以 SEO 概念維度為基礎的 Google 搜尋結果之分群

陸承志　　黃挺立　　林昱呈
元智大學資訊管理學系
imcjluh@saturn.yzu.edu.tw, {s1006207, s996242}@mail.yzu.edu.tw

摘要

本研究探討如何利用潛在語意分析 (Latent Semantic Analysis) 來找出 Google 搜尋結果前 20 筆到 50 筆的相關詞，並且依照語意分析產生的概念維度來將相關詞做分群，之後再利用每一筆網頁所包含的相關詞來將搜尋結果進行分群。本研究提出的方法會依據概念維度門檻為每一個查詢關鍵詞的搜尋結果判斷出適合的分群群數，而不用事先決定。搜尋結果中的網頁依據其所包含的相關詞在各相關詞群集的概念維度分數，分配到一個主要的文件群集或是分配到多個文件群集。最後，本研究使用 Silhouette Coefficient 來評估我們提出的文件分群方法的效能，並且與其他分群系統作比較。

關鍵詞：搜尋結果分群、文件分群、潛在語意分析、語意相關詞、搜尋引擎最佳化。

Abstract

This study proposed a Latent Semantic Analysis based method to find semantically related terms from top 20 to 50 Google search results for a given query and to group the terms into clusters. Each item of the search results is then grouped into one individual cluster based on the terms it contains. A heuristic method is proposed to conduct clustering of semantically related terms based on their concept dimension significance after LSA analysis. The proposed method determines the best fit number of clusters for each query, without the burden of defining the number of clusters in advance. Web pages containing multiple terms is assigned to a primary cluster or allocated them into multiple clusters based on concept dimensionality significance. Finally, the clustering quality is evaluated using silhouette coefficient on experiment results using a set of mixed popular and industrial keywords. The clustering quality of the proposed method is also compared with popular clustering methods.

Keywords: Search Results Clustering, Latent Semantic Analysis, Semantically Related Terms, Document Clustering， Search Engine Optimization

1. 前言

一般的搜尋引擎如 Google 或 Yahoo，所呈現的搜尋結果是依據使用者所下達的特定查詢關鍵字而來，但是這些特定查詢關鍵字所表達的主題概念不一定唯一，例如查詢關鍵字「 apple 」可能是農產品的水果，或是 3C 產品的蘋果公司，但搜尋引擎呈現的條列式結果往往會呈現一個主要的主題概念，例如在「 apple 」搜尋結果的前面幾筆幾乎都跟蘋果公司有關，這對想要了解蘋果公司的使用者之幫助很大，但對想要搜尋水果的使用者卻造成不方便。其他如「 snow leopard 」這個關鍵詞，Google 搜尋結果的前面幾筆包括 Mac OS 作業系統、野生動物、非公益保護動物組織等。

許多搜尋引擎使用者行為分析研究指出，大部分使用者會依據搜尋結果的順序跳著看；Google 的專利文件也指出，它們是把跟查詢關鍵詞相關的多個概念之文件綜合呈現，並且依照相關性決定呈現的數量與排序的順位。如果有一個機制可以把搜尋結果所呈現的概念萃取出來，並依這些概念來分群文件，對使用者搜尋精確合宜的文件有很大的幫助，同時能提升搜尋的效率。

本研究假設搜尋結果包含幾個主題概念，我們希望能清楚呈現每個概念由那些相關詞構成，再依相關詞的重要性將文件分群，這樣對於瞭解搜尋結果組成的人有很大的幫助。對於一個想要做搜尋引擎最佳化 (SEO) 的人來說，透過這個機制可以了解搜尋引擎在特定關鍵詞的搜尋結果偏好哪些主題概念，想要優化的網頁該具備哪些相關詞，才有機會獲得搜尋引擎的青睞；還有透過這個機制，網路行銷人員也可以了解在特定關鍵詞的搜尋結果中，搜尋引擎偏好的是哪一個概念或者哪一類型網站，進而判斷 SEO 成功的可能性。

本研究在查詢關鍵詞有多重主題概念的假設下，探討如何找出 Google 搜尋結果的相關詞，並且推算出 Google 在不同查詢關鍵字下的相關詞群聚概念所設定的分群數量，然後將相關詞適當分群。本研究提出的分群方法，能依據概念維度門檻為每一個查詢關鍵詞的搜尋結果判斷出適合的分群群數，而不用事先決定。接著，我們依據搜尋結果中網頁所包含的相關詞，將網頁分配到一個主要的文件群集或是分配到多個文件群集。

2. 文獻探討

根據許多眼球追蹤的研究 [1], [2], [3] 顯示，使用者會依照 Google 建議的排名順序來瀏覽網頁，即使內容可能不怎麼相關。所以網頁的瀏覽流量會隨著網站排名而遞減。

Carpineto et al. [4] 認為使用者從搜尋結果的前端按照順序閱讀下來，適合來找尋特定組織的首頁，但若是使用不同主題或意思的關鍵詞，搜尋結果通常會混合許多資訊在其中，使用者需要過濾大量不相關的資訊才能找到想要的資料。這個觀點也跟 Google 專利文件 [5] 提到的概念雷同，該文件指出搜尋結果若包含多個概念的文件，搜尋引擎會依照這些文件對查詢關鍵詞的相關性決定呈現的數量與排序順位，然後把這些文件綜合呈現。

提供搜尋結果分群的系統，一般稱為分群式搜尋引擎 [4], [6]。這類系統通常會包含四大步驟：(1)取得搜尋結果資料、(2)資料前置處理、(3)分群結構與(4)分群結果呈現。由於搜尋結果分群系統需要在短時間內完成有效的分群，通常會採用分割式分群法的 K-means 或者線性時間的 Suffix Tree Clustering (STC) 演算法。K-means 的基本步驟就是需先設定要分成 k 個群集，並隨機找出 k 個點來做為初始點，然後重複的尋找，直到得出 k 個點滿足停止的條件。對於不同查詢關鍵詞如何訂出適當的群數是使用 K-means 的最大挑戰。Suffix Tree Clustering (STC) [7]是一個利用文件中共同出現的詞彙的線性分群演算法，主要有下面幾個步驟：(1) 文件整理；(2) 建立基本群集； (3) 整合基本群集。STC 屬於遞增式、線性時間的演算法，受到許多研究的青睞; 但是 STC 產生的群集標題經常是片段的，而非適當的片語，而且尋找基本群集的複雜度會隨著文件字數的多寡而提升，因此後續有許多研究紛紛提出改善合併的過程。

Lingo 演算法[8], [9]的概念是先定義出群集的描述標題，然後根據各文件和這些描述標題的相似度，將超過預設門檻的文件指定到對應的群集當中。Lingo 演算法包含五個主要階段：(1). 資料前處理：這個階段透過 stemming 與 stop-word 處理，來移除網頁描述可能會影響分群品質的字或詞。(2).特徵詞挑選： 此階段藉由改良的 suffix-array，並且透過一些條件的篩選，例如出現的文件數、必須為完整片語、開頭結尾不能為停用詞等等，來找出有語意的常用詞彙或片語。(3).群組標題挑選： Lingo 使用奇異值分解 (Single Value Decomposition, SVD) 將使用單字詞建立的詞文矩陣 A 分解成三個矩陣 $A=USV^T$。 Lingo 認為 U 的每一個欄代表資料集的一個抽象概念。接著，再將縮減維度的 Uk 和使用常用詞彙建立的新詞文矩陣 P 做矩陣相乘，得到的矩陣每一欄的詞彙就來當作各個群集的標題。(4). 群集內容分配： Lingo 把每一個文件跟所有群集的標題做相似度計算， 然後將相似度分數大於一個預設文件指派門檻的文件分配可能的群集中。所以一個文件可能被分配到多個群集，而非單一。(5). 群集形成：最後步驟會將各群集的標題概念分數與其群組內文件數量進行乘積，來當作各群集的分數，在依照各群集的分數，從大到小依序顯示群集。

在利用最小成本擴張樹的研究中， Mecca et al. [10]利用奇異值分解 (Single Value Decomposition, SVD) 將詞文矩陣（Term-document matrix）轉化成「概念-文件矩陣」，然後將其中把有相似的概念的文件歸類成一個群。為了找出最適合的群數，該研究將可能的群數由小到大，把「概念-文件矩陣」分別轉換成最小成本擴張樹，再計算最小成本擴張樹的品質函數來找出最適合的群數。這個方法可以避免事先定義群集個數的困擾，但時間複雜度遠比其他方法來得高。

潛在語意分析 (Latent Semantic Analysis, LSA) 是用來分析文件與文件之間和文件所包含的詞彙之間關係的一個技術。它的基本做法是利用奇異值分解將詞文矩陣 A 分解成 U、S、V^T 三個矩陣，其中 U 代表詞的矩陣；V^T 代表文件的矩陣；矩陣 S 為一個 r 維的奇異值矩陣。LSA 將 S 矩陣的 r 個維度中挑選與保留出前面較重要的 k 個維度，以刪除雜訊維度。Wall et al. [11] 利用每個奇異值的 relative variance，亦即每個奇異值的平方除以所有奇異值的平方和，來決定每一個維度的相對重要性，並且利用前 k 個累積的相對變異數 (cumulated relative variance, CRV) 來顯示前 k 個維度的貢獻度。當到達某個 k 值的 CRV 值大於預設門檻值時，這個 k 就是我們的目標 k 值。LSA 利用三個縮減過的矩陣 U_k、S_k、V_k^T 之乘積來重建縮減過的矩陣 A_k,這時詞彙與文件之間的潛在語意就能出現。

分群結果評估可分為內部型 (internal) 評估與外部型 (external) 評估 [4], [12], [13]。內部型方法，例如 Lee & Kageura [14]提出的空間密度比 (Space Density Ratio) 即是計算了內部相似度與外部相似度的比值來評估分群的品質。Silhouette Coefficient (SC) [15]結合群組內部的內聚力 (Cohesion)和群與群之間的區別性 (Separation)，來進行分群結果的評估。如果每個物件都分配到適當的群集，那麼內聚力和區別性兩者都會高， SC 數值也會高。外部型則是使用一組類別當作基準來判斷分群結果是否有相符合。對搜尋結果來說，要找到一組可做基準的類別基準線並不容易，有許多研究採用 AMBIENT[1] (AMBIguous ENTities) 這個資料庫包含多個模糊查詢關鍵詞的 Yahoo 搜尋結果， 並非來自主流的 Google。人工評估則是使用 5 級量表，經由人工評估來衡量相關性 (coherence) 與有效性 (usefulness) [13]。Geraci et al. [16] 指出，評估人員需要回答下列三個問題： (1) 標題是否明確？ (2) 可否藉由標題猜出內容為何？ (3) 標題與內容是否相符？ 最後再將答案整理並評估。

[1] http://credo.fub.it/ambient/

3. 分群方法

本研究的架構如圖 1 所示，共分為四個步驟：(1) 網頁抓取與資料剖析；(2) LSA 分析；(3) 相關詞分群與挑選群集標題；(4) 文件分群。以下解釋相關詞分群與文件分群的流程。

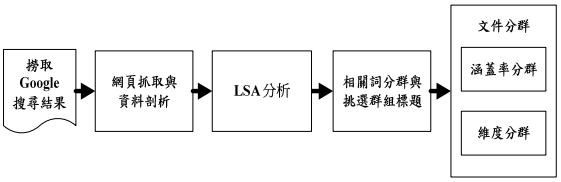

圖 1 分群流程圖

3.1 相關詞分群

我們利用經過 SVD 運算，縮減過維度的 U_k 來做相關詞的分群。U_k 是一個詞彙 - 概念矩陣（Term-concept Matrix），它的每一列代表一個相關詞，而每一行代表著不同概念維度。原則上，只要一個相關詞的概念維度在某個維度具有顯著的區別性，那麼這個相關詞就被分配到該維度所對應的相關詞群集中。首先，我們計算各個概念維度的所有元素之平均值、標準差與平均值+標準差三個數值。分配的流程如下：

1. 將第一維度中所有概念維度分數大於 Mean + SD 的相關詞挑選出來。
2. 將第一維度中其餘相關詞的概念維度分數與 Mean 比較，如果分數大於 Mean 且為所有維度當中分數最高的，就將它挑出來。
3. 接著，我們以相同方法將剩餘的詞依照下一個維度來挑選，以此類推。若維度數目越多，群集數也有可能會增加，但群集數必定不超過維度數，亦即有些詞無法分配到適當群集，我們就將最後剩餘的詞歸納成一個「其他」群集。表 1 為查詢關鍵詞 apple 的相關詞分群之結果：

在挑選完群集數與各群集內的相關詞之後，我們使用代表性 (Representativeness) 計算公式來找出一個簡單明瞭、可以描述群集的標題，而此標題不可與查詢關鍵詞相同。

表 1 依概念維度的相關詞分群範例

CT1	CT2	CT3
computer	nasdaq	encyclopedia
os	aapl	wikipedia
news	stock	
ipad	aapl stock	
mac	finance	
iphone		
apple		
ipod		
design		

我們使用的代表性計算公式，考量一個相關詞對於一個群集的重要程度 (涵蓋率) 以及這個詞對於不同群集的區別程度 (區別率)，亦即詞 t 在群集 *Ci* 的代表性 $R(t, Ci)$ 是一個包含涵蓋率和區別率的綜合指標，如公式 (1)：

$$R(t,C_i)=\alpha\times V(t,C_i)+(1-\alpha)\times D(t,C_i) \qquad (1)$$

其中 $V(t, Ci)$ 為詞 t 在群集 *Ci* 的加權涵蓋率 (coVerage)；$D(t, Ci)$ 為詞 t 在群集 *Ci* 的區別率；α 值代表涵蓋率與區別率之間的比重。簡言之相關詞 *t* 在群集 *Ci* 裡出現的文件數愈多，它的涵蓋率愈高；同時這些文件的排名愈好，它的加權愈大。此外，包含詞 t 的文件若其包含的所有相關詞愈集中於群集 C_i，那麼詞 t 的區別率 $D(t, Ci)$ 就愈高。詳細的定義，請參閱 [17]。

最後，我們會取每個群集中代表性數值的前 30% 來當成各個群集的標題。以表 1 的範例來看，三個群的標題分別為「news、ipad、iphone」，「aapl、finance」，和「wikipedia」。

3.2 概念維度文件分群

從上述相關詞的分群結果中，我們發現一個文件可能包含分布在多個群集的相關詞，我們利用這些已分群的相關詞，進一步的將包含這些相關詞的文件進行分群。分群的概念是每個相關詞群集(CT, Cluster of Terms) 有一個對應的文件群集(CD, Cluster of documents)，當一個文件對某個相關詞群集有顯著性時，我們就將文件分配到相對應的文件群集裡。

以圖 2 為例，CT1 內有 t_1、t_2、t_3、t_4 四個相關詞，CT2 有 t_5、t_6、t_7 三個相關詞。文件 d_1 包含 t_1、t_2、t_3、t_5，文件 d_2 包含 t_4、t_6、t_7，因為 d1 對 CT1 有顯著性，d_1 就可分配到 CT1 對應的文件群集 CD1；同樣地，d_2 對 CT2 有顯著性，我們將 d2 分配到 CD2。這裡我們考量文件的硬式分群 (Hard Clustering) 以及軟式分群 (Soft Clustering) 兩種方式。硬式分群允許一個文件只分到單一的文件群集，這個主要群集就是文件所隸屬的唯一群集；軟式分群允許一個文件分到多個文件群集，我們透過文件的概念維度來衡量文件對哪些群集有顯著性，將之分配到多個文件群集。

概念維度分群是利用每一個文件所包含的相關詞來計算該文件對應於每個相關詞群集的概念維度分數。由於每一個相關詞在各自的相關詞群集中都有一個維度分數，一個文件在某個相關詞群集所包含相關詞的概念維度分數之總和就是這個文件在那個相關詞群集的概念維度分數。

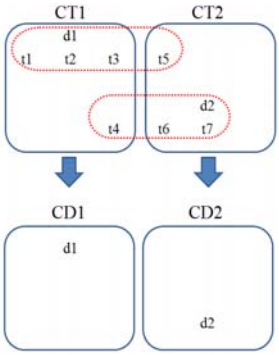

圖 2 基於相關詞群集的文件分群示意圖

以表 2 為例，文件 1 包含的相關詞為「webcam」與「camera」，它們在相關詞群集 CT1 的概念維度分數分別是 0.357 與 0.379，所以文件 1 在相關詞群集 CT1 的概念維度分數為 0.357 + 0.379 = 0.736；而文件 1 並沒有包含相關詞群集 CT2 的任何相關詞，所以文件 1 在相關詞群集 CT2 的概念維度分數為 0。另以文件 3 為例，包含的相關詞為「webcam」、「camera」與「world」，即文件 3 在相關詞群集 CT1 的概念維度分數為 0.357 + 0.379 = 0.736；文件 3 在相關詞群集 CT2 的概念維度分數為 0.755。在文件硬式分群時，我們選擇概念維度分數最高的相關詞群集所對應的文件群集做為該文件的主要群集。

表 2 文件的概念維度分布範例

群集	相關詞	出現文件	概念維度
CT1	webcam	1, 2, 3	0.357
	camera	1, 3	0.379
CT2	people	4, 5	0.623
	world	3, 4, 5	0.755

在軟式分群部分，我們使用所有文件概念維度的平均值來做為一個分群指派門檻。以表 1 的相關詞為例，CT1 的概念維度平均值為 0.366，所以文件 1-3 都會分到 CT1 所對應的文件群組 CD1；CT2 的概念維度平均值為 0.702，所以文件 3 - 5 都會分到 CT2 所對應的文件群組 CD2，如表 3。

表 3 依概念維度的文件分群範例

文件	概念維度		文件群集
	CT1	CT2	
1	0.736	0	**CD1**
2	0.357	0	**CD1**
3	0.736	0.755	*CD1*, **CD2**
4	0	1.378	**CD2**
5	0	1.378	**CD2**
平均值	0.366	0.702	

註：粗體代表文件主要群集，斜體代表次要群集

4. 實驗與評估

本實驗的變數包含文件分群方式(硬式分群與軟式分群) 和 Google 擷取的文件數量等。我們共挑選 100 個關鍵詞來進行實驗， 主要來源有專業的工業關鍵詞 50 個，涵蓋電子、電機、機器、橡膠等；另一種為 Google Zeitgeist 2012[2] 網站中，從搜尋關鍵字、人物、電影電視、消費性電子產品等主題當中挑選的熱門關鍵詞 50 個。

4.1 分群結果評估方法

本研究使用 Sihouette Coefficient [15]來評估分群的有效性。假設 $\pounds = \{C_{1,\ldots,}C_k\}$ 為將 n 個物件分成 k 個群集的分群結果。一個物件 $o \in C_j$ 的 Silhouette 則定義為公式 (2)：

$$s(o\pounds) = \frac{b(o,\mathcal{T}) - a(o,\ )}{\max\{b(o,\mathcal{T}), a(o,\ )\}} \qquad (2)$$

其中 $a(o\pounds)$ 為物件 o 到它自己的群組 C_i 的距離，$b(o\pounds)$ 為物件 o 到其他最接近的一個群組 C_j 的距離。我們參考 IBM[3] 的建議，將 物件 o 到群組 C_i 的距離改為物件 o 到 C_i 的 Centroid 中心的距離，以降低計算量。整體的 Silhouette Coefficient 則為所有物件 SC 值的平均值。

Silhouette Coefficient 的數值結果會落在 -1 與 +1 之間，越接近 +1 代表元素有被分配到適當的群集，反之越接近 -1 代表並沒有分配到適當的群集。在評估相關詞分群結果時，我們使用 LSA 縮減過維度的 U_k 詞的向量空間矩陣來當作相關詞的向量；而評估文件分群結果時，則使用 V_k^T 文件的向量空間矩陣當作文件的向量。

4.2 分群方法之調整

本研究選擇 Carrot2 分群系統的 version 3.7.1 workbench[4] 版本來做評估比較。Carrot2 分群系統內建許多不同的分群演算法，我們挑選出 Lingo、K-Means 與 Suffix Tree Clustering (STC) 三個不同分群演算法來進行比較。其中 Lingo 分群演算法的設計是以軟式分群為主要目標，所以與我們的文件軟式分群可以輕易的做比較，但是若要做硬式分群比較時，則需要做些額外的處理。Lingo 在做分群處理時，演算法會計算出每個群集的一個相似度分數，此相似度分數越高代表群集內文件之間的相似度越高，所以我們為 Lingo 軟式分群後的每一個文

[2] http://www.google.com/zeitgeist/2012/#the-world

[3] http://publib.boulder.ibm.com/infocenter/spssstat/v20r0m0/index.jsp?topic=/com.ibm.spss.statistics.help/alg_cluster-evaluation.htm

[4] http://project.carrot2.org/download.html

件，選擇其所在的群集中相似度分數最高的那個群集為其主要群集，就可以讓分群結果從軟式分群變為硬式分群結果，以利進行硬式分群的比較。

本研究把提出的分群方法命名為硬式概念維度分群 (Hard Clustering by Conceptual Dimensionality, HC-CD)，以及軟式概念維度分群 (Soft Clustering by Conceptual Dimensionality, SC-CD)，其中，若有包含關鍵詞則為 HC-CDQ 和 SC-CDQ，不包含關鍵詞則為 HC-CDN 和 SC-CDN。另外 K-Means 演算法適合硬式分群，STC 演算法適合軟式分群。分群演算法比較綜合整理如表 4。實驗時，我們一樣會使用矩陣 V_k^T 中的文件向量當作所有演算法的文件向量，其中 K-Means 演算法的 K 值則採用我們執行 LSA 所得到的 K 值， 也就是預設 K-means 跟我們的分群法相同的群數。。

表 4 分群演算法比較

方式	演算法			
硬式分群	HC-CDQ	HC-CDN	Lingo	K-Means
軟式分群	SC-CDQ	SC-CDN	Lingo	STC

4.3 實驗結果

搜尋引擎的使用者行為顯示，一般使用者大多只看搜尋結果的第一頁，而且不會看超過三頁。所所以搜尋結果的前 30 筆網頁是我們關注的重點。但我們發現，若只用前 10 筆搜尋結果，不僅相關詞數量很少，而且概念維度也經常低於兩個維度，造成分析上的困難。同時， 我們想要了解超過 30 筆之後，是否也有重要的概念維度出現，因此我們實驗的搜尋筆數範圍為 20 – 50。

4.3.1 文件硬式分群

圖 3 為擷取 20 - 50 篇網頁的實驗數據。實驗結果顯示使用概念維度分群並且包含關鍵詞本身 (HC-CDQ) 的表現最好，其次為 Lingo， 最差為 K-means。

從當中可以明顯看得出來，擷取 30 篇文件以上的平均值整體來說, 都比 20 篇來的好，而 30 篇、40 篇與 50 篇之間沒有明顯的差異，顯示擷取 30 篇所做的分群結果會較好而且也較穩定。

4.3.2 文件軟式分群

圖 4 為擷取 20-50 篇網頁，使用軟式分群的實驗結果。實驗結果顯示，當我們改為軟式分群時，因為查詢關鍵詞同時出現在很多文件，可看出使用維度分群並且不包含關鍵詞本身的結果 (SC-CDN) 的表現會比包含關鍵詞的效果來的好。而且當中可以看出，使用 30 篇文件以上的平均值整體來說, 都比 20 篇來的好，而 30 篇、40 篇與 50 篇之間沒有明顯的差異，結果與硬式分群相似。

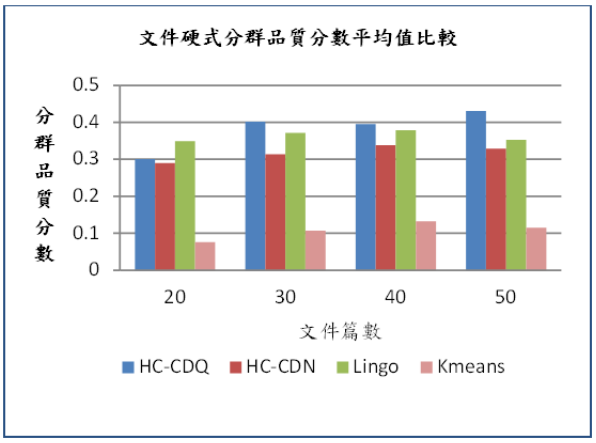

圖 3 文件硬式分群品質分數平均值比較

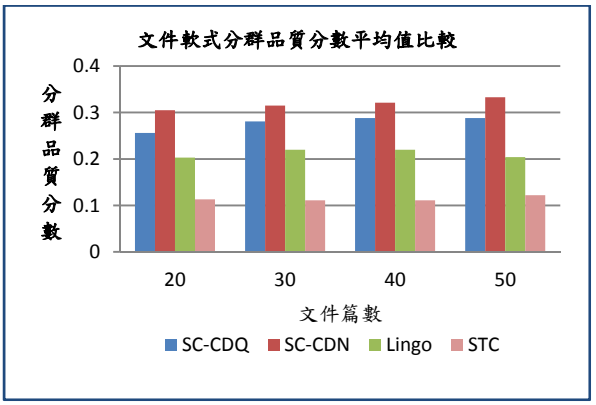

圖 4 文件軟式分群品質分數平均值比較

4.3.3 實驗討論

硬式分群的實驗結果顯示，在 30 到 50 筆文件的情況下，使用概念維度分群並且包含查詢關鍵詞（HC-CDQ）表現最好，其次是 Lingo 與概念維度分群並且不包含查詢關鍵詞（HC-CDN），最差的為 K-Means；而只有在 20 篇的情況下 Lingo 表現的比使用概念維度分群並且包含查詢關鍵詞來得好。文件軟式分群效果最好的反而是不包含查詢關鍵詞本身的概念維度分群（SC-CDN），其次為 Lingo、包含查詢關鍵詞本身的概念維度分群 (SC-CDQ) 與最後的 STC。由於大部分的文件都會包含查詢關鍵詞，所以當我們使用概念維度文件分群時，大部分的文件被分配到第一群集機率變高，造成第一群集過大，而且群集之間的區別性變低，分群品質下降。如果我們不考慮查詢關鍵詞本身，各文件就比較平均的分配於各群組當中，分群效果也會相對的比較好。

以文件數量來看，擷取 30 到 50 筆網頁的分群品質會比擷取 20 篇網頁來的好，而 30 到 50 筆網頁之間則沒有明顯差異。我們統計所有關鍵詞分群結果的分群數量，當擷取 30 篇網頁時，平均會分成 7.1 個群集；當擷取 20 篇網頁時，平均會將分成 4.5 個群集，簡言之，當我們多擷取第 21-30 篇網頁時，平均會增加 2.6 個群集， 也就是各個網頁要找到與他相似度較高的機會就會增加，因此分群結果會較好。

5. 結論

本研究利用潛在語意分析來找出 Google 搜尋結果前 20 到 50 筆網頁的相關詞，並且依照語意分析產生的概念維度來將相關詞以及文件分群，最後我們使用 Silhouette Coefficient，來對我們的文件分群結果做效能評估。

從實驗結果顯示，我們得到下面結論：

- 分群方法比較：硬式分群方面，抓取 30 到 50 筆網頁時，我們提出的使用概念維度分群並且包含查詢關鍵詞本身（HC-CDQ）表現最好；抓取 20 篇網頁時，則為 Lingo 結果較好；在文件軟式分群方面，不論抓取篇數多少，皆是使用概念維度分群並且不包含關鍵詞本身（SC-CDN）最好。
- 文件數量比較：我們提出的分群方法，在抓取 30 到 50 筆網頁時比抓取 20 篇網頁的分群結果較好，而抓取 30 到 50 筆文件之間則沒有明顯差異。其他的方法也有類似的情況，除了 STC 在 30 與 40 篇些微下降。

目前分群系統只按照相關詞的出現與否來決定文件與文件之間的語意，未來我們希望能利用文件之間更多的語意來做分群的依據，例如文件 A 與 B 都同時引用文件 C，那麼文件 C 就可視為跟 A 與 B 在同一個群集。或者來自同一類網站的文件內容，可以分配到同一個群組當中，例如社交網站、部落格等。此外，搜尋結果的第一頁已經加入許多的多媒體資料，例如圖片、影片、地圖、新聞與 SiteLink 等，如何有效的利用語意將這些不同型態的資訊做分群，也是未來需要擴展的方向之一。

誌謝

我們感謝審查委員的建設性回饋，讓本文的內容與呈現更臻完善。

參考文獻

[1] Joachims, T., L. Granka, B. Pan, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '05). New York, USA, pp. 154-161. August 15–19, 2005,

[2] Pan, B., Hembrooke, H., and Joachims, T. "In Google We Trust: Users' Decisions on Rank, Position, and Relevance," *Journal of Computer-Mediated Communication*, Vol. 12, pp. 801–823, 2007. doi: 10.1111/j.1083-6101.2007.00351.x

[3] Guan, Z. and Cutrell E. "An eye tracking study of the effect of target rank on web search." *Proceedings of the SIGCHI conference on Human factors in computing systems,* pp. 417-420, 2007.

[4] Carpineto, C., Osinski, S., Romano, G. and Weiss, D., "A Survey of Web Clustering Engines," *ACM Computing Surveys*, Vol. 41, No. 3, pp. 17:1-17:38, 2009.

[5] Patterson, A. L., "Phrase-based indexing in an information retrieval system," *U.S. Patent No.*7,536,408 B2, May 19, 2009.

[6] Cobos, C., et al. "Fitness Function Obtained from a Genetic Programming Approach for Web Document Clustering Using Evolutionary Algorithms," *J. Pavón et al. (Eds.): IBERAMIA 2012, LNAI 7637,* pp. 179–188.

[7] Zamir, O. and Etzioni, O. "Web Document Clustering: A Feasibility Demonstration," *SIGIR 98*, pp. 46-54, 1998.

[8] Osinski, S., Stefanowski, J. and Weiss, D., "Lingo: Search results clustering algorithm based on Singular Value Decomposition," Advances in Soft Computing, Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM´04 Conference, Zakopane, Poland, 2004, pp. 359-368.

[9] S. Osiński and Dawid Weiss, " A Concept-Driven Algorithm for Clustering Search Results," *IEEE Intelligent Systems*, May/June, Vol. 20, No. 3, 2005, pp. 48-54.

[10] G. Mecca, S. Raunich, and A. Pappalardo, "A new algorithm for clustering search results," *Data Knowledge and Engineering. Vol. 62, No.* 3. September 2007, pp. 504-522.

[11] Wall, Michael E., et al., "Singular value decomposition and principal component analysis," *A Practical Approach to Microarray Data Analysis*, D.P. Berrar, et al, eds. pp. 91-109, Kluwer: Norwell, MA., 2003.

[12] Jain, A.K., and Dubes, R.C, *Algorithms for Clustering Data*, Prentice Hall, New Jersey, 1998.

[13] Conrad, J.G., AI-Kofahi, K., Zhao, Y. and Karypis, G., "Effective Document Clustering for Large Heterogeneous Law Firm Collections," *10th International Conference on Artificial Intelligence and Law (ICAIL)*, pp. 177-187, 2005.

[14] Lee, K.S. and Kageura, K., "Korean-Japanese Story Link Detection Based on Distributional and Contrastive Properties of Event Terms," *Information Processing & Management,* Vol. 42, no. 2, pp. 538-550, 2006.

[15] Kaufman, L and Rousseeuw, P., *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York, 1990.

[16] Geraci, F., Maggini, M., Pellegrini, M. and Sebastiani, F. "Cluster generation and cluster labeling for web snippets: A fast and accurate hierarchical solution," *Internet Mathematics,* Vol. 3, No. 4, pp. 413-443, 2008.

[17] 黃挺立,陸承志, "基於語意相關詞的搜尋分群," ICIM 2013 國際資訊管理學術研討會, 2013 年 5 月 25-26 日,真理大學.