整合不同搜尋引擎的個人化網路資源搜尋器之研究與設計

李建億、周明仁、蔡政容
資訊教育研究所
國立台南師範學院
台南/台灣
E-mail：leeci@ipx.ntntc.edu.tw

摘要

隨著全球資訊網上資料量的迅速成長，使用者在全球資訊上學習時經常會漫無目標地學習或迷失學習的方向。因此，要想在全球資訊網中學習搜尋相關有用的資料，變成了一件困難的事。目前利用搜尋引擎來協助搜尋想要的資料成爲一種相當普遍和方便的方式。但是，由於各個搜尋引擎各有其特色，如果使用者在某一個搜尋引擎找不到他要的資料，則必須到另一個搜尋引擎，做相似反覆的查詢工作，如此查詢工作變成一件非常麻煩的事，而使用者也很容易失去了耐心，喪失了在網路上學習和蒐集資料的興趣。爲了解決此項缺失，而有了 Meta-Search 的研究發展，Meta-Search 本身不是一種搜尋引擎，而是當它接受了使用者的查詢詞之後，不做查詢檢索，而是把這個查詢詞再傳送到幾個不同的搜尋引擎，由這些搜尋引擎去做查詢的工作，但是 Meta-Search 所提供的網頁，對使用者而言，未必能滿足他的個別需求。所以在本論文中，我們將提出一種漸進式的合併排序法，來合併各種不同的搜尋引擎所傳回的結果，其做法是以相關的名次來做合併排序，並且可適應使用者對於相關資訊主觀認知上的差異，而在合併上做適切的調整。我們依據這個合併排序法實際建置一個系統，以提供適應個人化的合併查詢搜尋工具。經由實驗的結果，我們發現我們的合併排序方法相較於其他的方法，能有較佳的精準度。

關鍵字： Data fusion、Information Retrieval、MetaSearch、Ranking、WWW

1．緒論

隨著全球資訊網上資料量的日漸成長，全球資訊網中的資料變得非常龐雜，要想從中獲得相關有用之資料，利用搜尋引擎[17]來搜尋資料成爲一種相當普遍和方便的方式。然而，在目前存在的搜尋引擎中，如果使用者在某一個搜尋引擎找不到他要的相關資料，則必須到另一個搜尋引擎，做相似反覆的查詢工作，如果在這一個搜尋引擎又找不到使用者所需要的相關資料，則又必須到另一個搜尋引擎，還是做相似反覆的查詢工作，如此查詢工作，變成一件非常麻煩的事，而且使用者也很容易失去了耐心，喪失了在網路上學習和蒐集資料的興趣。爲了解決搜尋引擎的缺失，現有一種改進的方法，如 IQ99 搜尋家[3]，把查詢詞（Query Term）自動同時傳送到（可事先選定）幾個不同的搜尋引擎，然後再接收各個搜尋引擎所傳回的結果，並做簡單的呈列。然而該方法完全不對傳回的結果於相關性做進一步的合併排序（Ranking），因爲沒有經過排序的結果，呈現的資訊雖多，但卻顯得雜亂無序，並不能讓使用者快速得到他所想要的資

訊。爲了解決此項缺失，而有了 Meta-Search（Meta-Broker[8]的研究發展。

Meta-Search 本身不是一種搜尋引擎，而是當它接受了使用者的查詢詞之後，不做查詢檢索，而是把這個查詢詞再傳送到幾個不同的搜尋引擎，由這些搜尋引擎去做查詢的工作（如同前述 IQ99 搜尋家）。再則，Meta-Search 會根據這些搜尋引擎所傳回的網頁，依據查詢詞和網頁相關性的大小，將這些網頁整合在一起，合併這些搜尋引擎的搜尋結果，並加以排序，如此使用者便能藉由 Meta-Search 的合併排序，輕易地找到他想要之相關網頁，而不必再每次經由單一個搜尋引擎，做重覆查詢篩選的工作。然而，在 Meta-Search 中，網頁來自數個性質相異的搜尋引擎，如何合併這些網頁，並把網頁按相關性重新排序，是目前很重要的一個研究議題。有些研究的合併方法[10]是依據搜尋引擎傳回的相關屬性值，再利用自己的計算公式，計算出相關分數，再依據新的相關分數重新排列網頁。然而實際上絕大多數的搜尋引擎都不會提供如何計算權重的相關資訊，即使 Meta-Search 送出一連串查詢詞來猜測搜尋引擎的計算公式和網頁本身的相關屬性值，所得到的數據仍是缺乏正確性。再則，以現有的 Meta-Search，在某一段固定的時間內，不同的使用者，只要所輸入的查詢詞是一樣的，那輸出的結果就是一樣的，不會因爲使用者的差異，而有不同之結果，也就是說，無法因爲每一個人的資訊需求不同，而有不同的輸出結果，例如：同樣一個查詢詞"冬山河"，遊覽車司機和小學教師就有不一樣資訊需求，但目前的 Meta-Search 所呈現卻是一樣的內容，無法因人而異。

綜合以上的討論，我們可以了解到，使用者要在全球資訊網中獲得他認爲相關的網頁，將面臨下列二個問題：一是使用搜尋引擎雖然可以獲得相關的網頁，但是仍然必須經由一次又一次的使用不同的搜尋引擎，做相同的查詢工作，使用者才能獲得他認爲相關的網頁。另一個是使用 Meta-Search 雖然不必在不同的搜尋引擎之間一次又一次的反覆查詢，但是 Meta-Search 所提供的網頁，對使用者而言，未必能滿足他的個別需求。因此，針對 Meta-Search 我們將提出一種的合併排序法，改進現有 Meta-Search 所無法達到的效果，即是當不同的人輸入相同的查詢詞，會針對每一個人的資訊需求不同，而顯示出不同的結果。經過漸進式合併排序法之後的輸出，便有如教育上的因材施教一般，會因不同的人而輸出不同的結果。

在本論文中，我們將提出一種漸進式的合併排序法，來合併排序搜尋的結果，這是一種較爲適切的方法，並且可適應使用者對於相關資訊主觀認知上的差異。我們不以搜尋引擎的相關分數來合併排序，因爲不同的搜尋引擎，相關分數的計算方法、分數的高低範圍都不同，因此分數高的不一定相關性高，所以我們以相關名次來合併排序，因爲在任何一個搜尋引擎中，相關性大的一定排名在前。亦即，雖然網頁是來自不同的的搜尋引擎，但他們的相關名次，可以被用來比較相關性大小，名次在前的通常是相關性大的。並且在使用者瀏覽過網頁以後，由使用者決定每筆網頁相關與否。同時，針對每一個特定的查詢詞，漸進式合併排序法將計算出一組對應於各個搜尋引擎的權重（即是反應各個搜尋引擎各自對使用者而言所包含相關性資料的多寡）。當該使用者下次再做相同的查詢時，漸進式合併排序法便能依據對照該使用者的一組搜尋引擎之權重，來決定哪些搜尋引擎回覆的網頁應該有較高的排序名次，讓使用者能優先的瀏覽它，並再一次計算、儲存搜尋引擎的權重，以做爲該使用者下次合併排序時的依據。如此經過這種漸進式的合併搜尋，針對於每一個人的不同需求，利用漸進式合併排序法便能完全呈現出不同的搜尋結果，達成有意義的合併搜尋和資訊

個別化的目的。

2．文獻探討

在全球資訊網中，資訊量呈倍數成長，全球資訊網學習者要找到相關和有用的資訊，很多時候需要藉由搜尋引擎的輔助。一般而言，搜尋引擎系統[17][1]是一個架構在全球資訊網上的網站，除了搜尋引擎主系統以外，還包括摘要檢索資料庫(Annotation Database)等部份。並提供給學習者一個查詢介面，當學習者輸入適當的查詢詞後，搜尋引擎便從摘要資料庫中選取與查詢詞相關的網頁，將其網址甚至是網頁摘要說明，藉由網際網路(Internet)，傳送給在另一端的學習者。

搜尋引擎雖然可以幫助學習者尋獲相關的資料，然而，如果學習者在某一個搜尋引擎無法得到他需要的相關資料，則必須到另一個搜尋引擎，做相似反覆的查詢工作，如果在這一個搜尋引擎又找不到學習者所需要的相關資料，則又必須到另一個搜尋引擎，還是做相似反覆的查詢工作。而隨著在網路上更多搜尋引擎的建置，搜尋引擎之間難免會存在相同重複的網頁，因此學習者若在某一個搜尋引擎無法得到他需要的相關資料，到另一個搜尋引擎做相似反覆的查詢工作時，所得到的網頁資料，可能很多是前次相同查詢時已出現過的不相關網頁。

另有一種管理搜尋引擎的方法，稱之爲Meta-Search，在全球資訊網上，已有幾個類似Meta-Search 的產品，例如：MetaCrawler [19]、Internet Query 98 搜尋家[3]、Profusion [8][9]、SavvySearch [6][7]、Mamma [12]、Highway 61 [11]、MetaFind [13]、Ask Jeeves [4]、Dogpile[5]、Verio Metasearch[16]...等。而目前上述這些 Meta-Search 的產品，大都只是將搜尋引擎傳回的搜尋結果，依照傳回的時間先後次序，將結果作簡單的合併而已，其排名在前的，只是傳回結果給 Meta-Search 的時間較快而已，它的相關性和重要性並不一定是最大的。而較理想的 Meta-Search 法，是利用搜尋引擎傳回的屬性值、相關分數做爲參考值[10]，並設計自己的排序計算方法(Ranking Algorithm)，計算每一筆網頁屬於 Meta-Search 的相關分數，憑藉這個分數，把所有的搜尋結果加以合併，重新排列順序，相關分數高的網頁，便呈現在 Meta-Search 的愈前面。但是由於全球資訊網商業上競爭和商業機密，有時候搜尋引擎的建置者並不會提供給 Meta-Search 有關於搜尋引擎內部的排序計算方法、相關分數及有關的屬性值。此時 Meta-Search 只能用猜測的方式，傳送出一些查詢詞到搜尋引擎，由得到的結果，經過統計的計算方法，猜測搜尋引擎內部計算的排序計算方法、相關分數及有關的屬性值[10]。因爲是經由統計得來的結果，此方法不能保證搜尋合併的結果是正確的，所以相關性和重要性大的網頁並不一定呈現在最前面。除此之外，現有的 Meta-Search 法無法針對不同的學習者的需要，而有不同的呈現結果，不能滿足個別化不同需求。而UWRSA[2]雖能夠滿足個別化的需求，但在合併來自不同搜尋引擎的網頁時，僅考慮到相關網頁數量的多寡，並未考慮及相關網頁的名次。

3．漸進式合併排序法

以現有的 Meta-Search，在某一段固定的時間內，不同的使用者，只要所輸入的查詢詞是一樣的，那輸出的結果就是一樣的，不會因爲使用者的差異，而有不同之結果，也就是說，無法因爲每一個人的資訊需求不同，而有不同的輸出結果，例如：同樣一個查詢詞"冬山河"，遊覽車司機和小學教師就有不一樣資訊需求，但目前的 Meta-Search 所呈現卻是一樣的內容，無法因人而異。雖然上述的

UWRSA 能針對不同的使用者而有不同的輸出結果，但是其合併排名網頁的方式並未臻於合理與最佳化。因此，我們主要的構想乃是當不同的人輸入相同的查詢詞，能夠針對每一個人的資訊需求不同，而顯示出不同的結果。經過合併排序法之後的輸出，便有如教育上的因材施教一般，會因不同的人而輸出不同的結果。

一般來說當 Meta-Search 的作法是在合併搜尋結果時，根據搜尋引擎傳回的每一個網頁的分數，加以排列出其相關性的大小，在此便有一些問題產生。第一、搜尋引擎所提供的相關分數無法作爲參考；因爲相關分數是每一個搜尋引擎利用他自己獨特的計算方法演算得來的，不同的搜尋引擎有不同的計算方法，因此不同的搜尋引擎，分數大小無法比較相關性大或小。例如：在 Search Engine *A* 中的分數 0.9 其相關性可能大於 Search Engine *B* 中的分數 25 其原因是 *A*、*B* 的相關分數大小界限不同，$0 < A < 1$，$0 < B < 100$，也有可能 Search Engine *A* 和 Search Engine *B* 所採用的計算相關性大小方法不同，方法不同，搜尋引擎所提供的相關分數，自然無法比較其相關重要性的大小。第二、搜尋引擎根本沒有提供每一個網頁的相關分數；由於搜尋引擎建置不易和商業競爭上的機密，搜尋引擎除了提供相關資訊的全球資訊網網址（URL）和少許的資訊內容說明外，通常不會提供足夠相關有用的資訊（Weight Function，文件屬性），因此缺乏足夠的相關屬性值，Meta-Search 無法合併搜尋結果。爲了解決以上的問題，我們不以搜尋引擎所提供的相關分數作爲合併搜尋結果之依據，而是使用每一個網頁在搜尋引擎中的名次做爲合併搜尋結果的依據，而提出了一種新的合併排序法，稱之爲漸進式的合併排序法，如圖 3.1 所示，主要的作法是，在每一次查詢時，對於搜尋引擎給予適當的權重，經過一次又一次的合併查詢，能夠將使用者認爲最相關的網頁，漸進地排列於前而能優先呈現給使用者，使用者便能迅速地在前面排名的網頁獲得他所需要的資訊。

在圖 3.1 的漸進式合併排序法中，W_{hij} 爲使用者 *h*，執行查詢詞 *i* 時搜尋引擎 *j* 的「參考權重」；WT_{hij} 爲使用者 *h*，執行查詢詞 *i* 時搜尋引擎 *j* 的「總和權重」；*n* 爲 Meta-Search 所連結的搜尋引擎的數目（$n >= 2$）；Score（Sd_{hijk}）爲使用者 *h*，執行查詢詞 *i* 時搜尋引擎 *j* 中第 *k* 筆網頁的排序分數，由搜尋引擎 *j* 的參考權重 W_{hij} 乘以該搜尋引擎 *j* 中第 *k* 筆網頁的原始名次之倒數（$1 / k$），亦即，我們以網頁的「1 / 原始名次」爲網頁的新分數，再乘以搜尋引擎的權重。接著，根據 Score（Sd_{hijk}）分數由高而低合併網頁，Md_{hivjk} 爲合併後排名爲第 *v* 筆網頁且由搜尋引擎 *j* 所提供。假設 Md_{hivjk} 在搜尋引擎 *j* 中的原始位置爲第 *k* 筆，當使用者認爲 Md_{hivjk} 相關，則搜尋引擎 *j* 的權重 WT_{hij} 累加 $1 / k$。反之，當使用者認爲 Md_{hivjk} 不相關，搜尋引擎 *j* 的權重 WT_{hij} 減少 $Y * 1 / k$。再則，r_{hi} 爲使用者 *h* 在此次查詢詞 *i* 中所瀏覽合併結果的總網頁數目；R_{hi} 爲該使用者 *h* 對同樣查詢詞 *i* 的所瀏覽網頁的最大值，亦即，$R_{hi} = \text{Max}(r_{hi})$。因爲使用者所瀏覽的網頁數目，可能每次都不相同，所以每一次的 r_{hi} 值也都是不相同。在執行查詢動作時，若 $r_{hi} < R_{hi}$，表示沒有新的網頁被使用者瀏覽，不需更新搜尋引擎權重。若 $r_{hi} >= R_{hi}$，則 $R_{hi} = r_{hi}$，表示使用者瀏覽過了新網頁後，提供了更多的網頁相關與不相關之回饋，必須重新計算更新搜尋引擎權重。

另外，當兩個網頁的 Score（Sd_{hijk}）分數相等時，WT_{hij} 大的搜尋引擎的網頁先呈現，若網頁被搜尋引擎重複檢索時，合併後只呈現一次，其分數爲被檢索的所有分數的平均值。當 $WT_{hij} < 0$ 時，所有的 WT_{hij} 加上正規化係數 *Normals*，而加上 *Normals* 爲的是使 WT_{hij} 不至於爲負數且所有的 WT_{hij} 加上 *Normals* 之後，

彼此之間的大小差異不會因此改變。*SumWt* 爲所有搜尋引擎的 WT_{hij} 之總和，最後由 WT_{hij} / *SumWt* 求出搜尋引擎 *j* 的新 W_{hij}。

爲了說明起見，我們以下面的例子來解釋漸進式合併排序法的作法（如表 3.1）。假設 *M* 是一個使用漸進式合並排序法的 Meta-Search，S_A 以及 S_B 是二個在全球資訊網上的搜尋引擎，例如像 GAIS、Yahoo、Lycos…等，*i* 代表一個查詢詞。D_{A1}、D_{A2}、D_{A3}、…、D_{A10} 是代表 S_A 中回覆和查詢詞 *i* 相關的網頁，

這十個網頁的順序則是按照網頁和查詢詞 *i* 相關性的大小排列，相關性由大到小，相關性愈大的在愈前面，亦即 $D_{A1} > D_{A2} > D_{A3} \ldots > D_{A10}$。同樣的 D_{B1}、D_{B2}、D_{B3}、…、D_{B10} 是 S_B 中回覆和查詢詞 *i* 相關的網頁，這十個網頁的順序是按照網頁和查詢詞 *i* 相關性的大小排列，相關性由大到小，相關性愈大的在愈前面，亦即 $D_{B1} > D_{B2} > D_{B3} \ldots > D_{B10}$。假設現在有一個使用者，輸入了一個查詢詞 *i*，經由 Meta-Search 將查詢詞 *i* 同時傳送給了 S_A 以及 S_B。在 S_A 中和查詢詞 *i* 相關的網頁有 D_{A1}、D_{A2}、D_{A3}、D_{A4}、D_{A5}、D_{A6}、D_{A7}、D_{A8}、D_{A9}、D_{A10}，10 個網頁（相關性由大到小，如表 3.1（a））。在 S_B 中和查詢詞 *i* 相關的網頁有 D_{B1}、D_{B2}、D_{B3}、D_{B4}、D_{B5}、D_{B6}、D_{B7}、D_{B8}、D_{B9}、D_{B10}，10 個網頁（相關性由大到小，如表 3.1（b））。S_A 以及 S_B 把這些相關的網頁傳回給 Meta-Search，Meta-Search 把這 20 個網頁，根據 W_A / k，W_B / k，合併成 D_{A1}、D_{B1}、D_{A2}、D_{B2}、D_{A3}、D_{B3}、D_{A4}、D_{B4}、D_{A5}、D_{B5}、D_{A6}、D_{B6}、D_{A7}、D_{B7}、D_{A8}、D_{B8}、D_{A9}、D_{B9}、D_{A10}、D_{B10}（如表 3.1（c））。其中 W_A 以及 W_B 是 S_A 和 S_B 搜尋引擎的「參考權重」，$W_A = 0.5$，$W_B = 0.5$，初始值 $W_A = W_B$ 是假設對一個新的查詢詞，S_A 以及 S_B 對於使用者有著等量質的相關網頁，*k* 爲網頁在原搜尋引擎中的排序名次，從 1、2、3、…、*n*。另外，該使用者對此 20 個網頁作瀏覽以決定是否爲該使用者想要的網頁，其記錄於表 3.1（c）中的 Relative 欄中（"∨"表示使用者認爲他真正要需的網頁）。

最後，根據使用者瀏覽網頁後所決定相關與否資料來計算出下一次該使用者查詢相同查詢詞的搜尋引擎權重 W_A 和 W_B。WT_A 和 WT_B 是 S_A 以及 S_B 搜尋引擎的「總和權重」，爲搜尋引擎的累計相關加權重，其值是根據圖 3.1 的計算方法而產生，使用者認爲相關的網頁「總和權重」加上 $1 / k$，不相關者減去 $Y * (1 / k)$，其中 *k* 代表該網頁在原搜尋引擎上的名次。我們採用這種調整方式，是爲了要使排名高且使用者認爲相關的網頁能往前集中，因此排名越高者影響權重越大，給予的加權比重越大，排名低者給予較少的權重，因爲排名低者被使用者瀏覽的機會，本來就相對的少，所以其對「總和權重」影響應相對的變小，如此使用者在瀏覽前面幾筆網頁後，就能獲得他真正需要的資訊。根據圖 3.1，*Y* 爲一實數，爲減少「總和權重」運算式中的一個參數，*Y* 愈大，不相關個網頁出現時，所要減少的「總和權重」的部份愈大。*Y* 愈小，不相關個網頁出現時，所要減少的「總和權重」的部份愈少。我們使用的累計和減少搜尋引擎的「總和權重」的計算方法不同，不同的原因是我們認爲不相關的網頁出現時，減少的權重比例應該異於相關的權重計算方式。假設 $Y = 1 / 3$，利用圖 3.1 的計算方法，累計「總和權重」$WT_A = 2.48082$，$WT_B = 1.62897$，進而得到新的「參考權重」$W_A = WT_A / (WT_A + WT_B) = 0.6$，$W_B = WT_B / (WT_A + WT_B) = 0.4$。因此，當該使用者下次再輸入

/*對使用者 h，查詢詞 i 給定搜尋引擎 j 權重初始值 $WT_{hij} = 0$；$W_{hij} = 1 / n$；$r_{hi} = 0$*/

Do while (對使用者 h 一個查詢詞 i 被執行查詢)

/*計算網頁的分數 */

For j = 1 to n

  For k = 1 to m

    Score（Sd_{hijk}） = W_{hij} * （1 / k）

  Endfor

Endfor

根據 Score（Sd_{hijk}）重新合併陳列網頁

/*計算下一次使用者 h 查詢詞 i 的搜尋引擎 j 權重 W_{hij} */

If $r_{HI} > R_{HI}$

  $R_{HI} = r_{HI}$

  For v = 1 to R_{HI}

    If 使用者認爲 Md_{hivjk} 相關， $WT_{hij} = WT_{hij} + 1 / k$

    If 使用者認爲 Md_{hivjk} 不相關， $WT_{hij} = WT_{hij} - 1 /$ （$Y * k$）

  Endfor

/*Md_{hivjk} 爲合併後排名爲第 v 筆網頁且由搜尋引擎 j 所提供。假設 Md_{hivjk} 在搜尋引擎 j 中的原始位置爲第 k 筆 */

  For j = 1 to n

    If $WT_{hij} < 0$

      $Normals = 0 - WT_{hij}$

        For t = 1 to n

          $WT_{hit} = WT_{hit} + Normals$

        Endfor

    Endif

  Endfor

  For j = 1 to n

    $SumWt = SumWt + WT_{hij}$

  Endfor

  For j = 1 to n

    $W_{hij} = WT_{hij} / SumWt$

圖 3.1 漸進式合併排序法

S_A

Query	Document	*Rank*（k）
Q_1	D_{A1}	1
	D_{A2}	2
	D_{A3}	3
	D_{A4}	4
	D_{A5}	5
	D_{A6}	6
	D_{A7}	7
	D_{A8}	8
	D_{A9}	9
	D_{A10}	10

（a）搜尋引擎 *A* 提供十筆網頁

S_B

Query	Document	*Rank*（k）
Q_1	D_{B1}	1
	D_{B2}	2
	D_{B3}	3
	D_{B4}	4
	D_{B5}	5
	D_{B6}	6
	D_{B7}	7
	D_{B8}	8
	D_{B9}	9
	D_{B10}	10

（b）搜尋引擎 *B* 提供十筆網頁

Query	Document	Search Engine	W_j / *k*（*j*= *A* or *B*）	Relative
Q_1	D_{A1}	S_A	0.5	∨
	D_{B1}	S_B	0.5	∨
	D_{A2}	S_A	0.25	∨
	D_{B2}	S_B	0.25	
	D_{A3}	S_A	0.167	∨
	D_{B3}	S_B	0.167	∨
	D_{A4}	S_A	0.125	∨
	D_{B4}	S_B	0.125	
	D_{A5}	S_A	0.1	∨
	D_{B5}	S_B	0.1	∨
	D_{A6}	S_A	0.083	∨
	D_{B6}	S_B	0.083	∨
	D_{A7}	S_A	0.071	∨
	D_{B7}	S_B	0.071	∨
	D_{A8}	S_A	0.063	
	D_{B8}	S_B	0.063	
	D_{A9}	S_A	0.056	
	D_{B9}	S_B	0.056	∨
	D_{A10}	S_A	0.05	
	D_{B10}	S_B	0.05	

（c）第一次查詢後，網頁合併呈現情形

Query	Document	Search Engine	W_j / *k*（*j*= *A* or *B*）	Relative
Q_1	D_{A1}	S_A	0.6	∨
	D_{B1}	S_B	0.4	∨
	D_{A2}	S_A	0.3	∨
	D_{A3}	S_A	0.2	∨
	D_{B2}	S_B	0.2	
	D_{A4}	S_A	0.15	∨
	D_{B3}	S_B	0.13	∨
	D_{A5}	S_A	0.12	∨
	D_{A6}	S_A	0.1	∨
	D_{B4}	S_B	0.1	
	D_{A7}	S_A	0.086	∨
	D_{B5}	S_B	0.08	∨
	D_{A8}	S_A	0.075	
	D_{A9}	S_A	0.067	
	D_{B6}	S_B	0.067	∨
	D_{A10}	S_A	0.06	
	D_{B7}	S_B	0.057	∨
	D_{B8}	S_B	0.05	
	D_{B9}	S_B	0.044	∨
	D_{B10}	S_B	0.04	

（d）第二次查詢後，網頁合併呈現情形

表 3.1 一個漸進式合併排序法的例子

查詢詞 *i* 時，Meta-Search 便是以新的「參考權重」爲計算合併排名的根據，其結果如（如表 3.1（d））。我們可以發現，如此一次又一次漸進的方式來調整各搜尋引擎的權重，將使用者認爲相關的網頁漸進地往前呈現在合併搜尋結果的前面。因此，不同的使用者 *h* 對一查詢詞 *i* 於每一個搜尋引擎 *j* 將有不同的權重 W_{hij}，如此我們便以自動的方式，建立了使用

者之區別性，使得不同的使用者在輸入相同查詢詞便會呈現出不同的結果。

在我們的網頁分數計算方式 W_{hij} / k，亦即，W_{hij} * （1 / k）中，包括搜尋引擎權重和原始分數（1 / 名次）等兩部份，於此，我們將說明採行的原因。首先，我們將討論現有 Meta-Search 的作法及其缺失：

UWRSA 在合併網頁時，以搜尋引擎權重乘以原始分數當成網頁的最後分數。其原始分數的算法爲將整數 1，等分給所有的名例如共有 10 筆網頁，則其原始分數分別爲 1、9 / 10、8 / 10、7 / 10、6 / 10、5 / 10、4 / 10、3 / 10、2 / 10 以及 1 / 10。然而，UWRSA 取原始分數的方法，將使搜尋引擎彼此之間立於不平等的地位，因爲回覆較多網頁的搜尋引擎，相對於其他搜尋引擎，其網頁將會有較高的原始分數。

Profusion 在合併網頁時，以搜尋引擎權重乘以原始分數當成網頁的最後分數。其原始分分數除以所有網頁原分數最大者，例如共有 4 筆網頁，其搜尋引擎提供之原分數分別爲 1000、987、865 以及 776，則其原始分數分別爲 1、0.987、0.865 以及 0.776。Profusion 計算原始分數方法的缺失在於，當搜尋引擎根本不提供分數屬性時，將無法計算網頁的原始分數。

綜合上述的討論，我們認爲採行（1 / 名次）爲原始分數是較爲合理的計算方式，其有以下兩種原因：

有些搜尋引擎只有提供網頁排名，並未提供相關分數。因此，僅有名次可供我們爲合併排序之參考。

我們把不同搜尋引擎但名次相同的網頁，視爲相同的地位。亦即，名次相同，其原始分數及相同，如此每一個搜尋引擎才能立於平等之地位，不因回覆的網頁數目不同，而有不同的原始分數。

4．實驗系統的建構

在本章中，我們將實際建構一套漸進式合併排序法的 Meta-Search 系統原型，取名爲 PIS（Personal Information Search），而整個系統原型的功能與系統介面，將分述如下。

在 PIS 的系統介面上，使用者可輸入查詢詞，執行系統搜尋動作。針對此一查詢詞，系統連至自動選定或使用者選定的幾個搜尋引擎，在搜尋引擎傳回網頁以後，再依據我們的漸進示合併排序法，系統重新再計算網頁分數，合併排序網頁，網頁分數高的排名在前，以便使用者能優先的瀏覽。

在系統介面上輸入查詢詞，按下"開始查詢"執行搜尋的動作。在 PIS 系統回覆網頁的動作完成後，按下"合併"鍵，PIS 系統即執行合併網頁的動作，繼而將每筆網頁的摘要屬性（網址、標題…等），按照最終分數排列，由大到小。使用者直接在網頁的網址上雙擊滑鼠，即可執行瀏覽器，以瀏覽該網頁。使用者完成瀏覽網頁的動作後，回到 PIS 系統介面，對於之前剛瀏覽過的網頁，在相關的欄位上，用滑鼠加以點選，完成相關網頁認定的工作。使用者亦可不點選相關欄位，如此系統將認爲此網頁爲不相關。系統接收了使用者的相關回饋後，即可進行搜尋引擎的權重計算。計算出新的搜尋引擎權重後，將搜尋引擎個別的權重利用資料庫記錄起來，以作爲該使用者下次查詢同樣的查詢詞時的參考。PIS 系統合併查詢的過程，以下我們將以實際的查詢例子來說明 。

在進入 PIS 系統之後，首先會要求使用者輸入使用者名稱及密碼，在使用者選擇了若干搜尋引擎並輸入"mp3"爲查詢詞後，PIS 隨即執行查詢的動作，其結果如圖 4.1，在功能表檔案選項下，執行合併或直接在系統介面上按下"合併"，PIS 即按照網頁的最終分數，重新合併排名網頁，如圖 4.2。使用者在將滑鼠指

標移至欲瀏覽的網頁之網路位址上，雙擊滑鼠左鍵，隨即啓動瀏覽器以瀏覽該網頁。在使用者瀏覽該網頁的動作完成後，回到 PIS 主系統，在被使用者瀏覽過的網頁前面，系統將標示"*"號，已表示該網頁已經被使用者瀏覽過了，如圖 4.3，若使用者認爲在該網頁能獲得他所需要的資料，在該網頁相關欄位的小方框點選一下，相關欄位的小方框將出現"v"的記號，以表示該網頁爲使用者認爲相關的網頁，如圖 4.3。若使用者認爲在該網頁不能獲得他所需要的資料，則不需做任何動作。此外，按下系統介面上的"下一頁按鍵"，將出現下 10 筆的網頁供使用者瀏覽。當使用者對系統完成網頁相關與否的回饋後，結束這次的查詢，系統開始計算每個搜尋引擎各自對於查詢詞"mp3"的權重，並記錄在我們爲使用者所建置的個人資料表中，以作爲下次再輸入"mp3'爲查詢詞時，合併排名網頁之用。

圖 4.1 執行查詢詞"mp3"之後的結果

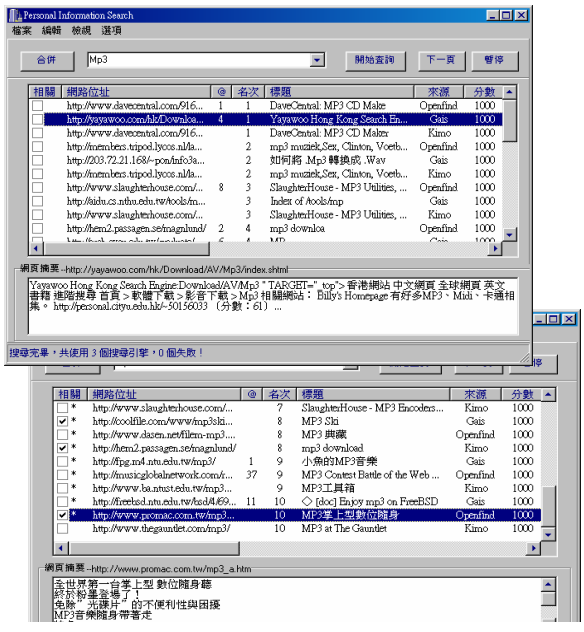

圖 4.2 執行合併之後的結果

圖 4.3 點選相關的網頁

5．結論以及未來的研究方向

在本論文中，我們針對 Meta-Search 之合併排序過程，提出了一個漸進式合併排序法，來改善合併排序之後的精準度。我們也實際建置了一套使用進式合併排序法 Meta-Search 系統 PIS（Personal Information Search），經過實際的測試，我們的漸進式合併排序法的確大大地提高了精準度。再則，在我們的實驗中，我們發現到有精準度無法上升的現象，其原因是在搜尋引擎所提供的原始網頁排名裡，使用者認爲是相關的網頁，都是排名於後，而排名在前的幾乎皆爲一些不相關的網頁。一般而言，搜尋引擎檢索網頁排名的技術大都是根據資訊檢索理論（Information Retrieval）[14]的方法，而我們在使用漸進式合併排序法時，並不重新檢索網頁中查詢詞出現次數，以計算網頁分數。因爲搜尋引擎已經檢索過了，所以我們將取回的網頁，若利用資訊檢索重新計算排名，所得到的網頁排名仍是相近的，而且重新檢索也將浪費太多時間和資源了。因此我們對於某單一個搜尋引擎所回覆的網頁，其自身排名在後的網頁，經過漸進式合併排序法，並不會改變搜尋引擎其自身內部的網頁排名。所以儘管我們了解對於某使用者，他所認爲相關的網頁排名集中於後時，我們仍然使用搜尋引擎其自身所提供之排名，來作合併排序，導致精準度因而無法提昇。對於以上這些現象發生時，未來我們將研究如何適當地調整搜尋引擎之權重，以達到最佳的合併結果。

參考文獻

[1] 卜小蝶（民 85）：圖書資訊檢索技術。台

北：文華

[2] 黃復光、何育琨（民 87）：一個使用在全球資訊網上之個人化智慧型搜尋引擎整合器。國立成功大學電機工程研究所碩士論文。

[3] 資訊人. Internet. 12 Jun. 1999. Available: http://www.inforia.com.tw/quest/index.asp

[4] Ask Jeeves Home Page. Internet. 11 Jun. 1998. Available: http://www.askjeeves. com/

[5] Dogpile Home Page. Internet. 16 Jun. 1999. Available: http://www.dogpile.com/

[6] Dreilinger, D., & Howe, A. E. （1997）. Experiences with Selecting Search Engines Using Metasearch. ACM Transaction on Information Systems, 15, 3, 195-222.

[7] Dreilinger, D. SavvySearch Home Page. Internet. 12 Jun. 1999. Available: http://www.savvysearch.com/

[8] Gauch, S., Wang, G., & Gomez, M.（1997）. Profusion*: Intelligent Fusion from Multiple, Distributed Search Engines *1. Journal of Universal Computing, Springe-Velag, 2（9）, 637-649.

[9] Gauch, S., & Wang, G. （1996）. Information fusion with Profusion. In Proceeding of the World Conference of the Web Society （WebNet'96）.

[10] Gravano, L., & Garcia-Molina, .H. （1997）. Merging Ranking from Heterogeneous Internet Sources. VLDB 1997, 196-205

[11] Highway 61 Home Page. Internet. 14 Jun. 1999.Available:http://www.highway61.com/

[12] Mamma: Mother of All Search Engines. Internet. 2 Jun. 1999. Available: http://www.mamma.com/

[13] MetaFind Home Page. Internet. 13 Jun. 1999. Available: http://www.askjeeves.com/

[14] Salton, G. （1989）. Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Reading, Mass.

[15] Selberg, E., & Etzioni, O. （1996）. Multi-Service Search and Comparison Using the MetaCrawer. Internet. 10 Jun. 1998. Available: http://www.w3.org/conferences /WWW4/Papers/169/.

[16] Verio Metasearch Home Page. Internet. 12 Jun. 1999. Available: http://search.verio.net/

[17] Yuwono, B., & Lee, D. L.（1996）. A World Wide Web Resource Database System. IEEE Transaction On Knowledge And Data Engineering, 8, 4, 548-554.

作者簡介

李建億

國立交通大學資訊科學博士，現爲台南師範學院資訊教育研究所助理教授，專長爲資訊檢索、資料探勘、電腦輔助教學系統

周明仁

現爲台南師範學院資訊教育研究所碩士班學生

蔡政容

現爲台南師範學院資訊教育研究所碩士班學生