文件剖析方法對提升資訊檢索精確率影響之研究

張錫正
私立華夏工商專科學校
台北縣中和市華新街 111 號
台灣科技大學資訊管理研究所
hcchang@cc.hwh.edu.tw

徐俊傑
台灣科技大學資訊管理研究所
台北市基隆路四段 43 號
Cchsu@cs.ntust.edu.tw

殷欣靖
台灣科技大學資訊工程研究所
台北市基隆路四段 43 號
M8815006@mail.ntust.edu.tw

摘 要

網際網路技術的快速進步及普及，網路上電子式文件(electric documents)資料亦巨幅成長，網路形成一個巨大的資料倉儲（data warehouse）。搜尋引擎（search engine）是目前網路上文件資料搜尋的主要工具，由於現有的搜尋引擎大多使用關鍵詞彙為基礎（keyword based）的查詢方式，此種查詢方式一般使用者所輸入的查詢詞彙個數有限，加上語言學上的「一字多義」與「一義多詞」的問題，易導致查詢主題模糊難於確認，在面對網路如此巨大的資料量時，常會找到太多不相關的資料，使真正的資料隱沒其中。另一方面，由於網路資料的動態變化性、資料格式的多樣性及資料的快速巨幅增長，使得搜尋引擎搜尋結果之精確率已大幅降低到其實用性受到嚴重考驗。

為解決上述問題，除了更新搜尋引擎所用之檢索技術外，如何協助及早確定使用者查詢主題，精確且有效地描述出查詢者的查詢意向，有效且大幅地過濾掉不相關的搜詢結果，無外乎是提高搜尋結果之精確率的最有效方法。本論文提出：（一）詞彙群組剖析（二）反查式搜尋剖析（三）段落式文件剖析（四）文件段落比對剖析及（五）相關性回饋調適等五種方法來協助及早確認使用者查詢主題與過濾掉不相關的搜詢結果，經由實驗結果來評估這四種文件剖析方式對於查詢結果之精確率的影響。另根據使用者搜尋紀錄與閱讀行為找出適當的回饋文件，使查詢主題更精確進而獲得最佳的查詢效能。

關鍵字：資訊檢索、文件剖析、查詢主題、相關回饋。

壹、 緒論

隨著網際網路的快速成長，網路上的資訊與日暴增，舉凡科技研究、商業經濟、政府的行政策施及人們日常生活的食、衣、住、行等，無不大量利用網路的方便性來完成。如何在這巨大浩瀚的網路中找尋有效的資訊，是目前使用網路所面臨的一個重要且日愈嚴重的問題。網路搜尋引擎無外乎是目前解決此問題的有效方法之一。然而由於網路上資訊格式的多變性， 資訊的動態變化性及目前所使用之資訊檢索(information retrieval)技術的問題，使得由搜尋引擎所搜得之資料的量之多與精確性之低，已大大地降低了其實用性。目前大家使用搜尋引擎所面臨的問題是，搜尋結果之資訊量過大，精確率過低，因而無法獲得自己真正想要的資料。

此一事實是目前從事資訊檢索技術研究者所急欲克服解決的問題，目前的資訊檢索系統， 大部分都是使用關鍵詞彙為基礎的方式查詢，面對巨大的資訊量這種資訊檢索系統在系統建構上較為簡易、快速。但這種技術在建構文件索引時，已徹底破壞了文件資料原有的文法結構、文句意涵、文件內文前後文修辭關係，使詞彙所代表的原意涵模糊了，當查詢系統在面對使用者所下達的有限個查詢關鍵詞彙（query keyword）時，對於在語言學上的「一字多義」與「一義多詞」的問題將全然無解，因而大大地降低了查詢結果的精確率。另一方面，當使用者對於自己所欲搜尋之資料概念模糊、對欲搜尋之資訊所屬的領域不甚熟悉、所下達的查詢條件不足、關鍵字使用不恰當亦或是語意模糊等，都將使問題更為嚴重最後終將無法找到自己真正想要的資料。

搜詢引擎是目前網際網路資料搜尋的重要工具之一，現有的網路中文資訊搜尋系統如openfind，google，yahoo 等，都是經由使用者下達查詢關鍵詞彙(query word)的方式來與系統事前建立的文件全文索引(full text indexing)關鍵詞資料庫進行比對，並以查詢關鍵詞彙出現在文件中的頻率，經一文件相似度計算機制來計算文件與查詢關鍵詞彙的相關度，然後依此相關度的大小依序將查詢結果排列顯示。這種簡單直覺且快速的作法，是完全以詞彙出現與否及出現頻率來作判斷，這種作法有下列幾個因素會直接影響到查詢結果的精確率(precision rate)[1]：

1. 使用者對查詢主題所屬領域的熟悉度：
   當使用者對於他所欲搜尋之資料所屬的

領域十分瞭解時，則他便可適切地使用足以代表該查詢主題的關鍵詞彙來搜尋資料。反之、若對其查詢主題所屬領域不甚熟悉時，則使用者可能因無法使用適當的詞彙而必需花費許多的時間，反覆下達不同的查詢詞彙來進行資料搜尋，如此將嚴重考驗使用者的耐心也造成網路使用的浪費，而且不保證找到所要資料。

2. 使用者所給予之查詢詞彙的多寡：

研究顯示，當使用者從網路上搜尋文件資料時，系統若能從使用者所給的查詢詞彙挖掘到越多的資訊，就越能夠找到相關的資料。但根據[2]實際統計數值顯示，一般網路使用者在網路上找尋資料時所下達的查詢詞彙的個數平均值為 2.48 個。當系統面臨查詢詞彙所能提供的資訊不足時，便無法掌握查詢主題，則所能挖掘到的資訊就會變少且不相關，所導致的結果就是查詢的召回率(recall rate)下降。

3. 檢索系統對於使用者查詢意向的掌握度：

每個使用者都有其感興趣的領域，單單計算使用者所下達的關鍵詞彙與文件索引詞彙的相關度來作為文件相關與否的判斷並不十分可靠，因為相同的詞彙使用在不同的領域可能具有完全相反的意義，在查詢時是否能掌握住查詢詞彙所要表現的真正意涵，使其真正代表每一位使用者的興趣方向，亦是影響搜尋結果的重要因素之一。

本研究既針對上述三項嚴重影響查詢結果精確率之要素尋求有效解決方法，系統依據使用者搜尋紀錄與閱讀行為模式來找出適當的查詢詞彙組與相關性回饋文件，經由詞彙群聚性剖析與文件內容剖析來找出最有效最經精緻的查詢詞彙組，以獲取最佳的查詢效能。經實驗測試證實，不同的文件內容剖析對不同的文件格式具有不同的效果，但綜合使用所有剖析方法確實對整體查詢結果有顯著的效果。

貳、 相關研究

解決因查詢關鍵詞彙不足所產生之查詢結果不精確之問題的有效方法之一既使用相關性回饋(relevance feedback)技術。首先，使用者下達關鍵詞彙來查詢資料，初步的查詢結果經使用者流覽過濾後並回饋相關文件給查詢系統，系統從此相關文件中萃取出更多的有效詞彙來補充原查詢詞彙之不足，然後再次進行查詢，如此重複進行以提高查詢結果的精確率。相關性回饋技術的研究如 Chia-Hui Chang 及 Ching-Chi Hsu[3]曾提出以查詢主題相關性回饋法（concept-relevance based feedback），來輔助查詢關鍵字的不足，以改善查詢結果的精確率。Chuan-Chuan Lin、Shou-Yi Tseng 與 Pei-Min Chen[4]提出以建構概念網路(concept network)的方式，透過概念矩陣(concept matrix)的運算，先找出文件的分類，然後再依照查詢所形成的概念向量與概念矩陣的比對運算，以找出相關文件，研究中並結合了模糊集合模型(fuzzy set model)與潛在語意索引模型(Latent Semantic Indexing Model)的運用。

另一類的作法則是針對已知的文件格式，作特徵詞彙的擷取，如 Steve Lawrence、Kurt Bollacker 與 C. Lee Giles[5,6,7]曾針對科技類文章作處理，找出該類文章的結構特性來解決文件查詢的問題。而 Jong P. Yoon 及 Sungrim Kim[8]則以 XML 格式文件作為處理對象。這些研究的目標都是放在解決特定文件格式查詢的方法上。但在面對非特定結構文件時，其查詢結果的精確率則有待改善。

Jinxi Xu 與 W.B. Croft [9] 曾研究探討「錯誤比對」 (mismatch)在資訊擷取中所產生的問題，研究發現使用者搜尋資料時所用的關鍵詞彙(keyword)經常與文件作者所使用的詞彙不同，所以系統在查詢時會因為在文件中找不到相同的查詢詞彙，而錯失原本是使用者所要的相關的文件。然而對於使用者而言，當他對所要搜尋之資料的主題在心中無法形成一個很清楚的概念時，則在他所下達查詢關鍵詞彙時，通常會很短或是使用了不適當的查詢關鍵詞彙，因而造成查詢結果的偏差。基於上述的問題，實有必要研究一個輔助機制來幫助使用者找出正確範圍的查詢詞彙。

有許多研究致力於改善此一問題，如 Jinxi Xu 與 W.B. Croft [9]就曾研究並提出從查詢出的文件中找出權重值較高的幾篇文件，分析文件中詞彙與查詢詞彙的關係，找出相關度高的詞彙來擴充查詢詞彙。A. M. Tjoa、M. Hofferer、G. Ehrentraut 與 P. Untersmeyer [10]則提出使用基因演算法(genetic algorithm)，去尋找與查詢詞彙最接近的文件，由該文件中的詞彙來擴充查詢詞彙，反覆計算以趨近使用者所要查詢的方向。M. Mitra、A. Singhal 及 C. Buckley[11]則提出利用模糊理論(fuzzy theory)的方式來過濾文件，由過濾出的文件中找出詞彙來擴充查詢詞彙。C. Buckley 、M. Mitra.、J. Walz 與 C. Cardie[12]提出將搜尋到的文件進行文件分群(cluster)，由查詢詞彙與各個文件分群進行比對，找出最接近的文件分群，再由該文件分群中挑出詞彙來擴充。

使用相關性回饋(relevance feedback)技術雖可改善查詢關鍵詞彙不足的問題，但也不可以無限制地使用，若使用過當也容易造成查詢主題發散，使最後的查詢結果越變越差。事實上，使用相關性回饋技術，是在使用者下達查詢指令查詢後，再經由初步結果來修正的技術，這種事後的補正工作，可以使最後結果便佳。但若使用者首次下達的查詢關鍵詞既有偏

差，會因修正回饋次數過多，讓使用者失去耐心，最後終將達不到預期之效果。事實上，前述的所有研究與努力無外乎是要及早確定出使用者心中所欲。使用者的查詢主題一經確立，查詢系統便能有高精確查詢結果輸出。

在日常生活中我們經常遇到，在閱讀完一篇文件後，很想再尋找與其類似或有關的文件來作為輔佐與參考。如果利用現有的搜尋引擎作為查詢的工具，則使用者必需自行由文件中挖掘出查詢資訊再輸入搜尋引擎進行搜尋。幸運的話，可能很快就能再次找到所需要的文件。反之，則可能需要經過多次反覆更改查詢詞彙才能查得。也有可能因所下達的查詢關鍵詞彙焦點不集中，而搜尋到太多根本不相關的資料，而失去找尋的耐心。所以如何能由文件自動去找尋相關的文件，是值得加以研究的課題。

欲及早確立查詢者的查詢主題，以文件為基礎的查詢（document-based query）不外乎是一種最有效的方法。基於這些考量，我們提出一個以文件為基礎的查詢系統，由使用者任意給定一查詢文件，系統直接進入文件內容自動剖析，挑出最足以代表該篇文章主題的重要關鍵詞句來進行相關文件搜尋。此外我們也將提出一個有別於一般相關回性饋的作法，系統會依個人的查詢行為模式作為量測依據，來加速確立使用者的搜尋目標，使得文件搜尋範圍趨近使用者所要尋找的方向，以求提高查詢速度與查詢的精確度。在本研究中希望在完成這整個系統的建構後，能夠達到以下的目的：

1. 提供個人化的查詢協助，幫助使用者查詢相關的文件。
2. 減輕使用者查詢時的負擔，達到較佳的查詢效能。
3. 增強與使用者的互動關係，建立有效的回饋機制。

參、 實作方法

如何從文件集與查詢文件中找出最適當的代表性詞彙來代表該文件是檢索系統首要面對的問題，經由個自之代表性詞彙的比對運算，以找出與查詢詞句相似程度最高的文件來，這看似乎簡單，其實是檢索系統最大的問題所在。這是因為文件是作者的思維表現，作者個人用字遣詞習慣不同文章的結構也不具一定的規則性，所以現有的詞彙比對技術僅能尋找一些較為合宜的數學模型，輔以一些經驗法則，來決定文件的相關與否。

本研究的整個系統架構如圖 1 所示，分為中文斷詞與關鍵詞辨識、文件索引、主題確認與關鍵詞萃取、查詢系統及相關性回饋五個子系統。前二者屬前置處理系統，後三個則為線上即時處理系統。所有蒐集到的文件都會先經由中文斷詞與關鍵詞辨識子系統找出有效的代表性詞彙，然後再將文件集建構成反向索引檔(inverted file)。線上即時處理系統則分為查詢主題確認與關鍵詞萃取，使用者查詢介面與相關性回饋機制，以下將分別詳述其作法。其中針對文件剖析的方式，在此將提出四種剖析方法並探討其有效性。

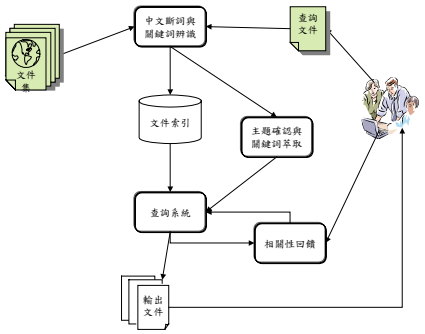

圖 1： 系統架構圖

一、詞彙群組剖析(Term Set Analysis)

依一般文章觀察研究結果，不同的作者在闡述某一相同主題時，會以個人所熟悉的詞彙來描述之，描述該主題事件的相關用語相對地也會重複出現。依據這個觀察事實，欲從查詢文件內容挑出足以代表該文件的關鍵詞彙。首先必需從查詢文件中的眾多詞彙挑出重要性較高的詞彙，依 Luhn[13]研究，指出統計一篇文件每個字出現的頻率，能夠初步有效地判斷一篇文件的關鍵詞彙。研究指出發生頻率越高的詞彙越能代表該文件，且可以作為該文件的索引詞彙。此外 Luhn 的研究也發現，通常文件中出現頻率最高的詞彙及出現頻率最低的詞彙，並不適用於代表該文件。原因是研究發現，出現頻率最多的詞彙通常是一些功能詞彙(stopword)，而出現最少的詞彙則為文件作者所使用的冷門詞彙，其他作者不一定會一樣使用此詞彙，這些詞彙對於相關文件查詢並無太大的助益。基本上這些出現頻率高的詞彙及出現頻率低的詞彙並不能有效的表達出該篇文章的闡述主題。所以依此研究結果，我們可將每一文章中的這類詞彙濾除到。根據 Zipf's Low[13]，文章初步去除這些功能字後我們約能降低文章的檔案大小約 20%~50%。

雖然 Luhn 與 Zipf 所研究的環境為英文，但在中文環境實質上亦有相同情形。依據由所蒐集到之文件的統計值，系統找出超過上限值的高頻率的詞彙共 498 個功能詞彙。

在去除功能詞彙後，我們依照 Luhn 的立論計算每個詞彙 i 在文件 j 中出現的頻率：

$$f_{i,j} = \frac{freq_{i,j}}{\max_l freq_{l,j}}$$

其中，$freq_{i,j}$ 為詞彙 i 出現在文件 j 的頻率，$\max_l freq_{l,j}$ 表示文件 j 中出現頻率最高的詞彙 l。

依 $f_{i,j}$ 的大小對每個詞彙排序，我們設定兩個變動的門檻值，分別為 threshold1 及 threshold2(可由使用者視查詢狀況更動，threshold1>threshold2)，依序挑出頻率大於 threshold1 的詞彙，並在查詢文件中找尋與該詞彙相鄰距離為 n (可由使用者視查詢狀況調整)且頻率大於 threshold2 的詞彙，經由這些步驟可以挑出不同的詞彙群組 $TermSet_i$。然後再使用布林運算模式將 $TermSet_i$ 內的 $term_{i,j}$ 及 $TermSet_j$ 之間關係組合起來，即：

$$(term_{1,1} \cap term_{1,2} \cap ..... \cap term_{1,n_1}) \cup ...$$

$$... \cup (term_{m,1} \cap term_{m,2} \cap ..... \cap term_{m,n_m})$$

作為查詢到先前所建立的反轉置檔案索引中找尋相關文件，每一篇搜尋到的文件 D_t，是經由計算 $TermSet_i$ 與 D_t 的相似度 $SimilarityScore(D_t)$作為相關度排序的依據。相似度計算方式如下所示：

$$SimilarityScore(D_t) = \underset{i=1}{\overset{m}{Max}}\left(\sum_{j=1}^{n_i} \left(freq(term_{i,j}, Q) \times freq(term_{i,j}, D_t) \times \log\frac{N}{Docfreq_{i,j}}\right) / n_i\right)$$

其中，$freq(term_{i,j},Q)$表示詞彙 $term_{i,j}$ 出現在查詢文件 Q 中的頻率，$freq(term_{i,j},D_t)$表示詞彙 $term_{i,j}$ 出現在文件 D_t 的頻率，N 表示文件集的總篇數，$DocFreq_{i,j}$ 表示詞彙 $term_{i,j}$ 出現在文件集中的篇數，n_i 表示 $TermSet_i$ 中 $term_{i,j}$ 的個數，m 表示 $TermSet_i$ 的個數。

二、反查式搜尋剖析(Term Set Track Analysis)

單考慮找尋詞彙群組作為搜尋運算尚嫌不夠周延，因為系統在建立反向索引檔前，會先分析所有文件，接著萃取每篇文件中的重要詞彙來代表該文件，然後再建成詞彙－文件索引形式，既詞彙共出現在那些文件中。所以當使用者給予一查詢文件時，系統既可查出查詢文件內之重要詞彙包含於哪幾篇文件中，及該詞彙在文件中的重要性(權重值)。使用反向索引檔式資料結構，我們很難反過來查知，被查出的相關文件中所包含的所有詞彙資訊，所以很難再進一步確認該文件的相關程度。當使用詞彙群組來作查詢時，我們所分析的詞彙群組是根據文件中相鄰近且出現頻率高的詞彙，在用於查詢時並不能保證所查詢到的相關文件中的詞彙，也是同於查詢文件中詞彙叢聚的情況分佈。例如文件範例 1，我們得到詞彙群組{“升高”,”氣象局”,”明天”}，這組詞彙群組中的各個詞彙可能散佈在被查詢到之相關文件中的各個不同段落中而無群聚的現象，雖然這三個詞彙都存在於文件中，但是並不像文件範例 1 的文件一樣，是三個詞彙叢聚在一起。這當文章各段落所描繪的主題不相同，但確有以上這三個詞彙出現且出現頻率也不小時，就會有查詢結果錯誤情形產生。為了要解決這個問題，我們使用一反查式搜尋剖析法來克服此一問題。作法是在求得查詢到的文件之 $SimilarityScore(D_t)$時，再逐一反查各個文件詞彙群組叢聚的頻率。所使用的公式為：

$$SimilarityScore(D_t) = \underset{i=1}{\overset{m}{Max}}\left(\sum_{j=1}^{n_i} \left(freq(term_{i,j}, Q) \times freq(term_{i,j}, D_t) \times \log\frac{N}{Docfreq_{i,j}}\right) / n_i\right) + \sum_{i=1}^{n_t} freq(TermSet_t, D_t)$$

其中，$freq(term_{i,j},Q)$表示詞彙 $term_{i,j}$ 出現在查詢文件 Q 中的頻率，$freq(term_{i,j},D_t)$表示詞彙 $term_{i,j}$ 出現在文件 D_t 的頻率，N 表示文件集的總篇數，$Docfreq_{i,j}$ 表示詞彙 $term_{i,j}$ 出現在文件集中的篇數，n_i 表示 $TermSet_i$ 中 $term_{i,j}$ 的個數，m 表示 $TermSet_i$ 的個數，$freq(TermSet_i, D_t)$表示詞彙群組 $TermSet_i$ 出現在文件 D_t 中的頻率。

三、段落式文件剖析法(Paragraph Query Set Analysis)

在一篇文章內容長度中等或是長文章中，其內容所討論的主題經常不是唯一，例如一篇名為「人工智慧於網際網路資訊檢索系統上的應用」之文章，其內容可能分段討論人工智慧技術（模糊理論，類神經網路，基因演算法 …等）網際網路技術及資訊檢索技術等。文章內容包含眾多主題，若將整篇文章整體看待，則在進行關鍵詞彙挑選時，所挑出用來代表該文章內涵之關鍵詞彙將易發散主題不集中。相對地，終將導致查詢結果不精確亦或是查詢結果之文件非常多但相關度都不高的情形。

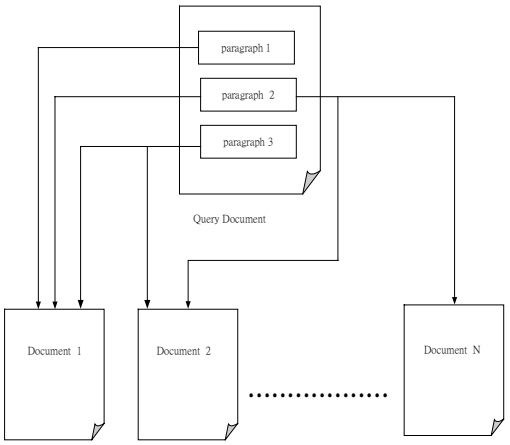

圖 2： 查詢文件與文件間段落之關聯狀況

前述兩節中所敘述作法一及作法二是在

查詢文件中找出詞彙群組 *TermSet*，並以($term_{1,1} \cap term_{1,2} \cap \ldots \cap term_{1,n1}$) $\cup \ldots \cup$ ($term_{m,1} \cap term_{m,2} \cap \ldots \cap term_{m,nm}$) 的模式進行查詢，此二作法尚未考慮到當詞彙群組中的詞彙是跨越文件的兩個段落的狀行。但由上述顯然可知，一篇文件所描述的主題事件不一是唯一情形是常有的事，也就是說極有可能上一個文件段落描述的是有 A 主題事件，下一個文件段落所描述的卻是 B 主題事件，而 A 主題事件與 B 主題事件兩者可能是南轅北徹的，所以當詞彙群組發生在這種情形時，其於查詢文件的代表性就有待考量且必須加以調整。對於這個狀況我們所提出的方法是，加入文件段落結構的因素考量，查詢文件的剖析不再是以文件整體為單位，而是以文件段落為剖析單位。我們分別以文件各個段落所決定的詞彙群組去作查詢運算，累計各篇查詢到的文件與查詢文件的相似度關係來決定文件相關程度。

如圖 2 所示，假設每一個箭頭代表文件段落與文件的關係，且設定每個關係的權重值皆為 1，我們可以看到 Document 1 與 Query Document 的三個段落都有關係，所以說 Document 1 與 Query Document 的相似程度因該很大才對。不管 Query Document 的三個段落所描繪的事件是否唯一，儘管三個段落分別描繪三個獨立的事件，那也表示 Document 1 的內容應該與這三個事件有密切關係，否則不該有箭頭指向文件。

我們一樣先去除功能詞彙，並統計在查詢文件 Q 中的各個詞彙的 $freq(term_{i,j}, Q)$，之後分別找出各個段落的查詢詞彙 $QueryTermSet_i$，由 $QueryTermSet_i$ 去查詢相關文件，計算查詢文件 Q 與文件 D_t 之 $SimilarityScore(D_t)$，公式表示如下：

$$SimilarityScore(D_t) = \sum_{i=1}^{s} \rho_i \left(\sum_{j=1}^{n_s} \left(freq(term_{i,j}, Q) \times \frac{n_{seg}}{n_Q} \times freq(term_{i,j}, D_t) \times \log \frac{N}{Docfreq_{i,j}} \right) / n_s \right)$$

其中，$freq(term_{i,j}, Q)$ 表示詞彙 $term_{i,j}$ 出現在查詢文件 Q 中的頻率，$freq(term_{i,j}, D_t)$ 表示詞彙 $term_{i,j}$ 出現在文件 D_t 中的頻率，n_{seg} 表示 $term_{i,j}$ 在第 i 個段落中出現的次數，n_Q 表示 $term_{i,j}$ 在查詢文件 Q 中出現的次數，N 表示文件集的總篇數，$DocFreq_{i,j}$ 表示詞彙 $term_{i,j}$ 出現在文件集中的篇數，n_s 表示 $QueryTermSet_i$ 中 *term* 的個數，m 表示查詢文件 Q 段落的數目，ρ_j 表示各個段落的權重值。

四、文件段落比對分析(Paragraph Match Analysis)

在上一節的作述的作法中，我們把原本以全文為單位的查詢，切成一個個以段落為單位的查詢，經由這些以段落為單位的查詢去累計文件 D_t 之 $SimilarityScore(D_t)$。此法在統計相似度上，對於被查詢到之文件而言並不是以段落來作為衡量，仍是以詞彙在 D_t 全文中出現的 TFIDF 來作為計算依據，為此我們設計完全以段落為考量的比對方法來作為和上一節中的作法之對照比較。

在這裡我們作法除了將查詢文件作段落切割外，也把被查詢文件作段落分割，查詢文件與被查詢文件中的段落兩兩進行相似度比對，而以段落之間的相似程度來決定文件的相關度。首先找出被查詢文件 D_t 中各個段落與查詢文件 Q 最為相關的段落，累加這些這些段落的相似值作為該文件 D_t 與查詢文件 Q 的相關聯程度。此時用於計算相似程度 $SimilarityScore(D_t)$ 的公式為：

$$SimilarityScore(D_t) = \frac{m}{n} \times \left(\sum_{i=1}^{m} \underset{j=1}{\overset{n}{Max}} (Similarity(P_{i,D_t}, P_{j,Q})) \right)$$

其中 $Similarity(P_{i,Dt}, P_{j,Q})$ 表示在文件 D_t 中的段落 P_i 與查詢文件 Q 中的段落 P_j 的相似度值，m 表示文件 D_t 的段落數，n 表示查詢文件 Q 的段落數。

在計算查詢文件段落 P_j 與經由查詢得到的文件 P_i 之間相似關係 $Similarity(P_{i,Dt}, P_{j,Q})$，最簡單的方式為 $|P_j \cap P_i|$，測量 P_j 和 P_i 的交集程度，不過使用這個方式僅能滿足一般性的檢測工作。在本研究中是採用 *cosine coefficient* 相似度量測法來計算文件之相似程度，其計算公式如下所示：

$$cosine(X_i, Y_j) = \frac{\sum_{i=1}^{n} (X_i \times Y_i)}{\sqrt{\sum_{i=1}^{n} X_i^2 \times \sum_{i=1}^{n} Y_i^2}}$$

其中 X_i 和 Y_i 分別表示為從文件 Q 及文件 D_t 中所挑選出的詞彙 TFIDF 值。

如圖 3 所示，假設被查詢文件 Document 1 中段落 e、h 分別與查詢文件的段落 a 有最大的關聯值 0.5 及 0.3，段落 f 與段落 d 有最大的關聯值 0.6，段落 g 與段落 b 有最大的關聯值 0.7，因此我們可以計算出 Document 1 與 Query Document 的相似程度為 (4/4)*(0.5+0.6+0.7+0.3) =2.1。

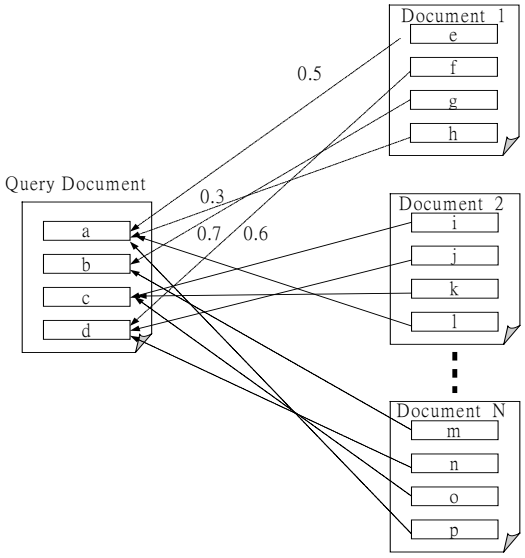

圖 3： 段落比對分析

五、相關性回饋機制

本研究所用的查詢方法，是由剖析文件去找出相關的文件，所以在查詢詞彙資訊方面要比由使用者下達查詢詞彙要多得多，相對地所要面臨的問題是如何從查詢文件內眾多的詞彙中挑選出適合的詞彙作為查詢詞彙，基本上我們並不以分析查詢詞彙與相關文件的相似度，找出擴充的詞彙。因為研究發現在一篇查詢文件中，所描述不只為單一事件，內容越長的文件可能描述的事件越多，針對不同的使用者在查詢時所專注的事件可能不盡相同，是故以全文作為查詢，可能會找出與該查詢文件相關的文件，但內容卻不是使用者所關心的事件描述。所以在製作輔助的機制時，為針對使用者個人閱讀興趣選擇，強調在找尋使用者的閱讀重心，由閱讀重心去擴充查詢資訊，而不是完全比照關鍵字查詢時，所用的擴充查詢詞彙的作法。

在此我們將加入時間參數的考量，主要目的是去偵測使用者瀏覽被查詢到之文件的行為，統計使用者在閱讀文件的時間，來界定使用者對於該文件的重視程度。所用的時間參數定義如下：

$$TimeFunction(D_t) = \frac{Time_{read}(D_t)}{\#\ words\ of\ document\ D_t}$$

其中 $Time_{read}(D_t)$ 表示使用者閱讀文件 D_t 所需的時間。

經由時間參數挑出使用者較重視的文件後，即可分析這些使用者所重視的文件與查詢文件之間的關係，藉由這些關係的分析，設法從中找尋使用者的查詢重心，剖析重心落於查詢文件那個段落之中，以更新查詢文件各段落的權重，而進一步趨近使用者所希望的查詢領域。

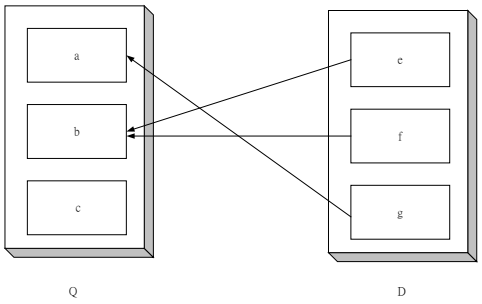

圖 4： 查詢文件段落重心找尋

在經由段落的相關資訊求得之後，我們可從中找出調整查詢文件各段落的權重比例，如同圖 4 所示，假設有查詢文件 Q 與經查詢後，經由時間參數找到使用者認定極具相關聯的文件 D 時。則可以經由段落比對方式，從文件 D 中計算出 e、f、g 段落分別與文件 Q 中的 a、b 段落有極高相關程度，由這些段落相關聯度關係，可以得知在文件 Q 中的三個段落，a 與 b 應該是屬於使用者較感興趣的兩個段落，是故在進行更進一步的查詢時查詢時，我們可以特別針對這兩個段落加重其權值，讓查詢重心落於這兩個段落中，經由這個方法在經過多次比對淬煉之後，可以更趨近使用者的瀏覽方向，讓查詢結果更為精確。

肆、 實驗分析

我們由新聞網站上所收集的 1200 篇的新聞文件作為測試文件，在以下的實驗中將一一分析系統在查詢相關文件的精確度。

實驗一:

在我們的詞彙群組式剖析法中需對三個參數進行設定，如表 4-1 所示(見附錄)，分別是 threshold1 、threshold2 與 distance（在表 1 中分別以 α 、β 、γ 表示）。我們共挑選出 21 組較為有效的參數設定，由我們所蒐集的 1200 篇文件中任選 10 篇文件進行查詢分析，並從實驗數據中找出較佳的參數設定。評估表 1 的實驗結果，我們發現有兩組較好的實驗數據分別是(0.8,0.5,9)及(0.9,0.5,9)。而這兩組的數據雖然都相差不大，但在(0.8,0.5,9)這組數據中，其值大都在 60%之前大於(0.9,0.5,9)這組數據的值，如圖 5 所示。

這表示以(0.8,0.9,9)為參數在進行文件查詢時，所查詢到的相關文件都集中在前面出現。因為我們的系統是一文件相關度的大小來排列，所以排列在前面的文件，其相關程度一定最大，所以出現在愈前面且確認為相關的文件，表示所得到的查詢結果愈為精確。因此我

精確度

們在詞彙式剖析法中將 threshold1 設為 0.8，threshold2 設為 0.5，distance 設為 9。

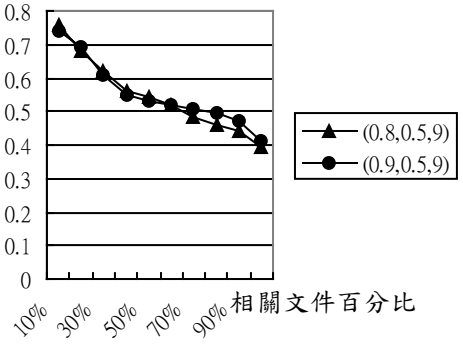

圖 5 數值比較折線圖

實驗二:

我們由詞彙群組式分析法與反查式搜尋剖析中各挑出五組實驗數據，來比對這兩種方法的數值變化，如表 2 所示。

由這五組實驗數據，我們發現在各組之間的數值及每一組實驗數據在各個百分比的數值變化上，反查式搜尋剖析都與詞彙群組式分析法的變化分佈大致上相同，如同圖 6 所示。這是因為反查式搜尋剖析是建構在詞彙群組式分析法之上，所以在各個數值的變化上大致上應該會詞彙群組式分析法相當。因此在反查式搜尋剖析上我們挑選與詞彙群組式分析法相同的參數設定。

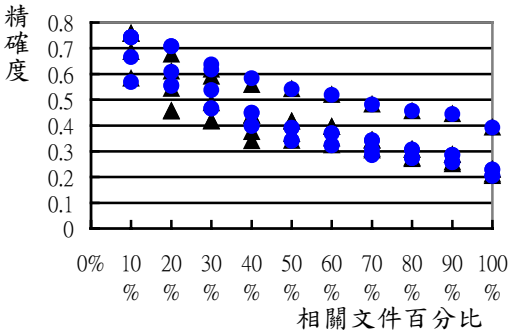

圖 6：詞彙群組式分析法與反查式搜尋剖析五組實驗的分佈情形
其中▲代表詞彙群組式分析法，●表示反查式搜尋剖析

實驗三:

在我們的段落式文件剖析法中需對一個參數進行設定，因此在這個實驗中，由我們所蒐集的 1200 篇文件中任選 10 篇文件進行查詢分析，把參數值的設定為 0.4 到 0.9 這六種情況，從這六組的實驗數據中找出較佳的參數設定，實驗結果數據如表 3 所示。評估表 3 的實驗數據，我們發現將參數設為 0.4 所得到的查詢精確率最高，如同圖 7 的數線分佈所示。

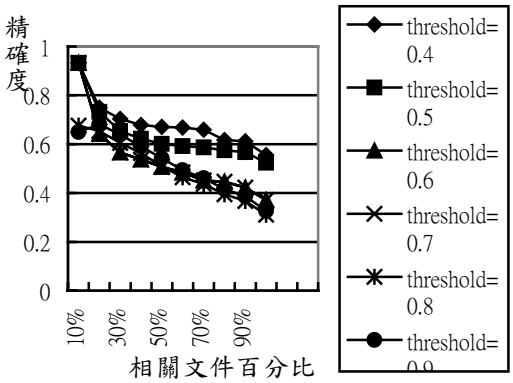

圖 7：段落式文件剖析法在不同參數下的數值比較折線

實驗四:

在這個實驗中將列出系統四種查詢相關文件方法的實驗數據比較，我們以之前三個實驗中所挑選出較好的參數設定值，所實驗出的數據來比較這四種方法的查詢效能。在第四個方法：文件段落比對分析，因為它的挑選關鍵詞彙的方式與方法三：段落式文件剖析法類似，所以在參數的設定上與方法三的參數設定值相同，一樣都設為 0.4。

表 4 列出詞彙群組式剖析法、反查式剖析法、段落式文件剖析法及文件段落比對分析這四種方法，在下達 20 篇查詢文件所得到的相關文件之平均精確度值。

我們可以從圖 8 中看出這四種方法的評估結果，發現以段落式文件剖析法的查詢效能最佳，文件段落比對分析最差，而詞彙群組式剖析法與反查式剖析法則是差不多。但是相關文件百分比在 50%時反查式剖析法的查詢效果比詞彙群組式剖析法還好，這表示雖然這兩個方法所找到的相關文件篇數都相同。但是反查式剖析法能夠將較為相關的文件排在前面，這使得使用者可以在閱讀較少的文件的情況下，找到自己所想要的文件。

因為段落式文件剖析法只將查詢文件作段落區分，並從各段落中萃取代表文件的關鍵詞彙，但對於被查詢文件並沒有進行段落區分，所以我們設計文件段落比對分析這個方法來作為段落式文件剖析法的對照比較。由實驗數據可以看見文件段落比對分的查詢效能並不是很好，這是因為單純以文件的段落以相似性比對來決定文件的相關聯性。可能會發生某一關鍵詞彙因不同用法而有不同的詞義，這會使得原本與查詢文件不相關的文件，因為包含有這一關鍵詞彙而列為極相關的文件。例如 ”

錦繡山河 ”這個關鍵詞彙可能會出現在地理類的文件中，也可能會出現在食譜的介紹的文件中。此外由於不同的作者雖然描述同一件事實，但是它們在文件中各段落的遣詞用句可能大不相同，所以單純以段落來進行比對極可能會因為這些因素而導致查詢效能的降低。

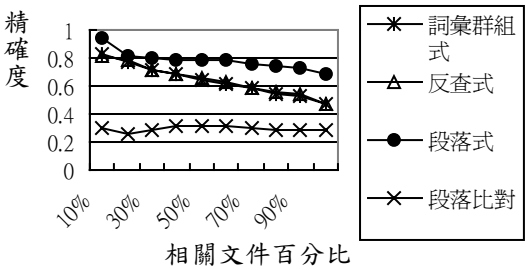

圖 8： 四種查詢方法，以 20 篇文件作查詢之效能比較折線圖

實驗五:

在這個實驗中主要為找出自動回饋機制對整個查詢系統效能的提升情形，由於要模擬使用者查詢的行為，且要由每一個查詢行為去偵測回饋資訊，並不是那麼的容易。因此我們改以另一種方式來進行實驗，在每一次查詢時中我們標定幾篇文件，在經由回饋之後，再一次的查詢使得這些標定的文件在相關度排序上提升與降低了多少，而以這兩個數值的差值來決定系統效能的提升與否。

我們在文件集中任選十篇文件作為查詢的文件，在獲得查詢結果後，考慮排序在前面的十篇文件。由前一個實驗的結果，發現以段落式文件剖析法查詢相關文件，相關文件大多出現在所查詢到的文件的前 40%。所以我們在挑選回饋上，前四篇文件不作為標定考慮，而以後面六篇文件作為回饋的文件。我們假設每一次以一篇標定的文件作為自動回饋機制的選擇，然後依回饋之後的再一次的查詢，來觀察該標定文件的排序位置是否有所提升或下降。如此每一篇查詢文件我們假設六次的回饋，考慮任選的十篇查詢文件在回饋之後的效能上有何改變。

我們經由表 5 中列出這十篇查詢文件的實驗結果，得到十篇文件的平均提升率為 0.416，而下降率則為 0.35，所以在整體上系統在使用自動回饋機制後，平均約讓查詢結果的效能提升了 6.6%。

伍、 結論

綜合以上的實驗，使用段落式文件剖析法來查詢相關文件，平均約比詞彙群組式剖析法提升了 14%。且對於查詢而言，所查詢到的相關文件大約都集中於文件排列的前面，對於查詢到不相關文件的比例也比其它方法來得少，因此所統計出的的精確度可以高於其它方法許多。

而與之作為實驗對照比較的文件段落比對分析法約比詞彙群組式剖析法降低了 34%。雖然同樣以段落為單位萃取關鍵詞彙，但是不同的是單純以段落的比對來決定相關程度，需要作比對的兩個段落內有相同的關鍵詞彙，且這些關鍵詞彙的 TFIDF 值還需高於設定的門檻值。這對於查詢文件與被查詢文件都以段落作為剖析，會造成計算每一個關鍵詞彙的 TFIDF 值低於門檻值，因此即使擁有相同的關鍵詞彙，卻因達不到門檻值而不列為比對統計。所以一些較為重要的關鍵詞彙可能會因此而被過濾掉或是 TFIDF 值過低，使得相關文件雖被查詢到，但排列在很後面的情形發生，也因此造成比對上的不精確。

除此之外以單純段落的比對，亦有可能會因其它不相關的文件段落因與查詢文件段落有共同的關鍵詞彙，且該關鍵詞彙所統計出的 TFIDF 值非常高，而使得不相關的文件被查詢出來，這也是文件段落比對分析法在查詢時所得到不相關的文件在平均上比使用其它方法多的原因之一。

對於反查式搜尋剖析法的查詢效能，僅高於詞彙群組式剖析法不到 1%的原因，可能是關鍵詞彙在查詢文件與被查詢文件內的分佈情形不同。也就是說查詢詞彙群組在查詢文件中是叢聚的情形，但是在被查詢文件中並不是呈叢聚的的分佈，所以再進行反向查詢詞彙群組叢聚在被查詢文件的比率時，效果並不是那麼的顯著。

而且對於詞彙群組式剖析法在挑選足以代表文件的詞彙群組時，有可能會摻雜一些不具代表性的詞彙群組，或是一些會模糊查詢主題的詞彙群組，使得在進行查詢時，具關聯性文件因這些雜訊而無法被排序到前面，或是有原本不相關的文件，因包含這些雜訊且權值所佔的比率極高，而被提升到排序的前面，而使得一些查詢的精確度沒有增加反而降低，因而造成整體的查詢效能沒有顯著地提升。

在自動回饋機制方面，由於我們基於時間與人力上的考量，無法全面模擬使用者的回饋情況。僅在下達查詢文件後，依其查詢結果標定回饋的文件，視所標定的回饋文件，再經一次的查詢後其排序的位置是否有所提升或下降，來瞭解回饋機制對於系統增進多少的效能，經過實驗五發現提升了 6.6%。雖然這個數值無法完全代表實際的查詢情況，但是由提升的數據顯示，我們所採用的這個回饋機制還是有增進一些查詢的效能。

參考文獻

[1] Ricaardo Baeza-Yates and Berthier Riberiro-Neto, *Modern Information Rertierval*, Addison- Wesley, 1999.

[2] Bernard J. Jansen, Amanda Spink and Tefko Saracevic, "Real life, real users, and real needs: a study and analysis of user queries on the web," *Information Processing and Management*, pp. 207-227, 2000.

[3] Chia-Hui Chang and Ching-Chi Hsu, "Enabling concept-based relevance feedback for information retrieval on the WWW," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 4, pp. 595-609, 1999.

[4] Chuan Chuan Lin, Shou Yi Tseng and Pei Min Chen, "A fuzzy document retrieval system based on concept network and cluster analysis," *Soochow Journal of Economics and Business*, pp. 39-60, 1999.

[5] Kurt Bollacker, Steve Lawrence and C.Lee Giles , "CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications," In Katia P. Sycara and Michael Wooldidridge, editors, *Proceedings of the Second International Conference on Autonomous Agents*, pp. 116-123, 1998.

[6] Kurt Bollacker, Steve Lawrence and C.Lee Giles, "A system for automatic personalized tracking of scientific literature on the web," In *Digital Libraries 99 –The Fourth ACM Conference on Digital Libraries*, pp. 105-113, 1999.

[7] Steve Lawrence , Kurt Bollacker and C. Lee Giles, "Indexing and Retrieval of Scientific Literature," *CIKM'99 of ACM* ,11/99 Kansas City, MO, USA, pp. 139-146, 1999.

[8] J.P. Yoon and Sungrim Kim, " Schema extraction for multimedia XML document retrieval," Web Information Systems Engineering, *Proceedings of the First International Conference*, pp. 113-120, 2000.

[9] Jinxi Xu and W. Burce Croft, "Improving the effectiveness of information retrieval with local context analysis," *ACM Transactions on Information Systems*, Vol. 18, No. 1, pp. 79-112, 2000.

[10] A. M. Tjoa, M. Hofferer, G. Ehrentraut and P. Untersmeyer, "Applying evolutionary algorithms to the problem of information filtering," *Database and expert systems applications,"Proceedings, Eighth International Workshop on 1997*, pp. 450-458, 1997.

[11] M. Mitra, A. Singhal and C. Buckey, "Improving automatic query expansion," *In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 206-214, 1998.

[12] C. Buckley, M. Mitra, J. Walz, and C. Cardie, "Using clustering and superconcepts within SMART," *In Proceedings of the 6th Text Retrieval Conference(TREC-6)*, E. Voorhess, Ed. pp. 107-124, 1998.

[13] C.J. van Risberrgen, "Information Retrieval," The URL of this paper can be found at http://www.dcs.gla.ac.uk/Keith/Preface.html

附錄：

表 1： 10 篇查詢文件在不同參數值下的平均精確度

(α,β,γ)	以 10 篇文件查詢的平均精確度									
	10%	***20%***	***30%***	***40%***	***50%***	***60%***	***70%***	***80%***	***90%***	***100%***
(0.8,0.5,1)	0.582	0.454	0.415	0.34	0.341	0.323	0.304	0.2735	0.2574	0.2046
(0.8,0.5,3)	0.686	0.456	0.417	0.374	0.367	0.349	0.31	0.27	0.25	0.21
(0.8,0.5,5)	0.757	0.542	0.487	0.438	0.416	0.394	0.342	0.3055	0.2858	0.2292
(0.8,0.5,7)	0.757	0.61	0.59	0.438	0.416	0.394	0.342	0.3055	0.2858	0.2292
(0.8,0.5,9)	0.757	0.675	0.617	0.558	0.541	0.519	0.482	0.4555	0.4438	0.3922
(0.8,0.6,3)	0.538	0.454	0.415	0.373	0.366	0.348	0.305	0.2737	0.2576	0.2047
(0.8,0.6,5)	0.657	0.542	0.487	0.438	0.416	0.389	0.339	0.3035	0.2838	0.2282
(0.8,0.6,7)	0.657	0.508	0.487	0.438	0.416	0.394	0.342	0.3055	0.2858	0.2292
(0.8,0.6,9)	0.657	0.608	0.537	0.472	0.441	0.419	0.382	0.3555	0.3438	0.2922
(0.9,0.5,3)	0.642	0.492	0.424	0.427	0.417	0.401	0.3646	0.3435	0.3169	0.2523
(0.9,0.5,5)	0.742	0.492	0.457	0.417	0.417	0.386	0.3596	0.3375	0.3119	0.2473
(0.9,0.5,7)	0.742	0.492	0.457	0.417	0.407	0.394	0.3666	0.3435	0.3159	0.2503
(0.9,0.5,9)	0.742	0.692	0.607	0.551	0.532	0.519	0.5066	0.4935	0.4739	0.4133
(0.9,0.6,3)	0.557	0.492	0.424	0.427	0.417	0.401	0.364	0.344	0.317	0.252
(0.9,0.6,5)	0.657	0.492	0.424	0.417	0.417	0.401	0.359	0.338	0.312	0.247
(0.9,0.6,7)	0.657	0.492	0.457	0.417	0.407	0.394	0.366	0.344	0.316	0.25
(0.9,0.6,9)	0.657	0.592	0.507	0.451	0.432	0.419	0.406	0.394	0.374	0.313
(0.9,0.7,3)	0.557	0.482	0.416	0.407	0.405	0.375	0.336	0.317	0.292	0.232
(0.9,0.7,5)	0.607	0.492	0.448	0.398	0.384	0.364	0.318	0.286	0.268	0.213
(0.9,0.7,7)	0.59	0.508	0.451	0.408	0.392	0.37	0.323	0.291	0.272	0.217
(0.9,0.7,9)	0.59	0.608	0.501	0.442	0.417	0.395	0.363	0.41	0.33	0.28

表 2：詞彙群組式分析法與反查式搜尋剖析平均精確度比對

	以 10 篇文件查詢的平均精確度(α=0.9, β=0.5, γ=1)									
	10%	***20%***	***30%***	***40%***	***50%***	***60%***	***70%***	***80%***	***90%***	***100%***
詞彙群組	0.582	0.454	0.415	0.34	0.341	0.323	0.304	0.2735	0.2574	0.2046
反查式	0.568	0.554	0.465	0.398	0.3048	0.323	0.2844	0.274	0.257	0.205

(a) 第一組實驗數據

	以 10 篇文件查詢的平均精確度(α=0.9, β=0.5, γ=3)									
	10%	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*	*80%*	*90%*	*100%*
詞彙群組	0.686	0.456	0.417	0.374	0.367	0.349	0.31	0.27	0.25	0.21
反查式	0.666	0.556	0.4666	0.3992	0.3417	0.3238	0.3051	0.274	0.258	0.205

(b) 第二組實驗數據

	以 10 篇文件查詢的平均精確度(α=0.9, β=0.5, γ=5)									
	10%	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*	*80%*	*90%*	*100%*
詞彙群組	0.757	0.542	0.487	0.438	0.416	0.394	0.342	0.3055	0.2858	0.2292
反查式	0.743	0.608	0.537	0.449	0.391	0.369	0.342	0.306	0.0.286	0.229

(c) 第三組實驗數據

	以 10 篇文件查詢的平均精確度(α=0.9, β=0.5, γ=7)									
	10%	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*	*80%*	*90%*	*100%*
詞彙群組	0.757	0.61	0.59	0.438	0.416	0.394	0.342	0.3055	0.2858	0.2292
反查式	0.743	0.708	0.637	0.449	0.391	0.369	0.342	0.306	0.0.286	0.229

(d) 第四組實驗數據

	以 10 篇文件查詢的平均精確度(α=0.9, β=0.5, γ=9)									
	10%	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*	*80%*	*90%*	*100%*
詞彙群組	0.757	0.675	0.617	0.558	0.541	0.519	0.482	0.4555	0.4438	0.3922
反查式	0.743	0.708	0.617	0.583	0.541	0.519	0.482	0.456	0.444	0.392

(e) 第五組實驗數據

表 3 段落式文件剖析法在不同參數下的平均精確度

threshold	以 10 篇文件查詢的平均精確度									
	10%	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*	*80%*	*90%*	*100%*
0.4	0.993	0.749	0.704	0.678	0.671	0.6685	0.659	0.617	0.611	0.555
0.5	0.993	0.732	0.654	0.622	0.6013	0.5915	0.587	0.575	0.567	0.523
0.6	0.933	0.642	0.566	0.537	0.507	0.4845	0.452	0.446	0.423	0.371
0.7	0.933	0.642	0.566	0.537	0.507	0.4845	0.452	0.446	0.423	0.371
0.8	0.676	0.654	0.605	0.5634	0.5091	0.4648	0.4372	0.395	0.368	0.311
0.9	0.65	0.381	0.639	0.597	0.542	0.4935	0.4612	0.416	0.387	0.327

表 4： 四種查詢方法在下達 20 篇查詢文件所得到的平均精確度

	以 20 篇文件查詢的平均精確度									
分析方法	***10%***	***20%***	***30%***	***40%***	***50%***	***60%***	***70%***	***80%***	***90%***	***100%***
詞彙群組式	0.83 (+0)	0.77 (+0)	0.72 (+0)	0.68 (+0)	0.65 (+0)	0.62 (+0)	0.58 (+0)	0.55 (+0)	0.53 (+0)	0.47 (+0)
反查式	0.82 (-0.01)	0.78 (+0.01)	0.72 (+0)	0.69 (+0.01)	0.66 (+0.01)	0.63 (+0.01)	0.59 (+0.01)	0.56 (+0.01)	0.54 (+0.01)	0.47 (+0)
段落式	0.95 (+0.12)	0.82 (+0.05)	0.8 (+0.08)	0.78 (+0.1)	0.78 (+0.13)	0.78 (+0.16)	0.76 (+0.18)	0.74 (+0.19)	0.73 (+0.2)	0.69 (+0.22)
段落比對	0.3 (-0.53)	0.26 (-0.51)	0.28 (-0.44)	0.32 (-0.36)	0.31 (-0.34)	0.31 (-0.31)	0.3 (-0.28)	0.29 (-0.26)	0.29 (-0.24)	0.28 (-0.19)

表 5： 以 10 篇查詢文件作自動回饋之實驗結果

	10 篇查詢文件各以六次回饋所得之查詢比較	
文件編號	***提升率***	***下降率***
1187	0.83	0
1236	0.17	0.67
1386	0.33	0.17
1456	0.17	0.5
1521	0.5	0.5
1621	0.67	0.17
1723	0	0.333
1725	0.33	0.5
1831	1	0
1921	0.17	0.67
平均	***0.416***	***0.35***