商品展覽會深網整合及其關鍵字查詢排名策略

周清江
淡江大學資訊管理學系
副教授
cjou@mail.tku.edu.tw

石永瑜
淡江大學資訊管理學系
碩士班研究生
yungyu.shih@msa.hinet.net

摘要

隨著網路使用量不斷地增加，搜尋引擎已成為蒐集資訊情報的重要工具，但仍然有許多有價值資料隱藏在網路表層網頁的資料庫內，是無法有效率的在傳統搜尋引擎中被找到，本研究以商品展覽會網路資料庫為例，提供一個解決方案。一個中小企業人員及參展廠商，在網路上常面臨到無法確實得知何時何地有國際展覽會舉行，而展覽會中又有哪些公司及相關產品參展，所花費的時間過長且找尋到資料也未必齊全，也無法真正地蒐集展覽會相關資訊。本研究將網路上來自相同領域不同展覽會的資料進行整合，並提供使用者進行產品關鍵字查詢，查詢結果包括了產品所屬的公司及該公司中與關鍵字相關產品。本研究由兩個系統完成：(1)爬行擷取系統使用網路機器人，蒐集多個展覽會網站資料來源、將不同網站所提供的資訊，整合於關聯式資料庫中；(2)查詢處理系統處理關鍵字查詢，且提供排名策略，除了參考過去研究之 Tuple Tree 大小標準化、文件長度標準化、文件頻率標準化及文件之間權重標準化的調整因素外，本研究加入特定欄位出現次數權重(α)及異質資料倍率權重(β)進行排序調整，讓公司及產品資訊與使用者輸入的關鍵字相關性較高者，排名較前面。經過實驗顯示，當α值為 4 及β值為 2 時，MRR 的結果為 0.3876，與過去的做法比較，約有 6%以上改善。

關鍵詞：深網整合、關鍵字查詢、排名策略

壹、緒論

一、 研究背景與動機

網際網路是因為軍事通信的發展而蓬勃，Tim Berners-Lee 在 1993 年以純文字格式基礎設計出超文件標示語言(HyperText Markup Language，HTML)，隨著 HTML 使用量的增加，將多種形式的多媒體加入網頁，不斷地擴充及發展網頁上的功能，使得網站設計有了基本規範，讓使用者透過瀏覽器觀看網頁，不僅增進了全世界的距離，也隨著網際網路快速發展及其不斷深入應用，網路的可用性研究也逐漸成為關注的焦點，網站的價值也逐漸上升。

一般將網頁內容分為表層網路(Surface Web)及深層網路(Deep Web)。表層網路藉由傳統搜尋引擎的搜尋機器在靜態網頁中爬行，挖掘得到的內容或資訊，例如：Google、Yahoo 等；深層網路是不能透過傳統的搜尋引擎得到內容，需要藉由查詢介面來存取後端資料庫，再經由網站伺服器動態產生頁面，將結果回傳給使用者。深層網路主要針對某一特定領域的網路搜尋服務，提供品質更好的資料，例如：書籍領域的 FindBook 網站。根據 BrightPlanet[5]在 2001 年調查統計顯示，深層網路的資料量約是表層網路的 400 至 550 倍，而且內容涵蓋各個主題領域。Fetterly 等人[7]的研究統計，到 2003 年為止，網際網路上的資料量已經超過 200000 Terabyte (TB)，並且這資訊仍然迅速成長中。為了從大量龐雜的資料中，找到所需的資訊，使用者已習慣依賴搜尋引擎網站所提供的服務尋找資料。

網際網路給人們帶來便利的同時，也產生了許多問題。就我們所知，在全球無數個展覽會網站中，過去沒有針對展覽會資訊提供整合的機制，對一位一般使用者或專業領域知識者而言，利用相關的關鍵字找尋展覽會參展產品，必需至各個展覽會的網站介面中搜尋，不但蒐集資料所花費的時間過長，且資料未必齊全。在展覽會網站資訊中，中小企業人員、參展廠商、採購人員可藉由展覽會資訊知道某一產品在哪些公司中參展、參展公司過去參加展覽會的頻率、經驗、參展產品的多寡等等訊息，來幫助本公司是否向某家公司採購產品或進行產業合作等決策。因此如果有系統自動蒐集相同領域不同展覽會網站資訊整合起來，建立在關聯式資料庫提供關鍵字查詢，讓使用者輸入產品相關關鍵字，完整性地獲得有哪些公司的產品在展覽會中參展，比較各公司參展產品的型號、規格、樣式等不同地資訊，這樣的系統可為使用者帶來許多便利性。

在關聯式資料庫中，展覽會資訊、公司資訊及產品資訊分別儲存在不同的資料表內，每張資料表具有多個欄位，過去相關文獻[3][6][8][9][10]研究指出，關鍵字查詢資料時不能只針對單一資料表，如果想要查詢不同資料表的內容，需要在資料表之間加入主鍵與外來鍵之間的連接，建立資料表與資料表之間的關係，並透過全文檢索方式查詢資料；但是在查詢結果中，過去研究將資料表的所有欄位型態視為全文檢索形式，且查詢結果直接列在雜亂的清單，未考慮到列出的資料是否符合使用者需求，未使用有效的排名績效，對使用者還是造成許多困擾。最後查詢資訊傳送給使用者前，對關鍵詞查詢得到的結果進行排序是非常重要的工作，因為使用者通常只對最相關的結果感興趣，一個成熟的資訊檢索系統，制定一套查詢結果排名策略是很重要的。

二、 研究目的

展覽會網站於深層網路中包含大量豐富的資料，為了讓使用者在特定領域不同網站，進行產品關鍵字查詢，找尋更專業及充分的資訊，並提出有效的排名策略，達到更貼近符合使用者查詢結果。本研究發展商品展覽會深網整合及其關鍵字查詢排名策略，

目的有二：

一、建立商品展覽會深網整合：全球資訊化時代，許多公司都透過商品展覽會參展，來提昇公司的技術、形象，並積極推廣參展的產品，逐步建立國際買家、廠商對產品的認知，擴大國際市場對於其產品的接受度。各個展覽會舉行前，會在官方網站公開列出本次展覽會資訊、參展的公司、參展的產品等資訊，有高度興趣的人員會在這裡找資料，對他們在公司經營決策上，展覽會資料是很重要的。然而，過去相關的展覽會深網整合文獻卻不曾探討這個議題，但中小企業人員及參展廠商對此類議題具有高度興趣，他們可透過本研究成果所開發之展覽會深網整合服務，尋找更專業精確的全球產品展資訊。本研究使用網路機器人，蒐集多個展覽會網站資料來源、將不同網站所提供的資訊，整合於關聯式資料庫中，並為使用者開發一個跨平台的單一全球查詢介面網站，可快速便利的查詢到來自不同展覽會中的資訊。

二、關鍵字查詢排名策略：根據關聯式資料庫中整合後的資料，建立展覽會資料表、公司資料表及產品資料表的部份欄位為全文檢索型態，應用資訊檢索(Information Retrieval)技術中關鍵字搜尋(Keyword Search)，讓使用者輸入產品關鍵字，找尋產品所屬的公司及該公司中與關鍵字相關產品。目前大部分的資訊檢索系統以相關性作為檢索結果排序基礎，但有許多排序後的結果未必為使用者的需求，因此本研究將參考公司與產品的相關性及重要性、資料表不同欄位間具有不同程度的差異，給予不同欄位的倍率權重，來提出有效的排名策略，重新調整原始排序結果，改善未排序前的準確性。

貳、文獻探討

本節探討與本研究相關的文獻，包含深層網路資料整合、關聯式資料庫中關鍵字查詢、查詢結果排名策略相關議題。

一、深層網路資料整合

CompletePlanet 目前為全球最大蒐集深層網路資料庫，共分為 43 個特定主題，約七萬個網站，但這樣巨大的資料量，卻只些微佔有深層網路資料量的一部分。深層網路包含大量豐富的資料，現在越來越多的網路開發商架設網站皆包含後端資料庫系統，提供使用者輸入不同關鍵字搜尋，獲得動態查詢內容，而這類內容是無法透過傳統搜尋引擎進行索引，但要以人工的方式分別至這些深層網路網站擷取資料，增加高度困難性。因此，如何有效利用深層網路網站中的資料是一個重要的議題。對使用者而言，希望有一個統一的查詢介面輸入關鍵字，可以直接擷取到多個深層網路資料庫的內容，並將資料加以整合，獲得有用的資訊。

劉偉[2]等人提出查詢結果處理模組，從深層網路資料庫的查詢結果，擷取並整合到一個統一結構化的模式下。此模組依查詢結果整合分為三個部分：結果的擷取、結果的注釋及結果的合併。來自各個網站伺服器提交查詢之後，當回傳結果以 HTML 的型式呈現，從 HTML 頁面中擷取所需資料，並以結構化的方式儲存，並識別資料欄位的意義，加入適當的註記，將查詢的結果進行有效的合併於一個統一模式下。

黃執強[1]提出一個自動化的屬性之間的對應，接受多個網站的網頁擷取資料當作輸入，一次處理某兩個網站，進行資料分析。利用不同網站中查詢到的資料、該資料所具有的資料型別特性，及重複資料出現部份，發展出一套多對多對應的資料分析整合系統，來整合同性質網站中的網頁資料。

遺憾的是劉偉[2]等人及黃執強[1]在資料整合時皆未使用到資料庫，無法利用全文檢索進行關鍵字搜尋，且資料整合後的結果，也未經過有效地的排名處理，直接傳送給使用者，在一個大量的資料中，依然給使用者一筆一筆點閱觀看資料，未考慮到這些資訊是否符合所需。

二、 關聯式資料庫中關鍵字查詢

在文本文件及網路搜尋引擎中，關鍵字搜尋已成為熱門的研究議題。許多主要資料庫供應商都有提供關鍵字查詢的工具。目前許多研究都關注在關聯式資料庫中關鍵字搜尋的問題[4][10][12]，特別是 Discover 系統[9]、Hristidis[8]、BANKS 系統[6]、DBXplorer 系統[3]及 Liu[10]等人皆支援關聯式資料庫中關鍵字查詢。

Discover 系統[9]利用資料庫系統主鍵及外來鍵的結合，根據關鍵字產生候選網路，將元素組合 Tuple 加入到網路中，並透過貪婪演算法，找尋查詢關鍵字的 Tuple。

Hristidis[8]採用資訊檢索文件相關性排名策略，改進 Discover 系統[20]的做法，提出了演算法提高查詢效率，並且加入 OR 語意，對 Top-k 查詢結果進行排序回傳給使用者。

BANKS 系統[6]提出將搜尋結果進行排序，並支援相關性排序和結果的呈現。讓使用者輸入關鍵字後，以啟發式向後搜尋整個 Tuple 圖形(Tuple graph)，這種作法是不考慮任何綱要資訊(Schema information)。

DBXplorer 系統[3]採用資訊檢索技術的索引方法，以廣度優先拜訪 Tuple 圖形中所有可能的 Tuple，建立結構化查詢語言(Structured Query Language, SQL)，在多個資料表中找到 Tuple 的位置，這種方法是專門處理關聯式資料庫，其不足之處在於只能處理 AND 語意的查詢。

表 1 彙整 Discover 系統[9]、Hristidis[8]、BANKS 系統[6]及 DBXplorer 系統[3]共同特徵性，每個系統皆允許使用者自由形式輸入關鍵字。

表 1 Discover 系統[9]、Hristidis[19]、BANKS 系統[6]及 DBXplorer 系統[3]特徵

(資料來源：本研究整理)

	自由形式關鍵字搜尋	圖形式回答	具有語意回答
Discover 系統[9]	V	V	AND Semantic
Hristidis[8]	V	V	OR Semantic
BANKS 系統[6]	V		AND Semantic
DBXplorer 系統[3]	V	V	AND Semantic

上述四個過去重要研究外， Liu[10]等人提出了一個新的排序策略，採用 Tuple Tree 大小標準化、文件長度標準化、文件頻率標準化及文件之間權重標準化的四個標準化因子，對結果進行有效排序，且傳回具有語義的結果，並透過大量實驗證明了查詢越複雜，搜尋效果改善越明顯。但本研究認為 Liu[10]等人提出的排序策略有三點不足之處：(1)將每一個 Tuple 中的資料皆進行全文檢索，當有資料為非全文檢索型式查詢時，沒有明確的處理方式。(2)每一個 Tuple 僅包含一個欄位，當 Tuple 中資料儲存於不同的欄位、且每個欄位具有不同的檢索型式時無法處理。(3) Tuple 中每一欄位重要性皆相同，無法區別不同欄位之間的重要性。

三、 查詢結果排名

網際網路搜尋引擎主要功能是針對查詢後的結果排名，如果單純透過資料庫的查詢結果，將未經過處理的資料，全部回傳給使用者，無法達到優良的效果。查詢結果排名需要一個理想的評分函數，可以真實地反應查詢結果與使用者輸入關鍵字的相關性，評分函數對所有可能的結果進行相關性評分，最後會依照結果的評分大小排序。

Discover 系統[9]首先提出了查詢結果排序的想法，後續許多研究都致力於排序方法的改進與優化，現今主要有四種常用的排序方法：(1)基於連接數量排序法：根據關連式資料庫主鍵與外來鍵的特性，按照結果所包含的連接數量多寡，進行資料的排序，當連接數量越多，相關性越低；缺點是無法對相同的連接數量進行排序。(2)基於權重值排序法：不僅考慮到關連式資料庫主鍵與外來鍵的特性，還增加資料庫的語意，不同的資料欄位會給予不同的權重值；缺點是欄位權重值需要特定領域的專家評估。(3)基於權威值排序法：根據資料庫的語意來給予的，對每一個資料及資料與資料之間的連接，皆給予一個不同的權威值，利用調整因子來調整排序的結果。(4)基於資訊檢索索引排序法：將關連式資料庫中的資料進行相關詞頻率(Term Frequency, TF)及反向文件頻率(Inverse Document Frequency, IDF)，透過得分函數獲得最終的分數。

Page[11]等人提出 PageRank 的排名理論，以被其他網站連結及向外連結其他網站的多寡計算而來，當 PageRank 值愈高，得到較高的排名，網頁就出現在搜索結果的愈前面，增加被瀏覽的機會。Balmin[4]等人以權威為基礎的排序法應用在關連式資料庫中，資料庫被建構在一個具有標記的有向圖中，首先定義查詢結果為資料圖形中的節點，每個節點有一個初始狀態，透過 Page[11]等人提出 PageRank 的演算法，使得每個節點到最後都會達到一個穩定的狀態，查詢支援 AND 語意及 OR 語意，查詢結果能找到相關的不包含任何查詢關鍵字的節點。

叁、系統分析與系統實作

本章節說明本研究系統分析設計與實作，於第一節說明整體系統架構以及各個系統元件的功能。於第二節說明商品展覽會深網爬行擷取系統。於第三節說明關連式資料庫中之關鍵字查詢排名策略。

一、 系統架構說明

本論文針對「香港貿易發展局」及「中經社」展覽會網站的電子及資訊產品類型產業，擷取技術採用非監督式，將資料儲存於關聯式資料庫中，提供使用者關鍵字查詢。過去的搜尋引擎讓使用者輸入關鍵字後，先至關聯式資料庫中查詢，由於查詢得到的結果，未經過有效的排名處理，較無法符合使用者需求，因此本研究會根據資料庫查詢結果與使用者輸入的查詢詞，提出有效的排名策略，重新調整排序的結果，將調適後的資訊回傳給使用者。本系統整體架構(圖 1)，依照其功能順序可分為兩個系統：左邊的「爬行擷取系統」及右邊的「查詢處理系統」。

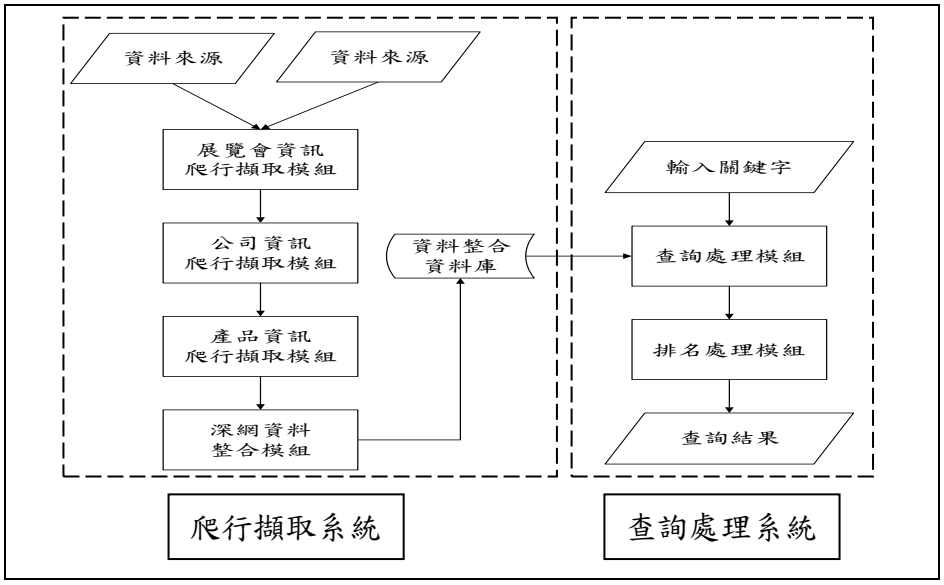

圖 1 系統架構圖

二、 爬行擷取系統

網路爬蟲爬行路徑如資料結構有向圖(Directed graph)，每一張網頁視為一個節點，網頁與網頁之間的連接視為節點的邊，本論文爬行方式採用廣度優先，從第一層開始爬行後，再繼續向下一層網頁爬行。本論文爬行擷取系統以 JAVA 語言開發，擷取方式採用 HTML Parser 及正規表達式(Regular Expression, REGEX)。HTML Parser 是一個開放原始碼，提供了線性及嵌套方式來分析網頁，類似於可延伸標記語言(eXtensible Markup Language, XML)中的樹狀結構，由於網頁上的文件許多為半結構化，HTML Parser 可將 HTML 的網頁擷取(Extraction)及網頁轉換(Transformation)。Regular Expression 可根據字串樣式規則，將字串進行全面性的格式化。一個正規表達式視為一個模式(Pattern)，描述符合句法規則的字串，例如「<a.+href*=*['\"]?.*?['\"]?.*?>」，表示在文字內容中尋找超連結資訊。

透過深層網路輸入關鍵字搜尋，回傳的查詢結果頁面中，包含許多幫助網路爬蟲完成爬行工作的重要資訊與特徵，幫助網路爬蟲完成爬行工作：(1)網頁原始檔由網頁標籤(Tag)與資料內容所構成，網頁標籤是網頁中的物件，而網頁標籤中可能還會包含網頁標籤，網頁標籤中的關鍵字，代表資料內容開始與結束的位置，例如「<table>…</table>」。(2)網頁的表頭或頁尾位置，常出現查詢結果的筆數及頁數等資訊，預先知道資料的範圍。

本論文爬行時會先到展覽會的網址的入口，先將展覽會資訊、公司資訊及產品資訊陣列初始化，分別取得回傳結果的 HTML 原始碼內容，爬行流程從展覽會資訊開始，至該展覽會參展的公司，到各公司所展示的產品。此外，必須先排除在許多頁面中有廣告，導覽列等非本研究感興趣的資訊，必須先排除，且參照一般使用者瀏覽網頁時的時間，加入隨機時間模擬使用者翻頁網頁的動作，避免網站開發商在瞬間偵測到大量的存取流量。

爬行擷取系統包含展覽會資訊爬行擷取模組、公司資訊爬行擷取模組、產品資訊爬行擷取模組及深網資料整合模組完成，詳細說明如下：

1. 展覽會資訊爬行擷取模組

在許多展覽會網站中，會依不同的參展類型分類，本論文以「電子與電腦」類型為例子，爬行相關的資訊，我們要找到符合「電子與電腦」類型的相關關鍵字，藉此能擷取所有展覽會資訊的內容，各網站展覽會資訊擷取完成後，儲存於關聯式資料庫中。

2. 公司資訊爬行擷取模組

根據「展覽會資訊爬行擷取模組」得到所有展覽會的清單，我們依循清單內的網址，進一步擷取公司資訊。由於每一家公司提供的資訊未必完整，並非所有的資訊都包含，其中常見的欄位包括：公司名稱、地址、電話、傳真號碼，網路爬蟲將擷取每一家公司提供的資訊，儲存於關聯式資料庫中。

3. 產品資訊爬行擷取模組

經過「公司資訊爬行擷取模組」的完成後，我們得到了所有公司的資訊，藉此可以更瞭解每家公司所參展的產品。由於產品資訊是本研究最關注的議題，進行資料整合時，我們必須要蒐集所有的產品資訊，每一個資訊來源提供的內容不同，產品資訊的項目繁多，因此我們採用最多欄位方式，擷取所有產品提供的資訊，儲存於關聯式資料庫中。

4. 深網資料整合模組

此模組的功能將爬行模組得到的資料，儲存於關聯式資料庫中，提供關鍵字查詢功能。資料庫建置實體關係圖(圖 2)，共有六個資料表，如下介紹：

(1) 展覽會資料表：根據「展覽會資訊爬行擷取模組」的結果，將展覽會的相關資訊，例如展覽會名稱、展覽會日期、展覽會地點等等欄位儲存於此資料表中。

(2) 公司資料表：根據「公司資訊爬行擷取模組」的結果，將參展的公司及其相關的公司參展資訊，例如公司名稱、公司描述、公司主要產品資訊、公司通路等等欄位儲存於此資料表中，。一家公司在不同展覽會中參展，視為「同」一家公司，在資料庫中也僅此一筆記錄，透過正規表達式，過濾公司的名稱是否重複。

(3) 產品資料表：根據「產品資訊爬行擷取模組」的結果，將各個公司參展的產品資訊，例如產品名稱、產品規格、產品描述、產品品牌等等欄位儲存於此資料表中。根據同一家公司的基準下，在單一展覽會中參展的產品，不論是產品名稱是否相同，皆視為「不同」的產品；但產品在多個展覽會中參展，如果產品名稱不同，視為「不同」的產品，如果產品名稱相同、產品的型號也相同，我們將視為「相同」的產品在不同展覽會中參展。

(4) 展覽會與公司關聯表：每一個展覽會有多家公司參展，而一家公司可以至多個展覽會參展，建立展覽會與公司之間的關連。

(5) 展覽會與產品關連表：每一個展覽會有多個產品參展；相同名稱的同一個產品可以到多個展覽會參展，但視為同一商品，建立展覽會與產品之間的關連。並根據一家公司在不同展覽會中參展，視為「同」一家公司的修正後，更新公司對應參展產品的關聯。

(6) 資料來源表：資料擷取前要明確確認資料來源的資訊。

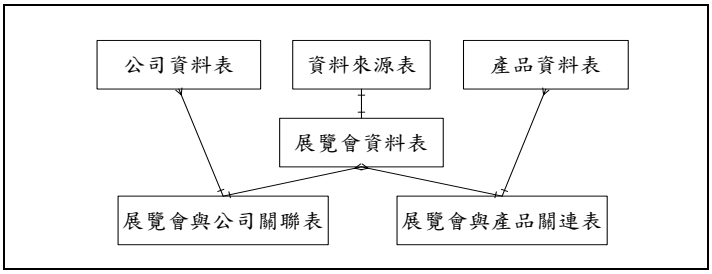

圖 2 實體關係圖

三、 查詢處理系統

深網整合於關聯式資料庫的結果，讓使用者進行產品關鍵字搜尋，找尋產品所屬的公司及該公司中與關鍵字相關產品。第一項查詢處理模組，說明輸入產品關鍵字後，查詢處理的建置。第二項排名處理模組，說明關鍵字查詢的排名策略機制。

1. 查詢處理模組

本研究將爬行後的資料，建構在 MySQL 中，MySQL 是關聯式資料庫管理系統(RDBMS)，提供全文檢索(Full Text)功能，能完整地查到所有內文出現關鍵字的文件。本研究針對公司資料表「公司描述(Company Description)欄位」、產品資料表中「產品名稱(Product Name)欄位」及「產品描述(Product Specifications)欄位」查詢資料，作為關鍵字搜尋的初步查詢。為了加快資料表中擷取資料列的速度，因此對公司資料表的公司編號(company_id)欄位與產品資料表的公司編號(company_id)欄位與產品編號(product_id) 欄位，建立索引(INDEX)型態，目的可以減少讀取傳回查詢結果的資料量，且可對資料表的資料列強制唯一性，以確保資料表資料的資料完整性。

結構化查詢語言(SQL)在資料查詢時，只需透過 SELECT 指令完成，完整的指令語法為「SELECT *column1, column2* FROM *table* WHERE *conditions*」，*column1* 與 *column2* 為記錄欄位、*table* 為資料表、*conditions* 為查詢條件，可包含一些條件子句，例如：AND、OR 及 ORDER BY 等。由於欄位的索引只能對欄位開頭的字串搜尋，如果資料內容是一篇文章，當文件長度過長時，索引的效果就失去了。因此，MySQL 提供了全文檢索的功能，當欄位內容向來冗長，藉此幫該欄位設定為全文檢索(FULLTEXT)型態，完整的指令語法為「SELECT *column1, column2* FROM *table* WHERE MATCH (*column1, column2,...*) AGAINST (*expr [search_modifier]*)，*column1* 與 *column2* 為記錄欄位、*table* 為資料表、MATCH( )函數是指在指定的欄位中，以自然語言搜尋符合資料、AGAINST( )函數是被給定的關鍵字搜尋字串。

本研究為了提供使用者進行關鍵字查詢，開發一個跨平台單一全球查詢介面網站，查詢頁面中只有一個輸入關鍵字入口，當提交(Submit)產品關鍵字後，會傳送到資料庫查詢資料，根據三個欄位分別查詢得到的資料，將去除資料中重複的結果，接下來傳送至排名處理模組中，計算排名順序。

2. 排名處理模組

此模組是本研究排名策略核心，利用資料存放於關聯式資料庫中的資料，讓使用者進行產品關鍵字搜尋，依照 Liu[10]等人作法，透過資訊檢索技術，應用在關鍵字搜尋排名策略，並修改其作法進行改良。以下三點分別說明 Liu[10]之關鍵字排名策略背景

介紹、本研究架構下之範例、以及特定欄位關鍵字出現次數權重與異質資料倍率權重之調整。

(1) Liu[10]之關鍵字排名策略背景介紹

關鍵字查詢可利用結構化查詢語言(SQL)語法完成，在關聯式資料庫中，往往依賴多個資料表組合得到查詢結果，需要建立資料表與資料表之間的關聯性。綱要圖(Schema Graph)方法視資料庫中每一個資料表為一個節點，資料表(R_i)與資料表(R_j)之間都具有主鍵與外來鍵之間的關係，綱要圖中的邊(Edge)即為節點(R_i)與節點(R_j)之間的聯接。資料表(R_i)中都具有多個欄位$(C_1^i, C_2^i, \dots C_{mi}^i)$。Tuple Tree 是由多個 Tuple 組合，簡稱「T」。Tuple Tree 中的每一個節點(t_i)即為一個 Tuple。當$(t_i \in R_i)$、$(t_j \in R_j)$，且$(t_i \cap t_j) \in (R_i \cap R_j)$，$(t_i,\ t_j)$即為 Tuple Tree(T)中的邊。Tuple Tree(T)的大小是所有 Tuple 的數量，每一個 Tuple 都包含多個文件$(\mathrm{D}_1, \mathrm{D}_2, \dots, \mathrm{D}_\mathrm{m})$，每一個文件$(D_i)$中，都具有多個欄位$(C_1^i, C_2^i, \dots C_{mi}^i)$。

過去研究[8]評估查詢詞與文件的相似度(公式 1)，是由關鍵字(k)在查詢詞(Q)上權重與關鍵字(k)在文件(D)中的權重決定。關鍵字(k)在文件(D)中的權重(公式 2)，較常使用資訊檢索中樞紐正規化權重法 (Pivoted Normalization Weighting Method) [8]。相關詞頻率(Term Frequency, TF)，指某一個給定的關鍵字在文件中出現的次數，出現的次數越多，代表的頻率就越重，為了防止偏向較長的文件，通常會被正規化，在這裡利用自然數(e)為基底取log函數兩次，將 *tf* 正規化為 *ntf* (公式 2.1)。反向文件頻率(Inverse Document Frequency, IDF)，指一個相關詞在文件中的重要性，當關鍵字出現在越多的文件中，代表越不重要，為了防止出現在過多的文件，通常會被正規化。某一特定相關詞的 *df*，可以由總文件數目除以包含該相關詞的文件數目，以自然數(e)為基底取log函數正規化為 *idf* (公式 2.2)。文件長度或相關詞在文件中的長度由位元組(bytes)的大小衡量(公式 2.3)，較大的文件通常包含多個相關詞及較高的相關詞頻率，導致關鍵字在文件中的權重往往過大，因此透過正規化來降低較大文件的相關詞權重，*avgdl* 為文件中的平均長度，*s* 為調整常數，過去研究通常預設為 0.2。

$$Sim(Q, D) = \sum_{k \in Q,D} weight(k, Q) * weight(k, D) \quad (1)$$

$$weight(k, D) = \frac{ntf}{ndl} * idf \quad (2)$$

$$ntf = 1 + \ln(1 + \ln(tf)) \quad (2.1)$$

$$idf = \ln \frac{N}{df + 1} \quad (2.2)$$

$$ndl = (1\text{-}s) + s * \frac{dl}{avgdl} \quad (2.3)$$

本研究將資料儲存於關聯式資料庫中，資料表由多個欄位組成，關鍵字查詢需要透過許多資料表的關聯，將查詢結果的公司與產品，依不同的公司區分，建立每家公司的 Tuple Tree。由於每一個 Tuple 都包含了一個文件　，　裡包含所有的欄位值，因此評估評估查詢詞　與文件　的相似度不再使用過去的作法[8] (公式 1)，將 Tuple 中的所有欄位值視為一個文件　，Tuple Tree(T)作為超級文件(Super-document)，計算查詢詞　與超級文件　之間的相似度(公式 3)。同時，原先計算每一個關鍵字與每一個文件　權重的做法，必須重新考量。過去研究[10]針對查詢詞　在超級文件的相似度作法，是由關鍵字　在查詢詞　的權重與關鍵字　在超級文件　中的權重決定。關鍵字　在超級文件　中的權重，必須先由關鍵字　在每一個文件　中的權重(公式 4)，再經過文件之間權重標準化　決定(公式 5)。

$$Sim(Q, T) = \sum_{k \in Q,T} weight(k, Q) * weight(k, T) \quad (3)$$

$$weight(k, D_i) = \frac{ntf * idf^{g}}{ndl * Nsize(T)} \quad (4)$$

$$weight(k, T) = Comb(weight(k, D_1), \ldots, weight(k, D_m)) \quad (5)$$

關鍵字(k)在每一個文件(D_i)中的權重(公式 4)，需參考四個影響排序重要性因素，包括 Tuple Tree 大小標準化$(Nsize(T))$、文件長度標準化(ndl)、相關詞頻率標準化(ntf)及文件頻率標準化(idf^g)。

① Tuple Tree 大小標準化$(Nsize(T))$找尋一個 Tuple Tree 中包含了多少個 Tuple，當 Tuple 數量越多，會得到更多的相關詞及相關詞的權重會越高(公式 6)，$size(T)$為文件長度(dl)。

$$Nsize(T) = (1\text{-}s) + s * \frac{size(T)}{avgsize} \quad (6)$$

② 文件長度標準化(ndl)目的是為了要區別欄位中所有資料的長度(公式 7)。文件長度(dl)為擷取到的結果中，該欄位中的內容長度；平均文件長度$(avgdl)$為擷取到的結果中，該欄位中的內容平均長度。

$$ndl = \left\{(1\text{-}s) + s * \frac{dl}{avgdl}\right\} * (1 + \ln(avgdl)) \quad (7)$$

③ 相關詞頻率標準化(ntf) 探討關鍵字在文件中出現的次數，出現的次數越多，代表的頻率就越重(公式 2.1)。

④ 文件頻率標準化(idf^g)探討相關詞在多少文件中出現(公式 8)。將查詢得到的結果視為一個集合，集合中有許多不同字詞，分布在不同的地方。原本計算範圍從查詢後的資料(Local)，修改為全域(Global)的資料，df^g為相關詞在資料庫所有文件中出現的頻率，N^g為在資料庫所有文件的數量。

$$idf^g = \ln \frac{N^g}{df^g + 1} \quad (8)$$

四個標準化因素計算關鍵字(k)在每一個文件(D_i)中的權重(公式 4)，僅代表每一個 Tuple 的值，接著要計算關鍵字(k)在超級文件(T)中的權重，需要透過文件之間權重標準化$(Comb())$，來計算文件與文件之間的權重，評估每一個 Tuple Tree 的分數(公式 9)。$maxWgt$為 Tuple Tree 中，關鍵字(k)在文件(D_i)中最大的權重值；$sumWgt$為 Tuple Tree 中，關鍵字(k)在文件(D_i)中所有權重值得加總。

$$Comb() = \max Wgt * \left\{1 + \ln(1 + \ln \frac{sumWgt}{\max Wgt})\right\} \quad (9)$$

雖然查詢詞(Q)與超級文件(T)之間的相似度計算完成，每一個 Tuple Tree 都有自己

的分數，將分數由高至低排序，越高的分數所代表的相關性較強，因此經過排序的結果，排名較高的文件，使用者想要獲得的資訊較高。

(2) 本研究架構下之範例

本系統讓使用者輸入產品關鍵字，找尋產品所屬的公司及該公司中與關鍵字相關產品，因此在資料庫中，需有公司資料表(Company, C)、產品資料表(Product, P)、展覽會與公司之間的關聯資料表(Company-Product, CP)。當使用者輸入關鍵字詞後，系統會將關鍵詞送到資料庫中，透過結構化查詢語言(SQL)於公司資料表中公司描述(Company Description)欄位及產品資料表中，產品名稱(Product Name)與產品規格(Product Specifications)欄位查詢。

為了提供使用者輸入產品關鍵字，找尋產品所屬的公司及該公司中與關鍵字相關產品。因此，以公司為基礎下，本研究建置 Tuple Tree 為例(圖 3)，分為兩個階層、至少一個 Tuple 組成，每一個 Tuple 只存在一個文件(D_i)。第一階層的 Tuple 代表公司資訊，為整個 Tuple Tree 的樹根，Tuple 中的文件僅有一個欄位為「公司描述」；第二階層的 Tuple 代表該公司與關鍵字查詢相關的產品資訊，Tuple 的數量會視該公司與關鍵字查詢相關的產品多寡決定，此階層中的每一個 Tuple 皆為 Tuple Tree 中的葉節點，用來連接與所屬公司之間的關連，每一個 Tuple 中的文件包含了兩個欄位為「產品名稱」及「產品描述」。因此，當資料庫查詢得到M個產品歸屬於N家公司參展，即可得到N個 Tuple Tree，再對N家公司進行排名順序。

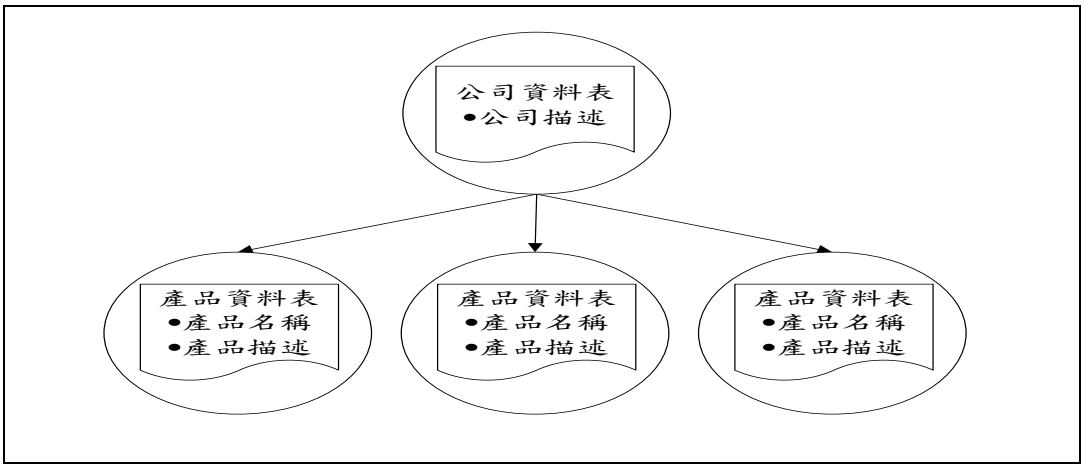

圖 3 本研究建置 Tuple Tree 示意圖

本範例以"notebook"關鍵字詞，說明本系統之排名處理模組作法。根據結構化查詢語言(SQL)的查詢，分別在公司資料表之公司描述欄位中找到 29 筆記錄、產品資料表之產品名稱欄位中找到 54 筆記錄及產品描述欄位中找到 122 筆記錄，經過重複性去除後，共有 133 筆產品資訊，隸屬於 29 家公司中參展。由於查詢結果資料量大，在此僅節錄一家公司說明查詢詞(Q)在超級文件(T)的相似度。在資料庫預設公司編號為「C430」，

該公司與關鍵字相關產品共有 2 筆記錄，令產品編號分別為「P1916 及 P1917」(圖 4)，建立 Tuple Tree 表示方式為「P1916→CP1←C430→CP2←P1917」，此 Tuple Tree 中共包含 3 個 Tuple，每一個 Tuple 僅有一個文件(D_i)。Tuple1 代表公司資訊，該文件中僅有一個「公司描述」欄位，其內容為「Notebook Coolers, Notebook Bags/Sleeves, i-Pad Stand」，令此 Tuple 以公司編號「C430」表示「Tuple1={C430}」；其餘 2 個 Tuple 皆為該公司與關鍵字查詢相關的產品資訊，每一個 Tuple 中的文件包含了兩個欄位為「產品名稱」及「產品描述」，令此 Tuple 依序以產品編號表示「Tuple2 = {P1916}及 Tuple3 = {P1917}」。

Company (C)

Company ID	Company Description
C430	Notebook Coolers, Notebook Bags/Sleeves, i-Pad Stand

Product (P)

Company ID	Product ID	Product Name	Product Description
C430	P1916	notebook cooling pad	this cooling pad helps release built-up heat in your laptop. it has dual cooling fans, a rich stereo system and a 6-angle adjustable workspace. the cooling fans provide excellent cooling performance, while the 2.0 stereo provides brilliant sound. the adjutable tilt prevents eye, neck and back strain while working for include colour box, 1 cooling pad, 1 usb power cable support up to 17 notebooks easily switch between 3 modes fans only, speakers only, fans and speakers convenient volume control wheel plug and play no drivers or software to install
C430	P1917	notebook desk	it is the perfect notebook cooling desk for users who enjoy watching movies or listening to music on their notebooks while relaxing in bed. this foldable cooling desk provides an excellent sound performance for the notebook, mp3 player or gaming device with its rich 2.1 stereo system. powered by only one usb cable to the notebook, the 2.1-channel amplifier drives two speakers and one subwoofer built into the desk. two ultra-silent fans under the notebook provides cooling due to long hours of desktop surface is angled at 15x which is an ideal position for typing while you are seated in bed buttons to control speaker volume and muting as well as turning the fan on and off independently foldable legs for easy storage

Company – Product (CP)

Company – Product ID	Company ID	Product ID
CP1	C430	P1916
CP2	C430	P1917

圖 4 "notebook"關鍵字詞為例之資料庫設計

根據 Tuple Tree 建置結果，需參考四個影響排序重要性因素，進行查詢詞與 Tuple Tree 的相似度計算(公式 3)，當每家公司獲得的分數，作為排序之準則，以關鍵字「notebook」說明相似度計算過程與結果(圖 5)：

		Nsize(T)	*ndl*		*ntf*		*idf²*		*weight(k, D)*	
Company	**Tuple**	**Nsize(T)**	**Company**	**Product**	**Company**	**Product**	**Company**	**Product**	**Company**	**Product**
430	430	1.4061	8.93904079	0	3.25283758	0	3.7763008	0	0.97728635	0
430	1916	1.4061	0	7.67320769	0	2.49562186	0	6.25485815	0	1.44678196
430	1917	1.4061	0	8.11626747	0	2.71572809	0	6.25485815	0	1.48843951

圖 5：關鍵字「notebook」相似度計算之各參數結果

① Tuple Tree 大小標準化：目的是了解 Tuple Tree 中包含了多少個 Tuple，不需對各個 Tuple 獨自計算，各個 Tuple 之值會皆相同(公式 6)。本 Tuple Tree 中共有 4 個 Tuple，Tuple 平均長度為$(avgsize)$為 0.3333，每個 Tuple 之$Nsize(T)$皆為 1.4061。

② 文件長度標準化：本參數計算個別文件的大小，需要將公司與產品分開計算文件長度與平均文件長度(公式 7)。關鍵字「notebook」查詢結果之公司平均文件長度為 3.0150、產品平均文件長度為 235.8158。Tuple1 的公司文件長度為 52，得到文件長度標準化為 8.9390。

③ 相關詞頻率(term frequency, TF)標準化：本參數探討關鍵字在文件中出現的次數，需要將公司與產品分開計算相關詞頻率(公式 2.1)。關鍵字「notebook」在 Tuple1 中僅出現過 2 次，相關詞頻率(TF)為 0.0050，得到相關詞頻率(term frequency, TF)標準化為 3.2528。

④ 文件頻率標準化：本參數探討關鍵字在多少文件中出現，需要將公司與產品分開計算文件頻率(公式 8)。本研究共蒐集 6,630 家公司及 23,077 筆各公司所參展產品，Tuple1 的文件為公司資訊，「notebook」關鍵字詞，有在公司文件中出現，文件頻率(DF)為 0.0001，文件頻率標準化為 3.7763。

關鍵字(k)在每一個文件(D_i)中的權重(公式 4)，需參考 Tuple Tree 大小標準化、文件長度標準化、相關詞頻率標準化及文件頻率標準化的結果而獲得。例如：Tuple1 的文件為公司資訊，Tuple Tree 大小標準化為 1.4061、文件長度標準化為 8.9390、相關詞頻率標準化為 3.2528 及文件頻率標準化為 3.7763，藉此獲得關鍵字(k)在此文件中的權重為 0.9773。文件之間權重標準化是利用關鍵字在不同文件中的權重，計算關鍵字(k)在超級文件(T)中的權重。"notebook"關鍵字詞為例之 maxWgt 為 1.4884 及 sumWgt 為 1.3042，得到關鍵字(k)在超級文件(T)中的權重為 2.6714，得到關鍵字(k)在超級文件(T)中的權重為 2.3891。然而，查詢詞(Q)與超級文件(T)之間的相似度(公式 3)，是由關鍵字(k)在查詢詞(Q)的權重與關鍵字(k)在超級文件(T)中的權重決定，現已得知關鍵字(k)在超級文件(T)中的權重，關鍵字(k)在查詢詞(Q)的權重作法與關鍵字(k)在超級文件(T)中的權重相同，關鍵字(k)在超級文件(T)中的權重為 2.2579，因此，本 Tuple Tree 之查詢詞(Q)與超級文件(T)之間的相似度為 5.3943。

(3) 特定欄位關鍵字出現次數權重與異質資料倍率權重之調整

每一家公司經過相似度計算，得到排名分數，將分數由高至低排序，得到越高的分數代表其所代表的相關性較強。但本研究認為評估關鍵字排名策略，不能僅考慮關鍵字在文件中的相似度。本研究針對關鍵字在特定欄位中出現次數權重(α)與異質資料倍率

權重(β)，加入排名策略調整。

① 關鍵字於特定欄位出現次數權重(α)：Tuple 中的文件是多個欄位所組成，由於每個欄位名稱都存在不同的意義，因此可對特定的欄位，增加關鍵字在文件中出現的次數。當重複出現關鍵字的次數越多，代表該文件對此關鍵字的相關性較強，對整體的排名順序會有幫助。在產品 Tuple 中，每個文件都包含了兩個欄位，分別為「產品名稱」及「產品描述」。本研究針對關鍵字出現在「產品名稱」欄位時，為了提高關鍵字出現的次數，進行相關詞頻率(TF)公式修正。當關鍵字(k)在「產品名稱」欄位中出現x次，將調整為$(x * \alpha)$次，α值將透過實驗來調整關鍵字在欄位中的重要性。根據相關詞頻率(TF)調整後結果，再進行相關詞頻率標準化(ntf)(公式 2.1)。

② 異質資料倍率權重(β)：由於不同的文件所存放的內容，代表了不同的意涵，例如：公司文件描述的是該公司的經營項目、服務範圍；而產品文件描述的是每個產品的型號與規格等等資訊。為了區別每個文件所代表意義不同，提高不同文件之間的重要性，本研究將關鍵字(k)在每一個文件(D_i)中的權重(公式 4)，分別給予不同的權重值，來調整排名績效。令公司與產品的倍率權重為β，公司之β值以 1 為基底，產品之β值透過實驗，重新調整關鍵上(D_i)的權重(公式 10)。

$$weight(k, D_i) = \frac{ntf * idf^{g}}{ndl * Nsize(T)} * \beta \quad (10)$$

最後根據查詢詞(Q)與超級文件(T)之間的相似度(公式 3)的定義，將關鍵字於特定欄位出現次數權重(α)及異質資料倍率權重(β)進行調整，獲得每一個 Tuple Tree 之值。

肆、實驗與討論

一、實驗建置

本研究之開發環境在 Microsoft Server 2008 R2、Intel Xeon CPU E5620 2.13GHz 處理器、12GB 記憶體、4TB 硬碟的伺服器，進行關鍵字排名策略之實驗。

在 B2BManufactures.com[1]網站中，消費電子產品、電子零件及電腦與電腦週邊目錄

[1] http://www.manufacturers.com.tw/

下，共收集 30 個關鍵字詞，包括「android、battery、camera、charger、coffee machine、computer adaptor、digital photo frames、earphone、electric socket、flash memory、galaxy projector、home audio、iphone case、iphone speaker、lenses、microphone、monitor、mouse、playstation、projector、remote control、scanner、server、smartphone、solid state disk、spotlight、switch、telephone、television、wireless keyboard (依字母排序)」，經由三位具有專業領域使用者，分別對 30 個關鍵字於 Liu[10]提出的方法測試。根據不同關鍵字的查詢結果，將排名前 10 名的公司進行勾選，評估哪些公司之展出的產品與輸入的關鍵字相關，作為目標答案的後選公司，而每家公司如果被一半以上使用者勾選，將視為目標答案，提供排名策略之特定欄位出現次數權重(α)與異質資料倍率權重(β)調整的依據。

為了評斷本研究提出的方法是否獲得良好的效果，我們採用 MRR 方法評估本系統之排名結果。Mean Reciprocal Rank(MRR)[13]，是一個評估排名好壞的方式，每一個結果的排名倒數，即為該結果的分數，最後評分為所有分數的平均數，此分數越高表示越能在前幾個答案中獲得資訊，n 為問題的個數、$rank(q_i)$表示第一個正確回答問題 q_i 的答案排行(公式 11)。

$$\frac{1}{n}\sum_{i=1}^{n}\frac{1}{rank(q_i)} \quad (11)$$

二、實驗結果與討論

根據三位使用者對 30 個關鍵字詞排名評估的目標答案，本研究將透過特定欄位出現次數權重(α)與異質資料倍率權重(β)進行調整，來提高公司重要性。不同α與β組合，會得到一組排序結果，將目標答案對應到每一組排序結果，計算 MRR 之值。將同α與β組合之不同關鍵字詞的 MRR 進行平均，找出最大之 MRR 結果，即為α與β調整之最適值。目標答案之α為 1、β為2^0，MRR 的結果為 0.3236，本研究分別將α值固定，探討β值調整對結果的影響及將β值固定，探討α值調整對結果的影響。

1. β值固定，探討α值調整對結果的影響

本研究設置β值$\{2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3\}$，探討$\alpha$範圍在$1 \leq \alpha \leq 30$的表現，實驗結果顯示(圖 6)，當$\beta$值為$2^1$且$\alpha$值為 4 時，MRR 之結果會達到最大值 0.3876。

2. α值固定，探討β值調整對結果的影響

本研究設置α值$\{1,2,3,4,5\}$，探討β範圍在$2^{-20} \leq \beta \leq 2^{20}$的表現，實驗結果顯示(圖7)，無論$\alpha$值為何，$\beta$值小於 2^{-6} 時，MRR 結果會趨近於 0.3804 達到收斂。最後，當α值為 4 且β值為 2^{1} 時，MRR 之結果會達到最大值 0.3876。

從實驗結果發現，不論以α值固定，探討β值調整對結果的影響或β值固定，探討α值調整對結果的影響，都可在α值為 4 及β值為 2^{1} 時找出最佳結果。

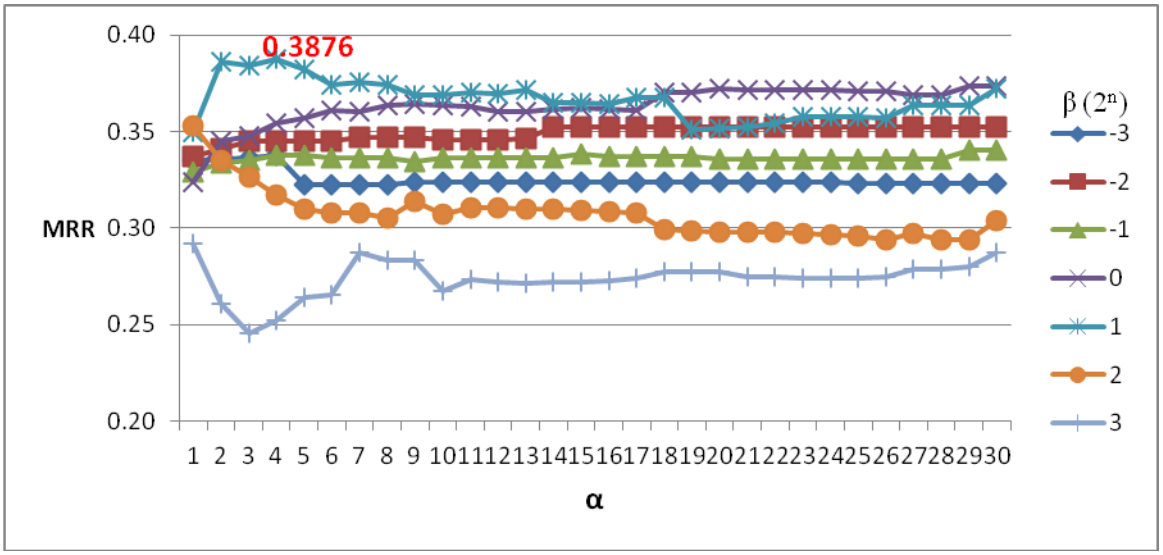

圖 6 β值固定， α值調整之結果

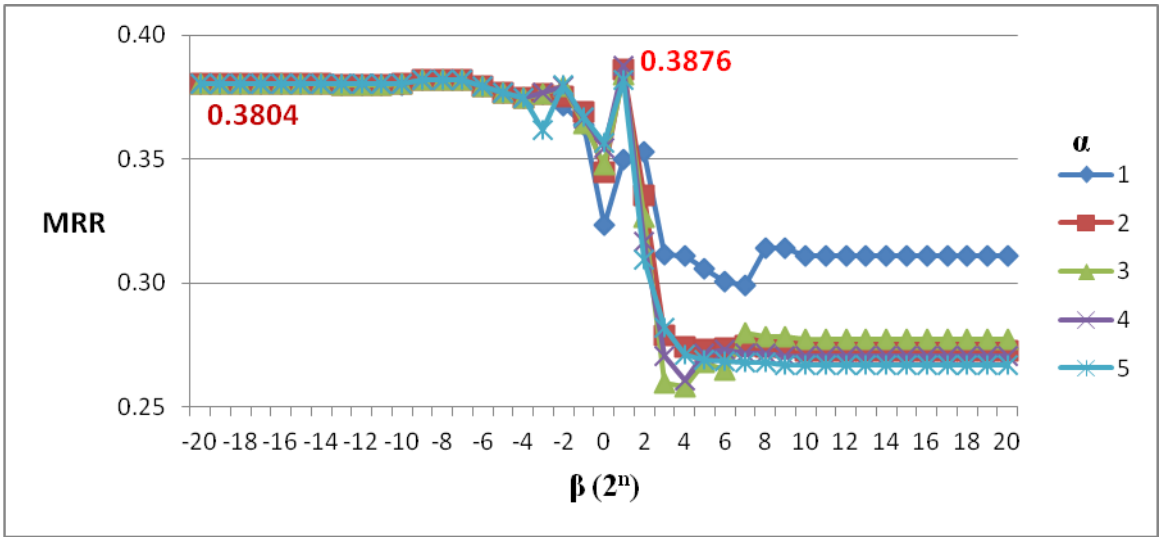

圖 7 α值固定， β值調整之結果

伍、結論與未來展望

本研究探討商品展覽會深網整合及其關鍵字查詢排名策略，共由兩個系統完成。為了提供中小企業人員、參展廠商、採購人員等使用者，提供在商品展覽會的不同網站中，藉由進行產品關鍵字查詢，而建立一個商品展覽會深網整合的單一全球查詢介面網站，讓查詢結果可獲得產品所屬的公司及該公司中與關鍵字相關產品等豐富資訊。

爬行擷取系統使用網路機器人，蒐集多個展覽會網站資料來源、將不同網站所提供的資訊，整合於關聯式資料庫中。查詢處理系統處理關鍵字查詢，然而查詢的結果未必符合使用者需求，本研究參照 Liu[10]的研究，加入 Tuple Tree 大小標準化、文件長度標準化、文件頻率標準化及文件之間權重標準化的調整因素，此外，本研究加入特定欄位出現次數權重(α)及異質資料倍率權重(β)進行排序調整。實驗階段邀請三位使用者針對 30 個關鍵字，在未使用特定欄位出現次數權重(α)及異質資料倍率權重(β)時，MRR 評估結果為 0.3236，作為本研究之標準答案，進行排序策略調整。經過實驗顯示，當α值為 4 及β值為 2 時，MRR 的結果為 0.3876，約有 6%以上改善。

未來的研究方向，可加入更多的資料來源整合，並可透過本研究為例，整合不同領域之間的資料。本研究之排名策略作法僅針對關鍵字計算相似度，未來可加入更多的因素，調整排名結果精確性。以資訊檢索技術角度來看，則可加入片語(Phrase)辨識、特殊關鍵字使用、根據使用者查詢紀錄，調整常用詞彙的重要性、可經過專家建議調整不同欄位之間的權重等因素，進而調整符合使用者需求的結果。

參考文獻

1. 黃執強. (2005). 同性質網頁資料整合之自動化研究 On.
2. 刘伟, 孟小峰, & 孟卫一. (2007). Deep Web 数据集成研究综述. *计算机学报, 30*(9).
3. Agrawal, S., Chaudhuri, S., & Das, G. (2002). *DBXplorer: A system for keyword-based search over relational databases*.
4. Balmin, A., Hristidis, V., & Papakonstantinou, Y. (2004). *Objectrank: Authority-based keyword search in databases*.
5. Bergman, M. K. (2001). White paper: the deep web: surfacing hidden value. *Journal of Electronic Publishing, 7*(1).
6. Bhalotia, G., Hulgeri, A., Nakhe, C., Chakrabarti, S., & Sudarshan, S. (2002). *Keyword searching and browsing in databases using BANKS*.
7. Fetterly, D., Manasse, M., Najork, M., & Wiener, J. L. (2004). A large‐scale study of the evolution of Web pages. *Software: Practice and Experience, 34*(2), 213-237.
8. Hristidis, V., Gravano, L., & Papakonstantinou, Y. (2003). *Efficient IR-style keyword search over relational databases*.
9. Hristidis, V., & Papakonstantinou, Y. (2002). *Discover: Keyword search in relational*

databases.
10. Liu, F., Yu, C., Meng, W., & Chowdhury, A. (2006). *Effective keyword search in relational databases*.
11. Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web.
12. Su, Q., & Widom, J. (2005). *Indexing relational database content offline for efficient keyword-based search*.
13. Voorhees, E. M. (1999). *The TREC-8 question answering track report*.

Abstract

With the rapid development of World Wide Web, the search engine has become an important tool to collect information. However, there are still lots of valuable information in the deep web that can't be found by traditional search engine efficiently. We tackle the problem using web exhibition product databases. A small and medium enterprises (SMEs) personnel and exhibitor often face a problem in the web that they could not exactly know when and where an international exhibition to would be held and they could not get the information about which companies and related products are in the exhibition. The collection of this information takes time. Furthermore, it may not be the complete information. In this study, we integrate different exhibition websites information in the same field. It provides users to search product through keyword query. Moreover, the query results include the product's company and its other products related to the keyword. The system is implemented by the combination of two systems. The first one is the crawler extracting system that uses network robot to collect many data of exhibition sites in the same field and to integrate these data into a relational database. The other one is the query processing system that answers a keyword query with its ranking strategies. Except for the tuple tree size normalization, the document length normalization reconsidered, the document frequency normalization and the inter-document weight normalization that were used in the past research, we join the specific field occurrences weight (α) and heterogeneous data weights (β) to adjust ranking list. The more company and product descriptions related to the keywords, the closer they will be put in the top of the result. Compared with past practices, when α is with value 4 and β with value 2, our experiments had a MRR value 0.3876, which was 6% improvement.