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

課程錄音之自動關鍵用語擷取及摘要

Automatic Key Term Extraction and Summarization from Spoken Course Lectures

陳縕儂

Yun-Nung Chen

指導教授：李琳山 教授

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

課程錄音之自動關鍵用語擷取及摘要

Automatic Key Term Extraction and Summarization from Spoken Course Lectures

本論文係 **陳縕儂** 君（學號R98922004）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

鄭秋豫

莊仁輝

系主任 呂育道

誌謝

首先，我要由衷的感謝我最敬愛的指導教授 – 李琳山老師。老師在整個碩士期間對我的指導與照顧不計其數，我非常感謝老師在研究上及生活上給我的協助。老師總是能夠給我很有用的建議，幫助我能在研究的主題上思考的更多更豐富，也時常鼓勵我，在碩士這段期間，我成長了許多，而這些有許多都是老師提供的，非常感謝老師。接著要感謝我一直以來的夥伴 – 黃宥，從大四決定要一起加入語音實驗室開始，我們就一直兩人合作努力，並且時常討論和聊天。我覺得加入語音實驗室最大最大的收穫應該就是和你成爲好朋友！還很幸運地和你一起去加州報告順便觀光，這是很特別的經驗，相信這樣的難得經驗會讓我永生難忘。眞的非常愛你，也很不捨碩士畢業以後就要各奔東西，你千萬別忘了我耶！再來還有另外一個合作夥伴 – 小蘋果，能跟你一起去布拉格讓我們變熟，也更了解你，覺得你眞可愛！接著還有在研究上給我建議和幫助並時常和我討論的宏毅哥、在數學上幫助我證明與學習的小安、喝半杯啤酒就醉的葉青峰、酸人實在是非常厲害的李尚文、打 BG 絕對不缺席的楊皓中、三國殺總是超入戲的哈哥、一路走來和我很相似的左逼、和我熟最久的牛、坐在旁邊以前曾是我偶像的歐吉桑、帶我進入語音領域的小妞、砲火莫名強大的阿邦、拯救工作站超辛苦的馬雅、和我一起去中研院錄音的李杰、眞誠的小銘和有型的 Jeff 、以及幫我改英文的 Aaron。當然還有一起當助教的趴呢張暘、一起做 project 的涂涂、貼心送我們畢業花束的薛琇文和陳筱雯、總是很辛苦的 train 我的 model 的楊易倫。當然還有一直支持我的家人們，能夠無憂無慮的念書，都要感謝你們一路的栽培，才能有今天的我。最後，還有一路上爲我打氣支持我，最重要的人 – 呂哲安，沒有你一路的陪伴，相信我也不會在碩士期間這麼快樂，有你的陪伴，眞好！謝謝大家，謝謝你們一路上的陪伴與支持，這是令我能夠順利完成碩士學位的最大動力。

摘要

本論文針對語音文件提出能夠自動擷取重要資訊的方法，其資訊包含關鍵用語及摘要。首先，本論文提出一有效擷取關鍵用語的方法，並以課程錄音做爲實驗語料，其中包含了語音訊號、錄音文本、以及課程所使用之投影片。我們將關鍵用語分成兩類型：關鍵片語及關鍵詞，並對兩種類型之用語發展不同的擷取方法，並依序擷取兩種類型之關鍵用語。首先使用分歧亂度擷取關鍵片語，接著使用韻律、詞彙、語意三大類的特徵藉由機器學習來訓練分類器並抽取關鍵詞，其中機器學習包含一個非督導式學習 (K 平均模範) 以及兩個督導式學習 (調適性推昇法及類神經網路) 的方法。實驗顯示本論文提出之關鍵用語擷取方法明顯較傳統方法好，而且三大類不同的特徵對於關鍵用語之選擇皆有幫助並具有加成性。有了關鍵用語的資訊，本論文進一步提出使用圖論之隨機漫步演算法爲此類語音文件自動抽取摘要。在此演算法中，文件中的每個句子被表示成一圖上的點，兩點相連邊的權重是根據兩句子間的主題相似度所估算的。 基本的概念爲，假設若一句子和較多重要的句子潛藏語意上較相似，則此句子應該較爲重要。此方法藉由關鍵用語的資訊來輔助估算語音文件中每個句子的重要性，並藉由圖論的方法來全面性地考慮所有句子彼此間的相似性，進而重新估算各句子的重要性。實驗結果顯示，不論在何種評估方法下，此方法所抽取出的摘要都較傳統基於潛藏主題亂度所抽取之摘要更好，而關鍵用語的資訊及隨機漫步演算法都對語音文件中的摘要抽取具有極大的貢獻。

Abstract

This thesis focuses on information extraction from spoken documents, extracting two important information: key terms and summaries. It proposes a set of approaches to automatically extract key terms from spoken course lectures including audio signals, ASR transcriptions and slides. We divide the key terms into two types: key phrases and keywords and develop different approaches to extract them in order. We extract key phrases using right/left branching entropy and extract keywords by learning from three sets of features: prosodic features, lexical features and semantic features from Probabilistic Latent Semantic Analysis (PLSA). The learning approaches include an unsupervised method (K-means exemplar) and two supervised ones (AdaBoost and neural network). Very encouraging preliminary results were obtained with a corpus of course lectures, and it is found that all approaches and all sets of features proposed here are useful. With key terms, this thesis proposes an improved approach for spoken lecture summarization, in which random walk is performed on a graph constructed with automatically extracted key terms and PLSA. Each sentence of the document is represented as a node of the graph and the edge between two nodes is weighted by the topical similarity between the two sentences. The basic idea is that sentences topically similar to more important sentences should be more important. In this way all sentences in the document can be jointly considered more globally rather than individually. Experimental results showed significant improvement in terms of ROUGE evaluation.

目錄

圖目錄

表目錄

第一章　導論

Introduction

1.1　研究動機

在這個充斥著數位內容的網路興盛時代，其中包含許多生活上的活動資訊，像是教育、娛樂、商業交易，以及社交網路等的互動，都是很好的例子。在網路資訊中，最引人注意的即爲多媒體。由於所有的多媒體文件都包含聲音訊號，而這些聲音訊號通常會描述核心概念，所以使用者需要在語音文件中有效率的索引、搜尋、以及瀏覽特定資訊。但是語音文件不像手寫文件一樣具有良好的結構，它們多半是影像或語音訊號，使用者並不能直接在螢幕上瀏覽並快速閱讀已呈現好的手寫文件，故在這些聲音資訊 (語音文件) 中擷取重要的資訊即成爲一非常重要的研究。

此論文提出一些方法在課程語料的語音文件中處理資訊擷取 (Information Extraction) 的問題。由於終身學習的概念現今越來越普遍，而網路上也提供了非常大量的課程錄音，其中像是麻省理工學院的課程錄音瀏覽器 (MIT Lecture Browser) [1] ，即爲一語音課程錄音的例子。然而，使用者無法非常方便而有效率地瀏覽這些大量的課程錄音，或是從中尋找特定的資訊，故由課程錄音中抽取重要的資訊 – 關鍵用語 (Key Term) 及摘要 (Summary) ，變得非常重要及具有吸引力。

同時，由於現在語音辨識還未完全成熟，故依然存在許多辨識錯誤的結果，造成由語音文件中擷取重要資訊變的更加困難。所以從語音文件中擷取資訊和從文字文件的最大不同，即爲要避免辨識錯誤的字詞造成較大的影響，同時考慮利

用語音特有的韻律特徵 (Prosodic Feature) 去幫助資訊擷取，進而發展出特爲語音文件資訊擷取的適當方法。

1.2 語音辨識

現今，在語音辨識上，最廣泛被採用的爲隱藏式馬可夫模型 (Hidden Markov Model)。隱藏式馬可夫模型是在1966年由統計學家波氏 (Baum) 提出 [2] 。五年後，由兩個實驗室獨立地將隱藏式馬可夫模型套用在語音上。一個是當時在卡內基美隆 (Carnegie Mellon University) 唸書的貝克氏 (Baker) ，在讀了波式的隱藏式馬可夫模型的論文後，將演算法應用到語音處理上。另一個則是當時在 IBM 瓦特森實驗室 (Watson Lab) 的詹氏 (Frederick Jelinek) ，受到夏農 (Shannon) 的資訊理論 (Information Theory) 影響而發展出跟隱藏式馬可夫模型等價的系統。IBM的系統跟貝克氏的系統基本上一樣，唯獨解碼演算法 (Decoding Algorithm) 不同。貝克氏的系統使用維特比 (Viterbi) 解碼演算法，而 IBM 則使用堆疊 (Stack) 解碼演算法。

之後的 20 年，由於美國國防部陸續生出了許多語音辨識方面的計畫。像是 Resource Management, Wall Street Journal, Air Traffic Information System, Broadcast News, CALLHOME 等等。藉由這些計畫的交流，隱藏式馬可夫模型也因此在研究社群中廣泛地被接受。也因爲它簡潔有效率，又有高正確率的優點，至今仍鮮有方法可以撼動它的地位。因此，以下將簡述現今被普遍使用在語音辨識的貝氏定理 (Bayes Theorem) 架構，隱藏式馬可夫模型便是建築在此架構上。

如果我們把語音辨識想做是給定一串訊號，找出一個句子它唸起來最像這串訊號。在機率模型的架構下我們可以用條件機率來表示。假設 x 是給定的觀測語句 (Observation) ，也就是處理訊號過後產生的特徵向量序列 (Feature Vector

Sequence)，X 是所有可能的觀測語句。則要從所有文句 Y 中找出機率最大的文句 y^* 可表示成

$$y^* = \arg\max_{y \in Y} P(y \mid x) \tag{1.1}$$

其中 y 爲所有文句 Y 中的某一句，$P(y \mid x)$ 代表在 x 發生時，文句 y 的事後機率。

剩下的問題就是，我們要如何對 $P(Y \mid X)$ 建立機率模型。以上說起來很簡單，但實際上由於語音辨識本身的複雜度，對於 $P(Y \mid X)$ 直接建立模型是有困難度的。於是我們就用貝氏定理 (Bayes Theorem) 來轉換建模方向。

$$\begin{aligned} y^* &= \arg\max_{y \in Y} P(y \mid x) \\ &= \arg\max_{y \in Y} \frac{P(x \mid y)P(y)}{P(x)} \end{aligned} \tag{1.2}$$

由於我們在辨識的時候都是一句一句，所以分母的 $P(x)$ 在辨識一個句子的時候並不會影響 y 的選擇，因此可以重寫成：

$$\begin{aligned} y^* &= \arg\max_{y \in Y} P(y \mid x) \\ &= \arg\max_{y \in Y} P(x \mid y)P(y) \end{aligned} \tag{1.3}$$

因此，語音辨識的問題便被拆成如何替 $P(X \mid Y)$ 及 $P(Y)$ 建立模型。其中 $P(X \mid Y)$ 可以看作是，給定一個句子的情況下，唸出來的訊號是 X 的機率分佈。而 $P(Y)$ 則可以看成，出現句子 Y 的機率分佈。它們可以分別使用聲學模型 (Acoustic Model) 及語言模型 (Language Model) 來計算出來。

在此篇論文，我們使用HTK [3] 來實做語音辨識。而語音辨識以詞爲單位的辨識率 (Word Accuracy) 在中英混合 (Code-Mixing) 下如本論文的實驗語料，會下

降許多。

1.3 相關研究

於文字文件中自動擷取關鍵用語及自動摘要在自然語言處理領域受到高度重視，但是卻鮮少有針對語音文件做資訊擷取之相關研究。近年來有幾篇論文對於經由語音辨識後結果進行關鍵用語擷取之研究 [4,5] ，套用文字領域的技術至語音辨識後結果上，而也有越來越多研究利用語音文件中的獨特訊號來進行自動摘要，例如抽取聲音上的特徵等等 [6,7] 。無論是關鍵用語擷取或是自動摘要，都是對用語或句子進行重要性之評估。目前針對文字文件中的資訊擷取技術，大致上可分爲三大類：依照經驗法則、透過機器學習，或是藉由圖論來擷取。

1.3.1 經驗法則 (Empirical Rule)

以下簡述三種不同的關鍵用語擷取方法，分別爲傳統的詞頻與逆向文件頻率 (Term Frequency - Inverse Document Frequency; TF-IDF) 、前後文相關 (Context Dependent) 、以及文字共同出現 (Co-Occurrence) 。

詞頻與逆向文件頻率 (Term Frequency - Inverse Document Frequency; TF-IDF)

在關鍵用語擷取方面，這一類的方法通常由簡單的想法出發，利用一些規則進行篩選。其中最爲廣泛應用的是詞頻與逆向文件頻率。此方法在資訊檢索領域被大量利用 [8] 。直覺來看，出現越多次的用語理應越重要，但是如果在每個文件都出現很多次，此用語很有可能只是高頻常用詞，像是功能詞 (Function Word) ，不帶有重要意義，並非我們目標擷取的關鍵用語。所以我們希望擷取到的詞應具有

高頻且分布集中的特性。嚴謹地定義詞頻與逆向文件頻率如下：

$$tf_{i,j} = \frac{n(w_i, d_j)}{\sum_k n(w_k, d_j)} \tag{1.4}$$

$$idf_i = \log \frac{|D|}{|\{d : w_i \in d\}|} \tag{1.5}$$

$$tfidf_{i,j} = tf_{i,j} \cdot idf_i \tag{1.6}$$

其中 $n(w_i, d_j)$ 代表某個詞 w_i 出現在文件 d_j 中的次數，$|D|$ 爲文件總數，而 $|\{d : w_i \in d\}|$ 代表包含詞 w_i 的文件數。由上式可看出，我們希望擷取的關鍵用語既具有很高的詞頻，也須具有很高的逆向文件頻率。因此定義詞頻與逆向文件頻率如 (1.6)，藉由排序 $tfidf_{i,j}$ 以擷取關鍵用語 [4]。

前後文相關 (Context Dependency)

此也爲一常被使用的方法。此方法想法很簡單：一詞如果常與另一詞一起出現，這兩個詞很有可能爲一高頻片語，也就很有可能爲關鍵用語。我們分別定義一個用語 x 的左鄰 (Left Context) 和右鄰 (Right Context) 如下：

$$LC_x(w) = \frac{freq(wx)}{freq(x)}, RC_x(w) = \frac{freq(xw)}{freq(x)} \tag{1.7}$$

其中 x 爲任一長度的用語，而 w 爲任一個詞。我們擷取具有高度前後文相關的用語爲關鍵用語；換句話說，若 $LC_y(x)$ 和 $RC_x(y)$ 皆很高，則 x 和 y 就可連接成爲一關鍵用語。而此方法需計算所有詞的左鄰及右鄰相當耗費工夫，爲了提高效率有些研究使用字首樹 (Prefix Tree) 或字尾樹 (Posfix Tree) 的方法。其中最符合要求的是後綴樹 (Radix Tree 或 Patricia Tree; PAT Tree)。這個方法最初來自資訊檢索 [9]

，之後被廣泛應用在中文用語擷取 [10–13] 。

文字共同出現 (Co-Occurrence)

文字共同出現 (Co-occurrence) 是另一種延伸的概念，想法是不單侷限在左鄰右鄰，還考慮了整體的分佈。假設某個詞不是均勻地跟每個高頻詞一起出現，而是跟某些特定高頻詞共同出現，那這個詞就比較重要。首先我們可以選出一個詞的集合 G ，這個詞集可以是經詞頻篩選過的高頻詞集合。於是某個詞跟這些高頻詞共同出現的次數可以視爲一個分佈，接著我們可以計算這個分布和均勻分佈之間的差距，計算分佈的差異可以使用卡方分佈 (χ^2 distribution) ：

$$\chi^2(w) = \sum_{g \in G} \frac{(freq(w,g) - n_w p_g)^2}{n_w p_g} \tag{1.8}$$

其中 n_w 是 w 出現的次數，p_g 是高頻詞 g 的機率，而 $freq(w,g)$ 是 w 與 g 共同出現的次數。若算出來的 $\chi^2(w)$ 越大，就代表詞 w 的分佈與均勻分佈的差異越大，也就表示 w 和某些高頻詞 g 時常一起出現，於是詞 w 就越可能是重要的詞 [14] 。

上述幾種方法均屬於經驗法則，這類的方法很多，出發點通常很簡單，而且實作也較容易，此類方法最大的缺點在於常有例外，我們只能說這類方法提供了一些趨勢和傾向，算出分數較高的詞有可能是關鍵用語，但並不絕對。

1.3.2 機器學習 (Machine Learning)

近年來機器學習方法日趨成熟，不管是關鍵用語擷取，或是自動摘要，此方法都廣泛地被應用在其中。例如支撐向量機 (Support Vector Machine; SVM) [15] 、條件隨機域 (Conditional Random Field; CRF) [16] 、單純貝式分類器 (Naïve Bayes Classfier) [17] 、還有一些使用推昇方法 (Boosting) [18] 的研究等等。

機器學習方法是透過學習訓練集 (Training Set) 的特徵 (Feature)，訓練一分類器 (Classifier) ，進而分辨出測試集 (Testing Set) 的分類。在關鍵用語及自動摘要方面，基本上都是假設欲分類的標記爲二元 (Binary) ，即 $+1$ 表此用語「屬於」關鍵用語或此句子「屬於」摘要，-1 則反之。需先對每個欲分類的對象抽取重要性特徵，基本上關鍵用語擷取部分可能會對其抽取詞性 (Part of Speech; PoS)、詞的位置、前後詞、還有一些關於文件的特性 (例如是否屬於標題等等) ，而摘要則對每個句子抽取其所在位置，其中所含詞之重要性等等特徵，藉由這些特徵訓練分類器來爲測試集進行分類。

機器學習方法有許多種，最主要的是督導式學習 (Supervised Learning) 和非督導式學習 (Unsupervised Learning)。督導式學習的效果雖然都不錯，但缺點是需要人工標註的正確答案，實作上較爲困難。基於人工標註答案的得來不易，也有部分研究使用半督導式學習 (Semi-Supervised Learning) [19] ，只需要少量的人工標註答案來幫助學習，近年來這類的方法很受注目 [20] 。

1.3.3 圖論 (Graph Theory)

透過圖論來計算重要性的方法也有相當多種，在關鍵用語擷取，較普遍的方法爲文字排名演算法 (TextRank) [21] ，是在圖上抽取重要的用語。而在自動摘要部分，也有一相似對應方法 – 詞彙特徵排名演算法 (LexRank) [22] ，是在圖上抽取重要的句子。以上兩方法都由資訊檢索享有盛名之網頁排名演算法 (PageRank) [23] 衍伸而來。以下針對三種方法做介紹。

網頁排名演算法 (PageRank)

在資訊檢索享有盛名的此演算法是由 Google 所提出，針對數量過大的網頁進行搜

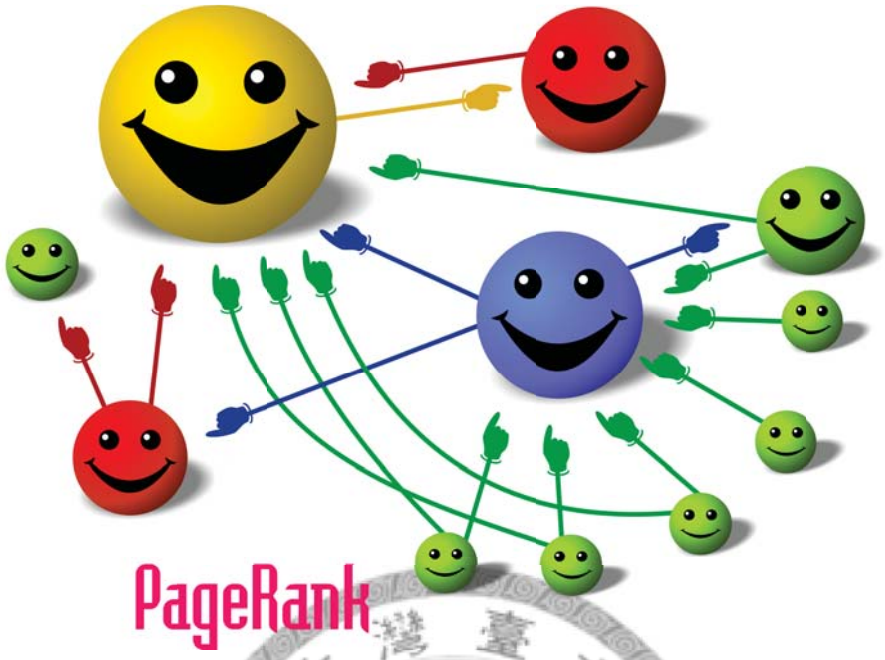


圖 1.1: 網頁排名演算法之圖示 (from Wikipedia)

尋並提出一較好的排序方法。此演算法的想法是，若一網頁很重要，則此網頁連結出去的網頁也很重要；換句話說，某個網頁的重要性取決於其他連結到這個網頁的頁面。故我們可以定義一張有向圖 (Directed Graph) $G = (V, E)$ ，在這張圖上，每一個點是一個網頁，每一條邊是網頁和網頁間的超連結，如圖 1.1。其中 V 是點的集合，E 是邊的集合。對於某個網頁 $v \in V$ ，計算 v 的重要性爲

$$PR(v) = \frac{1-d}{N} + d \sum_{u \in In(v)} \frac{1}{|Out(u)|} S(u) \tag{1.9}$$

其中 $In(v)$ 爲所有連結到 v 的網頁集合，而 $Out(v)$ 是 v 連結出去的網頁的集合。N 是考慮的網頁總數，d 是介於 0 到 1 之間的阻尼因數 (Damping Factor) ，代表使用者隨機瀏覽這個網頁的機率，用以避免有些網頁連結太少甚至沒有連結的問題。

文字排名演算法 (TextRank)

把網頁排名演算法套用至關鍵用語擷取上，得先替整份文件建構一個圖，於是上面每個點是一個詞，每個邊代表詞與詞的共同出現關係。這個演算法的想法則變成，若此詞很重要，與這個詞一起出現的詞也很重要。因此透過文字排名演算法，可以透過排序 $PR(v)$ 得到關鍵用語。

詞彙特徵排名演算法 (LexRank)

將相同概念套用到自動摘要上，先替整個文件建造一個圖，圖上的點則改爲文件中的句子，邊則表示句子和句子間的詞彙重複程度，也就是利用重複出現的詞彙來計算句子兩兩之間的相似性。故若有一句子很重要，則和此句子很相似的句子也應該很重要，最後依照句子的重要性來排序，得到自動摘要之結果。

除了此外，尚有其他圖論的方法 [24,25] 是藉由找出圖上連結兩個以上的連通單元 (Connected Component) 的這些點，這些點由於被連通單元所共用，代表彼此間的關係較緊密，可能是較爲核心、包含重要意義，因此比較重要，則可以抽取圖上的連通單元，作爲重要資訊 (用語或句子) ，此方法在此處則不多作贅述。

1.4 章節安排

本論文第二章將先介紹本論文會使用到的背景知識，包含了機率式潛藏語意分析模型 (Probabilistic Latent Semantic Analysis; PLSA) 及兩種督導式學習 (Supervised Learning) 的方法 – 調適性推昇法 (Adaptive Boosting; AdaBoost) 及類神經網路 (Artificial Neural Network) ，這皆爲在後面章節會使用到的基本知識與技術。接著，第三章將介紹本論文所提出，由語音文件中自動關鍵用語擷取的方法及流程，並且藉由實驗來驗證此方法之效度。第四章則針對語音文件之自動摘要所提

出之方法做介紹，而此處同時考慮了第三章的關鍵用語資訊，也進行實驗分析並驗證提出方法的效用。最後，第五章會總結本論文所提出之方法，並且陳述其貢獻， 最後對於未來提出了研究可能的規劃及展望。

第二章　背景知識

Background Review

本章會先介紹一重要潛藏主題模型之詳細數學理論，接著詳述兩種督導式 (Supervised) 機器學習 (Machine Learning) 的分類器 (Classifier) 訓練理論及流程，分別爲調適性推昇法 (Adaptive Boosting; AdaBoost) [26] 及類神經網路 (Artificial Neural Network) [27] 。

2.1　機率式潛藏語意分析模型 (Probabilistic Latent Semantic Analysis; PLSA)

機率式潛藏語意分析 (Probabilistic Latent Semantic Analysis; PLSA) 時常用來分析文件的潛藏主題，此爲霍氏 (Thomas Hofmann) 於 1999 年提出 [28] 。基本概念由傳統潛藏語意分析模型 (Latent Semantic Analysis) 衍伸而來 [29] 。

傳統潛藏語意分析藉由找出一組用語 (Term) 與文件 (Document) 之間的潛藏關係來分析文件內部的潛藏主題，其方法被應用在很多方面上，特別在資訊檢索分析方面展現的相當大的進步。主要的做法是將詞與文件之間的關係 – 文件-詞矩陣 (Document-Word Matrix) 進行奇異值分解 (Singular Value Decomposition; SVD)，達到縮減維度 (Dimension Reduction) 的效果。其想法是把詞與文件的關係投射到一個較小維度的空間上，而這個空間就是所謂的潛藏主題空間 (Latent Topic Space)，這空間的每一個維度都都代表一主題。

而機率式潛藏語意分析則針對傳統潛藏語意分析的缺失做了改進。由於傳統潛藏語意分析模型有一強烈假設，假設每一主題彼此之間皆爲獨立，但是在實際

情形來說較不合理，故機率式潛藏語意分析模型對此缺失做加強，發展更完整的理論架構。這套統計理論可以應用在任何一種雙模式型 (Two-Mode) 的資料上，以機率表示文件與索引之間的關係，引入了潛藏主題的概念，來表示文件與索引分別的主題分佈，以及兩者之間的相關程度，本論文會使用到此模型，故詳細模型內容詳述於下。

2.1.1 潛藏主題模型

它利用一組文件集 $\{d_j, j = 1, 2, ..., M\}$ 以及所有出現的用語 $\{t_i, i = 1, 2, ..., L\}$，並定義一組潛藏主題 $\{T_k, k = 1, 2, ..., K\}$，去分析用語及文件共同出現 (Term-Document Co-Occurrence) 之關係，概念如下：

1. 以機率 $P(d_j)$ 選取一文件 d_j。
2. 根據文件 d_j 涉及某個潛藏主題的機率 $P(T_k \mid d_j)$，選出一潛藏主題 T_k。
3. 在潛藏主題 T_k 中，產生出一用語 t_i 的機率爲 $P(t_i \mid T_k)$。

經由以上步驟進行機率之拆解，可以計算文件 d_j 產生一用語 t_i 的機率：

$$P(t_i, d_j) = P(d_j)P(t_i \mid d_j) = P(d_j) \sum_{k=1}^{K} P(t_i \mid T_k)P(T_k \mid d_j) \tag{2.1}$$

由此式中可看出，$P(t_i \mid d_j)$ 拆解成一組潛藏主題向量 $P(t_i \mid T_k)$ 的加總，其中每個不同的潛藏主題 T_k 具有不同的比重 $P(T_k \mid d_j)$。經由此拆解，造就了非單一向量之分群，而是將特定用語在文件中的機率以一混和模型 (Mixture Model) 的型式呈現。此模型建構於兩個基本假設上：

- 關於文件及用語 (t_i, d_j) 的出現機率爲相互獨立 (Independent)。

- 給定特定文件，在已知潛藏主題 T_k 下，產生用語 t_i 的機率爲獨立。

在此假設下，此模型可藉由最大期望值演算法 (Expectation Maximization; EM) 經由最大化目標函數去最佳化 [28] ，並且求取潛藏主題之模型。我們在下個章節會介紹最大期望值演算法。

爲了更容易理解潛藏主題模型所扮演的角色，我們可以將上面定義的潛藏主題 T_k 想像成新聞資料中的不同主題 (Topic)，例如政治、體育、財經等。假設希望估算用語「恐怖攻擊」出現在新聞「美國 911 事件...」的機率，雖然我們可以看出兩者相關性很高，但由於用語「恐怖攻擊」並沒有直接出現在此文件裡，我們無法利用共同出現關係計算它們的關係。潛藏主題模型如 (2.1) 的作法便是先定義好各個不同主題，像是政治、國際性新聞等，然後分別求出該文件屬於各主題的機率 $P(T_k \mid d_j)$ 及「恐怖攻擊」出現在各個主題裡的機率 $P(t_i \mid T_k)$ ，把這兩項機率相乘就可以得到 $P(t_i \mid d_j)$ 。這樣一來，即使用語「恐怖攻擊」並沒有直接出現在文件中，但因爲國際性新聞出現「恐怖攻擊」的機率很高，而此文件被分類到國際性新聞的機率也很高，所以我們能有效地估算用語「恐怖攻擊」出現在新聞「美國 911 事件...」的機率。

2.1.2 使用最大期望值演算法 (EM Algorithm) 求取潛藏主題觀念

我們希望對潛藏主題模型做最大可能性估測 (Maximum Likelihood Estimation) [30] ，藉由最大期望值演算法將目標函數最大化，此函數如下：

$$\sum_{j=1}^{M}\sum_{i=1}^{L} n(t_i, d_j)\log(P(t_i, d_j)) \tag{2.2}$$

式中 $n(t_i, d_j)$ 爲用語 t_i 在文件 d_j 中出現頻率 (Term Frequency) ，再用貝式定理

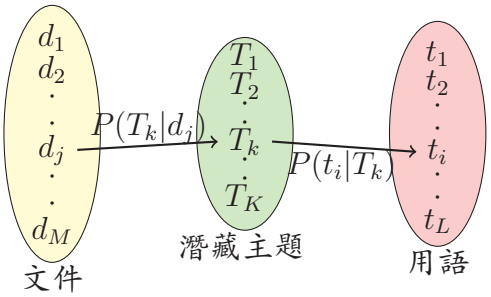


圖 2.1: 非對稱潛藏主題模型

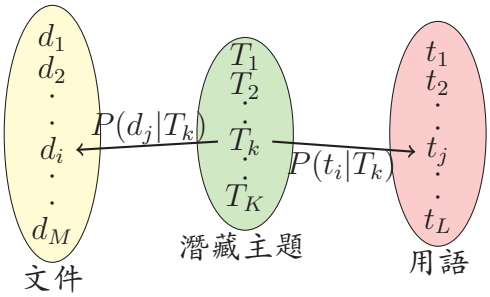


圖 2.2: 對稱潛藏主題模型

(Baye's Theorem) 改寫 $P(T_k|d_j)$ ，可得

$$P(T_k \mid d_j) = \frac{P(d_j \mid T_k)P(T_k)}{P(d_j)} \tag{2.3}$$

再將 (2.1) 和 (2.2) 代入 (2.3) ，得到

$$P(t_i, d_j) = \sum_{k=1}^{K} P(T_k)P(t_i \mid T_k)P(d_j \mid T_k) \tag{2.4}$$

(2.4) 爲一完全等價且對稱之模型。上述觀念可由圖 2.1 和 2.2 分別來表示 (2.1) 及 (2.4) 的意義。在圖 2.1中可看出此爲一非對稱結構，在潛藏主題 T_k 所展開的空間下將文件 d_j 展開；並經由貝式定理轉換後，文件 d_j 和用語 t_i 都獨立地用潛藏主題 T_k 展開，由圖 2.2 可看出這爲一對稱於潛藏主題 T_k 的結構。

估測潛藏主題模型的最大期望值法可分爲兩大步驟，分述於下。

期望步驟 (Expectation Step)

根據演算法過程中現有參數之預測，計算出潛藏變數之期望值，將最大化步驟更新後的潛藏主題模型，用來計算某一文件 d_j 的某一用語 t_i 對應潛藏主題 T_k 的機

率，更新式子如下：

$$P(T_k \mid t_i, d_j) = \frac{P(T_k)P(d_j \mid T_k)P(t_i \mid T_k)}{\sum_{T_k} P(T_k)P(d_j \mid T_k)P(t_i \mid T_k)} \tag{2.5}$$

最大化步驟 (Maximization Step)

根據期望步驟中計算之結果，於期望值最大化過程中估計新的參數值，其所要估計的參數分別如下所示：

$$P(t_i \mid T_k) \propto \sum_{d_j} n(t_i, d_j)P(T_k \mid t_i, d_j) \tag{2.6}$$

$$P(d_j \mid T_k) \propto \sum_{t_i} n(t_i, d_j)P(T_k \mid t_i, d_j) \tag{2.7}$$

$$P(T_k) \propto \sum_{d_j} \sum_{t_i} n(t_i, d_j)P(T_k \mid t_i, d_j) \tag{2.8}$$

藉由不斷重複此兩個步驟進行最大期望值法，可得到區域最佳化 (Local Optimal) 的機率式潛藏語意分析模型。若給定一文件，我們可以根據每個文件中的用語 t_i 在訓練語料中的分布來尋找其潛藏主題，經由模型轉換後，可得該文件的潛藏主題機率。

2.1.3 機率式潛藏語意分析模型與傳統潛藏語意分析模型之比較

傳統潛藏語意分析是透過將文件-詞的矩陣進行奇異值分解，達到維度上的縮減，因此對於相同的文件及詞集，每次分析出來的結果都是相同的。而機率式潛藏語意分析在使用最大期望值演算法時，是以隨機的狀態做爲起始機率的選取，來進行區域最佳化，故每次最佳化的結果都會依起始的不同而有所不同，雖然有可能會因爲初始狀態的不好而導致分析結果有落差，但是相較於傳統的潛藏語意分析

不會有隨機起始的問題，反而可以經由結合多個潛藏主題模型而得到多變化的語意分析。同時，機率式潛藏語意分析模型沒有主題之間須互相獨立之假設，較貼近實際情形。

2.1.4 基於機率式潛藏語意分析模型之特徵參數

本論文主要利用機率式潛藏語意分析模型提供的機率分布，計算以下三類不同的參數，進而利用這些參數來進行用語重要性之估算 [31] 。以下皆分別對潛藏主題機率 (Latent Topic Probability; LTP) 、潛藏主題重要性 (Latent Topic Significance; LTS) ，以及潛藏主題亂度 (Latent Topic Entropy; LTE) 三者做詳細介紹。

潛藏主題機率 (Latent Topic Probability; LTP)

我們可以藉由最終模型評估一個用語產生一潛藏主題的機率值如下：

$$\mathrm{LTP}(T_k \mid t_i) = \frac{P(t_i \mid T_k)P(T_k)}{P(t_i)} \tag{2.9}$$

此式 $P(t_i)$ 可藉由語料中獲得，而 $P(T_k)$ 可以根據 $P(T_k \mid d_j)$ 去估計。

潛藏主題重要性 (Latent Topic Significance; LTS)

一用語 t_i 對於一潛藏主題 T_k 之重要性被定義爲 [32]

$$\mathrm{LTS}_{t_i}(T_k) = \frac{1}{1 + \exp(-\lambda_{\mathrm{LTS}} C(t_i, T_k))} \tag{2.10}$$

此式爲一 $C(t_i, T_k)$ 定義的雙彎曲函數 (Sigmoid Function)，λ_{LTS} 爲一比例因子

(Scaling Factor) ，$C(t_i, T_k)$ 定義於下式：

$$C(t_i, T_k) = \sum_{d_j \in D} n(t_i, d_j) P(T_k \mid d_j) - \sum_{d_j \in D} n(t_i, d_j)[1 - P(T_k \mid d_j)] \tag{2.11}$$

$n(t_i, d_j)$ 爲用語 t_i 在文件 d_j 出現次數，$P(T_k \mid d_j)$ 爲潛藏語意分析模型的結果。在 (2.11) 的被減數部分，用語 t_i 在每個文件 d_j 中的頻率被 d_j 所隱含的特定主題 T_k 機率所加權， 然後加總在訓練語料中的所有文件 d_j 所貢獻的值。所以此部分爲訓練語料中，潛藏主題模型所估計 t_i 在主題 T_k 中的總頻率。而減數部分非常類似，只是將潛藏主題部分換成 T_k 以外的其餘主題，故將 $P(T_k \mid d_j)$ 取代爲 $[1 - P(T_k \mid d_j)]$ 。

潛藏主題亂度 (Latent Topic Entropy; LTE)

對於一用語 t_i 的潛藏主題亂度 $\mathrm{LTE}(t_i)$ 可以根據 (2.9) 的 $\mathrm{LTP}(T_k|t_i)$ 分布來估測 [33] 。

$$\mathrm{LTE}(t_i) = -\sum_{k=1}^{K} \mathrm{LTP}(T_k \mid t_i) \log \mathrm{LTP}(T_k \mid t_i). \tag{2.12}$$

較低的 $\mathrm{LTE}(t_i)$ 代表 $\mathrm{LTP}(T_k \mid t_i)$ 的分布較集中於少量的潛藏主題中，而 t_i 帶了較多的主題資訊，較具重要性。

2.2 機器學習 (Machine Learning)

機器學習分爲非督導式 (Unsupervised) 及督導式 (Supervised) 兩種。督導式學習是經由部分訓練集的正確答案標記來訓練分類器，而非督導式學習不需要正確答案之標記。近年來，許多督導式學習之發展已漸趨成熟，許多模型也被使用在關鍵用語擷取上，此論文主要使用兩個督導式學習之方法 – 調適性推昇法 (Adaptive

Boosting; Adaboost) 及類神經網路 (Artificial Neural Network) ，而詳細概念與訓練過程於下方做詳細的介紹。

2.2.1 調適性推昇法 (Adaptive Boosting; AdaBoost)

調適性推昇法 (Adaptive Boosting; AdaBoost) [26] 是一種把較弱的數個基本假設模型 (Hypothesis) 推昇 (Boosting) 爲較強模型之演算法。它根據前一個模型的表現決定它的權重以及下一個模型的重取樣，故稱爲調適性 (Adaptive) 的模型。

調適性推昇法先從單純的二類分類 (Two Class Classification) 著手，也就是俗稱的 Decision Stump 。假設訓練集 (Training Set) 爲 $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$ ，其中 $\mathbf{x}_n$ 爲一觀察樣本之 d 維特徵向量，標記答案 $y_n = 0$ 或 1 。此學習演算法需要一個基礎學習模型 h 及欲結合的個數 T ，詳細演算法列於 Algorithm 1 。

Algorithm 1 調適性推昇法 (Adaptive Boosting; AdaBoost)

Input: 訓練集 $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$

Output: 假設模型 $H(\mathbf{x})$

1: 對所有 n 都設定 $u_i = \frac{1}{N}$;

2: **for** $l = 1$ to T **do**

3: 基本假設 $h_l = \arg\min_h \sum_{n=1}^{N} u_n \cdot I[y_n \neq h(\mathbf{x}_n)]$;

4: 計算加權錯誤率 $\epsilon_l = \sum_{n=1}^{N} \frac{u_n}{\sum u_n} I[y_n \neq h(\mathbf{x}_n)]$;

5: 計算信心權重 $\alpha_l = \frac{1}{2} \ln \frac{1-\epsilon_l}{\epsilon_l}$;

6: 強調被 h_l 分類錯誤者 $u_n \leftarrow u_n \cdot \exp(-\alpha_l y_n h_l(\mathbf{x}_n))$;

7: **end for**

8: 結合後的假設模型 $H(\mathbf{x}) = sign(\sum_{l=1}^{T} \alpha_l h_l(\mathbf{x}))$;

由演算法可發現，每次迴圈中，藉由每次調整訓練集每筆資料的權重 u_n ，基礎模型會更專注在上一個模型分類錯誤者 ($y_n h_l(\mathbf{x}_n) < 0$ ，其 u_n 增加) ，同時降低分類正確者 ($y_n h_l(\mathbf{x}_n) > 0$ ，其 u_n 減少) 之重要性。而 α_l 的選取會使得權重經過更新後，上一個學習模型的預期正確率變爲 0.5 ，相當於完全和其互補的重取樣訓練集。可由此看出，當加權錯誤率 ϵ_l 越低，表示所學之假設 h_l 越正確，其權重 α_l 就越高。

調適性推昇法演算法之步驟，可以證明此等價於逐次去最小化 (Minimize) 一個目標函數 (Objective Function) [34] ，

$$\sum_{n=1}^{N} \exp(-y_n \cdot \sum_{l=1}^{T} \alpha_l h_l(\mathbf{x}_n)) = \sum_{n=1}^{N} \exp(-y_n h(\mathbf{x}_n)) \tag{2.13}$$

若加權錯誤率 $\epsilon_l < 0.5$ 一直成立，此目標函數會呈指數下降。若 $(\mathbf{x}_n, y_n)$ 分類錯誤，即 $y_n h(\mathbf{x}_n) < 0$ ，有 $\exp(-y_n h(\mathbf{x}_n)) > 1$ ，故

$$\sum_{n=1}^{N} \exp(-y_n h(\mathbf{x}_n)) \geq \sum_{y_n \neq h(\mathbf{x}_n)} exp(-y_n h(\mathbf{x}_n)) \geq \sum_{y_n \neq h(\mathbf{x}_n)} 1 \tag{2.14}$$

即目標函數式訓練集錯誤個數的上限，因此目標函數呈指數下降，會保證到訓練集錯誤率在 $O(\log N)$ 時間內降至 0 。

雖然此演算法只保證下降訓練集錯誤率，但文獻顯示此亦可推廣到測試集 (Testing Set) ，原因爲調適性推昇法亦有最大間距 (Maximal Margin) 之效果。而此演算法亦被推廣至多類分類問題 (Multi-Class Classification) ，如 AdaBoost.MH 及 AdaBoost.M2 [34] 。

此外，調適性推昇法沒有隨機性，所以模型呈現較小震盪，所形成之綜合模型也更快穩定下來。在訓練集充足且正確之情況下，調適性推昇法會有很好的效

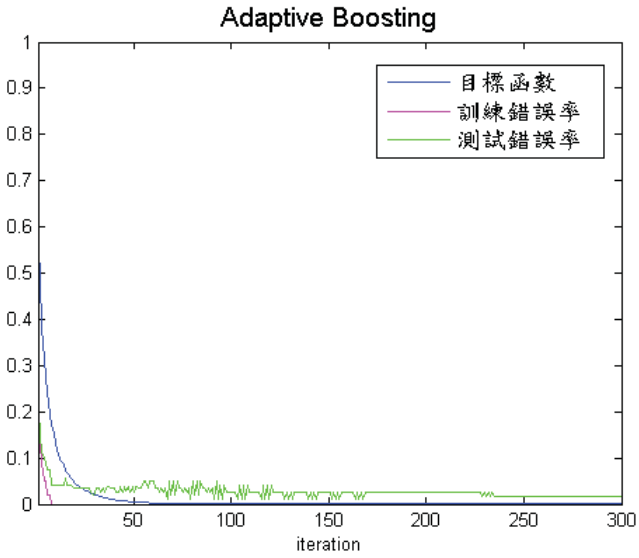


圖 2.3: 調適性推昇法對於訓練及測試集的錯誤率變化情形

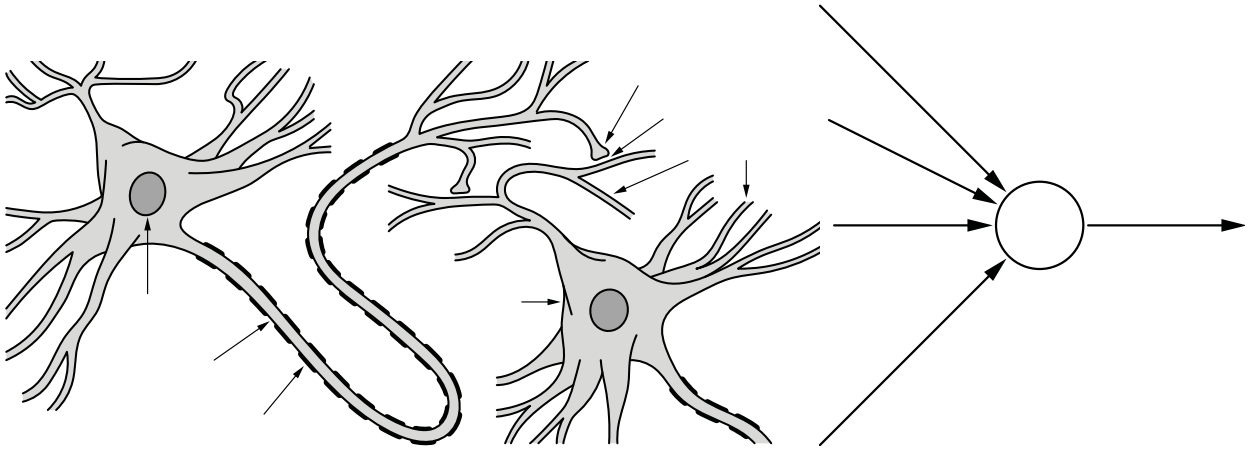
果。此情形可由圖 2.3 看出。

2.2.2 類神經網路 (Artificial Neural Network)

早期機器學習 (Machine Learning) 的研究者們受到神經科學 (Neuro Science) 的啓發，試著建立一個模型來模仿神經元突觸的行爲。如圖 2.4a ，一個神經元的作用就是將它接受到的訊號，經過某種轉換後，再傳給下一個神經元。 這跟數學上的函數其實是一樣的，因此我們只要想辦法找出一個函數符合眞實神經元的情況也許就可以模擬其行爲 [27] 。但實際上我們並不知道那是什麼樣的函數，只好假設它是輸入向量線性組合的某個轉換，寫成數學式子如下：

$$F(\mathbf{x}) = \varphi\left(\sum_i w_i x_i\right) \tag{2.15}$$

其中 $\mathbf{x}$ 是輸入訊號，φ 是某個數學轉換 (Transformation) ，而 w_i 是線性組合的加權 (Weight) 。如果將上面的概念畫成圖，就像圖 2.4b 一般。然而這樣的結構明顯太簡單，無法提供我們強而有力的模型，但我們可以將這些小的神經元串接起來



(a) 人體的神經元　　(b) 數學的神經元

圖 2.4: 神經元圖示

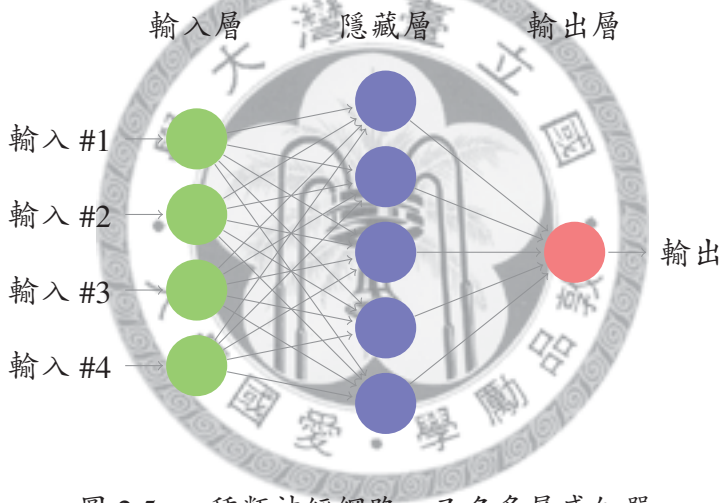


圖 2.5: 一種類神經網路，又名多層感知器

來得到一個強力的模型，而這樣串接起來的模型我們稱作類神經網路 (Artificial Neural Network) 。但如果串接的結構太複雜，又超出我們可以計算的能力，所以我們限制其串接的結構，規定串接的時候只能串接成一個無迴路 (Acyclic) 的類神經網路，叫做多層感知器 (Multi-Layered Perceptron; MLP) ，又名前向遞推類神經網路 (Feed-Forward Artificial Neural Network) ，爲類神經網路的一個子集。其結構如圖 2.5 。經常的用途是對付分類 (Classification) 問題、迴歸 (Regression) 問題，或將高維的特徵向量轉換到較低維度的向量。

一般來講，如果是在解二元分類問題 (Binary Classification) 或是單元迴歸問題

(Univariate Regression) 的時候，我們設定輸出的向量就是一維的實數向量。如果是分類問題就取一個閾值 (Threshold) ，高過閾值就是某一個分類，低於閾值就是另一個分類。如果是迴歸問題就直接使用這個實數當作我們的預測值。

當解的問題改成多元分類問題或多元迴歸問題時，我們可以增加輸出向量的維數爲要分類的類別數來解決問題，而用來估測模型參數的演算法並不會需要更動。

以下，我們用數學的語言來描述此種類神經網路。

數學定義

爲了方便說明模型的架構，這邊假設我們希望建立模型的函數是 $f : \mathbb{R}^d \to \mathbb{R}$ ，即輸入的向量爲 d 維，輸出的向量是 1 維。而我們建立模型的函數爲 g ，我們希望 g 跟 f 越相近越好。

則我們定義我們的前向遞推類神經網路如下：網路由 L 層串接起來的神經元 (Neuron) 所組成，第 l $(1 \leq l \leq L)$ 層總共有 $d^{(l)}$ 個神經元，將 $d^{(l-1)}$ 個輸入接成 $d^{(l)}$ 個輸出，這樣的結構表示第 $(l-1)$ 層的輸出就是第 l 層的輸入。若如果是有跳層情況的網路，也可以重新寫成第 l 層的輸入從第 $(l-1)$ 層輸出來的形式，所以定義由無迴路性質刻畫是正確的。由以上定義，我們知道整個系統的輸入維度爲 $d^{(0)} = d$ ，而整個系統的輸出維度爲 $d^{(L)} = 1$ 。

接著，我們再定義第 l 層的輸入爲 $x_i^{(l-1)}(1 \leq i \leq d^{(l-1)})$ 、輸出爲 $x_j^{(l)}(1 \leq j \leq d^{(l)})$ 。而第 l 層的加法權重爲 $w_{ij}^{(l)}(1 \leq l \leq L, 0 \leq i \leq d^{(l-1)}, 1 \leq j \leq d^{(l)})$則第 l 層中的第 j 個神經元的轉換函數可定義爲

$$x_j^{(l)} = \varphi\left(\sum_{i=0}^{d^{(l-1)}} w_{ij}^{(l)} x_i^{(l-1)}\right) \tag{2.16}$$

其中 $\varphi(s) = \tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$ 。

爲了後面演算法的推導方便，我們定義一個輔助變數 $s_j^{(l)}$ 將神經元函數中線性的部份跟非線性的部份分開來。

$$s_j^{(l)} = \sum_{i=0}^{d^{(l-1)}} w_{ij}^{(l)} x_i^{(l-1)} \tag{2.17}$$

$$x_j^{(l)} = \varphi(s_j^{(l)}) \tag{2.18}$$

假設模型的參數爲已知，則使用類神經網路來計算函數 $g(\mathbf{x})$ 的預測值便可以如下操作：將輸入向量 $\mathbf{x}$ 的每一維分別擺到 $x_1^{(0)}, \ldots x_{d^{(0)}}^{(0)}$ ，再從第一層開始一一按照上面的神經元函數計算這一層的輸出，直到第 L 層爲止。則最後我們便將 $x_1^{(L)}$ 當作 $g(\mathbf{x})$ 輸出的值。

模型參數估測

有了前一節定義的類神經網路之結構，這一節將探討如何利用訓練集 $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ ，來估計此模型的參數。

爲了要衡量我們得到的參數究竟讓模型是否更貼近我們希望建立模型的函數，我們需要一個錯誤衡量函數 (Error Function) 。一般來講，在訓練類神經網路的錯誤衡量函數皆爲

$$E_n(w) = (g(\mathbf{x}_n) - y_n)^2 \tag{2.19}$$

用二次項的好處就是可以微分，方便我們使用做最佳化 (Optimization) 的時候使用梯度下降法 (Gradient Decent) 。

在這邊我們使用一種在實做上比較容易的一種梯度下降法 – 隨機梯度下降法 (Stochastic Gradient Decent) 來做最佳化，因此我們需要計算 $\frac{\partial E_n}{\partial w_{ij}^{(l)}} (0 \leq i \leq$

$d^{(l-1)}, 1 \leq j \leq d^{(l)}, 1 \leq l \leq L)$ 。一旦我們有了以上梯度，則我們可以應用隨機梯度下降法來修正權重。

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial E_n}{\partial w_{ij}^{(l)}} \tag{2.20}$$

其中 η 是學習速率，隨機梯度下降法只要執行足夠多次更新，就可以得到一組滿意的參數。此演算法列於 Algorithm 2。

Algorithm 2 隨機梯度下降演算法 (Stochastic Gradient Decent Algorithm)

1: 視問題將 $w^{(0)}$ 初始化爲 0 或一組亂數;

2: **for** $t = 1$ to T **do**

3: $\quad w^{(t)} \leftarrow w^{(t-1)} - \eta \nabla E(w^{(t-1)})$;

4: **end for**

經由多次隨機梯度下降法的更新，權重能夠漸漸被修正以最佳化。故此時主要的問題是，是否有方法讓我們可以快速地算出 $\frac{\partial E_n}{\partial w_{ij}^{(l)}}$ 。這便是我們下面要講的後向推進演算法 (Backpropagation Algorithm) 的目的。

假設我們已經執行了一遍前向遞推來計算出對應訓練範例的輸出，在這過程中，注意到輔助變數 $x_j^{(l)}$ 跟 $s_j^{(l)}$ 已經被計算出來。則由於 $w_{ij}^{(l)}$ 只透過 $s_j^{(l)}$ 來影響輸出，根據鏈鎖法則 (Chain Rule) ，我們有

$$\frac{\partial E_n}{\partial w_{ij}^{(l)}} = \frac{\partial E_n}{\partial s_j^{(l)}} \cdot \frac{\partial s_j^{(l)}}{\partial w_{ij}^{(l)}} \tag{2.21}$$

其中 $\frac{\partial s_j^{(l)}}{\partial w_{ij}^{(l)}} = x_i^{(l-1)}$ 是已經計算出來的值。所以只剩下如何計算 $\frac{\partial E_n}{\partial s_j^{(l)}}$ 的問題。

先定義 $\delta_j^{(l)} = \frac{\partial E_n}{\partial s_j^{(l)}}$ 。則在最後一層 $l = L$ 的時候，我們有 $E_n = (x_1^{(L)} - y)^2$ ，

因此我們能計算 $\delta_1^{(L)}$

$$\delta_1^{(L)} = \frac{\partial E_n}{\partial s_j^{(l)}} = \frac{\partial E_n}{\partial x_1^{(l)}} \cdot \frac{\partial x_1^{(L)}}{\partial s_1^{(l)}} \tag{2.22}$$

其中 $\frac{\partial E_n}{\partial x_1^{(l)}} = 2(x_1^{(L)} - y)$ 、$\frac{\partial x_1^{(L)}}{\partial s_1^{(l)}} = \varphi'(s_1^{(L)})$ 。因此我們有了遞迴的初始條件。我們可以利用遞迴來算第 $l-1$ 的 $\delta_j^{(l-1)}$

$$\begin{aligned} \delta_i^{(l-1)} &= \frac{\partial E_n}{\partial s_i^{(l-1)}} \\ &= \sum_{j=1}^{d^{(l)}} \frac{\partial E_n}{\partial s_j^{(l)}} \cdot \frac{\partial s_j^{(l)}}{\partial x_i^{(l-1)}} \cdot \frac{\partial x_i^{(l-1)}}{\partial s_i^{(l-1)}} \\ &= \sum_{j=1}^{d^{(l)}} \delta_j^{(l)} \cdot w_{ij}^{(l)} \cdot \varphi'(s_i^{(l-1)}) \end{aligned} \tag{2.23}$$

其中 $\varphi'(s) = 1 - \varphi^2(s)$、$\varphi(s_i^{(l)}) = x_i^{(l)}$ 因此 $\delta_i^{(l-1)}$ 可以遞迴地表示成如下：

$$\delta_i^{(l-1)} = \left(1 - (x_i^{(l-1)})^2\right) \sum_{j=1}^{d^{(l)}} w_{ij}^{(l)} \delta_j^{(l)} \tag{2.24}$$

以上的式子中的變數都在前向遞推時能算出來，因此我們藉由後向推進演算法就可以算出所有的 δ。有了 δ，我們就可以算出錯誤對權重的梯度。因此我們可以做隨機梯度下降法來最佳化。

藉由上述方法，可以訓練出一類神經網路，其參數經由最佳化後符合訓練集，故此類神經網路則爲一分類器，測試集中的樣本可以透過此網路並決定其所屬之分類。

2.3 本章總結

本章節介紹了本論文中所使用的基礎數學工具，包含了非督導式的機率式潛藏語

意分析模型的訓練，以此模型進行潛藏主題的機率的估算。同時也詳細介紹了兩個督導式機器學習方法之原理及分類器的訓練流程，此兩項督導式學習爲調適性推昇法及類神經網路。在後面之章節，我們會介紹如何利用這些工具合理並有效地進行語音文件的關鍵用語擷取及自動摘要。

第 三 章　語音文件之關鍵用語擷取

Key Term Extraction from Spoken Documents

3.1　簡介

此章節提出一些方法在語音文件中處理關鍵用語擷取，而以課程語料作爲實驗。由於終身學習的概念普遍化，網路上提供了非常大量的課程錄音，使用者期望有效率地瀏覽這些大量的課程錄音，以及從中搜尋特定的資訊，由課程錄音中擷取關鍵用語成爲一重要研究。

許多研究都致力於由文字中擷取關鍵用語 [18,35,36] ，但較少研究針對語音文件去做特別的處理 [4,5] ，雖然一些自動摘要的研究使用了韻律等聲音訊號的特徵 [6,7,32,33] ，但是未有研究使用此特徵去幫助關鍵用語之擷取。

此篇論文將針對語音課程錄音的語料進行關鍵用語的擷取，語料包含了聲音訊號、大字彙辨識後的文本 (ASR Transcriptions) 、以及講者使用的投影片。此語料中的投影片皆爲英文，而錄音內容主要使用中文，途中穿插英文 (較口語的詞或專業詞彙) 。此類中英混合 (Code-Mixing) 的方式在台灣非常普遍，辨識結果及關鍵用語也包含中文及英文。

在進入各項細節之前，我們先定義一下使用的單位。所謂的用語 (Term) 可能爲一個詞 (Word) 或一個以上的詞所形成的片語 (Phrase) 。詞爲語言中具有意義的最小單位，上句話中的「定義」和「單位」均爲中文的一個詞，而英文的「Term」和「Word」亦皆屬一個詞 (Word) 。在此研究中，我們將關鍵用語分成兩類型：關鍵片語 (Key Phrase) 及關鍵詞 (Keyword)。關鍵片語像是「隱藏 馬可夫模型 (Hidden Markov Model) 」和「資訊 理論 (Information Theory) 」，但單詞「隱

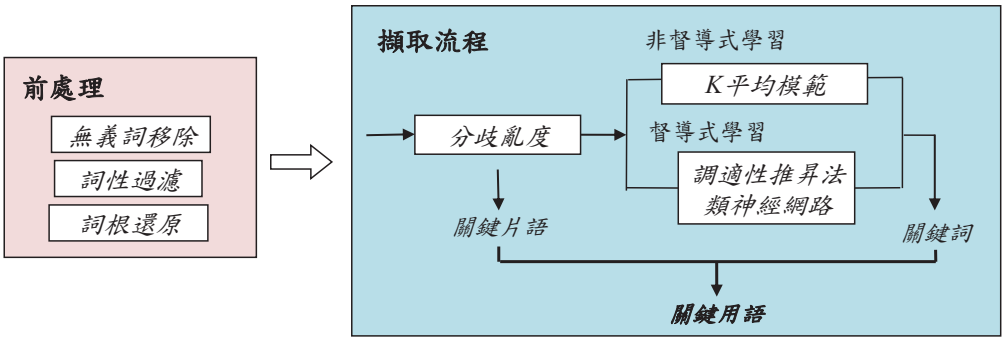


圖 3.1: 擷取關鍵用語的架構及流程

藏 (Hidden) 」或是「理論 (Theory) 」皆非關鍵用語。而也有單詞爲關鍵用語，在此稱做關鍵詞，如「亂度 (Entropy) 」。在此論文中，我們提出分歧亂度的方法先抽取出關鍵片語，接著使用三種不同類型之特徵訓練分類器 (Classifier) 來擷取關鍵詞。

之前已有初步關鍵用語擷取方法，實作在國立台灣大學 (National Taiwan University; NTU) 的課程錄音系統中，稱爲台大虛擬教師 (NTU Virtual Instructor) [37] ，此論文提出之方法更爲精緻，結果也更好。本章將詳細說明如何考慮課程錄音的特徵並抽取包含中英文的關鍵用語。

3.2 架構與流程 (Framework)

在關鍵用語的擷取方面，主要分爲兩個流程，如圖 3.1 。其中第一部分爲前處理 (Preprocessing) ，對於所有的用語做初步的過濾，留下的用語稱爲候選關鍵用語 (Candidate Key Term) ，此部分的詳細內容於第 3.3 節。接著第二個部分爲依序擷取關鍵片語及關鍵詞，詳述於後面章節。

圖 3.2: 高頻詞 (包含許多無義詞)

3.3 前處理 (Preprocessing)

對於所有抽取出來的用語，先經過三個步驟來過濾，留下的候選關鍵用語才進行後續的抽取流程。

3.3.1 無義詞移除 (Stop Word Removal)

無義詞 (Stop Word) 是指一些出現頻繁，本身並無意義的詞，像是冠詞 (「a」、「an」、「the」、「一個」)、介繫詞 (「of」、「in」、「for」、「的」、「是」) 和代名詞 (「it」、「their」、「his」、「它」、「這」) 等，因爲無義詞不具有特定意義，不可能成爲關鍵用語，因此我們將文件中出現的無義詞移除。我們利用的無義詞清單來自維基百科 [38]，其中包含 319 個無義詞。圖 3.2 爲英文中出現的高頻詞，字體越大代表出現頻率越高，可以發現其中有許多詞不具有實際意義，即爲無義詞，故移除無義詞是一重要的前處理。

3.3.2 詞性過濾 (PoS Filtering)

有鑑於關鍵用語通常是名詞、動詞或形容詞，我們只把這些詞性 (Part-of-Speech; PoS) 的詞保留，其餘皆過濾掉。此部分我們利用使用工具TreeTagger [39] 完成這項前處理。

3.3.3 詞根還原 (Word Stemming)

詞根還原 (Word Stemming) 是指將任何不同型態的詞轉回原形。考慮到在語音文件中的英文用語，一個詞可能以不同型態出現，像是複數型態 (如：「model」和「models」)、過去式 (如：「train」和「trained」) 等，無論其以何種型態出現，其意義皆爲同一詞，故在計算出現次數或重要性時，應該視其爲相同的詞來考慮。此步驟可以避免將不同型態的同樣的詞都擷取出來 (像「hidden Markov model」和「hidden Markov models」皆爲關鍵用語而被擷取出來)。我們對每個詞彙詞根原形化，將不同型態的詞變回原形。

3.4 藉由分歧亂度 (Branching Entropy) 抽取關鍵片語 (Key Phrase)

此章節的目的是抽取關鍵片語 (Key Phrase)。經過前處理後，保留下來的候選關鍵用語，再進行關鍵片語的擷取。像是「隱藏馬克夫模型 (Hidden Markov Model)」或「資訊理論 (Information Theory)」，也就是兩個以上的詞時常同時出現在語料中，頻繁的次數比許多其他的詞更多，有較高的機率爲關鍵片語。我們提出藉由一稱爲分歧亂度 (Branching Entropy) 的參數來抽取之，此方法使用一種符號串列的資料結構，稱爲後綴樹 (Patricia Tree; PAT-Tree) [9,40] 來實作，圖 3.3 爲一簡單的例子。此爲一專門儲存字串的樹狀結構，而在此處我們使用「詞」當作符號的單位。

在此樹狀結構下，每條路徑都爲半無限的字串 (Semi-Infinite String; Sistring)，此是由包含投影片和文本的語料中所含的句子的部分字串所形成的。半無限字串即爲在語料中，從句子的任意位置開始，至句子尾端的詞序列 (Word

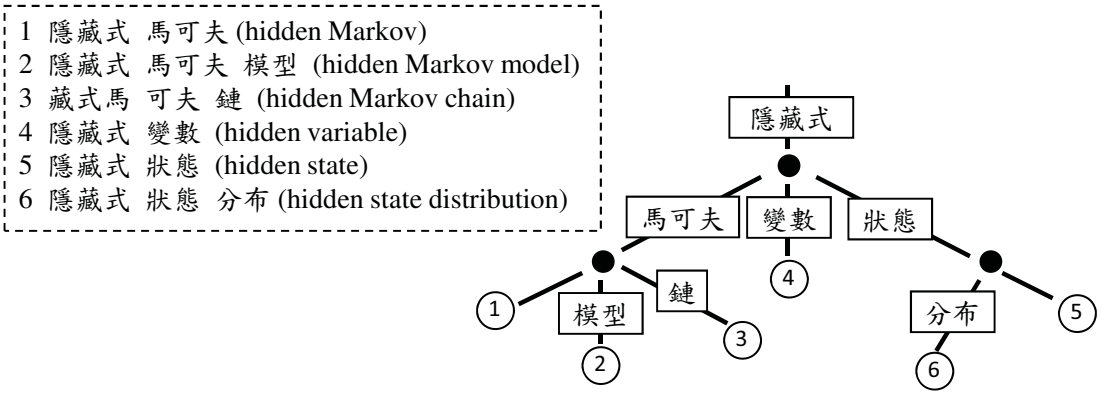


圖 3.3: 一個後綴樹上以詞爲單位的簡單例子

隱藏式 馬可夫 模型 是 關鍵 用語 . . . .
馬可夫 模型 是 關鍵 用語 . . . . . .
模型 是 關鍵 用語 . . . . . .
是 關鍵 用語 . . . . . .
關鍵 用語 . . . . . .
用語 . . . . . .

圖 3.4: 「隱藏式馬可夫模型是關鍵用語」的半無限字串例子

Sequence) [41] 。圖 3.4 爲幾個半無限字串的例子，我們將每個語料中 (例：「隱藏式 馬可夫 模型...」) 句子分成其半無限字串 (「隱藏式 馬可夫 模型...」、「馬可夫 模型...」、「模型...」等)，接著使用這些半無限字串去建造此後綴樹。然而，一些研究也利用後綴樹爲基礎來抽取中文的關鍵字詞，這些研究通常使用相互訊息 (Mutural Information) 來當成抽取的標準 [10,11,42]；但是在此論文中，我們提出另一個新的方法 – 分岐亂度，更有效地抽取關鍵片語。

超過兩個詞所組成的詞組 X 的右分歧亂度 (Right Branching Entropy) 被定義爲

$$p(x_i) = \frac{f_{x_i}}{f_X} \tag{3.1}$$

$$H_r(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i), \tag{3.2}$$

此處 X 爲一目標的詞序列 (如：「隱藏式 馬可夫」)，而 x_i 爲 X 在樹狀結構中的孩子 (如：「隱藏式 馬可夫 模型」及「隱藏式 馬可夫 鏈」爲「隱藏式 馬可夫」之孩子)，f_X 及 f_{x_i} 分別是 X 及 x_i 的詞頻，故 $p(x_i)$ 爲給定 X 之下，它的孩子爲x_i的機率，而 $H_r(X)$ 則稱做 X 的右分岐亂度 (Right Branching Entropy)，其中 n 則爲 X 在樹狀結構中所具有的不同孩子 x_i 的數目。

當一詞序列「隱藏式 馬可夫 模型」爲一關鍵片語時，不只此詞序列的頻率會較高，大部分「隱藏式 馬可夫」後面多半跟隨著「模型」，故「隱藏式 馬可夫」具有較低的 $H_r(X)$。而相反地，「隱藏式 馬可夫 模型」後方即可能銜接許多不同的詞，像是「的」、「是」...等等，故「隱藏式 馬可夫 模型」具有較高的 $H_r(X)$。故我們可以使用右分岐亂度 $H_r(X)$ 來定義關鍵片語的右方邊界，像上述例子中，「模型」後方比「馬可夫」後方更有可能爲一關鍵片語的右方邊界。我們藉由爲 $H_r(X)$ 設定一閾值 (Threshold) 來定義此處是否應爲右邊界。

同理，我們可以建造一反向後綴樹 (Reverse PAT-Tree) [40]，其中包含了許多反向的詞序列 (例：「模型 馬可夫 隱藏式」...)，並對每個詞序列 $\overline{X}$ 定義左分岐亂度 (Left Branching Entropy) $H_l(\overline{X})$。此值也用於定義關鍵片語的左方邊界，像是「馬可夫」前方比「隱藏式」前方有較高機率爲左邊界，因爲「隱藏式」前方可能會有很多不同的詞，而「馬可夫」前方通常爲「隱藏式」。

上面的流程需要在整個語料所建立的樹狀結構中快速搜尋並對每個詞序列計算左右分岐亂度，當語料較大時，這是一個較耗費時間的過程，但是後綴樹的結構可以讓此流程非常有效率，故實驗中，此流程並未耗費過多時間。

我們計算所有 X 的 $H_r(X)$ 及所有 $\overline{X}$ 的 $H_l(\overline{X})$，並抽取平均 $H_r(X)$ 及平均 $H_l(\overline{X})$ 來當成閾值，最後抽取 $H_r(X)$ 及 $H_l(\overline{X})$ 皆高於平均值的詞序列 X 當做關鍵片語。

3.5 抽取特徵以訓練分類器 (Classifier) 來抽取關鍵詞 (Keyword)

此章節爲關鍵詞之分類器特徵抽取。對於其餘的候選關鍵用語 (於關鍵片語擷取後)，抽取其多樣特徵，並藉由這些特徵來訓練分類關鍵詞的分類器。在此處使用了三種不同的特徵，分別爲韻律特徵 (Prosodic Feature) 、詞彙特徵 (Lexical Feature) 、語意特徵 (Semantic Feature) 。其詳細抽取方式詳述於下。

3.5.1 韻律特徵 (Prosodic Feature)

很多研究顯示韻律特徵對於抽取摘要有很大的幫助，因爲此類特徵可以幫助定義一個句子的重要性，講者可能會藉由不同的韻律 (如發音較大聲、音調較高、或較慢的敘述方式來強調較重要的部分) 。研究顯示，在研討會語料的韻律特徵會因爲講者的差異而較沒有明確的衡量方式 [6] 。由於課程錄音通常是出自於同一個講者，故韻律特徵在此處也許會較有幫助。故我們可以假設講者時常會使用韻律的變異、音高和音量上的改變、以及語速來強調較重要的資訊。用此概念來標記語音中重要的內容 [7]。

在此處，我們使用音素爲單位的方式對結果進行強迫性對齊 (Forced Alignment) [3] ，韻律特徵即可由此處抽取。我們只將每個用語在錄音中第一次出現的韻律特徵抽取出來，因爲我們假設講者通常在第一次提到該用語時才會進行明顯的強調。最後總共抽取出十二個數值來表達韻律特徵的部分。

音長特徵 (Duration Feature)

由於不同音素單位 (Phonetic Unit) 可能有不同的音長 (Duration) ，所以我們先對所

有的音素單位在此語料中計算一平均音長。接著對於每個用語，將其中所有出現的音素單位之音長計算出來，並對之平均值做正規化 (Normalization) 。最後抽取每個用語中，正規後的音素音長數值最大、最小、平均、及範圍四值，以此四值作爲此用語的音長特徵。

音高特徵 (Pitch Feature)

我們使用 ESPS [43] 從語音資訊中抽取每個用語中的音框 (Frame) 的基頻 (F0) 當作特徵值。爲了避免音高輪廓 (Pitch Contour) 中可能存在的不連續性，包括偵測到半頻、倍頻等情形，我們先對音高的特徵值做處理，將不連續處平滑化，此方法稱爲音高輪廓精緻化 [44] 。

1. 消除單獨一音框的半頻或倍頻

   首先對每個相鄰之音框，檢查它們的基頻相差是否在 12% 之內。若在 12% 內則加它們之基頻值加入音高輪廓中；若相差過多則刪去時間上較晚之音框的基頻值。經由這些步驟，會產生一些連續的音高輪廓片段。

2. 消除連續數個音框的半頻或倍頻

   檢查上一步驟產生之連續音高輪廓片段的長度是否大於 5 ，若小於 5 則刪除此段音高輪廓。

3. 線性內插補齊音高輪廓缺口

   經由上述兩步驟的追蹤與平滑化後，利用線性內插，將被刪除之音高輪廓缺口補齊。若被刪除的缺口出現在音高輪廓之首，直接複製剩下的音高輪廓中最早出現之音高數值；反之，若缺口出現在尾端，我們也直接複製剩下音高輪廓中最晚出現之音高數值。

若有單獨一個音框被抽取出半頻或是倍頻，會造成音高輪廓有特別高或特別低之不連續點，第一步即在消除這些錯誤。但若有連續數個音框被抽取出半頻或是倍頻，則無法被第一步驟偵查到。因此在第二步驟中，我們假設在語流中不會在五個音框之內出現兩個短暫停。所以若是抽取出的音高輪廓中有小於五個音框的片段，則認定其爲半頻或是倍頻誤差並刪除之。最後再經由線性內插將被刪除之缺口補齊。藉由上述步驟，我們便得以確保音高輪廓的平滑性及連續性。接著，同樣由每個用語的所有音框中抽取出最大、最小、平均、及範圍四值。

能量特徵 (Energy Feature)

對每個音框抽取第零項的倒頻譜係數 (0-th cepstral coefficient) 做爲能量特徵。同樣對每個用語中所有音框的最大、最小、平均、及範圍四值做爲此用語的能量特徵。

3.5.2 詞彙特徵 (Lexical Feature)

我們從語料的文本及投影片資訊內，抽取各個候選用語的詞彙特徵。這些特徵分別描述於下。

詞頻與逆向文件頻率 (Term Frequency-Inverse Document Frequency; TF-IDF)

詞頻與逆向文件頻率時常用來代表一個用語的重要性 (Significance) 。我們先將語料文本中的所有句子，以對應的投影片分割成不同的文件。接著並以投影片爲文件的單位來計算詞頻 (Term Frequency; TF) 、逆向文件頻率 (Inverse Document Frequency; IDF) 、以及詞頻與逆向文件頻率。

前文變異程度 (Left Context Variation)

我們由語料文本中，根據前文的變異程度抽取出兩種特徵。此處我們假設，在關鍵用語的前文，通常不會出現過多不同的詞。因爲若一用語爲關鍵用語，前文有很高的機率爲「使用」、「是」、「的」等等的字眼；但是若此用語不爲關鍵用語，前文可能出現的變異性就有可能很大。所以我們定義了一個特徵 – 前文變異程度 lcv_i ，爲此候選用語 t_i 在語料文本中的前文，有多少種不同的詞。除此之外，此特徵 lcv_i 可能會受到詞頻的高度影響，故我們也將此值對用語 t_i 的詞頻正規化，成爲 $lcvn_i$ 。

詞性 (Part of Speech; PoS)

由於動詞、名詞、以及形容詞有較高的機率爲關鍵片語，故我們需要多考慮候選用語的詞性來當成一特徵。我們給定不同詞性不同的數值來表示不同的詞性標記 (PoS Tag) [39]。若有一些用語不在定義的詞性中，則給定另外一個不同的數值。

3.5.3 語意特徵 (Semantic Feature)

由於語意是一能表達用語重要性的指標，故我們利用機率式潛藏語意分析模型 (Probabilistic Latent Semantic Analysis; PLSA) [28] 來分析每個文件的語意。同第 2.1 節所述，我們將每張投影片所對應的語料句子當成一文件進行此非督導式的訓練 (Unsupervised Training)，並可以得到潛藏主題下，文件與用語之間的機率關係。我們在此抽取了三種不同的特徵，詳述於下。

潛藏主題機率 (Latent Topic Probability; LTP)

如第 2.1 節所述，我們可以估算看到一用語 t_i 所對應潛藏主題 T_k 的機率

$\mathrm{LTP}(T_k \mid t_i)$ (2.9)，接著我們對每個候選用語，計算此潛藏主題機率分布的平均值 (Mean) 、變異數 (Variance) 、標準差 (Standard Deviation) 、對平均值正規化的變異數、以及對平均值正規化的標準差。

潛藏主題重要性 (Latent Topic Significance; LTS)

如第 2.1 節所述，可估算一用語 t_i 對應潛藏主題 T_k 的重要性評估 $\mathrm{LTS}_{t_i}(T_k)$ (2.10)。此參數考慮了用語 t_i 在文件 d_j 中出現的次數來估計此重要性。如潛藏主題機率一樣，我們也對每個候選用語計算上述五個數值來當成其特徵。

潛藏主題亂度 (Latent Topic Entropy; LTE)

潛藏主題亂度爲一用語在潛藏主題上分布的表現，如第 2.1 節 (2.12) 中的 $\mathrm{LTE}(t_i)$。由於較低的亂度代表此候選用語的分布多集中於少數主題下，故此值亦爲一有效評估重要性之特徵。我們對於每個候選用語皆抽取此值做爲特徵。

所有的特徵及其描述皆列於表 3.1。

3.6 藉由機器學習 (Machine Learning) 來抽取關鍵詞 (Keyword)

3.6.1 非督導式學習 (Unsupervised Learning)

由於關鍵用語的標記可能不易取得，故以下我們使用了兩個不同的非督導式學習之方法。一爲最普遍擷取關鍵用語的方法 – 詞頻與逆向文件頻率，另一爲我們提出之 K 平均模範 (K-Means Exemplar) 來擷取關鍵詞。

	特徵	描述
韻律特徵	音長 I	正規化音長之最大值
	音長 II	正規化音長之最小值
	音長 III	正規化音長之平均值
	音長 IV	正規化音長之範圍
	音高 I	基頻之最大值
	音高 II	基頻之最小值
	音高 III	基頻之平均值
	音高 IV	基頻之範圍
	能量 I	能量之最大值
	能量 II	能量之最小值
	能量 III	能量之平均值
	能量 IV	能量之範圍
詞彙特徵	詞頻	tf_i
	逆向文件頻率	idf_i
	詞頻與逆向文件頻率	$tf_i \cdot idf_i$
	前文變異程度	lcv_i
	正規化前文變異程度	$lcvn_i$
	詞性	詞性標記
語意特徵	潛藏主題機率 I	潛藏主題機率之變異數
	潛藏主題機率 II	潛藏主題機率之標準差
	潛藏主題機率 III	潛藏主題機率之平均值
	潛藏主題機率 IV	潛藏主題機率 I / 潛藏主題機率 III
	潛藏主題機率 V	潛藏主題機率 II / 潛藏主題機率 III
	潛藏主題重要性 I	潛藏主題重要性之變異數
	潛藏主題重要性 II	潛藏主題重要性之標準差
	潛藏主題重要性 III	潛藏主題重要性之平均值
	潛藏主題重要性 IV	潛藏主題重要性 I / 潛藏主題重要性 III
	潛藏主題重要性 V	潛藏主題重要性 II / 潛藏主題重要性 III
	潛藏主題亂度	潛藏主題分布的集中程度

表 3.1: 關鍵用語擷取所抽取之特徵及其描述

詞頻與逆向文件頻率 (TF-IDF)

此爲最普遍抽取關鍵用語的方法，其爲依照詞頻與逆向文件頻率的分數大小進行排序，抽取最高分的前 N 個用語當成關鍵用語，此方法爲我們評比的基準。

K 平均模範 (K-Means Exemplar)

我們使用 (2.10) 中的 $\mathrm{LTS}_{t_i}(T_k)$ 對每個用語 t_i 來建造一個特徵向量：

$$\mathbf{v_i} = (\mathrm{LTS}_{t_i}(T_1), \mathrm{LTS}_{t_i}(T_2), ..., \mathrm{LTS}_{t_i}(T_K)) \tag{3.3}$$

其中 K 爲潛藏主題的數量，所以 $\mathbf{v_i}$ 則爲在潛藏語意空間中的一向量。

我們使用 K 平均 (K-Means) 根據上述向量 $\mathbf{v_i}$ 來群集用語，並且抽取每個群集 (Cluster) 中的模範 (Exemplar) 當成關鍵用語。模範用語爲相同群集中最接近質量中心 (Centroid) 者，群集 j 中的質量中心 $\mathbf{v'_j}$ 的計算如下：

$$\mathbf{v'_j} = \frac{1}{|C_j|} \sum_{\mathbf{v_k} \in C_j} \mathbf{v_k} \tag{3.4}$$

其中 C_j 爲群集 j 內的所有向量集合。由於我們假設在潛藏語意空間中的群集通常和同一主題相關，故群集中最中心者則有很高機會爲關鍵用語 [45] ，故我們將群集分成 N 個，抽取總共 N 個關鍵詞。

3.6.2 督導式學習 (Supervised Learning)

我們使用第 3.5 節所有的特徵值以及兩種督導式學習 (Supervised Learning) 的方法來訓練分類器。分別爲第 2.2.1 節的調適性推昇法及第 2.2.2 節的類神經網路兩種學習法。

調適性推昇法 (Adaptive Boosting; AdaBoost)

調適性推昇法是組合許多基本的分類器結合而成的高準度分類器 [26] 。如第 2.2.1 節所述，對於訓練集中的樣本 $\{\mathbf{x}_n, y_n\}_{n=1}^{N}$ ，$\mathbf{x}_n$ 即爲訓練樣本的特徵向量，而 y_n 爲是否爲關鍵用語之標記 ($+1$ 表示爲關鍵用語， -1 表非關鍵用語)，數值 N 爲訓練用語的總個數。在此，如第 2.2.1 節所述，基本分類器選爲二元分類做爲基本假設 $h_{s,i,\theta}$ 。

經過 T 次的調適後，可以綜合每個時間點的假設，形成最終的假設 $H(x)$ 。便可利用此分類器來預測測試樣本是否爲關鍵用語。

$$H(x) = \text{sign}\,(\sum_{l=1}^{T} \alpha_l h_l(x)) \tag{3.5}$$

類神經網路 (Artificial Neural Network)

我們實作三層的後向推進演算法 (Backpropagation Algorithm) 的類神經網路 (b-J-1) ，其中各個神經元使用 tanh 的類型 [27] ，而 b 值代表特徵向量的維度，J 爲第二層神經元的數量。在每一次重複 (Iteration) 時都使用隨機坡降法 (Stochastic Gradient Descent) 來找最終的假設 (Hypothesis) ，並希望最小化訓練資料。

3.7 實驗基礎架構

爲了評估本論文所提出方法的成果好壞，我們設計幾個實驗來比較傳統方法與本論文所提出之方法的成果差異。以下爲實驗的設定。

3.7.1 實驗語料

本實驗使用的語料來自於單一一位老師上課的錄音，《數位語音處理》(Digital Speech Processing)。而使用的投影片爲全英文，課程錄音的內容則爲中英交雜(Code-Mixing)，主要語言爲中文，但時常使用英文來描述一些普通的詞 (如：「solution」) 及專業術語。此門課共有 45 堂，每一堂約爲一小時，總共語料的長度約爲 45.2 小時，其中總共使用 196 張投影片。在此實驗中，我們使用了人工文本 (Manual Transcriptions) 及辨識文本 (ASR Transcriptions) 的結果，進行斷句 (Sentence Segmentation) [46] 及主題分段 (Topic Segmentation) [47]，並根據對應的投影片段落分成數個文件 (一張投影片的內容及語料當成一文件)，接著分別對兩種文本擷取關鍵用語並做評估。此語料中共有 31975 句，句子又經過語言模型 (Language Model) 之中文斷詞 (Word Segmentation) [48]，英文一個單字爲一個詞，最後包含中英文共有 331509 個詞，其中有 6693 個不重複的詞，平均每個中文詞有 1.97 個字，而每句平均包含中英文共 10.37 個詞。經過前處理後，候選關鍵用語剩下 1950 個詞。而辨識文本所用到的辨識系統詳細說明於下節。

3.7.2 訓練與辨識系統

此辨識系統所用到的初始聲學模型 (Acoustic Model) 有兩個，一爲由聲碩麥克風 (ASTMIC) 所訓練的中文模型，以及由台灣中研院的語料所訓練的英文模型。藉由 25.2 分鐘的講者 (授課教授) 語料來做聲學模型的調適。而語言模型 (Language Model) 是藉由講者所開授的另外兩門課所訓練的，接著藉由投影片的文字來做些許調適 [48]。

此辨識系統在中文上的準確率 (Character Accuracy) 爲 78.15%，英文則爲 53.44%，總共辨識率爲 76.26%。英文辨識率較差的部分可以藉由使用投影片的

資訊來彌補。在此實驗中，我們將潛藏主題的數量設爲 32 ，此值對於共 17 個章節的課程來講假設爲大致合理數值。

3.8 實驗設計

在非督導式學習中，我們必須先定義欲抽取的關鍵用語數量。我們將此數量訂爲參考關鍵用語的總量。而在督導式學習中，使用三重交叉驗證法 (3-Fold Cross Validation) 來評估分類器之成效。

3.8.1 評估方法

在此我們使用三個最常使用的數值來評估抽取方法之好壞，分別爲精確率 (Precision) 、召回率 (Recall) 、以及 F 評估 (F-Measure)，並詳述於下。

精確率 (Precision)

精確率爲抽取回來的關鍵用語集合中有多少比例爲正確的：

$$P = \frac{|S_{ref} \cap S_{key}|}{|S_{key}|} \tag{3.6}$$

此處 S_{ref} 爲參考用語之集合，而 S_{key} 爲抽取之用語集合。

召回率 (Recall)

召回率爲評估有多少比例的正確關鍵用語有被抽取出來：

$$R = \frac{|S_{ref} \cap S_{key}|}{|S_{ref}|} \tag{3.7}$$

此值與 P 皆爲重要指標，但是抽取數量的多寡可能會對兩者有不同方向的影響。假如抽取的數量上升，R 值必定會提升 (因爲可能可以抽取到更多正確的關鍵用語)，但是 P 值很可能會下降 (因爲可能多抽到很多錯誤的用語)，故此兩指標皆很重要，我們在此論文主要便用 F 評估來當作最後評量成果的主要指標。

F 評估 (F-Measure)

此爲同時考慮精確率與召回率兩者，並將兩者以相同重要性組合而成的一評估指標。

$$F = \frac{2 \cdot PR}{(P + R)} \tag{3.8}$$

若此值越高，代表根據精確率與召回率，此結果有較佳的成果。

3.8.2 參考關鍵用語之生成

對於所有人 (儘管他非常熟悉課程錄音之內容)，標記關鍵用語的標準依然有很大的差異。這也是爲什麼定義一個標準答案來做評估是有點困難的事。爲了考慮不同人的意見，我們請了 61 個已修習過此門課程的學生來標記此語料中的關鍵用語。因爲不同人可能會標記出不同數量的關鍵用語，我們將標記數量的多寡考慮進每個用語重要性評估之中。也就是說，若有一個學生標記了 N 個關鍵用語，我們將其所標記的 N 個用語各獲得 $\frac{1}{N}$ 的分數。這代表我們考慮到，標記較少數量的學生 (標準較嚴格)，每個用語所獲得較高的分數；反之，每個用語獲得較低的分數。最後，我們將所有用語由此 61 個學生中獲得之分數做排序，並選出最高分的前 N_0 的用語當成關鍵用語，此處的 N_0 爲一最接近平均 N 值的整數。

根據上述程序，總共抽取出 154 個關鍵用語 (包含 59 個關鍵片語及 95 關鍵詞) 當成參考答案來做評估。表 3.2 列出幾個抽取出來的例子。

用語類型	參考用語
關鍵詞	summarization、entropy、codebook、音高
關鍵片語	map principle、speaker verification、language model、語言 模型

表 3.2: 參考關鍵用語之例子

根據抽取出的參考關鍵用語，我們可以計算所有標記者對應此答案之情形。平均精確率爲 66.13% 、平均召回率爲 90.30% 、F 評估則爲 76.37% 。

3.9 實驗結果及分析

我們藉由前述方法分別抽取關鍵片語及關鍵詞。接著使用第 3.8.1 節所述的三種評比指標來看此方法的有效性。

3.9.1 評比基準 (Baseline)

在以下的實驗中，針對關鍵片語的抽取結果，做爲評比基準 (Baseline) 的方法是 N 連文法 (N-Gram) 和相互訊息 (Mutual Information) ；N 連文法爲資訊檢索領域很常見的抽取片語之方法 [10,11,42] ，此方法列舉了所有由兩個以上的詞連接起來組合而成的片語，並依照其出現機率進行排序。這裡試圖找出前幾個出現機率較高的片語作爲關鍵片語，擷取的數量爲最接近參考關鍵片語數量之數值，此處抽出共 67 個關鍵片語當成比較基準。相互訊息亦是資訊檢索中相當常見的方法 [10,11,42] ，透過計算候選關鍵片語在子字串出現下的出現機率，相互訊息越高代表候選關鍵片語越重要，此方法亦利用後綴樹實作。

而針對關鍵詞的抽取結果，我們採用第 3.6.1 節所述傳統的詞頻與逆向文件頻率分數的方式來當作評比基準，此方法在做了前處理後根據詞頻與逆向文件頻率

來抽取前 N 個用語做爲關鍵詞。

3.9.2 分別針對關鍵詞 (Keyword) 及關鍵片語 (Key Phrase) 做評估

表 3.3列出了分別使用人工文本及辨識文本的抽取結果。在辨識文本的結果中，使用分歧亂度來抽取關鍵片語爲一有效方法，可以使 F 評估由原本約爲三成到達 68.09% (表中第三部分)，大大超越了 N 連文法及相互訊息之結果。而在關鍵詞的抽取結果則較關鍵片語來的低一點 (表中第四部分)，但是此結果依然明顯優於傳統的詞頻與逆向文件頻率方法 (表中第二行)。對非督導式學習而言，若用本論文提出之 K 平均模範可以使 F 評估到達 39.52%，此方法的表現明顯高於比較基準。

另一方面，督導式學習有較佳的結果。其中我們可以發現調適性推昇法的 F 評估有 49.44%，而類神經網路在 F 評估有 56.55%，明顯優於調適性推昇法。由此表中可看出定義關鍵詞比定義關鍵片語困難許多，在人工標記的參考關鍵用語中，關鍵詞的標記分歧度也較關鍵片語來的大。換句話說，當一個片語在語料中時常被提到，通常人們會認爲此片語爲關鍵片語，如語料中時常提到之「language model」及「語音 辨識」等片語；但若是一詞，人們認定其爲關鍵詞的標準則分歧性較大。而又因爲通常較不具意義但卻時常出現的用語，多半爲單詞，如語料中時常提到之「solution」及「語音」等詞，許多都不爲關鍵詞，是故在關鍵詞的抽取上，未來仍需要多發展更好之方法來改善擷取之結果。

而在人工文本的結果下，各方法的成效趨勢與辨識文本類似，但都稍優於辨識文本的結果。原因爲辨識文本中存在的辨識錯誤導致關鍵用語擷取較爲困難，但是本論文所提出之方法依然在辨識及人工文本上都有顯著效果。

	種類 (數量)	方法		精確率	召回率	F 評估
人工文本	1. 關鍵片語 (59)	N 連文法		25.37	28.81	26.98
		相互訊息		26.15	28.80	27.42
		分歧亂度		**59.26**	**81.36**	**68.57**
	2. 關鍵詞 (95)	非督導式	詞頻與逆向文件頻率	43.84	33.68	38.10
			K平均模範	**52.05**	**40.00**	**45.24**
		督導式	調適性推昇法	54.63	**62.11**	58.13
			類神經網路	**75.68**	58.95	**66.27**
辨識文本	3. 關鍵片語 (59)	N 連文法		21.43	25.42	23.26
		相互訊息		20.00	23.73	21.71
		分歧亂度		**58.54**	**81.36**	**68.09**
	4. 關鍵詞 (95)	非督導式	詞頻與逆向文件頻率	26.39	20.00	22.75
			K平均模範	**45.83**	**34.74**	**39.52**
		督導式	調適性推昇法	44.90	**55.00**	49.44
			類神經網路	**63.08**	51.25	**56.55**

表 3.3: 關鍵片語及關鍵詞分別在人工文本及辨識文本下的抽取成果 (%)

3.9.3 特徵效力分析

爲了分析我們抽取的特徵是否在訓練分類器時眞正具有效力，我們分別對三組不同類型的特徵進行實驗及分析。此實驗採用類神經網路 (表 3.3 中表現較好者) 在辨識文本中來抽取關鍵詞做爲分析對象，並且分析是否我們抽取的三種類型特徵都具有效果，而此實驗結果列於表 3.4中。

表中的 (a) (b) (c) 分別爲單獨使用某一類型的特徵，韻律特徵、詞彙特徵、或語意特徵，我們可以看到 F 評估的數值從 20% 一直到 42% 。表中 (d) 顯示出，整合韻律特徵及用語特徵，可以較單獨使用更好 (表中 (a) 或 (b))，表示兩者具有加成的效果。而 (e) 顯示語意特徵可以大量幫助關鍵用語的擷取，若多使用語意特徵，F 評估可以從 48.15% 上升至 56.55% ，上升幅度極大，也較單獨使用語意特徵更佳。於是我們可以得知，三種不同的特徵是在不同方面做重要性的評估，是

	特徵	精確率	召回率	F 評估
(a)	韻律	21.92	19.75	20.78
(b)	詞彙	33.57	**59.26**	42.86
(c)	語意	37.80	33.70	35.63
(d)	韻律 + 詞彙	48.15	48.15	48.15
(e)	韻律 + 詞彙 + 語意	**63.08**	51.25	**56.55**

表 3.4: 使用不同類型之特徵抽取關鍵詞 (不含關鍵片語) 之成果 (%)

故三種特徵可以整合並得到較好的結果。

而單獨一類特徵的使用上，其中又以詞彙特徵的表現最佳，F 評估達到 42.86% (如表中 (b))。在普遍的關鍵用語擷取研究上，多半都由詞彙特徵來拓展 (傳統的詞頻與文件頻率則屬於此類)，但是此特徵所衍伸的問題則爲，語音文件時常會有錯誤產生，是故可能造成無法完全符合 (Exact Match) 的情形產生，這樣特徵值可能會誤差較大，而其它兩類特徵則可以補償詞彙特徵帶來的問題。

韻律特徵由於較易受到訊號及雜訊的影響，故單獨使用韻律特徵很難得到足夠好的結果，F 評估只有 20.78% (如表中 (a))。但由於此特徵不受辨識結果之影響，它可以補足辨識錯誤所造成的詞彙特徵計算之誤差情形。

同理，語意特徵並非考慮辭彙上的完全符合，而是藉由文件中的前後文 (Context) 來進行潛藏主題的機率估計，故辨識錯誤對語意特徵的計算影響較不明顯，單獨使用可以得到 35.63% 的 F 評估 (如表中 (c))。

3.9.4 總結果

最後，我們將關鍵片語及關鍵詞組合起來做整體關鍵用語之評估，此結果列於表 3.5 。我們比較的基準方法使用 N 連文法抽取關鍵片語，以及使用詞頻與逆向文件頻率抽取關鍵詞，F 評估爲 32.19% 及 23.44% 。但從表中可以發現，若改用分

岐亂度抽取關鍵片語，依然使用傳統詞頻與逆向文件頻率的方法抽取關鍵詞，不管在人工文本或是辨識文本上，都可以得到不少進步。F 評估可以進步至 50.98% (人工文本) 或 45.61% (辨識文本)，代表利用分岐亂度之方法來擷取關鍵片語是非常有用的。換句話說，當一個詞序列 (如：資訊 理論) 被分岐亂度所擷取到時，此片語爲關鍵片語的機率則非常高。

由非督導式學習的 K 平均模範來看，可以有效進步 F 評估到達 55.84% (人工文本) 及 52.60% (辨識文本)。考慮到此方法沒有使用人工標記的答案來訓練，此表現也很好了。

而最好的結果是督導式學習中，藉由類神經網路分類關鍵詞所產生，此爲經由分岐亂度抽取關鍵片語後再做關鍵詞的擷取。在人工文本上的 F 評估爲 67.31%，在辨識文本上的 F 評估爲 62.70%，此值和人工標記的結果 (F 評估爲 76.37%) 相比，也沒有太大的差異，甚至在精確率的評估下，在人工文本上的擷取還以 67.10% 勝過人工標記 66.13% 的精確度。由此可知，本論文所提出之關鍵用語擷取方法非常有效。

3.9.5 結果分析及討論

有一些關鍵用語，每次被語者提到時都無法被辨識系統正確辨識出來，即使此用語在語料中不斷出現，這些用語依然沒有辦法在辨識文本中被擷取出來。人工文本和辨識文本的成果上的差距可能是因爲此原因所造成。要改善此問題，需要直接對語音訊號抽取重複出現的語音訊號。但是由於關鍵用語通常語者會說的較清晰，以強調的方式來發音，故多半關鍵用語並不會太難被辨識正確。是故即使人工文本和辨識文本上有些許的成果差異存在，但差距並不算太大，此論文提出之方法依然可以有效擷取到許多正確之關鍵用語。

	方法		精確率	召回率	F評估
人工文本	基準		33.81	30.72	32.19
	非督導式	詞頻與逆向文件頻率	50.98	50.98	50.98
		K平均模範	**55.84**	**55.84**	**55.84**
	督導式	調適性推昇法	56.61	**69.48**	62.39
		類神經網路	**67.10**	67.53	**67.31**
辨識文本	基準		26.67	20.92	23.44
	非督導式	詞頻與逆向文件頻率	49.24	42.48	45.61
		K平均模範	**52.60**	**52.60**	**52.60**
	督導式	調適性推昇法	51.11	**66.19**	57.68
		類神經網路	**60.61**	64.94	**62.70**
人工標記			66.13	90.30	76.37

表 3.5: 使用不同方法於人工及辨識文本的綜合成果 (%)

3.9.6 同類語音文件之實驗分析

爲了有效證明本論文所提出方法之有效程度，我們同時也對另一門課程錄音《訊號與系統》(Signal and System) 進行關鍵用語擷取之實驗。此課程與《數位語音處理》的內容具有些許差異，如包含了較多的數學參數、式子、以及定理的證明，由於上述內容不斷重複被提及，故關鍵用語擷取會更加的困難。

最終實驗結果顯示，擷取出的關鍵用語經過評估後，和前一門課程相似 [40]，包含關鍵片語及關鍵詞約有 50% 以上的 F 評估， 故可以驗證本論文所提出關鍵用語之擷取方法對於同類的課程錄音都是很有效的。

3.10 本章總結

在此章，我們提出了一有效於語音文件中擷取關鍵用語的方法，並且使用課程語音做爲實驗。我們發現關鍵用語的類型多半分爲兩種：關鍵片語及關鍵詞，故針

對此兩種類型發展了不同的方法。根據擷取流程架構，按照順序地擷取兩種不同類型之關鍵用語，經由實驗的驗證及分析，我們可以發現此論文提出的方法非常有效。

第 四 章　語音文件之自動摘要

Spoken Document Summarization

4.1　簡介

多媒體資訊在網路上非常豐沛，除了關鍵用語的精簡資訊外，許多研究致力於自動摘要之處理。因爲將多媒體資訊 (影像或語音) 進行自動摘要，使用者可以有效率地瀏覽此多媒體文件，經由較精簡的訊息，使用者便可以得到整個多媒體文件的內容。針對語音文件，比圖像更難瀏覽，必須由其中精確地抽取重要的語音，並結合成一較短之摘要。此方法稱爲抽取式摘要 (Extractive Summarization)，現今自動摘要的研究多半歸屬於此類，而也越來越多研究開始拓展此領域之發展 [49]。

在文字文件中，已有較多自動摘要之研究 [50]。後來漸漸有許多研究致力於語音文件自動摘要的發展，但其多半針對新聞內容作擷取，由於新聞內容的講者經過專業訓練，講話發音較清晰，而辨識的結果也會較佳。但最近許多研究開始將焦點轉移至新的方向，像是課程錄音 (Lecture) 和會議 (Meeting) 的語音紀錄 [1,51]，此類語音文件的語者多半是未經訓練的人，語音內容較爲自發性 (Spontaneous)，辨識率也較低。但在此類語料中自動摘要爲現今的研究趨勢，此論文也採用課程語料作爲自動摘要之實驗評估對象。

許多方法在原始的語音文件中抽取重要句子，重要句子的定義爲包含文件中心概念，較接近整份文件意義之句子，將這些句子按照原語音文件的順序進行排列，成爲最後的摘要 [52]。語音文件摘要最大的挑戰在於辨識錯誤、口語的語音帶來的問題，以及缺乏句子及段落的分界。下節會介紹此領域的傳統方法，針對

上面所提到的問題進行處理。

先前的一些研究顯示，由潛藏式語意分析模型所得到的主題資訊 (Topical Information) 對於估計一用語在文件中的重要性非常有幫助，利用潛藏語意的統計分析可以獲得比詞彙上分析更好的結果，並且藉此來定義每一用語在文件中的重要性評估，故在此論文中，我們也使用機率式潛藏語意分析模型來計算主題資訊，接著提出圖學式的隨機漫步演算法 (Random Walk) 來重新計算 $I(S,d)$ 的分數，此演算法不只考慮用語之重要性，也考慮整個文件中，句子彼此的相似性，所以若有一句子和很多重要的句子相似的話，此句子則會得到較高之分數。

4.2 傳統語音文件自動摘要

在語音文件中進行自動摘要最普遍的方法是由弗氏 (Furui) 所提出 [53]，其中針對文件 d 中的每個句子 $S = t_1 t_2 ... t_i ... t_n$ (一連串的用語 t_i 的集合)，評估一句子之重要性分數如下：

$$I(S,d) = \frac{1}{n}\sum_{i=1}^{n}[\lambda_1 s(t_i,d) + \lambda_2 l(t_i) + \lambda_3 c(t_i) + \lambda_4 g(t_i)] + \lambda_5 b(S) \qquad (4.1)$$

其中 $s(t_i,d), l(t_i), c(t_i), g(t_i)$ 分別代表之意義如下所示：

- $s(t_i,d)$：對於用語 t_i 統計學方面之評估 (如：詞頻與逆向文件頻率)
- $l(t_i)$：對於用語 t_i 語言學方面之評估 (如：不同詞性具有不同比重)
- $c(t_i)$：對於用語 t_i 的辨識信心分數 (Confidence Score)

  信心分數是評估聲學模型及語言模型下，此用語辨識出來的信心程度，也是一個用語被辨識出來的事後機率。此分數也爲解碼器 (Decoder) 藉由詞圖

(Word Graph) 來計算此詞與其它詞之比例數值。

- $g(t_i)$：對於用語 t_i 的 N 連文法分數 (N-Gram Score)

  N 連文法分數指出此詞在句子中藉由 N 連文法所計算之機率值，在實驗中多半使用語言模型中詞三連 (Trigram) 之機率。此分數是將辨識錯誤可能產生之語句上不自然的詞串的情形考慮進去，下降整個句子之重要性。

- $b(S)$：計算句子 S 的文法結構之分數

而 $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ 及 λ_5 都爲加權參數，對每個文件 d ，一句子是否應該被抽取出來當成摘要是根據此重要性分數 $I(S, d)$。

4.3 用語統計學方面之重要性評估

此章節介紹評估一用語在文件中的重要性依據，也就是 (4.1) 中的 $s(t_i, d)$ 的評估方法。會先介紹傳統之方法於第 4.3.1 節，接著在第 4.3.2 節介紹一改進之方法，最後詳細說明此論文提出的估算方法，考慮第三章擷取出來的關鍵用語當成額外的資訊來進行較準確之估計。

4.3.1 傳統重要性分數 (Significance Score)

傳統被提出來估計一用語之重要性分數 [53] 如下式：

$$s_{\mathrm{SS}}(t_i, d) = n(t_i, d) \log \frac{F_A}{F_{t_i}} \tag{4.2}$$

其中 $n(t_i, d)$ 爲用語 t_i 在文件出現次數，F_{t_i} 爲 t_i 在整個語料中出現次數，F_A 爲所有用語在整個語料中出現的總次數。此方法的基本概念和詞頻與逆向文件頻率的

概念很相像。

4.3.2 基於潛藏主題亂度之統計計算 (LTE-Based Statistical Measure)

先前的一些研究顯示 [32,33]，由潛藏式語意分析模型所得到的主題資訊 (Topical Information) 對於估計一用語在文件中的重要性 ((4.1) 中的 $s(t_i, d)$) 非常有幫助，利用潛藏語意的統計分析可以獲得比詞彙上分析更好的結果。(4.1)中用語在統計上的重要性評估 $s(t_i, d)$ 可由 (2.12) 中潛藏主題亂度 $\mathrm{LTE}(t_i)$ 來估計：

$$s_{\mathrm{LTE}}(t_i, d) = \frac{\beta n(t_i, d)}{\mathrm{LTE}(t_i)} \tag{4.3}$$

其中 β 爲一比例參數，故 $s_{\mathrm{LTE}}(t_i, d)$ 會與潛藏主題亂度 $\mathrm{LTE}(t_i)$ 成反比。一些先前之研究顯示 [32,33]，這個統計的評估方法比第 4.3.1 節的重要性分數在做語音摘要上有效許多，故此論文使用此值 $s_{\mathrm{LTE}}(t_i, d)$ 當作比較之基準。

4.3.3 基於關鍵用語資訊之統計計算 (Key-Term-Based Statistical Measure)

根據第三章自動擷取的關鍵用語 (包含一些錯誤的關鍵用語)，我們可以藉由這額外的資訊來估計一個基於關鍵用語的新的潛藏主題機率 $P_{\mathrm{KEY}}(T_k \mid d)$，並希望此機率可以較直接從機率式潛藏語意分析模型而計算的 $P(T_k \mid d)$ 來的更好。我們利用關鍵用語的資訊來估算 $P_{\mathrm{KEY}}(T_k \mid d)$ 如下：

$$P_{\mathrm{KEY}}(T_k \mid d) = \frac{\sum_{t_i \in key} n(t_i, d)\, \mathrm{LTP}(T_k \mid t_i)}{\sum_{k=1}^{K} \sum_{t_i \in key} n(t_i, d)\, \mathrm{LTP}(T_k \mid t_i)} \tag{4.4}$$

此處的 key 爲自動抽取出的關鍵用語之集合，而 $\mathrm{LTP}(T_k \mid t_i)$ 爲 (2.9) 中，一用語所對應的潛藏主題的機率值。由此式可以發現，在計算一文件所對應的潛藏主題的機率時，我們只考慮在文件 d 當中的關鍵用語 t_i 的潛藏主題機率分布，以此來避免其它不重要的用語之機率分布所帶來的影響。

在有了文件對潛藏主題的機率分布 $P_{KEY}(T_k \mid d)$ 後，接著我們定義用語 t_i 統計上的重要性評估 $s(t_i, d)$ 如下：

$$s_{\mathrm{KEY}}(t_i, d) = \sum_{k=1}^{K} \mathrm{LTS}_{t_i}(T_k) P_{\mathrm{KEY}}(T_k \mid d) \tag{4.5}$$

此處使用了 (2.10) 中 $\mathrm{LTS}_{t_i}(T_k)$ 的資訊，根據用語對潛藏主題的重要性當作權重來計算一用語 t_i 在文件 d 中的重要性評估。

4.4 圖論式的重估計法 (Graph-Based Re-Computation)

我們將選擇句子之問題看成一有向的圖，並藉由圖論的方法 – 隨機漫步演算法來估算每個點之重要性。首先將每個句子當成圖上的一個點，兩個點之間的邊則根據兩個句子之間的相似程度 (Similarity) 來加權之。此演算法的基本概念爲，若一個句子和很多重要的句子較相似，此句子也應該具有較高的重要性。如此一來，所有在文件中的句子都會一起被考慮進來，而非單一考慮少數句子。我們建立此有向圖，並且對點之間的邊以不同的兩個方向進行相似性之計算，此爲兩句子間不對稱之主題相似性。接著對於每個點保留前 N 個分數最高的外向邊 (Outgoing Edge)，而整個圖是藉由內向邊 (Incoming Edge) 來傳遞每個點的重要性。圖 4.1 爲一簡單之例子，其中 $Out(S_i)$ 爲點 S_i 經由外向邊連出之鄰居的點集合，而 $In(S_i)$ 則爲經由內向邊連到之鄰居點集合。

這類的概念及方法在語音用語偵測 (Spoken Term Detection) 及影像搜尋都有所應用 [54,55] 。而有些研究也有利用類似的概念應用在自動摘要上，但是由於這些研究在計算點之間的相似性時，都使用詞彙上的特徵 (Lexical) ，如詞彙特徵排名演算法 (LexRank) [22] ，並未考慮語意上的相似性，此論文提出藉由潛藏主題來評估兩句子間的相似性，在此稱爲主題相似度 (Topical Similarity) ，並重新估算句子之重要性。考慮主題相似度是由於語音文件中含有辨識錯誤之字詞，而辨識錯誤對於主題相似度的影響較詞彙相似度來的小，故此處使用主題相似度來針對語音文件做處理，進而得到較好的效果。

4.4.1 主題相似度 (Topical Similarity) 計算

此章節欲計算一文件中兩句子間的主題相似度，此相似度爲透過機率式潛藏語意分析模型所估計出的潛藏主題參數來估算的，詳述於下。

在文件 d 中，我們先計算一個句子 S_i 產生一潛藏主題 T_k 之機率：

$$P(T_k \mid S_i) = \frac{\sum_{t \in S_i} n(t, S_i)\, \mathrm{LTP}(T_k \mid t)}{\sum_{t \in S_i} n(t, S_i)} \tag{4.6}$$

$n(t, S_i)$ 爲語句 t 在句子 S_i 中出現的次數，此式是由 S_i 中所含的用語 t 在各個潛藏主題機率值的總和。 接著，要計算由 S_i 到 S_j 不對稱的主題相似度 $sim(S_i, S_j)$ $(S_i \rightarrow S_j)$，此式被定義成句子 S_i 所展開的潛藏主題空間 $P(T_k \mid S_i)$ 中，在句子 S_j 中所有用語 t 的潛藏主題重要性 $\mathrm{LTS}_t(T_k)$ 的總和，如下式：

$$sim(S_i, S_j) = \sum_{t \in S_j} \sum_{k=1}^{K} \mathrm{LTS}_t(T_k) P(T_k \mid S_i) \tag{4.7}$$

注意此處 $sim(S_i, S_j)$ 和 $sim(S_j, S_i)$ 的值爲不對稱的。計算完主題相似度

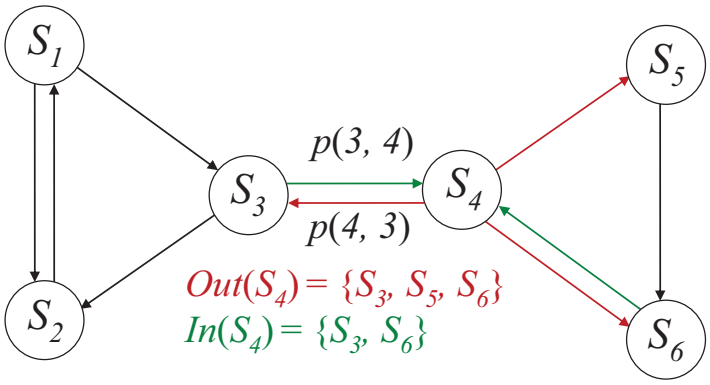


圖 4.1: 一簡單圖示例子：文件中每個句子被表示成圖上一點，而 $Out(S_i)$ 和 $In(S_i)$ 爲 S_i 分別透過向外邊及向內邊所連接的鄰居

$sim(S_i, S_j)$ 後，保留每個點 S_i 連出去的前 N 個最相似的鄰居 S_k，並對其的相似度總和正規化至 1。也就是圖 4.1中 S_i 到 S_j 邊上的權重 $p(i, j)$，是針對所有 $S_j \in Out(S_i)$ 進行正規化的相似度，如下式：

$$p(i,j) = \frac{sim(S_i, S_j)}{\sum_{S_k \in Out(S_i)} sim(S_i, S_k)} \tag{4.8}$$

而 $p(i, j)$ 會符合下列限制：

$$\sum_{S_j \in Out(S_i)} p(i,j) = 1 \tag{4.9}$$

所有由 S_i 連出去的邊上的相似度總和爲 1。

4.4.2 隨機漫步演算法 (Random Walk)

隨機漫步演算法是由網頁排名演算法 (PageRank) 衍伸而來 [23]，考慮一初始已知的事前分數 (Prior)，對整個網頁排序法進行前項之修改，接著在圖上進行類似網頁排序法的分數傳遞的動作。$v(i)$ 爲點 S_i 的新分數，此新分數爲以下兩個分數結合而成：

- 正規化後的句子 S_i 初始重要性 $r(i)$
- S_i 的所有鄰居 S_j 經由 $p(j,i)$ 之加權對 S_i 之分數貢獻

隨機漫步演算法的式子如下：

$$v^{(k+1)}(i) = (1-\alpha)r(i) + \alpha \sum_{x_j \in In(S_i)} p(j,i)v^k(j) \tag{4.10}$$

其中 α 爲一權重之分配參數，$In(S_i)$ 爲 S_i 透過向內連接的邊所連接的鄰居，而 $v^{(k+1)}(i)$ 代表點 i 在第 $(k+1)$ 個迭代 (Iteration) 的值，其中 $r(i)$ 爲正規化後的初始句子重要性：

$$r(i) = \frac{I(S_i,d)}{\sum_{S_j} I(S_j,d)} \tag{4.11}$$

此爲 (4.1) 所定義的句子重要性 $I(S_i,d)$，並對文件 d 中所有句子重要性之和做正規化的動作。由 (4.10) 可知，每次迭代都是根據每個句子與其他句子的主題相似度和句子之重要性來更新每次的值，故每個句子的重要性都會根據和其他句子之相似度將分數分出去，和越多重要句子相像之句子就可以漸漸獲得較高之分數。同時初始句子之重要性也一直被考慮在內 ((4.10) 的前項爲初始句子重要性)，故隨機漫步演算法可以有效使重要句子獲得較高的分數。而我們可以將之寫成向量與矩陣之形式，$\mathbf{v}^k = [v^k(i), i = 1,2,...,L]^{\mathbf{T}}$ 與 $\mathbf{r} = [r(i), i = 1,2,...,L]^{\mathbf{T}}$ 皆爲列向量 (Column Vector)，是文件中每個句子在第 k 次迭代的分數 $v^k(i)$ 以及原始分數 $r(i)$。其中 L 代表文件 d 中所有句子總數，而 $\mathbf{T}$ 表示轉置 (Transpose)。故我們可以將 (4.10) 以向量及矩陣方式如下表示：

$$\mathbf{v}^{(k+1)} = (1-\alpha)\mathbf{r} + \alpha\mathbf{P}\mathbf{v}^k \tag{4.12}$$

其中 $\mathbf{P}$ 爲一 $L \times L$ 的矩陣，其中皆爲 $p(j,i)$ 之數值，隨機演算法最後必定可以使 $\mathbf{v}$ 收斂，以下說明之。

假使 $\mathbf{v}$ 最後迭代收斂至 $\mathbf{v}^*$ ，我們可將 (4.12) 轉爲下式：

$$\mathbf{v}^* = (1-\alpha)\mathbf{r} + \alpha\mathbf{P}\mathbf{v}^* = ((1-\alpha)\mathbf{r}\mathbf{e}^{\mathbf{T}} + \alpha\mathbf{P})\mathbf{v}^* = \mathbf{P}'\mathbf{v}^* \tag{4.13}$$

其中 $\mathbf{e} = [1,1,...,1]^{\mathbf{T}}$ 爲一 L 維的向量，其中所有數值皆爲 1 。由 (4.10) 及 (4.11) 可知 $\sum_i v^*(i) = 1$ ，故 $\mathbf{e}^{\mathbf{T}}\mathbf{v}^* = 1$ 。隨機漫步演算法和網頁排序法一樣可以保證有最佳解，已有證明顯示 (4.13) 的解 $\mathbf{v}^*$ 即爲 $\mathbf{P}'$ 的支配特徵向量 (Dominate Eigenvector) ，其對應 $\mathbf{P}'$ 的最大特徵值 (Eigenvalue) 爲 1 。

此解 $\mathbf{v}^*$ 中的 $v^*(i)$ 可以當作句子 S_i 的新分數，由於 $v^*(i)$ 在以下兩種情況下會有較大的數值：

1. 句子 S_i 有較高的初始重要性
2. S_i 較相似於它的鄰居 S_j ，而 S_j 都具有較高的重要性分數

第一個情況會造成較高的 $r(i)$ ，而第二個情況會造成 S_j 傳遞過來的分數值較高，以上兩種情況都可能造成 $v^*(i)$ 有較高的值，也就是 S_i 會有較高的重要性。

由於以上已證明了最佳解之存在，我們可以根據資料量的大小選擇較適合之解法：

- 資料量大

  能力法 (Power Method) 可在極大資料量下有效率解出 (4.13) 中的 $\mathbf{v}$ [56,57]。此處我們使用一較簡單的方法來計算近似值：訂一較小閾值 (Threshold) ϵ ，每個迭代計算 $\mathbf{v}$ 改變量，當改變量小於 ϵ 時，則停止迭代，將 $\mathbf{v}$ 當成已收斂之值。

- 資料量小

  直接解 $\mathbf{P}'$ 的支配特徵向量 $\mathbf{v}$ 。

在資料量大時以迭代方式計算的隨機漫步演算法列於 Algorithm 3。

Algorithm 3 隨機漫步演算法 (Random Walk)

Input: 句子集合 $\mathcal{S} = \{S_i, i = 1, ...L\}$ 、閾值 $\epsilon = 0.0001$

Output: 重計算之重要性分數 $\mathbf{v}^*$

1: 建立正規化之初始重要性分數 $\mathbf{r}$;
2: 建立矩陣 $\mathbf{P}$;
3: 初始化 $k \leftarrow 1, \delta \leftarrow 0$;
4: **for** $k = 1$ to T **do**
5: $\mathbf{v}^{(k+1)} = (1-\alpha)\mathbf{r} + \alpha\mathbf{P}\mathbf{v}^k$;
6: $\delta \leftarrow |\mathbf{v}^{(k+1)} - \mathbf{v}^{(k)}|$;
7: $k \leftarrow k + 1$;
8: **if** $\delta < \epsilon$ **then**
9: $\mathbf{v}^* \leftarrow \mathbf{v}^{(k+1)}$;
10: break;
11: **end if**
12: **end for**

經過隨機漫步演算法後，由於此處的 $v^*(i)$ 數值爲介於 0 到 1 之間，故我們將此值與原始的重要性分數 $I(S_i, d)$ ((4.3) 中的基於潛藏主題亂度之統計計算或 (4.5) 中的基於關鍵用語資訊之統計計算所得到的句子重要性分數) 做結合如下：

$$\hat{I}(S_i, d) = I(S_i, d)(v^*(i))^{\gamma}, \tag{4.14}$$

其中 γ 爲一加權參數。 最後根據 $\hat{I}(S_i, d)$ 當作句子 S_i 的最終分數來排序句子。

4.5 實驗基礎架構

爲了評估本論文所提出方法的成果好壞，我們設計幾個實驗來比較傳統方法與本論文所提出之方法的成果差異。以下爲實驗的設定。

4.5.1 實驗語料與訓練辨識系統

本實驗使用的語料與第 3.7.1 節所述相同，亦爲《數位語音處理》 (Digital Speech Processing) 的課程錄音內容。而我們由 155 的語音文件中挑選出 35 個當作評估對象，每個文件的平均長度約爲 17.5 分鐘。而辨識文本所用到的辨識系統則如同第 3.7.2 節所述。

4.5.2 關鍵用語資訊

在第三章所提到的關鍵用語 (包含關鍵詞及關鍵片語) 經由本論文提出之方法被擷取出來，其中使用的關鍵用語 F 評估爲 62.70% (辨識文本) 及 67.31% (人工文本)。

4.6 實驗設計

在實驗中，將隨機漫步演算法 (4.13) 的 α 設爲 0.9 ，此數值是經驗上較好的選擇，在許多相關研究中都顯示了 $0.85 - 0.9$ 是較好的選擇 [54,55] 。而後，我們會做實驗分析不同的 α 是否會爲結果帶來不同的影響。

4.6.1 參考摘要之生成

我們請了兩個學生來產生參考摘要以供評估結果，我們請他們由文件中挑出重要的句子，並按照「最重要 (the most important) 」排到「一般重要 (of average importance) 」。根據挑出來的句子所形成的摘要，我們用以下方式來評估自動摘要的成效好壞。

4.6.2 評估方式

一有名的評估方式 – ROUGE 在許多研究中都被廣泛地使用 [58]。ROUGE 提供了許多不同的評估方法，在此實驗中，我們使用了較普遍的 ROUGE-N ($N = 1, 2, 3$) 以及 ROUGE-L 分數。$\mathrm{ROUGE_r}$-N 是自動產生之摘要和人工抽取之摘要之間 N 連文法的召回率，人工抽取的參考摘要爲一集合 R ，故 $\mathrm{ROUGE_r}$-N 如下式：

$$\mathrm{ROUGE_r\text{-}N} = \frac{\sum_{Sum \in R} \sum_{\mathrm{gram}_N \in Sum} \mathrm{Count_{match}}(\mathrm{gram}_N)}{\sum_{Sum \in R} \sum_{\mathrm{gram}_N \in Sum} \mathrm{Count}(\mathrm{gram}_N)} \tag{4.15}$$

其中 Sum 爲參考摘要集合 R 中的任一摘要，N 代表考慮詞彙之連續長度，也就是我們所稱的 N 連文法 – gram_N ，而 $\mathrm{Count_{match}}(\mathrm{gram}_N)$ 是 N 連文法同時出現於自動摘要及人工摘要的最大數量，$\mathrm{Count}(\mathrm{gram}_N)$ 則爲參考摘要出現 N 連文法總數量。ROUGE-L 的計算方式和 ROUGE-N 很相似，只是它只考慮自動摘要及參考摘要的「最長共同子字串 (Longest Common Subsequence; LCS) 」。在一些相關的研究中，ROUGE-1 及 ROUGE-L 被看作是較好評估短摘要的指標，而較長的摘要則較看重 ROUGE-2 、 ROUGE-3 。召回率可以上方式子得到，同理，我們也可以爲此四值計算精確率及 F 評估，以下我們使用 F 評估來分析自動摘要之成果。

4.7 實驗結果及分析

在以下的實驗中，摘要所佔之比例分別被設爲 10% 、 20% 及 30% 。另外，在包含關鍵用語資訊的實驗所訓練的機率式潛藏語意分析模型中，關鍵片語會被連結爲一用語，使估計結果可以更準確。

4.7.1 評量結果

圖 4.2 顯示 ROUGE-N 及 ROUGE-L 分別對應辨識文本 (圖 4.2 (a)-(d)) 及人工文本 (圖 4.2 (e)-(h)) 。在每一個實驗中，有三組條狀圖分別對應 10% 、 20% 及 30% 的摘要比例。而在每一組中共有四個條柱：

1. LTE (基準): 基於潛藏主題亂度之統計計算 ((4.3)中的 $s_{\mathrm{LTE}}(t_i, d)$)
2. LTE + RW: 基於潛藏主題亂度之統計計算後接著做隨機漫步演算法。
3. Key: 基於關鍵用語資訊之統計計算 ((4.5) 中的 $s_{\mathrm{KEY}}(t_i, d)$)
4. Key + RW: 基於關鍵用語資訊之統計計算後接著做隨機漫步演算法

在所有實驗中，我們可以發現，基於關鍵用語資訊之統計計算 (第 3 條) 總是較基於潛藏主題亂度之統計計算 (第 1 條) 來的好。故我們可以由實驗證明出，關鍵用語資訊 (即使其中具有些許錯誤之關鍵用語) 對於課程摘要上的抽取是非常有幫助的，尤其是人工文本的情況，進步量非常顯著。這可能是因爲在人工文本上，所有關鍵用語都是正確被辨識出來 (雖然可能包含一些錯誤的關鍵用語) ，所以基於關鍵用語的統計計算則能夠較準確的估計一用語在文件中的重要性。同理，在辨識文本上也可以看到顯著的進步，但是進步量可能不如人工文本來的高。這可能是因爲其中包含了辨識錯誤的詞，並非所有關鍵片語的重要性可以準確評估，故進步量不如人工文本來的多。

幾乎在所有的實驗中，根據主題相似度的隨機漫步演算法也明顯可以幫助基於主題亂度之統計計算 (第 2 條 vs 第 1 條)，而隨機漫步演算法並非總是可以使基於關鍵用語資訊之統計計算的結果進步。我們發現對於辨識文本來說 (圖 4.2 (a)-(d))，隨機漫步演算法可以使基於關鍵用語資訊之統計計算的結果進步 (第 4 條 vs 第 3 條)。但是在人工文本中摘要比例爲 10% 時 (圖 4.2 (e)-(h))，隨機漫步演算法並沒有像其他實驗一樣將基於關鍵用語資訊之統計計算的結果進步 (第 4 條 vs 第 3 條)。理由可能是因爲在人工文本上，在 10% 以內的重要句子幾乎和關鍵用語相關，故它們都可以被基於關鍵用語資訊之統計計算抽取出來，所以若加上不同句子間的額外的主題相似度可能無法將此結果有所進步。但是在排在 10% 到 30% 之間的句子重要性就沒有最前面的句子來的明確，所以隨機漫步演算法在這些情況下 (摘要比例爲 20% 及 30%) 就可以在 ROUGE-1 及 ROUGE-L 下提升基於關鍵用語資訊之統計計算的結果 (第 4 條 vs 第 3 條) (圖 4.2 (e)-(h))。

表 4.1 列出以上所有實驗與基準相比之最大相對進步率 (Relative Improvement; RI)。在辨識文本中，各個比例的摘要進步率都差不多，而最好的結果幾乎皆來自基於關鍵用語資訊計算並經由隨機漫步演算法重計算 (圖 4.2 中第 4 條)。原因是因爲辨識文本中存在許多辨識錯誤，利用主題相似度可以彌補辨識錯誤造成之影響，並且在各個比例都有效抽取重要的句子。

另一方面，在人工文本中，摘要比例爲 10% 時進步率爲最高的，來源有時爲只用基於關鍵用語資訊計算 (圖 4.2 中第 3 條)，有時又是多使用了隨機漫步演算法 (圖 4.2 中第 4 條)。這可能是因爲其中並未存在辨識錯誤，故藉由關鍵用語來估計句子重要性本身就有很高的準確性，多使用了隨機漫步演算法在此情況不一定能有效提升結果的好壞。而摘要比例爲 10% 時的進步率較其他明顯高出許多，故我們可以推斷，在短的摘要時，前 10% 的句子幾乎多半都和關鍵用語具有強烈

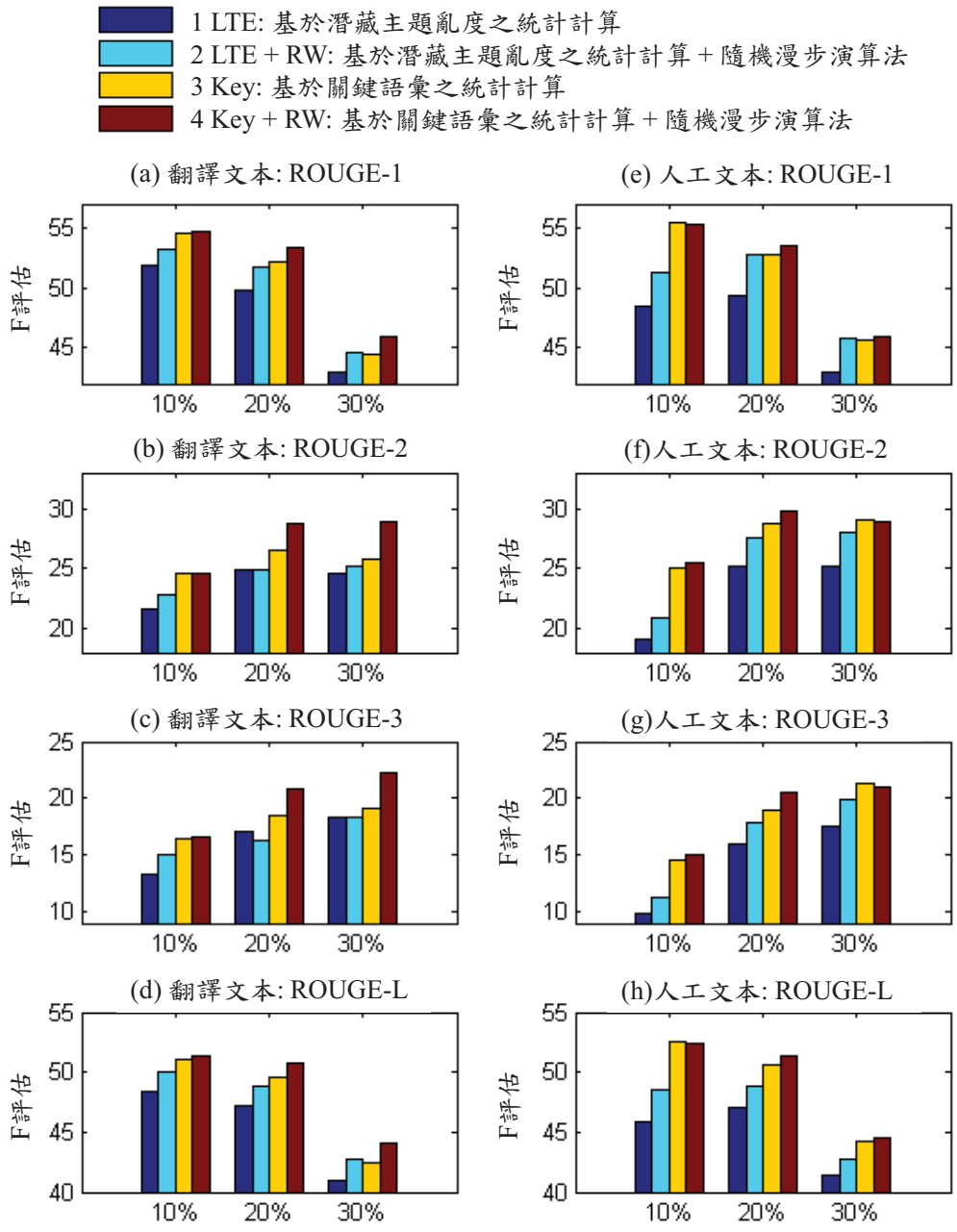


圖 4.2: 不同參數之選擇的 ROUGE 結果：「基於潛藏主題亂度統計計算」(1, 2)或「基於關鍵用語統計計算」(3, 4) 配合 (1, 3) 或不配合 (2, 4) 「隨機漫步演算法」對應於辨識文本 ((a)-(d)) 或人工文本 ((e)-(h)) ，其中摘要比例爲 10% 、 20% 及 30%

最大相對進步率	辨識文本			人工文本		
	10%	20%	30%	10%	20%	30%
ROUGE-1	5.30	7.19	7.13	14.37	8.27	6.74
ROUGE-2	13.94	15.48	17.45	33.31	18.33	15.33
ROUGE-3	24.98	22.23	21.33	50.98	28.21	22.25
ROUGE-L	6.23	7.51	7.92	14.52	9.12	7.49

表 4.1: 在所有實驗中與基準相比之最大相對進步率 (%)

的關係，所以在沒有辨識錯誤的情況下，前 10% 的重要句子可以被基於關鍵用語資訊之統計計算準確估計出來。但由於關鍵用語之 F 評估大約爲 67% ，故在比例 20% 和 30% 的情形下，只利用關鍵用語資訊之統計計算確實可能會抽出一些雜訊，也就是錯誤的句子，進步量就沒有 10% 時來得多。

4.7.2 α 之敏感程度 (Sensibility)

本論文所提出之方法中，(4.10) 包含了一可改變參數 α ，α 越接近 1 代表句子的重要性評估較看重句子彼此間的重要性。故本節對於不同 α 做了相同實驗，並且評估本論文所提出之方法是否在不同參數的選擇下，都可以獲得不錯的結果。

圖 4.3 爲辨識文本中，基於關鍵用語資訊之統計計算及隨機漫步演算法使用 (4.10) 中不同 α 值的相對進步率。我們可以發現，最後 ROUGE 結果幾乎不受 α 的影響，只有在 $\alpha = 0.8, 0.9$ 時進步量明顯較多。這代表本論文提出之隨機漫步演算法非常穩定，不會因爲參數之選擇而造成結果波動不穩之情形產生。而最佳的 α 值也和其他研究相似，多落在 $0.8 - 0.9$ 的位置。這也代表考慮相較於原始的統計重要性計算，句子彼此間的主題相似度更加重要。根據此句子和其他句子之主題相似性，可以幫助我們估算一句子之重要性，並抽取摘要。

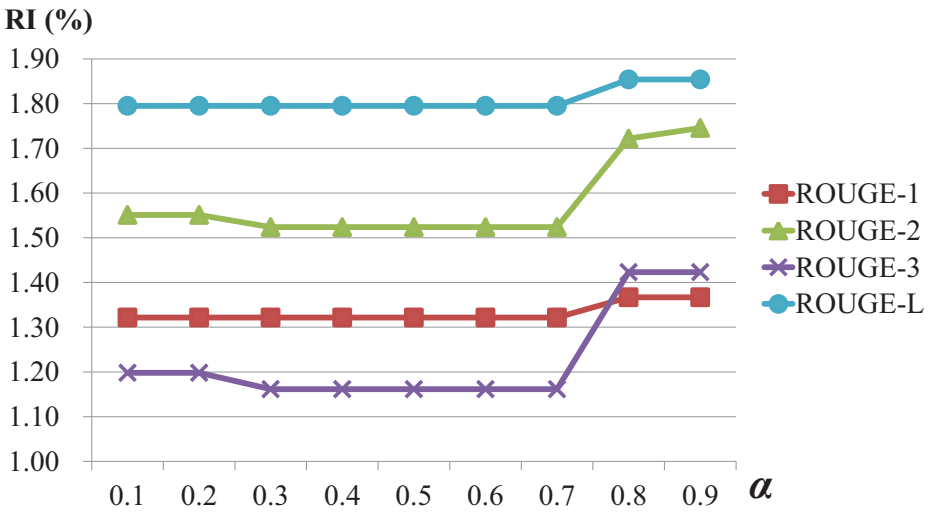


圖 4.3: 在基於關鍵用語資訊之統計計算中，經過隨機漫步演算法抽取辨識文本中 30% 摘要對應不同 α 值的相對進步率

4.7.3 結果分析

有些句子未包含明顯的重要用語，但可能敘述一些重要的觀念，這樣的句子在基於關鍵用語資訊之統計計算提出的方法上就沒辦法直接將之抽取出來。但是由於本論文有多考慮主題相似度，並藉由隨機漫步演算法做了第二階段的處理，故在此情況下，這些包含重要觀念的句子可能可以藉由和關鍵用語有較相似的潛藏主題分布而獲得較高分數，最終成爲摘要的句子。

另外，有時候每個句子的潛藏主題估計會有小誤差，由於計算句子潛藏主題是根據句子內部的用語所平均而來，有可能發生相鄰句子的潛藏主題變化太過劇烈，可能不符合實際情形。因爲在現實情況下，較少會有兩個相鄰的句子彼此的主題變化太大的情形，多半情況主題的改變是較平滑的，本論文所提出的方法並未考慮這樣的情形，故與現實情況還是存在一些誤差。

4.7.4 同類語音文件之實驗分析

同樣地，我們同時也對《訊號與系統》 (Signal and System) 的課程錄音進行自動

F 評估	人工文本		
	10%	20%	30%
ROUGE-1	36.24	57.47	58.68
ROUGE-2	27.73	46.51	47.21
ROUGE-3	26.16	43.31	44.81
ROUGE-L	35.89	57.12	57.50

表 4.2: 在《訊號與系統》的實驗中進行評估之最佳結果 (%)

摘要。在此門課程上，由於重複地提及數學式等等，重要資訊之擷取更加困難。故基於關鍵用語之統計計算在此處更能表現其明顯的優勢，再經由隨機漫步演算法也能更進一步獲得較好的結果。由於時間及資源上限制，在此實驗上進行的 ROUGE 評估，只使用了少量的參考摘要，而實驗結果秀於表 4.2 ，由表中我們可以相信並驗證本論文所提出摘要抽取的方法對於同類的課程錄音也是很有效的。

4.8 本章總結

於此章，我們提出基於關鍵用語之統計計算，可以有效地估計一句子於文件中的重要性，結果顯示較以往的基於潛藏主題亂度之計算更爲準確。此論文對於多樣的計算數值都進行了隨機漫步演算法的重計算，以圖學的方法根據句子彼此間的主題相似性來傳遞重要性，此方法可以用較宏觀的面向來考慮句子的重要性。最後實驗顯示，隨機漫步演算法於自動摘要中具有很大的增益效果，在語音文件中尤其需要語意方面的資訊來彌補語音辨識上的錯誤，而在不同的課程錄音實驗下皆顯示了很好的結果，故此章對語音文件自動摘要之領域貢獻了一很好的研究方向。

第 五 章　結論與展望

Conclusions and Future Work

由於現今越來越多課程錄音的資料，而也有越來越多研究致力於課程錄音語料之處理，故利用自動化的技術擷取重要之資訊 – 關鍵用語及摘要，儼然成爲一重要的方向。

5.1　本論文主要的研究方法及貢獻

本論文考慮課程錄音的結構與特性，並對其抽取重要的資訊，如關鍵用語 (Key Term) 及摘要 (Summary) 。

在關鍵用語擷取的研究上，本論文提出分岐亂度來有效偵測重要的片語，它可以有效偵測出片語的邊界，由此可以擷取出重要的關鍵片語。接著利用了三種不同類型之特徵，包含了語音訊號中的韻律特徵、最常用的詞彙特徵，以及訓練潛藏語意分析模型並抽取語意特徵，抽取出關鍵詞。以前的研究多半只使用了文字上面的資訊，未曾考慮講者在聲音訊號上留下的特徵及資訊。本論文首度考慮語音訊號所帶來的訊息，並利用此資訊來輔助關鍵用語的擷取 [59] 。實驗顯示本論文提出之方法在課程用語之擷取上非常有效。

在自動摘要的研究上，本論文提出一圖學式的架構，在語意空間中計算每個句子的重要性 (Importance) 。此處可以避免辨識錯誤所帶來的問題，如字詞比對無法完全相同 (Exact Match) ，將字詞轉換到語意空間可以較不受辨識率的影響。而在圖學架構上的重要性估算，也可以考慮間接性的語意關連，以往研究所考慮的多半爲間接性的詞彙 (Lexical) 關連，本論文是首度嘗試間接性的語意關連來抽

取摘要 [60] 。實驗顯示此方法在自動摘要上有非常顯著之增益效果。

此外，自動摘要同時也利用了自動擷取出來的關鍵用語資訊來幫助摘要的選擇，設計一估算用語在文件中重要性的評估方法，此額外的資訊可以更準確的估計文件及主題間的機率關係，這也是本論文首度嘗試的實驗。幸運地，許多嘗試都獲得了不錯的結果，同時也驗證了這些方法可以爲語音文件的資訊擷取帶來進步。

5.2 未來展望

由於資訊擷取會因爲語料的不同而有不同的特徵，其中也會有一些只存在於課程錄音中，像是整個課程的進行流程 (如：概念陳述、數學推導...等)，若我們能預測此時所進行的流程，我們可以根據此流程來考慮用語或句子之重要性，例如：在概念陳述部分的用語或句子多半較爲重要，這些特徵可以在未來被考慮進資訊擷取的方法中。

如同第 4.7.3 節所述，有些相連的句子其主題變化較爲劇烈，不符合現實情況。我們可以在考慮一句子之潛藏主題分布時，和前後句子做一平滑化 (Smoothing) 之動作，進而更準確估計一句子之主題分布。

未來自動摘要可能須拓展到多文件摘要 (Multi-Document Summarization) ，例如針對特定的用語，由不同文件中抽取出和此用語相關的句子並形成多文件之摘要。未來應由此方向來考慮，可以讓使用者輸入特定用語，並產生相應的摘要。

語音文件資訊擷取一直是一很重要之領域，此論文提供一簡單方向給未來學者在此領域做深入之研究。

參 考 文 獻

[1] J. Glass, T. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," in *Proceedings of 8th Annual Conference of the International Speech Communication Association*, 2007.

[2] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[3] S. Young and et al, *The HTK Book (for HTK Version 3.0)*, Cambridge University, 2000.

[4] F. Liu, D. Pennell, F. Liu, and Y. Lin, "Unsupervised approaches for automatic keyword extraction using meeting transcripts," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter*, 2009.

[5] F. Liu, F. Liu, and Y. Liu, "Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion," in *Proceedings of Spoken Language Technology*, 2008.

[6] J. Zhang, H. Chan, and P. Fung, "Improving lecture speech summarization using rhetorical information," in *Proceedings of Automatic Speech Recognition and Understanding*, 2007.

[7] J. Hirschberg, "Communication and prosody: functional aspects of prosody," *Speech Communication*, vol. 36, pp. 31–43, 2002.

[8] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley Longman Publishing Co., Inc. Boston, 1999.

[9] D. R. Morrison, "PATRICIA-practical algorithm to retrieve information coded in alphanumeric," *Journal of ACM*, 1968.

[10] L. Chien, "PAT-tree-based keyword extraction for chinese information retrieval," in *Proceedings of Special Interest Group on Information Retrieval*, 1997.

[11] L. Chien, "PAT-tree-based adaptive keyphrase extraction for intelligent chinese information retrieval," in *Proceedings of Information Processing and Management*, 1999.

[12] J. Horng and C. Yeh, "Applying genetic algorithms to query optimization in document retrieval," in *Proceedings of Information Processing and Management*, 2000.

[13] C. Chang and S. Lui, "Iepad: Information extraction based on pattern discovery," in *Proceedings of the 10th International Conference on World Wide Web*, 2001.

[14] Y. Matsio and M. Ishizuka, "Keyword extraction from a single document using word co-occurence statistical information," *International Journal on Artificial Intelligence Tools*, 2003.

[15] K. Zhang, H. Xu, J. Tang, and J. Li, "Keyword extraction using support vector machine," *Advances in Web-Age Information Management*, 2006.

[16] C. Chang, H. Wang, Y. Lui, D. W, Y. Liao, and B. Wang, "Automatic keyword extraction from documents using conditional random fields," *Journal of Computational Information Systems*, 2008.

[17] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning, "Domain-specific keyphrase extraction," in *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1999.

[18] A. Hulth, J. Karlgren, A. Jonsson, H. Boström, and L. Asker, "Automatic keyword extraction using domain knowledge," in *Proceedings of Computational Linguistics and Intelligent Text Processing*, 2004.

[19] D. Li, S. Li, W. Li, W. Wang, and W. Qu, "A semi-supervised key phrase extraction approach: learning from title phrases through a document semantic network," in *Proceedings of The Association for Computational Linguistics*, 2010.

[20] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning*, MIT Press, 2006.

[21] R. Mihalcea and P. Tarau, "TextRank: bringing order into texts," in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2004.

[22] G. Erkan and D. R. Radev, "LexRank: graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.

[23] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web," Tech. Rep., Stanford Digital Library Technologies Project, 1998.

[24] Y. Ohsawa, N. E. Benson, and M. Yachida, "Keygraph: automatic indexing by co-occurrence graph based on building construction metaphor," in *Proceedings of the Advances in Digital Libraries Conference*, 1998.

[25] Y. Matsuo, Y. Ohsawa, and M. Ishizuka, "Keyworld: extracting keywords from document's small world," *Discovery Science*, 2001.

[26] Y. Freund and R.E. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of The 13th International Conference on Machine Learning*, 1996.

[27] S. Haykin, *Neural networks: a comprehensive foundation*, 3 edition, 2008.

[28] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of Uncertainty in Artificial Intelligence*, 1999.

[29] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[30] A. P. Dempster, N. M. Laird, and D. B. Robin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of Royal Statist*, 1997.

[31] S. Kong and L. Lee, "Semantic analysis and organization of spoken documents based on parameters derived from latent topics," *IEEE Transactions on Acoustics, Speech, and Language Processing*, 2010.

[32] S. Kong and L. Lee, "Improved summarization of chinese spoken documents by probabilistic latent semantic analysis (PLSA) with further analysis and integrated scoring," in *Proceedings of Spoken Language Technology*, 2006.

[33] S. Kong and L. Lee, "Improved spoken document summarization using probabilistic latent semantic analysis (PLSA)," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 2006.

[34] R. Schapire, "The boosting approach to machine learning: An overview," in *MSRI Workshop on Nonlinear Estimation and Classification*, 2001.

[35] Y.H. Kerner, Z. Gross, and A. Masa, "Automatic extraction and learning of keyphrases from scientific articles," in *Proceedings of Computational Linguistics and Intelligent Text Processing*, 2005.

[36] P. Turney, "Learning algorithms for keyphrase extraction," in *Proceedings of Information Retrival*, 1999.

[37] S. Kong, M. Wu, C. Lin, Y. Fu, and L. Lee, "Learning on demand - course lecture distillation by information extraction and semantic structuring for spoken documents," in *Proceedings of Internation Conference of Acoustics, Speech and Signal Processing*, 2009.

[38] List from Wikipedia, "List of English stop words," `http://armandbrahaj.blog.al/2009/04/14/list-of-english-stop-words`.

[39] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proceedings of New Methods in Language Processing*, 1994.

[40] Y. Huang, "Automatic key term extraction and key term graph generation from spoken documents," M.S. thesis, National Taiwan University, 2011.

[41] D.E. Knuth, *The art of computer programming: sorting and searching*, vol. 3, Mass, 1973.

[42] T. Ong, "Updateable PAT-tree approach to chinese key phrase extraction using mutual information: a linguistic foundation for knowledge management," in *Proceedings of Second Asian Digital Library Conference*, 1999.

[43] Entropic, Inc., *ESPS/waves+ with EnSig 5.3 Release Notes*.

[44] W. Lin, "Tone recognition for fluent mandarin speech and its application on large vocabulary recognition," M.S. thesis, National Taiwan University, 2004.

[45] Z. Liu, P. Li, Y. Zheng, and M. Sun, "Clustering to find exemplar terms for keyphrase extraction," in *Proceedings of ACL-AFNLP*, 2009.

[46] C. Chou, "Chinese sentence segmentation using machine learning methods," M.S. thesis, National Taiwan University, 2009.

[47] S. Hsu, "Topic segmentation on lecture corpus and its application," M.S. thesis, National Taiwan University, 2008.

[48] C. Yeh, L. Sun, C. Huang, and L. Lee, "Bilingual acoustic modeling with state mapping and three-stage adaptation for transcribing unbalanced code-mixed lectures," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 2011.

[49] L. Lee and B. Chen, "Spoken document understanding and organization," in *Special Section, IEEE Signal Processing Megazine*, 2005.

[50] J. Goldstein, M. Kantrowitz, and J. Carbonell, "Summarizing text documents: Sentence selection and evaluation metrics," in *Proceedings of ACM SIGIR Conference on R&D in Information Retrieval*, 1999.

[51] S. Banerjee and A. Rudnicky, "An extractive-summarizaion baseline for the automatic detection of noteworthy utterances in multi-party human-human dialog," in *Proceedings of Spoken Lnaguage Technology*, 2008.

[52] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in *ACM SIGIR Conference on R&D in Information Retrieval*, 2001.

[53] S. Furui, T. Kikuchi, Y. Shinnaka, and C. Hori, "Speech-to-text and speech-to-speech summarization of spontaneous speech," in *IEEE Trans. on Speech and Audio Processing*, 2004.

[54] Y. Chen, C. Chen, H. Lee, C. Chan, and L. Lee, "Improved spoken term detection with graph-based re-ranking in feature space," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 2011.

[55] W. Hsu and L. Kennedy, "Video search reranking through random walk over document-level context graph," in *Proceedings of Mutimedia International Conference*, 2007.

[56] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of the Seventh International World Wide Web Conference*, 1998.

[57] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, "Extrapolation methods for accelerating PageRank computations," Tech. Rep., Stanford University, 1998.

[58] C. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proceedings of Workshop on Text Summarization Branches Out*, 2004.

[59] Y. Chen, Y. Huang, S. Kong, and L. Lee, "Automatic key term extraction from spoken course lectures using branching entropy and prosodic/semantic features," in *Proceedings of Spoken Language Technology*, 2010.

[60] Y. Chen, Y. Huang, C. Yeh, and L. Lee, "Spoken lecture summarization by random walk over a graph constructed with automatically extracted key terms," in *Proceedings of InterSpeech*, 2011.

作 者 發 表

1. Hung-Yi Lee, Yun-Nung Chen, and Lin-Shan Lee, "**Improved Speech Summarization and Spoken Term Detection with Graphical Analysis of Utterance Similarities**," in *Proceedings of APSIPA ASC 2011*: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2011, Xian, China, October 2011.

2. Yun-Nung Chen, Yu Huang, Ching-Feng Yeh, and Lin-Shan Lee, "**Spoken Lecture Summarization by Random Walk over a Graph Constructed with Automatically Extracted Key Terms**," in *Proceedings of ICSLP InterSpeech 2011*: The 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 2011.

3. 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 *Proceedings of IEEE ICASSP 2011*: 2011 International Conference on Acoustics, Speech and Signal Processing, Prague, Czech Republic, May 2011.

4. Yun-Nung Chen, Yu Huang, Sheng-Yi Kong and Lin-Shan Lee, "**Automatic Key Term Extraction from Spoken Course Lectures Using Branching Entropy and Prosodic/Semantic Features**," in *Proceedings of IEEE SLT 2010*: 2010 IEEE Workshop on Spoken Language Technology, Berkeley, U.S.A., December 2010. (SRI Best Student Paper Award)