

國立臺灣大學地理環境資源研究所

碩士論文

Department of Geography

College of Science

National Taiwan University

Master Thesis

語意關聯式的新圖文整合

— 以陽明山國家公園研究報告為例

Applying Semantic Matching Techniques to Integration of
Documentations and GIS --

A Case Study of Research Papers on YMS National Park



Yuan-Fu Luo

指導教授：朱子豪 博士、孫志鴻 博士

Advisor: Tzu- How Chu, Ph.D. Chih-Hong Sun, Ph.D.

中華民國 98 年 7 月

July, 2009



目錄

口試委員審定書	i
誌謝	ii
中文摘要	iii
英文摘要	iv
第一章 緒論	1
第一節 研究動機	1
第二節 研究目的	3
第三節 研究範疇與限制	4
一、研究範疇	4
二、研究限制	4
第二章 文獻回顧	7
第一節 圖文整合與自然語言斷詞	7
一、圖文整合	7
二、自然語言斷詞	9
第二節 語意搜尋 (Semantic Search) 與知識本體 (Ontology)	11
一、語意搜尋定義	11
二、知識本體定義	11
三、知識本體應用	12
四、知識本體種類	13
五、知識本體語言 (Ontology Language)	15
六、國內外過去與本體論相關的研究	19
第三節 語意網 (Semantic Web)	19



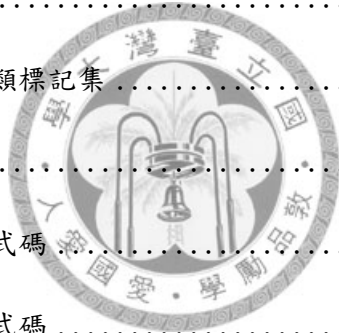
一、語意網.....	19
二、明顯的詮釋資料 (MetaData).....	20
三、邏輯 (Logical).....	21
四、語意網的演進架構.....	21
五、國內外過去與語意網相關的研究.....	23
第三章 研究方法	25
第一節 研究架構	25
第二節 研究流程	26
一、研究背景界定階段.....	27
二、研究範疇界定階段.....	28
三、文獻回顧階段.....	28
四、知識建構階段.....	29
五、知識整合階段.....	29
六、知識驗證階段.....	29
第三節 研究方法	29
一、建立語意網.....	29
二、定義語意網的空間與非空間關係.....	33
三、定義關係的反應行為.....	36
四、串聯語意網.....	44
五、聚焦.....	45
六、語意資料庫的擴充機制.....	48
第四節 研究工具	49
一、語意網建立工具：Protégé 3.4.....	49
二、查詢定位圖台：ArcGIS 9.2.....	50



三、系統整合工具：Java.....	51
四、詞性標註工具：中研院中文斷詞系統.....	51
第四章 研究成果與討論.....	55
第一節 詞彙庫及語意關聯研究成果.....	56
第二節 文件斷詞模組.....	59
一、系統簡介.....	59
二、系統範例.....	61
三、小結.....	62
第三節 語意推論模組.....	63
一、系統簡介.....	63
二、系統範例.....	65
第四節 查詢定位圖台.....	66
一、系統簡介.....	66
二、系統範例.....	69
第五節 查詢情境設計.....	71
情境一：無推論之關鍵字比對.....	71
情境二：透過空間外部語意關聯的推論.....	73
情境三：透過空間內部語意關聯的推論.....	74
情境四：透過屬性外部語意關聯的推論.....	75
情境五：透過屬性內部語意關聯的推論.....	76
第五章 結論與未來發展.....	77
第一節 結論.....	77
一、推論引擎.....	77
二、語意網的建立.....	77



三、詞性標註.....	78
四、簡單關係的定義.....	78
五、有限資源的最大化.....	78
第二節 未來發展.....	78
一、文本限制.....	78
二、自動化文件語意網粹取.....	79
三、引用現有的語意網.....	79
四、語意網來源的選擇.....	79
五、同義字的誤判.....	80
六、單向串連.....	80
參考文獻.....	81
附錄一、中研院平衡語料庫詞類標記集.....	85
附錄二、中文詞類分析總表.....	87
附錄三、文件斷詞模組部份程式碼.....	101
附錄四、語意推論模組部份程式碼.....	121
附錄五、查詢定位圖台部份程式碼.....	133



圖目錄

圖 1-1 語意對應流程.....	2
圖 1-2 彼此無任何關聯.....	2
圖 1-3 透過關鍵字串聯.....	2
圖 1-4 透過語意網串聯.....	3
圖 1-5 語意搜尋差異圖.....	3
圖 1-6 本體論將知識組織成有規律的架構.....	4
圖 2-1 圖文整合流程.....	8
圖 2-2 知識本體應用示意圖.....	12
圖 2-3 知識本體的種類.....	14
圖 2-4 各知識本體再利用性與實用性比較.....	14
圖 2-5 RDF 基本資料模式.....	16
圖 2-6 使用 RDF 圖形描述知識本體.....	17
圖 2-7 使用 RDF 圖形描述知識本體.....	17
圖 2-8 語意網的演進架構 (W3C Semantic Web Layers).....	22
圖 3-1 研究架構圖.....	26
圖 3-2 研究流程圖.....	27
圖 3-3 內部語意關聯.....	30
圖 3-4 外部語意關聯.....	30
圖 3-5 建立語意網的來源屬性表.....	31
圖 3-6 語意網建立結果.....	31

圖 3-7 外部語意關聯建立流程.....	32
圖 3-8 語意網建立流程圖.....	33
圖 3-9 不同尺度間資料的包含關係.....	34
圖 3-10 不同尺度間資料的包含關係.....	35
圖 3-11 空間相鄰關係.....	36
圖 3-12 語意網中關係定義.....	38
圖 3-13 呈現結果.....	38
圖 3-14 語意網中關係定義.....	38
圖 3-15 詞彙種類示意圖.....	39
圖 3-16 關係的反應行為.....	39
圖 3-17 語意網中關係定義.....	39
圖 3-18 呈現結果.....	39
圖 3-19 語意網中關係定義.....	40
圖 3-20 詞彙種類示意圖.....	40
圖 3-21 關係的反應行為.....	40
圖 3-22 語意網中關係定義.....	40
圖 3-23 呈現結果.....	40
圖 3-24 語意網中關係定義.....	41
圖 3-25 詞彙種類示意圖.....	41
圖 3-26 關係的反應行為.....	41
圖 3-27 語意網中關係定義.....	41



圖 3-28 呈現結果.....	42
圖 3-29 語意網中關係定義.....	42
圖 3-30 關係的反應行為.....	42
圖 3-31 語意網中關係定義.....	43
圖 3-32 呈現結果.....	43
圖 3-33 語意網中關係定義.....	43
圖 3-34 詞彙種類示意圖.....	43
圖 3-35 關係的反應行為.....	43
圖 3-36 不同語意網的實例可透過關係串聯.....	45
圖 3-37 本研究歸納之類別串聯關係圖.....	45
圖 3-38a 語意深度為 0 的推論結果.....	46
圖 3-38b 語意深度為 1 的推論結果.....	46
圖 3-38c 語意深度為 2 的推論結果.....	46
圖 3-38d 語意深度為 3 的推論結果.....	46
圖 3-39 語意擴張網路圖.....	47
圖 3-40 單一概念的不同主題關聯.....	47
圖 3-41 語意關係鏈.....	47
圖 3-42 語意資料庫的擴充流程圖.....	48
圖 3-43 Protégé 操作畫面.....	49
圖 3-44 ArcGIS 軟體架構.....	51
圖 4-1 系統運作流程圖.....	55



圖 4-2 空間內部語意關聯.....	58
圖 4-3 屬性內部語意關聯.....	59
圖 4-4 文件斷詞流程.....	60
圖 4-5 文件斷詞模組界面.....	61
圖 4-6 語意推論流程圖.....	63
圖 4-7 語意推論模組界面.....	64
圖 4-8 熔岩與火山地質的關係.....	65
圖 4-9 查詢定位圖台流程.....	67
圖 4-10 查詢定位圖台界面.....	68
圖 4-11 查詢定位圖台界面.....	70
圖 4-12 情境分類圖.....	71
圖 4-13 無推論之查詢流程圖.....	72
圖 4-14 有推論之查詢流程圖.....	73
圖 5-1 語意網的來源選擇.....	80



表目錄

表 2-1 超媒體 GIS 與傳統 GIS 之比較.....	7
表 3-1 語意關聯種類表.....	30
表 3-2 不同尺度的下層概念包含情形.....	34
表 3-3 不同尺度可查詢到的資料.....	35
表 3-4 空間關係行為反應表.....	37
表 3-5 語意網擴充程序表.....	50
表 4-1 空間詞彙庫部份資料.....	56
表 4-2 屬性詞彙庫部份資料.....	57
表 4-3 熔岩推論路徑表.....	63
表 4-4 空間條件與查詢資料的對應情形.....	69





國立臺灣大學（碩）博士學位論文
口試委員會審定書

語意關聯式的新圖文整合
— 以陽明山國家公園研究報告為例

The Semantic Matching Techniques Based On Ontology --
A Case Study of Research Papers of YMS National Park

本論文係羅元輔君（P96228002）在國立臺灣大學地理環境資源學系所完成之碩士學位論文，於民國九十八年五月二十一日承下列考試委員審查通過及口試及格，特此證明

口試委員：



（簽名）

（指導教授）

莊睦雄

賴建貴

孫志鴻

系主任、所長

（簽名）



誌謝

能夠順利完成研究所學業及碩士論文的撰寫，要感謝的人太多了！

首先感謝朱子豪及孫志鴻兩位教授的悉心指導，在論文撰寫的過程中，孫老師給我地理資訊系統最熱門趨勢及資訊，朱老師則在論文寫作的每個階段，給予務實並具體的指導，在追隨著兩位恩師的求學過程中，吸收到許多以往從未接觸的新知，並感謝蔡博文老師在 Seminar 時的指導，及賴進貴老師、莊睦雄老師在口試時的建議及指教，有了他們的栽培，才有今日的成長。

撰寫論文期間受到許多幫助，榮俊德、徐逸祥和張智傑三位學長，對於研究課題的指導，陳致元學長在我失意時的支持與鼓勵，林雅、郁樓、依璇同學在課業上的熱心幫助，我的同學兼上司俊豪同學的支持，及萬組長給我的無比信心，有了他們的幫忙，我的求學之路才能如此順利！

最後，謝謝我的家人默默在背後支持著我，使我在求學之路上無後顧之憂，心中還有對許多人道不盡的感謝，僅以此論文獻給所有關心我的人，謝謝大家！

羅元輔 謹誌於

國立臺灣大學地理環境資源研究所

2009/07/16



摘要

以往 GIS 領域在如何增進使用者介面親和力有許多的切入面向，自然語言即為其中一種，但自然語言缺乏正規的表達方式，常造成關鍵字搜尋與語意式查詢間的語意落差，導致查詢行為受到限制。

本研究嘗試援引「語意網」的技術於地理知識搜尋領域，並分成建立語意網、定義關係、定義反應行為、串聯、聚焦及擴充六階段，透過 Java 為系統撰寫語言，結合中研院詞性標註系統、Protege 及 ArcGIS 等工具，建構一基於語意網的圖文整合平台，期能於閱讀文件時適時提供空間元素對應的 GIS 資料，並經由文件斷詞、語意推論及圖台展示等過程，改善現有空間知識獲取的方式，最後再以五種模擬情境驗證系統，分析語意推論引擎介入前後影響搜尋結果的差異，實驗結果證明推論引擎可有效提升符合語意的搜尋結果滿意度。

本研究之成果，首先為調整現存之語意網建立與應用方法，加以正式化後用於空間知識搜尋領域，及本研究所建立之語意推論模組，可作為發展增加使用者語意彈性的搜尋引擎所使用，最後再將提出的概念落實於實務中，藉以改良現有之圖文整合架構。由新架構獲得的搜尋結果更能滿足人類的語意搜尋求，使閱讀的方式除了心象圖的產生之外，更有實際客觀的資料可加以輔助及驗證。

關鍵詞：空間知識管理、語意網、本體論

**Applying Semantic Matching Techniques to Integration of Documentations and
GIS --**

A Case Study of Research Papers on YMS National Park

Yuan-Fu Luo

The Graduate Institute of Geography
National Taiwan University

2009

Abstract

There have been many approaches to using natural languages to make user interfaces of GIS more friendly. However, the lack of systematic expressions in natural languages frequently causes semantic gaps between users' keyword searches and semantic inquiries and thus inhibits functions of inquiry.

This paper applies "semantic web" technology to search for geospatial knowledge, with the application involving the following six steps: establishment of semantic webs, definition of relationships, definition of the behavior of reactions, connection of multiple semantic nets, focus of the inference, extension the semantic net. This paper uses the Java language to combine tools such as Taiwan Academia Sinica's "Chinese Parsing System", Protégé and ArcGIS to create a semantic framework to integrate documentations and GIS, providing GIS data relevant to the document in use. In addition, Chinese Parsing Module, Semantic Matching Engine, and GIS display are used to improve existing methods of obtaining geospatial knowledge. Finally, this paper uses five scenarios to test the system mentioned above and analyze the effect of the Semantic Matching Engine in improving the performance of search. The results show the Semantic Matching Engine is significantly effective in this respect.

This paper aims to make adjustment in existing establishment and application of semantic webs to specifically make them suitable for use in search for geospatial information. In addition, the Semantic Matching Engine developed in this paper can be used in search engines for increasing users' semantic flexibility. The concepts in this paper are expected to be practically used to improve existing frameworks of GIS information integration to reach better performance in semantic searches.

Keywords: Spatial Information Management, Semantic Web, Ontology





第一章 緒論

第一節 研究動機

在知識經濟的時代，資訊爆炸使人們對資訊的需求激增，人類獲取知識的方法，已從以往需面對面傳授的經驗、紙張紀錄、書本閱讀，快速的演變為不限時間、地點，由網際網路獲取即時的資訊，任何網站、部落格、搜尋引擎儼然成為獲取資訊的途徑之一，而在空間知識獲取方式，近年已大量的採用 GIS，加上 GoogleEarth 所造成的效應與衝擊，令 GIS 需求由以往的學者專家轉變為一般使用者。以往 GIS 領域對於如何增進使用者介面親和力有許多的切入面向，自然語言即為其中一種。所謂的自然語言即是一般人日常生活中彼此溝通所用到的語言，若能使用人類本來就熟悉的語言與電腦進行溝通，其應用層次必能大幅提升，但事實上，此種非正規化語言（自然語言）與正規化語言（程式語言）間彼此是有落差的，而這種「語意落差 (Semantic Gap)」一直是人類與電腦之間溝通的障礙 (Wikipedia, 2009)。

此外，人類的思考模式及語言溝通是具有語意對應的，如「臺灣大學地理環境資源學系」和「台大地理系」指的是相同的概念；但現行的搜尋引擎大多為關鍵字搜尋，針對兩者之間的語意落差無法有效的進行對應，僅少部份頂尖公司如 Google 搜尋引擎，對於類似字詞的搜尋具有關聯的彈性。但對於搜尋引擎的使用者而言，如何用「正確的關鍵字」進行查詢卻是影響知識獲取量及花費時間的關鍵因素，以圖 1-1 為例，假設使用者要找「紗帽山的地質圖」，但現有資料庫中可能沒有相對應的資源，對於有經驗的學者專家來說可輕易的聯想到資料庫中有許多關於「陽明山國家公園」的各種資料，但一般使用者卻可能花上許多的時間將關鍵字從「紗帽山」修正為「陽明山國家公園」，此即為空間尺度上的語意落差，因此如何以最快的時間獲取「正確的關鍵字」即為重點之一。

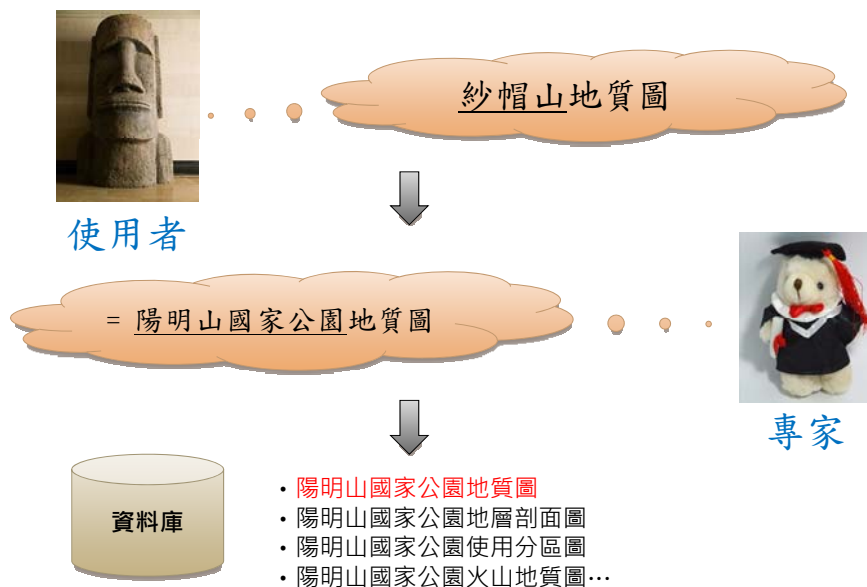


圖 1-1 語意對應流程

此外，本研究亦參考沈嘉慧 (2008) 所提出的「圖文整合」概念進行修改，其演進過程如下方所示。以往的關係如圖 1-2 所示，文件與 GIS 間以人類思考方式來進行連結，也就是說文件必需經由人類解讀後，在大腦中歸納 GIS 可呈現的地點及資料圖層後，採人工操作 GIS 方式來呈現所需要的結果，彼此間並無任何關聯，需由人類來連接彼此的關係；而在沈嘉慧提出了以文字挖掘技術來整合文件與 GIS 的架構後，文件與 GIS 才可透過關鍵字比對的過程，以不需透過人工介入的方式進行連結，有效增加彼此串連的效率 (如圖 1-3)。

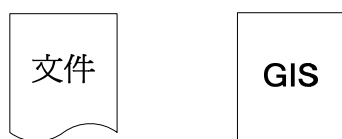


圖 1-2 彼此無任何關聯

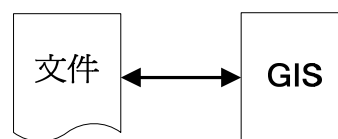


圖 1-3 透過關鍵字串聯

但接著卻遭遇關鍵字比對的限制，當文件的詞彙無法與 GIS 的詞彙對應時，便無法藉由 GIS 呈現文件中所需的資訊，因此本研究提出了圖 1-4 的「語意關聯式的新圖文整合」架構，在原本透過關鍵字串聯的架構中，加入了語意推論引擎，

當文件與 GIS 的詞彙無法透過關鍵字比對進行連結時，便啟動本研究所建立的語意搜尋引擎來進行詞彙的語意推論，藉由計算並提供與原關鍵字相關的其他詞彙後，將可大幅提升文件與 GIS 成功串聯的機會。

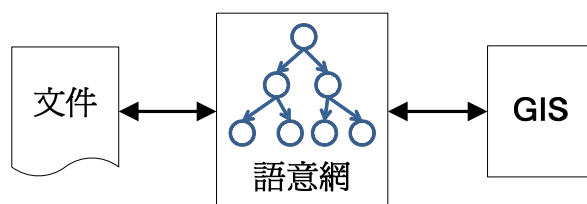


圖 1-4 透過語意網串聯

第二節 研究目的

目前一般 GIS 所提供的為關鍵字搜尋，並未在語意搜尋的機制上多做探討，造成使用者的搜尋往往無法得到令人滿意的結果，以搜尋特定關鍵字而言，當資料庫中沒有相對應的詞彙時，現有的關鍵字搜尋機制會直接結束整個搜尋程序，而本研究探討如何在現有的關鍵字搜尋過程中加入語意推論的概念，當關鍵字搜尋無法得到符合結果時進一步啟動語意搜尋機制，並期望藉由此方式提高使用者對搜尋結果的滿意度。

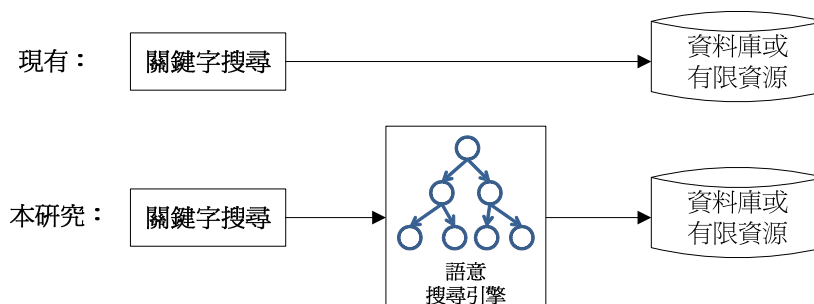


圖 1-5 語意搜尋差異圖

由於關鍵字搜尋與語意對應 (Semantic Match) 多是單向的個別發展，較少針對兩者的整合，因此本研究嘗試引入知識本體 (Ontology) 來解決人類與電腦間的語意落差 (Semantic Gap)，系統整合可以融合兩者的優勢，貢獻則在於提供使用者

在輸入查詢語句後，系統可藉推論引擎回傳符合其語意的搜尋結果，本研究主要的研究目標為建構符合人類語意搜尋式的推論引擎。

第三節 研究範疇與限制

一、研究範疇

從古至今，人類隨著科技進步及智慧的增長，累積的知識不斷的增加，而「語意網」更藉由概念關聯的特性，將原本分散各處的知識建構成如圖 1-6 的規律架構；因此若不根據未來用途加以限制語意網的範圍，將會造成知識體的無限延伸。本研究限制對象的概念主體為「陽明山國家公園」，包括此概念相關的研究報告及 GIS 資料，研究的產出為文件中所隱含的語意網，並將可行部份實際應用於系統發展；就應用層面而言，可發展於使用者搜尋行為的改善，讓搜尋結果更能符合使用者真實的語意需求。

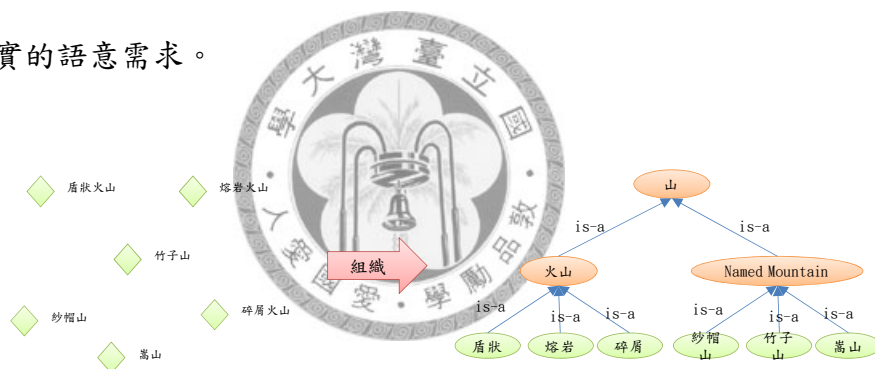


圖 1-6 本體論將知識組織成有規律的架構

本研究為探討關鍵字搜尋機制中的語意擴張部份，目的為建立符合使用者搜尋語意的推論引擎，因此引用沈嘉慧 (2008) 所建立的圖文整合模組，並改善其斷詞模組功能及在關鍵字搜尋過程中加入語意推論機制，因此圖文整合模組的運作過程本研究將不再探討。

二、研究限制

在語意網的部份，由於地理領域知識非常廣泛，全部整理完畢耗費相當時日，因此將範圍限定在「陽明山國家公園」相關的概念之上。在文件的部份，本研究

將只針對王鑫 (1986) 「陽明山國家公園地形及地質景觀」文件作出一個範例。在空間資料的部份，則限定於具有屬性表的向量資料，資料範圍則是陽明山國家公園。

另外在與陽明山國家公園相關的外顯化知識文件中，知識的分類、分享、呈現與儲存，僅佔整個知識管理領域的一小部份，屬知識管理整個導入的部分工作，不能代表整體導入計劃。而研究重心則以如何建立領域語意網及系統應用為優先，後續語意網其他議題則依時間及素材資源考量深入的程度。





第二章 文獻回顧

本研究主要是針對自然語言與 GIS 的空間定位關係中，加入語意搜尋的概念，有許多研究對自然語言與 GIS 做整合，故本研究會先對此部分詳加探討。第二部份則為語意搜尋與知識本體的相關技術。另一方面需關注的是人類搜尋時使用關鍵字時的語意需求，所以第三部份為語意網的相關文獻回顧。

第一節 圖文整合與自然語言斷詞

一、圖文整合

「圖文整合」的研究脈絡乃是由超媒體演化而來過去在結合 GIS 與其他展示系統的研究中，許多曾利用超媒體 (Hypermedia) 與 GIS 整合，例如謝奇峰 (1994) 曾針對玉山國家公園整合超媒體與地理資訊系統，藉由超媒體加強展示多媒的資料的能力和提供更靈活、更符合使用者學習的資料鏈結架構，並利用地理資訊系統來管理、展示與查詢地理資料，表 2-1 說明了超媒體 GIS 與傳統 GIS 間的差別。

表 2-1 超媒體 GIS 與傳統 GIS 之比較

資料來源：謝奇峰 (1994)

	超媒體地理資訊系統	傳統地理資訊系統
展示能力	展示能力較強 (引進多媒體)	展示能力較弱
資訊種類	資訊種類較多 (聲音、視訊)	資訊種類較少
資料鏈結介面	非線性、交互鏈結資料串接 (超鏈結)	傳統層狀資料串連
鏈結選擇介面	選擇性高、彈性大、可因人而異	選擇性低、彈性小 、固定不變
使用者介面	發揮空間較大 (圖形、影像、視訊)	發揮空間較小 (大多為文字)

沈嘉慧 (2008) 提出一般性文件與 GIS 的整合性概念，其圖文整合流程如圖 2-1，首先將文字藉由「中研院斷詞系統」執行斷詞後區分出單詞元素，接著依據其空間特性區分為空間、屬性兩類詞彙，接著透過該研究建立的空間、屬性詞彙庫分別達成定位及屬性條件查詢功能，完成自動提供自然語言語句可能需要的 GIS 輔助資料目的。

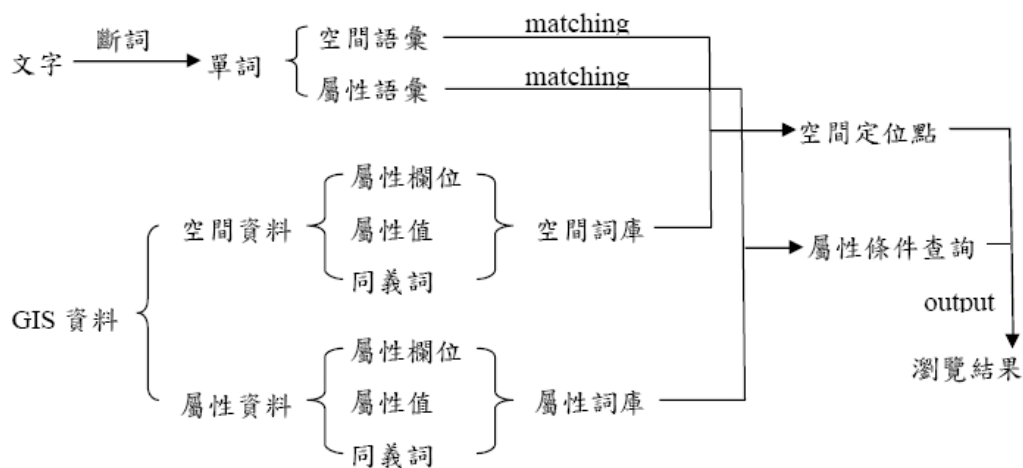


圖 2-1 圖文整合流程
資料來源：沈嘉慧 (2008)

黃如鈺 (2005) 則是將日常生活中的空間查詢語句，整理語句並抽取出其主要之構成元素，歸納出數種主要查詢語句的類型，並利用斷詞理論處理自然語言，並根據整理出來的語句訂定其後續處理之語意語法規則。在系統應用方面，則以餐飲服務查詢為案例，使用者直覺式的輸入自然查詢語句，系統進行斷詞處理、語意處理及空間查詢處理後，直接回覆使用者查詢結果的地圖資訊及相關訊息，研究中共分析出十種查詢語句類型，其類型的判別依據主要在於空間定位點、空間關係詞、查詢對象個數，依數量多寡分類至不同句型，進行不同的對應查詢處

理，此研究中處理的空間查詢來自研究者所調查到之查詢語句類型的整理，所以並未涉及一般文句中的空間關係。

二、自然語言斷詞

所謂自然語言 (natural language) 乃是我們人類所用的語言。任何一種自然語言所用的一個句子可由兩種不同觀點來分析，分別是語法 (syntax) 及語意 (semantics) (孫家麟，1988)。由於外顯化知識文件係以人類習慣的語言所撰寫，因此在文章中充滿各種自然語言，本研究欲對知識文件進行關鍵詞彙的抽離，因此需對於現有的「自然語言斷詞方法」加以探討，而在一般詞句的解析方式，大致分為以下幾種 (蔡純純，2002)：

1、詞庫式斷詞法

詞庫式斷詞是最早運用在文件斷詞的方法，也是目前普遍使用的斷詞方式，比對文件中出現的詞彙與詞庫中的詞彙進行斷詞，做法相當直覺且實作容易。利用事先建立的詞庫，對文件中的詞彙進行比對，完成斷詞動作。

2、統計式斷詞法

統計式斷詞法是參考一大型語料庫 (corpus) 上的統計資訊，單純以鄰近字元同時出現頻率高低作為斷詞的依據。由於語料庫屬於領域相關，不同語料庫間的統計資訊不適合互用。

3、混合式斷詞法

將詞庫斷詞法及統計斷詞法整合。利用詞庫斷出不同組合的詞彙，然後利用語料庫的統計資訊，找出最佳的斷詞組合，降低斷出無意義的詞彙機率，其缺點是需要詞庫的維護與大型的語料庫提供統計資訊；由於統計式斷詞法及混合式斷

詞法需利用大型語料庫來累計統計所需的資訊，本研究在時間及成本的考量上，在文章斷詞的部份決定採用詞庫式斷詞法。至於關鍵詞的擷取方面則有以下方式：

(1) 詞庫比對法 (Dictionary Approach)

利用事先訂定的詞彙字典來對文章進行關鍵詞擷取，其好處是快速及容易實行，尤其是使用的詞彙字典大小能夠包含所要擷取的資訊；但是這也是使用字典方式最大的缺點，如果要能夠有效率的擷取出所要的資訊，詞彙字典的大小及詞彙的相關性將是一大挑戰。

(2) 文法剖析法 (Linguistic Approach)

透過詞庫比對的方式，剖析出文件中的名詞、動詞、代名詞、介系詞等片語，再運用一些方法與準則擷取有意義的片語，過濾掉不適合的詞彙。大部份的剖析程式，需要藉助已經建立的詞典或語料庫，因此其缺點也和詞庫比對法一樣。

(3) 統計分析法 (Statistical Approach)

統計方式利用文字的數字資訊來擷取文章資訊，例如詞彙所出現的頻率、詞彙與詞彙出現的關係程度等，統計方式的好處是不用事先建立詞彙字典及文法也可以擷取出重要的專業領域用語或專用名詞，另一優點是，由於沒有用到詞典、語料庫或自然語言處理的技巧，因此具備擷取速度快、中英文均適用、擷取的詞彙沒有長度限制、可同時擷取廣義詞與狹義詞等特性。

本研究在關鍵詞的擷取則同時採用文法剖析法與詞庫比對法，首先標註文章中每個詞彙的各別詞性，接著將名詞詞彙抽離後與研究所建立的詞彙庫進行比對，綜合採用兩種方法的優點，完成研究提出的文件斷詞模組。

第二節 語意搜尋 (Semantic Search) 與知識本體 (Ontology)

一、語意搜尋定義

語意搜尋 (Semantic Search) 是使用語意網路 (Semantic networks) 實作模糊式查詢及網頁文字搜尋來產生較符合使用者語意的搜尋，Hilderbrand (2007) 等人分析、比較現有的語意搜尋系統，而 Guha (2003) 等人則將搜尋的方式區分為導覽式搜尋 (Navigational Search) 與研究式搜尋 (Research Search) 兩種。在「導覽式搜尋」中，使用者將搜尋引擎作為導覽工具搜尋特定且可預期的文件，此類搜尋方法無法達成語意搜尋的目標，而「研究式搜尋」則由使用者提供代表特定物件的慣用語來進行資訊的蒐集，此類搜尋方式得到的結果並不會完全與關鍵字具有絕對的對應關係，也因此適合作為語意搜尋的發展基礎。

Google 的 PageRank 使用的是偏好排列演算法，它是將無意義的關鍵字與搜尋結果進行排列，而語意搜尋使用的是語意網，一般而言，傳遞使用者所欲查詢的資訊會比單純排列無意義的搜尋結果來得好。另外有一部份人認為「語意搜尋」這個詞是一種將知識由結構較佳的知識體抽離的技術 (例如：Ontology)，使用這種技術可以連接較高層級的正式化知識與使用者的一般性查詢 (Wikipedia, 2009)。

二、知識本體定義

Gurber (2008) 認為 Ontology 一詞最早由哲學領域所提出，它代表的是使用許多屬性來對某件物進行特徵的描述，例如 Aristotle 的 Ontology 使用財富、地位等屬性來描述亞里斯多德的概念性特徵；但在電腦及資訊科技領域中的 Ontology 卻有著不同的意義，它被修改成為了某種目的而設計出來的方法論，用來將特定領域的知識加以結構化的整理、分類，提供後續系統應用。

知識本體是對於某一特定領域中，許多知識術語所組成的集合，包括語彙和語意上的相互連結及在推論和邏輯上的簡單規則。知識本體是用於描述特定領域

知識的一組概念或術語，可用以組織知識庫較高層次的抽象知識，也可用來描述特定領域的知識。知識本體是一個特定知識領域中有關物件的種類 (sorts of objects)、屬性 (properties of objects) 及物件間關係的內容原理 (content theories)。知識本體是對概念化的認同知識所作的正規與明確規範，所謂概念化是指世界上一些現象的抽象模型，正規是指機器可閱讀的，而明確的規範指概念、屬性關係、屬性、限制條件與原理是明確的。

三、知識本體應用

知識本體的應用如圖 2-2 所示，可應用在人與組織間的相互溝通、系統與系統間的交互操作及系統工程中，規則的討論、元件重複使用及使系統更具可靠性。

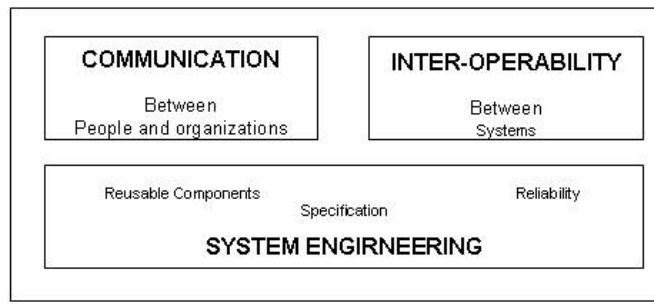


圖 2-2 知識本體應用示意圖

資料來源：Ushold and Gruniger (1996)

(一) 溝通 (Communication)：

基於來自不同背景的人，有不同的需求與觀點，彼此間可能使用不相同、重疊或是不協調的概念，藉由一致化的架構 (Unifying Framework)，可改善溝通的效果。

(二) 交互操作 (Inter-Operability)：

在不同應用系統間，由於有不同建模的方法 (Modeling Method)、語言及軟體工具，藉由知識本體可達成資訊的轉換與整合，達成彼此間的交互操作。

(三) 系統工程 (System Engineering) :

1. 重覆使用 (Re-Usability) : 共同的認知使用正規的呈現方式 (Formal Representation) 與語言, 用以表達對於某特定領域中重要的實體、屬性、流程與交互關係, 這樣的方式可視為一個重複使用的組成要素 (Re-usable Component), 在應用軟體系統間彼此分享。

2. 可信賴性 (Reliability) : 正規的呈現方式亦可用於自動化的一致性檢查 (Consistent Checking), 達成一個更可靠的應用軟體系統。

3. 系統規格 (Specification) : 共同的認知有助於在建立資訊系統時, 確認需求定義規格的整個流程。當系統需求的訂定涉及不同團隊時, 這些團隊來自相同領域或多個不同領域, 用不同的術語與概念時, 知識本體所扮演的角色尤其特別重要。

四、知識本體種類

而知識本體的種類以內容可分為圖 2-3 左方的領域知識本體 (Domain Ontology)、任務知識本體 (Task Ontology) 和通用性知識本體 (General/Common Ontology)。若以概念化議題區分則有如圖 2-3 右方的知識呈現形式概念化的呈現式知識本體 (Representation Ontology)、在不同領域間再利用的專屬知識本體 (Generic Ontology)、再利用的領域知識本體 (Domain Ontology) 與實用性高但較不可再利用的應用知識本體 (Application Ontology)。



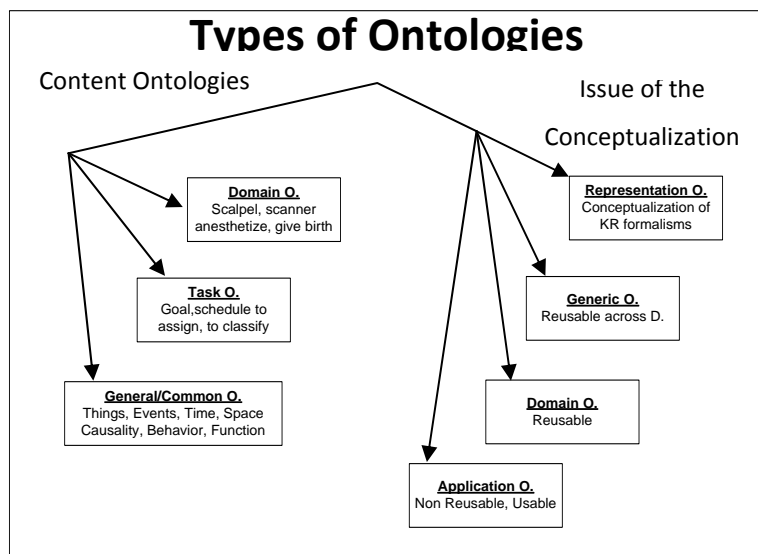


圖 2-3 知識本體的種類

資料來源：Mizoguchi *et al* (1995) 與 Van Heist *et al* (1997)

以再利用程度由低而高排列，特定領域應用的知識本體其再利用程度最低（如描述心臟疾病），其次依序為領域知識本體（如描述身體）、專屬的知識本體（如描述元件）、通用的知識本體（描述人事時地物）及呈現式知識本體（如以框架呈現）。以實用程度由低而高排列，實用性最低者為呈現式知識本體、通用性知識本體、專屬任務知識本體（如工作計劃）、領域任務知識本體（如手術計劃）及實用性最高的特定領域應用知識本體（如心臟手術計劃），如圖 2-4 所示。

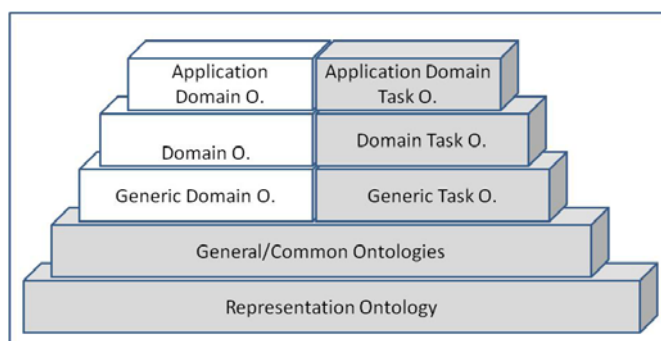


圖 2-4 各知識本體再利用性與實用性比較

資料來源：Gomez-Perez (1999)

五、知識本體語言 (Ontology Language)

(一) 可延伸標記語言 (eXtensible Markup Language, XML)

XML 是一種詮釋語言 (Meta-language)，其主要目的是用來描述文件內容，設計者可以使用自訂的標籤來定義文件資料所代表的意義，XML 具備以下優點：

1. 可延伸性：和 HTML 不同，XML 提供強大的可延伸性。其標籤是由使用者自行訂定的，如此使用者便可根據其欲描述的知識本體來訂定相關的詞彙；透過自訂的 XML 標籤，使用者可以了解標籤間所涵蓋的資訊內容。

2. 資料和展示是分開的：XML 文件可以視為獨立的半結構化資料，如此便可增加資料交換時的便利性。應用程式只要了解此 XML 的 DTD (Document Type Definition) 或綱要 (Schema)，便可輕易的存取和交換資料。若要將 XML 以不同的風格展示在瀏覽器上時，可使用樣式表 (eXtensible Stylesheet Language, XSL) 或 CSS (Cascading Style Sheet) 來做不同的呈現。

3. 可使用 XML 綱要來定義標籤屬性和標籤間的關係：藉由文件型別定義 (DTD) 和 XML 綱要來定義 XML 文件的規則和架構，如此便可描述知識本體中概念本身的屬性首概念間的關係。雖然短期內 DTD 仍有一定優勢，然而就長期的趨勢而言，XML 綱要依然具有 DTD 所沒有具備的特色：

- (1) 具有比 DTD 更豐富的資料型態和描述語法。
- (2) 支援命名空間 (NameSpace) 的使用。
- (3) XML 綱要使用 XML 的基本語法，也因此 XML 所具有的優點，XML 綱要都一併承襲。

(二) 資源描述架構 (Resource Description Framework, RDF) 與資源描述架構綱要 (RDF Schema, RDFS)

RDF 是 W3C 在 1999 年 2 月 22 日所制定的規格 (Recommendation)，其目的主要是為詮釋資料 (Metadata) 在網路上的各種應用提供一個基礎結構 (infrastructure)，使應用程式之間能夠在網路上交換詮釋資料，以促進網路資源的自動化處理。RDF 本身並沒有定義標籤的基本元素，而是提供一個標準模式 (Model) 和相關語法來描述標籤間的關係，其所使用的元素則透過 RDFS 來定義 (World Wide Web Consortium, 2005)。

RDF 的基本模式如圖 2-5 是由 Subject、Property (或 Predicate) 和 Object 所組成，其中 Subject 為所要描述的概念；而 Property 則為用來描述概念的特徵、屬性或關係；Object 則是描述屬性值，屬性值可以是文字或其他資源。

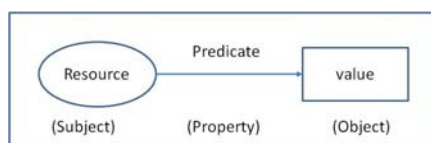


圖 2-5 RDF 基本資料模式

資料來源：Harmelen (2006)

RDFS 定義的基本元素有 Class、Property、Property 的 domain 和 Property 的 range。可將 Class 視為某知識領域中的一種概念，而 Property 即是使用來描述此概念的屬性，至於 Property 的 domain 則是用來表示此屬性是用來描述何種概念，而 Property 的 range 是用來說明此 Property 可能的屬性值範圍。RDFS 文件中所定義的概念、屬性或其他相關的定義，都是在 RDF 文件中會使用到的，就如同 XML 綱要和 XML 文件間的關係。以下為使用 RDFS 圖形表示「紗帽山溫泉」是「地標」的子概念 (Sub-Class)，其下分為「古蹟」、「國家公園」和「溫泉」等三個概念。據此，使用 RDFS 圖形將其表示成圖 2-6 和圖 2-7 的架構，透過此圖形，便可清楚看見景點類別間階層性的架構和用來描述概念的屬性。圖形中「名稱」與「位置」的資料型態為屬性中的 domain 而非元素，「座標」則為屬性中的 range，如此的描述使得標籤間主從關係更加詳細。

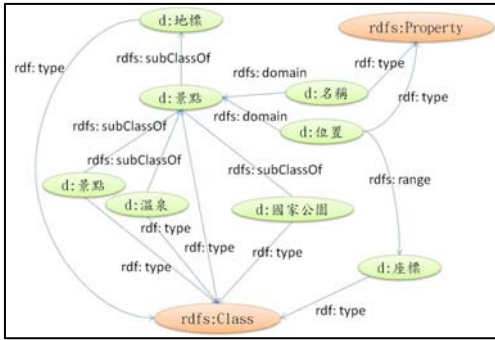


圖 2-6 使用 RDF 圖形描述知識本體

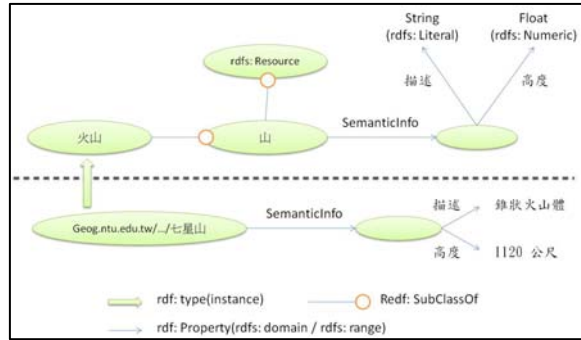


圖 2-7 使用 RDF 圖形描述知識本體

資料來源：Harmelen (2006)

(三) 知識交換格式 (Knowledge Interchange Format, KIF)

此語言的產生主要在於提供給不同應用系統間，利用一個共同語言達到資訊交換的目的。希望透過這個語言或其他共同語言以支援知識的交換，如此一來所有的應用系統便可分享與使用其他應用系統的知識。

知識交換格式提供表達一般第一階述語邏輯 (First Order Predicate Logic, FOPL) 陳述式 (Statement) 的機制，用以呈現有關知識的知識 (Knowledge about Knowledge) 及明確說明推論的規則。

(四) 框架邏輯語言 (Frame-Logic Language, F-Logic)

框架邏輯 (Kifer *et al*, 1995) 為一個物件導向 (Object-Oriented) 與應用邏輯 (Logic-Based) 的語言。一般而言，我們對於知識本體之定義是透過概念或類別 (Classes)、從屬 (Is-a) 關係、屬性、其他關連性與原理 (Axiom) 等概念，因此對於定義知識本體必需有一個適當之語言，以滿足上述之概念。F-Logic 主要包含三個部份：從屬階層 (Class Hierarchy)、屬性定義 (Attribute Definitions) 與相關規則 (Rules)。

1. 在從屬階層中，框架邏輯語言使用 X::Y 表示 X 為 Y 的子概念。

2. 屬性定義或繼承特性 (Inherited Properties) 則定義在知識本體用以表示概念間的關連性，這些關連性透過適當型別之屬性呈現，其屬性可為單一值，如字串或數字，亦可為其他概念間之關係。

3. 包含一些規則與公理，這些規則基於給定之事實，可用於推知新的知識，利用這些規則我們可以幫助應用系統找出不完整性之知識。

(五) 知識本體推論層級 (Ontology Inference Layer, OIL)

是一個應用網頁方式呈現以及對於知識本體的推論層級，它應用框架語言 (Frame-based Language)、描述邏輯 (Description Logic) 與網頁標準 (Web Standard)。OIL 定義了一些類別 (Classes) 並利用類別階層 (Class Hierarchy) 將其組織起來。OIL 建構於 RDFS 之上，使用 RDF 作為它的語法。

(六) DAML + OIL (DARPA Agent Markup Language + Ontology Inference Layer)

由美國國防部先進研究計劃機構 (Defense Advanced Research Projects Agency, DARPA) 所提出的一種較 XML 與 RDF 更新興的實作語意網概念語言。藉由 DAML 將可清楚描述網頁中的每一項內容，包括、圖片、網址等。並透過 RDF model 的概念建立資源間的關係。一般使用者檢視其語法建構成知本體時較容易理解。DAML+OIL 語法本身具有類似框架的架構 (frame-like) 和物件導向的概念，可以 XML 的基本語法來表現，也可與 RDF 利用專門的標籤對應表格而結合使用，還可以 RDFS 定義知識本體中相關的基本元素。

(七) OWL (Web Ontology Language)

前面所描述的 RDF、RDFS 表達力有限，RDF 是粗略的限定於二元基礎表達方式之上，而 RDFS 則是粗略的限定於次類別和性質間的關係，及這些性質的定義域 (Domain) 和值域 (Range)。隨著使用經驗的增加，W3C Web Ontology Working Group 歸納出多個無法以 RDF 及 RDFS 表達的語意網案例，也因此 W3C

Web Ontology Working Group 基於將資訊交換標準化、普及化的目的下，以 DAML+OIL 為基礎定義了 OWL (World Wide Web Consortium, 2005)。

六、國內外過去與本體論相關的研究

McCarthy *et al* (2008) 以本體論來詮釋災害監測系統所回傳的即時監測資料，使得原本看起來毫無意義的數據被賦予新的價值，並結合多種決策支援系統常用的工具如 CLIPS、ArcAgents 等來實作所提出的 REASON 架構，對監測領域及異質性資料庫間的互通性提出良好的架構。

李文元 (1996) 以 Protege-OWL 實作所需的本體論，並用其解決數位化學習過程中，老師與學生都會面臨的課程進度問題，透過研究過程建立的「能力本體論」及「課程本體論」來調整出適合學習者目前能力的教材，該文還模擬具備不同能力使用者於不同資訊載具（桌上型電腦、筆記型電腦及 PDA）的教材重構能力，達成學習者無縫式學習、無所不在學習的目標。

許孟淵 (1996) 主要運用自動建構之新聞本體論 (Ontology) 及先前研究所使用的主題地圖 (Topic Maps) 理論，利用新聞本體論之中概念的關聯及定義，針對個別新聞事件產生所屬的主題索引地圖，提供讀者一種創新且更符合新聞原文語意的新聞關聯知識瀏覽介面，讀者藉由基於新聞本體論之主題地圖幫助，可快速且正確的理解事件內包含到的主題及關聯。

第三節 語意網 (Semantic Web)

一、語意網

自從 1980 年發明了全球資訊網 (WWW, The World Wide Web) 後，以革命性的方式改變了人們之間通訊及交易的方式，但隨之而來的問題在於全球資訊網中的資訊內容並不是電腦可理解的，因此 berners-Lee 於 2001 年提出了語意網 (The Semantic Web) 的概念，並由 W3C (The World Wide Web Consortium) 的語意網推

動小組主導其發展與規範。全球資訊網改變了人們之間通訊及進行交易行為的方式，而趨使現有全球資訊網下一代的語意網演化的原因，乃因隨著科技進步，知識的累積已非常迅速，知識管理的方法論也變得非常重要；而全球資訊網的障礙在於其文章內容的語意不是機器可理解的，雖然電腦可快速的紀錄文字資料、並有效對其進行分割、檢查拼字、計算字數等處理；但卻無法解釋語句並抽出文章所隱含的語意，全球資訊網僅停留在「關鍵字搜尋」這個階段之上。

語意網不是要完全推翻現有全球資訊網的全新架構，而是由全球資訊網逐步演化而成，語意網的主要推動單位是 W3C，創始人為 1980 年代發明全球資訊網的 Berners-Lee (2001)，他期望這個新技術能夠實現他原本對資訊網的願景，這句話所隱含的意義，是指目前的全球資訊網其實並未達到 Berners-Lee 當初的期望，當中正是因為「語意」這個難以克服的問題。

在搜尋領域中，長久存在的問題是：人類所傳達的高階語意與電腦所實際執行的特徵萃取是不同的，而兩者之間的差距稱為語意落差 (莊家銘，2005)。舉例來說，當人們想要找與「狗」有關的所有影片，實際上電腦必需針對該圖片的詮釋資料進行關鍵字搜尋才可找到，但畢竟詮釋資料與圖片本身的內容仍有差距，這種比對的誤差便是語意落差。

二、明顯的詮釋資料 (MetaData)

所謂「詮釋資料」指的是描述資料的資料。目前全球資訊網的內容是適合人類讀取的格式而不是適合程式讀取的格式，對人們而言，這個資訊的呈現方式是令人滿意的，但卻是機器難以理解的，用來解決此類當今資訊網最常面臨難題的方法大致可歸納成兩類：

(一) 發展出超級智能代理者：

著重於發展出可理解當今資訊網使用率最高的 HTML 語法的代理者，這是屬於人工智慧的研究範疇，但實際上人工智慧整體發展的速度並不如人們所預期。人工智慧有長期歷史，但在商業上始終無法成功，難免讓人擔心語意網會重蹈人工智慧的錯誤。

(二) 針對的儲存格式作強化：

由 Tim Berner-Lee 所提倡的語意網願景並不依賴人類程度的智能，事實上它採用的是有別於以往人工智慧的發展技術。以往的人工智慧著重於代理人的功能提升 (或可以泛指搜尋引擎)，而語意網則強調要由資料的本身來克服這個問題。

XML 這種表達方式對於電腦來說，遠遠超過了現有 HTML 所能被電腦處理的價值，術語「詮釋資料」指的就是這種「描述資料的資料」，詮釋資料抓住資料所代表的意義，也就是資訊網中所指的語意。

三、邏輯 (Logical)

邏輯是推理領域的研究範疇，具體的案例為「自動推論者」，這類程式可以從給定的知識演繹出結論，並且提供一連串的推理過程，邏輯可以將內隱知識變成外顯知識，這種推理者在人工智慧中受到廣泛的研究，具體的例子可以學校來說：我們知道所有的教授是教員的份子，所有的教員是職員的一份子，而職員屬於臺灣大學，那麼在這兩層關係底下的教授也就是隸屬於臺灣大學的一份子。這種「若條件，則結論」式的邏輯判斷關係，就是自動推論者能夠依據所提供的條件來得出結論，而當使用者需要時還可提供一系列的推論步驟，甚至在最後提供驗證功能的主要原因。

四、語意網的演進架構

Koivunen 與 Miller 提出了圖 2-8 的語意網演進架構，它描述語意網中所需的相關技術建構層級 (引用自 Harmelen, 2006; 屠名正, 2006)。語意網的演進並非以

新一代語言來推翻舊有語言，而是在原有的發展架構下一步步向上建構，圖中底層代表最基礎的建構標準，頂層代表較難達成的抽象概念；由以往的經驗來看，不同團隊、機構間較容易將底層的協定作為標準來遵循，愈上層的觀念則在不同的研究團隊間，愈因理念、想法差異而不容易為大眾所遵循。

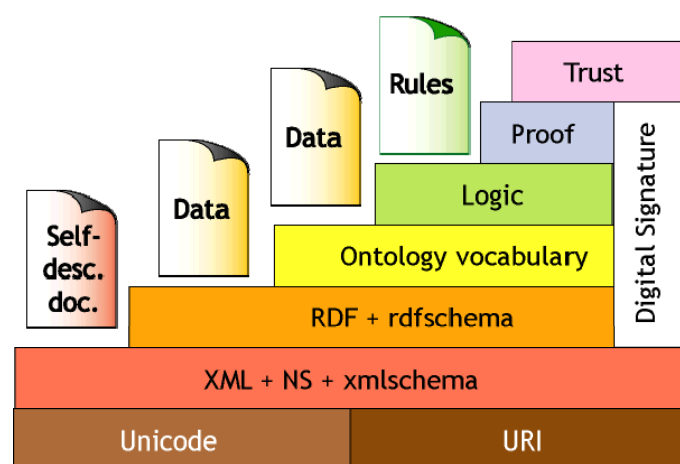


圖 2-8 語意網的演進架構 (W3C Semantic Web Layers)

資料來源：Harmelen (2006)；屠名正 (2006)

在接近底層的 XML 語言，是簡單而強大的資料交換架構；RDF 則可稱為基本的資料模型 (data model)，可用來描述如實體關係模型 (entity-relationship model)，它是基於 XML 發展的語法，因此在上圖中，它位於 XML 層的上方；RDFS 提供塑模規則制定，可將資訊網物件組織成階層，其關鍵要素為類別、性質、關係、領域和範圍限制，RDFS 可看作知識本體的原始語言，但用它來表達複雜的物件關係卻是不足的，因此才有如 OWL 來彌補其不足之處；Logic 層是以特定的描述性知識來應用知識本體的語言；Proof 層包含從較低層級的資訊網語言來推論及證明的語言；最後，再由所謂的「代理者」應用程式來達到使用的目的，時而有 Trust 層中的數位簽章需求，或可說將其定義為「將資訊網上分散和混亂的資訊以特定的方式組織起來的方式」，在金字塔的頂端，信任是高層次和重要的概念，只有在使用者信任其資料的安全和正確前提下，資訊網才能夠實現它全部的潛能。

五、國內外過去與語意網相關的研究

目前有許多探討如何應用語意網技術於空間測量資料整合的研究，如 Schevers 和 Drogemuller (2006) 提出基於語意網技術所開發的平台，用以整合「城市開發」領域中的多種不同分析工具，使彼此間的資料達成共通性及一致性，基於此平台設計的程式將可透過各種不同的來源資料進行推論。

王詠倩 (1997) 則是結合超媒體文件及 GIS，並利用語意網概念將資訊物件化且可由超媒體文件連結的 GIS 資料庫，而語意網可連結這些物件與概念，使資訊的瀏覽更具彈性與效率。在語意網的結構中，基本單元為「物件」，各物件必需明確定義其屬性、行為、關係，並於其中的定義進行與超媒體與 GIS 進行關連。






第三章 研究方法

第一節 研究架構

本研究主要是希望整合文件與 GIS 空間資料，並於圖文整合串連機制中，加入語意搜尋的概念，主要援引沈嘉慧 (2008) 提出的圖文整合架構，並改良其斷詞系統及展示模組，與以往研究不同之處，在於使用語意搜尋引擎來進行關鍵字的擴充；在現有的搜尋方式中，關鍵字需與資料完全相同才可相互對應，當資料庫中無相同之關鍵詞即停止搜尋行為，對語意上完成相同或相似的概念未提供對應功能，本研究則嘗試藉由語意網來找出與關鍵字 (概念) 間的各種關係，並以這些關係串聯至其他新概念，不斷重覆此種推論行為來達成語意關聯目的，提供搜尋的彈性。



在圖 3-1 的研究架構中，以橫向來看主要可分成文件斷詞、語意推論與查詢定位圖台三個模組，將文件輸入斷詞模組後引發需求，斷詞模組負責抽離空間詞彙及屬性詞彙，此時分別於空間詞彙庫及屬性詞彙庫尋找適合的單詞，若有合適的結果則判定為空間定位點或屬性條件查詢，代表可以關鍵詞找到對應的資源，直接進入到查詢定位圖台中呈現瀏覽結果，但若無法於空間、屬性詞庫中找到合適的詞彙，便啟動本研究所發展之推論引擎，藉由尋找空間或屬性語意網的方式來推論與輸入詞彙相關的其他詞彙，接著檢查所有詞彙中是否有相對應資源，在得到額外且具備資源的空間詞彙或屬性詞彙後，便輸出至查詢定位圖台中，並判斷其為空間詞彙時執行定位，屬性詞彙則用來作為圖層載入的依據。

而在垂直的部份則為語意網的建立流程，首先需將陽明山國家公園的先備資料依空間、屬性特性進行區分，接著歸納出空間詞彙群及屬性詞彙群後，將其定義為語意網的概念來源，再依不同概念的特性定義其屬性及其限制條件，接著建立

各個類別的實例 (Instance) 及不同實例間的關係，最後形成空間及屬性語意網，提供給語意推論引擎使用。

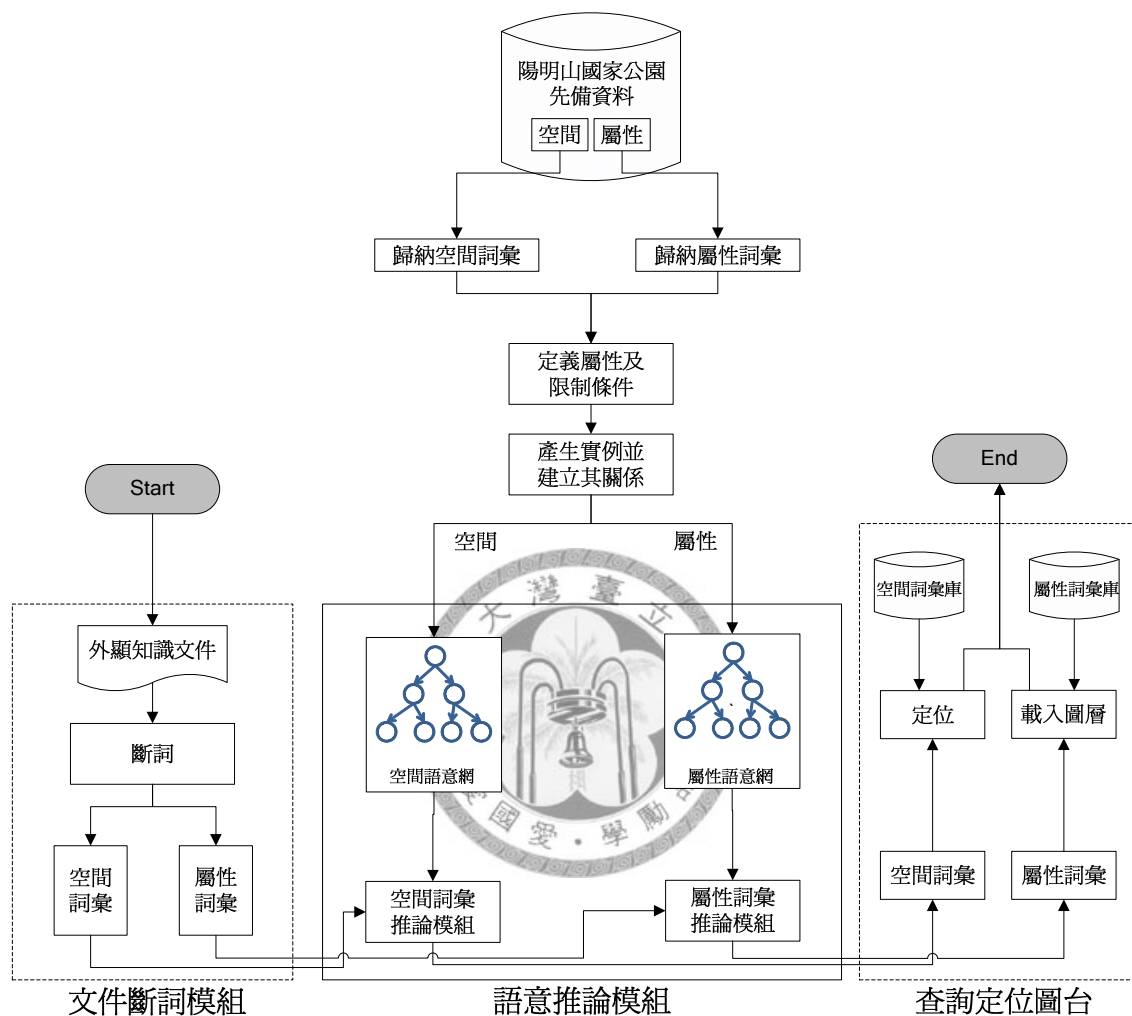


圖 3-1 研究架構圖

第二節 研究流程

本研究之流程可分為如圖 3-2 所示的六階段，詳述如下：

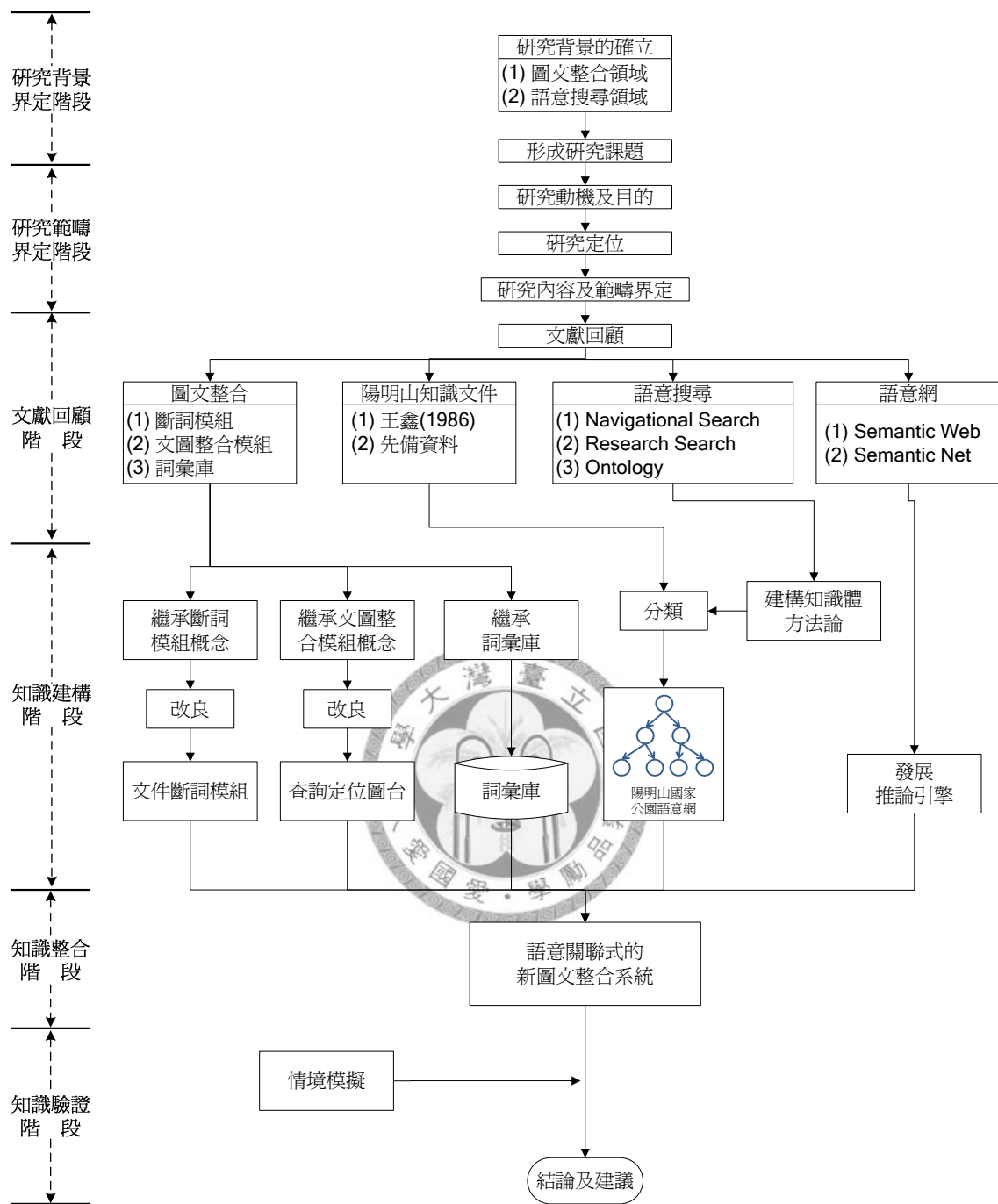


圖 3-2 研究流程圖

一、研究背景界定階段

本階段可分為研究背景確立與形成研究課題兩部份。

在研究背景確立部份，針對陽明山知識文件及圖文整合進行初步文獻回顧，以了解目前圖文整合發展的現況及限制，並回顧語意搜尋相關技術，瞭解其引入圖文整合領域的可行性。

經過初步文獻回顧及研究背景的瞭解後，本研究發現在圖文整合過程中無法有效的進行關鍵字對應是一個相當大的問題，而研究課題於焉形成。

二、研究範疇界定階段

本階段主要在釐清研究動機與目的、研究定位、研究內容與範圍等三部份（詳見第一章）。

三、文獻回顧階段

本研究之文獻主要可分為圖文整合、陽明山知識文件、語意網和語意搜尋四部份。圖文整合部份的文獻回顧主要在瞭解目前圖文整合的發展狀況，並深入瞭解其觀念、技術後改善其架構，將前人提出的斷詞模組、文圖整合模組及詞彙庫深入研究後加以改良，作為後續研究的概念來源。

陽明山知識文件方面的文獻回顧，則是分析現有研究報告及先備 GIS 資料後加以分類，嘗試將正式化的外顯知識進行分類，作為語意網的資訊來源。

而在語意網方面的文獻回顧，主要在探討目前語意網發展的趨勢，以作為研究後續展發的佐證資料，依搜尋的種類又可細分為 Semantic Web 和 Semantic Net 兩類，作為建構推論引擎的參考資料。

語意搜尋方面的文獻回顧，主要在瞭解語意搜尋相關的技術及原理，並帶出知識本體 (Ontology) 此一概念，藉以回顧知識本體相關的定義，作為組織語意網所需的方法論。

四、知識建構階段

本研究將前人圖文整合系統加以改良後，形成文件斷詞模組、查詢定位圖台及詞彙庫三部份，並將語意網所回顧並採用的方法來建構知識體，知識來源則為與陽明山相關的知識文件，加以分類、儲存後形成陽明山國家公園語意網，並由語意搜尋脈絡發展出推論引擎，作為後續階段的素材。

五、知識整合階段

在知識整合階段，主要以 Java 為系統撰寫語言，將上一階段之文件斷詞模組、查詢定位圖台、詞彙庫加以改良後形成圖文整合基本架構，再藉由本研究建置之語意網、推論引擎形成語意關聯式的搜尋架構，形成「語意關聯式的新圖文整合系統」，以增加使用者閱讀文件時的可用性及方便性。

六、知識驗證階段

在本階段中，主要的核心為驗證系統的正確性與可用性，對於系統介面的使用並不是重點；因此不考慮使用者的主觀認知，只進行專家評估，採用情境模擬方式，針對「語意關聯式的新圖文整合系統」所有可能發生情況設計五種查詢情境，若系統運作滿足所有查詢情境，則代表本研究之成果可提供更加符合使用者語意搜尋需求的結果，知識即獲驗證。

第三節 研究方法

一、建立語意網

本研究的語意網來源為與「陽明山國家公園」主題相關的向量圖層、屬性表資料等，藉由人工歸納的方式將資料分為空間詞彙及屬性詞彙，空間詞彙提供系統作為空間定位用途，屬性詞彙則作為圖層載入時的依據；此外，依語意網的關係特性又可區分為圖 3-3 的內部語意關聯及圖 3-4 的外部語意關聯，內部語意關聯主要用途在於將原本零散的地名詞彙彼此建立關係，其來源為現有資源的詞彙庫及外顯知識文件，使詞彙間亦可透過推論模組進行概念的巡覽，而外部語意關聯

則是以人工方式將原本不在詞彙庫中的概念透過特定關係連結到之前建立的「內部關聯語意網」，其來源則不侷限於現有資源及文件，只要與特定概念相關的概念、關係，皆可作為語意網的來源，使原先不在先備資料中的文件詞彙也可透過本研究的語意推論模組而受到先備資料的有效支援。另依空間、屬性特性又可細分成表 3-1 所示。

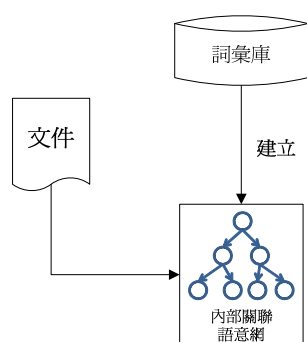


圖 3-3 內部語意關聯

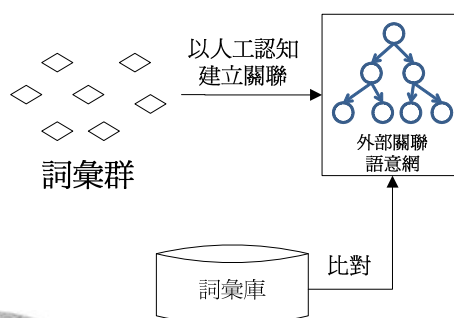


圖 3-4 外部語意關聯

表 3-1 語意關聯種類表

	內部語意關聯	外部語意關聯
空間	●	●
屬性	●	●

「內部語意關聯」為將先備資料中的字彙彼此建立自己的關聯性，僅能做為知識體的組織使用，以地名圖層的建立為例，圖 3-5 左方表格為「地名.shp」向量檔案的屬性表部份內容，在建立地名語意網的過程中，將圖層名稱 (*.shp) 作為一個概念，接著將 Type 欄位名稱作為 hasType 的屬性名稱，接著再將 Type 欄位的值當作地名的子類別，最後將 Name 欄位的值作為每個子類別的實例 (Instance)，如此便完成圖 3-6 的屬性語意網建構；

Name	TYPE
五指山	山嶺
真武寶殿	寺廟
湖底	居住地
富貴山墓園	墓地
綠峰農牧場	農場
...	...

地名.shp → Class Name

TYPE → Attribute Name

寺廟 → Sub Class

綠峰農牧場 → Instance of SubClasses

圖 3-5 建立語意網的來源屬性表

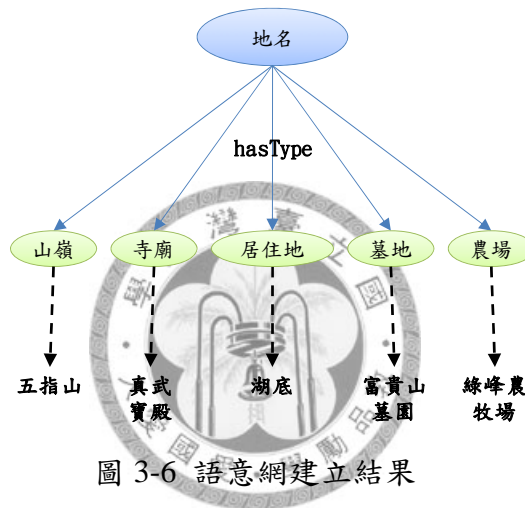


圖 3-6 語意網建立結果

「外部語意關聯」可依詞彙群的來源區分為針對性及一般性外部語意關聯，「針對性外部語意關聯」的詞彙群來源為特定的外顯知識文件或文章，如圖 3-7 的流程，首先將文件中的詞彙藉由斷詞系統進行拆解後成為許多詞彙，此時的詞彙群彼此無法透過任何關係連結，因此需由人工介入方式建立彼此間的相關性，此過程即為針對性外部語意關聯的建立流程。「一般性外部語意關聯」係指詞彙群的來源對象非特定的外顯知識文件或文章，只要研究人員認為與其具有高相關性的詞彙皆可加入，具體案例流程請參閱第四章。

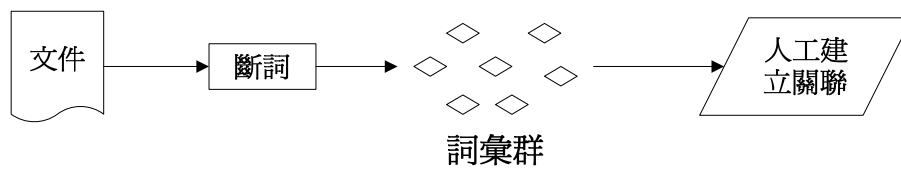


圖 3-7 外部語意關聯建立流程

本研究將空間詞彙作空間定位使用，屬性詞彙則為圖層載入依據，為達上述功能，語意網的定義需由具有資源的詞彙開始建立，當語意網涵蓋的詞彙數量愈多，代表未來針對使用者輸入關鍵字的語意擴張關聯詞數量同樣隨之增加，因此語意網的大小，將與推論引擎可推論的語意詞彙數量成正比關係。

而語意網的建立流程如圖 3-8，範疇中定義語意網需有效支援與陽明山國家公園相關的文件，因此第一步需以「陽明山國家公園先備資料」為初始的語意架構核心逐步增加語意網的規模，如同結晶核的細胞分裂一般，由先備資料中抽離研究所需的空間、屬性詞彙做為結晶核，接著第二步依據文件中對於結晶核的描述文字建立結晶核彼此間的「內部語意關聯」，第三步以人工方式對各結晶核擴張其「一般性的外部語意關聯」，此時的語意網係以一般性（非針對性）方式建立，因此無法完整而有效的解析文件所需的資源，程式將彙整出無法支援的詞彙清單，由人工方式將不支援的詞彙與結晶核建立關係，擴張其「針對性的外部語意關聯」，在完成詞彙與結晶核的關係建立後，此語意網將可有效支援文件所需的語意搜尋需求。

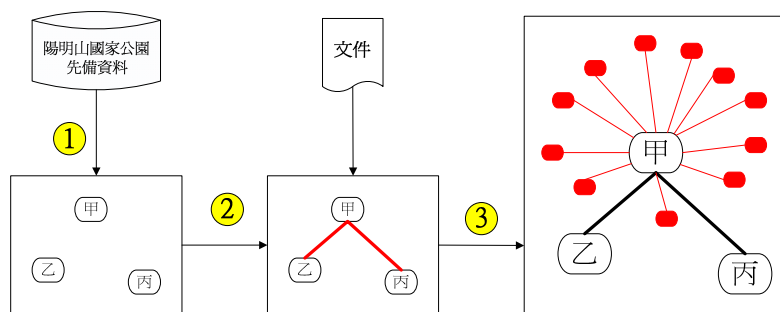


圖 3-8 語意網建立流程圖

以查詢「紗帽山的熔岩」為例，其中紗帽山為空間詞彙，熔岩為屬性詞彙，當資料庫中沒有與紗帽山對應的關鍵字時，便啟動語意推論引擎，至本研究建立的「空間語意網」進行額外概念的推論後得到陽明山這個字彙，達成空間詞彙的語意擴充目的；同樣的對於熔岩屬性詞彙而言，當資料庫中沒有與地質圖對應的關鍵字時，亦將啟動語意推論引擎，至「屬性語意網」進行額外概念的推論後得到火山地質圖，達成屬性詞彙的語意擴充目的。

二、定義語意網的空間與非空間關係

現有的文獻中，較少針對語意網的關係進行探討，因此本研究嘗試根據系統實作的實際需求，將其概分為空間性、非空間性兩大類，在空間性方面，主要以「空間包含關係」及「空間相鄰關係」來說明：

(一) 空間包含關係

在查詢的過程中，常會遇到因為關鍵字的不確定，造成無法搜尋到理想的結果，以空間包含關係為例，使用者面對的是資料「尺度」上的語意落差，圖 3-9 為不同尺度間資料的包含關係。

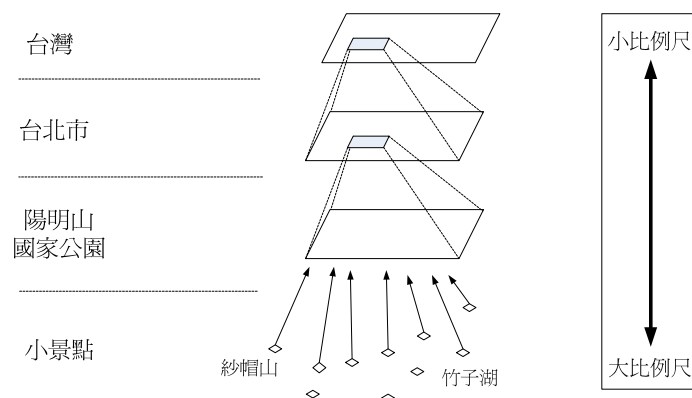


圖 3-9 不同尺度間資料的包含關係

表 3-2 歸納出不同的資料尺度間，其所包含的下層概念，在此僅列出與陽明山國家公園間有較高相關性的概念，而表 3-3 歸納出使用不同資料尺度關鍵字時，可查到的資料清單，在此僅列出與陽明山國家公園間有較高相關性的概念。

表 3-2 不同尺度的下層概念包含情形

資料尺度	包含 (hasContain)
台灣省	台北市、台北縣…
台北市	陽明山國家公園
陽明山國家公園	紗帽山、七星山、竹子山、嵩山… 市立陽明醫院、陽明山管理所、陽明之家… 竹子湖、山豬湖、頭湖、八煙… 陽明公園、六窟溫泉、擎天崗…

表 3-3 不同尺度可查詢到的資料

資料尺度	可查到的資料
台灣省	台灣地形圖、台灣行政區圖…
台北市	台北市旅遊景點地圖、中正區地圖、大同區地圖、中山區地圖、松山區地圖、大安區地圖、萬華區地圖、信義區地圖、士林區地圖、北投區地圖、內湖區地圖、南港區地圖、文山區地圖…
陽明山國家公園	火山地質圖、地質災害圖、地層剖面圖、地形圖、土壤厚度圖、使用分區圖、建築區圖、縣市界圖、鄉鎮市地圖、環境敏感圖、聚落分佈圖、道路圖、橋樑分佈圖…

針對語意推論的需求，建立如圖 3-10 的陽明山國家公園語意網，並在其中定義陽明山國家公園與其包含山脈的關係，經由推論引擎透過 hasContain 關係於不同節點中相互巡覽，達成不同尺度間語意關聯目的，因此當使用者欲查詢紗帽山的火山地質資料時，語意推論引擎將回傳陽明山國家公園火山地質圖，此例的結果對於使用者的查詢語意實際上是有達到相互符合的效果。

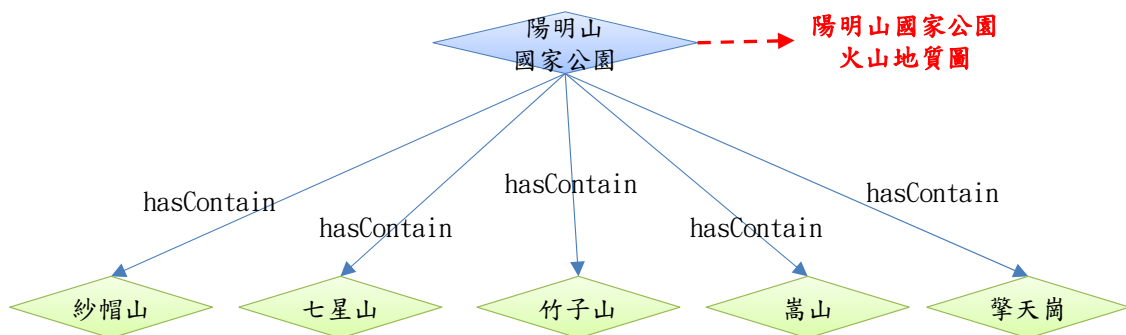


圖 3-10 不同尺度間資料的包含關係

李介中 (2004) 提出階層性地名關係概念，其定義在地名的結構中，較大尺度的地名常可關聯至許多相關的小尺度地名，本研究採用其地名階層架構，定義出不同尺度間地名關係，因此在現有資源的小型空間詞彙，可透過語意網對應到不一定具有資源的大型空間詞彙庫。

由於可用資源有限，本研究將利用大地名關聯至小地名的特性，僅以現有資源為初始詞彙，並定義概念、關係後連接至其他不一定具有資源的大型詞彙庫，藉由此種概念間關聯的特性，可使現有資源的查詢具備語意的彈性，系統也不需具備龐大的詞彙庫，便可達成語意查詢的效果。

(二) 空間相鄰關係

空間關係最常用的除了包含的關係外，亦有相鄰的關係，如圖 3-11 中的地名類別中有頂青巖及中青巖兩個實例 (Instance)，由於彼此在地圖上是具有空間相鄰特性的，因此藉由 hasNeighborWith 關係連接兩個實例，可有效藉由語意網來表達兩個實例的位置相鄰關係。

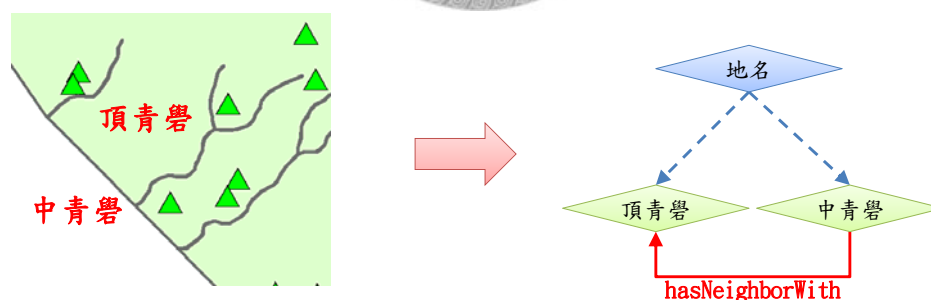


圖 3-11 空間相鄰關係

三、定義關係的反應行為

關係為語意推論時的路徑，本研究將關係區分為空間關係及屬性關係兩大類。空間關係所連接的個體 (Individuals)，依空間資料型態又可區分為點 (Points)、

線 (Polyline)、面 (Polygon) 三類，因此空間關係的反應行為亦需對於各種型態做出不同的定義，本研究初步探討不同關係間所反應的行為差異，其中空間關係對於查詢圖台所反應的行為具有較具體的對應方式，以下將依表 3-4 列出「相鄰」、「包含」及「被包含」三種最具代表性的空間關係反應行為：

表 3-4 空間關係行為反應表

關係	個體型態	反應行為
相鄰	點 A 對點 B	以點 B 為中心繪製 buffer， 且點 A 位於線 B 周圍 100 公尺 buffer 中任一點
	點 A 對線 C	以線 C 為中心製 buffer， 且點 A 位於線 C 周圍 100 公尺 buffer 中任一點
	點 A 對面 D	以面 D 為中心製 buffer， 且點 A 位於面 D 周圍 100 公尺 buffer 中任一點
包含	面 A 包含點 B	Zoom In/Out 至包含所有點 B、線 C、面 D 的最小範圍
	面 A 包含線 C	
	面 A 包含面 D	
被包含	點 A 被包含於面 B (簡單被包含關係)	Zoom In/Out 至面 B 範圍， 且點 A 位於面 B 的範圍中任一點
	點 A 被包含於面 B 及面 C (複雜被包含關係)	以面 B、面 C 產生 Overlay 後 Zoom In/Out 至該區，且點 A 位於面 B 及面 C 的 Overlay 區域中任一點

(一) 相鄰關係：

對於關係的反應行為，必需先定義相關參數供查詢定位圖台使用，例如「相鄰」代表多少距離範圍內的概念，若將其定義為 100 公尺，則當系統使用此關係查詢時，查詢定位圖台則會用已知點為中心繪製 buffer，依個體的型態可分為點對點的關係連接、點對線的關係連接、點對面的關係連接等；在點對點的關係中，當文章中有點 A 空間詞彙，但資料庫僅有與點 A 相鄰的點 B，推論引擎由圖 3-12

得到「點 A 相鄰於點 B」的關係，此時系統會以點 B 為中心繪製 buffer 圓形，並告知點 A 在點 B 附近，呈現圖 3-13 的結果。

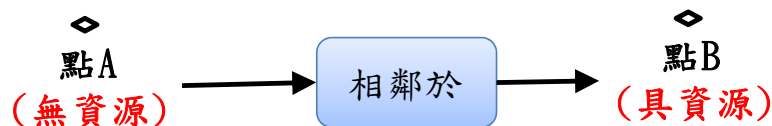


圖 3-12 語意網中關係定義

定義：點 A 相鄰於點 B。

反應：點 A 位於點 B 周圍 100 公尺的 Buffer 中任一點。

行為：以點 B 為中心繪製 Buffer。

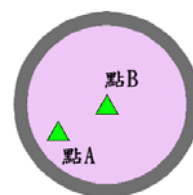


圖 3-13 呈現結果

舉例來說，當使用者閱讀某段文章「臺北市湖田國小位於北投區」時，文件斷詞模組將湖田國小視為未具資源 (Non Resources) 的詞彙，但因為圖 3-14 的語意網定義了「湖田國小相鄰於竹子湖」的關係，因此推論引擎會將湖田國小視為語意網可支援 (Supported) 的詞彙 (參考圖 3-15)，可關聯至有資源 (Resources) 的竹子湖後，查詢定位圖台定位至竹子湖的空間位置，並繪製半徑為 100 公尺的 buffer，告知使用者「湖田國小位於此 100 公尺的 buffer 內的位置」，完成空間關係行為的反應後呈現圖 3-16 的結果，此為點對點的關係定義。

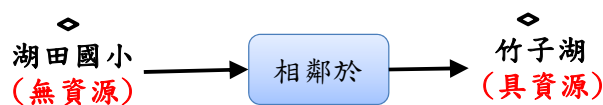


圖 3-14 語意網中關係定義

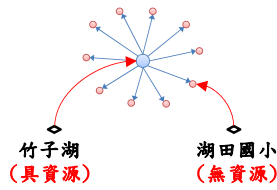


圖 3-15 詞彙種類示意圖



圖 3-16 關係的反應行為

在點對線的關係中，當文章中有點 A 空間詞彙，但資料庫僅有與點 A 相鄰的線 B，而推論引擎由圖 3-17 的語意網找到「點 A 相鄰於線 B」的關係，此時系統會以線 B 路徑為中心繪製 buffer 圓形（參考圖 3-18），並告知點 A 在線 B 附近，在自然語言中對於線的相鄰，同義詞有順著、沿著等。

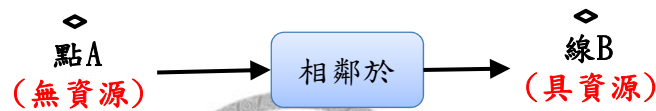


圖 3-17 語意網中關係定義

定義：點 A 相鄰於線 B。

反應：點 A 位於線 B 周圍 100 公尺的 Buffer 中任一點。

行為：以線 B 路徑為中心繪製 Buffer。



圖 3-18 呈現結果

舉例來說，當使用者閱讀某段文章「臺北市湖田國小位於北投區」時，文件斷詞模組將湖田國小視為未具資源 (Non Resources) 的詞彙（參考圖 3-20），但因為圖 3-19 的語意網定義了「湖田國小相鄰於竹子湖路」的關係，因此推論引擎會將湖田國小視為語意網可支援 (Supported) 的詞彙，可關聯至有資源 (Resources) 的竹子湖路後，查詢定位圖台定位至竹子湖路的空間位置，並繪製半徑為 100 公尺的 buffer，並告知使用者「湖田國小位於此 100 公尺的 buffer 內的位置」，完成空間關係行為的反應後呈現圖 3-21 的結果，此為點對線的關係定義。

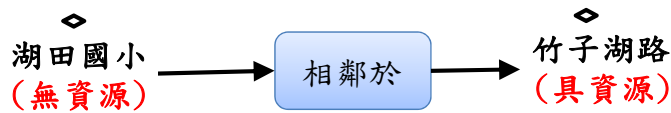


圖 3-19 語意網中關係定義

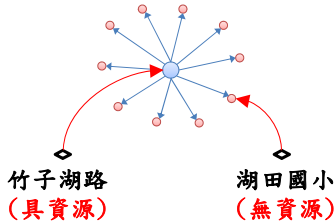


圖 3-20 詞彙種類示意圖



圖 3-21 關係的反應行為

在點對面的關係中，當文章中有點 A 空間詞彙，但資料庫僅有與點 A 相鄰的面 B 時，而推論引擎由圖 3-22 的語意網找到「點 A 相鄰於面 B」的關係，此時系統會以面 B 路徑為中心繪製 buffer 圓形，並告知點 A 在面 B 附近，呈現圖 3-23 的結果。

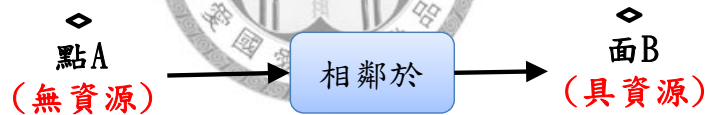


圖 3-22 語意網中關係定義

定義：點 A 相鄰於面 B。

反應：點 A 位於面 B 周圍 100 公尺的 Buffer 中任一點。

行為：以面 B 路徑為中心繪製 Buffer。

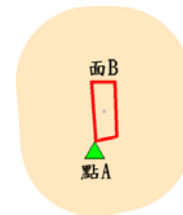


圖 3-23 呈現結果

舉例來說，當使用者閱讀某段文章「臺北市湖田國小位於北投區」時，文件斷詞模組將湖田國小視為未具資源 (Non Resources) 的詞彙，但因為圖 3-24 的語意網定義了「湖田國小相鄰於 A 聚落」的關係，因此推論引擎會將湖田國小視為

語意網可支援 (Supported) 的詞彙 (參考圖 3-25), 可關聯至有資源 (Resources) 的 A 聚落後, 查詢定位圖台定位至 A 聚落的空間位置, 並繪製半徑為 100 公尺的 buffer, 並告知使用者「湖田國小位於此 100 公尺的 buffer 內的位置」, 完成空間關係行為的反應後呈現圖 3-26 的結果, 此為點對面的關係定義。

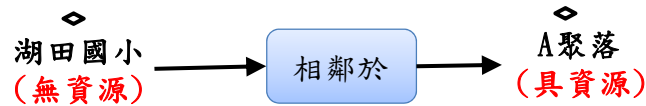


圖 3-24 語意網中關係定義

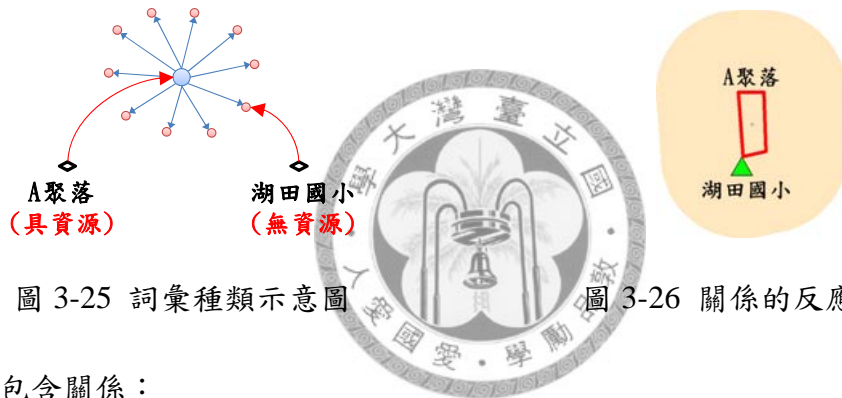


圖 3-25 詞彙種類示意圖

圖 3-26 關係的反應行為

(二) 包含關係：

包含關係依個體的種類可分為面包含點、面包含線、面包含面等, 本研究將其空間行為定義為 Zoom In/Out 至包含所有具備資源的點、線或面的最小範圍, 以圖 3-27 的語意網定義為例, 其最終的反應結果如圖 3-28 所示。

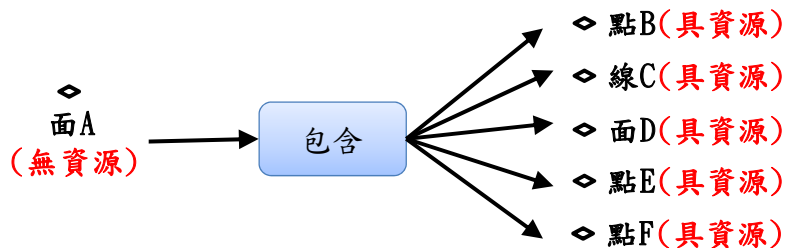


圖 3-27 語意網中關係定義

定義：面 A 包含點 B、線 C、面 D、點 E 和點 F。

反應：面 A 位於包含點 B、線 C、面 D、點 E 和點 F 的最小範圍。

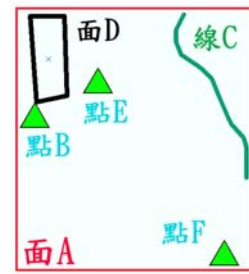


圖 3-28 呈現結果

(三) 被包含關係：

「被包含」關係的定義主要可區分為簡單被包含關係及複雜被包含關係，「簡單被包含」關係如圖 3-29，語意網中僅定義無資源聖道宮被單一具資源的五指山層分佈區所包含，此時僅需 Zoom In/Out 至包含聖道宮的「五指山層分佈區」範圍即可，呈現的結果如圖 3-30 所示，



圖 3-29 語意網中關係定義

定義：聖道宮被包含於五指山層分佈區。

反應：聖道宮位於包含它的五指山層分佈區範圍。

行為：Zoom In/Out 至五指山層分佈區。



圖 3-30 關係的反應行為

「複雜被包含」關係如圖 3-31 所示，語意網中定義了不只一個的包含實例，無資源的點 A 同時被具有資源的面 B、面 C 所包含，空間的反應行為則會將面 B、面 C 產生 Overlay 後 Zoom In 至該區塊，呈現如圖 3-32 的結果。

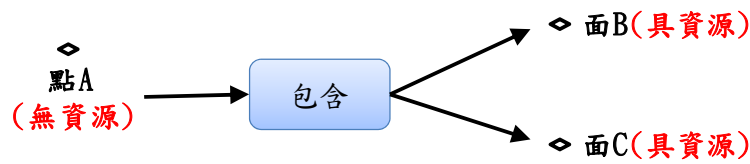
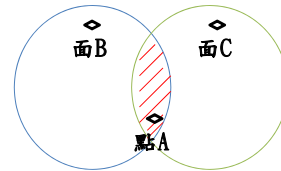


圖 3-31 語意網中關係定義

定義：點 A 同時被面 B、面 C 包含。

反應：點 A 位於面 B 及面 C 的 Overlay 中任一點。



行為：以面 B、面 C 產生 Overlay 後 Zoom In/Out。

圖 3-32 呈現結果

舉例來說，當使用者閱讀某段文章「中山樓除了歷經多次選舉總統，也是歷次修憲所在的場所。」時，文件斷詞模組將中山樓視為未具資源 (Non Resources) 的詞彙，但因為圖 3-33 的語意網定義了「中山樓被包含於凝灰角礫岩分佈區」及「中山樓被包含於 B 聚落」的關係，因此推論引擎會將中山樓視為語意網可支援 (Supported) 的詞彙 (參考圖 3-34)，關聯至有資源 (Resources) 的 B 聚落和凝灰角礫岩分佈區後，以有資源的個體產生 Overlap 後定位至該區，呈現如圖 3-35 的結果。

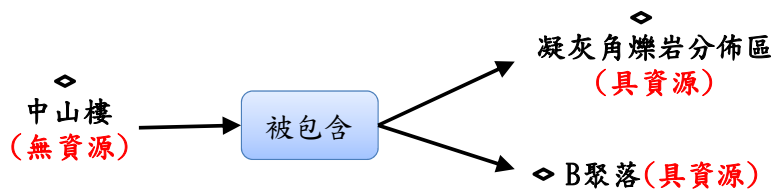


圖 3-33 語意網中關係定義

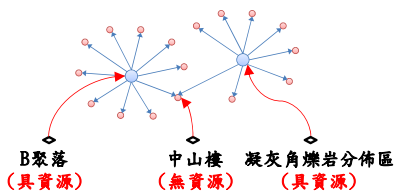


圖 3-34 詞彙種類示意圖



圖 3-35 關係的反應行為

屬性因其不具備空間特性，較難以定義對應的查詢行為，以下列出三種最具代表性的屬性關係：

(1) 在..之下：例如地質學中，五指山層通常位於木山層之下，可藉由定義地層間的出現順序來達到語意關係，具體的查詢案例為使用者想找出五指山可能的分佈情形，本系統藉由「五指山層通常位於木山層之下」這種關係，提供使用者木山層的分佈位置，並告知這些地區底下可能會有五指山層的分佈，達到語意搜尋的目的。

(2) 同義語：將兩個不同的詞彙進行串聯，使之完全相等。

(3) 相反語：明確宣告兩個概念為相異的，使推論機不會因彼此的定義相同而誤判為相等的概念。

四、串聯語意網

完成不同類別間語意網的定義後，可藉由前述的關係來串聯不同的小型語意網；例如目前已建立「地名語意網」及「道路語意網」，可藉由「isContainedBy」關係來串聯不同類別的實例 (Instance)，以圖 3-36 為例，道路語意網的五指山公路及湖底路，分別透過 isContainedBy 關係串聯至地名語意網的五指山與湖底這兩個實例。

而本研究藉由此種案例式的歸納，將不同類別的串聯關係以圖 3-37 表示，縣市類別包含鄉鎮區，鄉鎮區包含村里，村里包含地名，地名包含橋樑，道路包含地名 (如竹子湖路包含擎天崗、福德宮等)，河流包含地名等，在此定義的「hasContain 關係」為類別間的關係定義。

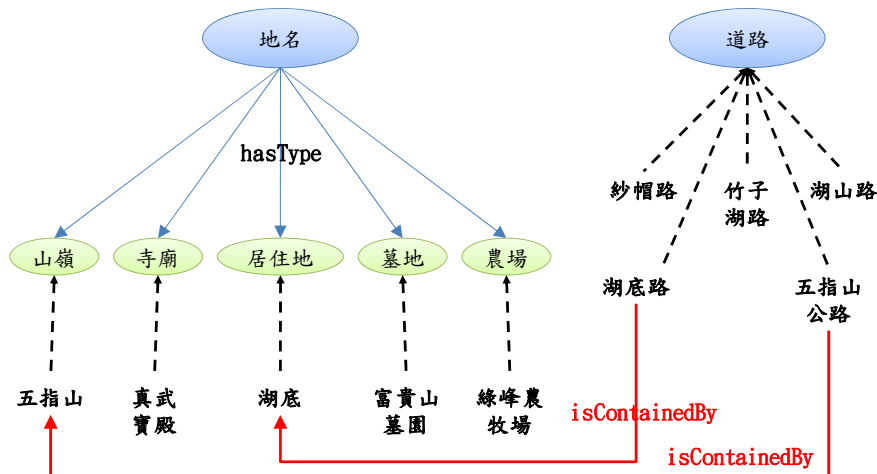


圖 3-36 不同語意網的實例可透過關係串聯

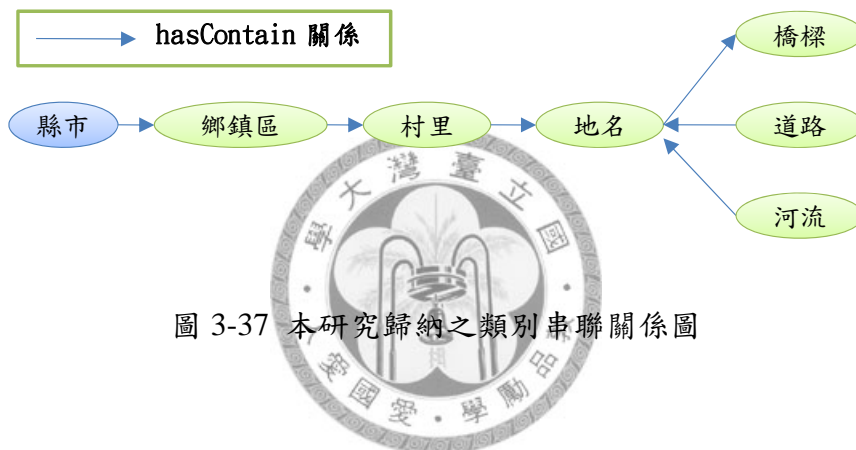


圖 3-37 本研究歸納之類別串聯關係圖

五、聚焦

由於語意網具備聯想的特性，必需依據用途來限定所需的知識文件，否則將會得到過多的結果，本研究依據語意網常使用的推論方式進行條件設定，分別為「限定深度」和「指定關係」兩種。

(一) 限定深度 (Depth)：

語意網的特性中，單一概念可向外延伸至多個概念，因此必需限定語意網擴張的程度。如圖 3-38a 時語意深度為 0，代表為關鍵字比對而無概念擴張，此時未推論新概念，其總概念數量為 1，屬本研究的例外情形，圖 3-38b 為語意深度 1 的推論結果，此時獲得的額外推論概念有 5 個，總概念數為 6 個；圖 3-38c 為語意深度 2 的推論結果，此時獲得的額外推論概念有 6 個，總概念數為 12 個；圖 3-38d

為語意深度 3 的推論結果，此時獲得的額外推論概念為 2 個，總概念數為 14 個，由圖 3-38a 到圖 3-38d 分別代表關聯程度由近而遠，圖 3-38a 為近關聯，圖 3-38d 為遠關聯；從上述例子可知，如果未限定語意網擴張的程度，將會造成概念間無限擴散，反而造成概念發散的負面效果。

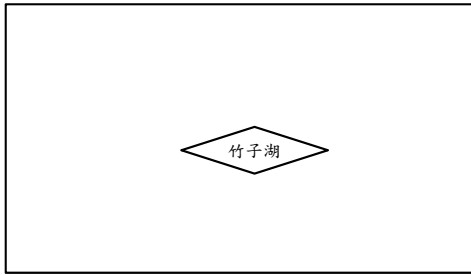


圖 3-38a 語意深度為 0 的推論結果

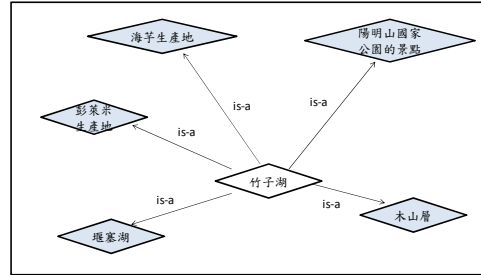


圖 3-38b 語意深度為 1 的推論結果

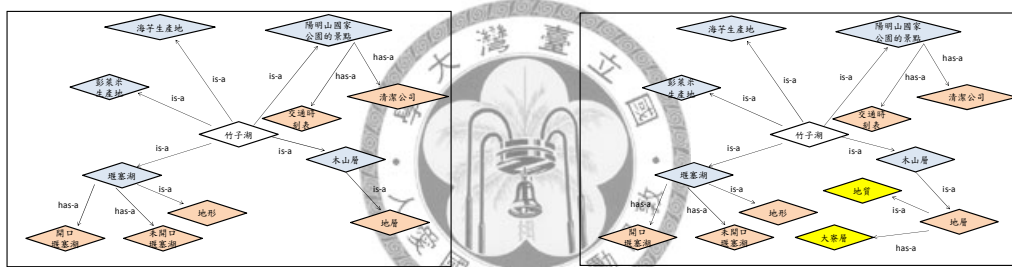


圖 3-38c 語意深度為 2 的推論結果

圖 3-38d 語意深度為 3 的推論結果

當使用者查詢某詞彙時，而資料庫中無相對應的資料時，透過語意擴充引擎的推論，以圖 3-39 為例，可在語意深度為 1 時透過各種不同關係找到 17 個未具備相對應資源的概念，無法提供符合語意搜尋的結果，而進行語意深度 2 的第二次語意推論時，得到 32 個額外的新概念，其中有一個為具有相對應的資源▲A，此時的結果為符合使用者的語意搜尋需求，若使用者對此結果不滿意，則再次整行語意深度為 3 的擴充，得到的 46 個額外概念中，有一個為具備資源的▲B，如此反覆的推論，便可有效達成語意擴充目的。

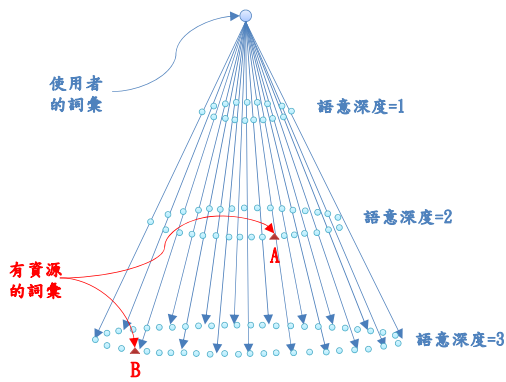


圖 3-39 語意擴張網路圖

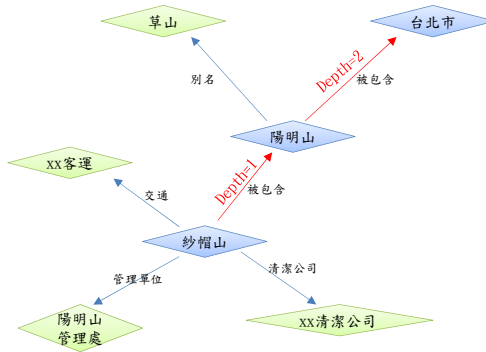


圖 3-40 單一概念的不同主題關聯

將不同深度的概念建立彼此關聯後，語意推論引擎可於特定語意網中巡覽，如圖 3-41 中 A 可關聯至 B，B 又可透過關係關聯至 C，由 A 到 B 的語意深度為 1，由 B 到 C 的語意深度為 2，藉由推論深度的增加，構成圖中的語意關係鏈，達到語意推論目的。

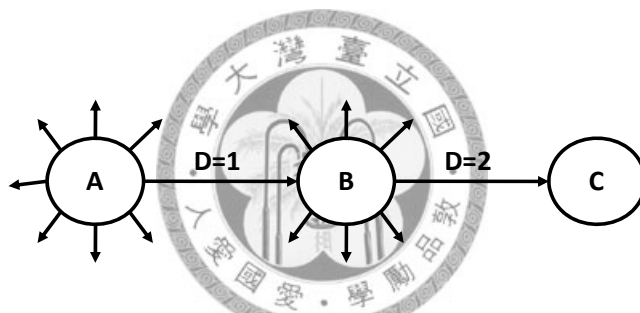


圖 3-41 語意關係鏈

(二) 指定關係 (Relationships)：

關係可用來連結兩種不同的概念，藉由定義多個不同概念間彼此的關係，可以建立龐大的語意網，以圖 3-2 的紗帽山為例，其具有各種關係如被包含、交通、管理單位及清潔公司等，假設使用者希望藉由系統取得與「紗帽山的地質圖」相似的資源，此時正確的關係為「被包含」，即透過「紗帽山被包含於陽明山」的關係，系統可取得陽明山的地質圖，而在語意深度為 2 時同樣藉由「被包含」關係得到台北市地質圖，因此由使用者選擇感興趣的主題將可提高搜尋結果的準確性。

六、語意資料庫的擴充機制

如前所述，建立語意網的方式是以陽明山相關的知識文件做為結晶核的外部語意關聯，完成建立特定文件與語意網間的關係後，此語意網便可有效針對特定文件中出現的詞彙推論其相關語意，但外顯化知識文件的數量隨時間不斷更新及增加，當新的文件運用此語意網推論時，難免有些詞彙無法被包含，在此提出語意資料庫的擴充策略，以確保語意網的可用性。

本研究的語意網是以特定知識文件作為外部語意關聯，若應用此本體於同領域的外顯知識文件時，將依圖 3-42 的流程圖運作，對於與來源文件相同的詞彙將可被完整支援，但若為新的詞彙則無法被語意推論模組所支援，此時將文件中無法被支援的詞彙進行歸納後回傳詞彙清單，並啟動語意網的擴充策略。

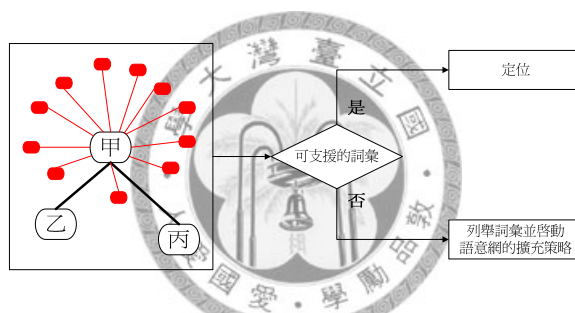


圖 3-42 語意資料庫的擴充流程圖

語意網的擴充策略依對象可分為詞彙庫的擴充及語意網的擴充兩種方式；其中詞彙庫又可依空間、非空間特性細分為「地名詞彙庫的擴充」及「屬性詞彙庫的擴充」，當地名、屬性詞彙庫的規模愈大表示圖 3-8 中的結晶核數量也會隨之增加，與關鍵字成功對應的機率亦將隨之提升；而語意網同樣依查詢對象（即語意網中的個體）對應資源不同分為「空間語意網」與「屬性語意網」，透過概念與關聯數量的增加，語意網規模也會逐漸增加。

本研究所提出的擴充策略主要以半自動化方式進行，將無法支援的詞彙自動歸納並列舉後提供給使用者，並由使用者決定要針對哪種資料對象進行擴充，若

為語意網的擴充時，需以人工方式建立列舉詞彙與語意網間的關係，以完成語意資料庫的擴充。

第四節 研究工具

一、語意網建立工具：Protégé 3.4

本研究採用 Protégé 3.4 rt1 版 (Stanford Medical Informatics, 2005) 所發展的來建立所需的語意網，由於 Protégé 可完整支援網頁本體語言 (Web Ontology Language, OWL) 格式，且其提供框架式的易擴充性架構，目前已提供豐富的外掛程式可以將語意網以圖形方式有效效現，所以我們採用 Protégé 來發展本研究所需要的語意網，其軟體操作畫面如圖 3-43 所示。

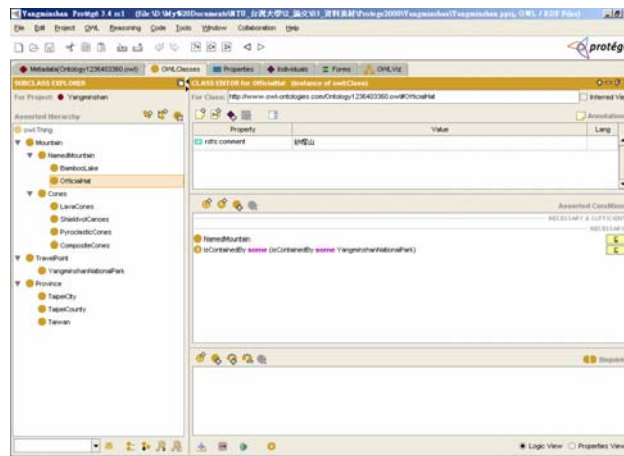





圖 3-43 Protégé 操作畫面

本研究考量語意資料庫的擴充性，空間、屬性詞彙庫使用一般的資料庫語法直接進行新增，而空間、屬性語意網則需使用本研究提供的界面新增，語意網擴充程序如表 3-5，首先將句子中無資源的名詞如「熔岩、火山、地質…」列舉供使用者選擇，接著列出資料庫中有資源的詞彙如「火山地質圖、使用分區圖…」供選擇，最後列出資料庫中已知的關係讓使用者選擇，此步驟所選取的關係用來連

接前兩步驟所選取的詞彙，將原本無資源的詞彙藉由關係連結至有資源的詞彙，供未來語意推論使用。

表 3-5 語意網擴充程序表

步驟	範例	示意圖
列出無資源詞彙清單	熔岩、火山、地質	
列出有資源詞彙清單	火山地質圖、地名圖、使用分區圖...	
列出關聯	包含、被包含、相鄰	

二、查詢定位圖台：ArcGIS 9.2

根據 Wikipedia (2009) 的定義，ArcGIS 是由 ESRI 公司所發展出來的一個可涵蓋 GIS 常用功能的軟體，桌上型 GIS 可分為 ArcReader、ArcView、ArcEditor 與 ArcInfo 四個等級，ArcReader 僅允許使用者瀏覽查詢但無法修改的功能，Arcview 允許使用者察看空間資料、建立地圖和執行基本的空間分析功能，ArcEditor 可說是 ArcView 進階版，包含許多針對 shapefiles 所提供較進階的工具和空間資料庫等，ArcInfo 包含資料編輯、維護和分析的功能，它們都是 ArcGIS 產品的一部份，而 ArcGIS 亦針對行動式裝置提供許多擴充套件。

圖 3-44 的 ArcGIS 9 軟體架構中，依產品用途定位不同區分為 Desktop GIS (桌上型 GIS)、Embendedd GIS (嵌入式 GIS)、Server GIS (伺服器端 GIS) 與 Mobile GIS (行動式 GIS)，其中大部份的功能可藉由 ArcObject 開發，本研考即透過 VBA 呼叫 ArcObject 的函式庫開發可運作於 Desktop GIS 的輔助系統。

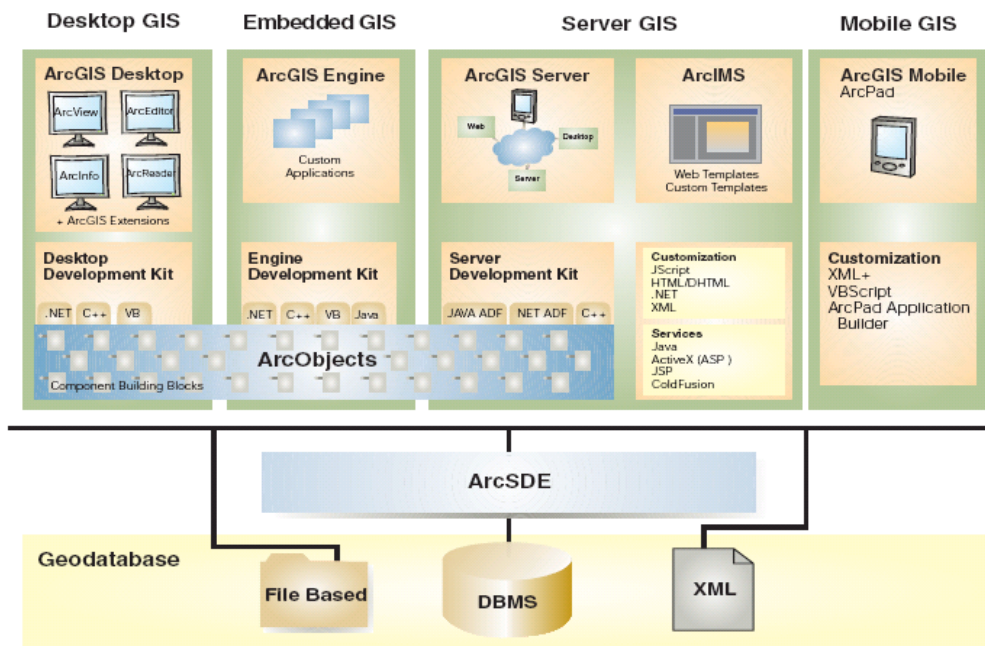


圖 3-44 ArcGIS 軟體架構

資料來源：ESRI (2004)

資料儲存格式可分為檔案式 (*.shp)、關聯式資料庫與 XML 三種，而在軟體設計的三層架構中，位於中介層的元件則可選擇使用直接存取或透過 ArcSDE，多種資料庫儲存及存取方式皆可依需求自行組合。

三、系統整合工具：Java

Java 是一種高階程式語言，可用來開發各類桌上型電腦應用程式、行動裝置應用程式、網頁、伺服器應用程式等，其具有跨平台的特性，也就是說在某作業系統所編寫的軟體，可運行在其他任何平台。此外，由於 Protégé 係由 Java 語言所撰寫而成，且其針對 Java 提供許多編輯 Ontology 的函式庫，因此本研究選擇 Java 作為軟體整合的平台。


四、詞性標註工具：中研院中文斷詞系統

詞是最小有意義且可以自由使用的語言單位。任何語言處理的系統都必須先能分辨文本中的詞才能進行進一步的處理，例如機器翻譯、語言分析、語言了解、

資訊抽取。因此中文自動分詞的工作成了語言處理不可或缺的技术（中研院詞庫小組，1993）。

目前中研院已提供中文斷詞系統供測試使用，斷詞服務開放一 API 供用戶端呼叫，資料的交換方式採用 XML，用戶端可自行撰寫程式經由 TCP Socket 連線傳送驗證資訊及文本至該伺服器，伺服器經過處理後經由原連線傳回結果。此線上服務為簡介中所提及斷詞系統之簡化版本，僅提供以基本詞典進行斷詞，並且輸出簡化詞類之標記，此系統僅支援 Big5 編碼之中文，在使用該系統前必需線上事先申請合法帳號，並撰寫程式送出指定格式的資料，例如將想進行標註的文章以<text></test>作為開始、結束的標記，其用戶端傳送資料之 XML 格式如下：

```
<?xml version="1.0" ?>
<wordsegmentation version="0.1">
<option showcategory="1" />
<authentication username="帳號" password="密碼" />
<text>台新金控 12 月 3 日將召開股東臨時會進行董監改選。</text>
</wordsegmentation>
```



接著系統將回傳完成詞性標記後的結果，原則上是以<sentence></sentence>作為開始、結束的標記，並在每個單詞之後以中括弧註記其詞性，詳細的詞性代碼意義請參考附錄一、二，系統回傳的結果如下：


```
<?xml version="1.0" ?>
<wordsegmentation version="0.1">
<processstatus code="0">Success</processstatus>
<result>
  <sentence> 台新 (N) 金控 (N) 1 2 月 (N) 3 日 (N) 將 (ADV) 召
  開 (Vt) 股東 (N) 臨時會 (N) 進行 (Vt) 董監 (N) 改選 (Vt) ，
  (COMMACATEGORY) </sentence>
</result>
</wordsegmentation>
```

本研究運用該系統所提供的詞性標註功能，作為關鍵詞抽離用途，目前僅針對名詞系列如 Na、Nb、Nc、Ncd...作為關注對象，其他詞性則留給未來相關研究者進行探討。





第四章 研究成果與討論

本研究之實作系統根據研究架構如圖 4-1 所示可分為三部分，第一部份是文件斷詞模組，此模組主要將文章中所隱含的空間、屬性詞彙進行抽取，所使用的是中研院詞庫小組提供之斷組模組線上伺服器，送出文字後由伺服器送回斷詞結果；第二部份是語意推論模組，利用 Protégé-OWL 結合本研究的推論模組進行語意推論，以達成使用者查詢語意理解的目的；第三部分是查詢定位圖台，在 GIS 介面下可呼叫第一部份之文件斷詞模組及第二部份之語意推論模組，並將結果送至資料庫查詢，所得之查詢結果即可展示於 GIS 畫面並進行簡單操作。

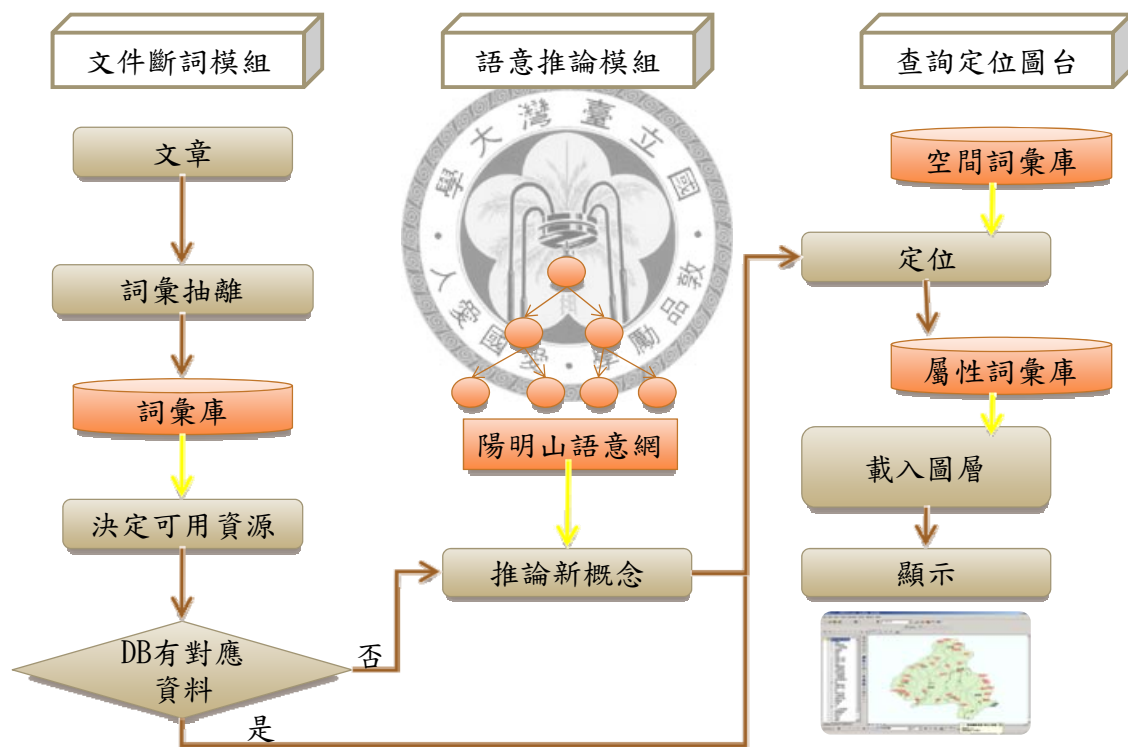


圖 4-1 系統運作流程圖

在系統運作流程中，使用者輸入的自然語言式查詢語句會由「自然語言解析模組」進行單詞的拆解，依詞性將字彙區分開後接著至詞彙庫中找尋可用的資源，如果有找到使用者所需要的資源時，便直接進入到圖文整合模組；但若未找到能夠符合使用者需求的對應資源時，便會啟用「語意推論模組」進行字彙的語意擴充，此模組的語意推論機制便是藉由本研究所建立的陽明山語意網所達成的，得到適合的單詞後，同樣導至「圖文整合模組」中。而在圖文整合模組，主要負責使用者需要的空間定位點及非空間定位屬性圖層的呈現。

第一節 詞彙庫及語意關聯研究成果

本研究所使用的詞彙庫可概分為空間詞彙庫及屬性詞彙庫，空間詞彙庫主要紀錄具有明確空間位置的點位，如陽明山國家公園為面型態的定位點，紗帽山為點型態的定位點等，目前已完成 370 筆資料的建置，其部份資料如表 4-1 所示；而屬性詞彙庫主要用來紀錄可對應資源有效關鍵字，如火山地質圖、土地分區圖等，目前已完成 53 筆資料建置，可對應至 18 個圖層，其部份資料如表 4-2 所示，在建立的詞彙庫將提供文件斷詞模組及語意推論模組做資源比對使用。

表 4-1 空間詞彙庫部份資料

類型	詞彙	類型	詞彙	類型	詞彙	類型	詞彙
縣市 (3)	台北縣	橋 (30)	二股橋	路 (62)	二子坪步道	地名 (218)	摸乳基
	基隆市		二重橋		十八分產業道路		七星山
	台北市		三重橋		大坪產業道路		頂山
鄉鎮區 (9)	七堵區		上磺溪橋		山城路		紗帽山
	三芝鄉		大屯一號橋		中正產業道路		鵝米山
	士林區		中正橋		中正路		土地公嶺
	北投區		永春橋		中湖戰備道路		嵩山
	石門鄉		竹里一橋		中興路		竹子山
	汐止市		行義橋		五指山公路		洪爐山
	金山鄉		西內柑宅橋		六股林口路		磺山
	淡水鎮		青巒橋		巴拉卡公路		後山

表 4-1 空間詞彙庫部份資料 (續)

類型	詞彙	類型	詞彙	類型	詞彙	類型	詞彙
	萬里鄉		流水小橋		半嶺產業道路		百拉卡山
村里 (31)	大屯里		峰頂橋		平菁街		二子山
	山溪村		馬槽橋		石子路		大屯山
	中心里		彩虹橋		尖山湖路		磺嘴山
	六股村		頂湖橋		竹子湖路		水井尾山
	友二里		湖田橋		老崩山路		五指山 (北五指)

表 4-2 屬性詞彙庫部份資料

類型	詞彙	類型	詞彙
地質災害 (8)	土石堆積	地質 (19)	大屯山凝灰角礫岩
	崩塌地		大寮層
	崩塌地		五指山層
	溫泉及天然氣		木山層
	礦場及採石場		火山泥流堆積物
	土石流		玄武岩質熔岩
	向源侵蝕		石底層
	河岸侵蝕		含普通輝石角閃石紫蘇輝石安山岩
土地利用 (8)	公園綠地		含普通輝石紫蘇輝石安山岩質熔岩
	行水區		含橄欖石角閃石安山岩質熔岩
	住宅區		含橄欖石兩輝角閃石安山岩質熔岩
	保護區		含橄欖石普通輝石安山岩質熔岩
	停車場用地		角閃石兩輝安山岩質熔岩
	墓地		兩輝安山岩質熔岩
	學校用地		兩輝角閃石安山岩質熔岩
	機關用地		現代沖積層
其他 (7)	地質敏感		紫蘇輝石角閃石安山岩質熔岩
	環境敏感		熱水換質帶

語意關聯依特性可分為詞彙庫的內部語意關聯及外部的語意關聯，依其空間、非空間特性又可分為具有空間性（空間類）及不具空間性（屬性類）資料，本研究建立圖 4-2 的「空間內部語意關聯」，此為與本研究範疇相關的行政區劃分資訊，並以縣市、鄉鎮區及村里為三種主要類別 (Class)，其餘如台北市、台北縣及基隆市則為隸屬於縣市類別的個體 (Individual)，不同個體隸屬於不同類別，藉由指定關係的 Domain 與 Range 後，即可將不同個體建立關係，達成彼此串聯的目的。

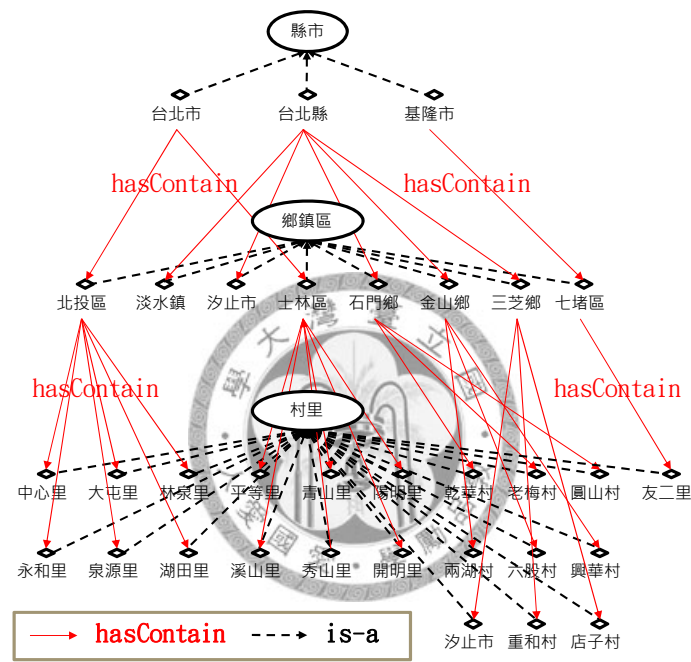


圖 4-2 空間內部語意關聯

圖 4-3 則為本研究建立的「屬性內部語意關聯」，不同的類別彼此具有繼承關係，如山嶺、寺廟、居住地等皆為地名的子類別，各類別的個體間同樣具有關係連結，如竹子湖路依 hasContain 的關係與竹子湖、竹子湖觀光農園及湖田國小三個體相關聯，在語意上來說即代表「竹子湖路包含竹子湖（居住地）、竹子湖觀光農園（農場）和湖田國小（學校）三個體」。

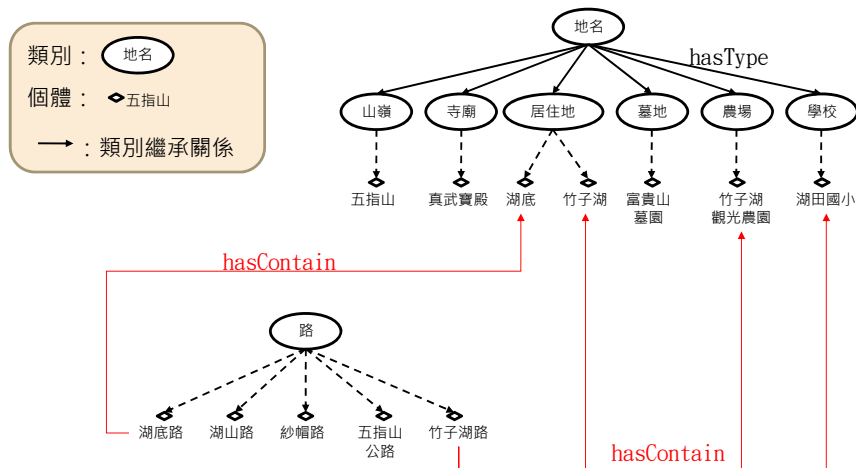


圖 4-3 屬性內部語意關聯

第二節 文件斷詞模組

一、系統簡介

此程式利用 Java 作為系統之開發語言，為繼承沈嘉慧 (2008) 提出之斷詞程式概念而來，係針對本研究目的及範疇加以改良後形成文件斷詞模組。文件斷詞的流程如圖 4-4 所示，首先由使用者選取欲處理的文章或段落進行斷詞，接著透過網路呼叫中研院提供之詞性標記系統執行詞性標註，第三步的詞彙分類則與空間、屬性詞彙庫進行關鍵詞的比對後區分為空間名詞、屬性名詞、語意網支援的名詞、未知名詞與未知其他詞 5 類詞彙，此時程式歸納的空間名詞、屬性名詞已可作為查詢定位圖台參數使用，因此由使用者判斷是否為足夠的搜尋結果，若足夠便產生「詞彙庫比對結果.txt」文字檔供查詢定位圖台使用；若不足則將「語意網支援的名詞」作為語意推論模組的參數，完成推論的運算後回傳具有資源的推論結果，產生「語意推論結果.txt」文字檔供查詢定位圖台使用。

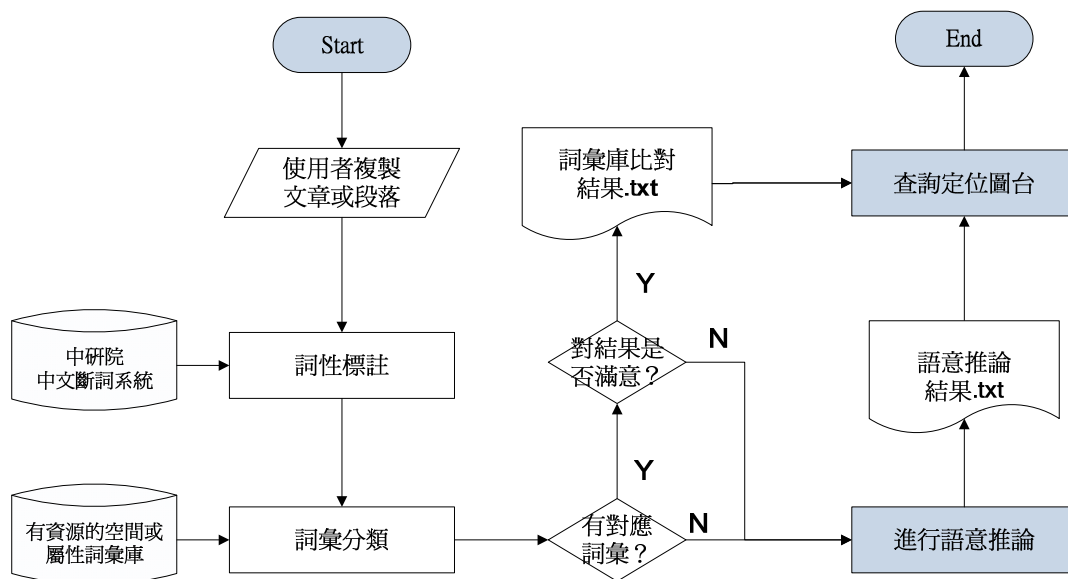


圖 4-4 文件斷詞流程

程式介面如圖 4-5 所示，左方為文件斷詞模組功能區，可分為來源文章（區塊 1）、斷詞結果（區塊 2）及詞彙分類（區塊 3）三部分，文件斷詞是將欲處理的文字貼至區塊 1 後，按下「送出」即執行文件斷詞及語意推論等程序，區塊 1 的文章將執行斷句、去除標點符號後，傳送至中研院斷詞伺服器處理並將回傳結果置於區塊 2，接著執行詞彙的分類，區分為空間名詞、屬性名詞、語意網支援名詞、未知名詞與未知其他詞 5 類詞彙，將分類結果置於區塊 3。

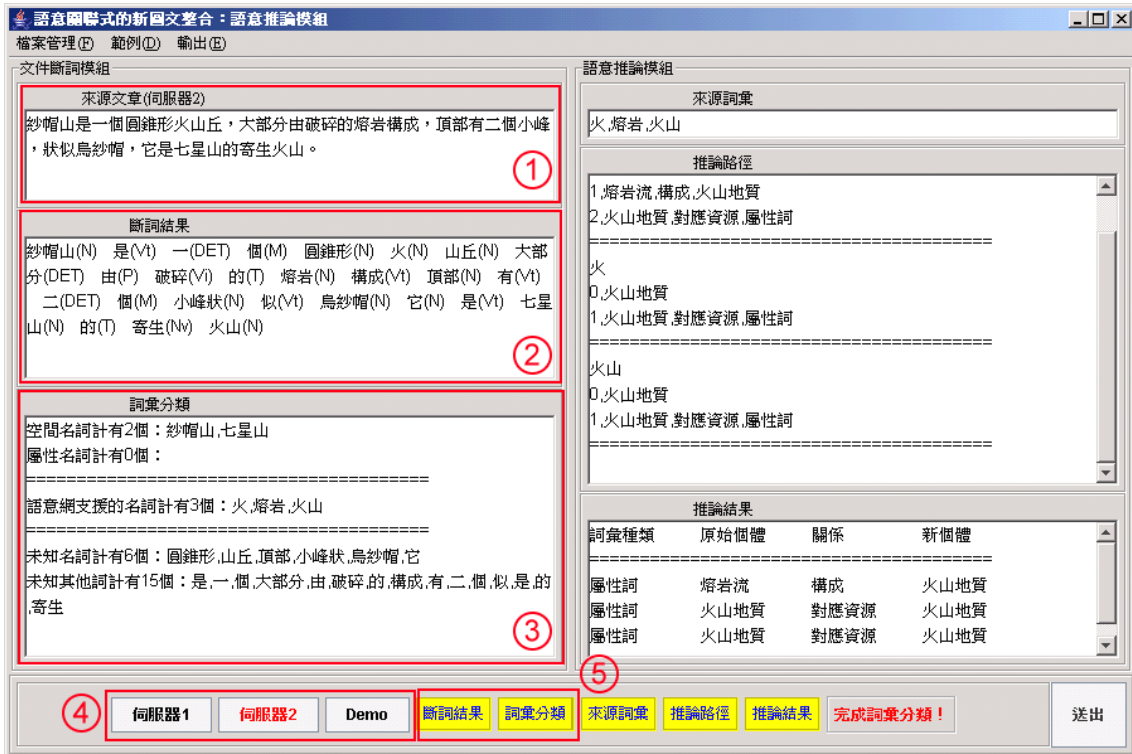


圖 4-5 文件斷詞模組界面

左下方 (區塊 4) 為中研院斷詞系統的伺服器選擇按鈕，由於斷詞系統僅供學術測試使用，不保證其可用性，因此本研究分別設定「伺服器 1」與「伺服器 2」為其位於中研院的主要伺服器及位於交通大學的備用伺服器，此外為了保證系統運作的可用性，本研究額外建立了「Demo」模式，將範例語句斷詞結果進行儲存，以確保後續語意推論程序不會因為伺服器停止服務而中斷。中下方 (區塊 5) 為系統運作的進度燈號，在完成文件斷詞後，斷詞結果燈號將變為黃色，而在完成詞彙分類後，詞彙分類燈號將變為黃色，如此可更明確理解系統運作的情形。

二、系統範例

接著以「紗帽山是一個圓錐形火山丘，大部分由破碎的熔岩構成，頂部有二個小峰，狀似烏紗帽，它是七星山的寄生火山。」段落為例進行程式功能驗證。

將例句貼至圖 4-5 區塊 1 的「來源文章」後按下送出，此例句將進行詞性標註處理，回傳結果置於區塊 2 的「斷詞結果」中，內容如下：「紗帽山 (N) 是 (Vt) 一 (DET) 個 (M) 圓錐形 (N) 火 (N) 山丘 (N) 大部分 (DET) 由 (P) 破碎 (Vi) 的 (T) 熔岩 (N) 構成 (Vt) 頂部 (N) 有 (Vt) 二 (DET) 個 (M) 小峰狀 (N) 似 (Vt) 烏紗帽 (N) 它 (N) 是 (Vt) 七星山 (N) 的 (T) 寄生 (Nv) 火山 (N)」，接著進行「詞彙分類」並將結果置於區塊 3，其中「紗帽山、七星山」在與詞庫做比對後被歸納為空間名詞，此例無屬性名詞詞彙，而「火、熔岩、火山」則為語意網支援的名詞，「圓錐形、山丘、頂部、小峰狀、烏紗帽、它」為未知名詞，「是、一、個、大部份、由、破碎、的、構成、有、二、個、似、是、的、寄生」為未知其他詞；此時使用者可決定是否要使用語意推論模組進行詞彙的語意關聯，或直接至查詢定位圖台查詢 GIS 資料。

三、小結

中研院斷詞系統處理文本之原則可接受文章斷詞，但考慮到本研究範疇，若以整篇文章加以斷詞，傳送資料花費時間較久，且結果過於複雜不易判斷閱讀者閱讀文字之資訊需求，所以斷詞原則以標點符號兩個句點間的一個文句為限，若使用者輸入多於一個文句的文字內容，程式會自動判斷第一個句點前的文字作為斷詞依據。

文件斷詞模組分類結果係作為語意推論模組及查詢定位圖台的來源參數，因此詞彙分類演算法為影響整體結果之關鍵因素，本研究依目的及範疇改良前人提出的斷詞程式，使得模組間的交互運作配合的更好。

第三節 語意推論模組

一、系統簡介

語意推論模組使用 Java 為開發語言，可透過 Protégé 提供的 Jena API (Knublauch, 2009) 進行知識本體的巡覽，及圖 4-6 的流程推論新概念，於步驟 A 時取得待推論的來源個體如「火、熔岩、火山」，步驟 B 則會以模糊比對找到「熔岩、火山熔岩…」等在語意網中無對應資源的個體，步驟 C 將出現如表 4-3 的推論路徑結果，並由程式判斷是否為「對應資源」關係，若是則為具有資源詞彙而依止推論行為，步驟 D 時得到推論結果如：熔岩（無資源）與火山地質（有資源）間的關係後，以「火山地質」詞彙作為查詢定位圖台的主題圖層（屬性詞）。

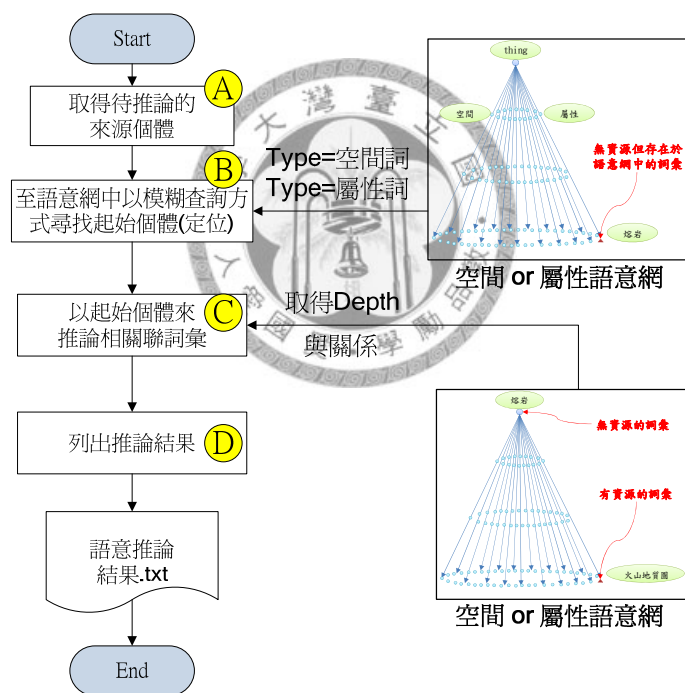


圖 4-6 語意推論流程圖

表 4-3 熔岩推論路徑表

深度	原概念	關係種類	關係	新概念	有資源
1	熔岩	ObjectProperty	構成	火山地質	
2	火山地質	DatatypeProperty	對應資源	火山地質圖	☆

關係種類依 OWL 的定義可以分為 Object Property 及 Datatype Property 兩類，ObjectProperty 為用來連接兩個不同個體的「物件關係」，如「熔岩・構成・火山地質」規則中的「構成」關係即屬之，係用以連接 2 個或多個不同個體，藉由串聯多個體彼此的 ObjectProperty 可達成個體巡覽目的；Datatype Property 則是用來連結概念與文字間的關係，本研究定義「對應資源」關係來串連概念與資源間的關係，並藉此建立研究所需的語意推論引擎。

程式界面如圖 4-7，右方為語意推論模組功能區，可分為來源詞彙（區塊 1）、推論路徑（區塊 2）與推論結果（區塊 3）三部份，來源詞彙係指由文件斷詞模組所完成分類的詞彙，將「語意網支援的名詞」作為區塊 1 的來源，並由本研究發展之推論引擎執行新個體的推論，透過研究建立的「陽明山國家公園語意網」進行關聯的尋找，因此語意網為推論引擎的知識來源。

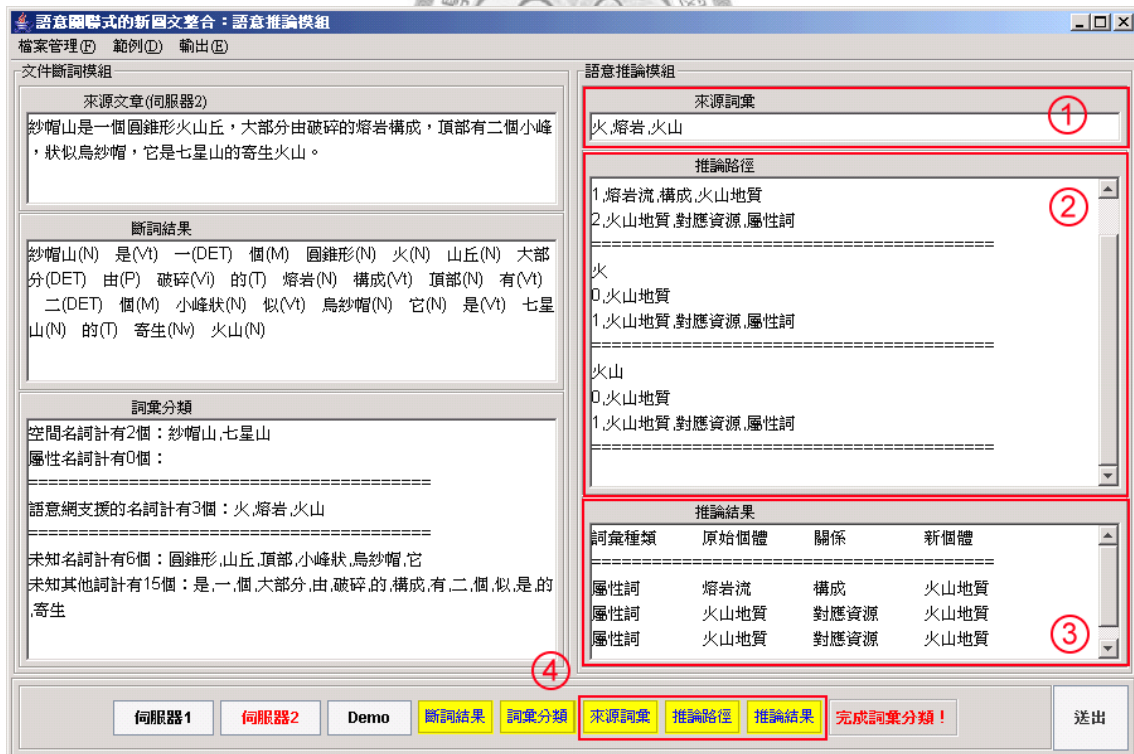


圖 4-7 語意推論模組界面

二、系統範例

同樣由文件斷詞模組處理「紗帽山是一個圓錐形火山丘，大部分由破碎的熔岩構成，頂部有二個小峰，狀似烏紗帽，它是七星山的寄生火山。」為例進行驗證，文件斷詞模組在處理完後無法找到相對應的資源，使用者決定將名詞詞彙交由語意推論模組進行關聯的尋找，推論引擎根據圖 4-8 的語意網計算出「熔岩」一詞的推論路徑如表 4-3 所示，語意網中定義「熔岩·構成·火山地質」規則，推論引擎以熔岩為起始概念，透過「構成」的關係連結到火山地質概念，此時推論引擎會尋找火山地質的「對應資源」屬性中是否有值，若有，則認定此概念為具備對應資源的新概念，完成語意的推論。



圖 4-8 熔岩與火山地質的關係

以「紗帽山」的推論路徑為例，代表紗帽山同時透過「被包含於」的關係與「陽明山國家公園」及「無資源面景點」兩個概念連接，此時推論引擎會各別至新概念中檢查其「對應資源」是否有值，若無，則推論引擎將在使用者限定的最大深度範圍中進行所有概念的巡覽，若有，則停止巡覽行為，回傳找到的新概念。區塊 1 的表格將隨詞彙的數量而改變，此例具有「紗帽山」及「熔岩」兩個詞彙，因此區塊 1 將產生兩個表格，並用以呈現各別的推論路徑。區塊 2 則為推論引擎的推論過程及每個指令執行的時間，將資訊呈現給使用者將有助於瞭解推論的情形及過程。

第四節 查詢定位圖台

一、系統簡介

查詢定位圖台採用 ESRI 的 ArcGIS 9.2 系列軟體作為平台，使用 VBA 作為開發語言，藉以讀取文件斷詞模組及語意推論模組結果，將空間詞彙提供定位用途，屬性詞彙作為圖層載入依據等。其流程如圖 4-9，程式在開啟時首先將「詞彙庫比對結果.txt」與「語意推論結果.txt」載入，功能依特性可分為無推論的藍色區塊、有推論的紅色區塊、尋找定位點、Buffer、Overlay 及重新讀取六大功能區。「無推論的藍色區塊」主要是讀取由詞彙庫比對結果.txt 而來的空間定位點及屬性詞彙，其中屬性詞彙又依載入、未載入分別提供載入圖層及刪除圖層等功能；「有推論的紅色區塊」則讀取由語意推論結果.txt 而來的空間定位點及屬性詞彙，其中屬性詞彙亦依載入、未載入分別提供載入圖層及刪除圖層等功能；「尋找定位點」則是提供搜尋具有空間位置的紀錄，首先輸入關鍵字進行尋找後完成定位；

「Buffer」提供繪製緩衝區的功能，但需事先選定具有空間位置的定位點（包含點、線、面資料型態），最後按下 buffer 按鈕後完成繪製；「OverLay」提供計算兩封閉多邊形的交集區域，因此需事先選擇 2 個 Polygone，接著按下 OverLay 按鈕後完成繪製；「重新讀取」則是提供重新由文字檔讀取推論結果的功能。

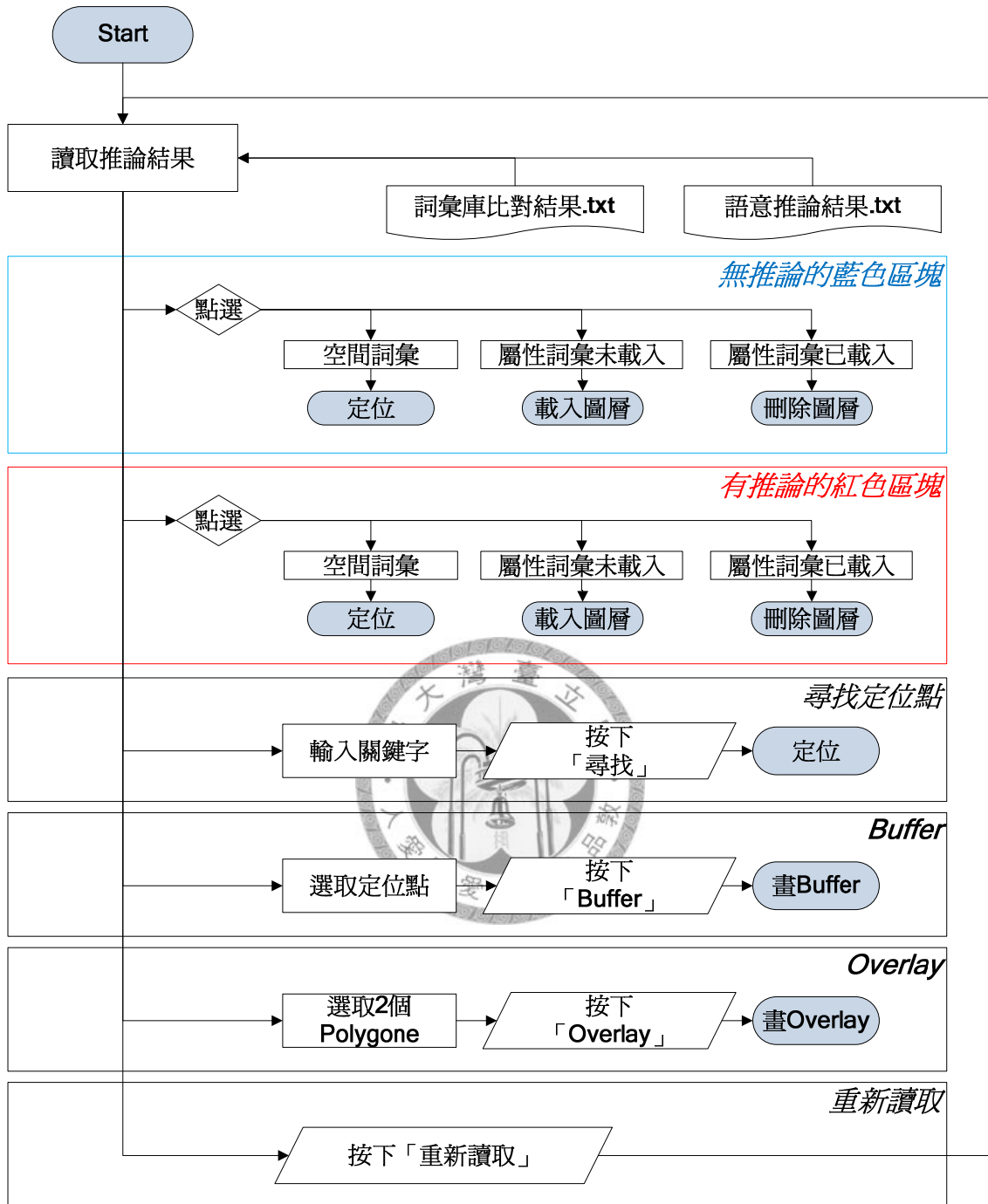


圖 4-9 查詢定位圖台流程

查詢定位圖台界面如圖 4-10 可分為 7 個區塊，其中區塊 1~3 內容係由「詞彙庫比對結果.txt」文字檔讀取而來，區塊 4~6 內容則由「語意推論結果.txt」文字檔讀取而來，區塊 7 則提供空間定位點的名稱搜尋（分為名稱完全比對與名稱模糊比

對兩種)、Buffer 繪製功能、OverLay 繪製功能與重新讀取推論結果等功能，作為滿足本研究程式驗證所需的各項功能。

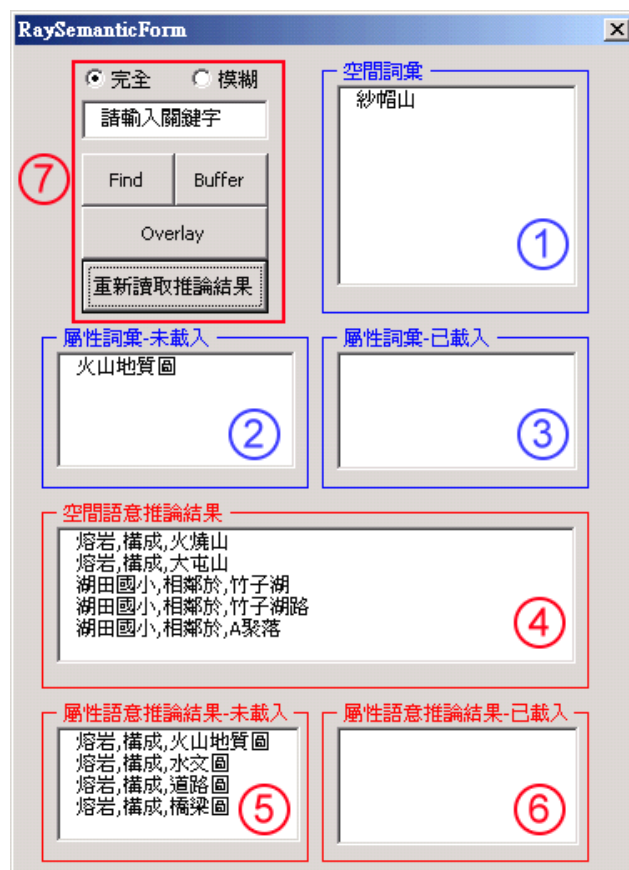


圖 4-10 查詢定位圖台界面

文字在經過斷詞及比對後篩選出欲使用於查詢之關鍵字，並將關鍵字送入查詢相關圖層之 ShapeFile 檔案，以模糊查詢尋找相關資料，最後則將這些資訊呈現於 GIS 系統畫面上，表 4-4 為空間條件與查詢資料的對應情形。

表 4-4 空間條件與查詢資料的對應情形

	單一空間條件	多空間條件
無對應查詢對象	情形 A：直接展示空間條件之位置分佈	先判斷是指向單一空間或多空間，再展示位置分佈
單一對應查詢對象	情形 B：唯一解，展示空間條件之對應查詢對象	先判斷是指向單一空間或多空間，再展示對應之查詢對象
多對應查詢對象	情形 C：展示空間條件的多個對應查詢對象	先判斷是指向單一空間或多空間，再展示多個對應查詢對象

文件斷詞程式如前述介紹可將文章中的句子進行詞性標註，得到詞性後與預設之各項詞庫進行比對，比對到的項目即為具有 GIS 可提供之相對資料，在文件斷詞模組中將這些詞存入文字檔案，回到查詢定位圖台就可讀取這些關鍵詞彙進行查詢；文句經過斷詞後會產生「詞彙庫比對結果.txt」與「語意推論結果.txt」兩個文字檔，「詞彙庫比對結果.txt」中所記錄的資料為與詞彙庫相符的比對結果，其型態可分為空間詞和屬性詞；「語意推論結果.txt」中所記錄的資料則是經由語意推論模組運作後得到的新個體，同樣分為空間詞和屬性詞兩大類，空間詞為句子中的空間查詢條件，可提供使用者在多空間條件中進行選擇；屬性詞則是句子中提及之屬性條件，轉換為對應之 GIS 圖層檔名，可選擇圖層的開啟及刪除，另外由於有疊圖功能，所以可以一次加入多個圖層開啟瀏覽，也可依需求刪除不想瀏覽的圖層；圖層開啟之後主要功能為 GIS 基本操作功能，如放大、縮小、平移，藉由資料瀏覽可引發更多資料需求。

二、系統範例

接著以「紗帽山是一個圓錐形火山丘，大部分由破碎的熔岩構成，頂部有二個小峰，狀似烏紗帽，它是七星山的寄生火山。」進行驗證。此例句經由文件斷詞模組與語意推論模組處理後，產生的「詞彙庫比對結果.txt」文字檔中將「紗帽

山、七星山」視為空間名詞，此例無屬性名詞詞彙，而將「火、熔岩、火山」由語意推論引擎運算後，得到「熔岩流、火山地質」兩個新的個體。

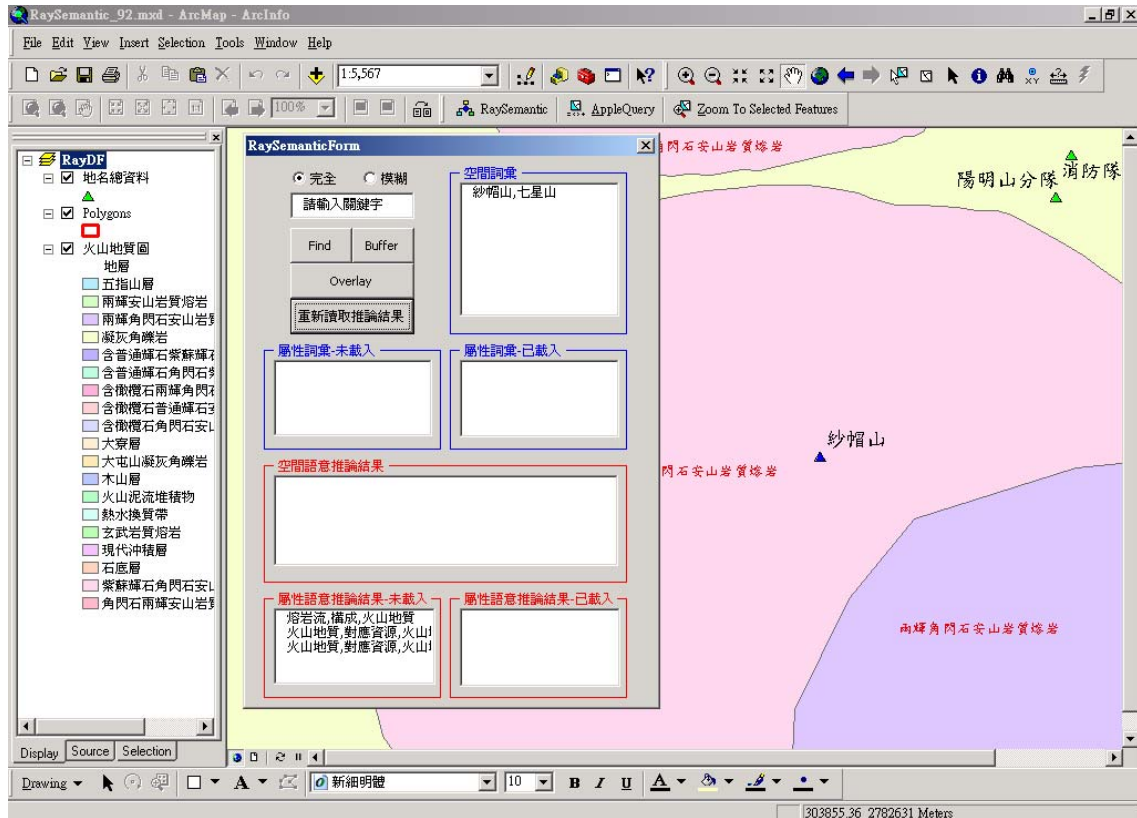


圖 4-11 查詢定位圖台界面

使用者點選「空間詞彙」後，可按下「Zoom To Selected Features」進行定位，而選擇屬性詞彙則可將對應之 GIS 圖層載入或刪除，達到定位及載入圖層目的。

第五節 查詢情境設計

本研究將以情境模擬的方式進行程式的運作驗證，各情境說明及種類彙整如圖 4-12 所示，並分述如下：



圖 4-12 情境分類圖

情境一：無推論之關鍵字比對

由使用者閱讀的文章中抽離「竹子湖」詞彙，經比對後資料庫中具有對應資源，直接輸出結果到查詢定位圖台，不需進行語意推論，查詢流程如圖 4-13：

步驟 1：使用者閱讀某文件時，讀到「竹子湖地區遠在五十萬年前曾為堰塞湖」後引發需求，該句作為文件詞系統的來源參數。

步驟 2：斷詞系統將「竹子湖地區遠在五十萬年前曾為堰塞湖」句子送至中研院詞性標註系統，並回傳詞性標記結果為「竹子湖 (Nc) 地區 (Nc) 遠 (VH) 在 (P) 五十萬 (Neu) 年前 (Nd) 曾 (D) 為 (P) 堰塞 (Nb) 湖 (Na) 」。

步驟 3：完成詞性標記後，將詞彙進行分類，並以名詞作為主要關注對象，此時分類的結果為名詞 3 個 (竹子湖、地區、堰塞)，非名詞 6 個 (遠、在、五十萬、年前、曾、為)。

步驟 4：接著將名詞詞彙與空間/屬性詞彙庫比對，其中「竹子湖」名詞於空間詞彙庫中可找到對應資源，因此將其視為空間詞彙，其餘則為「未具資源名詞」類型詞彙。

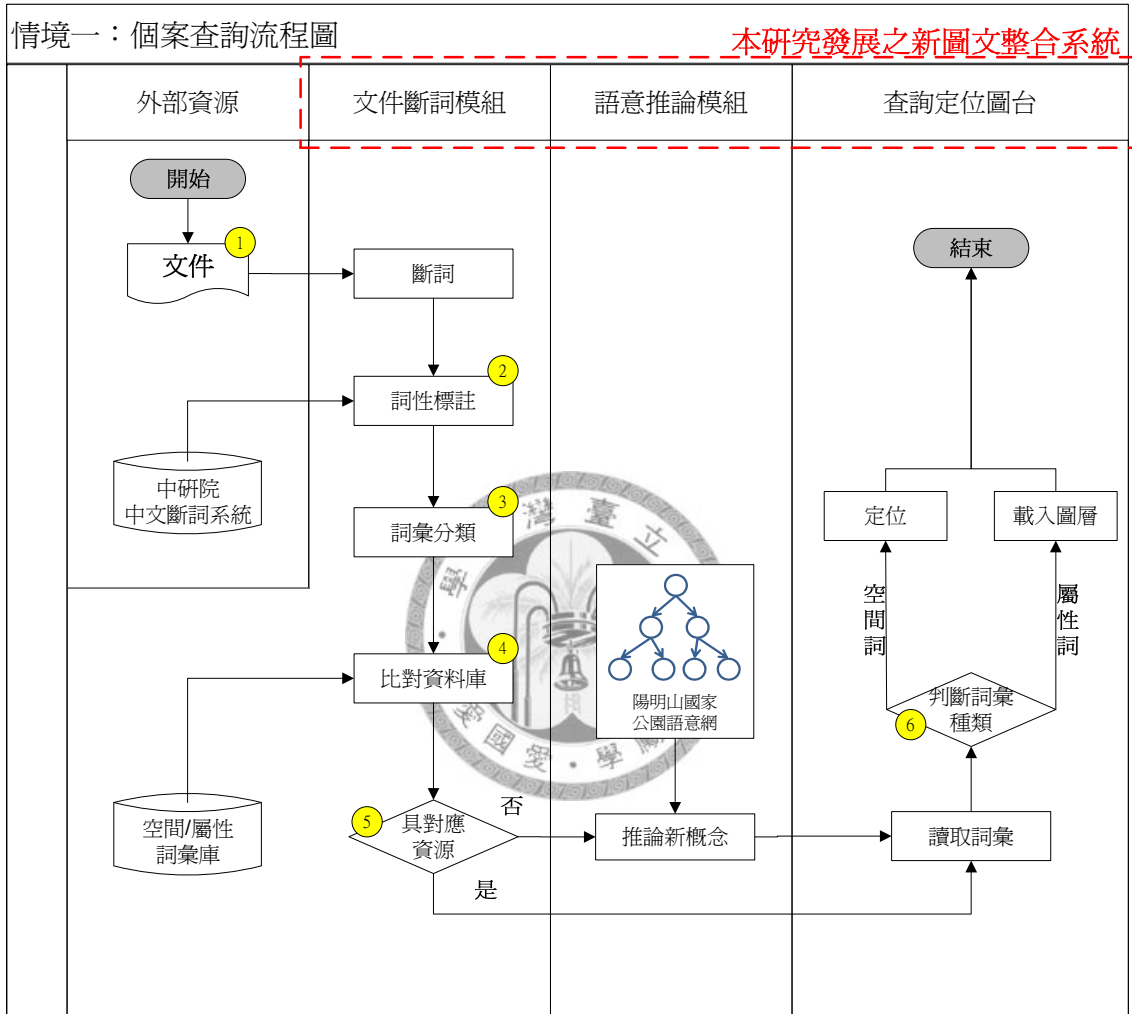


圖 4-13 無推論之查詢流程圖

步驟 5：由使用者決定是否需繼續推論，由於目前的「竹子湖」空間詞彙已足夠圖台定位使用，因此不推論新的概念。

步驟 6：將空間詞彙送至查詢定位圖台，定位至該點後結束整個查詢流程。

情境二：透過空間外部語意關聯的推論

藉由使用者閱讀的文章中抽離「湖田國小」詞彙，現有詞彙庫中無法找到對應資源，經由推論引擎獲得「竹子湖」此一具對應資源的詞彙，有效解決空間尺度上的語意落差，查詢流程如圖 4-14：

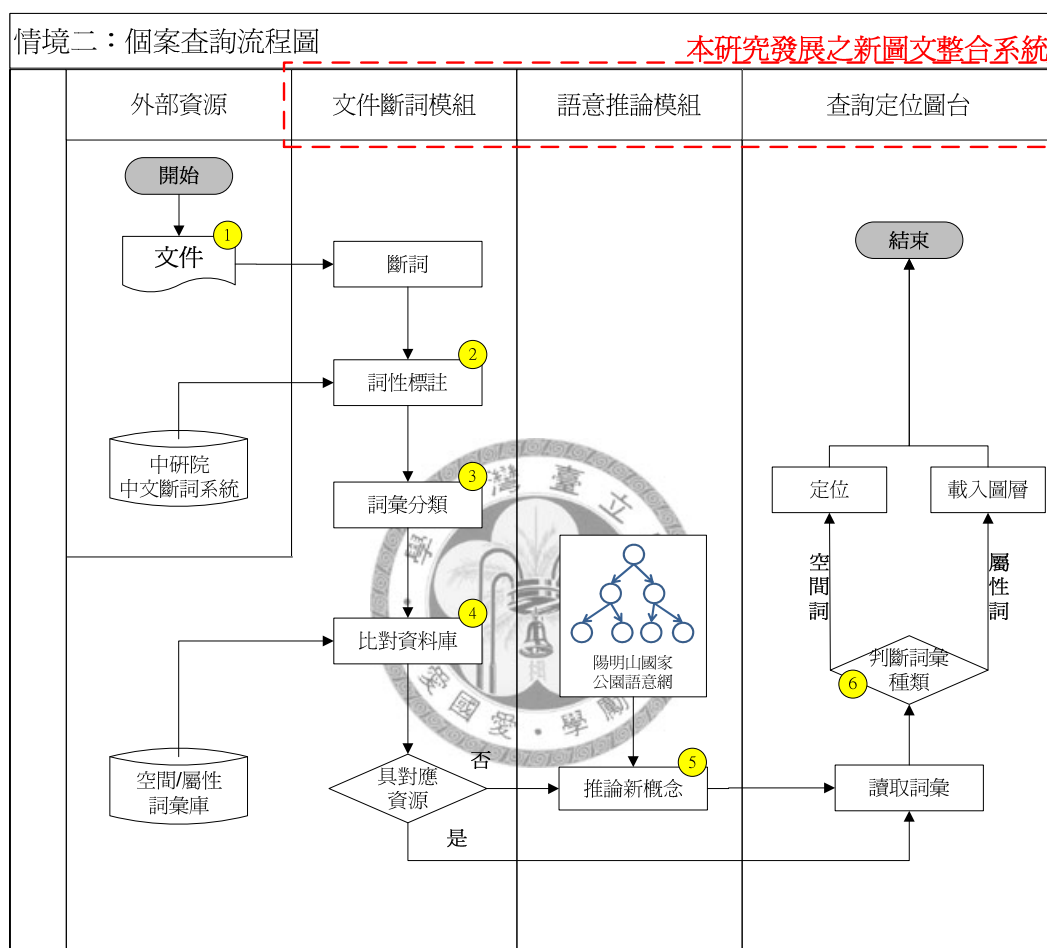


圖 4-14 有推論之查詢流程圖

步驟 1：使用者閱讀某文件時，讀到「湖田國小座落於此窪地湖底窪地」後引發需求，該句作為文件詞系統的來源參數。

步驟 2：斷詞系統將「湖田國小座落於此窪地湖底窪地」句子送至中研院詞性標註系統，並回傳詞性標記結果為「湖田 (N) 國小 (N) 座落 (Vt) 於 (P) 此 (DET) 窪地 (N) 湖 (N) 底 (N) 窪地 (N)」。

步驟 3：完成詞性標記後，將詞彙進行分類，並以名詞作為主要關注對象，此時分類的結果為名詞 3 個（湖田國小、湖底、窪地），非名詞 3 個（座落、於、此）。

步驟 4：接著將名詞詞彙與空間/屬性詞彙庫比對，上述名詞皆無法於詞彙庫中找到對應資源，因此將其作為語意推論引擎的來源參數。

步驟 5：在啟動語意推論引擎後，藉由語意網來找尋新概念，此時原始概念「湖田國小」可經由「湖田國小·相鄰於·竹子湖」的關係，得到新概念「竹子湖」，並回傳推論路徑表給使用者瀏覽。

步驟 6：將新概念「竹子湖」送至查詢定位圖台後，判斷其為具資源的空間詞彙後，定位至該點後結束整個查詢流程。

情境三：透過空間內部語意關聯的推論

本推論情境由使用者閱讀的文章中抽離「紗帽山」詞彙，現有詞彙庫中可直接找到對應資源，因此將沒有機會啟動推論引擎，運作流程如下：

步驟 1：使用者閱讀某文件時，讀到「紗帽山是一個圓錐狀火山」後引發需求，該句作為文件詞系統的來源參數。

步驟 2：斷詞系統將「紗帽山是一個圓錐狀火山」句子送至中研院詞性標註系統，並回傳詞性標記結果為「紗帽山 (Na) 是 (SHI) 一 (Neu) 個 (Nf) 圓錐狀 (Na) 火山 (Na) 」。

步驟 3：完成詞性標記後，將詞彙進行分類，並以名詞作為主要關注對象，此時分類的結果為名詞 3 個（火山、紗帽山、圓錐狀），非名詞 7 個（是、一、個）。

步驟 4：接著將名詞詞彙與空間/屬性詞彙庫比對，其中「紗帽山」名詞可於詞彙庫中找到對應資源，因此直接將「紗帽山」詞彙送至查詢定位圖台後，判斷其為具資源的空間詞彙，定位至該位置後完成查詢行為。

情境四：透過屬性外部語意關聯的推論

藉由使用者閱讀的文章中抽離「熔岩」詞彙，現有詞彙庫中無法找到對應資源，經由推論引擎獲得「火山地質」此一具對應資源的詞彙，有效解決屬性間的語意落差，查詢流程如圖 4-14：

步驟 1：使用者閱讀某文件時，讀到「大部分由破碎的熔岩構成」後引發需求，該句作為文件詞系統的來源參數。

步驟 2：斷詞系統將「大部分由破碎的熔岩構成」句子送至中研院詞性標註系統，並回傳詞性標記結果為「大部分 (Neqa) 由 (P) 破碎 (VH) 的 (DE) 熔岩 (Na) 構成 (VG) 」。

步驟 3：完成詞性標記後，將詞彙進行分類，並以名詞作為主要關注對象，此時分類的結果為名詞 1 個 (熔岩)，非名詞 4 個 (大部分、由、破碎、的)。

步驟 4：接著將名詞詞彙與空間/屬性詞彙庫比對，上述名詞皆無法於詞彙庫中找到對應資源，因此將其作為語意推論引擎的來源參數。

步驟 5：在啟動語意推論引擎後，藉由語意網來找尋新概念，此時原始概念「熔岩」可經由「熔岩·構成·火山地質」的關係，得到新概念「火山地質」，並回傳推論路徑表給使用者瀏覽。

步驟 6：將新概念「火山地質」送至查詢定位圖台後，判斷其為具資源的屬性詞彙後，將圖層「火山地質圖」載入專案後結束整個查詢流程。

情境五：透過屬性內部語意關聯的推論

藉由使用者閱讀的文章中抽離「木山層」詞彙，現有詞彙庫中無法找到對應資源，經由推論引擎獲得「地層分佈圖」此一具對應資源的詞彙，有效解決空間尺度上的語意落差，查詢流程如圖 4-14：

步驟 1：使用者閱讀某文件時，讀到「木山層是顏滄波和陳培源提出來的一個地層名稱」後引發需求，該句作為文件詞系統的來源參數。

步驟 2：斷詞系統將「木山層是顏滄波和陳培源提出來的一個地層名稱...」句子送至中研院詞性標註系統，並回傳詞性標記結果為「木山層 (Na) 是 (SHI) 顏滄波 (Nb) 和 (Caa) 陳培源 (Nb) 提出來 (VC) 的 (DE) 一 (Neu) 個 (Nf) 地層 (Na) 名稱 (Na)」。

步驟 3：完成詞性標記後，將詞彙進行分類，並以名詞作為主要關注對象，此時分類的結果為名詞 4 個（木山層、地層、顏滄波、陳培源），非名詞 6 個（是、和、提出來、的、一、個）。

步驟 4：接著將名詞詞彙與空間/屬性詞彙庫比對，上述名詞皆無法於詞彙庫中找到對應資源，因此將其作為語意推論引擎的來源參數。

步驟 5：在啟動語意推論引擎後，藉由語意網來找尋新概念，此時原始概念「木山層」可經由語意深度為 1 的「木山層·欄位為·地層欄位」關係，及語意深度為 2 的「地層欄位·資料庫為·地層分佈圖」關係，得到新概念「地層分佈圖」，並回傳推論路徑表給使用者瀏覽。

步驟 6：將新概念「地層分佈圖」送至查詢定位圖台後，判斷其為具資源的屬性詞彙，載入相對應的圖層後結束整個查詢流程。

第五章 結論與未來發展

第一節 結論

本研究為一應用語意網知識庫於圖文整合面向之研究，首先回顧相關理論，包括自然語言的拆解方式、建構知識本體的技術、語意網知識表現特色及應用時需注意的地方，接著再以各種關係建立本研究自行發展的語意網及推論引擎，應用於文件與 GIS 的串聯流程之中。研究動機主要可分為兩點：首先利用語意網來解決人類自然語言與電腦關鍵字查詢間的語意落差，及在現行透過關鍵字串聯文件與 GIS 的架構中，加入語意推論引擎來增加搜尋的彈性，使得查詢的行為獲得更好的支援。最後，再針對推論引擎、語意網的建立、詞性標註、簡單關係的定義及有限資源的最大化這四個面向歸納出研究的結論如下，來回應本研究的目的與目標：

一、推論引擎

本研究提出詳細且可實際應用的語意網搜尋引擎，可改善人類查詢語句與搜尋引擎間的語意落差問題，藉由語意網完成了推論人類查詢語意的基本架構，嘗試改變以往的閱讀方式，以新圖文整合的概念將文件與 GIS 串聯，並在原有的關鍵字搜尋機制上加以擴充，建立「語意搜尋式的新圖文整合」架構，對文件與 GIS 間的整合提供一個新的架構。

二、語意網的建立

本研究首先分析陽明山國家公園相關文件中所隱含的語意結構，並依其建立「陽明山語意網」的內部關聯架構，接著以彼此具備高相關性的各種概念建構出「陽明山語意網」的外部關聯架構，再以資料庫中可供使用的資源建立相關的索引詞彙形成空間和屬性詞庫，並提供文件斷詞模組使用，最後以 GIS 軟體呈現文件中與語意相對應的資源，滿足使用者的查詢需求。但在語意網的建置過程中係

採人工方式，因此知識工作者的主觀意識為影響語意網結果的關鍵的因子，因此如何將建置知識本體方法的標準化、模組化即變得非常的重要。

三、詞性標註

中研院目前已提供成熟之詞性標註模組可以使用，本研究應用其研究成果作為標註文章詞性用途，並自行開發詞彙粹取及比對模組找出重要的關鍵詞彙應用於後續查詢行為中，提出建立語意網的半自動化方式，藉由實做語意網的過程歸納各種結晶核、關聯建立的可能規則，以供未來建立全自動化的研究題材，期許未來能建立「語意結構庫」，輔助語意網的建立行為。

四、簡單關係的定義

本研究僅針對空間「相鄰」、「包含」、「被包含」及屬性「構成」、「上方地層為」等關係進行實作，是一簡單關聯的語意網建立方式，若能基於本研究提出的建議來產生所需語意網將可使推論引擎支援更多的關係。

五、有限資源的最大化

因為人類的資源有限，如何將現有資源效益最大化即變得非常的重要，本研究嘗試將多個詞彙與特定資源建立關係，形成「多個詞彙可對應到同一資源」的架構，在研究完成後，可有效輔助使用者尋找與閱讀主題相關的資源，不論是對於使用者的資源輔助或資源的重覆利用，均具有實質上的效益提升。

第二節 未來發展

本研究主要希望能夠解決使用者在查詢上所遇到的問題：即當資料庫中無對應的關鍵詞時，現有的關鍵字比對引擎將無法找到對應資源；藉由結合語意網的技術來達到語意的查詢，希望踏出改變傳統關鍵字查詢模式的第一步，但其中仍有些構想及功能尚待後續研究改進，在此提出幾個議題進行探討：

一、文本限制

本研究的文句歸納文本主要是陽明山研究報告「陽明山國家公園地形及地

質景觀」，但是每個寫作者慣用的語言及文法因人而異，所形成的規則可能也不同，所以本研究所建構出的規則若以其他文本輸入進行使用時，可能會有些許出入，此點仍尚待改進。但本研究所建立之文句規則仍以符合一般大多數情形為主，希望將文本差異程度降到最低。透過此研究及系統，能讓使用者在面對文字資料時，透過文字資料與GIS 資料中相關者的抽離連結，進而獲得更多資訊，並增加GIS的使用及普及。

二、自動化文件語意網粹取

本研究使用人工方式由陽明山研究報告將文件語意網進行抽離，期未來能藉由語法歸納、對應架構的經驗累積，歸納出一套自動化文件語意網粹取的理論，以達到文件輸入、語意網自動輸出的目的。此外，自動化由文件產生語意網的機制，可能會在多個不同的文件間產生語意網的矛盾與衝突，未來是否需由人工或專家進行確認，則留到後續研究持續探討。

三、引用現有的語意網

Natalya F.Noy (2001) 提出的「語意網發展流程」中，第二階段為考慮重複使用，即是希望知識工作者在建立語意網時，能夠使用現有的一般普及性語意網來做為發展的基礎，一方面除了將多個語意網串聯起來成為一個大型的語意網，另一方面更將具經驗的知識工作者分類方式，直接引用於自己的語意網中，如此不僅建置語意網的時間縮短了，更可建置出符合標準化基礎的領域語意網。

四、語意網來源的選擇

若能進一步將研究拓展於網路系統，即可結合網路上更為龐大之資訊量，且不僅止於文字資訊，包括各式多媒體如圖像、聲音、影片，皆可以 metadata 的文字資訊與GIS 串連，且達到相互連結的功用，例如將旅遊相關各式資訊與GIS 連結，就數位書籍而言可成為電子書的另一模式，另一方面亦可將虛擬旅遊更佳深度化，當使用者在瀏覽某地區的GIS 資料時可同時查詢究架構同地區相關圖片、

文字資訊或是網路上景點的即時影像，可隨時尋找相關之文字與多媒體資訊，滿足更多層面之需求。本研究資料來源為陽明山知識文件，建議未來可以Web Service的Metadata為來源 (參考圖5-1)，增加資料來源的豐富性。

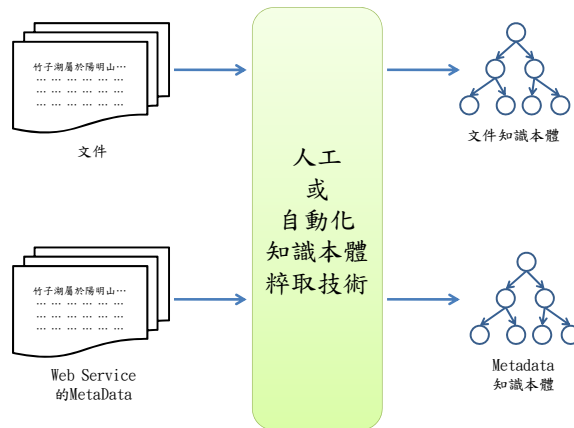


圖 5-1 語意網的來源選擇

五、同義字的誤判

在文字與 GIS 的串連方法，本研究採取的是文字的直接對照查詢，優點在於查詢結果較為精確，但相對而言在使用者可能產生錯誤的時候無法判斷，例如輸入的是同義字，就無法得到相關資訊，若能加以模糊查詢的功能則可擴大查詢範圍，但相對而言必須負擔查詢不夠準確之風險。

六、單向串連

本研究主要關注的面向在於文字向GIS 資料的單向串連，就功能性而言較為薄弱，只能從文字部分去尋找相關的GIS 資料，若能將GIS資料向文字的串連也加入的話，系統會更加完善，使用者亦能在瀏覽GIS 資料的過程當中，隨時查詢相關文字資訊，文字與GIS 兩者結合將更為緊密，並且符合使用者之需求，功能也更加強大。

參考文獻

中研院詞庫小組 (1993) 中文詞類分析，中文詞知識庫小組技術報告，南港，

<http://godel.iis.sinica.edu.tw/CKIP/>, [3, April, 2009]。

王治立 (2004) 旅遊語意網整體服務系統之建置，大葉大學資訊管理學系碩士論文。

王詠倩 (1997) 超媒體輔助 GIS 使用環境之研究—以語意網為系統整合中介，國立臺灣大學地理學研究所碩士論文。

王鑫 (1986) 陽明山國家公園地質及地形景觀，內政部營建署，台北。

李介中 (2004) 碎形理論應用於台灣地區建地空間型態之研究，國立臺灣大學地理環境資源學研究所碩士論文。

李文元 (2007) 語意格網為基礎之普適學習環境，中原大學資訊管理研究所碩士論文。

沈嘉慧 (2008) 應用文字挖掘技術以整合文件與 GIS — 以陽明山國家公園研究報告為例，國立臺灣大學地理環境資源學研究所碩士論文。

陳建軍、周成虎、王敬貴 (2006) 地理本體的研究進展與分析，地學前緣，13 (3)：81-90。

陳靖允 (2004) 以語意網建置旅遊解說資料庫—以陽明山的自然資源為例，國立臺灣大學地理環境資源研究所碩士論文。

黃如鈺 (2005) 自然語言式 GIS 查詢介面—以大安區餐飲服務為例，國立臺灣大學地理環境資源研究所碩士論文。

黃燕萍 (1999) 中文社會新聞文件資訊擷取，雲林科技大學資訊管理研究所碩士論文。

蔡純純 (2003) 中文新聞文件空間資訊擷取之研究—以火災、搶劫、車禍事件為例，國立臺灣大學地理環境資源所碩士論文。

鄭伊善 (2007) 一套將關聯式資料庫對映至知識本體的方法，臺灣大學資訊管理學研究所碩士論文。

謝奇峰 (1994) 超媒體地理資訊查詢展示系統之建立—以玉山國家公園為例，國立臺灣大學地理學研究所碩士論文。

簡月秀 (2003) 以知識本體為基礎的壽險客服應用，國立政治大學經營管理碩士論文。

Antoniou, G. and van Harmelen, F. (2006) 語意網技術導論，屠名正譯，台北市：碁峰資訊。



Berners-Lee, T., Hendler, J. and Lassila, O. (2001) The Semantic Web, *Scientific America*, <http://www.sciam.com/> [3, April, 2009].

ESRI (2004) What is ArcGIS, http://www.esri.com/library/whitepapers/pdfs/what_is_arcgis.pdf, [19, March, 2009].

Guha, R., McCool, R. and Miller, E. (2003) Semantic Search, *WWW2003*, May 20-24 <http://www2003.org/cdrom/papers/refereed/p779/ess.html>, [5, March, 2009].

Gruber, T. (2008) Ontology, <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>, [5, March, 2009].

- Hildebrand, M., van Ossenbruggen, J.R., Hardman, L. (2007) An analysis of search-based user interaction on the Semantic Web, *INformation Systems*, Report INS-E0706 May 2007.
- Jupp, S., Moulton, G., Rector, A., Stevens, R., Wroe, C. (2007) A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools Edition 1.1, <http://www.co-ode.org>, [2, January, 2008]
- Knublauch, H. (2009) Protégé-owl api programmer's guide, <http://protege.stanford.edu/plugins/owl/api/guide.html>, [20, June, 2008].
- McCarthy, J. D., Graniero, P. A. and Rozic, S. M. (2008) An Integrated GIS-Expert System Framework for Live Hazard Monitoring and Detection, *Sensors*, 2008(8): 830-846.
- Noy, N. F. and McGuinness, D. L. (2000) Ontology Development 101: A Guide to Creating Your First Ontology, <http://protege.stanford.edu/>, [11, September, 2007]
- OWLviz (2004) OWLviz-Guide, <http://www.co-ode.org/downloads/owlviz/OWLviz-Guide.pdf>, [5, July, 2008].
- Schevers, H.A.J. and Drogemuller, R.M. (2006) Semantic Web for an Integrated Urban Software System, <http://www.mssanz.org.au/modsim05/papers/schevers.pdf>, [15, June, 2008]
- Stanford Medical Informatics (2005) Protégé, <http://protege.stanford.edu>, [2, January, 2008].
- Sun (2009) Java, <http://www.java.com/en/about/>, [10, December, 2008].
- Sun, H., Li, S., Li, W., Ming, Z. and Cai S. (2005) Semantic-Based Retrieval of Remote Sensing Images in a Grid Environment, *IEEE GEOSCIENCE AND REMOTE SENSING LETTERS*, 2(4): 440-444.
- World Wide Web Consortium (2005) RDF, <http://www.w3.org/RDF/>, [4, February, 2008].

World Wide Web Consortium (2005) OWL, <http://www.w3.org/OWL/>, [21, January, 2007].

Wikipedia (2009) Semantic Gap, http://en.wikipedia.org/wiki/Semantic_gap, [9, March, 2007].

Wikipedia (2009) ArcGIS, <http://en.wikipedia.org/wiki/ArcGIS>, [16, March, 2009].



附錄一、中研院平衡語料庫詞類標記集

簡化標記	對應的 CKIP 詞類標記	
A	A	/*非謂形容詞*/
Caa	Caa	/*對等連接詞，如：和、跟*/
Cab	Cab	/*連接詞，如：等等*/
Cba	Cbab	/*連接詞，如：的話*/
Cbb	Cbaa, Cbba, Cbbb, Cbca, Cbcb	/*關聯連接詞*/
Da	Daa	/*數量副詞*/
Dfa	Dfa	/*動詞前程度副詞*/
Dfb	Dfb	/*動詞後程度副詞*/
Di	Di	/*時態標記*/
Dk	Dk	/*句副詞*/
D	Dab, Dbaa, Dbab, Dbb, Dbc, Dc, Dd, Dg, Dh, Dj	/*副詞*/
Na	Naa, Nab, Nac, Nad, Naea, Naeb	/*普通名詞*/
Nb	Nba, Nbc	/*專有名稱*/
Nc	Nca, Ncb, Ncc, Nce	/*地方詞*/
Ncd	Ncda, Ncdb	/*位置詞*/
Nd	Ndaa, Ndab, Ndc, Ndd	/*時間詞*/
Neu	Neu	/*數詞定詞*/
Nes	Nes	/*特指定詞*/
Nep	Nep	/*指代定詞*/
Neqa	Neqa	/*數量定詞*/
Neqb	Neqb	/*後置數量定詞*/
Nf	Nfa, Nfb, Nfc, Nfd, Nfe, Nfg, Nfh, Nfi	/*量詞*/
Ng	Ng	/*後置詞*/
Nh	Nhaa, Nhab, Nhac, Nhb, Nhc	/*代名詞*/
I	I	/*感嘆詞*/

附錄一、中研院平衡語料庫詞類標記集 (續)

簡化標記	對應的 CKIP 詞類標記	
P	P*	/*介詞*/
T	Ta, Tb, Tc, Td	/*語助詞*/
VA	VA11,12,13,VA3,VA4	/*動作不及物動詞*/
VAC	VA2	/*動作使動動詞*/
VB	VB11,12,VB2	/*動作類及物動詞*/
VC	VC2, VC31,32,33	/*動作及物動詞*/
VCL	VC1	/*動作接地方賓語動詞*/
VD	VD1, VD2	/*雙賓動詞*/
VE	VE11, VE12, VE2	/*動作句賓動詞*/
VF	VF1, VF2	/*動作謂賓動詞*/
VG	VG1, VG2	/*分類動詞*/
VH	VH11,12,13,14,15,17,VH21	/*狀態不及物動詞*/
VHC	VH16, VH22	/*狀態使動動詞*/
VI	VI1,2,3	/*狀態類及物動詞*/
VJ	VJ1,2,3	/*狀態及物動詞*/
VK	VK1,2	/*狀態句賓動詞*/
VL	VL1,2,3,4	/*狀態謂賓動詞*/
V_2	V_2	/*有*/
DE	/*的, 之, 得, 地*/	
SHI	/*是*/	
FW	/*外文標記*/	



附錄二、中文詞類分析總表

一、述詞，是謂語中心。（依動作/狀態、及物性、論元個數以及述詞後接成分的詞組形式分為十二大類）

VA 動作不及物述詞，這類述詞只需要一個名詞組參與論元即可。（依論旨角色、語意特性的不同分為四類）

VA1 表移動或存在的述詞，論旨角色為客體（theme）。（依語意及內部結構的不同分為三類）

VA11 表移動或行動的述詞，可後接地方成分，有主語倒置的現象。例：跑、飛、走。

VA12 表存在或靜態的述詞，可後接地方成分，有主語倒置的現象。例：坐、躺。

VA13 內部結構為述賓結構且賓語表地方成分的行動述詞。例：逛街、上臺、出場。

VA2 作格述詞，論旨角色為客體（theme），述詞前可有一個肇始者（causer），原來出現在述詞前的客體移到賓語的位置。例：出動、轉。

VA3 氣象述詞，論旨角色為客體（theme）。

例：下雨、颱風、打雷。

VA4 一般的動態述詞，論旨角色為主事者（agent）。

例：違規、謀生、開會。

VB 動作類單賓述詞，語意上需要兩個參與論元，但它的賓語不能直接出現在述詞後，而以介詞引介或賓語前提的方式出現。（依論旨角色的不同分為兩類）

VB1 賓語為動作施行的對象，其角色為終點（goal）。（依句型的不同分為兩類）

VB11 終點一定要以介詞引介出現在述詞前或後。

例：求婚、拜年。

VB12 終點可以是名詞組出現在主語位置。

例：立案、整容、解體。

VB2 賓語的角色為客體 (theme)。例：充公、除名、送醫。

VC 動作單賓述詞，語意上需要兩個參與論元。(依論旨角色的不同分為三類)

VC1 表移動或存在的述詞，主語為客體 (theme)，賓語為表地方的終點 (goal)。例：進、闖入、經過、逃離、住、世居。

VC2 以主事者 (agent) 為主語，終點 (goal) 為賓語。

例：打、學、訪問、使用、破壞、照顧。

VC3 以主事者 (agent) 為主語，客體 (theme) 為賓語。(依句型的不同分為三類)

VC31 述詞後除了賓語外不需再接一個地方成分。

例：買、賺、吃、生產、組織、收取、洩露。

VC32 述詞後除了賓語外，通常還接一個由介詞「到」引介的地方詞。

例：走私、引渡、調遣、押送、發射、搭載。

VC33 述詞後除了賓語外，通常還接一個由介詞「在」或「到」引介的地方詞，而且有地方詞倒置的現象。例：放、埋、懸掛、儲存、搭建、囚禁。

VD 雙賓述詞，這類述詞在語意上有傳遞事物的動作訊息，需要三個參與論元。(依間接賓語的論旨角色的不同分為兩類)



VD1 表將一事物傳遞給對方的述詞，主事者 (agent) 具有起點特徵 (+source)，間接賓語是終點 (goal)。

例：寄、送、捐。

VD2 表向對方取得一事物，主事者 (agent) 具有終點的特徵 (+goal)，間接賓語是起點 (source)。

例：搶、敲詐、索取。

VE 動作句賓述詞，後接句賓語的動作及物述詞。(依論元個數的不同分為兩類)

VE1 三元述詞。(依語意上的不同分為「問類」及「說類」兩類)

VE11 問類，以主事者 (agent) 為主語，以終點 (goal) 為間接賓語，客體 (theme) 為直接賓語 (句賓語)，句賓語為疑問句式，且疑問範圍只到包接句。例：責問、詢問。

VE12 說類，和 VE11 的論旨角色相同，不同的是：

VE12 的句賓語不限於疑問句。

句賓語的疑問範圍不限於包接句。

主語或間接賓語與句賓語之主語間可有共指關係。

例：提示、許諾、指引。

VE2 二元述詞，以主事者 (agent) 為主語，終點 (goal) 為句賓語，語意多為表語言行為之述詞。

例：悲歎、自誇、下令、研究、討論、探索、反省、強調、猜想、說、提到。

VF 動作謂賓述詞，後接述詞組賓語的動作及物述詞。(依論元個數的不同分為兩類)

VF1 二元述詞，以主事者 (agent) 為主語，終點 (goal) 為賓語，語意多含有「打算」之意。例：企圖、想、打算。

VF2 三元述詞，以主事者 (agent) 為主語，後帶一個終點 (goal) 的名詞組賓語，再帶一個表客體 (theme) 的述詞組賓語。其中這個名詞不但是主要述詞的賓語，也是述詞組賓語的主語，是一般所謂的「兼語式」述詞，此類述詞語意多表「鼓勵」、「命令」、「強迫」、「請求」。例：任用、勸。

VG 分類述詞，擔任主語和補語間連結的角色。(依論元個數的不同分為兩類)

VG1 三元述詞，這類述詞帶有主事者 (agent)、客體 (theme) 和範圍 (range) 三個論元。例：稱呼、喊、命名。

VG2 二元述詞，典型的分類述詞，帶客體 (theme) 和範圍 (range) 兩個論元。例：姓、當。

VH 狀態不及物述詞，用以描述事物所呈現的某種狀態，這類述詞只需要一個參與論元即可。(依論旨角色的不同分為兩大類)

VH1 論旨角色為客體 (theme)。(依句型的不同分為七類)

VH11 一般的不及物述詞。例：動聽、浪漫、特別。

VH12 能夠後接定量詞表示量度的述詞。

例：入超、增值、淨重。

VH13 能夠後接比較對象及兩者差額的述詞。

例：大、高、慢。

VH14 可以後接地方成分，有地方詞倒置句型。

例：瀟灑、矗立。

VH15 可以以句子作為主語，且可將句子移至述詞後。例：值得、夠、適合。

VH16 作格述詞，述詞前可有一個肇始者 (causer)，原來述詞前的客體 (theme) 移到一般賓語的位置。例：辛苦、豐富、穩固。

VH17 述詞前可有一個接受者 (recipient)，是客體 (theme) 的擁有者。
例：丟、瞎、斷。

VH2 論旨角色為經驗者 (experiencer)。(依句型的不同分為兩類)

VH21 非作格述詞。例：心酸、想不開。

VH22 作格述詞，述詞前可有一個肇始者 (causer)，原來述詞前的經驗者 (experiencer) 移到賓語的位置。例：震驚、為難、急煞、感動。

VI 狀態類單賓述詞，語意上需要兩個參與論元，但它的賓語不能直接出現在述詞後，而以介詞引介或賓語提前的方式出現。(依論旨角色的不同分為三類)

VI1 以經驗者 (experiencer) 為主語，終點 (goal) 為賓語，表心靈狀態。
例：心動、灰心、傾心。

VI2 以客體 (theme) 為主語，以終點 (goal) 為賓語。

例：內行、不利、為例。

VI3 以客體 (theme) 為主語，以起點 (source) 為賓語。

例：受教、取材、取決。

VJ 狀態單賓述詞，這類述詞在語意上需要兩個參與論元。(依論旨角色不同分為三類)

VJ1 以客體 (theme) 為主語，以終點 (goal) 為賓語。

例：迎合、代表。

VJ2 以經驗者 (experiencer) 為主語，終點 (goal) 為賓語，表心靈狀態。
例：景仰、惦念、嫌忌。

VJ3 以客體 (theme) 為主語，以範圍 (range) 為賓語。



例：長達、剩餘。

VK 狀態句賓述詞，後接句賓語的狀態及物述詞。（依照主語論旨角色的不同分為兩類）

VK1 以經驗者（experiencer）為主語，以終點（goal）為賓語，表心靈狀態。例：不滿、嫌惡。

VK2 以客體（theme）為主語，以終點（goal）為賓語。

例：反應、在於、干係。

VL 狀態謂賓述詞，後接述詞組的狀態及物述詞。（依照主語論旨角色或論元個數的不同分為四類）

VL1 以經驗者（experiencer）為主語，終點（goal）為賓語的二元述詞，表心靈狀態而其語意多表「意願」。

例：樂於、甘願。

VL2 以客體（theme）為主語，終點（goal）為賓語的二元述詞。其語意多表「專門」之意。

例：擅長、專門、擅於。

VL3 不帶主語的二元述詞，後接一個終點（goal）和一個表客體（theme）的述詞組論元，例：輪、該。其中表客體的述詞組中賓語部分常常會移到輪、該等主要述詞前面的位置。

VL4 使役述詞，帶肇始者（causer）、終點（goal）、客體角色（theme）的三元述詞。例：使、讓。

二、體詞（N），體詞通常出現在主語或賓語的位置。（依其語意、作用分八類）

Na 名詞（下分五類）

Naa 物質名詞，是不可數的實體名詞。



例：泥土、鹽、水、牛肉。

Nab 個體名詞，是可數的實體名詞，可受個體量詞修飾。

例：桌子、杯子、衣服、刀。

Nac 可數抽象名詞，是可數的非實體名詞。

例：夢、符號、話、原因。

Nad 抽象名詞，是不可數的非實體名詞。

例：風度、香氣、愛心、馬後砲。

Nae 集合名詞：這類名詞不能指涉個體，只能指涉複數，且不可以受個體量詞修飾，又依是否受定量式複合詞修飾分二類。

Naea 不能加任何定量式詞組來修飾的集合名詞。

例：三餐、五臟六腑、四肢。

Naeb 可用定量式詞組來修飾。

例：車輛、船隻、夫妻。



Nb 專有名稱（下分兩類）

Nba 正式專有名稱，包含時間、地方以外的專有名稱。

例：吳大猷、余光中、詩經、雙魚座。

Nbc 姓氏。例：張、王、李。

Nc 地方名詞（下分五類）

Nca 專有地方名詞，特指某一地方、行政單位或機構，通常不能用定量式複合詞來修飾。例：西班牙、台北。

Ncb 普通地方名詞，可以用定量式詞組來修飾。

例：郵局、市場、學校、農村。

Ncc 名方式地方名詞。例：海外、身上、腳下。

Ncd 表事物相對位置的地方詞，大部分由獨用的方位詞或方方式或定方式複合詞構成（下分二小類）。

Ncda 單音節位置詞，其後不能加"的"。

例："上"有天堂。

Ncdb 雙音節位置詞。例：上頭、中間、左方、西北。

Nce 定名式地方名詞，例：四海、當地。

Nd 時間名詞（下分三類）

Nda 時間名稱（下分兩類）

Ndaa 歷史性的時間名稱（下分四小類）

Ndaaa 特指的時代名稱。例：洪荒時代、五〇年代。

Ndaab 朝代名稱。例：唐朝、西漢。

Ndaac 歷代帝王的年號名稱。

例：乾隆、光緒、天寶。

Ndaad 年份名，用以計數年份的紀元。

例：公元、西元。

Ndab 可循環重複的時間名稱（下分六小類）

Ndaba 年稱。例：今年是"辛未"年。

Ndabb 季節，即春、夏、秋、冬四季。

例：今年"春天"雨水多。

Ndabc 月份名稱。例："十二月"又叫"臘月"。

Ndabd 日期。例：三月"六日"、冬至。



Ndabe 日以內的時間名稱。例：傍晚、大清早。

Ndabf 時期，指一段時間。例：寒假、年假、春節。

Ndc 名方式時間名詞，由附著語位的時間成分加上方位詞複合而成。例：年底、週末、日後。

Ndd 副詞性時間詞（以下分三類）

Ndca 表過去的副詞性時間詞。例：過去、從前、當初。

Ndcb 表將來的副詞性時間詞。例：以後、後來、將來。

Ndcc 表現在及其他的副詞性時間詞。

例：現在、當今、眼前、近來。

Ne 定詞，用以表示物品的指涉或物品的數量。例：這、哪、少許。

Nf 量詞，用以計量的連用語位，常和定詞構成定量式詞組。

Nfa 個體量詞，表示每一個名詞所屬的專門單位詞。

例：一"張"桌子、一"個"杯子、一"件"衣服、一"把"刀子。

Nfb 跟述賓式合用的量詞，放於述詞與賓語之間。

例：下一"盤"棋、寫一"手"好字、說一"口"標準國語。

Nfc 群體量詞，語義上能標示出一組或一群的物體。

例：一"對"夫妻、一"雙"筷子、一"副"耳環、一"群"鴨子。

Nfd 部分量詞，表示事物的部分而非整體的概念。

例：一"部分"原因、一"節"甘蔗、一"段"文章、一"點"事情。

Nfe 容器量詞，用器皿式的名詞來作量詞，表示概括性的容量。例：一"箱"書、一"櫃子"衣服、一"盤"水梨、一"碗"飯。

Nff 暫時量詞，是以名詞作量詞，加在定詞後面。

例：一"肚子"牢騷、一"頭"秀髮、一"鼻子"灰、一"地"落葉。

Nfg 標準量詞，是正規的量詞，為名副其實的量詞。包括：

長度單位。例：尺、寸、丈。

面積單位。例：頃、畝。

重量單位。例：公斤、磅。

容量單位。例：升、斗。

時間單位。例：分、秒、時。

錢幣單位。例：元、法郎、先令。

數量單位。例：刀、令。

能量單位。例：馬力、燭光、卡路里。

Nfh 準量詞，由名詞轉化而來的單位化量詞，是獨立自主的，它不是後頭名詞的量詞。例：國、面、撇。

Nfi 述詞用量詞，是動作述詞的量詞，表示動作發生的次數。

例：看一"遍"、摸一"下"。

Ng 後置詞。它是一個附著成分，前接一個詞組形成時間成分或表情況的成分。例：睡覺"之前"、夜"裡"、三百人"以上"。

Nh 代名詞（下分三類）

Nha 人稱代名詞（下分三小類）

Nhaa 常用的人稱代名詞，是我、你、他及其複數式、同義詞。

Nhab 一般的人稱代名詞，可與第一、二、三人稱同位並列。例：自己。

Nhac 特別的人稱代名詞，有所專指的代名詞。例：您、足下、令尊、本人、賤內、小犬。

Nhb 疑問代名詞，包括誰、什麼及其別體甚麼、啥等。

Nhc 泛指代名詞，可通用於人、物的代名詞。例：之、其。

三、介詞（P），用以引介一個角色，作述詞的修飾成分或必要論元。（依介詞組所可能表示的角色、介詞對其論元之語意及語法限制的不同分為六十五類）

四、副詞（D），主要用作謂語的修飾語。（依語意下分十類）

Da 表範圍和數量的副詞。例：只、僅僅。

Db 表示評價的副詞。（下分三類）

Db_a 法相副詞。例：也許、大概、一定。

Db_{aa} 推測用法。例：也許、大概、可能、一定。

Db_{ab} 義務用法。例：必須、可以、不得。

Db_b 表示說話者的評斷的副詞。例：幸虧、果然。

Db_c 由"-起來"與述詞組成的評價詞。

例：這條路"看起來"很平直。

Dc 表否定的副詞。包括：未、沒有、沒、不。

Dd 時間副詞。例：先、立刻。

Df 程度副詞。（下分兩類）

Df_a 述詞前程度副詞。例：很、非常。

Df_b 述詞後程度副詞。例：得很、之至。

Dg 地方副詞。例：處處、到處。

Dh 方式副詞。例：逐一、從頭、一起。

Di 標誌副詞，幾乎都緊接在述詞之後，表現時態。例：過、著。



Dj 疑問副詞。例:為什麼、幹麼。

Dk 句副詞，有轉變或連接語氣的功能。例:總之、據說。

五、連接詞 (C)，用以表示並列關係或標明兩分句關係的詞。(依連接成分組合關係的不同下分兩類)

Ca 並列連接詞，連接兩個詞性相似的成分形成向心式結構，其中每一個成分的功能都跟整個結構相同。(下分兩類)

Caa 這類連接詞多半同時具有介詞的特性。例:和、跟。

Cab 連接兩個同類的成分，後一成分常可省略。包括:等、等等、之類。

Cb 關聯連接詞，能夠把幾個分句連成複句形式的連接詞。(下分三類)

Cba 移動性前繫連接詞，語意上具起頭作用，後面常須接一個分句，其所在分句可能移位至複句的後半段。(下分兩類)

Cbaa 偏正句移動性連接詞。

例:雖然、因為、即使、只有。

Cbab 偏正句句尾連接詞。這一類只有"的話"和"起見"。

Cbb 非移動性前繫連接詞。語意上具起頭作用，後面常須接一個分句，位置固定在前一分句。(下分兩類)

Cbba 偏正句非移動性前繫連接詞。例:雖、既、就是。

Cbbb 聯合句前繫連接詞。例:不但、一來。

Cbc 後繫連接詞，能將一個分句聯繫於前一個句子的連接詞。(下分兩類)

Cbca 偏正句後繫連接詞。例:可是、所以、那麼、否則。

Cbcb 聯合句後繫連接詞。例:而且、二來。

六、語助詞 (T)，附加於詞組或句子後的連用詞。(依語助詞間共存的次序分為四類)

Ta 了、的。

Tb 沒、沒有、而已、罷了、也好、也罷、云云、等等、之類、爾爾、來哉、著。

Tc 啊、呀、哇、哪、吶、呢、哩、啲、唷、嘛、嚙、麼、哦、喔、嘔、誼、耶、囉、嘍、吧、罷、啦、咧。

Td 了嗎、了否、而已嗎、啦云云、咧云云、嗎、否、不、與否、哉、耶、矣、啵。

如果有一個以上的語助詞一起出現，其先後的順序依序為：Ta，Tb，Tc。
Td 不與前三類共存。

七、感歎詞 (I)，表示說話者的口氣或態度的獨用語式。例：啊、喂、唉。

八、非謂形容詞 (A)，是純粹的形容詞，不具謂語作用。例：公共、共同。





附錄三、文件斷詞模組部份程式碼

```
package Thesis;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;

public class RayTextParser {

    private static String SpatialFileName = "D:\\My Documents\\Java\\Eclips
Workspace\\Protege3\\bin\\Thesis\\空間詞彙庫.txt";

    private static String AttributeFileName = "D:\\My Documents\\Java\\Eclips
Workspace\\Protege3\\bin\\Thesis\\屬性詞彙庫.txt";

    private static String SemanticVocabularyTmpStr = ""; // 儲存要用來推論的名詞詞彙
    private static String SpatialVocabularyTmpStr = ""; // 儲存具資源的空間詞彙
    private static String AttributeVocabularyTmpStr = ""; // 儲存具資源的屬性詞彙

    public String getSemanticVocabularyTmpStr () { // 取得語意推論的來源字串
        return SemanticVocabularyTmpStr;
    }

    public String getSpatialVocabularyTmpStr () { // 取得具資源的空間詞彙
        return SpatialVocabularyTmpStr;
    }

    public String getAttributeVocabularyTmpStr () { // 取得具資源的屬性詞彙
        return AttributeVocabularyTmpStr;
    }
}
```

```

public String ParseSemanticTable (String tmpStr) {           // 將推論路徑表拆解成輸出
文字檔的格式

    //String tmpStr = "熔岩\n0,熔岩流\n1,熔岩流,構成,火山地質\n2,火山地質,對應資源,屬
性詞\n===== \n 湖田國小\n0,湖田國小\n1,
湖田國小,相鄰於,A 聚落\n1,湖田國小,相鄰於,竹子湖路\n1,湖田國小,相鄰於,竹子湖\n2,
竹子湖,對應資源,空間詞\n===== \n";

    String tmpStr1 = tmpStr.replace
("===== ", ";");

    String[] tmpArr2 = tmpStr1.split (";");

    String Result = "";           // 最終結果

    for (int i=0;i<tmpArr2.length;i++) {

        String Result1 = "";           // 詞彙種類
        String Result2 = "";           // 起始個體
        String Result3 = "";           // 關係
        String Result4 = "";           // 新個體

        if (!tmpArr2[i].trim () .equals ("")) {
            String[] tmpArr3 = tmpArr2[i].split ("\n");
            for (int j=0;j<tmpArr3.length;j++) {
                // 取得起始個體
                // -----
                if ( ( i<3)
                    if (!tmpArr3[j].trim () .equals ("") && tmpArr3[j].length () >2) {
                        if (tmpArr3[j].substring (0,2) .equals ("0,") ) {
                            Result2 = tmpArr3[j].substring (2, tmpArr3[j].length () );
                        }
                    }
                }

                // 取得詞彙種類 & 關係種類
                // -----
                if (tmpArr3[j].indexOf ("對應資源") !=-1) {

```



```

String[] tmpArr4 = tmpArr3[j].split(",");

// 取得詞彙種類
Result1 = tmpArr4[3];
Result4 = tmpArr4[1];

// 取得關係種類
String[] tmpArr5 = tmpArr3[j-1].split(",");
if (tmpArr5.length==2) {
    Result3 = "對應資源";
}else if (tmpArr5.length==4) {
    Result3 = tmpArr5[2];
}else{
    Result3 = "錯誤";
}
}
}
Result += Result1 + "\t" + Result2 + "\t" + Result3 + "\t" + Result4 + "\n";
}
}
return Result;
}

public String ClassifyVocabulary (String sourceStr) { // 詞彙分類
/*
* 說明：
*
* 輸入：
*     紗帽山 (Na) 是 (SHI) 一 (Neu) 個 (Nf) 圓錐形 (Na) 火山 (Na) 丘
(Nb)， (COMMACATEGORY) 大部分 (Neqa) 由 (P) 破碎 (VH) 的 (DE) 熔岩 (Na)
構成 (VG)， (COMMACATEGORY)
*
*/

```



```

* 輸出：
*     空間名詞計有 0 個：
*     屬性名詞計有 0 個：
*     未知名詞計有 5 個：紗帽山,圓錐形,火山,丘,熔岩
*     未知其他詞計有 10 個：是,一個,,大部分,由,破碎,的,構成,,
*/

```

```

String result = "";

//addMessage ("開始將詞彙分類...");

SemanticVocabularyTmpStr = "";
SpatialVocabularyTmpStr = "";
AttributeVocabularyTmpStr = "";

String[] tmpStr = (this.getVocabularyTmpStr (sourceStr) ).split (";"); // 回傳結
果：紗帽山,圓錐形,火山,丘,熔岩;是,一個,,大部分,由,破碎,的,構成

String[] NounArr = tmpStr[0].split (","); // 取得
「名詞」字串陣列，結果：{"紗帽山","圓錐形","火","山丘","熔岩","，頂部","小峰","，
狀","烏紗帽","，它","七星山","火山",";"}

String[] UnNounArr = tmpStr[1].split (","); //
取得「非名詞」字串陣列，結果：{"是","一","個","，大部分","由","破碎","的","構成
","有","二","個","似","是","的","寄生","。"}

System.out.println ("NounArr = " + tmpStr[0] );

System.out.println ("UnNounArr = " + tmpStr[1] );

// 將「未知名詞」中的詞彙與詞彙庫互相比對

// 記得要將逗號排除掉

String hasResourceVocabularyOfSpatialTmpStr = ""; // 儲存具有資源的空間
詞彙

String hasResourceVocabularyOfAttributeTmpStr = ""; // 儲存具有資源的
屬性詞彙

String hasResourceVocabularyOfSemanticTmpStr = ""; // 儲存具有資源的
語意網個體

String NoResourceVocabularyTmpStr = ""; // 儲存無對
應資源的名詞詞彙

```

```

String ResourceStringOfSpatial = getSpatialResourceStr () ;
String ResourceStringOfAttribute = getAttributeResourceStr () ;
String ResourceStringOfSemantic = getSemanticResourceStr () ;
String[] ResourceArrayOfSpatial = ResourceStringOfSpatial.split (",");
String[] ResourceArrayOfAttribute = ResourceStringOfAttribute.split (",");
String[] ResourceArrayOfSemantic = ResourceStringOfSemantic.split (",");

boolean hasMatchedSaptialTag;           // 與空間詞彙庫比對成功的 tag
boolean hasMatchedAttributeTag;         // 與屬性詞彙庫比對成功的 tag
boolean hasMatchedSemanticTag;         // 與語意網比對成功的 tag
boolean TwoNounConnTag=false;          // 是否為兩個名詞合併的 tag
boolean PreTwoNounConnTag=false;       // 紀錄前一詞彙是否為兩個名詞合併的 tag
//boolean BreakFromAttributeTag=false; // 由掃描屬性詞彙庫後決定是否為為串聯
詞，並跳過第二個名詞

for (int i=0; i<NounArr.length; i++) { // 掃描「輸入詞彙」
    System.out.println ("NounArr[" + i + "] = " + NounArr[i]);

    // 跳過串聯名詞的第 2 個名詞：空間
    if (PreTwoNounConnTag) {
        //System.out.println ("【" + NounArr[i] + "】 跳過不執行");
        PreTwoNounConnTag=false;
        continue;
    }

    // 比對空間詞彙庫
    // -----
    hasMatchedSaptialTag = false;
    hasMatchedAttributeTag = false;
    hasMatchedSemanticTag = false;

```



```

for (int j=0; j<ResourceArrayOfSpatial.length; j++) { // 掃描「空間詞彙庫」
    // 設定變數初始值
    hasMatchedSaptialTag = false;
    TwoNounConnTag = false;

    // Step1:長詞合併
    if (i < NounArr.length-1) { // 陣列最後一個元素不進行長詞合併
        if (ResourceArrayOfSpatial[j].equals (NounArr[i]+NounArr[i+1]) ) {

            hasMatchedSaptialTag = true;
            TwoNounConnTag = true;
        }
    }

    // Step2:單詞比對
    if (ResourceArrayOfSpatial[j].equals (NounArr[i]) ) {
        hasMatchedSaptialTag = true;
        TwoNounConnTag = false;
    }

    if (hasMatchedSaptialTag) { // 找到符合字詞
        if (TwoNounConnTag) { // 為長詞合併
            hasResourceVocabularyOfSpatialTmpStr += NounArr[i] +
NounArr[i+1] + ",";
            PreTwoNounConnTag = true;
        }else{ // 不為長詞合併
            hasResourceVocabularyOfSpatialTmpStr += NounArr[i] + ",";
        }
        break; // 找到單詞後，馬上結束掃描空間詞彙庫以節省時間
    }
} // 掃描「空間詞彙庫」迴圈的 Ending

```



```

// 比對屬性詞彙庫
// -----
if (!hasMatchedSaptialTag) { // 空間詞彙庫有對應資源時便不掃描屬性
詞彙庫
    for (int k=0; k<ResourceArrayOfAttribute.length; k++) { // 掃描「屬性詞彙
庫」

        // 設定變數初始值
        hasMatchedAttributeTag = false;
        TwoNounConnTag = false;

        // Step1:長詞合併
        if (i < NounArr.length-1) { // 陣列最後一個元素不進行長詞合併
            if (ResourceArrayOfAttribute[k].equals
(NounArr[i]+NounArr[i+1])) {
                hasMatchedAttributeTag = true;
                TwoNounConnTag = true;
            }
        }

        // Step2:單詞比對
        if (ResourceArrayOfAttribute[k].equals (NounArr[i])) {
            hasMatchedAttributeTag = true;
            TwoNounConnTag = false;
        }

        if (hasMatchedAttributeTag) { // 找到符合字詞
            if (TwoNounConnTag) { // 為長詞合併
                hasResourceVocabularyOfAttributeTmpStr += NounArr[i] +
NounArr[i+1] + ",";
                PreTwoNounConnTag = true;
            }else{ // 不為長詞合併

```



```

        hasResourceVocabularyOfAttributeTmpStr += NounArr[i] +
";";
    }
    break;          // 找到單詞後，馬上結束掃描屬性詞彙庫以節省
時間
}
} // 掃描「屬性詞彙庫」的迴圈 Ending
}

```

// 比對語意網個體清單

// -----

```

    if (hasMatchedSaptialTag || hasMatchedAttributeTag) {          // 空間詞彙庫、屬
性詞彙庫有對應資源時便不掃描語意網

```

```

}else{

```

```

/*

```

```

for (int k=0; k<ResourceArrayOfSemantic.length; k++) { // 掃描「語意網」

```

```

    if (ResourceArrayOfSemantic[k].indexOf (NounArr[i]) !=-1) {

```

```

        hasResourceVocabularyOfSemanticTmpStr += NounArr[i] + ",";

```

```

        hasMatchedSemanticTag = true;

```

```

        break;          // 找到單詞後，馬上結束掃描語意網以節省時間

```

```

    }

```

```

} // 掃描「語意網」的迴圈 Ending

```

```

*/

```

```

for (int k=0; k<ResourceArrayOfSemantic.length; k++) { // 掃描「語意網」

```

```

    // 設定變數初始值

```

```

    hasMatchedSemanticTag = false;

```

```

    TwoNounConnTag = false;

```

```

    // Step1:長詞合併

```



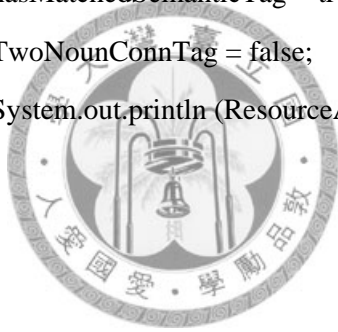
```

        if (i < NounArr.length-1) { // 陣列最後一個元素不進行長詞合併
            //System.out.println (ResourceArrayOfSemantic[k] + " == " +
NounArr[i]+NounArr[i+1]);
            if (ResourceArrayOfSemantic[k].equals
(NounArr[i]+NounArr[i+1])) {
                hasMatchedSemanticTag = true;
                TwoNounConnTag = true;
            }
        }

        if (!hasMatchedSemanticTag) {
            // Step2:單詞模糊比對
            if (ResourceArrayOfSemantic[k].indexOf (NounArr[i]) !=-1) {
                hasMatchedSemanticTag = true;
                TwoNounConnTag = false;
                System.out.println (ResourceArrayOfSemantic[k] + " like " +
NounArr[i]);
            }
        }

        if (hasMatchedSemanticTag) { // 找到符合字詞
            if (TwoNounConnTag) { // 為長詞合併
                hasResourceVocabularyOfSemanticTmpStr += NounArr[i] +
NounArr[i+1] + ",";
                PreTwoNounConnTag = true;
            }else{ // 不為長詞合併
                hasResourceVocabularyOfSemanticTmpStr += NounArr[i] +
",";
            }
            System.out.println ("中斷!");
            break; // 找到單詞後，馬上結束掃描屬性詞彙庫以節省
時間
        }

```



```

        //System.out.println ("繼續執行 1");
    } // 掃描「語意網」的迴圈 Ending
    //System.out.println ("繼續執行 2");
}
//System.out.println ("繼續執行 3");
if (!hasMatchedSaptialTag && !hasMatchedAttributeTag
&& !hasMatchedSemanticTag) {
    NoResourceVocabularyTmpStr += NounArr[i] + ",";
}
} // 掃描「輸入詞彙」的迴圈 Ending

// 去除結尾的逗點
if (hasResourceVocabularyOfSpatialTmpStr.length () >1)
hasResourceVocabularyOfSpatialTmpStr =
hasResourceVocabularyOfSpatialTmpStr.substring (0,
hasResourceVocabularyOfSpatialTmpStr.length () -1);

if (hasResourceVocabularyOfAttributeTmpStr.length () >1)
hasResourceVocabularyOfAttributeTmpStr =
hasResourceVocabularyOfAttributeTmpStr.substring (0,
hasResourceVocabularyOfAttributeTmpStr.length () -1);

if (hasResourceVocabularyOfSemanticTmpStr.length () >1)
hasResourceVocabularyOfSemanticTmpStr =
hasResourceVocabularyOfSemanticTmpStr.substring (0,
hasResourceVocabularyOfSemanticTmpStr.length () -1);

if (NoResourceVocabularyTmpStr.length () >1) NoResourceVocabularyTmpStr =
NoResourceVocabularyTmpStr.substring (0, NoResourceVocabularyTmpStr.length () -1);

// 將 tmpStr 轉換為 tmpArr，用來計算詞彙數量
String[] hasResourceVocabularyOfSpatialTmpArr =
hasResourceVocabularyOfSpatialTmpStr.split (",");

String[] hasResourceVocabularyOfAttributeTmpArr =
hasResourceVocabularyOfAttributeTmpStr.split (",");

String[] hasResourceVocabularyOfSemanticTmpArr =
hasResourceVocabularyOfSemanticTmpStr.split (",");

String[] NoResourceVocabularyTmpArr = NoResourceVocabularyTmpStr.split (",");

SemanticVocabularyTmpStr = hasResourceVocabularyOfSemanticTmpStr;

```

```

    if (hasResourceVocabularyOfSpatialTmpArr[0].equals ("") &&
hasResourceVocabularyOfSpatialTmpArr.length==1) {

        result += "空間名詞計有 0 個：\n";

    }else{

        result += "空間名詞計有" + hasResourceVocabularyOfSpatialTmpArr.length + "
個：" + hasResourceVocabularyOfSpatialTmpStr + "\n";

        SpatialVocabularyTmpStr = hasResourceVocabularyOfSpatialTmpStr;

    }

    if (hasResourceVocabularyOfAttributeTmpArr[0].equals ("") &&
hasResourceVocabularyOfAttributeTmpArr.length==1) {

        result += "屬性名詞計有 0 個：\n";

    }else{

        result += "屬性名詞計有" + hasResourceVocabularyOfAttributeTmpArr.length + "
個：" + hasResourceVocabularyOfAttributeTmpStr + "\n";

        AttributeVocabularyTmpStr = hasResourceVocabularyOfAttributeTmpStr;

    }

    result += "=====\n";

    if (hasResourceVocabularyOfSemanticTmpArr.length>0) {

        if (hasResourceVocabularyOfSemanticTmpArr[0].equals ("") &&
hasResourceVocabularyOfSemanticTmpArr.length==1) {

            result += "語意網可支援名詞計有 0 個：\n";

        }else{

            result += "語意網支援的名詞計有" +
hasResourceVocabularyOfSemanticTmpArr.length + "個：" +
hasResourceVocabularyOfSemanticTmpStr + "\n";

        }

    }else{

        result += "語意網可支援名詞計有 0 個：\n";

    }

    result += "=====\n";

```

```

if (NoResourceVocabularyTmpArr[0].equals ("") &&
NoResourceVocabularyTmpArr.length==1) {

    result += "未知名詞計有 0 個：\n";

}else{

    result += "未知名詞計有" + NoResourceVocabularyTmpArr.length + "個：" +
NoResourceVocabularyTmpStr;

}

```

```

if (UnNounArr[0].equals ("") && UnNounArr.length==1) {

    result += "未知其他詞計有 0 個：\n";

}else{

    result += "\n 未知其他詞計有" + UnNounArr.length + "個：";

}

```

```

for (int i=0; i<UnNounArr.length; i++) result += UnNounArr[i] + ",";
result = result.substring (0, result.length () -1);
//addMessage ("完成詞彙分類!");
return result;
}

```



```

public String getVocabularyTmpStr (String sourceStr) { // 取得抽離名詞後的暫存字串

```

```

/*
* 說明：
* 取得抽離名詞後的暫存字串，資料型態：「名詞暫存字串;非名詞暫存字串」
* 分號 (;)：用來做為暫存字串的分隔字元
* 逗號 (,)：用來做為暫存字串中，每個詞彙的分隔元
* 詞性部份僅針對 Na、Nb、Nc 三種做為關注的名詞詞性
* 輸入：
* 紗帽山 (N) 是 (Vt) 一 (DET) 個 (M) 圓錐形 (N) 火 (N) 山丘 (N) ， (COMMACATEGORY)， 大部分 (DET) 由 (P) 破碎 (Vi) 的 (T) 熔岩 (N) 構成 (Nv) ， (COMMACATEGORY)， 頂部 (N) 有 (Vt) 二 (DET) 個 (M) 小峰 (N) ， (COMMACATEGORY)， 狀 (N) 似 (Vt) 烏紗帽 (N) ， (COMMACATEGORY)， 它 (N) 是 (Vt) 七星山 (N) 的 (T) 寄生 (Nv) 火山 (N) 。 (PERIODCATEGORY)，

```

```

* 輸出：
*  紗帽山,圓錐形,火山,丘,熔岩 ； 是,一個,，,大部分,由,破碎的,構成
*/

String result;

String tmpNounStr = "";           // 名詞詞彙暫存字串：「紗帽山,圓錐
形,...」
String tmpUuNounStr = "";       // 非名詞詞彙暫存字串：「是,一個,...」

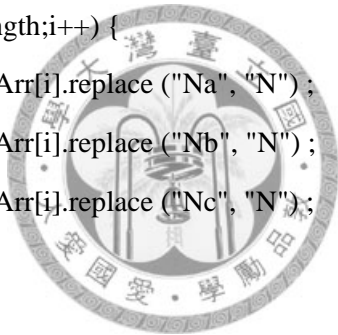
sourceStr = sourceStr.replace (" (", ",");
sourceStr = sourceStr.replace (" )", ",");
String[] tmpStrArr = sourceStr.split (",");

for (int i=0; i<tmpStrArr.length;i++) {
    tmpStrArr[i] = tmpStrArr[i].replace ("Na", "N");
    tmpStrArr[i] = tmpStrArr[i].replace ("Nb", "N");
    tmpStrArr[i] = tmpStrArr[i].replace ("Nc", "N");
}

for (int i=0; i<tmpStrArr.length-2; i+=2) {
    if (tmpStrArr[i+1].equals ("N") ) tmpNounStr += (tmpStrArr[i].trim () ) .replace ("
", "") + ",";
    else if (tmpStrArr[i+1].equals ("COMMACATEGORY") ) {
        // 排除標點符號，不做任何事
    }
    else tmpUuNounStr += (tmpStrArr[i].trim () ) .replace (" ", "") + ",";
}

// 去除結尾多餘的逗號
if (tmpNounStr.length () >0) tmpNounStr = tmpNounStr.substring (0, tmpNounStr.length ()
-1);

```



```

    if (tmpUuNounStr.length () >0) tmpUuNounStr = tmpUuNounStr.substring (0,
tmpUuNounStr.length () -1);

    result = tmpNounStr + ";" + tmpUuNounStr;

    return result;

}

public String MarkSyntax (String sourceStr, String ServerType) { // 連接至中研院斷詞系
統進行詞性標註

/*
* 說明：
* 連接至中研院斷詞系統進行詞性標註
* 輸入：待標註詞性字串
* 輸出：完成詞性標註字串
*/

boolean ServerTag = true;
String hostname = "140.109.19.104";
int Port = 1501;
String account = "jaudan";
String pwd = "12345";

// 連線參數設定
if (ServerType.equals ("來源文章 (伺服器 1)")) {
    // 中研院斷詞系統主伺服器
    hostname = "140.109.19.104";
    Port = 1501;
    account = "jaudan";
    pwd = "12345";
    ServerTag = true;
} else if (ServerType.equals ("來源文章 (伺服器 2)")) {

```



```

// 交大斷詞系統備用伺服器
hostname = "140.113.39.209";
Port = 1501;
account = "jaudan";
pwd = "12345";
ServerTag = true;
}else if (ServerType.equals ("來源文章 (Demo) ")) {
    ServerTag = false;
}else{
    ServerTag = false;
}

String responseStr;
PrintWriter out = null;
BufferedReader networkIn = null;

if (ServerTag) {
    //addMessage ("開始標註詞性...");
    try{
        Socket theSocket = new Socket (hostname, Port) ;
        networkIn = new BufferedReader (new InputStreamReader
(theSocket.getInputStream () ) ) ;
        //BufferedReader userIn = new BufferedReader (new InputStreamReader
(System.in) ) ;
        out = new PrintWriter (theSocket.getOutputStream () ) ;
        //addMessage ("連接至詞性標註系統..");
        out.println ("<?xml version=\"1.0\" ?><wordsegmentation
version=\"0.1\"><option showcategory=\"1\" /><authentication username=\"\" + account +
\"\" password=\"\" + pwd + \"\" /><text>\" + sourceStr + \"</text></wordsegmentation>");
        out.flush () ;

```



```

// 取出結果字串 ...<sentence> 陽明山 (N) 的 (T) 地上 (N) 有
(Vt) 東西 (N) </sentence>...

responseStr = networkIn.readLine ();

responseStr = responseStr.replace ("</sentence><sentence>", " , ");

responseStr = responseStr.substring (responseStr.indexOf ("<sentence>")
+1+10 , responseStr.indexOf ("</sentence>"));

//addMessage ("完成 !");

return responseStr;
}
catch (IOException e) {
return "錯誤 : [" + e + "];
}
finally{
try{
if (networkIn != null) networkIn.close ();
if (out != null) out.close ();
}
catch (IOException e) {}
}
}
}

if (sourceStr.equals ("湖底窪地因為河流侵蝕熔岩台地而形成湖田國小和泉源國
小座落於此窪地紗帽橋位於窪地西側出口")) {

return "湖 (N) 底 (N) 窪地 (N) 因為 (C) 河流 (N) 侵蝕 (Nv)
熔岩 (N) 台地 (N) 而 (C) 形成 (Vt) 湖田 (N) 國小 (N) 和 (C) 泉
源 (N) 國小 (N) 座落 (Vt) 於 (P) 此 (DET) 窪地 (N) 紗帽橋 (N)
位於 (Vt) 窪地 (N) 西側 (N) 出口 (Vt)";

}

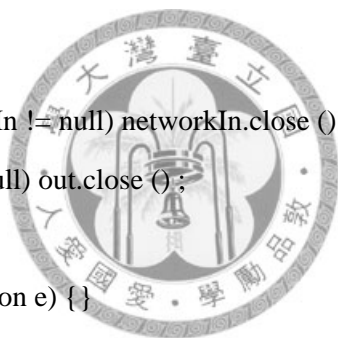
}else if (sourceStr.equals ("紗帽山是一個圓錐形火山丘大部分由破碎的熔岩構成
頂部有二個小峰狀似烏紗帽它是七星山的寄生火山")) {

return "紗帽山 (N) 是 (Vt) 一 (DET) 個 (M) 圓錐形 (N) 火
(N) 山丘 (N) 大部分 (DET) 由 (P) 破碎 (Vi) 的 (T) 熔岩 (N) 構
成 (Vt) 頂部 (N) 有 (Vt) 二 (DET) 個 (M) 小峰狀 (N) 似 (Vt) 烏
紗帽 (N) 它 (N) 是 (Vt) 七星山 (N) 的 (T) 寄生 (Nv) 火山 (N) ";

}

}else if (sourceStr.equals ("竹子湖地區遠在五十萬年前曾為堰塞湖")) { //
竹子湖由破碎的熔岩構成

```




```

        return "竹子湖 (Nc) 地區 (Nc) 遠 (VH) 在 (P) 五十萬 (Neu) 年前 (Nd)
曾 (D) 為 (P) 堰塞 (Nb) 湖 (Na)";// 竹子湖 (N) 由 (P) 破碎 (Vi) 的 (T) 熔岩 (N)
構成 (Vt)

    }else if (sourceStr.equals ("湖田國小座落於湖底窪地")) {

        return "湖田 (N) 國小 (N) 座落 (Vt) 於 (P) 此 (DET) 窪地 (N) 湖 (N)
底 (N) 窪地 (N) ";

    }else if (sourceStr.equals ("紗帽山是一個圓錐狀火山")) {

        return "紗帽山 (Na) 是 (SHI) 一 (Neu) 個 (Nf) 圓錐狀 (Na) 火山 (Na)
";

    }else if (sourceStr.equals ("大部份由破碎的熔岩構成")) {

        return "大部分 (Neqa) 由 (P) 破碎 (VH) 的 (DE) 熔岩 (Na) 構成 (VG)
";

    }else if (sourceStr.equals ("木山層是顏滄波和陳培源提出來的一個地層名稱")) {

        return "木山層 (Na) 是 (SHI) 顏滄波 (Nb) 和 (Caa) 陳培源 (Nb) 提出
來 (VC) 的 (DE) 一 (Neu) 個 (Nf) 地層 (Na) 名稱 (Na) ";

    }else{

        return "非範例語句，請使用伺服器 1 或 2 !" + sourceStr;

    }

}

}

```



```

public String getSpatialResourceStr () { // 取得空間詞彙庫
的名詞清單

```

```

/*

```

```

* 說明：

```

```

* 取得空間詞彙庫的名詞清單，以逗號分隔

```

```

* 輸出：

```

```

* 台北縣,基隆市,台北市,七堵區,三芝鄉,士林區,北投區,...

```

```

*/

```

```

String s = "";

```

```

try{

```

```

    //String SpatialFileName = "D:\\My Documents\\Java\\Eclips
Workspace\\Protege3\\bin\\Thesis\\空間詞彙庫.txt";

```

```

    File f = new File (SpatialFileName) ;

```

```

    FileReader fr = new FileReader (SpatialFileName) ;
    char buffer[] = new char[ (int) f.length () ];
    fr.read (buffer) ;

    for (int i=0; i<buffer.length; i++) s += buffer[i];
    s = s.replace ("\n", ",");
    s = s.replace ("\r", "");
    s = s.trim () ;
    s = s.substring (0, s.length () -1) ;
} catch (IOException e) {
    System.out.println ("檔案路徑錯誤！ ");
}
return s;
}

public String getAttributeResourceStr () { // 取得屬性詞彙庫
    的名詞清單
    /*
    * 說明：
    * 取得屬性詞彙庫的名詞清單，以逗號分隔
    * 輸出：
    * 土石堆積,崩塌地,崩塌地,溫泉及天然氣,...
    */

    String s = "";
    try{
        //String AttributeFileName = "D:\\My Documents\\Java\\Eclips
        Workspace\\Protege3\\bin\\Thesis\\屬性詞彙庫.txt";

        File f = new File (AttributeFileName) ;
        FileReader fr = new FileReader (AttributeFileName) ;
        char buffer[] = new char[ (int) f.length () ];
        fr.read (buffer) ;

```



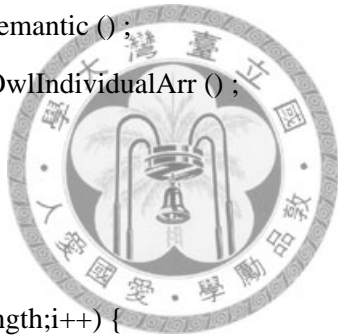
```

    for (int i=0; i<buffer.length; i++) s += buffer[i];

    s = s.replace ("\n", ",");
    s = s.replace ("\r", "");
    s = s.trim ();
    s = s.substring (0, s.length () -1);
} catch (IOException e) {
    System.out.println ("檔案路徑錯誤！ ");
}
return s;
}

public String getSemanticResourceStr () { // 取得詞意
網支援的個體清單
    RaySemantic s = new RaySemantic ();
    String[] TmpResult = s.getOwlIndividualArr ();
    String ResultStr = "";
    // 去除「unnamed:」字串
    int j=0;
    for (int i=0;i<TmpResult.length;i++) {
        ResultStr += TmpResult[i].replace ("unnamed:", "") + ",";
        j++;
    }
    if (ResultStr.length () >0) ResultStr = ResultStr.substring (0, ResultStr.length () -1);
    // 去除結尾逗號
    return ResultStr;
}
}

```






```

*         getOwlClassArr ()
           取得語意網中的「類別清單」                                ok
*         getOwlIndividualArr ()
           取得語意網中的「個體清單」                                ok

*         getOWLModel ()
           初始 owlModel 物件
ok
*/

String OwlUri =
"http://localhost/owl/for%20Protege%203.4%20OWL/Yangminshan_Program/Yangminshan_Program_owl.owl";

static int LineNum = 0;           // Log 檔行號
static String Log = "";          // Log 檔，紀錄函式運作情形
static int Depth = 0;            // 紀錄語意深度
static int DepthLimit = 3;      // 限制最大語意深度

public String getLog () {
    return Log;
}

public void setDepthLimit (int LimitNum) { // 設
    定最大語意深度，預設值為 3
    DepthLimit=LimitNum;
}

public String[] getLikeIndividual (String IndividualName) { // 用
    來取得類似的個體清單
    // 回傳結果：{"湖田里","湖田國小"}
    String LikeIndividualNameTmpStr = ""; // 類似的個體
    清單

    String[] IndividualArr = this.getOwlIndividualArr ();
    for (int i=0;i<IndividualArr.length;i++) {
        if (IndividualArr[i].indexOf (IndividualName) !=-1) {
            //System.out.println (IndividualArr[i] + " = " + "unnamed:" +
            IndividualName + "\n");

```



```

        //if (IndividualArr[i].equals ("unnamed:" + IndividualName) ) {
            LikeIndividualNameTmpStr += IndividualArr[i] + ",";
        }
    }

    if (LikeIndividualNameTmpStr.length () >1)
    LikeIndividualNameTmpStr = LikeIndividualNameTmpStr.substring
    (0, LikeIndividualNameTmpStr.length () -1) ;

    LikeIndividualNameTmpStr = LikeIndividualNameTmpStr.replace
    ("unnamed:", "");

    return LikeIndividualNameTmpStr.split (",") ;
}

String InferNewIndividual (String IndividualName) {           // 以
各種關係開始向外串連，分別處理 DatatypeProperty 和
ObjectProperty
/* 回傳結果：
*      1,竹子湖,ObjectProperty,被包含於,陽明山國家公園\n
*      2,竹子湖,ObjectProperty,被包含於,無資源面景點\n
*      2,陽明山國家公園,DatatypeProperty,對應資源,陽明山國家
公園.Lyr
*/
Log += "[" + ++LineNum + "];

Log += "列舉「 " + IndividualName + "」個體的關係，並取得其關
係清單：\n";

//String IndividualName = "unnamed:" + sourceIndividualName;
String Result = "";
Depth++;           // 語意深度 + 1
if (Depth <= DepthLimit) {
    OWLModel owlModel = getOWLModel ();           // 初始
owlModel 物件

    OWLIndividual individual = owlModel.getOWLIndividual
    (IndividualName) ;

    if (individual != null) {

```

```

//Result += "Msg:以下為「 + IndividualName + "」的推論
路徑表\n";

Collection properties = individual.getRDFProperties ();
String ObjectPropertyTmpStr = "";
String DatatypePropertyTmpStr = "";
for (Iterator it = (Iterator) properties.iterator (); it.hasNext ();)
{
    RDFProperty eachProperty = (RDFProperty) it.next ();
    String PropertyType = (eachProperty.getProtegeType
()) .toString ();
    if (PropertyType.indexOf ("ObjectProperty") != -1) {

        ObjectPropertyTmpStr += eachProperty.getBrowserText
() + ",";
    }else if (PropertyType.indexOf ("DatatypeProperty") != -1) {
        DatatypePropertyTmpStr +=
eachProperty.getBrowserText () + ",";
    }else{
        // 對於其他種類的 Property 不做任何處置
    }
}

// 針對 Datatype Property 做處理 (檢查「對應資源名稱」關
係中有無字串，即代表有無資源)
//
-----

if (DatatypePropertyTmpStr!="") {
    if (DatatypePropertyTmpStr.length () >0) Log +=
"\tDatatypeProperty : \t (" + DatatypePropertyTmpStr.substring (0,
DatatypePropertyTmpStr.length () -1) + ") \n";

    String[] DatatypePropertyArr =
DatatypePropertyTmpStr.split (",");
    for (int i=0; i< DatatypePropertyArr.length; i++) {
        if (!DatatypePropertyArr[i].equals ("null")
&& !DatatypePropertyArr[i].equals ("")) {

```



```

Collection values = ind.getPropertyValues (hasResource) ;
String[] sArray = new String[values.size () ];
int i = 0;
for (Iterator it = (Iterator) values.iterator () ; it.hasNext () ;) {
    sArray[i] = (String) it.next () ;
    i++;
}
return sArray;
}

String[] getObjectPropertyValue (String ObjectPropertyName, String
IndividualName) { // 取得指定 ObjectProperty 的個體名
稱清單
/* 回傳結果 :
*   MyArray[0] = unnamed:無資源面景點
*   MyArray[1] = unnamed:陽明山國家公園
*   MyArray[2] = null <-- 記得要排除之
*/
Log += "[" + ++LineNum + "]" ;
Log += "取得 「" + ObjectPropertyName + "」 (ObjectProperty) 的
值。 \n";

    OWLModel owlModel = getOWLModel () ; // 初始 owlModel
物件

    OWLIndividual ind = owlModel.getOWLIndividual
(IndividualName) ;

    OWLObjectProperty hasResource =
owlModel.getOWLObjectProperty (ObjectPropertyName) ;

    Collection values = ind.getPropertyValues (hasResource) ;

    String[] sArray = new String[values.size () ];

    int i = 0;

    if (values.size () >0) {
        for (Iterator it = (Iterator) values.iterator () ; it.hasNext () ;) {

```

```

        OWLIndividual ind1 = (OWLIndividual) it.next () ;
        sArray[i] = (String) ind1.getBrowserText () ;
        i++;
    }
}
return sArray;
}

public String getInferRouteTable (String InputIndividualTmpStr) {
// 取得傳入多個個體的推論路徑表

/*      傳入：「窪地,河流,熔岩,台地,窪地,湖田,紗帽橋,窪地,西側」
*      回傳結果：
*          熔岩
<-- 使用者的詞彙
          0,熔岩流
<-- 語意網中對應的個體
          1,熔岩流,岩層可構成地質,火山地質
<-- 推論路徑表
          2,火山地質,對應資源,屬性詞

=====

          湖田國小
<-- 使用者的詞彙
          0,湖田國小
<-- 語意網中對應的個體
          1,湖田國小,相鄰於,A 聚落
<-- 推論路徑表
          1,湖田國小,相鄰於,竹子湖路
          1,湖田國小,相鄰於,竹子湖
          2,竹子湖,對應資源,空間詞

=====

*/

String InferRouteTalbe=""; // 推論路徑表

```

```

String[] InputIndividualTmpStrArr = InputIndividualTmpStr.split
(",");

String SemanticSupportedIndividualTmpStr = "";           // 可
被語意網支援的詞彙 (輸入的詞彙)

String SemanticSupportedIndividualTmpStr1 = "";         //
可被語意網支援的詞彙 (語意網的個體)

RaySemantic t = new RaySemantic ();

String[] IndividualTmpArr = t.getOwlIndividualArr ();    // 取
得語意網中的「個體清單」
for (int i=0;i<IndividualTmpArr.length;i++) {
    for (int j=0;j<InputIndividualTmpStrArr.length;j++) {
        if (IndividualTmpArr[i].indexOf
(InputIndividualTmpStrArr[j]) !=-1) {                   // 模糊比對
            SemanticSupportedIndividualTmpStr +=
InputIndividualTmpStrArr[j] + ",";
            SemanticSupportedIndividualTmpStr1 +=
IndividualTmpArr[i] + ",";
        }
    }
}

SemanticSupportedIndividualTmpStr =
SemanticSupportedIndividualTmpStr.substring (0,
SemanticSupportedIndividualTmpStr.length () -1);       // 去除結
尾逗號

SemanticSupportedIndividualTmpStr1 =
SemanticSupportedIndividualTmpStr1.substring (0,
SemanticSupportedIndividualTmpStr1.length () -1);

String[] SemanticSupportedIndividualTmpStrArr =
SemanticSupportedIndividualTmpStr.split ("");

String[] SemanticSupportedIndividualTmpStrArr1 =
SemanticSupportedIndividualTmpStr1.split ("");

// 語意推論
for (int i=0;i<SemanticSupportedIndividualTmpStrArr1.length;i++) {

```

```

    InferRouteTalbe += SemanticSupportedIndividualTmpStrArr[i] +
"\n";

    String[] LikeIndividualTmpArr = t.getLikeIndividual
(SemanticSupportedIndividualTmpStrArr1[i]); // 熔岩流、紗帽山

    for (int k=0;k<LikeIndividualTmpArr.length;k++) {

        System.out.println
("SemanticSupportedIndividualTmpStrArr1[" + i + "] = " +
SemanticSupportedIndividualTmpStrArr1[i]);

        System.out.println ("\tLikeIndividualTmpArr[" + k + "] = " +
LikeIndividualTmpArr[k]);

        InferRouteTalbe += "0," + LikeIndividualTmpArr[k] + "\n";

        InferRouteTalbe += t.InferNewIndividual ("unnamed:" +
LikeIndividualTmpArr[k]); // 執行個體的推論

    }

    InferRouteTalbe +=
"=====\n";

}

return InferRouteTalbe.replace ("unnamed:", "");

}

String[] getOwlClassArr () { // 取得語意網中的「類別清單」
函式

    Log += "[" + ++LineNum + "]\n";

    Log += "產生個體索引表。 \n";

    OWLModel owlModel = getOWLModel (); // 初始 owlModel
物件

    String[] ClassNameArr;

    Collection classes = owlModel.getUserDefinedOWLNamedClasses
();

    ClassNameArr = new String[classes.size () ];

    int i=0;

    for (Iterator it = classes.iterator () ; it.hasNext () ;) {

        OWLNamedClass cls = (OWLNamedClass) it.next () ;

```



```

        ClassNameArr[i] = cls.getBrowserText () ;
        i++;
    }
    return ClassNameArr;
}

String[] getOwlIndividualArr () {    // 取得語意網中的「個體清單」
函式
    OWLModel owlModel = getOWLModel () ;    // 初始 owlModel
物件

    Collection Instances = owlModel.getUserDefinedRDFIndividuals
(false) ;

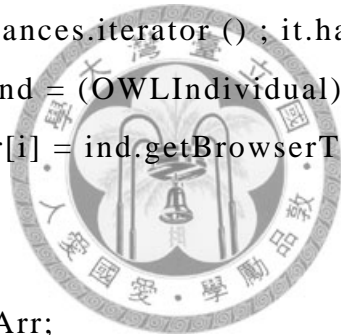
    String[] InstanceNameArr = new String[Instances.size () ] ;
    int i=0;
    for (Iterator it = Instances.iterator () ; it.hasNext () ;) {
        OWLIndividual ind = (OWLIndividual) it.next () ;
        InstanceNameArr[i] = ind.getBrowserText () ;
        i++;
    }
    return InstanceNameArr;
}
/*

String[] SortedOwlIndividualArr (String[] UnSortIndividualArr) {
// 將「個體清單」重新排序的函式

}*/

OWLModel getOWLModel () {    // 初始 owlModel 物
件
    OWLModel owlModel = null;
    try {
        owlModel = ProtegeOWL.createJenaOWLModelFromURI
(OwlUri) ;
    } catch (OntologyLoadException e) {

```



```
// TODO Auto-generated catch block
e.printStackTrace ();
}
return owlModel;
}
}
```



附錄五、查詢定位圖台部份程式碼

Dim WorkingFolderPath As String ' 程式運作資料夾路徑

Dim LayerFolderPath As String ' 圖層所在的路徑

Dim KeywordMatchResultFilePath As String ' 詞彙庫比對結果文字檔路徑

Dim SemanticResultFilePath As String ' 語意推論結果文字檔路徑

Dim strBufferDefaultDistance As String ' Buffer 函式預設的半徑

Private Sub cbOverlay_Click ()

 Call PolygonsOverlay

End Sub

Private Sub lbxKeywordAttributeLoaded_Click ()

 If lbxKeywordAttributeLoaded.ListCount > 0 Then

 ' 將圖層由地圖中移除

 Dim LoadedArr As Variant

 LoadedArr = VBA.Split (lbxKeywordAttributeLoaded.Text, ",")

 RemoveLayer (LoadedArr (UBound (LoadedArr))) ' 移除指定的 Layer

 ' 將「已載入圖層」中的項目移至「未載入圖層」中

 lbxKeywordAttributeUnLoad.AddItem lbxKeywordAttributeLoaded.Text

 lbxKeywordAttributeLoaded.RemoveItem lbxKeywordAttributeLoaded.ListIndex

 End If

End Sub

Private Sub lbxKeywordAttributeUnLoad_Click ()

 If lbxKeywordAttributeUnLoad.ListCount > 0 Then

 ' 將圖層加到地圖中

 Dim UnLoadArr As Variant

 UnLoadArr = VBA.Split (lbxKeywordAttributeUnLoad.Text, ",")

```
LoadLayer (UnLoadArr (UBound (UnLoadArr))) ' 載入指定的 Layer
```

```
' 將「未載入圖層」中的項目移至「已載入圖層」中
```

```
lbxKeywordAttributeLoaded.AddItem lbxKeywordAttributeUnLoad.Text
```

```
lbxKeywordAttributeUnLoad.RemoveItem lbxKeywordAttributeUnLoad.ListIndex
```

```
End If
```

```
End Sub
```

```
Private Sub lbxKeywordSpatial_Click ()
```

```
Dim ArrEchoTmp As Variant
```

```
Dim LastStr As String
```

```
ArrEchoTmp = VBA.Split (lbxKeywordSpatial.Text, ",")
```

```
LastStr = ArrEchoTmp (UBound (ArrEchoTmp)) '取得最後一個空間詞彙
```

```
Call FindAndSelect (LastStr) ' 尋找並選擇空間詞彙
```

```
End Sub
```

```
Private Sub txtKeyword_MouseUp (ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
```

```
txtKeyword.Text = ""
```

```
End Sub
```

```
Private Sub UserForm_Initialize ()
```

```
WorkingFolderPath = "D:\My Documents\NTU_臺灣大學\2_論文\01_文獻回顧  
\01_TextMining\沈嘉慧\嘉慧的論文光碟資料\DATA\"
```

```
LayerFolderPath = WorkingFolderPath & "Layers\"
```

```
KeywordMatchResultFilePath = WorkingFolderPath & "\Ray\詞彙庫比對結果.txt"
```

```
SemanticResultFilePath = WorkingFolderPath & "\Ray\語意推論結果.txt"
```

```
strBufferDefaultDistance = "100" ' 預設 buffer 的距離為 100 公尺
```

```
Call btnReadTxt_Click
```

End Sub

Private Sub btnBuffer_Click ()

Call BufferFeatures (strBufferDefaultDistance)

End Sub

Private Sub btnFind_Click ()

Call FindAndSelect (txtKeyword.Text)

End Sub

Private Sub FindAndSelect (strKeywords As String) ' 尋找空間定位點並選擇的函式

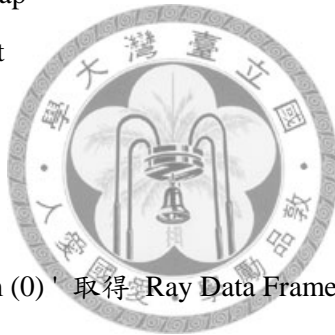
' IMxDocument -> IMaps -> IMap

Dim pMxDoc As IMxDocument

Set pMxDoc = ThisDocument

Dim pMap As IMap

Set pMap = pMxDoc.Maps.Item (0) ' 取得 Ray Data Frame 變數



' IMap -> IFeatureLayerDefinition -> .DefinitionExpression=strQuery

Dim pLocationLayerDef As IFeatureLayerDefinition

Dim pLayer As ILayer

Dim pLocationLayer As ILayer

Dim ResultMsg As String ' 訊息

For i = 0 To pMap.LayerCount - 1

Set pLayer = pMap.Layer (i)

' 尋找指定的圖層，且屬性表中必需要有 name 欄位才可查找

If pLayer.name = "地名總資料" Or pLayer.name = "道路圖" Then

```
Set pLocationLayerDef = pLayer ' 取得「地名總資料」Layer 變數
Set pLocationLayer = pLayer
```

```
Dim strQuery As String
If obEqual.Value Then
strQuery = "Name = '" & strKeywords & "'" ' 完全比對
Else
strQuery = "Name like '%" & strKeywords & "%'" ' 糊糊比對
End If
```

```
Dim pActiveView As IActiveView ' 取得目前的 ActiveView
Set pActiveView = pMap
```

```
Dim pLocationLeryFeatureSelection As IFeatureSelection
Set pLocationLeryFeatureSelection = pLocationLayer
```

```
If txtKeyword.Text = "" Then ' Zoom Out 至全景
pLocationLayerDef.DefinitionExpression = ""
pActiveView.Extent = pLocationLayer.AreaOfInterest
```

```
Else ' Zoom In 至選擇的點
```

```
Dim pFilter As IQueryFilter
Set pFilter = New QueryFilter
```

```
pFilter.WhereClause = strQuery
```

```
pLocationLeryFeatureSelection.SelectFeatures pFilter, esriSelectionResultNew, True
```

```
' 顯示搜尋結果訊息
```

```
If pLocationLeryFeatureSelection.SelectionSet.Count = 0 Then
```

```
'ResultMsg = ResultMsg & ""
```

```
Else
```

```
ResultMsg = ResultMsg & "在「" & pLayer.name & "」中找到 " &  
pLocationLeryFeatureSelection.SelectionSet.Count & " 筆紀錄!" & vbCrLf
```

```
End If
```

```
Dim pRedColor As IRgbColor ' 設定顏色
```

```
Set pRedColor = New RgbColor
```

```
pRedColor.Blue = 255
```

```
Set pLocationLeryFeatureSelection.SelectionColor = pRedColor
```

```
' 如何 Zoom In 至選擇的點？
```

```
pMxDoc.ActiveView.Refresh
```

```
pActiveView.PartialRefresh esriViewGraphics, Nothing, Nothing
```

```
End If
```

```
End If
```

```
Next
```

```
' 顯示搜尋結果訊息
```

```
If ResultMsg = "" Then
```

```
MsgBox "查無資料!"
```

```
Else
```

```
MsgBox ResultMsg
```

```
End If
```

```
End Sub
```

```
Private Sub btnLocate_Click ()
```

```
' Zoom In 至該點
```

```
'Dim pLocationLeryFeatureSelection As IEnumFeature
```

```
'Set pLocationLeryFeatureSelection = pMxDoc.FocusMap.FeatureSelection
```



```
Dim pSelSet As ISelectionSet
Set pSelSet = pLocationLeryFeatureSelection.SelectionSet 'bug
```

```
Dim pEnumGeom As IEnumGeometry
Dim pEnumGeomBind As IEnumGeometryBind
```

```
Set pEnumGeom = New EnumFeatureGeometry
Set pEnumGeomBind = pEnumGeom
```

```
pEnumGeomBind.BindGeometrySource Nothing, pSelSet
```

```
Dim pGeomFactory As IGeometryFactory
Set pGeomFactory = New GeometryEnvironment
```

```
Dim pGeom As IGeometry
Set pGeom = pGeomFactory.CreateGeometryFromEnumerator (pEnumGeom)
```

```
pMxDoc.ActiveView.Extent = pGeom.Envelope
pMxDoc.ActiveView.Refresh
```

```
End Sub
```

```
Private Sub btnReadTxt_Click ()
```

```
' 重置元件狀態
```

```
lbxKeywordSpatial.Clear
```

```
lbxKeywordAttributeUnLoad.Clear
```

```
lbxKeywordAttributeLoaded.Clear
```

```
lbxSpatial.Clear
```

```
lbxAttributeUnLoad.Clear
```

```
lbAttributeLoaded.Clear
```

```
' 讀取「語意推論結果.txt」文字檔
```

```
F = FreeFile
```

```
Open SemanticResultFilePath For Input As #F
```

```
Dim StrTmp As String
```

```
Dim StrEchoTmp As String
```

```
Dim LineNumber As Integer
```

```
LineNumber = 0
```

```
Do
```

```
Line Input #F, StrTmp
```

```
If VBA.Left (StrTmp, 3) = "空間詞" Then
```

```
StrEchoTmp = VBA.Right (StrTmp, VBA.Len (StrTmp) - 4)
```

```
lbSpatial.AddItem StrEchoTmp
```

```
ElseIf VBA.Left (StrTmp, 3) = "屬性詞" Then
```

```
StrEchoTmp = VBA.Right (StrTmp, VBA.Len (StrTmp) - 4)
```

```
lbAttributeUnLoad.AddItem StrEchoTmp
```

```
End If
```

```
Loop Until EOF (F)
```

```
' 讀取「詞彙庫比對結果.txt」文字檔
```

```
F1 = FreeFile
```

```
Open KeywordMatchResultFilePath For Input As #F1
```

```
Dim StrTmp1 As String
```

```
Dim StrEchoTmp1 As String
```

```
Dim LineNumber1 As Integer
```

```
LineNumber1 = 0
```

```
Do
```

```

Line Input #F1, StrTmp1
If VBA.Left (StrTmp1, 3) = "空間詞" Then
StrEchoTmp1 = VBA.Right (StrTmp1, VBA.Len (StrTmp1) - 4)
' 以逗點分隔多個詞彙
If True Then
If VBA.InStr (StrEchoTmp1, ",") > 0 Then
Dim StrEchoArr As Variant
StrEchoArr = VBA.Split (StrEchoTmp1, ",")
For i = 0 To UBound (StrEchoArr)
lbxKeywordSpatial.AddItem StrEchoArr (i)
Next i
Else
lbxKeywordSpatial.AddItem StrEchoTmp1
End If
ElseIf VBA.Left (StrTmp1, 3) = "屬性詞" Then
StrEchoTmp1 = VBA.Right (StrTmp1, VBA.Len (StrTmp1) - 4)
lbxKeywordAttributeUnLoad.AddItem StrEchoTmp1
End If
Loop Until EOF (F1)
End Sub

```



```

Public Sub PolygonsOverlay ()
Dim pMxDoc As IMxDocument
Dim pFLayer As IFeatureLayer
Dim pFCursor As IFeatureCursor
Dim pFeature As IFeature
Dim i As Long
Dim pTopoOP As ITopologicalOperator
Dim pPolygonA As IPolygon
Dim pPolygonB As IPolygon

```



```

Dim pPolygonC As IPolygon
Dim pElement As IElement
Dim pSimpleFillSymbol As ISimpleFillSymbol
Dim pSimpleLineSymbol As ISimpleLineSymbol
Dim pRGBColor As IRgbColor
Dim pGC As IGraphicsContainer

Set pMxDoc = ThisDocument

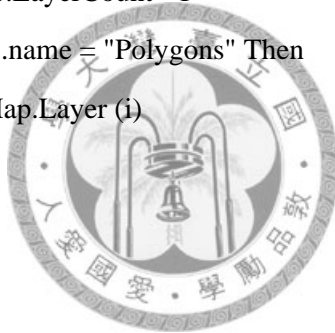
' 設定 intersect 的來源為兩個 Shape
' -----
'設定 pFLayer 變數為 name = Polygons 的圖層
For i = 0 To pMxDoc.FocusMap.LayerCount - 1
If pMxDoc.FocusMap.Layer (i) .name = "Polygons" Then
Set pFLayer = pMxDoc.FocusMap.Layer (i)
End If
Next i

If pFLayer Is Nothing Then
MsgBox "can't find Polygons layer"
Exit Sub
End If

' 設定 pPolygonA、pPolygonB 為前兩個 Shape
Set pFCursor = pFLayer.FeatureClass.Search (Nothing, False)
Set pFeature = pFCursor.NextFeature

i = 0
Do Until pFeature Is Nothing
i = i + 1

```



If i = 1 Then

Set pPolygonA = pFeature.Shape

End If

If i = 2 Then

Set pPolygonB = pFeature.Shape

End If

Set pFeature = pFCursor.NextFeature

Loop

' 取得 intersect (交集)

Set pElement = New PolygonElement

Set pTopoOP = pPolygonA

Set pPolygonC = pTopoOP.Intersect (pPolygonB, esriGeometry2Dimension)

' 改變樣式為 esriSLSSolid

' -----

pElement.Geometry = pPolygonC

Set pRGBColor = New RgbColor

pRGBColor.Red = 0

pRGBColor.Green = 0

pRGBColor.Blue = 0

' 設定 SimpleLineStyleSymbol 為 esriSLSSolid

Set pSimpleLineStyleSymbol = New SimpleLineStyleSymbol

pSimpleLineStyleSymbol.Width = 2

pSimpleLineStyleSymbol.Color = pRGBColor



```
pSimpleLineSymbol.Style = esriSLSSolid
```

```
' 設定 SimpleFillSymbol 為 esriSLSSolid
```

```
Set pSimpleFillSymbol = New SimpleFillSymbol
```

```
pSimpleFillSymbol.Style = esriSFSSolid
```

```
Set pRGBColor = New RgbColor
```

```
pRGBColor.Red = 255
```

```
pRGBColor.Green = 234
```

```
pRGBColor.Blue = 190
```

```
pSimpleFillSymbol.Color = pRGBColor
```

```
pSimpleFillSymbol.Outline = pSimpleLineSymbol
```

```
Dim pFillShapeElement As IFillShapeElement
```

```
Set pFillShapeElement = pElement
```

```
pFillShapeElement.Symbol = pSimpleFillSymbol
```

```
Set pGC = pMxDoc.FocusMap
```

```
pGC.AddElement pElement, 0
```

```
pMxDoc.ActiveView.Refresh
```

```
End Sub
```

```
Public Sub BufferFeatures (strBufferDistance As String)
```

```
'說明： 此為畫 buffer 的函式
```

```
'用法： Step1: 選取某點、線、面
```

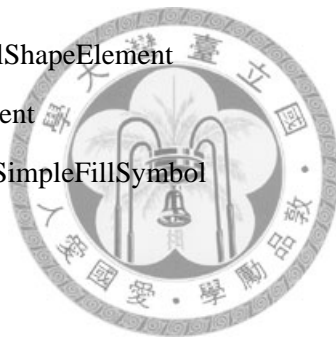
```
' Step2: 選擇距離來畫 buffer
```

```
Dim pMxDoc As IMxDocument
```

```
Dim pActiveView As IActiveView
```

```
Dim pGraphicsContainer As IGraphicsContainer
```

```
Dim pEnumFeature As IEnumFeature
```



```

Dim pFeature As IFeature
Dim pTopoOP As ITopologicalOperator
Dim pElement As IElement

Set pMxDoc = Application.Document
Set pActiveView = pMxDoc.FocusMap
Set pGraphicsContainer = pMxDoc.FocusMap

'Verify there is a feature selection
If pMxDoc.FocusMap.SelectionCount = 0 Then Exit Sub
If strBufferDistance = "" Or Not IsNumeric (strBufferDistance) Then Exit Sub

'Buffer all the selected features by the BufferDistance
'and create a new polygon element from each result
Set pEnumFeature = pMxDoc.FocusMap.FeatureSelection
pEnumFeature.Reset
Set pFeature = pEnumFeature.Next
Do While Not pFeature Is Nothing
Set pTopoOP = pFeature.Shape
Set pElement = New PolygonElement
pElement.Geometry = pTopoOP.Buffer (CInt (strBufferDistance) )

' 改變 buffer 的 renderer 如何寫？

' zoom in 至該位置
Dim pBufferEnv As IEnvelope
Set pBufferEnv = pElement.Geometry.Envelope
pBufferEnv.Expand 5, 5, True 'kimi 0980604 此時已定位完成，只差 refresh
pMxDoc.ActiveView.Extent = pBufferEnv

```



'kimi 0980604 以下為 Renderler 寫法

```
Dim pSimpleFillSymbol As ISimpleFillSymbol
```

```
Dim pSimpleLineSymbol As ISimpleLineSymbol
```

```
Dim pRGBColor As IRgbColor
```

```
Set pRGBColor = New RgbColor
```

```
pRGBColor.Red = 0
```

```
pRGBColor.Green = 0
```

```
pRGBColor.Blue = 0
```

```
Set pSimpleLineSymbol = New SimpleLineSymbol
```

```
pSimpleLineSymbol.Width = 2
```

```
pSimpleLineSymbol.Color = pRGBColor
```

```
pSimpleLineSymbol.Style = esriSLSSolid
```



```
Set pSimpleFillSymbol = New SimpleFillSymbol
```

```
pSimpleFillSymbol.Style = esriSFSSolid
```

```
Set pRGBColor = New RgbColor
```

```
pRGBColor.Red = 255
```

```
pRGBColor.Green = 234
```

```
pRGBColor.Blue = 190
```

```
pSimpleFillSymbol.Color = pRGBColor
```

```
pSimpleFillSymbol.Outline = pSimpleLineSymbol
```

```
Dim pFillShapeElement As IFillShapeElement
```

```
Set pFillShapeElement = pElement
```

```

pFillShapeElement.Symbol = pSimpleFillSymbol
pGraphicsContainer.AddElement pElement, 0 'kimi 0980604 新增至 GraphicsContainer
Set pFeature = pEnumFeature.Next
Loop

```

```

pActiveView.Refresh 'kimi 0980604 refresh

```

```

End Sub

```

```

Private Sub lbxAttributeLoaded_MouseUp (ByVal Button As Integer, ByVal Shift As Integer,
    ByVal X As Single, ByVal Y As Single)

```

```

    If lbxAttributeLoaded.ListCount > 0 Then

```

```

        ' 將圖層由地圖中移除

```

```

        Dim LoadedArr As Variant

```

```

        LoadedArr = VBA.Split (lbxAttributeLoaded.Text, ",")

```

```

        RemoveLayer (LoadedArr (UBound (LoadedArr))) ' 移除指定的 Layer

```

```

        ' 將「已載入圖層」中的項目移至「未載入圖層」中

```

```

        lbxAttributeUnLoad.AddItem lbxAttributeLoaded.Text

```

```

        lbxAttributeLoaded.RemoveItem lbxAttributeLoaded.ListIndex

```

```

    End If

```

```

End Sub

```

```

Private Sub lbxAttributeUnLoad_MouseUp (ByVal Button As Integer, ByVal Shift As Integer,
    ByVal X As Single, ByVal Y As Single)

```

```

    If lbxAttributeUnLoad.ListCount > 0 Then

```

```

        ' 將圖層加到地圖中

```

```

        Dim UnLoadArr As Variant

```

```

        UnLoadArr = VBA.Split (lbxAttributeUnLoad.Text, ",")

```



```
LoadLayer (UnLoadArr (UBound (UnLoadArr))) ' 載入指定的 Layer
```

```
' 將「未載入圖層」中的項目移至「已載入圖層」中
```

```
lboxAttributeLoaded.AddItem lboxAttributeUnLoad.Text
```

```
lboxAttributeUnLoad.RemoveItem lboxAttributeUnLoad.ListIndex
```

```
End If
```

```
End Sub
```

```
Private Sub lboxSpatial_Click () ' 抽離空間詞彙
```

```
Dim ArrEchoTmp As Variant
```

```
Dim LastStr As String
```

```
ArrEchoTmp = VBA.Split (lboxSpatial.Text, ",")
```

```
LastStr = ArrEchoTmp (UBound (ArrEchoTmp)) '取得最後一個空間詞彙
```

```
Call FindAndSelect (LastStr) ' 尋找並選擇空間詞彙
```

```
'Call BufferFeatures ("50") ' 畫 buffer
```

```
End Sub
```

```
Private Sub LoadLayer (strLayerName As String) ' 載入 Layer
```

```
Dim pMxDoc As IMxDocument
```

```
Dim pMap As IMap
```

```
Dim pGxLayer As IGxLayer
```

```
Dim pGxFile As IGxFile
```

```
Dim LayerLoadedTag As Boolean
```

```
Dim pLayer As ILayer
```

```
Set pMxDoc = ThisDocument
```

```
Set pMap = pMxDoc.FocusMap
```

```
Set pGxLayer = New GxLayer
```

```
Set pGxFile = pGxLayer
```

```
pGxFile.Path = LayerFolderPath & strLayerName & ".lyr"
```

```
' 檢查該 Layer 是否已被載入
```

```
LayerLoadedTag = False
```

```
For i = 0 To pMap.LayerCount - 1
```

```
Set pLayer = pMap.Layer (i)
```

```
If pLayer.name = strLayerName Then
```

```
LayerLoadedTag = True
```

```
End If
```

```
Next
```

```
If LayerLoadedTag Then
```

```
MsgBox strLayerName & " 已存在", vbOKOnly, "test"
```

```
Else
```

```
pMap.AddLayer pGxLayer.Layer
```

```
pMap.MoveLayer pMap.Layer (0), pMap.LayerCount - 1
```

```
End If
```

```
pMxDoc.ActiveView.Refresh
```

```
End Sub
```

```
Private Sub RemoveLayer (strLayerName As String) ' 移除 Layer
```

```
Dim pMxDoc As IMxDocument
```

```
Dim pMap As IMap
```

```
Dim pGxLayer As IGxLayer
```

```
Dim LayerLoadedTag As Boolean
```

```
Dim LayerLoadedIndex As Integer
```

```
Dim pLayer As ILayer
```

```
Set pMxDoc = ThisDocument
```




```
Set pMap = pMxDoc.FocusMap
```

```
LayerLoadedTag = False
```

```
For i = 0 To pMap.LayerCount - 1
```

```
Set pLayer = pMap.Layer (i)
```

```
If pLayer.name = strLayerName Then
```

```
LayerLoadedTag = True
```

```
LayerLoadedIndex = i
```

```
End If
```

```
Next
```

```
If LayerLoadedTag Then
```

```
pMap.DeleteLayer pMap.Layer (LayerLoadedIndex)
```

```
Else
```

```
MsgBox strLayerName & " 不存在", vbOKOnly, "test"
```

```
End If
```

```
pMxDoc.ActiveView.Refresh
```

```
End Sub
```

