國立臺灣大學電機資訊學院資訊工程學研究所

碩士論文

Graduate Institute of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

中英雙語環境下使用詞群及隨機森林的語言模型調適

Language Model Adaptation for Mandarin-English Code-Mixed Lectures Using Word Classes and Random Forests

黃昭瑜

Huang, Chao-Yu

指導教授：李琳山 博士

Advisor: Lin-shan Lee, Ph.D.

中華民國一百年六月

June, 2011

國立臺灣大學碩士學位論文
口試委員會審定書

中英雙語環境下使用詞群及隨機森林的語言模型調適

Language Model Adaptation for Mandarin-English Code-Mixed Lectures Using Word Classes and Random Forests

本論文係 **黃昭瑜** 君（學號R98922053）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 100 年 6 月 20 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

（指導教授）

陳信宏　　王小川

簡仁宗　　鄭秋豫

系 主 任　呂育道

誌謝

能夠順利完成碩士論文研究， 首先要感謝的就是我的指導教授李琳山老師。 老師提供了充分的研究資源以及自由發展的研究環境， 讓實驗室的同學們能夠心無旁騖的進行各種研究。 在 meeting 時， 老師總是十分專注的聽著台上同學的報告， 並在我遇到困難時提供意見， 讓我研究得以順利進行。 和老師合寫會議論文的過程中， 亦讓我發現了許多盲點， 使我學到如何嚴謹的來表達自己的研究成果， 這些都是求學至今獨一無二的收穫。

此外，在實驗室的討論風氣十分盛行， 常常在和實驗室同學的討論過程中激發出彼此的想像力， 進而找到許多研究方向， 這一切都要感謝各位學長以及同學們的教導與鼓勵。 從和小銘及牛的合作專題研究的那一刻起， 便一腳踏入了語音研究的大門。 在哲光學長的身上我學到了專業研究人員的精神、 在唐玥蓮學姊和 Aaron 學長的幫助之下我確立了研究題目的主要方向、 在向皓中 murmur 的過程中我確立了實驗的步驟、 在看歐吉桑有條不紊的 comment 時我感受到名師的風範、 在看宏毅哥和研究盡情的戰鬥時我感受到無比的熱血、 在看小安報告的時候我感受到强者的氣息、 總是在實驗室認真跑實驗的阿邦、 温文儒雅的馬雅、 强者儂大和黃宥、 愛點人肩膀的小蘋果、 高音渾厚的鋒哥、 靦腆的左逼、 歡樂的李尚文、 帥氣到爆表的 Jeff 和李杰、 義氣相挺的涂涂、 總是適時幫助我們的金燕， 有你們的陪伴， 在實驗室的生活每一天都多采多姿。

最後，感謝我的家人、 B94 的好捧油們、 彰友會温馨的大家、 魔術社嘴砲的 low 人們、 以及可愛的佳盈。

謝謝大家！

摘要

語言模型在語音辨識中一向扮演著極為重要角色， 然而自然語言的語法千變萬化， 隨著國際化的風潮， 人們日常生活中的語言也由單語轉向雙語或多語， 於是雙語混合的語言模型變成一個迫切需要卻又難解的問題。

雖然雙語在現今社會十分流行， 但可收集到的雙語語料和單語相較之下仍是九牛一毛， 於是在本論文中使用基於詞群之 N 連語言模型來辨識雙語混合語料。 藉由同時使用統計學和語言學的方式建立雙語詞群， 勾勒出雙語詞彙之間的互動模式， 並以此建立語言模型， 以補益雙語語料的不足。

基於詞群之 N 連語言模型 是將 N 連事後機率中欲估測的歷史詞串和目標詞都加以分群以共享資訊的方法， 較為粗糙。 相較之下， 決策樹語言模型則是僅將歷史詞串分群以共享資訊。 同時，可集合眾多的隨機決策樹， 假設一棵樹能達到的是區域最佳解， 那麼在一片森林中，應會有機會接近全域最佳解， 這就是隨機森林語言模型。

為了能夠使用各種背景語料來強化語言模型， 本論文亦使用基於隨機森林的語言模型調適法以進行調適。 首先使用大量的背景語料來生成隨機決策樹後， 再用目標課程的訓練語料來修剪樹， 使得經修剪過後的樹能更貼近目標課程， 接著集結經由各領域的背景語料而來的許多片隨機森林， 形成眾林之林語言模型。 和最初未經調適的基礎語言模型相比， 其絕對的辨識正確率進步約 1.78%。

關鍵字　語言模型、雙語混合、詞群、隨機森林、語言模型調適

Contents

圖目錄

表目錄

第一章　導論

1.1　研究動機

由於近年來超高速網路的發展以及多媒體播放軟體介面的普及，有越來越多的影音資訊提供使用者在終端裝置上閱覽使用。其中，有許多大學將其所開設的各類課程錄音、錄影後，上傳到網路上供人瀏覽，使學生和一般民眾得以經由這些線上影音課程進行更有效率的學習，如知名播放軟體 iTunes U 上，就提供超過 350,000 個世界各地大學的課程影片、投影片及其他各種資源 [1]。

儘管如此，一門課程整個學期下來常常動輒數十個小時，使用者往往不得不從第一章的影片一路看到最後一章。有些章節的內容使用者可能早就知道了，但想跳過卻不知下一個章節的影片要跳到哪邊；有些使用者讀到後面的章節時早已忘掉了前面的章節，此時要回過頭來複習卻又不知從何找起，於是學習的效率就因此而大打折扣。

於是，若能利用自動語音辨識技術 (automatic speech recognition, ASR) 先對整門課程進行前端辨識處理，再將辨識後的結果提供給後續資料檢索 (information retrieval)、關鍵字抽取 (key term extraction) 及文件摘要 (document summarization) 等技術進行更進一步的分析，便能讓使用者充分使用系統所提供的各種查詢及瀏覽功能來增進學習的效率。由此可知，一個好的辨識結果對於整個課程系統而言是最初步且必要的。

在傳統上由於自動語音辨識技術的結構相當複雜，故通常會藉由貝氏定理 (Bayes' Theorem) 將問題拆解成聲學模型 (acoustic model, AM) 與語言模型 (language model, LM) 兩個子問題，再分別使用統計方法替兩個子問題建立模型。

本論文著力在語言模型上的方法與改進。

1.2 相關研究

傳統大字彙連續語音辨識 (large vocabulary continuous speech recognition, LVCSR) 所處理的語料， 大多是屬於朗讀式語音 (read speech) 或是經過規劃的語音 (planned speech) ， 如新聞和廣播等等。 而課程語音則是偏向於自發性語音 (spontaneous speech) ， 其內容充滿著停頓、重複的語句、語助詞和各種不流暢 (disfluencies) 的語句。 再加上課程的主題與內容和一般生活中的語句相距甚遠， 於是很難收集到足夠的訓練語料以建立統計模型， 故在課程語音的辨識上往往比一般的大字彙連續語音辨識更為困難。 但這也使得有越來越多的研究單位投身其中， 如麻省理工學院 [2] [3] [4] 和康乃爾大學 [5] 均建立其線上影音課程學習系統。

然而，上述系統處理的皆為單語的課程語料。 在非英語系國家的大學課程中， 課程的內容往往是雙語混合 (code-mixed) 的， 其特性為大部分的語句仍是使用本國國語， 但在專業名詞及部分口語則偏向於使用英語，如 「這是一個使用 fourier transform 的好處」、「最好的 solution 是這個 viterbi algorithm」等類似的語句 在雙語混合的課程中十分常見。

目前對於雙語語料的研究也不在少數， 如香港中文大學的廣東話和英語的雙語混合辨識系統 [6]、 新竹交通大學的國客雙語混合辨識系統 [7] 等， 皆是針對雙語的特性考慮辨識的問題。 而本論文所研究的臺大課程系統則同時處理了課程語音與中英雙語混合的問題。

現今的單語辨識系統已具備一定的辨識率， 但若單獨使用兩個單語語言模型來處理雙語辨識卻又顯得不切實際， 故該如何在現有的基礎上建立雙語混合語言模型， 以及該用怎樣的方式來補強稀疏的課程語料， 皆是本論文主要的研究方向。

1.3 主要研究方法及成果

首先針對資料稀疏性 (data sparseness) 的問題， 在本論文中採用了基於詞群之 N 連語言模型 (class-based N-gram language model) [8]， 其中詞群的分群演算法主要採用了下列兩種不同的做法： 一是利用統計， 使用資料驅動 (data-driven) 的自動化分群演算法， 讓詞群間的平均相互資訊 (mutual information) 最大化； 二是考慮中英混合的特性， 先將訓練集 (training set) 裡的文字做好詞性標記 (part-of-speech tagging) 後， 再依照詞性做為詞群分類的準則。 為了讓基於詞群之 N 連語言模型更具强健性， 本論文亦採用了傑氏 (F. Jelinek) 的線性內差調適法 (linear interpolation) [9] [10]， 使其和經過聶氏平滑法 (Kneser-Ney Smoothing) [11] [12] 的 N 連語言模型做相等權重的線性內差。 最後和經過聶氏平滑法的 N 連語言模型相比， 絕對的辨識正確率在中文的部份進步約 0.42% ， 英文的部份則為 1.42% 。

此外，本論文亦使用了隨機森林 (random forest, RFs) 語言模型 [13]， 其利用快速交換演算法 (fast exchange algorithm) 將歷史詞串分群 [14]， 建立衆多隨機決策樹 (randomized decision trees) 後， 最後集合所有隨機決策樹而成為隨機森林語言模型。 和經過聶氏平滑法的N連語言模型相比， 其絕對的辨識正確率在中文的部份進步約 1.3% ， 英文的部份則為 3.4% 。

最後，為了能夠充分使用其他領域的各種語料來補强稀疏的課程語料， 本論文亦分別使用了傳統的模型內差調適法、隨機森林語言模型調適法 [15]， 以及衆林之林 (forest of random forests, FRF) 語言模型 [16] 來調適課程語料。 和傳統的方法相比之下，衆林之林語言模型絕對的辨識正確率在中文的部份進步約1.2%， 英文的部份則為2.52%。

1.4 章節安排

本論文接下來的章節安排簡述如下：

- **第二章：**

  簡介雙語混合語料的特性、 傳統語音辨識的架構、 語言模型的背景知識， 以及實驗所使用的資源與環境。

- **第三章：**

  介紹基於詞群之 N 連語言模型和本論文主要的詞群分群演算法， 並以實驗探討分析各種詞群語言模型的表現。

- **第四章：**

  介紹決策樹語言模型的基本架構、 隨機決策樹的生長 (growing) 與修剪 (pruning) 演算法， 和如何產生隨機森林語言模型。 最後以實驗測試隨機森林語言模型的辨識正確率。

- **第五章：**

  簡介語言模型調適的理論基礎， 和使用隨機森林以調適語言模型的方法， 以及改進自隨機森林的衆林之林語言模型調適。 最後比較各種未經調適和已調適過後的語言模型。

- **第六章：**

  本章為論文總結與建議未來可發展之方向。

第 二 章　理論背景與實驗環境介紹

本章就語音辨識中所使用的語言模型作一簡述。 首先在 2.1 節， 介紹本論文中所使用的雙語語料的特性， 而 2.2 節， 則簡介目前的大字彙連續語音辨識架構， 以及語言模型在其中所扮演的角色。 接著在 2.3 節，介紹統計式語言模型 (statistical language model, SLM) 和其中最多人採用的 N 連語言模型， 以及評估語言模型好壞的標準—混淆度的定義。 以及在 2.4 節，描述本論文的實驗環境與語料。 最後 2.5 節是本章總結。

2.1　雙語混合 (code-mixing) 的介紹

目前在語言學上對於雙語系統已有許多相關研究 [17] [18]。 一般在學理上，將母語定義為主位語言 (host language) ， 而第二外語則定義為客位語言 (guest language) 。 同時雙語系統又可細分為雙語切換 (code-switching) 和雙語混合 (code-mixing) 。

當在一句話中僅用了一種語言， 但在句子和句子間卻出現了語言的切換 (inter-sentential switching) ， 此時我們將這種現象定義為雙語切換。 如：「今天天氣很好，let's take a walk」便是一種雙語切換。 其中主位語言和客位語言的句子間或許存在著因果關係， 但是兩者之間的互動並不明顯， 於是，往往將其視為一種語言識別 (language identification, LID) 的問題處理 [19]。

然而當一句話中用了不只一種語言， 即句子內部就有語言轉換的發生 (intra-sentential switching) ， 此時我們將這種現象定義為雙語混合。 如：「這是一個使用 fourier transform 的好處」便是一種雙語混合。 此時客位語言常常以詞 (word) 或片語 (phrase) 鑲嵌於主位語言的句子中， 這兩者之間的互動是十分豐富

的， 且也較符合一般大眾的用法。 而本論文主要研究的語料即屬於雙語混合，底下將針對雙語混合的特性進行探討。

2.1.1 語言差異

由於大多數人皆對其母語較為熟悉， 因此當說到客位語言時， 常常會發生「外語母語化」的情形， 也就是使用主位語言的方式來表達客位語言。 有些會用主位語言的腔調說客位語言， 如：「這支唉鳳的功能好强大」便是將「iPhone」的音唸為「唉鳳」。 有些更明顯的則是會將主位語言的語句結構套用在客位語言上頭， 譬如：「晚餐吃水餃 O 不 OK 呢？」 便是將英文的「OK」套用到中文的「好不好呢」這樣的語法中。 這類句子的意思對於使用雙語混合的主位語言語者而言是很容易理解的， 但是當客位語言的語者聽到這樣的句子大概只能說「一點也不OK。」

於是，若只是單獨為兩種語言分別建立模型， 到最後才想要找方法來合併他們以得到雙語混合的模型， 是相當不切實際的。 因此，應直接由雙語混合語料著手，觀察其特殊的結構， 並建立雙語混合模型， 會是較合理的做法。

2.1.2 語言借用

首先在觀察雙語語料後， 不難發現大部分的客位語言常被用來填補主位語言所沒有的詞彙， 又或者是藉以加强句子的語氣， 這類的現象稱為語言借用。 被借用的客位語言通常以詞或片語為主， 例如：「你有沒有什麼好的 idea？」 便是借用了英文詞彙「idea」來加强表達中文詞彙「想法」的意思。

又或者是使用一些專有名詞如「Hidden Markov model」時， 由於「隱藏式馬可夫模型」此類的翻譯可能並不夠原汁原味， 故大部分的人還是會傾向於保留原

文， 甚至當其有縮寫時，基於方便也很有可能會直接使用縮寫， 如：「HMM」便是「Hidden Markov model」的縮寫， 而前者出現的次數往往高於後者。

然而語言借用的範圍及多寡又常因語者本身的習慣或是談話的主題及內容而有所不同。 但反過來想，若語者真有其習慣的用法， 或某些談話的主題內容就是會出現一些特定的借用者， 那我們就可以收集其相關的雙語混合語料， 並想辦法從中建立良好的雙語混合語言模型。

2.1.3 雙語混合的課程語料

本論文所研究的語料， 是臺灣大學電機系教授所開設的大學課程， 其內容充滿著中英混合的語句， 底下將其歸納出下列幾項特性：

- **課程中專業術語的呈現：**

  根據觀察，課程語料中的專業術語常以英文方式呈現， 如：「其實就有一個非常有名的在 information theory 裡面有所謂的叫做 binary entropy function。」 又或者原文與縮寫並列， 如「那我的那些 HMM 那些 Hidden Markov model 呢就是所有的這些基本的音。」 甚至縮寫「HMM」出現的次數是原本「Hidden Markov model 」的兩倍之多。

- **語者慣用之口頭禪、連接詞和語助詞：**

  在講述課程內容時， 常會因為語者的習慣而有許多口頭禪、連接詞以及在語句開頭或結尾的語助詞。 以本論文所處理的一門課程為例， 該門課程總共有 34,475 個句子，裡頭「OKAY」出現的次數總共有 443 次， 其中就有 296 次「OKAY」出現在句首、 76 次「OKAY」出現在句尾。

- **數學方程式的呈現：**

  當課程中需要講解到數學方程式時， 很容易以英文表示式中代數的部分，

如：「A X I 加上 B 減掉 Y I 的平方等於 MINIMUM」。 但此類的語料十分難以獲得， 導致很難使用統計的方式去訓練模型。 故常見的一個做法是藉由觀察數學式子的文法結構， 使用詞群或其他文法的組合來增進辨識率。

2.2 雙語大字彙連續語音辨識系統簡介

自動語音辨識系統在過去數十年間已逐漸發展成熟， 而雙語系統在過去數年間亦有著許多發展 [6] [19] [20] [21]。 目前本實驗室的做法是從語料、辭典乃至音素集皆同時處理了中、英文的部份， 使其並陳在系統之中 [22]。

現今統計式的大字彙連續語音辨識系統， 大致可以想成是將一段連續語音的數位訊號 (signal) 解碼為文字轉寫 (transcription) 的過程， 而中英雙語混合辨識系統之整體架構如圖 2.1 所示：

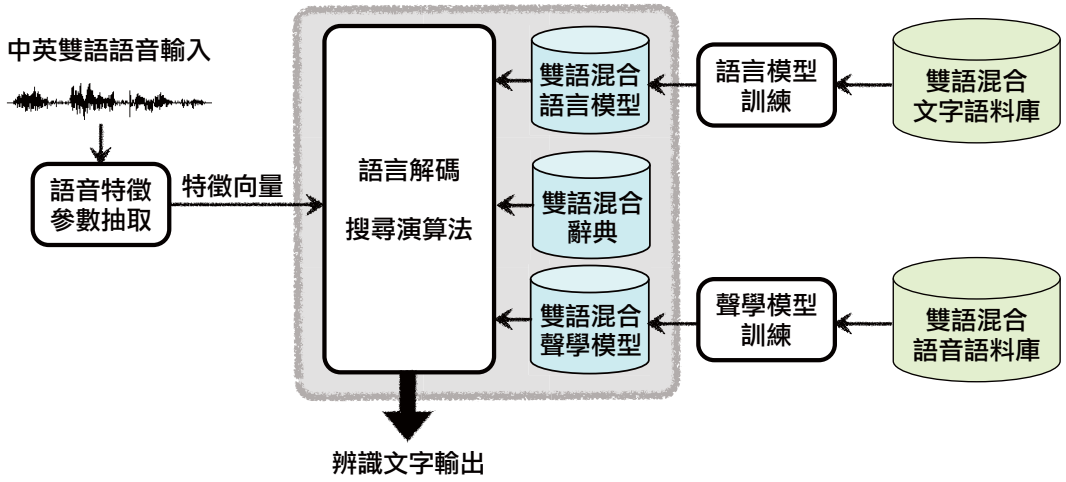

圖 2.1: 中英雙語混合辨識流程圖

上圖可將整個流程細分為三部分： 前端語音特徵抽取 (feature extraction)、 聲學模型和語言模型的建立、 和最後結合聲學和語言知識以及包含中音雙語音素集之雙語混合辭典， 以解碼辨識出結果。

2.2.1 特徵抽取

特徵抽取的目標，乃是從欲辨識的聲波訊號裡，擷取其中重要的部份，並經過壓縮、消除雜訊等程序，讓經過特徵抽取的語音訊號方便後端程式處理。

聲音訊號處理上最常用的特徵抽取方法，主要透過計算梅爾倒頻譜係數 (Mel-frequency cepstral coefficients, MFCC) 所獲得。首先將連續語音訊號切割為時間軸上一連串相疊的音框 (frame)，使音框內包含短時間內穩定且接近的訊號。針對人耳聽覺特性，每個音框經過 18 個梅爾倒頻譜濾波器與餘弦轉換後，即得 13 維特徵向量，並在時間軸上取其一階及二階導數的微分，得到總共 39 維的梅爾倒頻譜特徵向量 (MFCC feature vector)。

2.2.2 音素集

音素乃最小的發音單位，也是每個有意義詞的基本聲學單位，而音素集則是以最小的音素集合涵蓋所有字詞發音，故在雙語混合辨識系統上，音素集必須同時考慮中文和英文之發音特性。

傳統中文辨識系統上的音素集普遍使用「聲母-韻母 (initial-final)」之單音節 (syllable) 結構，若對照至英文音素 (phoneme) 的組成，聲母相當於是由一個子音音素所構成 (空聲母除外)，而韻母則是由一到三個音素，亦稱類音素(phone-like units, PLU) 所構成。

為使辨識系統能同時處理中英文發音特性，故本論文之雙語辨識系統中所使用的音素集同時包含了中英文音素，共記 32 個中文音素及 39 個英文音素以及靜音 (sil) 和短暫停音 (short pause)，如表 2.1 。

Plosive	CH_b CH_d CH_g CH_p CH_t CH_k EN_B EN_D EN_G EN_P EN_T EN_K
Fricative	CH_h CH_s CH_s' CH_ts CH_ts' CH_dz CH_dz' CH_Z' EN_F EN_HH EN_S EN_TH EN_SH EN_CH EN_JH EN_DH EN_V EN_ZH EN_Z
Voiced Consonant	CH_n CH_n# CH_N# CH_@ CH_@' CH_l EN_M EN_N EN_NG EN_ER EN_L EN_R
Vowels	CH_dz| CH_s| CH_ts| CH_y CH_o CH_U CH_U' CH_i CH_e CH_# CH_E CH_u EN_AO EN_AH EN_AY EN_AW EN_AA EN_AE EN_IH EN_IY EN_Y EN_EY EN_EH EN_OW EN_OY EN_UW EN_UH EN_W
Others	sil sp

表 2.1: 中英文音素集：中英文音素分別以"CH"及"EN"開頭以識之

2.2.3 辭典

辭典定義了中英文字詞與發音的對應關係， 使辨識時解碼器 (decoder) 可將音素串列轉化為可能的字詞串列， 故辭典所包含的詞必須盡量貼近測試語料， 以避免當欲辨識的字詞為辭典外詞彙(out of vocabulary, OOV) 時， 因解碼器無法將音素串列對應至辭典外詞彙， 而導致這些詞彙無法辨識的情形。

中文裡有意義文字的基本單位為單字 (charactor)， 故藉由在辭典中盡可能的包含單字和詞後， 再以辭典將文字語料斷詞， 便可避免辭典外詞彙的出現， 甚至可進一步透過 Pat-Tree 抽辭程式將許多單字合併為詞， 以長詞為優先， 縮減辭典大小進而降低語言模型之複雜度。

而辭典中英文的部份，以課程系統為例，可先將目標課程的教科書中的專有名詞先列舉出來，再加上日常生活中常見詞彙以及語者習慣之發語詞，便可降低英文辭典外詞彙的出現。

2.2.4 辨識解碼

令 $\bar{X}$ 為雙語混合語音 X 透過特徵抽取而得到的 39 維梅爾倒頻譜特徵向量，在數位語音處理的環境下，可將其視為一連串的時間序列 (time-series sequence)，亦即在時間軸上 T 個音框所對應的語音 x_t 及其所抽取出其特徵向量 $\bar{x}_t$，序列的每一單元則代表當時聲音的資訊：

$$\bar{X} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_T), \tag{2.1}$$

也就是中英雙語語音輸入經語音特徵參數抽取後之特徵向量。將特徵向量 $\bar{X}$ 輸入解碼器中辨識後，得到辨識文字輸出：

$$W = (w_1, w_2, ..., w_m) = w_1^m \tag{2.2}$$

即為經辨識後長度為 m 的中英雙語混合詞串 (word string)。

於是，$P(W|\bar{X})$ 可表示為給定一連串欲辨識的特徵向量 $\bar{X}$，並將其解碼為 W 的事後機率。所以整個辨識解碼的過程可視為是在給定聲音資訊 $\bar{X}$ 的條件下，於所有可能的詞串 W 中找出最有可能的詞串 W^* 作為辨識結果。因此我們以最大事後機率 (maximum a posterior probability, MAP) 來求得最佳解：

$$\begin{aligned} W^* &= \arg\max_W P(W|\bar{X}) \\ &= \arg\max_W \frac{P(W)P(\bar{X}|W)}{P(\bar{X})} \\ &= \arg\max_W \frac{P(W)P(\bar{X}|W)}{\sum_{W'} P(W')P(\bar{X}|W')} \\ &= \arg\max_W P(W)P(\bar{X}|W). \end{aligned} \tag{2.3}$$

故整個大字彙連續辨識系統便以式 (2.3) 為基礎， 將原本的事後機率 $P(W|\bar{X})$ 拆成 $P(W)$ 和 $P(\bar{X}|W)$ 兩個子問題。

其中 $P(\bar{X}|W)$ 可視為是聲學層次的處理， 其機率可由聲學模型獲得。 透過雙語混合語音語料庫訓練跨詞三連音素 (triphone) 聲學模型， 並由辭典之中英文字詞與其發音音素之對應， 來判別給定中英混合詞串 W 時， $\bar{X}$ 之音素串列為 W 之發音的機率。

而 $P(W)$ 的部份， 由貝氏機率的觀點， 則為語者所說出的中英雙語混合詞串 W 的事前機率 (prior probability)。 故中英雙語混合語言模型所扮演的角色， 是當給定任一中英雙語混合詞串 W 時， 能夠估算出其發生機率的高低。 底下將對 $P(W)$ 的計算做進一步的探討。

2.3 統計式語言模型

語言模型大致可分為以文法為基準 (grammar-based) 的語言模型， 以及統計式語言模型 (statistical LM)。 但由於口語的使用往往偏離文法甚多，導致找出合適文法的困難度增加。 相對而言，統計式語言模型只要擁有夠多的訓練語料， 便能獲得不錯的辨識率， 所以現今普遍採用統計式語言模型。

語言模型主要欲估算的是詞串的機率 $P(W)$ 。 根據式 (2.2) ，可藉由鏈鎖法則 (chain rule) 將其拆解成：

$$\begin{aligned} P(W) &= P(w_1^m) \\ &= P(w_1) \times P(w_2|w_1) \times P(w_3|w_1^2) \times ... \times P(w_m|w_1^{m-1}) \\ &= \prod_{i=1}^{m} P(w_i|w_1^{i-1}) = \prod_{i=1}^{m} P(w_i|h_i). \end{aligned} \tag{2.4}$$

其中h_i 則是詞 w_i 出現之前的歷史詞串 (history) ： $w_1, w_2, ..., w_{i-1}$ ， 故 $P(w_i|h_i)$ 可

看成是給定歷史詞串 h_i 時， 詞 w_i 出現的條件機率。 於是，經由式 (2.4) ，我們便可從計算 $P(w_i|h_i)$ 中估測 $P(W)$ 。

然而在實際辨識時， 會發現 $P(w_i|h_i)$ 所要估測的參數量隨著 i 的長度呈指數成長， 其大小為 $|\mathcal{V}|^i$ ， $|\mathcal{V}|$ 為辭典 $\mathcal{V}$ 所包含的總詞數大小。 於是， 常見的做法是使用等價類別分類函數 (equivalence class classifier) ： $\Phi(\cdot)$ ， 將 h_i 對應到其等價類別 (equivalence class) ： Φ_i ， 即 $\Phi_i \equiv \Phi(h_i)$ 。 如此一來，可以使數個不同的歷史詞串對應到同一個等價類別中，以簡化計算量。 所以將此套用在式 (2.4) 後，可得：

$$P(W) = \prod_{i=1}^{m} P(w_i|h_i) \approx \prod_{i=1}^{m} P(w_i|\Phi(h_i)) = \prod_{i=1}^{m} P(w_i|\Phi_i). \tag{2.5}$$

於是如何決定出合適的 $\Phi(\cdot)$ 以及如何估測 $P(w_i|\Phi_i)$ 便成為語言模型的重要課題。

隨著 $\Phi(\cdot)$ 的不同， 語言模型可特化為 N 連語言模型或是決策樹語言模型 (decision tree LM) 等許多種變形。 而 $P(w_i|\Phi_i)$ ，在傳統上則是使用 最大相似度估測法 (maximum likelihood estimating, MLE) 計算其值， 其做法是在離散空間 (discrete space) 上去估算目標詞串在訓練集中出現的次數， 同時進行標準化 (normalization) ， 其數學式為：

$$P(w_i|\Phi_i) = \frac{C(\Phi_i, w_i)}{\sum_{w_j \in \mathcal{V}} C(\Phi_i, w_j)}, \tag{2.6}$$

其中 $C(\cdot)$ 表示觀察到特定事件 $\cdot$ 所發生的次數， 故 $C(\Phi_i, w_i)$ 表示在訓練集中出現 Φ_i 且其後面又正好相鄰著詞 w_i 的次數， 而分母則是 Φ_i 後面出現所有可能的相鄰詞之次數的總和， 以使機率和為 1 。 因此， $P(w_i|\Phi_i)$ 可視為是一種相對頻率的估算。

2.3.1 N 連語言模型

在目前的語言模型中，最被廣為使用的就是 N 連語言模型。 由於其具有簡單又具效率的特性， 往往成為初建系統時的基礎模型。 而 N 連語言模型最主要的精神， 便是藉由馬可夫假設 (Markov assumption) 將等價類別分類函數 $\phi(\cdot)$ 定義為：

$$\Phi_i = \Phi(h_i) = \Phi(w_1^{i-1}) = w_{1-N+1}^{i-1}, \tag{2.7}$$

也就是假設詞 w_i 的出現只和其前面的 $N-1$ 個詞有關。

以三連語言模型 (trigram LM) 為例，將 $N = 3$代入式 (2.5) 及式 (2.7) 得到：

$$P(W) \approx P(w_1) \times P(w_2|w_1) \times \prod_{i=3}^{m} P(w_i|w_{i-2}, w_{i-1}). \tag{2.8}$$

而 $P(w_i|w_{i-2}, w_{i-1})$ 則可由式 (2.6) 及式 (2.7) 得到：

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{\sum_{w_j \in \mathcal{V}} C(w_{i-2}, w_{i-1}, w_j)}. \tag{2.9}$$

於是，只要找出訓練集中所有長度為 3 的詞串， 並統計他們所出現的次數， 便可計算出 $P(W)$。

然而，經過式 (2.7) 的簡化後， $P(w_i|w_{i-N+1}^{i-1})$ 要估計的參數仍多達 $|\mathcal{V}|^N$ ， 對於一般動輒上萬字的辭典來說， 訓練語料還是很有可能不夠多， 除了統計出來的模型可能因為樣本數的不足而欠缺代表性， 更有可能因為欲統計的詞串 W 中 出現了一些在訓練集中從沒出現過的未見事件 (unseen event) ， 使得式 (2.9) 中分子的部份為零， 並連帶導致式 (2.8) 中的 $P(W)$ 歸零。

在真實世界中， 詞串 W 裡包含未見事件的詞是十分常見的情形， 若因此將其機率估測為零， 將導致 W 永遠不可能在式 (2.3) 中被辨識出來， 將嚴重影響辨識的正確率。 以上的情形稱之為 N 連語言模型的稀疏性 (sparseness) 。

2.3.2 統計式語言模型的平滑化

為了解決 N 連語言模型的稀疏性， 我們使用語言模型的平滑化 (smoothing) ， 以調整式 (2.6) ， 使包含未見事件的機率經平滑化的估測後得以被賦予一非零值，以避免$P(W)$ 為零。 目前平滑法大致可分為線性內差法 (linear interpolation) [23] 和退化平滑法 (back-off smoothing) [24] [25]， 以及其中最被廣為使用的聶氏平滑法 (Kneser-Ney smoothing) [11]。

在聶氏平滑法中，以 N 連語言模型為例，其主要的方程式是：

$$\begin{aligned} P_{KN}(w_i|w_{i-N+1}^{i-1}) = & \frac{max(C(w_{i-N+1}^{i}) - D, 0)}{C(w_{i-N+1}^{i-1})} \\ & +\lambda(w_{i-N+1}^{i-1})P_{KN}(w_i|w_{i-N+2}^{i-1}), \end{aligned} \tag{2.10}$$

其中 D 是一折扣常數 (discount constant) ， 而 $\lambda(w_{i-N+1}^{i-1})$ 則是接收這些折扣的累計值。 D 值的決定方式是使用留一法 (leave-one-out) ， 使得 $D = \frac{n_1}{n_1+2n_2}$ ， 其中 n_1 是訓練集中出現次數恰好是 1 次的所有 N 連詞的總個數， 而 n_2 則是出現次數恰好是 2 次的所有 N 連詞的總個數。

接著，為了讓機率總和為 1 ，定義 λ 為：

$$\lambda(w_{i-N+1}^{i-1}) = \left(\sum_{w_j} \|C(w_{i-N+1}^{i-1}, w_j) > 0\| \right) \frac{D}{C(w_{i-N+1}^{i-1})}, \tag{2.11}$$

其中 $\|\cdot\|$ 為真假值函數， 表示當條件 $\cdot$ 為真時，其值為一，否則為零。 於是，在式 (2.10) 中 N 連次數高於 D 而因此被折扣掉的部分， 最後都會累積在 λ 中， 並將這些折扣依照每個 N 連詞的退化 $N-1$ 連的機率當做權重， 重新分配回的所有 N 連詞當中， 以達到平滑化的效果。

至於式 (2.10) 中的退化 $N-1$ 連機率則為：

$$P_{KN}(w_i|w_{i-N+2}^{i-1}) = \frac{\sum_{w_{i-N+1}} \|C(w_{i-N+1}^{i-1}, w_i) > 0\|}{\sum_{w_{i-N+1}, w_j} \|C(w_{i-N+1}^{i-1}, w_j) > 0\|} \tag{2.12}$$

或是繼續依照式 (2.10) 進行遞迴求值， 直到退化至單連 (unigram) 機率為止。

2.3.3 混淆度

一般來說，若要判定語言模型的好壞， 可將其直接使用在語音辨識上，測試辨識正確率是否進步。 但是由於在真實辨識時影響正確率的變因很多， 所以一般也常使用混淆度 (perplexity, PPL) 和辨識正確率並陳。

混淆度的計算不牽涉到辨識系統， 僅需使用未用來訓練語言模型的測試集 (testing set) 來估算語言模型的好壞。 其主要的計算式如下：

$$PPL = exp\left(-\frac{1}{N_w}\sum_{s=1}^{N_s}\sum_{i=1}^{n_s}\log\left(P(w_i|w_1^{i-1})\right)\right), \tag{2.13}$$

表示測試集共有 N_w 個詞，N_s 個句子， 每個句子 s 又包含 n_s 個詞， 於是由式 (2.13) 可看出混淆度為測試集中所有詞的事後機率的幾何平均倒數。

而由資訊理論 (information theory) 來看， 可發現 $-\frac{1}{N_w}\sum_{s=1}^{N_s}\sum_{i=1}^{n_s}\log\left(P(w_i|w_1^{i-1})\right)$ 其實就是一種熵 (entropy) 的概念， 而熵在壓縮演算法中又可看成是每個單位平均要使用多少位元進行編碼。 故熵的值越低， 所需要的編碼量就越少。

若將熵應用在語言模型上， 則可想成是一個等機率辭典中平均每個詞需要多少位元進行編碼。 於是在給定歷史詞串的條件下， 當熵的值越低， 編碼的位元有限， 其所能代表的詞的量就越少， 因此可以接在歷史詞串後頭的詞也就越少， 代表可以更容易的預測接下來的詞， 便可在辨識過程中更精確的找到答案。

而熵和混淆度之間的關係只差在指數的部分， 當熵的值越小時混淆度也會跟著越小， 且由於指數的關係，混淆度對於數值的變化較為靈敏， 於是最後普遍使用混淆度作為評估測試語料的平均分歧度 (average branching factor) 。

一般來說當混淆度越小，辨識正確率相對也會較高。 雖然兩者之間並無一定比例之關係， 但皆為良好的語言模型評估參考。

2.4 實驗環境

接下來各章節的實驗，主要使用臺灣大學電機系同一位教授所開設的兩門個別獨立的課程， 其分別為數位語音訊號處理 (digital speech signal processing, DSP) 及信號與系統 (signal and system, SS) 。

本論文中所有的語言模型， 都對於這兩門課程分別跑了一次完整的語音辨識， 以測試其在不同的辨識場景中是否都能有穩定的進步。

2.4.1 辭典

本論文的辨識系統係採用雙語混合辭典， 辭典的建立主要參照 [22]。 辭典定義了語言模型計算時之基本單位， 包含了中文詞、英文詞、以及詞與發音的對照表，其中主要以中文為主， 而英文則是常見的語助詞及專有名詞較多。 辭典所包含的詞數及中英比例請見表 2.2。

課程	中文詞數	英文詞數
數位語音訊號處理	11,175 (84.07%)	2,117 (15.93%)
信號與系統	11,178 (83.70%)	2,177 (16.30%)

表 2.2: 課程系統中辭典的中英文詞數及比例

2.4.2 語料庫

語料庫可分為文字語料以及語音語料。 而文字語料依照來源主要分為底下幾類：

- **目標課程語料：**

  為辨識目標課程本身的內容，係課程語音經標註及斷句過後的人工轉寫(manual transcription)。 接著抽取其中一部分作為訓練、調適與實驗材料，分別為訓練集(training set)、發展集(development set)及測試集(testing set)，其兩兩之間並無重複之內容。 訓練集取每節課前 16 分鐘左右的內容，總計約 12 小時， 發展集和測試集則是在每堂課各隨機選取一段約 4 分鐘的連續語料，總計各約 3 小時。

  語言模型主要使用訓練集依照各種模型訓練而成， 發展集則僅做語言模型調適時，調整各語言模型的內差權重之用。 而測試集則是用來判斷語言模型的好壞，為實驗數據的主要來源。

- **相似課程語料：**

  由於課程語料的量往往遠小於廣播和新聞語料， 因此若能夠由藉由一些相似的課程來輔助欲辨識的課程， 相信定能為統計式語言模型帶來進步。 以大學課程為例， 共同必修課程常常由許多不同的教授所同時開設，其主題和內容是十分相似的， 如微積分、經濟學等等， 故利用相似課程語料可加強其專業內容的辨識效果。

  又或者一個教授可能會同時開設兩門以上不同的課程， 這些課程間的主題和內容雖然不一定相關， 但是仍可藉由這類的相似課程語料去捕捉語者的說話習慣。

  在本論文中主要採取的是第二種， 也就是使用同位教授所開設的不同

課程。 而這些課程之間則互相使用對方的訓練集作為自身調適之用， 如數位語音訊號處理和信號與系統的訓練集皆互為對方的相似課程語料。

- **背景語料：**

  在傳統的辨識系統上， 為了讓訓練出來的語言模型具有一般性 (generalization) ， 通常會將各種領域所能獲得的各種語料集結而成為背景語料， 並使用這些背景語料來訓練語言模型，所以稱呼其為背景模型 (background model) 。 但背景模型往往和測試語料 (testing corpus) 的領域相距甚遠 (out-of-domain) ， 所以常採用調適語料 (adaptation corpus) 來調適背景模型， 利用調適語料和測試語料間的具有同領域 (in-domain) 的特性以補背景模型之不足。

  本論文所使用的背景語料主要有 雅虎奇摩新聞網所整理的台灣各大新聞社的新聞 [26]， 以及近來在台灣十分流行的社群網站噗浪網 (Plurk) [27] 上網友各式各樣的訊息， 兩者皆為中英混合語料。

所有文字語料皆經辨識目標課程的辭典所斷詞完畢，主要理論係依據 [28] 中第四章的中文斷詞法。 而斷詞後的語料庫的句數、中英詞數請見表 2.3。

文字語料	句數	中文詞數	英文詞數
數位語音訊號處理 - 訓練集	9,174	69,799(87.48%)	9,986(12.52%)
數位語音訊號處理 - 發展集	2,279	17,004(86.37%)	2,684(13.63%)
數位語音訊號處理 - 測試集	2,279	16,756(86.76%)	2,556(13.24%)
信號與系統 - 訓練集	5,599	58,841(84.69%)	10,641 (15.31%)
信號與系統 - 發展集	1,383	15,469(84.45%)	2,849(15.55%)
信號與系統 - 測試集	1,383	15,761(86.83%)	2,390(13.17%)
雅虎奇摩新聞網	5,223,719	53,019,262 (99.13%)	465,865 (0.87%)
噗浪網	1,860,175	21,098,014 (95.74%)	939,500 (4.26%)

表 2.3: 各種文字語料的句數及中英文詞數分佈

2.4.3 語音辨識系統

本實驗主要使用隱藏式馬可夫模型程式集 (Hidden Markov Model Toolkit, HTK) 為語音辨識器 。 課程語音之原始錄音為雙聲道且取樣頻率 (sample rate) 為 44 赫茲 (Hz) ， 而為了抽取恰當之特徵參數，降其頻至 16 赫茲且僅取用單聲道。 聲學模型是使用課程語料之中訓練集所對應的音檔所訓練成的語者相關 (speker dependent, SD) 模型， 為跨詞三連音素聲學模型， 使用的音素集同時包含 32 個中文音素及 39 個英文音素及靜音和短暫停音， 為 24 高斯合成分佈 (Gaussian-mixtures) 模型。 辨識時聲學模型和語言模型的比例係數 (scale factor) 分別為 0.5 和 8.0。

而基礎實驗中所使用的語言模型 是採用史丹福研究所語音技術與研究實驗室所開發的語言模型程式集 (SRILM) [29] 訓練出三連語言模型， 並使用聶氏折扣法

(Kneser-Ney discounting) 及古德—圖靈式折扣法 (Good-Turing discounting) 進行平滑化。

最後，辨識正確率的算法為：

$$Accuracy = \frac{H - I}{T}, \tag{2.14}$$

其中 H 為辨識正確的次數 (hit) ， I 為插入性錯誤 (insertion error) 的次數，T 則為總次數 (total)。 要注意的是，中文計算次數的單位是單字 (charactor) ，而英文則為詞 (word) ， 例如：「model」、「HMM」和「OKAY」。 實驗時所測試的辨識正確率分別有：

$$Mandarin\ Accuracy = \frac{H_{Mandarin} - I_{Mandarin}}{T_{Mandarin}}, \tag{2.15}$$

$$English\ Accuracy = \frac{H_{English} - I_{English}}{T_{English}}, \tag{2.16}$$

$$Overall\ Accuracy = \frac{H_{Mandarin} + H_{English} - I_{Mandarin} - I_{English}}{T_{Mandarin} + T_{English}}, \tag{2.17}$$

2.5 本章總結

本章簡介雙語混合課程語料的性質， 以及介紹語言模型在大字彙連續語音辨識中所扮演的角色及目的， 接著介紹統計式語言模型和基礎實驗所使用的聶氏三連語言模型， 最後列出實驗使用的語料庫和辨識環境。 本章亦列出語言模型評估參考：混淆度以及中英辨識正確率的計算方法。

第 三 章　基於詞群之雙語語言模型

本章介紹基於詞群之 N 連語言模型以及詞群之分群演算法。 首先在 3.1 節簡介基於詞群之 N 連語言模型的數學架構， 接著在 3.2 節詳述本論文所使用的詞群分群演算法。 當建立基於詞群之 N 連語言模型後， 在 3.3 節中使用線性內差法强化基於詞群之 N 連語言模型， 並在 3.4 節以實驗分析各種分群演算法。 最後 3.5 節是本章總結。

3.1　基於詞群之 N 連語言模型

傳統上 N 連語言模型僅考慮訓練語料中詞的前後關係。 舉例來說，「狗」和「貓」都是動物的名稱， 而「run」和「walk」都是這些動物可能會出現的動作，但若雙連詞「狗 run」和「貓 walk」在訓練語料中頻繁出現， 而「狗 walk」及「貓 run」出現的次數卻很少， 則其語言模型的分數就會很低， 並連帶影響到辨識結果。 但在實際說話時，「狗」和「貓」後面接著「run」和「walk」理應具有相近的機率， 可是統計式 N 連語言模型並無法達到此目的。

在缺少訓練語料的情況下， 上述的問題會更加嚴重。 以課程系統為例， 課程語料往往僅能透過騰寫逐字稿的方式獲得， 需要耗費許多時間與人力， 故其數量往往遠小於一般語料。 再加上中英混合的特性， 使得英語詞彙更容易因為量少而出現極多的未見事件， 導致機率估測的困難。

這時，基於詞群之 N 連語言模型 (class-based Ngram LM) [8] 便是一種有效率的去增强語言模型强健性的方法。 其將語言中具有相近語意、詞性與文法結構的詞彙合為一群， 使機率計算的單位由詞 (word) 提昇至詞群 (word class) ， 並讓詞群內的詞之間共享參數， 以減少 N 連機率的參數量， 並解決稀疏性的問題。

首先，依據不同的分群演算法得到各種分類函數 $\pi(\cdot)$ ， 將辭典 $\mathcal{V}$ 中所有的詞代入分群函數中：

$$c_i = \pi(w_i), w_i \in \mathcal{V}. \tag{3.1}$$

即可得到 w_i 所對應的詞群 c_i 。

接著，用詞群的概念重新定義 N 連語言模型， 以雙連機率為例，其式為：

$$P_{class}(w_i|w_{i-1}) = P(w_i|c_i) \times P(c_i|c_{i-1}), \tag{3.2}$$

其中 $P(w_i|c_i)$ 表示訓練集中詞 w_i 在其所屬詞群 c_i 中所佔次數的比例， 而 $P(c_i|c_{i-1})$ 則是以詞群為單位之雙連機率， 計算方式分別如下：

$$P(w_i|c_i) = \frac{C(w_i)}{\sum\limits_{w_j \in c_i} C(w_j)}, \tag{3.3}$$

$$P(c_i|c_{i-1}) = \frac{C(c_{i-1}, c_i)}{\sum\limits_{c_j} C(c_{i-1}, c_j)}. \tag{3.4}$$

於是，先根據分群演算法得到分類函數 $\pi(\cdot)$ ， 將訓練集中所有的詞都替換為詞群後， 接著使用式 (3.3) 算出每個詞在其所屬詞群中所站次數的比例， 並使用式 (3.4) 估算詞群間的 N 連機率， 最後套入式 (3.2) ， 得到基於詞群之 N 連語言模型。

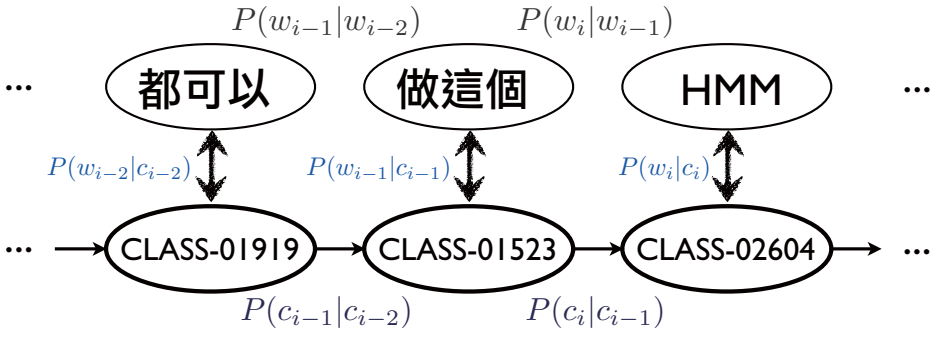

圖 3.1: 詞和詞群之間的轉換對應關係

如此一來，就算有些 N 連詞並未出現在訓練集中，還是可以透過其對應詞群的分佈來重新估算其機率。例如本節一開始的例子，只要出現「狗 run」，那「狗 walk」、「貓 run」乃至「貓 walk」的機率都會一起被調整，同詞群內部的參數是可以共享的，而詞群中若有某些詞特別具代表性，亦可反應在式 (3.3)。故接下來的重點就在於如何得到一個好的分群演算法。

3.2 詞群分群演算法

詞群的分群演算法主要有兩種做法，一是使用資料驅動 (data-driven) 的自動化分群演算法，二是透過語言學的知識 (linguistic knowledge driven) 做基於規則 (rule-based) 的對應。

3.2.1 以平均相互資訊最大化為基準

首先使用資料驅動 (data-driven) 的自動化分群演算法。在統計上主要是以最大化文字訓練語料的對數相似度 (log likelihood) 為準則 (criterion)，其定義為：

$$\begin{aligned}
L(\pi) &= \frac{1}{T-1}\log P(w_2^T|w_1) \\
&= \sum_{w_{i-1},w_i} \frac{C(w_{i-1},w_i)}{T-1}\log(P(c_i|c_{i-1})\times P(w_i|c_i)) \\
&= \sum_{c_{i-1},c_i} \frac{C(c_{i-1},c_i)}{T-1}\log\frac{P(c_i|c_{i-1})}{P(c_i)} + \sum_{w_i}\frac{\sum_{w_j\in\mathcal{V}} C(w_j,w_i)}{T-1}\log P(w_i|c_i)P(c_i) \\
&= \sum_{c_{i-1},c_i} P(c_{i-1},c_i)\log\frac{P(c_i|c_{i-1})}{P(c_i)} + \sum_{w_i} P(w_i)\log P(w_i) \\
&= I(c_{i-1},c_i) - H(w_i).
\end{aligned} \tag{3.5}$$

其中 T 是訓練集中的總詞數、$I(c_{i-1},c_i)$ 是雙連詞群之間的平均相互資訊、而

$H(w_i)$ 則是單連詞 w_i 的熵， 其值並不受分群演算法 π 的影響。 因此最大化相似度亦即最大化平均相互資訊 (average mutual information) 。

但是平均相互資訊的計算相當複雜， 因此很難有效率的去找出最佳解， 故一般採用採用貪婪演算法 (greedy algorithm) 去找出近似解， 其中漸進式貪婪合併法 (incremental greedy merging) 便是十分常見的分群演算法 [8]。

這邊的實驗主要使用 SRILM 程式集 [29] 所提供的程式：ngram-class 以進行漸進式貪婪合併法。 其主要步驟如下：

– 步驟 1： 設定想要得到的詞群數目 N_c 。

– 步驟 2： 以出現次數排序訓練語料並取前 N_c 個高頻詞，使其各自分成一群。

– 步驟 3： 將尚未分群的詞中最高頻的詞獨立定義為新的詞群。

– 步驟 4： 任意合併兩詞群，並記錄合併前後平均相互資訊的差值。

– 步驟 5： 找出合併前後平均相互資訊下降量最小的組合，並實際合併之。

– 步驟 6： 重複步驟 3 至 5 ，直到所有的詞都已分群完畢。

於是，便可得到中英混合詞群， 並建立以平均相互資訊最大化為基準的基於詞群之 N 連語言模型，底下簡稱為 MI。

3.2.2 以詞性標記為基準

這邊主要使用詞性標記 (part-of-speech tag) 為分群主要依據 [30]。 首先使用中研院中文斷詞系統 [31] 將未斷詞過的訓練集進行詞性標記後， 將訓練集中的詞依據其所標記的詞性進行分群， 亦即具有相同詞性的詞將會被分配至同一詞群中， 以便將語言中具有相近語意、詞性與文法結構的詞彙合為一群。

然而一個詞在不同位置可能被標上不同的詞性， 所以在分群時， 皆以該詞被標記的詞性裡出現最多次的詞性為主， 使每一個詞最後僅會被分至一個詞群中。

此外，由於中研院中文斷詞系統僅能為中文進行詞性標記， 故英文的部份皆會被標記為外來語 (foreign words, FW)， 若直接使用詞性標記的結果， 將會使所有英文詞都被分配至同一詞群中。 其顯而易見的壞處是不同英文詞之間的語意及文法並不見得相同， 如「OKAY」及「HMM」等。 但隱藏的好處是， 由於課程語料中英文的部份大多是專有名詞及語助詞等， 藉由將英文詞統一分配到同一詞群， 可觀察哪些中文詞的前後特別容易鄰接英文詞， 譬如「叫做」後面接中文詞和英文詞的比例分別為 23.30% 及 76.70% ， 這在以中文為主的中英混合語料中是十分特殊的。 所以針對英文的部份， 可讓所有的英文詞都分配到詞群「FW」中， 或讓每個英文詞不分群， 直接以詞為單位， 抑或額外查詢英文字典以標記詞性並和中文同類詞性合併， 皆為可行的方法。 然而在之前的實驗中 [32] 發現由於課程語料太過稀少且其中英文詞彙的詞性過於集中， 使得低頻的英文詞性並無法藉由詞群獲得進步， 故底下並未使用英文的詞性。

最後，雖然在低頻詞的部份基於詞群之模型有較好的表現， 但在高頻詞的部份 N 連模型則較為精準， 所以一個常見的做法是僅將低頻詞加入分群演算法中， 而高頻詞則不分群。

綜合以上的觀察，主要採取下列四種分群演算法：

– 演算法 1： 中文的部份依據其所標記的詞性分群，而英文全分群為「FW」。

– 演算法 2： 僅將中文的部份依據其所標記的詞性分群，而英文詞則不分群。

– 演算法 3： 和 (1) 相同，但是僅讓出現次數低於平均值的低頻詞參與分群。

– 演算法 4： 和 (2) 相同，但是僅讓出現次數低於平均值的低頻詞參與分群。

分別使用這四種分群演算法， 建立以詞性標記為基準的基於詞群之 N 連語言模型， 底下稱之為 POS-1 、 POS-2 、 POS-3 以及 POS-4， 如 POS-4 即表示僅將中文且次數低於平均的詞彙依據其所標記的詞性分群， 而英文詞則不分群。 或以 POS 通稱以詞性標記為基準的基於詞群之 N 連語言模型。

3.2.3 以複合詞性標記為基準

然而，回顧標記詞性時，其使用的乃是未經斷詞過的文字訓練語料， 亦即最後斷詞的位置是依據中研院中文斷詞系統所判定， 這和使用課程辭典為依據的斷詞法 [28] 在中文的部份有著極大的差異。 中研院中文斷詞系統所斷出來的詞由於要在上面標註詞性的關係，所以長度顯得較為細緻， 但是課程辭典中的中文詞則強調「長詞優先」，所以長度較長。 於是導致在為辭典中的詞依據詞類標記結果而決定分詞函數 $\pi(\cdot)$ 時， 很容易會因為斷詞的不一致而導致某些詞並未獲得詞性標記， 最後便不能參與分群。

為了解決斷詞不一致的問題， 標註詞性時不再使用文字訓練語料， 而是改為標記辭典。 此時某些長度較長的詞可能被斷成好幾個部分， 即使其在辭典中確實為一個詞， 如「語音處理」便被標記為「語音(N) 處理(VT)」。 於是使用程式強制其再度合併為一個詞， 即「語音處理(N+VT)」， 其中「N+VT」即為「語音處理」的複合詞性 (Compound POS) 。 表 3.1 列出部分複合詞性及其所包含的詞。

最後利用這些複合詞性，讓相同複合詞性的詞分配到同一詞群， 並依樣採用上述 (1) 至 (4) 的分群演算法， 建立以複合詞性標記為基準的基於詞群之 N 連語言模型， 故底下稱之為 POS-C-1 、 POS-C-2 、 POS-C-3 以及 POS-C-4， 或以 POS-C 通稱以複合詞性標記為基準的基於詞群之 N 連語言模型。

複合詞群	複合詞群總次數	包含詞	該詞出現的次數
ADV+ADV+VI	34	不太好	4
		不太容易	14
		不太一樣	16
ADV+VI+T	151	很好的	4
		最小的	4
		最理想的	5
		最重要的	6
		很清楚的	7
		最常用的	7
		很重要的	9
		很大的	11
		最簡單的	13
		不一樣的	18
		最大的	43
ADV+VT+T	78	要說的	5
		要找的	9
		要講的	14
		所講的	16
		所說的	34
P+DET+M	98	到這個	13
		用一個	24
		在這個	61
T+DET	14	之一	3
		的部份	5
		的部分	6

表 3.1: 部分複合詞性及其在訓練集中所包含詞以及出現次數的資訊

3.3 使用線性內差法强化語言模型

由於 N 連語言模型和基於詞群之 N 連語言模型 在高頻詞及低頻詞上的統計各有優缺點， 故一個常見的做法是將兩者的機率做等權重的線性內差 (interpolation)。

於是，以平均相互資訊最大化為基準 (MI) 、 以詞性標記為基準 (POS) 、 和以複合詞性標記為基準 (POS-C) 的這三種基於詞群之模型模型， 都分別和經過聶氏平滑法的 N 連語言模型 (KN) 做等權重的線性內差：

$$\bar{P}_{MI+KN}(w_i|h_i) = \frac{1}{2}\, P_{MI}(w_i|h_i) + \frac{1}{2}\, P_{KN}(w_i|h_i), \tag{3.6}$$

$$\bar{P}_{POS+KN}(w_i|h_i) = \frac{1}{2}\, P_{POS}(w_i|h_i) + \frac{1}{2}\, P_{KN}(w_i|h_i), \tag{3.7}$$

$$\bar{P}_{(POS-C)+KN}(w_i|h_i) = \frac{1}{2}\, P_{POS-C}(w_i|h_i) + \frac{1}{2}\, P_{KN}(w_i|h_i). \tag{3.8}$$

最後為了同時使用資料驅動及語言學的知識， 底下分別將兩個使用語言學並經强化過後的模型， 與使用資料驅動並經强化過後的模型再做一次等權重的線性內差：

$$\tilde{P}_{[POS+KN]+[MI+KN]}(w_i|h_i) = \frac{1}{2}\, \bar{P}_{POS+KN}(w_i|h_i) + \frac{1}{2}\, \bar{P}_{MI+KN}(w_i|h_i), \tag{3.9}$$

$$\tilde{P}_{[(POS-C)+KN]+[MI+KN]}(w_i|h_i) = \frac{1}{2}\, \bar{P}_{(POS-C)+KN}(w_i|h_i) + \frac{1}{2}\, \bar{P}_{MI+KN}(w_i|h_i). \tag{3.10}$$

以上流程如圖 3.2 所示：

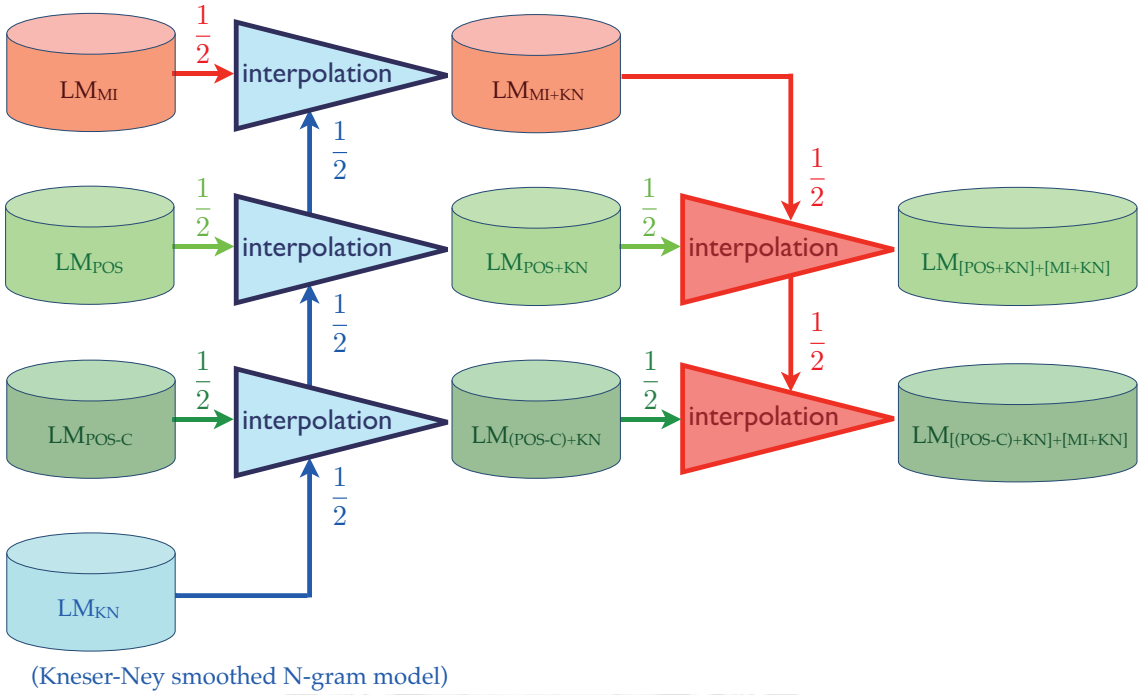

圖 3.2: 將各種根據不同分群演算法的基於詞群之模型和 N 連模型之間做等權重之線性內差以强化模型。

3.4 實驗結果與比較

底下所有實驗皆同時實作在「數位語音訊號處理」及「信號與系統」這兩門教學系統上。基礎實驗的部份為經過聶氏平滑法的三連語言模型，簡稱為 Trigram。

3.4.1 基於詞群之模型 - 以平均相互資訊最大化為基準

首先討論以平均相互資訊最大化為基準的模型，如 3.2.1 節所述。詞群數目 N_c 預設為 200，亦即將所有出現在訓練集中的詞分為 200 個詞群。

由表 3.2 及表 3.3 可看出，和基礎實驗相比之下，基於詞群之模型較基礎實

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	備註
Trigram	250.981	83.81	62.45	82.13	基礎實驗
MI	347.162	82.26	60.41	80.54	$N_c = 200$
MI+KN	**241.489**	**84.11**	**63.83**	**82.52**	與 N 連模型線性內差

表 3.2: 「數位語音訊號處理」基於詞群之模型實驗結果 1：使用平均相互資訊為基準

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	備註
Trigram	211.085	77.35	72.21	76.95	基礎實驗
MI	286.67	75.92	69.62	75.43	$N_c = 200$
MI+KN	**204.713**	**77.55**	**71.88**	**77.11**	與 N 連模型線性內差

表 3.3: 「信號與系統」基於詞群之模型實驗結果 1：使用平均相互資訊為基準

驗差， 其原因可能是因為單一語者的語料可能有其習慣的用法， 因此訓練集中出現次數較高的部份亦在測試集中密集出現， 此時强化低頻詞並無法帶來太大的好處， 導致使用詞群的效果不佳。

然而，經由和 N 連模型做線性內差强化過後， 所得到的模型便獲得許多進步， 推測是由於 N 連模型在高頻詞及基於詞群之模型在低頻詞上的估算皆有其獨到之處， 使得最後得到全面性的良好的估算。 在「數位語音訊號處理」以及「信號與系統」中， 其絕對的辨識正確率分別進步了 0.39% 與 0.16%。

3.4.2 基於詞群之模型 - 以詞性標記為基準

接著討論以詞性標記為基準的模型。 由於使用詞性標記來分群時， 如 3.2.2 節所述， 可根據英文詞以及高頻詞是否參與分群而分為 POS-1 到 POS-4， 故底下四種方法並陳， 並使用基礎實驗 (Trigram) 以及前面實驗中的 MI+KN 來做比較。

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	備註
Trigram	250.981	83.81	62.45	82.13	基礎實驗
MI+KN	241.489	84.11	63.83	82.52	强化之資料驅動模型
POS-1	498.202	80.92	58.46	79.16	英文皆分群為 FW
(POS-1)+KN	260.543	83.78	63.17	82.16	與 N 連模型線性內差
[(POS-1)+KN]+[MI+KN]	242.344	84.09	63.95	82.50	資料驅動＋語言學知識
POS-2	458.627	81.03	62.63	79.58	英文獨立不參與分群
(POS-2)+KN	262.981	83.60	63.17	82.00	與 N 連模型線性內差
[(POS-2)+KN]+[MI+KN]	243.076	84.08	63.72	82.47	資料驅動＋語言學知識
POS-3	315.691	82.62	59.86	80.83	低頻詞＋英文皆分群為 FW
(POS-3)+KN	243.707	83.97	63.17	82.34	與 N 連模型線性內差
[(POS-3)+KN]+[MI+KN]	**235.656**	**84.22**	**64.15**	**82.65**	資料驅動＋語言學知識
POS-4	301.688	82.73	62.12	81.11	低頻詞＋英文獨立不參與分群
(POS-4)+KN	246.361	83.87	62.55	82.19	與 N 連模型線性內差
[(POS-4)+KN]+[MI+KN]	236.54	84.17	63.60	82.55	資料驅動＋語言學知識

表 3.4: 「數位語音訊號處理」基於詞群之模型實驗結果 2：使用詞性標記為基準

由表 3.4 所示， 在POS-1 與 POS-3 之間以及 POS-2 與 POS-4 之間比較的結果顯示， 讓高頻詞不參與分群而僅用低頻詞會讓結果變得比較好， 這可能是因為參與分群的詞越少時， 其模型就越接近 N 連模型。

而基於詞群之模型和 N 連模型做線性內差以强化模型後， 再藉由同時使用

統計學和語言學的分群演算法的好處， 進一步將其與强化後之資料驅動模型做合併， 皆可使得基於詞群之模型獲得穩定的進步。

此外，在POS-1 與 POS-2 之間以及 POS-3 與 POS-4 之間比較的結果顯示， 讓英文不參與分群，亦即讓英文偏近 N 連模型似乎是比較好的作法， 但是，在與 N 連模型線性內差後， 最後反而是讓英文皆分群為「FW」的結果較佳， 其原因可能是和 N 連模型的互補性所導致， 但最後在與强化後之資料驅動模型做合併後， 則差別不大。

最後的結果是， 使用低頻詞做分群、英文皆分群為「FW」、 以及結合 N 連模型和資料驅動模型後的結果表現最佳， 其絶對的辨識正確率和基礎實驗以及强化之資料驅動模型相比， 分別進步了 0.52% 及 0.13%。

表 3.5 的結果和表 3.4 大致雷同， 在混淆度上最好的結果和表 3.4 相同， 為使用低頻詞做分群、英文皆分群為「FW」、 以及結合 N 連模型和資料驅動模型後的結果表現最佳， 其總正確率較基礎實驗為佳。 然而可能受到辨識時其他環境的影響， 總正確率最好的模型傾向於讓高頻詞也參與分群， 其絶對的辨識正確率和基礎實驗以及强化之資料驅動模型相比， 分別進步了 0.18% 及 0.02%。

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	備註
Trigram	211.085	77.35	72.21	76.95	基礎實驗
MI+KN	204.713	77.55	71.88	77.11	强化之資料驅動模型
POS-1	448.107	74.45	66.54	73.84	英文皆分群為 FW
(POS-1)+KN	219.972	77.30	71.33	76.84	與 *N* 連模型線性內差
[(POS-1)+KN]+[MI+KN]	204.611	**77.57**	71.96	**77.13**	資料驅動＋語言學知識
POS-2	403.643	74.42	70.71	74.13	英文獨立不參與分群
(POS-2)+KN	220.861	77.33	**72.25**	76.94	與 *N* 連模型線性內差
[(POS-2)+KN]+[MI+KN]	205.267	77.54	71.83	77.10	資料驅動＋語言學知識
POS-3	265.678	76.01	69.46	75.50	低頻詞＋英文皆分群為 FW
(POS-3)+KN	207.675	77.24	71.75	76.81	與 *N* 連模型線性內差
[(POS-3)+KN]+[MI+KN]	**200.889**	77.46	71.58	77.00	資料驅動＋語言學知識
POS-4	257.689	76.10	71.42	75.73	低頻詞＋英文獨立不參與分群
(POS-4)+KN	208.579	77.26	71.83	76.84	與 *N* 連模型線性內差
[(POS-4)+KN]+[MI+KN]	201.219	77.41	71.58	76.95	資料驅動＋語言學知識

表 3.5:「信號與系統」基於詞群之模型實驗結果 2：使用詞性標記為基準

3.4.3 基於詞群之模型 - 以複合詞性標記為基準

最後討論以複合詞性標記為基準的模型。 和以詞性標記為基準的模型相同， 底下亦根據英文詞以及高頻詞是否參與分群而分為 POS-C-1 到 POS-C-4， 以及使用基礎實驗 (Trigram) 以及前面實驗中的 MI+KN 來做比較。

由表 3.6 比對表 3.4 以及表 3.7 比對表 3.5 ， 可發現使用複合詞性標記的模型和使用詞性標記的趨勢大致相同， 然而兩者最好的模型在「數位語音訊號處理」及「信號與系統」各擅勝場， 無法區分何者較佳。

其原因可能是因為詞性標記和複合詞性乃是在長詞被斷的支離破碎時才會有

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	備註
Trigram	250.981	83.81	62.45	82.13	基礎實驗
MI+KN	241.489	84.11	63.83	82.52	强化之資料驅動模型
POS-C-1	503.374	80.67	57.99	78.88	英文皆分群為 FW
(POS-C-1)+KN	258.131	83.83	63.64	82.24	與 N 連模型線性內差
[(POS-C-1)+KN]+[MI+KN]	241.93	84.07	63.83	82.48	資料驅動＋語言學知識
POS-C-2	440.445	80.94	62.94	79.53	英文獨立不參與分群
(POS-C-2)+KN	257.182	83.81	63.13	82.18	與 N 連模型線性內差
[(POS-C-2)+KN]+[MI+KN]	241.096	84.08	63.48	82.46	資料驅動＋語言學知識
POS-C-3	315.009	82.48	60.76	80.77	低頻詞＋英文皆分群為 FW
(POS-C-3)+KN	243.963	83.74	63.33	82.14	與 N 連模型線性內差
[(POS-C-3)+KN]+[MI+KN]	**235.961**	**84.23**	**63.87**	**82.63**	資料驅動＋語言學知識
POS-C-4	302.338	82.64	62.39	81.05	低頻詞＋英文獨立不參與分群
(POS-C-4)+KN	245.882	83.76	62.59	82.09	與 N 連模型線性內差
[(POS-C-4)+KN]+[MI+KN]	236.508	84.18	63.48	82.55	資料驅動＋語言學知識

表 3.6: 「數位語音訊號處理」基於詞群之模型實驗結果 3：使用複合詞性標記為基準

所區分， 且複合詞性標記的詞群數目會隨著斷開的片段數而呈現指數成長， 導致每個詞群內所包含的詞數下降， 無法發揮詞群內資訊共享的長處。

綜合以上所有實驗， 在「數位語音訊號處理」中最佳的方法為表 3.4 中的[(POS-3)+KN]+[MI+KN]， 即高頻詞不參與分群、使用 N 連模型强化、以及結合資料驅動的詞性標記模型。 和基礎實驗相比， 其在中英及總正確率分別進步了 0.41% 、 1.7% 及 0.52%；

而在「信號與系統」中最佳的方法為表 3.7 中的[(POS-C-1)+KN]+[MI+KN]，即高頻詞亦參與分群、使用 N 連模型强化、以及結合資料驅動的複合詞性標記模

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	備註
Trigram	211.085	77.35	72.21	76.95	基礎實驗
MI+KN	204.713	77.55	71.88	77.11	强化之資料驅動模型
(POS-C-1)	449.908	74.08	66.79	73.52	英文皆分群為 FW
(POS-C-1)+KN	216.779	77.33	71.79	76.90	與 N 連模型線性內差
[(POS-C-1)+KN]+[MI+KN]	203.915	77.60	72.00	77.16	資料驅動＋語言學知識
(POS-C-2)	391.792	74.10	70.08	73.78	英文獨立不參與分群
(POS-C-2)+KN	216.017	77.33	71.83	76.91	與 N 連模型線性內差
[(POS-C-2)+KN]+[MI+KN]	203.74	77.58	72.00	77.14	資料驅動＋語言學知識
(POS-C-3)	263.112	75.99	70.00	75.52	低頻詞＋英文皆分群為 FW
(POS-C-3)+KN	206.247	77.21	71.50	76.77	與 N 連模型線性內差
[(POS-C-3)+KN]+[MI+KN]	200.158	77.39	71.46	76.93	資料驅動＋語言學知識
(POS-C-4)	257.369	76.04	71.12	75.66	低頻詞＋英文獨立不參與分群
(POS-C-4)+KN	206.992	77.17	71.75	76.75	與 N 連模型線性內差
[(POS-C-4)+KN]+[MI+KN]	200.431	77.36	71.71	76.92	資料驅動＋語言學知識

表 3.7: 「信號與系統」基於詞群之模型實驗結果 3：使用複合詞性標記為基準

型。 在中文和總正確率分別進步了 0.25% 及 0.21%，而英文則退步了 0.21%。

3.5 本章總結

本章介紹了基於詞群之語言模型，並希望改善一般 N 連語言模型稀疏性的問題。主要使用的分群演算法為：以平均相互資訊為基準、 以詞類標記為基準、 和以複合詞類標記為基準， 接著使用等權重之線性內差法分別强化之。 最後總辨識正確率上在兩門不同的課程系統中皆獲得進步。

第 四 章　隨機森林語言模型

本章介紹決策樹語言模型以及隨機森林語言模型。 在 4.1 節介紹決策樹語言模型的生長與修剪演算法， 接著在 4.2 節介紹隨機森林語言模型， 並在 4.3 節以實驗分析隨機森林語言模型並比較其結果。 最後 4.4 節是本章結論。

目前隨機森林語言模型不管是在單語語言模型、 或是在語言模型調適上皆得到穩定的進步 [13] [15]， 故接下來本章將介紹隨機森林語言模型以及其應用在雙語環境下的成果。

4.1　決策樹語言模型

在 2.3 節中曾經提過， 統計式語言模型所要估測的， 是給定歷史詞串 $h_i = w_1, w_2, ..., w_{i-1}$ 時， 預測接下來出現的詞是 w_i 的機率， 亦即 $P(w_i|w_1^{i-1})$ 。 然而要估計的參數量高達 $|\mathcal{V}|^i$ ， 故一般使用等價類別分類函數將歷史詞串對應到其等價類別之中以減少參數量：

$$P(W) = \prod_{i=1}^{m} P(w_i|w_1, w_2, ..., w_{i-1}) \approx \prod_{i=1}^{m} P(w_i|\Phi(w_1, w_2, ..., w_{i-1})) \qquad (4.1)$$

其中 $\Phi : \mathcal{V}^* \mapsto \mathcal{C}$ 即為等價類別分類函數。 根據馬可夫假設， 若等價類別分類函數僅取歷史詞串中接近 w_i 的前 $N-1$ 個詞做統計， 則此即為 N 連模型：

$$P(w_i|h_i) \approx P(w_i|\Phi_{Ngram}(h_i)) = P(w_i|w_{i-N+1}^{i-1}). \qquad (4.2)$$

而若等價類別分類函數是經由決策樹提問一連串的問題來分類歷史詞串， 使得每一個在相同節點中的歷史詞串共享機率分佈， 則此模型即為決策樹模型：

$$P(w_i|h_i) \approx P(w_i|\Phi_{DT}(\Phi_{Ngram}(h_i))) = P(w_i|\Phi_{DT}(w_{i-N+1}^{i-1})). \qquad (4.3)$$

其中 $\Phi_{DT}()$ 為決策樹之等價分類函數。

歷史詞串 (w_{i-2}, w_{i-1})	目標詞 w_i	三連詞 (w_{i-2}, w_{i-1}, w_i) 出現次數
這個 學期	之內	1
	的內容	1
這個 HMM	的	2
	就會	1
那些 HMM	的	4
	是	3
一堆 Gaussian	Mixture	3
	相加	1

表 4.1: 某虛擬訓練集，及其所包含的歷史詞串和w_{i-2}^{i} 的出現次數

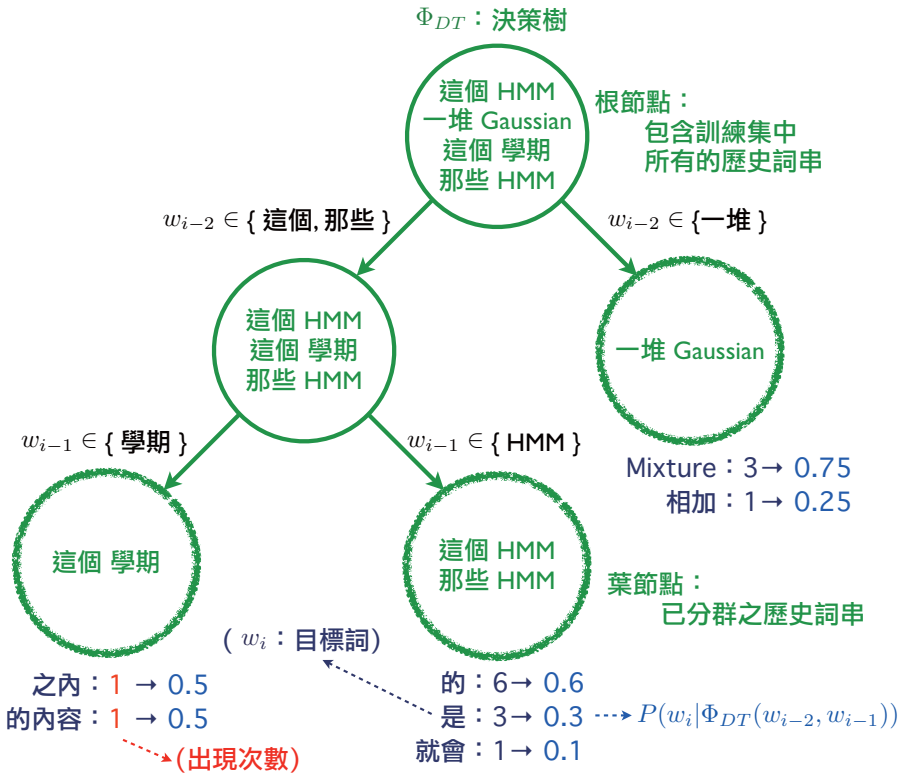

圖 4.1: 決策樹 Φ_{DT}，將表 4.1 中的歷史詞串分群以共享機率分佈

舉一個虛擬訓練集為例， 如表 4.1 所示， 其列舉了此訓練集內所有三連詞及其出現次數。 假定今欲估算三連機率 P (是 | 這個 HMM)， 由於在訓練集表 4.1 中期出現次數是零， 故若直接使用一般的 N 連語言模型將會導致其機率為零。

這時若存在一棵決策樹 Φ_{DT} ， 如圖 4.1 所示， 此決策樹將某些歷史詞串分群後， 使存在於同一個類別的歷史詞串放在同一個葉節點內， 例如「這個 HMM」和「那些 HMM」存在於同一個葉節點， 故如果使用 Φ_{DT} (這個 HMM) ， 將會得到包含「這個 HMM」和「那些 HMM」的葉節點， 令其為 $leaf_f$ ， 而葉節點內的歷史詞串彼此之間的次數是共享的， 故其機率依最簡單的算法為：

P (是 | 這個 HMM) $\approx P$(是 | Φ_{DT} (這個 HMM))

= P (是 | $leaf_f$) = $\frac{3}{2+1+4+3}$ = 0.3

如此一來，便可避免將此機率估計為零， 達到平滑化與減少預估計的參數量的效果。 底下將詳細介紹決策樹 Φ_{DT} 的生長與修剪演算法。

4.1.1 決策樹生長(growing)演算法

決策樹的生長是一連串將樹的節點 (node) 分裂為左右子節點的過程。 每個節點中包含了一些歷史詞串， 隨著節點的分裂， 其所含的歷史詞串也會隨之分配到其子節點當中。

由於決策樹的生長與修剪演算法皆需使用到文字訓練語料， 故底下皆隨機挑選訓練集中約 75% 的語料作為決策樹生長之用； 而剩下的 25% 則用來修剪決策樹，並稱做留存資料 (held-out data)。

一開始決策樹僅含有一個根節點 (root node)， 其包含了訓練集以內留存資料以外的所有歷史詞串， 此時根節點也是樹中唯一的葉節點 (leaf node)。 接著，在每一回合開始， 依序分裂一個葉節點， 並產生兩個新的葉節點， 直到生長演算

法結束為止。 底下使用快速交換演算法為主要生長演算法。

假定現在想要分裂一個葉節點 n， 並標示 $h(n)$ 為節點 n 中所包含的所有歷史詞串。 要注意的是，在此演算法中，所有歷史詞串的長度都是固定的， 也就是仿效 N 連模型僅取前 $N-1$ 長度的詞作為歷史詞串。 接著，定義位置(position, p)為歷史詞串中的詞與目標詞 w_i 的距離， 如表 4.2。

歷史詞串中的詞	w_{i-N+1}	w_{i-N+2}	...	w_{i-1}
位置 p	$N-1$	$N-2$	...	1

表 4.2: 歷史詞串中的詞與欲預測的詞 w_i 之間的距離

分裂葉節點時，主要考慮的是， 到底歷史詞串應該要分配到左子節點中比較好，還是分到右子節點中比較好。 但若一個一個的去判別 $h(n)$ 裡的歷史詞串該如何分配， 會耗費相當多的時間， 故在此處定義一個新的基本單位 $\beta_p(v)$， 其包含 $h(n)$ 中所有位置 p 的詞剛好是詞 v 的歷史詞串。

同樣以表 4.1 和圖 4.1 為例， 此決策樹在根節點分裂時， 其選擇的 p 值為 2， 而歷史詞串中p 值為 2 的詞只有「這個」、「一堆」及「那些」， 所以這時可將所有的歷史詞串分為：

- β_2 (這個) = {「這個 HMM」，「這個 學期」}
- β_2 (一堆) = {「一堆 Gaussian」}
- β_2 (那些) = {「那些 HMM」}

由此可看出， 當 p 值固定時 ，

$$h(n) = \cup_v \beta_p(v), \quad v \in \mathcal{V} \tag{4.4}$$

恆成立。 而作為基本單位，$\beta_p(v)$ 是無法被分割的， 故若選定 p 值為 2 時， {「這個 HMM」，「這個 學期」} 將會一起被放到左節點或右節點之中， 不會出現分開放的情形。 而當根節點分裂完後， 接著選定 p 值為 1， 此時「這個 HMM」和「這個 學期」不再屬於同一個基本單位， 故兩者可獨立放在左右子節點之中， 而換成「這個 HMM」和「這些 HMM」都屬於β_1 (HMM) ， 所以換 {「這個 HMM」，「這些 HMM」} 要被一起考慮。

由於不同的 p 會得到一堆不同的 $\beta_p(v)$， 故接下來的作法是先選定一個固定的 p ， 得到對應的基本單位 $\beta_p(v)$ 後， 再一一分配至左節點或右節點當中。

令左節點所包含的歷史詞串為 $\mathcal{L}_p$、 右節點為 $\mathcal{R}_p$， 則

$$\mathcal{L}_p \cup \mathcal{R}_p = h(n) \tag{4.5}$$

且

$$\mathcal{L}_p \cap \mathcal{R}_p = \phi \tag{4.6}$$

一開始，先以 $\beta_p(v)$ 為基本單位將 $h(p)$ 隨意分配至 $\mathcal{L}_p$ 及 $\mathcal{R}_p$， 使其成為兩非空集合且互相獨立的子集。 接著計算和節點 n 相關的文字訓練語料， 並依不同分裂狀況下所得到的對數相似度 $L(\mathcal{L}_p)$ 為分裂的準則。

當機率的估算是採用最大相似度估測法且未經平滑化，如式 (2.4) 時， 則其對數相似度為：

$$\begin{aligned} L(\mathcal{L}_p) &= \Sigma_w \left[C(\mathcal{L}_p, w) \log \frac{C(\mathcal{L}_p, w)}{C(\mathcal{L}_p)} + C(\mathcal{R}_p, w) \log \frac{C(\mathcal{R}_p, w)}{C(\mathcal{R}_p)} \right] \\ &= \Sigma_w \left[C(\mathcal{L}_p, w) \log C(\mathcal{L}_p, w) + C(\mathcal{R}_p, w) \log C(\mathcal{R}_p, w) \right] \\ &\quad - C(\mathcal{L}_p) \log C(\mathcal{L}_p) - C(\mathcal{R}_p) \log C(\mathcal{R}_p), \end{aligned} \tag{4.7}$$

其中 $C(\cdot, w)$ 表示詞 w 出現在所有屬於 $\cdot$ 之歷史詞串的後面之次數總和， 而 $C(\cdot)$ 則為 $\cdot$ 中所有歷史詞串的次數總和。 由於所有的計算皆可透過次數的計算來達成， 故最佳化次數的存取即可加速運算。

有了對數相似度後， 接下來便開始嘗試搬移左右子節點中的 $\beta_p(v)$ 以最大化對數相似度。 假設現在 $\beta_p(v) \in \mathcal{L}_p$ 且我們想要嘗試將其搬運至右節點 $\mathcal{L}_p$ 中， 則搬運後對數相似度僅需重新計算式 (4.7) 中次數的值：

$$\begin{aligned} C(\mathcal{L}_p, w) &\to C(\mathcal{L}_p, w) - C(\beta_p(v), w) \\ C(\mathcal{R}_p, w) &\to C(\mathcal{R}_p, w) + C(\beta_p(v), w) \\ C(\mathcal{L}_p) &\to C(\mathcal{L}_p) - C(\beta_p(v)) \\ C(\mathcal{R}_p) &\to C(\mathcal{R}_p) + C(\beta_p(v)) \end{aligned} \tag{4.8}$$

若搬運之後可導致對數相似度的上升， 則此搬運成功並付諸實行， 同時更新 $\mathcal{L}_p$ 與 $\mathcal{R}_p$ 及式 (4.7) 中的對數相似度和相關的次數。 反之若搬運 $\beta_p(v)$ 並無法使對數相似度上升， 則 $\beta_p(v)$ 繼續留在原節點。

於是，嘗試搬移左右節點中各個 $\beta_p(v)$ ， 直到不再有任何的搬移可導致對數相似度的上升時， 我們得到 $\mathcal{L}_p^*$ 與 $\mathcal{R}_p^*$ 以及其所對應的對數相似度， 此時位置 p 便已計算完畢。

最後，當所有的位置 p 都計算完成時， 再從中選擇對數相似度最高的 p^*， 並得到 $\mathcal{L}^*$ 與 $\mathcal{R}^*$， 並分別將其分配至節點 n 的左右子節點之中， 於是節點的分裂便完成了。

新分裂出來的左右子節點將被標示為葉節點， 原葉節點 n 則為成為非葉節點。 當所有葉節點的分裂皆無法再增加對數相似度時， 則決策樹生長演算法亦宣告完成。

4.1.2 決策樹修剪(pruning)演算法

綜觀整個決策樹的生長演算法，其乃是一由上而下的過程 (top-down)，而接下來的修剪演算法，則是由下而上的過程 (bottom-up)。修剪演算法主要是為了解決以往樹的生長不知該何時停止生長的問題 (early-stop)，故這邊先將決策樹 Φ_{DT} 進行完整的生長後，再使用留存資料以修剪決策樹，解決了太早或太晚停止生長演算法的問題。

接下來便開始介紹決策樹修剪演算法。決策樹修剪的過程中主要是以最大化留存資料相似度為目標，然而其中機率的計算則是參考內差式聶氏平滑法 (interpolated KN smoothing)：

$$P_{DT}(w_i|\Phi_{DT}(w_{i-N+1}^{i-1})) = \frac{max(C(\Phi_{DT}(w_{i-N+1}^{i-1}), w_i) - D, 0)}{C(\Phi_{DT}(w_{i-N+1}^{i-1}))} + \lambda(\Phi_{DT}(w_{i-N+1}^{i-1}))P_{KN}(w_i|w_{i-N+2}^{i-1}), \tag{4.9}$$

其中 $\Phi_{DT}(w_{i-N+1}^{i-1})$ 是歷史詞串 w_{i-N+1}^{i-1} 最後被分配到的葉節點所包含的所有歷史詞串，而 $P_{KN}(w_i|w_{i-N+2}^{i-1})$ 的計算則是如 2.3.2 節中式 (2.12) 所述。

接著定義所有非葉節點皆有其所屬的潛能 (potential)，潛能的計算乃是由非葉節點在分裂後與分裂前之間留存資料的對數相似度的差值。

於是，由深度最深的非葉節點開始一路修剪決策樹，若此非葉節點的分裂並不能使留存資料的對數相似度增加 (潛能為負) 又或者其潛能低於某個閾值，則修剪此非葉節點使其成為葉節點，並將其子節點中的歷史詞串全數返回，直到所有葉節點的父非葉節點的潛能皆符合需求為止。此修剪演算法和傳統的分類迴歸樹 (classification and regression trees, CART) 十分相似。至此，決策樹已成長完成，而決策樹最後僅取每個葉節點作為等價類別分類函數，如式 (4.9) 中的$\Phi_{DT}(\cdot)$。

然而，在式 (4.9) 中若需計算到訓練集中未見的歷史詞串時，由於此歷史詞串很有可能不會被分配到任何一個葉節點之中，於是此歷史詞串便不能夠

享有與其他歷史詞串共享同樣的機率分佈。 此時僅使用式 (4.9) 中的退化機率 $\Phi_{DT}(w_{i-N+1}^{i-1})$ 來做估算， 也就是對於任意詞w_i而言 $C(\Phi_{DT}(w_{i-N+1}^{i-1}), w_i) = 0$ 且 $\lambda(\Phi_{DT}(w_{i-N+1}^{i-1})) = 1$ 。

4.2 隨機森林語言模型

由於決策樹的生長時的快速交換演算法仍然是一種貪婪演算法， 於是其無法保證能夠得到最佳的決策樹， 於訓練集中如此， 對於測試集則更是如此， 故在早期的實驗中，決策樹語言模型並未能勝過 N 連語言模型， 或是和 N 連語言模型線性內差後才稍微進步了一些 [33]。

直到後來才有人開始嘗試使用隨機森林來處理這些問題。 隨機森林主要是經由集合眾多隨機決策樹 (randomized decision tree) 而來， 故底下將介紹如何生成隨機決策樹。

4.2.1 隨機決策樹

在 4.1 節訓練決策樹時， 會計算所有位置 p，並取其中使得對數相似度最大的 p^* 的結果作為分裂的依據， 故最後得到的決策樹只有一種可能 (deterministic)。 而隨機決策樹則是僅隨機選擇其中的一些位置 p 參與計算， 雖然這樣做無法達到節點 n 中的區域最佳解， 但是可能因此有機會得到一棵更好的決策樹。

每個 p 值是否參與計算主要是經由伯努力試驗 (Bernoulli trial) 而來， 並設其成功機率為 r， 每個位置的選擇與否於其他位置無關， 經隨機挑選出來的位置 $\bar{p}$ 將納入快速交換演算法中， 並同樣求得其中最佳的 $\bar{p}^*$ 值。

對於歷史詞串中的 $N-1$ 個位置而言， 最終還是會得到區域最佳解的 p^* 值

的機率為：

$$Q(r) = \frac{r}{1-(1-r)^{N-1}}, \tag{4.10}$$

且

$$\begin{aligned} \lim_{r\to 0} Q(r) &= \frac{1}{N-1} \\ \lim_{r\to 1} Q(r) &= 1 \end{aligned} \tag{4.11}$$

r 的值在整顆樹之中是固定的。 一般而言，當 r 的值越小以及 N 的值越大時， 生成的隨機決策樹之間就會越亂。 在最後的實驗中 r 的值為 0.5。

最後，當選擇了一組非空集合之位置 $\bar{p}$ 時， 便開始進行快速交換演算法。 然而和決策樹不同的是， $\beta_{\bar{p}}(v)$ 一開始的分配亦是隨機的。 對於每一個 $\beta_{\bar{p}}(v)$ ， 藉由成功機率為 0.5 的伯努力試驗去詢問：

1. 是否 $\beta_{\bar{p}}(v)$ 屬於 $\mathcal{L}_{\bar{p}}$？
2. 是否 $\beta_{\bar{p}}(v)$ 屬於 $\mathcal{R}_{\bar{p}}$？

若 (1) 得答案為真，則將其分配至 $\mathcal{L}_{\bar{p}}$； 同理，若 (2) 得答案為真，則將其分配至 $\mathcal{R}_{\bar{p}}$。 而若 (1) 和 (2) 的答案皆為否，則此些歷史詞串將不再往下處理， 亦即某些存在於訓練集中的歷史詞串最終並不會在存在於決策樹的葉節點之中。

接下來的步驟就和一般的決策樹語言模型一致。

4.2.2 由樹而林

當執行許多次隨機決策樹的生成後， 將會得到許多隨機決策樹， 也就是隨機森林。 每一棵隨機決策樹生成後皆可由式 (4.9) 得到一個經平滑化後的決策樹語言模型。 若總共有 K 顆樹， 且這些決策樹語言模型分別為 $DT_1, DT_2, ..., DT_K$， 則

最後隨機森林語言模型機率的估算便可經由集合這些決策樹語言模型而來：

$$P_{RF}(w_i|w_{i-n+1}^{i-1}) = \frac{1}{K}\sum_{k=1}^{K} P_{DT_k}\left(w_i|\Phi_{DT_k}(w_{i-n+1}^{i-1})\right), \tag{4.12}$$

其中 $\Phi_{DT_k}(w_{i-n+1}^{i-1})$ 表示歷史詞串 w_{i-n+1}^{i-1} 在隨機決策樹 DT_k 中所對應的葉節點。若 w_{i-n+1}^{i-1} 並不存在於葉節點之中，則仿照未見的歷史詞串將其機率退化至$P_{KN}(w_i|w_{i-N+2}^{i-1})$。

經由大數法則 (the Law of Large Numbers)，式 (4.12) 最終將會隨著樹的增加而逐漸收斂。

對於決策樹語言模型而言，當每一個歷史詞串皆對應至一個葉節點之時，則此模型即特化為 N 連模型。同理，隨機森林語言模型亦可特化為 N 連模型。在產生隨機決策樹的過程中，雖然經由每一棵決策樹所得到等價類別分類函數可能有好有壞，但隨著樹的數目增加，某些樹的弱點可能也因此被其他的某些樹所掩蓋，導致最終隨機森林語言模型的成功。

4.3 實驗結果與比較

與第三章的實驗相同，底下的實驗亦同時實作在「數位語音訊號處理」及「信號與系統」這兩個教學系統上。而其基礎實驗亦如 3.4 節中所述，乃經過聶氏平滑法的三連語言模型，簡稱為 Trigram。

隨機森林語言模型的生成主要使用約翰·霍普金斯大學所開發的隨機森林語言模型程式集 [34]，使用三連模型，且所有隨機森林語言模型皆由 1,000 顆隨機決策樹所構成。

需要注意的是，由於決策樹的修剪需要用到留存資料，故底下使用 4 摺的交叉驗證法(4-fold cross-validation)。首先將 1,000 顆樹分成 4 區，也就是每一區要

訓練出 250 顆樹。 接著隨機將訓練集中的文字語料拆成 4 份， 每一份輪流當這 4 區之中的留存資料 (25%) 以做修剪樹之用， 而剩下獨立於留存資料的部分 (75%) 則作為生長樹之用。 最後依式 (4.12) 將所有隨機決策樹集合在一起。

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)
Trigram	250.981	83.81	62.45	82.13
RF	**222.603**	**85.11**	**65.85**	**83.60**

表 4.3: 「數位語音訊號處理」隨機森林語言模型實驗結果

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)
Trigram	211.085	77.35	72.21	76.95
RF	**187.138**	**78.45**	**72.67**	**78.01**

表 4.4: 「信號與系統」隨機森林語言模型實驗結果

由表 4.3 及表 4.4 中可看出， 隨機森林語言模型不管是在中文或英文的辨識正確率皆勝過經過聶氏平滑法的三連語言模型， 在數位語言處理其絕對的辨識正確率在中文的部份進步約 1.3%， 英文的部份則為 3.4%。 而在信號與系統中其絕對的辨識正確率在中文的部份進步約 1.1%， 英文的部份則為 0.46%。 故隨機森林語言模型不僅適用在單語語言模型， 在雙語環境下亦能帶來許多進步。

4.4 本章總結

本章介紹了決策樹語言模型以及隨機森林語言模型。 這兩個模型和基於詞群之 N 連語言模型相似， 都希望能透過群組化的方式將相似的歷史詞串做分類以共享機率分佈， 解決一般 N 連語言模型資料稀疏性的問題。 透過隨機決策樹的生成並集合為隨機森林， 解決了一般決策樹語言模型因為使用貪婪演算法而無法得到最佳解的缺點， 最後不管是在混淆度或中英辨識率皆勝過經過聶氏平滑法的三連語言模型。

第 五 章　強健性語言模型調適

本章介紹如何將背景語言模型調適至目標課程的各種方法。一開始在 5.1 節先介紹何謂語言模型調適以及傳統的模型內差調適法，接著在 5.2 節使用隨機森林及眾林之林進行語言模型調適，並在 5.3 節以實驗分析各種調適方法的好壞，最後 5.4 節是本章總結。

5.1　語言模型調適法

在前幾章討論的語言模型，皆是直接使用和目標課程同領域 (in-domain) 的語料庫來訓練語言模型，這樣訓練出來的語言模型稱為特定領域語言模型 (domain-specific LM)，其優點是較貼近欲辨識的目標。

儘管如此，同領域的語料往往十分稀少，特別是像大學的課程系統，其內容充滿著許多專有名詞及學術詞彙，需要有許多修過課或相關領域的同學逐字逐句謄出人工轉寫，十分耗費心力，甚至一不小心就會謄出錯的語料，例如將數學名詞：特徵值「eigenvalue」標成「i 根 value」等，這對於統計式語言模型的傷害相當大。於是原本就飽受資料稀疏性所苦的 N 連語言模型，在課程系統上更顯得嚴重。

然而，還是有許多領域外 (out-of-domain) 的語料庫可供使用，常見的做法是將這些領域外的語料當做背景語料以訓練出背景模型，並同時使用和目標課程同領域的語料當做調適語料，將背景模型調適至目標課程之中，以解決資料稀疏的問題。

目前最常見的語言模型調適法為模型內差法 (model interpolation)，其將特定

領域語言模型和各種背景模型做加權相加，

$$P_{interp}(w_i|h) = \sum_k \lambda_k P_k(w_i|h), \tag{5.1}$$

其中 $P_k(w_i|h)$ 為特定領域語言模型以及其他各種背景模型的機率， 而 λ_k 則為各別模型所對應的權重， 通常使用發展集並以最大化事後機率為準則估算之， 並使其總和為 1，即：

$$\sum_k \lambda_k = 1. \tag{5.2}$$

在最後的實驗中主要使用 SRILM 程式集 [29] 所提供的程式：compute-best-mix 以進行內差權重的估算。

5.2 基於隨機森林的語言模型調適法

除了模型內差法之外， 亦有許多新的語言模型調適的方法陸續發表在論文上，如鑑別式語言模型調適 (discriminative language model adaptation) [35] [36] [37] 等。而基於隨機森林的語言調適法， 則是到這一兩年來才開始有相關研究在單語模型上 [15] [16]， 底下將詳細介紹之。

5.2.1 隨機森林語言模型調適法

在第四章中， 隨機森林語言模型不管是在混淆度或是辨識正確率上都獲得許多進步， 其使用大量的訓練語料以生長出隨機決策樹後， 再使用留存資料以修剪隨機決策樹， 最後集合所有隨機決策樹而成為隨機森林語言模型。 而隨機森林語言模型調適法亦使用了同樣的隨機決策樹生成與修剪演算法。

不同的是， 在生成隨機決策樹時， 一開始於根節點中放置的乃是所有可用語料的歷史詞串， 含目標課程訓練集、相似課程訓練集、以及其他各種背景語料

等， 並使用快速交換演算法直到節點的分裂無法使這些語料對數相似度上升為止， 此時的樹乃是一顆全面領域的樹。

接著，在修剪隨機決策樹時， 潛能的計算則是由原先的留存資料改為使用目標課程訓練集， 使得樹的修剪能夠反應出目標課程的特性， 以達到調適的效果，如圖 5.1。

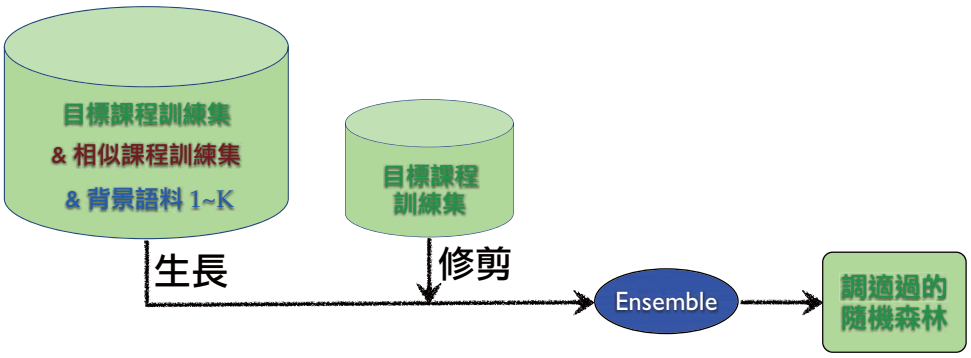

圖 5.1: 隨機森林語言模型調適法

5.2.2 眾林之林語言模型調適法

然而，由於背景語料的數量十分龐大， 導致在生成隨機決策樹時需要耗費相當長的時間， 且背景語料常常來自許多不同領域以及擁有不同主題， 僅用數量稀少的目標語料訓練集是否能夠好好的修剪樹是令人懷疑的。

故眾林之林 (forest of random forests, FRF) 語言模型調適法的想法是， 先將各種可用來訓練隨機森林的語料分門別類後， 再分別使用這些語料， 使其和目標課程訓練集共同生長出對應的隨機森林， 最後將這些隨機森林做模型線性內差而成為眾林之林， 如圖 5.2。

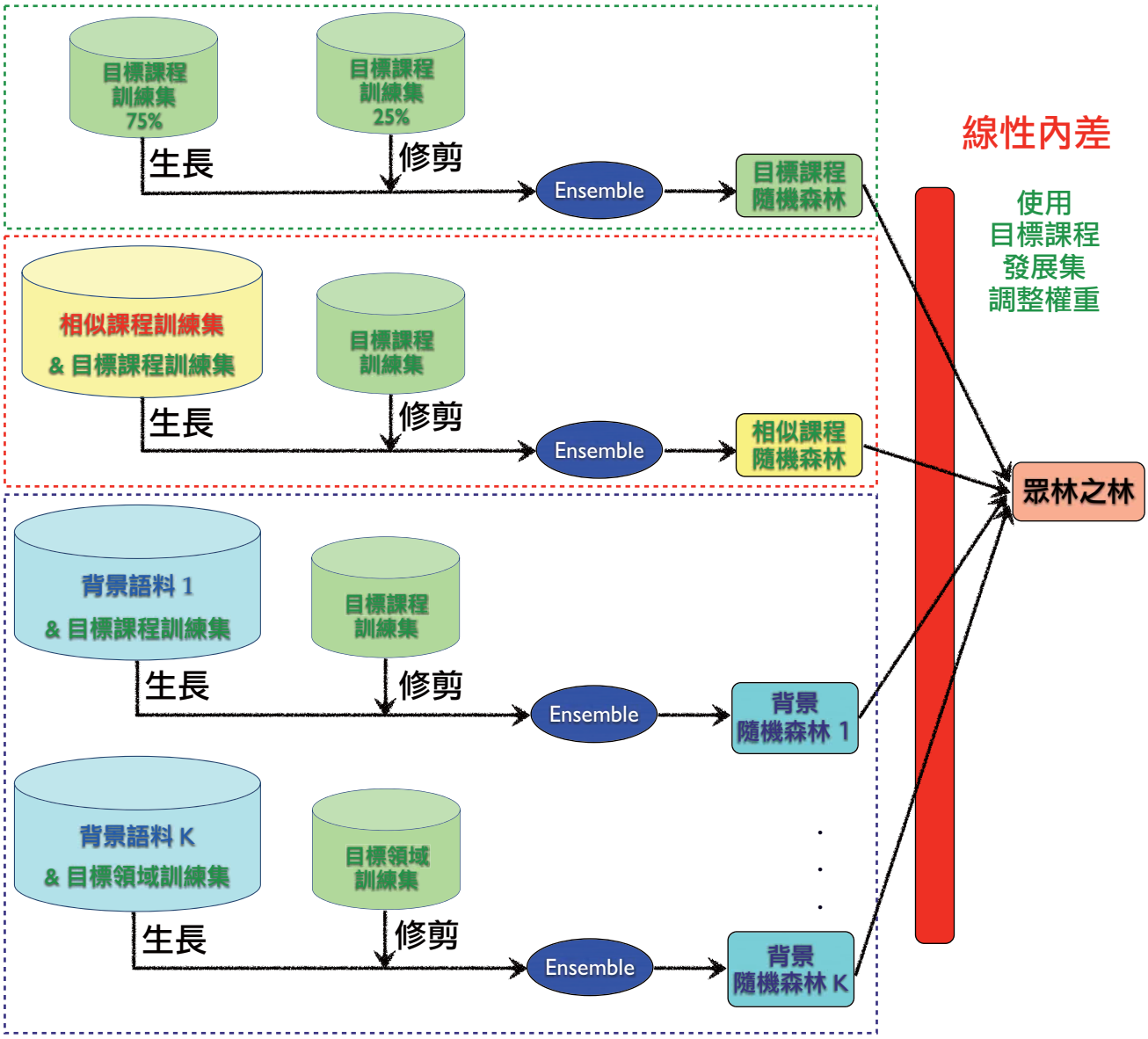

圖 5.2: 眾林之林語言模型調適法

5.3 實驗結果與比較

底下的實驗亦同時實作在「數位語音訊號處理」及「信號與系統」這兩門課當中， 除了課程本身的訓練語料之外，亦使用各種相似課程語料與背景語料以調適並貼近目標語料， 如表 5.1。

目標課程	目標課程訓練集	相似課程訓練集	其他各種背景語料
數位語音訊號處理	數位語音訊號處理訓練集	信號與系統訓練集	雅虎奇摩新聞網 + 噗浪網
信號與系統	信號與系統訓練集	數位語音訊號處理訓練集	雅虎奇摩新聞網 + 噗浪網

表 5.1: 調適課程系統時所使用的語料

各種語料的大小及中英分佈請參見第 2.4.2 節中表 2.3 所述。

5.3.1 串接所有語料並以其直接訓練模型

首先實驗最基本的方法， 也就是將目標課程系統中可以獲得的所有語料： 目標課程訓練集、 相似課程訓練集、 以及其他各種背景語料串接在一起後， 再以其訓練出經過聶氏平滑法的三連語言模型。

由表 5.2 及表 5.3 可看出， 直接使用所有語料來訓練 N 連語言模型的效果，居然比單純使用目標課程訓練集還來得差， 可見語料間的不匹配， 故必須使用語言模型調適技術才能為正確率帶來進步。

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	使用訓練語料
Trigram	**250.981**	**83.81**	**62.45**	**82.13**	僅使用目標課程訓練集
Trigram	1415.19	75.87	40.06	73.05	串接各種語料

表 5.2: 「數位語音訊號處理」語言模型調適實驗結果 1：串接所有語料並以其直接訓練模型

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	使用訓練語料
Trigram	**211.085**	**77.35**	**72.21**	**76.95**	僅使用目標課程訓練集
Trigram	1408.76	69.58	50.12	68.07	串接各種語料

表 5.3: 「信號與系統」語言模型調適實驗結果 1：串接所有語料並以其直接訓練模型

5.3.2 模型內差調適法

接著實驗傳統的模型內差調適法， 亦即使用個別語料來訓練出個別所屬的三連模型後， 再使用目標課程發展集來調整各模型的內差權重 λ_k ， 最後套用式 (5.1) 得到模型內差調適模型。

由表 5.4 及表 5.5 可看出經過線性內差過後的語言模型較未經調適的模型佳。其中，使用相似課程訓練集所訓練出來的三連模型， 其辨識率僅稍微落後於使用目標領域訓練集所訓練出來的三連模型， 在最後模型內差時， 其內差權重也比其他背景模型大上不少， 故模型內差的成功有絕大部分要歸功於此相似課程語料。

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	使用訓練語料	內差權重 λ_k
Trigram	250.981	83.81	62.45	82.13	目標領域訓練集	0.75655
Trigram	539.675	80.55	48.03	78.00	相似領域訓練集	0.185684
Trigram	3404.67	70.57	21.74	66.73	雅虎奇摩新聞網	0.0258864
Trigram	2039.56	74.50	32.67	71.22	噗浪網	0.0318796
模型內差以上所有模型	**227.79**	**84.25**	**63.41**	**82.61**		

表 5.4: 「數位語音訊號處理」語言模型調適實驗結果 2：使用模型內差法調適語言模型

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	使用訓練語料	內差權重 λ_k
Trigram	211.085	77.35	72.21	76.95	目標領域訓練集	0.767651
Trigram	468.852	74.78	61.08	73.72	相似領域訓練集	0.201737
Trigram	4486.67	62.48	24.71	59.55	雅虎奇摩新聞網	0.00671167
Trigram	2478.12	66.59	33.71	64.04	噗浪網	0.0238995
模型內差以上所有模型	**190.458**	**78.20**	**71.88**	**77.71**		

表 5.5: 「信號與系統」語言模型調適實驗結果 2：使用模型內差法調適語言模型

5.3.3 隨機森林語言模型調適法

然後實驗隨機森林語言模型調適法， 其在生長隨機決策樹時， 同時使用了目標課程訓練集、相似課程訓練集、以及其他各種背景語料， 使這些語料的歷史詞串皆置入根節點中以生長樹， 並使用目標課程訓練集以修剪樹。

表格上排 由表 5.6 及 5.7 可看出， 當直接使用所有語料來生長樹時， 僅靠少量目標課程訓練集並無法為樹做適當的修剪。

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	使用訓練語料
Trigram	**250.981**	**83.81**	**62.45**	**82.13**	僅使用目標課程訓練集
隨機森林語言模型調適法	739.933	84.05	55.52	81.81	串接所有語料

表 5.6: 「數位語音訊號處理」語言模型調適實驗結果 3：使用隨機森林語言模型調適法。

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	使用訓練語料
Trigram	**211.085**	**77.35**	**72.21**	**76.95**	僅使用目標課程訓練集
隨機森林語言模型調適法	625.273	77.48	62.83	76.35	串接所有語料

表 5.7: 「信號與系統」語言模型調適實驗結果 3：使用隨機森林語言模型調適法

同時，亦有可能是因為目標課程語料的句數太少， 再加上衆語料間的不匹配， 於是當一開始就將所有語料置全部入根節點時， 就註定了其失敗的命運，和串接所有語料並以其直接訓練模型的失敗有同工異曲之妙。

雖然如此，其辨識率仍然比直接串接所有語料並以其直接訓練模型佳。

5.3.4 衆林之林語言模型調適法

最後，使用衆林之林語言模型調適法， 先將目標課程訓練集、相似課程訓練集、以及其他各種背景語料分門別類後， 分別使其和目標課程訓練集串接以生長樹，並使用目標課程訓練集以修剪樹， 形成許多片隨機森林， 而衆林之林則是模型內差這許多片隨機森林而來。 要注意的是若有樹單純使用目標課程訓練集， 則其留置資料使用 4 摺的交叉驗證法取得， 如第四章所述。

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	使用訓練語料	內差權重 λ_k
RF	222.603	85.11	65.85	83.60	目標課程訓練集	0.580853
RF	244.245	84.55	64.80	83.00	相似課程訓練集+目標課程訓練集	0.4021
RF	869.162	83.43	53.94	81.11	雅虎奇摩新聞網+目標課程訓練集	0.0101375
RF	704.904	83.64	56.08	81.47	噗浪網+目標課程訓練集	0.00690957
FRF	**209.036**	**85.45**	**65.93**	**83.91**		

表 5.8: 「數位語音訊號處理」語言模型調適實驗結果 4：使用衆林之林模型調適法

語言模型	混淆度	中文正確率(%)	英文正確率(%)	總正確率(%)	使用訓練語料	內差權重 λ_k
RF	187.138	78.45	72.67	78.01	目標課程訓練集	0.69755
RF	202.874	78.21	71.88	77.72	相似課程訓練集+目標課程訓練集	0.300381
RF	678.333	76.99	63.62	75.95	雅虎奇摩新聞網+目標課程訓練集	0.000123158
RF	570.923	76.39	46.63	74.05	噗浪網+目標課程訓練集	0.00194567
FRF	**175.522**	**78.79**	**72.71**	**78.32**		

表 5.9: 「信號與系統」語言模型調適實驗結果 4：使用衆林之林模型調適法

由表 5.8 及表 5.9 ， 比較前面各實驗， 可看出衆林之林語言模型調適法， 和模型內差法、以及隨機森林語言模型調適法相比之下都較為進步， 顯示將背景語料依照來源拆分為幾個部分後再訓練隨機森林是可行的作法， 惟背景語料所訓練出來的隨機森林語言模型之內差權重仍然低落， 但相似語料之內差權重則提高甚多。

經過衆林之林語言模型調適法後， 最後混淆度和辨識正確率皆有上升。 和傳統的模型內差法相比， 在數位語言處理其絶對的辨識正確率在中文的部份進步約 1.2%， 英文的部份則為 2.52%。 而在信號與系統中其絶對的辨識正確率在中文的部份進步約 0.59%， 英文的部份則為 0.83%。

5.4 本章總結

本章介紹各種語言模型調適法以調適背景模型。 在最後的實驗中顯示由不匹配的語料直接訓練隨機森林語言模型並未能帶來進步， 但是藉由將背景語料依照來源分開後再個別訓練隨機森林語言模型， 最後集結個別隨機森林而成為衆林之林語言模型， 此時和傳統的模型內差法相比， 不管是在混淆度或者是中英辨識率上皆有所提昇。

第六章　結論與展望

6.1　總結與討論

本論文主要研究在中英雙語混合環境下的大學課程辨識系統之中， 各種語言模型的訓練與調適。

目前最受歡迎的語言模型仍然是傳統經過平滑化的 N 連語言模型， 其訓練的時間很短且已有許多寫好的程式可直接使用， 故一般皆採用經過平滑化的 N 連語言模型作為基本實驗之用。

然而， N 連語言模型對於訓練集中的稀有事件的掌握度仍然不佳， 特別是在中英雙語混合的語料上更是充斥著稀有事件， 此時基於詞群之 N 連語言模型便解決了這個問題。 其群組化各種在語意上或文法上相近的詞， 使其能夠共享機率分佈， 為稀有及未見事件帶來較好的估測。 本論文主要採用兩種分群演算法， 分別為以平均相互資訊最大化為基準 和以詞性標記為基準的作法， 兩者分別得力於資料驅動以及語言學的知識。 同時經由和 N 連語言模型的等權重做線性內差之後， 更彌補了其群組化後對於常見詞彙的傷害。 最後，將兩種不同的分群演算法所得到的模型再做一次等權重的線性內差， 得到了最大的進步。

接著，同樣是群組化的概念， 決策樹語言模型和隨機森林語言模型主要著力於歷史詞串的分群， 其利用快速交換演算法以及決策樹修剪演算法解決了傳統決策樹生成過程緩慢、 以及太早或太晚停止生長決策樹的缺點。 而隨機森林語言模型又較決策樹語言模型具有一般性， 其利用衆多的隨機決策樹嘗試尋求廣義最佳解， 改善傳統決策樹語言模型使用貪婪演算法而導致的解通常不是最佳解的窘境。 於是使用隨機森林最終為雙語環境的實驗帶來了穩健的進步。

最後是嘗試使用各種背景語料來改善課程語料稀疏的問題， 本論文中使用了

傳統的模型調適法以及基於隨機森林的語言模型調適法。 雖然直接使用隨機森林語言模型調適法並未能為混淆度和辨識率帶來進步， 但藉由分類背景語料以個別訓練隨機森林語言模型後再模型內差之， 亦即所謂的衆林之林語言模型調適法，其在最後的實驗帶來了穩定性的進步， 如表 6.1 及圖 6.1。

目標課程	基礎實驗	基於詞群之語言模型	隨機森林語言模型	衆林之林語言模型
數位語音訊號處理	82.13	82.65	83.60	83.91
信號與系統	76.95	77.16	78.01	78.32

表 6.1: 各語言模型之辨識總正確率(%)

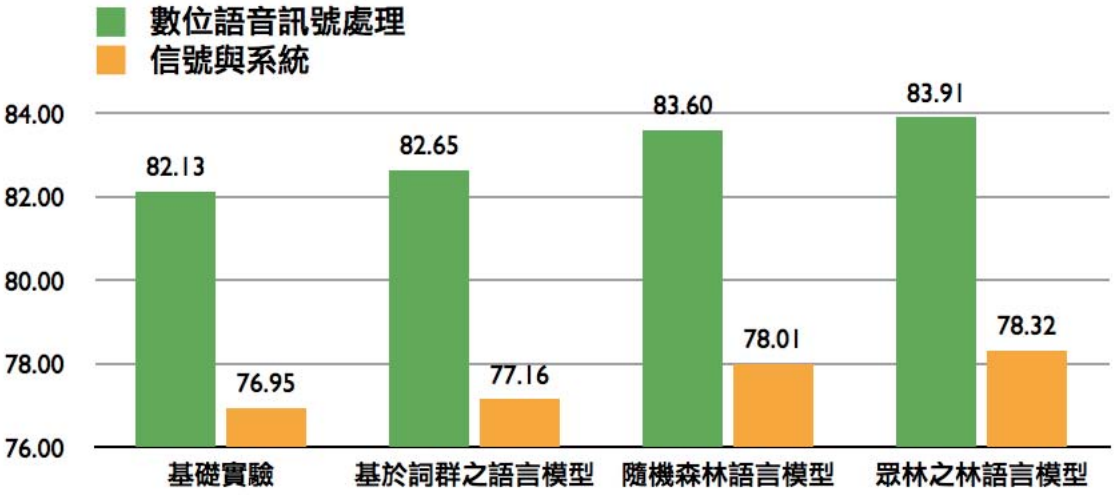

圖 6.1: 各語言模型之辨識總正確率(%)

6.2 雙語混和課程系統中最好的聲學及語言模型

如 2.2 節所述， 雙語混和課程語音辨識系統主要奠基在良好的聲學模型以及語言模型的估測上。 其中在語言模型的部份， 已於本論文中提出衆林之林語言模型調適法 (FRF)， 並以實驗證實其在不同系統中皆獲得穩定進步； 而聲學模型的部份， 則是以「以單位映射與模型回復建立之雙語混合聲學模型」這篇論文 [38] 中所提出的：經高斯層級映射回復並以疊加式調適法訓練之語者調適模型 (SA)， 以及經高斯層級映射回復並以最大相似度法則訓練之語者特定模型 (SD) 為最佳。實驗部分如表 6.2。

課程系統	聲學模型	中文正確率(%)	英文正確率(%)	總正確率(%)
數位語音訊號處理	語者調適模型	79.47	53.33	77.42
	語者特定模型	86.32	75.16	85.45
信號與系統	語者調適模型	72.86	59.95	71.86
	語者特定模型	80.08	81.02	80.16

表 6.2: 雙語混和課程系統實驗結果：使用衆林之林語言模型調適法及最好的聲學模型：語者調適模型及語者特定模型。

6.3 未來展望

在衆林之林演算法中僅單純將背景語料依來源做分類。然而近幾年來基於主題分析的各種强健性語言模型調適正不斷的發展當中，若能夠使用主題分析來分類各種背景語料，並配合衆林之林語言模型調適法，想必會比單純使用來源做分類的方法更為精緻，

且在第五章的實驗中，雅虎新聞以及噗浪網語料庫之內差權重皆幾乎為零，若完全不採用這兩個語料庫，想必也不會對最後的結果造成太大的更動。故到底是這兩個語料庫和課程系統語料庫幾乎不匹配，還是必須要使用其他更進步的方法，例如上面提到的使用主題分析等才能夠更加有效的利用這些背景模型，都是未來值得研究的方向。

此外，本論文主要研究的都是對於 N 連語言模型的改進。但是 N 連語言模型的計算主要皆是透過最大相似度估測法計算次數的比值，故這樣的計算僅是在離散空間中去做演算。

然而，以聲學模型為例，在過去十幾年來有許許多多各式各樣的聲學模型調適法在連續空間中做演算，於是若能將語言模型的問題也放到連續空間中去解，屆時就可能像聲學模型一般找到許多模型調適法以增進辨識率，這是筆者認為目前在語言模型中最有潛力的一個新題目。若能將其應用在中英混合環境的課程系統上，想必中英的特殊結構定能讓此新題目更添挑戰性。

參 考 文 獻

[1] "iTunes U - Learn anything, anywhere, anytime," http://www.apple.com/education/itunes-u.

[2] James Glass, Timothy J. Hazen, Lee Hetherington, and Chao Wang, "Analysis and processing of lecture audio data: Preliminary investigations," in *HLT-NAACL Speech Indexing and Retrieval Workshop*, 2004.

[3] A. Park, T. J. Hazen, and J. R. Glass, "Automatic processing of audio lectures for information retrieval: Vocabulary selection and language modeling," in *ICASSP*, 2005.

[4] J. R. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," in *Interspeech*, 2007, pp. 2553–2556.

[5] S. Mukhopadhyay, B. Smith, "Passive capture and structuring of lectures," in *Proceedings of the ACM International Conference on Multimedia*, 1999, pp. 477–487.

[6] Y.C. Chan, P.C. Ching, T. Lee and H. Cao, "Automatic speech recognition of Cantonese-English code-mixing utterances," in *Interspeech*, 2006.

[7] T.-L. Tsai, C.-Y. Chiang, H.-M. Yu, L.-S. Lo, Y.-R. Wang, and S.-H. Chen, "A study on Hakka and mixed Hakka-Mandarin speech recognition," in *ISCSLP*, 2010.

[8] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," in *Computational Linguistics*, 1992, vol. 18, pp. 467–479.

[9] F. Jelinek, "Up from trigrams! The struggle for improved language models," in *EUROSPEECH*, 1991, p. 1037–1040.

[10] R. A. Solsona, E. Fosler-Lussier, H.-K. J. Kuo, A. Potamianos, and I. Zitouni, "Adaptive language models for spoken dialogue systems," in *ICASSP*, 2002.

[11] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *ICASSP*, 1995.

[12] S. F. Chen, J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Association for Computational Linguistics*, 1996.

[13] P. Xu and F. Jelinek, "Random forests in language modeling," in *EMNLP*, 2004.

[14] S. Martin, J. Liermann, and H. Ney, "Algorithms for bigram and trigram word clustering," in *Speech Communication*, 1998, vol. 24, pp. 19–37.

[15] A. Deoras, F. Jelinek, and Y. Su, "Language model adaptation using random forests," in *ICASSP*, 2010.

[16] I. Oparin, L. Lamel, and J.-L. Gauvain, "Improving Mandarin Chinese STT system with random forests language models," in *ISCSLP*, 2010.

[17] S. N. Sridhar and K. K. Sridhar, "The syntax and psycholinguistics of bilingual code mixing," in *Canadian Journal of Psychology 34(4)*, 1980.

[18] P. Li, "Spoken word recognition of code-switched words by Chinese-English bilinguals," in *Journal of Memory and Language*, 1996.

[19] D.-C. Lyu, R.-Y. Lyu, "Language identification on code-switching utterances using multiple cues," in *Interspeech*, 2008.

[20] D.-C. Lyu, R.-Y. Lyu, Y.-C. Chiang, and C.-N. Hsu, "Speech recognition on code-switching among the Chinese dialects," in *ICASSP*, 2006.

[21] R. Lejeune, J. Baude, C. Tchong, H. Crepy, and C. Waast-Richard, "Flavoured acoustic model and combined spelling to sound for asymmetrical bilingual environment," in *Interspeech*, 2005.

[22] M.-R. Wu, "Initial study on Chinese/English bilingual speech recognition based on lecture recording," in *M.S. thesis, NTU*, 2007.

[23] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Workshop Pattern Recognition in Practice*, 1980.

[24] I. Good, "The population frequencies of species and the estimation of population parameters," in *Biometrika*, 1953, vol. 40, pp. 237–264.

[25] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," in *IEEE Transactions on Acoustics, Speech, and Signal Process-35, no.3*, 1987, pp. 400–401.

[26] "Yahoo! Kimo News portal," `http://tw.news.yahoo.com`.

[27] "Plurk," `http://www.plurk.com`.

[28] S.-P. Liao, "Enhanced language modeling for Chinese speech recognition," in *M.S. thesis, NTU*, 2003.

[29] A. Stolcke, "SRILM-An extensible language modeling toolkit," in *ICSLP*, 2002.

[30] T.R. Niesler, E.W.D. Whittaker and P.C. Woodland, "Comparison of part-of-speech and automatically derived category-based language models for speech recognition," in *ICASSP*, 1998.

[31] "Academia Sinica, Part-of-Speech Tagger," `http://ckipsvr.iis.sinica.edu.tw`.

[32] C.-F. Yeh, C.-Y. Huang, L.-C. Sun, and L.-S. Lee, "An integrated framework for transcribing Mandarin-English code-mixed lectures with improved acoustic and language modeling," in *ISCSLP*, 2010.

[33] G. Potamianos and F. Jelinek, "A study of n-gram and decision tree letter language modeling methods," in *Speech Communication*, 1998, vol. 24(3), pp. 171–192.

[34] Yi Su, "Random forest language model toolkit," `http://www.clsp.jhu.edu/~yisu/rflm.html`.

[35] X. Liu, W. J. Byrne, M. J. F. Gales, and P. C. Woodland, "Discriminative language model adaptation for Mandarin broadcast speech transcription and translation," in *ASRU*, 2007.

[36] X. Liu, M. J. F. Gales, and P. C. Woodland, "Context dependent language model adaptation," in *Interspeech*, 2008.

[37] X. Liu, M. J. F. Gales, and P. C. Woodland, "Use of contexts in language model interpolation and adaptation," in *Interspeech*, 2009.

[38] C.-F. Yeh, "Bilingual code-mixed acoustic modeling by unit mapping and model recovery," in *M.S. thesis, NTU*, 2011.