南　華　大　學

資訊管理學系

碩士論文

應用領域本體論設計整合網路上搜尋引擎機制

Design of an Integrated Mechanism of Search Engines on Web by Domain Ontology

研 究 生：鄒昇衛

指導教授：王昌斌

中華民國九十六年六月

南　華　大　學

資訊管理學系

碩 士 學 位 論 文

應用領域本體論設計整合網路上搜尋引擎機制

研究生：劉昇衛

經考試合格特此證明

口試委員：

指導教授：

系主任(所長)：資訊管理學系 系主任吳光閃

口試日期：中華民國　96　年　6　月　21　日

誌　　謝

碩士班學習尾聲即將響起，此刻的心情洋溢著喜悅與不捨。回首這三年來，昇衛要感謝的人很多，首先最要感謝的是恩師王昌斌教授，在論文研究期間不辭辛勞地指導與提攜，以及時常地關心與鼓勵支持昇衛，非常由衷地感謝，雖隻字片語，意更甚於言表，老師，真的很謝謝您！再者，十分感謝口試委員中正資管所阮金聲教授，以及所上邱宏彬教授，對於本論文悉心指正與精闢的建議，使得學生在研究寫作與思考邏輯上更加成長增進，論文能得以愈趨完善，非常地謝謝您們！此外，謝謝陳裕民教授、陳宗義老師和蔡德謙老師，幫助釐清思考困境與提供寶貴的意見。

這段期間感謝南華同儕們彼此的鼓勵與照應，尤其是文天學長、建磐學長在論文上的幫忙，以及實驗室成員育銘、育弘、政宇在研究上的協助，與你們一同成長的相渡時光非常愉快。

最後要感謝我最親愛與敬愛的家人感謝你們的勉勵與支持，讓我能勇往前進，追求人生目標的完成，謝謝你們！

鄒昇衛 謹識

于 南華大學資管所

九十六年 六月

應用領域本體論設計整合網路上搜尋引擎機制

研究生：鄒昇衛　　指導教授：王昌斌博士

南　華　大　學　資訊管理學系碩士班

摘　　要

近來由於產業及科技的競爭，以致於相關知識的蒐集、獲取、整合、儲存、管理、分享與運用之重要性相對提升。隨著網際網路發展，如何以自動化的方式有效獲取網路上的資訊提供使用者所需的知識是一項很大的挑戰。

本研究結合利用資料探勘發掘網頁內容知識並檢視其相似性且導入領域實體概念，發展強化搜尋引擎的過濾及排序機制，透過演算法去除格式不完整、有重覆性網址且針對格式化的摘要及標題進行資訊含量之運算，其值若介於本研究所設立之可接受範圍，便進一步計算摘要權重值；若遇到描述不同但意思相仿的摘要，會應用領域實體所建立的法則計算詞彙相似程度，其後給予適當權重值，本研究的領域實體是著重於國小數學學習方面，系統則將每篇摘要之權重排列順序，其後檢視符合原意與否，再取回其網頁內容，經由擷取就變成可利用知識，此知識可提供給使用者解決問題之參考。希冀能節省使用

者自行過濾檢索時間與減少頻寬資訊量。

關鍵詞：網頁內容探勘、資訊檢索、本體論、搜尋引擎

Design of an Integrated Mechanism of Search Engines on Web by Domain Ontology

Student：Sheng-Wei Tsou　　Advisors：Dr. Chin-Bin Wang

Department of Information Management
The M.I.M. Program
Nan-Hua University

ABSTRACT

Due to the rapid development of information technology, it is important to search, gain, integrate, store, share, reuse and manage the different scopes of professional knowledge. This issue becomes increasingly essential for users to extract the appropriate internet information efficiently and automatically for a great variety of resources on websites.

The research includes both web content mining and information retrieval to design a strengthened mechanism of search engines on web by domain ontology. We try to develop and design the algorithms which have the functions of filtering, ranking and weighting. The purpose is to filter the dump link and the advertisement link according to the web document titles, the ranking of the abstract's URL and the weighting of the information content. Then the users can retrieve more suitable information and capture the web content knowledge efficiently. In the process of filtering and ranking, the knowledge can be refined to useful one which can offer end users to decide whether or not the knowledge meets their demands. In this way, the users can save the time to filter and retrieve and decrease the

loading of internet.

Keywords：web content mining, information retrieval, ontology, search engine

目　錄

表　　目　　錄

圖　　目　　錄

第一章、緒論

第一節　研究背景

隨著資訊科技日新月異，人們可透過網路資源獲取知識，一般人遇到問題時，第一反應就是藉由搜尋引擎檢索答案，儼然搜尋引擎已成為知識獲取的工具之一，加上資訊數位化盛行，且各種電子型態的文件及資訊如文字(text)、影像(image)及聲音(audio)等傳播媒體，持續地以互動(interactive)及整合(integrated)的方式放置於網際網路上供大眾存取[1]，所以網路上的資訊琳瑯滿目，彷彿是一座超大型資料庫。

根據普遍觀察，使用者利用搜尋器進行檢索常會受到輸入界面和搜尋方式兩種因素影響；設計複雜界面常讓使用者難以上手，因此造成檢索結果精確度下降，且每具搜尋引擎其檢索方式不同，使用者需求已無法從單一搜尋器獲取足夠資訊[11][12]。除此之外，搜尋引擎效能也受到網路負載能力與頻寬大小的影響。針對前述之原因，使用者若想在茫茫網海中找尋需要的資訊，就猶如大海撈針一樣困難，殷鑑於此，本研究利用資料探勘挖掘網頁內容知識，配合專家學者所提出相關資訊檢索方法及本體論概念，便可解決前述之問題。另外本體論概念中所衍生的相關詞及同義詞則有助於進一步判斷資訊是否符合使

用者所需。

第二節　研究動機

目前各家搜尋引擎功能都相當強大，其搜尋結果相去不遠，但缺乏有效率的過濾機制且資料同質性及重複性太高又散亂，因此造成人們耗費大量時間與精力自行檢索，加上詞彙之間存在著某些隱含關係，若不應用實體論概念建立彼此的關聯，往往讓使用者無法獲得想要的知識，所以如何快速正確從網路上獲取知識是學者長久以來追求的目標之一。針對前述種種問題，本研究透過資料探勘中的網頁內容探勘(web content mining)方法、資訊檢索(information retrieval)的相似度與應用領域實體論(domain ontology)建立數學的本體，設計並強化搜尋引擎的過濾及排序機制且應用領域實體論建構出同義詞規則為本研究之動機。

當使用者有問題時可至平台上的討論區陳述，系統根據使用者的問題描述，根據後端所提供的領域知識庫為基礎，解析使用者所描述問題之語意，並於網路上擷取與問題相關的知識，建立知識間彼此的關聯，日後將自動化建構成類似樹的結構並儲存於知識庫。系統第一階段是分析解剖問題，產生圖形化語意網(graphical semantic net)，第二階段透過搜尋引擎檢索網路上資料，然後第三階段會針對網頁萃取出所需的隱含知識。如圖1-1所示。

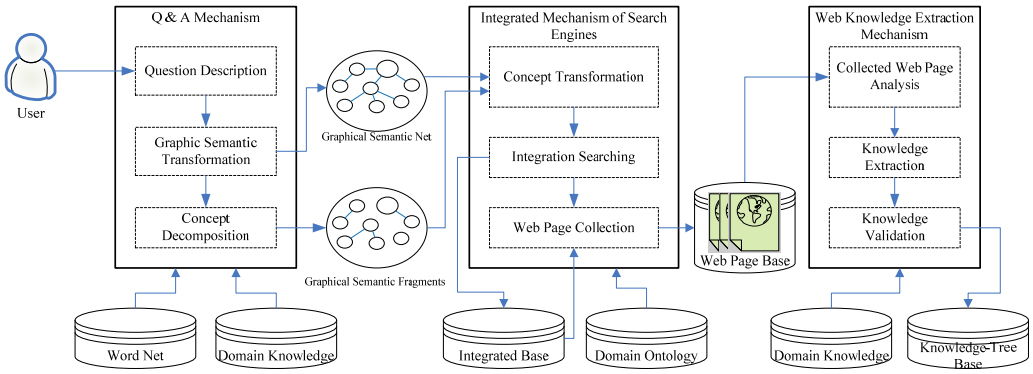

圖 1-1：整體機制架構

本研究為整體機制中的第二階段，主要是針對搜尋後的重要資訊進行資訊含量之計算，如摘要及標題；透過演算法刪除重覆性網址，並依摘要資訊含量多寡給予權重，若資訊含量介於本研究所設立之可接受範圍，便透過所建立的法則計算詞彙之間相似程度，再給予適當權重，此後將自動地依每篇摘要權重高低逐一排序且判斷是否符合原意，其後取回其網頁內容，經由擷取就變成可利用知識，此知識可提供給使用者解決問題之參考，節省使用者自行過濾檢索時間並降低頻寬負載量。

第三節　研究目的

由於實體論的發展已跨越由圖書館科學、哲學和知識表達的學術領域外，漸漸受到市場部門和企業注意，但在各個領域有其不同定義及使用方式，關注的焦點也不同，面對眾多數位化資源，需要一個多用途且具彈性的表達工具以便能順應智慧型的資訊表達和檢索，利用

概念來描述特定領域之知識為當今熱門趨勢[5][16]。

因此，本論文透過實體概念處理詞彙相關的詞意，包含同義詞及相關詞。先計算每則摘要的資訊含量並訂立門檻值，分類為低資訊量、中等資訊量與高資訊量摘要，其後經運算給予適當權重值並排序，依排名逐一與原意判斷符合程度，當中含有意義相同但描述不同之語意，最終提供於知識擷取機制所要的客制化資訊，將使知識更易於編修、彙整、互通及整合，大幅提升其再用性(reusability)與可分享性(sharability)。

第四節　研究流程

本研究流程如圖1-2所示。首先，領域研究階段會閱讀有關於網頁內容探勘、資訊檢索、搜尋引擎與實體論的各種定義與作法；探討相關技術與方法階段則更深入地研討資料探勘、自然語言處理、資訊檢索、可擴充標記語言、資訊檢索、過濾及排序技術的方法與應用；設計階段則依圖形化語意網設計出資訊標準化機制、比對與排序演算法；最後實作與驗證階段透過蒐集的20篇摘要進行實驗，檢視所設計的機制與演算法。

基於此設計建構出整合式搜尋引擎機制(Integrated Mechanism of Search Engines on Web)，主要核心是過濾與排序機制中的演算法；過濾重覆網址且運算資訊含量，稱為「比對演算法」，依摘要權重自動排

序，稱為「排序演算法」，若網路上無適合答案，則回傳予概念轉換機制(concept transformation)重新組合關鍵字集，稱為「回饋機制」，而且藉由實體概念建立有關國小數學領域的相關詞及同義詞規則，導入實體論概念來協助檢索描述不同意義相同之摘要，最終檢索符合原意的網頁知識，其結果可應用於知識擷取方面，細節部份將於第三章詳細介紹。

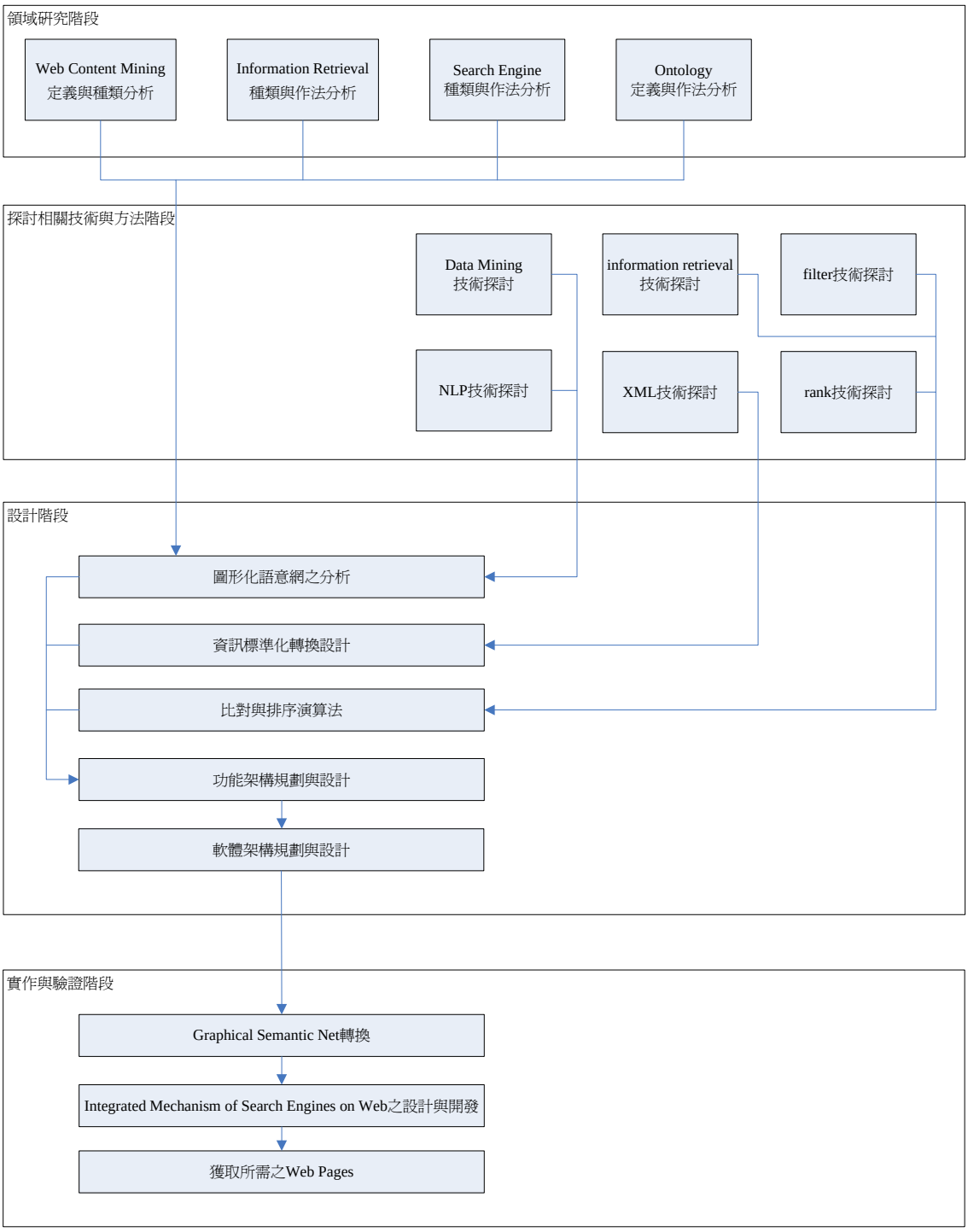

圖 1-2：本研究之研究流程

第五節　論文架構

本論文的章節架構如下：第一章研究背景、動機、目的、流程與論文架構，並概括描述研究的整體架構。第二章則是相關研究之詳細探討，陳述網頁內容探勘的基本概念，以及資訊檢索相關的作法與技術，和本研究所彙整四個著名搜尋引擎的簡介、特性、功能並比較其差異性及本體論的定義與應用面。第三章則簡介圖形化語意網，針對資訊搜尋設計搜尋代理人機制及相關資訊制定客製化格式，詳述比對演算法資訊含量運算與排序演算法中權重值計算，並利用同義詞規則處理意思相同的摘要。第四章將敘述實際運作出研究的核心部份「過濾與排序機制」，且利用數則網路上蒐集的文章，來檢視評估執行後其資訊效率的優劣性。最後，第五章將簡短地為這篇論文做總結及貢獻處，進一步描述未來尚可研究的方向。

第二章、文獻探討

第一節　網頁內容探勘

Cooley等學者於1997年提出網頁探勘(web mining)的概念，網頁探勘主要是利用文字或資料探勘(data mining)的技術，針對網頁的特性，自動從網頁上擷取、發掘出一些特徵與規律(pattern)，並應用於各個領域，簡言之是透過資料探勘技術於全球資訊網上，用以探勘使用者行為，網頁內容或是網站架構的方法[11][18]。網頁探勘應用範圍非常廣泛，舉凡利用在建立適性化網站、個人化電子商務系統、個人化搜尋引擎或控制代理伺服器(proxy)的流量；網頁探勘可分為三類，一為網頁內容探勘(web content mining)，二為網頁用法探勘(web usage mining)，三為網頁結構探勘(web structure mining)。本研究只探討網頁內容探勘其他部份不予闡述。

其定義是自全球資訊網上發現資訊，根據網頁本身的內容進行資料探勘，其內容包括文字(text)、超連結與目錄結構[20][23][24]；或是過濾網路資料並設法找出結構化的資訊，針對網路資料進行語意分析，以獲取符合使用者的資訊[20][23][24]，換句話說就是檢索出資訊中的隱含知識。網頁內容探勘可再細分為以代理人為基底的方法(Agent Based Approach)及以資料庫為基底的方法(Database Approach）

[11][18]。前者可分成三大類：智慧型搜尋代理人(Intelligent Search Agents)、資訊過濾/分類(Information Filtering/Categorization)、個人網站代理人(Personalized Web Agents) [11][18]。後者再細分為多維度資料庫(Multilevel Databases)與網路查詢系統(Web Query System)二大類[11][18]，如圖2-1所示。

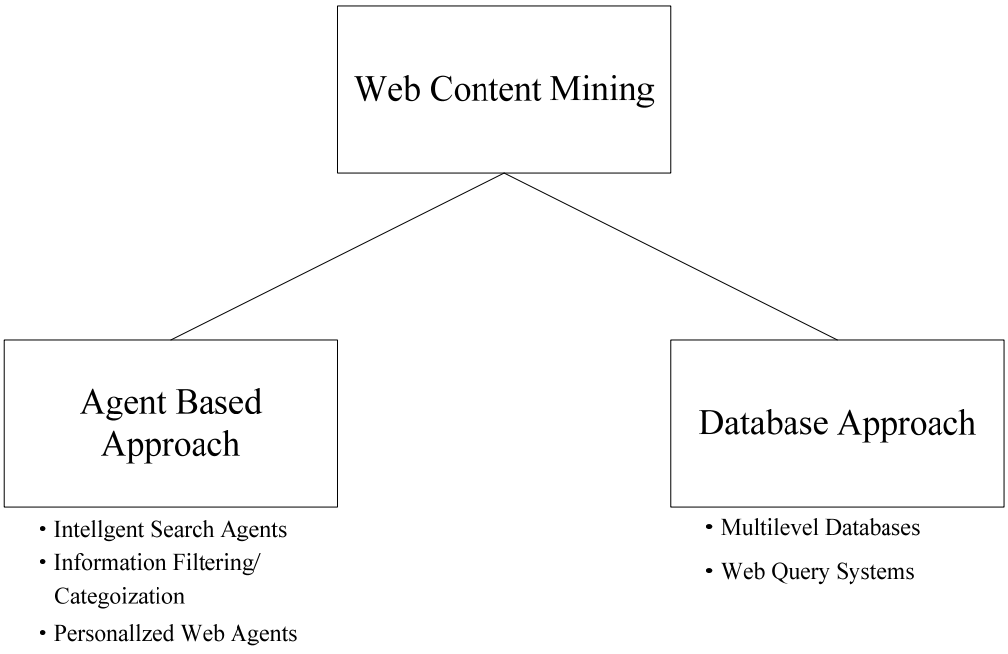

圖 2-1：網頁內容探勘分類

(資料來源：[11][18])

第二節　搜尋引擎

搜尋引擎可謂資訊檢索系統(Information Retrieval)的一種，第一個網路搜尋引擎是基於傳統資訊檢索的演算法和技術而發展，其明確的定義是由專業搜尋網站所提供的搜尋服務程式，讓瀏覽者自行輸入想

要查詢的關鍵字，搜尋服務程式會自動將符合條件的相關資料條列出來，以供使用者點取運用[11][12]。搜尋器主要是根據使用者輸入的關鍵字至網路上展開檢索、擷取與過濾相關資訊，彙整結果後傳回予使用者參考，因此搜尋引擎屬於網頁內容探勘一種[17]。

網路搜尋引擎的架構可以分成三種：集中式架構(Centralized Architecture) 、仲介式架構(Meta-search Architecture)和分散式架構(Distributed Search Architecture)[12]。Jenkins等學者於1998年提出可以將搜尋引擎依服務方式的不同分成為以下三大類：分類目錄(Classified Directories)、自動式搜尋引擎(Automated Search Engines)及匯總式搜尋引擎(Meta Search Engines)[11][17][21]。分類目錄則利用人工的方式對網路資源進行分類；自動式搜尋引擎則由機器主動至網路上搜尋資源；匯總式搜尋引擎提供一個界面讓使用者在單一網頁裡查詢數個搜尋引擎與分類目錄[11][17][21]。依資源存取方式將分類目錄稱為被動式搜尋引擎(Passive Search Engines)，而自動式搜尋引擎可稱主動式搜尋引擎(Active Search Engines) [11][17][21]。

被動式搜尋引擎初期是透過人力直接瀏覽網路上的新網站，並由工作人員為新網站建立其所屬分類。目前搜尋引擎則提供檢索界面，讓使用者直接透過界面進行登錄並搜尋所需，但往往分類目錄無法符合使用者的意思；主動式搜尋引擎利用機器人(robot)或蜘蛛(spider)的

技術，在全球資訊網上自動地搜尋新網站或更新網站資源，並建立URL(Uniform Resource Locator)索引及相關的抬頭(title)或標題(heading)資訊[7]，如圖2-2所示。

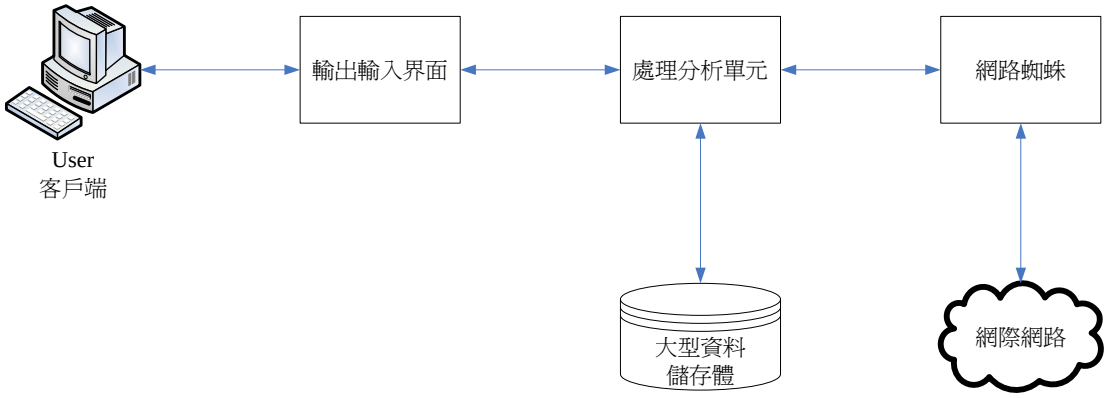

圖 2-2：搜尋引擎基本架構

(資料來源：[7])

壹、AltaVista

AltaVista創立於1995年的美國加州，會依據使用者檢索時的特徵、習慣與經驗來改善其系統功能，首位提供多國語言進行搜尋同時首創圖片、音訊與視訊等搜尋功能且擁有六十一項搜尋技術之專利，是目前公認最強的搜尋器[28]，如表2-1與圖2-3所示。

表 2-1：AltaVista整理表

項目	AltaVista
搜尋型態	網頁、圖片、音訊、視訊、新聞
搜尋格式	html、pdf、ppt、doc、xml、txt
搜尋功能	基本與進階搜尋
搜尋型式	http://www.altavista.com/web/results?itag=ody&q=&kgs=0&kls=0 itag=ody－網頁介面參數 kgs=0&kls=0－網頁字型 &－AND

	間隔各參數
字數上限	中文 800 字，英文 800 字
查詢字的參數	q—可任意輸入字或句子 兩個字詞以上或句子，中間可用+或%20(利用 ASCII)隔開
特性	1.第一個使用網頁檢索技術 2.首先採用多國語言進行搜尋 3.首創圖片、音訊與視訊搜尋功能的引擎 4.可進行多媒體與特色搜尋並且可翻譯 5.擁有 61 項搜尋技術之專利

圖 2-3：AltaVista搜尋介面

貳、GAIS

GAIS(Global Area Information Server)是由國內中正大學資訊工程吳昇教授的網際網路實驗室於 1995 年發展出一套多用途、可調式網路資源搜尋系統，並以之建構資訊服務站，提供給國內使用者一個方便的網路資源搜尋服務，如表 2-2 與圖 2-4 所示，其資訊伺服系統包含以下五個子系統[29]：

一、資料蒐集子系統：

用來蒐集網路上之資訊或內部的資訊[29]。

二、資料分析管理子系統：

可用來過濾分析摘要轉換或管理資料，並可去除重覆多餘的資料[29]。

三、虛擬代理伺服器子系統：

提供虛擬的 Cache 空間，並用來架構階層式資訊搜尋與資訊分佈，並可嵌入智慧型代理人(Intelligent Agent)，提供方便的資訊過濾與擷取的功能[29]。

四、WWW 界面軟體子系統：

是一些界面程式，用來將 GAIS 搜尋軟體架設在 WWW 站上[29]。

五、索引查詢子系統：

GAIS 系統最重要的核心軟體，它提供高效率的資料索引與強大的搜尋功能[29]。

表 2-2：GAIS 整理表

項目	GAIS
搜尋型態	網頁
搜尋格式	htm、html、xml、txt
搜尋功能	基本搜尋
搜尋型式	http://gais.cs.ccu.edu.tw/GAIS32/search.php?query=&l=big5&g=H&s=10 l=big5－繁體字型 g=H－網頁介面參數 s=10－結果顯示幾頁，範圍 0 至 20 &－AND 間隔各參數

字數上限	中文無，英文無
查詢字的參數	query－可任意輸入字或句子 兩個字詞以上或句子，中間可用+或%20(利用 ASCII)隔開
特性	1.可以容錯搜尋或近似搜尋 2.提供中文同音搜尋 3.可以做全文檢索、欄位檢索或兩者混合檢索 4.可以使用自然語言檢索 5.可使用有序性之布林檢索(Ordered Boolean) 6.可使用正規表示式(Regular Expression)檢索 7.索引空間極省而且可調 8.可檢索巨量資料，且檢索速度快 9.多用途與富彈性

圖 2-4：GAIS 搜尋介面

參、Google

Google於1998年由創辦人Larry Page和Sergey Brin在史丹佛大學研發出來，搜尋方式是使用精密的內文比對技術，只會找回

包含全部搜尋字詞的網頁，最大特色為可選擇查尋時的語言範圍，並提供網頁排行榜「PageRank」突破了傳統網路分類的概念，PageRank是其核心工具，而檢索結果是將網頁重點簡介，標明出所有的關鍵字，方便快速查閱[30]，如表2-3與圖2-5所示。

表 2-3：Google整理表

項目	Google
搜尋型態	網頁、圖片、新聞、論壇、網頁目錄
搜尋格式	rtf、ps、pdf、xls、ppt、doc、txt
搜尋功能	基本與進階搜尋
搜尋型式	http://www.google.com/search?hl=en&lr=&q= hl=en－英文字型 lr－網頁介面參數 &－AND 間隔各參數
字數上限	中文無，英文 2048 字
查詢字的參數	q－可任意輸入字或句子 兩個字詞以上或句子，中間可用+或%20(利用 ASCII)隔開
特性	1.查詢 PDF 檔案 2.頁庫存檔 3.匯率兌換 4.計算機 5.類似網頁 6.指定網域 7.錯別字改正 8.中英文字典 9.好手氣

圖 2-5：Google搜尋介面

肆、Yahoo

Yahoo是由Yet Another Hierarchical Officious Oracle所組成，由David Filo和楊致遠於1994年創立，1995年才變成公司直到2000年轉變成稱霸全球入口網站的龍頭；搜尋結果的排序原則是以所輸入的關鍵字來比對，會列出網站名稱或敘述內容中有符合使用者所置入關鍵字的搜尋結果，相關程度越高會排在越前面，而相關程度的高低是由搜尋系統程式運算，以各種因子複合計算出來，並無絕對的單一因子[32]，如表2-4與圖2-6所示。

表 2-4：Yahoo整理表

項目	Yahoo
搜尋型態	網頁、圖片、視訊、新聞、網頁目錄、購物
搜尋格式	htm、html、pdf、xls、ppt、doc、xml、txt
搜尋功能	基本與進階搜尋
搜尋型式	1. http://www.yahoo.com/ 2. http://search.yahoo.com/search?p=&fr=FP-tab-web-t&toggle=1&cop=&ei=UTF-8 3. http://search.yahoo.com/?fr=FP-tab-web-t 4. http://search.yahoo.com/search?ei=UTF-8&fr=FP-tab-web-t&p= 1 轉成 2(www.yahoo.com) 3 轉成 4(search.yahoo.com/?fr=FP-tab-web-t) fr=FP-tab-web-t －網頁介面參數 ei=UTF-8 －字集編碼 toggle=1&cop=－般參數 &－AND 間隔各參數
字數上限	中文無(www.yahoo.com) 100 字(search.yahoo.com/?fr=FP-tab-web-t) 英文無(www.yahoo.com) 100 字(search.yahoo.com/?fr=FP-tab-web-t)
查詢字的參數	p－可任意輸入字或句子 兩個字詞以上或句子，中間可用+或%20(利用 ASCII)隔開
特性	1.相關詞提示 2.英文搜尋詞拼字校正 3.多元資料類型搜尋 4.自動拆字功能 5.查詢 PDF 檔案 6.指定網域

圖 2-6：Yahoo搜尋介面

表2-5為本研究所整理AltaVista、GAIS、Google、Yahoo四個著名搜尋引擎比較表，其中自然語言查詢是根據各搜尋引擎之特性所獲得[28][29][30][32]，搜尋引擎排名來源是[26]。

表 2-5：AltaVista、GAIS、Google、Yahoo比較表

	AltaVista	GAIS	Google	Yahoo	
搜尋方式	以關鍵字查詢為主	以關鍵字查詢為主	以關鍵字查詢為主	以分類目錄瀏覽為主	
搜尋字數上限	中文 800 字 英文 800 字	中文無 英文無	中文無 英文 2048 字	中文無 中文 100 字	英文無 英文 100 字
搜尋格式	html、pdf、ppt、doc、xml、txt	htm、html、xml、txt	rtf、ps、pdf、xls、ppt、doc、txt	htm、html、pdf、xls、ppt、doc、xml、txt	

類似查詢	有	有	有	有
自然語言查詢	無	有	有	無
多國語言查詢	有	中文、英文	有	有
欄位查詢	url	url	link、related	title、url
刪除重複網址	無	無	無	無
刪除無效連結	無	無	無	無
刪除廣告	無	無	無	無
搜尋引擎排名	全球第 1	台灣第 6	全球第 6	全球第 2

第三節　資訊檢索

所謂資訊檢索是屬於電腦科學一部份，擁有自動儲存和擷取文件功能。資訊檢索的行為簡單的說就是將使用者需求資訊與大量已經儲存的資訊作比對，使檢索結果能滿足使用者所需。早期資訊檢索主要是用於資料庫方面；隨著網際網路大幅成長，產生大量結構化與非結構化的資料，不少學者近年來相繼提出不同的方法，期盼能解決諸如此類的問題[6]，而檢索技術主要分成三種：

一、內文比對查詢(text pattern search)：

又稱為全文掃描搜尋(full text scanning)，主要是將使用者欲查詢的關鍵字與文章內容做比對，若有符合條件時結果便回饋予使用者。此種方法優點是儲存容易，因為資訊不做任何處理直接存於資料庫；但在檢索時由於是全文比對，所以當資料量龐大時，效率會大打折扣[6][13][25]。

二、反轉檔案搜尋(inverted file search)：

當資料儲存於資料庫時，會進行斷詞處理，記錄每個關鍵詞所存在的文章，利用此文章關鍵詞做為文章的索引值；當查詢者輸入關鍵詞時，搜尋引擎不直接比對全文，而是藉由索引值，找到含有此字彙的文件資料。此種方法可解決前述之缺失，由於需要建立索引表，相對會佔用較多的資料儲存空間[6][13][25]。

三、特徵搜尋(signature search)：

使用重疊編碼(superimposed coding)的方式，來建立文件的特徵，主要是先過濾不符合搜尋條件的文章，再針對初步過濾後的文章進行詳細的比對[6][13][25]。

第四節　網路文件自動摘要

人與人之間是透過語言溝通，傳遞彼此的想法與觀點；對於摘要而言，能撰寫出反應原文所要傳述的意旨是很重要之議題。由於全球資訊網蓬勃發展，資訊檢索技術已逐漸被應用到研究自動摘要產生的

適用性，進一步有學者探討應用於超連結文件方面，自動摘要是由電腦自動地從原始資料中萃取出重要資訊的過程[6][13]。摘要主要目的是產生一個言簡意賅的文件描述，它應比文件簡短且標題更具敘述性，有學者實驗出一篇合適的摘要大約占全文的20％至30％是為一般大眾所能接受[6]。所以評估摘要之優劣準則應包含下列四點[6][13][14]：

一、頻率關鍵詞法：

動詞與名詞是句子的核心部份，文件中每一個動詞與名詞皆視為重要詞彙，而詞彙的重要程度，則視該詞彙在文件中所發生次數多寡[6][13]。

二、標題關鍵詞法：

一篇文章的標題往往選取與主題相關的字詞所組合而成，因此出現在標題的字詞要給予較高的權重值[6][13]。

三、位置法：

一篇文章最重要的部分大部分位於文章首段與末段；學者曾指出簡單的摘錄文件中的前60、150或250個詞彙，便達到了90％以上的可接受度[6][13]。

四、標籤線索法：

超文件提供某些特殊標籤，如：斜體字、粗體字、底線與大小寫字體，都可以呈現相關重要的訊息[6][13]。

以上四準則乃需要依實際情況去做調整，如：吳郁瑩與邱立豐兩位研究者紛紛加入一些主觀想法以解決所遇的問題。

至於網路文件自動摘要的形成步驟，Ronald Brandow學者將網路文件自動摘要區分為統計式樣本分析、特徵字選取、句子權重加總計算及句子選取四大步驟[6]；黃純敏博士則分成六個步驟：擷取網路超文件、超文件分析作業、關鍵詞萃取、句子權重計算、重要句子選取及產生自動摘要[6]；總結上述網路文件自動摘要的基本元素包含了斷詞、句子權重計算、重要句子選取並鏈結三個元素。

壹、 相似度(Similarity)

一般要了解句子間之關聯程度，可利用特徵字來求彼此的相似性，其作法會設定一個門檻值做為基準，判斷是否已達到此基準值，換言之相似度大小是藉由計算兩句子所擁有共同特徵字所占的比重而斷定，表 2-6 列出常用的相似度計算公式：

表 2-6：相似度計算公式

Similarity Measure sim(X,Y)	Evaluation for Binary Term Vectors	Evaluation for Weighted Term Vectors
Inner product	$\lvert X \cap Y \rvert$	$\sum_{i=1}^{t} x_i \cdot y_i$
Dice coefficient	$2\frac{\lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert}$	$\frac{2\sum_{i=1}^{t} x_i \cdot y_i}{\sum_{i=1}^{t} x_i^2 + \sum_{i=1}^{t} y_i^2}$

Cosine coefficient	$\frac{\lvert X \cap Y \rvert}{\lvert X \rvert^{1/2} \bullet \lvert Y \rvert^{1/2}}$	$\frac{\sum_{i=1}^{t} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{t} x_i^2 \bullet \sum_{i=1}^{t} y_i^2}}$
Jaccard coefficient	$\frac{\lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert - \lvert X \cap Y \rvert}$	$\frac{\sum_{i=1}^{t} x_i \cdot y_i}{\sum_{i=1}^{t} x_i^2 + \sum_{i=1}^{t} y_i^2 - \sum_{i=1}^{t} x_i \cdot y_i}$
$X=(x_1, x_2, ..., x_t)$ $\lvert X \rvert$=number of terms in X $\lvert X \cap Y \rvert$=number of terms appearing jointly in X and Y		

(資料來源：[1])

表中的Inner product、Dice coefficient、Cosine coefficient與Jaccard coefficient共同特點是它們的分子都有兩個資料交集，等同計算詞彙共同出現在兩篇文章的程度，其中Cosine coefficient是最常應用的方法[1]，適用於任兩句子相似度運算，至於本研究是採用Dice coefficient方式來計算領域實體論中詞彙之間的相似程度，由文獻[8]可得知計算兩個句子中的某詞彙相似度可利用Dice coefficient最為適合。

第五節　實體論(Ontology)

Ontology最早是由哲學家所提出，可稱為「本體論」或「實體論」，是指用來處理生命體或現實事物本質之存在理論，日後被延伸應用在很多領域方面，乃用以描述知識或表現知識，然而其定義、表達方式

並沒有一致的標準[5]。一般都採取Gruber學者所定義的規則，實體論是一種對某一個概念的詳細描述，包括對於概念、關聯、實體的描述；本體論能清楚定義出所欲表達的概念，主要目的可用於知識分享與再利用[19]。

當使用實體來描述特定領域下的知識，可以看成是概念(concept、object或class）、屬性(attribute、property、slot或role）、關係(relation)與實例(instance)這四元素之組合，分別詳述如下[16]：

一、概念：

以多個底層物件所組成範圍，如由多個字彙所組成之集合，這個集合能夠作為一個概念性描述，透過這個集合能讓電腦了解到概念所代表的意思[16][22]。

二、屬性：

用來輔助描述一個物件，描述出該物件的特性或特徵，物件與物件會存在著某些關係，而且各自擁有不同之屬性，可協助得知每個概念與概念間關係，便於推知整體的重要性高低[16][22]。

三、關係：

用來更清楚地表達概念，所以通常在實體架構中最底層的部份是用來定義實例[16][22]。

四、實例：

若只使用物件與其屬性來清楚描述特定知識領域內的概念與結構是不足夠地，所以建構出整個實體後，除了清楚描述出物件與物件屬性之外，還可以為這些物件定義其彼此間所有的關係[16][22]。圖2-7為一個簡單的本體論架構例子[16]。

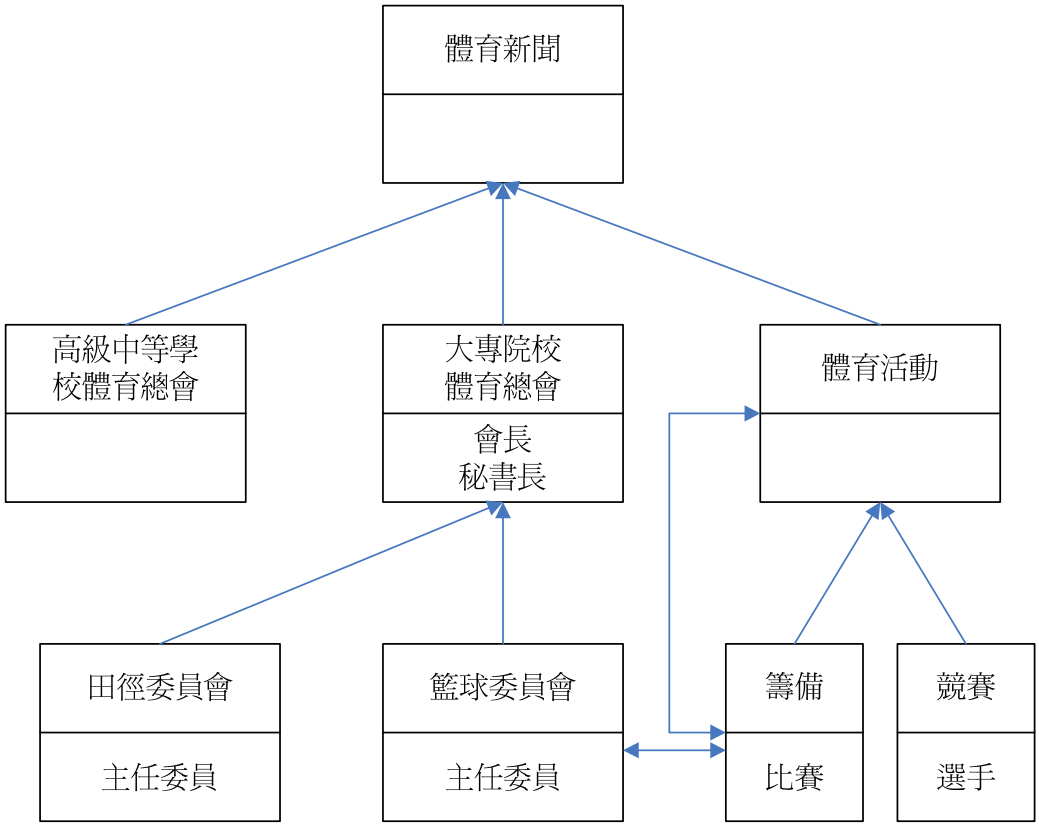

圖 2-7：體育新聞之ontology架構

上圖是體育新聞的實體架構，體育新聞是這個實體中最上層的概念，其中高級中等學校體育總會、大專院校體育總會與體育活動都是屬於體育新聞的一種，跟上層存在著「is-a」的關係，都隸屬體育新聞下所衍生出之概念；而在大專院校體育總會中，會長與秘書長為其屬性，表示有二種職位，可利用屬性來描述其所屬概念下的某些特徵與

特性；然而在體育新聞下的大專院校體育總會概念中，分別由田徑委員會及籃球委員會所組成，彼此便存在一種「part-of」的關係，且各自有其屬性來表達委員會內的職稱。

然而一個實體發展應包含下面四個步驟，藉由這些步驟，即可建構出一個領域實體論[5]。

(一)、定義ontology中的class[5]。

(二)、定義class與class之間的階層關係 (subclass- superclass) [5]。

(三)、定義class中的屬性，並且說明對於屬性值的限制[5]。

(四)、將instance的屬性值填入[5]。

綜合上述本體論是用來定義某個領域中的基本概念及彼此關聯，其主要用意是為了讓電腦更容易閱讀這些知識，讓某特定領域的知識與資訊能夠以人與電腦皆可理解的架構來描述與呈現，並清楚地說明抽象資料與概念間關係，使得資訊能搭配電腦系統以支援知識分享及再利用，因此實體內所定義元素皆是代表特定領域內的相關知識，理論上應該由領域下的領域專家(domain expert)來制定，但已有相關研究提出自動化或半自動化建構實體的方式與機制[5][16]。

目前本體技術被廣泛應用於資訊領域上，例如智慧代理人系統、知識管理系統、電子化商務系統，而且藉由概念性模組構成的實體來精確定義出不同符號或詞彙意義[5][16]。另外有些研究中則指出語意網

路，也隸屬於其中一項技術，期盼利用實體概念來表達其完整語意性，輔助機器代理人於語意網尋覓最佳解，以解決人們所遭遇之難題[16]。

第三章、網路整合型搜尋引擎機制架構

本研究將其流程架構分為三個部分：前處理機制(concept transformation)、整合型搜尋機制(integration searching)與後處理機制(format standardizing、filter & ranking、web page collection)，其中又以後處理機制的過濾與排序機制(filter & ranking)為重點核心研究，如圖3-1所示。

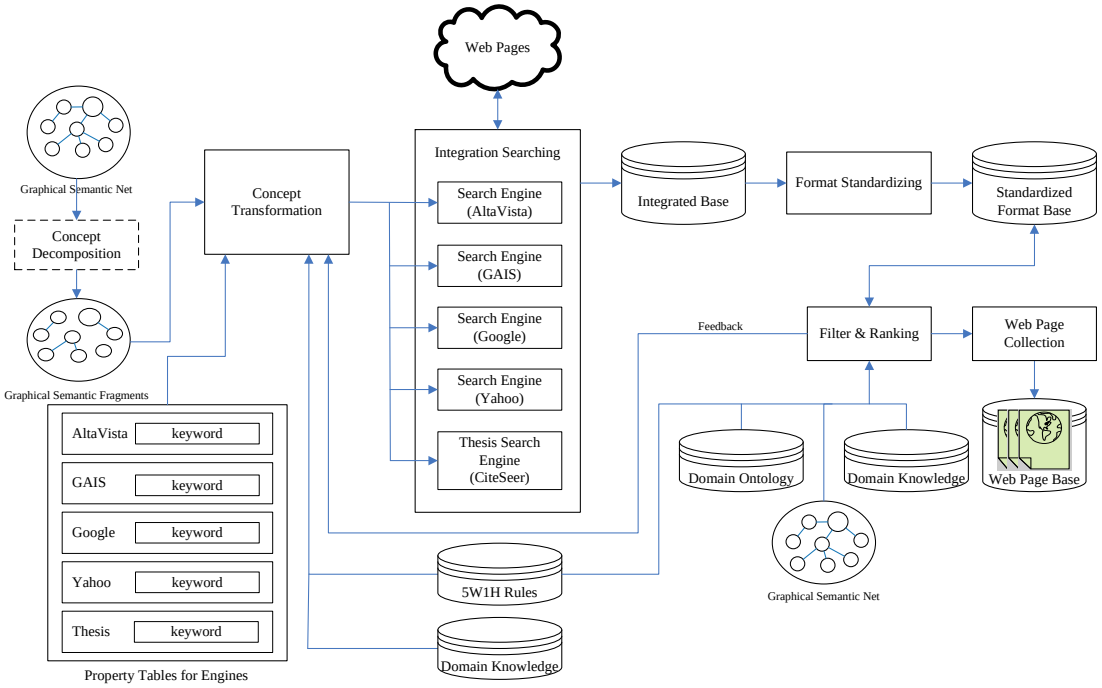

圖 3-1：網路整合型搜尋引擎機制架構

第一節　前處理機制

前處理機制會依輸入的圖形化語意網，針對關鍵字進行處理將其組合成有意義之字組，而且置換成符合每具搜尋器的搜尋語法，其步

驟如下：

一、concept transformation：

將類似網狀結構的圖形化語意網輸入至 concept decomposition mechanism，主要功能是把完整圖形化語意網進行分割的動作，分成多個有意義的子概念，此作用是為避免完整的概念於網路上無法找到符合的結果。簡單來說，圖形化語意網是根據使用者的問題解析並過濾無意義部份，搭配 5W1H(who、why、what、when、where、how)規則及意圖推算出最佳語意網，彙整出有關此問題的關鍵字集；此外透過實體論找出另一組關鍵字，其優勢可讓不同特性的概念與概念間，依某種關係尋找所需的語意內容，建立起真正可表示使用者意圖的關鍵字。因此搜尋的項目分別包含兩類，一是根據 5W1H 規則與意圖所推算出的語意網，二是本體論的語意網，稍作彙整後將傳遞至本研究所選定的四具著名搜尋引擎(AltaVista、GAIS、Google、Yahoo)進行資訊搜尋。

此機制包含 keyword processing 與 specialized keyword composition 兩個部份；keyword processing 是針對語意網中的關鍵字進行處理，經由領域知識(domain knowledge)與 5W1H 規則判斷形成關鍵字集(keyword set)，其判斷方法為下列公式 1[8]：

$$Sim(W_1, W_2) = \frac{2 \times |S(W_1) \cap S(W_2)|}{|S(W_1)| + |S(W_2)|} \quad \text{……(公式 1)}$$

W_i：中文詞彙

$S(W_i)$：將中文詞彙W_i拆解成詞素，所得的詞素集合

$|S(W_i)|$：詞素集合 $S(W_i)$長度

Specialized keyword composition 是依據每具搜尋引擎之特性或屬性特徵，透過程式客製化成符合搜尋語法，形成領域關鍵字集(domain keyword set)。關鍵字配合布林函數 AND 符號(&)作排列組合，但排除重複情況及零排列，計算出總共有幾種，若檢索無解則採用關鍵字遞減方式逐一搜尋，如有三組關鍵字分別為國小、四則運算及乘法，先利用三組字組合出關鍵字，從中擇其一較貼切原意的字組優先搜尋；無解則利用兩組關鍵字或單組關鍵字進行排列組合，再置入搜尋器搜尋；若檢索依然呈現無效結果便啟動後處理機制中的回饋機制(feedback)，其內容將於過濾與排序機制中詳細說明，其他依此類推繼續執行前述步驟，如圖 3-2 所示。

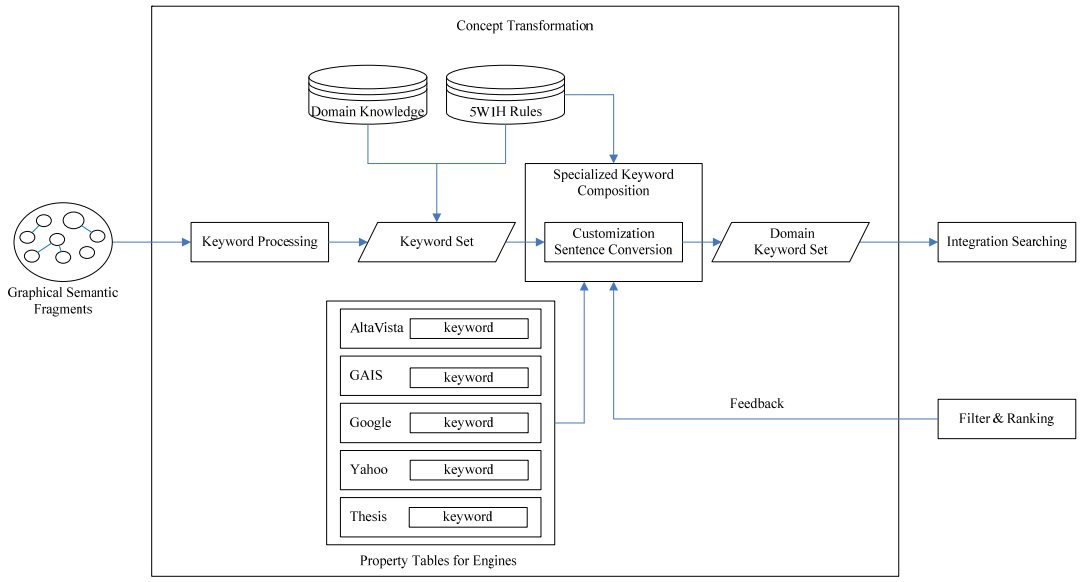

圖 3-2：concept transformation 機制

第二節　整合型搜尋引擎機制

利用前處理機制產出的領域關鍵字集，透過程式轉換成搜尋器句型，其後放入四具引擎進行網路檢索。

一、integration searching：

每具搜尋引擎至全球資訊網檢索，其搜尋結果經由彙整將被置入 integrated base，由於網路上搜尋的資訊大多以非結構為主，容易造成系統負荷與讀取誤判，為求資訊一致性，所以每則資訊經處理且儲存於資料庫，其資料表欄位包含搜尋引擎名稱、網址、關鍵字、標題、標籤、詞彙數目、摘要位置、摘要與原始文件等訊息。

此機制會設計一個代理人(agent)進行自動化搜尋，其代理人應包含前端、中端與後端[9]；前端則由程式所組成，程式可依研究需要的資

訊項目，如標題及網址，下指令給系統搜尋此類資訊，中端則負責溝通前後端，使用適當的通訊協定作為聯繫並管理整合搜尋後資訊，後端則屬於資料庫，把每則資訊作簡單處理並存入integrated base，而且對將來欲儲存新的關鍵字資訊可作分析判斷，若有重複則剔除，避免後處理機制工作量龐大，執行速度緩慢，相對而言可提高效率與時間，如圖3-3所示。

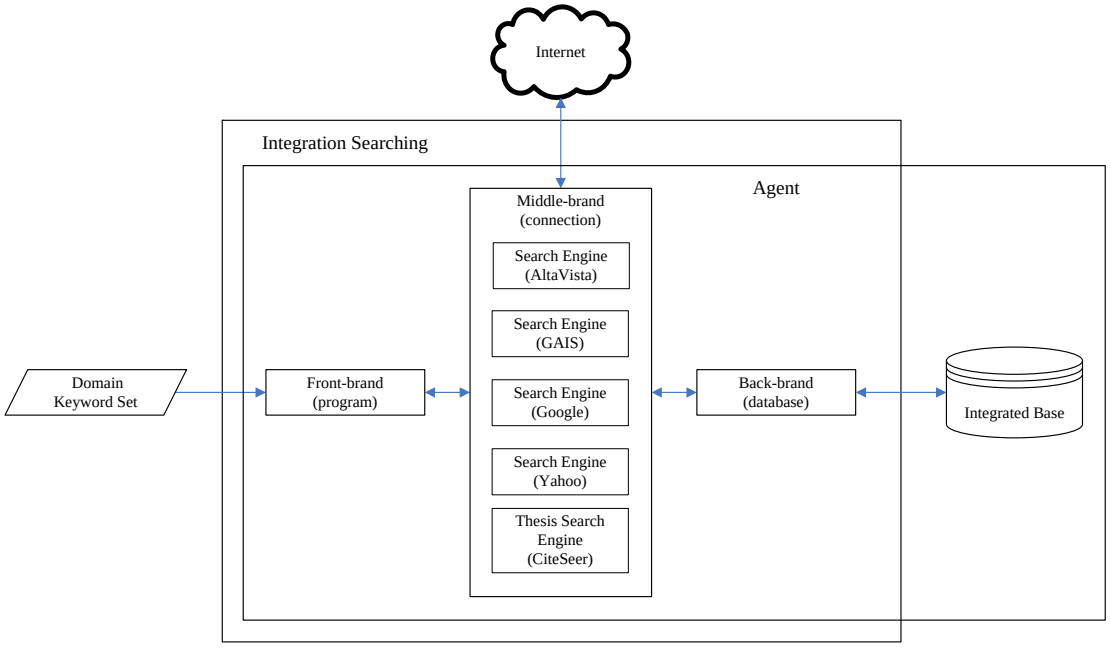

圖 3-3：integration searching機制

第三節　後處理機制

針對後處理機制檢索後的資訊格式重新整理，使其有一致性，接續藉由本研究所設計比對演算法與排序演算法計算每則摘要之資訊含量及權重值，同時訂定門檻值作排序，其後則應用領域實體論(domain

ontology)概念處理摘要中涉及同義詞部份，這是本研究重點核心之一，最後經由判斷並獲取符合原意的網頁，以便於知識萃取。

當整合型搜尋引擎檢索出所有的資訊，會進一步將資訊處理成一致性格式，並暫存於整合型資料庫，其後步驟如下：

壹、 格式標準化機制(format standardizing)

主要透過程式執行任務；網路上的資料格式大多以HTML(Hyper Text Markup Language)與PDF(Portable Document Format)呈現且內容都是非結構與半結構性，利用XML(Extensible Markup Language)可達成結構化目的[2][31]；XML是一套資料的描述語言，主要是用來設計網頁中可攜帶結構化的資訊，並且允許使用者可以自行定義和它們文件相關的標籤，同時可透過自訂標籤、屬性、XML schema與DTD(Document Type Definition)[2][30]，對於每則摘要的相關資訊進行定義成為標準格式，可謂format standardizing，隨後儲存至standardized format base。

貳、 過濾與排序機制(filter & ranking)

如圖3-4所示，將每則資訊已格式標準化的摘要與標題進行一連串處理，主要依據圖形化語意網和領域知識所提供的知識為基準，搭配領域實體論與原意，透過演算法運算資訊含量、權重值與相似度；其中相似度是利用本研究所建立的同義詞規則，計

算句子中詞彙與資料庫中詞彙的相似程度，可以解決意思同描述不同之摘要問題；最後給予適當門檻值並排列順序且逐一與原意比對找出真正符合的資訊。

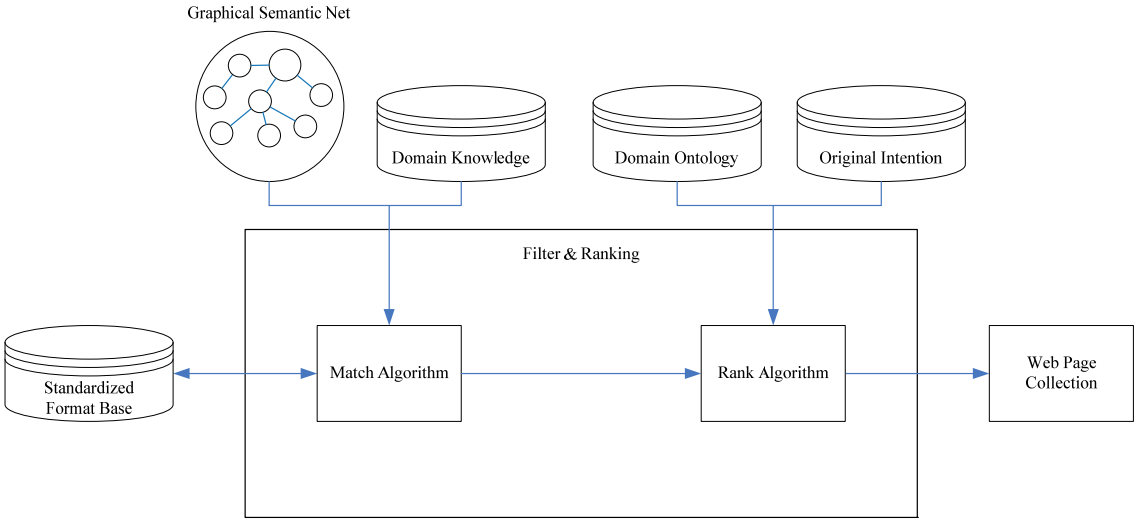

圖 3-4：filter與ranking機制

參、 比對演算法(match algorithm)

演算流程與演算法如圖3-5及圖3-6所示。首先系統過濾網址，藉由程式來判斷URL是否有重複，重複則直接刪除資料庫中的相關資料列，反之針對摘要與標題中含有關鍵字的句子處理成可比較單位與N元詞(N-gram)，由N個字所組成則稱為N元詞，利用N元詞特性及句子結構建立句子中所有的可比較單位[6][15]，其後運算每則摘要之資訊含量值，資訊含量相關計算公式如公式2，公式3所示[15]。

$$R(i,j)=\frac{(U_i \cap U_j)*2}{U_i \cup U_j} \cdots\cdots\cdots\cdots\text{(公式2)}$$

$$S(i)=\sum_j R(i,j) \cdots\cdots\cdots\cdots\text{(公式3)}$$

U_i：中文詞彙

$U_i \cup U_j$：所有句子I與句子J中相同的可比較單位

運算時必須遵循下列四個比較法則[15]：

(一)、N元詞只能與N元詞比對

(二)、標注詞只能與標注詞比對

(三)、每個詞只能比對成功一次

(四)、詞的比對不考慮順序性

當摘要與標題中的句子分別處理成N元詞與可比較單位，其後依數學領域知識為基準，運算出摘要與標題句子的各別R值，若摘要句子有兩句以上，也是計算每句的R值，其後將摘要句子的R值與標題句子的R值相加，便產生資訊含量值，而且每篇摘要的資訊含量會分成高、中與低三類，值低於或等於0.1則為低資訊含量值，系統會立即剔除，高於或等於0.3則為高資訊含量值，其他部份則是中等資訊含量值。

例如：某摘要的關鍵字是「九九乘法」，數學領域知識庫中也有「九九乘法」，其可比較單位有九九、九乘、乘法共3個，標

注詞1個，共有4個比較單位；其標題句子：來玩九九乘法表，可比較單位有來玩、玩九、九九、九乘、乘法、法表共6個，標注詞為九九乘法，共有7個比較單位；摘要句子一「一開始我們想解出九九乘法表中」，可比較單位有8個，標注詞1個，共有9個比較單位；摘要句子二「更發現九九乘法表的奧妙與許多有趣的玩法」，可比較單位有13個，標注詞1個，共有14個比較單位。

分別計算每句的R值，R(標題) = (1*2)/(7+4)=0.18118，R(摘要一) = (1*2)/(9+4)=0.1538，R(摘要二) = (1*2)/(14+4)=0.1111；其後將R值相加便產生資訊含量值，S(某摘要)=0.1818+0.1538+0.1111=0.4467，此值大於0.3，故為高資訊含量的摘要。

簡言之上述是將不符合的項目自動地刪除，剩下符合項目，換句話說不符合項目是指資訊含量低與重複網址，反之符合項目包含資訊含量高與中等的摘要，統稱為比對演算法(match algorithm)。

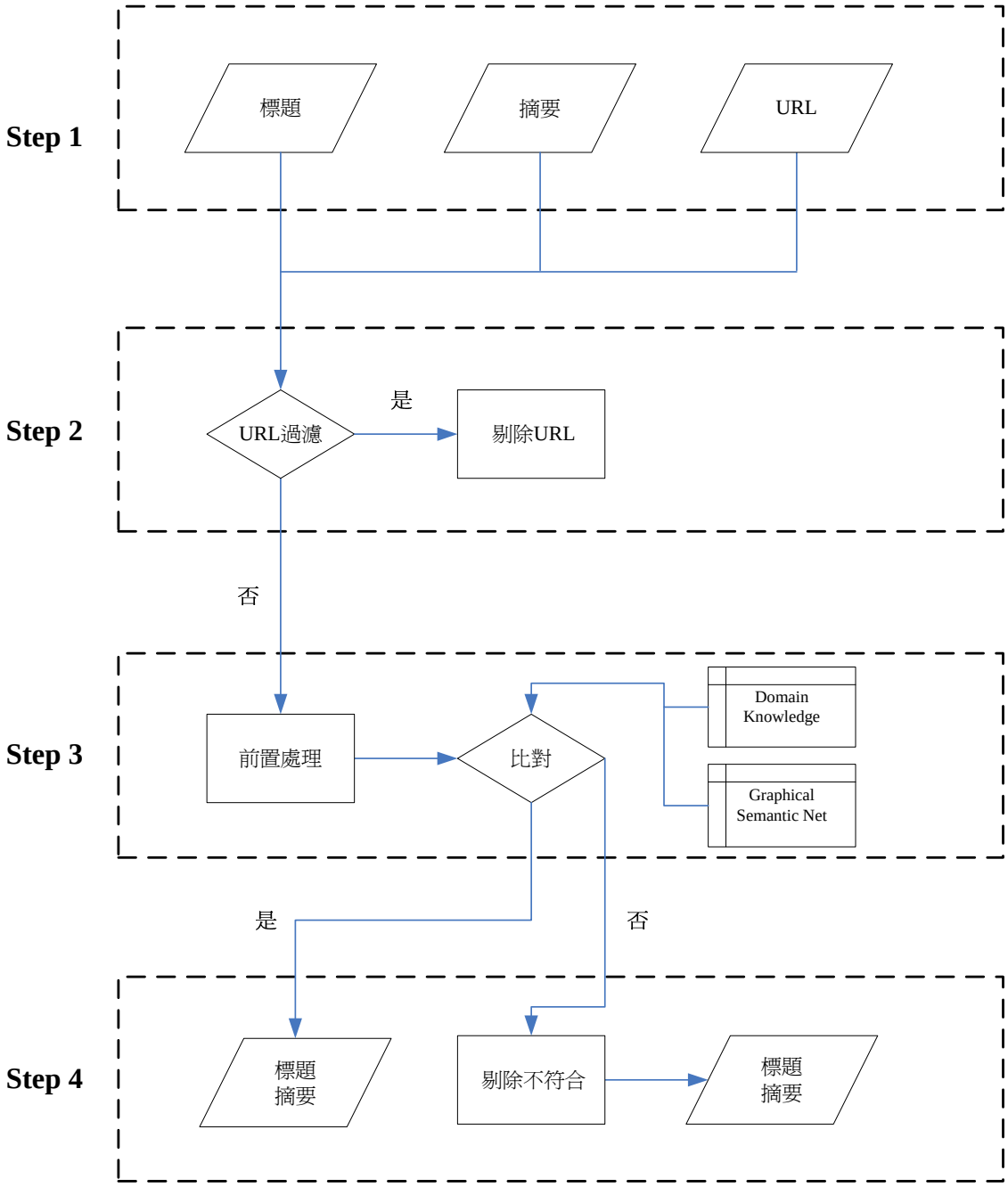
Step 1
標題
摘要
URL
Step 2
URL過濾
是
剔除URL
否
Step 3
前置處理
比對
Domain Knowledge
Graphical Semantic Net
是
否
Step 4
標題
摘要
剔除不符合
標題
摘要

圖 3-5：比對演算法流程

Match Algorithm

Input：the headline, abstract, and URL of Standardized Format Base

Output：the content of information is high and medium-sized summary

```
void match (int n, x, y,
                number headline [], abstract [], URL[],sentence R[][],
                content of information S[], value V[])
{
    int i;
    int j;
    double v;
    double a;
    double b;
    double c;
    a=S[i]; b=S[i]; c=S[i];
    a=low; b=mid; c=high;
    for (int u=0; u<n; u++) {
        extraction URL;
        u1[u]=URL[u];
        u2[u]=u1[u];
        if(u1= =u2) {
        delete u1[u];
        }
        else {
        deliver to n-gram;
        }
    }
    for (int x=0; x<n; x++) {
       for (int y=0; y<n; y++) {
            extraction headline and abstract;
            he[x]=headline[x];
            ab[y]=abstract[y];
            headline and abstract dealt with n-gram;
            R[i][j]=(U_i ∩ U_j) *2/U_i ∪ U_j;
            S[i]=sum R[i][j];
            if (a value S[i] with lower than V[v]) {
               S[i]=low;
               delete S[i];
```

```
        }
        else if (a value S[i] with higher than V[v]) {
          S[i]=high;
          deliver to Rank Algorithm;
        }
        else {
          S[i]=mid;
          deliver to Rank Algorithm;
        }
      }
    }
}
```

圖 3-6：比對演算法

n表示第某篇資料，x表示標題，y表示摘要，u表示URL，R表示句子，S表示資訊含量，V表示門檻值。其步驟說明如下：

步驟一：取出已經標準化的標題、摘要與 URL。

步驟二：進行 URL 過濾，是則進行前置處理，否則剔除相關資訊。

步驟三:標題與摘要先處理成 N 元詞與比較單位,接續跟 Domain Knowledge 與 Graphical Semantic Net 進行資訊含量之運算。

步驟四：是則傳送給排序演算法，否則將資訊含量低的標題與摘要剔除。

肆、 排序演算法(rank algorithm)

演算流程與演算法如圖3-7及圖3-8所示。首先將關鍵字次數、摘要位置與標題標籤取出，進行權重值運算並註記於標準格

式資料庫裡，同時摘要進行同義詞的分析判斷，因為此部分摘要有涉及描述方式不同但本意卻相同情況，所以可給予相對等的權重值，至於計算權重的法則有下列五點，前四項已於第二章第四節介紹，本章節不加以闡述：

1、頻率關鍵詞法。

2、標題關鍵詞法。

3、位置法。

4、標籤線索法。

5、領域實體論法：

句子是由詞彙所組成，但詞彙之間會存在著某些特定關係；同義詞包含廣義上的相關詞與狹義上的同義詞，前者是指某篇摘要描敘不同，但意義與原意類似的詞[3][4]；後者是指某篇摘要述敘不同，但意義與原意相同的詞[3][4]，本研究透過數學實體與國教專業社群網[27]，蒐集國小數學詞彙並觀察詞彙拆解後的意思，設法從中建構出相關詞與同義詞的詞彙規則，如：幾何與代數同屬於數學；先乘除後加減是意指四則運算等諸如此類關係，如表3-1與3-2所示。

表 3-1：相關詞規則表

規則一(縮寫相關詞)	
規則二(單字相關詞)	幾何學－數學
規則三(多字相關詞)	統計與機率－數與量
規則四(中英文相關詞)	

表 3-2：同義詞規則表

規則一(縮寫同義詞)	國小一年級加法－小一加法
規則二(單字同義詞)	數－值
規則三(多字同義詞)	先乘除後加減－四則運算
規則四(中英文同義詞)	

目前研究彙整出各四條規則分別為縮寫相關詞、縮寫同義詞、單字相關詞、單字同義詞、多字相關詞、多字同義詞、中英文相關詞及中英文同義詞，經實際觀察與蒐集，沒有縮寫相關詞的詞彙規則，由於所依據的實體概念以中文為主，所以中英文相關詞與中英文同義詞暫時不予研討，日後若涉及到此部分，會逐一建立其餘規則。簡言之領域實體論法有五大步驟：

(1)、從Domain Ontology中蒐集整理相關詞與同義詞並建立其規則。

(2)、依摘要中的關鍵字選定適合之規則進行運算或比對，關鍵字可能是N元詞或標注詞。

(3)、適用規則一(縮寫同義詞)則利用詞彙之詞素與關鍵字計算相

似程度，其公式陳列於本章第一節，如公式1所示，判斷此值位居何種值域中，給予相對等的權重。

(4)、適用規則二與三(單字相關詞、單字同義詞、多字相關詞、多字同義詞)則利用自行建立的關係表直接判斷，其後給予已訂立於資料庫中的權重值。

(5)、完成一至四步驟，即可得知同義詞權重(Synonym Weight)。

規則一是縮寫同義詞，依摘要的句子經計算產生值若相似度小於0.2表示不是同義詞，所以會刪除此摘要，反之就是符合規則，因為縮寫代表詞彙減少轉變成另一種形式來闡述相同的意思，故權重值會給予5；適用規則二與三直接給予已定義的SW值，單字相關詞與單字同義詞值分別為6，多字相關詞與多字同義詞分別為7，單字是指單一或單獨的辭彙意思，多字則是可拆解成兩個以上的辭彙意思。

當前述逐一完成後將運算總體權重值，如公式4所示[13]：

$$SCORE = \sum_{k=1}^{n} TP_k + PW + \sum_{l=1}^{m} T_l W + SW \text{.....(公式4)}$$

TP_k：摘要中第k個詞彙的權重

n：重要詞彙總數

PW：位置權重

$T_l W$：第l個加權詞彙的標題與標籤權重

m：加權詞彙總數

SW：同義詞權重

SCORE：總得分

權重值是參照文獻加上測試及觀察後才設立權重值[13]，TP為關鍵字在一篇文章中出現的次數，理論上詞彙出現愈多重要，其值定為5；PW為環繞關鍵字的摘要位居整篇何處位置，出現於第一段值定為10，第二段為5，第三段以後為1；TW則是由標題與標籤所構成，標題於網路超文件分別由<TITLE>、<H？>與<FONT SIZE>所表示，<TITLE>用於說明文件主題，設值為5，<H？>分為六種等級，此處只針對<H1>至<H4>作處理，設值為3，<FONT SIZE>用來呈現文件主題大小字體，此處設定<FONT SIZE>大於3以上才給予權重值3，<FONT SIZE>小於3表示不重要資訊，故不作處理，標籤是用以呈現文件重要性的資訊，分為<META>、<B>、<I>、<EM>、<STRONG>、<BLINK>、<BIG>，<META>會記載文件額外重要訊息，設值為5，其他部份則以特殊方式表達詞彙的重要性，設值為2[13]。

例如：某摘要關鍵字是「九九乘法」，藉由觀察此摘要的超文件檔，關鍵字出現2次，所以TP為5+5=10；包含關鍵字的句子位居整篇第一段與第三段，故PW為10+1=11；摘要的標題為

5+3=8，標籤有9個故值為2+2+2+2+2+2+2+2+2=18，相加之後TW值為26；關鍵字經系統判斷隸屬同義詞庫中的多字相關詞規則，所以SW為7；總體權重值為10+11+26+7=54。

經運算後此處會設定適當門檻值，此值是指取前20或30篇的摘要，取出門檻值以上摘要並由高至低排序，依順序取每筆摘要與原意進行相似度運算判斷是否符合，原意是使用者於討論區描述的句子，判斷方法是利用公式1，計算每句的相似程度，其後相加產生相似度，若相似度小於或等於0.15，表示此篇摘要不符合原意，會主動啟動回饋機制重新組合關鍵字集；相似度大於0.21，代表符合原意可作為解答，至於其他部分則表示部份符合原意，進一步透露此摘要資訊量過低，不適合推薦予使用者，系統會立即刪除。不符合時則擷取下筆資料來檢驗，若判斷結果無一符合立刻啟動回饋機制返回concept transformation，依關鍵字集重新採用詞彙遞減方式來組合，並藉由5W1H規則中映對(mapping)關係找尋有關的關鍵字，其後透過specialized keyword composition客製化成搜尋器語法繼續於網路上檢索，其他則依此類推繼續執行一連串機制與演算法。

例如：某摘要關鍵字是「九九乘法」，摘要句子一「一開始我們想解出九九乘法表中」，摘要句子二「更發現九九乘法表的

奧妙與許多有趣的玩法」,原意是「小朋友不會背九九乘法」,S(摘要一)=2*4/14+10=0.3333，S(摘要二)=2*4/19+10=0.2759，故此篇摘要相似度0.3333+0.2759=0.6092，表示符合原意可作為解答。此外使用者若本欲查詢有關「輕度障礙生對四則運算認知程度」方面的問題，但所檢索的摘要都與原意不相符，導致啟動回饋機制重新組合關鍵字，同時5W1H規則裡卻映對到「數學閱讀障礙程度」，兩者之間看起來存在某種程度的關聯性，因此可利用這對映關係一同進行搜尋，希冀能檢索出符合的答案。上述內容統稱為排序演算法(rank algorithm)。

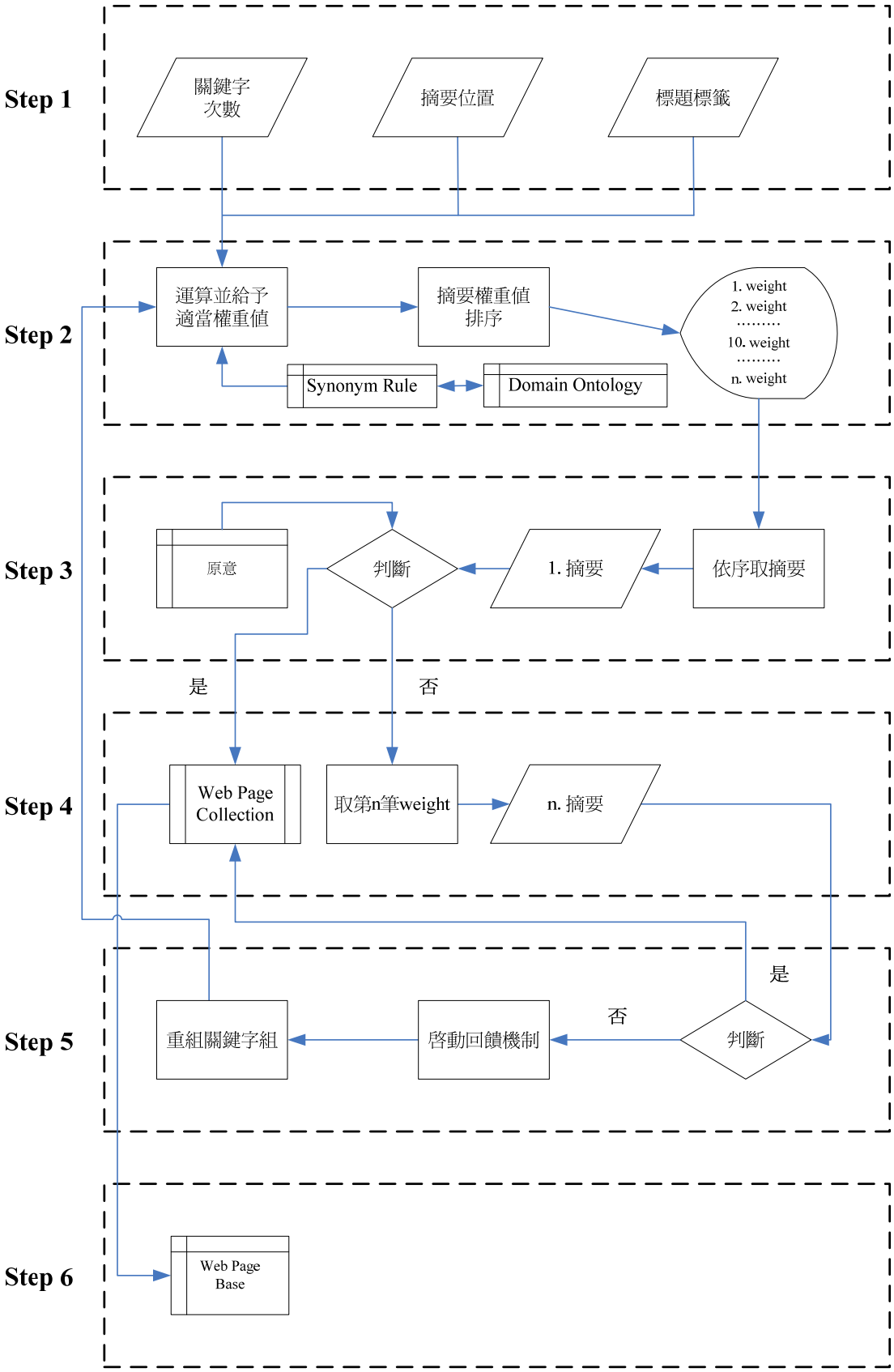
Step 1
關鍵字
次數
摘要位置
標題標籤
Step 2
運算並給予
適當權重值
摘要權重值
排序
1. weight
2. weight
10. weight
n. weight
Synonym Rule
Domain Ontology
Step 3
原意
判斷
1. 摘要
依序取摘要
是
否
Step 4
Web Page
Collection
取第n筆weight
n. 摘要
是
Step 5
重組關鍵字組
啓動回饋機制
否
判斷
Step 6
Web Page
Base

圖 3-7：排序演算法流程

Rank Algorithm

Input：the content of information is high and medium-sized abstract

Output：accord with the webpage of the original meaning

```
void rank (int n, x, y,
            number abstract [], frequency [], location [], headline [], tag[],
            synonym [], value V[], SCORE SC[], TP[] , PW[], TW[], SW[])
{
    int i=0;
    for (int y=0; y<n; y++) {
        extraction y;
        ab[y]=abstract [y];
        if (frequency, location, headline, tag, and synonym does exist) {
           TP[i]= frequency [i];
           PW[i]= location [i];
           TW[i]= headline [i] and tag[i];
           if(synonym is rule 1) {
            synonym [i]= Sim(W1,W2) = 2×|S(W1)∩S(W2)| / |S(W1)|+|S(W2)|;
            Wi =Chinese words;
            SW[i]= synonym [i];
           }
           else {
              offer weight value directly;
              SW[i]= synonym [i];
           }
           SC[i]= Σ_{k=1}^{n} TPk + PW + Σ_{l=1}^{m} TlWt + SW ;
           i++;
        }
    }
       for (int y=0; y<n; y++) {
          fetch SC[i] of each y in accordance with the order;
          if (judge whether its abstract accords with the original meaning) {
             return to start Web Page Collection mechanism;
          }
          else {
             SC[i-1] which fetches the next abstract is implement;
             if (URL[i-1] accords with the original meaning) {
                return to start Web Page Collection mechanism;
```

```
    }
     else {
         there is no answer accorded with;
         start Feedback mechanism;
         reconfigure Keyword Set;
     }
  }
  while (start Web Page Collection mechanism) {
    pick and fetch its real webpage;
    if (webpage accorded with) {
      store on Web Page Base;
    }
  }
}
```

圖 3-8：排序演算法

n表示第某篇資料，x表示標題，y表示摘要，V表示門檻值，SC表示權重值，TP表示詞彙權重值，PW表示摘要位置權重值，TW表示標題與標籤權重值，SW表示相關詞與同義詞權重值。其步驟說明如下：

步驟一：取出已運算的標題與摘要。

步驟二:依五個準則計算得分,給予摘要適當權重值以進行排序。

步驟三:取每筆 weight 的摘要與原意進行相似度運算判斷是否符合。

步驟四：是則啟動網頁取回機制，否則取第 n 筆摘要繼續運算。

步驟五：依第 n 筆摘要符合(是)則將至 Step 4，若無檢索結果(否)

則啟動回饋機制返回前處理機制，重新組合關鍵字集，之後至Step 2 繼續執行。

步驟六：將符合摘要的真正網頁擷取並儲存於網頁庫。

伍、 網頁擷取機制(web page collection)

從網路取回經由格式標準化及過濾與排序的實際網頁內容，並儲存於web page base，資料庫裡包含原始資料檔與經機制處理的檔案。

第四章、系統開發與實作

根據第三章所設計的架構，本研究設計發展強化搜尋引擎的比對與排序機制，此機制會針對相關資訊經由一連串處理運算資訊含量、權重值與相似度，讓使用者最後獲取網路上最貼近原意的網頁知識。

第一節　實驗環境介紹

本實驗與測試是利用下列環境所建造，主要可分為軟體與硬體兩部份。

壹、 軟體

一、 作業系統：Microsoft Windows XP Professional Edition Service Pack 2。

二、 資料庫系統：Microsoft SQL Server 2005 Express Edition。

三、 程式語言：Java。

四、 程式開發平台：Gel RC40、JDK 1.6.0、SQL JDBC 1.1.1501.101。

貳、 硬體

一、 機器：ASUS A8Jc Notebook。

二、 CPU：Intel Duo T2250 1.73GHz。

三、 記憶體：1024MB。

實驗目的主要是探索加入同義詞權重(SW)項目，是否比未使用 SW 計算權重值更能凸顯重要網頁的排名順序，以及其效能有所提升。

第二節　資料來源與限制

由於網路上資訊格式繁多，經實際搜尋發現所需資訊都以 PDF 格式為主，PDF 格式的檔案處理上有眾多限制，因此本研究透過搜尋引擎輸入二十個關鍵字，取得二十篇有關國小數學的摘要，針對摘要進行格式標準化並儲存於標準格式資料庫。

至於資料庫，則建立數學領域知識庫(Math Domain Knowledge Base)包含 289 條國小數學名詞、動詞與專有名詞；數學同義詞規則(Math Correlation Rule)有 151 條規則，分為單字與多字相關詞；數學相關詞規則(Math Synonym Rule)有 76 條規則，分為縮寫、單字與多字同義詞，縮寫是詞彙經縮減所組成；標準格式資料庫(Standardized Format Base)目前先行建立二十篇摘要的相關資訊，如：搜尋引擎名稱、關鍵字、網頁格式、網址、標題、標籤數目及摘要等資訊欄位。

比對演算法的資訊含量值，是經由程式運算後所求得，藉由觀察及實際測試二十篇摘要文章後，分別訂定資訊含量值，圖 4-1 則為區分低、中、高資訊含量值之程式碼。

```
int i = 0;
int k = 0;
double vi1[]= new double[20];
double vi2[]= new double[20];
double vi3[]= new double[20];
double vi4[]= new double[20];
while(rs4.next()) {
  double vi0 = rs4.getDouble("value_of_information");
  //System.out.println(vi0);
  vi1[i] = vi0;
  i++;
}

for(int j=0; j<20; j++) {
  //System.out.println(j + ":" + vi1[j]);
  if(vi1[j]<=0.1) {
    vi2[k] = vi1[j];
    System.out.println("低資訊含量-第"+ (k+1) + "篇值爲" + vi2[k]);
    k++;
    stmt5.executeUpdate("DELETE FROM test WHERE value_of_information <= 0.1");
  }
  else if(vi1[j]>=0.3) {
    vi3[k] = vi1[j];
    System.out.println("高資訊含量-第"+ (k+1) + "篇值爲" + vi3[k]);
    k++;
    stmt6.executeUpdate("UPDATE test SET value_of_information_level ='高資訊含量'WHERE keyword ='牛頓'");
    stmt6.executeUpdate("UPDATE test SET value_of_information_level ='高資訊含量'WHERE keyword ='加法'");
    stmt6.executeUpdate("UPDATE test SET value_of_information_level ='高資訊含量'WHERE keyword ='圓周率'");
    stmt6.executeUpdate("UPDATE test SET value_of_information_level ='高資訊含量'WHERE keyword ='等值分數'");
    stmt6.executeUpdate("UPDATE test SET value_of_information_level ='高資訊含量'WHERE keyword ='比例尺'");
    stmt6.executeUpdate("UPDATE test SET value_of_information_level ='高資訊含量'WHERE keyword ='統計'");
    stmt6.executeUpdate("UPDATE test SET value_of_information_level ='高資訊含量'WHERE keyword ='公因數'");
    stmt6.executeUpdate("UPDATE test SET value_of_information_level ='高資訊含量'WHERE keyword ='先乘除後加減'");
    stmt6.executeUpdate("UPDATE test SET value_of_information_level ='高資訊含量'WHERE keyword ='周長'");
    stmt6.executeUpdate("UPDATE test SET value_of_information_level ='高資訊含量'WHERE keyword ='循環小數'");
    stmt6.executeUpdate("UPDATE test SET value_of_information_level ='高資訊含量'WHERE keyword ='九九乘法'");
  }
  else {
    vi4[k] = vi1[j];
    System.out.println("中資訊含量-第"+ (k+1) + "篇值爲" + vi4[k]);
    k++;
    stmt7.executeUpdate("UPDATE test SET value_of_information_level ='中資訊含量' WHERE keyword ='通分'");
    stmt7.executeUpdate("UPDATE test SET value_of_information_level ='中資訊含量' WHERE keyword ='平行四邊形'");
    stmt7.executeUpdate("UPDATE test SET value_of_information_level ='中資訊含量' WHERE keyword ='分配律'");
    stmt7.executeUpdate("UPDATE test SET value_of_information_level ='中資訊含量' WHERE keyword ='小二乘法'");
    stmt7.executeUpdate("UPDATE test SET value_of_information_level ='中資訊含量' WHERE keyword ='國小四年級數學'");
    stmt7.executeUpdate("UPDATE test SET value_of_information_level ='中資訊含量' WHERE keyword ='等差數列'");
  }
}
```

圖 4-1：資訊含量程式碼

排序演算法中所需要權重值，如：TP、PW、TW，則是參照文獻加上測試及觀察後才設立權重值[13]。同義詞的作法，主要是根據相關及同義規則計算詞彙的相似程度，再給予相對 SW 值。

經由上述所定義的值，加總後便可得知每則摘要的總體權重值，圖 4-2 為加總的原始權重值與新權重值之程式碼。接續會依總權重值進行排序，其後根據排序的摘要分別與原意進行相似度判斷，相似度門檻值是經實際測試二十篇摘要文章後所訂定，圖 4-3 則為二十篇摘要與原意所運算之相似度，並區分其符合程度。

```
int a = 0;
int tp1;
int pw1;
int tw1;
int sw1;
int tp2[] = new int[20];
int pw2[] = new int[20];
int tw2[] = new int[20];
int sw2[] = new int[20];
int oweight[] = new int[20];
int nweight[] = new int[20];
while(rs10.next()) {
  tp1 = rs10.getInt("TP");
  pw1 = rs10.getInt("PW");
  tw1 = rs10.getInt("TW");
  sw1 = rs10.getInt("SW");
  tp2[a] = tp1;
  pw2[a] = pw1;
  tw2[a] = tw1;
  sw2[a] = sw1;
  a++;
}
for(int j=0; j<17; j++) {
   System.out.println((j+1) + "_TP:" +tp2[j]);
   System.out.println((j+1) + "_PW:" +pw2[j]);
   System.out.println((j+1) + "_TW:" +tw2[j]);
   oweight[j] = tp2[j] + pw2[j] + tw2[j];
   System.out.println("第" + (j+1) + "篇原始權重值:" + oweight[j]);
   stmt11.executeUpdate("UPDATE test SET original_weight ='15' WHERE keyword ='通分'");
   stmt11.executeUpdate("UPDATE test SET original_weight ='36' WHERE keyword ='平行四邊形'");
   stmt11.executeUpdate("UPDATE test SET original_weight ='38' WHERE keyword ='牛頓'");
   stmt11.executeUpdate("UPDATE test SET original_weight ='18' WHERE keyword ='加法'");
   stmt11.executeUpdate("UPDATE test SET original_weight ='13' WHERE keyword ='分配律'");
   stmt11.executeUpdate("UPDATE test SET original_weight ='75' WHERE keyword ='圓周率'");
   stmt11.executeUpdate("UPDATE test SET original_weight ='45' WHERE keyword ='等值分數'");
   stmt11.executeUpdate("UPDATE test SET original_weight ='26' WHERE keyword ='比例尺'");
   stmt11.executeUpdate("UPDATE test SET original_weight ='121' WHERE keyword ='統計'");

    stmt11.executeUpdate("UPDATE test SET original_weight ='41' WHERE keyword ='公因數'");
    stmt11.executeUpdate("UPDATE test SET original_weight ='28' WHERE keyword ='小二乘法'");
    stmt11.executeUpdate("UPDATE test SET original_weight ='29' WHERE keyword ='先乘除後加減'");
    stmt11.executeUpdate("UPDATE test SET original_weight ='32' WHERE keyword ='國小四年級數學'");
    stmt11.executeUpdate("UPDATE test SET original_weight ='16' WHERE keyword ='等差數列'");
    stmt11.executeUpdate("UPDATE test SET original_weight ='105' WHERE keyword ='周長'");
    stmt11.executeUpdate("UPDATE test SET original_weight ='47' WHERE keyword ='循環小數'");
    stmt11.executeUpdate("UPDATE test SET original_weight ='47' WHERE keyword ='九九乘法'");

    System.out.println((j+1) + "_SW:" +sw2[j]);
    nweight[j] = oweight[j] + sw2[j];
    System.out.println("第" + (j+1) + "篇新權重值:" + nweight[j]);
    stmt11.executeUpdate("UPDATE test SET new_weight ='15' WHERE keyword ='通分'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='36' WHERE keyword ='平行四邊形'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='38' WHERE keyword ='牛頓'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='18' WHERE keyword ='加法'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='13' WHERE keyword ='分配律'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='75' WHERE keyword ='圓周率'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='45' WHERE keyword ='等值分數'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='26' WHERE keyword ='比例尺'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='121' WHERE keyword ='統計'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='41' WHERE keyword ='公因數'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='33' WHERE keyword ='小二乘法'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='34' WHERE keyword ='先乘除後加減'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='39' WHERE keyword ='國小四年級數學'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='22' WHERE keyword ='等差數列'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='111' WHERE keyword ='周長'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='54' WHERE keyword ='循環小數'");
    stmt11.executeUpdate("UPDATE test SET new_weight ='54' WHERE keyword ='九九乘法'");
  }
```

圖 4-2：權重值程式碼

```
String st6 = new String();
String st7[] = new String[17];
String st8 = new String();
String st9[] = new String[17];
String st10 = new String();
String st11[] = new String[17];
double sim4[] = new double[17];
double sim5[] = new double[17];
while(rs17.next()) {
  st6 = rs17.getString("new_abstract");
  st7 = st6.split(" ");
  if(rs18.next()) {
    st8 = rs18.getString("original_intention");
  }
  if(rs19.next()) {
    st10 = rs19.getString("keyword");
  }
  for(int i=0; i<st7.length; i++) {
     //System.out.println("---"+st7[i]); //列印包含關鍵字的句子
     st9[i] = st8;
     //System.out.println("~~~"+st9[i]); //列印包含關鍵字的原意句子
     st11[i] = st10;
     //System.out.println("@@@"+st11[i]); //列印關鍵字
     sim4[i] = (double)(2*st11[i].length())/(double)(st7[i].length() + st9[i].length());
     sim5[i] = Math.rint(sim4[i]*10000)/10000;
     //System.out.println((i+1)+"^^^"+sim5[i]); //列印每句相似度
     stmt20.executeUpdate("UPDATE test SET sim ='2.1084' WHERE keyword ='統計'");
     stmt20.executeUpdate("UPDATE test SET sim ='1.641' WHERE keyword ='周長'");
     stmt20.executeUpdate("UPDATE test SET sim ='1.52' WHERE keyword ='圓周率'");
     stmt20.executeUpdate("UPDATE test SET sim ='1.0201' WHERE keyword ='循環小數'");
     stmt20.executeUpdate("UPDATE test SET sim ='0.6092' WHERE keyword ='九九乘法'");
     stmt20.executeUpdate("UPDATE test SET sim ='1.2388' WHERE keyword ='等值分數'");
     stmt20.executeUpdate("UPDATE test SET sim ='0.7286' WHERE keyword ='公因數'");
     stmt20.executeUpdate("UPDATE test SET sim ='0.25' WHERE keyword ='國小四年級數學'");
     stmt20.executeUpdate("UPDATE test SET sim ='0.7514' WHERE keyword ='牛頓'");
     stmt20.executeUpdate("UPDATE test SET sim ='0.5024' WHERE keyword ='平行四邊形'");
     stmt20.executeUpdate("UPDATE test SET sim ='0.9661' WHERE keyword ='先乘除後加減'");
```

```
      stmt20.executeUpdate("UPDATE test SET sim ='0.2286' WHERE keyword ='小二乘法'");
      stmt20.executeUpdate("UPDATE test SET sim ='0.6768' WHERE keyword ='比例尺'");
      stmt20.executeUpdate("UPDATE test SET sim ='0.3333' WHERE keyword ='等差數列'");
      stmt20.executeUpdate("UPDATE test SET sim ='0.3961' WHERE keyword ='加法'");
      stmt20.executeUpdate("UPDATE test SET sim ='0.1667' WHERE keyword ='通分'");
      stmt20.executeUpdate("UPDATE test SET sim ='0.2069' WHERE keyword ='分配律'");
    }
  }
  int d = 0;
  int g = 0;
  double similar1;
  double similar2[]= new double[17];
  double similar3[]= new double[17];
  double similar4[]= new double[17];
  double similar5[]= new double[17];
  while(rs21.next()) {
    similar1 = rs21.getDouble("sim");
    //System.out.println(similar1);
    similar2[d] = similar1;
    d++;
  }
  for(int i=0; i<17; i++) {
     if(similar2[i]<=0.15){
       similar3[g] = similar2[i];
       System.out.println("不符合原意,啓動回饋機制");
       g++;
     }
     else if(similar2[i]>0.21) {
       similar4[g] = similar2[i];
       System.out.println("符合原意,啓動擷取網頁機制-" + similar2[i]);
       g++;
       stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='統計'");
       stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='周長'");
       stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='圓周率'");
       stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='循環小數'");
       stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='九九乘法'");
       stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='等值分數'");

  stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='等值分數'");
  stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='公因數'");
  stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='國小四年級數學'");
  stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='牛頓'");
  stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='平行四邊形'");
  stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='先乘除後加減'");
  stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='小二乘法'");
  stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='比例尺'");
  stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='等差數列'");
  stmt22.executeUpdate("UPDATE test SET sim_mark ='符合' WHERE keyword ='加法'");
}
else {
  similar5[g] = similar2[i];
  System.out.println("部分符合原意-" + similar2[i]);
  g++;
  stmt22.executeUpdate("UPDATE test SET sim_mark ='部分符合原意' WHERE keyword ='通分'");
  stmt22.executeUpdate("UPDATE test SET sim_mark ='部分符合原意' WHERE keyword ='分配律'");
  File fr1 = new File("C:\\Documents and Settings\\stanley\\桌面\\test\\general\\通分.pdf");
  File fr2 = new File("C:\\Documents and Settings\\stanley\\桌面\\test\\general\\分配律.pdf");
  if(fr1.exists()) {
    fr1.delete();
  }
  if(fr2.exists()) {
    fr2.delete();
  }
  else {
    //System.out.println("error");
  }
}
```

圖 4-3：相似度程式碼

第三節　實驗結果

表4-1為20篇摘要經由程式擷取處理後，呈現包含關鍵字的所有句子及標題。

表 4-1：標題與摘要句子彙整表

摘要編號－關鍵字	標題	摘要句子
1－通分	高雄市國一學生分數乘除法運算錯誤類型與成因之探究	先通分再相乘與整數相乘加上分數相乘等
2－平行四邊形	淺談行列式及其幾何性質	第一段中從平面上的平行四邊形面積來推導空間中二個向量所張出的平行四邊形面積 藉由方向餘弦的性質來計算空間中的平行四邊形面積
3－牛頓	牛頓的啟示	牛頓是偉大的物理學家和數學家 他在伽利略等人研究的基礎上總結出牛頓三大定律和萬有引力定律 牛頓從小就喜歡思考 應該像牛頓一樣
4－加法	郭伯臣施淑娟	在本文中以分數加法的診斷測驗為例

		學生作答分數加法測驗時
5－分配律	在課堂討論情境下國一學生文字符號概念及對運算相關法則的認知	而且對分配律與結合律的意義和應用不甚了解
6－圓周率	圓周率單元教學之探討	本研究探討四位國小教師的圓周率單元教學 了解教師如何進行圓周率的教學 圓周率單元教學主要的困難是如何將抽象的圓周率概念教給學生 學生要如何去瞭解圓周率的意義 與教師引入圓周率概念的方法 圓周率意義的說明所造成的
7－等值分數	貝氏網路在數學領域數與量主題測驗之應用以國小五年級等值分數單元為例	本文旨在探討以國小五年級等值分數單元為例 以等值分數錯誤類型 學童在等值分數單元子概念的精熟度介於 學生最常以分子和分母差不變的錯誤概念處理等值分數的問題

8－比例尺	中華民國中小學教師自然科學與數學教學設計競賽	比例尺這個單元 藉由計算縮圖和實際長度的比和比值正式進入比例尺的學習 平面圖並將比例尺的觀念真正應用在生活中在遊戲中學數學是本單元設計的最終目的
9－統計	國小統計課程之內函與教學理念	統計的主體為資料 統計的目的則在解決生活上的問題 因此統計的教學應強調資料與問題之間的聯繫以及對統計結果的影響 統計內涵除了包括以數學為基礎之統計基本概念和方法外 更重要的是必須包含統計思維 統計過程的建構 本文將提出統計過程教學概念圖 作為國小教師統計教學的參考 除了從統計過程的思維來

		闡述統計教學的理念外 本文並針對九年一貫統計能力指標中 教師有疑意的統計專有名詞和概念 教師對統計教學的認知影響統計課程的有效運用 本文中將以美國數學科學委員會所提出的教師統計知識作為依據 說明統計過程在統計教學知識中所扮演的角色 最後提出國內目前統計課程和統計教學的盲點與困境
10－公因數	公因數與公倍數解題模組	在國中數學一年級課程中公因數與公倍數的應用問題乃為師生最頭疼的單元之一 老師用心講解公因數與公倍數的基本概念 當公因數與公倍數應用到解決一些數論問題或應用問題時 今特將吾人所發展之公因

		數與公倍數解題模組供與大家參考
11－小二乘法	應變計試驗配合最小二乘法計算複合材料之材料係數	本論文使用最小二乘法來計算複合材料之應變及應力關係
12－先乘除後加減	先乘除後加減	某老師將之簡化為先乘除後加減 學生解釋是因為習慣了背誦先乘除後加減
13－國小四年級數學	整數數感融入國小四年級數學科教學之研究	分解或合成數以便於運算和使用參考值以便於解題融入國小四年級數學科教學有關整數教材的單元中
14－小三數學	無	本文報告一項有關通達學習法對學生的小三數學學習成果的影響
15－三位數	HPM通訊第四卷第十一期	這三位數學家的著作可說是阿拉伯幾何的三支柱
16－幾分之幾	在裡面變多少	形成的內多邊形面積為原多邊形面積的幾分之幾
17－等差數列	協助學生發展數學化能力之探究	教材單元包含了乘法公式和等差數列
18－周長	三角形周長等分線的作圖與數量分佈	過三角形頂點作出周長等分線 發現三條過頂點的周長等

		分線交於一點 畫出過P點的周長等分線 在眾多周長等分線中 我們發現了一條特殊的周長等分線 其他的周長等分線 中點皆落在此特殊周長等分線上 作出過P點的周長等分線 當P點在包絡區內部可繪出三條周長等分線 只能作出一條周長等分線
19－循環小數	探討循環小數的循環節	循環小數的循環節是一個令人玩味的主題 經由探討循環小數的循環節 本文嘗試解釋學生如何利用回答下列有關循環小數的相關問題
20－九九乘法	來玩九九乘法表	一開始我們想解出九九乘法表中 更發現九九乘法表的奧妙與許多有趣的玩法

表4-2為20篇摘要句子與標題分別經由程式運算加總後，所求得資訊含量值。

表 4-2：摘要句子與標題資訊含量值

摘要編號－關鍵字	資訊含量值	等級
1－通分	0.1176	中資訊含量
2－平行四邊形	0.1576	中資訊含量
3－牛頓	1.0555	高資訊含量
4－加法	0.3333	高資訊含量
5－分配律	0.1053	中資訊含量
6－圓周率	0.9892	高資訊含量
7－等值分數	0.6039	高資訊含量
8－比例尺	0.4213	高資訊含量
9－統計	2.2648	高資訊含量
10－公因數	0.4826	高資訊含量
11－小二乘法	0.1574	中資訊含量
12－先乘除後加減	0.4207	高資訊含量
13－國小四年級數學	0.1234	中資訊含量
14－小三數學	0.0741	低資訊含量
15－三位數	0.1	低資訊含量
16－幾分之幾	0.0952	低資訊含量
17－等差數列	0.125	中資訊含量
18－周長	1.9463	高資訊含量
19－循環小數	0.5127	高資訊含量
20－九九乘法	0.4467	高資訊含量

根據第三章所述，系統先將低資訊含量的摘要刪除，程式則進一步處理一連串步驟，包含TP、PW、TW所獲得的原始權重值，以及加

入SW所得新權重值，表4-3為每篇摘要的TP、PW、TW與SW，表4-4則為原始摘要排序、原始權重值、新摘要排序與新權重值之整理表。

表 4-3：TP、PW、TW與SW

摘要編號－關鍵字	TP	PW	TW	SW
1－通分	5	5	5	0
2－平行四邊形	15	10	11	0
3－牛頓	20	11	7	0
4－加法	10	10	8	0
5－分配律	5	1	7	0
6－圓周率	35	16	24	0
7－等值分數	20	15	10	0
8－比例尺	15	11	0	0
9－統計	105	16	0	0
10－公因數	20	16	5	0
11－小二乘法	5	10	13	5
12－先乘除後加減	10	11	8	5
13－國小四年	5	10	17	7

級數學				
17－等差數列	5	5	6	6
18－周長	50	16	39	6
19－循環小數	15	16	16	7
20－九九乘法	10	11	26	7

表 4-4：權重值與摘要排序

摘要編號 (原始摘要排序)	原始權重值	摘要編號 (新摘要排序)	新權重值
9	121	9	121
18	***105***	18	***111***
6	75	6	75
19	***47***	19	***54***
20	***47***	20	***54***
7	45	7	45
10	41	10	41
3	38	13	***39***
2	36	3	38
13	***32***	2	36
12	***29***	12	***34***

4	28	11	***33***
11	***28***	4	28
8	26	8	26
1	25	1	25
17	***16***	17	***22***
5	13	5	13

表4-4中共有7篇相關詞與同義詞摘要，其摘要則以粗斜體表示，排序有變動的摘要則以方框表示，由上表可知第11篇與第13篇摘要，原始權重值分別為28與32，因為加入SW所產生的新權重值分別為33(縮寫同義詞權重為5)與39(單字同義詞權重為7)；接續系統會根據新權重值順序，讓每則摘要與原意逐一進行相似度運算，原意是指使用者於討論區所描述的句子，表4-5為其運算結果。

表 4-5：摘要句子與原意相似度

摘要編號	相似度
9	2.1084
18	1.641
6	1.52
19	1.0201

20	0.6092
7	1.2388
10	0.7286
13	0.25
3	0.7514
2	0.5024
12	0.9661
11	0.2286
4	0.3961
8	0.6768
1	0.1667
17	0.6092
5	0.2069

表4-5中呈現編號1與5摘要系統會予以刪除，因為此兩篇隸屬於部分符合原意，表示參考價值不大，其他篇數都符合原意。

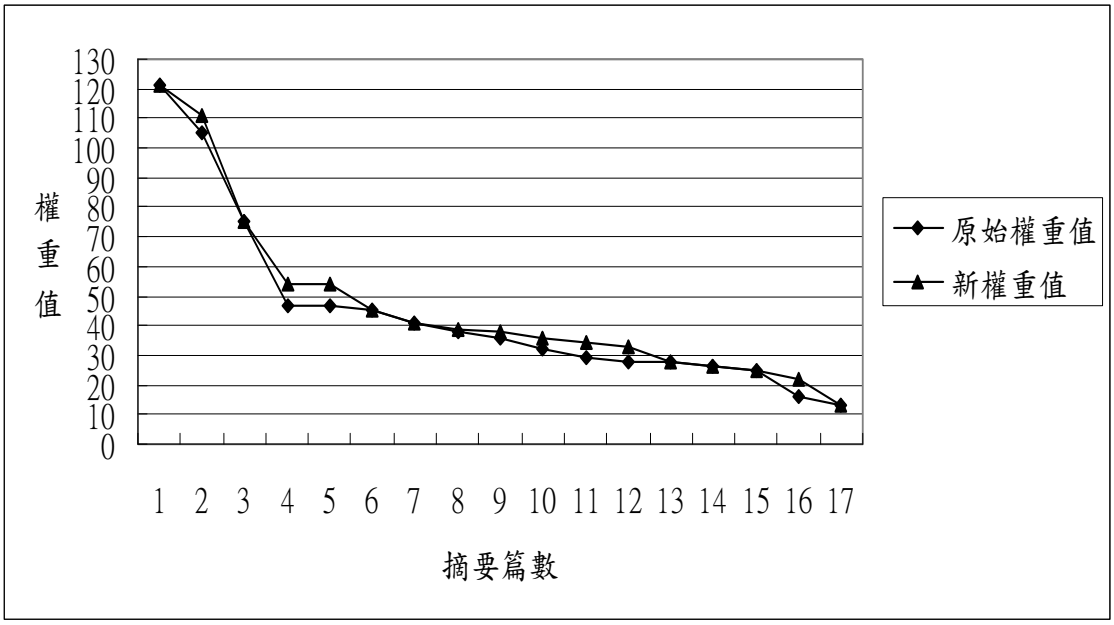

圖 4-4：權重值實驗結果

圖4-4是參照表4-4所得折線圖，縱軸為17篇摘要的各別權重值，橫軸則為17篇摘要篇數，其中已剔除低資訊含量的摘要(第14篇、第15篇和第16篇)；圖中可得知TP、PW、TW與TP、PW、TW及SW之各別運算結果，會產生兩種不同的權重值，搭配表4-5數據表佐證，顯示加入SW確實影響摘要排序，進一步證實導入同義詞權重(SW)有助於一篇描述不同意思卻相同的摘要重要性提升，如此可避免忽略此摘要所潛藏的隱性知識。

透過文獻[8]採用應用率與正確率兩項評估準則，應用率是指全部文章裡，有使用同義詞文章與未使用同義詞文章之多寡；正確率指某類型文章有被真正使用之高低程度。藉由兩項準則衡量經本研究機制處理的摘要是否應用同義詞權重，計算SW使用程度，並且針對含有SW

的中等及高資訊量摘要與未含SW的高資訊含量評估各別正確率，如表4-6所示。

表 4-6：應用率與正確率

經處理摘要篇數	17	經處理摘要篇數	17
實際應用SW摘要篇數	7	未含SW摘要篇數	10
應用率	**0.4118(41.18%)**	**應用率**	**0.5882(58.82%)**
中等資訊量摘要篇數	6	無	無
實際應用SW摘要篇數	3	無	無
正確率	0.5(50%)	正確率	無
高資訊量摘要篇數	11	高資訊量摘要篇數	11
實際應用SW摘要篇數	4	未含SW摘要篇數	7
正確率	0.3636(36.36%)	正確率	0.6363(63.63%)
正確率	**0.8636(86.36%)**	**正確率**	**0.6363(63.63%)**

參照表4-6所示，表中左半部分是依設計機制所執行的結果分析，右半部分是對照組，所以不會處理同義詞只執行高資訊量的摘要；中等資訊量裡實際應用同義詞權重的摘要分別為第11、13和17篇，其正

確率為0.5；高等資訊量中實際應用的篇數分別為第12、18、19和20篇，其正確率為0.3636；所以含有SW之應用率為0.4118，正確率為0.8636(粗黑底線所示)。未含SW的應用率為0.5882，正確率為0.6363(粗黑底線所示)。綜合上述可知道摘要中若涉及同義詞權重(SW)是會小幅影響整體排序，而且含有SW的正確率可達八成六，未含SW達五成八，更有助於系統較迅速提供最佳解，間接節省了使用者的檢索時間與精神，因此效能進而提升。

第五章、結論與未來展望

第一節　結論

本研究發現目前各家搜尋引擎功能都相當齊全且強大，檢索結果差異性不多，重點要能減少頻寬負載量，避免使用者浪費精神與時間自行過濾檢索結果，因此本研究設計網路搜尋引擎的強化機制，透過比對演算法先剔除重複網址，接續將摘要與標題進行前置處理，並以領域數學知識為基準，運算每則摘要的資訊含量。

排序演算法則利用 TP、PW、TW 與 SW 值，計算每則摘要之總體權重值，此處 SW 是以相關詞規則與同義詞規則為準則；適用規則一者，藉由相似度運算取得適當權重值；適用規則二和三者，則是從資料庫中直接給予定義的權重值。導入領域實體論法是為解決敘述不同但意義相同的摘要問題，結果經實作發現確實影響摘要權重排序，進一步會促進資訊迅速被使用者發覺。因此，本研究的產出及貢獻如下：

一、設計建構強化搜尋引擎的過濾與排序機制。

二、提出領域本體論法解決描述不同但意思相通的摘要問題；利用原方法的 TP、PW、TW 值搭配同義詞權重(Synonym Weight)進行運算，經實驗證明能加強部分摘要排序，優先提供較貼切原意之隱含知識，以便給予使用者參考。

三、減少使用者檢索時間讀取重複性太高的網頁且提高正確性與效能。

現今網路上搜尋的摘要，都以關鍵字為主，經研究發現呈現結果都以摘要特定內容段落為主，導致無法確實檢索網頁裡所潛藏的知識，未來可改用網狀結構圖形取代關鍵字為主之搜尋，其效能雖尚待評估，但是對於格式標準化過程能嵌入更有意義的內容與標題，有助於往後知識萃取或知識樹建構。

第二節　未來展望

第三章中曾簡述圖形化語意網的型態，若能將圖形化語意網模組應用於其他相關領域，例如知識地圖、討論區，加上專家驗證或自我學習機制必能挖掘出特徵與規律（pattern），進而組織成知識。

網路上資訊格式大致可分為 htm、html、pdf、xls、ppt、doc、xml、txt、rtf 與 ps，其中 pdf 是 Adobe 公司所制定的一種格式，但內容卻相當複雜，pdf 採用二進位與純文字混合的編碼模式，並不像 xml 或 html 容易處理，而且未採用 Unicode 等標準字元編碼，而是透過內建的編碼表進行字元編碼[10]，使得使用者處理此類資訊時更顯困難，此外 pdf 提供類似安全碼機制，當原創者撰寫完某篇文章，可對文件進行加密，其用意就是不希望使用者任意竄改內容，甚至禁止對於文件進行複製與剪下動作，基於前述理由未來可研究探討設計一套機制或方法以解除加密限制並處理編碼問題。

領域本體論概念目前已廣泛應用於資訊領域，針對摘要中關鍵字歧義詞部分，若關鍵字無法檢索答案，可運用其反義詞或建立關鍵字與歧義詞的規則關係二度嘗試搜尋，以求得最佳解；藉由概念性模組構成的實體能精確定義出不同符號或詞彙意義，以協助解決類似問題；若能有效應用本體論方法於各種領域，乃是其最大優勢。

參　考　文　獻

一、中文部份

[1] 王志浩，「知識發掘之技術於智慧型資訊檢索系統之研究」，元智大學資訊工程研究所碩士論文，92 年 7 月。

[2] 王常威，「以內容為基礎之 XML 文件分類方法之研究」，成功大學資訊管理研究所碩士論文，93 年 6 月。

[3] 陳光華、莊雅蓁，「應用於資訊檢索的中文同義詞之建構」，中國圖書館學會會報，第六十七期，93～107 頁，90 年。

[4] 石逸民，「從全球資訊網擷取同義詞」，中正大學資訊工程研究所碩士論文，92 年 7 月。

[5] 李健興、陳雅絹、郭雅琪及莊宏翊，「基於Ontology之中文文件自動摘要技術之研究」，輔仁管理評論，92 年。

[6] 邱立豐，「互動式概念查詢應用於網路文件自動摘要之效益」，雲林科技大學資訊管理研究所碩士論文，91 年 6 月。

[7] 林欣彥，「權重式超搜尋引擎與網頁偵測之研究」，朝陽科技大學資訊管理研究所碩士論文，92 年 7 月。

[8] 柯淑津，“從詞網出發的中文複名詞的語意表達”，International Journal of Computational Linguistics and Chinese Language Processing, pp. 93-108, 2003.

[9] 陳同孝、謝俊宏及張家力，“智慧型網際代理人系統之建構”，台灣區網際網路研討會TANET，89 年。

[10] 陳鍾誠、廖先志，「OpenPDF-將 PDF 轉為 XML 的開放原始碼軟體」，94 年 10 月。

[11] 陳麴合，「超連結與關鍵字頻分析之搜尋引擎研究」，屏東科技大學資訊管理研究所碩士論文，90 年 6 月。

[12] 許志新，「分散式搜尋引擎之設計與實作」，中正大學資訊工程研究所碩士論文，85 年 6 月。

[13] 黃純敏、吳郁瑩，“網路中文文件自動摘要”，台灣區網際網路研討

會TANET，88 年。

[14]黃純敏、楊存一及邱立豐，「多語文超文件自動摘要與評估」，行政院國家科學委員會專題研究計畫成果報告，計劃編號：NSC89-2416-H-224-053，90 年。

[15]謝文泰、陳文錶、張履平，「以句子資訊量來產生文件摘要之模式」，第七屆人工智慧與應用研討會，661～666 頁，91 年 11 月。

[16]鍾明強，「基於 Ontology 架構之文件分類網路服務研究與建構」，成功大學資訊工程研究所碩士論文，93 年 6 月。

[17]蕭榮賢，「基於詞彙分析之資訊搜尋系統的設計與實作」，中正大學電機工程研究所碩士論文，94 年 7 月。

二、西文部份

[18]R. Cooley, B. Mobasher, and J. Srivastava, "Web mining：information and pattern discovery on the World Wide Web," 9th IEEE International Conference on Tools with Artificial Intelligence, pp. 558-567, 1997.

[19]T. R. Gruber, "A translation approach to portable ontology specifications," Knowledge Acquisition, pp. 199-220, 1993.

[20]J. Han and K. Chang, "Data Mining for Web Intelligence," IEEE Computer, pp. 64-70, 2002.

[21] C. Jenkins, M. Kackson, P. Burden, and J. Wallis, "Searching the world wide web：an evaluation of available tools and methodologies," ELSEVIER Journal on Information and software technology, pp. 985-994, 1998.

[22]A. Maedche, B. Motik, L. Stojanovic, and R. Studer, "Ontologies for Enterprise Knowledge Management," IEEE Intelligent System, pp. 26-33, 2003.

[23]Sunil Kr. Pandey and R.B. Mishra, "Intelligent Web Mining Model to Enhance Knowledge Discovery on the Web," Seventh International Conference on Parallel and Distributed Computing Applications and Technologies, pp. 339-343, 2006.

[24]E. Spertusm, "Mining Structural Information on the Web," The Sixth International World Wide Web Conference, pp. 1205-1215, 1997.

三、網站部份

[25]資訊檢索與知識探勘, http://www.lac.org.tw/20040518/93_teacher_03.doc

[26]搜尋引擎排名, http://www.promote168.com.tw/search-engine-list.htm

[27]國教專業社群網, http://teach.eje.edu.tw/

[28]AltaVista, http://www.altavista.com/

[29]GAIS, http://gais.cs.ccu.edu.tw/

[30]Google, http://www.google.com.tw/

[31]W3C web site, http://www.w3.org/XML

[32]Yahoo, http://tw.yahoo.com/