A Concept Extraction Approach for Document Clustering and Visualization

Student：Chia-Ning Chang

Supervisors：Dr. Hao-Ren Ke
Dr. Wei-Pang Yang

Institute of Computer Science and Engineering

National Chiao Tung University

ABSTRACT

The World Wide Web (WWW) contains a giant amount of information, but finding relevant information from WWW is also a great challenge. Keyword-based querying usually returns many documents; however, they are neither strongly related nor presented in a comprehensible order. Clustering is capable of solving such a problem by grouping relevant documents. Users are able to find relevant documents through groups containing documents with similar concepts.

This thesis attempts to extract concepts from a corpus, each of which is defined as a collection of keywords in documents, and conduct document clustering on the basis of the extracted concepts. The overall processes are as follows. First, a clustering algorithm groups similar keywords to create concepts. Second, a document is represented by a vector, each element of which indicates the similarity between the document and a concept. Then, documents are clustered according to the abovementioned vector. Furthermore, citations between documents are used to construct documents connections. Such connections are further used for discovering group relations and concept relations. In addition to extracting concepts and clustering documents, this thesis uses the visualization technique to present clustering results and show the relationship between concepts. Several experiments with CiteSeer documents are performed in order to show that concepts extracted by our method can not only clearly represent each group, but also achieve good clustering accuracy, which is about 80%.

Keywords: Document Clustering, Keyword Clustering, Concept Extraction, Topic Keyword, Visualization, Citation

以概念萃取為基礎之文件分群與視覺化

研究生：張家寧

指導教授：柯皓仁 博士
楊維邦 博士

國立交通大學資訊科學與工程研究所

摘要

近年來，網際網路已經成為取得資訊最方便的管道，其中又以在搜尋引擎輸入關鍵字取得資訊的方式最為普遍。然而，搜尋引擎通常不會對搜尋結果進行過濾與篩選，過多的資料提高了評估資料相關性的複雜度，如何在獲取的資料中去蕪存菁，並建立出容易讓使用者了解的模型，進而讓資料有效率地轉化為使用者容易吸收的知識，是目前重要的研究課題之一。分群演算法可以將資料分析之後，依照相似度將類似的資料群聚，不同的群具有不同的含意與概念，如何從群中自動萃取出其含意並賦予概念，是本研究的主要目的之一。

本研究提出以關鍵字分群的方式達到概念萃取的目的，且將文件以多種概念描述後，基於這些概念進行文件分群。進行概念萃取主要分為以下幾個主要的步驟：特徵選擇、特徵關係的建立，以及特徵分群；特徵分群的結果即為所有文件包含的概念。此外，透過文件內引用文章 (Citing Article)的相似度，建立文件間的引用關係 (Citation Relation)，進而建立群與群之間的引用關係，達到建立概念之間的相關性。最後，取代傳統條列式的顯示方式，以視覺化的方式展現分群結果並呈現出概念之間的相關性。

本研究採用 CiteSeer 資料庫的論文做為語料庫，選取標題、摘要及引用做為資料來源，摘要部分所收錄的文字大約只有 1000 個字元，這個數量相當於在搜尋引擎中以關鍵字查找所得到的結果資料。根據實驗結果分析，本研究所萃取出的概念可以適合地表達出文件的整體概念，在文件分群的準確率(Accuracy)上亦有一定水準，可達到 80%的準確率。

關鍵字：文件分群、關鍵字分群、概念萃取、主題關鍵字、視覺化、引用

誌謝

隨著這兩年的研究所生涯到達尾聲，六年的新竹生活也即將畫下句點，一路走來，心中除了感謝還是感謝。

首先感謝三位指導老師柯皓仁、黃明君以及楊維邦老師。感謝楊老師引領我入門，得以在溫暖的資料庫實驗室裡開啟我的研究所生涯；感謝柯皓仁、黃明居老師不厭其煩的細心指導與建議，引導我一步步地將論文順利完成。三位老師除了指引課業的方向之外，亦培養學生獨立思考的學習態度，而這些都將成為我展開下一段人生旅程的知識寶藏。

再來感謝實驗室博士班學長們這兩年來多方面的照顧。感謝鎮源學長，當實驗評估遇到瓶頸時，不斷地提供多種方法與建議，同時也分享經驗並且指點迷津；感謝忠億學長，總是耐心地指導我撰寫論文的技巧，雖然人在美國仍是零時差地給予我鼓勵；感謝信源學長，不但提供免費又好喝的咖啡，在 meeting 時也傳授了不少寶貴的經驗與觀念。除了學長們之外，還有實驗室同學昕潔，不論在課業上或是生活上，總是一起努力及互相打氣的好伙伴，共同分享這兩年的酸甜苦辣。感謝資料庫實驗室的大家，與大家相處的點點滴滴，都將是我人生中美好的回憶。

最後則是感謝我的家人無條件的包容與愛護，讓我能無後顧之憂的繼續朝目標前進，有你們的支持才有現在的我！

June, 2006

目錄

表目錄

圖目錄

第一章　緒論

1.1 研究動機與目的

隨著資訊爆炸的時代來臨，利用網路取得資訊已經成為最方便的管道，搜尋引擎就是一個最好的例子，在搜尋引擎輸入關鍵字之後，便可以取得許多相關資訊，找不到資料已不再是最大的煩惱。然而，透過搜尋引擎搜尋到的資料大多是基於關鍵字匹配所獲得，而且為了提升所尋得之資料量，通常不會進行過濾與篩選。於是，過多的資料提高了資料的複雜度，也增加了使用者取得符合需求的資料的困難度。另外，在搜尋結果當中，每一筆資料都是互相獨立的，無法得知哪些資料是屬於同質性，哪些則是完全不相關；若能將搜尋結果經過有系統的整理及分析，分成多個以主題概念表示之類別，再讓使用者根據其目的選擇相對應的類別，將可以有效減低資料的複雜度，引導使用者獲取真正有幫助之資訊。同時，當替搜尋結果建立類別之後，可以更進一步地建立出類別之間的關係，並且表達出類別之間的相關性，讓使用者更容易了解搜尋結果的意義，或是以此相關性去進行更深入的瀏覽與學習。

所謂的對搜尋結果進行分群(Clustering)，即是對搜尋結果進行分析之後，依給定的相似度評估法則，將相似的搜尋結果以群的方式聚集，並在呈現時以群做為單位。分群主要應用在非監督式資料(Unsupervised Data)中，所謂的非監督式資料是指資料的特性模糊(Fuzzy)，或是資料量不足以分割出可以讓分類演算法學習之訓練資料(Training Data)；由於沒有可供分類的相關資訊，分群通常是利用統計上的方法，將資料量化後，以資料之間的相似度判斷同質性。若將分群應用在搜尋引擎上，則是將每一筆搜尋結果（即一篇文件）的內容經過處理及過濾之後，再利用文件分群(Document Clustering)演算法將相似的資料群聚在一起，以便使用者更容易組織及瀏覽搜尋結果。

本論文的研究目的是希望利用以關鍵字為基礎的分群演算法在搜尋結果中有效率及準確地萃取出其所包含的概念(每一個概念是由一些相似的關鍵字所組成)，並以這些概念進行文件分群，同時建立概念之間的相關性，輔以視覺化的方式讓使用者瀏覽搜尋結果，並提供更多相關資訊給使用者參考。

1.2 研究方法與範圍

大多數的文件分群演算法都是將文件內容轉換成向量空間(Vector Space)，向量空間的特性為可以透過距離表示文件間的相似度，並且將相似度高的資料群聚在一起。直覺上來說，表達文件的方式通常是選取文件中較具有代表性的關鍵字做為特徵，將這些特徵給予較多的權重，顯示它在文件當中的重要性。但是事實上，單單一個關鍵字並不足以表示整篇文章的概念，一個概念應該是由多個關鍵字組合而成；除此之外，有時文章會包含二個以上的概念，當兩篇文章在多個概念上都有某種程度的相似時，即可以認定這兩篇文章是相似的。

本研究希望透過特徵選擇(Feature Selection)之方法，取出文件中重要的關鍵字，再利用關鍵字分群的方式達到概念萃取的目的。且將文件以多種概念描述後，基於這些概念進行文件分群。最後，取代傳統條列式的顯示方式，以視覺化的方式展現分群結果並呈現出概念之間的相關性。

另外，本論文的研究範圍是專注於英文論文文件之概念萃取，而不考慮一般性的非學術文章，如新聞文件與一般網頁等；而在論文概念萃取方面，由於作者對於概念萃取沒有太大的幫助，為了簡化系統的複雜度以及處理人名的問題，因此不加入作者相關資訊，僅利用文件的標題與摘要進行概念萃取。

1.3 論文架構

本論文首先在第二章介紹文件分群演算法的相關研究，其中包含自然語言處理、分群演算法、分群準則、文件分群演算法及視覺化之應用；第三章闡述概念萃取與文件分群之演算法設計，以及視覺化的方法及過程。第四章說明實驗結果並根據實驗結果進行討論。最後，在第五章總結本研究，並探討未來的發展研究方向。

第二章　相關研究工作

本章說明文件分群及視覺化應用的相關研究工作。首先介紹文件分群，共分為 1) 分群演算法(Clustering Algorithm), 2) 分群準則(Clustering Criterion Function), 3) 文件分群演算法(Document Clustering)。其次，則接著介紹視覺化(Visualization)概念及應用。圖 1 為依年份及研究範疇整理相關研究發展過程。

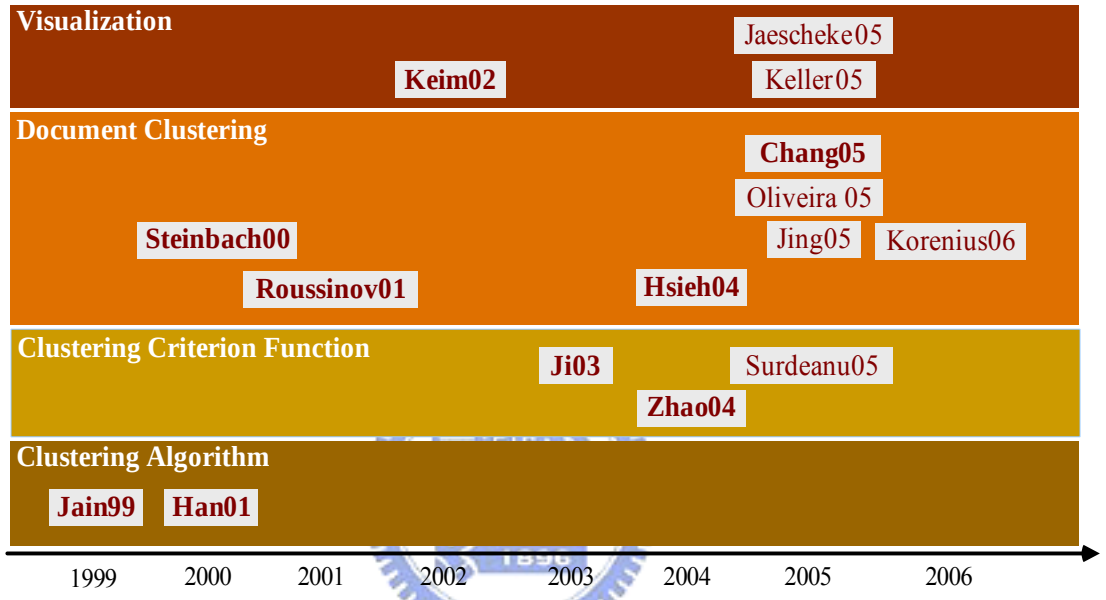

圖 1: 文件分群及視覺化應用之相關研究發展

2.1 分群演算法

本節介紹具代表性並與本論文相關的分群演算法，包括：1) 劃分式分群法(Partitioning Methods); 2) 階層式分群法(Hierarchical Methods); 3) 基於模型分群法(Model-based Methods) [8]，並以 *k*-Means 演算法 [17]，Agglomerative & Divisive 演算法 [16]及 Self-Organizing Map (SOM) [22]演算法為例說明。除了說明這些代表性的分群演算法之外，在 2.1.4 節中會介紹本研究在萃取文件概念主要參考的演算法：Topic Keyword Clustering [2]。

2.1.1 劃分式分群法 – 以 *k*-Means 為例

劃分式分群法的作法為給定 n 個資料物件及參數 k，將 n 個資料物件分成 k 群，一

個劃分代表一個群，換句話說，在分群前必須先定義目標分群個數，此亦為劃分式分群法的特色。*k*-Means [17]為一個應用廣泛的分群演算法。分群過程中同時須確保兩個條件：1) 位於相同群內的物件，彼此間相似度高，此處的相似度定義為物件與群中心點的歐基里德距離(Euclidean Distance)，其中群中心點為即為該群的重心，即所有物件的向量平均值，物件與群中心點的距離愈小表示相似度愈高; 2) 位於不同群內的物件，彼此間相似度低，意即屬於不同群的物件其歐基里德距離愈大愈好。*k*-Means 演算法的詳細步驟如下所示：

k-Means 演算法：

輸入：1) *k* 值; 2) *n* 個物件

輸出：*k* 個群

1 於 *n* 個物件中隨機選取 *k* 個物件做為初始群，並以其為群中心點。

2 重複以下步驟，直到群的分佈不再改變

2.1 依序計算每個物件與 *k* 個群之群中心點距離做為該物件與群中心點的相似度，同時將每個物件指定給與其相似度最高的群。

2.2 重新計算該群的群中心點

判斷群的分佈是否會再改變的標準是依照判斷準則函數，當函數收斂時即表示群的分佈不再改變，判斷準則函數之定義如方程式(1)：

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 \tag{1}$$

其中，E 表示所有物件與其所屬群之群中心的平方差總和，p 為屬於群 C_i 的某物件，m_i 為 C_i 的中心點，且 p 及 m_i 皆為多維的向量。該準則主要的目的在於盡可能地讓每個群內部物件的關係緊密。

k-Means 測試不同的分割方式，試圖找出具有最小 E 值的 *k* 個群劃分。因此，當群越密集，且群與群間有明顯區隔時，*k*-Means 的表現較佳。然而，*k*-Means 初始時隨機選擇 *k* 個物件當做初始群，其分群結果會受到初始群的影響，因此 *k*-means 演算法經常得到的是一個局部最佳化(Local Optimum)的群劃分。此外，*k*-means 的計算複雜度

(Computational Complexity)為 $O(nkt)$，n 是所有物件的數目，k 是叢集的數目，t 是疊代(iteration)的次數。一般而言，$k<<n$ 且 $t<<n$；因此若需處理的大量資料時，k-Means 亦可成比例延展，同時維持相當的效率。

k-Means 最大的缺點可分為四方面討論。首先，k-Means 只適合用於能定義中心點的群。第二，必須預知分群的群數；第三，若每群內的物件個數差距過大，或是群的形狀不為凸多邊形，都較不適合使用 k-Means [8]；最後，由於 k-Means 是根據群中心點做為分群的基礎，當資料量過小時，容易受到雜訊的影響，進而影響分群的結果。

2.1.2 階層式分群法 – 以 Agglomerative & Divisive 為例

本節將介紹另一種常見的分群方法 — 階層式分群法 [8]。階層式分群法乃是訂定終止條件，當滿足終止條件時即停止程序。判斷停止程序的終止條件通常定義為達到目標分群數，或是經由合併或分化後，群與群之間的相似度達到門檻值等。階層式分群法與 2.1.1 節中說明 k-Means 演算法最大的不同之處，在於其並不預先將物件分割為 k 群。

一般而言，此類演算法的架構相似於樹狀圖，分群方式可大略分為以下兩種 [11]：

1. 凝聚式(Agglomerative)

   凝聚式的分群法為一種自下而上(Bottom-Up)的分群方式。初始時，每個物件自成一群，計算兩兩群的相似度，當相似度大於既定的臨界值時，則合併兩群為一個較大的群，直到所有的物件都屬於同一個群，或是符合終止條件才停止。

2. 分裂式(Divisive)

   分裂式的分群法為一種由上而下(Top-Down)的分群方式。初始時，將所有的物件視為同一群，再依照物件間的相似度分割為較小子群，直到每個子群都只有一個物件或是符合終止條件才停止。

假設有物件{a, b, c, d, e}，圖 2 舉一個簡單的例子說明上述兩種演算法。

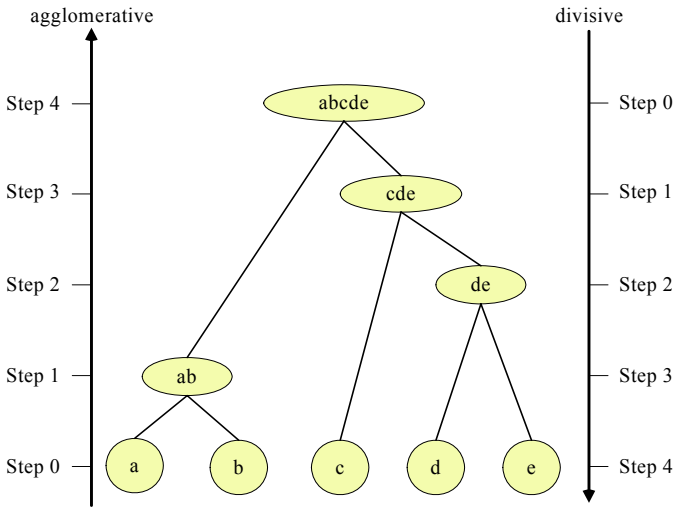

圖 2: 階層式演算法

由圖 2 可知，分裂式分群法實為凝聚式的反向。判別兩群是否可以進行合併(或分割)，通常是依據既定的準則。常用的準則如方程式(2)(3)(4)(5)，其中 C_i, C_j 代表群，m_i, m_j 是群 C_i, C_j 的群中心，p 及 p' 代表群中的物件，$|p-p'|$ 表示物件間的距離。

$$\text{Minimum distance: } d_{\min}(C_i, C_j) = \min_{p\in C_i, p'\in C_j} |p-p'| \tag{2}$$

$$\text{Maximum distance: } d_{\max}(C_i, C_j) = \max_{p\in C_i, p'\in C_j} |p-p'| \tag{3}$$

$$\text{Mean distance: } d_{mean}(C_i, C_j) = |m_i - m_j| \tag{4}$$

$$\text{Average distance: } d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p\in C_i} \sum_{p'\in C_j} |p-p'| \tag{5}$$

舉例來說，假設有兩個群 C_1 和 C_2，當 C_1 與 C_2 間有一組物件距離小於臨界值，就將 C_1 及 C_2 合併，換句話說，當兩群的最近距離小於臨界值即可合併，此種方式亦稱為 Single-Link 分群，如方程式(2)所示。當兩群的最大距離小於臨界值時才可合併，此種方式稱為 Complete-Link，如方程式(3)所示。

階層式分群法的概念雖然簡單，但是要決定在什麼條件下進行合併或分裂卻是非常困難。決定合併及分裂的條件是非常關鍵的，因為群一旦被合併或分裂，下一步驟的處理將依循上一步驟的結果繼續進行，並且群與群之間也不能交換物件。因此，若選擇不適當的合併或分裂的條件，可能會導致最終分群結果不佳。除此之外，此種分群方法的

可擴展性不佳，因為進行合併或分裂的處理時需要檢查及估算大量的物件或是群。

2.1.3 基於模型分群法 – 以 Self-Organizing Map 為例

基於模型分群法為試圖依照某些數學模型，將給定的資料最佳化，此種方法經常是根據下列的假設：資料是根據潛在的機率分佈生成的。其主要有兩類：統計學方法和類神經網路方法，本節介紹具分群及視覺化的 Self-Organizing Map (SOM) [22]演算法，是屬於類神經網路方法的一種，SOM 的主要概念為將多維空間的資料對應到二維平面上，並且於二維平面上依然維持在多維空間中空間距離的關係。換句話說，在多維空間中距離相近的資料，於二維空間亦會被群聚在相近的點上，進而達成分群的目的。

SOM 應用於文件分群上，主要分成下列六個步驟 [28]：

1. 初始化輸入點(Input Node)及輸出點(Output Node)

   利用向量空間模型(Vector Space Model)將文件向量化，每個向量都代表一個輸入點，稱為輸入向量(Input Vector)。輸出點通常是排列為一個矩形平面，因此需設定矩形的長與寬以決定輸出點之個數。輸出點的維度與輸入點的維度相同，輸出向量可視為輸入向量所在空間的一點，啟始值可用亂數決定。

2. 選擇勝利點(Winning Node)

   依序用所有輸入向量對所有輸出點的模型向量做調整，計算每個輸入向量 X 跟所有輸出點模型向量之距離，模型向量與輸入向量 X 最近的輸出點，定義為勝利點。常用的距離計算公式為歐基里德距離，如方程式(6)所示，其中 $m_i(t)$ 表示第 t 次調整模型向量時，輸出點 i 的模型向量，Dim_j 表示向量中第 j 個維度的值。

$$Similarity\left(X, m_i(t)\right) = \sqrt{\sum_j \left(Dim_j(X) - Dim_j\left(m_i(t)\right)\right)^2} \qquad (6)$$

3. 調整模型向量

調整模型向量值的計算方法如方程式(7)所示。$h_{c(x),i,t}$ 主要是控制分群過程中模型向量的學習速度及影響鄰近區域點的能力。$h_{c(x),i,t}$ 的定義如方程式(8)所示，第 i 個輸出點在第 t 次調整模型向量時，將輸入向量 X 的勝利點 c 代入後，所得的函數值。其中 r_c、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。α(t)為一單調遞減(Monotonically Decreasing)函數，值介於 0~1 之間，用來影響模型的學習速度，亦控制勝利點影響鄰近區域點的能力。經此調整後，所有的模型向量向輸入向量 X 移動的程度都將有所不同。

$$m_i(t+1) = m_i(t) + h_{c(X),i,t} \times \left(X - m_i(t)\right) \tag{7}$$

$$h_{c(X),i,t} = \alpha(t) \exp\left(\frac{\|r_c - r_i\|}{2\sigma^2(t)}\right) \tag{8}$$

4. 重複步驟 2 及 3 直到 SOM 收斂或是達到循環次數的門檻值才繼續步驟 5 及 6
5. 指定群聚

當 SOM 分群過程結束後，計算每個輸入向量與所有輸出向量的距離，將輸入向量指定至距離最近的輸出點。當每一筆文件向量都指定至其最近的輸出點後，即可產生一個依照文件相似度分群的二維地圖 (2D Map)。

6. 標記(Labeling)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。在 [22] 提到一種適合文件分群的群聚標記法：根據每個輸出點的模型向量，選出座標軸值最大的相對應關鍵字來代表該點。若相鄰的輸出點有一樣的關鍵字，則合併為同一群。

2.1.4 關鍵字分群 Topic Keyword Clustering

Topic Keyword Clustering [2]，此演算法的基本精神相似於 Chameleon [12]。Chameleon 分群法隸屬於階層式分群法(Hierarchical Method)，採用動態模型(Dynamic Modeling)來進行分群 [8]。群與群能否合併取決於該群之間的互連性和相似度，只要定

義相似度函數即可應用於各類型的資料。Chameleon 分群法的步驟如下所列，流程圖如圖 3 所示。

1. 將資料建立成一個稀疏圖形 (Sparse Graph)
2. 以 *k*-Nearest Neighbor 演算法 [5]將圖形分割為多個子圖
3. 依照相似度定義將群合併

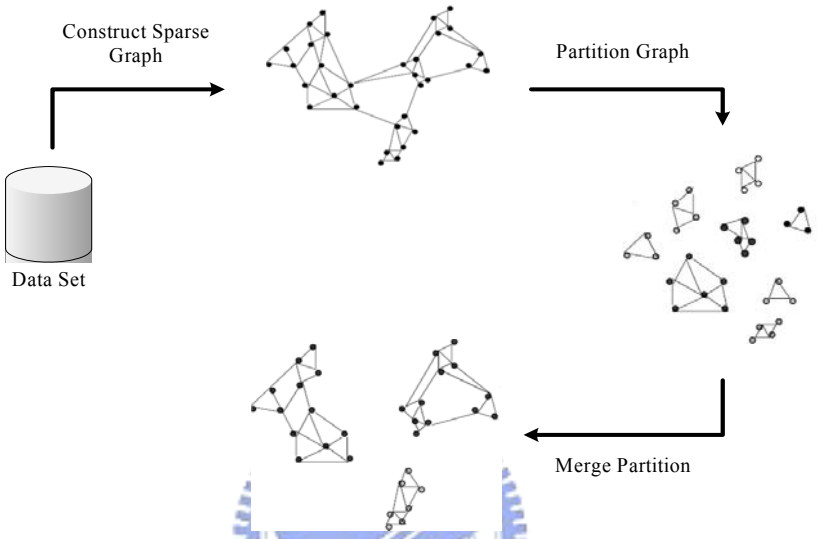

圖 3: Chameleon 分群演算法流程圖

Topic Keyword Clustering 分群法即是將 Chameleon 應用於文件分群。首先取出文件中的關鍵字(Keyword)，將關鍵字分群之後，計算文件與關鍵字子群之相似度，最後將文件對應至最相似的關鍵字子群。詳細分群步驟如下：

1. 選擇關鍵字

   根據先前研究，用來表示文件最適當的關鍵字個數為 10~25 個 [14]，過多的關鍵字反而會降低重要關鍵字的顯著性。因此，透過停用字集(Stop Word)可移除詞頻較高的關鍵詞及功能詞(Function Word)，以達到篩選關鍵字的目的。

2. 計算關鍵字間的關聯性

   當某些關鍵字經常同時於同一文章、段落或語句中出現，便可說這些關鍵字具有關聯性。利用 Mutual Information [1]，可以定義兩兩關鍵詞的共現關聯性作為定義兩

兩關鍵詞的關係。計算方式如方程式(9)，分子為兩關鍵詞共同出現的個數，分母則取該兩關鍵詞在語料庫中出現次數的最大值。

$$r_{ij} = f\left(t_i \cap t_j\right) \Big/ MAX\left(f\left(t_i\right), f\left(t_j\right)\right) \tag{9}$$

3. 建立關鍵字網路圖

利用關鍵字及語意關聯性建立網路圖，並將網路圖內的連線進行刪減，只保留大於平均語意相關度的連線，將原本的網路圖修正為稀疏網路圖。

4. 進行關鍵字分群

關鍵字分群的主要步驟如圖 4 所示。首先由稀疏網路圖中選取權重大於平均的點，並將這些點加入候選關鍵字集。點權重的計算方法如方程式(12)所示。w_i 表示點 v_i 的權重值，定義為方程式(11)；計算得出，r_{ij} 表示點 v_i 與相連點 v_j 間之語意相關度，m 則為與點 v_i 有線相連的點個數，$\sum_{j=1}^{m} r_{ij} \Big/ m$ 為點 v_i 及其相連點間的平均語意相關度。

$$\begin{aligned} & tf_{i,j} = freq_{i,j} \quad \Rightarrow \text{ frenquency of term } i \text{ in the document } d_j \\ & idf_i = \log_2 \frac{N}{n_i} \quad \Rightarrow \begin{cases} N: \text{ \# of documents} \\ n_i: \text{frenquncy of term } i \text{ in document collections} \end{cases} \\ & w_{i,j} = tf_{i,j} \times idf_i \Rightarrow w_{i,j}: \text{the weight of term } i \text{ in document d}_j \end{aligned} \tag{10}$$

$$w_i = \sum_{j=1}^{N} w_{ij} \tag{11}$$

$$CW_i = w_i + \frac{\sum_{j=1}^{m} r_{ij}}{m} \tag{12}$$

將候選關鍵字利用 *k*-Nearest Neighbor 進行分群，產生候選關鍵字組後，以每個候選關鍵字組為中心還原連線，每個關鍵字組都會產生一候選關鍵字子群，再由 Greedy 演算法找出候選關鍵字子群中互連性(Inter-Connected)最強的兩個群將之合併，直到子群間的互連相關度(Relative Inter-Connective)都小於門檻值後停止，即得

到關鍵字子群。

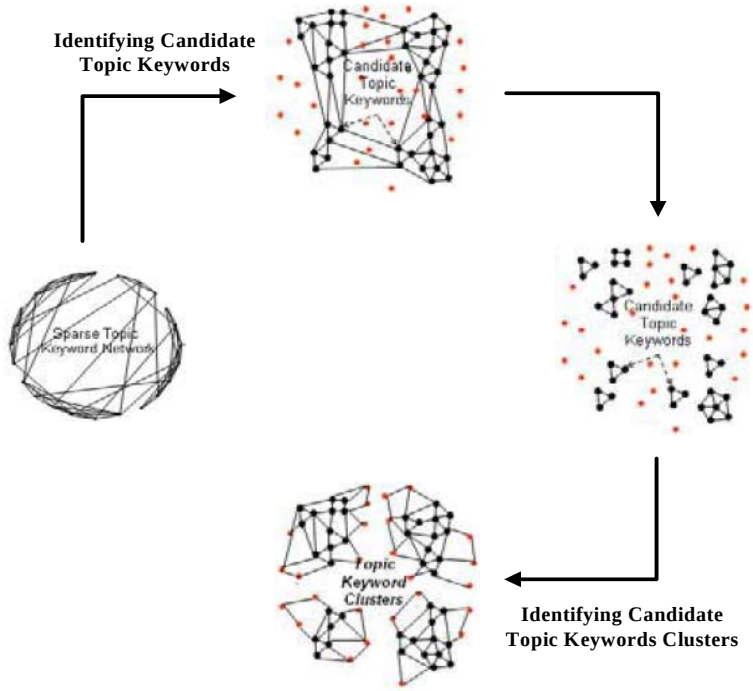

圖 4: Topic Keyword Clustering 主要步驟

5. 修正關鍵字分群結果及文件分群

在合併候選關鍵字子群的過程中，會造成每一群包含的關鍵字個數產生差距，同時也會影響文件分群的正確性，因此需要將每個子群內的關鍵字保持在一定的差距內。當子群內的關鍵字數目大於平均關鍵字數目時，則依序移除與群本身最不相關的關鍵字，以圖形的觀點來說，即為權重最小的點。關鍵字子群修正完畢後，將文件與每一個關鍵字群以 Cosine Similarity 計算相似度，並將文件對應至相似度最高的關鍵字子群中，以達到文件分群的目的。

2.2 分群準則

一般說來，群的內聚力 (Cluster Compactness)高表示群內的物內之間相似度高，群的分離度高 (Cluster Separation)則表示不同群的物件相似度低，故群的內聚力及分離度可以做為判別分群優劣的方法之一。分群準則的目的則是將與分群相關的這些特性加以最佳化，以提高分群的準確率。本節介紹五種分群時常用的準則，依序為 I_1、I_2、E_1、

H_1 及 H_2，計算分式整理如方程式(13)(14)(15)(16)(17)。

$$I_1 = \max \sum_{r=1}^{k} n_r \left(\frac{1}{{n_r}^2} \sum_{d_i, d_j \in S_r} \cos\left(d_i, d_j\right) \right) \tag{13}$$

$$I_2 = \max \sum_{r=1}^{k} \sum_{d_j \in S_r} \cos\left(d_i, C_r\right) \tag{14}$$

$$E_1 = \min \sum_{r=1}^{k} n_r \cos\left(C_r, C\right) \tag{15}$$

$$H_1 = \max \frac{I_1}{E_1} \tag{16}$$

$$H_2 = \max \frac{I_2}{E_1} \tag{17}$$

I_1 測試屬於同一群的兩兩物件相似度，將其兩兩物件相似度加總後平均能達到最大值，S_r 為群 r ，n_r 為群 r 中包含的物件個數，d_i 及 d_j 表示物件 i 及物件 j。I_2 的目的為將物件對應至與其相似度最高的群，計算物件與群相似度的方法乃是考慮該物件與該群的群中心 C_r 相似度。I_2 亦為物件分群時常用準則。E_1 的目的為使得群與群之間的相似度達到最小值；群與群相似度的計算方式為將每一群的中心點與所有物件的中心點 C 計算相似度後，再乘上該群包含的物件個數，故當群包含的物件愈多，其權重也會愈大，相對值也會提高。H_1 與 H_2 兩個準則公式將前三個準則加以應用結合，同時考慮群內的相似度強度及群與群之間的不相似度。其目標為 I_1 及 I_2 值盡可能大且 E_1 的值盡可能小。

一般而言，I_1 與 I_2 可以得到比較好的分群結果，E_1 則可以得到群大小較相近的分群結果。然而，如何選擇適當的分群準則，必須考慮到分群的物件類別及分群結果的應用，並透過實驗嘗試才能得知。

2.3 字詞語意關聯度

進行文件分群前，必須透過自然語言處理進行語意分析，以了解文件的主題及概念。本節將介紹幾個以統計為基礎的字詞相關度計算方法。

2.3.1 Pearson's Chi-Square Test

Pearson's Chi-Square Test [4]，又稱為卡氏檢定。對於兩個字詞 t_i, t_j，假設其出現的機率為獨立，彼此間不具相關性，但當實際出現的頻率與期望出現的頻率差距過大時，即表示字詞 t_i, t_j 間彼此相關，不互相獨立。卡式檢定計算方法如方程式(18)所示，其中，O_{ij} 表示實際 t_i 與 t_j 一起出現的頻率，E_{ij} 則為期望出現的頻率。

$$\chi^2 = \sum_{i,j} \frac{\left(O_{ij} - E_{ij}\right)}{E_{ij}} \tag{18}$$

2.3.2 Likelihood Ratio

Likelihood Ratio [6], [10]對於兩個字詞 t_i 與 t_j，測試兩個假設：1) H_1: t_i 及 t_j 間彼此獨立，其出現與否不具相關性；2) H_2: t_i 與 t_j 的出現有相關性。H_1 與 H_2 的定義如方程式(19)(20)(21)所示。

$$H_1 : P\left(t_j \mid t_i\right) = p = P\left(t_j \mid \overline{t_i}\right) \tag{19}$$

$$H_2 : P\left(t_j \mid t_i\right) = p_1 \neq p_2 = P\left(t_j \mid \overline{t_i}\right) \tag{20}$$

$$\begin{aligned} p &= P\left(t_j \mid t_i\right) = P\left(t_j \mid \overline{t_i}\right) = P\left(t_j\right) \\ p_1 &= P\left(t_j \mid t_i\right) = \frac{P\left(t_i \cap t_j\right)}{P\left(t_i\right)} \\ p_2 &= P\left(t_j \mid \overline{t_i}\right) = \frac{P\left(\overline{t_i} \cap t_j\right)}{P\left(\overline{t_i}\right)} \end{aligned} \tag{21}$$

t_i 與 t_j 間的事件分佈關係，可表示為表 1。O_{11} 為 t_i 及 t_j 共同出現的次數；O_{12} 為 t_i 出現的次數，但不包含與 t_j 一同出現的次數；O_{21}：為 t_j 出現的次數，但不包含與 t_i 一同出現的次數；O_{22} 為不出現 t_i 也不出現 t_j 的次數。

表 1: t_i 與 t_j 事件分佈關係

	t_i	$\bar{t}_i$
t_j	O_{11}	O_{21}
$\bar{t}_j$	O_{12}	O_{22}

根據表 1，假設 t_i 與 t_j 出現的頻率為二項式方佈(Binomial Distribution)，方程式(24)為二項式分佈的計算方法，如此一來，便可以計算 H_1 與 H_2 的 Likelihood Ratio，如方程式(22)(23)所示：

$$L(H_1) = b(O_{11}; O_{11}+O_{12}, p)\, b(O_{21}; O_{21}+O_{22}, p) \tag{22}$$

$$L(H_2) = b(O_{11}; O_{11}+O_{12}, p_1)\, b(O_{21}; O_{21}+O_{22}, p_2) \tag{23}$$

$$b(k; n, x) = \binom{n}{k} x^k (1-x)^{(n-k)} \tag{24}$$

最後，依據 $L(H_1)$與 $L(H_2)$可以計算 Likelihood Ratio 值 λ，如方程式(25)所示。最後，再根據統計學中信賴水準 (Confidence Level)來估計關係的可信度；信賴水準通常以方程式(26)表示，α 為落在信賴區間 (Confidence Interval)外側的機率。舉例來說，當 α =0.005 時，表示信賴水準為 99.5%，再經查表得知門檻值為 7.88，當 λ 大於 7.88 時，表示 t_i 及 t_j 之間具有關係存在。

$$\lambda = \frac{L(H_1)}{L(H_2)} \tag{25}$$

$$(1-\alpha)\times 100\% \tag{26}$$

2.3.3 Mutual Information

Mutual information (MI) [1]亦是一種計算字詞語意相關度的方法，其計算方法為對於兩個字詞 t_i 及 t_j，當 t_i 及 t_j 共同出現的機率可以表示如方程式(27)所示。

$$MI\left(t_i,t_j\right)=\log_2\frac{P\left(t_i\cap t_j\right)}{P\left(t_i\right)P\left(t_j\right)}$$
$$=\frac{P\left(t_i\mid t_j\right)}{P\left(t_i\right)}$$
$$=\frac{P\left(t_j\mid t_i\right)}{P\left(t_j\right)} \tag{27}$$

Mutual Information 適合用於判別兩個字詞間的出現關係是否獨立，而不適用於判別兩個字詞是否相關[19]。表 2 進一步說明這個現象。表中，(*house, chambre*)的關係，若使用 x^2 來計算，則(*house, chambre*)的相關性遠大於(*house, communes*)。然而，已知 x^2 的結果是正確的。由此例可知，以 Mutual Information 評估兩個字詞的相關度，(*house, communes*)及(*house, chambre*)這兩組字詞相關性皆差不多，換句話說，Mutual Information 雖可計算字詞相關性，但更適合用於判別兩個字詞間的不相關性。

表 2: Mutual Information 範例

	chambre	*~chambre*	MI	χ^2
house	31950	12004	4.1	553610
~house	473	848330		
	communes	*~communes*		
house	4974	38980	4.2	88405
~house	441	852682		

2.4 視覺化之應用

視覺化的定義為將資料以圖形化的方式呈現，其目的在於讓使用者更容易了解資料的特性與整體的概念，舉例來說，可方便判別資料是否屬於同質性或是資料是否為雜訊等等。故在進行視覺化的過程前，需要將資料經由人工分析後再整合呈現，也才能達到視覺化的目的。下列幾項是進行視覺化的原則 [13]：

1. 將資料用某些視覺化的方式呈現

   一般而言，視覺化大多採用圖形介面來呈現，且避免使用不直覺或複雜的數學式及演算法。

2. 讓使用者容易了解資料所代表的意義及內容

在呈現資料的同時，資料數量的控制也是很重要的一環，尤其當資料量過多時，需要思考如何在質與量中取得平衡點以達到視覺化的目的。

3. 可直接與資料進行互動

透過互動的方式了解使用者對於資料上的需求，更進一步的可以做為未來進行系統改良的基礎。

在說明視覺化的原則及目的之後，根據這些原則有幾個常用的視覺化方法，在以下列舉並簡單說明之：

1. 二維平面

這是最常使用的視覺化表示方法之一，將物件向量化之後，再映射至二維平面上，表現物件群聚的狀態，同時以顏色區分不同的群，如圖 5 所示。

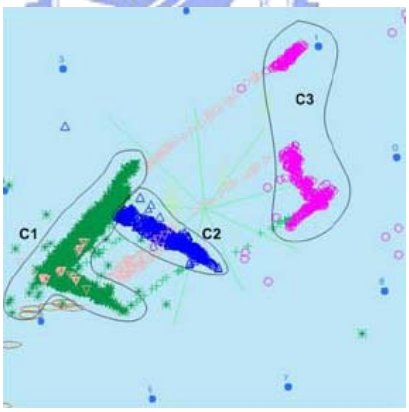

圖 5: 以二維平面表示群聚 [3]

2. 樹狀圖

常用於階層式分群法，根據決定分裂或合併的條件建立圖形，愈下層相似度愈高，如圖 6 所示。

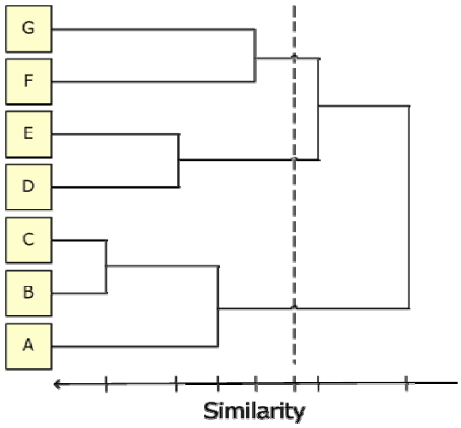

圖 6: 以樹狀圖表示群聚結果

第三章　概念萃取之文件分群與視覺化

在本章中，將闡述本研究提出的概念萃取之文件分群演算法，以及如何將分群結果加以視覺化。前置處理可以用來增進其後進行概念萃取的效益，故將先說明前置處理的方式；再詳細說明進行概念萃取、文件分群，以及視覺化的方法。系統整體架構如圖 7，主要分為四個程序，分別為資料前置處理(Data Pre-processing)、文件分群(Document Clustering)、群聚後置處理(Cluster Post-processing)，以及視覺化(Visualization)，本章會在每一節詳述每一個流程的步驟。

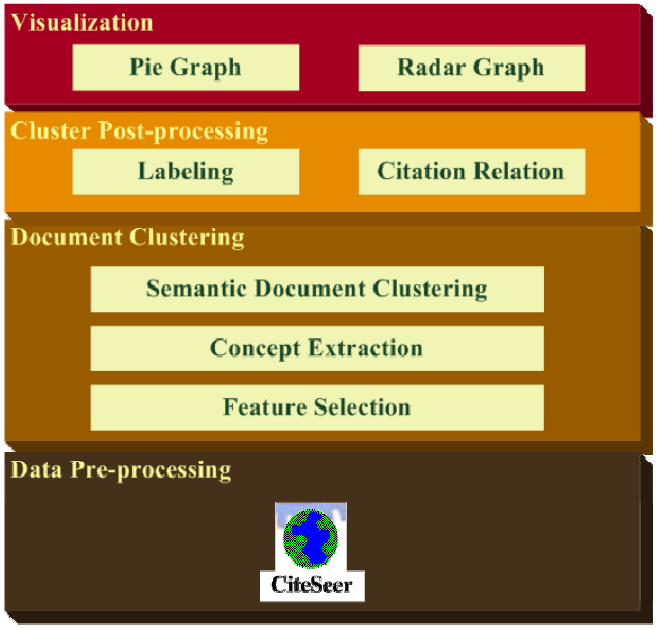

圖 7: 系統架構圖

3.1 前置處理

在說明前置處理方法之前，先說明本研究使用的語料庫(Corpus)。本研究是以 CiteSeer [29]內的論文做為語料庫，CiteSeer 是一個收集科學領域論文的數位圖書館，其收錄的主題相當多，包含計算機結構 (Architecture)、人工智慧 (Artificial Intelligence), 資訊擷取 (Information Retrieval)、計算機網路 (Networking)、作業系統 (Operating Systems)，以及計算機理論 (Theory)等，並且提供論文的相關欄位，包含標題(Title)、作者(Author)、摘要(Abstract)、出版社(Publisher)及引用 (Citation)等資訊。除了論文本

身的欄位之外，CiteSeer 亦建立了引用索引(Citation Index)，完整地記錄文章間的引用關係。引用的應用範圍很廣泛，如論文中參考書目，Blog 及網頁間的連結，甚至 Google 的 Page Rank [32]，都是透過引用來建立關係，因此在 3.3.2 節我們將會說明如何透過引用關係，建立文章間的關聯性。

本研究選取標題、摘要及引用做為資料來源，摘要部分所收錄的文字大約只有 1000 個字元，這個數量相當於以搜尋引擎透過關鍵字查找所得到的結果資料。在這種資訊不足的情況下，本論文將探討如何準確地萃取出合適的概念。

在 CiteSeer 的眾多主題中，本研究選定 Information Retrieval 相關論文做為語料庫，並針對文件標題與摘要進行前置處理，主要的前置處理包含斷詞切字(Tokenization)、小寫化(Lowercase)、刪除停用字(Stop Word)、詞性判斷(Part of Speech, POS)、詞幹轉換(Word Stemming)、片語化(Chunk) [27]，茲分別說明如下。

3.1.1 斷詞切字與小寫化

在前置處理中，斷詞切字是第一步驟。英文的斷詞切字主要是利用空格和標點符號來判斷段落與句子，以達到斷詞切字的目的。字彙的字首大小寫會影響到詞性的判斷，為了避免判斷錯誤，本研究將全部的字小寫化，再進行斷詞切字的處理。

3.1.2 停用字之處理

斷詞切字後，需將分隔出來的字詞進行第一次的過濾，此步驟便是移除停用字。停用字是指文章中沒有語意，但可用來平順語意的詞，可能包含介系詞、指代詞、連接詞與助詞等。由於本研究是以計算頻率做為判別字彙重要程度的依據，這些停用字經常出現於文件中，反而會被誤判為具有相當程度之重要性，因此需將在前置處理中將其過濾，以達到資料清除(Data Cleaning)的目的。表 3 為停用字列表的一部分。

表 3: 部分停用字列表

.	,	the	p	In	to	of
a	and	that	said	With	for	is
was	u	he	from	Have	he	not
had	by	it	they	Who	been	on
but	has	news	an	As	where	at
be	which	e1	states	About	him	you’re
whenever	wherever	whom	willing	Still	way	take

3.1.3 詞性標記(Part of Speech, POS)

一般而言，英文文件中的重要概念大多是由名詞或名詞片語所組成，故詞性標記也是篩選字詞的重要步驟之一。本研究中，詞性標記是以愛丁堡大學在 1990 年發展的 NLP Processor [30]來實作，NLP Processor 使用改良式 Penn Treebank Tag-Set [31]進行訓練，此工具之訓練語料集(Training Set)為網路上所搜集到的英文文件，大約包含一百萬字。以圖 8 的 CiteSeer 原文輸入 NLP Processor 之後，經由訓練語料集的比對找出每個字彙可能的詞性，並以 XML 標籤標註，結果如圖 9 所示。

we propose a hierarchical demand based replication strategy that optimally disseminates information from its producer to servers that are closer to its consumers

圖 8: CitesSeer 原文範例

詞性標示之方式如表 4 及表 5 所示。主要分成兩大部分：第一部分為開放類別分類(Open-Class)，此類別用以表示一般性詞彙，像是常用的名詞(Noun)、形容詞(Adjective)、動詞(Verb)，以及副詞(Adverb)等；第二部分為封閉類別(Closed-Class)，此類別的詞彙可以用少數有限的字來規定，或是藉由固定的組成方式來判斷。例如介系詞(Preposition)、冠詞(Article)，以及字首為 WH 的字彙。

```
<S>
<NG>
<W C='PRP'>we</W>
</NG>
<VG>
<W C='VBP'>propose</W>
</VG>
<NG>
<W C='DT'>a</W>
<W C='JJ'>hierarchical</W>
<W C='JJ'>demandbased</W>
<W C='NN'>replication</W>
<W C='NN'>strategy</W></NG>
<NG>
<W C='WDT'>that</W>
</NG>
<VG>
<W C='RB'>optimally</W>
<W C='VBZ'>disseminates</W>
</VG>
<NG>
<W C='NN'>information</W>
</NG>
<W C='IN'>from</W>
<NG><W C='PRP$'>its</W>
<W C='NN'>producer</W>
</NG>
<W C='TO'>to</W>
<NG>
<W C='NNS'>servers</W>
</NG>
<NG><W C='WDT'>that</W>
</NG>
<VG>
<W C='VBP'>are</W>
</VG>
<W C='JJR'>closer</W>
<W C='TO'>to</W>
<NG><W C='PRP$'>its</W>
<W C='NNS'>consumers</W>
</NG>
<W C='.' T='.'>.</W>
</S>
```

圖 9: NLP Processor 處理結果

表 4: 開放類別的種類

開放類別的種類 (open class categories)		
POS Tag	**Description**	**Example**
JJ	形容詞 (adjective)	Green
JJR	比較級形容詞 (adjective comparative)	greener
JJS	最高級形容詞 (adjective superlative)	greenest

RB	副詞 (adverb)	however, usually, naturally, here, good
RBR	比較級副詞 (adverb comparative)	Better
RBS	最高級副詞 (adverb superlative)	Best
NN	一般名詞 (common noun)	Table
NNS	複數名詞 (noun plural)	Tables
NNP	專有名詞 (proper noun)	John
NNPS	複數專有名詞 (plural proper noun)	vikings
VB	動詞 (verb base form)	Take
VBD	動詞過去式 (verb past)	Took
VBG	動名詞 (gerund)	Taking
VBN	過去分詞 (past participle)	Taken
VBP	非第三人稱動詞 (Verb, present, non-3d)	Take
VBZ	第三人稱動詞 (verb present, 3d person)	Takes
FW	外國字 (foreign word)	d'hoevre

表 5: 封閉類別的種類

封閉類別的種類 (closed class categories)		
POS Tag	**Description**	**Example**
CD	數字 (cardinal number)	1, third
CC	連接詞 (coordinating conjunction)	and
DT	指定詞 (determiner)	the
EX	there 存在詞 (existential there)	there is
IN	介系詞 (preposition)	in, of, like
LS	列表標題字 (list marker)	1)
MD	語氣詞 (modal)	could, will
PDT	前限定詞 (predeterminer)	both the boys
POS	所有格結尾 (possessive ending)	friend's
PRP	人稱代名詞 (personal pronoun)	i, he, it
PRP$	所有格代名詞 (possessive pronoun)	my, this
RP	質詞 (particle)	give up
TO	To (both "to go" and "to him")	to go, to him
UH	感嘆詞 (interjection)	uhhuhhuhh
WDT	WH 開頭限定詞 (wh-determiner)	which
WP	WH 開頭代名詞 (wh-pronoun)	who, what
WP$	WH 開頭所有格代名詞 (possessive wh-pronoun)	whose
WRB	WH 開頭副詞 (wh-adverb)	where, when

3.1.4 詞幹轉換

英文時常因為時態或是句型文法變化將字詞的形態改變，當進行資料擷取時，形態變化會導致無法準確地計算字詞出現的頻率，進而影響字詞相關度的計算結果。詞幹轉換即是刪去型態學(Morphology)上的詞類型態變化，用以解決上述的問題。本研究選擇由英國薩西格斯大學及劍橋大學合作發展的工具 Morpha [33]；它首先將每個字都標上詞性，再利用 Flex 規則 [15]進行詞幹轉換。

在此列舉幾個規則，如方程式(28)(29)所示：

$$\{\mathrm{A}\}+\{C\}\text{"ied"}\{\text{return}(\text{lemma}(3,\text{"y"},\text{"ed"}));\} \Rightarrow \text{carry}+\text{ed} \quad (28)$$

$$\text{"boogied"}\{\text{return}(\text{lemma}(3,\text{"y"},\text{"ed"}));\} \Rightarrow \text{boogie}+\text{ed} \quad (29)$$

左式是一般正規式表示法，右式則是程式輸入時的判斷方式；方程式(28)(29)都是描述以 ied 結尾的動詞變化，亦可從此例中得知，動詞變化通常具有多種形式，需挑選出最符合該字詞變化的規則將之轉換。

3.1.5 片語化

英文的片語由多個字彙組成，有時候單一字彙並不能表達出正確的語意，將其組合成片語之後便有了特殊的詞義，例如：next month、recent years 與 a critical point。經過觀察得知，論文的關鍵字通常以名詞片語表示，關鍵字則代表論文的主要概念。另外，未來建立字與字之間的關係時，必須統計字出現的頻率，若沒有經過片語化會把字面相同但是意義不同的單字計算在一起，這樣的統計效果就無法區分出語義的歧異 (Word Sense Ambiguity)，這是另一個必須片語化的理由。

本研究使用 NLP Processor 工具進行片語化，它是藉由上下文文法來判別斷句 (或斷詞)的位置，分隔出名詞片語及動詞片語。

3.2 文件分群演算法

經過前置處理將大部分文件內的雜訊或不必要的資訊刪除後，在這些經過資料清除(Data Cleaning)後的字中，挑選出較能代表文件概念的字彙及片語，計算這些字彙與片語之間的相關度，並據以建立網路語意關係圖，再依據圖形理論(Graph Theory)原理進行過濾及分群，以達到概念萃取的目的，也做為本研究文件分群的基礎。

3.2.1 特徵選擇

本小節中，將深入探討如何透過特徵選擇(Feature Selection)挑選較具代表性的概念字詞。除了傳統上依字詞出現在文件或語料庫內的次數進行過濾之外，也會利用字詞間的相關度來進行特徵選擇。

1. 特徵過濾

在以文件做為資訊來源時，通常是以一個有意義的「詞」做為特徵，然而，特徵過多一直是文件分群面臨的困難之一，過多的特徵將會造成分群演算法耗費時間在處理不具代表性或甚至是無意義的特徵，同時也可能降低重要特徵的顯著性。解決此類問題的方法稱之為維度縮減(Dimension Reduction) [23]，運用維度縮減的技術不但可以簡化文件自動分群的計算過程，同時也更能正確地表達文件的概念。

本研究依據下面幾項規則對特徵進行過濾及篩選：

a) 移除單一字元

單一字元的字彙通常不具有意義。

b) 移除副詞及代名詞

本研究主要為萃取短文件的概念，而文件概念常以名詞和名詞片語來表示之。代名詞雖然常常表示主詞，但目前用來替換代名詞為主詞的工具準確度只達50%，對於短文的助益不大；另一方面，為了減少處理的複雜度，不使用代名詞替換而直接移除副詞與代名詞。

c) 移除在語料庫出現次數過多的字詞

當一個字詞在語料庫內經常出現時，幾乎可以確定此字詞屬於過於常見且不具有代表性之字詞。本研究將出現次數定義為在不同文章出現該字詞的文章篇數，且出現次數的上限為語料庫內文件個數的 8%。

d) 移除在語料庫出現次數過少的字詞

若一個字詞出現的次數太少，則此字詞幾乎可以確定不適合用以表達文件的概念，本研究訂定出現次數的下限為 3 次。

2. 計算特徵間之語意相關度

兩個字詞之間的語意相關度是以「共現」(Co-occurrence)的方式來計算。在計算共現時，首先需要訂出一個範圍，本研究以句子為範圍，即當兩個字詞在同一句內出現才表示其具有語意相關度。

計算相關度是採用由 Likelihood Ratio 衍生的統計方法：Log Likelihood Ratio [6][10]。其優點為經由數學函數的轉換後，即可產生一個易於計算的統計函數分佈，其計算方法為將 Likelihood Ratio 值 λ 先取對數值後，再乘上一常數 -2。計算 $-2\log\lambda$ 的過程如(30)所示，本研究使用最後得出的化簡後公式來進行語意相關度之計算。

$$\begin{aligned}
&-2\log\lambda \\
&= -2\log\frac{L(H_1)}{L(H_2)} \\
&= -2\log\frac{b(O_{11};O_{11}+O_{12},p)\,b(O_{21};O_{21}+O_{22},p)}{b(O_{11};O_{11}+O_{12},p_1)\,b(O_{21};O_{21}+O_{22},p_2)} \\
&= -2\left(\begin{array}{l}(O_{11}+O_{21})\log p+(O_{12}+O_{22})\log(1-p)- \\ \left(O_{11}\log p_1+O_{12}\log(1-p_1)+O_{21}\log p_2+O_{22}\log(1-p_2)\right)\end{array}\right)
\end{aligned} \tag{30}$$

由於本研究採用論文為語料庫，根據論文的特性，標題(Title)通常是表達文件的主題概念，摘要(Abstract)則是簡略地說明文件的主題和相關背景等等。故對於出現於標題中之特徵往往具有較重要的特徵關係，在計算特徵間語意相關度時，透過增加其權重來強化這些特徵關係之代表性，在本研究中取權重為 2.0，故 t_i 及 t_j 的語意相關度 r_{ij} 表示如(31)所示。

$$\begin{aligned}
&r_{ij} = -2\log\lambda\times weight \\
&weight = \begin{cases}2.0\text{ , if } t_i \text{ \& } t_j \text{ are both in title} \\ 1.0\text{, otherwise}\end{cases}
\end{aligned} \tag{31}$$

計算 r_{ij} 之後，以信賴水準方式估計，本研究取 α=0.001，僅保留信賴度大 99.9%的特徵關係，意即當值大於 10.83 時，才保留這個特徵關係，表 6 為列舉一些特徵關係以供參考。

表 6: 特徵關係實例

t_i	t_j	associate weight
self-organizing map	map	149.719
Wapper	induction	149.719
mobile agent	mobile	140.27
Question	answer	138.388
Latent	semantic	136.149
Label	unlabeled	134.316
computer science	science	131.792
Precision	recall	106.873
Smooth	bandwidth	93.8361
Information extraction	learning	93.1027

3.2.2 概念萃取與特徵分群

在 3.2.1 節介紹了如何從語料庫中挑選出合適且具代表性的特徵，並且計算了特徵之間的關係，接下來將介紹如何利用這些特徵之間的關係進行概念萃取。概念萃取的方法為將 2.1.4 節提到的關鍵字分群 Topic Keyword Clustering 演算法 [2]加以修改，以符合本研究的需要。圖 10 為概念萃取的步驟，本小節將會依步驟舉例並詳細說明之。

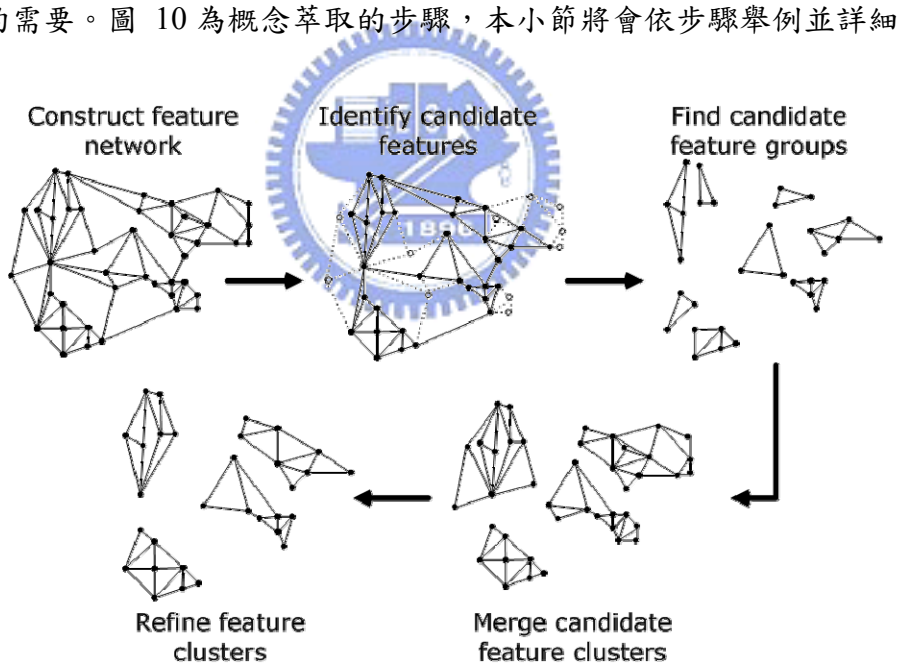

圖 10: 概念萃取之步驟

1. 建立特徵語意網路圖

在前一小節得到特徵間語意相關度後，根據圖形理論之原理，每項特徵都可表示為一個點(Vertex)，特徵間的關係以一個邊(Edge)做為表示，由這些點與邊即可組合成一個

基本的網路圖，並依照下列的步驟對此網路圖進行處理：

a) 移除網路圖內不重要的邊

對網路圖內的邊進行刪減，取門檻值為網路圖內所有語意關係度的平均，刪除小於門檻值的連線，將原本的網路圖修正為稀疏網路圖，如圖 11 所示，未來將以此圖做為本研究說明概念萃取步驟的範例。圖中的每個點是與 Information Retrieval 相關的關鍵字詞，邊則是大於語意關係度平均值的特徵關係，但邊的長度與字詞之間的相關度無關。

圖 11: 稀疏網路圖

b) 移除稀疏網路圖內不重要的點

移除較不重要的邊得到稀疏網路圖之後，便可計算每個點於稀疏網路圖之權重，計算方法如(32)所示，r_{ij} 為點 V_i 及點 V_j 的字詞相關度，V_j 為與點 V_i 相鄰的點，m 為與點 V_i 相鄰的頂點個數，因此點權重即為該點相連之邊的平均權重。接續前一步驟移除不重要的邊之後，繼續移除稀疏網路圖中不重要的點，將門檻值訂為所有點權重的平均，移除小於平均的點，如圖 12 所示，虛線空心的點表示不重要的點，將 codec，story，news，google，yahoo 移除。

$$CW_i = \frac{\sum_{j=1}^{m} r_{ij}}{m} \tag{32}$$

圖 12: 選出重要的關鍵字

2. 概念萃取之方法

這一節將說明概念萃取之方法及步驟。在前一步驟所建立的候選特徵之語意網路圖，圖內的點即代表語料庫內重要的特徵，將這些特徵利用演算法進行分群運算，所獲得的每一個群即代表萃取出的一種概念，這些步驟在本研究中稱之為概念萃取，將重要的步驟詳述如下：

a) *k*-Nearest Neighbor Graph Approach [5]

考慮圖中的每個點，取與該點最相近的 k 個點為一組，每組都為一個連通圖(Connected Graph)，本研究稱之為特徵候選組。k 值的選擇會影響分群數目的多寡，當 k 值愈大，群數便會愈少，而每一群包含的特徵也會較多，但當群內的特徵數量過多時，反而無法表達出清楚的概念，降低重要特徵的代表性。k 值不能過大，亦不能 k 值過小，否則會產生過多的群；根據本研究的實驗結果，當 k 值取 2 時效果較佳，故在本研究中取 k 為 2。圖 13 為範例進行 *k*-Nearest Neighbor Graph Approach 後的結果，共分為四個特徵候選組。

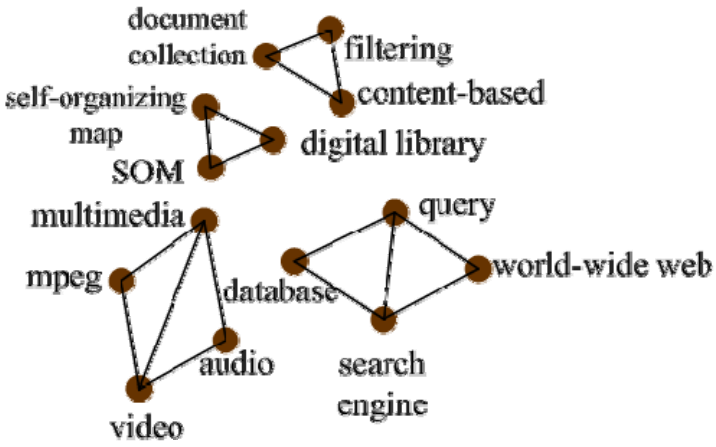

圖 13: *k*-Nearest Neighbor Graph Approach 分群結果

b) 由特徵候選組產生候選特徵子群

以每個候選特徵組為中心，向外還原先前與候選特徵組內的點有直接連線關係的邊，形成候選特徵子群，並計算每個子群的權重，權重計算方式為該群內所有邊權重的總和，如(33)所示，G_m 表示某一特徵候選組 m，r_{ij} 則是 G_m 內包含的特徵關係。

$$W_{G_m} = \sum_{r_{ij} \in G_m} r_{ij} \tag{33}$$

c) 合併候選特徵子群

產生候選特徵子群之後，使用 Greedy 演算法找出互連性(Inter-Connected)最強的候選特徵兩個子群，並將之合併。互連性強度的判別是依據兩個候選特徵子群的互連相關度 (Relative Inter-Connective)來計算。計算方式為兩個子群內交集的邊之權重總和再除上兩個子群的權重總和，如方程式(34)。由上述方程式也可以了解，當兩個子群共同包含的邊達到一定比例時，即表示該二子群有某種程度的相關性，因此可以將他們合併。以圖 14 為例，候選特徵組{digital library, self-organizing map, SOM} 及 {document collection, filtering, content-based}還原連線後，且形成的候選特徵子群之間的互連相關度夠大，故將之合併成一個特徵子群。經由實驗得知，取門檻值 0.5 時會有最佳的效果。

$$RI(G_i, G_j) = \frac{\left|W_{E(G_i,G_j)}\right|}{\left|W_{G_i}\right| + \left|W_{Gj}\right|} \tag{34}$$

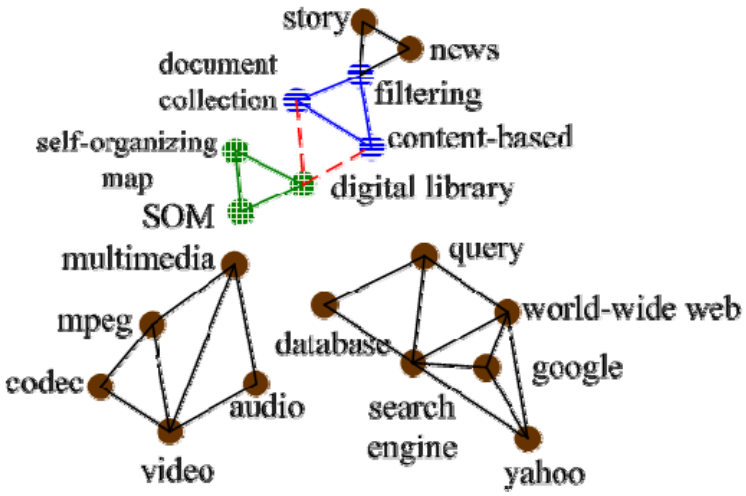

圖 14: 合併特徵子群

d) 修正並產生概念子群

合併候選特徵子群之後，子群內特徵的個數會產生差距，此現象會造成在進行文件分群時，包含特徵較多的子群會涵蓋較多文章，此種情形將會影響分群的正確性。因此本研究希望每個子群內的特徵個數保持在一定的差距內，同時保留原來特徵子群的概念。

利用每個子群之邊權重平均值及連線密度(Connected Density)來計算子群的權重。邊權重的計算方式如(35)。連線密度為該子群內的邊個數除以該子群可能最大邊數，如(36)所示。當子群的權重小於平均權重或是子群內的特徵個數大於平均個數時，則移除與該群最不相關 (意即點權重最小)的特徵，子群權重的計算方法如(37)所示。

$$AS(G) = \frac{\sum_{r_{ij} \in E(G)} r_{ij}}{\left|E(G)\right|} \tag{35}$$

$$CD(G)=\frac{(E(G))}{|V(G)|\times|V(G)-1|/2} \tag{36}$$

$$CW=CD(G)\times AS(G) \tag{37}$$

以圖 15 為例，{multimedia, mpeg, audio, video, codec}包含了五個特徵，{SOM, digital library, content-based, filtering, document collection, self-organizing map, story, news}一共包含了八個特徵，{search engine, database, query, world-wide web, google, yahoo}則包含了六個特徵，每個群的平均個數則為(8+6+5)/3=6.33，表示每個子群最多只能包含六個特徵。由上述可知，需將包含特徵 SOM 的子群進行修正，將最不重要的特徵 story 及 news 移除，以達到平衡子群內特徵個數的目的。另外，若是子群內包含的特徵少，但是子群權重卻大於平均權重時，表示此群內包含重要的特徵關係，故將該群保留；反之，若子群經修正後仍小於平均權重，則表示該群內的沒有重要的特徵關係，故將該群直接刪除。由於分群結果中的每一個子群即代表一種概念，故本研究將特徵的分群結果稱為概念子群。

圖 15: 修正特徵子群

3.2.3 語意相似度向量之文件分群

在 3.2.2 萃取出文件中的重要概念之後，將文件內容以這些概念描述，進而將文件分群，最後透過群聚標記的動作將分群結果視覺化。

1. 產生語意相似度向量

首先將文件 D_j 及概念子群 C_m 向量化，根據 3.2.2 產生 k 個概念子群，一共包含了 n 個特徵，將文件 D_j 及概念子群 C_m 以這 n 個特徵表示。首先說明文件向量化的方式。當特徵 i 出現於該文件時，向量值為特徵 i 在該文件中的 TF-IDF 值，以 w_i 表示，若特徵 i 未出現於文件中，則向量值 w_i 為 0；概念子群的向量化方式為，當特徵 i 出現於概念子群 m 時，向量值為 1，否則為 0，如(38)所示。使用 Cosine Similarity 計算文件向量與每一個概念子群向量的相似度，相似度即代表文件包含該概念的程度。每個文件將會產生一個可描述多個概念的語意相似度向量，如(39)所示。

$$
\begin{aligned}
&C_m = \langle f_1, f_2, ..., f_n \rangle \text{ where } m = 1, 2, ..., k; \\
&D_j = \langle w_1, w_2, ..., w_n \rangle \text{ where } j = 1, 2, ..., N \\
&n: \text{ \# of features; } k: \text{\# of clusters} \\
&f_n = \begin{cases} 0, \text{ if } f_n \notin C_m \\ 1, \text{ if } f_n \in C_m \end{cases} \\
&N: \text{ \# of documents in corpus} \\
&w_i: \text{the weights of the features } i \text{ are estimated as TF-IDF}
\end{aligned}
\tag{38}
$$

$$
\begin{aligned}
&SD_j = \langle sim(D_j, C_1), sim(D_j, C_2) ..., sim(D_j, C_k) \rangle \\
&\text{where } j = 1, 2, ..., nd; \; i = 1, 2, ..., k
\end{aligned}
\tag{39}
$$

2. 進行文件分群

文件的分群動作是利用以劃分方法為基礎的 Bisection k-Means 演算法 [24]，首先將所有文件視為一群，該演算法步驟如下：

Bisection k-Means 演算法：

輸入：n 個物件

輸出：k 個群

1 選擇一個欲分割的群

2 將該群以 2-Means 演算法分為兩個子群

3 以 2.2 節提到之 I_2 準則做為分群最佳化的條件，計算文件與所有群中心點向量的相似度，再對應文件至與其相似度最高的群。

4　重覆進行步驟 1 及 2，直到達到目標分群個數

4.1　若預計分為 *k* 群，則需要執行 *k*-1 次

4.2　選擇要分割哪一群則是找出當時群內文件間之平均相似度最小的群，換句話說，即是找出文件相似度總合最小的群。

Bisection *k*-Means 結束後，文件分群大致完成。為了將分群結果以視覺化呈現，必須控制分群的數量，同時也要維持群內部的相似度與群外部的分離度。

3.3 群之後置處理

在本節中將會敘述如何對前一節所得到的分群結果進行後置處理。後置處理主要分為兩個方向，第一是群聚標記(Labeling)，挑選出合適的特徵代表該群的主題。第二則是以論文內的引用文章(Citation)之相似度建立群與群之間的關係。

3.3.1 群聚標記

文件分群視覺化通常採取標記的方式，而標記之主要目的是為讓使用者能夠快速且容易了解每一個群所代表的概念及意義。本研究所採取的標記方法是依據群中心點的語意相似度向量來選擇相似度最高的概念子群，並取概念子群內的特徵為候選標記特徵。在概念子群的特徵當中，由觀察得知名詞及名詞片語較能表達群的概念，且名詞片語的重要性又大於名詞。經過多次實驗及分析，本研究依照下列幾項規則來挑選特徵：

1. 選出權重前十名的特徵
2. 在這十項特徵中，優先選取名詞及名詞片語為候選標記特徵
3. 利用人力輔助過濾出有意義的特徵
4. 取權重最高的二項特徵做為最後群的標記

3.3.2 以論文之引用文章建立群聚關係

除了將分群結果視覺化之外，還有什麼資訊可以協助使用者了解分群的意義及概念，並且可以透過視覺化以供參考？由於本研究是以 CiteSeer 做為語料庫，而 CiteSeer 的特色在於其建立了完整的引用索引(Citation Index)，利用此項特性，本研究先透過文

件內引用文章(Citation Article)的相似度，建立文件間的引用關係(Citation Relation)，進而建立群與群之間的引用關係，並將結果用於輔助視覺化。在本研究中，文件間的引用文章的相似度是依據兩方面來定義的，分別為：

1. 引用文章之超連結 (Hyperlink)

在 CiteSeer 資料庫中，文件內的引用文章絕大部分都有超連結可以連結到該篇引用文章，將此超連結視為一文件之代碼，可藉此得知經常被引用的文章為何，又哪些文件引用相同文章。

兩篇文章的引用相似度計算方法如(40)所示。對兩篇文件 d_i 及 d_j，令 $link(d_i)$ 與 $link(d_j)$ 為 d_i 與 d_j 所引用之文章，計算 d_i 及 d_j 共同的引用文章數，再除上 $link(d_i)$ 及 $link(d_j)$ 之最小值。

$$linksim_{i,j} = \frac{\left|link(d_i) \cap link(d_j)\right|}{\min\left(\left|link(d_i)\right|, \left|link(d_j)\right|\right)} \tag{40}$$

2. 引用文章之標題

雖然 CiteSeer 在文章引用方面的資料很完備，但若是該篇引用文章之超連結遺失，或是同一篇被引用文章具有兩個以上的超連結，便會影響相似度的計算，因此除了考慮引用文章的超連結之外，還需要考慮引用文章的標題來計算相似度，以修正此種資料錯誤的問題。對於每篇文件，擷取其引用文件的所有標題，並進行前置處理與向量化之後，使用 Cosine Similarity 進行相似度的計算。

本研究將上述兩項的相似度以線性組合調整，以計算文件 d_i 及 d_j 的引用相似度 $DR(i,j)$，$linksim$ 表示引用文件之超連結相似度，$textsim$ 則為用文件之標題相似度，計算方法如(41)所示，本研究取 α 為 0.5。

$$DR(i,j) = \alpha \cdot \left(linksim_{i,j}\right) + (1-\alpha) \cdot \left(textsim_{i,j}\right) \tag{41}$$

群與群的引用關係則是建立在文件間的引用相似度，以下舉例說明。若群 C_m 中有

文件 d_i(即 $d_i \in C_m$)，群 C_n 有文件 d_j(即 $dj \in C_n$)，若 d_i 及 d_j 的引用相似度大於門檻值 θ，則將 C_m 及 C_n 的引用次數加 1，直到群與群之間的引用關係都建立完成；再依公式(42)計算 C_m 及 C_n 之相似度。最後進行視覺化時，將 *CR(m,n)* 由大到小排列，列出前十個最相關的類別以供使用者參考。

Define a threshold θ

$$CR(m,n) = \frac{\#\text{ of document pairs } (d_i, d_j) \text{ that } DR(i,j) > \theta}{\max(|C_m|, |C_n|)} \tag{42}$$

3.4 視覺化過程

本研究以二種視覺化的方式來呈現分群結果。第一種為文件的分群結果，本系統將每一群以一個圓餅圖表示，其優點在於可以用半徑大小來表示該群內包含的文件數量，並且用顏色的深淺表示該群內文件的同質性是否一致，顏色愈深代表同質性愈高，群內部緊密，以圖 16 為例，每一群上頭的文字表示該群的標記，亦可稱之為群的主題概念；同質性則依相似度分為四種層級，依相似度高低排序，分別為 collaborative filtering > speech recognition > self-organizing map > decision tree。

第二種視覺化方式是用來表達群與群之間的關係，本系統以雷達圖表現之，雷達圖的特性在於可以明確地描述群與群之間的關係強度，以圖 17 為例，最上層的文字為這一群的主題概念，雷達圖每一個頂點即代表一個與該群具有相關度的群，透過雷達圖可輕易地了解該群與其他群之間的相關程度，在此例中 text categorization 與 self-organizing map 的相關度最高。

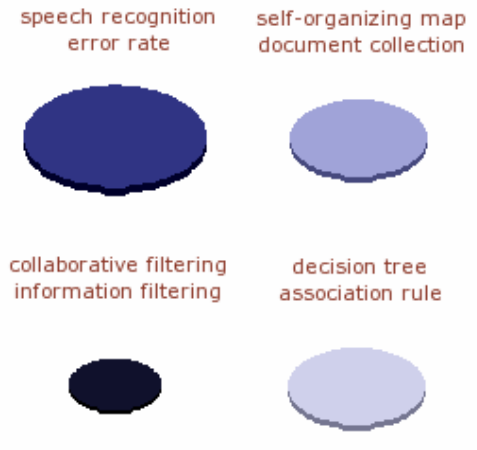

圖 16: 分群結果之視覺化

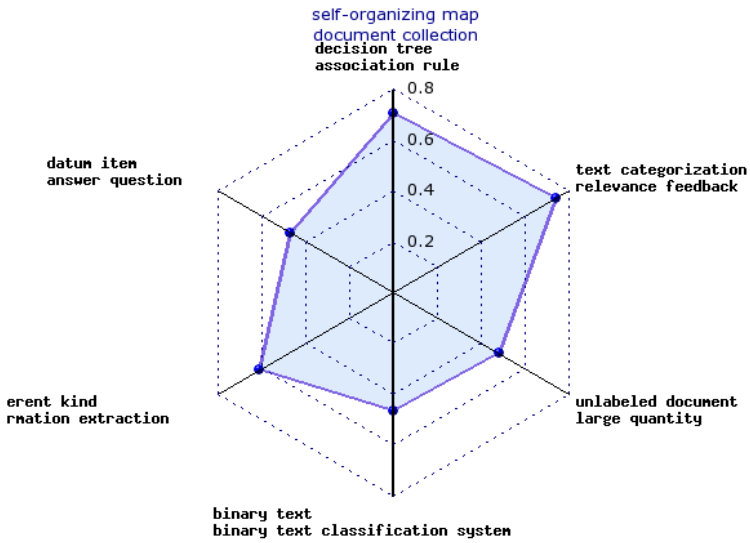

圖 17: 群與群關係之視覺化

第四章　實驗結果分析與評估

本章將敘述評估方法與實驗結果，並根據實驗的結果進行討論。評估分群的結果有很多種，但是要從中找出適合的方法卻很困難，分群方法的效能會根據評估方法的不同而改變，因此需以多種方式來考量。本研究共使用三種評估方法，首先是由專家將語料庫內的論文分群，根據專家的分群結果計算 Entropy、Purity、Recall 及 F-measure。第二種是計算群的內聚力及分離度以判別分群結果的優劣。第三種則是請專家標示兩兩文件之間的相似度。此外，將本研究提出的分群方法及 2.1.4 中提到的 Topic Keyword Clustering 進行比較與分析。

4.1 評估方法

4.1.1 以專家分群結果評估

由於本研究使用的語料庫沒有分類的架構，故先請專家將語料庫內的文件分群，本研究是採用與 Information Retrieval 相關的論文做為資料，因此根據 Information Retrieval 領域內特有的專有名詞來定義分類架構，在本實驗中，專家將之分為三十五群，包含的主題及文件篇數如表 7 所示，共 541 篇文件。

表 7 專家標示的結果

Class Number	Class Label	Number of documents
1	Search engine	12
2	Data mining	20
3	Speech recognition	16
4	SVM	12
5	Machine learning	12
6	Labeling	9
7	Information filtering	35
8	Agent	18
9	Computational complexity	22
10	Natural language processing	40
11	Text analysis	18
12	Programming language	7
13	Text categorization	28
14	Document clustering	12
15	Information extraction	30
16	Tracking	9
17	Digital library	16
18	Biological	4
19	Information retrieval system	7
20	Self-organizing map	9
21	Multimedia retrieval	8
22	Image encryption algorithm	3

23	Markov model	9
24	Personalized web	15
25	Supervised learning	19
26	Query expansion	20
27	Naïve bayes	18
28	Part-of-speech tagging	11
29	Wapper induction	16
30	Content-based retrieval	25
31	Middleware／midation	10
32	Broadcast disks	10
33	File system	11
34	Network	17
35	Visualization	13

在說明評估方法之前，先介紹查準率 (Precision)及查全率 (Recall) [18]，查準率及查全率的公式如方程式(43)(44)，查準率與查全率原來是用在評估資訊檢索系統的效能。當查準率與查全率用於評估分群效能時，查準率代表分群的結果為正確的比率，而查全率的意義為根據分群的結果，某一類別內被查找出來的比率，意即被正確歸類的比率。

$$\text{precision}=\frac{\#\text{ of relevant retrieved}}{\#\text{ of retrieved}} \tag{43}$$

$$\text{recall}=\frac{\#\text{ of relevant retrieved}}{\#\text{ of relevant}} \tag{44}$$

接下來則說明評估分群的方法 [21]，本研究使用分類中常使用的 Purity、Recall、Entropy 、F-measure，這四個數值皆是由上述的查準率及查全率衍生而來的，詳細說明如下：

1. Purity

分別對於每一個群 i，都可以計算它與每一個類別 j 的查準率 (p_{ij})，p_{ij} 的意義為在群 i 中，標示為類別 j 的比率。計算方法就如方程式(45)所示，n_{ij} 為群 i 包含之類別 j 的文件篇數，n_i 為群 i 的文件篇數。群 i 的 purity 值 ρ_i 為查準率最高的類別之 p_{ij}，如方程式(46)所示。再將每一群包含的文件篇數佔文件總數的比例當做權重，計算整體分群結果的 Purity 值 ρ ，如方程式(47)所示，其中 n 表示文件總數。

$$Pij = \frac{n_{ij}}{n_i} \tag{45}$$

$$\rho_i = \max\left\{p_{ij}\right\} \tag{46}$$

$$\rho = \sum_i \frac{n_i}{n}\rho_i \tag{47}$$

2. Recall

Recall 的計算方法和 Purity 大致相同，分別由對於每一個群 i，都可以計算它與每一個類別 j 的查全率 (r_{ij})， r_{ij} 的意義為是在類別 j 中，被標示為群 i 的比率。計算方法就如方程式(48)所示，n_{ij} 為群 i 所包含類別 j 的文件篇數，n_j 為屬於類別 j 的文件篇數。類別 j 的 recall 值 γ_j 為查全率最高的群之 r_{ij}，再根據 γ_j 計算整體分群結果的 Recall 值 γ ，如方程式(49)(50)。

$$r_{ij} = \frac{n_{ij}}{n_j} \tag{48}$$

$$r_j = \max\left\{r_{ij}\right\} \tag{49}$$

$$\gamma = \sum_j \frac{n_j}{n}\gamma_j \tag{50}$$

3. Entropy

根據每一群的查準率，亦可以延伸出另一種常用的計算方法 — Entropy。Entropy (熵)，或稱亂度，當分群結果愈好數值愈小。對於每個群 i，可以用方程式(51)計算其 Entropy 值 E_i，再根據 E_i 計算整體分群結果的 Entropy，如方程式(52)所示。

$$E_i = -\sum_j p_{ij}\log p_{ij} \tag{51}$$

$$E = \sum_i \frac{n_i}{n}E_i \tag{52}$$

4. F-measure

由於 Entropy 只有參考查準率，其結果往往不夠公正，因此最常用的評估方式就是結合查準率與查全率的 F-measure。對群 i 及類別 j 的 F-measure，計算方法如方程式(53)所示。在專家標示的分群結果中，每一個類別的 F-measure 值 F_j，是取所有群與該類別 j 的 Fij 之最大值，如方程式(54)所示。和先前評估整體分群的方式相同，將類別包含的文件篇數佔文件總數的比例當做權重，計算整體分群的 F-measure，如方程式(55)。

$$F_{ij} = \frac{2 \cdot r_{ij} \cdot p_{ij}}{p_{ij} + r_{ij}} \tag{53}$$

$$F_j = \max_i \left\{ F_{ij} \right\} \tag{54}$$

$$F = \sum_j \frac{n_j}{n} F_j \tag{55}$$

4.1.2 以群聚分佈評估

分群的基本概念即為透過分群演算法，在非監督式的資料中，將相似的資料群聚在一起，並且預期群聚分佈為群內相似度高，群與群之間分離度高。一般評估分群結果的優劣是以群的內聚力 (Cluster Compactness)、分離度 (Cluster Separation)以及綜合前兩者的 Overall Cluster Quality。在此詳細說明這三種評估方式的計算方法 [9]：

1. 內聚力

欲計算內聚力，首先要計算所有文件向量 X 的變異數 (Variance) $v(X)$，以及計算群 c_i 的變異數 $v(c_i)$，計算 $v(X)$ 的公式如方程式(56)所示。N 為文件總數，$\bar{x}$ 則表示所有文件向量的平均，如方程式(57)；$d(x_i,x_j)$ 表示向量 x_i 與 x_j 的距離，距離定義如方程式(58)。$v(c_i)$ 的計算方式和 $v(X)$ 大致相同，以群為單位，計算群內文件與群中心點之間的距離，如方程式(59)所示。

$$v(X) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} d^2 \left(x_i, \bar{x} \right)} \tag{56}$$

$$\bar{x} = \frac{1}{N}\sum_i x_i \tag{57}$$

$$d\left(x_i, x_j\right) = 1 - \cos\left(x_i, x_j\right) \tag{58}$$

$$v\left(c_i\right) = \sqrt{\frac{1}{|c_i|}\sum_{j=1}^{|c_i|} d^2\left(c_{ij}, \overline{c_i}\right)} \tag{59}$$

接下來將每一群 (c_1,c_2,…c_c)的變異數值除以 $v(X)$後取平均值，計算公式如(60)所示，其中 C 為群聚個數，得一值 Cmp，即為分群結果的內聚力，Cmp 值域介於 0~1 之間，當 Cmp 的值愈小，表示每一群的內聚力愈強。

$$Cmp = \frac{1}{C}\sum_i^C \frac{v\left(c_i\right)}{v\left(X\right)} \tag{60}$$

2. 分離度

分離度的計算方法如(61)所示，得到值 Sep。其中 σ 為 Gaussian Constant，C 為群聚個數，$d(x_{ci},x_{cj})$為群 c_i 與群 c_j 中心點的距離，Sep 值域介於 0 與 1 之間。當 Sep 的值愈小，表示每一群的分離度愈高。

$$Sep = \frac{1}{C\left(C-1\right)}\sum_{i=1}^{C}\sum_{j=1, j\neq i}^{C} \exp\left(-\frac{d^2\left(x_{c_i}, x_{c_j}\right)}{2\sigma^2}\right) \tag{61}$$

3. Overall Cluster Quality

將上述的 Cmp 及 Sep 利用線性組合得到綜合分數，稱之為 Overall Cluster Quality，用以評估分群的品質。公式如(62)所示，β 為常數，且 $\beta \in [0,1]$，用以調整 Cmp 及 Sep 的比重。當 Ocq 值愈低表示該群聚分佈效果愈好。

$$Ocq\left(\beta\right) = \beta \cdot Cmp + \left(1-\beta\right) \cdot Sep \tag{62}$$

4.1.3 以專家標示兩兩文章相似度評估

在進行人工判別實驗結果的時候，我們必須評估專家對樣本相似度的同意度，通常利用可信度 (Reliability)及有效性 (Validity)來區別。可信度是指專家在評估過程中標示的一致性，而有效性是指專家評估的樣本中可用的樣本數。本研究採用 Kappa Statistics[34]來計算專家的同意度值，以表 8 為例說明之。假設有 29 位病人分別由兩位醫生診斷病情，Yes 表示診斷結果為不健康，No 則表示健康，則

Kappa= (Observed agreement – Chance agreement) / (1 – Chance agreement)
Observed agreement = (10+12) / 29 = 0.76
Chance agreement = 0.586 * 0.345 + 0.655 * 0.414 = 0.474
Kappa = (0.76 – 0.474) / (1-0.474) = 0.54

表 8 Kappa Statistics 範例 [7]

		Doctor A		
		No	**Yes**	**Total**
Doctor B	**No**	10 (34.5%)	7 (24.1%)	17 (58.6%)
	Yes	0 (0.0%)	12 (41.4%)	12 (41.4%)
Total		10 (34.5%)	19 (65.5%)	29

並計算標準差得到 0.134，當信賴水準(Confidence Level)達 95%時，信賴區間 (Confidence Interval)為 (0.279, 0.805)，將結果由 Kappa 的參考表格，如表 9 所示，此範例的同意度介於 Fair 及 Almost Perfect 之間。

表 9: Kappa Table

Kappa	**Strength of agreement**
0.00	Poor
0.01~0.20	Slight
0.21~0.40	Fair
0.41~0.60	Moderate
0.61~0.80	Substantial
0.81~1.00	Almost perfect

在本研究中，將語料庫內的文章隨機抽選二百組配對組合，請專家標示該組內的兩篇文章是否相似，結果如表 10 所示。

表 10: 專家標示答案分佈表

		Expert A		
		No	**Yes**	**Total**
Expert B	**No**	97 (90.65%)	10 (9.35%)	107 (53.5%)
	Yes	4 (4.3%)	89 (95.7%)	93 (46.5%)
Total		101 (51.5%)	99 (49.5%)	200

Observed agreement = (97+89) / 200 = 0.93
Chance agreement = 0.535 × 0.515 + 0.465 × 0.495 = 0.5057
Kappa = (0.93 -0.5057 / (1-0.5057) = 0.8584

並計算標準差得到 0.036，當信賴水準達 95%時，信賴區間為 (0.7893, 0.9305)，將結果由表 9 分析，同意度介於 Substantial 及 Almost Perfect 之間，亦即專家 A 及專家 B 評估的答案具有相當高的一致性。在這二百組之中，專家 A 與專家 B 一致性的答案有 186 組，本研究即以這 186 組文章配對做為實驗的樣本。

若某樣本被專家標示為「相似」，則表示專家認為兩篇文章應該屬於同一群，同理，若被標示為不相似，則表示應要屬於不同群，並分別計算正確率，計算方法如方程式(63)所示[8]，事件分佈表如表 11 所示。

表 11: 以專家標示兩兩文章相似度之事件分佈表

		專家標示	
		Y	N
分群結果	Y	t_pos	f_pos
	N	f_neg	t_neg

t_pos：專家標示相似，且分群結果亦屬於同一群之樣本個數
f_pos：專家標示不相似，但分群結果為屬於同一群之樣本個數
t_neg：專家標示不相似，且分群結果亦不屬於同一群之樣本個數
f_neg：專家標示相似，但分群結果為不同群之樣本個數
pos：專家標示為相似的樣本個數，即正例的個數。
neg：專家標示為不相似的樣本個數，即反例的個數。
sensitivity：正例的正確率，即分群結果為同群且專家亦標示為相似之比率
specificity：反列的正確率，即分群結果為不同群且專家亦標示為不相似之比率
accuracy：綜合 *sensitivity* 及 *specificity* 而得的正確率

$$sensitivity = \frac{t\_pos}{pos}$$

$$specificity = \frac{t\_neg}{neg} \tag{63}$$

$$accuracy = sensitivity\frac{pos}{(pos+neg)} + specificity\frac{neg}{(pos+neg)}$$

4.2 實驗結果

4.2.1 以專家分群結果評估

根據 4.1 的評估方法，首先以專家分群結果進行分析。本研究將文件分別分為 50, 75 及 100 群，並與 2.1.4 節中的 Topic Keyword Clustering 演算法比較，調整門檻值為 0.1, 0.15, 0.5，使其亦為 50, 75 及 100 群。由數值中可以了解，當分群個數愈多，Purity 愈高，但 Recall 也愈低，為了綜合考慮 Purity 或 Recall 的值使結果不致於偏移，本研究將以綜合上述兩個值的計算方法來評估，主要參考 Entropy 及 F-measure。

表 12: 以專家分群結果評估之結果

# of clusters	Topic Keyword Clustering			Our Method		
Estimation	50	75	100	50	75	100
Purity	0.3008	0.4084	0.5259	0.5359	0.6096	0.6295
Recall	0.3239	0.2569	0.2138	0.4709	0.3811	0.3298
Entropy	2.6289	2.0581	1.4362	1.7427	1.348	1.1622
F-measure	0.2472	0.2971	0.3078	0.4339	0.4643	0.4586

將表 12 以圖表顯示。如圖 18、圖 19、圖 20、圖 21 所示。除了 Entropy 愈小愈好之外，其它的值都是愈大愈好，由下列的圖表可知，本研究的分群演算法在這四項數值的評估中的結果都較 2.1.4 的 Topic Keyword Clustering 好。

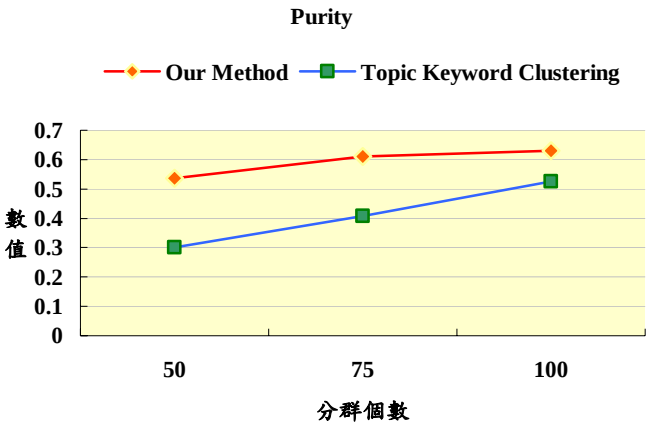

圖 18: Purity 數值之比較

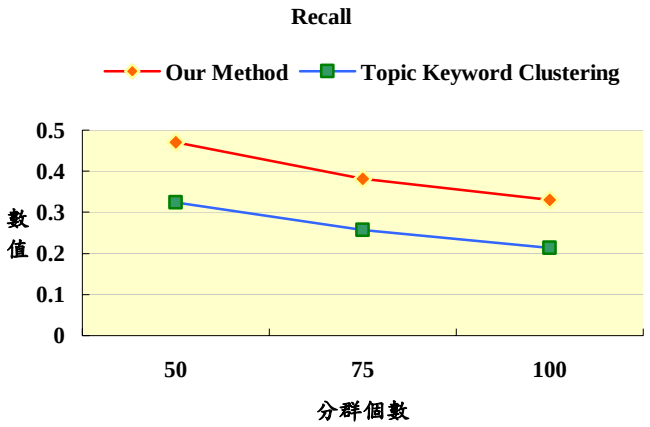

圖 19: Recall 數值之比較

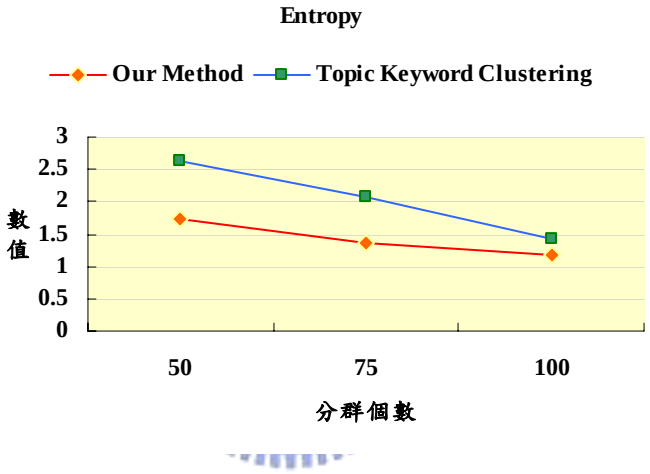

圖 20: Entropy 數值之比較

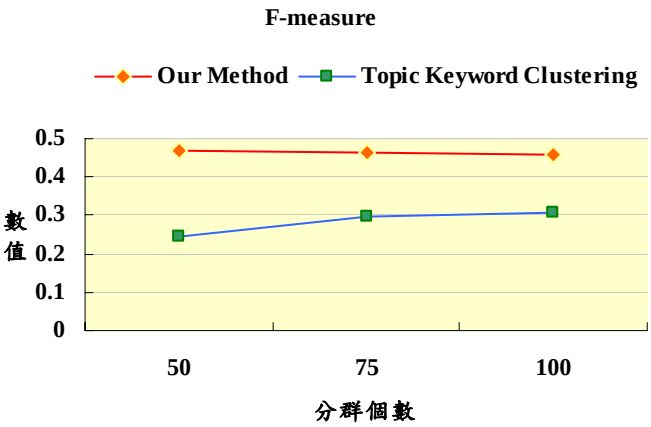

圖 21: F-measure 數值之比較

4.2.2 以群聚分佈評估

使用 Topic Keyword Clustering 與本研究的分群方法，分別將文件分為 50, 75 及 100 群後，分析文件在空間分佈的情形，實驗結果如表 13。

表 13: 以群聚分佈評估之結果

# of clusters	Topic Keyword Clustering			Our Method		
Estimation	50	75	100	50	75	100
Compactness	0.9217	0.8520	0.7841	0.4914	0.4350	0.3806
Separation	0.6086	0.4820	0.4531	0.5031	0.4991	0.4911
Overall Cluster Quality	0.7652	0.6670	0.6186	0.4973	0.4671	0.4358

根據 4.1 的評估方法，是以距離關係表示相似度，故內聚力、分離度及整合內聚力和分離度的 Overall Cluster Quality 的值都是愈小愈好。將表 13 轉化為圖 22、圖 23、圖 24，經觀察可得知，本研究提出的分群方法在內聚力及綜合比較數值的表現上都優於 Topic Keyword Clustering，如圖 22 及圖 24 所示。分離度雖然沒有優於 Topic Keyword Clustering，但是數值上的差距也不大，因此，本研究以群聚分佈的評估方式亦優於 Topic Keyword Clustering。

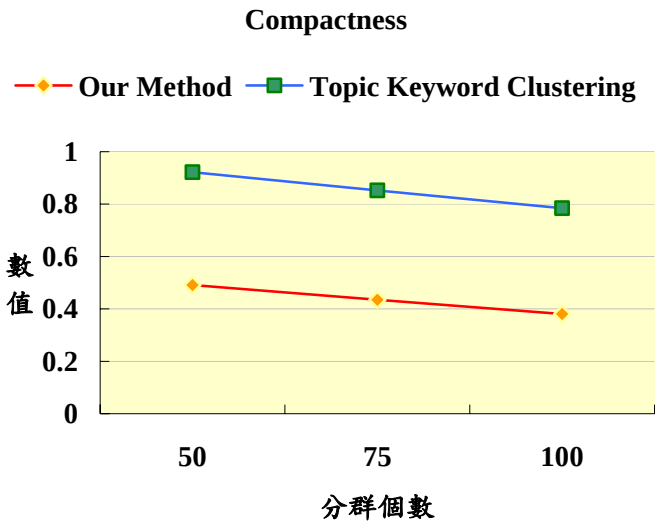

圖 22: Compactness 數值之比較

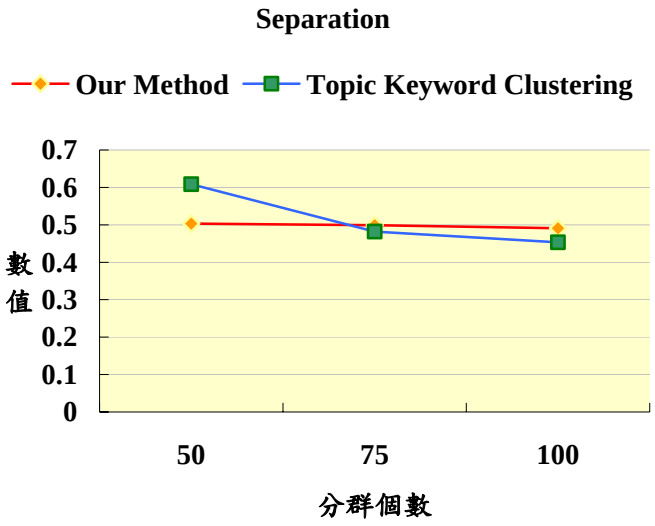

圖 23: Separation 數值之比較

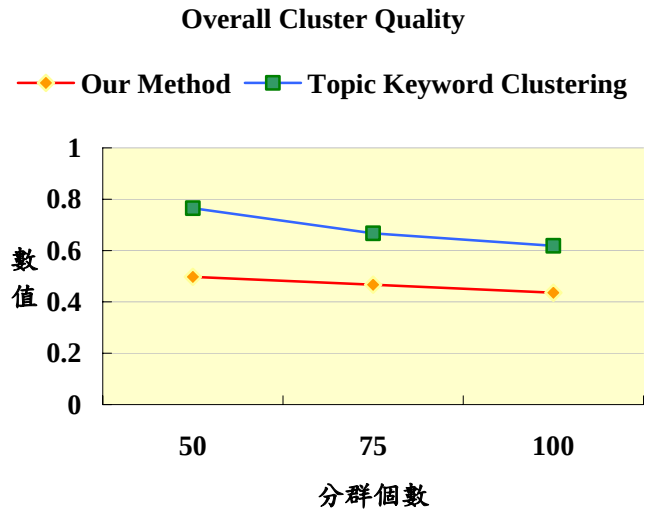

圖 24: Overall Cluster Quality 數值之比較

4.2.3 以專家標示兩兩文章相似度評估

以專家標示兩兩文章相似度評估法的事件分佈表，如表 14、表 15，再比較本研究與 Topic Keyword Clustering 的準確率，如表 16 所示，若將這兩種演算法用於不相似的文章之判斷，其準確度並無太大的差異。根據實驗結果，本研究在相似性文章的判別上較 Topic Keyword Clustering 準確度高，其原因為可以有效地提升正確率(*t_pos*)，將原本以 Topic Keyword Clustering 分群結果為錯誤的樣本，意即落在

f_neg 的樣本，修正至 *t_pos* 中。

表 14: 以 Topic Keyword Clustering 進行專家標示兩兩文章相似度之實驗結果

		專家標示	
		Y	N
分群結果	Y	19	3
	N	69	92

表 15: 以本研究進行專家標示兩兩文章相似度之實驗結果

		專家標示	
		Y	N
分群結果	Y	57	3
	N	32	94

表 16: 以專家標示兩兩文章相似度實驗結果之綜合比較

	Sensitivity	**Specificity**	**Accuracy**
Topic Keyword Clustering	0.2045	0.9485	0.6011
Our Method	**0.6405**	**0.9691**	**0.8118**

4.2.4 實驗討論

根據 4.2.1 及 4.2.2，本研究在兩種評估方法中都有較佳的成果。除了數據上的分析之外，亦需了解這兩種文件分群演算法主要的差異並進行比較，比較的結果如表 17 所示。由表中可知，本研究與 Topic Keyword Clustering 有兩項差異，分別是計算語意相關度與最後文件分群的方法，接下來將針對這兩項進行深入的討論與分析。

表 17 本研究與 Topic Keyword Clustering 之比較

	Topic Keyword Clustering	**Our Method**
Semantic Correlation	Mutual Information	Log likelihood ratio
Keyword Clustering	*k*-nearest neighbor graph approach	
Document Clustering	Cosine similarity measurement	Bisection *k*-Means

1. 語意相關度 (Semantic Correlation)

欲得到良好的文件分群結果，必須先產生適當的概念子群；而適當的概念子群則是建立在正確的語意相關度。在 2.1.4 提到的關鍵字分群 Topic Keyword Clustering 方法，其使用的語意相關度計算公式如(64)所示，即所謂的 Mutual Information (MI) [1]計算方式，本研究則是以 Log Likelihood Ratio [6][10]計算語意相關度。藉由觀察關鍵字分群 Topic Keyword Clustering 演算法與本研究的文件分群演算法所產生的概念子群，了解語

意相關度對分群結果的影響。

$$r_{ij} = f\left(t_i \cap t_j\right) \Big/ MAX\left(f\left(t_i\right), f\left(t_j\right)\right) \tag{64}$$

表 18 是一實例，由此例可以觀察出，使用 MI 做為語意相關度的計算方法，其概念子群內包含的特徵個數較多，但特徵彼此之間的概念並不一致，例如 part-of-speech 及 markov model 是表示不同的概念，但是卻被判斷其屬於相同的特徵子群；反觀以 log likelihood ratio 建立的特徵子群，part-of-speech 及 markov model 分屬於不同群，子群內的特徵概念也較為一致。

表 18: MI 與 log likelihood ratio 產生的概念子群

mutual information	**Log likelihood ratio**
part-of-speech tagging, tagger, institute, markov, text segmentation, markov model, estimation, disambiguation, information retrieval ir, workshop, resolution, tagging, rule-based, probability, series	part-of-speech tagger, part-of-speech tagging, text segmentation, story, tagger
	markov, markov model, threshold, relevance feedback

2. 文件分群方法 (Document Clustering)

當文章包含多種主題或是概念時，本研究使用的分群演算法的正確率會高於 Topic Keyword Clustering，其原因在於 Topic Keyword Clustering 只單單將文章對應至概念子群相似度最高的群，而本研究是依據文章對每一個概念子群的相似度，當兩篇文章都共同包含某些概念時，或是同時趨近於某些概念，即可判斷此兩篇文章為相似。以表 19 為例，文章 A 與文章 B 的主題都是以 Machine Learning 的方式在網路上尋找資料，使用本研究的分群方法可以正確地將文章 A 與文章 B 分在同一群，但使用 Topic Keyword Clustering 則會得出錯誤的結果。

表 19: 比較實例

A	Title: using reinforcement learning to spider the web efficiently consider the task of exploring the web in order to find pages of a particular kind or on a particular topic. this task arises in the construction of search engines and web knowledge bases. this paper argues that the creation of efficient web spiders is best framed and solved by reinforcement learning, a branch of machine learning that concerns itself with optimal sequential decision making. one strength of reinforcement learning is that it provides a formalism for measuring the utility of actions that give benefit only in the future. we present an algorithm for learning a value function that maps hyperlinks to future discounted reward by using naive bayes text classifiers. experiments on two real-world spidering tasks show a three-fold improvement in spidering efficiency over traditional breadth-first search, and up to a two-fold improvement over reinforcement learning with immediate reward only. keywords: reinforcement learning, text classification, world wide web, spidering,
B	Title: A machine learning approach to building domain-specific search engines domain-specific search engines are becoming increasingly popular because they offer increased accuracy and extra features not possible with general, web-wide search engines. unfortunately, they are also difficult and time-consuming to maintain. this paper proposes the use of machine learning techniques to greatly automate the creation and maintenance of domain-specific search engines. we describe new research in reinforcement learning, text classification and information extraction that enables efficient spidering, populates topic hierarchies, and identifies informative text segments. using these techniques, we have built a demonstration system: a publicly-available search engine for computer science research papers.

綜合本節的實驗結果分析，本研究提出的以概念萃取為基礎之文件分群演算法，修正了 Topic Keyword Clustering 的問題，且經多方面評估後都得到不錯的結果；故使用 Log Likelihood Ratio 能夠建立正確的字詞相關度，經由概念萃取得出的概念也能正確地包含語料庫中的主題概念，並且以概念相似度表示文件，亦能準確地表達文件內容，自然可以提升文件分群的正確率。

第五章　結論與未來研究方向

5.1 結論

綜合本論文所述，本研究著重於二個方面：第一為以特徵分群萃取文件中所包含的概念，再依概念的相似度描述文件，最後利用文件分群演算法將文件分群；第二則是將分群的結果以視覺化的方式呈現給使用者，在本論文中，視覺化系統除了以圖形化的方式展現分群結果，同時亦可顯示出群與群之間的關係。

在進行概念萃取時，本研究使用了幾項重要的步驟：

1. 建立相關度

將過濾後的特徵建立關係，且相關度的計算方法是以特徵出現的頻率為基礎，無需事前訓練與學習，並利用 Log Likelihood Ratio 建立字詞之間的相關度。

2. 建立語意網路

根據共現原理建立的相關度可自動化地產生語意網路，不需專家人工建立。且將語意網路圖以圖形理論的原理產生。

3. 產生概念

由語意網路圖找出最重要的核心概念，採用 *k*-Nearest Neighbor Graph Approach 演算法找出核心，由核心向外延伸，同時考慮分群結果的平衡性並維持群內內聚力及群外分離度，並以最適當的特徵個數表現概念。

4. 描述文件

利用概念相似度表示文件，以了解文章整體的概念性，有助於提高文件分群的滿意度。此種描述文章的方法對於提升分群正確率亦有一定程度的效果。

在視覺化的過程中，除了以群聚標記表示每一群的概念，另外也提出以文件中相互引用的關係建立群與群之間的關係，自動化地產生概念之間的相關性。

5.2 未來研究方向

本研究主要在探討萃取文件內的重要概念以及將文件分群視覺化，經實驗結果得知，將文件以概念相似度描述後再進行文件分群，可以獲得不錯的效果。礙於時間因素，本研究尚有多項未完成的想法與進一步發展的空間，討論如下：

1. 建立階層式的概念

在進行合併特徵子群的步驟時，亦保留合併前的架構，以產生概念階層 (Concept Hierarchy)。但概念階層必須建立在較大的語料庫中才有意義。此外，建立階層式架構的關鍵點在於要選擇適當的合併及分割的條件，如何改進演算法及如何決定分割(合併)條件，並考慮應用在大型語料庫中的時間複雜度與實用性等等，都是未來可以研究的方向。

2. 將視覺化系統設計的更具互動性

提供一個可以讓使用者線上搜尋且自行決定資料群數的系統，使得進階使用者能夠根據其需求，調整希望得到的結果，以達到協助使用者更有效率地獲得資訊的目的。更進一步地，可考慮採取使用者自訂模式，將視覺化系統界面以模組化設計，提供使用者自訂界面的功能，並設計更多不同的可用視覺化圖形元件，進以改善視覺化系統的友善度。

3. 加入論文的書目資訊作為分群的條件

對於論文語料庫的未來工作，可將進行人名、期刊名稱、研討會名稱與……等等可用資訊的處理，直覺上同一期刊或同一研討會所收錄的論文應該具有一定程度的相似性，這些相似性將有利於分群工作。

參考文獻

[1] P. Athanasios, *Probability, Random Variables and Stochastic Processes.* ,Second Edition ed.New York: McGraw-Hill, 1984.

[2] H. C. Chang and C. C. Hsu, "Using topic keyword clusters for automatic document clustering," IEICE Trans. Inf. Syst., vol. E88D, pp. 1852-1860, AUG. 2005.

[3] K. Chen and L. Liu, "ClusterMap: Labeling clusters in large datasets via visualization," in *CIKM '04: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management,* 2004, pp. 285-293.

[4] H. Chernoff and E. L. Lehmann, "The use of maximum likelihood estimates in χ^2 tests for goodness-of-fit," *The Annals of Mathematical Statistics,* vol. 25, pp. 579-586, 1954.

[5] K. Chidananda Gowda and G. Krishna, "Agglomerative clustering using the concept of mutual nearest neighbourhood," Pattern Recognit, vol. 10, pp. 105-112, 1978.

[6] L. E. Lehmann, , Testing Statistical Hypotheses. Wiley, 1986.

[7] B. S. Everitt, Statistical Methods for Medical Investigations. ,2nd Edition ed.Edward Arnold, 1994.

[8] J. Han and M. Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.

[9] J. He, A. Tan, C. L. Tan and S. Y. Sung, "On quantitative evaluation of clustering systems." in Clustering and Information Retrieval Anonymous 2003, pp. 105-134.

[10] J. Neyman, E.S. Pearson, "Joint statistical papers," 1967.

[11] A. K. Jain, M. N. Murty and P. J. Flynn, "Data clustering: A review," ACM Comput. Surv., vol. 31, pp. 264-323, SEP. 1999.

[12] G. Karypis, E. H. Han and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," Computer, vol. 32, pp. 68-+, AUG. 1999.

[13] D. A. Keim, "Information Visualization and Visual Data Mining," IEEE Trans. Visual. Comput. Graphics, vol. 8, pp. 1-8, 2002.

[14] D. Koller and M. Sahami, "Hierarchically classifying documents using very few words," in ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning, 1997, pp. 170-178.

[15] J. R. Levine, T. Mason and D. Brown, Lex & Yacc. ,2nd ed.O'Reilly & Associates, Inc, 1992.

[16] K. Leonard and J. R. Peter, *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley-Interscience, 1990.

[17] J. MacQueen, "Some methods for classification and analysis of multivariate observations," Math. Statist, Prob., vol. 1, pp. 281-297, 1967.

[18] J. Makhoul, F. Kubala, R. Schwartz and R. Weischedel, "Performance measures for information extraction," in *Proc. DARPA Broadcast News Workshop*, pp. 249-252, 1999.

[19] C. D. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing.* 1999.

[20] G. Minnen, J. Carroll and D. Pearce, "Applied morphological processing of English," Nat. Lang. Eng., vol. 7, pp. 207-223, 2001.

[21] M. Rosell, V. Kann and J. Litton, "Comparing comparisons: Document clustering evaluation using two manual classifications," in *Proc. Int. Conf. on Natural Language Processing (ICON -- 2004),* 2004, pp. 207-216.

[22] D. G. Roussinov and H. C. Chen, "Information navigation on the web by clustering and summarizing query results," *Information Processing & Management,* vol. 37, pp. 789-816, NOV. 2001.

[23] F. Sebastiani, "Machine learning in automated text categorization," ACM Comput. Surv., vol. 34, pp. 1-47, 2002.

[24] M. Steinbach, G. Karypis and V. Kumar, "A comparison of document clustering techniques," *KDD Workshop on Text Mining,* 2000.

[25] Y. Yang, "Noise reduction in a statistical approach to text categorization," in SIGIR '95: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1995, pp. 256-263.

[26] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," Mach. Learning, vol. 55, pp. 311-331, JUN. 2004.

[27] 劉政璋 Cheng-Chang Liu, "以概念分群為基礎之新聞文件自動摘要系統 Concept Cluster Based News Document Summarization," pp. 67, 民 94.

[28] 謝佩原 Pei-Yuan Hsieh, "目標導向之 SOM 應用於文件分群 Goal-Oriented SOM for Document Clustering," pp. 46, 民 93.

[29] CiteSeer - http://citeseer.ist.psu.edu/

[30] NLP process - Text Analysis Toolkit. Available as http://www.infogistics.com/textanalysis.html

[31] Infogistics, POS -tag - http://www.infogistics.com/tagset.html

[32] Page Rank - http://www.webworkshop.net/pagerank.html

[33] Mopha - http://www.informatics.susx.ac.uk/research/nlp/carroll/morph.html

[34] Kappa Statistics - http://www.dmi.columbia.edu/homepages/chuangj/kappa

附錄

視覺化系統簡介

本系統目前有 541 篇文章，主要是 CiteSeer 資料庫中與 Information Retrieval 相關的論文，圖 25 為首頁畫面，畫面的左邊為分類主題，即是經由文件分群之後選出最適當的特徵表示該類別的主題概念。畫面的右邊則是分群結果，圓上頭的文字表示該類別的主題，圓的大小代表包含文章數的多寡，當圓的半徑愈大，則該類別包含的文章數愈多；顏色的深淺則代表群內的相似度，當相似度愈高顏色愈深。

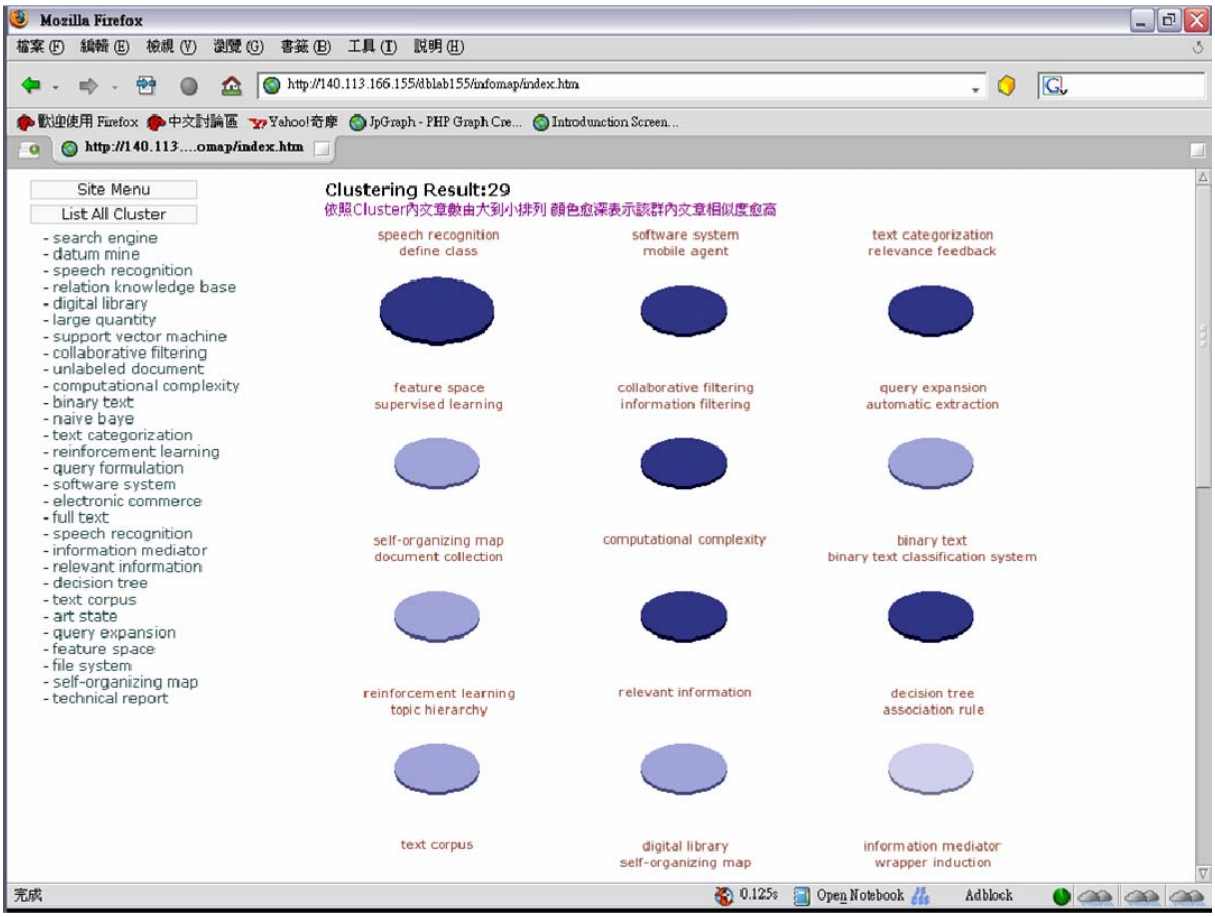

圖 25: 視覺化系統首頁

假設以類別尋找相關的文章，以 Self-Organizing Map 為例，可以從圖 26 畫面左邊的分項類別中尋找與 SOM 相關的主題，在畫面的右邊就會列出 SOM 的相關文章，當

滑鼠移到任一文章的標題時，系統會顯示該文章的摘要，以方便使用者判別該文章是不是他所需要的。除了相關文章的訊息之外，亦可知道有哪些主題與 SOM 相關具有高度的相關性，如圖 26 中以條列式呈現的 Related Topic。為了讓使用者更加了解主題與主題之間的相關性，除了以條列式呈現之外，亦使用雷達圖表示，使主題之間的關係更為顯著。同時可以將圖點選放大，如圖 27 所示。

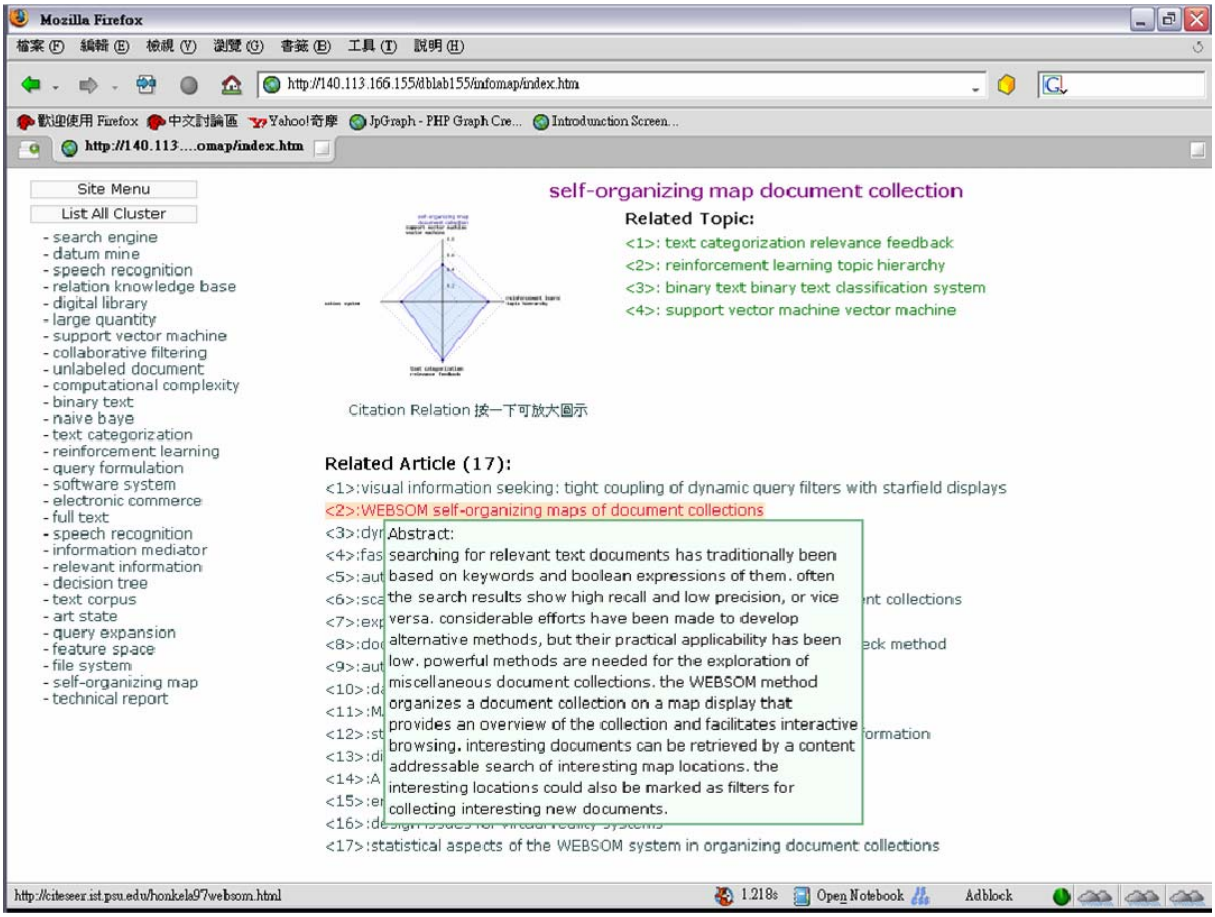

圖 26: 以類別尋找相關文章

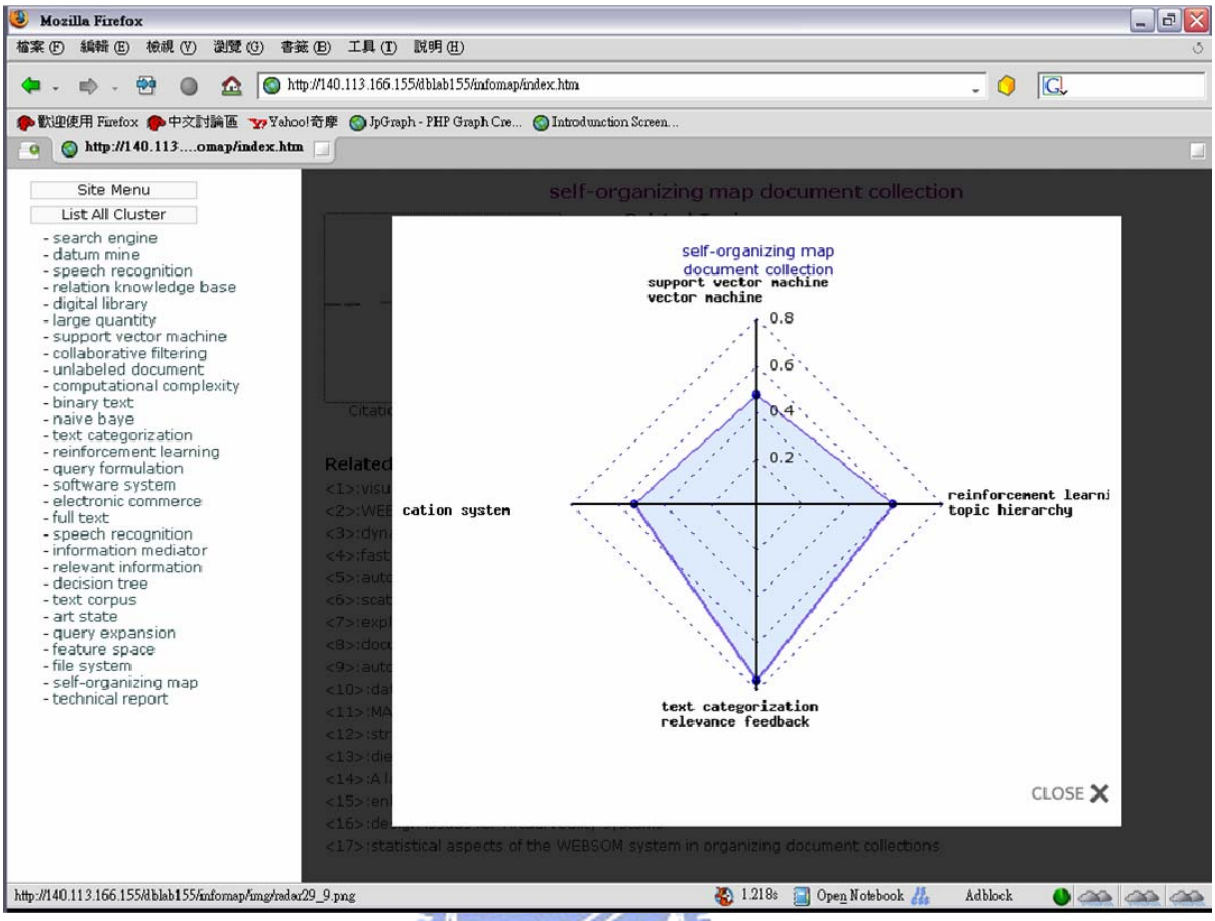

圖 27: Citation Relation 雷達圖放大圖示

除了透過上述的分類架構瀏覽之外，亦可直接利用關鍵字搜尋想要的文章，圖 28 即為關鍵字搜尋結果的畫面，提供使用者與該關鍵字相關的文章及主題，並且以條列式及雷達圖表示搜尋結果在各主題之間的分佈性。

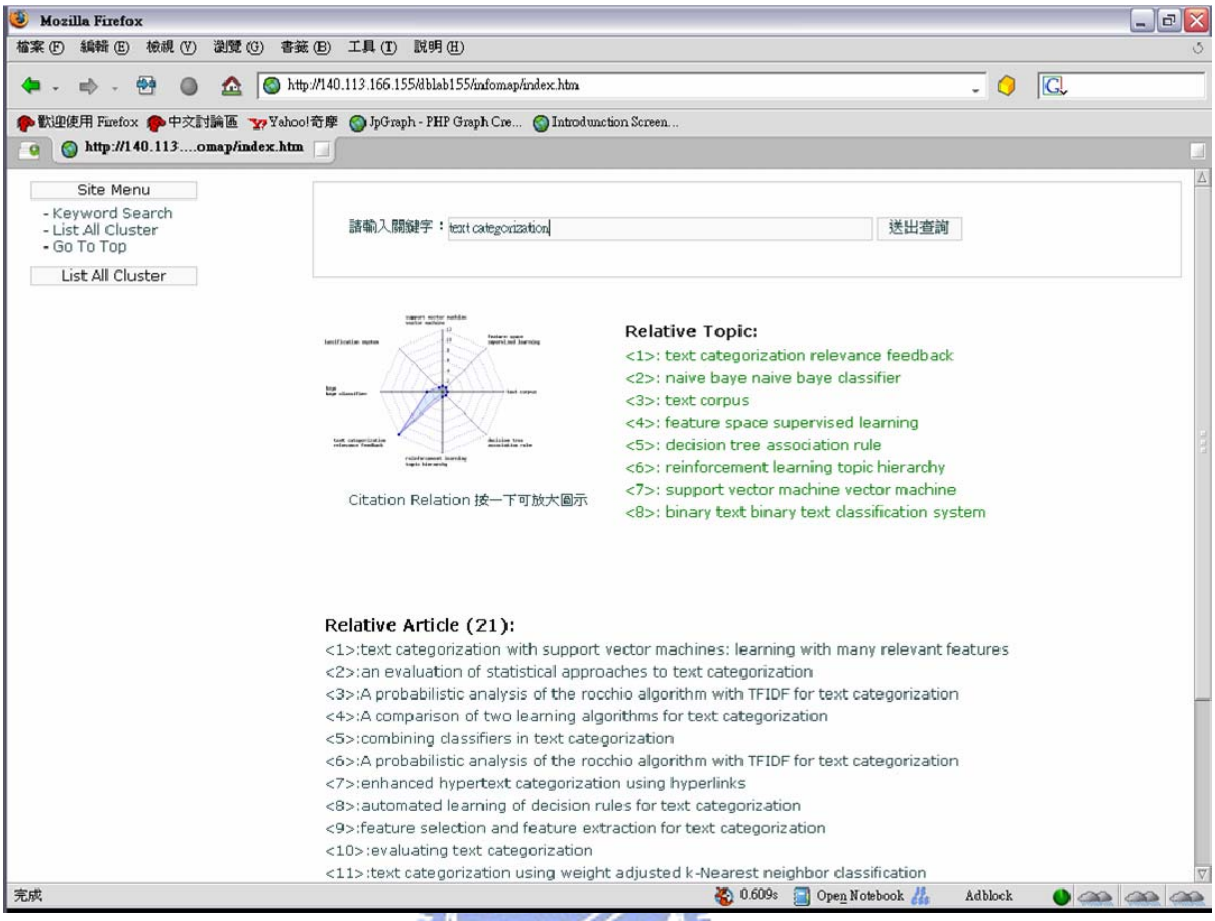
Mozilla Firefox
檔案(F) 編輯(E) 檢視(V) 瀏覽(G) 書籤(B) 工具(T) 說明(H)
http://140.113.166.155/dblab155/infomap/index.htm
歡迎使用 Firefox 中文討論區 Yahoo!奇摩 JpGraph - PHP Graph Cre... Introduction Screen...
http://140.113....omap/index.htm
Site Menu
- Keyword Search
- List All Cluster
- Go To Top
List All Cluster
請輸入關鍵字：text categorization
送出查詢
Citation Relation 按一下可放大圖示
Relative Topic:
<1>: text categorization relevance feedback
<2>: naive baye naive baye classifier
<3>: text corpus
<4>: feature space supervised learning
<5>: decision tree association rule
<6>: reinforcement learning topic hierarchy
<7>: support vector machine vector machine
<8>: binary text binary text classification system
Relative Article (21):
<1>:text categorization with support vector machines: learning with many relevant features
<2>:an evaluation of statistical approaches to text categorization
<3>:A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization
<4>:A comparison of two learning algorithms for text categorization
<5>:combining classifiers in text categorization
<6>:A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization
<7>:enhanced hypertext categorization using hyperlinks
<8>:automated learning of decision rules for text categorization
<9>:feature selection and feature extraction for text categorization
<10>:evaluating text categorization
<11>:text categorization using weight adjusted k-Nearest neighbor classification
完成
0.609s
Open Notebook
Adblock

圖 28: 關鍵字搜尋結果