的行動。如此一來， Q-函數的物理意義便昭然若揭：整個對話流程中系統可以獲得的獎勵期望值總和。最後，系統最佳的決策 (optimal policy) 則必須藉由間接最大化每個狀態及行動組合 (state-action pair) 的 Q-函數值：

$$Q^*(s,a) = E_{s'|s,a}[R_a(s,s') + \gamma \max_{b \in A} Q^*(s',b)] \tag{2.12}$$

式 2.12經過不斷地遞迴形式讓 Q-函數漸漸收斂到一個穩定的值，而這個穩定的 Q-函數則可以再用來萃取出最佳的決策：

$$\pi^*(s) = \max_{a \in A} Q^*(s,a) \tag{2.13}$$

由此可知求取最佳決策其實等同於最佳化 Q-函數。式 2.12 在強化學習 (Reinforcement Learning) 領域的研究中，被廣泛稱為值迭代法 (Value Iteration) [43]。

Part I

個人化語音辨識系統

Chapter 3

社群網路群力模式

3.1 群力模式概念介紹

群力模式 (crowdsourcing) [44, 45] 是一種分佈式 (distributed) 的問題解決和生產模式，由雜誌記者 Jeff Howe 在 2006 年提出用來專指這類型透過網際網路將問題外包給群眾的新型商業模式。它的整個運作方式大概如下列所述：(1) 委託工作者 (requester) 將原本性質較為單純但仰賴大量人力且耗時、且在現階段電腦仍無法解決的工作，藉由分割的方式切分成許多細小瑣碎的工作。這些個別的小工作通常獨立運作而且可以在相對上較短的時間內完成。(2) 這些獨立的小工作再以按件計酬的方式，直接在網路上號招大量有意願的群眾參與，每個小部份由不同的工作者 (Worker/Turker) 完成。這樣子的網路外包通常在一些有許多工作者聚集的網路平台上發生，例如 Amazon Mechanical Turk(AMT)、CloudCrowd、或者 InnoCentive 等。(3) 在工作完成之後，通常必須要有一套審查機制，對這些被分配出去的小工作進行過濾，以防有不負責任的工作者進入造成工作不完整或錯誤。這些審查方案往往也是藉由群眾的力量完成，機制類似同儕評審 (peer review)。(4) 最後，檢視過的工作由委託工作者回收，並給予這群工作者適當的報酬。一般而言，這類外包平台上的參與群眾多半不具特定專業背景，且多半是利用平時閒暇之餘參與外包工作以索取低廉的酬勞。所以以工作者角度來說，此平台不僅成了使用網路並消磨時間的一項新選擇，同時也是增加額外收入的管道。而委託工作者角度來看，相較於僱請專人處理的傳統做法，利用群力模式的多工方式不僅成本低廉，更提升了處理速度，同時仍可得到與傳統作法相當的工作品質。不同

現行的群力模式系統分類大致上可以整理如下圖 3.1所示。

工作者 貢獻的形式	系統架構	雇用工作者	工作內容	範例	目標問題
顯性貢獻	獨立系統	需要	評估： 評價、投票、標記	del.ici.ous.com 網頁標記, Yelp, Amazon	評估物件、收集等
			分享： 1. 物件 2. 文字資訊 3. 結構化知識	1. Youtube, Flickr, Napster … 2. Yahoo 知識家, Quora … 3. Swivel, Google base …	建構一個可以共享的平台
			社交互動	LinkIn, MySpace, Facebook	建構社群網路
			建構平台或系統： 1. 軟體 2. 文字資料庫 3. 系統 4. 其他	1. Linux, Apache, Hadoop 2. Wikipedia, openmind, Intellipedia 3. Wikia Search, mahalo, Freebase 4. Digg.com, Second Life	建構平台或系統
			執行工作	尋找朋友、電子產品 … 等等	可以是任何的問題
隱性貢獻	獨立系統	需要	1. 有目的的遊戲 2. 打賭預測市場 3. 解影像驗證碼 4. 買、賣、拍賣，多人線上遊戲	1. ESP 2. intrade.com, Iowa Electronic Market 3. recaptcha.net 4. eBay, World of Warcraft	1. 照片標記 2. 預測事件 3. 數位化手寫文字 4. 建構社群(為了未來收費或廣告)
	依附 現有系統下	不需要	1. 關鍵字搜尋 2. 購買產品 3. 瀏覽網頁	1. Google, Microsoft, Yahoo 2. Amazon 3. Yahoo! front page	1. 拼字更正、關鍵詞推薦 2. 推薦產品 3. 調整網頁設計

Figure 3.1: 不同群力模式系統按照不同層面做檢視與分類

群力模式在機器學習 (machine learning) 上的應用越來越重要，主要原因是近幾年機器學習越來越常被應用到許多不同層面的真實世界問題，但往往在處理這些問題的時候才會發現，有標記且高品質的訓練資料庫 (training data set) 其實也十分難以得到。因此，利用群力模式的方式來獲得這些初始化的學習資料便變成一個十分直覺的結合方式。而在近幾年，也有越來越多的研究開始著重在自我適應學習 (self-adaptive learning) 的系統，如 MIT 的電影瀏覽器 (MIT MovieBrowser) [46,47]。這些研究的主要論點在於，很多的應用其實會隨著時間及人們習慣的改變而不斷改變，一個靜態 (static) 的系統其實對於這樣的轉變是沒有任何應變的能力。為了要讓系統能在任何時刻都能盡量符合使用者的需求，這個系統必須要是動態 (dynamic) 不斷在根據現階段搜集進來的資料不斷地做演化。而這些會隨時間不斷調適的系統，往往便是採用如圖 3.1 中的隱性貢獻機制。而本部分再來要討論的內容，主要便是落在隱性貢獻機制這個群力模式的光譜之內。

3.2 具備語音界面的社群網路瀏覽器

本章節我們將介紹一個具備語音界面的社群網路瀏覽器 [15, 48]，作為一個我們個人化辨識系統的應用程式載具。藉由這個應用程式載具我們提供應用服務給使用者，而使用者則藉由使用這項服務留下個人化的語料。而後，藉由搜集到的語料庫，我們可以調適現有的模型，使這些模型隨著時間與使用者特型不斷演進。

3.2.1 系統特點

圖 3.2 便是我們提出的以語音為接口的社群網路瀏覽器系統架構圖。這個系統主要包含三個主要的部分：(1) 手機用戶端：為使用者用戶端 (client side)，其功能為輔助使用者透過語音或非語音指令來瀏覽社群網路，其功能包含社群瀏覽 (browsing)、語音動態更新 (speech posting)、及辨識錯誤更正 (error correction) 等等。(2) 第三方 (third-party) 社群網站：此為使用者真正要瀏覽的目標網站，也是我們搜集語言模型個人化語料庫的主要來源。它可以是任意現存的個人網站，如臉書 (Facebook)、推特 (Twitter) 或噗浪 (Plurk) 等等，本論文以臉書作為例子。 (3) 雲端個人化辨識系統：此為應用程式的伺服器端 (server side)，主要包含個人化的聲學模型、語言模型、辨識解碼器 (decoder)、及聲學模型與語言模型的個人化機制。當使用者一開始使用這個應用程式的時候，應用程式會要求使用者輸入一個現存臉書帳號並綁定該使用者為服務對象。在此同時，使用者會被額外要求一個獲得過去動態文章及朋友動態的權限，關於臉書權限我們將在章節 3.2.2作進一步說明。倘若使用者同意了此權限，則雲端伺服器端將會將這些過去的動態文章下載下來形成一個該使用者的個人化語料庫，成為個人化語言模型調適的資料來源。再來，當使用者透過手機用戶端應用程式輸入語音時，這段語音錄音會先透過網際網路傳到我們的雲端伺服器端。一方面，這段語音會被儲存下來在未來做非監督式 (unsupervised) 的聲學模型調適，以期獲得一個更接近該語者聲音的聲學模型；另一方面，這段語音會被辨識解碼器藉由個人化的聲學模型與語言模型辨識成文字並回傳回手機用戶端。使用者可以在手機端更正辨識結果，或者是直接將它上傳到自己的臉書動態牆 (wall) 與大家分享。

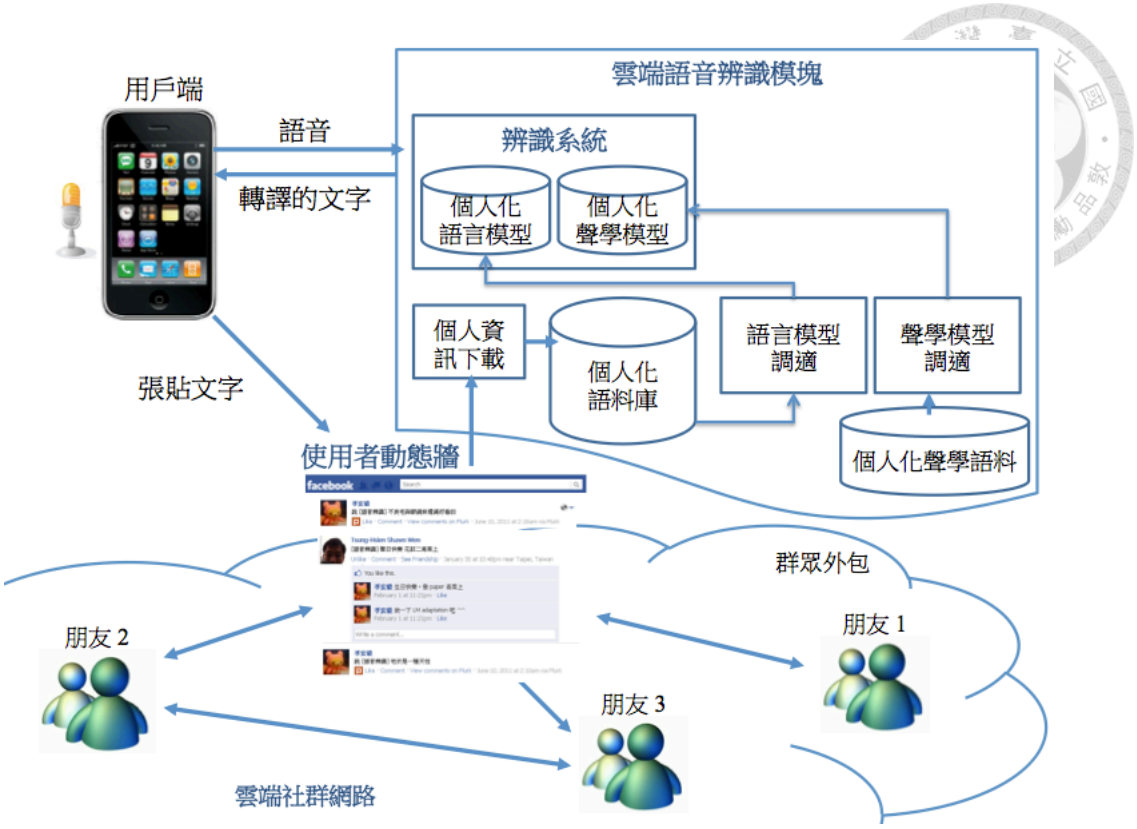
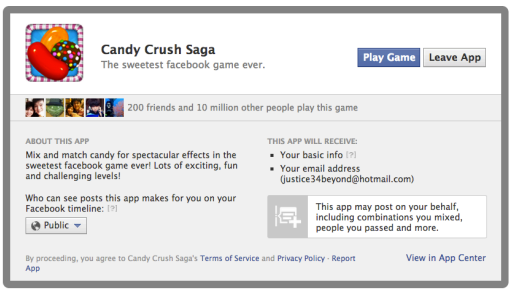


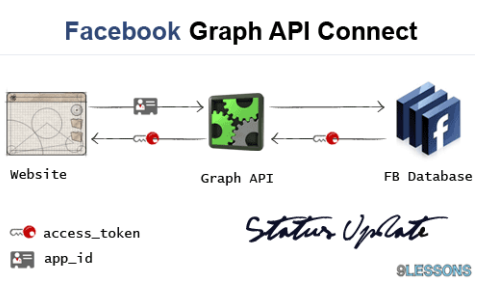
Figure 3.2: 以語音為接口的社群網路瀏覽器系統架構

3.2.2 臉書認證機制

為了保護使用者的隱私權，許多知名的社群網站都有隱私權的協定，以保護使用者個人資料非經過同意被其他第三方人士使用。以臉書為例，每支獲取使用者特定個人資料的應用程式界面 (API：Application Programming Interface)，都必須額外附加一個相對應的存取令牌 (access token)，才能獲得相對應的資料權限。而這個存取令牌的獲得，是在使用者第一次使用該應用程式的時候，該應用程式便必須向使用者要求並獲得同意之後，由臉書官方藉由網際網路傳送給該應用程式。 這個存取令牌有各式各樣不同的類型，包含：讀取使用者的朋友清單、



(a) 存取令牌使用者要求界面



(b) 存取令牌獲得示意圖

Figure 3.3: 臉書認證系統

朋友詳細資訊、個人動態牆發文、聊天記錄、以使用者身份發文、分享文章、按" 讚" 等等，更多有關存取令牌類型資訊請參考臉書開發者官方文件 (Facebook Developer Guide)。除此之外，存取令牌也有時間的限制，倘若超過有效時間便必須重新向使用者索取。使用者也擁有在任意時間取消存取令牌的權力。

3.2.3 系統生態系

藉由這樣子的一個與社群網路結合的雲端辨識系統，我們能夠在相當程度上，做到語音辨識系統的個人化，也就是聲學模型與語言模型的個人化。首先，我們不再有過去語料庫歸屬的問題，主要原因是有二：一是由於智慧型手機的使用普遍已經趨向一個個人化的使用經驗，通常不會有一支手機有多個使用者的情形；二是因為系統綁定了使用者的社群網路帳號，藉由這個帳號我們同樣可以確認使用者的身份。語言模型調適的語料庫來源主要來自於使用者綁定個人社群網路帳號之後，系統所獲得的個人化語料庫。除了語者本身的語料庫有助於模型調適之外，我們也可以藉由估計語者與不同朋友之間的語言相似度，將不同的朋友語料庫加進來幫助模型調適。雖說隨著使用者使用的過程中，我們可以將辨識後的文字當作新的語料庫進行動態調適，但許多過去的經驗顯示，動態利用轉譯過後的文字做語言模型調適是有風險的，主要原因在於這樣的做法會將許多辨識錯誤的結果給學進去。因此，由於我們有提供手機端更正功能，比較適當的做法應該是只將使用者更正過的句子拿來做動態調適，而忽略使用者沒有更正過的句子，這樣一來我們至少確定更正後的結果是在使用者可以接受的範圍之內。關於語言模型的個人化，我們將在第 4章做更深入的討論。最後，由於一開始並未有該使用者的聲音語料庫，聲學模型的個人化資料來源主要來自於使用者不斷使用而留下來的錄音，故相對語言模型來說，它是一個迭代的過程。在這裡我們主要進行的是非監督式 (unsupervised) 的聲學模型調適，主要原因是標記的聲學語料庫在這樣的系統設計裡仍然難以取得。儘管如此，這樣子的個人化聲學模型已經能夠提供我們相當好的辨識準確率了。最後，這個系統符合了我們要追求的宗旨：一個自給自足的系統生態系。主要原因在於其調適的語料庫都是取之於系統本身，不需要外力的干涉便能夠隨著不同的使用者與不同的使用習慣而適應演進。

Chapter 4

個人化語言模型

4.1 N 連文法語言模型 (N-gram LM) 個人化

在上一章我們討論了個人化辨識系統的架構，以及它與應用程式的用戶端、第三方社群網路之間的運作機制，藉此整合這三塊內容達到一個自給自足的系統。而本章我們要討論的內容則著重在如何建構一個個人化的語言模型，而本節以 N 連文法語言模型為主要探討對象 [15]。儘管藉由系統設計與社群網路掛鉤讓我們可以獲得品質較高個人化語料庫，但個人化語言模型仍然面臨了許多問題。最首先遇到的問題就是，搜集到的個人化語料庫極小，通常只有幾百到幾千個句子不等，這樣的小語料庫無法直接訓練出一個獨立 (standalone) 且強健 (robust) 的語言模型，因此勢必採取如章節 2.1.4 所提到的語言模型調適的方法。假設我們有個目標使用者 u 我們想建其個人化 N 連文法語言模型 $P^{(u)}(w_q|h_q)$，而我們手邊有的語言模型分別為：

1. $P_u(w_q|h_q)$，目標使用者 u 的個人語料庫訓練的模型。

2. $P_i(w_q|h_q)$，任意一個我們擁有的其他使用者 i 的個人語料庫訓練的模型。

3. $P_B(w_q|h_q)$，背景語言模型。

為了讓我們的個人化語言模型調適能夠套用現有框架，我們調整式子 2.3 為：

$$\begin{aligned} P^{(u)}(w_q|h_q) &= \alpha^{(u)}P_u(w_q|h_q) + \\ &\quad \beta^{(u)}\Sigma_{i\in U}\lambda_i^{(u)}P_i(w_q|h_q) + \\ &\quad (1-\alpha^{(u)}-\beta^{(u)})P_B(w_q|h_q) \end{aligned} \tag{4.1}$$

其中 U 是除了該目標使用者 u 以外的所有系統已註冊的使用者集合，$\lambda_i^{(u)}$ 則是對應的 u 中每位使用者 i 的的權重，$\sum_{i\in U}\lambda_i^{(u)}=1$。$\alpha^{(u)}$ 與 $\beta^{(u)}$ 則是同時考慮上述三個不同來源的語言模型之間加權的權重。剩下的問題便是如何估計多組權重 $\lambda_i^{(u)}$、$\alpha^{(u)}$、$\beta^{(u)}$ 使得最後的整合模型對該使用者的語言最俱有預測能力。

雖然在過去最常見的做法，是將目標語料庫再切出一組發展語料庫 (development set)，再套用最大期望演算法 (EM：Expectation-Maximization) 去調整參數 $\lambda_i^{(u)}$、$\alpha^{(u)}$、及 $\beta^{(u)}$，以期最佳化該發展語料庫的混淆度 (perplexity)。但這樣的方法在我們現在面對的問題，可以預測的是無法發揮太大個功效，主要原因在於以下兩點：(1) 由於可以被系統觀測到的使用者數目其實是相當多的，但每個使用者可以被觀察到的語料庫相對上比較稀疏 (sparse)，約只有幾百到幾千句不等。根據機器學習 (ML：Machine Learning) 的理論我們可知，用一個擁有成千上萬個參數 (parameter) 的模型去貼合一個過小的訓練語料庫 (training set) 很容易造成過貼合的問題，導致訓練出來的模型缺乏廣泛性 (generalization capability) 而無法在未知的測試語料上有好的表現。(2) 社群網路上搜集下來的個人化語料庫通常包含許多不同的主題。通常，即使是同一個使用者的語料庫，在不同的時間點其討論的主題可能就會完全不同。這跟社群網路大家常用來抒發自己對某個事件的看法的使用習慣有關，因為跨了一段時間該使用者關心的主題可能就不同。這樣的問題導致我們切出來的訓練語料庫與發展語料庫，可能與未知的測試語料庫之間有很大的主題鴻溝，而使得訓練語料庫與發展語料庫不再那麼可信，無法忠實的反映該使用者的未來語言特性。考慮到上述我們遇到的這些困境，儘管再來提出的方法都是本著式子 4.1 的基礎之上，估計權重的方法則不再全然依靠最大期望演算法，而是依照許多不同我們可以觀測到的指標去估計，包含社群網路互動關係 (social relationships)、潛藏式主題模型 (latent topic models)、及使用者網路的圖學

結構 (user graph structure) 等等。

4.1.1 模型調適整體架構

為了解決上述所遇到的問題，這裡我們提出一個新的語言模型調適架構，如圖 4.1 所示。當一個目標使用者 u 利用臉書帳號登入了我們系統之後，我們的系統便獲得了該使用者的存取令牌 (access token)。藉由這個存取令牌，我們的系統便可以開始撈取使用者社群網路上的語料，包含使用者動態牆上的發文、其他使用者的回應、聊天紀錄、以及其他他可以觀察到的網站上公開資訊。使用者本身的語料庫會被我們切成訓練語料庫 (training set， A_u) 以及發展語料庫 (development set， D_u) 兩部分。除了使用者本身的語料庫之外，我們也同時考慮了其他不同的使用者 i 的語料庫 A_i 對該使用者 u 的影響。這些小的個人語料庫，A_u、 A_i， $i = 1, 2, ...$，則由兩種不同的模擬方案 (modeling scheme) 先聚集產生少數幾個中途語言模型 (intermediate LMs)。這兩種不同的模擬方案分別為再來 (1) 章節 4.1.2、章節 4.1.3、及章節 4.1.4 要討論的個人化語料庫加權法 (Personal Corpora Weighting)，主要精神是利用某些使用者之間可以觀察到的指標 (indicator)，對目標使用者 u 與另外其它每個使用者 i 間都計算一個權重，而這個權重便直接拿來對個人語料庫做加權產生一個中途語言模型。(2) 章節 4.1.5 要討論的主題，句子集群方法 (Sentence Clustering)， 則是利用某些指標將句子根據相似程度作分群

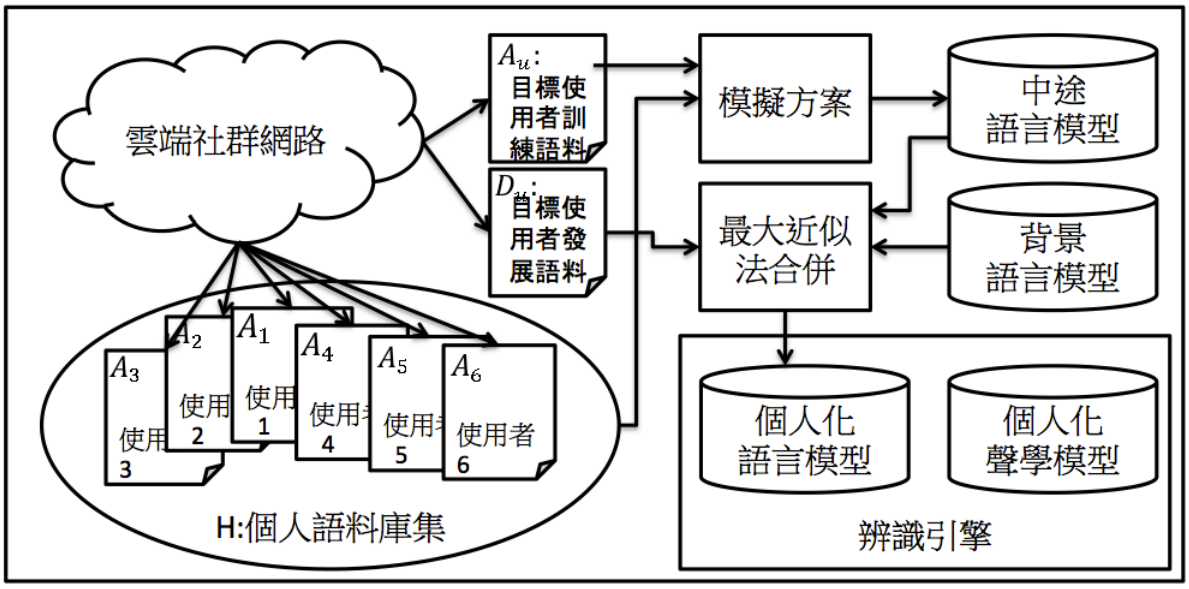


Figure 4.1: 個人化 N 連文法語言模型調適架構

(clustering)，並對每個群集都訓練一個中途語言模型。最後，這些中途語言模型與背景語言模型 (background LM) 之間才會再利用最大期望值演算法，藉由調整幾個語言模型間的權重，以期最小化該使用者的發展語料庫 D_u 混淆度。這樣做的目的主要是為了保留最大期望值演算法能夠根據資料特型估計一組最佳參數的好處，但同時避免在每個語言模型都太小觀察的語料不足時便使用而造成過貼合 (overfitting) 問題。最後，我們將經過最大期望值演算法產生出來的個人化語言模型配置進雲端的個人化辨識系統中，便完成了個人化語言模型的建構。

4.1.2 基於社群網路互動關係之相似度估計

在個人化語料庫加權法中，我們可以採用許多不同的指標來衡量兩個使用者之間的相似程度 (similarity)。我們第一個直覺想到的，便是使用者在社群網路中的互動關係。根據過去一些社會語言學家 (sociolinguistic) [49, 50] 的研究中指出，社群網路上的關係或者是人們在社會上的接觸，都是很重要的語言交換 (language exchange) 指標。人們往往在不知不覺中，將對方的語言使用習慣也學習起來。延續著這個論述，我們做一個假設：兩個使用者倘若在社群網路中顯示為朋友關係，那他們之間可能存在某種共同使用的語言 (shared language usage)；倘若兩個使用者除了是朋友之外，他們的互動也十分密切，如常常互傳訊息、互相留言等等，那他們的共同語言的使用應該更明顯且強烈。

為了模擬這樣的互動關係，我們定義一組社群關係的特徵向量 (feature vector)，$f_j(u,i), j = 1, 2, ...$。這組特徵向量的每個特徵值 (feature value) 是從目標使用者 u 與其他任一使用者 i 過去的互動記錄中抽取。在後面的實驗中我們總共抽取了七個特徵值，列表成表 4.1。 有了這一組特徵向量之後，我們便可以根據這組特徵向

Table 4.1: 使用者的社群網路互動特徵 $f_j(u,i)$

j	描述
1	目標使用者 u 與某使用者 i 的共同好友數目
2	目標使用者 u 對某使用者 i 的留言次數
3	目標使用者 u 從某使用者 i 接收到的留言次數
4	目標使用者 u 對某使用者 i 的按" 讚" 次數
5	目標使用者 u 從某使用者 i 接受到的按" 讚" 次數
6	目標使用者 u 與某使用者 i 共同參加的社團數目
7	目標使用者 u 對某使用者 i 共同按讚的粉絲團數目

量去計算兩個使用者在社群網路上的相關程度 (relevance score)，以式子表示成：

$$R(u,i) = \sum_j b_j \cdot \log(f_j(u,i)) \tag{4.2}$$

$b = \{b_j, j = 1, 2...\}$ 是一組權重向量 (weighting vector)，同樣可以由使用者的發展語料庫調整。而在上式中我們對特徵值取 log 主要原因是考慮到這些特徵與相關度其實不是呈現線性成長的關係，因此採用 log 在特徵值大時壓抑其效果。最後，目標使用者 u 與使用者 i 之間語料的加權權重 $\lambda_i^{(u)}$ 可以寫作：

$$\lambda_i^{(u)} = \frac{R(u,i) + \epsilon}{\sum_i [R(u,i) + \epsilon]} \tag{4.3}$$

ϵ 是一個平滑化參數 (smoothing factor)，主要為了避免 $\lambda_i^{(u)}$ 有 0 的狀況發生。

4.1.3 基於潛藏式主題模型之相似度估計

另外一種估計權重的方法，是直接考慮使用者的個人化語料庫裡面呈現的潛藏主題 (latent topic) 相似程度。雖然有很多種不同的模型方法可以去估計潛藏在文章集合裡的主題分佈 [51]，但基本上做法並不離藉由統計字詞之間共同出現 (co-occurrence) 的關係，將關聯性高的字詞歸類在一起，這套方法我們統稱潛藏主題模型 (latent topic models)。而在過去其實就已經有很多有關語言模型調適的研究 [14, 52, 53]，採用潛藏主題模型將大文章集合細分而作細緻的領域導向 (domain oriented) 語言模型調適。而在這當中，最受大家廣泛使用的、也一直有不錯表現的，便是潛藏狄氏分配 (LDA：Latent Dirichlet Allocation) [54]。本文也主要是採用此潛藏主題模型去估計文章潛藏主題分佈。

潛藏狄氏分配根據機率圖學模型 (PGM：Probabilistic Graphical Model) 可以將其表示成圖 4.2。其中 W 代表詞、Z 代表某個潛藏主題、T 代表該文章裡總共的文字數目，θ 是一個主題對文章的多項分佈 (multinomial topic distribution over document)、α 是一個控制 θ 疏密的超參數 (hyperparameter)，ϕ_k 則是第 k 個主題的文字對主題多項分佈 (multinomial word distribution over k-th topic)、而 β 同樣的是控制 ϕ 疏密的超參數，M 則代表整個文章集合裡的文章數目。它本身為一個生成模型 (generative model)，主要用來模擬語料庫 D 中每一篇文章 d 的生成過

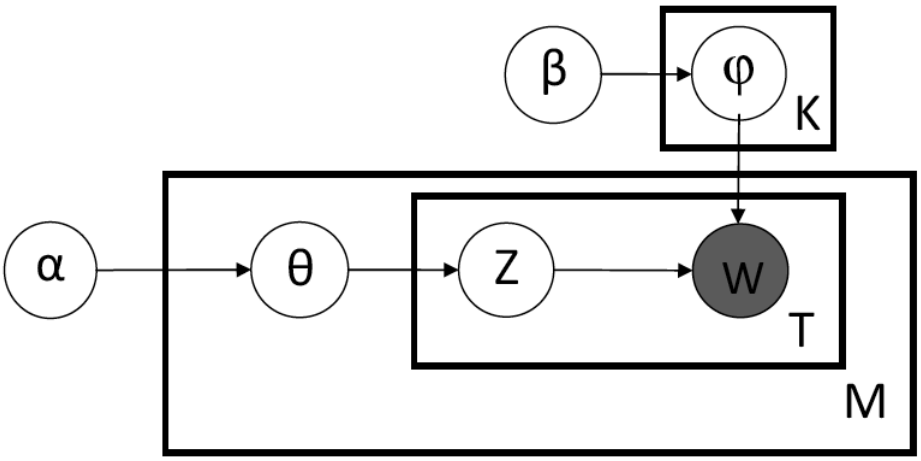


Figure 4.2: 潛藏狄氏分配

程：

1. 挑選一個主題多項分佈 θ_d， $\theta_d \sim$ Dirichlet(α)。

2. 對一篇長度為 T 的文章的每一個詞 W 來說：

   (a) 挑選一個潛藏主題 Z， $Z \sim$ Multinomial(θ_d)。

   (b) 挑選了 Z 之後，便可得該主題下的詞多項分佈 ϕ_Z， $\phi_Z \sim$ Dirichlet(β)

   (c) 挑選一個詞 W， $W \sim$ Multinomial(ϕ_Z)

潛藏狄氏分配的學習過程，主要是調整其內部參數 θ_d 與 ϕ_Z 使其能夠最大化訓練語料庫的機率，其目標函數 (objective function) 我們可以寫為：

$$P(D|\alpha,\beta) = \prod_{d=1}^{M} \int P(\theta_d|\alpha)(\prod_{n=1}^{T_d} \sum_{Z_{dn}} P(Z_{dn}|\theta_d)P(W_{dn}|\phi_{Z_{dn}}))d\theta_d \tag{4.4}$$

超參數 α， β 一般是手動設定，但也有一些數學方法可以在一定迭代次數過後自動最佳化之 [55]。解這個最佳化問題，一般遭遇到的最大麻煩在於將上式 4.4 展開後會發現 θ_d 與 $\phi_{Z_{dn}}$ 會出現耦合 (coupling) 的情形而無法直接最佳化求解。因此，許多替代的近似方案在過去的幾年陸陸續續被提出來 [56, 57]，而在本篇論文裡面我們採取塌陷式吉氏取樣程序 (Collapsed Gibbs Sampling) [55, 56] 求解，其主要概念在於：(1) 首先將 θ_d 與 $\phi_{Z_{dn}}$ 分別積分掉減少未知變數的數目，此步稱為塌陷 (collapse)。(2) 再來利用吉氏取樣程序 (Gibbs Sampling) 從聯合機率分佈 $P(W, Z)$

作取樣。(3) 最後利用採樣來的範例 (example) 做計算，便可以獲得主題的多項分佈 θ_d 與詞的多項分佈 $\phi_{Z_{dn}}$。

而本篇論文在潛藏狄氏分配的實際操作程序上與傳統略有不同：比起輸入一篇篇的文章，我們則是將每個使用者的整份個人化語料庫都當成是一整篇文章當作該模型的輸入。由於潛藏狄氏分配是根據詞與詞在同一篇文章中共同出現 (co-occurrence) 的關係來作為分析的依據，當每篇文章代表一個人的語料庫之後，其分析出來的主題就不再只是傳統我們認知的文章主題關係，而也包含了人們常用的生活用語習慣的分類。由於這類用語都十分地個人化，因此對於我們個人化語言模型的建立會有很大的幫助。最後，如式子 4.1所示，目標使用者 u 與使用者 i 之間語料的加權權重 $\lambda_i^{(u)}$ 可以被理解為兩個使用者個人語料庫主題機率分佈 $\theta^{(u)}$ 與 $\theta^{(i)}$ 之間的相似程度，以餘弦相似度 (cosine similarity) 來表示可以寫為：

$$\lambda_i^{(u)} = sim(u,i) = \frac{\theta^{(u)} \cdot \theta^{(i)}}{|\theta^{(u)}| \times |\theta^{(i)}|}, \tag{4.5}$$

4.1.4 基於隨機漫步演算法之相似度重估

不論基於社群網路互動關係 (章節 4.1.2) 抑或潛藏主題模型相似度 (章節 4.1.3) 去計算權重，其考量都只限於使用者之間兩兩的相似程度，並沒有考量到宏觀的社群網路上不同使用者間的拓樸 (topology) 結構。假設我們能夠考慮這樣的關係，直觀上應該是很有幫助的，例如：在一個社群網路中，使用者 A 與使用者 B 的相似度很高，使用者 B 又跟使用者 C 十分接近。因此，即使我們並不知道使用者 A 與 C 之間的關聯性，我們也可以藉由他們的共同鄰居 B 推測 A 與 C 也十分接近，因此當我們要估計 A 的個人化語言模型時，C 的個人語料庫便可以作為強力的互補素材。

隨機漫步演算法 (Random Walk Algorithm)，主要是一套圖學 (graph) 上的演算法，將節點上的分數不斷藉由與其他節點相連的相似度權重互相傳播以期到達穩定平衡。此方法已經被證明在許多不同的應用上都有很不錯的表現，包含影片搜尋 (video search) [58]、口述語彙偵測 (spoken term detection) [59]、或是語音文件摘要 (spoken document summarization) [60] 等。在這裡我們採用此演算法去幫助考慮網路之間使用者互聯的關係。首先，某個使用者被當做一個節點 (node)，兩兩使

用者 i， j 間的雙向連結 (two-way directed edge) 有一個權重，在此我們初始化為 i， j 間的主題相似程度：

$$sim(i,j) = \frac{\theta^{(i)} \cdot \theta^{(j)}}{|\theta^{(i)}| \times |\theta^{(j)}|}, \tag{4.6}$$

此式即為式子 4.5，其中 $\theta^{(i)}$、 $\theta^{(j)}$ 為潛藏狄氏分配所推論出來的潛藏主題分佈。為了增進計算的效率與降低雜訊的影響，對每個節點我們只保留其前 K 個擁有最高權重的離開連結 (outgoing edge)，並進一步正規化之：

$$\rho(i,j) = \frac{sim(i,j)}{\sum_{j \in O_i} sim(i,j)} \tag{4.7}$$

其中 O_i 是節點 i 以 K 個最高離開連結權重相連的鄰居的集合。而在時間點 $t = 0$，節點 i 上的分數是根據目前的目標使用者 u 來定義，將式子 4.3或 4.5正規化寫為 $\nu_0^{(u)}(i) = \lambda_i^{(u)} / \sum_i \lambda_i^{(u)}$。設定這些以後，隨機漫步迭代更新節點分數的算式便可寫為：

$$\nu_{t+1}^{(u)}(i) = (1-\gamma)\nu_t^{(u)}(i) + \gamma \sum_{j \in I_i} \rho(j,i)\nu_t^{(u)}(j), \tag{4.8}$$

式子 4.8中第一項是節點 i 在時間點 t 時的初始分數，第二項則是在該時間 t 欲更新的分數， γ 是兩項之間的折衷權重 (trade-off weight)，而 I_i 則是節點 i 以進入連結 (incoming edge) 相連的鄰居集合。隨機漫步理論保證式子 4.8在某個迭代數目 N 之後，節點上的分數會漸漸收斂到一組穩定的值。收斂後的分數 $\nu_N^{(u)}(i)$ 則被我們當作是新的權重 $\lambda_i^{(u)}$ 來使用。

4.1.5 語句集群法

另外一種不同於個人化語料庫加權的方法，我們採用語句集群法來產生中途語言模型。我們首先將所有的語料庫都收集在一起形成一個大型的個人語料庫集合 $\{A_u, A_i, i = 1, 2, ...\}$。再來，我們同樣採用章節 4.1.3的潛藏主題模型方法，訓練一個潛藏狄氏分佈模型。根據潛在狄克利里分佈模型，我們可以對整個語料集合的每個句子 s 都推論它的潛藏主題分佈，並根據其推論出來的主題分佈對句子作分

群：

$$C_k = \arg \max_{1 \le k \le L} \theta_k^{(s)} \tag{4.9}$$

L 是潛在主題的數目，k 代表某的主題的索引，C_k 代表第 k 個集合 (cluster)，$\theta_k^{(s)}$ 代表該句子 s 的潛在主題機率分佈中的第 k 個主題的機率值。根據分群出來的每個句子群集，我們便可以訓練一個該群集的語言模型 $P_k(w_q|h_q)$。最後，給定群集語言模型及背景語言模型 $P_B(w_q|h_q)$ 之後，個人化的語言模型 $P^{(u)}(w_q|h_q)$ 便可以根據下式估得：

$$P^{(u)}(w_q|h_q) = \lambda_B^{(u)} P_B(w_q|h_q) + \sum_{k=1}^{L} \lambda_k^{(u)} P_k(w_q|h_q) \tag{4.10}$$

$\lambda_B^{(u)}$ 與 $\lambda_k^{(u)}$ 是各別模型的權重，$\lambda_B^{(u)} + \sum_k \lambda_k^{(u)} = 1$。它們可由 (1) 吉氏取樣法對目標使用者發展語料庫 D_u 作採樣並估計各個主題的機率而得，或 (2) 使用最大期望演算法最佳化目標使用者發展語料庫 D_u 的混淆度而來。

4.2 個人化 N 連文法語言模型評估

為了評估本論文提出之 N 連文法語言模型個人化成果的好壞，我們設計了幾個實驗來比較個人化與否的語言模型表現的差異。以下為實驗設定。

4.2.1 實驗設定

誠如章節 3.2 所言，我們建構了一個具備語音界面的社群網路瀏覽器應用程式，並利用該系統綁定了使用者的臉書帳號，並向使用者要求爬取其社群網路上的資料作為語言模型調適個人化語料庫的來源。在初期實驗裡總共有 21 個使用者登入並允許了系統獲取個人語料庫的權限。這 21 個使用者在後來的實驗中便是我們的主要實驗對象，每一個即稱為目標使用者 u (target user)，而我們的目的便是要對每個目標使用者都建立一套個人化語言模型並評估其效能。藉由這套群力模式的機制，我們的系統便可以獲得這 21 個目標使用者的個人語料庫以及該社群網路上其他公開給這 21 個目標使用者的資訊。因此我們在初期實驗的資料收集裡，總共收集到 21 個目標使用者語料、其他 12365 個匿名使用者語料、以及這些使用

者之間社群網路上互動的資料，包含共同朋友數、互相留言數、互相按讚數、共同參加社團數，及共同喜歡的粉絲頁數目等等。從這些使用者中，總共被存取下來的句子數目共有 45 萬句，在經過前處理 (preprocessing) 後被留下的當作實驗語料的句子共有 28 萬句。每個使用者句子的數目範圍很大，在 1 到 2943 之間，平均 23.12 句，每句平均包含 12.12 個詞 (中文或英文)。每個使用者平均擁有 250 個朋友。由於這些統計數據只根據 21 個目標使用者所能觀測到的資料為主，因此未知數據的問題 (missing data problem) 十分嚴重。另外，我們在這組語料庫中還觀察到了以下現象：(1) 語料庫主要來自使用者的牆上發文 (wall post)，但這類文章通常只是代表在該時間點該使用者的部分想法，與上下發文的主題通常不會有太大的相依性 (dependecy)，亦即主題具有十分高的不連續性 (discontinuity)。(2) 在同一個使用者的語料庫中，常常有一些詞彙或片語會一再重複出現，這些通常就代表了該使用者的口頭禪 (signature phrase)。因此個人化的語言模型倘若能好好模擬這種口頭禪現象，相信能有很大的進步空間。

實驗中，對每個目標使用者 u 而言，我們會將他的個人語料庫切成三個部分：

- (1) $3/5$ 為訓練語料，主要是作為訓練語言模型之用，
- (2) $1/5$ 為發展語料，作為擁有多個模型之後調整權重之用，
- (3) $1/5$ 為測試語料，作為最後評估模型表現的語料。

背景語言模型的訓練，則是由另一批搜集自另一個社群網站噗浪 (Plurk) 的語料庫為主，共包含 2.5 億多個句子，中英夾雜，比例約為 9:1。N 連文法語言模型的訓練與調適我們採用知名的 SRILM 工具 [61]，平滑化則主要採用強化聶氏平滑法 (Modified Kneser-Ney Smoothing) [26]。

一般衡量語言模型的指標主要有兩個，一個是計算其在測試語料 $T = \{w_1, w_2, ...w_N\}$ 上的混淆度 (PPL：Perplexity)，亦即該模型 $P_M(w_q|h_q)$ 對詞的平均預測能力的倒數，寫作：

$$PPL_T(P_M) = \frac{1}{(\prod_{i=1}^{N} P_M(w_i|h_i))^{\frac{1}{N}}} \tag{4.11}$$

此部分我們將在 4.2.2 節做深入分析。另一個則是直接計算它對在某個應用上的

表現，例如語音辨識的正確率 (Accuracy) 或錯誤率 (WER：Word Error Rate) 的影響，我們將在 4.2.3節中作討論。

4.2.2 混淆度實驗結果與結果分析

圖 4.3 為混淆度對考慮的使用者語料庫數目作圖。縱軸為混淆度，橫軸為其語料庫加入參考的使用者數目，加入以權重越高者優先考慮。不同的顏色代表不同權重計算方式：相等權重、社群互動關係、主題模型分別跑在單連文法及三連文法語料上。四筆數據都顯示出，當加入的使用者數目很少的時候 (圖表左半部)，混淆度下降的幅度十分可觀。這證明了個人語料庫與背景語料庫之間的不匹配相當嚴重，儘管我們採用的同是來自於社群網路的資料來源，Plurk 與 Facebook。同時，加入考量的使用者語料庫越多 (圖表右半部)，混淆度仍是下降，只是下降幅度比較不再那麼大，代表其實每個個人語料庫其實還存在相當大的數據稀疏 (data sparseness) 的問題，因此或許要更多不同的個人語料來彌補這個問題，儘管這些使用者可能跟目標使用者很不相似。很明顯的，精細挑選權重計算的方式會帶來比較好的效果 (社群關係, 潛藏主題 > 相等權重)。其中，利用社群網路的方式計算權重似乎在加入使用者數目少的時候比較有效 ($m = 10 - 200$)，但是當加入的使用者多的時候就變成計算潛藏主題相似度的方法較優 ($m > 1000$)。主要原因可能在於使用者一般的朋友數目都是在幾百左右，當數目只有幾百個的時候我們可以

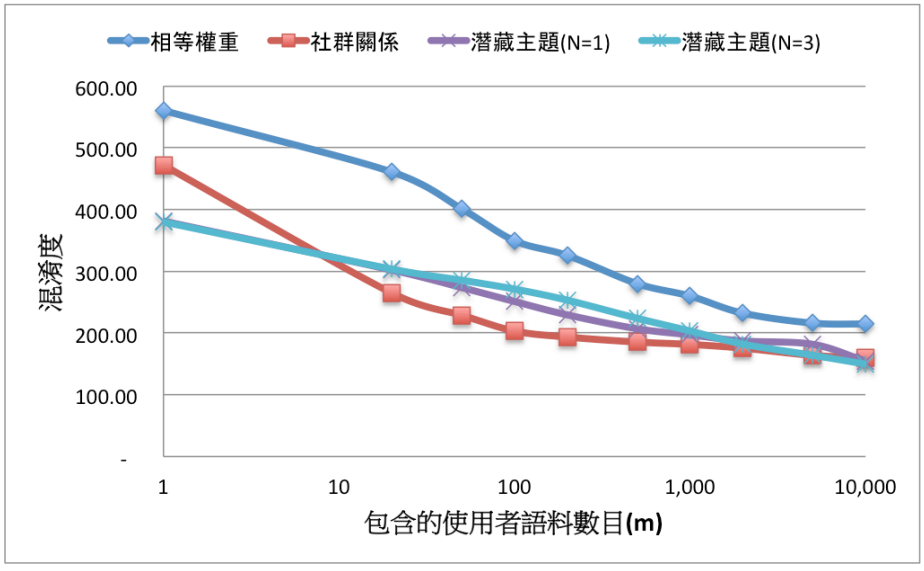


Figure 4.3: 混淆度對考慮的使用者語料庫數目作圖

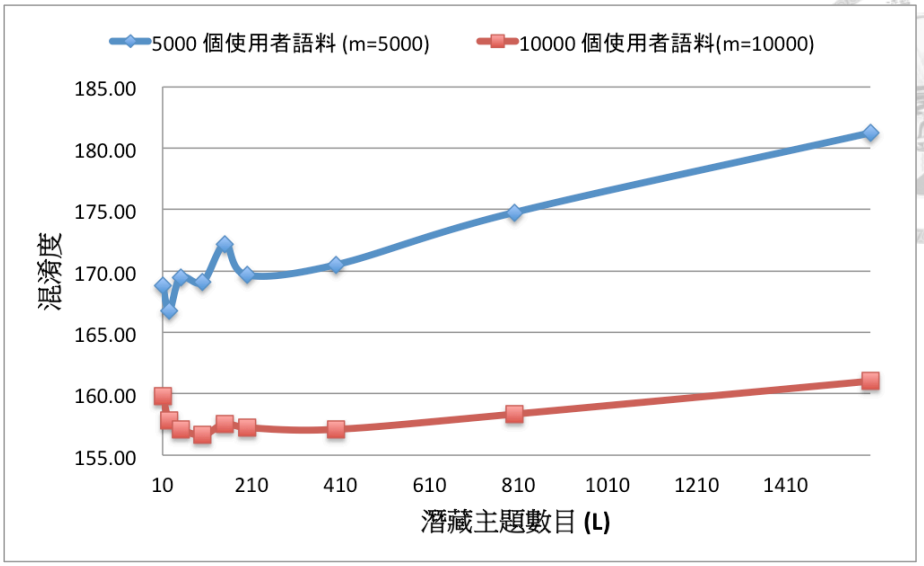


Figure 4.4: 混淆度對潛藏主題數目作圖

利用目標使用者與朋友們之間的互動精確的計算出每個人應有的權重。但當使用者數目超出了一般朋友的數目範圍之後，社群網路的資訊就不足以給予一個合理的權重，因此便只能夠仰賴直接計算語言上的相似度了。

由於在潛藏主題模型中，潛藏主題數目的選擇一直以來都是一個很大的問題，因此圖 4.4 則是我們測試混淆度對潛藏主題數目作圖的結果。縱軸為混淆度，橫軸為潛藏主題的數目，不同的顏色分別代表考慮的使用者人數。從圖中我們可以發現，潛藏主題數目倘若挑選在一定範圍內 (約 30 – 200)，對於實驗的混淆度表現不會有太大的影響，這項觀察在考慮不同的使用者數目時結果是一致的。

圖 4.5中呈現了混淆度分別對不同目標使用者作圖的結果。縱軸是混淆度，橫軸代表 21 個不同的目標使用者，不同的顏色代表四種不同的做法 (modeling scheme)。 我們可以發現不同的使用者所產生的語料庫，有十分不同的混淆度結果，代表模型對不同使用者說話的方式預測能力不一。不同的模擬方案基本上都可以減少使用者語料的混淆度，增加預測的準確性，但效果不一，有優劣之分。以本實驗結果而論，語句群集法並採用最大化期望值估計權重的方法 (4.1.5) 有好的混淆度表現，其次為潛藏主題模型 (4.1.3) 再加隨機漫步演算法重估權重 (4.1.4)，再來才是社群網路 (4.1.2) 及相等權重方法。

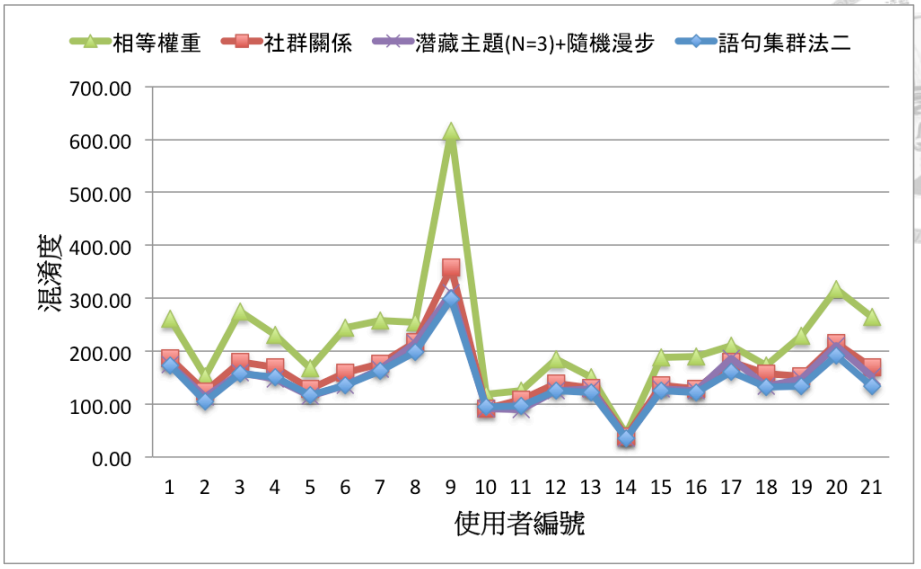


Figure 4.5: 混淆度分別對不同目標使用者作圖

4.2.3 語音辨識實驗結果

本節將介紹我們提出的個人化 N 連文法語言模型在語音辨識上的成效。為了對提出的個人化語言模型做初步的實驗測試，我們從 21 個目標使用者的測試語料庫中挑出其中最長的 1000 句作為我們語音辨識的目標。這 1000 句共由兩個男性的研究生利用我們安卓手機上的應用程式介面錄音並上傳到伺服器端做辨識。由於一些錄音成效的問題最後只採用了其中 840 句，後來的實驗數據中的準確率 (Accuracy) 便是這 840 句平均的準確率結果。聲學模型的訓練、調適，及最後的辨識解碼都是採用 HTK [62] 工具。聲學模型的部分，中文的三連音素 (triphone) 是根據聲碩麥克風語料庫訓練而得，其中包含 37 個中文的音素。而英文的三連音素模型則是根據中研院 (Academic Sinica Taiwan) 的英文語料庫訓練的，包含 35 個英文音素。兩個語料庫都包含上百個語者，因此為語者不特定 (user independant) 模型。在解碼的參數設定方面，聲學模型與語言模型的權重分別設定為 0.5 與 5.0，解碼的搜尋寬度 (beamwidth) 為 100。在聲學模型調適上，我們採用了最大相似度線性迴歸法 (MLLR)。

圖 4.6 便是我們提出的不同個人化 N 連文法語言模型方法在語音辨識上的實驗結果。縱軸為辨識準確率，橫軸為提出的不同個人化 N 連文法語言模型方法，對於各個語言模型，我們同時也提供了語者不特定與語者調適不同聲學模型上的測試結果。首先我們可以觀察到，不管有沒有採取語者調適，個人化語料庫可以提

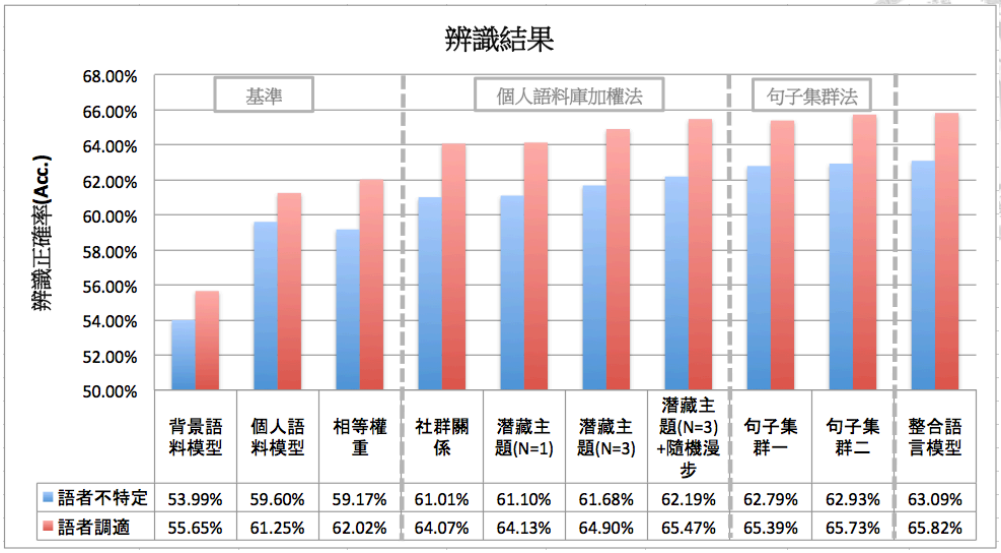


Figure 4.6: 個人化 N 連文法語言模型在語音辨識上的實驗結果

供相較於純粹用背景語言模型約 5.6% 的穩定進步量 (個人語料模型 > 背景語料模型)。倘若我們再把更多的個人語料庫加進來，即使不仔細去計算每個語料庫的權重，我們仍然可以得到進步量 (相等權重 > 個人語料模型)，這主要是因為更多的語料彌補了資料稀疏的問題。 到此這三筆結果可以被視為實驗的基準 (baseline)。在圖 4.6中左則是我們提出來的個人化語料庫加權法 (Personal Corpora Weighting)，包含各種不同計算權重的方式，大致上都可以勝過基準點約 2.0% 或更多。其中利用社群網路關係 (社群關係) 的方法大致與潛在主題模型跑在單連文法語料庫上潛藏主題 *(N=1)* 差不多，但潛在主題模型跑在三連文法語料庫 (潛藏主題 *(N=3)*) 的結果大致勝出跑在單連文法語料庫潛藏主題 *(N=1)* 的結果約 0.6%。這可能是因為跑在三連文法的語料庫上表現出更多俱有語言模型訓練時的統計意義。倘若我們將目前最好的結果 (潛藏主題 *(N=3)*)，再利用隨機漫步演算法對其權重再度重新估計，我們會額外再得到 0.5% 的辨識進步量 (潛藏主題 *(N=3)*+ 隨機漫步)，主要原因在於隨機漫步演算法又額外考慮了整個網路的宏觀結構。在圖 4.6中右則是語句集群法，包含吉氏取樣 (語句集群法一) 及最大化期望演算法 (語句集群法二) 兩種不同的權重估計法。雖然最後三種方法 (潛藏主題 *(N=3)*+ 隨機漫步 *vs* 語句集群法一 *vs* 語句集群法二) 看似沒有十分大的優劣之分，但語句集群法二 仍然是相對上較好的，主因可能是因為若自由度 (degree of freedom) 足夠時，最大化期望演算法仍然是比較好的估計參數法。最後，考量到社群網路關係、潛藏主題相似度、及語句集群法 (社群關係, 潛藏主題 *(N=3)*+ 隨機漫步, 及語句集群法二)，似乎參考

了不同的使用者相似度上的關係，我們將這三個方法所產生的中途語言模型放在一起，用最大化期望演算法一起估計權重並把將這多個模型整合。而這個整合的結果 (整合語言模型) 也似乎成功地展現了各個模型的優點，勝過了所有單獨模型的結果約 0.1% 到 0.15%。

4.3 遞迴式類神經網路語言模型 (RNNLM) 的個人化

在上一節我們介紹了許多種針對 N 連文法語言模型的個人化方法，在這一節中我們要探討另外一種語言模型，也就是類神經網路語言模型 (NNLM：Neural Network Language Model) 的個人化方法 [48]。在這一個節中我們特別著重在遞迴式類神經網路語言模型 (RMMLM：Recurrent Neural Network Language Model) 的個人化，除了它是最近拿來做語言模型最熱門的主題之外，也在於其理論上能記憶任意長度的歷史 (arbitrary length history) 所帶來對於序列極高的預測能力，而這正是傳統前饋式類神經網路 (FFNN：Feed-Forward Neural Network) 所欠缺的優點。

4.3.1 模型結構

如圖 4.7所示，遞迴式類神經網路的模型主要由三個層級組成：

1. 輸入層 (input layer)，大小為辭典大小 (lexicon)，採用 1-of-N 編碼 (1-of-N encoding) 形式，輸入為前一個詞在辭典裡的索引 $w(t)$，只有該索引位置值為一，其餘都是零。

2. 輸出層 (output layer)，大小同為辭典大小，同樣採用 1-of-N 編碼 (1-of-N encoding) 形式，輸出為該模型對下一個字出現機率分布的預測 $y(t)$。

3. 潛藏層 (hidden layer)，或稱上下文層 (context layer)，通常維度較小，代表在該時間點模型保存的上下文資訊 $s(t)$。 除了與輸入層、輸出層各有一組權重矩陣 (weight matrix)， $\mathcal{W}$ 與 $\mathcal{O}$ 外，與上一個時間點的上下文向量 $s(t-1)$ 也存在一組矩陣 $\mathcal{S}$，其用意即是模擬上下文間相依的關係。

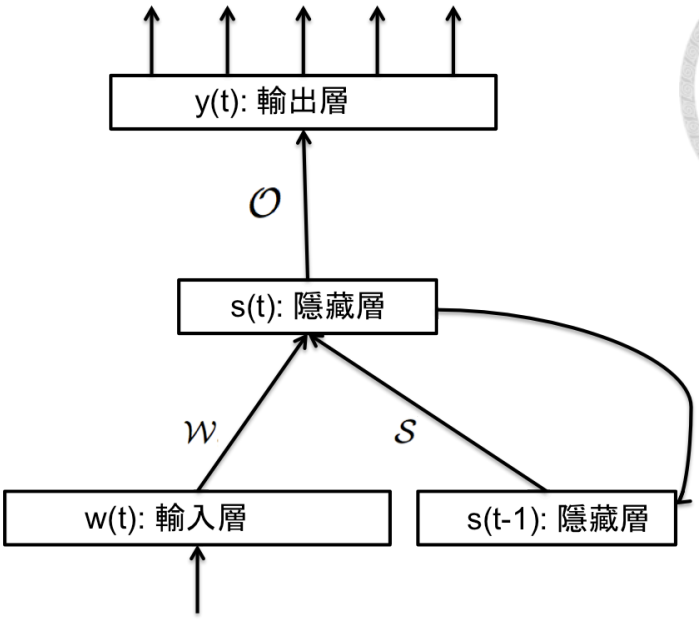


Figure 4.7: 遞迴式類神經網路語言模型

在定義了這些向量與矩陣之後，輸入層、潛藏層、及輸出層的關係可用下列二式表達：

$$s(t) = f(\mathcal{W} \cdot w(t) + \mathcal{S} \cdot s(t-1)) \tag{4.12}$$

$$y(t) = g(\mathcal{O} \cdot s(t)) \tag{4.13}$$

而 $f(\cdot)$ 與 $g(\cdot)$ 分別為羅輯函式 (logistic function) 與 soft-max 函式：

$$f(z) = \frac{1}{1+e^{-z}} \tag{4.14}$$

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}} \tag{4.15}$$

在整個結構裡面，其主要賣點在於潛藏層裡的遞迴式結構 (recurrent structure)。藉由該遞迴式結構，一方面可以免去如前饋式網路，必須設定明確的前後文的長度，去模擬長度本質上未知的上下文的缺點；另一方面藉由不斷的遞迴資訊，可以模擬任意長度的上下文關係。而這樣子的模型訓練，包含了調整所有的權重矩陣 W、S、O 使得該模型對訓練語料的預測能力被最佳化。而一般最常使用的，便是沿時間反向傳播法 (BPTT：Back Propagation Through Time)。

4.3.2 最佳化演算法 - 沿時間反向傳播 (BPTT)

要理解沿時間反向傳播演算法，要先理解其基本形式：反向傳播演算法 (Back Propagation)。反向傳播演算法包含三個部分：

1. 給定一筆訓練範例 (training example) 及現階段的模型參數組合，從輸入層向輸出層做順向傳遞 (forward pass)，輸出現階段模型預測結果 $y_j, 0 < j < |V|$。

2. 根據第一步中模型的輸出結果 y_j，與真正的結果 d_j 間計算誤差值，在語言模型中誤差函數 (error function) 為交叉熵 (cross entropy)：

$$E = \sum_j d_j \log y_j \tag{4.16}$$

3. 根據第二步算出來的誤差函數，計算一階倒數 (first derivative)，調整近來的權重 (incoming weight)，並重複將這個誤差訊號利用連鎖法則 (chain rule) 往後傳遞，直到輸入層為止。

我們利用圖 4.8作為一個範例來解釋該演算法的向後傳遞過程。假設 y_j 與 z_j 分別是是輸出節點 j 的輸出訊號與輸入訊號， y_i 是潛藏節點 i 的輸出訊號。其中 $y_j = f(z_j)$ 經過一個激活函數 (activation function) $f(\cdot)$ 轉換。假設已知誤差函數為式 4.16的形式，因此我們可以很輕易計算它對 y_j 的一次偏微 $\partial E/\partial y_j = d_j/y_j$。根據連鎖法則，我們又可知：

$$\frac{\partial E}{\partial z_j} = \frac{\partial y_j}{\partial z_j}\frac{\partial E}{\partial y_j} = f'(z_j)\frac{\partial E}{\partial y_j} \tag{4.17}$$

且：

$$\frac{\partial E}{\partial y_i} = \sum_j \frac{\partial z_j}{\partial y_i}\frac{\partial E}{\partial z_j} = \sum_j w_{ij}\frac{\partial E}{\partial z_j} \tag{4.18}$$

故我們便可推得誤差函數 E 對權重 w_{ij} 的一階導數：

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial z_j}{\partial w_{ij}}\frac{\partial E}{\partial z_j} = y_i\frac{\partial E}{\partial z_j} \tag{4.19}$$

而此值便被拿來作為梯度下降法 (gradient decent) 中更新權重 w_{ij} 更新的依據。

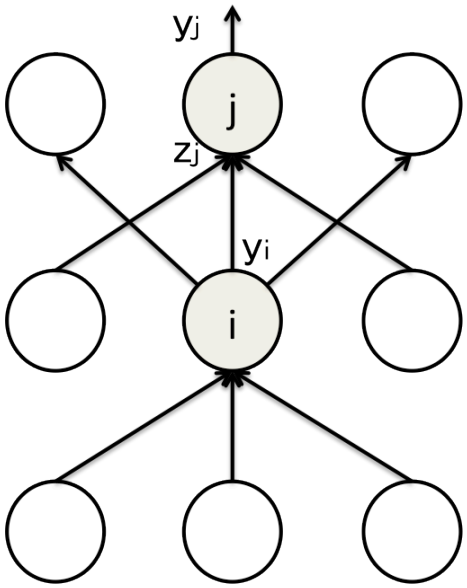


Figure 4.8: 反向傳播演算法範例示意圖

而沿時間反向傳播演算法 (BPTT) [63] 與正規的反向傳播演算法唯一不同之處便是，在做模型訓練之前，必須根據設定的展開時間參數，將潛藏層往前展開 N 層，如圖 4.9所示，為將時間展開 3 層之後的結果。展開之後其實相當於訓練兩個前饋式類神經網路，一為原來的上半部，包含輸出層、原始潛藏層、及輸入層；另一個為下半部，包含所有沿著時間展開的潛藏層及其對應的輸入層。另外，由於模型隱含序列的特性，因此在訓練的過程中必須保持序列的順序一筆一筆餵入作訓練。訓練後各個時間點的潛藏層與潛藏層間的權重矩陣將被取平均，以期讓各個時間點的權重都相同。

4.3.3 上下文相關的遞迴式類神經網路與詞輔助特徵

除了由輸入層直接提供字的索引作為特徵值之外，仍然有很多有用的詞的特徵，可以拿來作為我們判斷下一個字詞應該為何的依據，例如詞性 (POS tag)、詞義 (word lemma)、以及該詞常出現的社交場合 (social situational feature) [64] 等等。為了提供模型這些輔助的詞特徵 (auxiliary word features)，上下文相關的遞迴式類神經網路模型 (Context-dependent RNNLM) [33] 被提出，如圖 4.10。其特色為修改原有的遞迴式類神經網路結構，在原有的架構下，加入了一層輔助特徵層 (feature layer)，並對輸出層與隱藏層都各有一組權重矩陣相連接。這樣一來，藉由在特徵層餵入全域的資訊 (global information)，例如該句子的潛藏主題分佈向量，我們

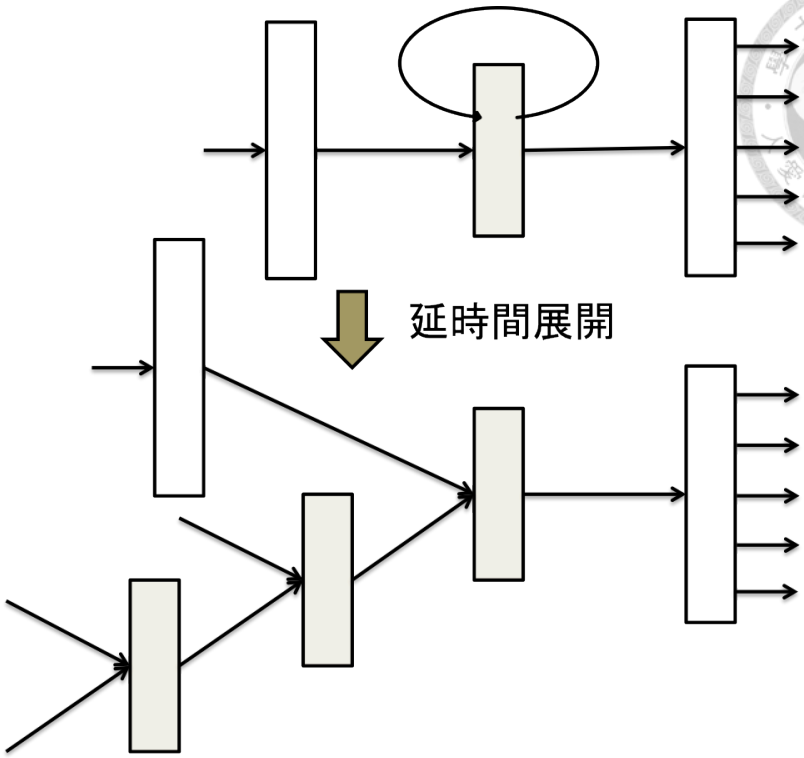


Figure 4.9: 沿時間反向傳播演算法將潛藏層沿時間展開 (unfold through time)

就可以減小梯度漸失 (vanishing gradient problem) [65]，這個長久以來存在，在訓練遞迴式類神經網路模型的時候會遇到的問題。在這樣新的架構下，我們只要對式 4.12 與式 4.13 略作修改，便可以獲得新的網路各層的數學關係：

$$s(t) = f(\mathcal{W} \cdot w(t) + \mathcal{S} \cdot s(t-1) + \mathcal{F} \cdot f(f)) \tag{4.20}$$

$$y(t) = g(\mathcal{O} \cdot s(t) + \mathcal{G} \cdot f(t)) \tag{4.21}$$

$f(\cdot)$ 與 $g(\cdot)$ 同樣分別為羅輯函式 (logistic function) 與 soft-max 函式。故此，模型的訓練同樣藉由沿時間反向傳播演算法，調整權重矩陣 $\mathcal{W}$、$\mathcal{S}$、$\mathcal{O}$、$\mathcal{F}$、$\mathcal{G}$ 使得該模型對訓練語料的預測能力被最佳化。

4.3.4 三步驟調適機制

類神經網路語言模型的調適機制，相對於 N 連文法語言模型，在現有的學術文獻上仍是較欠缺的。目前比較在類神經網路語言模型調適這塊領域有名的的研

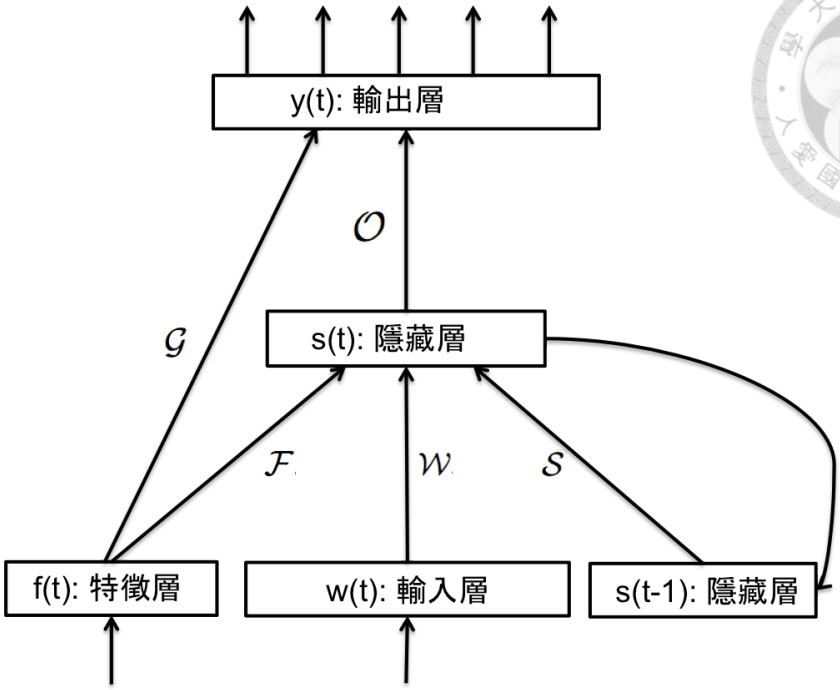


Figure 4.10: 上下文相關的遞迴式類神經網路結構

究，可以算是劍橋大學做在前饋式類神經網路語言模型上的方法 [29, 66] 為主。其提出來的調適方法為，在前饋式類神經網路語言模型的架構之下 (圖 2.3)，於投射層與隱藏層間再加入一層調適層 (adaptation layer)，其激活函數為一線性轉換 (linear transform)。當模型在訓練時，背景語料被拿來訓練整個模型的權重，此時忽略該調適層；當調適開始時，模型一樣做反向傳播，惟只有調適層的權重被調整，其他層的權重則維持不變。如此的做法好處在於，這層調適層可以作動態抽換，以根據需要的辨識目標領域作動態調適，很適合領域導向的語言模型調適。但這樣的做法在這裡卻有一些潛在的問題：(1) 該線性調適層無法模擬較複雜的資料結構，尤其是語言個人化的特性。人類語言十分複雜，尤其每個人的用字遣詞特性更是不同，單純一層的線性轉換難以模擬這樣的複雜度。(2) 梯度漸失的問題主要因為遞迴式網路對時間作展開後，需要訓練的層數太多，造成梯度在傳遞到較早的潛藏層時容易出現梯度過小或過大的情形。這樣的狀況導致多加調適層在遞迴式網路中極有可能使得網路變得不穩定，越容易出現梯度漸失的問題。因此考量到以上兩點，我們提出在遞迴式類神經網路語言模型中，不另外加調適層而直接根據調適語料微調模型每個層的權重。這樣一來，同時調適多個層級不但使得模擬個人化的能力較強，也不另外增加訓練的困難。

給定目標使用者 u、背景語料庫 B、目標個人化語料庫 A、以及目標使用者的社群網路上朋友的語料庫集合 C，在同時考量到計算複雜度與資料數量可行性之下，我們提出以下的三步驟個人化調適機制：

1. 將背景語料庫 B 切成訓練語料 T_b(training set) 與驗證語料 V_b(validation set)，利用 T_b 作為最佳化的目標提升模型對其的預測能力，並同時利用 V_b 控制訓練的紀元數 (epoch)，訓練一個背景遞迴式類神經網路語言模型。訓練得的背景模型包含 $\mathcal{W}_0$、$\mathcal{F}_0$、$\mathcal{S}_0$、$\mathcal{G}_0$、與 $\mathcal{O}_0$ 等五組語者獨立的權重矩陣。

2. 給定目標使用者的個人化語料庫 A，同樣將其切分成訓練語料 T_a 與驗證語料 V_a。同樣地，T_a 作為利用沿時間反向傳播 (BPTT) 調整參數時的最佳化目標，V_a 控制訓練的紀元數。此步驟後獲得的模型參數 $\mathcal{W}'$、$\mathcal{F}'$、$\mathcal{S}'$、$\mathcal{G}'$、與 $\mathcal{O}'$ 已經帶有初步個人化的色彩。

3. 給定目標使用者朋友的個人化語料集合，我們將其串成一個完整的語料，並直接將其當作訓練語料。此時控制訓練紀元數的驗證語料同樣由 V_a 擔任，確認訓練的過程除了降低朋友們語料庫的混淆度外也同時降低了目標使用者的語料混淆度。此步之後產生最終參數組合：$\mathcal{W}''$、$\mathcal{F}''$、$\mathcal{S}''$、$\mathcal{G}''$、與 $\mathcal{O}''$

我們認為，以上的三步驟調適機制，俱有以下許多優點：

- 首先，遞迴式神經網路語言模型的計算 (computation overhead) 是眾所皆知相當高的。而我們第一步訓練出的背景語言模型，由於預先學習了許多共同的上下文關係，所以往後的訓練只需要根據特別的個人化上下文特徵作參數微調，因而省下許多不必要的多餘計算。

- 第二，雖然拿來做調適的語料相當少 (約幾百至幾千句)，但由於類神將網路語言模型將詞與句子打散入網路中形成分散式的表達形式 (distributed representation)，這樣子的表達形式被認為具有將少量訓練放大的效果。原因在於同樣一個句子在分散的連續空間中的訓練，其實等同告訴了模型許多相似句子的上下文特性。

- 第三，不像我們在本章前半部敘述的 N 連文法語言模型個人化的方法需要以很多經驗法則來計算朋友語料間權重，我們在這裡第三步的調適機制，利

用 V_a 的掌控讓整個訓練過程完全資料導向 (data-driven)。我們不必再擔心怎麼挑選每個權重，而是讓模型自己根據資料特性決定。

- 最後，同上述，考量到遞迴式的結構造成梯度漸失的問題，以及多餘的特徵層讓網路變得更加複雜，我們為了保持網路結構簡單，故不再另外加入調適層，以增進訓練的穩定與效率。另外當然也是考量了單層線性轉換恐無法模擬複雜個人化語料的特性。

4.3.5 使用者導向的詞特徵

誠如章節 4.3.3中圖 4.10所示，上下文相關的遞迴式類神經網路語言模型 (CD-RNNLM) 藉由新加入的特徵層，作為字詞輔助特徵向量的輸入點，使得模型根據已知的上下文關係估計下一個字詞的機率時，也同時可以參考更多的除了該字詞以外的其他額外資訊。在一些過去的研究成果 [64] 中指出，加入額外的附加資訊，如：詞性 (POS tags)、詞義 (word lemma)、及主題資訊 (topic information) 等等，不只在混淆度方面有所下降，也有效降低了語音的辨識錯誤率。在本節裡面，除了上述提到的，詞本身已知的特徵之外，我們希望更加著重於這些詞在不同的使用者之間被使用的分佈狀況。

本論文提出兩種使用者導向的詞特徵 (user-oriented lexical features)：

1. 第一，我們著重在傳統的人口組成資訊 (demographic information) 對語言使用的影響。這樣子的假設其實是十分合理的：想像如果目標使用者是一個男孩，那麼在男孩言談中討論到超級英雄 (superhero) 這個詞彙的頻率理論上應該是比女孩相對上高很多。再者，時下年輕人愛好的網路用語及新式詞彙，對於老一輩的人肯定就相當陌生。儘管這層關係一直以來對我們來說都很直觀，但是這樣的語料在過去卻很缺乏而因此缺少研究。幸運的是，這些資訊很容易在現今的社群網路上被收集、觀測，因此對我們想要個人化語言模型來說無疑是一大福音。假設給定一組個人化語料庫及一個人口組成指標 (demographic indicator)，如性別、年齡、語言等等，而可以將 p 個使用者切

分成 K 個群體，則我們便可以根據每個詞 w 計算它在群體裡的機率分佈：

$$P(w|C_k) = \frac{\sum_{p \in C_k} n(w,p)}{\sum_k \sum_{p \in C_k} n(w,p)} \tag{4.22}$$

其中 $n(w,p)$ 是詞 w 出現在 p 這個使用者個人語料庫裡的次數。而後這個機率分佈 $P(w|C_k)$ 就可直接被當作是輸入層的輔助特徵使用。我們可以串聯很多組不同人口組成指標所產生的特徵向量來提供模型更豐富的輔助資訊。

2. 另外一種思維我們可以想成，其實很多時候，對於我們語言的使用，人口組成還是比較次要的。相對上，該使用者的興趣、嗜好、偶像、他所在的社群、朋友圈等等，可能才是塑造其使用語言的特性。這些理論我們可以在很多相關領域的社會科學研究上看到 [49, 50]。但想要模擬這些關係其實是很難的，主要難點有二：(1) 這些關係很多往往都很抽象 (arbitrary)，要用數學直接模擬很多時候不是件容易的事。(2) 同時模擬越多的特徵就需要約高的向量維度，而這樣子高維度的向量便會對已經很難訓練的遞迴式類神經網路語言模型帶來更高的複雜度，使得訓練愈加難以實行。為了克服這些困難，我們改採用潛在因子模型 (latent factor model) [67, 68] 來直接模擬資料裡展現出來的不同層面特性。首先，我們建立一個使用者對詞單元的矩陣 M，矩陣中的每個元素 M_{ij} 即是詞單元 w_j 出現在使用者 p_i 中語料的次數，為了簡單起見，在這裡的詞單元我們只考慮單連文法 (unigram)，也就是詞。擁有這個矩陣之後，為了找出其中隱含的潛在因子 (latent factor)，奇異值分解 (SVD：Singular Value Decomposition) 便可被採用來對該矩陣做拆解分析：

$$M = U \cdot \Sigma \cdot V^T \tag{4.23}$$

而後我們保留前 K 個擁有最大特徵值 (eigenvalue) 的維度，我們便可以將使用者與詞單連文法同時降維到這個 K 維的潛在因子空間，每個維度都代表了使用者及詞單連文法所擁有的某種潛在因子。而後這些詞單連文法的潛在因子向量便可作為特徵向量輸入網路特徵層。雖然此方法乍看之下很像潛藏語義分析 (LSA：Latent Semantic Analysis) [68]，但其中確有根本意義上的不同。潛藏語義分析是分解文章對詞單連文法矩陣以期找出文章與詞單連文法

當中潛在的主題關係，而這裡提出的方法則是分解使用者對詞單連文法的矩陣，理論上可以找出任何兩兩間可能存在的潛在關聯性。

4.4 個人化遞迴式類神經網路語言模型評估

為了評估本章節提出的遞迴式類神經網路語言模型個人化成果的好壞，我們設計了幾個實驗來比較個人化與否的語言模型表現的差異，其中亦包含了與 N 連文法語言模型個人化成果的比較。以下為實驗設定。

4.4.1 實驗設定

實驗的整體基本設定與章節 4.2.1 相去不遠，唯一差別在於進行了第二次收集資料之後，我們擁有了更多的個人化語料庫，因此許多語料庫的統計數據有了更動，但其他未提及的部分基本與章節 4.2.1相同。

在第二次的資料收集裡總共有 42 個使用者登入並允許了系統獲取個人語料庫的權限。這 42 個使用者在後來的實驗中便是我們的主要實驗對象，稱為目標使用者 (target user) u，而我們的目的便是要為每個目標使用者都建立一套個人化語言模型並評估其效能。藉由這套群力模式的機制，我們的系統便可以獲得這 42 個目標使用者的個人語料庫以及該社群網路上其他公開給這 42 個目標使用者的資訊。因此我們在第二次資料收集裡，總共收集到 42 個目標使用者語料、其他約 93000 個匿名使用者語料、以及這些使用者之間社群網路上互動的資料，包含共同朋友數、互相留言數、互相按讚數、共同參加社團數，及共同喜歡的粉絲頁數目等等。從這些使用者中，總共被存取下來的句子數目共有 330 萬句，在經過前處理後 (preprocess) 被留下的當作實驗語料的句子共有 240 萬句。每個使用者句子的數目範圍很大，在 1 到 8566 之間，平均 25.7 句，每句平均包含 10.6 個字 (中文或英文)。每個使用者平均擁有 238 個朋友。

斷詞採用中研院 (Academic Sinica Taiwan) 的斷詞系統 [69, 70]。其中詞典被我們重新根據新的語料庫做過調整，以語料庫中出現頻率最高的 18000 個英文單字與 46000 個中文詞組成。切分訓練語料、驗證語料、測試語料的比例同樣為 3 : 1 : 1。背景語言模型的訓練，仍是由社群網站噗浪 (Plurk) 的語料庫為主，共

包含 2.5 億多個句子，中英夾雜，比例約為 9:1。N 連文法語言模型的訓練與調適我們採用知名的 SRILM 工具 [61]，遞迴式類神經網路語言模型個人化採用 RNNLM 工具 [71]，平滑化則主要採用強化聶氏 (Modified Kneser-Ney Smoothing) 平滑法 [26]。在遞迴式類神經網路語言模型的部分，我們額外加入了許多輔助的特徵值幫助增進模型的預測能力，加入的特徵值包括由中研院斷詞系統及普林斯頓詞網 (Princeton Word Net) [72] 獲得詞性標記 (POS)、從臉書上撈得的每個使用者的人口統計資訊 (DE) 包含男女性及語言設定、100 個維度的潛在使用者群體 (LUG) 特徵。在後面的實驗中我們測試使用單一特徵及組合特徵，測試不同的特徵組合對混淆度與語音辨識率的影響。

4.4.2 混淆度實驗結果與結果分析

表 4.2列出了我們對遞迴式類神經網路語言模型，在不同模型調適階段、不同特徵值組合、不同潛藏層數目的混淆度實驗結果。五個主要的大列分別主要為：(A) 背景語料訓練的語言模型；(B) 只用使用者個人語料做模型調適；(C) 經 (B) 調適後再用使用者的朋友語料調適；(D) 經 (C) 調適後，在測試階段同時根據測試語料做動態訓練 (dynamic training)，學習率設為 0.05；(E) 經 (C) 調適後與 N 連文法語言模型結果做線性內差，內插權重設為 0.5。其中相對應的個人化 N 連文法語言模型表現以標記 KN3 表示 (KN3：Kneser-Ney 3-gram)，標示在每個大列主旨的下方。不同潛藏層數目的實驗結果同時以 H 參數標示於各小列，不同的特徵值組合則標示於各行中。首先我們可以發現的是，我們提出來的三步驟調適機制在本混淆度實驗裡相當成功， 混淆度都有顯著的下降 (二步調試後 < 一步調試後 < 背景模型)。主要的進步還是在於利用此調適的方法，我們成功的減少了背景模型與個人用語的不匹配程度，因此增進模型的預測能力。考量到觀察過的測試語料應該也對於未來的測試語料預測有所幫助，因此 (D) 欄便列出我們用學習率 = 0.05 (learning rate) 在測試時同時做動態調適的結果，我們也可以觀察到顯著的進步 (二步調試後 + 動態調適 < 二步調試後)。第五大列則是二步調適後的模型再與相對應的 N 連文法語言模型進一步線性內差的結果，跟過去的許多文獻顯示的結果相似，結合之後的預測能力會大於兩者單獨預測 (二步調試後 + KN3< 二步調試後)，代表兩種模型中存在著互補性。不同的潛藏層數目對於遞迴式類神經網路語

Table 4.2: 遞迴式類神經網路根據不同模型調適階段、不同特徵值組合、不同潛藏層數目的混淆度表。

混淆度 (PPL)		特徵值						
		無	詞性	人口組成	潛在群體	詞性＋人口	詞性＋潛在	全
(A) 背景模型 (KN3=343.57)	H=50	341.90	331.91	335.33	319.23	332.25	315.72	317.84
	H=100	309.50	305.77	308.88	295.45	302.83	292.71	292.29
	H=200	289.14	284.82	282.14	271.48	279.97	270.85	271.15
	H=500	267.38	262.08	261.51	258.60	261.19	255.61	253.92
(B) 一步調適後 (KN3=299.32)	H=50	296.55	286.25	287.83	280.74	286.39	274.63	273.97
	H=100	280.74	265.26	262.04	252.19	259.01	252.72	251.93
	H=200	253.97	243.24	243.11	239.77	242.51	235.88	234.05
	H=500	234.55	224.76	225.20	224.21	225.90	217.85	220.53
(C) 二步調適後 (KN3=233.20)	H=50	270.88	263.77	266.09	251.64	265.05	249.57	243.30
	H=100	250.95	237.52	238.27	234.59	237.60	228.26	226.23
	H=200	225.12	214.87	216.85	210.29	216.21	208.64	207.37
	H=500	209.56	196.50	199.23	195.59	195.76	192.85	192.83
(D) 二步調試後＋動態調適學習率 γ=0.05	H=50	240.18	244.40	236.33	233.87	235.00	233.15	224.55
	H=100	226.35	217.06	212.08	210.26	212.00	212.64	206.38
	H=200	202.28	203.52	194.02	193.50	193.27	194.85	188.61
	H=500	187.66	179.69	179.34	182.86	177.53	178.87	175.82
(E) 二步調試後+KN3 權重 $\lambda = 0.5$	H=50	182.24	182.20	179.45	175.86	178.62	175.07	173.38
	H=100	174.61	172.94	171.72	168.03	171.77	168.80	167.74
	H=200	168.98	165.21	164.86	163.14	165.16	162.83	161.33
	H=500	161.88	158.45	159.19	157.23	156.96	156.27	155.40

言模型的表現也有著決定性的影響。從表 4.2中看出，一般而言，越高的潛藏層數目俱有越高的預測能力，也就是擁有越低的測試混淆度 (H=500 < H=200 < H=100 < H=50)，但隨著層數漸增，混淆度降低的幅度也漸漸呈現飽和現象。因此我們大概可以整理出一個結論：在我們的實驗中最佳潛藏層數目應該高於 500，小於 500 的值都還不足以模擬臉書語料中上下文的特性。最後，不同的輔助特徵也可以提供模型不同的預測能力。一般來說，有加輔助詞特徵都會略優於沒有加詞特徵 (其他行 < 無)，主要原因在於輔助的詞特徵提供了當上下文特性不明朗化時，模型有更廣泛的特徵可以參考。雖說混合不同特徵共同使用在潛藏層數目少的時候還可以獲得相當的進步量，但隨著潛藏層數目漸漸增加，其混合的效果就漸漸消失了。主要可能是因為越大的潛藏層數目便可以模擬更複雜的上下文關係，因

此也就比較不需要額外特徵的輔助了。最後，我們可以發現我們最好的模型 ((E) 欄，H=500，混合全部特徵) 可以獲得最低的混淆度表現，降低了背景 N 連文法語言模型混淆度約 57.4 個百分點、基本型遞迴式類神經網路語言模型 (背景模型，H=500，無特徵) 混淆度約 41.8 個百分點。

4.4.3 前 N 最佳結果重評分實驗結果

由於目前尚未有任何一個解碼器 (decoder) 支援類神經網路語言模型直接做語音辨識，因此本實驗設定為利用遞迴式類神經網路語言模型，對 N 連文法語言模型產生的前 N 最佳結果重評分 (N-best rescoring)。我們的辨識系統同樣以 HTK 工具 [62] 為基礎，產生前 1000 最佳辨識清單 (1000-best list)。用來產生前 1000 最佳結果的語言模型為個人化過的、強化聶氏平滑法過的三連文法語言模型 (Personalized KN3)。辨識的聲學模型基本上也與章節 4.2.1的設定相同。非監督式的最大相似度線性迴歸法 (MLLR) 同樣也被拿來作為語者調適的方法。為了比較對於不同語者狀況及錄音情況對辨識結果的不同，我們測試在兩組不同的錄音語料上，語料 (I) 與語料 (II)。基本上語料 (I) 與我們在前面章節 4.2.3 對個人化 N 連文法語言模型時的測試語料是同一組，由於是乾淨背景加上只有兩個錄音者，故是一個較簡單的辨識環境。語料 (II) 是我們在第二次搜集資料以後重新錄製的語料，在多種不同的背景環境之下，包含路邊、實驗室、臥室、客廳等等不同的地點由不同的語者錄製，是較困難的辨識環境。此兩組資料特性整理如表 4.3。

Table 4.3: 前 N 最佳結果重評分實驗所用的兩個錄音語料統計資料

	目標使用者數目	錄音語者數目	總共測試句子數	錄音環境
語料 (I)	21	兩位	840 句	乾淨背景環境
語料 (II)	42	多數	948 句	多種不同環境

由表中可以看出，語料 (I) 由於初始辨識率已經較高，因此重評分後的進步量較少，在潛藏層數目太少的狀況下 (H=50 或 100) 甚至很容易退步。而語料 (II) 則因初始辨識率本來就較低，因此給予重評分方法比較多的進步空間，因此基本上重評分過後都是有進步的。再者，表中也顯示出，聲學模型調適雖然可以增進初始的辨識率，但是同樣的就會壓縮到重評分可以再進步的空間。不同的潛藏層數目基本上影響很大，大部份的數據都顯示出潛藏層大小在低於兩百的時候，在大

部份狀況下基本上是不夠用的。因此我們知道，當潛藏層數目夠大的時候，個人化遞迴式類神經網路語言模型廣泛上是比個人化 N 連文法語言模型還要好的。我們比較單獨使用遞迴式類神經網路語言模型重評分，及遞迴式類神經網路語言模型線性內差個人化 N 連文法語言模型的結果重評分後的結果。雖然在語料 (I) 上我們發現，結合兩個模型的結果會相較單純使用一個模型還要好，與過去其他文獻的結論相同。但在語料 (II) 上結果卻是結合後的結果反而較單純使用遞迴式類神經網路語言模型還要差。這可能是因為語料 (II) 的辨識環境較為嚴峻，許多辨識錯誤造成了上下文資訊出現了許多雜訊，而這些雜訊造成了較短的上下文資訊十分不可靠，必須觀察較長的上下文關係來彌補。而模擬較長的上下文關係正是遞迴式類神經網路語言模型優於 N 連文法語言模型的主要點，因此出現了結合 N 連文法語言模型反而更差的實驗結果。最後，結合不同的詞輔助特徵在本實驗裡未必較好，反而是單一使用某種特徵通常獲得最好的結果。這極有可能是因為過大的特徵向量會造成資訊過飽和，使得不同資訊之間互相干擾，不止沒有發揮一加一大於二的功效，反而使得結果變差。

表 4.4 便是我們前 1000 最佳結果重評分的實驗結果，表中列出兩組不同的錄音語料、聲學模型有無調適，這四種不同排列組合的背景條件下測試的結果。由於語料 (II) 是較困難的辨識語料，因此基本上語料 (I) 的初始辨識結果高於語料 (II) (66.05>53.01%，69.87%>58.05%)；在同樣的語料條件下，有做聲學模型調適的結果又優於沒做聲學模型調適的結果 (69.87>66.05%，58.05%>53.01%)。每個背景條件我們都測試了，遞迴式類神經網路語言模型單獨 (RNNLM)、及遞迴式類神經網路語言模型線性內差 N 連文法語言模型 (RNNLM+KN3) 的結果，兩者之間的權重設定為 $\lambda = 0.75$。如同表 4.2，我們同樣測試了多種不同潛藏層數目 (H=50、H=100、H=200、及 H=500) 與不同輔助特徵的排列組合對結果的影響。對應的 N 連文法語言模型辨識的最佳結果 (first-pass result) 則顯示於左側邊欄，而表中的數據都是相比該側邊欄辨識結果的絕對進步率，也就是經過該特定設定下的遞迴式類神經網路語言模型重新對前 1000 結果評分之後的結果與原始數據的差。為了方便瀏覽，差值在一定值以上的結果以綠色標定 (進步較顯著)、差值為負以紅色標定 (退步)，其餘以黑色表示。其中兩組錄音語料由於進步比例不一，語料 (I) 進步 0.5 百分點即以綠色標定，語料 (II) 則必須進步超過 1.0 個百分點才以綠色標定。

Table 4.4: 遞迴式類神經網路在兩個不同錄音語料、不同特徵值組合、不同潛藏層數目的前 N 最佳結果重評分實驗結果。

正確率 (Accuracy) 絕對進步率 (%)			無	單一特徵值			兩組特徵值		全
				詞性	人口組成	潛在群體	詞性＋人口	詞性＋潛在	
語料 (I)，AM，66.55%	RNN	H=50	-0.45	-0.67	-0.80	-0.53	-0.79	-0.66	-0.44
		H=100	0.08	-0.53	0.24	-0.46	0.01	-0.03	0.15
		H=200	0.27	0.41	0.57	0.49	0.44	0.34	0.56
		H=500	0.86	0.94	0.80	0.50	0.95	0.99	0.73
	RNN +KN3 $\lambda = 0.75$	H=50	0.38	-0.06	0.11	0.06	-0.06	-0.10	0.04
		H=100	0.77	0.37	0.46	0.08	0.30	0.46	0.13
		H=200	0.71	0.84	0.86	0.60	0.81	0.91	0.45
		H=500	0.96	0.92	1.08	0.94	0.97	0.70	0.91
語料 (I)，MLLR，69.87%	RNN	H=50	-0.95	-0.68	-1.17	-0.34	-0.81	-0.67	-0.81
		H=100	0.13	-0.82	-0.51	-1.06	-0.41	-0.75	-0.54
		H=200	-0.19	0.02	-0.16	-0.36	0.05	-0.28	-0.33
		H=500	0.33	0.63	0.18	0.17	0.32	0.55	0.44
	RNN +KN3 $\lambda = 0.75$	H=50	-0.19	-0.29	-0.39	-1.20	-0.21	-0.43	-0.07
		H=100	-0.02	-0.18	-0.01	-0.17	0.03	0.32	-0.13
		H=200	0.08	0.32	0.27	0.05	0.18	0.24	0.21
		H=500	0.76	0.87	0.22	0.87	0.44	0.38	0.32
語料 (II)，AM，53.19%	RNN	H=50	1.42	1.36	1.33	1.44	1.30	1.56	1.59
		H=100	1.46	1.66	-0.67	1.80	1.54	1.75	1.51
		H=200	1.69	1.87	1.70	1.90	1.83	1.72	1.86
		H=500	1.94	1.97	1.87	2.03	1.82	1.95	1.95
	RNN +KN3 $\lambda = 0.75$	H=50	0.86	0.84	0.89	0.95	0.93	0.94	0.99
		H=100	0.96	1.32	1.09	1.28	1.15	1.22	1.17
		H=200	1.20	1.20	1.27	1.37	1.38	1.16	1.32
		H=500	1.50	1.58	1.34	1.46	1.37	1.49	1.42
語料 (II)，MLLR，58.05%	RNN	H=50	1.57	1.51	1.57	1.59	1.55	1.59	1.59
		H=100	1.69	1.69	1.66	1.75	1.60	1.71	1.74
		H=200	1.75	1.60	1.82	1.87	1.70	1.70	1.89
		H=500	1.85	1.89	1.77	1.78	1.82	1.78	1.88
	RNN +KN3 $\lambda = 0.75$	H=50	1.00	1.08	1.05	1.10	1.03	1.05	1.10
		H=100	1.27	1.26	1.21	1.22	1.15	1.20	1.27
		H=200	1.38	1.30	1.37	1.36	1.47	1.23	1.41
		H=500	1.38	1.57	1.55	1.42	1.32	1.43	1.43

4.5 個人化語言模型結論

上一章提出一個自給自足、具有語音介面的雲端系統生態系，訓練的語料取之於使用者，訓練完的模型則用來回饋使用者增進辨識率。本章就整個系統過去最少人研究、也是整個系統成功關鍵的環節：語言模型，做一系列其個人化能力的研究，包含了傳統的 N 連文法語言模型及遞迴式類神經網路語言模型。從本章實驗的內容中，我們可以大致提煉出幾個重點：

- 利用系統綁定社群網路帳號的方式，提供了我們立即的個人化語料庫來源，讓我們可以馬上初始一個個人化語言模型。
- 朋友的語料庫也是個很有幫助的資料來源，藉由這些語料庫的幫助，可以補齊稀疏的個人語料庫欠缺的許多上下文特性。
- 除了個人語料之外，社群網路上的許多額外資訊，不管對於 N 連文法語言模型或者遞迴式類神經網路語言模型的個人化，都有相當程度上的幫助。
- N 連文法語言模型的調適，由於本身對於資料採不連續的表達形式 (discrete representation) 的問題，自由度相當大，但資料卻很少，直接套用機器學習最佳化之很容易造成過貼合問題，故必須以許多先驗知識 (prior knowledge) 整合多組語料降低學習自由度作為前處理，才能套用最佳化方法。
- 遞迴式類神經網路語言模型，藉由將不連續的字詞投影到連續空間 (continuous space)，對每個字詞俱有更簡練的表述能力；更甚者，其遞迴式架構理論上可以模擬任意長度的上下文關係，更容易捕捉到長距離的文本相依關係。因為這樣的特性，遞迴式類神經網路語言模型做個人化是比 N 連文法語言模型更加適合的，因為藉由以上的兩個優點，它可以藉由投射到連續空間擴大少量的個人化語料訓練的效果，亦可以藉由模擬長距離的文本關係，達到更強的個人化能力。
- 最後，這兩種模型理論上都可以動態地根據新近來的使用者語料，做線上的更新 (online update)。藉由這整塊配套措施的結合，我們才能真正創建一個不斷根據該使用者語言使用習慣自我調適的個人化語言模型。

Chapter 5

個人化語言模型與個人化聲學模型整合

5.1 個人化聲學模型

在介紹了許多語言模型個人化的方法之後，我們在本章將重新審視前人 [73] 提出的聲學模型調適機制，並藉由將兩者整合得到一個個人化的辨識系統。圖 5.1為前人提出的串接式聲學模型調適架構，輸入端為語者獨立模型 (SI model：Speaker independent model)，輸出端為語者調適模型 (SA model：Speaker dependent model)，中間經過了三階段的調適步驟：

1. 全域性最大相似度線性回歸法 (GMLLR：Global Maximum Likelihood Linear Regression)：如同章節 2.1.1 所敘述相同，最大相似度線性回歸法藉由在調適語料上用最大相似度準則估計一套語者仿射轉換，而達到調適模型參數的目的。若模型中所有聲學單位 (如高斯混合) 共享同一個線性轉換，則稱為全域性最大相似度線性迴歸（global MLLR），可大致調整模型方向。此方法固

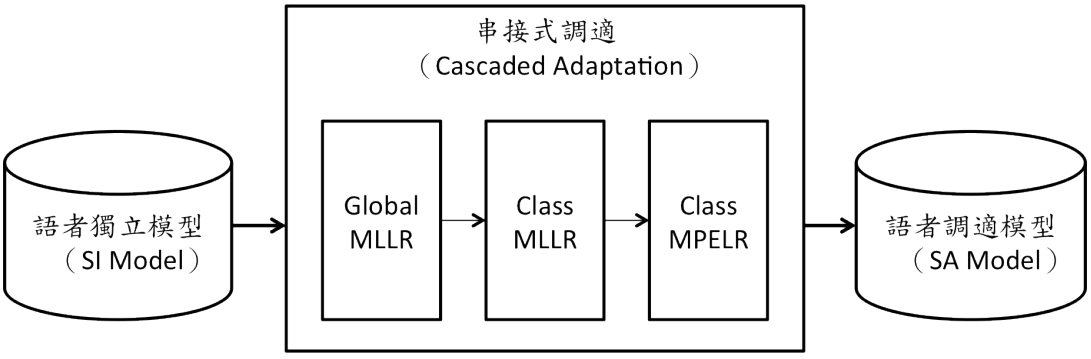


Figure 5.1: 串接式聲學模型調適架構

然較為粗糙，但在語料稀少的情況下可避免資料稀疏的問題。

2. 多群式最大相似度線性回歸法 (CMLLR：Class-based Maximum Likelihood Linear Regression)：此方法基本上與一般最大相似度線性回歸法無異，差別在於它是根據聲學單位的特性分成多個迴歸群 (regression class)，而每個群裡面的每個聲音單元共享同一個仿射轉換，是一種較為細緻的做法。

3. 最小音素錯誤線性迴歸 (MPELR：Minimum Phone Error Linear Regression)：此方法語最大相似度的準則不同，而是屬於鑑別式訓練 (Discriminative Training) 方法的一種。其做法為先對訓練語料進行初步辨識產生多組結果 (如前 N 最佳)，透過調整參數盡量拉開正確轉譯與每筆辨識錯誤的結果，而提升模型對於易混淆音素的鑑別能力。一般使用最大交互資訊 (MMI：Maximum Mutual Information) [74]、最小音素錯誤 (MPE：Minimum Phone Error) [75] 等不同鑑別準則，均表現相當成功。

其主要的概念是，先經由全域性最大相似度線性回歸法調整大致模型走向後，再利用類別性最大相似度線性回歸法，與最小音素錯誤線性迴歸作更細緻的調適。除此之外，在前人論文 [73] 中也設計了一個漸進式的調適機制，如圖 5.2。該系統隨著使用者使用次數越來越多，搜集越來越多的語料對模型逐步做個人化調適。

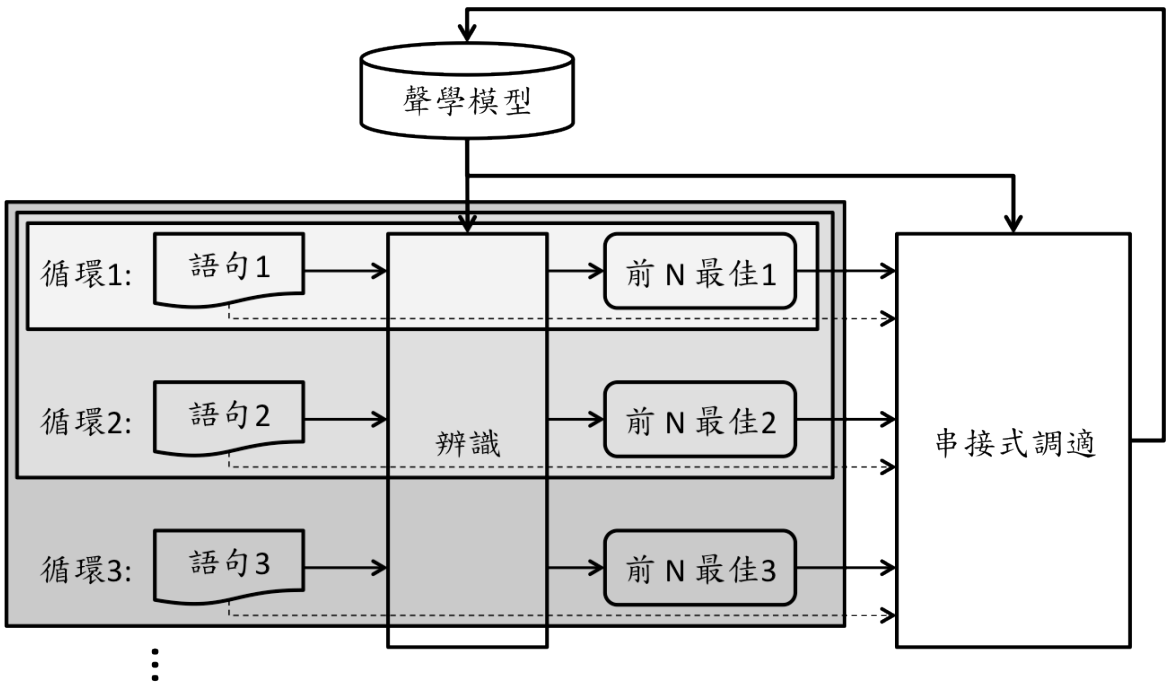


Figure 5.2: 循環漸進式聲學模型調適機制

5.2 整合系統架構

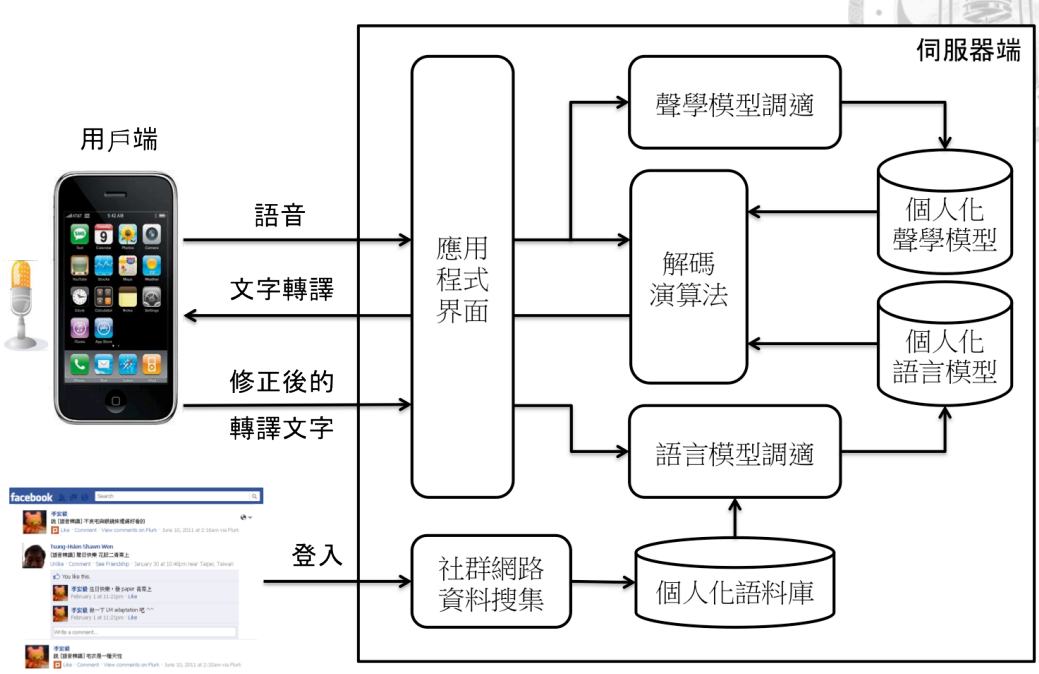


Figure 5.3: 個人化辨識系統整合系統

最後，我們預期中的個人化辨識系統將被整合如同圖 5.3，包含了個人化語言模型與個人化聲學模型的調適。圖中最下面的一部份，是使用者一開始用系統前，系統藉由使用者登入後提供的權限抓取社群網路語料，並離線 (offline) 建立個人化的語言模型，希望藉此快速提高語言模型對每個個體語言的辨識能力。而上半部分主要著重於線上 (online) 不斷調適與更新。藉由使用者輸入的語音訊號，系統的工作主要是回傳辨識後的結果回用戶端並顯示給使用者看。但此同時，這些使用者的聲音資料也會被記錄在伺服器端，並採用上述章節 5.1中的方法，逐步不斷地調適聲學模型。語言模型的線上更新，則仰賴於使用者覺得辨識不滿意後，利用手機界面修改後的文字結果。這段修改完的文字，除了讓使用者作後續的用途外，也會被回傳到伺服器端記錄下來，並同時採用語言模型調適再次調整語言模型。最後，這樣子的系統的好處在於，在調適語料少的時候可以藉由社群網路的資料快速個人化；而在使用者越用越頻繁後，聲學模型與語言模型也同時慢慢被導向更適合該使用者的方向，且當中的調適資料盡可能地自給自足。我們相信像這樣子的系統未來將會越來越多，使得真正的智慧型學習系統能夠漸漸真正地落實於我們生活中。

Part II

互動式搜尋系統

Chapter 6

互動式語音搜尋系統

6.1 簡介

第一部分我們介紹了如何將每個使用者的辨識系統個人化，以達到更精準的辨識正確率。因為本論文提出的這個雲端系統依靠語音為主要介面，因此我們可視為前一部份為系統的基礎建設。本論文的第二部分，我們利用一個例子來介紹如何在這樣的系統環境之下，架設一個真正用戶端的應用程式。本章我們以互動式語音搜尋系統為一個例子，探討互動性在純語音操作環境下的重要性。

6.1.1 研究動機

互動式搜尋系統 (IIR：Interactive Information Retrieval) [76, 77]，主要利用一個互動式的使用者介面來跟使用者作互動，以期能夠驅使使用者提供更多有助於搜尋其心中所要內容的資訊。藉由這樣子的互動，系統能夠漸漸地釐清使用者真正的需求，因此能提供品質更好的搜尋結果給使用者。在過去的研究裡，" 京都語音對話導覽系統"(Dialogue navigator for Kyoto city) [10, 78]，可以視為在這個領域裡，相當完整的一套互動式對話系統。該系統採用基於貝式風險的對話管理員 (Bayes risk-based dialogue manager)，根據現階段觀察到的使用者反應，挑選統計上風險最低的動作作為系統回應。利用這個方式，該系統被實作在京都市的導覽系統，效果相當不錯。"MIT 電影瀏覽器"(MIT MovieBrowser) [46, 47] 則是最近的一套互動式搜尋系統。該系統利用條件隨機場域 (CRF：Conditional Random Field) 對使

用者輸入語句進行語意標記 (Semantic tagging)，對使用者語意進行了解，並將了解之後的資訊作為檢索資料庫的查詢詞，進行電影資訊的高精度的檢索。該系統的整個搜尋流程相當順暢且精確度相當高。這樣子的互動式系統，如同上述的兩個例子，背後都依賴一個結構化的資料庫 (structured database) 作為檢索的目標。但如今由於網際網路的發達，許多非結構化的知識 (unstructured knowledge) 大量產生，我們同樣需要一套有效率的互動式系統幫助我們找到這些非結構化資訊背後我們要的內容，但現階段這樣子的系統是十分欠缺的。

互動式搜尋系統對於語音內容，如上課錄音、電視新聞、或者演講內容等等尤其重要，其主要原因在於：(1) 語音內容由於是訊號內容，它很難顯示在螢幕上給使用者看。儘管列出了排序的清單 (ranking list)，使用者仍然必須一筆一筆點進去聽才有辦法判斷這筆內容究竟是否是他想要的。因此它很難瀏覽也很難選擇。(2) 在數位內容產生地如此快速的大環境之下，已經不再可能用人工轉寫所有內容，而必須轉而依靠電腦自動轉寫。但電腦自動轉寫包含許多不可避免的錯誤，這樣子的錯誤結果將對搜尋技術造成另一層挑戰，使其更難做到精確的字詞比對。因為這兩個原因的影響，導致系統回傳給使用者的排序清單效果較差，加上難以瀏覽，使用者就必須花很多時間去聽更多的內容才能找到他所想要的內容。因此，一個互動式的系統假若可以根據當下的狀態做出判斷，例如：倘若判斷第一遍搜尋結果 (first-pass) 太差或者查詢指令 (query) 太過模糊，系統會自動跟使用者要求更多的資訊；反之，倘若系統判斷結果已經夠好了則會將結果回傳給使用者。藉由這樣子的互動，可以給使用者較好的點選清單，讓使用者避免必須瀏覽大量聲音才能找到想要內容的麻煩，將對很多現存語音內容檢索的質量有所改善。

6.1.2 系統組成

本論文提出的互動式語音搜尋系統架構圖如圖 6.1所示，主要包含兩個主要的模組，分別為對話管理員 (DM：Dialogue manager) 及搜尋模組 (Retrieval module)。最左邊當使用者輸入查詢指令時 (Query)，搜尋模組裡的搜尋引擎 (Search engine) 會先根據該查詢指令，在後端已建好索引的口述語料庫 (spoken achive) 中搜尋比對相關的文章內容，對之做排序，便得到一組檢索排序清單。在傳統搜尋系統裡，該清單會被直接回覆給使用者；而在我們這個互動式搜尋系統中，系統並不

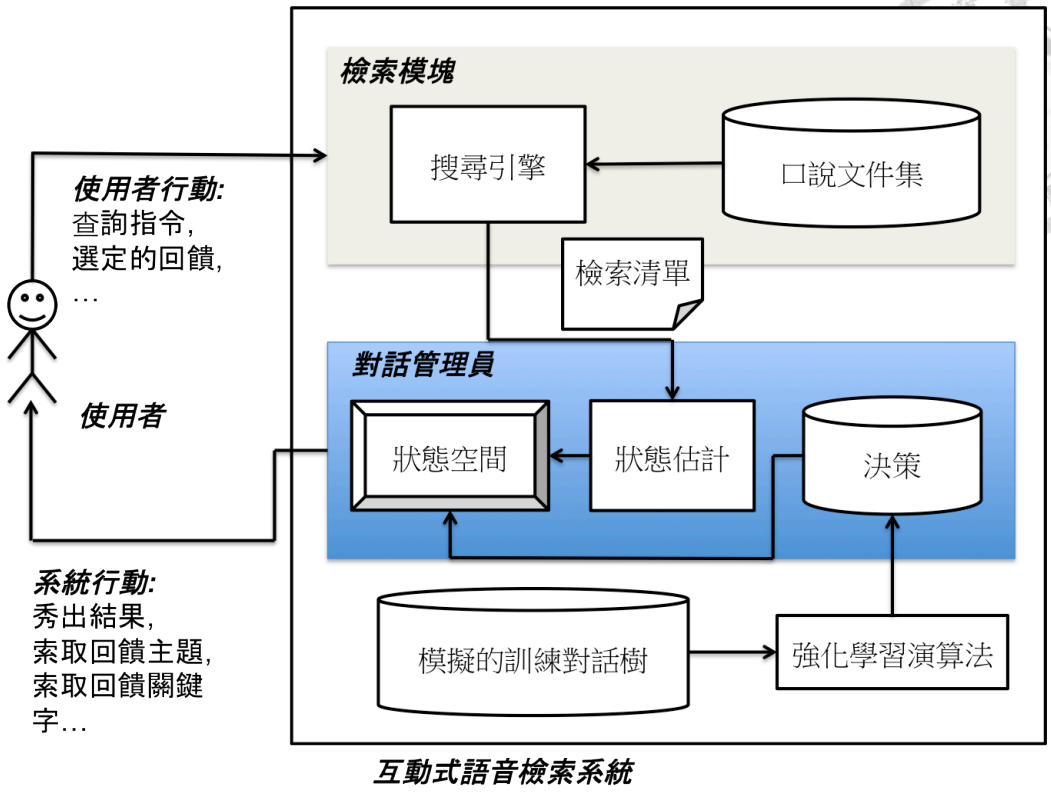


Figure 6.1: 本論文提出的互動式語音搜尋系統架構圖

直接回覆該清單，而是先對該清單抽取一些特徵值，並由這些特徵值估測該清單的品質，而此估計出來的品質則被視為該系統目前所在的狀態 (State)。另一方面，系統的決策 (Policy)，也就是在什麼狀態該採取怎麼樣的行動 (Action)，可以根據一組訓練的對話語料，藉由強化學習 (Reinforcement Learning，RL) 而得。給定了系統決策及現階段的系統狀態，系統便知道如何採取行動，來應對使用者的行為。

6.2 語音檢索的相關回饋

圖 6.1的搜尋引擎其整體架構主要承接章節 2.2及圖 2.5的概念，而延續之更進一步探討在向量空間模型 (Vector Space Model, VSM) 檢索與語言模型 (Language Model, LM) 檢索兩種不同的比對演算法架構下，不同的相關回饋機制 (Relevance Feedback, RF)。

6.2.1 向量空間模型相關回饋

在向量空間模型檢索中 (章節 2.2.3)，初始的查詢指令與每篇文章都被表示成在一高維空間中的向量，$\overrightarrow{Q_0}$ 與 $\overrightarrow{D_j}$，其中該空間為 $|V|$ 維度，即為詞典的維度，故每一維度均代表一個詞。當使用者對系統進行相關回饋之後，回饋的資訊可能是一個詞、新的查詢指令、亦或是某一篇文章相關與否，我們也都可以將這些資訊表達成向量的形式，只是會有正相關 (positive) 與負相關 (negative) 的差別。給定了正相關與負相關的資訊後，我們採用羅氏回饋演算法 (Rocchio formula)，將更新過後的查詢指令向量寫為：

$$\overrightarrow{Q}_{i+1} = a \cdot \overrightarrow{Q_i} + \frac{b}{|R|} \cdot \sum_{\overrightarrow{v} \in R} \overrightarrow{v} - \frac{c}{|I|} \cdot \sum_{\overrightarrow{v} \in I} \overrightarrow{v} \tag{6.1}$$

i 代表互動回合，R、I 分別為所有正相關資訊向量與負相關資訊向量的集合，a、b、c 為權重參數。雖然一般我們稱計算向量空間中的兩個向量的相似度是用餘弦相似度 (cosine similarity)，但實際上由於查詢指令的平滑項 $||\overrightarrow{Q}||$ 並不影響排序結果，因此我們可以忽略之：

$$R(\overrightarrow{Q}, \overrightarrow{D_j}) = \cos(\overrightarrow{Q}, \overrightarrow{D_j}) = \frac{\overrightarrow{Q} \cdot \overrightarrow{D_j}}{||\overrightarrow{Q}|| \cdot ||\overrightarrow{D_j}||} \approx \frac{\overrightarrow{Q} \cdot \overrightarrow{D_j}}{||\overrightarrow{D_j}||} \tag{6.2}$$

結合式 6.1 與 6.2，再加上在我們的互動系統裡，每回合只會有一組新的回饋資訊進來，因此藉由忽略式 6.1裡的加項 (summation term)、並為了簡單起見先忽略負回饋項只考慮正回饋項的效果、同時亦將回饋的資訊向量 $\overrightarrow{v}$ 對包含的詞數目 $dim(\overrightarrow{v})$ 進行平滑化以利選取權重，我們可以推得文件向量 $\overrightarrow{D_j}$ 與查詢指令的相關性分數隨著互動回合更新的式子：

$$\begin{aligned} R(\overrightarrow{Q}_{i+1}, \overrightarrow{D_j}) &= \frac{\overrightarrow{Q}_{i+1} \cdot \overrightarrow{D_j}}{||\overrightarrow{D_j}||} \\ &\approx a\overrightarrow{Q_i} \cdot \frac{\overrightarrow{D_j}}{||\overrightarrow{D_j}||} + d\frac{\overrightarrow{v}}{dim(\overrightarrow{v})} \cdot \frac{\overrightarrow{D_j}}{||\overrightarrow{D_j}||} \\ &= a \cdot R(\overrightarrow{Q}_i, \overrightarrow{D_j}) + \frac{d}{dim(\overrightarrow{v})} \sum_{m \in |V|} \overrightarrow{v_m} \cdot \frac{\overrightarrow{D_j}}{||\overrightarrow{D_j}||} \\ &= a \cdot R(\overrightarrow{Q}_i, \overrightarrow{D_j}) + d \cdot \frac{1}{dim(\overrightarrow{v})} \sum_{m \in |V|} R(\overrightarrow{v_m}, \overrightarrow{D_j}) \end{aligned} \tag{6.3}$$

其中式 6.3裡的第一項為上一個回合得到的相關性分數，而第二項則為回饋的資訊向量 $\overrightarrow{v}$ 的相關性分數，a、d 分別代表兩項間的權重。我們將回饋的資訊向量 $\overrightarrow{v}$ 拆成其各維度的線性組合 $\overrightarrow{v} = \sum_{m \in |V|} \overrightarrow{v_m}$，故每個 $\overrightarrow{v_m}$ 都代表了一個詞典裡的詞。藉由這樣的逼近與分解，我們可以在離線 (offline) 時預先計算每個 $R(\overrightarrow{v_m}, \overrightarrow{D_j})$，而在線上 (online) 時只需要將使用者回饋回來的資訊按照這些分數組合起來即可，有利於整個系統的加速。最後真正在系統實用時，倘若回饋回來的是一篇文章，我們只採用它所包含的最重要的 M 個詞，其集合設為 Γ，式子寫為：

$$R(\overrightarrow{Q}_{i+1}, \overrightarrow{D_j}) = \alpha \cdot R(\overrightarrow{Q}_i, \overrightarrow{D_j}) + \frac{\beta}{M} \sum_{\overrightarrow{t} \in \Gamma} R(\overrightarrow{t}, \overrightarrow{D_j}) \tag{6.4}$$

倘若回饋回來的是一個詞 t，則更新分數的式子寫為：

$$R(\overrightarrow{Q}_{i+1}, \overrightarrow{D_j}) = \alpha \cdot R(\overrightarrow{Q}_i, \overrightarrow{D_j}) + \gamma \cdot R(\overrightarrow{t}, \overrightarrow{D_j}) \tag{6.5}$$

以上的推導雖只根據正回饋項作推導，但負回饋項的推導做法均同。

6.2.2 語言模型相關性回饋

語言模型檢索 (章節 2.2.4) 將查詢指令與文章均表示成一個語言模型 (或詞類分佈)，每個維度一樣都代表一個詞。在跟使用者互動的過程中，系統漸漸累積了越來越多的資訊，包含正相關或負相關的。為了模擬與查詢指令負相關的資訊，除了 θ_Q 外，我們額外模擬了一個負向查詢指令語言模型 θ_N [79]。因此，這些回饋的資訊在語言檢索模型的架構當中，便被分別拿來重新估計一個較好的查詢指令語言模型 θ'_Q 及負向查詢指令模型 θ'_N。我們採用了過去廣泛應用在虛擬相關性回饋 (Pseudo relevance feedback, PRF) 的方法而將它應用在我們互動式的回饋上：基於查詢詞限制條件的混合模型 (Query-regularized mixture model) [16, 80]。以下以正回饋為推導例子，但負回饋都可以依此類推。該模型假設了一篇被使用者指定為相關集合 R 的文章裡的每個詞，是由一個與查詢指令相關的語言模型 θ'_Q 及背景語言模型 θ_B，以一個文件特定 (document-dependant) 的比例 γ_D 所組合的混合模型，所產生的，如圖 6.2所示。該文件相關集合 R 除了可以是文件的詞分佈外 (document distribution over words)，亦可以是主題的詞分佈 (topic distribution

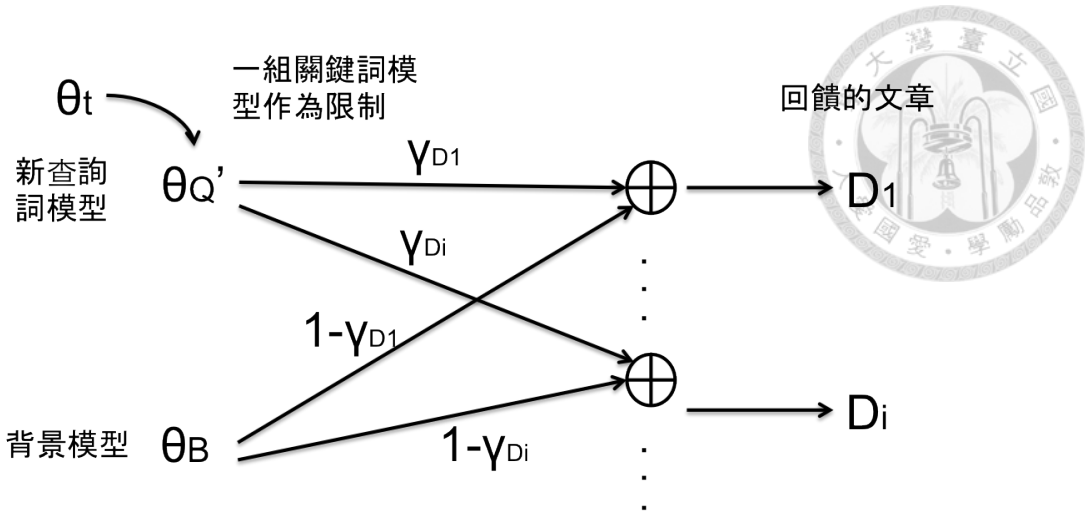


Figure 6.2: 基於查詢詞限制條件的混合模型假設

over words)，是系統藉由於使用者的互動中，慢慢收集而擴增而來。每個詞究竟是從 θ'_Q 抑或從 θ_B 來的，及每篇文件的 γ_D 均是未知，都是我們要求的模型參數，可以從使用者回饋回來的正相關文件集合 R 裡估計。給定一篇文件 d 及其根據式 2.9計算而得的語言模型 θ_d，我們希望可以找到一個如同上述的混合模型 θ'_d：

$$P(w|\theta'_d) = \gamma_d P(w|\theta'_Q) + (1-\gamma_d)P(w|\theta_B) \tag{6.6}$$

使得 θ_d 與 θ'_d 兩者之間的 KL 散度最接近。但於此同時，我們又希望新估計的查詢指令語言模型 θ'_Q 不要偏離我們希望的限制 θ_t 太多。θ_t 主要由原始查詢指令及使用者回饋的單詞資訊而得，可寫為：

$$P(w|\theta_t) = \frac{N(w,R')}{||R'||} \tag{6.7}$$

R' 是查詢指令與使用者回饋的詞的集合，而初始時應該只包含所有查詢指令裡的詞，隨著互動發生會漸漸擴增。$N(w,R')$ 代表詞 w 在 R' 裡出現的次數。根據上述的兩個條件，我們可以寫出我們想要最佳化的目標函數 (objective function) 為：

$$F(\theta'_Q, \{\gamma_d\}_{d\in R}) = \sum_{d\in R} KL(\theta_d|\theta'_d) + \mu KL(\theta_t|\theta'_Q) \tag{6.8}$$

式 6.8中的第一項，即是這個生成模型 (generative model) 的似然函數 (likelihood function)，即要最佳化的目標本體；而第二項，即是此模型的先驗限制項 (prior

constraint)，用先驗知識 (prior knowledge) 來限制模型避免過貼合於資料。μ 則是兩項之間的權衡權重。求取的參數為新的查詢指令語言模型 θ'_Q，及每篇文章的混合權重 $\{\gamma_d\}_{d\in R}$。我們可以視此問題為最大後驗概率的問題 (MAP：Maximum a posteriori)，可用最大期望演算法 (EM) 解之。

最後，整個系統的運作為：當使用者正向回饋了一個詞分佈 (點擊文件或點擊主題)，這個文件或主題的詞分佈將會被加到 R；倘若使用者正向回覆了一個詞，則此詞將會被加入到 R'。如此一來，當搜尋系統要重新排序結果時，便會先利用這些收集到的資料，根據上述提出的最大後驗概率方法估計一組現階段最適合的查詢指令語言模型，再用此模型根據式 2.10對文件重新排序，整體重估流程如圖 6.3所示。

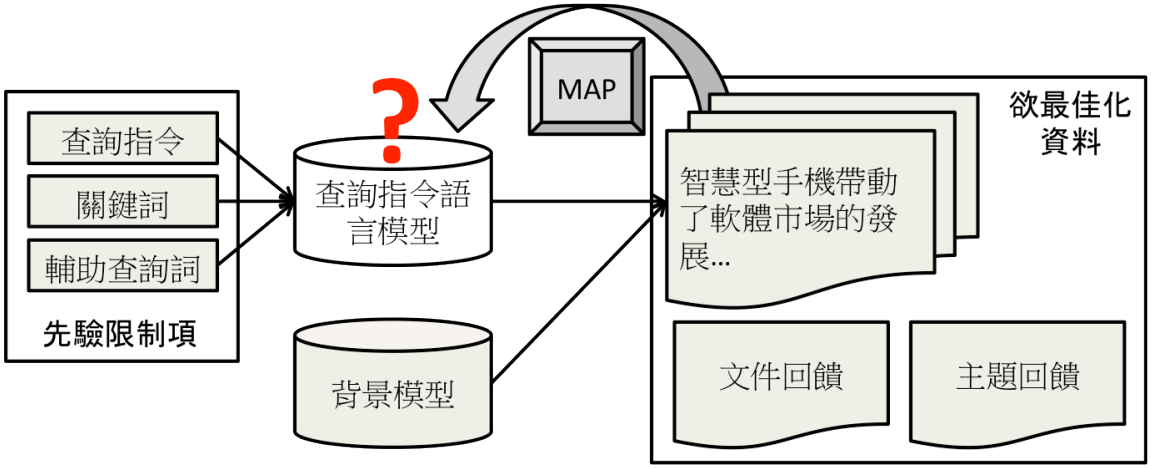


Figure 6.3: 基於最大後驗概率法的查詢指令語言模型重估

Chapter 7

馬可夫決策模型作為對話管理員

7.1 馬可夫決策模型模擬互動式搜尋系統

承接章節 2.2.5，馬可夫決策模型 (MDP：Markov Decision Process)，可由五個參數表示之：$\{S, A, T, R, \gamma\}$，分別代表狀態、行動、轉移機率、獎勵、以及折扣參數。為了用馬可夫決策模型模擬互動式搜尋系統，我們必須將它的五項參數一一設定。其中，在互動式搜尋這樣的問題下，轉移機率 T 屬於外部環境動態參數 (Environment Dynamics)，我們難以模擬。因此，在轉移機率的部分，我們並不直接模擬，而改用取樣 (sampling) 的辦法逼近環境的特性。而折扣參數部分我們設為 1.0，原因在於我們的使用者並不在乎哪一回合增加了系統效能，反正只要在最後秀給使用者看的時候，他有觀測到該進步即可。另外三項參數，則在下面幾個小節詳述。

7.1.1 狀態 (State)

狀態 (State) 的挑選是馬可夫決策模型的一個很重要的課題。馬可夫決策模型最後的目標，是要學習出一個系統決策 (Policy) π，使得給一個狀態，系統可以採取最佳的行動，去最佳化價值函數 Q，如同式 2.13所示。由於狀態是函數 π 的輸入，因此倘若挑選的狀態不夠俱有鑒別度，系統便很難學出每個狀態採取不同行動的差別；又倘若系統的狀態被挑選的太過細緻，那需要訓練的狀態數目則又太多，訓練起來十分困難且很沒有效率。因此，挑選好的狀態，可以達到事半功倍

的效果。在我們設計的系統裡，由於狀態參數必須反映系統對於當前排序清單的信心指數，一個最直接的方法，便是設其中一種資訊檢索的衡量指標 (evaluation metric) 作為狀態參數之一，例如：準確度 (precision)、召回率 (recall)、F– 度量 (F-measure)、或平均準確度 (MAP：Mean Average Precision)。在我們的系統設計裡，考量到平均準確度 (MAP) 考慮了排序清單整體的表現，是一個較全面的指標，因此我們採用它作為系統的狀態參數之一。而後來我們比較了

- 將此狀態參數量化 (quantization) 成離散空間參數，或
- 保留其連續狀態參數的特性兩種方法，

來比較系統表現能力的差異。另一個參數則設為系統互動的回合數 (dialogue turn)。一個很容易想像的原因是，系統越互動到越後面的回合，由於前幾輪的互動已經在某些程度上彌補了查詢指令資訊的不匹配，因此能夠再造成的進步量可能已經很有限了。倘若經過好幾輪表現仍然不見起色，那可能代表這個查詢指令對於系統而言太過困難或甚至系統根本沒有使用者想要的資訊，因此持續互動下去極有可能也是徒勞。但光是平均準確度這個參數並無法告訴我們這件事情，因此我們多加了一個互動回合數作為狀態參數。

給定一有 N 筆結果的排序清單，考慮單一查詢指令平均準確率 (AP：Average Precision) 的計算：

$$AP = \frac{\sum_{n=1}^{N} Prec(n) \cdot rel(n)}{||R||} \tag{7.1}$$

$Prec(n)$ 代表該排序清單前 n 個排序結果的準確率； $rel(n)$ 是一個示性函數 (indicator function)，倘若第 n 比結果相關則輸出 1，否則輸出 0。 R 是所有相關的文件集合。由式 7.1中可知，平均準確率的估計需要知道使用者心中那組相關的文件集合，也就是正確答案 (ground truth)。但事實上，這個資訊對系統而言是未知的，因此在過去此指標都只能拿來最後作為衡量檢索系統效能用。在章節 7.3，我們提出一些方法來估計現在排序清單的平均準確率，也就是系統狀態。

7.1.2 行動 (Action)

在本系統中，我們對系統總共定義了五個不同的行動 (action)。在每個互動的時間點 i，系統處於會根據定義的相關性分數計算方式 (式 2.7或式 2.10)，計算每

篇文章對當前查詢指令的相關度 $R(Q_i, D_j)$ 並依此排序。根據這個排序清單，系統可以採取的行動如下：

1. 列出結果 (*Show List*)：當系統採取這個行動時，當按照 $R(Q_i, D_j)$ 排序的清單將會回傳給使用者作為最後結果，並結束本次搜尋。如果系統不是選擇這個行動的話，系統便會採取下列四種回饋動作的其中一種對使用者做回饋，並根據回饋的資訊按照新的相關度分數 $R(Q_{i+1}, D_j)$ 更新排序清單。

2. 要求挑選文件 (*Return Document*)：系統回傳了當前按照 $R(Q_i, D_j)$ 排序的清單，並要求使用者由上往下看，挑選第一篇相關的文章當做回饋。

3. 要求回覆某一關鍵詞是否相關 (*Return Key-term*)：系統從現階段的排序清單中挑選一個關鍵詞 t^*，要求使用者回覆某一關鍵詞是否相關。在向量檢索模型中：

$$t^* = \arg\max_t \sum_{d \in \mathcal{D}} f(d,t) ln(1 + idf(t)), \tag{7.2}$$

$f(d,t)$ 為詞 t 在文章 d 裡的詞頻 (TF：term frequency)，$idf(t)$ 為詞 t 的倒文件頻 (IDF：inverse document frequency)。而在語言模型檢索模型中：

$$t^* = \arg\max_t \sum_{q \in \theta_Q'^{(i)}} P(q|\theta_Q'^{(i)}) \times Jaccard(q,t) \tag{7.3}$$

$\theta_Q'^{(i)}$ 是第 i 個時間點的查詢指令語言模型，可由式 6.8解得，而 q 我們實際上只取了前 M 個最重要的詞。傑氏係數 (Jaccard coefficient) $Jaccard(q,t) = \|D_q \cap D_t\| / \|D_q \cup D_t\|$，為兩個詞 q 與 t 之間共同出現在文章中次數的一個指標，越高代表共同出現次數越多。

4. 要求新查詢詞 (*Return a Query*)：系統要求使用者回饋另一個查詢指令。

5. 要求回饋主題 (*Return Topic*)：系統會回覆給使用者一組由潛藏主題模型 [54, 81] 推論出來的主題清單，每一個主題是由該主題內出現機率最高的前 k 個詞來代表，並要求使用者挑選一個跟他想要的內容最相關的一個主題。被挑選到的主題中詞的多項分佈則會被當作一篇文章回饋到系統之中，該文件長度則被設為該文件集平均的文章長度。

7.1.3 獎勵 (Reward) 與回報 (Return)

給定一個系統決策 π，一個檢索對話過程 (retrieval session) 開始於狀態 s_0，並根據該決策 π 採取行動，產生一連串的對話過程 $\Gamma_\pi(s_0) = \{s_0, s_1, ...s_K\}$，而 s_K 為最終狀態 (final state)。對於每一個在狀態 s_k 採取的行動 $a_k = \pi(s_k)$ 而言，系統會獲得一個獎勵函數 $r_k = C(a_k)$，該函數定義於所有系統可以採取的行動之上。在本論文中提出，由於所有的回饋行動 (feedback actions) 都對使用者帶來更多的額外工作負擔，因此都被設定為獲得負的獎勵 (negative reward)，或代價 (cost)。但在每個會話的最後一個回合採取的行動，也就是" 列出結果" 這項行動，則會獲得一個跟平均準確率 (MAP) 有關的獎勵，定義為

$$r_K = \tau[E(s_K) - E(s_0)] \tag{7.4}$$

$E(s)$ 為在狀態 s 時我們定義的檢索評量指標，也就是平均準確率 (MAP)。而 τ 則是一個權衡參數，取決於我們系統較著重的是檢索精確度或者是盡量減少使用者的工作負擔。最後，系統的總回報 (Return) 可以寫為 $G = \sum_{k=0}^{K} r_k$。

7.2 強化學習演算法

倘若觀察人類甚或動物的學習過程，我們其實不難發現，這些學習行為的基礎來自於與環境的互動。就像一個新生兒，對周遭環境並不熟悉，也沒有一個指導者 (supervisor) 在旁指導他應該怎麼做，怎樣做才是正確的。而是在他每次動作探索之後，環境藉由某種形式給之反饋。藉由這些回饋，他漸漸知道每個動作所可能造成的結果，也知道該怎麼做才能達到他的目的。這樣子的學習，我們稱作強化學習 (RL：Reinforcement Learning)，也是在日常生活中，人們學著如何跟環境互動的主要方式。

在機器學習的領域裡面模擬強化學習的主要目的，便是讓電腦自己去學習一個應對外在環境的策略 (Policy)。不同於監督式學習 (supervised learning)，學習代理人 (agent) 必須去嘗試執行各種不同的動作，接著觀察環境回饋回來的訊號。假若學習代理人採取了一個好的動作，則環境會藉由此訊號來獎勵他；反之，則用此訊號來懲罰。這個演算法體現到馬可夫決策模型上，便是大家熟知的值迭

代 (value iteration) 學習法，如章節 7.2.1。為了解連續空間馬可夫決策模型的問題，值迭代法更進一步被改良成貼合值迭代法 (FVI：Fitted Value Iteration)，如章節 7.2.2

7.2.1 值迭代學習法

回憶我們曾在章節 2.2.5說過，馬可夫決策模型的主要目的在於學出一個系統決策 π，為一從狀態空間映射到動作空間的函數，$\pi : S \rightarrow A$。為了輔助策略的學習，我們也定義了 Q-函數 (Q-function) 為 $Q^{\pi} : S \times A \rightarrow \mathbb{R}$，為一個從狀態與動作的聯合空間映射到一個實數 (real number) 空間的函數。這個 Q-函數代表的意義，便是在某個狀態採取某個行動之後，整個對話流程中系統可以獲得的獎勵期望值總和，寫成如式 2.12。該式為一個迭代式的函數，我們可以更進一步將其期望值項展開：

$$Q^*(s,a) = \sum_{s'} P_a(s,s')[R_a(s,s') + \gamma \max_{b \square nA} Q^*(s',b)] \tag{7.5}$$

$P_a(s,s')$ 為在狀態 s 上採取行動 a 後跳到狀態 s' 的轉移機率，是由外在的環境決定。一般而言，在很多的應用情況下，如對話系統 (dialogue system)，外在環境造成的轉移機率系統是無法知道的，因此系統往往需要對外在環境建模型模擬這個轉移機率，我們稱之以模型為基礎的方法 (model-based method)。倘若我們純粹藉由從跟環境的互動中做取樣 (sampling) 而不建立模型模擬，我們則稱為無模型的方法 (model-free method)。考量到對話系統本身的特性，我們採用無模型的方法。

在無模型的方法裡，我們所需要的學習資料則必須藉由與真實使用者互動，或者藉由一個模擬的使用者 (simulated user) 來產生。而產生出來的這些互動資料，則利用基於動態規劃的概念 (DP：dynamic programming)，在每個迭代回合更新部分 Q-函數，漸漸讓 Q-函數收斂到穩定的值，我們稱為最佳 Q-函數 (optimal Q-function)。而這個最佳的 Q-函數，便可以用來萃取出我們所要的決策。

7.2.2 貼合值迭代學習法

值迭代法雖然可以讓我們求出最佳 Q-函數，但它假設狀態空間是離散的，因此這個對應關係是一個確切的表示形式 (exact representation)，當狀態空間是

連續的時候就難以表示。在這種時候，一個最簡便的方法便是將 Q-函數參數化 (parameterization)，也就是寫成一個由許多基底函數 (basis function) 組成的線性組合：

$$Q_{\underline{\rho}_i}(s,a) = \sum_m \rho_m \phi_m(s,a) = \underline{\rho}^T \underline{\phi}(s,a) \tag{7.6}$$

其中 $\{\phi_m(s,a)\}_{1\leq m\leq M}$ 是給定的基底函數，$\underline{\rho} \in \mathbb{R}^M$ 則是用來對基底函數做線性組合 (linear combination) 的參數。$\underline{\rho}$ 與 $\underline{\phi}(s,a)$ 則是他們兩者分別對應的向量形式 (vector form)。而 $Q_{\underline{\rho}_i}(s,a)$ 則是我們對 $Q_i^*(s,a)$ 的一個估計，而目標也就是要找出一組參數 $\underline{\rho}$，使得兩者越像越好。擬合值迭代演算法 (FVI：Fitted Value Iteration) [82--84] 提供了一套在連續空間中估計值迭代近似解的方法。擬合值迭代法大致分成以下兩個步驟，不斷迭代直至估計的 $Q_{\underline{\rho}_i}(s,a)$ 收斂或者滿足了一定的迭代數之後：

1. 值取樣 (Sampling)：此步驟基本延續傳統值迭代演算法。給定一筆訓練資料 (s_i, a_i, r_i, s_i')，分別代表當前狀態、採取的行動、獲得的獎勵、及下一個狀態，我們可以從當前的 Q-函數中取樣出新的函數值：

$$D(Q(s_i,a_i)) = r_i + \gamma \max_{a\in A} Q(s_i', a) \tag{7.7}$$

   $D(\cdot)$ 代表取樣的運算子 (operator)。式 7.7稱為取樣貝氏最佳方程式 (Sampled Bellman Optimal Eqaution)。由於此取樣函式並沒有包含環境的轉移機率項，因此它可被視為是一種無模型的做法。

2. 值貼合 (Fitting)：由於取樣未必會包含於我們預先設定的基底函數所展開的空間中，因此我們必須將取樣出來的值，想辦法貼合到這組基底函數可以表示的空間上。而解決這個問題的方法，便是給定現在的參數 $\underline{\rho}_{i-1}$、一組訓練語料 $\{(s_j, a_j, r_j, s_j')_{1\leq j\leq N}\}$、及取樣出來的點們 $D(Q_{\underline{\rho}_{i-1}}(s_j,a_j))$，藉由解正規化線性回歸 (regularized linear regression) 找到一組新的參數 $\underline{\rho}_i$ 能夠使資料均方誤差 (mean square error) 最小：

$$\underline{\rho}_i = \arg\min_{\underline{\rho}\in\mathbb{R}^K} \sum_{j=1}^{N} \left(D(Q_{\underline{\rho}_{i-1}}(s_j,a_j)) - Q_{\underline{\rho}}(s_j,a_j)\right)^2 + \frac{\eta}{2}\|\underline{\rho}\|^2 \tag{7.8}$$

式 7.8中的第二項是正規化項，主要目的在於避免學出來的參數過貼合於資料點，而 η 則是我們在兩項之間作權衡的參數。此式可以用矩陣成法得出一個關閉形式解 (closed form solution)。此演算法操作將從一組隨機初始化的參數 $\underline{\rho}_0$ 開始，迭代重複上述的兩個步驟直到收斂為止。

7.3 狀態估計

雖然說當系統在進行訓練的時候，是有該查詢指令對應到的正確文章們作為計算當前狀態的依據，但當跟真正使用者做線上互動的時候，系統其實是缺乏這方面資訊的。因此我們勢必不能仰賴真正的檢索指標，而必須用一套方法直接從檢索出來的清單與查詢指令的許多面向，直接做估計。

7.3.1 狀態特徵值

為了採用機器學習的方法直接從查詢指令及排序清單估計當前的清單好壞 (或者系統狀態)，我們必須首先抽取特徵值 (feature)。這些特徵值在許多文獻中主要分成兩種，一種主要是在產生排序結果前，只利用查詢指令與文件集的資訊抽取，這種我們稱作檢索前指標 (pre-retrieval indicator)。而倘若在產生排序清單以後才整合清單資訊及上述提到的幾種特性計算的指標，我們稱作檢索後指標 (post-retrieval indicator)。在我們後來的實驗中，採用的指標主要有以下幾種：

- 查詢指令長度：越長的查詢指令可能對資訊需求 (information need) 描述得越清楚，對於系統來說越容易找到相關的結果。
- 目前互動回合數 [19]：到了越後面的互動回合，使用者加入的資訊越多，所以理論上應該會有更好的結果
- 過去採取的行動 [20]：倘若過去系統曾經採取了比較積極的回饋動作，理論上也會獲得較好的結果。
- 清楚度分數 (clarity score) [85]：

$$Clarity = \sum_{w \in V} P(w|Q) \log_2 \frac{P(w|Q)}{P(w|C)} \tag{7.9}$$

$P(w|Q)$ 為查詢指令的詞類分佈，而 $P(w|C)$ 為文件集 C 的詞類分佈，故式 7.9可視為兩個機率分佈的交叉散度 (cross entropy)。當該式值越大，兩者越不像，代表該查詢指令有其特定主題，足以跟整個文件集分布作區分。

- 查詢指令涵蓋範圍 (query scope) [85]：

$$Scope = -\log(n_Q/N) \tag{7.10}$$

n_Q 是文件集中至少出現其中某個查詢指令詞的文件總數，N 是文件集的大小。按照字面上的意義來詮釋，意思便是查詢指令所涵蓋的文件集範圍，值越大，涵蓋的範圍越廣。

- 查詢指令與文件集相似度 (SCQ：Similarity between Query and Collection) [86]：

$$SCQ = \sum_{t \in Q}(1 + \ln(f_{c,t})) \cdot \ln(1 + \frac{N}{f_t}) \tag{7.11}$$

$f_{c,t}$ 為詞 t 出現在文件集 c 裡的次數，而 f_t 則是文件集中出現該詞 t 的文章數目。本權值衡量該查詢指令對於文件集的相似程度，倘若越相似該查詢指令越容易在文件集中檢索出好的結果。

- 加權資訊增益 (WIG：Weighted information gain) [87]：

$$WIG(Q, C, L) = \frac{1}{K} \sum_{D_j \in T_K(L)} \log \frac{P(Q, D_j)}{P(Q, C)} \tag{7.12}$$

Q 是查詢指令，C 是文件集，L 是檢索清單，K 是可以調整的排名參數，代表我們只要看前 K 個結果。$T_K(L)$ 是一個運算子，代表只保留前 K 筆排序結果而忽略這之後的結果。$P(Q, D_j)$ 是查詢指令與文章 D_j 共同出現的聯合機率 (joint probability)，而 $P(Q, C)$ 則為查詢指令與文件集的聯合機率。倘若此值越大，代表查詢指令出現在排序前 K 的文件中，相對於文件集整體而言比例越高，代表系統更有效率地將包含查詢指令的文件排在越前面。

- 查詢指令回饋法 (QF：Query feedback) [87]：此方法的概念在於，假若檢索出來的清單很能夠反映查詢指令所想代表的資訊內容，那代表我們倘若利用

一套方法，將此檢索清單壓縮回一個新的查詢指令，此新查詢指令應該要與原有的查詢指令十分相似才是。其實作方法主要如下

1. 依據排序清單估計一個排序清單語言模型 (ranked list language model)：

$$P(w|L) = \sum_{D \in L} P(w|D)P(D|L) \tag{7.13}$$

$P(D|L)$ 一般設為一個隨著文章排名線性遞減的函數。

2. 我們將 $P(w|L)$ 裡的所有詞按照以下準則排序：

$$P(w|L) \log \frac{P(w|L)}{P(w|C)} \tag{7.14}$$

$P(w|C)$ 是詞 w 在文件集中的機率。

3. 式 7.14中的前 N 個詞則被選出來形成一個新查詢指令 $Q' = \{(w_i, t_i)\}_{i=1,...N}$。$w_i$ 是第 i 個詞而 t_i 則是該詞的權重，是該詞對式 7.14貢獻的比例。

- 排序清單的前 N 相似度分數、平均值、與標準差 [19, 20]：排序清單的前幾名相似度分數越高，其實相當程度上反映了該查詢指令越容易匹配到文件集中的文件。為了考慮不同排名的影響，我們考慮了多組前 N 個的組合。除了分數之外，我們也考慮了其巨觀統計數據，例如平均值與標準差等等。

7.3.2 離散狀態估計 - 多類別支持向量機

對於離散狀態分類演算法的部分，我們採用支持向量機 (SVM：Support Vector Machine) 來幫助我們分類系統狀態。支持向量機與許多傳統的分類演算法不同，主要的基礎奠定於最小化結構風險 (SRM：Structural Risk Minimization) 定理而來。利用這個方法，支持向量機試圖改善尚未觀察到的資料被分錯的機率，以達到不錯的一般化效果 (generalization)。給定一組資料即每組資料對應的類別 (正或負，二元分類)，支持向量機首先將資料點投影到一組高維空間，並嘗試在這個高維空間中，尋找一個超平面 (hyperplane)，來幫助系統做決策，此決策函數可以表示為：

$$f(\mathbf{x}) = sigmoid(\mathbf{w} \cdot \mathbf{x} + b) \tag{7.15}$$

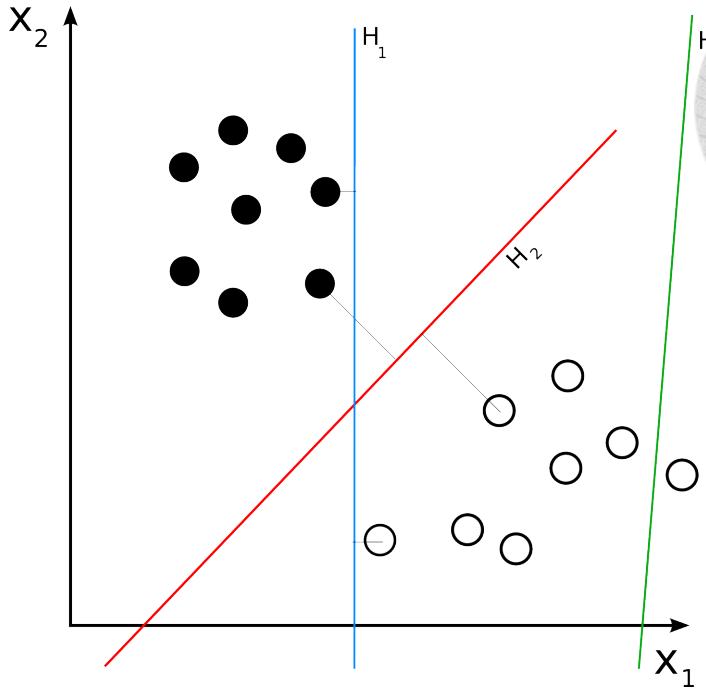


Figure 7.1: 支持向量機學出來的超平面示意圖

其中 **x** 為資料點，**w** 為超平面參數，b 為一常數。系統的決策則根據式 7.15的輸出做決策，可能是正或負。超平面是由參數決定，需要學習的參數共有 **w** 及 b。但支持向量機學出來的平面，除了希望能把資料點盡量分開之外，還希望能夠最大化正與負資料點之間的間距 (margin)。圖 7.1很能夠表示這樣的概念。三個平面中 $H1$、$H2$ 雖然都可以成功把兩組俱有不同特性的資料點分開，但我們會說 $H2$ 這個平面對於這組資料點而言優於 $H1$，原因乃是它最大化了兩者之間的間距，而這也正是支持向量機希望找出來的平面。

在實作的部分，由於我們的問題是一個多類別分類 (multi-class classification) 的問題，目的是將抽出來的特徵向量映射到多個類別中的某一個。我們採用支持向量機的多類別版本，對每個類別，並把其他的類別都當成同一類訓練多個一對多 (1-vs-N) 的分類器，並由多個分類器投票整合最後結果。再者，由於此問題對於預測真實的平均準確率還是有相當大的差距，我們首先根據第一筆結果的分數做量化 (quantization) 及互動回合，這兩個做資訊分組。這個分組機制很像圖 7.2中的樹狀架構，而對某一個樹狀架構的葉端節點 (leaf node)，都訓練一個多類別支持向量機做更進一步的分類。

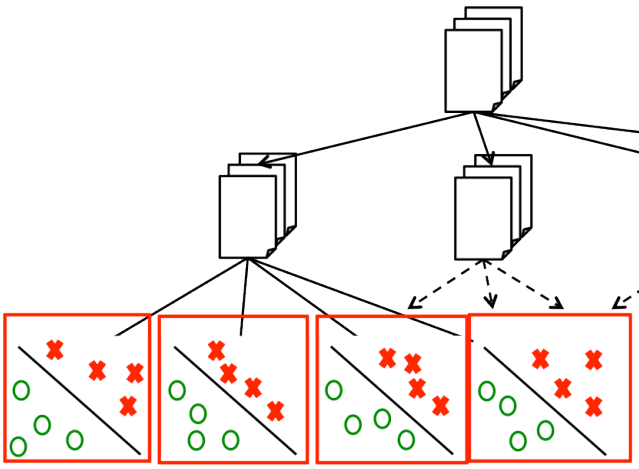


Figure 7.2: 結合樹狀結構與多類別支持向量機的狀態分類架構

7.3.3 連續狀態估計 - 正規化線性回歸法

對於連續狀態的估計則較為單純，因為我們要做的決定是柔性決策 (soft decision)，而不是硬性決策 (hard decision)，系統比較有辦法可以抵抗因為雜訊造成的狀態估計錯誤，系統較為強健。連續狀態估計的問題，不同於離散狀態估計是一個分類問題，而是一個回歸問題 (regression)。給定多維的特徵向量，我們的目的就是學出一組參數 σ 能夠最貼近我們訓練資料點所標定的平均準確率值：

$$\sigma^* = \arg\min_{\sigma \in \mathbb{R}^q} \sum_{j=1}^{N} \left(E_j - \sigma^T f(L)\right)^2 + \frac{\lambda}{2}\|\sigma\|^2 \tag{7.16}$$

$f(L)$ 便是如章節 7.3.1裡定義的，對檢索清單 L 運算的特徵函數， E_j 代表第 j 比資料點對應到的目標平均準確率， q 為參數維度也是特徵維度。式 7.16同樣是正規化線性回歸，可以解出一個封閉解。

7.4 系統評估

為了評估我們提出的互動式搜尋系統表現，我們在新聞語料庫上測試初始的實驗結果。

7.4.1 實驗設定

我們採用的實驗語料是台灣公視在 2001 到 2003 年間，從台北的電視新聞或廣播電台錄下的語料庫。該語料庫總共有 5047 篇語音文件，總長度為 198 個小時，這些語音文件便是我們系統需要檢索的目標。辨識系統方面，60 萬個最常出現的字詞被選出來構成字典，語言模型方面則是用三連文法語言模型 (trigram LM) 訓練在 39M 的奇摩新聞上，而聲學模型則調整不同方法以利評估系統在不同辨識環境下的檢索效能，敘述於後。在檢索方面，我們採用平均準確率來估計檢索效能。為了避免系統過度互動而無法停止，我們限制系統的互動回合最多只能到五次。倘若互動數目超過五次，則系統會立刻回傳結果給使用者並強制結束對話內容。決策的訓練都是從完全未知開始學起。每個回饋行動所對應到的代價，考慮不同行動帶給使用者不同程度的額外工作，我們按照經驗法則設定，大致上的代價大小關係是：要求新查詢詞 > 要求回饋主題 > 要求挑選文件 > 要求回覆某一關鍵詞是否相關。檢索的文件由 22 個研究生提供總共 163 個文字形態的查詢指令，並標定了其對應到的多筆相關文件。相關文件的數目從 1 筆到 50 筆不等，平均比數則是 19.5，每個查詢指令的長度從 1 到 4 個中文詞不等。我們的實驗結果主要由模擬使用者 (simulated user) 進行訓練與測試，實驗中回報了不同參數、不同辨識率、連續空間與離散空間等各種不同條件下的系統表現，指標包含平均準確率 (MAP) 與回報 (Return)。

7.4.2 模擬使用者

在本實驗中我們利用模擬使用者進行訓練與測試，作為快速驗證想法的一個初步實驗，未來將可以做到實際使用者上評估系統效能。每個模擬使用者會根據 163 組查詢指令與對應的相關文件，作為其取樣的參考。根據系統採取的不同動作，我們的模擬使用者會有以下回覆：

- 當系統採取" 要求挑選文件"：模擬使用者按照系統的指示，將回傳的清單由上而下檢視，並根據所知的相關文件資訊，回饋第一篇他看到的相關文件。
- 當系統採取" 要求回覆某一關鍵詞是否相關"：模擬使用者會回答" 是的"，如果那個回傳的關鍵字出現在相關文件集一半以上的文件裡，反之，則回答"

不是"。

- 當系統採取" 要求新查詢詞"：則系統會根據以下準則挑選回饋的新查詢詞：

$$t^* = \arg\max_t \sum_{d \in \mathcal{R}} f(d,t) ln(1 + idf(t)) \tag{7.17}$$

$f(d,t)$ 與 $idf(t)$ 分別為詞 t 的文件頻與倒文件頻， R 是相關的文件集。

- 當系統採取" 要求回饋主題"：模擬使用者則隨機挑選一則相關的主題作為回應，相關的主題也是由研究生手動標定的。

7.4.3 離散空間向量模型初始實驗

本實驗主要測試離散空間、向量檢索模型做在兩種不同辨識度的語料庫下的實驗結果。我們採用兩種不同聲學模型辨識出不同的結果，並以此來測試在不同辨識環境下對檢索準確率的影響：

- (a) 聲學特徵採用傳統的梅爾倒頻譜係數 (MFCC：Mel Frequency cepstral coefficient) 實作，模型則用高斯混合模型 (GMM：Gaussian Mixture Model) 訓練在 24.5 小時的新聞語料上，每個隱藏馬可夫模型 (HMM：Hidden Markov Model) 上的狀態有 8 個高斯混合，辨識正確率為 45.64%，稱為文件集 (I)。

- (b) 聲學特徵同樣採用梅爾倒頻譜係數，模型同樣用高斯混合模型訓練在 24.5 小時的新聞語料上，每個隱藏馬可夫模型上的狀態有 64 個高斯混合，辨識正確率為 52.15%，稱為文件集 (II)。

在本實驗中，所有結果都是藉由 4 套交叉驗證 (4-fold cross validation) 的結果。

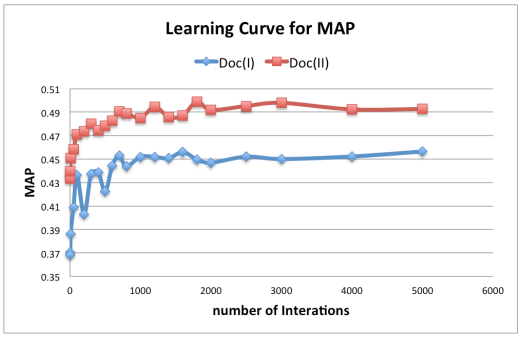
表 7.11 便是此實驗的實驗結果。縱軸方向的兩個主要欄位分別為上述提到的，利用不同的聲學模型跑出來的不同辨識度的文件集，(I) 與 (II)。同時，我們亦分別測試了兩種權衡參數 τ，其影響如章節 7.1.3所示。其中我們四個比較的基礎實驗 (baseline) 分別為：(1) 無回饋，系統只負責檢索，沒有提供互動的效能；(2)(3)(4) 只能採取該特定回饋動作，採取 N 次並選擇回報最好的一次結果。選擇最好的原因乃在於我們系統最主要的衡量指標是回報。而 (5)(6) 則是我們提出的利用馬

Table 7.1: 平均準確率與回報在向量檢索模型、離散空間馬可夫決策的實驗結果

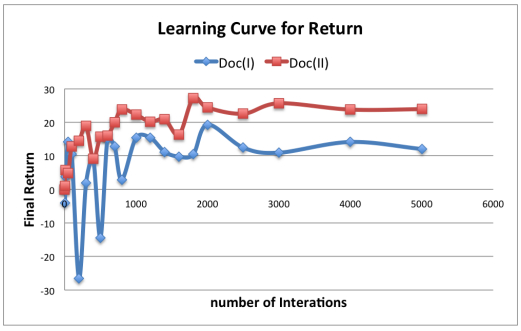
		文件集 *(I)*				文件集 *(II)*			
		τ=1000		τ=2000		τ=1000		τ=2000	
決策		MAP	回報	MAP	回報	MAP	回報	MAP	回報
基礎實驗	(1) 無回饋	0.3703	--	0.3703	--	0.4335	--	0.4335	--
	(2) 要求挑選文件	0.3341	-56.31	0.3341	-92.61	0.3807	-72.87	0.3807	-125.7
	(3) 要求新查詢詞	0.3693	-11.23	0.3693	-12.28	0.4283	-15.49	0.4283	-20.79
	(4) 詢問關鍵詞相關	0.4241	4.18	0.4241	57.99	0.4890	5.72	0.4890	61.08
MDP	**(5)** 全知狀態	**0.4607**	**46.75**	**0.4735**	**139.73**	**0.5249**	**40.63**	**0.5267**	**124.71**
	(6) 估計狀態	**0.4335**	**19.92**	**0.4558**	**91.25**	**0.4920**	**23.30**	**0.4975**	**84.05**

可夫決策模型與使用者進行互動的結果，其中 (5) 全知狀態，乃是假設倘若系統狀態是全然已知的狀態沒有估計錯誤的問題，所得到的結果，故可視為本系統的表現上限 (upper bound)。在所有的基準裡面，我們可以發現只有" 要求新查詢詞" 這項動作，才能同時可以在平均準確率與回報兩項指標中都獲得進步，而且單獨採取" 要求挑選文件" 或" 詢問關鍵詞相關" 亦都無法提供任何進步。但是，我們卻可以很清楚發現我們提出的互動模型利用" 全知狀態" 的資訊下，卻可以在平均準確率與回報上都雙雙獲得相當大的突破。而如果我們考慮了真實狀態下狀態其實是未知，並採用" 估計狀態" 這項方法之後，不可避免的估計錯誤影響了表現，故使得其表現低於" 全知狀態"((6) < (5))。但儘管如此，" 估計狀態" 的方法仍然好過於任何" 基礎實驗" 方法 ((6)>(1)(2)(3)(4))。並且，不管在哪一個辨識狀態或者參數 τ 的設定下，這樣的趨勢都是相同的。

圖 7.3(a)和圖 7.3(b)分別顯示了平均準確率與回報的學習曲線，採用估計狀態方法，在不同的訓練迭代數下對應到的值所畫出來的結果。圖中可以發現，兩個指標都隨著學習過程中漸增。在訓練的早期，由於觀測到的互動資料還不是太多



(a) 平均準確率學習曲線



(b) 回報學習曲線

Figure 7.3: 向量模型離散空間互動系統學習曲線

而有一些波動，但系統大概也在近 2000 個迭代數之後趨近飽和。

7.4.4 離散、連續空間語言模型實驗與結果比較

本實驗比較了離散與連續空間，語言模型檢索模型做在兩種不同辨識度的語料庫下的實驗結果。我們同樣採用兩種不同聲學模型辨識出不同的結果，並以此來測試在不同辨識環境下對檢索準確率的影響。另外，值得一提的是，對於不同的辨識結果，我們同時測試了最佳序列 (1-best) 與詞圖 (lattice) 對實驗的影響：

- (a) 聲學特徵採用梅爾倒頻譜係數疊上倒頻譜正規化法 (CMVN：Cepstral Mean and Variance Normalization)，模型則用高斯混合模型 (GMM：Gaussian Mixture Model) 訓練在 24.5 小時的新聞語料上，每個隱藏馬可夫模型上的狀態有 64 個高斯混合，辨識正確率為 54.43%，為文件集 (III)。

- (b) 聲學特徵串接感知線性預測 (PLP：Perceptual Linear Predictive) 特徵以及由多層感知器 (MLP：Multilayer Perceptron) 產生的音素事後機率，都是訓練在 10 個小時的新聞語料上，每個隱藏馬可夫模型上的狀態有 48 個高斯混合，辨識正確率為 62.13%，為文件集 (IV)。

在本實驗中，所有結果都是藉由 10 套交叉驗證 (10-fold cross validation) 的結果，每一套中 8 份查詢指令與對應的相關文件集作為訓練語料、1 份作為調參數用的驗證語料、最後 1 份則是測試語料。

Table 7.2: 平均準確率與回報在語言模型檢索模型、離散或連續空間 MDP 的結果

決策		文件集 (III)：最佳序列		文件集 (IV)：最佳序列		文件集 (III)：詞圖		文件集 (IV)：詞圖	
		MAP	回報	MAP	回報	MAP	回報	MAP	回報
基礎實驗	(1) 無回饋	0.4521	--	0.4950	--	0.4577	--	0.5044	--
	(2) 要求挑選文件	0.5205	38.32	0.5484	28.31	0.5343	46.59	0.5618	27.40
	(3) 詢問關鍵詞相關	0.4475	-19.02	0.4854	-19.66	0.4480	-19.75	0.4931	-21.33
	(4) 要求新查詢詞	0.4704	-31.78	0.4907	-54.32	0.4906	-17.16	0.4993	-55.11
	(5) 要求回饋主題	0.4766	4.44	0.5074	-7.66	0.4957	17.91	0.5437	19.24
全知狀態	(6) 離散空間	0.5796	93.81	0.6178	91.96	0.5926	110.04	0.6409	107.73
	(7) 連續空間	0.5839	99.29	0.6231	99.91	0.5921	102.67	0.6400	108.58
估計狀態	**(8)** 離散空間	**0.5354**	**61.63**	**0.5889**	**72.95**	**0.5491**	**68.12**	**0.6166**	**91.96**
	(9) 連續空間	**0.5398**	**67.07**	**0.5964**	**81.38**	**0.5626**	**84.54**	**0.6204**	**96.15**

表 7.2便是我們的實驗結果，我們列了平均準確率 (MAP) 與回報這兩項系統指標，在語言模型檢索模型、離散或連續空間馬可夫決策模型的表現。如上所述，實驗被驗證在不同的兩種辨識率、最佳序列或詞圖，排列組合後共四種文件集上。我們的比較基礎實驗 (baseline) 仍然同上述實驗一般，比較了 (1) 無回饋；與 (2)(3)(4)(5) 只能採取該特定的回饋動作 N 次並只取一筆回報指標最好的結果來比較。在本實驗中正式加入了第四項回饋動作：" 要求回饋主題"。我們藉由比較可以發現，採取特定的回饋行動，跟全然無互動的情況下並不保證得到進步量 ((2)(3)(4)(5) vs (1))。每個不同回饋行動的在語言檢索模型上的表現大致上是" 要求挑選文件" 優於" 要求回饋主題" 優於" 要求新查詢詞" 優於" 詢問關鍵詞相關"((2)>(5)>(4)>(3))。而 (6)(7)(8)(9) 則是我們提出來的，系統提供互動之後的結果。我們可以發現，提出的方法基本上都優於任一個基準結果 ((6)(7)(8)(9)>(1)(2)(3)(4)(5)) 同樣的，(6)(7) 全知狀態代表乃是假設系統狀態是全然已知的，狀態沒有估計錯誤的問題所得到的結果，故視為本系統的表現上限 (upper bound)。(8)(9) 考慮了真實狀態下狀態其實是未知, 並採用" 估計狀態" 這項方法之後, 不可避免的估計錯誤再度影響了表現, 故使得其表現低於" 全知狀態"((8)(9) < (6)(7))。但儘管如此," 估計狀態" 的方法仍然好過於任何" 基礎實驗" 方法 ((6)(7)>(1)(2)(3)(4)(5))。最後，我們比較連續狀態馬可夫決策與離散狀態馬可夫決策的表現，我們可以發現，不管在" 全知狀態" 或" 估計狀態" 兩種實驗底下，連續狀態馬可夫決策都優於離散狀態馬可夫決策 ((7) > (6), (9) > (8))，這主要原因出於連續空間的設定給予了訓練馬可夫決策的時候更大的狀態表示彈性，因此能夠更細微的模擬價值函數隨著狀態改變而改變的關係。上述的實驗結果在四組我們測試的不同文件集設定下，亦都顯示出了一樣的規律。

圖 7.4顯示了平均準確率與回報的學習曲線，採用估計狀態方法，在不同的訓練迭代數下對應到的值所畫出來的結果。圖中可以發現，兩個指標都隨著學習過程中漸增。在訓練的早期，由於觀測到的互動資料還不是太多而有一些波動，但系統大概也在近 2000 個迭代數之後趨近飽和。在圖中，連續狀態馬可夫決策以實線表示，離散狀態馬可夫模型則以虛線表示。我們可以發現，不管在哪一組測試文件集下，連續狀態馬可夫決策在訓練穩定之後，都優於離散狀態馬可夫決策。

7.5 互動式搜尋系統結論

本章討論的互動式搜尋系統，我們大致上可以列出以下重點：

- 互動性對於語音文件檢索來說特別重要，主要原因在於 (1) 語音文件很難被呈現且瀏覽耗時，且 (2) 過差的辨識率可能導致檢索結果更不如人意，因此回傳的排序清單必須有高準確度以確保使用者能夠快速找到他想要的結果。而與使用者互動來獲得更多資訊便是一個有效的做法。

- 不同的檢索模型有不同的預期效果，但一般當前學術界認知而言，語言模型檢索模型的好處在於數學推導嚴謹、需要調整的參數較少、可以套用完整的貝式機率理論，因此目前在學術界是比較得寵的檢索模型。

- 馬可夫決策模型模擬對話式的互動系統藉由權衡於檢索效能與使用者的額外工作之間可以達到相當程度的功效。不僅在平均準確率上有所上升，系統回報也有所進步。

- 連續狀態馬可夫決策相較於連續狀態馬可夫決策而言，擁有較好的狀態空間表達能力，因此對於系統學習而言，擁有較好的基礎，可以學習許多細緻的價值函數在些微狀態上的差異。

- 狀態估計是一個不容易但很重要的問題，做得越好越能夠提升整體系統的效能。最大的限制在於系統能知道的資訊太少，很多資訊可能也未必真正反映了真實狀態的趨勢。

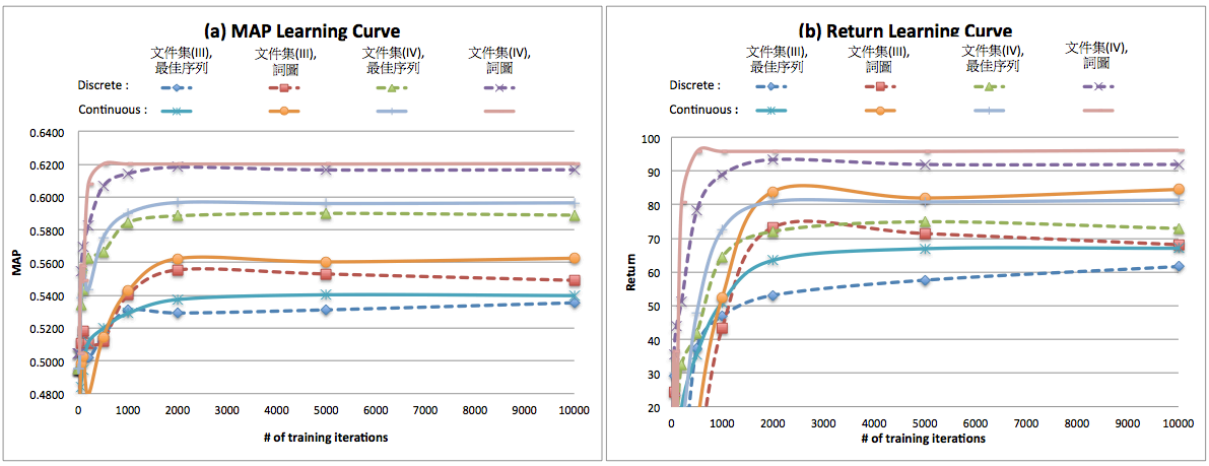


Figure 7.4:

Chapter 8

結論與未來展望

8.1 雲端化的應用程式平台

現在的資訊世界，大量的資料流竄在網路之上，應用程式與服務亦隨著網路的發達及大量流通的資料獲得了一大步的躍進。而人類行為的快速改變，讓許多這些應用程式與服務必須不斷地跟緊腳步，使自己能跟得上潮流 (up-to-date)。本論文所探討的，便是這一系列以語音為接口的雲端應用平台技術，該怎麼設計才能讓自己擁有源源不絕、由使用者端提供的資料，並藉由這些資料，讓自己的服務自動化地，跟上大多數使用者的使用情境與習慣。在這樣的大架構底下，我們以兩套系統作為例子：

1. 個人化的雲端辨識系統，及
2. 互動式語音文件搜尋系統。

希望對於如何達到這一步，提供一點簡單的想法與理念。這兩個系統，除了在本論文中提到的方法之外，仍然存有很多可以繼續改善與思考的空間，以下是我們認為的幾個可以再繼續改善的地方：

8.2 個人化語音辨識系統

對於個人化語音辨識系統，我們主要著重在語言模型的部分：

- 推論個人化的未知上下文結構：儘管我們能從社群網站上取得所需的個人化調適語料，為建立個人化的語言模型建立了基礎，但社群網路的語料仍然存在著極高的資料稀疏 (data sparseness) 的問題。怎樣利用許多存在的個人語料庫與社群網路上的關係，更精確的去推論個人化的未知上下文結構，是一個可以有效利用社群網路的力量去彌補資料不足問題的方法。這方面的研究，必須去假設、檢視、並實驗許多不同社群關係對語言使用的影響。倘若做得好的話，也許可以發現許多潛藏在人類語言機制之下的不同觀點。

- 遞迴式類神經網路語言模型的結構改良：現階段的遞迴式類神經網路雖然被證明了很適合拿來做語言模型，但倘若將此類模型放到社群網路的架構底下，我們便會發現他的不足之處。主要原因在於，它並沒有考慮進社群網路中，人與人之間利用不同的互動或關係互聯的關係。一個有趣的研究方向，將是將整個社群的結構也考慮進語言模型本身結構之中，藉由深度整合而獲得更好的個人化語言模型。

8.3 互動式搜尋系統

本文提出來的互動式搜尋系統，還處於初步利用實驗測試演算法的階段。對其未來而言，應該會有許多十分有趣且值得探討的研究方向：

- 建立及時線上系統與真實使用者互動學習：在當下，對於此系統來說，設計一個及時的線上系統，讓該學習機制可以被放到雲端跟真正的使用者互動並學習肯定是第一要務。但在這之前，仍有許多的準備工作，例如：雲端應用程式界面 (API：Application Programming Interface) 的建立、準備線上學習演算法 (online learning)、例外處理 (Exception Handling) 等等。

- 自然語言使用者互動界面：一個自然語言的使用者介面，肯定會使得使用者跟系統的互動變得流暢許多，也能大大提升使用者經驗。但想要訓練一個自然語言理解的單元，需要許多標定的使用者做監督式學習，因此初始的系統利用 Amazon Mechanical Turk 搜集資料訓練一個初始的系統，而後再把它放到雲端去更新與學習是比較實際的做法。

- 互動式問答 (Q & A) 系統：更終極的目標，是能夠對非結構性的資料 (unstructural data) 建立一套互動式的問答系統。對於結構性資料來說，這件事情相對容易，但是非結構性的資料卻是十分困難的。我們堅信，要做非結構性資料問答系統的第一步，便是將非結構性資料轉為結構性資料的過程。

Bibliography

[1] D. Hakkani-Tur, G. Tur, and L. Heck. Research challenges and opportunities in mobile applications. Signal Processing Magazine, IEEE, 2011.

[2] G. Hinton, Li Deng, Dong Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. Signal Processing Magazine, IEEE, 2012.

[3] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, ICML '08. ACM, 2008.

[4] G.E. Dahl, Dong Yu, Li Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. Audio, Speech, and Language Processing, IEEE Transactions on, 2012.

[5] Tee Kiah Chia, Khe Chai Sim, Haizhou Li, and Hwee Tu Ng. Statistical lattice-based spoken document retrieval. ACM Trans. Inf. Syst., 2010.

[6] Yi-Chen Pan, Hung-Yi Lee, and Lin-Shan Lee. Interactive spoken document retrieval with suggested key terms ranked by a markov decision process. Audio, Speech, and Language Processing, 2012.

[7] S. E. Johnson, P. Jourlin, G.L. Moore, K.S. Jones, and P.C. Woodland. The cambridge university spoken document retrieval system. In Acoustics, Speech, and

Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on, 1999.

[8] Diane J. Litman and Scott Silliman. Itspoke: an intelligent tutoring spoken dialogue system. In Demonstration Papers at HLT-NAACL 2004, 2004.

[9] Stephanie Seneff and Joseph Polifroni. Dialogue management in the mercury flight reservation system. In Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems, 2000.

[10] Teruhisa Misu and Tatsuya Kawahara. Bayes risk-based dialogue management for document retrieval system with speech interface. Speech Commun., January 2010.

[11] Berlin Chen and Yi-Ting Chen. Extractive spoken document summarization for information retrieval. Pattern Recognition Letters, 2008.

[12] Lin shan Lee and B. Chen. Spoken document understanding and organization. Signal Processing Magazine, IEEE, 2005.

[13] Jerome R. Bellegarda. Statistical language model adaptation: review and perspectives. Speech Communication, 2004.

[14] Aaron Heidel and Lin-Shan Lee. Robust topic inference for latent semantic language model adaptation. In Proc. on ASRU, 2007.

[15] Tsung-Hsien Wen, Hung-Yi Lee, Tai-Yuan Chen, and Lin-Shan Lee. Personalized language modeling by crowd sourcing with social network data for voice access of cloud applications. In Proc. on IEEE SLT workshop, 2012.

[16] Hung-Yi Lee, Tsung-Hsien Wen, and Lin-Shan Lee. Improved semantic retrieval of spoken content by language models enhanced with acoustic similarity graph. In SLT, 2012.

[17] Yun-Nung Chen, Chia-Ping Chen, Hung-Yi Lee, Chun-An Chan, and Lin-Shan Lee. Improved spoken term detection with graph-based re-ranking in feature space.

In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pages 5644--5647. IEEE, 2011.

[18] Hung-yi Lee and Lin-shan Lee. Integrating recognition and retrieval with user feedback: A new framework for spoken term detection. In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pages 5290--5293. IEEE, 2010.

[19] Tsung-Hsien Wen, Hung-Yi Lee, and Lin-Shan Lee. Interactive spoken content retrieval with different types of actions optimized by a markov decision process. In Interspeech, 2012.

[20] Tsung-Hsien Wen, Hung-yi Lee, Pei-hao Su, and Lin-Shan Lee. Interactive spoken content retrieval by extended query model and continuous state space markov decision process. In ICASSP, 2013.

[21] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. Computer Speech and Language, 1995.

[22] luc Gauvain Jean and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. IEEE Transactions on Speech and Audio Processing, 1994.

[23] P. C. Woodland. Speaker adaptation for continuous density hmms: A review. In Proc. on ITRW on Adaptation Methods for Speech Recognition, 2001.

[24] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. Computational Linguistics, 1992.

[25] William Gale. Good-Turing Smoothing Without Tears. Technical report, AT&T Bell Laboratories, 1994.

[26] Frankie James. Modified kneser-ney smoothing of n-gram models modified kneser-ney smoothing of n-gram models. Technical report, 2000.

[27] Georey Hinton. A Practical Guide to Training Restricted Boltzmann Machines. Technical report, 2010.

[28] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. Journal of Machine Learning Research, 2003.

[29] Junho Park, Xunying Liu, Mark J. F. Gales, and Philip C. Woodland. Improved neural network based language modeling and adaptation. In Proc. on InterSpeech, 2010.

[30] Hai-Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and Francois Yvon. Structured Output Layer neural network language model. In Proc. on ICASSP, 2011.

[31] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In Proc. on InterSpeech, 2010.

[32] T. Mikolov, S. Kombrink, L. Burget, J.H. Cernocky, and S. Khudanpur. Extensions of recurrent neural network language model. In Proc. on ICASSP, 2011.

[33] T. Mikolov and G. Zweig. Context dependent recurrent neural network language model. In Proc. on IEEE SLT workshop, 2012.

[34] R.M. Iyer and M. Ostendorf. Modeling long distance dependence in language: topic mixtures versus dynamic cache models. IEEE Transactions on Speech and Audio Processing, 1999.

[35] Aaron Heidel, Hung-An Chang, and Lin-Shan Lee. Language model adaptation using latent dirichlet allocation and an efficient topic inference algorithm. In Proc. on InterSpeech, 2007.

[36] Marcello Federico. Efficient language model adaptation through mdi estimation. In Proc. on EuroSpeech, 1999.

[37] Ciprian Chelba and Frederick Jelinek. Structured language modeling. Computer Speech and Language, 2000.

[38] Hung yi Lee and Lin-Shan Lee. Enhanced spoken term detection using support vector machines and weighted pseudo examples. IEEE Transactions on Audio, Speech & Language Processing, 2013.

[39] Hung yi Lee, Tsung wei Tu, Chia-Ping Chen, Chao-Yu Huang, and Lin-Shan Lee. Improved spoken term detection using support vector machines based on lattice context consistency. In ICASSP, 2011.

[40] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. SIGIR '01. ACM, 2001.

[41] Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. Cambridge Journal, 1999.

[42] Richard Bellman. Dynamic programming. 1957.

[43] Stuart Dreyfus. Richard bellman on the birth of dynamic programming. Oper. Res., January 2002.

[44] Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy. Crowdsourcing systems on the world-wide web. Communications of the ACM, 2011.

[45] Munro and Robert. Crowdsourcing and language studies: the new generation of linguistic data. In Proc. on NAACL, 2010.

[46] Jingjing Liu, Scott Cyphers, Panupong Pasupat, Ian McGraw, and Jim Glass. A conversational movie search system based on conditional random field. In Proc. on InterSpeech, 2012.

[47] Ian McGraw, Scott Cyphers, Panupong Pasupat, Jingjing Liu, and Jim Glass. Automating crowd-supervised learning for spoken language systems. In Proc. on InterSpeech, 2012.

[48] Tsung-Hsien Wen, Aaron Heidel, Hung-Yi Lee, Yu Tsao, and Lin-Shan Lee. Recurrent neural network based language model personalization by social network crowdsourcing. In to be published.

[49] John Paolillo. The virtual speech community: Social network and language variation on irc. Journal of Computer-Mediated Communication, 1999.

[50] Devan Rosen and Margaret Corbit. Social network analysis in virtual environments. In Proc. on ACM Hypertext, 2009.

[51] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America, 2004.

[52] Hsu Bo-June and James Glass. Style and topic language model adaptation using hmm-lda. In Proc. on EMNLP, 2006.

[53] Tam Yik-Cheung and Tanja Schultz. Unsupervised language model adaptation using latent semantic marginals. In Proc. on InterSpeech, 2006.

[54] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. J. Mach. Learn. Res., 2003.

[55] Gregor Heinrich. Parameter estimation for text analysis. Technical report, 2004.

[56] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In KDD. ACM, 2008.

[57] Teh Yee Whye, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In NIPS, 2006.

[58] Hsu Winston H., Lyndon S. Kennedy, and Chang Shih-Fu. Video search reranking through random walk over document-level context graph. In MULTIMEDIA. ACM, 2007.

[59] Chen Yun-Nung, Chen Chia-Ping, Lee Hung-Yi, Chan Chun-An, and Lee Lin-Shan. Improved spoken term detection with graph-based re-ranking in feature space. In ICASSP, 2011.

[60] Chen Yun-Nung, Yu Huang, Ching-Feng Yeh, and Lee Lin-Shan. Spoken lecture summarization by random walk over a graph constructed with automatically extracted key terms. In Interspeech, 2011.

[61] Andreas Stolcke. Srilm - an extensible language modeling toolkit. In Proc. on Spoken Language Processing, 2002.

[62] Steve J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. The HTK Book Version 3.4. Cambridge University Press, 2006.

[63] Mikael Bod´en. A guide to recurrent neural networks and backpropagation. 2002.

[64] Yangyang Shi, Pascal Wiggers, and Catholijn M. Jonker. Towards recurrent neural networks language models with linguistic and contextual features. In Proc. on InterSpeech, 2012.

[65] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult, 1994.

[66] Xunying Liu, Mark J. F. Gales, and Philip C. Woodland. Improving lvcsr system combination using neural network language model cross adaptation. In Proc. on InterSpeech, 2011.

[67] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. Computer, 2009.

[68] Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An Introduction to Latent Semantic Analysis. Discourse Processes, 1998.

[69] Keh-Jiann Chen and Shing-Huan Liu. Word identification for mandarin chinese sentences. In Proceedings of the 14th conference on Computational linguistics - Volume 1, COLING '92. Association for Computational Linguistics, 1992.

[70] Wei-Yun Ma and Keh-Jiann Chen. A bottom-up merging algorithm for chinese unknown word extraction. In Proceedings of the second SIGHAN workshop on Chinese language processing - Volume 17, SIGHAN '03. Association for Computational Linguistics, 2003.

[71] Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukas Burget, and Jan Cernocky. Rnnlm - recurrent neural network language modeling toolkit. In Proc. on ASRU, 2011.

[72] George A. Miller. Wordnet: A lexical database for english. Communications of the ACM, 1995.

[73] Yi-Lun Yang. Plutalk: A spoken social network application with speaker adaptation and error correction. 2012.

[74] L. Bahl, P. Brown, P.V. De Souza, and R. Mercer. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86., 1986.

[75] D. Povey and P.C. Woodland. Minimum phone error and i-smoothing for improved discriminative training. In Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on, 2002.

[76] David Robins. Interactive information retrieval: Context and basic notions. Informing Science Journal, 2000.

[77] Ian Ruthven. Interactive information retrieval. Annual Review of Information Science and Technology, 2008.

[78] Teruhisa Misu and Tatsuya Kawahara. Speech-based interactive information guidance system using question-answering technique. In ICASSP, 2007.

[79] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. CIKM '01. ACM, 2001.

[80] Tao Tao and ChengXiang Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In SIGIR'06, 2006.

[81] Thomas Hofmann. Probabilistic latent semantic indexing. In ACM SIGIR, 1999.

[82] Richard Bellman and Sherman Dreyfus. Functional approximation and dynamic programming. Mathematical Tables and Other Aids to Computation, 1959.

[83] Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. Journal of Machine Learning Research, 2008.

[84] Csaba Szepesvári and Rémi Munos. Finite time bounds for sampling based fitted value iteration. ICML '05. ACM, 2005.

[85] Ben He and Iadh Ounis. Query performance prediction. Inf. Syst., 2006.

[86] Ying Zhao, Falk Scholer, and Yohannes Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In Advances in Information Retrieval. 2008.

[87] Yun Zhou and W. Bruce Croft. Query performance prediction in web search environments. SIGIR '07. ACM, 2007.