逢　甲　大　學

資訊工程學系專題報告

以.NET 技術實作個人化 Internet 網頁搜尋服務

學生：鄭宗碩（四乙）

指導教授：劉安之

中華民國九十三年五月

目錄

圖目錄

表目錄

摘要

在網際網路的時代，搜尋引擎幾乎是人們賴以取得所需資訊的重要管道。 然而面對各大入口網站的搜尋引擎所產生的幾乎是數以萬計的資訊，如何在大量資訊中取得知識又是一項非常值得探討的議題。

Meta Search Engine 亦是搜尋引擎的一種。 它利用回收其他搜尋引擎的搜尋結果來做處理並加以彙整成新的搜尋結果，提供使用者另類的資訊來源。

在本專題實驗將利用微軟.NET技術實作一個 Meta Search Engine，針對目前國內四個極具盛名的入口網站，GAIS（Global Area Information Servers)、Google 台灣網站、微軟 MSN 台灣網站及 Yahoo 奇摩，四家搜尋引擎所產生的結果加以彙整、分析，提供一份新的搜尋結果給使用者。 並針對使用者在使用時所遇到的一些問題加以提出解決之道，讓使用者能夠省下更多的時間並獲得更多的資訊。

以下為本專題實驗報告各章節內容扼要的簡介：

第一章、導論。 了解搜尋引擎還能為使用者提供什麼功能。 介紹目前搜尋引擎的種類，以及本專題實驗所採用的四家搜尋引

擎的特點比較。　並描述使用者在使用搜尋引擎時所遭遇到的狀況。

第二章、系統架構與資料處理流程介紹。　介紹本搜尋引擎的架構及所提供的功能，還有取得資料之後彙整出新結果的處理流程。

第三章、系統實作介紹。　介紹本專題實驗所實作的各項功能及原理；並利用測試搜尋字串集合來檢測本搜尋引擎在大量使用者使用之下所能達到的最大效能平均值。

第四章、討論與未來展望。　並針對本專題實驗報告作檢討，提出在開發的過程中所遭遇的困難以及將來能夠再繼續改進的地方。

第一章、 導論

使用者在使用搜尋引擎時，常常有些資料在某個搜尋引擎的搜尋結果中出現，有些資料則在另外的搜尋引擎的搜尋結果中出現。 如果兩部分的資料對使用者來說都有必然的重要性，有沒有一種方法能夠整合這些搜尋的結果，然後重新整現一份新的結果，更能夠滿足使用者的要求呢？

當使用者搜尋一份結果時，嘗試點選了搜尋結果的連結，卻發現竟是無法顯示網頁或者網頁不存在的錯誤訊息，是否能夠有辦法事先知道哪些連結可用，哪些連結無法使用，避免因為點選後因為等待所付出的時間代價呢？

我們將在本專題實驗中實作這個整合搜尋服務。 提供使用者一份整合的搜尋結果，並能事先檢查搜尋結果目標網頁的狀態，藉此讓這些搜尋服務時能夠更加滿足使用者的需求。

1.1 搜尋引擎種類的簡介：

相較於歷史上印刷產業快速發展的年代造成的資訊爆炸，如今網際網路在短時間內的驚人發展，彷彿也開始進行一場另類的資訊大爆炸。 面對與日俱增的 Internet 網頁以及各種資源，搜尋引擎似乎是

人們所仰賴及使用來獲得所需資訊的工具。

一般來說搜尋引擎共分成以下五類：Internet 網頁內文全文搜尋、目錄式搜尋、網站網頁內文全文檢索、網站版代理搜尋及個人版代理搜尋*（如表 1.1）。

本專題實驗實作的搜尋引擎屬於「個人版代理搜尋」，即是摘要中所說明的 Meta Search Engine。 主要運作模式為將使用者欲搜尋字串的 Request 送到各來源搜尋引擎, 並取得 Response 回來的搜尋結果, 然後取得其中有意義的資訊並加以彙整，再利用 3.4 節所介紹的排名規則產生一份整理過的搜尋結果，提供使用者對於搜尋結果的另外一個選擇。 也利用使用者的點選習慣來調整排名順序，希望藉以提高搜尋結果的準確度。

種類	優點	缺點	代表網站
Internet 網頁內文全文搜尋	搜尋範圍擴及整個 Internet	查詢到的資料筆數太多	GAIS、Openfind
目錄式搜尋	以網站分類的觀念，幫助使用者歸納站台	無法立即更新，時效性差	Yahoo、Yahoo奇摩、Yam、MSN
網站網頁內文全文檢索	以特定網站為搜尋範圍，檢索資料的精確度高	無法提供範圍外的功能查詢	GAIS、龍捲風
網站版代理搜尋	可以同時使用多個搜尋引擎查詢	使用者費用	MetaCrawler、Inference Find
*個人版代理搜尋	除了網站版代理搜尋的功能外，使用者可以針對這些站台，加以分類，配合自己的查詢習慣	使用者費用	龍捲風

表 1.1　五種搜尋引擎種類分類，優缺點比較及代表性搜尋引擎

資料來源：

http://www.ctjh.tpc.edu.tw/www/center/computer/www/content12-c32.htm

1.2 來源搜尋引擎介紹：

本專題實驗所採用的四個搜尋引擎來源 － 分別為 GAIS、Google

台灣、MSN 台灣及 Yahoo 奇摩 － 都有各自的特色並提供的不同的附加功能。

1.2.1來源搜尋引擎進階功能比較：

以下則將進一步了解這四大搜尋引擎能夠提供哪些更進階的搜尋功能：(參考下表 1.2)

	GAIS	Google 台灣	MSN 台灣	Yahoo 奇摩
字串	ˇ	ˇ	ˇ	ˇ
輸出結果筆數	ˇ	ˇ		ˇ
查詢網頁語言		ˇ	ˇ	ˇ
更新日期選擇		ˇ		ˇ
字詞位置			ˇ	ˇ
網域搜尋	ˇ	ˇ	ˇ	ˇ
完整字句	ˇ	ˇ		ˇ
任一字句	ˇ	ˇ		ˇ
不包含字詞	ˇ	ˇ		ˇ
搜尋特定網頁		ˇ		ˇ
結果排序	ˇ		ˇ	
網頁地區		ˇ	ˇ	ˇ
檔案類型	ˇ		ˇ	
文件目錄層級	ˇ		ˇ	
單網域單結果			ˇ	
找圖	ˇ	ˇ		
找新聞	ˇ			
找新聞群組		ˇ		
找產品				ˇ
網站組群功能	ˇ			

表 1.2　四家搜尋引擎所提供的進階搜尋功能選項比較圖

1.2.2來源搜尋引擎搜尋結果一覽：

首先讓我們了解所採用的搜尋引擎搜尋完畢之後所提供的資訊有哪些。 利於我們取得其中我們所需要的資訊。

本實驗中我們將從搜尋結果取得的資訊有：搜尋結果於單一搜尋引擎的排名順序，搜尋結果網頁內容的 Title 內容、URL 網址及摘要內容。

1.2.2.1 GAIS 搜尋引擎搜尋結果顯示：

GAIS 引擎的搜尋結果所提供的資訊有搜尋排名順序、搜尋結果網頁內容的 Title 內容、URL 網址及摘要內容。 此外還提供了「暫存網頁」、「相關網頁」等連結以及網頁大小及維護更新日期。 此外也提供了「網頁內容別名」的額外資訊更利於搜尋的比對。

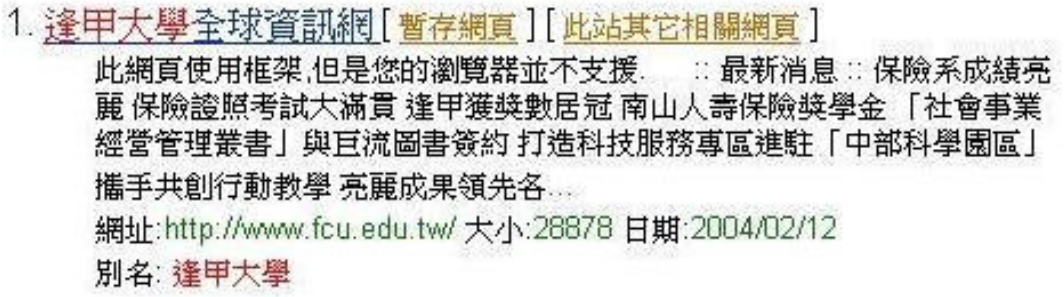
1. 逢甲大學全球資訊網[暫存網頁][此站其它相關網頁]
此網頁使用框架,但是您的瀏覽器並不支援. :: 最新消息 :: 保險系成績亮麗 保險證照考試大滿貫 逢甲獲獎數居冠 南山人壽保險獎學金 「社會事業經營管理叢書」與巨流圖書簽約 打造科技服務專區進駐「中部科學園區」攜手共創行動教學 亮麗成果領先各...
網址:http://www.fcu.edu.tw/ 大小:28878 日期:2004/02/12
別名: 逢甲大學

圖 1.1 以「逢甲大學」為關鍵字搜尋 GAIS 搜尋引擎所得結果情況一

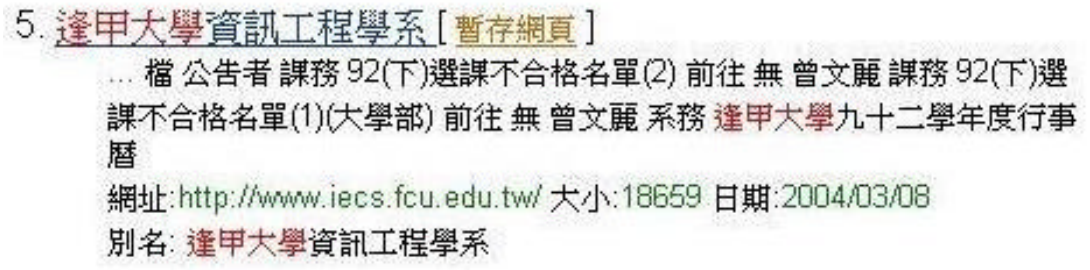
5. 逢甲大學資訊工程學系 [暫存網頁]
... 檔 公告者 課務 92(下)選課不合格名單(2) 前往 無 曾文麗 課務 92(下)選課不合格名單(1)(大學部) 前往 無 曾文麗 系務 逢甲大學九十二學年度行事曆
網址:http://www.iecs.fcu.edu.tw/ 大小:18659 日期:2004/03/08
別名: 逢甲大學資訊工程學系

圖 1.2　以「逢甲大學」為關鍵字搜尋 GAIS 搜尋引擎所得結果情況二

1.2.2.2　Google 台灣搜尋引擎搜尋結果顯示：

GAIS 引擎的搜尋結果所提供的資訊有搜尋結果網頁內容的 Title 內容、URL 網址及摘要內容。　此外還提供了「暫存網頁」、「相關網頁」連結、網頁大小及維護更新日期等資訊。　GAIS 引擎也提供了搜尋結果目標的檔案型態標記顯示，有一般 HTML 檔案（預設內容無標示）、[DOC]型態表示微軟 Word 檔、[PPT]形態表示微軟 PowerPoint 檔、[XLS]形態表示微軟 Excel 檔以及 [PDF]形態表示 Acrobat PDF 檔。


逢甲大學全球資訊網
此網頁使用框架,但是您的瀏覽器並不支援.
www.fcu.edu.tw/ - 1k - 頁庫存檔 - 類似網頁

逢甲大學通識教育網頁
www.fcu.edu.tw/support/academic/education/ - 2k - 頁庫存檔 - 類似網頁
[www.fcu.edu.tw 的其它相關資訊]

圖 1.3　以「逢甲大學」為關鍵字搜尋 Google 台灣所得結果情況一

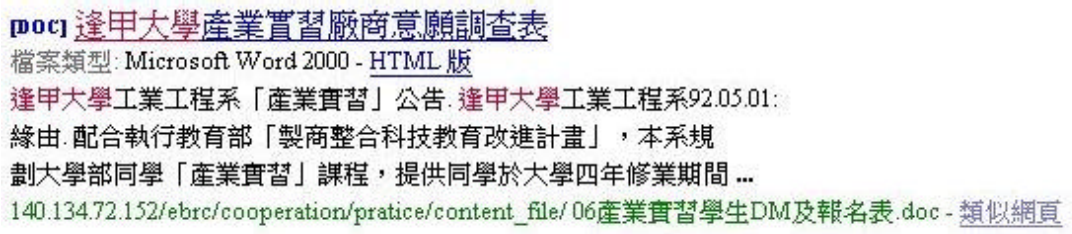
[DOC] 逢甲大學產業實習廠商意願調查表
檔案類型: Microsoft Word 2000 - HTML 版
逢甲大學工業工程系「產業實習」公告. 逢甲大學工業工程系92.05.01:
緣由. 配合執行教育部「製商整合科技教育改進計畫」，本系規
劃大學部同學「產業實習」課程，提供同學於大學四年修業期間 ...
140.134.72.152/ebrc/cooperation/pratice/content_file/ 06產業實習學生DM及報名表.doc - 類似網頁

圖 1.4　以「逢甲大學」為關鍵字搜尋 Google 台灣所得結果情況二

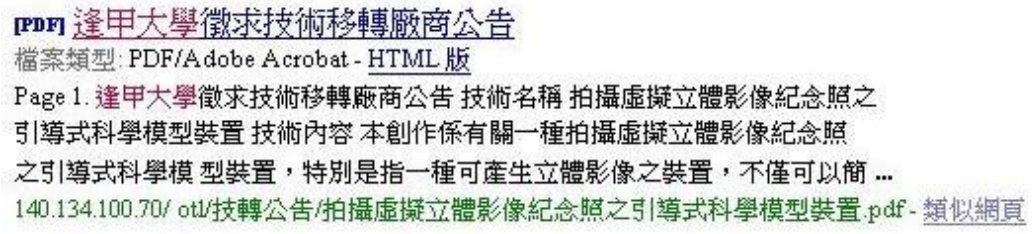
[PDF] 逢甲大學徵求技術移轉廠商公告
檔案類型: PDF/Adobe Acrobat - HTML 版
Page 1. 逢甲大學徵求技術移轉廠商公告 技術名稱 拍攝虛擬立體影像紀念照之
引導式科學模型裝置 技術內容 本創作係有關一種拍攝虛擬立體影像紀念照
之引導式科學模 型裝置，特別是指一種可產生立體影像之裝置，不僅可以簡 ...
140.134.100.70/ otl/技轉公告/拍攝虛擬立體影像紀念照之引導式科學模型裝置.pdf - 類似網頁

圖 1.5　以「逢甲大學」為關鍵字搜尋 Google 台灣所得結果情況三


[XLS] www.gale.com/tlist/sb5022.xls
類似網頁

圖 1.6　以「Excel」為關鍵字搜尋 Google 台灣所得結果情況一


[XLS] Harbinger
檔案類型: Microsoft Excel 5 - HTML 版
Harbinger. A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U. 1, Lawful Good. ——
— Melee ———. —— Ranged ———. Ranged. 2, #. Name. R. CR. Cmd. Diff. Size. Lvl.
Spd. AC. HP. Atk. Dmg. Atk. Dmg. Cost. Count. Points. Count. Points. 3, 1. Cleric of
Order. U. 4. 5. —. M. 4. 6. 16. 25. +4. 5. —. —. 24. 2. 48. 0. 4, 2. ...
www.spilth.org/dnd/minis/miniatures.xls - 類似網頁

圖 1.7　以「Excel」為關鍵字搜尋 Google 台灣所得結果情況二


[PPT] www.stic.gov.tw/fdb/pq/PQNextTraining.ppt
檔案類型: Microsoft Powerpoint - HTML 版
類似網頁

圖 1.8　以「簡報」為關鍵字搜尋 Google 台灣所得結果情況一


[PPT] 簽到
檔案類型: Microsoft Powerpoint 97 - HTML 版
...與國家高速網路中心聯繫，建請提供高速頻寬供視障者使用. 至淡江大學參加「如何整合視障資源」，並向教育部范次長. 巽綠作業務簡報。. 與國聲電台合辦「學習為視障者錄一本有聲Book～錄音志工. 培訓營」。. ...
163.23.207.15/ncue/htm/main/news_download/ Korea-report.ppt - 類似網頁

圖 1.9　以「簡報」為關鍵字搜尋 Google 台灣所得結果情況二

1.2.3.3　　MSN 台灣搜尋引擎搜尋結果顯示：

MSN 台灣的搜尋結果所提供的資訊有搜尋排名順序、搜尋結果網頁內容的 Title 內容、URL 網址及摘要內容。　除此之外還提供了搜尋結果目標的檔案型態標記顯示，有一般 HTML 檔案（預設內容無標示）、[Microsoft Word]型態表示微軟 Word 檔、[Microsoft Powerpoint]形態表示微軟 PowerPoint 檔、[Microsoft Excel]形態表示微軟 Excel 檔以及 [PDF/Adobe Acrobat]形態表示 Acrobat PDF 檔。

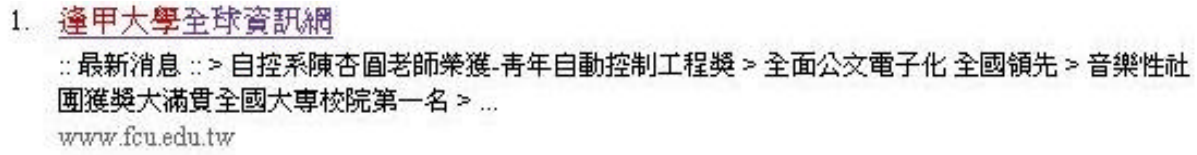
1. 逢甲大學全球資訊網
:: 最新消息 :: > 自控系陳杏圓老師榮獲-青年自動控制工程獎 > 全面公文電子化 全國領先 > 音樂性社團獲獎大滿貫全國大專校院第一名 > ...
www.fcu.edu.tw

圖 1.10　以「逢甲大學」為關鍵字搜尋 MSN 台灣搜尋引擎所得結果情況一

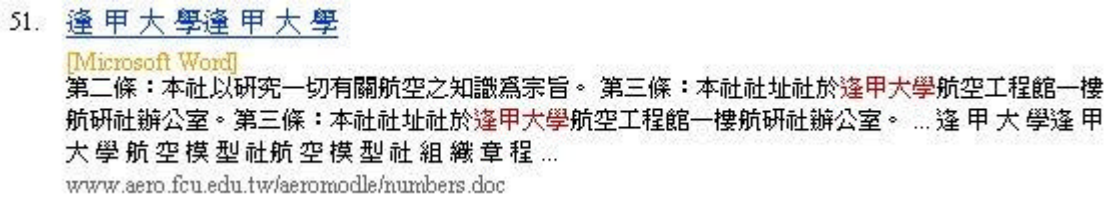
51. 逢甲大學逢甲大學
[Microsoft Word]
第二條：本社以研究一切有關航空之知識爲宗旨。 第三條：本社社址社於逢甲大學航空工程館一樓航研社辦公室。第三條：本社社址社於逢甲大學航空工程館一樓航研社辦公室。 ... 逢甲大學逢甲大學航空模型社航空模型社組織章程 ...
www.aero.fcu.edu.tw/aeromodle/numbers.doc

圖 1.11 以「逢甲大學」為關鍵字搜尋 MSN 台灣引擎所得結果情況二


132. 逢甲大學資訊工程研究所碩士班碩士論文
[PDF/Adobe Acrobat]
... Ad-hoc Wireless Networks 逢甲大學e-Thesys(91 學年度) 逢甲大學資訊工程研究所碩士班 ...
www.mclab.iecs.fcu.edu.tw/thesis/%AAL%A8%D8%BBT.pdf

圖 1.12 以「逢甲大學」為關鍵字搜尋 MSN 台灣引擎所得結果情況三


448. http://www.iaalab.ncku.edu.tw/cac/92%A5%C1%AF%E8%B7%A7%BD%D7%A6%A8%C1Z.xls
[Microsoft Excel]
... 9 林柏志 逢甲大學 航空系 二 64 46 ... 曹仕翰 逢甲大學 航空系 二 82 42 ...
www.iaalab.ncku.edu.tw/cac/92%A5%C1%AF%E8%B7%A7%BD%D7%A6%A8%C1Z.xls

圖 1.13 以「逢甲大學」為關鍵字搜尋 MSN 台灣引擎所得結果情況四

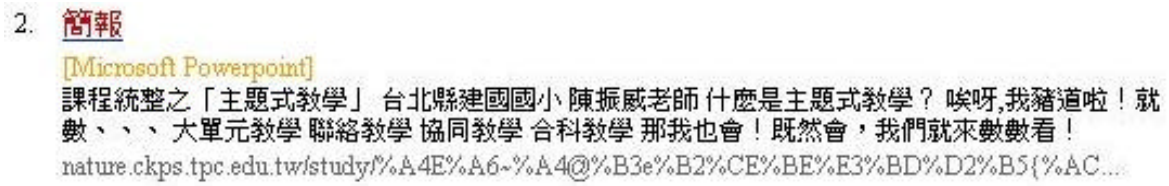

圖 1.14　以「簡報」為關鍵字搜尋 MSN 台灣引擎所得結果情況一

1.2.3.4　Yahoo TW 搜尋引擎搜尋結果顯示：

GAIS 引擎的搜尋結果所提供的資訊有搜尋結果網頁內容的 Title 內容、URL 網址及摘要內容。　此外還提供了「庫存頁面」、「同類網頁」連結、網頁大小、維護更新日期等資訊。

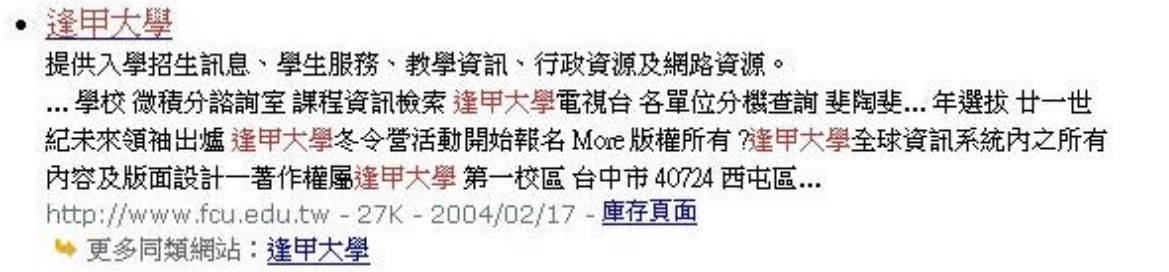

圖 1.15　以「逢甲大學」為關鍵字搜尋 Yahoo 台灣引擎所得結果情況

以上為本專題實驗所採用的四家來源搜尋引擎的搜尋結果。　先了解搜尋結果所有可能出現的狀況後，才能針對取回的資料作處理，取出我們所需要的有意義的資訊。

第二章、　系統架構與資料處理流程

2.1 系統介面及功能介紹：

以下將一一介紹我們所實作的系統使用介面以及該控制項所提供的功能：

2.1.1　一般搜尋介面：

如圖 2.1 所示，這部份依序包含了關鍵字串輸入盒、開始搜尋按鈕、搜尋深度選擇、顯示筆數選擇、搜尋結果網站狀態檢視及顯示搜尋結果分析資訊。

其中搜尋深度選擇功能的意義是指從每個搜尋引擎所回收的資料量。　當設定的深度越深，將可取得越多的資訊，但相對地會需要花費更多的時間。　詳細的花費時間平均請參考 3.6 節－實際效能檢測的結果。

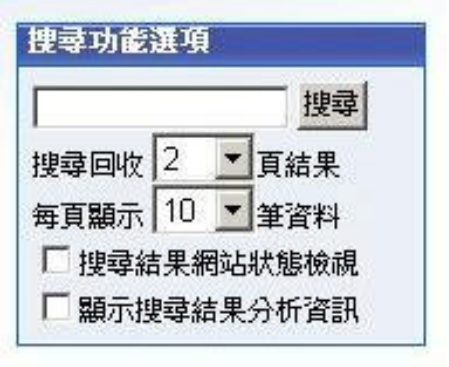

圖 2.1　搜尋功能選項

2.1.2 一般搜尋結果：

在一般的搜尋結果當中，每次的搜尋都會顯示搜尋結果的排名、Title 內容以及摘要內容。 此外也會將該連結的 URL 設為超連結供使用者直接點選。 並顯示本次搜尋所花費的時間，以秒為單位（如圖 2.2 所示）。

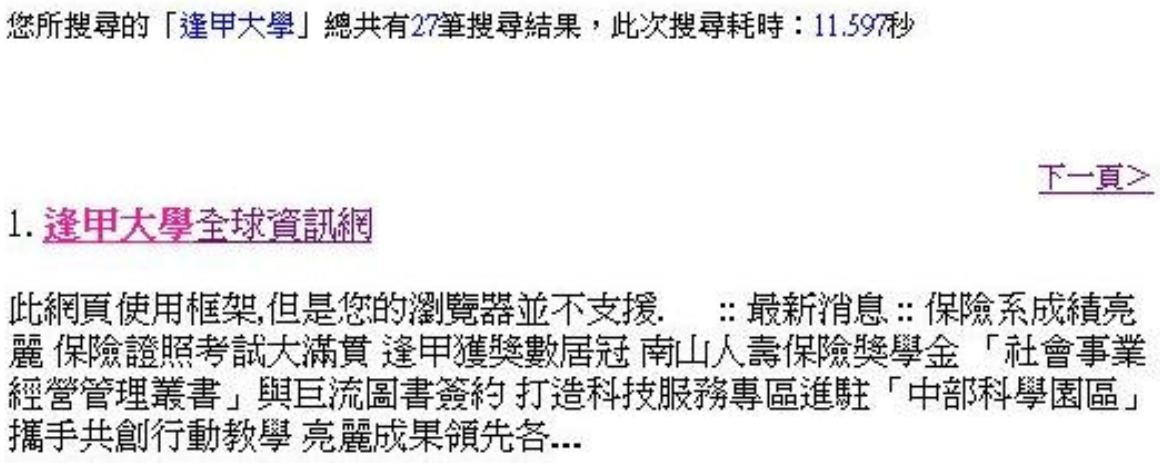

圖 2.2 一般搜尋的結果

2.1.3 熱門搜尋介面：

在這個控制項當中（如圖 2.3），我們利用紀錄使用者搜尋過的關鍵字串，並加以統計，找出前五名最常被搜尋的關鍵字串，顯示為熱

門搜尋。　其他使用者若是對這熱門搜尋的字串有興趣的，便可直接點選，從站內的資料庫中找到搜尋的結果，而不需要再送出要求給其他搜尋引擎並花費時間等待結果回收並整合。

圖 2.3　熱門搜尋

2.1.4　檢視搜尋結果連結狀態：

搜尋結果網站狀態檢視功能則是會自動偵測搜尋結果的各個連結目前的狀態。　若是搜尋結果的連結是可以連通的，則在顯示結果中顯示如圖 2.4 的正大拇指符號；反之則出現 2.5 的反大拇指符號。　藉此以圖形化活潑的風格告知使用者目前該連結的狀態：

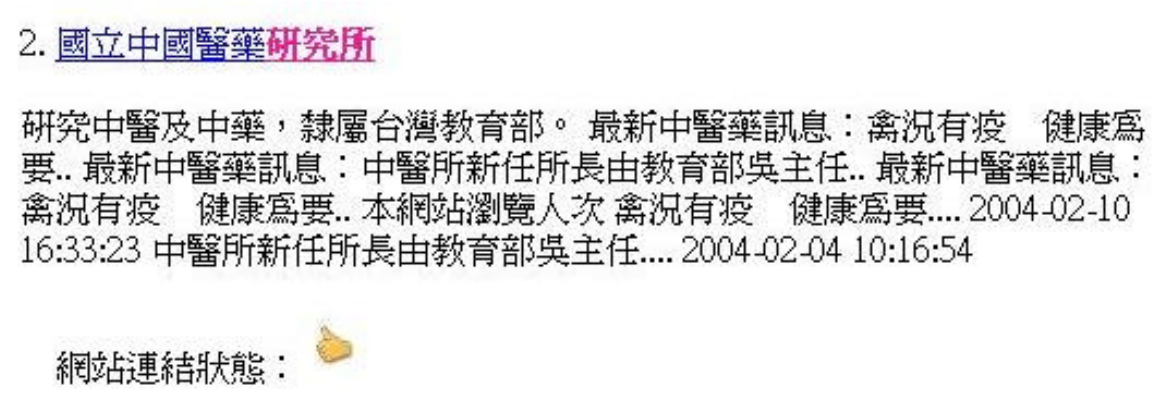

圖 2.4　連結正常且開啟速度小於 10 秒的搜尋結果網頁

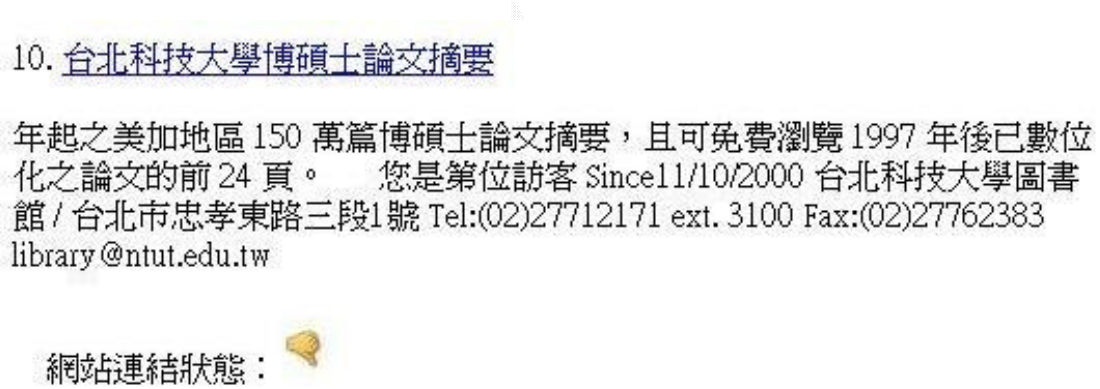

圖 2.5　無法連結或者開啟速度太慢的搜尋結果網頁

2.1.5顯示排名分析結果：

顯示搜尋結果分析資訊的選項將會在結果顯示中顯示該連結在四個來源搜尋引擎中排名的位置，如下圖圖 2.6 顯示。　提供使用者了解該搜尋結果在各個不同的搜尋引擎中所排列的重要性。

圖 2.6　顯示排名分析搜尋結果

2.2 搜尋流程介紹：

搜尋流程請參照圖 2.7 。　如同一般搜尋引擎，就是等待使用者輸入搜尋字串然後做搜尋。　為了避免每次使用者輸入搜尋字串之後都得等候到其他搜尋引擎搜尋的結果，我們會將每一次的搜尋結果先存放在本地伺服器端。　則每次使用者在搜尋時我們會先檢索站內是否已有搜尋的結果，如果是已有資料存在，我們則直接從資料庫輸出，否則送出 Request 到其他外部的搜尋引擎搜尋並取得結果。　回收結果之後再存在本地伺服器資料庫並輸出給使用者。

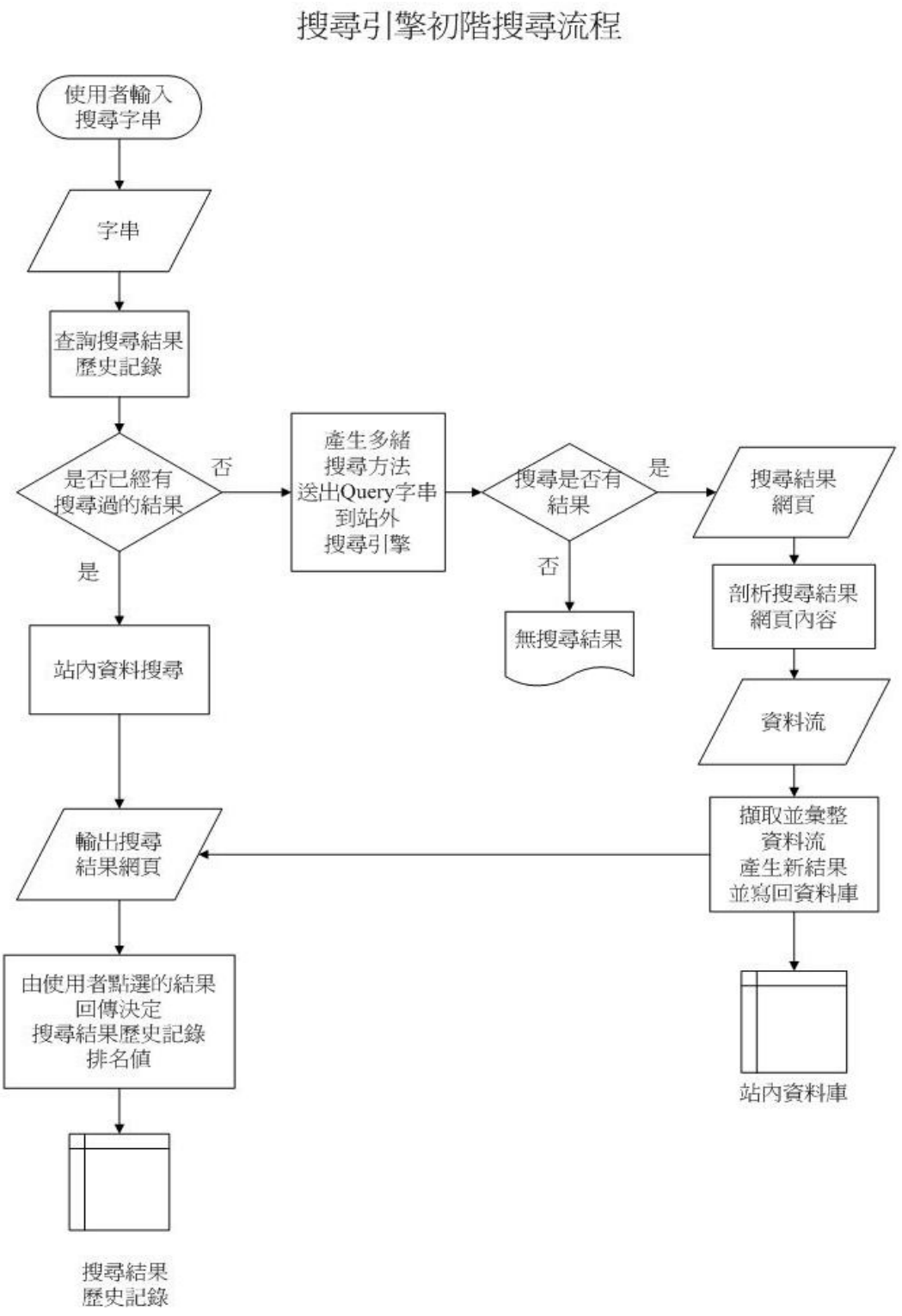

圖 2.7　本專題實驗搜尋引擎搜尋流程圖

2.3 搜尋結果排名方法介紹：

由於搜尋結果是由四個來源搜尋引擎的結果所彙總起來的，每一個搜尋結果在四個來源搜尋引擎其中之一都有其各自的排名順序，所以排名的方式也就勢必需要有所調整。

因此我們提供了簡單的排名方法，也就是最簡單的平均法。每一個搜尋結果都會有四個排名值，也就是以該連結之於搜尋關鍵字在 GAIS、Google 台灣、MSN 台灣及 Yahoo 奇摩各自的排名值。

而當有一個連結只出現在 GAISGAIS、Google 台灣及 MSN 台灣的搜尋結果，則將該連結的 Yahoo 奇摩排名值設為本次搜尋中 Yahoo 奇摩搜尋結果總數再加一計算。

以下為兩搜尋結果 A 與 B。 以 A_1、A_2、A_3、A_4分別代表 A 結果於本次搜尋出現在四個搜尋引擎的排名，數值越低表示排名越前面；B_1、B_2、B_3、B_4以此類推。 另外以 M_1、M_2、M_3、M_4分別代表四個搜尋結果的總數。

假設一函數 f(x)的值表示某 x 連結在一次搜尋中的鍵值，以下為

A與B結果的 f(x)值：

$$f(A) = \frac{A_1 + A_2 + A_3 + A_4}{4} \qquad f(B) = \frac{B_1 + B_2 + B_3 + B_4}{4}$$

則兩連結的相對排名順序以 f(A)及 f(B)為排序依據，越低的值具有較高的排名。 因此我們能夠得到一個結論：當一個結果出現的次數越高，會有較低的平均值，也將能得到較高的排名。

又以 M1、M2、M3、M4 當作四個搜尋引擎各自搜尋結果的全部個數。假設現在 A 的結果並沒有出現在第四個搜尋引擎的結果中, 則將 A4 設為（M4 + 1 ）

$$f(A) = \frac{A1 + A2 + A3 + (M4 + 1)}{4} \qquad f(B) = \frac{B1 + B2 + B3 + B4}{4}$$

若 B 的結果有出現在第四個搜尋引擎的結果中, 則 (M4 + 1) 必然大於 B4。 也因為這樣會使得 f (A) 變大，依照之前所訂的排名規格，A 連結的排名將會往後拉。 依照我們所訂定的規則，將會使得出現在四個搜尋引擎的搜尋結果中越多次的連結，將能夠獲得較好的排名。

2.4 資料表設計：

由於最初所設定的搜尋引擎種類是實作 meta search engine，所以在資料表的設計上我們採取如圖 2.8 的設計：

圖 2.8　資料表設計

圖 2.8 的資料表中，搜尋記錄資料表所記錄的是使用者所搜尋過的字串、搜尋時間及被搜尋的次數。　其中我們可以利用被搜尋的次數來判定熱門搜尋字串的結果。

而搜尋結果記錄資料表所記錄的是使用者所搜尋的搜尋字串及所

取得的每一筆資料內容：包含了 URL 位置、各家搜尋引擎中出現的排名數、網頁的鏢頭內容及網頁的摘要內容顯示。　這部份是最後輸出時所顯示的內容。

使用者個人資料夾資料表所記錄的是使用者所搜尋過的關鍵字串以及搜尋結果中對使用者自己本身有用的資訊，由使用者自己選取記錄起來，可供下次需要時快速點選。

第三章、　系統實作介紹

3.1 取得外站搜尋結果資訊：

Meta search engine 實作的第一步就是先收集其他搜尋引擎的搜尋結果。　我們所利用的方法是先了解四個來源搜尋引擎的 URL 參數及編碼方式，如此一來我們就可以送出 URL 的 Request 來取得串流（stream）型態的資料流。　再透過字串剖析的方式來取得我們要的資訊。

3.1.1　送出搜尋 Request

我們在利用.NET Framework的HttpWebRequest物件送出URL查詢字串。 URL查詢字串包含了來源搜尋引擎的網址以及編碼過的使用者搜尋字串， 亦或者包含其他代表不同功能的參數屬性。

首先我們先來了解四個來源搜尋引擎的URL以及其參數的意義。

- GAIS搜尋引擎：

http://gais.cs.ccu.edu.tw/cgi-bin/GAIS30/gaisweb.cgi?q="使用者所要搜尋的字串" & p= "起始面數" & u="zh-TW(表台灣地區)"

- Google台灣搜尋引擎：

http://www.google.com.tw/search?q="使用者所要搜尋的字串" & start= "起始面數" & hl="zh-TW(表台灣地區)"。

- MSN台灣搜尋引擎：

http://search.msn.com.tw/advresults.aspx?q="使用者所要搜尋的字串"。

- Yahoo奇摩搜尋引擎：

http://tw.search.yahoo.com/search/kimo?p="使用者所要搜尋的字串"。

此外，為了符合來源搜尋引擎的所能接受的使用者搜尋字串，我們必須將伴隨著 URL 送出的使用者搜尋字串加以編碼。 在此我們利用.NET Framework 的 HttpUtility 物件的 UrlEncode 方法來作編碼的動作。 編碼方式常見的有兩種：UTF-8 (數字代碼 65001)及 Big-5 (數字代碼 950)。 其中 Google 台灣及 MSN 台灣採用的 URL 編碼是 UTF-8；而 GAIS 及 Yahoo 奇摩採用的 URL 編碼則為 Big-5。

在實作上，我們建了一個 SendQueryAndGetResponse ( ) 副程式規格如下：

```
protected string SendQueryAndGetResponse
                                (string QueryURL,int
                                usedencode)
```

其中 QueryURL 所代表的是四個來源搜尋引擎的 URL 以及其參數。usedencode 則代表是四個來源搜尋引擎的所採用的編碼。 並將所輸入的 URL 當作參數然後利用 HttpWebRequest 物件送出 URL，再利用 HttpWebResponse 物件取得 HttpWebRequest 物件所獲得的回應。

```
//建立 Uri 物件代表所要檢查的 URL 位置
Uri HttpSite = new Uri(QueryURL);
//建立 HttpWebRequest 物件並送出 HttpSite 物件的 Request
HttpWebRequest userRequest = (HttpWebRequest)
```

```
WebRequest.Create(HttpSite);
//取得 HttpWebResponse 物件的結果
HttpWebResponse userResponse = (HttpWebResponse)
                                    userRequest.GetResponse();
```

如此就完成送出 Request 的動作了。

3.1.2 接收網頁串流及取回資料的編碼問題

接著我們必須將取得的所接收的資料串流，如下我們宣告 Stream 型態的物件來接收回傳的資料串流：

```
//宣告 Stream 型態的物件來接玻所取得的串流資料
Stream RespStream = userResponse.GetResponseStream();
```

接下來則是將 Stream 物件寫到暫存的 StringWriter 物件中, 等候後續處理。 接著我們從 StringWriter 物件中將資料一個 Byte 一個 Byte 給取出來到 Byte 陣列 StringData 中並做編碼，最後才傳回編碼過後的字串型態內容到 strContainer 回傳。

```
//宣告 Encoding 物件的 targetEncoding 並設定編碼為(65001)
Encoding targetEncoding;
targetEncoding = Encoding.GetEncoding(65001);
Char[] CharData = new char[40000];
String StringData = ConsoleWriter.ToString();
Byte[] ByteData = targetEncoding.GetBytes(StringData);
//進行 ByteData 陣列內容的編碼
targetEncoding.GetChars(ByteData,0,ByteData.Length,CharDa
ta,0);
//將編碼過後的字串型態加到 strContainer 回傳
```

```
string strContainer = new string(CharData);
```

3.2 利用 Regular Expression 處理串流資料：

從來源搜尋引擎端取得搜尋結果回來之後，是以串流型態來儲存的，而串流的內容其實就是大量的 HTML 原始碼及一些內碼。 一般我們是利用瀏覽器避開不需要的內碼而將 HTML 轉換成我們所熟悉的網頁內容，但在此我們則將要利用 Regular Expression 的表示式來進行剖析，跳過 HTML 的 Tag 及其他無用的內碼將網頁中對我們有意義的資料給擷取出來。

3.2.1 一覽搜尋結果來源 HTML 原始碼

現在我們先來了解從四個來源搜尋引擎取得的串流的 HTML 碼以及內碼的內容有哪些，再來進行下一步的剖析的準備工作。

首先我們先來看看從 GAIS 搜尋引擎取回的 HTML 原始碼及內碼內容，如下表（表 3.1）：

```
[空白]<dl>[跳行] [空白]<dt>[跳行]
[空白]<font color="#000000">[該搜尋結果於總搜尋結果之排
名]. </font><a href=" [搜尋結果目標 URL] ">
[搜尋結果目標網頁之 Title 內容]</a>[跳行] [空白]
 [ <a href="[站存網頁連結網址]" target="_blank"><font
color=#cc6600 size=-1>暫存網頁</font></a> ] [跳行] [跳行]
[空白] [ <a href=" [其他相關網頁連結網址]">
<font color=#cc6600 size=-1>此站其它相關網頁</font></a> ]
[跳行] [跳行] [空白] </dt>[跳行] [跳行] [空白] <dd>[跳行]
[空白] <font class=Abstract> [搜尋結果目標網頁之摘要內
容]</font>[跳行] [空白] </dd>[跳行] [空白] <dd>[跳行]
[空白]<font size="-1">[跳行]
[空白]網址:<font color="#008000">[搜尋結果目標
URL]</font>[跳行]
[空白]大小:<font color="#008000">[檔案大小以位元組為單
位]</font>[跳行]
[空白]日期:<font color="#008000">[最近更新日
期]</font>[跳行]
[空白]<!--     Score:<font color="#008000">[搜尋內含
值]</font> -->[跳行]
[空白]<!--     (##SCORE##) -->[跳行]
[空白]</font>[跳行] [空白]</dd>[跳行] [跳行] [空
白]<dd>[跳行]
[空白]<font class=Abstract>別名:  [搜尋結果目標網頁之別
名]</font>[跳行] [空白]</dd>[跳行] [跳行] [空白]</dl>[跳
行]
```

表 3.1　GAIS 搜尋結果 HTML 原始碼及內碼

```
<p>[跳行]
<a onmousedown="return clk(排名,this)" style="color: #00c;
font-family"
href="[搜尋結果目標 URL]">[跳行]
[搜尋結果目標網頁之 Title 內容]</a><br>[跳行]
<font size="-1">
[搜尋結果目標網頁之摘要內容] <b>...</b> <br>[跳行]
<font color="#008000">[搜尋結果目標 URL] – [資料大小] -[跳
行]
</font>[跳行]
<a class="fl" href="[庫存頁面網址連結]">[跳行]
頁庫存檔</a> -
<a class="fl" href="[同類網站網址連結]">[跳行]
類似網頁</a></font> </p>[跳行]
```

表 3.2　Google 台灣搜尋結果 HTML 原始碼及內碼

```
<ol style="MARGIN-TOP: 0px; MARGIN-BOTTOM: 0px">
<li><a class="t" href="[搜尋結果目標 URL]">
[搜尋結果目標網頁之 Title 內容]</a><div>
<span class="d">[搜尋結果目標網頁之摘要內容] </span><br>
<span class="u">[搜尋結果目標 URL]</span></div>
</li></ol>
```

表 3.3　MSN 台灣搜尋結果 HTML 原始碼及內碼

```
[空白]<LI><BIG><A href="" target="_blank">
[搜尋結果目標網頁之 Title 內容]</A></BIG>[跳行] [空白] [跳
行]
[空白]<br>[跳行]
[空白]  [搜尋結果目標網頁之摘要內容] <BR>
[跳行] [空白] [跳行]
[空白]<span class="wurl">[搜尋結果目標 URL] [跳行]
[空白]- 27K[跳行] [空白] [跳行]
[空白]- 2004/02/17[跳行]
[空白]-
<a href="[庫存頁面網址連結]">庫存頁面</a>[跳行]
[空白] [跳行] [空白]</span><br>[跳行] [空白]
<img src="http://tw.yimg.com/i/tw/search/lookin.gif"
width="15" height="9" alt="網站分類" hspace="3"><font
color="#666666">更多同類網站：
<A href="[同類網站網址連結]">[同類網站 Title 介
紹]</A></font><br>[跳行]
[空白]<br></LI>[跳行]
```

表 3.4　Yahoo 台灣搜尋結果 HTML 原始碼

3.2.2　產生對應的 C#語言 Regular Expression 規則

微軟.NET Framework 提供 Regular Expression 的物件提供使用者進行特定規則文字的剖析。　所在類別函式庫為 System.Text.RegularExpressions。　在此我們利用 Regular Expression 的字詞剖析能力來將取得的字串流（stream）中滿足我們所需要取得的資訊給擷取出來。　表為.NET Framework 提供的 Regular

Expression 的剖析規則：

依照.NET Framework 所提供的 Regular Expressions 的剖析規則，我們可以將 3.1 節的 GAIS 搜尋引擎的 HTML 結果轉成如下表（表）的 Regular Expression 規則：

```
<dl>\\s+<dt>\\s+<font\\s\\S+>\\d+\\.\\s</font><a[^>]*>\\s+(?<TIT
LE>.+)\\s*</a>\\s+(?:(\\[.+\\]\\s+)|(\\[.+\\]\\s+\\[.+href=\"(?<
Relative>[^\"]*)\".+\\]\\s+))\\s+</dt>\\s+<dd>\\s+<[^>]*>
(?<CONTENT>.+)</font>\\s+</dd>\\s+<dd>\\s+.+\\s+.+>(?<URL>http[^
<]*)</font>\\s+.+>(?<SIZE>\\d+)<.+\\s+.+>(?<DATE>\\d+/\\d+/\\d+)
<.+\\s+"
```

表 3.5　GAIS 轉換成.NET Framework 適用之 Regular Expression

```
<p\\s*class=g><a\\s*href=(?<URL>http[^>]*)>(?<TITLE>[^<]*)</a
>(?<CONTENT>[^<]*)<[^>]*>
```

圖 3.6　Google 台灣轉換成.NET Framework 適用之 Regular Expression

```
<li><a\\s*href=\"(?<URL>http[^\"]*)\"\\s*class=\"t\">(?<TITLE
>[^<]*)</a><div><span class=\"d\">(?<CONTENT>[^<]*)<br/>
```

圖 3.7　MSN 台灣轉換成.NET Framework 適用之 Regular Expression

```
<LI><BIG><A\\s*href=[^\\*]*\\*(?<URL>http://[^\\s*]*)\\s*targ
et=_blank> (?<TITLE>[^<]*)</A></BIG>(?:(\\s*)|(\\s+<[^>]*>
簡)|(\\s+<^>]*>英))\\s+(?<CONTENT>
(?:([^<]*<BR>)|(\\S+)))\\s+<span\\s*class=wurl>
```

圖 3.8　YAHOO 奇摩轉換成.NET Framework 適用之 Regular Expression

3.3 多執行緒於本專題實作應用介紹：

微軟.NET Framework提供了多執行緒的功能，方便使用者能夠同時執行多道指令。　這對於撰寫網路程式設計的我們更是受益匪淺。

在本專題實驗中，由於需要大量地向站外作送出Request及接收Response的動作，且在於網路狀況並不總是很穩定的情況下，多執行緒網路程式可以利用等待前一個Request或者Response的時間再來執行其他指令或者程式。

以下片段程式的功能是將送出查詢及接收回傳資料串的副程式加到程式的多緒執行。　先宣告一個Thread物件，然後引入所要執行的副程式，然後將該Thread物件啟動（ eg. t1.Start( ) ），然後再將Thread物件加入到多執行緒的佇列中執行（ eg. t1.Join( ) ）：

```
try
{
    //宣告ThreadStart物件啟動SearchProcess物件的搜尋函式多緒
    ThreadStart SearchGAIS = new ThreadStart(this.GaisQuery);
    ThreadStart SearchGoogle = new ThreadStart(this.GoogleQuery);
    ThreadStart SearchYahooTW = new ThreadStart(this.YahooTWQuery);
    ThreadStart SearchMSN = new ThreadStart(this.MSNQuery);
    //宣告t0、t1、t2、t3四個Thread物件
    Thread t0 = new Thread(SearchGAIS);
    Thread t1 = new Thread(SearchGoogle);
    Thread t2 = new Thread(SearchMSN);
    Thread t3 = new Thread(SearchYahooTW);
    //將t0、t1、t2、t3使作用
    t0.Start();
    t1.Start();
    t2.Start();
    t3.Start();
    //將t0、t1、t2、t3加入到執行行列
    t0.Join();
    t1.Join();
    t2.Join();
    t3.Join();
}
```

我們透過多緒執行能夠有效地利用傳送查詢及接收回傳資料這之間的等候時間，再送出其他的查詢。　如此一來可以省下大量的時間，也能進而加快整個程式執行的速度。

3.4 資料彙整：

由於取回來的資料包含了四家搜尋引擎的搜尋結果，這之中有許多可能是重複的連結，或者需要利用一些排名值的順序來做最後結果的排序，我們利用.NET Framework 提供的 ArrayList 物件來快速地儲

存回收的結果，並利用 ICompare 物件來快速依搜尋條件做搜尋。

3.4.1 利用 ArrayList 物件儲存自訂 SearchedEntry 物件

ArrayList 是.NET Framework 所提供用來儲存資料的資料結構。他的原理如同C語言的 Link List 能夠隨時新增資料內容，又能透過繼承的 ToArray( )將 ArrayList 的內容當成陣列利用 Index 來使用。

在實作中，我們宣告了兩個 ArrayList 物件 ResultContainer 及 filter。 ResultContainer 紀錄的是搜尋完之後全部的結果，而 filter 儲存的則是 ResultContainer 做完彙整的內容。

```
public ArrayList ResultContainer = new ArrayList();
public ArrayList filter = new ArrayList();
```

新增資料時只要利用 Add( )函式來新增資料內容即可。 新增的資料型態則依使用者個人的需求來新增不受限制，但同一個 ArrayList 內的資料型態必須一致。

3.4.2 實作 ICompare 介面來快速排序搜尋結果

接下來要介紹的是利用我們用來儲存資料內容的 SearchedEntry 類別繼承 ICompare 類別並實作排序函式。 如此一來我們能夠利用以

下如同 CompareTo( )函式及 SortBtURL( )函式只要定義排序的條件就能快速地排序一個資料結構的內容。

```
//與現行的物件互相比較，比較的物件為 SearchedEntry 型態物件
//的 intOrder 屬性內容
public int CompareTo(object obj)
{
    SearchedEntry se2;
    se2 = (SearchedEntry) obj;
    return intOrder.CompareTo(se2.intOrder);
}

//將現行的兩SearchedEntry型態物件的strHref屬性依照string
型態
//的排序比較方式做排序
public class SortByURL:IComparer
{
    public int Compare(object obj1,object obj2)
    {
        SearchedEntry test1 = (SearchedEntry) obj1;
        SearchedEntry test2 = (SearchedEntry) obj2;
        return
(string.Compare(test1.strHref,test2.strHref));
    }
}
```

最後我們只要如下地叫用即可快速地完成排序工作：

```
//所使用的排序規則為 CompareTo 所定義的比較方式
filter.Sort();

//所使用的排序規則為自訂的 SortByURL 所定義的比較方式
ResultContainer.Sort((IComparer)new
SearchedEntry.SortByURL());
```

3.5 偵測搜尋結果連結狀態功能介紹：

這項功能最主要的意義是在於利用.NET Framework 所提供的多執行緒功能幫助使用者先行檢測所搜尋的結果目標網頁連結是否為死連結；亦或者連線狀態很慢，幫使用者在結果中做標記供參考。

在實作的部分，當使用者勾「選檢視搜尋結果連結狀態」選項時，系統會將 IfCheckStatus 的值設為 true 並將儲存已經排序過搜尋結果的 ArrayList 物件內容的 strURL 屬性所代表的 URL 加到多執行緒的佇列中呼叫 CheckStatus( )副程式執行檢測目標網頁連結的狀態。

```
if (IfCheckStatus == true)
{
    for(index = 0;index<filter.Count;index++)
    {
        ThreadStart thtest = new
        ThreadStart(this.CheckStatus);
        Thread t0 = new Thread(thtest);
        t0.Start();
        t0.Join(10000);
    }
}
```

CheckStatus( )副程式所執行的功能其實與 3.1.1 節的功能是一樣的。 唯一不同的是我們利用 HttpWebResponse 物件所取得的

Response StatusCode 屬性的結果來判定狀態。 當回傳值為 OK 時，表示目前所檢查的 URL 連線狀態是成功的，反之則為死連結。

```
public void CheckStatus()
{
    //將代表目前所要檢查的 SearchEntry 物件實體化為 temp
    SearchedEntry temp = (SearchedEntry)
    this.filter[index];
    try
    {
        Uri HttpSite = new Uri(temp.strHref);
        HttpWebRequest myRequest = (HttpWebRequest)
        WebRequest.Create(HttpSite);
        HttpWebResponse myResponse = (HttpWebResponse)
        //取得 HttpWebResponse 物件的結果
        myRequest.GetResponse();
        if (myResponse.StatusCode.ToString() == "OK")
        //回傳值為 OK 則將代表狀態的屬性 StatusCode 設為 true
            temp.StatusCode = true;
        else
        //回傳值為 OK 則將代表狀態的屬性 StatusCode 設為 false
            temp.StatusCode = false;
    }
    catch
    {
        //例外發生則將代表狀態的屬性 StatusCode 設為 false
        temp.StatusCode = false;
    }
}
```

最後版面輸出時就會依照代表該搜尋結果的 StatusCode 來輸出如圖 2.4 及圖 2.5 一樣的輸出標記告知使用者目前網頁連結的狀態。

3.6 實際效能檢測：

因為 meta search engine 在絕大部分的時候，都必須到站外的其他搜尋引擎作搜尋獲得資訊，所以會比一般站內已具存有大量資料的搜尋引擎要花費更多的時間。 如果我們能夠了解所需要花費的平均時間，也就代表著搜引擎的執行效能，至少在等待時能夠知道還需要多少的時間。

要獲取搜尋的平均時間，我們利用從其他搜尋引擎獲得搜尋結果的方法 － 即送出 URL 參數的 Request 並獲得 Response 的網頁的方法，計時每次搜尋需要多少時間，最後取平均值來了解每次的平均搜尋時間。

以下是我們利用 2004 年 5 月 3 日，Yahoo 奇摩的熱門搜尋排行榜的前十大熱門搜尋作為實驗樣本，每次搜尋結果回收 2 到 6 頁結果，不啟動任何特殊功能所得的平均時間結果：

回收頁數 / 搜尋字串	2	3	4	5	6
reesion	1.125	1.75	1.781	2.844	3.922
琉璃仙境音樂網	1.469	1.234	1.89	2.782	5.969
瘋狂阿給	1.297	1.204	1.641	3.187	4.01
遊戲區	0.984	0.922	1.688	2.531	3.297
天堂官方網站	1.625	1.282	1.984	2.375	4.515
仙境傳說	0.781	0.688	2	2.828	4.282
希望	1.078	0.906	2.703	2.688	3.5
史萊姆	0.937	0.953	1.766	3.297	3.563
104	0.766	1.172	2.015	3.078	3.672
遊戲基地	1.031	1.375	2.297	2.843	3.64
音樂網	0.938	0.719	1.422	2.547	2.781
王心凌	0.75	0.766	1.593	2.125	3.313
天堂 2	0.844	0.797	1.485	2.172	2.953
中華電信	0.766	0.703	1.484	2.187	2.891
小遊戲	0.781	0.75	1.641	2.281	3.172
加總	14.391	14.471	25.749	37.484	52.308
平均	0.9594	0.964733	1.7166	2.498933	3.4872

表 3.9　搜尋效能實測數據列表

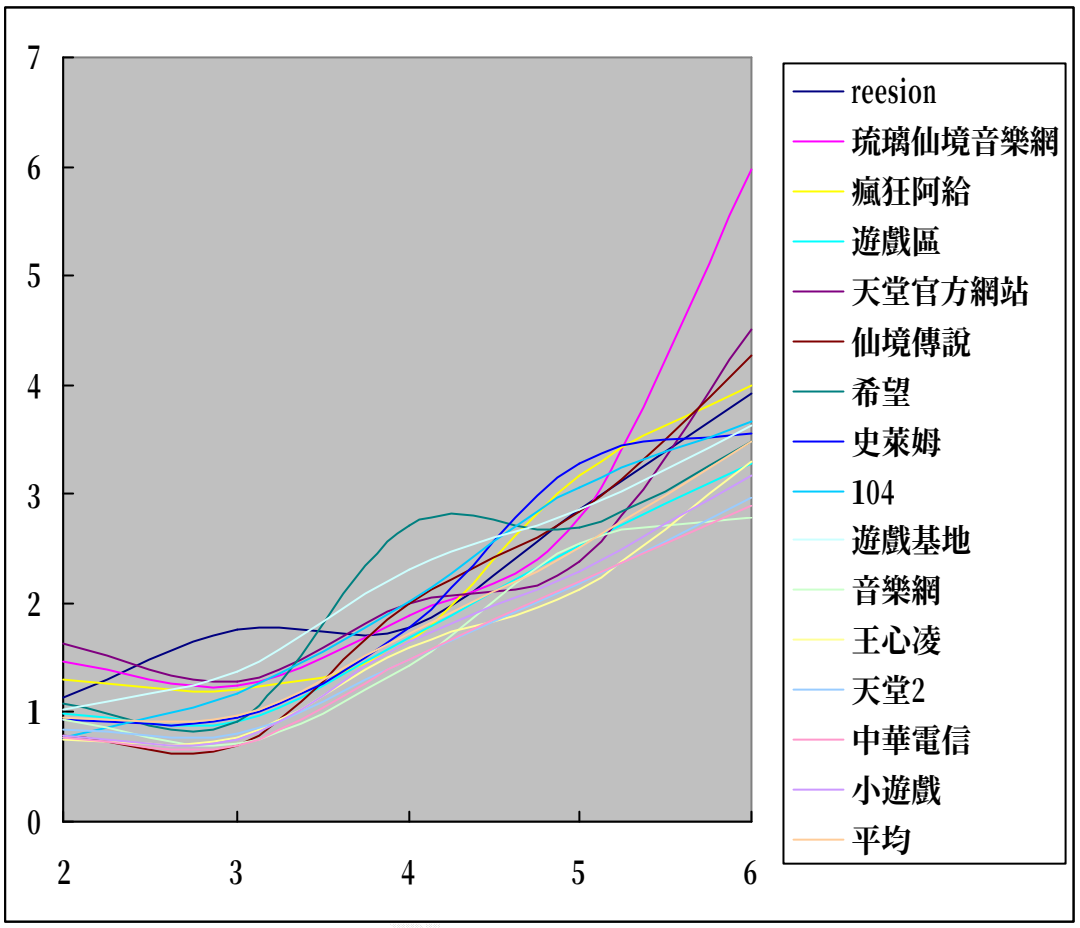

圖 3.1　搜尋效能實測數據曲折圖

以上為利用 2004 年 5 月 3 日 Yahoo 奇摩熱門關鍵字為測試搜尋字串樣本集合之結果，(X 軸座標表示回收的頁面數；Y 軸表示所花費的時間，單位為秒)。

由以上的數據可以了解到這必然的道理，當需要同時回收更多的搜尋結果頁面時，勢必是需要大量時間。　不過我們利用 .NET Framework 多執行緒的優點就能夠大幅減少等待時間。　這也是我們選擇以微軟.NET 技術來實作的理由之一。

第四章、結論

4.1 總結：

最後的成果，是完成一個能夠蒐集各家搜尋引擎搜尋結果的整合工具。 又能讓使用者花費些許時間來預先偵測搜尋結果網頁的狀態，並能查看搜尋結果在各家搜尋引擎的重要性排名。 目前這個搜尋工具已經以控制項的型態內嵌在本校.NET 社群網站內，另外亦提供使用者記錄自己搜尋過的結果，節省使用者再次搜尋所花費的時間。

4.2 遭遇到的困難：

在開發這個工具的過程當中，所遭遇到最大的困難就是在於語言的熟悉度。 由於第一次使用微軟的.NET 技術的 C#語言及 SQL Server 2000 來開程式，面對具有物件導向程式設計能力的 C#語言，著實花費了不少功夫在熟悉語言的使用上。

此外在剖析處理回傳資料時也花費了很大的時間，才漸漸地抓到利用 Regular Expression 剖析大量文字串的訣竅，將所要處理的大量文字串的規律式解析出來，進而利用其強大的能力來幫助我們完成這

項工作。

4.3 未來展望：

其實在開發的過程當中，發覺了幾項非常值得探討的問題方向。在以下將一一提出說明，以提供未來接續開發人員可以繼續鑽研開發的方向：

一、 搜尋字串的選擇： 其實目前的搜尋引擎所能做到的最基本的功能就是比對使用者輸入的字串內容，然後找出字串比對符合率最高的結果，並加以排名輸出。 然而選擇搜尋字串有很多小技巧，使用者能透過增加或者減少搜尋字串的內容來找到的幾乎完全不同的搜尋結果。 如果能夠找出這些被新增或者減少的關鍵子字串所搜尋的結果與使用者所需求的資訊的關係，或許能進而建議使用者改變搜尋字串已取得可能是更符合使用者需求的搜尋結果。

二、 取得並剖析不規則的網路資源內容： 在這個實驗中所取得的資料來源是非常有規律的搜尋結果內容。 但更積極的搜尋引擎應該要有能力自行在網際網路上探索，找尋資源並能剖析所找到的資源內容, 再加到自身的資料庫當中以供查詢。

這個部份由於無法善用 Regular Expression 的優點，所以必須利用堆疊來剖析網頁資源的 HTML 標籤，以取得網頁上有意義的資訊。

三、 資料的分類： 這其實是很大的一個學問。 由二的網路探索之後所取得的資源內容應該要能被分類儲存，才能更有效率地被搜尋。 目前的分類式搜尋引擎都是利用人工的方式來做分類。 將來若是本搜尋服務也能將鎖探索得到的資訊加以分類來儲存，在搜尋速度及準確度上將能更為有效率及準確。