Goal-Oriented SOM for Document Clustering

Student: Pei-Yuan Hsieh Advisor: Dr. Hao-Ren Ke, Dr. Wei-Pang Yang.

Institute of Computer and Information Science
National Chiao Tung University

ABSTRACT

In this thesis, a Goal-Oriented Self-Organizing Map (GOSOM) is proposed to cluster documents according to user's goals. GOSOM is motivated by Self-Organzing Map (SOM) model and allows the user to specify what kinds of results should be clustered. The specified goals are analyzed by Latent Sematic Analysis (LSA) to determine their relationships to input vectors. GOSOM properly enhances the features of input vectors when caculating similarity in the clustering process; in this manner, GOSOM is capable of guiding the clustering result toward user's goals. Additionally, a weighted majority voting algorithm is provided to label the clustering result with respect to the specified goals. Furthermore, GOSOM presents a user relevance feedback mechanism to improve the performance of clustering. A system called Goal-Oriented Document Clustering system (GODOC) is implemented to verify that GOSOM is superior to convensional SOM. Experiment results show that GOSOM significantly improve 21.67% in accuracy, 28.47% in recall.

Keywords: Self-Organizing Map, Latent Semantic Analysis, Document Clustering, Goal-Oriented

目標導向之 SOM 應用於文件分群

研究生: 謝佩原　　指導教授: 柯皓仁博士，楊維邦博士

國立交通大學資訊科學研究所

摘 要

在這篇論文中，我們提出一目標導向之 SOM (Goal-Oriented Self-Organizing Map, GOSOM) 來將文件依使用者的目標分群。GOSOM 是基於 Self-Organzing Map (SOM) 加以改良，可讓使用者指定想要的分群結果種類。使用者指定的目標是透過潛在語意分析 (Latent Semantic Analysis, LSA) 方法，來分析其與輸入向量 (Input Vector) 的關係。GOSOM 適當地加強了輸入向量的特徵，以致於在分群過程中計算相似度時，可將分群結果導向使用者想要的目標。此外，我們也提出一個權重的多數決 (Weighted Majority Voting) 方法，將分群結果以使用者的觀點作適當標記。最後，GOSOM 提供了一個使用者相關回饋 (User Relevance Feedback) 的機制，以改善分群的結果。我們實作了「目標導向文件分群系統」(Goal-Oriented Document Clustering system, GODOC) 來驗證 GOSOM 優於傳統的 SOM 模型。實驗結果證明，相較於傳統的 SOM，GOSOM 在準確率上 (Accuracy) 平均增加 21.67%、求全率則 (Recall) 增加 28.47%。

關鍵字： Self-Organizing Map、潛在語意分析、文件分群、目標導向

誌謝

本論文能夠順利完成，首先得感謝兩位指導教授，柯皓仁老師與楊維邦老師。在他們的教導下，我得以學會獨立研究的精神與能力；也感謝他們不厭其煩地點出我犯的錯誤，並予以細心指正。

謝謝交大資訊科學資料庫實驗室的學長姊、學弟妹和同學們，因為有你們的集思廣義才讓我的碩士論文得以順利進行。尤其要感謝鄭培成學長、黃夙賢學長、葉鎮源學長、林忠億學長，常撥空與我討論問題，助我解決諸多疑難雜症。

最後，要感謝關心我的家人，一直在背後默默支持我，讓我能全心全意地投入學業與研究中。也謝謝可愛的 eponie，總在我沮喪、情緒低落時，包容我的壞脾氣。謹將此論文，獻給你們。

May, 2004

目錄

圖目錄

表目錄

方程式目錄

第一章 簡介

第一節 文件分群系統

隨著網際網路的普及，越來越多的資訊以數位化的形式呈現在網路上。使用者常會使用搜尋引擎 (例如:Google) 來搜尋想要的資訊，但是往往遇到許多的問題。第一、一般搜尋引擎皆以關鍵字來當成搜尋的條件，但對於使用者而言，大部分搜尋的目的只是存在一個概念，很難透過簡單的關鍵字去涵蓋想要搜尋的目的。第二、越來越多的數位化資訊導致大量的查詢結果。使用者無法從頭至尾瀏覽全部的搜尋結果，導致搜尋出來的結果無法提供有效的幫助。大量的搜尋結果需要透過有效的方法，整理成使用者能夠接受的數量以及格式。許多研究者提出將搜尋的結果分群成不同的概念，讓使用者根據所欲搜尋的目的，選擇想要的概念。對於使用者來講，搜尋結果根據不同的概念來分群，可以讓使用者對於想搜尋的文件有更完整的參考依據，並且透過分群縮小搜尋結果的數量，幫助使用者瞭解哪些搜尋結果才符合使用者的目的。如果再加上回饋式的查詢讓使用者修正查詢的條件，查詢的結果便能夠越來越正確地達到使用者的目的。

文件由文字所組成，所以基本上文件分類的問題就是文字分類的問題。自動決定文件所欲歸類的分類，稱之為文件分群 (Document Clustering)。文件分群應用在訓練資料 (Training Data) 不足或者分類的目的不固定的時候，在分類的時候無法透過訓練資料獲得分類的資訊，同時也要決定所欲歸類的分類。文件分群的應用有許多，例如搜尋引擎，可以透過文件分群系統決定文件所屬的分類；例如報社可以將編輯好的新聞透過文件分群主動分類；電子報提供的新聞自動派送服務，也可透過文件分群系統將使用者有興趣的文章用電子郵件傳

送給使用者；甚至網頁也可透過文件分群建立起階層性質的分類目錄方便使用者根據分類來瀏覽網頁。

最傳統的文件分群方法，是利用一些分群規則，根據文件是否符合規則來決定文件所屬分群。此種分群方法的優點是簡單而且分類有效率，但缺點是需要大量的專家來編輯分群規則，而且對於新進的文件必須時常修改分群規則以保持分群系統的正確性。於是有許多研究，希望能夠自動決定文件的分群，透過自動學習的過程，將文件由關鍵字來表達，以關鍵字跟各種類別的關連程度來決定文件所屬的分群。此時會面對兩個問題。一是有效的文件表示法，二是如何將文件自動分群。

關於有效的文件表示法，一般最常採用的模型 (Model) 就是向量空間模型 (Vector Space Model) 。其主要精神在於，將欲自動分類的文件，以向量來表示，而向量中不同的座標軸，代表不同的字詞，該座標軸的值，即為該字詞於該文件中的權重。關於向量空間模型的研究很多，大略如下：

1.在向量空間模型中，以詞 (Term) 為文件的基本單位，因此，如何適當斷詞切字是相當重要的。最直覺的方法就是採用字典檔 (Dictionary)，將所有的詞都建立在一個檔案中，然後用逐字比對的方式來斷詞。這個方法的準確率相當高，但卻有個缺憾：當出現字典裡沒記載的詞時，斷詞就會斷錯[Lin96]。

2.由於字詞的數目太多，因此必須選取重要且具有代表性的關鍵字來表達文件，以簡化文件自動分群的計算過程。此類問題稱之為維度縮減（Dimension Reduction），運用維度縮減的技術在某些情況下可以用 1/10 的關鍵字來代表文件，而不會降低文件表達的正確性[Sebastiani02]。

3.字詞在一文件中的權重，一般常定義為字詞於該文件中出現次數，稱作 Tf (Term Frequency)；但也有考慮到，同一個字詞若在太多篇文件出現，反而無法區別不同類型文件。因此有把字詞出現的文章篇數 (Document Frequency) 也加以考慮的 TFIDF 方法[Salton88]。

在如何將文件自動分群方面，類神經網路 (Neural Network) 是個被廣為使用的方法。類神經網路為一個大型具有權重的網路，透過學習 (Learning) 的方式來調整權重，然後利用學習過的網路來分群資料。一般的類神經網路，需要額外的訓練資料來進行其學習過程；但採用非監督式學習 (Unsupervised Learning) 的類神經網路便可以不需要訓練資料－它可以直接從欲分群的資料中自我學習，對權重進行適當的調整[Haykin94]。

誠如文章一開始提到的，現今使用者需要一個文件自動分群的系統，來幫助他們快速掌握文件的內容。但在全球資訊網蓬勃發展的今日，資訊爆炸讓使用者對於搜尋文件的要求，從資訊的層次提升到知識的層次。使用者對於大量資訊不再只有瀏覽的目的，而是要從大量的資訊歸納出知識以便後續學習與應用。對於使用者來說，如果能夠根據需要的的主題，透過搜尋系統自動將搜尋的結果分門別類呈現給使用者，將會大大減少使用者人工整理資料的時間。這對於從事系統性的分析寫作：如學術研究、新聞報導、歷史事件分析…等等，有相當大的幫助。舉例來說，若使用者希望蒐集關於「人工智慧」方面的資料，首先，他會用關鍵字「人工智慧」，透過搜尋引擎找回可能相關的文件。接下來，他發現找回來的文件實在太多，要分門別類整理過後，才容易使用。但是，使用者因為應用上的需求，希望能依照「專家系統」、「資訊檢索」這兩個特定方向來分群。除此之外，蒐集來的資料，可能還包含其他不相關的如「基因演算法」、「模糊控制」...等等文件，使用者會希望將其他不相關的文件排除在分群結果之外，或者分到”其他”的分群當中。由以上的應用得知，傳統的文件分群演算法，著重在分群結果，卻忽略使用者有一定的分群目標，往往使得分群結

果不合乎使用者的需求，於是，如何讓使用者設定想要分群的方向，又能夠兼顧文件自動分群的特性，便成為一個重要的研究議題。

然因使用者的搜尋條件不固定，每次不同的搜尋都要準備不同的訓練資料；這對使用者而言，不但麻煩，而且也有實行上的困難。另外，訓練資料還需事先判定答案，在資料量較大時，這亦成為另一難題。所以，在此採用不必額外訓練的文件自動分群方法，較為恰當。

要達到不必額外訓練的文件自動分群，則必須採用非監督式學習的文件分群演算法。Kohonen 在 1984 年提出的 Self-Organzing Map (SOM) [Kohonen84]就是這類的演算法。它被大量運用在語音辨識 (Speech Recognition) 上。後來，Kohonen 在 1996 年進行了 WEBSOM [Honkela96]的研究計畫，將 SOM 運用在新聞群組 (News Group) 文件的文件自動分群上。此後，關於運用 SOM 在文件自動分群的研究，絡繹不絕[Roussinov01] [Guerrero Bote02]。隨著 SOM 的應用日漸廣泛，各種改良的版本也就應運而生。如 DSOM [Su01]， GHSOM [Dittenbach02]，TASOM [Shah-Hosseini03]，ADSOM [Ressom03]…等等，都是改良原本 SOM 演算法。但關於將 SOM 應用在依使用者需求分群這類的研究，並不多見。例如前述，讓使用者設定分群方向，引導文件進行自動分群，就是一個需求很高，具相當實用性的應用。因此，我們希望能藉由探討「如何以特定目標引導 SOM 自動分群」這樣的研究，實際將 SOM 應用到一個目標導向文件分群系統上，以滿足使用者進階的需求。

第二節 研究動機與目的

在 Kangas(1996)、Kohonen(1998)中可發現，SOM 分群的過程中，相似度計算公式是主導因素之一。因此，我們認為若能使用潛在語意分析 (LSA)，導出所有詞之間的相互關係[Deerwester90]，進而在分群過程中計算相似度

時，依其與使用者目標相關程度予以加強或減弱，應有助於「引導 SOM 分群過程，以符合使用者需求」此目的。同樣的，我們亦將此概念應用於提出的群聚標記 (Cluster Labeling) 方法及使用者相關回饋機制 (User Relevance

1. 改良 SOM，使其具備目標導向分群的能力。
2. 提出一個配合使用者設定目標的群聚標記法，以利使用者辨別分群結果。
3. 設計一個適合的使用者相關回饋機制，能有效改善前面兩個步驟的分群結果，使其更符合使用者需求。
4. 實作目標導向文件分群系統 (Goal-Oriented Document Clustering System, GODOC)，其核心為本論文中改良過的 SOM 演算法

本論文基於前述原因及需求，提出一個目標導向之 SOM 模型 (Goal-Oriented SOM)，並以其為核心實作 GODOC。GOSOM 基於 SOM，融合 LSA 加以改良，使 SOM 具有目標導向分群的能力。此外，關於群聚標記，我們提出一權重的多數決 (Weighted Majority Voting) 方法[Mavroudi02]，可以配合使用者設定的目標對群聚進行標記，以便於讓使用者更容易理解各個群聚的內容。此外，我們提出了一個能配合 GOSOM 的使用者相關回饋機制，以期讓文件分群的結果，更符合使用者的要求。

第三節 論文架構

本論文首先在第二章介紹目標導向文件分群的相關研究，包含傳統 SOM 演算法、LSA、使用者相關的文件分群法、以及將 SOM 應用於文件分群的研究。接下來，在第三章詳述我們提出的 GOSOM 主要精神及各模組的功能。接著，介紹以 GOSOM 為核心的 GODOC 系統架構及採用的技術；第四章是以此

系統進行實驗，評估 GOSOM 的準確性及穩定程度，以證明 GOSOM 的可行性；最後，在第五章總結本論文，並探討未來的研究發展方向。

第二章 文件分群相關研究工作

本章說明目標導向文件分群相關的研究工作。關於本論文提出的「目標導向之 SOM」應用於文件分群上，相關研究工作主要分三方面：

- SOM 性質探討：[Kangas96][Kohonen98][Guerrero Bote02][Mavroudi02] [Ressom03][Hagenbuchner03]
- 使用者相關的文件分群系統：[Yu85][Deogun86][Bhuyan91] [Hatano99]
- SOM 應用於文件分群：[Honkela96] [Orwig97] [Kohonen00] [Roussinov01]

圖 1 依年份及技術整理了相關研究的發展，而粗體的部分是本論文主要參考的研究。

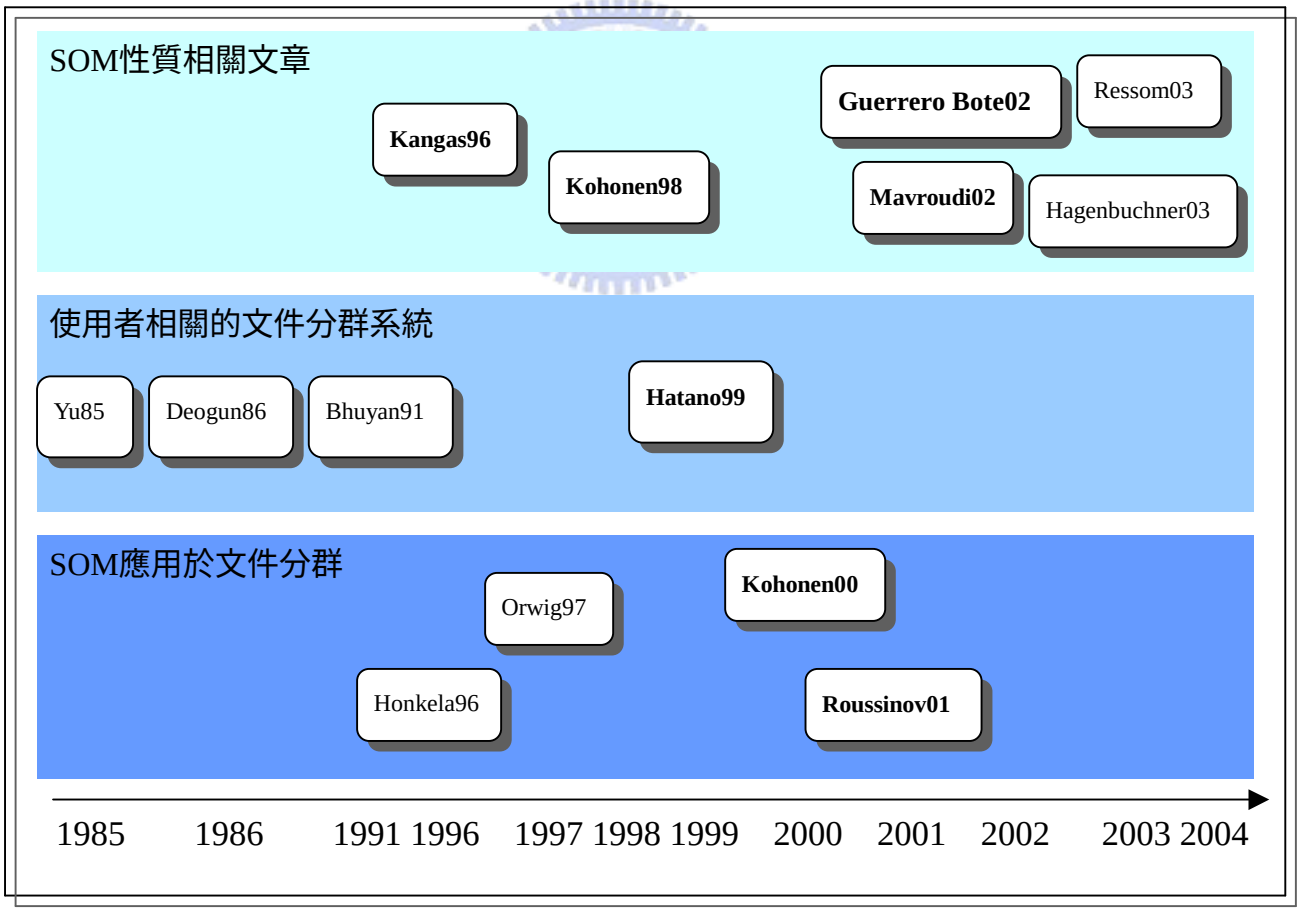

圖 1：相關研究發展

由於我們的「目標導向文件分群系統」核心 — GOSOM，是以 SOM 為基礎改良，所以，首先在第一節介紹傳統 SOM 演算法，藉以瞭解其性質。而「目標導向文件分群系統」是以使用者的興趣為目標，所以在第二節，會介紹與使用者相關的文件分群系統。接下來，因為我們的系統用了 LSA，以連結使用者目標及其他詞間的相關程度，所以在第三節中，我們會簡單介紹 LSA，並以一實例說明。最後，在第四節中，會提到兩個將 SOM 應用於文件分群的系統 — WEBSOM 及 Adaptive Search，以瞭解將 SOM 應用於文件分群系統的各種議題。

第一節　Self-Organizing Map (SOM)

SOM 是種非監督式學習 (Unsupervised Learning) 的類神經網路模型，由 Kohonen 於 1984 提出。與監督式學習 (Supervised Learning) 的類神經網路模型不同的是，SOM 不需額外的訓練資料。SOM 將多維空間的資料對應到二維的平面上，並且在二維的平面上維持多維空間中空間距離的關係。亦即，在多維空間上相近的資料，在二維空間上會被群聚在相近的點上，進而達成分群的目的。圖 2 是其網路拓樸 (Network Topology) 示意圖。SOM 應用在文件分群上主要分成幾個步驟：

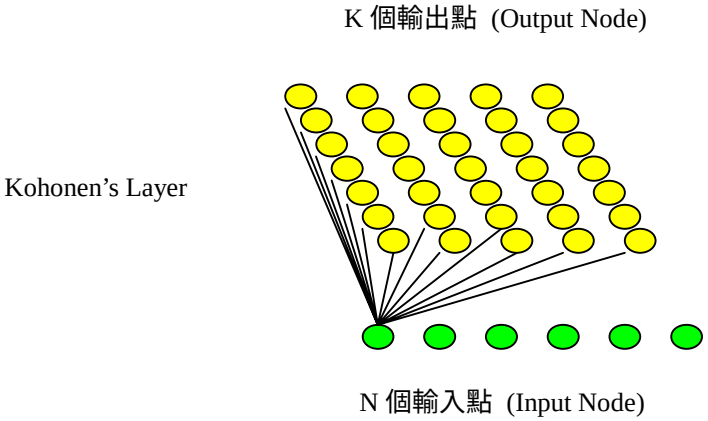

圖 2：SOM 的 Network Topology[Roussinov01]

1. 初始化輸入點 (Input Node)：

套用向量空間模型 (Vector Space Model) 將每筆文件量化，使其成為一個座落在關鍵字空間中的向量。而每個向量，就代表一個輸入點，稱為輸入向量 (Input Vector)。

2. 建立及初始化輸出點 (Output Node)：

通常輸出點是以一個矩形平面排列，所以，需設定矩形長寬以決定輸出點總個數。而每個輸出點都有一個模型向量 (Model Vector)，此向量的維度需與輸入向量的維度相同，此向量可視為輸入向量所在空間中的一點，其起始值可用亂數決定。

3. 進行學習 (Learning) 過程：

以下所描述的，為一次完整循環 (Iteration) 的過程，要反覆進行數次，直到整個 SOM 收斂，或者超過某一循環次數門檻限制才停止。

一個循環要依序用所有輸入向量對所有輸出點模型向量做調整，調整過程如下：

A. 選擇勝利點 (Winner Node)：計算輸入向量 **X** 跟所有輸出點模型向量的距離，模型向量與輸入向量 **X** 最近的輸出點為勝利點。最常被使用的距離計算公式如方程式 1。其中，$\mathbf{m}_i(t)$ 代表在第 t 次調整模型向量時，某一輸出點 i 的模型向量。

$$Similarity\ (\mathbf{X}, \mathbf{m}_i(t)) = \sqrt{\sum_j (Dim_j(\mathbf{X}) - Dim_j(\mathbf{m}_i(t)))^2}$$

方程式 1：以 Euclidean Distance 為相似度定義

B. 調整模型向量：根據 0 方程式 2 來調整模型向量的值：

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{c(\mathbf{X}),i,t} \times (\mathbf{X} - \mathbf{m}_i(t))$$

0 方程式 2：Model Vector Update

0 方程式 2 中，$h_{c(\mathbf{X}),i,t}$ 控制了分群過程中模型向量學習速度的快慢及影響鄰近區域點 (Neighborhood Node) 的能力。$h_{c(\mathbf{X}),i,t}$ 一般的定義如方程式 3 所示，為第 i 個輸出點在第 t 次調整模型向量時，將輸入向量 $\mathbf{X}$ 的勝利點 $c(\mathbf{x})$ 帶入後，所得的函數值。其中 $\mathbf{r}_c$、$\mathbf{r}_i$ 分別為勝利點及第 i 個輸出點在矩形平面的座標向量。

$$h_{c(\mathbf{X}),i,t} = \alpha(t) \exp\left(-\frac{\|\mathbf{r}_c - \mathbf{r}_i\|^2}{2\sigma^2(t)}\right)$$

方程式 3：$h_{c(\mathbf{X}),i,t}$

在方程式 3 中，α(t)介在 0~1 之間，是個單調遞減 (Monotonically Decreasing) 的函數，它影響了學習速度的快慢。σ(t)亦為一單調遞減的函數，它牽涉到勝利點影響鄰近區域點的能力。經此調整後，所有的模型向量，都會有不同程度向輸入向量 X 移動的趨勢。但在不同的應用中，$h_{c(\mathbf{X}),i,t}$ 有較為簡易的定義法，例如方程式 4：

$$h_{c(\mathbf{X}),i,t} = \begin{cases} \alpha(t), \ if \ \|\mathbf{r}_c - \mathbf{r}_i\| < R \\ 0, \ otherwise \end{cases}$$

方程式 4：另一種 $h_{c(\mathbf{X}),i,t}$ 定義法

在方程式 4 裡，只有與距離勝利點距離在範圍 R 內的鄰近區域點，才會被調整值。

4. 指定群聚：

在學習過程完成後，每個輸出點的模型向量，就是分群的依據。方法如下：計算每個輸入向量與所有輸出點模型向量的距離，把該輸入向量指派給距離最近的輸出點。將所有代表文件的輸入向量指派完後，即可得到一個依照文件相似度分群的 2D 地圖。

5. 標記 (Label) 群聚區域：

由於輸出點的模型向量是分群的依據，因此可做為群聚標記時的參考。而 Roussinov (2001)提到一種適合文件分群工作的群聚標記法如下：從每個輸出點的模型向量中，選出一個值最大的座標軸相對應的關鍵字，來代表該點。若相鄰的輸出點擁有一樣的關鍵字，則合併為同一個群聚。

第二節 使用者相關文件分群 (User-Involved Document Clustering)

一般的自動分群過程，其目的是針對某種群聚內部結構的測量 (Internal Measure)，找出最佳化的解，諸如在一個群聚內的相似度測量…等等[Kim00]。其中，使用者的想法並沒有被考慮在內。但從實際應用的角度而言，使用者的想法是很重要的。Yu (1985)提出了以使用者意見為主的分群法 — Adaptive Document Clustering。其主要過程如下[Yu85]：

步驟 1：

1. 若所有文件為 $R_1, R_2, \ldots, R_N$，則將每個文件 R_i 指定到一個在一維實數線 $(-\infty, \infty)$ 上的位置 X_i，$1 \le i \le N$
2. 給定一個查詢 (Query)
   A. 用之前儲存(或預先設定)的群聚，作為回應。

B. 請使用者勾選跟查詢相關 (Relevant) 的文件。

C. 更動文件在實數線上的位置，使得在步驟 B 中使用者勾選的、與此查詢相關的文章，在實數線上的位置互相靠近；不相關的文章中，隨機選出文件，使其遠離其重心。

3. 重複步驟 2 數次。

步驟 2：經過數次查詢後，依據下列方法決定群聚：

在實數線上以遞減方向，確認文件。設 d 為兩個相鄰文件在步驟 1 開始前的平均距離。若經步驟 1 之後，兩相鄰文件的距離小於 $d/3$，則將此兩個文件歸到同一個群聚中。否則，就歸到不同的群聚。

這種依據使用者意見分群的相關研究，被稱為使用者導向分群法 (User-Oriented Clustering) [Raghavan87a][Raghavan87b][Bhuyan89]。此類分群演算法的主要精神，是以使用者所提供的正/負面答案，逐步對分群結果進行調整，形成最後答案[Bhuyan91]。而 Hatano (1999)提出了基於 SOM 的互動式分類 (Interactive Classification)，他提供給使用者五個回饋式運算 (Feedback Operation)，讓使用者與 SOM 達成互動。其五個運算如下：

- 擴大 (Enlarge)：使用者若對某個詞進行「擴大」運算，系統會針對文件的特徵向量 (Feature Vector) 作適當調整，使得歸類為該詞的群聚數盡量增多。
- 刪除 (Delete)：使用者若針對某個詞進行「刪除」運算，則系統會將含該關鍵字的文章，排除在分群過程之外。
- 合併 (Merge)：使用者可針對數個詞進行「合併」運算，指定其成為另一個新詞。系統將把文件的特徵向量中，此數個詞的頻率相加，視為指定的新詞頻率。

- 注意 (Notice)：使用者可對所有文件進行「注意」運算，系統會針對所有的索引詞 (Index Term) 做 Tf-IDf 的計算，取分數高的前 m 個。接下來，將文件的特徵向量，轉換成以此 m 個索引詞表示。
- 分析 (Analysis)：使用者可對所有文件進行「分析」運算，系統會在決定文件的特徵向量時，排除使用者不熟悉的索引詞。

與前述的使用者導向分群法的主要精神相似，此系統也運用回饋式的方法，讓使用者表達意見[Hatano99]。本論文提出的目標導向文件分群概念，修正 Hatano (1999)的方法，讓使用者在分群前，即可用自己的興趣為目標，主動地在分群過程中引導系統，使系統產生符合使用者觀點的分群結果，而不必等到初次分群完成，才使用回饋式的方法，讓使用者逐步修正分群結果。

第三節　潛在語意分析 (Latent Semantic Analysis)

2.2.1 LSA 簡介

潛在語意分析 (Latent Semantic Analysis, LSA) 是種對於字與字之間，基於在文件中被使用的關係，以統計方法來做分析、做為大量文件輔助訓練資料的方法。主要的精神是基於在文件中，字詞的共同出現與否，來作為其字義上相似程度的判斷。它可以適當地表現人類知識的推演建立過程。LSA 有兩大步驟：奇異值分解 (Singular Value Decomposition, SVD) 和維度約化 (Dimension Reduction)。SVD 是一種數學矩陣的分解技術，能將文件所隱含的知識抽象轉換到語意空間中，而維度約化能萃取文件知識在語意空間中的較重要的部分，使 LSA 能更精確地推演出文件所隱含的的知識。若將 LSA 應用到文件索引 (Document Indexing)，則能發現文件間或字詞間深層的相似關係 [Deerwester90]。

2.1.3 LSA 的實例說明 [Landauer98]

以下，舉在 Landauer (1998)中，所提到的一個簡單例子，來說明 LSA 的流程。在這個例子中，文字的內容是給定九篇技術文件的標題，有五篇是人機互動 (Human Computer Interaction, HCI) 的相關文件，分別稱為 c1、c2、c3、c3、c4、c5。其他 4 篇則是關於數學圖形理論 (Mathematical Graph Theory) 的相關文件，分別稱做 m1、m2、m3、m4。圖 3 是這些文件標題的列表。

Example of text data: Titles of Some Technical Memos

c1: *Human* machine *interface* for ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user* perceived *response time* to error measurement

m1: The generation of random, binary, ordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
M4: *Graph minors*: A *survey*

圖 3：範例文件列表

首先，從這些標題中，挑選出現 2 次以上的詞 (圖 3 中斜體部分)，共有 12 個。然後根據這些詞跟文件標題，建構一個詞-文字內容 (Word-by-Context) 的矩陣 X。在 X 中，列的部分，是代表各個不同的詞；行的部分，是代表各篇不同的文件標題；值是該詞在某篇文件標題中出現的次數。

$$X = \begin{array}{c|ccccccccc} & c1 & c2 & c3 & c4 & c5 & m1 & m2 & m3 & m4 \\ human & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ interface & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ computer & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ user & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ system & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ response & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ time & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ EPS & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ survey & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ trees & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ graph & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ minors & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

而矩陣 X 在經過 SVD 分解之後，可分解成 W 、 S 、 P^T 三個矩陣的連乘。

$$W = \begin{bmatrix} 0.22 & -0.11 & 0.29 & -0.41 & -0.11 & -0.34 & 0.52 & -0.06 & -0.41 \\ 0.20 & -0.07 & 0.14 & -0.55 & 0.28 & 0.50 & -0.07 & -0.01 & -0.11 \\ 0.24 & 0.04 & -0.16 & -0.59 & -0.11 & -0.25 & -0.30 & 0.06 & 0.49 \\ 0.40 & 0.06 & -0.34 & 0.10 & 0.33 & 0.38 & 0.00 & 0.00 & 0.01 \\ 0.64 & -0.17 & 0.36 & 0.33 & -0.16 & -0.21 & -0.17 & 0.03 & 0.27 \\ 0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\ 0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\ 0.30 & -0.14 & 0.33 & 0.19 & 0.11 & 0.27 & 0.03 & -0.02 & -0.17 \\ 0.21 & 0.27 & -0.18 & -0.03 & -0.54 & 0.08 & -0.47 & -0.04 & -0.58 \\ 0.01 & 0.49 & 0.23 & 0.03 & 0.59 & -0.39 & -0.29 & 0.25 & -0.23 \\ 0.04 & 0.62 & 0.22 & 0.00 & -0.07 & 0.11 & 0.16 & -0.68 & 0.23 \\ 0.03 & 0.45 & 0.14 & -0.01 & -0.30 & 0.28 & 0.34 & 0.68 & 0.18 \end{bmatrix}$$

$$S = \begin{bmatrix} 3.34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.54 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.35 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.64 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.50 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.31 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.85 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.56 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.36 \end{bmatrix}$$

$$P^T = \begin{bmatrix} 0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.01 & 0.02 & 0.08 \\ -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \\ 0.11 & -0.50 & 0.21 & 0.57 & -0.51 & 0.10 & 0.19 & 0.25 & 0.08 \\ -0.95 & -0.03 & 0.04 & 0.27 & 0.15 & 0.02 & 0.02 & 0.01 & -0.03 \\ 0.05 & -0.21 & 0.38 & -0.21 & 0.33 & 0.39 & 0.35 & 0.15 & -0.60 \\ -0.08 & -0.26 & 0.72 & -0.37 & 0.03 & -0.30 & -0.21 & 0.00 & 0.36 \\ 0.18 & -0.43 & -0.24 & 0.26 & 0.67 & -0.34 & -0.15 & 0.25 & 0.04 \\ -0.01 & 0.05 & 0.01 & -0.02 & -0.06 & 0.45 & -0.76 & 0.45 & -0.07 \\ -0.06 & 0.24 & 0.02 & -0.08 & -0.26 & -0.62 & 0.02 & 0.52 & -0.45 \end{bmatrix}$$

再經過維度約化，取新維度為 2，亦即保留矩陣 S 中最高的兩個數值，其餘的數值均設為 0，相當於只取三個矩陣 W 、 S 、 P 的前 2 列，其餘值設為 0，分別得到 W' 、 S' 、 P'^{T} 。由此，我們可得到一重建矩陣 $X' = W'\ S'\ P'^{T}$ 。

$$X'=\begin{array}{c|ccccccccc} & c1 & c2 & c3 & c4 & c5 & m1 & m2 & m3 & m4 \\ \hline human & 0.16 & 0.40 & 0.38 & 0.47 & 0.18 & -0.05 & -0.12 & -0.16 & -0.09 \\ interface & 0.14 & 0.37 & 0.33 & 0.40 & 0.16 & -0.03 & -0.07 & -0.10 & -0.04 \\ computer & 0.15 & 0.51 & 0.36 & 0.41 & 0.24 & 0.02 & 0.06 & 0.09 & 0.12 \\ user & 0.26 & 0.84 & 0.61 & 0.70 & 0.39 & 0.03 & 0.08 & 0.12 & 0.19 \\ system & 0.45 & 1.23 & 1.05 & 1.27 & 0.56 & -0.07 & -0.15 & -0.21 & -0.05 \\ response & 0.16 & 0.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\ time & 0.16 & 1.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\ EPS & 0.22 & 0.55 & 0.51 & 0.63 & 0.24 & -0.07 & -0.14 & -0.20 & -0.11 \\ survey & 0.10 & 0.53 & 0.23 & 0.21 & 0.27 & 0.14 & 0.31 & 0.44 & 0.42 \\ trees & -0.06 & 0.23 & -0.14 & -0.27 & 0.14 & 0.24 & 0.55 & 0.77 & 0.66 \\ graph & -0.06 & 0.34 & -0.15 & -0.30 & 0.20 & 0.31 & 0.69 & 0.98 & 0.85 \\ minors & -0.04 & 0.25 & -0.10 & -0.21 & 0.15 & 0.22 & 0.50 & 0.71 & 0.62 \end{array}$$

而觀察 X' 與 X 後我們可發現，原本”trees”這個詞彙，在 X 中的 m4 並沒有出現，所以值為 0。但因為 m4 的標題中出現了”graph”、 ”minor”這兩個與”tree”相關的詞，所以在經過重建的矩陣 X' 中，”trees”在 m4 的值被修改成 0.66。另外，”survey”這個詞彙原本在 m4 的值是 1，但在經過重建的矩陣 X' 中，其值被降為 0.42，恰準確地反映了”survey”這個詞對於 m4 這個領域 — ”Mathematical Graph Theory”的文件標題來說，並不重要。由此可知，LSA 的確能萃取出文件知識在語意空間中較重要的部分, 進而估量每個詞彙對不同文字內容的重要性。

第四節　SOM 在文件分群的應用

2.3.1 WEBSOM 系列

赫爾辛基大學 (Helsinki University) 開發了一個文件分群系統 WEBSOM，及其改良版本 WEBSOM2。這個系統主要是用來幫助使用者在極

大量的文件中，快速而有效地找出想要的資料。主要概念是，將文件依據其相似度來分群，以減少使用者搜尋的時間。系統是基於 SOM 演算法來實作的，SOM 中所需的輸入向量則是以文件本身的詞彙統計資料來表示。整個系統主要提供二種功能：

- 內容強調的搜尋 (Content Addressable Search)：

  使用者用系統提供的介面，來描述他們想要文件的概念。系統會將此概念模擬成一短文，然後將此短文位於 SOM 結果地圖 (Result Map) 的位置標出。而此位置周圍的文件就是使用者可能要找的文件了。

- 關鍵字搜尋 (Keyword Search)：

  在系統將所有文件做完分群、產生地圖之後，會以適當方法選出代表地圖上各個群聚的關鍵字。若使用者特別對某關鍵字文件有興趣，則可選擇關鍵字搜尋。系統會將符合查詢關鍵字的區域標示出來，讓使用者可快速瀏覽這些文件。

這個系統遇到的主要問題是，如何在有限資源及時間內，將極大量的文件依相似度分群。因此，主要探討的問題就是如何加快 SOM 的速度。在文件的前處理 (Preprocessing) 上，Kaski (1997)提出了用隨機對映 (Random Mapping) 的方法來解決關鍵字過多的問題。在 SOM 演算法上，Kohonen (2000)提出了下面幾點改進：

- 在代表文件的輸入向量中，會有許多維度的值為 0。這可以用來加速文件相似度的計算，以應用於 SOM 中所有用到計算文件相似度處。
- 建構一個規模 (Size) 較小的 SOM 來進行計算，再把結果當成之後完整、大規模的 SOM 的起始值依據，以增進大規模 SOM 收斂的速度。

- 在 SOM 分群的過程進行中，可採用許多逼近法來加速找尋勝利點，又保持相當的準確性。如何有效的平行化處理 (Parallelized) 也被提出[Kohonen00]。

2.3.2 適應式搜尋 (Adaptive Search)

Roussinov (2001)提到：傳統的關鍵字搜尋並不是一個好方法，因為使用者可能對他所要搜尋的主題並不完全瞭解，以致於無法給定精確的搜尋關鍵字。而 Cutting (1992)提出一個新方法：先把文件分群，再從各個分群文件挑出關鍵字，描述該文件群，讓使用者方便點選某群文件，再繼續進行分群、讓使用者點選的工作，直到使用者滿意為止。但是，這種結合分群的瀏覽式搜尋，花的時間比傳統關鍵字搜尋還多，而且找到的文章，相關性反而比較低。因此，Roussinov (2001)提出一種新方法，名為 "Adaptive Search"，主要概念是：在搜尋引擎跟使用者之間，多加一個中間層 (Layer)。其流程如圖 4，主要步驟為：

1. 使用者先透過中間層對傳統關鍵字搜尋引擎下關鍵字搜尋。
2. 中間層把搜尋引擎傳回來的文件用 SOM 分成幾個較小的群聚，且將每個群聚做適當的標記，呈現給使用者。

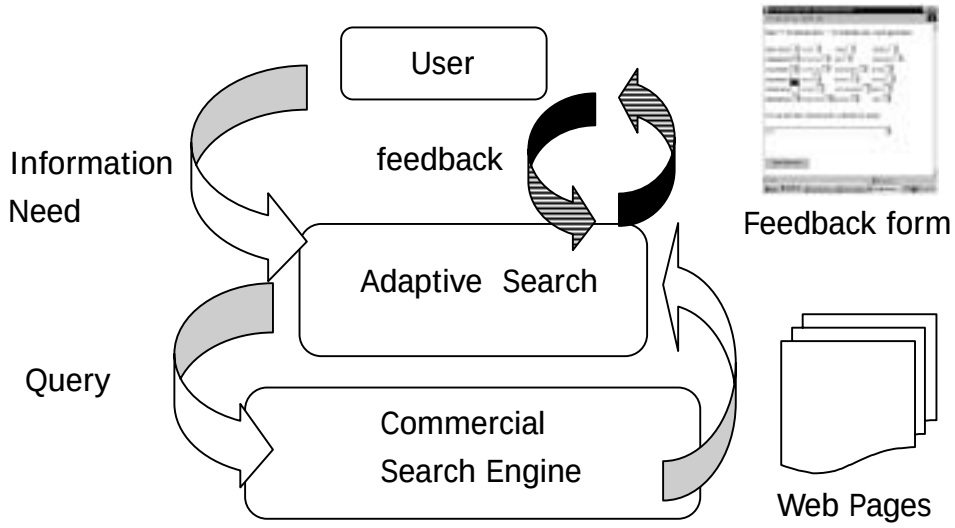

圖 4：Adaptive Search 流程示意[Roussinov01]

3. 使用者透過中間層的”feedback form”做使用者相關回饋，指出哪幾群是符合使用者興趣的。另外，使用者也可以再補上有興趣的關鍵字。
4. 中間層根據使用者的 feedback，再形成更精準的查詢字串，向搜尋引擎發出查詢。之後就重複(2)~(4)的步驟，直到找出使用者滿意的文件為止[Roussinov01]。

第三章 目標導向文件分群模型及系統實作

本論文提出一個目標導向之 SOM (Goal-Oriented Self-Organizing Map, GOSOM) 模型。GOSOM 基於傳統 SOM，並結合了潛在語意分析 (Latent Semantic Analysis, LSA)，達成以使用者興趣，引導文件分群過程的功能，使之更符合使用者的需求。在群聚標記方面，為了讓使用者更易瞭解分群結果，我們提出一權重的多數決 (Weighted Majority Voting) 方法。此外，我們針對此模型特性, 提出了使用者相關回饋 (User Relevance Feedback) 方法, 並將 GOSOM 實作在目標導向文件分群系統 (Goal-Oriented Document Clustering System, GODOC)。

首先，在本章的第一節，先描述 GOSOM 的架構，再逐一介紹模型中各元件的細節。在談到 GOSOM 時，共有三個重點。一、介紹 SOM 在文件分群上的傳統作法及 GOSOM 如何改良；二、解釋權重的多數決群聚標記法；三、說明使用者相關回饋的方法。在本章的第二節，會介紹 GODOC 的流程，並分別介紹各元件。

第一節 目標導向之 SOM (Goal-Oriented SOM, GOSOM)

由第二章的介紹中可知，傳統 SOM 主要是藉由輸入點一次次對輸出點的模型向量做調整，使得輸出點逐漸形成相似的群聚，最後才將每個輸入點，依其跟所有輸出點模型向量的相似度大小，來指定其分群結果，因此，輸出點的模型向量，具有代表分配到該輸出點內所有輸入點特徵的功能。由此可知：

- 若要達成「依使用者興趣為目標，引導分群過程」的目的，可從「依使用者興趣來定義相似度」著手。因為在 SOM 中，輸出點群聚形成的過程，正是依據相似度大小來決定的。相似度越大，越容易形成群

聚。所以，若採用具某特性的相似度函式，則分群過程就會被此特性影響。

- 分群的解釋方法，必須要跟模型向量的內容有相當程度關係，因為這代表了 SOM 的分群結果。

GOSOM 就是基於這兩個基本精神，融入目標導向分群的概念，所提出的模型。圖 5 是 GOSOM 模型示意圖，其流程如下：

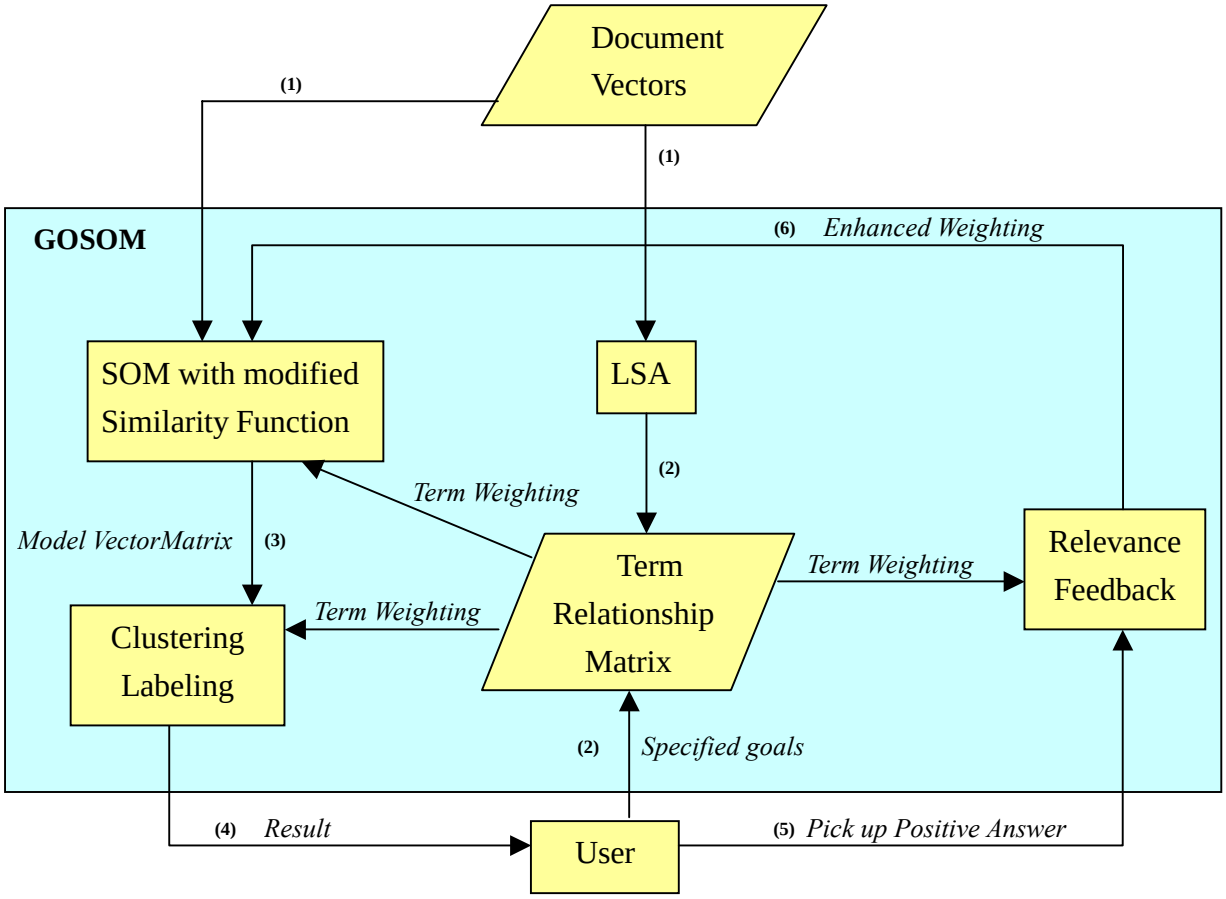

圖 5：GOSOM 模型示意圖

1. 將文件經處理過所形成的文件向量，輸入到 LSA 及搭配改良相似度定義的 SOM 中。
2. 運用 LSA，可產生詞-詞的關係矩陣，再以使用者輸入的分群目標配合此矩陣，可將使用者分群目標代表的概念擴展到其他詞，可對所有詞產生適當的權重。

3. 將傳統 SOM 配合改良過的相似度定義，及步驟 2 所得的詞權重，可產生代表初步分群結果的模型向量矩陣。
4. 以步驟 2 所得的詞權重，配合本論文提出的群聚標記法，可為步驟 3 的模型向量矩陣作適當標記，產生分群結果，呈現給使用者。
5. 使用者可針對分群結果，勾選分群正確的文件，做使用者相關回饋。
6. 系統根據使用者的意見，對步驟 2 所得的詞權重作調整，再次進行分群。

在接下來的小節內，會分別介紹 GOSOM 的四個元件：LSA、改良的相似度定義、群聚標記、以及使用者相關回饋。

3.1.1 LSA

我們實作了一個能執行 LSA 方法的元件。在 LSA 中最主要的矩陣運算技術，就是 SVD 分解。我們採用 http://math.nist.gov/javanumerics/jama/[NIST]上的 Jama-1.0.1 套件來實作，而其他的矩陣運算應用，也是基於此套件延伸發展而來。

3.1.2 改良的相似度定義

在傳統應用 SOM 於文件分群的系統裡，大部分是採用向量模型空間，所有文件都以詞為特徵，將文件化為實數的特徵向量來描述。而本研究欲達成之目的，是要依「使用者目標」分群；也就是說，分群過程中，相似與否要從「使用者目標」的觀點來決定。在此，「使用者目標」不是一個詞，而是一個概念 (比如說：「使用者目標」可能為「捷運偷拍」這種模糊的概念)。因此，為了將「概念」這種特色引入，我們使用 LSA 方法，深層分析文件中詞與詞的關係，可得一詞 – 詞關係矩陣，藉此找出與「使用者目標」較相關的詞。如此，我們便可用「依與使用者目標相關的詞分群」這種方法，在向量空間模型中達成「依使

用者目標分群」的目的。參考之前的文獻，SOM 最常將相似度定義成 Euclidean Distance (方程式 5)[Kohonen98]。

$$Similarity\ (\mathbf{X}, \mathbf{m}(t)) = \sqrt{\sum_i (Dim_i(\mathbf{X}) - Dim_i(\mathbf{m}(t)))^2}$$

方程式 5：以 Euclidean Distance 作為相似度定義

而這樣的相似度定義，無法強調特定目標，如：使用者的興趣或喜好。我們認為，與使用者目標較相關的詞，在相似度計算中，應為其主要因素。因此，我們提出了一個改良後的相似度定義如方程式 6：

$$Similarity\ '(\mathbf{X}, \mathbf{m}(t)) = \sqrt{\sum_i W_i (Dim_i(\mathbf{X}) - Dim_i(\mathbf{m}(t)))^2}$$

方程式 6：改良後的相似度定義

在輸入向量 **X** 與 $\mathbf{m}(t)$ 兩個向量相減時，於每個不同字詞對應的座標軸 i，乘上反映其重要程度的權重 (Weight) W_i，將較重要的詞，乘以較大的權重；較不重要的詞，乘以較小的權重。如此在計算相似度時，會強調在較重要詞上的差異，而相對忽略其他不重要的詞。接著，我們將 W_i 定義如方程式 7：

$$W_i = F(\max_j \mathbf{R}(i, usergoal_j))$$

方程式 7：改良後的相似度公式中，詞的權重

在此，$usergoal_j$ 代表第使用者 j 個使用者目標。**R** 是透過 LSA 方法得到的詞－詞關係矩陣，$\mathbf{R}(i, j)$ 表示第 i 個詞與跟第 j 個詞的相關程度 － 其值介在-1~1 之間[Deerwester90]。$F(\mathbf{R}(i, j))$ 是一個映射方程式，將 $\mathbf{R}(i, j)$ 的值域對映到 0~1，定義為 $\frac{\mathbf{R}(i,j)-1}{\mathbf{R}(i,j)-(-1)} = \frac{F(\mathbf{R}(i,j))-1}{F(\mathbf{R}(i,j))-0}$。經整理，得 $F(\mathbf{R}(i,j)) = \frac{\mathbf{R}(i,j)+1}{2}$。舉例來說，若透過 LSA 得到的矩陣如：

$$\mathbf{R} = \begin{bmatrix} & Term1 & Term2 & Term3 & Term4 & Term5 & Term6 & Term7 \\ Term1 & 1 & 0.6 & -0.1 & 0.3 & 0.1 & 0.4 & 0.2 \\ Term2 & 0.6 & 1 & -0.3 & -0.1 & -0.75 & -0.9 & -0.5 \\ Term3 & -0.1 & -0.3 & 1 & -0.4 & -0.4 & -0.7 & -0.2 \\ Term4 & 0.3 & -0.1 & -0.4 & 1 & -0.8 & -0.8 & 0.9 \\ Term5 & 0.1 & -0.75 & -0.4 & -0.8 & 1 & 0.7 & -0.9 \\ Term6 & 0.4 & -0.9 & -0.7 & -0.8 & 0.7 & 1 & -0.85 \\ Term7 & 0.2 & -0.5 & -0.2 & 0.9 & -0.9 & -0.85 & 1 \end{bmatrix}$$

圖 6：經 LSA 方法得到的詞－詞關係矩陣範例

而使用者的分群目標為 Term 2、4、7 三類(亦即 $usergoal_1 = Term\,2$、$usergoal_2 = Term\,4$、$usergoal_3 = Term\,7$)，我們若欲決定 W_1，則套用方程式 7 後，計算如下：由 **R** 中可得，Term 1 這個詞與三者關係程度為[0.6,0.3,0.2]。因此，$\max \mathbf{R}(i, usergoal_j) = 0.6$，$W_1 = F(0.6) = 0.8$。

3.1.3 群聚標記 (Cluster Labeling)

在 SOM 分群過程結束後，每個輸出點會有一個模型向量，其維度與輸入點同。而輸出點的模型向量，代表分配到該輸出點內，所有輸入點的特徵 (Feature)。所以，若要對一個輸出點做標記，以代表分配到此輸出點的輸入點(也就是文件)，一定要參考模型向量。而將這些輸出點分群結果作適當標記 (Labeling)，有助於讓使用者瞭解該群聚文件的特性。

3.1.3.1 權重的多數決 (Weightd Majority Voting)

根據 Roussinov (2001) 所提出的適應式搜尋 (Adaptive Search) 系統中，是根據每個輸出點的模型向量，選出其中值最高的座標軸，將其對應的字詞，當作該輸出點代表的群聚之標記。本論文提出一融合多數決 (Majority Voting) 精神[Mavroudi02]與權重概念的群聚標記方法，稱「權重的多數決」。方法敘述如下：

1. 首先，定義與第 i 個詞的最相關的使用者目標為：

$$TermGoal(i) = \arg \max_{\text{usergoal}_j} \mathbf{R}(i, usergoal_j)$$

R 是透過 LSA 方法得到的詞－詞關係矩陣。

2. 延續 3.1.2.中改良相似度定義時的計算公式，設第 i 個詞的權重為：

$$W_i = F(\max_j \mathbf{R}(i, usergoal_j))$$

3. 接著，定義第 j 個使用者目標，在輸出點 (p,q) 上，對於第 i 個詞的分數：

$$Score(i, usergoal_j) = \begin{cases} 0, & if \ \ TermGoal_i \neq usergoal_j \\ \mathbf{m}_{p,q}(i), & else \end{cases}$$

$\mathbf{m}_{p,q}(i)$ 表示在輸出點 (p,q) 的模型向量 **m** 中，第 i 個詞對應的座標軸值。

4. 有了 $Score(i, usergoal_j)$ 後，我們便可定義第 j 個使用者目標在輸出點 (p,q) 中的總分為：

$$SumofGoal(usergoal_j) = \sum_i Score(i, usergoal_j) \times W_i$$

5. 最後，依據各個使用者的目標總分來決定輸出點 (p,q) 的標記：

$$Label_{p,q} = \arg \max_{usergoal_j} SumofGoal(usergoal_j)$$

舉例來說，套用「權重的多數決」決定輸出點(1,1)的步驟如下：

- 若輸出點的規模為 3X3，輸入向量維度及輸出點的模型向量維度皆為 7。使用者的目標有三個，分別為 Term 2、4、7。**R** 是透過 LSA 方法

得到的詞－詞關係矩陣如圖 6、$F(\mathbf{R}(i,j)) = \frac{\mathbf{R}(i,j)+1}{2}$、**M** 為 SOM 分群後所得的模型向量矩陣，內容如圖 7：

$$\mathbf{M} = \begin{bmatrix} \begin{bmatrix} 1.3 \\ 5.1 \\ 6.9 \\ 7.1 \\ 4.5 \\ 5.5 \\ 9.1 \end{bmatrix} & \begin{bmatrix} 1.1 \\ 4.3 \\ 5.6 \\ 5.5 \\ 3.2 \\ 4.5 \\ 8.7 \end{bmatrix} & \begin{bmatrix} 1.1 \\ 2.1 \\ 2.2 \\ 3.5 \\ 7.1 \\ 6.2 \\ 5.0 \end{bmatrix} \\ \begin{bmatrix} 2.5 \\ 4.3 \\ 3.6 \\ 3.8 \\ 6.7 \\ 8.1 \\ 4.1 \end{bmatrix} & \begin{bmatrix} 3.8 \\ 3.5 \\ 4.2 \\ 4.9 \\ 8.7 \\ 9.1 \\ 5.1 \end{bmatrix} & \begin{bmatrix} 4.5 \\ 4.4 \\ 5.1 \\ 6.1 \\ 7.1 \\ 9.8 \\ 6.5 \end{bmatrix} \\ \begin{bmatrix} 5.8 \\ 4.7 \\ 2.1 \\ 5.6 \\ 7.3 \\ 2.1 \\ 5.5 \end{bmatrix} & \begin{bmatrix} 2.5 \\ 2.2 \\ 3.9 \\ 3.7 \\ 6.7 \\ 5.5 \\ 9.6 \end{bmatrix} & \begin{bmatrix} 7.6 \\ 1.3 \\ 5.9 \\ 3.5 \\ 4.3 \\ 1.1 \\ 8.2 \end{bmatrix} \end{bmatrix}$$

圖 7：經 SOM 分群後的模型向量矩陣範例

- 決定與第 i 個詞的最相關的使用者目標：

$$TermGoal(1) = \arg\max_{\text{usergoal}_j} \mathbf{R}(i, usergoal_j) = \arg\max_{usergoal_j} [0.6 \quad 0.3 \quad 0.2]$$
$$= \text{usergoal}_1$$
$$TermGoal(2) = \text{usergoal}_1 \quad TermGoal(3) = \text{usergoal}_3$$
$$TermGoal(4) = \text{usergoal}_2 \quad TermGoal(5) = \text{usergoal}_1$$
$$TermGoal(6) = \text{usergoal}_2 \quad TermGoal(7) = \text{usergoal}_3$$

- 計算第 i 個詞的權重：

$$W_1 = F(\max_j \mathbf{R}(i, usergoal_j)) = F(\max_j [0.6 \quad 0.3 \quad 0.2]) = F(0.6) = 0.8$$

$$W_2 = 1 \qquad W_3 = 0.4$$

$$W_4 = 1 \qquad W_5 = 0.125$$

$$W_6 = 0.1 \qquad W_7 = 1$$

- 計算三個使用者分群目標，在輸出點(1,1)上，對於第 i 個詞的分數：

$$Score(1, usergoal_1) = 1.3 \quad Score(1, usergoal_2) = 0 \quad Score(1, usergoal_3) = 0$$
$$Score(2, usergoal_1) = 5.1 \quad Score(2, usergoal_2) = 0 \quad Score(2, usergoal_3) = 0$$
$$Score(3, usergoal_1) = 0 \quad Score(3, usergoal_2) = 0 \quad Score(3, usergoal_3) = 6.9$$
$$Score(4, usergoal_1) = 0 \quad Score(4, usergoal_2) = 7.1 \quad Score(4, usergoal_3) = 0$$
$$Score(5, usergoal_1) = 4.5 \quad Score(5, usergoal_2) = 0 \quad Score(5, usergoal_3) = 0$$
$$Score(6, usergoal_1) = 0 \quad Score(6, usergoal_2) = 5.5 \quad Score(6, usergoal_3) = 0$$
$$Score(7, usergoal_1) = 0 \quad Score(7, usergoal_2) = 0 \quad Score(7, usergoal_3) = 9.1$$

- 計算三個使用者分群目標的總分如下：

$$SumofGoal(usergoal_1) = 1.3 \times 0.8 + 5.1 \times 1 + 4.5 \times 0.125 = 6.7025$$

$$SumofGoal(usergoal_2) = 7.1 \times 1 + 5.5 \times 0.1 = 7.65$$

$$SumofGoal(usergoal_3) = 6.9 \times 0.4 + 9.1. \times 1 = 11.86$$

- 最後，依據各個使用者的目標總分決定標記：

$$Label_{1,1} = \arg \max_{usergoal_j} [6.7025, 7.65, 11.86] = usergoal_3$$

結果，輸出點(1,1)的標記為第三個使用者分群目標。

3.1.3.2 其他類 (Unrelated)

在前面我們提到用「權重的多數決」方式來決定一個群聚的標記。此方法是考慮到「相對多數」的概念。但在真實應用中，還要考慮到「絕對多數」的問題。例如：欲分群的文件，是從搜尋引擎中得到，其中可能有些不屬於使用者指定的任一目標。若使用前述的「權重的多數決」，將會錯將這些文件，歸到其中某目標裡。為了解決這個問題，我們採用一個「總分門檻」方法，敘述如下：

1. 從 LSA 方法獲得的詞－詞關係矩陣為 **R**，$\mathbf{R}(i,j)$ 表示第 i 個詞與跟第 j 個詞的相關程度。$F(\varepsilon)$ 是一個映射方程式，定義為：$F(\varepsilon)=\frac{\varepsilon+1}{2}$。
2. 將所有分群目標與所有詞的關係程度取平均值 $\varepsilon=\frac{1}{\|j\|}\sum_{j}\frac{1}{\|i\|}\sum_{i}\mathbf{R}(i,j)$，$\|i\|$、$\|j\|$ 分別為所有詞的個數，及使用者分群目標個數。
3. 設一總分門檻 α （$\alpha=\frac{1}{2}$，$\frac{1}{3}$，..等等）。在「權重的多數決」的步驟 3 中，若沒有任一分群目標總分 ≥ $\alpha\times F(\varepsilon)\times\sum_{i}\mathbf{m}_{p,q}(i)$，則將此輸出點標記為「其他」類，而非統計總分最高的分群目標。$\mathbf{m}_{p,q}(i)$ 表示在輸出點 (p,q) 的模型向量 **m** 中，第 i 個詞對應的座標軸值。

舉例說明：

- 延續前一個例子，**R** 如圖 6、**M** 如圖 7。決定輸出點(1,3)的步驟如下：
- 三個使用者目標的累計總分 $SumofGoal(usergoal_j)$ 依序分別為：3.8675、4.12、5.88。
- 設總分門檻值為 α=0.5。經由上述「總分門檻」方法計算得 ε=-0.057，檢查發現：

  $3.8675 < 4.12 < 5.88 < 0.5\times F(-0.057)\times(\ 1.1 + 2.1 + 2.2 + 3.5 + 7.1 + 6.2 + 5.0\ ) = 0.5\times 0.4715\times 27.2 = 6.4124$

則表示此輸出點與三個分群目標都不太相關，應標記成「其他」類比較適合。

3.1.4 使用者相關回饋 (User Relevance Feedback)

本論文在 3.1.2 提出的改良相似度定義，是以詞權重為基礎。因此，若能正確地找出接近使用者觀點的詞權重值，有助於改善分群結果。在一般檢索系統 (Retrieval System) 的使用者相關回饋策略裡，認為使用者勾選的正面答案中，其向量的主要特徵較為重要，因此，會在計算時將這些特徵加強。所以，基於此一精神，我們提出了依照使用者勾選的正面答案調整詞權重的使用者相關回饋法。其演算法敘述如下：

定義：n 為使用者勾選系統分群正確文件的篇數、f 為使用者相關回饋的次數、t 為輸入向量的維度、$\mathbf{d}_i$ 為代表使用者勾選系統分群正確的第 i 篇文件的文件向量、$Label(\mathbf{d}_i)$ 為使用者勾選第 i 篇文件的系統標記、$\mathbf{R}(term_j, term_k)$ 為 $term_j$ 跟 $term_k$ 在詞－詞關係矩陣中的值、β 為使用者相關回饋的衰退率、γ 為詞調整的擴充程度，$0 \le \beta, \gamma \le 1$。

步驟：

1. 計算每個詞在 $\mathbf{d}_i$ 中的重要性如方程式 8：

$$importance(\mathbf{d}_i, term_a) = \mathbf{d}_i(term_a) \times W_a$$

方程式 8：每個詞在文件向量 $\mathbf{d}_i$ 中的重要性

$\mathbf{d}_i(term_a)$ 表 $term_a$ 在 $\mathbf{d}_i$ 中的值、W_a 定義如方程式 7。

2. 將所有詞根據方程式 8 算出重要性，依序由高到低排列， 取 $\gamma \times t$ 個詞。

3. 對每個在步驟 2 中被挑中的$term_j$，計算出此詞與所有詞平均關係值如方程式 9。然後對所有的詞進行測試調整：若$\mathbf{R}(term_k, term_j) \geq Avg(term_j)$，則將$term_k$與$Label(\mathbf{d}_i)$所代表使用者目標$usergoal_l$的關係值如方程式 10 提高。

$$Avg(term_j) = \sum_{k=1}^{t} \frac{\mathbf{R}(term_j, term_k)}{t}$$

方程式 9：某詞與所有詞平均關係值

$$\mathbf{R}(term_k, Label(\mathbf{d}_i))_{new} = \mathbf{R}_{old} + [\mathbf{R}_{old} - (-1)] \times \beta \times (1-\beta)^f$$

方程式 10：將詞與使用者目標$usergoal_l$關係值提高

4. 對其他被勾選的文件重複進行步驟 1、2、3。

舉某一篇使用者勾選分群正確的文件$\mathbf{d}_i$做說明：

- $\gamma \times t$=3、β=0.1、f=1、$Label(\mathbf{d}_i)$=Term2、$\mathbf{R}$如圖 6。
- 設依步驟 1、2 計算重要性後，依高低順序排列為：Term3、Term5、Term6、Term1、Term4、Term7、Term2。則應取前 3 個詞：Term3、Term5、Term6 進行測試調整。
- 以 Term3 為例：先計算門檻$Avg(term_3) = \sum_{k=1}^{7} \frac{\mathbf{R}(term_3, term_k)}{7} = -0.157$。再對所有詞進行測試，只有 Term1、Term3 通過，因此對二者與使用者目標 Term2 的關係值進行如下調整：

$$\mathbf{R}(term_1, term_2)_{new} = 0.6 + 1.6 \times 0.1 \times (1-0.1)^1 = 0.744$$
$$\mathbf{R}(term_3, term_2)_{new} = -0.3 + 0.7 \times 0.1 \times (1-0.1)^1 = -0.237$$

第二節　目標導向文件分群系統 (Goal-Oriented Document Clustering System, GODOC)

在參考了 Kohonen (1998) WEBSOM 及 Hotano (1999)後，本論文提出一套 GODOC，其核心為 3.1 所提出的 GOSOM，希望將傳統應用 SOM 於文件分群的方式，加以改進，融入目標導向以及使用者相關回饋的概念，以期文件分群準確度更加提升，更符合使用者需求。圖 8 是 GODOC 的流程圖。處理的步驟如下：

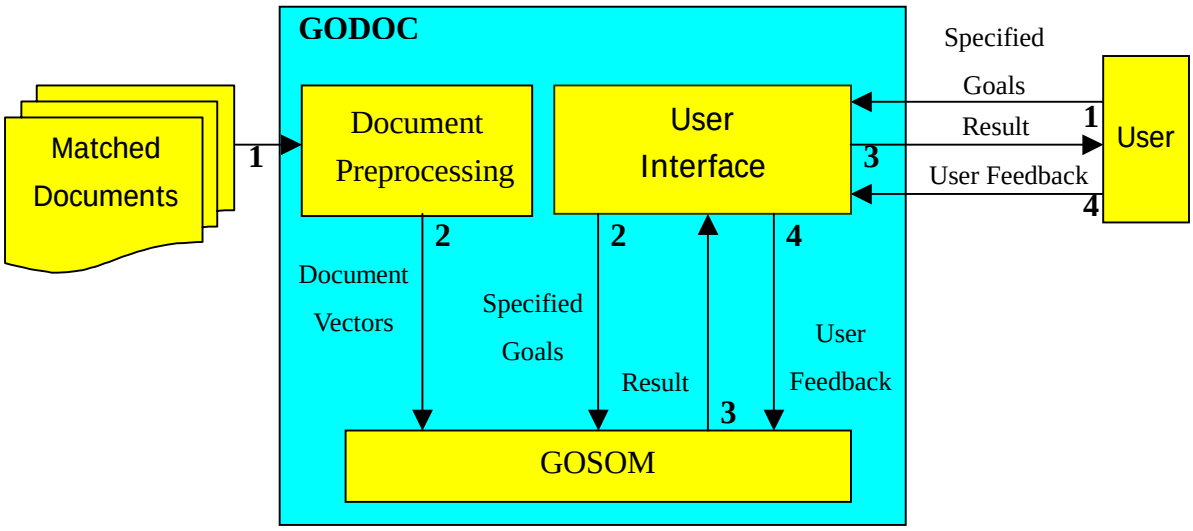

圖 8：GODOC 流程圖

1. 將欲分群的文件及分群目標，輸入到 GODOC 中。
2. 文件經過處理後，轉化成文件向量，跟使用者的分群目標一起輸入 GOSOM。
3. 系統將 GOSOM 的分群結果透過使用者介面，呈現給使用者
4. 使用者透過使用者介面，勾選分群正確的文件做使用者回饋，讓系統再次進行分群。

接下來會介紹各元件：包括了文件前處理，及使用者介面。GOSOM 則在本章第一節已介紹過。

3.2.1 文件前處理 (Document Preprocessing)

GODOC 的文件事先處理含兩部分：斷詞切字與去除停用字 (Stop Word)。在文件分群的研究領域裡，斷詞切字本身就是一個重要的研究項目。本論文在斷詞切字這方面係採用「字典檔比對法」，以 http://www.mandarintools.com 上的中文斷詞切字工具為基礎，加以改寫。而去除停用字部分，一樣係採用「字典檔比對法」處理。

3.2.2 使用者介面 (User Interface)

使用者介面的部分，如圖 9 所示，是採用 html 的介面，網頁伺服器採用 Caucho 公司所出版的 Resin。GOSOM 係利用 JAVA 程式語言來發展。一個格子就是 SOM 的一個輸出點，也就是一個群聚。每個群聚會有一個標記，代表這個群聚內的文件，都被標記成該類文件，不同的顏色代表不同的標記。

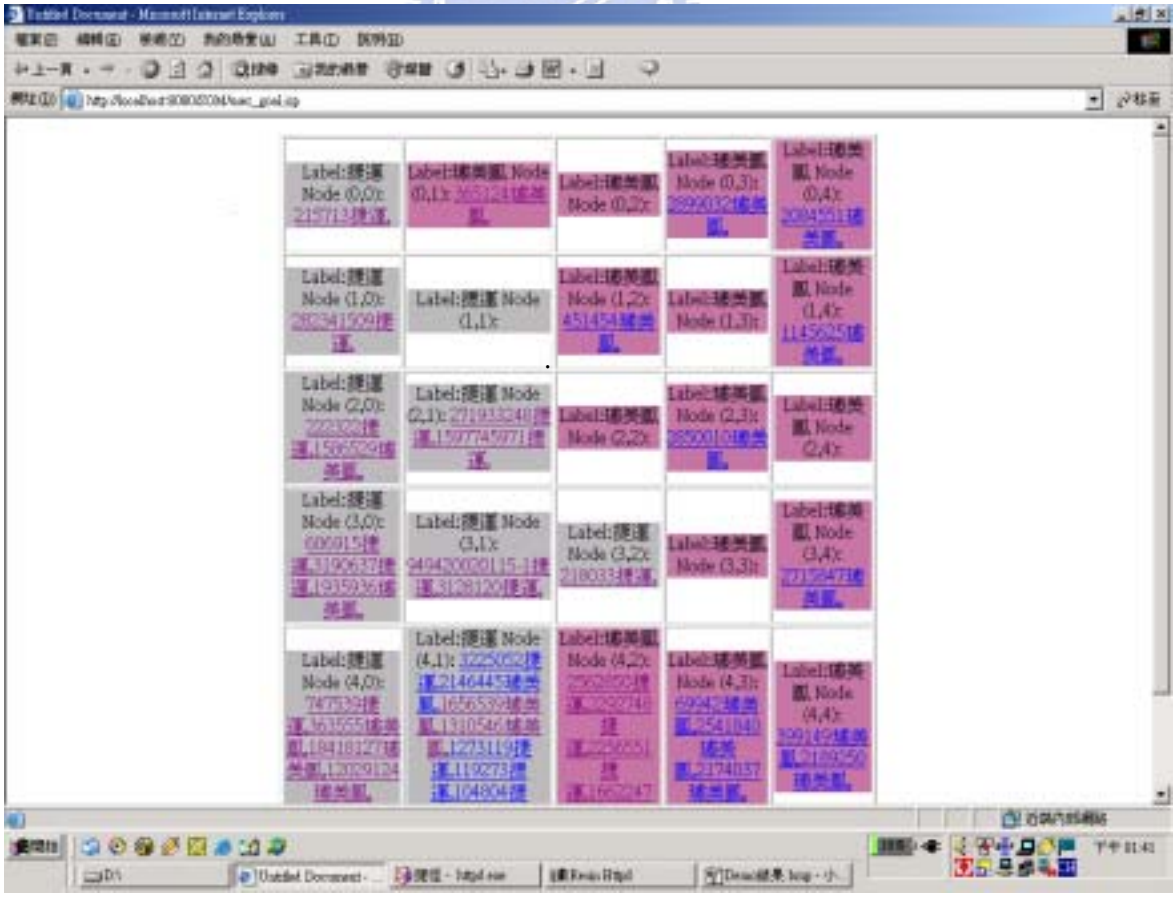

圖 9：使用者介面

第四章 實驗結果分析與評估

第一節 實驗資料及實驗設計

由於系統是為了「以使用者興趣為目標，進行目標導向的分群」這樣的目而設計，因此我們希望能測試出在「目標導向」這種概念下，系統的效能表現。所以我們便以文件內容中的文字主要特性，假設為分群目標，以此進行測試。

實驗資料的來源為網路新聞文件，共分兩類。第一類實驗資料共有兩組，大部分是聯合知識庫 (http://udndata.com/library) 中五大報的新聞；另有少部分透過入口網站 PChome (http://www.pchome.com.tw) 及搜尋引擎 Google (http://www.google.com)找到的新聞文件。第二類全部都是和 Yahoo 奇摩合作的中央社 (http://tw.news.yahoo.com/polity/cna/) 提供的網路新聞。

在第一類的文件中，文件都有兩個主題，一個是同組內所有文件共有的「中心主題」，另一個是跟「中心主題」相關的「特性主題」。舉例來說：有兩篇文件都跟「洩密」有關係，其中一篇是關於「花蓮選舉洩密案」，另一篇是跟「陳水扁飛彈洩密案」有關，則因此對這兩篇文件可設定共通的中心主題是「洩密」，而二者分別的「特性主題」則為「選舉」、「飛彈」。在第一類實驗中，我們把「特性主題」當成使用者感興趣的主旨，設定為分群目標，以此進行分群實驗。本類實驗文件中，「特性主題」較為明確，是某事件的專有名詞，文件數量較少。兩組文件的內容特性如下表：

	第一組資料集	第二組資料集
時間	2002/01/01~2004/02/13	2001/01/01~2004/02/13
中心主題	洩密	偷拍
總篇數	100	100

特性主題(分群目標)數	3	2
特性主題(分群目標)篇數	「選舉洩密」:26 「國安密帳洩密」:27 「飛彈洩密」:27 「其他」:20	「捷運偷拍」:40 「璩美鳳偷拍」:40 「其他」:20
文件長度	大部分約 1000 字內	大部分約 1000 字內

表格 1：第一類實驗資料說明

第二類的實驗文件特性，與前一類不同，只設定有一個較為模糊的「特性主題」。在前述中提到，本類的實驗文件皆為中央社的網路新聞。我們就以原網站上，將新聞略分成十大類的類別為主題，當成「分群目標」。本類的文件較第一類文件不同處為：文件長度較長、其分群目標較為模糊無特定之專有名詞。其內容特性如下表：

	第一組資料集
時間	2003/11/11~2003/11/30
中心主題	無
總篇數	200
特性主題(分群目標)數	3
特性主題(分群目標)篇數	「健康類新聞」:55 「財經類新聞」:55 「政治類新聞」:55 「其他類」:35
文件長度	大部分約 1000 字~1500 字

表格 2：第二類實驗資料說明

在實驗設計方面，總共有三個實驗。第一個實驗是使用第一類實驗文件，目的在證明本系統的效能改善。第二個實驗則使用第二類實驗文件，目的在證明，雖然文件數變多，分類目標較為模糊，但系統的效能仍能穩定地維持一定水準。第三個實驗則混合使用第一、二類實驗文章，希望測出我們的使用者回饋在不同資料下的反應。

第二節 評估方法

本論文所提出的演算法，主要的目的是希望能「依照使用者的興趣為目標」來引導分群，傳統分群系統的評估法，如 intra-cluster, inter-cluster…等等並不符合我們的目的。因此參考 Slonim (2000)、Liu (2002)，透過改良監督式文件分類系統中的（Supervised Text Classfifcation）準確度（Accuracy）來做評估。原公式如方程式 11：

$$Accuracy = \frac{\sum_{i=1}^{N} \delta(\alpha_i, Ans_i)}{N}$$

方程式 11：原 Accuracy 計算公式[Liu02]

當 x=y 時，δ(x,y)=1，否則δ(x,y)=0；而 α_i 是系統給第 i 篇文件的標記，Ans_i 是第 i 篇文件的人工答案標記。但考慮到我們系統額外加入「其他」這類標記，所以將原公式改良如方程式 12。其中，當 x≠「其他」、y≠「其他」，且 x=y 時，$\delta'(x,y)=1$，否則 $\delta'(x,y)=0$；而 α_i 是系統給第 i 篇文件的標記，Ans_i 是

$$Accuracy = \frac{\sum_{i=1}^{N} \delta'(\alpha_i, Ans_i)}{N\text{-}Sum_{other}}$$

方程式 12：改良後 Accuracy 計算公式

第 i 篇文件的人工答案標記，Sum_{other} 是被系統標記為「其他」的文件篇數。

為避免標記時，「其他」類過多，造成 Accuracy 很高，但實際系統效能不好的誤判情形，我們還加入了資訊檢索系統的評估法，來評估實驗文件中，各類文件的求全率 (Recall)，其公式如方程式 13：

$$Recall(Class_A) = \frac{\|\alpha_A\|}{\|Ans_A\|}$$

方程式 13：Recall 公式

其中，$\|\alpha_A\|$ 是系統將文件標記為「A 類」的總篇數、$\|Ans_A\|$ 是人工答案中「A 類」標記的總篇數。當「其他」類過多時，將會反映在 Recall 的數值上 (Recall 會很低)。因此，要同時對此兩項評估指標做觀察，才能得知系統的實際效能表現。

第三節 實驗結果與討論

4.2.1 實驗一

此實驗目的在於比較 GOSOM 依使用者特定的觀點分群的效能，藉由 Accuracy 跟 Recall 來估計其效能。在斷詞切字的部分，我們將在整個資料集 (Data Set) 中，只出現 1 次 (Tf=1) 的詞去掉，不予使用。表格 3 是其實驗結果。

從實驗結果中我們可以發現，隨著不同的實驗資料，Accuracy 跟 Recall 的值會跟著相對變動。針對特定概念的引導分群，在 Accuracy 上，GOSOM 平均將 SOM 改善 24.6%；在 Recall 方面則改善 24.25%。

	第一類實驗資料 1		第一類實驗資料 2	
Method	SOM	GOSOM	SOM	GOSOM
Size	6*6	6*6	6*6	6*6
Accuracy	0.44	0.56	0.59	0.72
Recall	0.548	0.693	0.737	0.9

表格 3：實驗 1 結果數據

4.2.2 實驗二

此實驗目的在於希望證明雖然文件數變多，分群目標較為模糊，但系統的效能仍能穩定地維持一定水準，依使用者喜好，將關於某一主題的文字內容，依使用者特定的觀點分群，並藉由 Accuracy 跟 Recall 來估計其效能。表格 4 是其實驗結果。

從實驗數據中，我們可以發現，與分類目標數同為 3 的第一類實驗資料相比，雖然在 Accuracy 方面改善效果下降至 18.75%，但在 Recall 方面的改善卻上升至 32.7%。因此，我們認為 GOSOM 對於 SOM 的改善效果尚稱穩定。

	第二類實驗資料 1	
Method	SOM	GOSOM
Size	6*6	6*6
Accuracy	0.48	0.57
Recall	0.424	0.563

表格 4：實驗 2 結果數據

4.2.3 實驗三

此實驗目的在於測試我們提出的使用者相關回饋策略，是否能進一步改善 GOSOM 效能。若可，則表示策略成功。此外，由於我們的使用者相關回饋機制是調整詞權重，所以實驗結果亦可證明透過其他方法，只要能提升詞權重的

品質，GOSOM 的效能表現會更好。與前兩個實驗相同，我們透過 Accuracy 跟 Recall 來估計其效能。表格 5 是其實驗結果。

	第一類實驗資料 1		第一類實驗資料 2		第二類實驗資料 1	
使用者回饋前後	回饋前	回饋後	回饋前	回饋後	回饋前	回饋後
使用者勾選出的正確答案比例	0	0.1	0	0.15	0	0.15
Accuracy	0.56	0.61	0.72	0.78	0.57	0.525
Recall	0.693	0.751	0.9	0.975	0.563	0.527

表格 5：實驗 3 結果數據

在第一類實驗資料中，我們的使用者相關回饋機制在 Acuuracy 及 Recall 的兩項測量上，效能都有提升 — 第一組資料 Acuuracy 較回饋前改善 8.9%，Recall 改善 8.3%；第二組資料 Acuuracy 較回饋前改善 8.3%，Recall 改善 8.3%。這表示了透過使用者相關回饋來調整詞權重，的確可以在提升 GOSOM 的效能。但在第二類的實驗資料中，Recall 及 Accuracy 都變差，這表示若策略失敗，無法將真正重要的特徵反映於詞權重，GOSOM 的效能便會下降。至於在第一類實驗資料中，改善效果會明顯比第二類來的好，其主要原因是由於我們在 3.1.4 提出的策略中，步驟 3 會依詞 – 詞關係矩陣，將原本使用者勾選文件中的主要特徵加以延伸，找出其他相關的詞，加強其詞權重，所以步驟 3 對於詞 – 詞關係矩陣的品質反應較為靈敏。而第二類實驗資料的使用者目標較為模糊，所以經由 LSA 方法找出的詞 – 詞關係矩陣，較無法有效凸顯重要的詞。因此，在詞 – 詞關係矩陣品質較不好的情形下，步驟 3 的延伸動作效果便減弱，而容易對詞權重做不適當的調整。

4.2.3 綜合討論

綜合前面三個小節的實驗，我們可以發現，GOSOM 的效能表現關鍵，在於詞權重是否能適度加強，而這跟詞 – 詞關係矩陣有直接的關係。在本論文提

出應用 LSA 方法的架構下，使用者目標若為較清晰的概念，GOSOM 的分群結果會較好。因此，若未來能找到比 LSA 更有效的方法，能將使用者目標所代表的概念，擴展到其他詞上，將能使 GOSOM 更具實用性。

關於使用者相關回饋：要根據使用者的意見，對分群系統作有效而顯著的改進並不容易。本論文提出的使用者相關回饋機制，受詞－詞關係矩陣影響太大，因此即便使用者正確地表達了意見，系統也不一定能有顯著改善。如何參考類似 LVQ[Kohonen98]的演算法，找到在 SOM 中可做使用者相關回饋機制的切入點，從而提出有效的回饋法，是個可嘗試的方向。在圖形檢索 (Image Retrieval) 方面，有人提出將輸入資料的特徵分開，套用多個 SOM，來落實使用者相關回饋機制，亦或是個可行的方法。但如何應用在本系統中，仍待研究[Laaksonen01]。

第五章 結論與未來研究方向

第一節 結論

本論文基於現今使用者的實際需求，提出一個目標導向之 SOM (Goal-Oriented SOM, GOSOM) 模型，並以其為核心，實作目標導向文件分群系統(Goal-Oriented Document Clustering System, GODOC)。本論文主要貢獻有：

1. 融合 LSA (Latent Sematic Analysis) 方法以改良 SOM，使其具備目標導向分群的能力。
2. 提出一權重的多數決 (Weighted Majority Voting) 群聚標記 (Cluster Labeling) 方法，其具有目標導向能力，可配合使用者設定的目標進行群聚標記，以利使用者辨別分群結果，容易理解各個群聚的內容
3. 提出一個能配合 GOSOM 的使用者相關回饋 (User Relevance Feedback) 機制，讓文件分群的結果，更符合使用者的要求。

實驗結果分別在準確率 (Accuracy)、求全率 (Recall) 方面，平均較 SOM 改善了 21.67%、28.47%，這表示提出的方法，確實表現的比之前的方法要好。而本論文提出針對 GOSOM 設計的使用者相關回饋機制，也透過實驗被證明是可行的。

綜合以上論述可知，「目標導向之 SOM」模型，的確能符合使用者的需要。而透過本論文所提出的方法跟實驗，驗證了這個模型的確可以成功地被實作完成。

第二節 未來研究方向

本論文為了融入使用者目標的概念，使用 LSA 來導入概念空間 (Concept Space) 的觀念，獲得了初步成功。若能透過其他的方式來結合使用者概念跟文件關鍵字，例如知識本體 (Ontology) 或語意網路 (Semantic Network)…等等，使其獲得更準確的「使用者概念 – 文件關鍵字」關係，相信能獲得更進一步的效果。另外，未來關於 SOM 模型仍可加以改良的部分，有下列兩點：

- 鄰近區域函式 (Neighborhood Function)：從 2.1 中所提到的方程式 3，一個文件對其勝利點對其周圍的鄰近點，影響力的大小，對 SOM 的分群結果應該是關鍵的。但如何將賦予其意義，做有效的應用，仍然不甚清楚。
- 模型向量 (Model Vector)：SOM 的分群結果，能否用更具威力、具其他意義的表示法，而不單單只是一個模型向量呢？關於這方面的研究，非常的多。在本系統中，若從改良模型向量著手，或可將對於知識專家非常有用的知識本體 (Ontology) 加入，使得系統成為更具實用性。

最後，如何找到在 SOM 中可做使用者相關回饋機制的切入點，從而提出有效的回饋法，也是我們未來努力的方向。

參考文獻

[Bhuyan91] J. N. Bhuyan, J. S. Deogun, and V. V. Raghavan, “Cluster-Based Adaptive Information Retrieval,” *Proc. of the 24th Hawaii International Conference on System Sciences-Architecture and Emerging Technologies Tracks,* vol. 1, pp. 307-316, 1991

[Bhuyan89] J. N. Bhuyan, J. S. Deogun, and V. V. Raghavan, “Near Optimal Algorithm for Boundary Selection Problem in User-Oriented Information Retrieval,” *Information Processing 89*, *Proc. of the IFIP 11th World Computer Congress*, pp. 275-280, 1989

[Cutting92] D. R.Cutting, D. R. Karger, J. O. Pederson, and J. W. Tukey, “Scatter/gather: A cluster-based approach to browsing large document collections,” *Proc. of the 15th Annual International ACM Conference on Research and Development in Information Retrieval*, pp. 318-329, 1992

[Deerwester90] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing By Latent Semantic Analysis,” *Journal of the American Society For Information Science*, vol. 41, no. 6, pp. 391-407, 1990

[Deogun86] J. S. Deogun, and V. V. Raghavan, “User-Oriented Document Clustering,” *Proc. of the 9th annual international ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 157-163, 1986

[Dittenbach02] M. Dittenbach, A. Rauber, and D. Merkl, “Uncovering hierarchical structure in data using the growing hierarchical self-organizing map,” *Neurocomputing*, vol. 48, no.1-4, pp. 199-216, 2002

[Guerrero Bote02] V. P. Guerrero Bote, F. M. Anegon, and V. H. Solana, “Document organization using Kohonen’s algorithm,” *Information Processing Management,* vol. 38, no. 1, pp. 79-89, 2002

[Hagenbuchner03] M. Hagenbuchner, A. Sperduti, and A. C. Tsoi, “A Self-Organizing Map for Adaptive Processing of Structure Data,” *IEEE Transactions on Neural Networks,* vol. 14, no. 14, pp. 491-505, 2003

[Shah-Hosseini03] H. Shah-Hosseini, and R. Safabakhsh, “TASOM: A New Time Adaptive Self-Organizing Map,” *IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics,* vol. 33, no. 2, pp. 271-282, 2003

[Haykin94] S. Haykin, *Neural Networks: A Comprehensive Foundation.* Macmillan, 1994

[Hatano99] K. Hatano, R. Sano, Y. Duan, and K. Tanaka, “An Interactive Classification of web Documents by Self-Organizing Maps and Search Engines,” *Proc. of the 6th International Conference on Database Systems for Advanced Applications,* pp. 35-42, 1999

[Honkela96] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, “Newsgroup exploration with WEBSOM method and browsing interface,” *Technical Report A32 Helsinki University of Technolnogy, Laboratory of Computer and Information Science Espoo Finland*, 1996

[Jiang97] MS Thesis of ingqian Jiang, “Using Latent Semantic Indexing for Data Mining,” *Department of Computer Science, University of Tennessee,* 1997.

[Kangas96] J. Kangas, and T. Kohonen, “Developments and applications of the self-organizing map and related algorithms,” *Mathemetics and Computers in Simulation,* vol. 41, no.1-2, pp. 3-12, 1996

[Kaski97] S. Kaski, “Computationally efficient approximation of a probabilistic model for document representation in the WEBSOM full-text analysis method,” *Neural Processing Letters,* vol. 5, no 2, pp. 139-151, 1997

[Kaski98] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen, “WEBSOM — Self-organizing maps of document collections,” *Neurocomputing,* vol. 21, no.1-3, pp. 101-117, 1998

[Kaski99] S. Kaski, “Fast Winner Search for SOM-based Monitoring and Retrieval of High-Dimentional Data,” *the 9th International IEE Conference on Artificial Neural Networks,* vol. 2, pp. 940-945, 1999

[Kim01] H. J. Kim , and S. G. Lee, “A semi-supervised document clustering technique for information organization,” *Proceedings of the 9th International*

Conference on Information and Knowledge Management, pp. 30-37, 2000

[Kohonen84] T. Kohonen, *Self-organization and associative memory.* Springer Verlag, 1984

[Kohonen95] T. Kohonen, *Self-organizing map.* Springer, 1995

[Kohonen98] T. Kohonen, "The self-organizing map," *Neurocomputing,* vol. 21, no. 1-3, pp. 1-6, 1998

[Kohonen00] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, and J. Honkela, "Self Organization of a Massive Document Collection," *IEEE Transactions on Neural Networks,* vol. 11, no. 3, pp. 574-585, 2000

[Lin96] C. S. Lin, and H. Chen, "An Automatic Indexing and Neural Network Approach to Concept Retrieval and Classification of Multilingual (Chinese-English) Document," *IEEE Transactions On Systems, Man, And Cybernetics-part B: Cybernetics,* vol. 26, no. 1, pp. 1-14, 1996

[Landauer98] T. K. Landauer, P. W. Foltz, and D. Laham, "Introduction to Latent Semantic Analysis," *Discourse Processes,* vol. 25, no. 2-3, pp. 259-284, 1998

[Laaksonen01] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja, "Self-Organizing Maps as a Relevance Feedback Technique in Content-Based Image Retrieval," *Pattern Analysis and Applications*, vol. 4, no. 2-3, pp. 140-152, 2001

[Liu02] X. Liu, Y. Gong, and W. Xu, "Document Clustering with Cluster Refinement and Model Selection Capabilities," *Proc. of the 25th annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 191-198, 2002

[Mavroudi02] S. Mavroudi, S. Papadimitriou, and A. Bezerianos, "Gene expression data analysis with a dynamically extended self-organized map that exploits class information," *Bioinformatics,* vol. 18, no.11, pp. 1446-1453, 2002

[Martin-Merino01] M. Martin-Merino, and A. Munoz, "Incorporating Asymmetry into SOM and Sammon Algorithm for Visual Map Generation," *Proc. of International Joint Conference on Neural Networks,* pp. 1908-1913, 2001

[NIST] http://math.nist.gov/javanumerics/jama/

[Orwig97] R. E. Orwig, H. Chen, and J. F. Nunamaker, “A Graphical, Self-Organizing Approach to Classifying Electronic Meeting Output,” *Journal of the American Society for Information Science,* vol.48, no.2, pp. 157-170, 1997

[Raghavan87a] V.V. Raghavan, and B. Agarwal, “Optimal Determination of User-Oriented Clusters: An Application for the Reproductive Plan,” *Prof. of the 2nd International Conference on Genetic Algorithms*, pp. 241-246, 1987

[Raghavan87b] V. V. Raghavan, and J. S. Deogun, “Optimal Determination of User-Oriented Clusters,” *Proc. of the 10th Annual ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 140-146, 1987

[Roussinov01] D. G.. Roussinov, and H. Chen, “Information navigation on the web by clustering and summarizing query results,” *Information Processing and Management,* vol. 37, no.6, pp. 789-816, 2001

[Ressom03] H. Ressom, D. Wang, and P. Natarajan, “Adaptive double self-organizing maps for clustering gene expression profiles,” *Neural Networks*, vol. 16, no. 5-6, pp. 630-640, 2003

[Salton88] G.. Salton ,and C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513-523, 1988

[Sebastiani02] F. Sebastiani, ” Machine Learning in Automated Text Categorization,” *ACM Computing Surveys,* vol.34, no. 1, pp. 1–47, 2002

[Slonim00] N. Slonim ,and N. Tishby, “Document Clustering using Word Clusters via the Information Bottleneck Method,” *Proc. of the 23rd annual international ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 208-215, 2000

[Su01] M. C. Su, and H. T. Chang, “A new model of self-organizing neural networks and its application in data projection,” *IEEE Transactions on Neural Networks*, vol.12, no.1, pp. 153-158, 2001

[Yu85] C. T. Yu, T. Wang, and C. H. Chen, “Adaptive Document Clustering,” *Proc. of the 8th ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 197-203, 1985