第3章 語言模型應用於語音辨識

對於語音辨識而言，語言模型能夠輔助辨識過程中的聲學混淆。除此之外，針對特定主題的辨識，語言模型調適更是不可或缺。本章主要回顧過去幾年，語言模型應用於語音辨識之相關研究。並著重於介紹使用不同語言資訊的語言模型。

3.1 語言模型研究

3.1.1 統計式語言模型研究方向

統計式語言模型於語音辨識的研究大抵可以分成三個方向：(1) 語料處理與使用、(2) 模型建立及改善、(3) 訓練法則及模型調適方式。首先，因為資料稀疏問題可能造成模型訓練不可靠，所以在語料方面，我們可以透過網頁搜尋的方式找出更多相關的語料，再進一步訓練語言模型[Zhu and Rosenfeld 2001]。Google Research也在 2006 年釋出Google *N*-gram英文語料，包含了極大量的單連詞到五連詞，能夠幫助各領域的研究[Google 2006]。然而，不像英文語料詞與詞之間有空白隔開，處理中文或日文等語料會遇到模糊斷詞(Segmentation Ambiguity)的問題，如何使用未經斷詞的原始語料(Raw Corpus)訓練語言模型也是一個研究的主

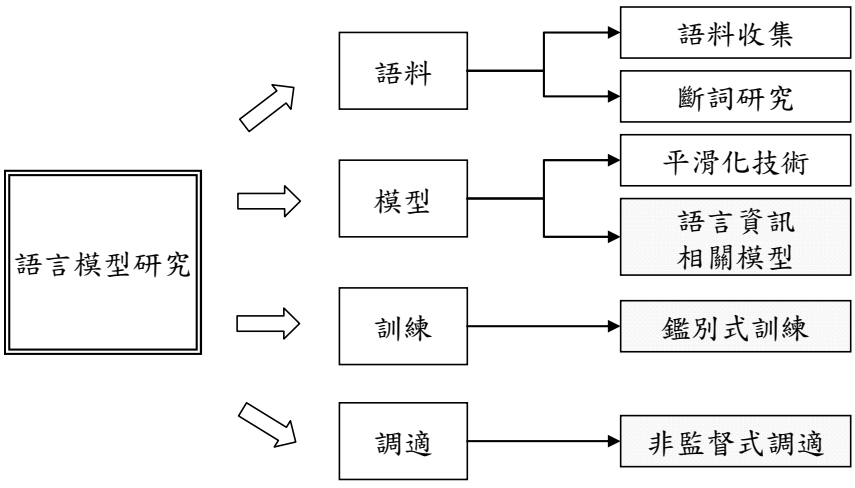


圖 3-1 語言模型研究方向

題[Mori and Takuma 2004]。模型方面，基本的模型是*N*連詞模型，早期的研究在於解決資料稀疏問題，所以有許多平滑化的技術被提出，如搭配後向法(Back-off)或插補法(Interpolation)的Katz平滑化[Katz 1987]與Modified Kneser Ney平滑化[Chen and Goodman 1999]等。

模型的主要研究方向是針對各種不同的語言資訊建立語言模型，用以補充使用傳統詞彙資訊的*N*連詞模型不足的地方，例如文件主題(Document Topics)、文件組織(Document Organization)、語句結構(Sentence Structure)及詞類別(Word Class)資訊等。過去有學者提出使用詞類別改善*N*連詞模型資料稀疏問題的*N*連類別模型(Class-based *N*-gram Model)[Brown *et al.* 1992]、針對語句結構提出的結構化語言模型(Structured Language Model)[Chelba and Jelinek 1999]、使用線性代數方法找出文件潛藏語意資訊的潛藏語意分析(Latent Semantic Analysis, LSA)[Bellegarda 2000]、使用機率架構的機率式潛藏語意分析(Probabilistic Latent Semantic Analysis, PLSA)[Glidea and Hofmann 1999; Mrva and Woodland 2004]與潛藏狄利克雷分配(Latent Dirichlet Allocation, LDA)[Tam and Schultz 2005; Mrva and Woodland 2006]等，藉由不同層次語言資訊的輔助，讓語言模型更加強健(Robust)。

傳統的訓練方式是使用最大相似度估測(Maximum Likelihood Estimation, MLE)訓練模型參數，期望正確訓練語料的相似度越大越好。而另一種作法是採

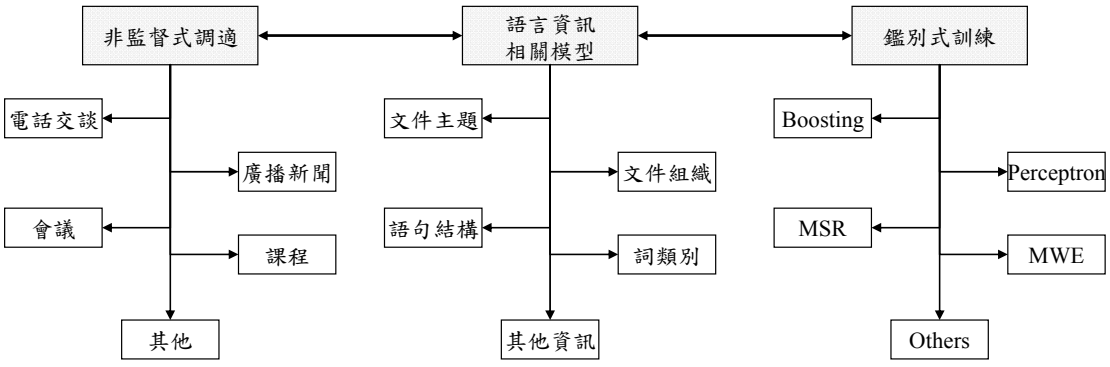


圖 3-2 語言模型研究方向(續)

用鑑別式訓練(Discriminative Training)。這類方法多以特徵(Feature)為基礎，透過減損函數(Loss Function)的定義，採用不同的鑑別式訓練找出特徵對應的參數。鑑別式訓練方式有許多，例如最小化指數型減損函數(Minimum Exponential Loss Function)的Boosting演算法[Collins 2000]、最小化平方差(Minimum Square Error, MSE)減損函數的Perceptron演算法[Roark *et al.* 2004]、最小化樣本風險(Minimum Sample Risk, MSR)法則[Gao *et al.* 2005]或是最小化詞錯誤(Minimum Word Error, MWE)法則[Kuo and Chen 2005]等等。這些方法都是先定義出特徵，如*N*連詞，並使用正確特徵資訊及與其競爭(Competing)的特徵資訊，如辨識產生的*N*最佳詞序列(*N*-best)或是詞圖(Word Graph)，透過最小化錯誤的方式，調整特徵參數。除了*N*連詞特徵外，觸發對(Trigger Pair)亦可當作特徵[Singh-Miller and Collins 2007]。值得一提的是，目前的鑑別式訓練多採用詞層次的特徵，如*N*連詞或觸發對，而少有其他類型的特徵。最主要的原因可能是因為詞特徵是最明確的一種特徵，而其它像詞類別或是語意，比較難以正確地鑑別。

在模型調適研究方面，傳統的調適方式是使用一份正確的調適語料，擷取出其資訊，加入語音辨識系統之中。近年來則著重於非監督式調適(Unsupervised Adaptation)應用於不同類型語音辨識，除了電話交談語音(Conversational Telephone Speech, CTS)外，還有廣播新聞(Broadcast News)[Chen *et al.* 2003]、課程語音(Lecture)[Niesler and Willett 2002]、會議語音(Meeting)[Tur and Stolcke 2007]等等。因為收集到的文字調適語料可能與語音測試語料相關性不夠，所以我們可以先對少部份語音語料進行辨識，得到文字結果。可想而知，這文字結果會與語音語料較相關。非監督式調適即是使用語音辨識系統輸出的轉寫文件，如*N*最佳詞序列，從中擷取資訊，再應用於語音辨識。所以，傳統的調適技術或語言模型亦可用於非監督式調適，例如使用最大事後機率概念(MAP-based)的模型插補法(Model Interpolation)與詞頻數混合法(Count Merging)[Bacchiani and Roark 2003; 蔡文鴻 2005]。需要注意的是，辨識結果可能會有錯誤，因此找出來的資訊不完全正確，如何解決辨識錯誤影響調適技術的問題亦是研究的方向之一。

3.1.2 語言資訊相關模型應用於語音辨識

語言模型應用於語音辨識系統最主要的方式是從歷史詞序列h得到資訊，進而預測目前可能的詞w，即估測機率$P(w|h)$。假設歷史詞序列h的辨識正確率不低，我們應可從中獲得有用的自然語言資訊，例如詞彙、詞類別資訊及語意主題資訊等。接下來將回顧過去學者所提出的模型與方法，並根據其利用的資訊概略地分成三類：

1. 詞相關語言模型：N 連詞模型是基本的語言模型，而為了改進其只能捕捉短距離詞彙資訊的限制，此類模型嘗試模擬二連、三連甚至更長距離的詞彙資訊，且模型複雜度不會急遽增加。
2. 詞類別相關語言模型：主要是以詞典中的詞為模型單位，嘗試預測另外一個詞的可能，類似於二連詞模型。然而，詞與詞的關係可以透過固定或非固定的詞類別建立。建立詞之間的關係後，因為歷史詞序列亦是由詞組成，因此可以進一步找出序列中的詞與欲預測詞之間的關係。
3. 文件主題相關語言模型：透過隱藏或非隱藏的參數估測，針對一篇或一群文件的主題性建立模型。非隱藏參數的模型通常是使用 N 連詞分布，而參數隱藏的模型通常是詞袋(Bag of Words)模型，即僅考慮單連詞。歷史詞序列可視為尚未完成的文件，假設完成的程度已經能呈現某些主題，透過此類模型可找出其主題性。

值得注意的是，我們雖然根據不同層次的資訊將模型分為三類，但過去學者提出的模型可能會同時滿足兩種甚至三種以上的模型特徵，也就是說能夠同時捕捉住多樣化的資訊，而提出能夠同時利用各種資訊的模型亦是我們努力的目標。

3.2 詞相關語言模型(Word-based Language Model)

N 連詞模型是詞相關語言模型中最基本的模型，然而因為馬可夫假設而僅能捕捉到短距離的資訊，所以有許多輔助或改進的模型被提出，例如快取模型(Cache Model)及略詞模型(Skipping Model)等。於此節中我們則選擇介紹觸發對語言模型(Trigger-based Language Model)及混合階層馬可夫模型(Mixed-order Markov Model)，這兩種模型分別可視為快取模型與略詞模型的延伸。

3.2.1 觸發對語言模型(Trigger-based Language Model)

為了找出長距離歷史詞序列中重要的資訊，我們可以使用觸發對(Trigger Pair)[Lau *et al.* 1993]。如果某個詞序列 A 與某個詞序列 B 在語意上或詞彙上很相關，我們稱 $A \rightarrow B$ 是一個觸發對，且 A 是觸發項，B 是被觸發項；這表示當詞序列 A 出現於某篇文章時，詞序列 B 很可能伴隨出現。為了方便統計及使用，我們通常會將詞序列大小限制為單連詞。若進一步限制觸發項 A 與被觸發項 B 為相同的詞，則可視為快取模型的一種，也就是說，快取(Cache)可視為是一種自我觸發對(Self-Trigger)。首先是觸發對選擇的準則：一般使用平均交互資訊(Mutual Information, MI)決定觸發對的相關性：

$$\begin{aligned} MI(w_j, w_i) = {} & P(w_j, w_i)\log\frac{P(w_i \mid w_j)}{P(w_i)} + P(w_j, \overline{w}_i)\log\frac{P(\overline{w}_i \mid w_j)}{P(\overline{w}_i)} \\ & + P(\overline{w}_j, w_i)\log\frac{P(w_i \mid \overline{w}_j)}{P(w_i)} + P(\overline{w}_j, \overline{w}_i)\log\frac{P(\overline{w}_i \mid \overline{w}_j)}{P(\overline{w}_i)} \end{aligned} \tag{3-1}$$

$\overline{w}$ 表示詞 w 沒有出現，例如 $P(w_j, \overline{w}_i)$ 表示詞 w_j 出現而詞 w_i 沒有出現的機率。式(3-1)亦可簡化成只考慮第一個項。除了交互資訊之外，亦有學者提出以詞頻數與反文件頻數(TF*IDF)的方式決定觸發對[Troncoso *et al.* 2004]：

$$TDIDF(w, d) = \frac{tf_{w,d}\log(N / n_w)}{\sqrt{\sum_{j=1}^{|d|}\left(tf_{j,d}\log(N / n_j)\right)^2}} \tag{3-2}$$

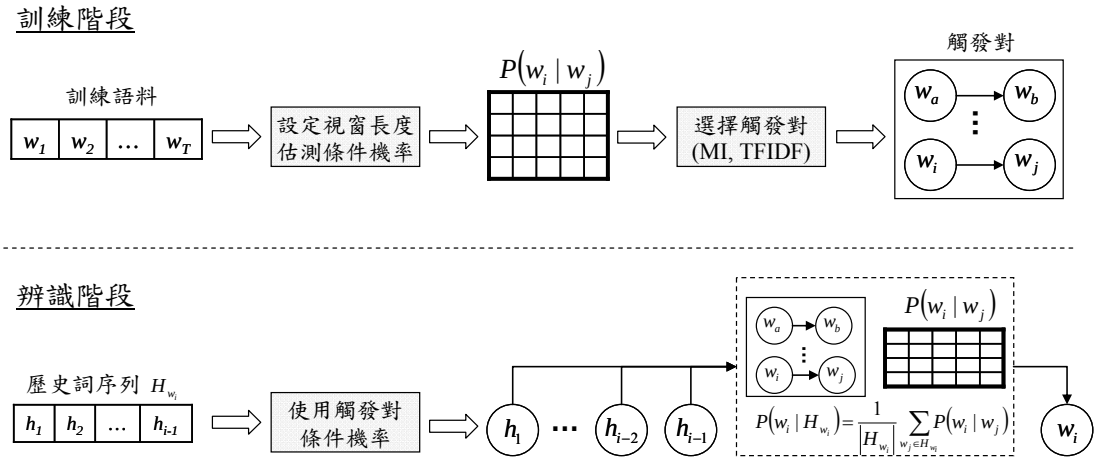


圖 3-3 觸發對語言模型使用流程圖

$tf_{w,d}$ 是詞 w 於文件 d 出現的次數，N 是訓練文件總數，n_w 是訓練文件中包含詞 w 的篇數，$|d|$ 是文件長度。由於 TFIDF 值不能直接決定詞配對重要與否，所以會設定一個門檻值保留 TFIDF 較高的詞，再對這些內容詞(Content Words)進行配對。接著估測觸發對的條件機率。先設定一個長度為 l 的視窗，然後移動並統計訓練語料視窗內的任兩個詞同時出現的次數：

$$P(w_i \mid w_j) = \frac{n_l(w_j, w_i)}{\sum_{w_l} n_l(w_j, w_l)} \tag{3-3}$$

$n_l(w_j, w_i)$ 表示訓練語料中詞 w_j 與詞 w_i 在長度為 l 的視窗內同時出現的次數。

在語音辨識過程中，觸發對可使用最大熵值法與 N 連詞等詞彙上的資訊整合，或是將歷史詞序列視為一連串詞的組成，所以觸發對語言模型可視為一連串的詞預估機率的線性組合：

$$P(w_i \mid H_{w_i}) = \frac{1}{|H_{w_i}|} \sum_{w_j \in H_{w_i}} P(w_i \mid w_j) \tag{3-4}$$

H_{w_i} 是 w_i 的歷史詞序列，$|H_{w_i}|$ 是歷史詞序列 H_{w_i} 的長度。圖 3-3 為觸發對語言模型使用流程圖，於訓練階段找出相關的觸發對及機率分布，並在測試階段使用。

3.2.2 混合階層馬可夫模型(Mixed-order Markov Model)

為了改善*N*連詞模型隨著*N*變大，模型參數量也急遽變多的缺點，混合階層馬可夫模型(Mixed-order Markov Model)被提出[Saul and Pereira 1997]。混合階層馬可夫模型在不同距離k使用不同的轉移矩陣$\mathbf{M}_k(w_{i-k}, w_i)$，不同距離的轉移矩陣視為不同階層(Order)的二連詞模型。透過轉移矩陣$\mathbf{M}_k$，能夠根據前k個詞w_{i-k}預測目前的詞w_i。轉移矩陣類似於略詞模型(Skipping Model)[Kuhn and Mori 1990]，而混合階層馬可夫模型嘗試結合這些略詞二連模型。比起*N*連詞模型的參數隨著*N*指數成長，混合階層馬可夫模型的距離k增加時，模型參數為線性成長。混合階層馬可夫模型定義成：

$$P(w_i \mid w_{i-m},...w_{i-2}, w_{i-1}) = \sum_{k=1}^{m} \lambda_k(w_{i-k}) M_k(w_{i-k}, w_i) \prod_{j=1}^{k-1} \left[1 - \lambda_j(w_{i-j})\right] \tag{3-5}$$

$\lambda_k(w_{i-k})$是詞w_{i-k}在位置k的權重，或是由詞w_{i-k}在位置k觸發目前詞的可能性；$\mathbf{M}_k(w_{i-k}, w_i)$是一個$V \times V$的轉移矩陣，能夠定義詞$w_i$與相距$k$的詞$w_{i-k}$之間的相依性，$V$是詞典大小；$m$是最遠可預測的距離，亦表示有$m$個轉移矩陣被結合。因此，從位置 1 的詞$w_{i-1}$觸發詞$w_i$的機率為在位置 1 的詞$w_{i-1}$觸發詞的可能性，乘上位置 1 的詞$w_{i-1}$觸發詞$w_i$的機率，即$\lambda_1(w_{i-1})\mathbf{M}_1(w_{i-1}, w_i)$。由位置 2 的詞$w_{i-2}$觸發詞$w_i$的機率則是由不是從位置 1 的詞$w_{i-1}$觸發的可能性，乘上位置 2 的詞$w_{i-2}$觸發詞$w_i$的機率，即$\left[1 - \lambda_1(w_{i-1})\right]\lambda_2(w_{i-2}) M_2(w_{i-2}, w_i)$，這表示，若詞$w_i$是被$w_{i-2}$所觸發，則不可能先被$w_{i-1}$觸發，依此類推。最後，再將每個位置的詞觸發目前詞的可能性加總。明顯地，$\lambda_m(w_{i-m})$需固定為 1，表示不會由超過距離m的詞觸發，如此才會滿足$\sum_{w_i} P(w_i \mid w_{i-m},..., w_{i-1}) = 1$的條件。混合階層馬可夫模型參數可以透過最大化訓練語料相似度估測：

$$L = \prod_{i=1}^{T} P(w_i \mid w_{i-m},..., w_{i-2}, w_{i-1}) \tag{3-6}$$

T是訓練語料總詞數。將式 (3-5)代入式 (3-6)並使用期望值最大化法求得參數：

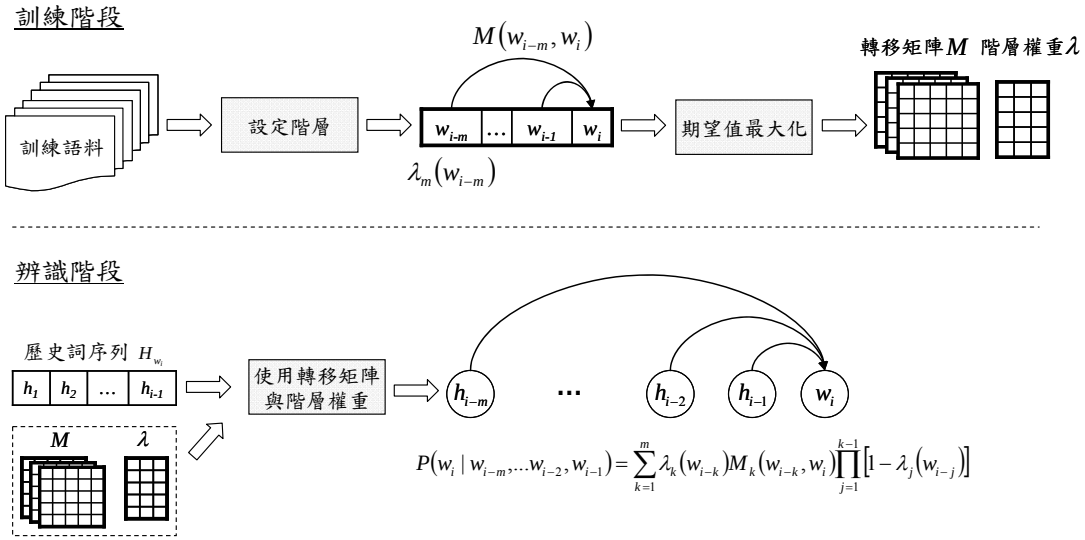


圖 3-4 混合階層馬可夫模型使用流程圖

E-step

$$\phi_k(w_i) = \frac{\lambda_k(w_{i-k}) M_k(w_{i-k}, w_i) \prod_{j=1}^{k-1} [1 - \lambda_j(w_{i-j})]}{P(w_i \mid w_{i-m},...w_{i-2}, w_{i-1})} \tag{3-7}$$

M-step

$$\lambda_k(w) = \frac{\sum_i \delta(w, w_{i-k}) \phi_k(w_i)}{\sum_i \sum_{j=k}^{m} \delta(w, w_{i-k}) \phi_j(w_i)} \tag{3-8}$$

$$M_k(w,v) = \frac{\sum_i \delta(w, w_{i-k}) \delta(v, w_i) \phi_k(w_i)}{\sum_i \delta(w, w_{i-k}) \phi_k(w_i)} \tag{3-9}$$

$\phi_k(w_i)$ 是由位置 k 觸發詞 w_i 的事後機率，$\delta(w,v)$ 是一個零一函數(Zero-One Function)，當 $w=v$ 時，其值為 1，反之為 0。初始的轉移矩陣機率分布可以使用略N詞二連語言模型(Skip-N Bigram)。圖 3-4 為混合階層馬可夫模型使用流程圖，於訓練階段求得轉移矩陣跟權重後，於辨識階段直接使用。我們可以用混合階層馬可夫模型來近似較高階層的馬可夫模型，一般而言，距離 m 亦不會太大，常設定為 2 至 5 之間。

3.3 詞類別相關語言模型(Word Class-based Language Model)

於此節中我們簡介詞類別相關語言模型，如 *N* 連類別語言模型、聚合式馬可夫模型。我們可以透過詞類別的加入，降低*N*連詞模型資料稀疏的問題。

3.3.1 *N*連類別模型(Class-based *N*-gram Model)

傳統的*N*連詞模型會遭遇到資料稀疏性的問題，且當*N*越大時，所遇到的問題越嚴重。為了改進這個缺點，*N*連類別模型(Class-based *N*-gram model)被提出[Brown *et al.* 1992]。假設語句中每一個詞都有所屬的詞類別，屬於同一個詞類別的詞可能在語意上或是文法上具有相似的意義。而如果透過對詞做分類，*N*連類別模型的參數量會大幅的減少。*N*連類別模型定義如下：

$$P(w_i \mid w_{i-n+1},...,w_{i-1}) \approx P(w_i \mid c_i)P(c_n \mid c_{i-n+1},...,c_{i-1}) \tag{3-10}$$

$P(w_i \mid c_i)$表示詞w_i在所屬類別c_i所佔的比例，即

$$P(w \mid c) = \frac{C(w)}{\sum_{w \in c} C(w)} \tag{3-11}$$

$C(w)$是詞w在語料中的總詞頻數。$P(c_i \mid c_{i-n+1},...,c_{i-1})$表示 *N* 連類別機率，且

$$P(c_i \mid c_{i-n+1},...,c_{i-1}) = \frac{C(c_{i-n+1},...,c_i)}{C(c_{i-n+1},...,c_{i-1})} \tag{3-12}$$

$C(\cdot)$表示詞類別頻數。接下來，我們可以透過最大化訓練語料相似度找出每個詞所屬的類別。以二連類別模型為例，訓練語料對數相似度可表示成：

$$\begin{aligned} \log L &= \frac{1}{T-1} \log P(w_2,...,w_T \mid w_1) \\ &\approx \sum_{w_{i-1},w_i} \frac{C(w_{i-1},w_i)}{T-1} \log[P(c_i \mid c_{i-1})P(w_i \mid c_i)] \end{aligned} \tag{3-13}$$

T 是訓練語料總詞數。根據*N*連類別模型定義，$P(w_i) \approx P(w_i \mid c_i)P(c_i)$，將式 (3-13) 重新安排後，可以得到：

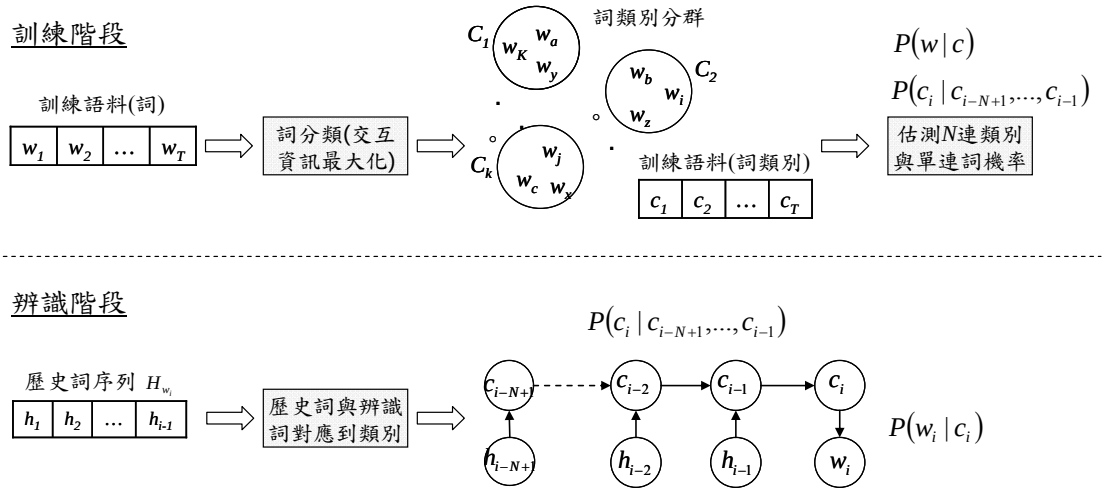


圖 3-5 *N* 連類別模型使用流程圖

$$
\begin{aligned}
&\log L \\
&= \sum_{w_{i-1},w_i} \frac{C(w_{i-1},w_i)}{T-1} \log \frac{P(c_i \mid c_{i-1})}{P(c_i)} + \sum_{w_{i-1},w_i} \frac{C(w_{i-1},w_i)}{T-1} \log P(w_i \mid c_i) P(c_i) \\
&= \sum_{c_{i-1},c_i} \frac{C(c_{i-1},c_i)}{T-1} \log \frac{P(c_i \mid c_{i-1})}{P(c_i)} + \sum_{w_i} \frac{C(w_i)}{T-1} \log P(w_i) \\
&= MI(c_{i-1},c_i) - H(w_i)
\end{aligned}
\tag{3-14}
$$

$MI(c_{i-1},c_i)$是相鄰類別c_{i-1}與c_i的交互資訊，$H(w_i)$是詞w_i的熵值。所以如果採用二連類別模型，最大化訓練語料相似度等同於最大化詞類別交互資訊(Mutual Information)。然而，目前沒有比較有效率的方式找出最佳的詞類別組合，我們可以採用貪婪演算法(Greedy Algorithm)來找出較佳的解，希望每一次將詞分到某一類別時，類別交互資訊能夠最大。然而，當詞典很大時，對詞分群的動作相當費時，所以在 2001 年亦有學者透過簡化模型的方式改進原來的分類步驟[Whittaker and Woodland 2001]。圖 3-5 是*N*連類別模型使用方式，先透過訓練語料決定每個詞可能的類別與每個類別的單連詞機率，再將歷史詞對應到所屬的類別。

3.3.2 聚合式馬可夫模型(Aggregate Markov Model)

有別於*N*連類別模型是將每個詞對應到固定的詞類別，聚合式馬可夫模型(Aggregate Markov Model, AMM)定義了柔性(Soft)的詞類別[Saul and Pereira

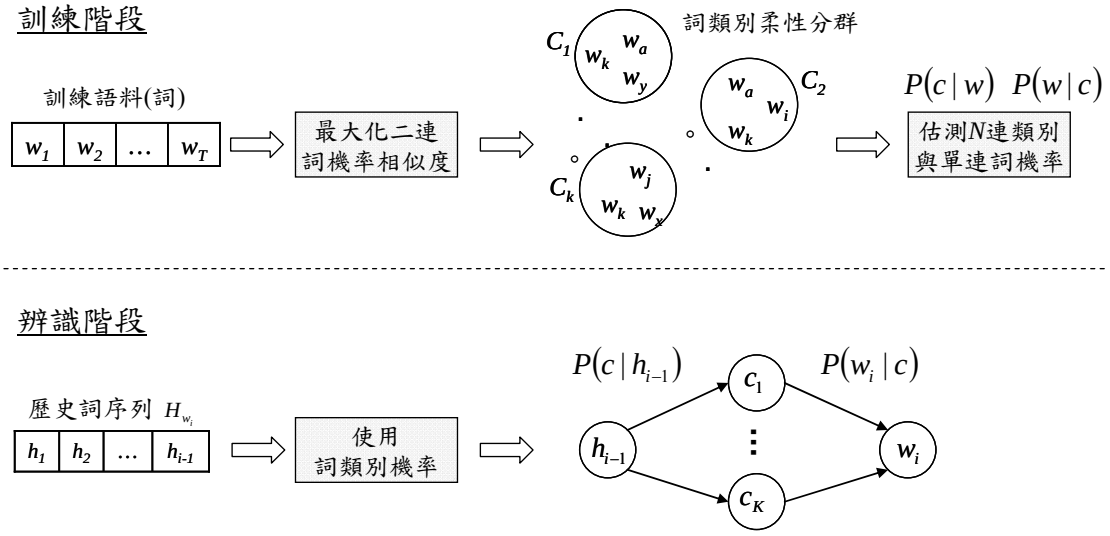


圖 3-6 聚合式馬可夫模型使用流程圖

1997]，也就是說，每個詞不只是屬於一種類別，而是可以屬於許多類別，這種多類別的定義也較符合自然語言真正的情況，例如中文裡有些某些詞可以是動詞，也可以是名詞等，所以用法及相鄰詞也會有所不同。聚合式馬可夫模型放寬了*N*連詞類別模型的限制，且原始的二連詞模型透過機率式的詞類別聚合之後，亦可以降低模型的複雜度。其模型可以表示成：

$$P(w_i \mid w_{i-1}) = \sum_{c=1}^{C} P(w_i \mid c) P(c \mid w_{i-1}) \tag{3-15}$$

C 表示所有可能的詞類別數，$P(w_i \mid c)$ 為類別 c 產生詞 w_i 的機率，$P(c \mid w_{i-1})$ 為詞 w_{i-1} 產生類別 c 的機率。所以模型複雜度由 V^2 降為 $2CV$，V 是詞典大小。所以，訓練語料對數相似度可表示成：

$$\log L = \sum_{i=1}^{T} \log P(w_i \mid w_{i-1}) = \sum_{i=1}^{T} \log \sum_{c=1}^{C} P(w_i \mid c) P(c \mid w_{i-1}) \tag{3-16}$$

T 是訓練語料總詞數，C 為可能的詞類別數。我們可以透過期望值最大化法最大化訓練語料相似度求得模型參數 $P(w_i \mid c)$ 與 $P(c \mid w_{i-1})$：

E-step:

$$P(c \mid w_{i-1}, w_i) = \frac{P(w_i \mid c)P(c \mid w_{i-1})}{\sum_{c'} P(w_i \mid c')P(c' \mid w_{i-1})} \tag{3-17}$$

M-step:

$$P(c \mid w_{i-1}) = \frac{\sum_{w} n(w_{i-1}, w)P(c \mid w_{i-1}, w)}{\sum_{c'}\sum_{w} n(w_{i-1}, w)P(c' \mid w_{i-1}, w)} \tag{3-18}$$

$$P(w_i \mid c) = \frac{\sum_{w} n(w, w_i)P(c \mid w, w_i)}{\sum_{w'}\sum_{w} n(w, w')P(c' \mid w, w')} \tag{3-19}$$

$n(w, w')$為詞w與w'的共同出現且相鄰的次數。圖 3-6 為聚合式馬可夫模型使用流程圖，可以看出其使用柔性類別建立詞與詞之間的關係，與N連類別模型相比，更有彈性。於辨識過程中，聚合式馬可夫模型只考慮單一歷史詞，即二連詞。

3.4 文件主題相關語言模型(Document Topic-based Language Model)

於此節中我們介紹幾種與文件主題相關的語言模型，如混合主題式語言模型、潛藏語意分析、機率式潛藏語意分析及潛藏狄利克雷分配。

3.4.1 混合主題式語言模型(Mixture-based Language Model)

建立及使用語言模型的過程中常會遇到的問題就是訓練語料與測試語料不一致，如果從一個特定主題的語料訓練出的模型，測試於同一個主題會有效果，用於其他主題可能沒有效果，甚至變差；如果從主題較平均的一般語料訓練出的模型，用於特定主題也往往不能突顯效果，這也是需要語言模型調適的原因。因此，混合主題式語言模型(Mixture-Based Language Model)被提出，可以呈現訓練或調適語料中不同主題性的詞分布[Clarkson and Robinson 1997]。混合主題式語言模型的建立方式是：先將訓練或調適語料以文章為單位，根據不同的主題分成K份，然後分別對每一份具有相同或相似主題的語料訓練語言模型，例如N連詞模型，最後將每個具有不同主題特色的語言模型透過不同比例的權重結合在一起，以使用三連模型為例：

$$P(w_i \mid w_{i-2}, w_{i-1}) = \sum_{j=1}^{K} \lambda_j P(w_i \mid w_{i-2}, w_{i-1}, M_j) \tag{3-20}$$

K為主題的個數，λ_j為對應主題j的權重，且$\sum_{j=1}^{K} \lambda_j = 1$，$M_j$為模型$j$的參數。

針對此模型有幾點可以探討。首先，訓練語料必須依據不同的主題分群(Clustering)，然而原始語料可能沒有明確的主題標記，所以我們可以透過分群技術得到不同主題的文章群，例如K平均分群法(K-means Clustering)。此外，由於我們對原始語料作分割，再對每一個主題語料訓練模型，分割後的語料可能會太少，所以對於每一個主題語言模型，可以進行平滑化的動作，以減少資料稀疏的問題，最後整合所有的主題語言模型時，亦可以加入原始未分割語料所訓練的語言模型，作為一般(General)語言模型平滑化。不同模型的權重可於語音辨識過程

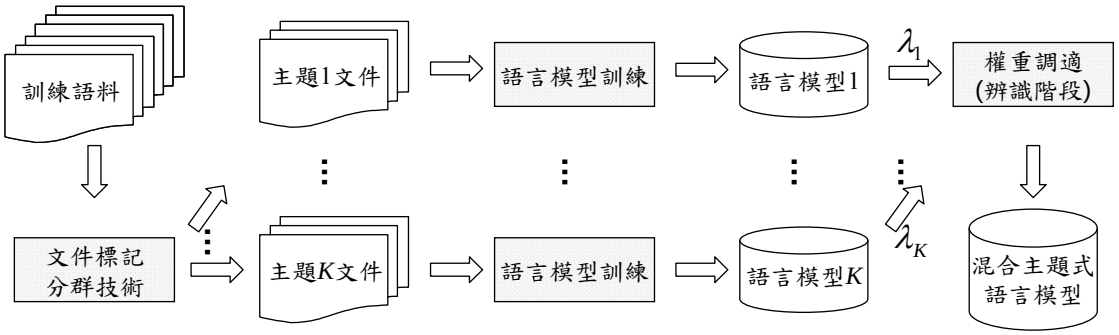


圖 3-7 混合主題式語言模型使用流程圖

中透過期望值最大化法(EM Algorithm)調整，例如針對歷史詞序列、或是語音辨識第一階段所產生的第一名(Top 1)詞序列進行最大化相似度估測，以求得更適合的權重。圖 3-7 為混合主題式語言使用流程圖。

3.4.2 潛藏語意分析(Latent Semantic Analysis)

潛藏語意分析(Latent Semantic Analysis, LSA)最早應用於資訊檢索[Deerwester *et al.* 1990]，其透過線性代數方法中的奇異值分解(Singular Value Decomposition, SVD)，將詞與文件的關係建立於某特徵空間。潛藏語意分析應用於語言模型可分為兩階段：訓練階段，包括相關性矩陣建立及奇異值分解；測試階段，包括虛擬文件摺入(Pseudo-document Folding-in)及機率估測[Bellegarda 2004]。

訓練階段：假設詞典大小為 M ，訓練文件數為 N ，則相關性矩陣 $\mathbf{W}$ 可表示成一個 $M \times N$ 矩陣，其中詞 w_i 與訓練文件 d_j 對應的元素(Element) $w_{i,j}$ 可表示成：

$$w_{i,j} = (1-\varepsilon_i)\frac{c_{i,j}}{n_j} \tag{3-21}$$

$c_{i,j}$ 表示詞 w_i 在文件 d_j 出現的次數，n_j 表示文件 d_j 的總詞數，ε_i 表示訓練語料中詞 w_i 的正規化熵值(Normalized Entropy)：

$$\varepsilon_i = -\frac{1}{\log N}\sum_{j=1}^{N}\frac{c_{i,j}}{\sum_j c_{i,j}}\log\frac{c_{i,j}}{\sum_j c_{i,j}} \tag{3-22}$$

根據定義，$0 \le \varepsilon_i \le 1$，當 $c_{i,j} = \sum_j c_{i,j}$ ，ε_i 等於 0，$c_{i,j} = \sum_j c_{i,j}/N$ 時，ε_i 等於 1。當 ε_i 接近 0，表示詞 w_i 只出現在特定文件，ε_i 接近 1 則表示詞 w_i 平均分布於各文件，所以 $1-\varepsilon_i$ 的用途類似於反文件頻數(Inverse Document Frequency, IDF)，使得兩個詞在文件中有相同詞頻數的情況下，亦能有不同的資訊量。

建立完相關性矩陣 $\mathbf{W}$ 之後可得知，對於 M 個詞而言，每個詞 w_i 可以表示成 N 維的向量 $\vec{w}_i$；總共有 N 篇文件，每一篇文件 d_j 可以表示成 M 維的向量 $\vec{d}_j$。然而，這樣的詞與文件向量表示方法有些缺點，例如詞只在部分文件中出現或是文件的詞數不多，則詞向量與文件向量的許多維度值為 0。而且其向量分屬於不同的向量空間，沒辦法直接找出詞與文件的相關性。為了解決這些問題，我們可以對相關性矩陣 $\mathbf{W}$ 進行奇異值分解(Sigular Value Decomposition, SVD)：

$$\mathbf{W} \approx \mathbf{USV}^T \tag{3-23}$$

經過奇異值分解後，可以把矩陣 $\mathbf{W}$ 分解成三個矩陣，其中 $\mathbf{U}$ 是 $M \times K$ 維的左奇異矩陣(Left Sigualar Matrix)，$\mathbf{S}$ 是奇異值對角矩陣(Sigular Diagonal Matrix)，其值為 $s_1 \ge s_2 \ge \ldots \ge s_K > 0$ ，$\mathbf{V}$ 是 $N \times K$ 維的右奇異矩陣(Right Sigular Matrix)，$K << \min(M,N)$ 是分解的維度。由於我們限制了 K 的大小，所以是一個近似的分解。$\mathbf{U}$ 與 $\mathbf{V}$ 都是行向量單範正交(Column-orthonormal)，即 $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_K$，$\mathbf{I}_K$ 是 K 維單位矩陣，而矩陣 $\mathbf{S}$ 可以視為一個潛藏語意空間(Latent Semantic Space)。所以，詞 w_i 及文件 d_j 可以用維度為 K 的向量 $\bar{u}_i$ 及 $\bar{v}_j$ 表示。如果將詞向量 $\bar{u}_i$ 或文件向量 $\bar{v}_j$ 投影到潛藏語意空間 $\mathbf{S}$，如 $\bar{u}_i\mathbf{S}$ 與 $\bar{v}_j\mathbf{S}$，可以表示詞 w_i 或文件 d_j 在潛藏語意空間 $\mathbf{S}$ 的位置。透過投影，我們可以進一步找出詞與文件的相關性。

測試階段：語音辨識過程中，歷史詞序列 h 可視為是一個未完成的文件，而

我們希望得到辨識詞 w 與歷史詞序列 h 的關聯性。要找出詞 w 與歷史詞序列 h 的關係，我們可以使用訓練語料的奇異值分解結果，求得歷史詞序列 h 在語意空間 **S** 的向量 $\vec{v}\mathbf{S}$：

$$\vec{v}\mathbf{S} = \vec{h}^T\mathbf{U} \tag{3-24}$$

$\vec{h}$ 是將歷史詞序列 h 表示成與訓練文件 $\vec{d}$ 相同形式的向量，T 是轉置，$\vec{v}$ 是新的 K 維歷史詞序列向量，**U** 與 **S** 是訓練階段所產生的奇異矩陣，這樣的方法稱為摺入(Folding-in)。直接使用訓練語料結果的前提是歷史詞序列 h 的詞用法與原來的訓練語料一致，不會影響 **U** 與 **S** 矩陣，否則需要重新建立相關性矩陣，再進行奇異值分解運算。歷史詞序列 h 可視為虛擬文件(Pseudo-document)，摺入後，相關性矩陣 **W** 多出一行向量 $\vec{h}$，右奇異矩陣 **V** 多出一列向量 $\vec{v}$。虛擬文件亦可以使用漸進(Incremental)方式求得。假設訓練語料足夠，正規化熵值 ε 不會被影響，則歷史詞序列向量 $\vec{h}$ 可表示成：

$$\vec{h} = \frac{n_h - 1}{n_h}\tilde{\vec{h}} + \frac{1-\varepsilon_i}{n_h}[0...1...0]^T \tag{3-25}$$

n_h 是目前歷史詞序列 h 的長度，$\tilde{\vec{h}}$ 是前一個歷史詞序列 $\tilde{h}$ 的向量，$[0...1...0]^T$ 是一個維度 i 為 1，其餘維度為 0 的 M 維向量，即詞 w_i 向量，T 是轉置。也就是說，目前的歷史詞序列可視為前一個歷史詞序列 $\tilde{h}$ 加上一個新詞 w_i。所以，歷史詞序列摺入可以有效率地於語意空間中更新：

$$\vec{v}\mathbf{S} = \vec{h}^T\mathbf{U} = \frac{1}{n_h}\left[(n_h - 1)\tilde{\vec{v}}\mathbf{S} + (1-\varepsilon_i)\vec{u}_i\right] \tag{3-26}$$

$\tilde{\vec{v}}\mathbf{S}$ 是前一個 K 維歷史詞序列 $\tilde{\vec{v}}$ 於空間 **S** 中的向量，$\vec{u}_i$ 是詞 w_i 在 **U** 矩陣對應的 K 維詞向量。$\vec{v}\boldsymbol{S}$ 的初始向量為 $\vec{0}$，然後隨新詞 w_i 的加入進行更新。

我們欲求得給定歷史詞序列 h，詞 w 的機率 $P(w|h)$，需要先找出詞 w 與歷

史詞序列 h 的相關性。根據 $\mathbf{W}=\mathbf{USV}^{\mathbf{T}}$，可由 $\vec{u}_i\mathbf{S}^{1/2}$ 與 $\vec{v}\mathbf{S}^{1/2}$ 兩向量的夾角值判斷，且最直接的方式是使用餘弦測量(Cosine Measure)：

$$K(w,h)=\cos\left(\vec{u}\boldsymbol{S}^{1/2},\vec{v}\boldsymbol{S}^{1/2}\right)=\frac{\vec{u}\boldsymbol{S}\vec{v}^{T}}{\left\|\vec{u}\boldsymbol{S}^{1/2}\right\|\left\|\vec{v}\boldsymbol{S}^{1/2}\right\|} \tag{3-27}$$

餘弦值介於的範圍是[-1,1]，所以通常會透過正規化方法使機率 $P(w|h)$ 介於 0 到 1 之間。

因為詞袋(Bag of Words)特性，潛藏語意分析(LSA)無法完全取代 N 連詞模型，所以需要將潛藏語意分析整合於 N 連詞模型，使其具備語意的資訊。首先假設 w_i 是辨識詞，$H_{w_i}^{(l)}$ 表示包含潛藏語意的歷史詞序列，潛藏語意語言模型的機率 $P\left(w_i\mid H_{w_i}^{(l)}\right)$，可以透過歷史詞序列摺入及餘弦測量求得。而潛藏語意語言模型與 N 連語言模型的整合如下：

$$P\left(w_i\mid H_{w_i}^{(n+l)}\right)=P\left(w_i\mid H_{w_i}^{(n)},H_{w_i}^{(l)}\right) \tag{3-28}$$

H_i 表示詞 w_i 的歷史詞序列，(n) 表示包含 N 連資訊的詞序列，(l) 表示包含潛藏語意資訊的詞序列，$(n+1)$ 表示整合資訊後的詞序列。所以可以進一步表示成：

$$\begin{aligned}P\left(w_i\mid H_{w_i}^{(n+l)}\right)&=\frac{P\left(w_i,H_{w_i}^{(l)}\mid H_{w_i}^{(n)}\right)}{\sum_{w_i\in V}P\left(w_i,H_{w_i}^{(l)}\mid H_{w_i}^{(n)}\right)}\\&=\frac{P\left(w_i\mid H_{w_i}^{(n)}\right)P\left(H_i^{(l)}\mid w_i,H_{w_i}^{(n)}\right)}{\sum_{w_i\in V}P\left(w_i\mid H_{w_i}^{(n)}\right)P\left(H_i^{(l)}\mid w_i,H_{w_i}^{(n)}\right)}\end{aligned} \tag{3-29}$$

在這邊做一個假設：給定詞 w_i 的情況下，N 連歷史詞序列 $H_{w_i}^{(n)}$ 不影響包含潛藏語意的歷史詞序列 $H_{w_i}^{(l)}$，所以整合的機率會變成：

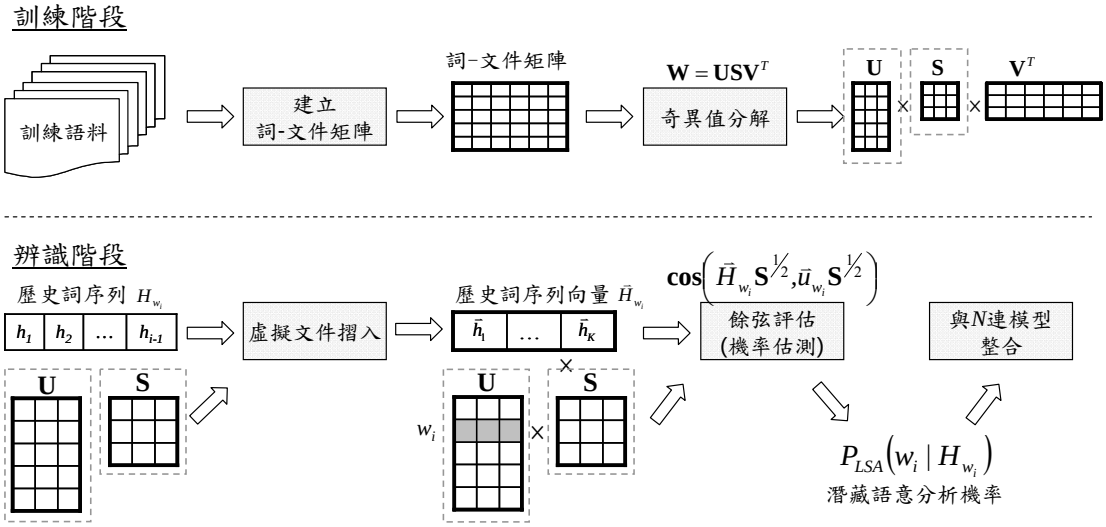


圖 3-8 潛藏語意分析使用流程圖

$$P\left(w_i \mid H_{w_i}^{(n+1)}\right)=\frac{P\left(w_i \mid H_{w_i}^{(n)}\right)P\left(H_{w_i}^{(l)} \mid w_i\right)}{\sum_{w_j \in V} P\left(w_j \mid H_{w_i}^{(n)}\right)P\left(H_{w_i}^{(l)} \mid w_j\right)} \tag{3-30}$$

再透過貝氏定理將 $P\left(H_{w_i}^{(l)} \mid w_i\right)$ 轉換成單連機率 $P\left(w_i\right)$、潛藏語意分析機率 $P\left(w_i \mid H_{w_i}^{(l)}\right)$及包含潛藏語意分析資訊的詞序列 $P\left(H_{w_i}^{(l)}\right)$，$P\left(H_{w_i}^{(l)}\right)$在分子與分母項不會改變，所以最後可以得到式子：

$$P\left(w_i \mid H_{w_i}^{(n+1)}\right)=\frac{P\left(w_i \mid H_{w_i}^{(n)}\right)\dfrac{P\left(w_i \mid H_{w_i}^{(l)}\right)}{P\left(w_i\right)}}{\sum_{w_j \in V} P\left(w_j \mid H_{w_i}^{(n)}\right)\dfrac{P\left(w_j \mid H_{w_i}^{(l)}\right)}{P\left(w_j\right)}} \tag{3-31}$$

式 (3-31)在n>1 時才有整合N連與潛藏語意資訊的意義。圖 3-8 為潛藏語意分析使用流程圖。我們可以看到，基本的潛藏語意分析在辨識階段只有使用**U**與**S**矩陣，所以後來有學者提出了一種平滑化技術的作法：將歷史詞序列與訓練語料中的文件做相似度(Similarity)的比對，找出較相似的文件，作為與歷史詞序列的相關文件集合，即對訓練時產生的**V**矩陣做進一步利用[Chien *et al.* 2004]。

3.4.3 機率式潛藏語意分析(Probabilistic Latent Semantic Analysis)

機率式潛藏語意分析(Probabilistic Latent Semantic Analysis, PLSA)，顧名思義，其概念是從潛藏語意分析(LSA)延伸而來，而不同的地方在於潛藏語意分析是將高維度的詞向量與文件向量投影到低維度的潛藏語意空間，藉此表示詞與文件的關係，而機率式潛藏語意分析則是以機率方式定義了一個生成模型(Generative Model)，透過隱藏的主題，對文件建立模型，能表示詞與文件的關係，且能呈現詞在不同主題中有一詞多義的情況[Hofmann 1999]。

機率式潛藏語意分析主要概念來自於觀點模型(Aspect Model)[Saul and Pereira 1997]。觀點模型是一個統計式混合模型(Mixture Model)，主要透過結合 K 個隱藏變數 $z_k \in \{z_1,...,z_K\}$ 的分布，達到對共同出現的事件建立模型。而機率式潛藏語意分析為表示詞與文件共同出現的觀點模型，變數 z_k 可視為是文件中隱藏的主題。我們可以透過對文件建立生成模型 M_{d_j} ，表現詞 w_i 與文件 d_j 的關係。使用生成模型產生語料的流程為：

1. 根據 $P(M_{d_j})$ 選擇一篇文件 d_j
2. 根據 $P(z_k \mid M_{d_j})$ 選擇一個隱藏變數 z_k
3. 根據 $P(w_i \mid z_k)$ 產生一個詞 w_i

$P(M_{d_j})$ 是文件 d_j 的事前機率，或是詞 w_i 會確切地出現在文件 d_j 的機率；$P(z_k \mid M_{d_j})$ 是文件 d_j 的潛藏空間機率；$P(w_i \mid z_k)$ 是主題 z_k 產生詞 w_i 的機率。表示詞 w_i 與文件 d_j 共同出現的可能性可用一個聯合機率 $P(w_i, d_j)$ 表示：

$$P(w_i, d_j) = P(M_{d_j})P(w_i \mid M_{d_j}) = P(M_{d_j})\sum_{k=1}^{K} P(w_i \mid z_k)P(z_k \mid M_{d_j}) \tag{3-32}$$

式 (3-32) 中需要加總所有主題 z_k，因為每一主題 z_k 都有一定機率產生詞 w_i。觀點模型有兩個獨立假設：第一個是給定主題 z_k ，詞 w_i 與文件 d_j 為條件獨立，第二

個是文件d_j中的詞w_i彼此獨立。所以文件d_j的生成機率可進一步表示成：

$$P(d_j \mid M_{d_j}) = \prod_{n=1}^{|d_j|} P(w_n \mid M_{d_j}) = \prod_{w_i=1}^{M} \left(\sum_{k=1}^{K} P(w_i \mid z_k) P(z_k \mid M_{d_j}) \right)^{n(w_i, d_j)} \tag{3-33}$$

$|d_j|$是文件d_j的長度，$n(w_i, d_j)$是詞w_i在文件d_j出現的次數，M是詞典大小。有別於群組模型(Clustering Model)是將文件分到某一群(主題)，再使用該群(主題)的詞分布，觀點模型讓每一文件d_j擁有所屬的主題權重$P(z_k \mid M_{d_j})$。接著我們透過對訓練語料(即每一篇文件)對數相似度最大化來找出較佳的參數：

$$\begin{aligned} \log L &= \sum_{i=1}^{M} \sum_{j=1}^{N} n(w_i, d_j) \log P(w_i, d_j) \\ &= \sum_{j=1}^{N} \left[n(d_j) \log P(M_{d_j}) + \log \sum_{i=1}^{M} n(w_i, d_j) \sum_{k=1}^{K} P(w_i \mid z_k) P(z_k \mid M_{d_j}) \right] \end{aligned} \tag{3-34}$$

M是詞典大小，N是訓練文件數，$n(w_i, d_j)$是詞w_i在文件d_j出現的次數，$n(d_j)$是文件d_j的長度，其中$P(d_j)$不包含潛藏類別z_k，最佳化時可以不考慮。然後使用期望值最大化(EM)演算法[Bilmes 1998]：

E-step：

$$P(z_k \mid w_i, M_{d_j}) = \frac{P(w_i \mid z_k) P(z_k \mid M_{d_j})}{\sum_{k'=1}^{K} P(w_i \mid z_{k'}) P(z_{k'} \mid M_{d_j})} \tag{3-35}$$

M-step：

$$P(w_i \mid z_k) = \frac{\sum_{j=1}^{N} n(w_i, d_j) P(z_k \mid w_i, M_{d_j})}{\sum_{i=1}^{M} \sum_{j=1}^{N} n(w_i, d_j) P(z_k \mid w_i, M_{d_j})} \tag{3-36}$$

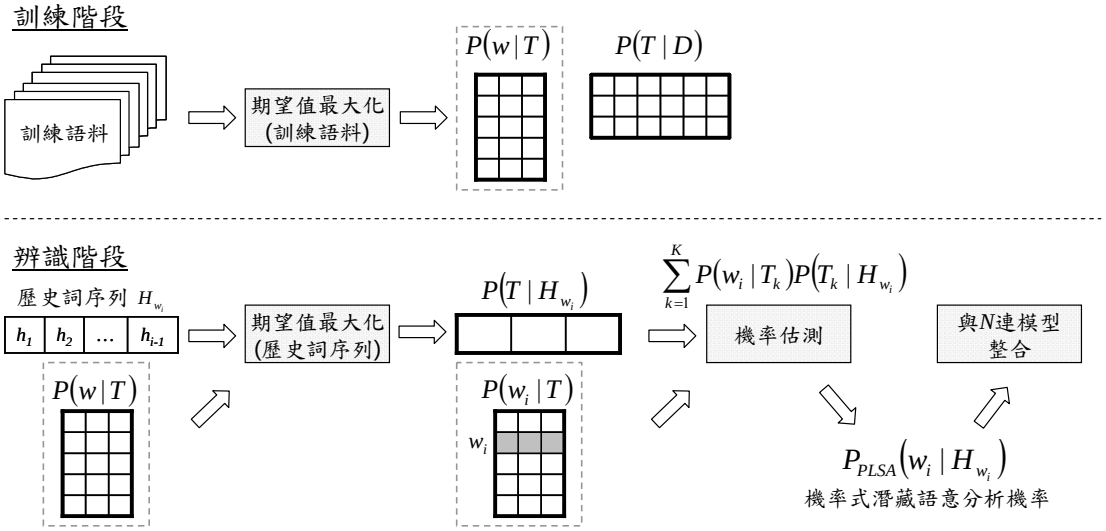


圖 3-9 機率式潛藏語意分析使用流程圖

$$P(z_k \mid M_{d_j}) = \frac{\sum_{i=1}^{M} n(w_i, d_j) P(z_k \mid w_i, M_{d_j})}{n(d_j)} \tag{3-37}$$

$P(z_k \mid w_i, M_{d_j})$是給定詞w_i與文件d_j，產生潛藏類別z_k的事後機率。

語音辨識過程中，要計算的是給定歷史詞序列H_{w_i}，預測詞w_i的機率$P(w_i \mid H_{w_i})$，而歷史詞序列H_{w_i}可以視為一篇虛擬文件，所以同樣地可以計算：

$$P_{PLSA}(w_i \mid H_{w_i}) = \sum_{k=1}^{K} P(w_i \mid z_k) P(z_k \mid H_{w_i}) \tag{3-38}$$

由於歷史詞序列H_{w_i}隨辨識過程一直在改變，所以需要即時估測。如同潛藏語意分析(LSA)的作法，假設詞於主題的機率分布$P(w_i \mid z_k)$不會隨歷史詞序列H_{w_i}改變，所以我們可以先透過最大化語料相似度訓練求得。而主題機率分布$P(z_k \mid H_{w_i})$則是透過最大化歷史詞序列相似度訓練求得，其更新式子類似式 (3-35)式與(3-37)。

有了潛藏語意分析機率$P_{PLSA}(w_i \mid H_{w_i})$之後，可以參考潛藏語意分析採用機率

規模調整法(Probablity Scaling)與*N*連詞模型整合[Mrva and Woodland 2004]：

$$\hat{P}\left(w_i \mid H_{w_i}\right)=\frac{P_{N-gram}\left(w_i \mid w_{i-N+1},\ldots,w_{i-1}\right)\dfrac{P_{PLSA}\left(w_i \mid H_{w_i}\right)}{P_{N-gram}\left(w_i\right)}}{\sum_{w_j \in V} P_{N-gram}\left(w_j \mid w_{i-N+1},\ldots,w_{i-1}\right)\dfrac{P_{PLSA}\left(w_j \mid H_{w_i}\right)}{P_{N-gram}\left(w_j\right)}} \tag{3-39}$$

或是直接使用模型插補法(Interpolation)與 N 連詞模型結合：

$$\hat{P}\left(w_i \mid H_{w_i}\right)=(1-\lambda)P_{N-gram}\left(w_i \mid w_{i-N+1},\ldots,w_{i-1}\right)+\lambda P_{PLSA}\left(w_i \mid H_{w_i}\right) \tag{3-40}$$

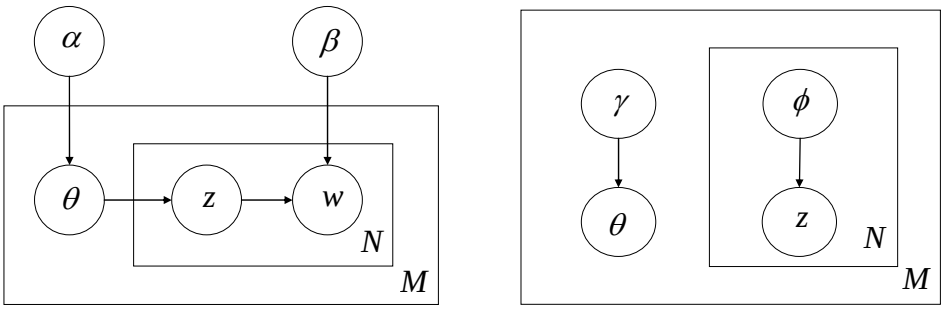
λ 是 N 連詞模型與潛藏語意分析模型的比重。

圖 3-9 是機率式潛藏語意分析使用流程圖，由於是使用期望值最大化法(EM)，所以可能會有過度符合(Overfitting)訓練語料的問題，所以有針對期望值最大化法改良的調合式期望值最大化法(Tempered Expectation Maximization, TEM)等訓練安排(Annealing Schedule)研究[Hofmann 1999]，或是引入文件事前機率參數條件並使用最大化事後機率(Maximum a Posteriori, MAP)的潛藏語意分析(MAP-PLSA)及使用近似貝氏期望值最大化法(Quasi-Bayes Expectiation Maximization)漸近地(Incrementally)使用訓練語料的近似貝氏潛藏語意分析(Quasi-Bayes PLSA, QB-PLSA)[Chien *et al.* 2005]等。

3.4.4 潛藏狄利克雷分配(Latent Dirichlet Allocation)

與機率式潛藏語意分析(PLSA)相似，潛藏狄利克雷分配(Latent Dirichlet Allocation, LDA)亦是一種語料生成模型[Blei *et al.* 2003]。潛藏狄利克雷分配透過幾個步驟來產生語料中的文件 d ：

1. 決定一件文章 d 長度為 N ，且 N 屬於波氏(Possion)分布
2. 決定一隨機變數 θ ，且 θ 屬於 k 維的狄利克雷(Dirichlet)分布，其參數為 α
3. 對於文件 d 中的詞 w_n ，根據參數為 θ 的多項式(Multinominal)分布選擇主題 z_n
4. 最後根據條件機率 $P\left(w_n \mid z_n,\beta\right)$ 產生出詞 w_n



(a) (b)

圖 3-10 圖形模型 (a) 潛藏狄利克雷分配 (b) 簡化之潛藏狄利克雷分配

β 是 $P(w_n \mid z_n)$ 的模型參數，且假設文件長度 N 與主題無關，暫不考慮。狄利克雷的分布可表示成：

$$p(\theta \mid \alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \ldots \theta_k^{\alpha_k - 1} \tag{3-41}$$

α 是 k 維向量，$\alpha_i > 0$；$\theta_i \geq 0$ 且 $\sum_{i=1}^{k} \theta_i = 1$；$\Gamma(x)$ 是 Gamma 函數。所以詞 w 與文件 d 的主題分布 θ_d 的聯合機率可表示成：

$$P(w, \theta_d \mid \alpha, \beta) = P(\theta_d \mid \alpha) \sum_{z} P(w \mid z, \beta) P(z \mid \theta_d) \tag{3-42}$$

文件 d 的邊際相似度(Marginal Likelihood)則可表示成：

$$P(d \mid \alpha, \beta) = \int P(\theta_d \mid \alpha) \prod_{n=1}^{N_d} \sum_{z} P(w \mid z, \beta) P(z \mid \theta_d) \theta_d \tag{3-43}$$

N_d 是文件 d 的長度。語料 C 的邊際相似度則可表示成：

$$P(C \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{d,n}} p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid z_{d,n}, \beta) \right) d\theta_d \tag{3-44}$$

M 是訓練文件數，$w_{d,n}$ 是文件 d 中的詞 w_n，$z_{d,n}$ 是文件 d 中詞 w_n 對應的主題。根據式 (3-44)，潛藏狄利克雷分配可表示成一個三層架構的圖形模型(Graphic Model)。如 圖 3-10 (a)所示，α 與 β 是語料層級(Corpus-level)參數，在產生語料

C 前就先決定好。θ 是文件層級(Document-level)參數，每一篇文件 d 有不同的 θ_d，而 z 與 w 則是詞層級(Word-level)參數，每篇文件 d 的每個詞 $w_{d,n}$ 及其主題 $z_{d,n}$ 參數不同。機率式潛藏語意分析(PLSA)則是一個兩層架構的圖形模型，即沒有語料層級參數 α 及 β。

接著我們會進行最大化訓練語料對數相似度的方式求取模型參數：

$$\begin{aligned}\log P(C|\alpha,\beta) &= \sum_{d=1}^{M}\log\int\sum_{z}P(\theta_d,z,d\,|\,\alpha,\beta)d\theta_d \\ &= \sum_{d=1}^{M}\log\int p(\theta_d\,|\,\alpha)\left(\prod_{n=1}^{N_d}\sum_{z_{dn}}p(z_{dn}\,|\,\theta_d)p(w_{dn}\,|\,z_{dn},\beta)\right)d\theta_d\end{aligned} \tag{3-45}$$

使用期望值最大化法時，於E-step估測給定某篇文章 d 中的詞 w_{dn}，隱藏參數 θ_d 與 z_{dn} 共同出現的事後機率 $P(\theta_d,z_{dn}\,|\,w_{dn},\alpha,\beta)$，而其分母項為詞 w_{dn} 的邊際相似度 $P(w_{dn}\,|\,\alpha,\beta)$，需要同時對參數 θ_d 積分及加總 z_{dn}。然而 z_{dn} 與參數 θ_d 有關，不易估測，所以有幾類作法，其中一種是透過蒙地卡羅取樣(Monte Carlo Sampling)逼近事後機率[Griffiths and Steyvers 2004]，一種則是透過變動性貝氏期望值最大化法(Variational Bayesian Expectation Maximization, VBEM)，進行推論(Inference)[Blei et al. 2003]。

我們採用變動性貝氏期望值最大化法訓練參數。變動性貝氏期望值最大化法主要是使用較易訓練的下界(Lower Bound)函數當作輔助函數(Auxiliary Function)，最直接的方式是使用 Jensen 不等式先找出訓練語料相似度的下界。而為了使 θ 與 z 無關，再引入變動參數(Variational Parameters) γ 與 ϕ，形成新的下界：

$$\begin{aligned}\log P(C|\alpha,\beta) &= \sum_{d=1}^{M}\log\int\sum_{z}P(\theta_d,z,d\,|\,\alpha,\beta)d\theta_d \\ &\geq \sum_{d=1}^{M}\left(\begin{array}{l}\int\sum_{z}q(\theta_d,z\,|\,\gamma,\phi)\log P(\theta_d,z,d\,|\,\alpha,\beta)d\theta_d \\ -\int\sum_{z}q(\theta_d,z\,|\,\gamma,\phi)\log q(\theta_d,z\,|\,\gamma,\phi)d\theta_d\end{array}\right)\end{aligned} \tag{3-46}$$

γ 替代 α 當作 θ 的超參數，ϕ 則為替代 β 的參數。新的下界會形成新的圖形模型，

如 圖 3-10 (b)所示，θ與z的關聯被移除。然後同樣使用期望值最大法求得每一篇文件d的最佳變動參數γ與ϕ：

$$\phi_{dni} = \frac{\beta_{idn} \exp\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{k} \gamma_j\right)\right)}{\sum_{v=1}^{V} \beta_{iv} \exp\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{k} \gamma_j\right)\right)} \tag{3-47}$$

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni} \tag{3-48}$$

β_{idn}是給定主題i，文件d的詞w_{dn}的機率，即$P(w_{dn} \mid z_i)$；ϕ_{dni}是給定主題i，文件d的詞w_{dn}的變動機率；$\Psi(\cdot)$是對數 Gamma 函數的第一階微分值，γ_i是主題i的變動超參數。得到ϕ與γ後，可以再求得原始參數α與β：

$$\beta_{ij} = \frac{\sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dni}}{\sum_{v=1}^{V} \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dni}} \tag{3-49}$$

α則需要使用牛頓-拉菲生(Newton-Raphson)演算法求得。關於變動性貝氏期望值最大化法概念可參考 附錄A 變動性貝氏期望值最大化法，參數求解及數學式推導，可參考[Blei *et al.* 2003]的附錄。

語音辨識中，潛藏狄利克雷分配使用方式類似於機率式潛藏語意分析，於辨識過程中，詞機率$P(w \mid h)$則需根據不同的歷史詞序列h求得，而β可以先從訓練語料求得。基本的作法是使用變動性貝氏期望值最大化法計算所有詞w的$\log P(w \mid h)$下界，再取指數，最後透過機率正規化求得詞機率$P(w \mid h)$。然而，因為正規化運算量過大，所以我們直接透過最大事後機率(MAP)估測θ，即：

$$P(w \mid h) = \sum_{k=1}^{K} \beta_{kw} \hat{\theta}_k \tag{3-50}$$

$$\hat{\theta}_k = \frac{\gamma_k}{\sum_{k=1}^{K} \gamma_k} \tag{3-51}$$

當辨識進行到歷史詞序列已經足夠以呈現某一主題時，參數α亦可進行更新：

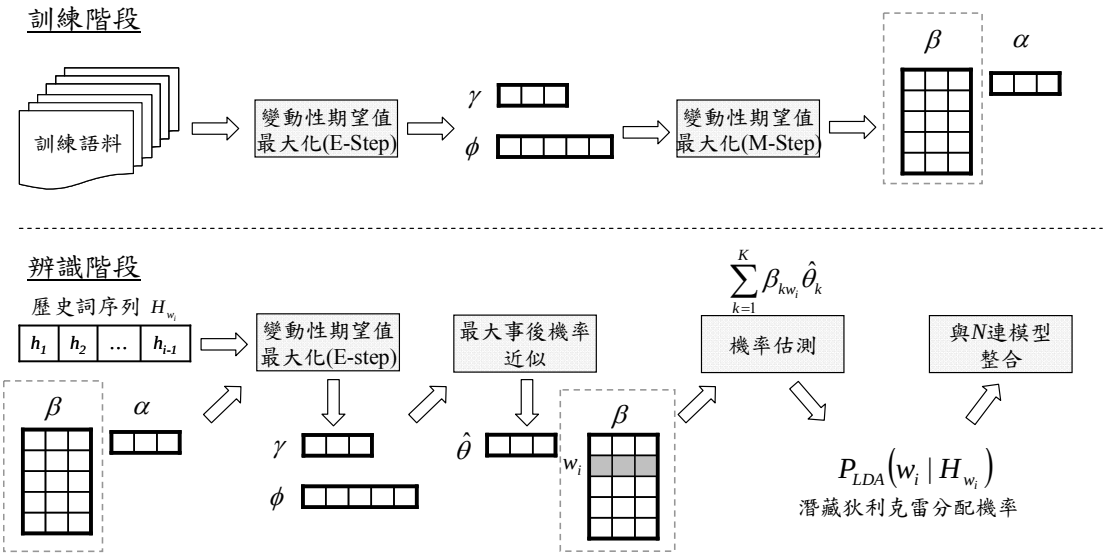


圖 3-11 潛藏狄利克雷分配使用流程圖

$$\hat{\alpha}_i = \varepsilon\alpha_i + \sum_{j=1}^{H} c_j \cdot \phi_{ij} \tag{3-52}$$

H 是歷史詞序列長度，ε 表示舊參數的影響力，c_j 表示歷史詞序列中的詞 w_j 的影響力，ε 可以經驗地調整，c_j 則可以使用信心度分數估測。圖 3-11 為潛藏狄利克雷分配使用流程圖。近年來已有同樣使用狄利克雷分布，並且延伸為更複雜的模型被提出，例如作者主題關聯模型(Author-Topic Model)[Rosen-Zvi et al. 2004]、隱藏主題馬可夫模型(Hidden Topic Markov Model)[Gruber et al. 2007]等，都是對潛藏狄利克雷分配進一步的改進。

3.5 語言資訊相關模型實驗結果

我們的調適實驗進行於辨識系統的詞圖重計分(Word Graph Rescoring)階段。我們於每一辨識詞 w_i 後返(Backtrace)最有可能的歷史詞序列 h_{w_i} ，讓不同的模型使用。調適方式是採用模型插補法(Model Interpolation)，透過線性結合方式，整合背景三連模型及調適語料訓練出來的各種語言模型。實驗分為發展集與評估集，插補參數於發展集調整至最佳，再用於評估集。

3.5.1 快取模型

快取模型(Cache Model)主要概念為：過去曾出現過的詞語，再次出現的可能性較高。於此實驗中，我們僅採用單連詞快取。我們先於辨識過程中記錄歷史詞序列，當接下來的辨識詞為歷史詞序列出現過的詞時，則增加其機率。所以使用背景語言模型並結合快取模型可以表示成：

$$\hat{P}(w \mid h) = (1-\lambda) P_{N-gram}(w \mid h) + \lambda P_{cache}(w) \tag{3-53}$$

$P_{N-gram}(w \mid h)$ 是背景語言模型機率，λ 為可調整的參數權重，$P_{cache}(w)$ 則直接透過歷史詞序列中的詞頻數估測。除此之外，亦可使用調適語料訓練出來的單連詞模型作為快取模型機率。根據歷史詞長度的不同，我們可以使用不同的比重。我們初步地將歷史詞長度分成四個情況，0~9 詞、10~19 詞、20~49 詞及 50 詞以上。表 3-1 與 表 3-2 為快取模型實驗結果，基準是背景三連詞模型結果。於字錯誤率實驗結果觀察到，快取模型的比重不能太高。這是因為歷史詞序列不夠長，估測出的機率不可靠；此外，辨識歷史詞序列不一定正確，所以亦會快取到錯誤的資訊，其效果有限，甚至比直接使用調適語料單連模型插補結果(表 2-4、表 2-5)更差。然而，語言複雜度的實驗則因為是使用正確的轉寫文字做測試，所以當比重越大時，機率越大，語言複雜度也越低。

表 3-1 使用歷史詞序列估測之快取模型實驗結果

發展集					
0~9 詞	10~19 詞	20~50 詞	50 詞以上	字錯誤率(%)	語言複雜度
基準				20.79	667.23
0.00	0.01	0.01	0.01	20.65	518.88
0.00	0.03	0.03	0.03	20.60	496.41
0.00	0.05	0.05	0.05	20.64	484.76
0.00	0.1	0.1	0.1	20.85	468.08
0.00	0.01	0.03	0.05	20.59	493.43
0.00	0.01	0.05	0.1	20.70	480.50
評估集					
基準				20.32	682.10
0.00	0.01	0.03	0.05	20.23	510.52
0.00	0.1	0.1	0.1	20.32	485.88

表 3-2 使用 SetMAT 單連詞機率之快取模型實驗結果

發展集					
0~9 詞	10~19 詞	20~50 詞	50 詞以上	字錯誤率(%)	語言複雜度
基準				20.79	667.23
0.00	0.01	0.01	0.01	20.78	578.87
0.00	0.03	0.03	0.03	20.76	577.38
0.00	0.05	0.05	0.05	20.75	576.44
0.00	0.1	0.1	0.1	20.74	575.19
0.00	0.1	0.1	0.2	20.78	575.48
0.00	0.1	0.2	0.2	20.82	575.06
0.00	0.2	0.2	0.2	20.83	575.03
0.00	0.01	0.03	0.05	20.75	577.20
0.00	0.01	0.05	0.1	20.76	576.31
評估集					
基準				20.32	682.10
0.00	0.1	0.1	0.1	20.38	581.01
0.00	0.1	0.2	0.2	20.39	580.93

3.5.2 觸發對語言模型

我們設定選擇觸發對的視窗長度為 5，並估測條件機率 $P(w_j \mid w_i)$，最後再根據交互資訊(MI)或是詞頻數反文件頻數(TFIDF)設定不同門檻值選出部分觸發對。隨著門檻值的降低，保留的觸發對也越多。表 3-3、表 3-4、表 3-5 與 表 3-6 是採用SetMAT語料的觸發對語言模型(Trigger-based Language Model)實驗結果，基準是背景三連詞模型結果。首先我們可以發現，隨著保留的觸發對數量增加，字錯誤率與語言複雜度亦下降。最佳的字錯誤率由 20.32%降為 19.76%，相對改善 2.75%；語言複雜度方面，在使用全部可能的觸發對的情況下，由 682.10 降到 427.15，相對改善 37.37%。此外，我們亦觀察到，將使用詞頻數反文件頻數與使用交互資訊的觸發對語言模型在相同觸發對個數數量級的情況下相比，效果略差。我們認為，這可能是因為使用詞頻數反文件頻數選擇出的觸發對是考慮整個文件的內容詞(Content Words)部分再進行配對，對於不同時期的新聞語料而言，內容差異較大。因此，從調適語料擷取出的觸發對可能在測試語料上不會出現。過去的學者是將詞頻數反文件頻數方式應用在自動轉寫文件及課程語音辨識上，較不易發生此現象。相較之下，交互資訊的統計資訊是在某固定窗長度內收集，所以觸發對的使用，對於調適與測試語料來說，較為一致。

表 3-3 SetMAT 之交互資訊觸發對模型於發展集結果

MI	SetMAT	
觸發對數	字錯誤率(%)	語言複雜度
基準	20.79	667.23
30,106	20.13	514.35
41,983	20.16	509.76
51,594	20.11	507.54
65,042	20.10	504.18
88,475	20.09	499.64
134,706	20.11	491.74
240,307	20.07	481.87
465,722	19.99	467.63
894,665	20.00	447.18
1,542,395	19.99	421.46
2,494,496	19.75	418.14

表 3-4 SetMAT 之詞頻數反文件頻數觸發對模型於發展集結果

TFIDF	SetMAT	
觸發對數	字錯誤率(%)	語言複雜度
基準	20.79	667.23
50,572	20.69	627.60
88,310	20.63	614.70
164,991	20.65	596.90
379,680	20.63	559.84
914,159	20.25	501.02

表 3-5 SetMAT 之交互資訊觸發對模型於評估集結果

MI	SetMAT	
觸發對數	字錯誤率(%)	語言複雜度
基準	20.32	682.10
30,106	19.69	521.91
41,983	19.67	516.95
51,594	19.64	514.58
65,042	19.62	510.40
88,475	19.63	505.68
134,706	19.61	499.79
240,307	19.64	489.66
465,722	19.76	477.08
894,665	19.76	456.61
1,542,395	19.93	428.34
2,494,496	19.77	427.15

表 3-6 SetMAT 之詞頻數反文件頻數觸發對模型於評估集結果

TFIDF	SetMAT	
觸發對數	字錯誤率(%)	語言複雜度
基準	20.32	682.10
50,572	20.27	641.75
88,310	20.30	630.61
164,991	20.21	610.48
379,680	20.07	572.72
914,159	20.03	512.48

3.5.3 混合階層馬可夫模型

表 3-7 與 表 3-8 是混合階層馬可夫模型(Mixed-order Markov Model)的實驗結果。階層數等於 1 是未平滑化的二連詞模型與背景三連詞模型插補的結果。我們可以發現，於SetET語料，當階層數增加時，效果提升。而在SetMAT語料，其效果沒有變好。然而，兩者的基準不同，因為SetET語料涵蓋的主題較廣，訓練出的二連詞模型較為一般(General)，而SetMAT語料所訓練出的二連詞模型則接近測試語料，所以基準較高。SetET於評估集的字錯誤率從 19.82%降為 19.40%，相對改進 2.11%，語言複雜度從 608.03 降為 561.72，相對改進 7.61%。而SetMAT則無進步。但是我們亦發現，兩組語料的效果是差不多的，如SetET與SetMAT於評估集的字錯誤率分別為 19.40%與 19.49%。我們認為，雖然是不同的語料，但是其語句組成架構仍是類似的。

表 3-7 SetET 與 SetMAT 之混合階層馬可夫模型於發展集結果

	SetET		SetMAT	
階層數	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
1	20.26	573.86	19.53	440.17
2	19.88	547.78	19.75	496.28
3	19.82	538.29	19.86	489.68
4	19.74	531.22	19.74	487.34

表 3-8 SetET 與 SetMAT 之混合階層馬可夫模型於評估集結果

	SetET		SetMAT	
階層數	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
1	19.82	608.03	19.23	449.10
2	19.59	578.65	19.54	504.03
3	19.40	567.93	19.49	499.02
4	19.44	561.72	19.57	495.33

3.5.4 二連類別模型

表 3-9 與 表 3-10 是二連類別模型(Class-based Bigram Model)實驗結果，基準是插補未平滑化之二連詞模型結果。於辨識過程中，我們將歷史詞 h_{i-1} 與辨識詞 w_i 對應到訓練階段就決定的詞類別 c_{i-1} 與 c_i 。再透過類別二連模型 $P(c_i \mid c_{i-1})$ 及辨識詞在詞類別的機率分布 $P(w_i \mid c_i)$ 計算模型機率，再與背景語言模型插補。我們可以發現，隨著類別數的增加，效果會越來越好。然而，其效果不如基準結果，這是因為其模型複雜度為 $C^2 + V - C$ ，C 是類別數，V 是詞典大小，$C \le V$ ，小於二連詞模型的 V^2 ，所以能夠保留的資訊有限。而因為SetMAT語料是與測試語料相同領域，詞分布 $P(w_i \mid c_i)$ 會較接近測試語料，所以結果會比SetET好。

表 3-9 SetET 與 SetMAT 之二連類別模型於發展集結果

	SetET		SetMAT	
類別數	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
基準	20.26	573.86	19.53	440.17
8	20.40	605.76	20.21	566.31
16	20.53	599.54	20.13	546.70
32	20.49	594.79	19.92	526.66
64	20.43	588.48	19.95	509.38
128	20.39	584.80	19.88	497.48
256	20.29	580.44	19.70	483.72

表 3-10 SetET 與 SetMAT 之二連類別模型於評估集結果

	SetET		SetMAT	
類別數	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
基準	19.82	608.03	19.23	449.10
8	20.11	625.70	19.86	576.96
16	20.25	620.77	19.62	555.82
32	20.10	613.75	19.61	535.45
64	20.16	608.62	19.61	520.53
128	20.04	605.76	19.52	507.03
256	20.19	602.00	19.31	493.80

3.5.5 聚合式馬可夫模型

表 3-11 與 表 3-12 是聚合式馬可夫模型(Aggregate Markov Model)的結果，基準是插補未平滑化之二連詞模型結果。我們可以發現，隨著類別數增加，效果大抵越好。而SetMAT的效果普遍較好。我們認為可能是因為SetMAT與測試語料較為相關。我們亦可以與二連類別模型的結果作比較，兩種皆為詞類別相關模型。我們可以發現，聚合式馬可夫模型的效果較好。這是因為聚合式馬可夫模型允許每個詞有不同的詞類別，而且模型數量較大的關係。聚合式馬可夫模型的模型複雜度為 $2\times C\times V$ ，二連類別模型則為 C^2+V-C ，C 是類別數，V 是詞典大小，$C\le V$ ，所以二連類別模型的參數較少，效果亦較差。

表 3-11 SetET 與 SetMAT 之聚合式馬可夫模型於發展集結果

	SetET		SetMAT	
類別數	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
基準	20.26	573.86	19.53	440.17
8	20.09	573.55	19.97	539.60
16	19.96	565.15	19.67	515.00
32	19.94	561.65	19.67	504.69
64	19.94	559.99	19.70	501.97
128	19.82	558.66	19.79	499.30
256	19.73	554.90	19.85	498.48

表 3-12 SetET 與 SetMAT 之聚合式馬可夫模型於評估集結果

	SetET		SetMAT	
類別數	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
基準	19.82	608.03	19.23	449.10
8	19.74	594.39	19.57	548.41
16	19.65	582.08	19.43	528.17
32	19.71	584.74	19.45	515.84
64	19.78	581.17	19.39	510.57
128	19.73	583.93	19.70	509.13
256	19.67	582.30	19.56	505.76

3.5.6 混合主題式語言模型

表 3-13 與 表 3-14 是混合主題式語言模型(Mixture-based Language Model)的結果。我們首先使用*K*平均值(*K*-means)演算法將調適語料文件分群。每一群再個別訓練*N*連詞語言模型，且每個*N*連詞模型都經過Katz平滑化。主題數等於 1 表示未分群調適語料的平滑化*N*連詞模型，而大於 1 的主題數的結果皆不使用未分群語料所訓練的語言模型(一般語言模型)，即不包含主題數為 1 的語言模型。混合主題式語言模型的結合參數則在辨識過程中動態調整。我們可以觀察到，語料經分群後，效果提升，且大抵隨分群數越多，效果越好。於評估集中，使用SetMAT的字錯誤率從 19.23%降為 18.80%，相對改進 2.23%，語言複雜度從 426.59 降為 367.12，相對改進 13.94%。我們亦嘗試使用未平滑化的模型以及加入一般語言模型。然而，使用未平滑化的模型效果會變差。而SetET加入一般語言模型則會讓效果變差，SetMAT加入一般語言模型則會接近只使用一般語言模型的效果。我們認為，在SetET語料中，不同主題性語言模型的特性可能會被一般語言模型所遮蔽；而在SetMAT語料中，則是因為分群後語料較少，所訓練出的模型不可靠，所以加上一般語言模型可以補償效果。

表 3-13 SetET 與 SetMAT 之混合主題式語言模型於發展集結果

主題數	SetET		SetMAT	
二連	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
1	19.89	540.10	19.51	439.16
2	19.73	485.23	19.79	394.74
3	19.55	485.74	19.63	396.90
4	19.55	491.84	19.63	396.06
8	19.39	490.76	19.40	398.31
16	19.36	493.10	19.42	403.08
三連	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
1	19.65	507.30	19.46	426.59
2	19.46	442.97	19.62	381.95
3	19.44	438.56	19.58	380.49
4	19.45	438.00	19.53	379.74
8	19.25	417.50	19.41	371.96
16	19.22	405.26	19.41	367.12

表 3-14 SetET 與 SetMAT 之混合主題式語言模型於評估集結果

主題數	SetET		SetMAT	
二連	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
1	19.65	576.04	19.23	447.55
2	19.61	514.01	19.09	400.09
3	19.62	513.33	18.99	401.72
4	19.66	518.19	19.24	401.53
8	19.46	517.31	18.94	405.00
16	19.32	518.51	18.82	410.60
三連	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
1	19.29	544.04	19.23	434.46
2	19.14	473.00	19.12	388.00
3	19.26	472.55	19.06	385.90
4	19.31	468.41	19.17	384.26
8	19.21	451.26	18.95	377.64
16	19.08	437.24	18.80	372.26

3.5.7 潛藏語意分析

潛藏語意分析(Latent Semantic Analysis, LSA)主要採用機率調整方式整合背景模型與調適語意機率，其中可設定一個權重α來調整語意機率的影響程度：

$$\hat{P}\left(w_i \mid H_{w_i}\right)=\frac{P_{N-gram}\left(w_i \mid H_{w_i}\right)\left(\frac{P_{LSA}\left(w_i \mid H_{w_i}\right)}{P_{N-gram}\left(w_i\right)}\right)^{\alpha}}{\sum_{w_j \in V} P_{N-gram}\left(w_j \mid H_{w_i}\right)\left(\frac{P_{LSA}\left(w_j \mid H_{w_i}\right)}{P_{N-gram}\left(w_j\right)}\right)^{\alpha}} \tag{3-54}$$

α越大，表示語意資訊的比重越大。經過實驗，我們將α設定為 0.08。表 3-15 是潛藏語意分析於發展集實驗結果。令人意外的是，其效果並不如預期的好，甚至比基準結果還差。我們認為，主要的原因是在建立詞-文件矩陣時，其中的元素包含了類似反文件頻數的 1-正規化熵值。我們希望能夠呈現的調適語料詞分布，會與 1-正規化熵值的分布衝突，例如幾乎每一篇新聞報導結尾都會出現記者姓名或用詞，其正規化熵值較高，經過 1-正規化熵值的運算之後，反而會被認為是沒有用的資訊。加上採用機率調整方式來估測機率，更容易被語意機率所影響。而各維度間並無明顯差異，我們認為是因為詞-文件矩陣經過奇異值分解後，詞向量的元素值差異會很小，計算出的餘弦值差異不會太大，也因此語意機率也很相似。

表 3-15 SetET 與 SetMAT 之潛藏語意分析於發展集結果

	SetET		SetMAT	
維度	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
基準	20.79	667.23	20.79	667.23
25	20.80	659.55	20.89	659.20
50	20.78	659.86	20.89	659.62
75	20.81	659.89	20.89	659.87
100	20.81	660.02	20.90	660.12
125	20.81	660.14	20.90	660.28
150	20.81	660.24	20.90	660.41

3.5.8 機率式潛藏語意分析

表 3-16 與 表 3-17 是使用不同主題數的機率式潛藏語意分析(Probabilistic Latent Semantic Analysis, PLSA)及與調適語料之N連詞模型比較的結果。我們先從調適語料訓練詞w於隱藏主題T的機率分布$P(w|T)$，然後於辨識過程中動態調整歷史詞序列主題權重$P(T|H)$。我們可以發現，隨著隱藏主題數的增加，效果會越來越好，而在 256 個主題數時有收斂的現象。在SetET中，評估集最佳的字錯誤率為 128 主題數的 19.20%，而三連詞模型為 19.34%，相對改進為 0.72%，而SetMAT中，機率式潛藏語意分析模型的表現不如三連詞模型。我們亦可從語言複雜度觀察到此問題。SetET於評估集的語言複雜度在 128 主題數能夠比三連詞模型低，而SetMAT則仍高於三連詞模型。SetET比SetMAT效果來的好，我們認為，因為SetET語料較大，訓練出來的詞分布$P(w|T)$較佳，有助於線上估測主題權重$P(T|H)$。

潛藏語意分析模型與三連詞模型相比，改進不大，這是因為潛藏語意分析模型本身是一個詞袋(Bag of Words)模型，沒辦法捕捉區域性規則(Local Regularity)。若與單連詞或二連詞模型比較，較為合理。評估集中，與單連詞模型相比，SetET 的字錯誤率由 20.09%降為 19.20%，相對改進為 4.43%，語言複雜度由 606.31 降為 531.51，相對改進 12.33%；與二連詞模型相比，SetET 的字錯誤率由 19.65%降為 19.20%，相對改進為 2.29%，語言複雜度由 576.04 降為 531.51，相對改進 7.73%。所以，若與單連詞或二連詞模型相比，潛藏語意分析模型的確利用了更長的歷史詞序列主題資訊。

表 3-16 SetET 與 SetMAT 之機率式潛藏語意分析於發展集結果

	SetET		SetMAT	
主題數	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
8	20.17	579.35	20.15	549.11
16	19.81	558.49	20.13	540.52
32	19.77	543.26	20.06	533.07
64	19.73	530.02	19.99	527.82
128	19.54	514.10	19.95	519.71
256	19.54	502.14	19.95	515.17
單連	20.34	606.31	20.31	574.31
二連	19.89	540.10	19.51	439.16
三連	19.65	507.30	19.46	426.59

表 3-17 SetET 與 SetMAT 之機率式潛藏語意分析於評估集結果

	SetET		SetMAT	
主題數	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
8	19.71	606.51	19.76	563.70
16	19.57	585.44	19.77	554.07
32	19.63	575.07	19.60	545.14
64	19.35	555.98	19.71	539.61
128	19.20	540.08	19.55	533.29
256	19.29	531.51	19.48	526.56
單連	20.09	626.33	19.93	586.32
二連	19.65	576.04	19.23	447.55
三連	19.34	544.04	19.23	434.46

3.5.9 潛藏狄利克雷分配

表 3-18 與 表 3-19 是潛藏狄利克雷分配(Latent Dirichlet Allocation, LDA)的結果。於實驗中，我們根據歷史詞序列動態地調整狄利克雷分布參數α，參數α可視為從歷史詞序列中快取到的每一個主題的重要性，透過對其設定，能夠使主題分布能夠與辨識語料相符。於實驗中觀察到，隨著主題數的增加，效果也越好，而在主題數為 256 時，有收斂的現象。然而，若與機率式潛藏語意分析比較，我們發現其效果於低主題數時較好，而在高主題數時較差。我們認為，主要的原因在於參數α影響，對於潛藏狄利克雷分配而言，主題權重的調整受到參數α的限制。而歷史詞序列的辨識錯誤或語句主題的轉換會讓參數α產生偏差，進而影響主題分布，在主題數較多時，影響更明顯。

表 3-18 SetET 與 SetMAT 之潛藏狄利克雷分配於發展集結果

	SetET		SetMAT	
主題數	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
8	20.13	573.80	20.13	547.88
16	19.86	553.40	20.06	537.40
32	19.73	540.16	19.98	527.21
64	19.74	528.16	19.94	526.48
128	19.79	526.47	20.00	522.62
256	19.63	541.43	20.04	533.94

表 3-19 SetET 與 SetMAT 之潛藏狄利克雷分配於評估集結果

	SetET		SetMAT	
主題數	字錯誤率(%)	語言複雜度	字錯誤率(%)	語言複雜度
8	19.73	599.62	19.80	561.13
16	19.73	580.53	19.83	549.46
32	19.44	571.55	19.73	538.86
64	19.34	555.56	19.46	537.35
128	19.27	550.36	19.57	535.78
256	19.39	571.67	19.50	546.70

3.6 本章結論

我們於本章中簡介了過去學者所提出的語言模型，並將其應用於語音辨識之中。我們將模型略分為幾類：詞相關語言模型(Word-Based Language Model)、詞類別相關語言模型(Word Class-Based Language Model)及文件主題相關語言模型(Document Topic-Based Language Model)，而使用潛藏主題來建立語言模型是一個趨勢。由實驗結果可以發現，除了傳統的詞模型，文件主題相關語言模型的確有很大的幫助。表 3-20 是各種模型的分析，n 是使用的N連數，V 是詞典大小，m 是所使用的階層數，C 是可能的詞類別數，T 是可能的主題數，D 是訓練文件數。值得注意的是，潛藏狄利克雷分配在更新參數時仍需要使用額外的變動參數空間。

表 3-20 各種模型之分析

	模型層次	模型使用	模型複雜度
N 連詞模型	詞	事先訓練	V^n
(單連)觸發對模型	詞	事先訓練	V^2
混合階層馬可夫模型	詞	事先訓練	$m \times V^2$
N 連類別模型	詞類別	事先訓練	$C^n + V - C$
聚合式馬可夫模型	詞類別	事先訓練	$V \times C + C \times V$
混合主題式語言模型	文件主題	需即時調適	$V^n \times T$
潛藏語意分析	文件主題	需即時調適	$V \times T + T + T \times D$
機率式潛藏語意分析	文件主題	需即時調適	$V \times T + T \times D$
潛藏狄利克雷分配	文件主題	需即時調適	$V \times T + T$