第 2 章 相關文獻探討

自動摘要不是一種新興觀念，在 1950 年代至 1960 年代學者們就已開始在這方面的研究。因為當時尚無較大的文字語料集，在自然語言處理上也無較為成熟的統計模型，再加上電腦的計算能力及記憶體的容量也有所限制，因此當時的研究重點在於精簡的流程處理，著重於下列技術 [Luhn 1959；Edmundson 1969] ：

- 文字所在的位置：位於重要段落的字句佔有較高的權重，如第一段或者位於標題如『簡介、目的、結論』的段落，被視為重要
- 語彙的隱含：字句中包含重要字詞，如『重要的、艱難的』等主題句
- 位置：每一段落的第一句和最後一句被視為重要

雖然上述的方法有效，然而它們非常依賴於特別的寫作格式與風格。例如利用第一段形成摘要，僅在新聞及新聞雜誌類型的文件中適用。是以本研究試圖能發展對不同文件類型皆能通用的自動摘要模型，並不專注於特定文件的寫作方式與風格；換言之，本論文希望所探討的摘要模型，能經過一些處理（訓練）進而能自動獲得這方面的資訊，如構成摘要的重要語彙。

在回顧自動摘要模型上，可以發現其技術裡的許多重要觀念來自於資訊檢索（Information Retrieval, IR），此外資訊檢索上許多成功的檢索模型，也被驗證同樣適用於自動摘要上，如向量空間模型、潛藏式語意分析模型等 [Gong and Liu 2001; 葉鎮源 2002; 何遠 2003; 黃建霖 2004; Hirohata *et al.* 2005] 。

資訊檢索處理的問題是如何依使用者的問句（Query）從大量的文件中找出相關（即符合使用者需求）的文件；而自動摘要常常假定使用者的需求為 『看看文件中的最重要的部分是什麼?』 來找出與文件最相關的字句。資訊檢索與自動摘要的比較，可由圖 2. 1 所示。

以下小節介紹幾個自動摘要中常用且來源於資訊檢索的觀念。

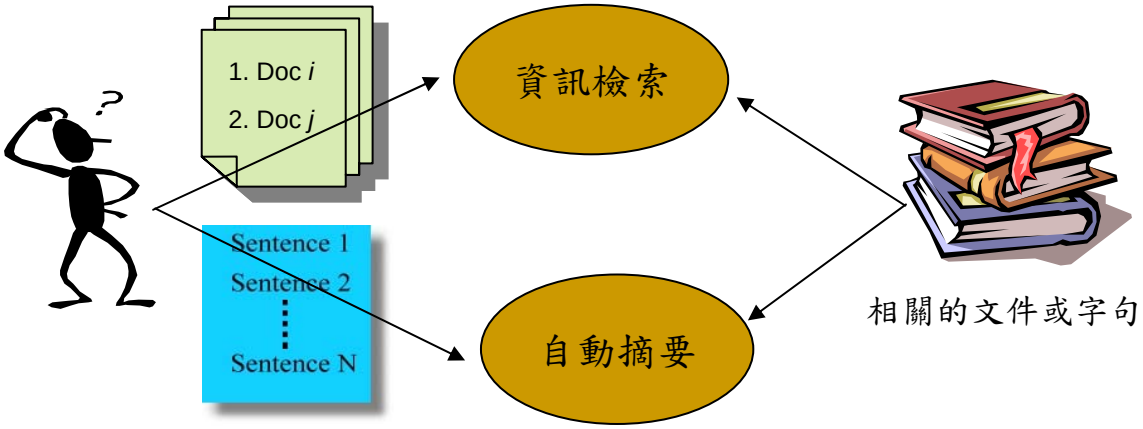

圖 2. 1 資訊檢索與自動摘要比較圖

2.1 向量空間模型（Vector Space Model, VSM）

在資訊檢索（Information Retrieval, IR）領域中，向量空間模型是個典型的檢索模型 [Baeza-Yates *et al.* 1999]。其將每一篇文件 d_j 與問句 q 視為一 T-維的向量（T 是索引特徵的總數）：

$$\vec{d_j} = (w_{1,j}, w_{2,j}, ..., w_{T,j}) \tag{2. 1}$$

$$\vec{q} = (w_{1,q}, w_{2,q}, ..., w_{T,q}) \tag{2. 2}$$

向量的權重 $w_{i,j}$ 代表索引特徵 i 在文件 d_j 的權重，其計算常使用 詞頻-反文件頻（Term Frequency-Inverse Document Frequency, TF-IDF）乘積來表示。詞頻統計其出現的頻率來決定其重要性，越常出現愈重要，其值經由正規化算出；反文件頻用以決定一索引特徵是否具有鑑別力，如一索引特徵在每一篇文件都存在（如中文：的、了），則應降低其權重，上述討論可由以下數學式來表示：

$$w_{i,j} = tf_{i,j} * idf_i = \frac{freq_{i,j}}{\max_h freq_{h,j}} * \log\frac{N}{n_i} \tag{2. 3}$$

其中

$freq_{i,j}$ ：索引特徵 i 在文件 d_j 中出現的次數

N ：文件集總數

n_i ：索引特徵 i 在文件集中出現的文件數

最後每一篇文件 d_j 與問句 q 經由估測兩向量的餘弦(Cosine)值來決定其相關性：

$$sim(d_j, q) = \frac{\overrightarrow{d_j} \cdot \vec{q}}{\left|\overrightarrow{d_j}\right| \times \left|\vec{q}\right|} = \frac{\sum_{i=1}^{T} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{T} w^2_{i,j}} \times \sqrt{\sum_{i=1}^{T} w^2_{i,q}}} \quad (2.4)$$

近年來有學者應用向量空間模型於自動摘要上，其拿整篇文件做問句（Query）去檢索文件中的每一字句，得到一相關度排名（句排名），並依摘要比例將字句摘錄出來形成摘要 [何遠 2003]。

2.2 相關評估（Relevance Measure, RM）

Gong 提出使用相關評估的方法來產生摘要 [Gong and Liu 2001]，其方法主要以向量空間模型為基礎，試圖找出文件中不同主題的重要字句為標的，其步驟如下：

1. 將文件 D 斷句，$D = \{S_1, S_2, ..S_i.., S_N\}$，這些字句 S_i 用來組成候選句 S
2. 對於每一字句 S_i 產生詞頻（Term-Frequency, TF）向量 $\overrightarrow{S_i}$，以及對於整篇文件 D 的詞頻向量 $\overrightarrow{D}$
3. 對於 S 中每一字句 S_i，估測 $\overrightarrow{S_i}$ 與 $\overrightarrow{D}$ 之間的相關分數（餘弦分數）
4. 選取最大相關分數的字句 S_k，並將其置於摘要中
5. 將 S_k 自 S 中移除，並將 S_k 中所含的字詞自文件 D 中移除；並重新計算向量 $\overrightarrow{D}$
6. 如摘要的字句達到摘要比例的量則終止運算，否則回到步驟 3 執行

本方法在步驟 4 中，選取文件中最大的相關分數的字句，代表其含有文件的主要意涵。為了使相關分數所選取到的摘要可覆蓋整篇文件的主要主題，是以在步驟 5 去除第 k 句中所含的字詞，讓接下來所選取的字句與第 k 句具有最小重覆，此傾向於所摘要的字句間具有最小的重覆。

2.3 潛藏語意分析（Latent Semantic Analysis, LSA）

潛藏語意分析 [G. Furnas *et al.* 1988; Bellegarda 2000] 是基於線性代數方法為核心的模型，包括了奇異值分解（Singular Value Decomposition, SVD）與維度約化（Dimension Reduction）兩個處理過程。LSA 的應用非常廣泛，諸如同義詞建構、判斷字詞與字句間的關係、跨語言語言模型調適（Language Model Adaptation）[KIM *et al.* 2004]、與自動摘要 [Gong and Liu 2001; 葉鎮源 2002] 等。

2.3.1 索引與字句矩陣

在進行奇異值分解之前，要將文件轉換成 索引-字句矩陣（Term-Sentence Matrix）。假設一篇文件中不同的索引字或詞有 M 個，此外文件可斷句成 N 句。

所以 索引-字句矩陣 A 的維度是 $M \times N$，矩陣中每個元素 w_{ij} 的值，可使用對數-熵（Log-Entropy）來計算 [Bellegarda 2000; Giles *et al.* 2003] ：

$$w_{ij} = l_{ij} \times g_i \tag{2. 5}$$

l_{ij} 代表索引 i 在字句 j 的對數權重，g_i 代表索引 i 的熵權重：

$$l_{ij} = \log\left(1 + f_{ij}\right) \tag{2. 6}$$

$$g_i = 1 - \varepsilon_i \tag{2. 7}$$

其中

f_{ij}：索引 i 在字句 j 中出現的次數

p_{ij}：索引 i 在字句 j 中的機率值 $= \dfrac{f_{ij}}{\sum_{j=1}^{N} f_{ij}}$

ε_i：索引 i 在字句中的正規化熵值 $= -\dfrac{1}{\log N}\sum_{j=1}^{N} p_{ij} \log p_{ij}$，即 $0 \le \varepsilon_i \le 1$。$\varepsilon_i$ 越接近 0 代表索引 i 在越少的字句中出現，越具有鑑別力

N ：字句數

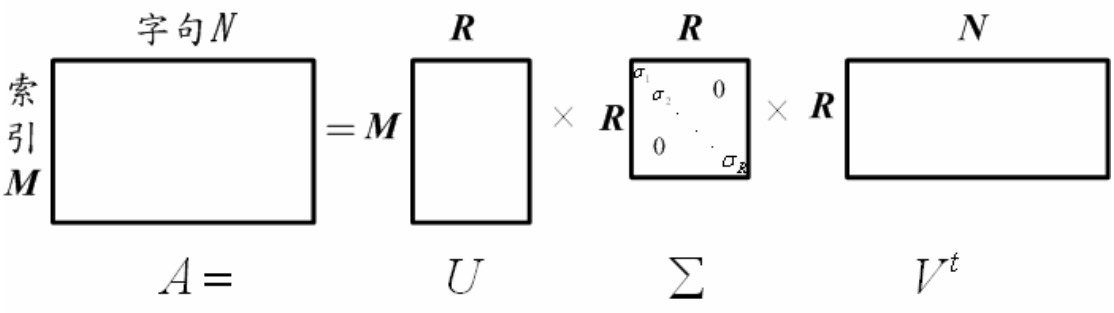

圖 2.2 奇異值分解圖示

使用 Log-Entropy 權重的方式在大部份潛藏語意分析為基礎的實驗中皆有不錯的效果 [Berry and Browne, 1999; Bellegarda 2000]。

2.3.2 奇異值分解（Singular Value Decomposition, SVD）

建立好索引-字句矩陣之後，便可進行奇異值分解：

$$A = U \sum V^T \tag{2.8}$$

其中 $\sum$ 是 $R \times R$ 維的對角奇異值矩陣，U 是 $M \times R$ 維的左奇異向量（Left Singular Vector）矩陣，V^T 是 $R \times N$ 維的右奇異向量（Right Singular Vector）矩陣，奇異值分解如圖 2.2 所示。

經過奇異值分解後，索引和字句都被投影到新的空間，稱為潛藏語意空間（LSA Space），此空間的維度是 R 維，R 小於 M 與 N。換句話說，奇異值分解透過降維的方式，將在高維度（M 與 N）內互不相關的索引與字句，投影到低維度的同一空間內，如此即可於潛藏語意空間估測其相關性。

2.3.3 潛藏語意分析摘要模型

Gong 提出應用潛藏語意分析於摘要模型上 [Gong and Liu 2001]，其方法如下：

1. 將文件 D 斷句，$D = \{S_1, S_2, ..S_i.., S_N\}$，這些字句 S_i 用來組成候選 S，設 k=1
2. 由 D 建立 索引×字句矩陣 A
3. 對 A 進行奇異值分解，在右奇異向量（Right Singular Vector）矩陣 V^t 中，

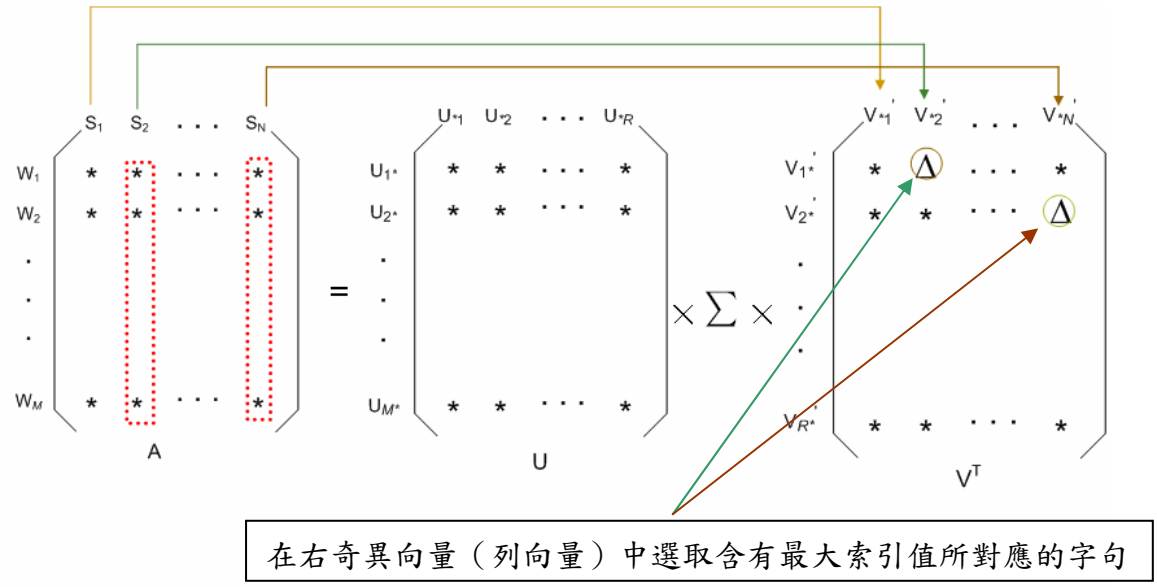

圖 2.3 潛藏語意分析摘要模型示意圖

每一字句 S_i 可由 V^t 中的行向量 $[v_{i1}, v_{i2}, ..., v_{iR}]^T$ 表示

4. 在 V^t 中選取第 k 個右奇異向量（列向量）

5. 由上述向量中，選取含有最大索引值所對應的字句，將其加入摘要中

6. 如 k 達到摘要比例的量則終止運算，否則將 k 加 1，並執行步驟 4

此方法假設，每一奇異值分別代表一概念或主題。是以奇異值所對應到的列向量（V^t 中某一列），用以描述各字句所能表達的概念或主題。因奇異值矩陣 $\sum$ 是經由遞減排序，是以第 k 個所選取的列向量，代表第 k 名重要的概念或主題，而其含有最大索引值所對應的字句就代表第 k 名重要的字句；且每一右奇異向量是相互獨立，是以所選取的字句間具有最小的重覆，如圖 2.3 所示。

A 代表原索引-字句矩陣，$\sum$ 是 $R \times R$ 維的對角奇異值矩陣、U 是 $M \times R$ 維代表索引在此語意空間的表示法、V^T 是 $R \times N$ 維代表字句在此語意空間的表示法。如在 V^T 的第 1 個右奇異向量（列向量），以第 2 個索引值為最大，是以將其所對應原始文件 D 中的第 2 句加入摘要；同理，第 2 個右奇異向量，加入第 N 句。

2.4 馬可夫模型（Markov Model）

隱藏式馬可夫模型是由馬可夫模型演變而來，根據 [Rabiner *et al.* 1989] 馬可夫

模型之相關定義，如下所示：

定理 1：若隨機過程（Stochastic Process）$\{S_t, t \geq 0\}$ 中，第 $t+1$ 的時間狀態只和第 t 的時間狀態有關，並與之前的時間狀態無關：

$$p\{S_{t+1} = s_{t+1} \mid S_0 = s_0, S_1 = s_1, ..., S_t = s_t\} = p\{S_{t+1} = s_{t+1} \mid S_t = s_t\} \quad (2.9)$$

則稱這個隨機過程為一階馬可夫鏈（First Order Markov Chain），此乃馬可夫模型中最簡單的模型。

一階馬可夫鏈在 N 個狀態下，可用三個元素來表示 (S, A, Π)

- S 表示所有狀態的集合，$S = \{s_1, s_2, ..., s_N\}$，其中 N 為狀態的個數
- $A = (a_{ij})$ 代表狀態轉移機率矩陣，$a_{ij} = p\{S_{t+1} = s_j \mid S_t = s_i\}$，$1 \leq i, j \leq N$ 表示從狀態 i 跳到狀態 j 的機率，且必須滿足 $a_{ij} \geq 0, \sum_{j=1}^{N} a_{ij} = 1$
- $\Pi = (\pi_i)$ 代表狀態初始的機率向量 $\pi_i = p(S_1 = s_i)$，$1 \leq i \leq N$ 表示在 t=1 時，狀態為 i 的機率，且需滿足 $\sum \pi_i = 1$ 的條件

若馬可夫鏈中每一時間的可能狀態均來自一有限集合 $S = \{s_1, s_2, ..., s_N\}$，則稱之為有限狀態馬可夫鏈（Finite State Markov Chain）。

定理 2：若隨機過程 $\{S_t, t \geq 0\}$ 的轉移機率 a_{ij} 不隨時間改變，也就是說滿足性質：

$$p\{S_{t+1} = s_j \mid S_t = s_i\} = p\{S_2 = s_j \mid S_1 = s_i\} = a_{ij} \quad (2.10)$$

則稱為穩定型之有限狀態馬可夫鏈（Stationary Finite State Markov Chain）。

滿足上述定理 1 與定理 2 的隨機過程即可稱之為馬可夫鏈或具有馬可夫之性質。

2.5 隱藏式馬可夫模型（Hidden Markov Model, HMM）

隱藏式馬可夫模型最早是由 Baum 和 Petrie 在 1966 年所發展出來 [Baum *et al.* 1966]，其植基於統計的機率模型，並於近十幾年來逐漸被廣泛應用，概因其擁

有豐富的數學架構及基礎能夠成功地解決所欲處理的問題。

目前，除了被廣泛應用在語音辨識（Speech Recognition）[Rabiner *et al.* 1989] 、自然語言 [Thede *et al.* 1999] 處理，甚至被應用於影像處理（Image Processing）之分析 [Aas *et al.* 1999] 與網路通訊 [Salamatian *et al.* 2001] 上。

根據 [Rabiner *et al.* 1989]對離散型隱藏式馬可夫模型之定義為：它是一個雙層隨機程序，包含了隱藏的狀態層和可觀察的輸出層；隱藏層無法直接觀察，但可從另一能產生輸出序列之輸出層觀察得出。

隱藏式馬可夫模型在 N 個狀態下，可用四個元素來表示(S,Π, A,B)

(一) 符號的表示意義如下：

- S 表示所有狀態的集合，$S=\{s_1, s_2, ..., s_N\}$，其中 N 為狀態的個數
- $V=\{v_1, v_2, ..., v_M\}$代表可觀察輸出的集合，其中 M 為所有可能輸出符號的數目
- $O=\{o_1, o_2, ..o_t.., o_T\}$表示可觀察的輸出序列，$o_t$ 代表在時間 t 下，對於任一狀態所有可能產生的觀察輸出符號，且需滿足$o_t \in V$

(二) 模型的參數為：

- $\Pi=(\pi_i)$代表狀態初始的機率向量$\pi_i = p(S_1 = s_i)$, $1 \le i \le N$ 表示在 t=1 時，狀態為 i 的機率，且需滿足$\sum_{i=1}^{N} \pi_i = 1$的條件
- $A=(a_{ij})$代表狀態轉移機率矩陣，$a_{ij} = p\{S_{t+1} = s_j \mid S_t = s_i\}$, $1 \le i, j \le N$ 表示從狀態 i 跳到狀態 j 的機率，且必須滿足$a_{ij} \ge 0, \sum_{j=1}^{N} a_{ij} = 1$
- $B=\{b_j(k)\}$ 代表可觀察輸出矩陣 $b_j(k) = p\{o_t = v_k \mid S_t = s_j\}$, $1 \le j \le N$, $1 \le k \le M$ ，表示在狀態為 j 時，v_k 的發生機率，且滿足$\sum_{k=1}^{K} b_j(k) = 1$

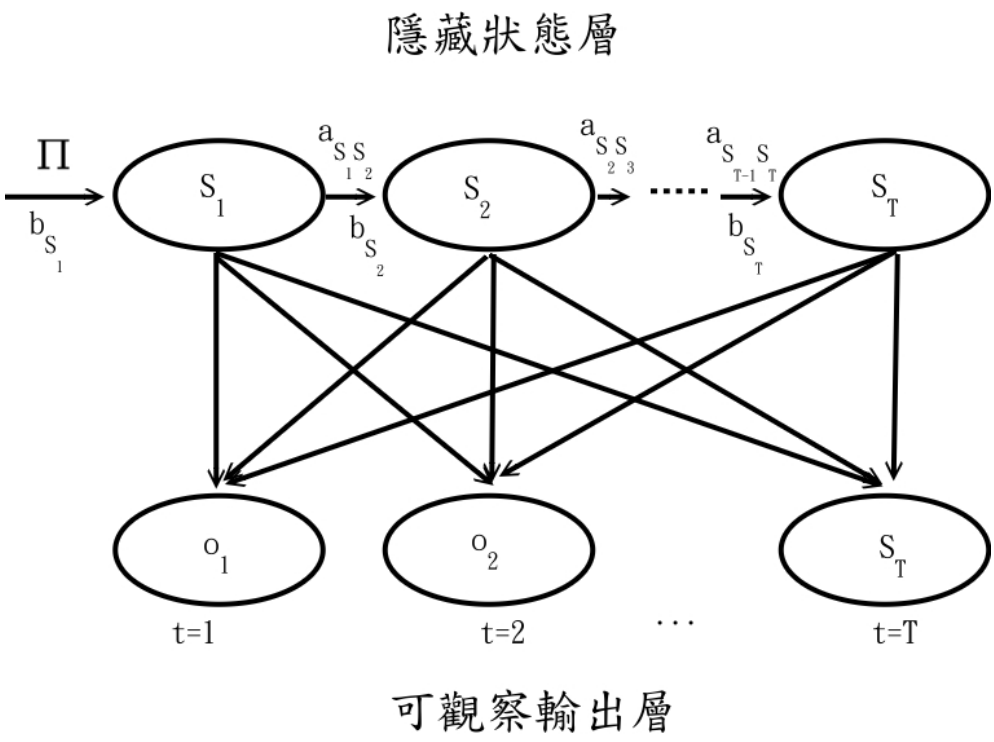

圖 2.4 隱藏式馬可夫示意圖

當適當地決定Π、A 和 B 時，隱藏式馬可夫模型的產生過程、運作方式如下：

1. 根據起始狀態機率分佈Π決定 S_1
2. 設定 t=1。
3. 由 $b_{S_t}(k)$ 的機率分佈產生 o_t
4. 由狀態轉移機率矩陣 $a_{S_tS_{t+1}}$ 的機率分佈決定 S_{t+1}
5. 設定 t=t+1，當 t<T 時回到步驟 3，否則結束程式。

上述的步驟，可由圖 2.4 所示。

2.6 統計式語言模型（Statistical Language Model, SLM）

以統計式語言模型（Statistical Language Model, SLM），來觀察字詞間可能相接的情形，已被廣泛於語音辨識器上 [Rosenfeld 2000; Siivola *et al.* 2001]，給定一長度為 n 之詞串 W，$W = w_1, w_2, ..., w_n$，要估測 W 的機率，$P(W)$，可以利用連鎖律（Chain Rule）將其分解：

$$
\begin{aligned}
P(W) &= P(w_1)P(w_2,...,w_n \mid w_1) \\
&= P(w_1)P(w_2 \mid w_1)P(w_3,...,w_n \mid w_1,w_2) \\
&= \prod_{i=1}^{n} P(w_i \mid w_1,...,w_{i-1}) \\
&= \prod_{i=1}^{n} P(w_i \mid h_i)
\end{aligned}
\tag{2. 11}
$$

其中 h_i 是詞 w_i 的歷史詞串（history），$h_i = w_1,...,w_{i-1}$。

假設 $|V|$ 為詞典大小，則式(2.11)中 $P(w_i \mid h_i)$ 的 w_i 與歷史詞串 h_i 之參數量為 $|V|^i$，此為一極其龐大的計算量而無法估測，勢必要做簡化。是以 *N*-連語言模型廣泛的被使用來處理這個問題，N 連語言模型是帶入 $N-1$ 階馬可夫模型假設，即假設詞 w_i 的出現只與其前面 $N-1$ 個詞有關聯，而與 $N-1$ 個詞以前的詞沒有關聯，所以式(2.11)可以改寫成：

$$
P(W) = \prod_{i=1}^{n} P(w_i \mid h_i) = \prod_{i=1}^{n} P(w_i \mid w_{i-N+1},...,w_{i-1}) \tag{2. 12}
$$

如三連語言模型（Tri-gram Language Model）可表示成

$$
P(W) = \prod_{i=1}^{n} P(w_i \mid w_{i-2}, w_{i-1}) \tag{2. 13}
$$

要估測式(2.13)中的 $P(w_i \mid w_{i-2}, w_{i-1})$ 可使用最大相似度估測法（Maximum Likelihood Estimation, MLE）得到：

$$
P(w_i \mid w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})} \tag{2. 14}
$$

$C(w_{i-2}, w_{i-1}, w_i)$ 與 $C(w_{i-2}, w_{i-1})$，分別為 w_{i-2}, w_{i-1}, w_i 同時出現的次數與 w_{i-2}, w_{i-1} 同時出現的次數

2.7 主題混合模型（Topical Mixture Model, TMM）

主題混合模型最早由 [Chen *et al.* 2004b; Chen 2005] 所提出並使用於語音文件檢索上。在傳統資訊檢索上，給定一使用者查詢 Query $Q = q_1 q_2 .. q_n .. q_N$，一文件 D_i 可根據其機率 $p(D_i \mid Q)$，得到相關程度的排名，經貝式定理可表示為：

$$
p(D_i \mid Q) = \frac{p(Q \mid D_i)\, p(D_i)}{p(Q)} \tag{2. 15}
$$

$p(Q|D_i)$ 是文件 D_i 產生查詢 Q 的機率，$p(D_i)$ 是文件 D_i 相關的事前機率，$p(Q)$ 是查詢 Q 的事前機率（Prior Probability）。對於所有文件來說 $p(Q)$ 是相同的且不影響文件的排名，是以可省略。於外，估計 $p(D_i)$ 的機率仍然未知，是以可進一步簡化假設 $p(D_i)$ 是均勻分佈（Uniform Distribution），也就是對於所有的文件是相同的 [Miller *et al.* 1999]。如此便可藉由 $p(Q|D_i)$ 來近似 $p(D_i|Q)$ 。

另一方面，假設查詢 $Q = q_1 q_2 .. q_n .. q_N$ 中，每個查詢項的發生互為獨立事件，因此估測 $p(Q|D_i)$ 可視為查詢 Q 中每一查詢項 q_n 於文件 D_i 機率分佈的連乘積，數學式如下：

$$p(Q|D_i) = \prod_{n=1}^{N} p(q_n|D_i) \tag{2.16}$$

在此研究中，每一篇文件 D_i 可被詮釋為混合模型（Mixture Model），模型中定義 K 個潛藏主題，各由一個主題單連語言模型（Topical Unigram）所表示，且每一潛藏主題在各文件都有不同的權重。換句話說，每一篇文件可以產生許多主題，每個主題都有相對應的單連語言模型，因此查詢 Q 與每一文件 D_i 的相關程度，可進一步改寫為：

$$p(Q|D_i) = \prod_{n=1}^{N} \sum_{k=1}^{K} p(q_n|T_k) p(T_k|D_i) \tag{2.17}$$

$p(q_n|T_k)$ 指特定潛藏主題 T_k 產生查詢項 q_n 的機率，$p(T_k|D_i)$ 是潛藏主題 T_k 在文件 D_i 的權重，且須滿足 $\sum_{k=1}^{K} p(T_k|D_i) = 1$ 的限制。總結來說，主題混合模型的主題單連語言模型，$p(q_n|T_k)$，是經由整個文件集訓練而來，且每一潛藏主題 T_k 在各文件 D_i 都有其所屬的權重，$p(T_k|D_i)$，如圖 2.5 所示。

在主題混合模型中，不同於逐字比對（Literal Term Matching），如向量空間模型，是以查詢 Q 中每一查詢項 q_n 在文件 D_i 出現的次數做計算，在主題混合模型中是以 q_n 發生在主題 T_k 與文件 D_i 產生主題 T_k 的機率來表示。是以即使查詢項並未出現在文件 D_i 中，經由主題混合模型還是可以給予 $p(Q|D_i)$ 較高的值，而達到概念比對的目的。

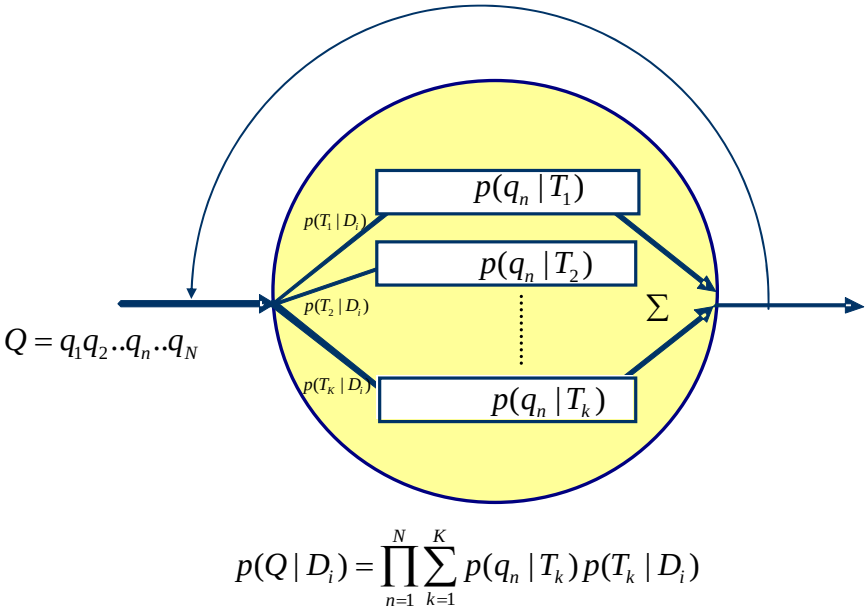

圖 2. 5 主題混合模型示意圖

2.7.1 主題混合模型訓練

在訓練時，*K*-means 演算法 [Ball and Hall 1967; Duda and Hart 1973] 被用來事先切割整個文件集為 K 個潛藏主題。因此，對於每一潛藏主題，其初始的主題單連語言模型可用主題所包含的文件來估測；而其在每一文件 D_i 的權重，可由與中心 C_k 的鄰近程度來估計，如下所示：

$$p(T_k \mid D_i) = \frac{R(\overrightarrow{D_i}, \overrightarrow{C_k})}{\sum_{r=1}^{K} R(\overrightarrow{D_i}, \overrightarrow{C_r})} \tag{2. 18}$$

其中 $R(\overrightarrow{D_i}, \overrightarrow{C_k})$ 代表利用餘弦估測文件 D_i 與中心 C_k 的距離，如下所示：

$$R(\overrightarrow{D_i}, \overrightarrow{C_k}) = \frac{\overrightarrow{D_i} \cdot \overrightarrow{C_k}}{\left\|\overrightarrow{D_i}\right\| \times \left\|\overrightarrow{C_k}\right\|} \tag{2. 19}$$

2.7.1.1 主題混合模型訓練—監督式

更進一步來說，主題單連語言模型與其在各文件的權重，可使用期望值最大化（Expectation-Maximization, EM）演算法來優化此二者的機率分佈 [Dempster *et al.* 1977]。給定一訓練集，如每一查詢 Q 均有與其相關文件的資訊，則主題混合

模型可迭代更新，利用下面三個公式：

$$\hat{p}(q_n \mid T_k) = \frac{\sum_{Q \in [TrainSet]_Q} \sum_{D_i \in [Doc]_{R\ to\ Q}} n(q_n, Q)\, p(T_k \mid q_n, D_i)}{\sum_{Q \in [TrainSet]_Q} \sum_{D_i \in [Doc]_{R\ to\ Q}} \sum_{q_s \in Q} n(q_s, Q)\, p(T_k \mid q_s, D_i)} \tag{2. 20}$$

$$\hat{p}(T_k \mid D_i) = \frac{\sum_{\substack{Q \in [TrainSet]_Q \\ st.\ D_i \in [Doc]_{R\ to\ Q}}} \sum_{q_s \in Q} n(q_s, Q)\, p(T_k \mid q_s, D_i)}{\sum_{\substack{Q \in [TrainSet]_Q \\ st.\ D_i \in [Doc]_{R\ to\ Q}}} |Q|} \tag{2. 21}$$

$$p(T_k \mid q_n, D_i) = \frac{p(T_k \mid D_i)\, p(q_n \mid T_k)}{\sum_{l=1}^{K} p(T_l \mid D_i)\, p(q_n \mid T_l)} \tag{2. 22}$$

其中，$[TrainSet]_Q$ 是查詢範例的訓練集合，$[Doc]_{R\ to\ Q}$ 是與特定查詢範例 Q 相關的文件集合，$n(q_n, Q)$ 是每一查詢項 q_n 出現在查詢範例 Q 的次數，$|Q|$ 是查詢範例 Q 的長度，$Q \in [TrainSet]_Q \ \ st.\ D_i \in [Doc]_{R\ to\ Q}$ 表示查詢範例 Q 滿足 D_i 在文件集中是與其相關的條件，$p(T_k \mid q_n, D_i)$ 是在查詢項 q_n 與文件 D_i 出現的條件下潛藏主題 T_k 發生的機率。

2.7.1.2 主題混合模型訓練—非監督式

如果訓練資料集，沒有與使用者查詢 Q 相關文件的資訊，則可將每一文件 D_i 視為與自己相關，用以訓練主題混合模型，經由簡單的更改式(2.20)-(2.22) 得到：

$$\hat{p}(q_n \mid T_k) = \frac{\sum_{D_i \in [D]} n(q_n, D_i)\, p(T_k \mid q_n, D_i)}{\sum_{D_i \in [D]} \sum_{q_s \in D_i} n(q_s, D_i)\, p(T_k \mid q_s, D_i)} \tag{2. 23}$$

$$\hat{p}(T_k \mid D_i) = \frac{\sum_{q_s \in D_i} n(q_s, D_i)\, p(T_k \mid q_s, D_i)}{|D_i|} \tag{2. 24}$$

[D]代表整個文件集，$|D_i|$ 是文件 D_i 的長度，$n(q_n, D_i)$ 是查詢項 q_n 出現在文件 D_i 的次數，$p(T_k \mid q_n, D_i)$ 是在查詢項 q_n 與文件 D_i 出現的條件下潛藏主題 T_k 發生的機率。