

## 第四章

### 系統實作與實驗評估

#### 4.1 系統架構

圖 4.1 為本研究的系統架構圖，圖 4.2 為系統首頁。本研究將系統分成三大部分，一部分處理長期情境，一部分處理短期情境，最後一部分則處理長期情境與短期情境的結合。在長期情境的方面。為了搜集使用者與搜尋引擎互動的一切紀錄，我們實作了一個介面，稱為 Log Collector。從 Log Collector 蒐集回來的資料經由分析之後，由 Web Page Collector 去將當初搜尋引擎針對使用者查詢回傳的網頁都抓取回來。接著 HTML Parser 做語法分析以找出關鍵字，進而找出長期情境所記錄的主題關鍵字。

在短期情境方面，利用我們實作的介面與使用者作互動，同時記錄了目前使用者與本系統之間的互動過程，一樣透過 Web Page Collector、HTML Parser 做分析與處理，以產生符合短期情境的主題關鍵字。

最後，Search Behavior Miner 這個模組，包含處理 Filtering 與探勘兩大部分的程式，Filtering 主要必須去計算長期情境中各個 Query Session 的查詢關鍵字密度以及點選網頁密度；而探勘部分主要是實作探勘演算法，探勘出以往經驗使用者的關鍵字集合，以做為修正查詢的依據。

1. Log Collector：在這一個部份中，本研究實作了一個中介系統來處理與紀錄使用者與

搜尋引擎之間的一切互動過程。所謂中介系統是我們想提供一個介面系統介於使用者

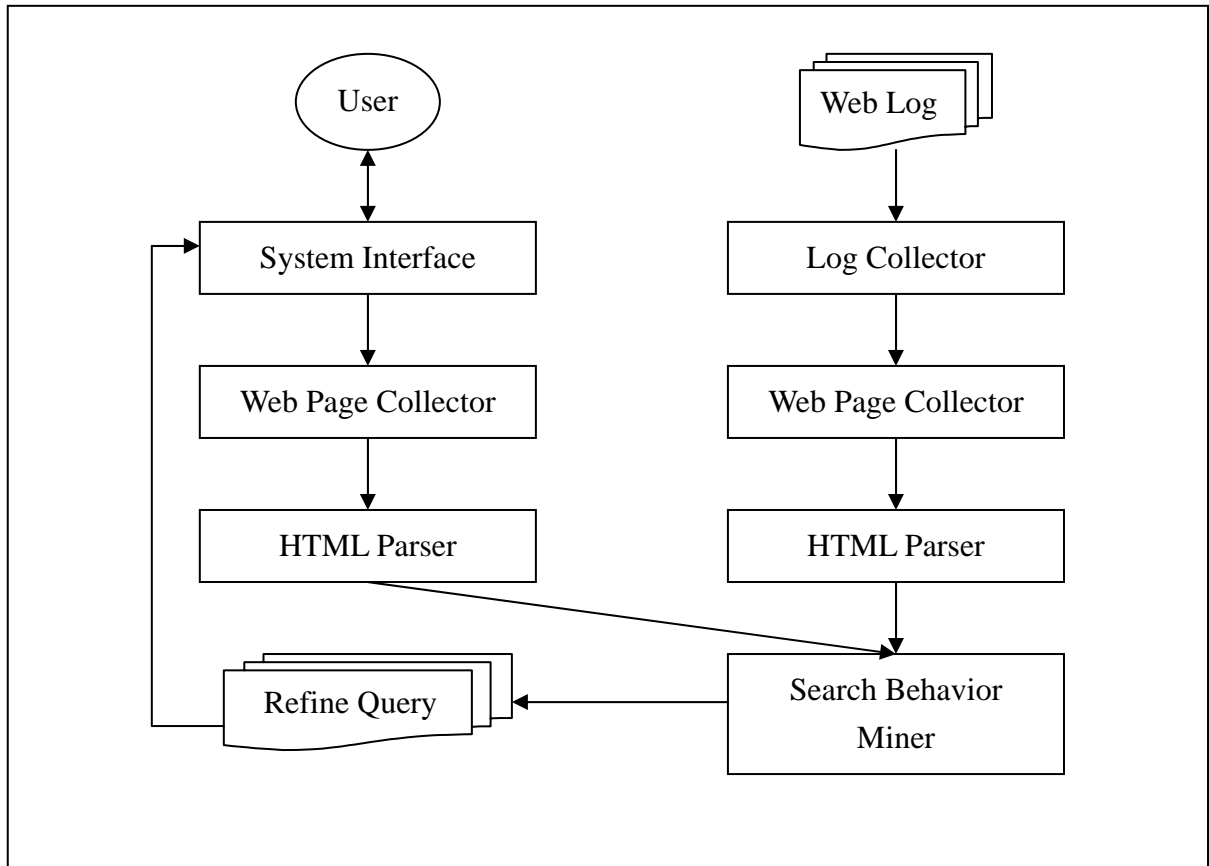


圖 4.1：系統架構圖。

與搜尋引擎之間，我們提供了讓使用者輸入關鍵字做查詢的服務，同時將使用者的關鍵字送交搜尋引擎做查詢的動作。接著再將搜尋引擎回傳的網頁由實作的介面輸出給使用者瀏覽且允許使用者做下一步點選瀏覽的動作。而這些使用者與搜尋引擎間的所有互動過程，本研究都會將其記錄下來，以供日後分析、處理之用。在搜尋引擎方面，我們選定 Google 來作為我們的實作機制，因此，我們利用了 Google SOAP Search API (<http://www.google.com/apis/>)來幫助我們達到這樣的目的。Google SOAP Search API 是

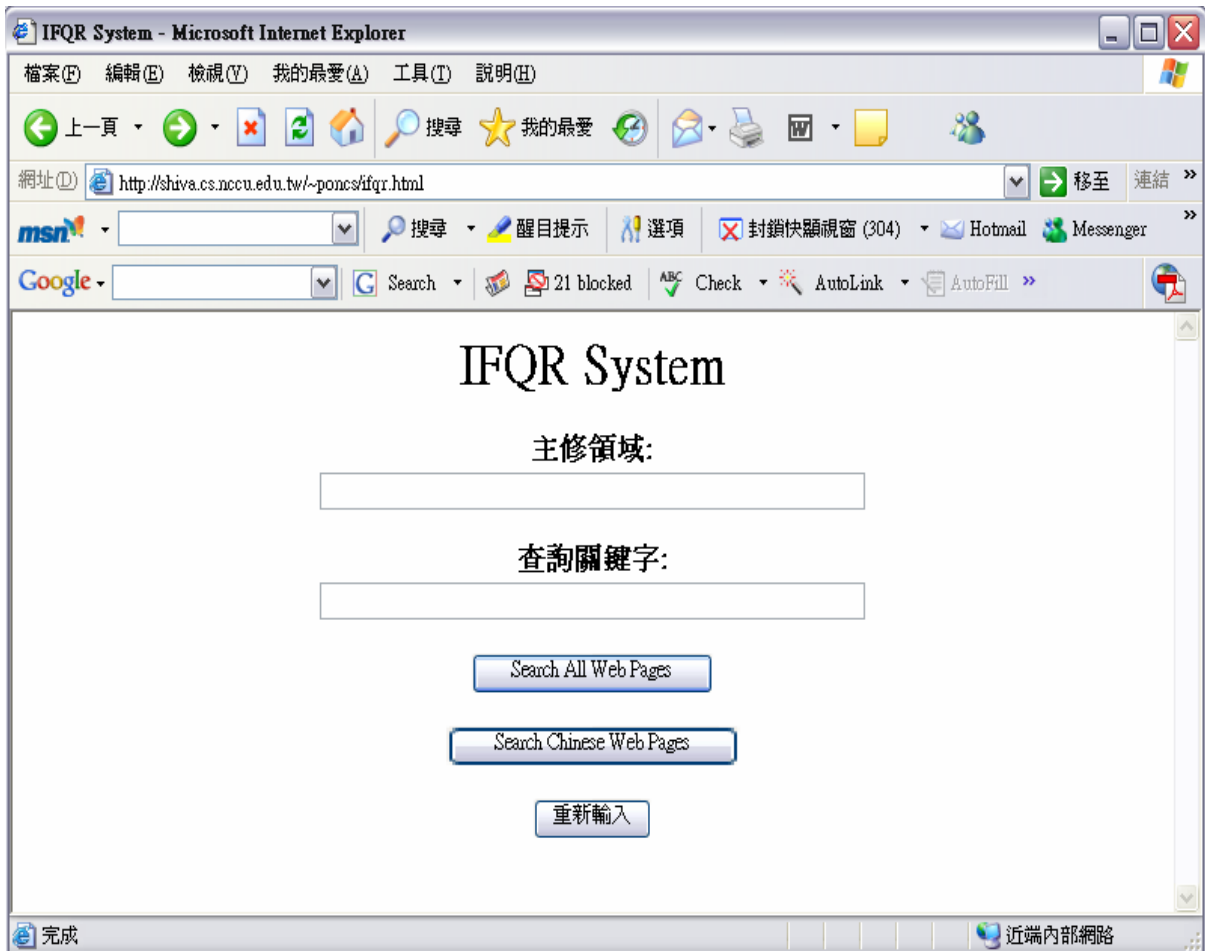


圖 4.2：系統首頁。

由 Google 所提出來幫助一些程式設計師或研究者有需要使用到 Google 搜尋網頁的功能時，所方便使用的一套 Web 服務。Google SOAP Search API 是由程式語言 Java 所實作而成，其中包含了許多 Google 自己所寫的 Java 函式庫裡的函式。它允許使用者遠端地去聯繫上系統並利用程式來操作，同時也提供了許多程式設計師或研究者有更多的需求時方便解決的途徑，並且在回傳給使用 Google SOAP Search API 的使用者網頁時，是以 structured data 的形式回傳，更是讓使用者作後端處理時，更為方便。Google SOAP Search API 將所有他們可以提供的功能分成六個 class 來提供我們呼叫，分別是，GoogleAPIDemo、GoogleSearch、GoogleSearchDirectoryCategory、

GoogleSearchFault、GoogleSearchResult 以及 GoogleSearchResultElement。涵蓋了較多的功能且使用 Google SOAP Search API 的使用者較常使用到的是 GoogleSearch、GoogleSearchResult、GoogleSearchResultElement 這三個 class，GoogleSearch 主要是提供使用者在做搜尋之前要求的一些功能，例如：搜尋非英文的網頁、設定搜尋結果的個數等等；GoogleSearchResult 則是提供了使用者在針對搜尋回傳結果所要求的一些功能，例如：搜尋時間等等；而 GoogleSearchResultElement 則是針對回傳頁面裡的一筆結果來提供功能，例如：取得某一筆結果的 URL 等等。不過，相對的，Google SOAP Search API 也有一些限制條件的存在，例如：針對一組帳號密碼，一天能夠允許查詢的次數最多為一千次；每次針對查詢關鍵字回傳的網頁最多會回傳前一千筆網頁等等均是 Google SOAP Search API 的一些限制條件。圖 4.3 為使用者在我們的中介系統上查詢關鍵字“data mining”所回傳英文頁面第一頁的網頁資料。圖 4.4 為中文頁面第一頁資訊。

2. Web Page Collector：不論是在長期情境的部份或者是短期情境的部份，網頁的內容，網頁所傳達給讀者的資訊，都是本研究的一大重點。因此，將網頁抓取回來提供給我們做分析，也是我們的程式裡，佔有舉足輕重地位的一部分。也因此，在 Web Page Collector 這一個程式模組中，最主要的目的就是抓取回使用者查詢時，搜尋引擎回傳的網頁，並且根據使用者有點選與未點選的，也將其分門別類的記錄下來。而在這部分，同樣的也利用 Google SOAP Search API 來協助實作。在長期情境方面，我們要還原出當初搜尋引擎回傳給使用者的前一百筆網頁，因此，根據 Search Log 中所記錄的使用者關鍵字查詢，我們同樣的輸入給 Google SOAP Search API，根據 Google SOAP Search API 的回傳，一筆一筆的分析出來，也一筆一筆的抓取其網頁回來。

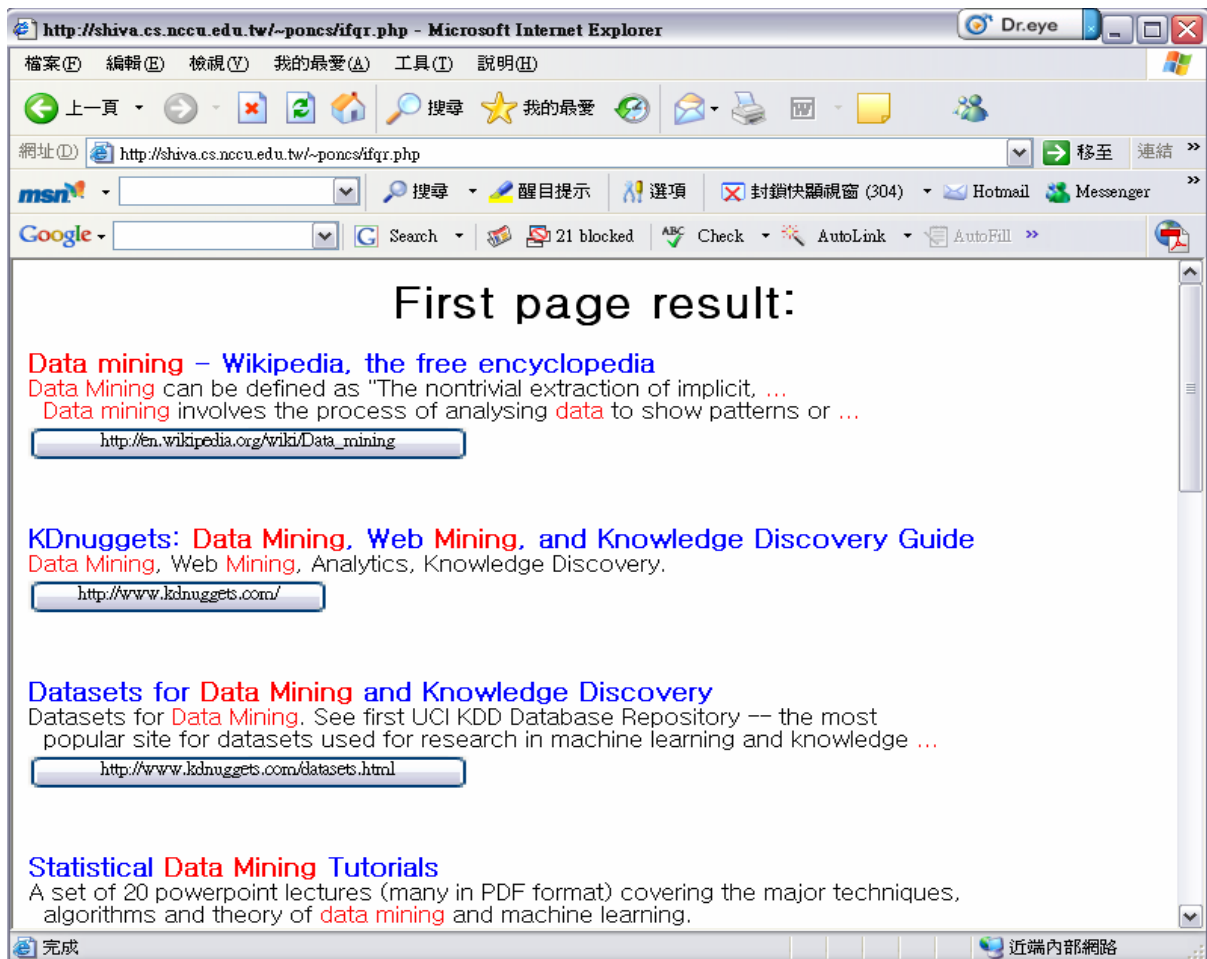


圖 4.3：系統針對“data mining”關鍵字所回傳第一頁的資訊(英文)。

不過，在這部份可能會遇到的一個問題是，因為時間的關係，因為搜尋引擎 Ranking Algorithm 的關係，因此，現階段還原的網頁排序可能會跟當初回傳給使用者的網頁排序，有著些許的出入。但由於本研究是取前一百筆網頁，數目不算太少，這個問題對我們而言影響並沒有這麼大，且經由實驗統計來看，以人直覺重要的網頁還是有保留著，出入僅僅只有一兩個網頁而已，所以這問題對本研究而言，影響並不會太大。而在短期情境方面，即根據目前使用者瀏覽多少個回傳網頁，則抓取多少回來做分析與整理。

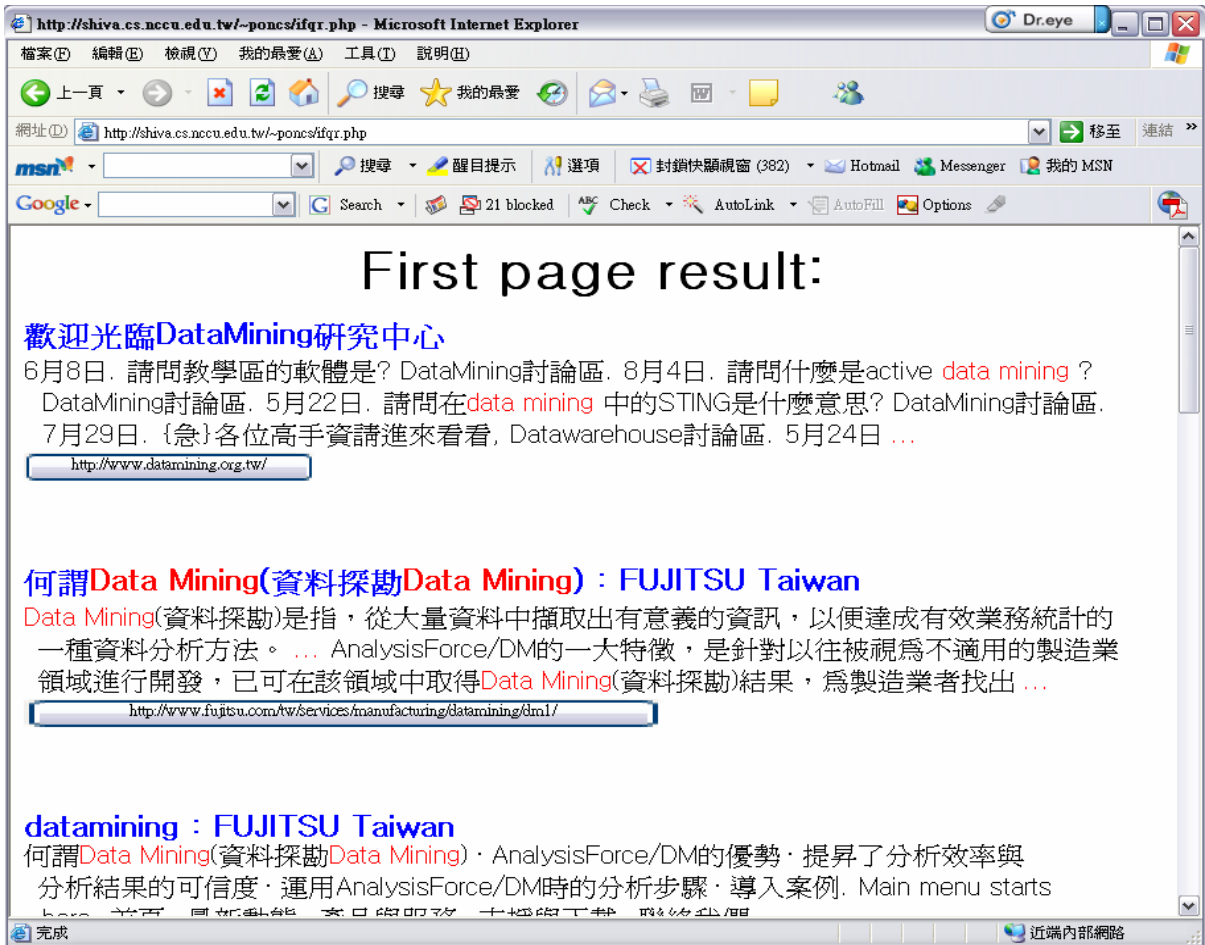


圖 4.4：系統針對“data mining”關鍵字所回傳第一頁的資訊(中文)。

3. HTML Parser：在一般使用者做網路搜尋的時候，無非是想找到一些網頁，根據讀取、吸收網頁的內容，來取得其真正需要的資訊需求。而在介紹前一個程式模組 Web Page Collector 的時候，我們也有提到網頁的內容、網頁所要傳達給讀者的真正意義，在本研究中是佔了相當重要的地位。因此，如何 parse 網頁，則成了本研究相當重要的課題。在之前我們所提的觀點之中，其實不難看出，網頁還是由眾多關鍵字所組成，萃取出網頁重要的關鍵字依舊是本程式模組重要的依據之一。本研究所提出的方法，除了針對網頁全文作 parse 外，同樣也針對了 HTML 重要的格式，提出了萃取包含於其中的關鍵字這樣的觀點。因此，在 parse 網頁全文方面，首先必須將網頁中所有的 HTML

tag 給消除掉，再針對剩下的內容萃取關鍵字。而萃取 HTML 重要格式裡關鍵字的部份，我們鎖定了 Anchor Text、標題、粗體字，這三種重要的 HTML 格式集合，而它們在 HTML tag 裡的表示法則見表 4.1。因此，以表 4.1 為例來說，根據表 4.1 舉出的例子，則取出這三種重要的格式裡所包含的關鍵字均為“Data Mining - Home Page”這樣的關鍵字。

而在這個程式模組中，另外的一些事項是，因為我們所解決的網頁包含了英文網頁也包含了中文網頁，牽扯到了中文網頁，首先所要解決的事項就是斷詞的問題。在這邊斷詞的問題，我們藉助了中央研究院資訊科學所詞庫小組所開發出的一套系統--「中文斷詞系統」(Sinica) (<http://ckipsvr.iis.sinica.edu.tw/>)來幫助我們。中文斷詞系統開放了一套 API 以供我們 Client 端來做呼叫，我們將想斷詞的本文整體的包裹在系統所規定的 XML format 中，藉由 TCP Socket 連線傳送驗證資訊以及 XML 文件至 Server，同時等待 Server 回傳斷詞後的結果，如表 4.2 所示。以表 4.2 為例子，我們傳送了「國立政治大學」內文給中文斷詞系統，且將其包裹在系統規定的 XML tag 中，而系統則回傳「國立(A) 政治(N) 大學(N)」這樣斷詞後的內容回覆於我們，括號內的英文字母則是代表了括號前中文字的詞性，例如：(A)為「非謂形容詞」，(N)為「名詞」等等，而詳細的詞性內容各代表什麼，則參照中文斷詞系統(<http://ckipsvr.iis.sinica.edu.tw/>)裡的說明。

此外，中文網頁還有編碼的問題，在本篇研究中，我們選定了 Big5 來當成我們處理中文網頁統一的編碼，因此，我們使用了專門用在作業系統 FreeBSD 上轉碼的工具 AutoConvert 來做轉碼的動作，我們發現，一般中文網頁的編碼不外乎 Utf-8 與 Big5 為主，因此，本系統大部分都會將 Utf-8 編碼的中文網頁轉碼成以 Big5 編碼的中文網頁，再做 parse。

同時，在 HTML Parser 程式模組中，我們一樣地會找出每一個網頁的主題關鍵字。依據所找出的每類關鍵字之重要程度大小來訂定它們該給予的權重值，根據這些權重

值，依權重值最大的前幾關鍵字，則將其稱為主題關鍵字。依照測試的結果，我們認定權重值大小的給予依序為 Anchor Text > 標題 > 粗體字 > 全文萃取的關鍵字，這樣的權重值大小順序較能符合網頁意義。

表 4.1：三種重要格式於 HTML 裡的表示法。

格式名稱	於 HTML 裡的表示法
Anchor Text	<code>&lt;a href=" http://www.the-data-mine.com/ "&gt; Data Mining – Home Page &lt;/a&gt;</code>
標題	<code>&lt;title&gt;Data Mining - Home Page&lt;/title&gt;</code>
粗體字	<code>&lt;b&gt; Data Mining – Home Page&lt;/b&gt;</code>

表 4.2：使用中文斷詞系統傳送本文前後之內容。

傳送給系統前後	本文內容
傳送給系統的內容	<pre>&lt;?xml version="1.0" ?&gt; &lt;wordsegmentation version="0.1"&gt; &lt;option showcategory="1" /&gt; &lt;authentication username="xxx" password="xxx" /&gt;&lt;text&gt; 國立政治大學 &lt;/text&gt;&lt;/wordsegmentation&gt;</pre>
系統回傳的內容	<pre>&lt;?xml version="1.0" ?&gt; &lt;wordsegmentation version="0.1"&gt; &lt;processstatus code="0"&gt;Success&lt;/processstatus&gt; &lt;result&gt;&lt;sentence&gt; 國立(A) 政治(N) 大學(N) &lt;/sentence&gt;&lt;/result&gt;&lt;/wordsegmentation&gt;</pre>

4. Search Behavior Miner：在長期情境部分，做完 HTML Parser 後，會將每一位以往使用者的關鍵字查詢以及點選過網頁資訊之主題關鍵字，存入資料庫裡，成為了代表每一位以往使用者搜尋行為的資料庫。短期情境的部分，記錄下的即是目前使用者的關



鍵字查詢以及未點選網頁摘要之主題關鍵字。而本程式模組的輸入資料即是長期情境資訊以及短期情境資訊兩者，最主要的工作在於實作出 Filtering 與探勘兩個部份。在 Filtering 部分，主要是針對長期情境中各個 Query Sessions，去計算出其查詢關鍵字密度以及點選網頁密度，再將高於查詢關鍵字密度門檻值以及低於點選網頁密度門檻值的 Query Sessions 給去除掉。另外就是針對每個短期情境未點選網頁摘要之主題關鍵字，去掃描長期情境資料庫，將有包含這些主題關鍵字的 Query Sessions 也予以去除。另外一部份是實作資料探勘，這部分主要實作 Apriori 演算法，探勘出超過最小出現次數的關鍵字頻繁項目集合，再經由兩個條件的篩選，將最後答案送至系統介面，回傳給使用者瀏覽。

## 4.2 實驗資料來源

欲分析使用者在搜尋引擎上的行為，直觀而言，在資料來源的部份上，就必須取得搜尋引擎與使用者互動的一切過程，也就是搜尋引擎的查詢日誌。但由於目前搜尋引擎的查詢日誌對於使用者而言，是非常私密的資料，裡頭可能包含了關於使用者個人相當重要的資料以及紀錄，同樣的，這些查詢日誌對於搜尋引擎公司而言，也是相當關鍵的商業機密。因此，由上述原因可得知，要取得這些機密的查詢日誌有實際上的困難。而本研究的中心理念正是建構於存在著一個龐大的查詢日誌資料庫的基礎之上。既然須取得搜尋引擎的查詢日誌如此地困難，我們因此考慮使用代理伺服器日誌(Proxy Server Logs)，但是代理伺服器日誌同樣地也會牽扯到使用者個人隱私權的問題，此外，有些搜尋引擎的設定並無法讓我們從 Client 端經由代理伺服器日誌去探知使用者在搜尋引擎上點選了哪些網頁，而這也是使用代理伺服器日誌的最大問題所在。因此，基於這些關於資料來源取得的種種重大問題，我們解決的方案即是實作了一個介於搜尋引擎與使用者之間的中介系統：Implicit Feedback Query Refinement System (IFQR System)，也藉著

此系統，我們來取得使用者與搜尋引擎之間所有的互動紀錄。

要建立這樣的一個中介系統，搜尋引擎的選擇是非常重要的，我們選定了 Google 來當作我們挑選搜尋引擎的對象，因為 Google 是目前一般使用者最喜歡使用的搜尋引擎，其介面、其編排方式，更是為一般使用者所喜歡且熟稔的，尤其是它的網頁排序演算法，更是公認為在這方面最進步的搜尋引擎。換句話說，我們藉著受測者在 IFQR System 與 Google 上查詢行為的比較，來比較自動查詢修正(IFQR System)與沒有自動查詢修正(Google)間的效能差異。

我們系統蒐集資料的時間由西元 2006 年的 5 月中旬到 6 月下旬，歷經一個半月。  
表 4.3 為所蒐集的查詢紀錄之詳細資訊。

表 4.3：資料來源的詳細資訊。

資料來源項目	資料來源詳細資訊
時間	2006 年 5 月中旬~6 月下旬
檔案大小	5MB
查詢紀錄的筆數	4994 筆
User Session 數目	77 個
Query Session 數目	206 個
查詢關鍵字的數目	4024 筆

### 4.3 實驗設計

本研究的主要目標是想要幫助搜尋引擎生手或者不懂得如何給予關鍵字的使用者藉由自動查詢修正來達到他們的資訊需求。但是，資訊需求是相當主觀的，而如何判定

是否有達到資訊需求，每個使用者的判定方式也都各有不同。因此，我們認為，唯有將資訊需求規則化、清楚化，在判定是否有達到資訊需求時，才能夠非常的統一且一致。

為了客觀地評估實驗的效果，除了上述蒐集的查詢日誌外，我們設計特定的題目，邀請測試者透過我們的中介系統在搜尋引擎上搜尋資訊。我們針對五大類不同性質的主題，設計了七個不同的問題，每一個問題都清楚的點明了一個資訊需求，同時也都有清楚且正確的答案，如此一來，在判定是否達到資訊需求時，則會非常的清楚且明白。以下為七個問題的詳細描述：

(1)請問在 Association Rule Mining 中，有哪一種方法是直接將資料拿到記憶體來做探勘，減少了掃描資料庫的時間，也不需要 candidate 的產生，請問是哪一種方法(第一個提出的方法)，且希望大家找出主要內容為介紹這個方法的網頁。

(2)在物理學中，有一個定律主要是在說當流體流速減少時，壓力會增加，試想空氣流經飛機機翼截面，通過機翼上方的流體因為行徑路線比較長(機翼上半部曲面弧長較長)，因此速度比較快。因而壓力會較下方流體壓力來的小，所以下方氣壓大於上方氣壓，造成飛機的昇力。試問，這是哪一個定律，又請找出主要內容為說明這個定律的網頁。

(3)相信大家小時候都有聽過這樣的一個故事：

故事摘要：

故事發生在中世紀，一個因長相醜陋而被人丟棄的嬰兒，在一個偶然的機緣下被很有名望的教堂副主教收養成為養子並扶養成為教堂的敲鐘人；雖然他的面貌醜陋但內心卻很高尚善良。

在愚人節那天，流浪的吉普賽藝人在廣場上表演歌舞，有個吉普賽姑娘，她長得美麗

動人舞姿也非常優美吸引了來往的行人；同時，吉普賽姑娘的美貌也吸引副主教，副主教瘋狂地愛上了她。原本一向專心於聖職的副主教在愛上吉普賽姑娘後，一股強烈的慾望使得副主教開始謀計將吉普賽姑娘佔為己有。於是他命令面貌醜陋的男主角把吉普賽姑娘搶來。結果法國國王的弓箭隊長救下了吉普賽姑娘，抓住了男主角。他把男主角帶到廣場上鞭笞，善良的吉普賽姑娘不計前仇，反而送水給男主角喝。男主角雖然外貌醜陋，內心卻純潔高尚，他非常感激吉普賽姑娘，也因此愛上了吉普賽姑娘。天真的吉普賽姑娘對弓箭隊長一見鍾情，兩人約會時，副主教悄悄在後面跟著，出於嫉妒，他用刀刺傷了弓箭隊長，然後逃跑了。吉普賽姑娘卻因謀殺罪被判死刑。男主角因為自己對吉普賽姑娘的愛慕之心，而把吉普賽姑娘從絞刑架下搶了出來，藏在巴黎聖母院內，但副主教卻趁機威脅吉普賽姑娘，在遭到拒絕後，副主教將吉普賽姑娘交給了國王的軍隊，讓無辜的吉普賽姑娘被絞死了。在男主角得知養育他多年的副主教竟是害死吉普賽姑娘的兇手時，痛苦的掙扎之下親手把副主教從教堂的鐘塔上推下。

請大家運用聰明的頭腦，利用搜尋引擎找出這篇故事的英文名稱，也順便找出主要內容為描述這篇故事的網頁。

(4)請問提出地球繞太陽轉的科學家為誰，請找出主要內容為介紹這位科學家生平的網頁。

(5)相信大家小時候都有聽過這樣的一個故事：

故事摘要：

長髮姑娘還在媽媽的肚子裏時，就註定一生下來就必須送給隔壁的女巫領養。因為媽媽想吃隔壁女巫種的花耶菜，那是種綠油油的蔬菜。可是當父親去偷摘時，卻不幸被女巫發現了。女巫很生氣，於是她提出一個不合理的要求。她要他們的第一個孩子。於是，長髮姑娘出生後的第一天就被抱走了。一轉眼，長髮姑娘已經十六歲了，但她

從沒離開過古塔。因為古塔很高，又沒梯子，每次巫婆來看她時，總是喊著「長髮姑娘，把你的頭髮放下來吧！」於是，長髮姑娘就把她一頭長到地的頭髮給放下，讓女巫爬上來。女巫告訴長髮姑娘，只要她一滿十八歲，成為女巫後就可以離開古塔了。因此，長髮姑娘沒見過女巫外的其他一人。她很想離開這裏，去外面的世界看看。但因為沒有梯子，所以她只能用她的歌聲，來表達她的情緒。

有一天，一個年輕的王子在附近打獵，聽到了歌聲，於是他尋著歌聲的方向而去。他找到了古塔，看見了長髮姑娘。他被她的美貌及歌聲吸引住了。

而長髮姑娘也被眼前的年輕王子所吸引。於是她和王子談起戀愛。王子每天都來看她，並且要她做他的新娘。而長髮姑娘也答應了。

可惜，這一切都被巫婆知道了。女巫很生氣，她剪去了她的長髮，並且讓她離開古塔，在森林裏流浪。隔天，當王子再度來找長髮姑娘時，女巫放下她所剪下的長髮，讓王子爬上來，就在王子爬到一半時，告訴他這個悲慘的消息。並用刀子割斷長髮，打算讓王子摔死。

女巫詛咒王子，就算他沒摔死，也別想看見長髮姑娘，她要他一輩子因為看不見長髮姑娘而痛苦致死。結果，王子命大沒有死，可是眼睛卻受了詛咒瞎了。

於是王子只好在森林附近流浪，他因為看不見長髮姑娘而傷心著。突然他聽見了他所熟悉的聲音，他朝著那個方向走去。

他找到了長髮姑娘，但他的眼睛再也看不到她了。長髮姑娘安慰王子說「別擔心，就讓我當你一辈子的眼睛吧。」於是長髮姑娘用吻去王子臉上的淚珠。結果奇蹟發生了，長髮姑娘的吻破除了女巫的魔咒。王子又可以看見東西了，他高興的抱著長髮姑娘，帶著她回去他的城堡，遠離這座森林。

而女巫呢？因為她一時的大意，竟忘了帶梯子過來，現在她還在古塔上呢，她正高喊著「誰來救救我呀！」！

請大家運用聰明的頭腦，利用搜尋引擎找出這篇故事的英文名稱，也順便找出主要內

容為描述這篇故事的網頁。

(6)為什麼在一般平年之中，二月只有 28 天，請找出有詳述這個原因的網頁。

(7)1992 年巴塞隆納奧運籃球項目中美國夢幻一隊陣中唯一的大學生是誰，且請找出該位球員的職業生涯紀錄。

針對上述的七個問題，首先，我們邀請非常熟悉搜尋引擎且熟悉這些問題所屬主題的經驗使用者，在我們的中介系統上作測試，確定這七個題目是可以經由搜尋引擎且在全球資訊網上找到答案。並且，我們也會將這些測試使用者的搜尋過程，一併地納入我們所蒐集的查詢紀錄檔中。雖然我們也蒐集了一個半月的查詢記錄檔，但是，我們都非常清楚，每位使用者的查詢都是非常不一致且包羅萬象的，因此，將測試使用者的搜尋過程納入我們的查詢紀錄檔中，則我們可確保接下來給受測者的實驗，我們的查詢紀錄檔中至少會有相關資料的存在。

#### 4.4 實驗評估

本研究在實驗評估這方面，主要想要以「查詢的回合數」來評估我們的自動查詢修正的效果：

查詢的回合數：在討論這項評估方法之前，我們先定義何謂「查詢的回合數」。使用者每轉換一次查詢關鍵字做搜尋，則稱為查詢過程的一個回合。例如，某位使用者的查詢過程為， $Q_1 \rightarrow W_1 \rightarrow Q_2 \rightarrow W_2$ ， $Q_k$  為下關鍵字查詢， $W_k$  為網頁點選，因此，這位使用者的查詢過程我們可以解讀為，先下了  $Q_1$  這個關鍵字，並在回傳的網頁中點選了  $W_1$  這個網頁，接著又下了  $Q_2$  關鍵字，最後點選  $W_2$ ，結束整個查詢過程。因為這位使用者在整個查詢過程中變換了兩次的關鍵字查詢，所以此次搜尋的查詢回合數則為 2。因此，

直觀而言，查詢回合數可以很明顯的去衡量出搜尋的效果成效如何。

根據上述的評估方式，在實驗評估的模式中，我們一樣地以前一小節提及的那七個問題，邀請了數位搜尋引擎生手以及較不熟悉這些問題所屬主題的使用者來接受我們的實驗。首先，我們選定的比較對象是 Google，受測者先詳細的閱讀了這七個問題，輔以專人解說，確實的了解了資訊需求為何，以受測者對於這些問題的了解程度加上受測者本身對於這些問題的想法，針對每一個問題都先在 Google 上做一次完整的搜尋，同時記錄下上述評估方法的數據，也就是在 Google 上最後搜尋到所需資訊的搜尋回合數。接著，受測者再到我們的系統上針對每一個問題再搜尋一次，首先第一回合，受測者鍵入與剛剛在 Google 上受試時第一回合一模一樣的關鍵字查詢，同時在回傳的網頁中也點選了與剛剛相同的網頁集合，接著，在之後的回合中均使用我們的查詢修正方法。同樣的也去記錄下我們評估方法的數據，來做比較。

## 4.5 實驗結果

在本節中，我們將清楚呈現本研究的結果數據，同時對於本研究的結果數據作稍微的論述。

首先，先大致地介紹本研究針對上述七個問題所蒐集到的一些例子，包含有經驗使用者的查詢行為以及搜尋引擎生手的查詢行為，還有本研究所回傳給使用者的修正查詢。我們將以第二題、第三題以及第五題為例：

以第二題來說，我們列舉一個搜尋引擎生手的例子為：{airplane, rise, law, air, aeromechanics} → {airplane, rise, theme} → {airplane, wing, rise, theme, reason} → {airplane, wing, rise, reason} → {airplane, atmospheric, pressure, rise, reason} → {airplane, atmospheric, pressure, rise, reason, wing} → {airplane, atmospheric, rise, theme}

→ {airplane, invent, rise, theme} → {why, airplane, rise, theme} → {airplane, fly, reason}, 經驗使用者的其中一個例子為：{flight, rise, theory}, 我們回傳的修正查詢為 {Bernoulli, pressure, flow, air, fluid}。

以第三題來說，搜尋引擎生手的例子為：{church, ugly, fool, Gypsy} → {church, archdeacon, suffragan, ugly, Gypsy} → {church, archdeacon, suffragan, ugly, fool, Gypsy, bell, Paris} → {church, archdeacon, ugly, Gypsy} → {church, bell, archdeacon, Gypsy, “Notre Dame”}, 經驗使用者的例子為：{gypsy, bell, story, Abandoned, Infants} → {gypsy, girl, bell, tower, story, Abandoned, Infants}, 我們回傳的修正查詢為：{Dame, Notre, Esmeralda, Quasimodo, Hunchback}。

而第五題，搜尋引擎生手其中之一例為：{longhair, princess} → {long, hair} → {story, longhair, princess} → {children’s, book} → {children’s, book princess} → {Children’s, Literature} → {Princess, with, longhair} → {children, Princess, longhair} → {literature, Princess, longhair}, 經驗使用者其中之一例為：{long, hair, witch, tower, prince, blind}, 而我們回傳的修正查詢為：{long, hair, Rapunzel, girl, witch}。

本研究總共邀請了五位搜尋引擎生手以及較不熟悉這些問題所屬主題的使用者來接受我們的實驗，在評估方面，由於本研究想比較的是搜尋引擎生手分別在本系統與 Google 搜尋引擎上搜尋這些問題的查詢回合數為何，因此，本研究必須定義在一次的查詢回合中，前多少筆的網頁有滿足使用者資訊需求的才能構成這次查詢回合是成功搜尋的。根據以往的研究統計[28]指出，使用者平均一次查詢所瀏覽頁數是 1.7 頁，如果一頁回傳網頁以十筆網頁計算，那麼，平均來說使用者每查詢一次會瀏覽 17 筆網頁。也由於是實驗的階段，為求精準、詳細，我們設定一次查詢回合中所瀏覽網頁的筆數以前五十筆為限，在前五十筆之中，若有符合使用者資訊需求，我們即認為此回合的查詢滿



足使用者的資訊需求。圖 4.5 以及圖 4.6 則設定不同的網頁筆數來做比較，圖 4.5 為前十筆網頁，圖 4.6 則為前五十筆網頁。可以很清楚的對比到，無論是本系統或者是 Google 搜尋引擎，只以十筆網頁即決定是否此次查詢回合有滿足使用者需求的，效果方面很明顯的都比以五十筆網頁決定的都差。圖 4.5 以及圖 4.6 均為本研究的實驗數據，主要為搜尋引擎生手根據本研究設計的七個問題，分別在 Google 與本研究的系統 IFQR System 搜尋時，所需查詢回合數的比較。其中，X 軸為問題的題號，Y 軸為所需查詢回合的次數。依照圖 4.5 以及圖 4.6 所示，我們可以很清楚的發現，搜尋引擎生手在 IFQR 上搜尋的查詢回合數遠比在 Google 上搜尋還來的少，這代表了使用本系統的查詢修正方法可以更快速也更準確地找到資訊需求，也表示本系統確實可以幫助到搜尋引擎生手使用者，並且是相當有幫助的。

另外，值得探討的是，根據圖 4.5 與圖 4.6 所示，IFQR System 在第一題與第六題的表現遠比 Google 來的好，平均相差四個查詢回合。而第三題、第五題與第七題，IFQR System 平均只比 Google 相差一個查詢回合。第一題會有這麼大差距源自於題目的主題性較為專業，較專業的問題對於一般的使用者原本就較為困難。此外，本系統的使用者多數具有資訊科學背景。因此，經驗使用者的查詢經驗對於搜尋引擎生手有很大的幫助。第六題必須找出題目所述之原因的網頁，由觀察得知，搜尋引擎生手查詢的關鍵字所回傳的網頁往往只是提到二月有 28 天，但並沒有描述其原因。因此，由本題可看出，經驗使用者的查詢過程，對於生手的查詢有很大的幫助。相反的，第三題與第五題均為故事題，故事題最大的重點在於使用者必須較熟悉整個故事的情節，藉由對故事的瞭解來判斷重要的關鍵字。由於這兩題的本意是假設受測者聽過童話故事鐘樓怪人與 Rapunzel，但卻想搜尋其詳細故事內容。為了確認受測者瞭解故事內容，因此在題目設計上我們將故事劇情描述得相當仔細。因此，搜尋引擎生手對於故事題也是得心應手，以致於經驗使用者在故事題的幫助並不算太大。而第七題也是因為題目的描述非常詳

盡，以致於經驗使用者與生手在查詢回合數中差異不大。

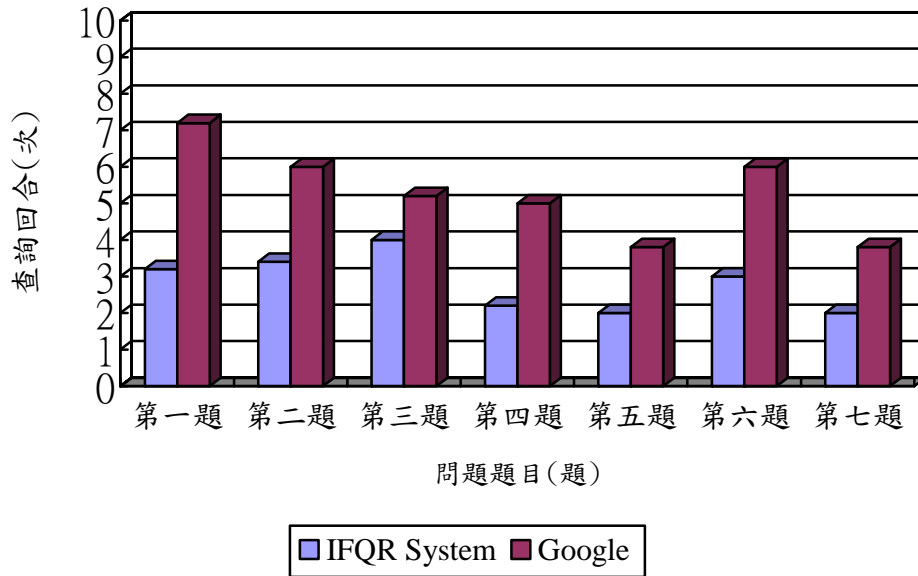


圖 4.5： IFQR 系統與 Google 之比較(前十筆網頁)。

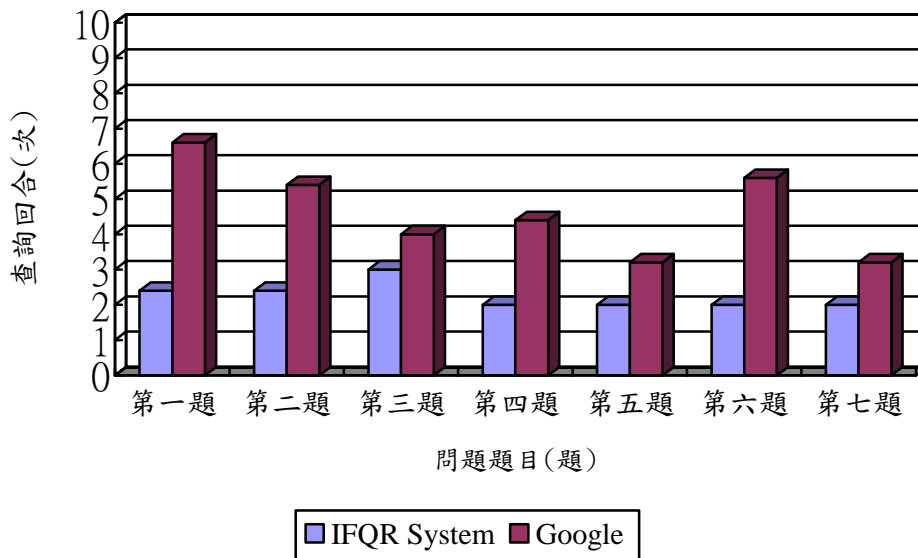


圖 4.6： IFQR 系統與 Google 之比較(前五十筆網頁)。