

國立臺灣大學文學院圖書資訊學系

碩士論文



Department of Library and Information Science

College of Liberal Arts

National Taiwan University

Master Thesis

以任務佇列模式設計整合替代計量分析系統

The Design and Implementation of an Integrated Altmetrics Analysis System

Based on Task Queue Model

董冠伯

Kuan-Po Tung

指導教授：陳光華 博士

Advisor: Kuang-hua Chen, Ph.D.

中華民國 105 年 1 月

Jan, 2016



誌謝



能夠完成這篇論文，首先要感謝指導老師陳光華教授，在一開始找我參與了 Digital Humanities 計畫，讓我有機會接觸到文獻計量分析與系統設計。隨著計畫的進行，我的論文主題也呼之欲出，讓我不用在主題的選擇上耗費太多心力。同時也感謝指導教授非常有耐心的在我課業與工作皆忙碌，進度非常緩慢的狀況下，逐步指導我將系統設計出來，並完成相關研究數據。也感謝口試委員唐牧群教授與楊東謀教授，在口試當下提供了許多建議，讓論文更加完善。尤其楊東謀教授在軟體工程與開發方法上的不吝指導，讓我對軟體開發模式有更深入的了解。

另外，還要感謝在計畫中一同努力的幾位同學，分別是鈞湄、芊儒、弼心與舒軒，若不是他們在計畫中所整理的會議紀錄與分析資料，在面對龐大的系統與程式之下，會更難消化這麼大量的分析數據。

當然還有因為過度忙碌沒有太多機會見面的家人，以及許多朋友、同事，公司老闆等，在我為了論文必須來回學校奔波的同時，替我照顧了許多我無法顧及的事務，讓我無後顧之憂的完成學業。

最後，則要感謝改變我人生的賴毓敏學長，帶領我進入全新未知的領域，並在我需要同時顧及學業與事業時，帶我進入現在的公司，讓我能夠適當的分配時間在工作與論文上。感謝大家。



中文摘要



替代計量學 (altmetrics) 是文獻計量分析領域的新興研究領域，用以取代傳統書目計量學 (bibliometrics) 與資訊計量學 (informetrics) 在資料來源上的限制，其整合了網路計量學的優點與特性，將資料來源擴展至許多非正式的學術活動行為，讓學者的各項網路活動、討論交流、課程與程式產出等等也能列入統計。可以用來補足傳統計量學只能研究專業資料庫所收錄之文獻，且更新與散播速度較慢的問題。雖然國外已經有許多 Altmetrics 分析工具，但這些工具對國內在地化研究資訊分析能力的不足，需要一個能夠以中文作者姓名來分析其著作影響力的系統。本研究旨在探索 Altmetrics 適合用於國內學者網路影響力的資料來源與類型，並根據這些分類打造中文的替代計量分析系統，補充國外替代計量分析系統的不足，作為探討學者與其著作影響力的參考資源。由於此分析系統必須適應各類不同的網站平台，並抓取對應的資料作分析，而這些資料都是非結構化的，且分析的規則也各不相同。為此必須設計一套能夠適應多種環境，並且容易擴充新資料來源的架構。在本研究中，將採用任務佇列模式，藉由多組任務列隊依次執行並指派新任務，來達到彈性擴充的目的。最後所設計出來的系統擁有優良的 GUI 介面，並且分析內容與人工分析的差異不大，有一定程度的參考價值，足以取代大量人工查詢的工作。在未來，期許能有更多的研究，在本研究之上，設計出更多的資料分析類型，與更精確的分析演算法，讓國內學者的努力被更多人看見。

英文摘要



Altmetrics is a new research field which to resolve the limit of traditional Bibliometrics and Informetrics. It's integrated benefits and features from Webometrics, extends the data source from many informal activities of academic and made all activities of internet, discussion, courses and code that can be included to the statistic. It can be used to solve the problem of traditional Informetrics which is too slow on spreading and only includes data from professional database. Although there are many altmetrics tools in foreign country, but they are limited on analysis local information, we need a new system to analysis the influences of scholars through Chinese name. The subject of this thesis is that to explore the data source and data type to analysis the influence of local scholars, design a Chinese Altmetrics analysis system based on the result of research and covers the shortage of the tools in foreign country on researching local scholars. This system must adapt many different websites/platforms and fetch corresponding data, but these data are unstructured, and must analysis by different rules. We must design a structure which is able to adapt to many environments and easily to extend new data source. In this thesis, will use task queue pattern, execute and assign tasks by many tasks in a list. In the result, we created a system with well GUI, and there are very few differences between system auto analysis and manual analysis. It has something referential value and is available to replace the manual analysis. We can expect there will be more research to find more data type and develop more algorithm to make the effort of local scholars be known by more people.

目次



誌謝.....	III
中文摘要.....	V
英文摘要.....	VI
目次.....	VII
表次.....	IX
圖次.....	XI
第一章 緒論.....	1
第一節 問題陳述.....	1
第二節 研究目的.....	4
第三節 研究範圍與限制.....	4
第四節 名詞解釋.....	5
第二章 文獻分析.....	7
第一節 資訊計量學.....	7
第二節 非傳統計量方式.....	10
第三節 現有的替代計量學分析工具.....	16
第四節 軟體開發流程.....	23
第三章 研究方法與步驟.....	31
第一節 系統設計與任務流程.....	31
第二節 資料來源.....	32
第三節 分析資料類型.....	33
第四節 前導研究.....	35
第四章 系統實作.....	39

第一節	開發平台選擇.....	39
第二節	介面規劃.....	41
第三節	任務結構與流程.....	42
第四節	任務流程實作.....	57
第五節	實際畫面導覽.....	72
第五章	結論與未來展望.....	81
第一節	結論與貢獻.....	81
第二節	建議.....	82
參考文獻.....		85



表次



表 3-1：Altmetrics 分析資料來源統整.....	33
表 3-2：資料過濾規則.....	33
表 3-3：實驗進行時的資料來源與類型.....	35
表 3-4：實驗用學者名單與編號.....	36
表 3-5：數據比對一覽.....	36
表 4-1：課程大綱任務流程.....	44
表 4-2：報章雜誌任務流程.....	45
表 4-3：維基百科任務流程.....	46
表 4-4：社群網站任務流程.....	48
表 4-5：Google Scholar 任務流程	49
表 4-6：學位論文被引次數任務流程.....	51
表 4-7：學位論文指導次數任務流程.....	51
表 4-8：TCI 被引次數任務流程.....	53
表 4-9：TCI 著作篇數任務流程.....	53
表 4-10：WOS 任務流程	55
表 4-11：Scopus 任務流程	56
表 4-12：Mendeley 任務流程	56
表 4-13：課程大綱任務流程	58
表 4-14：佇列儲存的訊息參數	60
表 4-15：課程大綱系統設計表	63
表 4-16：報章雜誌系統設計表	64

表 4-17：維基百科系統設計表	65
表 4-18：社群媒體系統設計表	66
表 4-19：Google Scholar 系統設計表	67
表 4-20：學位論文系統設計表	68
表 4-21：TCI HSS 系統設計表	69
表 4-22：Web of Science 系統設計表	70
表 4-23：Scopus 系統設計表	71
表 4-24：Mendeley 系統設計表	72



圖次



圖 2-1：Cave 的分類體系	14
圖 2-2：Konkiel 的分類體系	15
圖 2-3：Lin 與 Fenner 的分類體系	15
圖 2-4：Altmetric.com 的甜甜圈表示法與 Facebook 上發現的引用	17
圖 2-5：Impactstory 的彙整頁面	18
圖 2-6：ALM Reports 入口頁面	19
圖 2-7：文章頁面的 Metrics 分頁	20
圖 2-8：PLUMX 的 Altmetrics 分析結果	21
圖 2-9：PaperCritic 首頁搜尋介面	22
圖 2-10：Publish or Perish 系統畫面	22
圖 2-11：瀑布流的開發階段	25
圖 3-1：系統查詢任務流程	31
圖 3-2：人工總和與系統總和折線圖	37
圖 4-1：Joomla.org 網站首頁	40
圖 4-2：Joomla CMS 後台操作介面	40
圖 4-3：查詢介面首頁	41
圖 4-4：查詢結果整合顯示介面	42
圖 4-5：Google Scholar 搜尋結果的被引用次數	49
圖 4-6：博碩士論文聯邦查詢系統的搜尋筆數結果	51
圖 4-7：華藝的搜尋筆數結果	52

圖 4-8：TCI 每筆搜尋結果的被引用次數顯示.....	53
圖 4-9：TCI 搜尋結果總筆數.....	54
圖 4-10：查詢結果整合顯示介面	61
圖 4-11：系統首頁	73
圖 4-12：搜尋結果頁面	73
圖 4-13：詳細的搜尋結果列表	74
圖 4-14：課程大綱頁面總覽	75
圖 4-15：報章雜誌頁面總覽	75
圖 4-16：社群媒體頁面總覽	76
圖 4-17：維基百科頁面總覽	76
圖 4-18：單一頁面詳細資料	77
圖 4-19：搜尋條目管理	77
圖 4-20：任務管理	78
圖 4-21：搜尋引擎頁面管理	78
圖 4-22：搜尋結果頁面管理	79
圖 4-23：Webometrics 網址快取管理.....	79
圖 4-24：佇列管理	80

第一章 緒論



第一節 問題陳述

替代計量學 (altmetrics)，是文獻計量分析的新興研究領域，其概念最早來自書目計量學 (bibliometrics)。普里查德 (Pritchard) 於 1996 年提出書目計量學一詞，並定義為：「應用數學與統計方法至書或其他形式的文字傳播」，意味著書目計量是一種測量的技術。書目計量學被認為是一種量化科學，可分為描述型 (descriptive) 與評估型 (evaluative)。前者用於學術生產力的計算，例如某學科領域中論文、圖書、期刊等等的出版量或索引數量。後者則用於計算文獻被參考、引用的情形 (蔡明月，2003)。同時間，也有許多學者提出不同的文獻計量理論，例如圖書館計量 (librametrics) 或科學計量 (scientometrics) 等。書目計量與科學計量的目的各有重疊，無法明確加以區分。例如蔡明月 (2004) 便提到，多數書目計量的研究成果皆刊載於《科學計量學》(Scientometrics) 期刊，而科技研究的具體成果也是以「文獻」(literature) 為主要的呈現形式，這意味著無法將兩者明確區隔開來。

隨後，納基 (Nacke) 在 1979 年提出了資訊計量 (informetrics)，其意義是將數學的方法應用到資訊現象的測量。後來納基更建議將科學計量作為資訊計量的附屬領域，並將資訊計量置於資訊科學的領域內 (Nacke, 1979)。資訊計量建立在電腦科學逐漸成熟的環境，相較於書目計量只能應用於小規模的文獻集合，以分析其中的文獻特性與引用現象，資訊計量更能夠大規模的統計多個索引資料庫的文獻，甚至直接分析全文。實際上，資訊計量兼容了書目計量與科學計量，三者是息息相關的，不只可以統計學科貢獻程度，更能夠對未來的科學發展政策做出建議。當然，也各自有獨立的分支領域，例如書目計量更著重在出版文獻的分析，而科學計量可用於統計科學發展的活躍程度等 (蔡明月，2003、2004)。



到了 1997 年，當網路開始蓬勃發展的時候，網路計量（webometrics）被提出來。作為資訊計量的一個分支，網路計量同樣著重在取得學術活動的量化數據，但更進一步包含了一切在網路上的活動與資訊傳播的狀態。網路計量的研究層面包含了網路連結、網路內容、網路探勘（Web mining）與網路影響因素（Web Impact factor）等（蔡明月，2002）。對當時已有數十年歷史的資訊計量而言，網路計量無疑的替其帶來新的生命力，當然也面臨新的問題。首先網路本身含有大量非結構化資訊，網路計量必須依賴於搜尋引擎的檢索功能，若搜尋引擎無法提供足量資訊，則資訊蒐集就會遇到困難或者受到影響。再者，網路上的引用動機也不全然與過去的文獻引用相同，也可能出現大量交互引用與連結，對於引用效力的判斷也會遇到困難。最後，由於任何人皆可建立網站，編寫文章，使得網路上的資訊很難判斷擁有多少價值。總而言之，網路計量帶來了更大規模、更快速、更即時的分析能力，但其結果的可信任度也受到不少質疑。

隨著 Web 2.0 的發展，網路上的內容生產逐漸轉為群眾導向，也使得許多人開始致力於 Web2.0 的計量研究。這些研究包含網路上的資料探勘，例如論壇、個人部落格、社群媒體等等。由於 Web 2.0 的特色，便是任何人皆可在網路平台上發表自己的非正式觀點，因此對這些網路平台的言論做分析，就形成新型態的計量學，如部落格計量、Wiki 計量、Facebook 計量等等。而這些新型態的計量學，也就成為替代計量學（altmetrics）興起的基礎。

Altmetrics 全名為 Alternative Metrics。中國學者翻譯為「替代性計量學」或「選擇性計量學」，臺灣則有人稱之為「另類計量學」。本研究，取 alternative 有「替代」、「額外的選擇」之意，統一稱其為「替代計量學」。在替代計量提出前，學者們通常以 ALMs（Article-Level Metrics）為主要概念去測量單篇文獻的影響力。2010 年，Priem 在自己的 Twitter 上使用 altmetrics 一詞來補足原本 ALMs 的不足。相較於 ALMs 以測量單篇文獻為主，Altmetrics 融合了多種新的資料來源，可以測量文獻、期刊或學者的影響力，不只涵括了 ALMs，還能夠提供更多

應用。短短幾年間，altmetrics 獲得大量的關注，許多學者、出版商與評鑑者都開始探討這門新學問，網路上也出現大量討論文章。此潮流被稱為 Altmetrics 運動 (Altmetrics movement) (邱均平、余厚強，2013)。一直到 2013 年，如 Public Library of Science (PLOS) 與 Elsevier 等機構也公開支持 Altmetrics，國際間也舉辦了越來許多專門的研討會，使得 Altmetrics 逐漸成為一門重要的研究領域。

作為 Altmetrics 分析標的的常用資料來源，有 Elsevier 旗下的 Mendeley (<http://www.mendeley.com>)，專門提供參考文獻管理工具、以及由 12 家學術出版商共同成立的 Cross Reference (<http://crossref.org>)，專門提供文獻引文鏈接服務、還有學術文獻書籤工具 CiteULike (<http://citeulike.org>) 和 Faculty of 1000 (<http://f1000.com>) 生命科學論文評價系統等等，並不限於任何特定領域，只要有使用者能夠發表評論、互相交流、收藏或引用任何他人著作的網站，幾乎都可作為 altmetrics 的分析來源。

目前已經提供線上服務的 altmetrics 分析工具有 Altmetric.com (<http://www.altmetric.com>)、Impactstory (<https://impactstory.org>)、PLOS Article Level Metrics (<http://article-level-metrics.plos.org>) 與 PLUMX (<http://plu.mx>) 等等，皆用各自的方法去分析網路上各個資料來源以取得 Altmetrics 分析結果。但這些網站主要皆以全球範圍與英文資訊為主，因此英文文獻較容易被這些分析工具收錄，若學者發表的是在地化的研究內容，或文獻以母語寫成，現有的工具無法衡量其價值。

有鑑於國外提供的 Altmetrics 分析工具對國內在地化研究資訊分析能力的不足，需要一個能夠以中文作者姓名來分析其著作影響力的系統，除了從一般的 Altmetrics 資料來源如 Google、社群媒體與 Web of Science 等等之外，還要能夠分析國內期刊文獻資料庫的內容。本研究旨在探索 Altmetrics 適合用於國內學者網路影響力的資料來源與類型，並根據這些分類打造中文的替代計量分析系統，補充國外替代計量分析系統的不足，作為探討學者與其著作影響力的參考資源。

第二節 研究目的



本研究以建構國內學者在地化資料分析為主的中文整合替代計量分析系統，研究目的為：

- 一、彌補國際現有 Altmetrics 工具的不足，提供在地化學術分析數據。
- 二、探索適合用於國內 Altmetrics 分析的資料來源與類型。
- 三、建立整合替代計量分析系統的操作流程，並可在單一介面顯示所有的分析數據。

由於可用的搜尋引擎與資料來源不一定都有提供完整的 API，操作介面也不盡相同，本研究開發的系統所面臨的研究問題包含：

- 一、如何根據不同類型的資料來源網站與搜尋引擎制定合適的分析策略。
- 二、如何使單一系統能夠適應不同網站、搜尋引擎與分析數據的邏輯，並將這些數據整合在單一介面顯示。
- 三、需設計高度可維護化（Maintainability）、可擴充化（Extendibility）的程式架構，當需要增加新的資料來源與數據類型時，無需回頭修改舊的程式碼。

第三節 研究範圍與限制

本研究的研究限制如下：

- 一、本研究蒐集之參考文獻，以國內可取得之中、英文文獻為限。
- 二、本研究分析之資料來源，以 Google 搜尋引擎有索引且程式系統可正常訪問取得之頁面資料，或者該網站本身提供 API 查詢為限。

第四節 名詞解釋



(1) Bibliometrics

中文又稱「書目計量學」，旨在以統計學方法，分析學者的學術活動、被引次數與出版文獻的成長和分布狀況。

(2) Scientometrics

中文為「科學計量學」，與書目計量學很相似，但主要研究科學活動的成長與分布狀態，通常用在提供科學發展政策制定的參考。

(3) Informetrics

中文稱「資訊計量學」，主要依賴電腦科學的發展所帶來的運算能力，大範圍計算資料庫文獻，取得學術活動的分佈狀況。由於其與書目計量學和科學計量學的分析目標有大量重疊，目前主流以「資訊計量學」作為文獻計量分析研究的總和名詞。

(4) Webometrics

中文稱「網路計量學」，是資訊計量學在網路環境的延伸，藉由 World Wide Web 網頁相互超連結的特性，分析網頁內容和引用的資料數據。

(5) Altmetrics

Altmetrics 是以網路為媒介，分析學者在非正式學術交流中的貢獻度與被討論程度。與網路計量與資訊計量的最大差別在於其著重學者在各種網路社群媒體上的發言、被談論與被收藏次數等等，許多數據類型都是網路時代才有的新式數據（例如書籤收藏次數、軟體被下載次數等）。本研究以「替代計量學」為中文翻譯。

(6) Software Engineering

中文為「軟體工程」，探討軟體製作過程、管理架構、團隊合作等等的一門學問，旨在以科學化的管理方式建立標準軟體開發流程的一門學問。



(7) Task Queue

Queue 的中文稱「佇列」或「列隊」，是軟體技術名詞，描述一組資源在儲存器中依序存入並取出，其原理與日常生活中的「排隊」類似，有「先進先出」的概念，與 Queue 概念相反者為先進後出的 Stack（堆疊）。Task Queue（任務佇列）則是描述一個程式有許多任務要進行時，將任務先後順序安排成一個佇列，再依序執行的過程。

第二章 文獻分析



本章將探討從 1970 年代起發展的傳統的書目計量學與資訊計量學，了解文獻計量分析的歷史、由來，還有常見的指標。接著將說明 1990 年代以後興起的新興計量方式，我們將這些計量方式稱為非傳統計量學，與 1990 年代之前的傳統計量學做出區隔。

第一節 資訊計量學

由於替代計量學是建立在資訊計量學的基礎上，而資訊計量學最早的前身又是從書目計量學開始，因此本小節將從傳統計量學中的書目計量學開始探討，了解其目的與形成的背景，慢慢推展到後來的科學計量學與資訊計量學，釐清文獻計量分析這門學問的發展脈絡。

一、書目計量學

在蔡明月(2003)的《資訊計量學與文獻特性》一書中提到，最早在 1917 年，柯爾 (Cole) 與伊勒斯 (Eales) 就曾經依靠統計方法來計算文獻成長的情形，他們比較解剖學在 1543 到 1860 年間的出版品成長狀況。到了 1923 年，休姆(Hulme) 使用書目統計 (statistical bibliography) 作為評估現代文明發展的量化技術。圖書館學大師阮加納桑 (Ranganathan) 則提出圖書館計量學 (librametrics) 一詞作為分析圖書館運作的量化研究工具。現在統稱的書目計量學 (bibliometrics)，則是由普里查德 (Pritchard) 在 1969 年提出，其定義為：應用數學與統計方法至書或其他形式的文字傳播。也因為書目計量學的提出，讓這門學問脫離了過去單純被歸類在統計學下的文獻分析研究。

隨後，越來越多對書目計量學的新定義被提出，試圖完善這門新興的科學。

費耳松尼 (Fairthorne) 擴大書目計量學的定義為：對記錄文獻的特性及與其有關之行為的量化研究。英國標準局定義為：運用統計和數學的方法來研究文獻的使用及出版的形式。雖然許多學者各自給予了書目計量學不同的定義，但大抵上仍脫離不了是藉由科學化的統計技術來分析文獻特性與傳播狀況的一門學問。

書目計量學的範圍也各有不同的定義，例如尼古拉 (Nicholas) 與瑞奇 (Ritchie) 認為，書目計量學分為描述型 (descriptive) 與行為型 (behavioral)，前者描述文獻特性，後者則檢視文件內涵之間的關係 (Nicholas & Ritchie, 1978)。史蒂文斯 (Stevens) 則認為書目計量分成描述型 (descriptive) 與評估型 (evaluative)，前者用在計算學術的生產力，例如出版量、被收錄數量等；後者運用參考書目與引用文獻來計算文獻被使用的情形 (Stevens, 1983)。目前運用書目計量學探討的現象，包含文獻成長與老化、文獻生命半衰期等等，也有研究專注在分析文獻的分布情形，例如三大定律 (布拉德幅、洛特卡與齊普夫) 就各自從不同角度分析了文獻在數學上的分佈特性。另外，引用分析也是很重要的研究領域，例如被引、自引、共被引等等指標，能夠用來評估文獻的價值或者學者的貢獻程度。

二、科學計量學

科學計量學是由杜部洛夫 (Dobrov) 與克倫諾伊 (Karennoi) 於 1969 年提出的，其定義為：資訊科學處理的測量 (measurement of informetrics process)。不過科學計量學真正受到廣泛關注，則要到 1978 年，布勞恩 (Braun) 發行《科學計量學》(Scientometrics) 期刊才開始。原先布勞恩只是打算作為一個國際學術交流的空間，但這份期刊日後卻大大超出原先設定的目標，並與社會學和經濟學等領域形成密切聯繫 (蔡明月，2003)。

相較於書目計量學，桑古特 (Sengupta) (1985) 認為科學計量學更強調於科學研究活動的成長與傳播等面相，例如生產力和研究進度等。莫拉夫西克 (Moravcsik) 則認為科學計量學是一門科際整合的學科，包含許多傳統學科而非

局限於單一窄化的主題 (Moravcsik, 1985)。整體而言，科學計量學可用於分析學術研究活動的熱絡程度，協助制定政策。例如貝克 (Beck) 就認為，科學計量學能夠幫助科學管理成功的執行，並平衡研究經費的分配，增加研究效率等等 (Beck, 1978)。



三、資訊計量學

資訊計量學 (informetrics) 最早由納基 (Nacke) 在 1979 年以德文 *informetrie* 提出，意義為以數學方式應用到資訊現象的測量，並建議將科學計量學納入資訊計量學的範圍，而資訊計量學則納入資訊科學的範圍 (Nacke, 1979)。Rajan (1985) 出任國際文獻聯盟資訊計量學委員會時，定義資訊計量學為：匯集各種資訊生產力的測量研究。蔡明月 (2004) 也在比較三大計量學時，提出資訊計量學是測量資訊的使用、流動及其產生的影響。Brookes (1990) 則主張可以把書目計量學與科學計量學併入資訊計量學，以便更有效的掌握電腦化資訊系統的發展。

資訊計量學的應用層面較為廣泛，莫拉萊斯 (1985) 認為，資訊計量學的應用包含文獻的量化成長、資訊的分佈與退化、資訊產物與服務的效率、資訊系統和資訊建設的效率、科學家的引用習慣以及引用文獻分析還有以參考書目來研究學科之間的關係等等數種應用層面。蔡明月 (2004) 則認為，由於資訊科技進步，電子化文獻大量增加，可以用電腦直接分析、量測文獻的全文，使得齊普夫定律的字詞分布研究能夠被強化，也能提昇引用分析的研究與探討。

由以上的應用範圍分析，可以看到資訊計量學與書目計量學有不少重複的地方，可以說資訊計量學其實便包含了書目計量學與科學計量學，到了 1990 年代，由於網路興起，資訊計量學開始採用網路作為統計分析的媒介，此時另外興起的網路計量學便與資訊計量學息息相關。書目計量學詞彙的出現頻率也逐漸降低，世人漸漸將資訊計量學作為多數文獻特性分析研究的總和名詞。

第二節 非傳統計量方式



本小節探討的是自 1990 年代網路興起以後流行的新興文獻計量研究方式，其中最主要的项目就是網路計量學。由於網路環境下，開始發展出許多新的學術活動行為，例如運用網站媒體進行非正式學術交流，以及將自身研究結果上傳個人網站供搜尋引擎檢索並被眾人取用等。上述的行為並非原先書目計量學能夠掌握的，也呈現出許多傳統研究方式無法分析的面向，例如研究結果如何被應用於教學上、一般大眾如何使用學術研究成果，以及呈現科學研究結果的網頁如何被利用等等（王鈞湄，2013）。由於這些指標與研究面向與過去的書目計量學有許多不同，不只更為廣闊，且對於人與人之間的非正式交流更為重視，本研究暫時將這類新興的研究方式歸類為非傳統計量，以與過去單純分析文獻與作者本身特性的書目計量學做出區隔。

一、網路計量學

二十一世紀以後，全球資訊網（World Wide Web）大幅度的影響著資訊科技發展走向，也為資訊計量學注入新的契機。這種新研究方式被稱為網路計量學（Webometrics）。網路計量學興起於 1997 年，主要是利用原本資訊計量學與科學計量學的方法去分析網路空間中的文獻分布特性、學術活動與應用情形等等。網路計量學主要探討的內容包括：網路連結、網路內容、網路探勘（web mining）與網路影響因素（web impact factor）等等。

「網路連結」探討的是網頁與網頁之間的交互參照關係。由於在網路世界中，文件是以全球資源定位器（Uniform Resource Locator--URL）來標註其定位，而網頁之間要相互連結時，也會在自己的內文中，以 URL 來標註其他參考資源的位址。當一個網頁連結其他網頁時，表示這些被連結的網頁與原始網頁是有相關的；而被連結的網頁，則是一種權威指標，被越多網頁連結的網頁，表示其內容重要性高於一般網頁。這種網頁之間的參照行為，與傳統出版品的參考

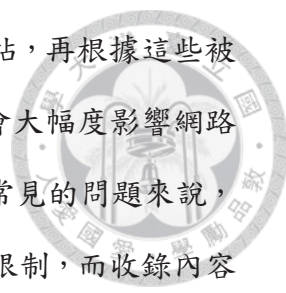
(reference) 和引用 (citation) 有非常相似的行為。因此，分析這些 URL 的交互狀態，就成為網路計量學的研究指標之一。1997 年，勞倫斯研究了六個主要的搜尋引擎，包括 Alta Vista、HotBot、Northern Light、Excite、Lycos 及 Infoseek，結果發現被索引的網頁達 32 億個（蔡明月，2003、2004），也就是在網路上已存在 32 億個 URL 能夠定位到的不同網頁，這些網頁之間的交互參照數量將會是更巨大的數字，其中也蘊含難以想像的趨勢等待被挖掘。

「網路內容」研究的是網頁的本文，例如藉由引用分析來判斷單一網頁的權威性，與網路連結的研究息息相關。同時也可以研究網頁的內連 (inlinked) 以及字詞分布研究，來得出網頁的重複性與相關性等等。也可以利用布拉德福定律來界定某個領域裡面裡面的核心網站。

「網路探勘 (web mining)」來自資料探勘 (data mining)，原本是用來分析大量非結構化資料中隱藏的趨勢與特性的工具，在網路計量學中更是發揮了重要的影響力。網路內容的成長非常快速，可以視為是一個以指數成長的分散式資料庫，且大多數都是非結構化的資訊。這種快速成長、無中心、且變化大的資訊環境正好成為網路探勘的沃土，我們可以利用網路探勘來尋找特定主題的權威網站、了解某議題的成長與演變過程或者從網路上的社群交流得出可能的趨勢分析。

Ingwersen 提出網路影響因素 (Web Impact Factor--Web-IF)，運用 Impact Factor 影響係數的概念分析網頁的影響力，並計算網頁被引用的程度與連結的強度。例如 Ingwersen 分析 Alta Vista 索引的網頁，並計算國家層級網域與個人層級網域的連結影響力，並得出國家層級網域的連結影響力高於個人層級網域 (Ingwersen, 1998)。

以上四個層面大概描述了網路計量學的主要特色與研究方向，可以看出網路計量學所研究的內容，是由人們所產出，分散在網路上的非結構化訊息，並試圖從這些訊息中分析出有價值的資訊。也由於這些分析的資料來源是高度分散且沒有結構的，檢索工具在網路計量學中變得更重要，一般來說，目前常用的檢索工



具是搜尋引擎，藉由搜尋引擎來尋找特定詞彙與主題的相關網站，再根據這些被找到的資源做分析。因此，搜尋引擎的功能性與資料完整度，會大幅度影響網路計量學研究的結果，也無法辨識真正的生產力分布狀況。以最常見的問題來說，搜尋引擎通常檢索功能較為簡單，回傳的搜尋結果有最大數量限制，而收錄內容也時常變動，只會收錄搜尋引擎自己判斷較有價值的內容，這使得研究樣本的蒐集會受到不同搜尋引擎特性的影響與限制。而被研究的對象，也就是網頁本身，也充滿變數。例如在網路時代，任何人皆可申請網域，架設自己的網站並發表文章與評論，不同的網頁之間也可以大量交互連結，這使得判斷網頁品質與知識價值變得更為困難，這都是網路計量學在研究時需要克服的。

網路時代使得學術傳播行為出現許多變化，大量的研究草稿可以在網路上即時發佈，人們也可以迅速做出非正式評論與交流討論。甚至許多正式的學術內容直接選擇網路出版而不印刷為紙本。這使得傳統的書目計量學與資訊計量學必須尋找新的研究方式與指標來判斷網路上的學術傳播，進階促成了網路計量學這門新的學問。

二、替代計量學

傳統的文獻計量方式，可以給予一個約略的統計數值來評判學者貢獻度。但在網路時代，現有的文獻計量研究方式與指標顯的不夠周延，例如時間延遲過長就是很常見的問題。科學期刊的發表週期一般在三個月到三年之間，而基於引文的文獻計量分析還要多出一個出版週期才能計算，無法即時反應學者的研究成果。另外，影響力的計算也不夠周延。許多學者的貢獻不見得只有發表學術文獻，可能有更多表現形式，例如編寫程式軟體、發表評論與部落格文章、製作教學簡報等等，單純計算文獻本身無法反應學者的貢獻價值（邱均平、余厚強，2013）。尤其到了網路計量的時代，如果繼續保守原有的研究方式與指標，以上問題顯的更加嚴重。



在這樣的氛圍下，許多學者察覺到舊有計量方式的不足，試圖探索新的研究指標。例如 PLoS 的「論文層級計量」(Article-Level Metrics--ALMs)，就是運用傳統引用次數與部分社群媒體的數據來分析單篇文獻、單本期刊或單一學者的影響力(蔡明月、曾苓莉，2014)。2010 年，Priem 最先在自己的 Twitter 上使用 altermetrics 一詞來彌補 ALMs 的局限性。隨後，連同 Taraborelli 等人在 Altmetrics 的專設網站(<http://altmetrics.org>)上發表《Alternative metrics: A manifesto》宣言，正式提出「Altmetrics」專有詞彙。Altmetrics 是 Alternative Metrics 的縮寫，一開始使用的是 alt-metrics，到了 2011 年的第二版宣言開始，移除中間的連接符，成為現在廣泛流通的 altmetrics 一詞。Altmetrics 仿照 Scientometrics、Informetrics、Webometrics 等詞彙，以 metrics 結尾，意味著是替傳統的文獻計量分析提供替代性方案。當 Altmetrics 的理念對外宣告之後，很快便引起許多相關研究者的關注，而首次 Altmetrics 研討會「altmetrics11」也於 2011 年 6 月召開，無論是學者、社群網站、學術出版商還是學術評鑑者皆對此次研討會的內容有熱烈迴響，網路上也開始出現 Altmetrics 討論群組和部落格文章。此時期 Altmetrics 研究的理論與應用以及可能遭遇的問題皆被充分討論，同時也深入探討資料收集、數據刻意操作等等已預見的問題，此潮流被稱為 Altmetrics 運動(Altmetrics movement)(邱均平、余厚強，2013)。

由於網路的特性，學術文獻的傳播變得更即時，許多文獻可以在線上即時發表，較傳統印刷出版的期刊節省更多時間。而由於文獻在網路上可以自由瀏覽、取用、評論、推薦、標註與連結等等特性，Altmetrics 具有更廣泛的同儕評鑑(peer review)特性(顧立平，2013)。Piwowar (2013) 則認為 Altmetrics 有以下四種潛在優勢：1. 對學術影響力更細微深入的分析，了解文獻被閱讀、討論、儲存與評論的情形。2. 以「日」為基礎的更為即時的計算資料。3. 了解如部落格文章、影片、軟體等網路原生(web-native)學術產出物的影響。4. 了解如學者、從業人員、臨床醫生、教育家和一般大眾等不同使用者的影響力。相較於過去的文獻

計量分析只有權威期刊或資料庫作為分析內容的來源，Altmetrics 展現了一種分散式，由下而上且即時的新計量模式，以人在網路虛擬空間的交互行為作為學術產出受重視程度與貢獻價值的判斷依據。



三、替代計量學的指標與分類體系

Altmetrics 的資料指標發展非常多元，許多學者各自提出不同的分類體系。Richard Cave (2012) 在第 32 屆 Charleston Conference 發表了《Overview of the Altmetrics Landscape》一文，認為 Altmetrics 的資料指標可以分為使用率(Usage)、擷取(Capture)、提及與談論(Mention)、社群媒體(Social Media)與引用(Citation)五種。

Cave 的分類體系



圖 2-1：Cave 的分類體系

印第安那大學圖書館館員 Konkiel (2013) 認為 Altmetrics 的指標應該要包括分享 (Shares)、儲存 (Saves)、評論 (Reviews)、改寫 (Adaptations) 與社群媒體使用統計 (Social Usage Statistics) 等五大類別。

Konkiel 的分類體系



圖 2-2：Konkiel 的分類體系

Lin 與 Fenner (2013) 則整理了 Public Library of Science (PLOS) 採用的資料指標，分別為觀看 (Viewed)、儲存 (Saved)、討論 (Discussed)、推薦 (Recommended) 及引用 (Cited) 等。

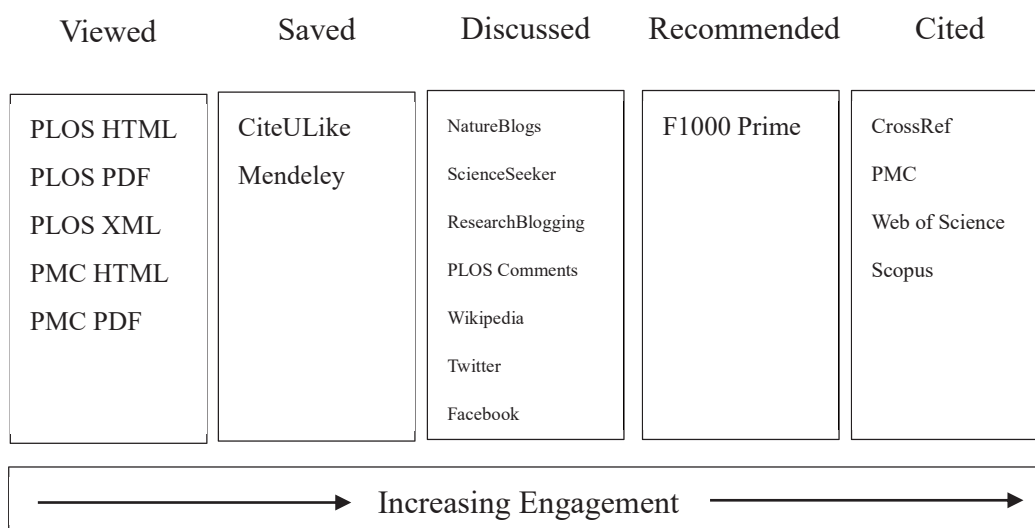
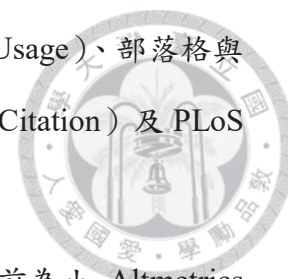


圖 2-3：Lin 與 Fenner 的分類體系

但到了 2014 年，PLoS 採用的資料指標已經改為使用率 (Usage)、部落格與媒體 (Blog and Medias)、社群網站 (Social Network)、引用 (Citation) 及 PLoS 五種類型 (蔡明月、曾苓莉, 2014)。



由以上羅列之各種不同的資料指標分類方式，可以看出目前為止 Altmetrics 的資料指標尚缺乏一致性的標準，對於各類別的定義也缺乏詳盡規範，因此不同單位或學者所做的 Altmetrics 研究分析很難放在同一個比較基準上做評量。為解決此問題，美國國家資訊標準組織 (NISO) 於 2013 年，在史隆基金會 (Alfred P. Sloan Foundation) 的贊助下開始進行第二階段的替代性評鑑計量研究計畫 (NISO Alternative Assessment Metrics)，預計於 2015 年下半年完成以學術社群為基礎的 Altmetrics 標準與應用規範建置 (蔡明月、曾苓莉, 2014; Lagace, 2013)。

第三節 現有的替代計量學分析工具

自 2010 年 Altmetrics 被提出以來，短短幾年間已經有不少相關的工具與網站可以用在分析 Altmetrics。其中，由 Priem 與 Piwowar 建立的 Altmetrics (<http://altmetrics.org>) 上面就列出不少常用的網站，不過這份清單可能已許久沒更新，在本研究撰寫的當下，已經有部分網站的連結失效了。本小節從各個學者的文獻中，統整出一些還在運作中的 Altmetrics 工具，並一一介紹。

一、Altmetric.com

Altmetric.com (<http://altmetric.com>) 是一個專門提供 Article-Level Metrics 數據分析服務的商業網站，由 Euan Adie 於 2011 年在倫敦創立，並以 Altmetric 名詞來稱呼他們的整合分析產品，以避免商標名稱與專有名詞混淆。此網站在 Altmetrics 運動期間蓬勃發展，且其所開發的 Altmetrics App 應用程式也榮獲了 Elsevier 舉辦的 App for Science 競賽首獎 (蔡明月、曾苓莉, 2014)。Altmetric.com 的客群包含機構、研究者與出版商，其客戶可以登入網站並觀看特定學術文章的

影響力，Altmetric.com 會定期分析這些學術文章，並追蹤其在各大社群網站、部落格或媒體等資料來源中的被提及或被談論狀況，並計算其影響力分數。Altmetric.com 最有特色的一點，在於其稱為「甜甜圈」(donuts) 的呈現方式，以顏色區塊組成的圓環來表達被分析文章的相關資訊，請參見圖 2-4。圓環中心是該篇文章的影響力分數，而周邊的顏色分別代表了 Twitter、Facebook 等社群媒體（藍色）、部落格文章（黃色）以及主流媒體（紅色）等等。Altmetric.com 並提供了免費的部落格 Widget 與 API，可以將特定文章的分析結果嵌入在任何網站上，也提供簡易的瀏覽器小工具，可以立即知道當下閱讀的這篇文章的 Altmetrics 分數等等。Altmetric.com 可說是一套功能非常完善的影響力分析工具。

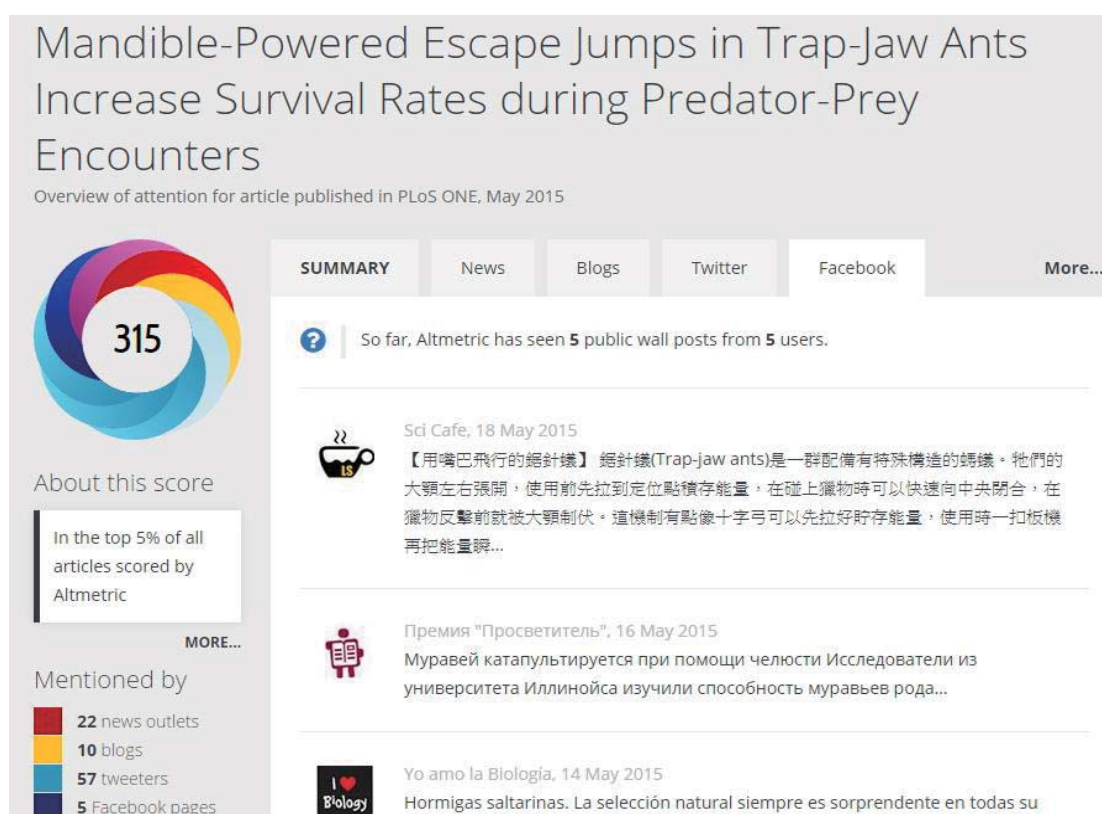


圖 2-4：Altmetric.com 的甜甜圈表示法與 Facebook 上發現的引用

一、Impactstory

Impactstory (<https://impactstory.org>) 原名 Total Impact，是一套以個人為基礎的網路貢獻度分析工具，請參見圖 2-5。使用者註冊之後，可以根據自己在網路上各大網站所註冊的會員名稱或 Email，彙整所有的個人產出到 Impactstory，並標註哪些是精選的作品。由於 Impactstory 是運用各大網站所提供的 API 來抓取資料，使用者只需要輸入自己在網站上的 ID 或 Username，就能通通彙整進來。其資料來源較為新穎，包含許多以網路服務為基礎的新興網站，如 Slideshare (簡報分享平台)、Twitter (微網誌社群) 或 Github (程式碼代管與分享平台) 等，另外學術相關的網站如 figshare (線上研究成果分享平台) 與 Publons (線上同儕審查平台) 也都是較新的網站，除此至外還有較為傳統的學術資訊網站如 Google Scholar 等。值得注意的是，Impactstory 雖然是收費服務，但其本身的原始碼是開源 (open source) 的，並由 Python 語言撰寫，任何人都可以從 GitHub 上下載其原始碼來修改或建立自己的服務，也可以貢獻修改的程式碼回饋給官方的程式。Impactstory 的原始碼可以至 <https://github.com/total-impact> 取得。

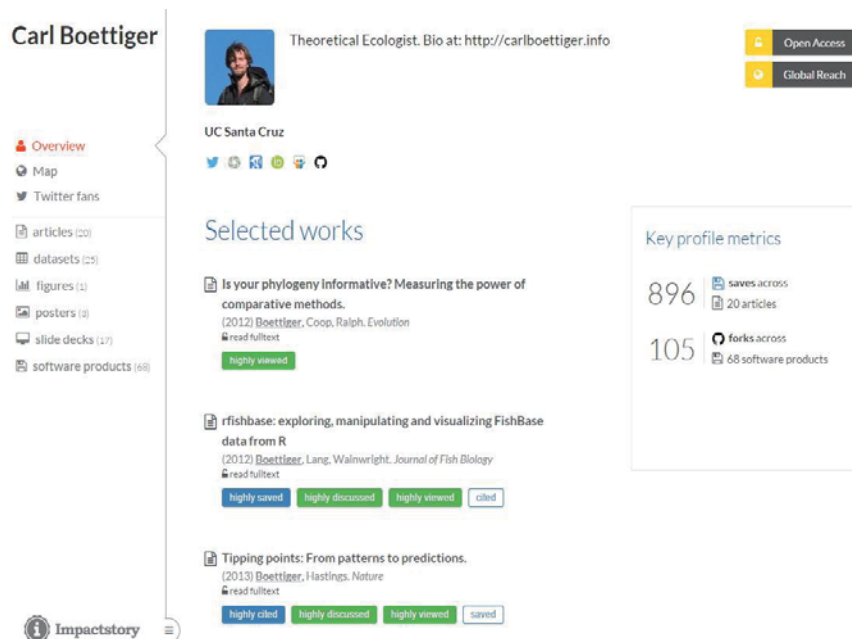
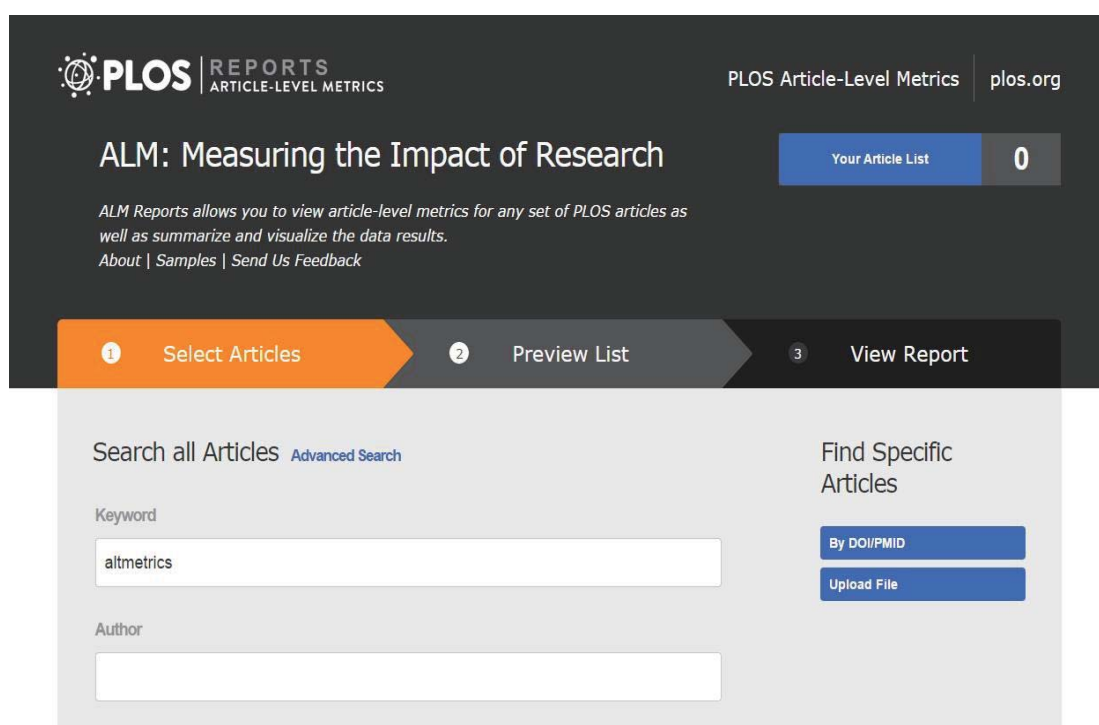


圖 2-5：Impactstory 的彙整頁面

二、PLOS ALM Reports

PLOS ALM Reports (<http://almreports.plos.org>) 提供 ALMs 計量的線上平台，由於 PLoS 本身即擁有大量學術論文資料庫，因此很容易進行引文分析。ALM Reports 提供一個簡易的入口查詢頁面，可以搜尋作者、標題、關鍵字等等，搜尋結果會導入到 PLoS Biology 的單一文章頁面，包含文章本文 (Article)、作者 (About the Authors)、計量分析 (Metrics)、評論 (Comments) 與相關內容 (Related Content) 等等資訊，請參見圖 2-6 與圖 2-7。



The screenshot shows the PLOS ALM Reports website. At the top, there is a navigation bar with the PLOS logo, the text "REPORTS ARTICLE-LEVEL METRICS", and links to "PLOS Article-Level Metrics" and "plos.org". Below the navigation bar, the main heading is "ALM: Measuring the Impact of Research". To the right of this heading is a button labeled "Your Article List" with a counter showing "0". Below the heading, there is a descriptive paragraph: "ALM Reports allows you to view article-level metrics for any set of PLOS articles as well as summarize and visualize the data results." followed by links "About | Samples | Send Us Feedback". A progress bar with three steps is visible: "1 Select Articles" (highlighted in orange), "2 Preview List", and "3 View Report". Below the progress bar, there is a search section. On the left, under "Search all Articles", there are input fields for "Keyword" (containing "altmetrics") and "Author". On the right, under "Find Specific Articles", there are two buttons: "By DOI/PMID" and "Upload File".

圖 2-6：ALM Reports 入口頁面

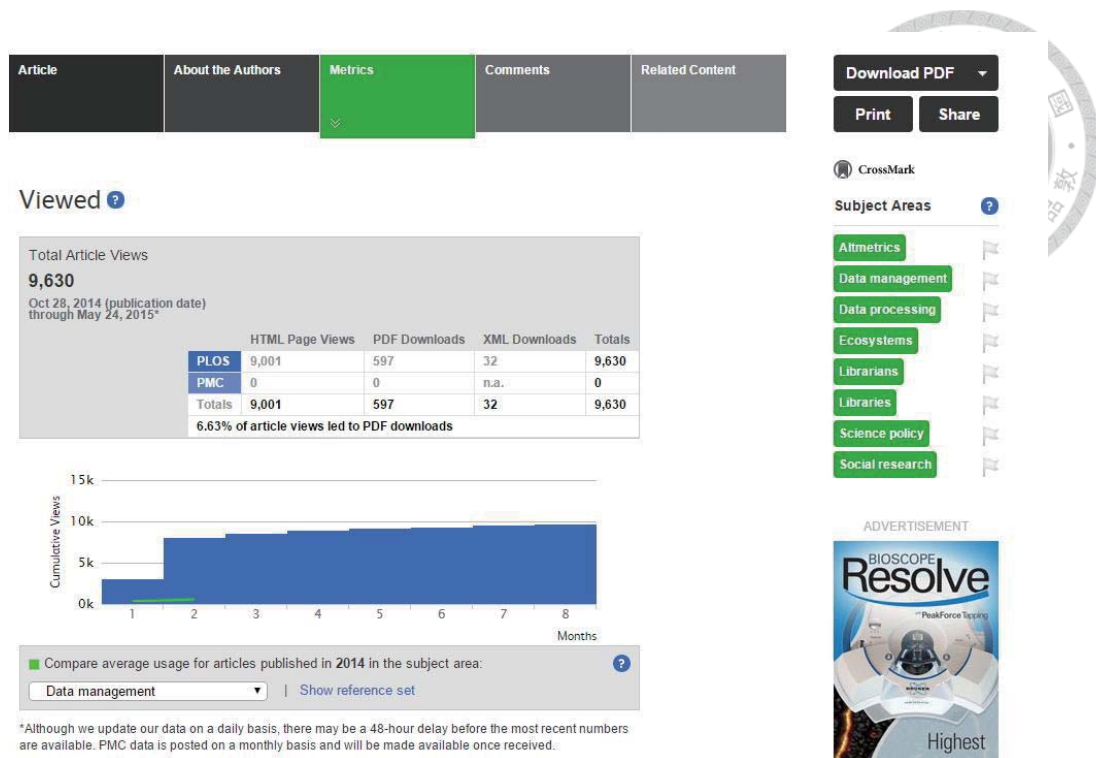


圖 2-7：文章頁面的 Metrics 分頁

三、PLUMX

PLUMX (<https://plu.mx>) 由 Plum Analytics 公司於 2013 年推出，提供強大的學術評鑑服務，追蹤所有形式的研究成果，以補強傳統引用分析在學術評鑑上的不足。Plum Analytics 於 2014 年被 EBSCO 收購，預期將整合旗下相關產品來擴大服務內容。PLUMX 採用 Cave (2012) 主張的 Altmetrics 指標分類體系，以貌似梅花五片花瓣的五個圓圈 (PLUM PRINT) 來分別代表使用率 (Usage)、擷取 (Capture)、提及與談論 (Mention)、社群媒體 (Social Media) 與引用 (Citation) 五種資料指標。五個圓圈的大小呈現出不同資料類型的影響力 (圓圈愈大代表該類型的資料來源愈多)。請參見圖 2-8

PLUMX 的資料來源部分，在使用率類別包含資料庫使用統計還有 WorldCat 顯示的各圖書館館藏數量。在提及與談論類別包括 Amazon 網路書店的書評。引用類別除了傳統的 PubMed Central、Scopus 之外，還包含專利引用資料。PLUMX

不只可以針對單篇文章分析影響力，還可以用至機構典藏系統，針對研究者、系所、學校、機構或單一期刊來做 Altmetrics 分析（蔡明月、曾苓莉，2014）。

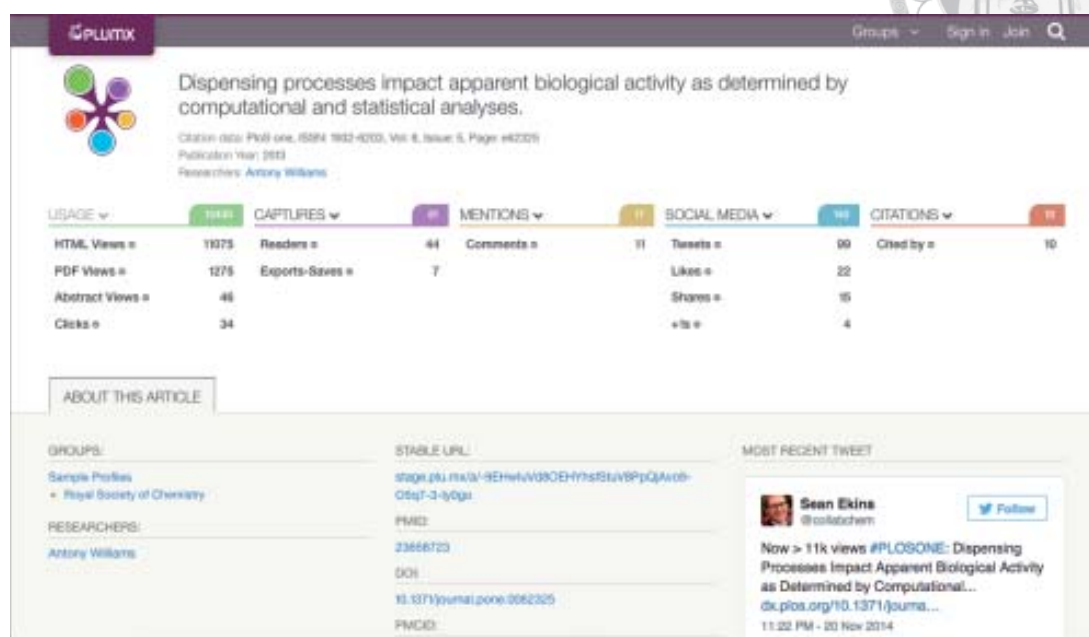


圖 2-8：PLUMX 的 Altmetrics 分析結果

四、PaperCritic

PaperCritic (<http://www.papercritic.com>) 是一個專注在分析 Mendeley 的 Altmetrics 系統，藉由 Mendeley 提供的 API 來抓取資料，能獲取文獻著作的意見與回饋資料，研究者也可在此網站評論他人撰寫的文章。網站首頁搜尋介面參見圖 2-9。

於本文撰寫時，PaperCritic，與 Mendenly 的連結有錯誤，搜尋結果會回傳「Mendeley API Error：Something went wrong when we tried to connect to the Mendeley API. Please click here to try again!」，因此暫時無法抓取到搜尋結果頁面。

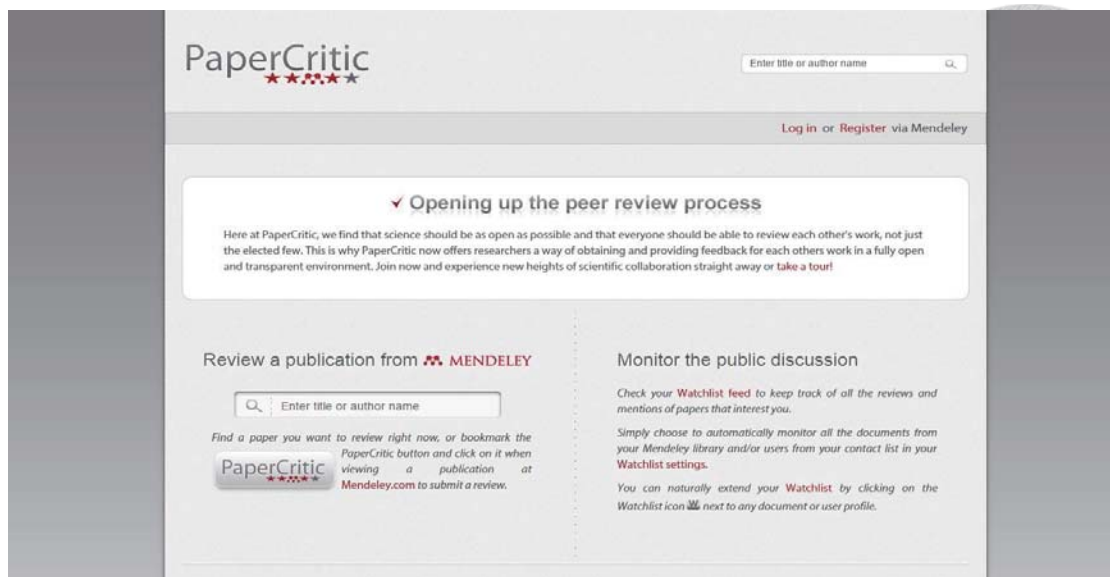


圖 2-9：PaperCritic 首頁搜尋介面

五、Publish or Perish

Publish or Perish (PoP) (<http://www.harzing.com/pop.htm>) 是一個可以免費下載的程式，支援 Windows、Mac 與 Linux。該軟體直接由 Google Scholar 獲取資料，並提供 Google Scholar 上的引用數據，也可以計算 h-index 等數種計量指標。請參見圖 2-10 的使用者介面。

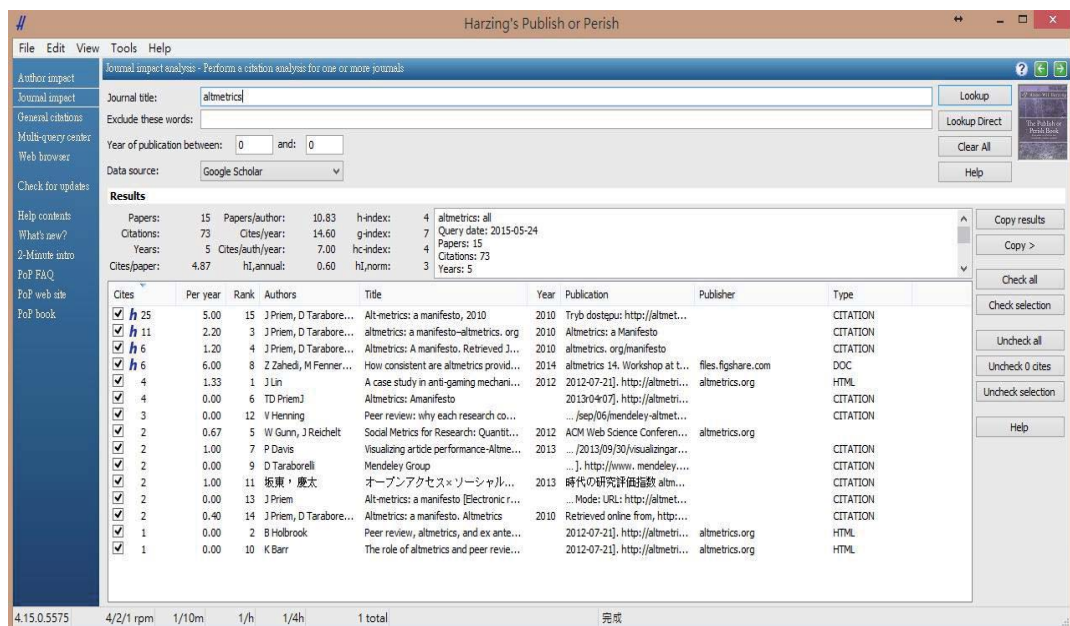


圖 2-10：Publish or Perish 系統畫面

第四節 軟體開發流程



本節探討「軟體工程」的簡易發展歷程，並介紹目前常見的軟體開發流程，簡述其適合的開發情境。

一、軟體危機與軟體工程

在 1920 到 1940 年代電腦被發明出來，人們意識到單純依賴機器的操作將會越來越複雜且不靈活。例如一個專門用來計算數據的計算機（電腦），若要改處理文字文件，必須更改線路、結構，甚至重新設計機器。對那個年代來講，更改電腦程式意味著重新進行紙上作業，規劃機器架構，然後製作出新的機器。為了應付越來越大型的電腦開發，馮紐曼於 1945 年為 EDVAC 電腦計畫提出的《First Draft of a Report on the EDVAC》報告中，設計了「馮紐曼架構」（Von Neumann architecture），又稱「儲存型（Stored）程式架構」或「普林斯頓架構」。此架構設計了一組指令集，將程式的運算轉成一連串指令執行細節。藉由將指令也當成一種資料來儲存，一台馮紐曼架構的電腦可以輕易改變其程式邏輯，以計算不同的內容。從這個時候開始，電腦的發展逐漸將抽象化軟體與硬體區隔開來，到了 1950 年代，現代化程式語言如 Fortran、ALGOL 與 COBOL 也陸續誕生。

軟體的發展非常迅速，專案也越來越龐大，開發人力越來越多。到了 1980 年代，許多軟體的規模開始失控，不只品質大幅下降，時程也嚴重延遲，導致成本劇烈上升，這段期間被稱為「軟體危機」。軟體品質的下降甚至造成人員傷亡，例如 1985 到 1987 年間，輻射治療的機器 Therac-25 的劑量計算錯誤導致許多治療中患者遭到嚴重輻射灼傷。由於這個意外，使得軟體的開發方法逐漸被重視，因而喚起軟體開發工程化管理方法論的思維。

「軟體工程」（Software Engineering）由 Margaret Hamilton 在 1968 年提出，這個名詞被作為北約科技委員會（NATO）所舉辦研討會的名稱，也在此時對外宣布這個概念。軟體工程旨在研究和應用如何以系統性的、規範化的、可量化的

程序化方法去開發和維護軟體，以及如何把合適的管理技術與軟體開發執行技術結合起來的一門學問。(Wikipedia, 2015)

在 ACM 與 IEEE Computer Society (2014) 聯合修定的《Software Engineering Body of Knowledge》提到，軟體工程領域中的核心知識包括：

- 軟體需求 (Software requirements)
- 軟體設計 (Software design)
- 軟體建構 (Software construction)
- 軟體測試 (Software test)
- 軟體維護與更新 (Software maintenance)
- 軟體構型管理 (Software Configuration Management, SCM)
- 軟體工程管理 (Software Engineering Management)
- 軟體開發流程 (Software Development Process)
- 軟體工程工具與方法 (Software Engineering Tools and methods)
- 軟體品質 (Software Quality)

由此可見軟體工程的領域已經非常龐大，需要夠大的開發團隊，才能夠掌握所有面相。其中「軟體開發流程」是影響一個軟體開發速度與品質管理的重要項目。採用正確的發流程，小型團隊也能夠掌握複雜的開發工作，降低失敗的機率及避免系統開發者陷入「焦油坑」(Brooks, 1975)。以下簡介幾個目前常見或常被討論的軟體開發流程與方法，皆被視為實踐軟體工程的方式之一。

二、瀑布流模型

瀑布流模型 (Waterfall Model) 是軟體開發流程與步驟的一種理論模式，其概念在名詞被提出前已廣泛流傳在不同的開發者之間，之後 Winston W. Royce 在 1970 年作了一次完整的描述。在瀑布流模型中，軟體開發被分為為需求分析，設計，實作，測試，整合與維護這樣的步驟依序進行，並強調系統開發應該要有完整的週期，每一次的調整、修改，都必須從頭依序經歷這些階段，如圖 2-11 所示：

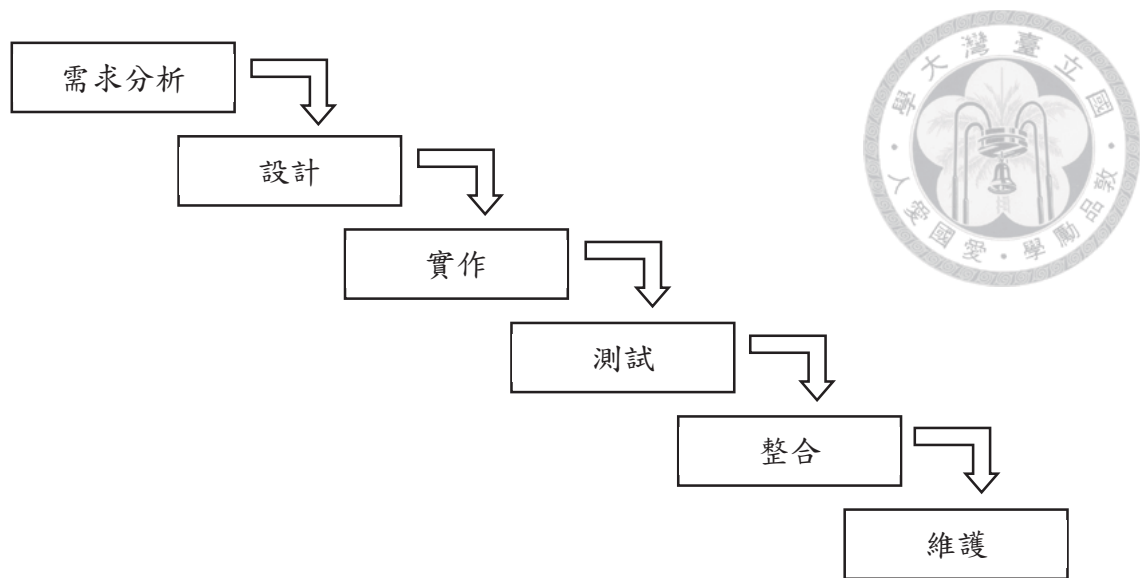


圖 2-11：瀑布流的開發階段

由於每一個週期都要自上而下依序進行，因此被稱為瀑布流（Waterfall），或稱「系統發展生命週期」（System Development Life Cycle，SDLC）。瀑布流的支持者認為，在軟體規劃時期的成本遠低於開發時期，因此應該要在規劃時清楚的分析需求，才能進入開發階段。一旦有任何錯誤在開發階段或開發完成後才被察覺，其補救成本可能是預先排除問題的 50 到 200 倍。另外，瀑布流也非常重視文件的撰寫。支持者認為大型軟體的開發沒有人可以獨力了解所有面相，隨著人員的去留，只有完整的文件才能確保軟體結構穩定且健壯，新進人員可以藉由閱讀文件快速進入狀況，參與開發。

由於瀑布流強調必須嚴格遵循整個開發與規劃流程的循環，因此導致的是其執行效率低落，即便在 Royce 的文章中，也認為這個模型不見得可以正常的運作。Parnas 等人（2011）則認為客戶常常看見實際開發的系統後，才察覺到需要調整的地方，因此會造成軟體設計被持續更改、重建，造成不必要的成本支出，意味軟體的需求規格總是無法在規劃期就完美的確認。但整體而言，多數人還是認為瀑布流適用於大型企業的軟體開發工作，且嚴格遵循瀑布流的軟體在結構穩定性與品質上是非常優良的。

三、敏捷軟體開發

敏捷軟體開發 (Agile software development) (以下簡稱「敏捷開發」) 是 1990 年代開始逐漸引起關注的新興軟體開發方式，此名詞在 2001 年，由一群已經在使用此開發方式的實踐者，聚集在美國猶他州雪鳥滑雪勝地共同宣布，此次會議又稱「雪鳥會議」。

此開發方式的特色是能夠應付快速需求變化，讓開發團隊更緊密的溝通協調，也更適合中小型開發團隊使用。其開發過程採用迭代的方式，每個階段重功能後再進行下一階段。與其他「非敏捷」開發方式相比，敏捷開發的重點在強調四個原則：

- (1) 重視人與人之間的互動，而非工具與規範。
- (2) 直接開發出可用的軟體，並以程式本身做為文件與討論依據，而非依賴大量文書文件。
- (3) 與客戶密切合作、討論，尋求解決方案，而非依賴合約。
- (4) 隨時快速的回應變化做調整，而非永遠遵循預定計畫。

由此可見，敏捷軟體開發並不意味著開發速度的「快」，而是對變動的即時反應與彈性，並把程式碼本身當成溝通用文件，省去程式以外不必要的大量繁瑣細節，因為每一次的需求變動，會讓那些辛苦寫完的文件失去價值。對比於瀑布模型嚴格遵循計畫流程來說，敏捷開發較能適應高度變化的市場與不確定最終需求的專案。

適用於敏捷開發的團隊與組織會與瀑布模型有所不同，由於敏捷軟體開發重視人與人之間的溝通、交流，也重視與客戶溝通的層面，因此一個能夠成功實踐敏捷開發的組織需要有以下特色 (Wikipedia, 2015)：

- (1) 組織文化必須支持談判
- (2) 人員彼此信任
- (3) 人少但是精幹

(4) 開發人員所作決定得到認可

(5) 環境設施滿足成員間快速溝通之需要

但除了「敏捷」一詞以外，其實有大量強調快速的開發模式與敏捷軟體開發非常相似，但使用不同的名詞，這些開發方式也被認為是敏捷開發的一部分。因此「敏捷開發」一詞並非一個完全固定的開發模型，而是泛指任何一個相較於傳統「非敏捷」開發模式來說，較能即刻回應需求變動的方式，皆可被稱為「敏捷（Agile）方法」。目前被認為屬於敏捷方法的開發模式有：極限編程（Extreme programming, XP）、Scrum、AUP（Agile Unified Process）與 Crystal 等等。而使用在敏捷開發的技術，則有 TDD（測試驅動）與 BDD（行為驅動）等。

敏捷開發主要的缺點在於過短的迭代開發時間可能會導致軟體功能無法如期完成，一旦時程脫勾越來越嚴重，反而會造成軟體發佈的延遲。另外敏捷開發強調人與人的溝通，在網路時代逐漸流行遠端或分散工作的情況下，需要更合適的溝通工具，來彌補人員無法見面討論的不足。

四、快速應用程式開發

快速應用程式開發（Rapid Application Development）（以下簡稱 RAD），比起敏捷軟體開發更早被提出來，是為了面對早期「非敏捷」開發模式而設計的一種加速開發的方法。由電腦顧問專家 James Martin 在 1991 年提出，是一種快速生成系統並不會犧牲品質的方法。RAD 與原型開發方法（Prototyping）有一定程度相似，目標都是對用戶需求做出快速反應。RAD 的目標是在 30 天到 90 天內產出符合需求的軟體，根據 80/20 法則，最重要的 80% 功能常常只佔有 20% 的開發時間，因此若開發者與客戶能夠適度妥協，將能夠讓軟體開發速度大幅提昇。RAD 要求由經過通分技能訓練的小型團隊來執行，並在過程中要盡可能的與客戶保持密切交流，系統終端使用者的參與在開發過程是非常重要的。當第一階段軟體交付後，還有需要調整或擴充的地方，再藉由多次迭代開發來滿足（Jeffrey、Joey、Joseph, 1996）。雖然 RAD 並不列為敏捷開發的一種，但在迭代

開發以滿足需求變動的部分本質上是非常相似的。

RAD 在開發技術上有額外的特色，採用 RAD 模式的軟體開發需要合適的工具來支援，軟體工具的成熟發展有機會蓬勃發展。例如所謂的「第四代程式語言」(4GL)，便是實現 RAD 開發的一個重要輔助工具。第四代程式語言主要描述不需要紮實的專業技能就能使用的開發技術，例如 SQL 查詢語言、圖形語言、報表產生工具、應用程式產生器（例如 Visual Basic）等等，是一般程式語言再做了更高階抽象化後的樣貌。運用第四代語言，可以快速替軟體進行原型開發，建立基本功能，上線測試之後再持續迭代修改，這也就是 RAD 與原型開發模式的開發流程。

目前常見的 RAD 開發工具，以大型 IDE 所提供的程式產生器(code generator)、GUI 元件拖拉編輯器等為主，例如微軟的 Visual Basic、寶蘭的 Delphi 皆自稱為 RAD 開發工具，蘋果的 Xcode 也屬於 RAD 的工具之一。但雖然在主流上稱這些具備 GUI 編輯功能的環境為 RAD 工具，但其實廣泛的定義下，任何語言的函式庫，或者提供簡潔的指令用來產生介面元件的純開發框架，都被認為是一種第四代語言或 RAD 工具。

王要武（2008）制定了幾個適合採用 RAD 開發模式的情境：

- (1) 開發的軟體相對獨立，不與其他軟體交互
- (2) 軟體可利用許多現有的函式庫
- (3) 系統效能並非關鍵的考慮因素時
- (4) 應用的開發可以使用高端的軟體開發工具（如 4GL）
- (5) 系統使用範圍較窄（內部使用或面向垂直市場）
- (6) 專案的執行範圍與時程受到高度限制
- (7) 系統可靠性並非關鍵的考慮因素
- (8) 系統可分解成多個獨立的模組
- (9) 所採用的技術相對成熟（已使用超過一年）

至於 RAD 的缺點則與敏捷開發差不多，由於兩者皆是採迭代開發，有很高的機率在短期迭代內無法產出令人滿意的成果，長期下來將會拖垮專案進度。但同樣也可以藉由合適的工具與成熟的管理技術來避免。

本節介紹了三種常見的開發模式，各自適用於不同的情境與團隊規模。而本研究開發系統時，考量到開發者為研究者一人，為了確保開發效率，將以 RAD 開發模式為主。而由於過程中需求變動並不多，大多在開發前便能掌握此系統的多數規格，不需有太過多次的迭代，在需求分析上比較近似瀑布模型，預先訂出所有的分析資料來源，擬定分析策略後才開始開發。但由於資料掌握在第三方網站上，因為這些網站定期更新的關係，有時分析功能會失效，必須依照該網站的更新重新組織分析策略。這個反覆修改的過程也會近似於敏捷開發。至於以 RAD 模式開發所採用的快速開發工具，將於後續章節說明。



第三章 研究方法與步驟



本研究以設計適用於中文學術資訊傳播的 Altmetrics 分析系統為主，目的是希望能彌補國際現有 Altmetrics 工具或服務的不足，藉由分析學者可能互相交流、或著作可能被收錄的網站來取得單一學者的 Altmetrics 數據。

第一節 系統設計與任務流程

本系統查詢流程較簡單，主要任務是根據使用者輸入的學者中英文姓名，組合出查詢關鍵字，至預先定好的各個資料來源查詢資料，分析統計結果，接著儲存回系統資料庫，並顯示在分析結果頁面上供使用者參考。由於分析過程需時較久，因此第一次查詢會請使用者稍候，之後搜尋時才顯示結果。請見圖 3-1 所示：

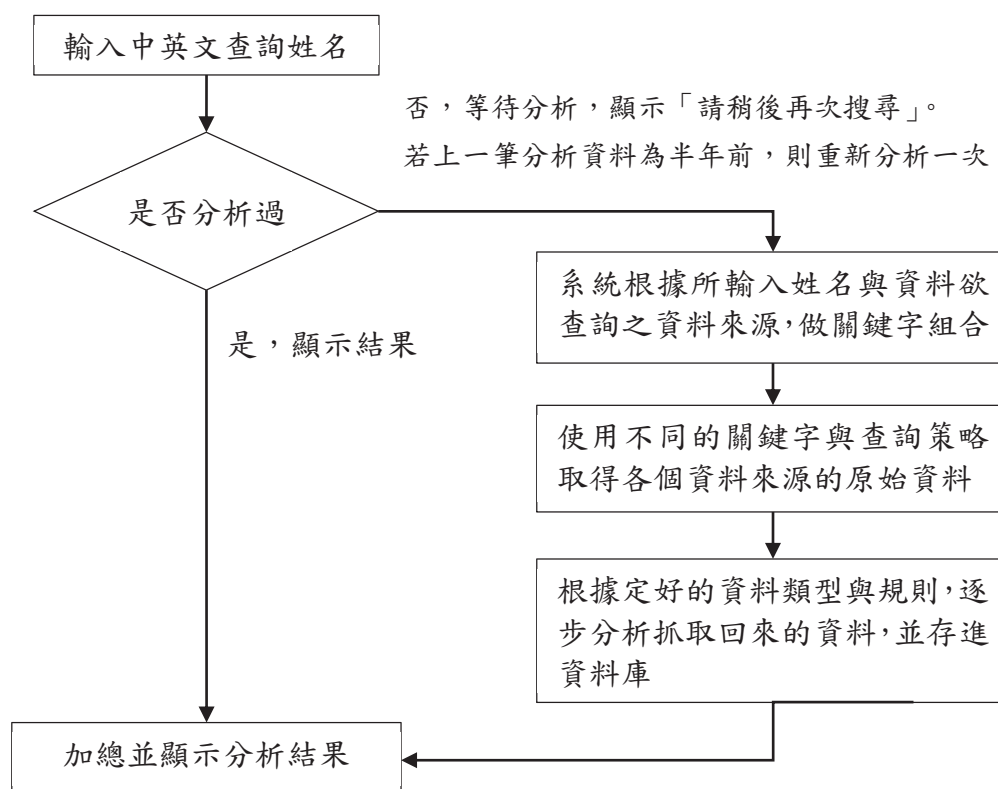


圖 3-1：系統查詢任務流程

第二節 資料來源



為了要分析國內學者與其著作，本研究挑選了以下幾個網站，作為分析時的資料來源。多數網站以 Google 作為主要的搜尋引擎，方便系統根據學者姓名檢索相關網頁。部分無法被 Google 收錄之網站，則以該網站自備之搜尋系統或 API 來檢索。

- (1) 課程大綱：分析學者的著作是否被課程當做參考書目。
- (2) 學位論文：分析學者著作是否被作為參考文獻，或者為指導教授。
- (3) 社群媒體：分析學者的著作是否在 Twitter、Facebook 等社群媒體上被討論或提及，或者本身發起任何評論與討論。
- (4) 報章雜誌：分析學者的著作是否在報章雜誌的網站上被討論或提及，或者本身即是作者。
- (5) 維基百科：分析學者是否在維基百科的條目中被引用或提及，以及該學者是否有獨立條目。
- (6) Google Scholar：分析學者著作是否被 Google Scholar 收錄，以及在 Google Scholar 上統計的被引次數。
- (7) TCI-HSS：TCI-HSS 是台灣人文社會期刊論文的引文索引資料庫，可由此資料庫分析學者在人文社會研究領域的貢獻狀態。
- (8) Web of Science：分析學者在 Web of Science 上的被收錄著作篇數以及被引用次數。
- (9) Mendeley：分析學者著作在 Mendeley 上被引用的次數。

表 3-1 羅列資料來源，與本研究採用之資料蒐集工具以及對應的工具網址。

表 3-1：Altmetrics 分析資料來源統整

資料來源	搜尋引擎	網址
課程大綱	Google	https://www.google.com.tw
學位論文	華藝線上圖書館	http://www.airitilibrary.com
	OAI 博碩士論文聯邦查詢系統	http://fedetd.mis.nsysu.edu.tw
社群媒體	Google	https://www.google.com.tw
報章雜誌	Google	https://www.google.com.tw
維基百科	Google	https://www.google.com.tw
Google Scholar	Google Scholar	https://scholar.google.com.tw
TCI-HSS (臺灣人文及社會科學引文索引資料庫)	TCI-HSS	http://tci.ncl.edu.tw
Web of Science	WOS API	http://webofscience.com
Mendeley	Mendeley API	https://www.mendeley.com

第三節 分析資料類型

根據上一章文獻探討中提到的資料指標分類，並搭配所採用的資料來源選擇合適的資料類型，並根據一組特定規則來分析，如表 3-2 所示：

這些資料類型參考了 Cave (2012) 所列之 Altmetrics 資料分類，並整合傳統計量分析中常用的貢獻度所訂定，解釋如下：

(1) 課程大綱數量：

該學者出現在課程大綱中的數量，包含被引與自引，以及可能被提及的次數，作為一個整合分類。

(2) 被引次數與自引次數：

較為傳統的被引與自引次數，常見在書目計量學與文獻計量學中，可以了解學者出版之著作在網路社群、或是資料庫系統所收錄的文獻中被引用的數量。也與 Cave 所提的 Altmetrics 「引用」(Citation) 分類相同。

表 3-2：資料過濾規則

資料來源	資料類型	規則
課程大綱	課程大綱數量	內文有「課程大綱」、「課綱」、「大綱」等關鍵字的網頁
	被引次數	書目資料出現在他人的課程大綱上
	自引次數	書目資料出現在自己的課程大綱上
學位論文	被引次數	書目資料出現在論文參考書目中的數量
	指導次數	姓名出現在指導教授欄位的論文數量
社群媒體	被引次數	姓名出現在參考書目中
	著作篇數	以個人帳號發表的訊息數量
	被提及次數	姓名在他人發表的訊息與評論中出現的次數
報章雜誌	被引次數	姓名出現在參考書目中
	著作篇數	姓名出現在作者欄位
	被提及次數	姓名在他人發表的文章與評論中出現的次數
維基百科	被引與提及	姓名出現在條目的次數與參考書目的書目數量
	條目名稱	姓名被作為單一條目名稱
Google Scholar	著作篇數	著作被 Google Scholar 收錄的數量
	被引次數	搜尋指定作者，並加總每篇文章的被引用次數。
TCI-HSS	著作篇數	指定搜尋作者姓名的文章數量
	被引次數	搜尋指定作者，並加總每篇文章的被引用次數。
Web of Science	被引次數	搜尋指定作者，並加總每篇文章的被引用次數。
Mendeley	被引次數	搜尋指定作者，並加總每篇文章的被引用次數。

(3) 指導論文次數：

由於文獻計量分析旨在分析學者的貢獻程度數據，指導論文也是學術產出的一種，在網路上也容易藉由資料庫系統提供的資訊取得數據，因此本研究列入指導論文次數作為分類之一。

(4) 被提及次數：

較近似 Cave 所提的「提及與談論」(Mention)，或 Konkiel (2012) 所提的「評論」(Reviews) 分類。通常出現在文章內文而非參考書目中，表示學者或其著作在網路文章的內文中被提及或是評論的數據。

(5) 著作篇數：

藉由統計學者本身在網路社群發表的文章、留言、評論等，可以了解此

學者在網路上的活躍程度，若著作篇數可能反映學者的學術生產力、社群交流程度或者意見影響力等等。例如在 Impactstory 中，主要的著作分類也包含了程式的產出。本研究以學者在網路社群的帳號發表的文章為主。



第四節 前導研究

本系統以敏捷模式持續迭代發展，並如同 RAD 模式般，每一次完成的原型都可以正常運作。本小節將以 Pilot Study 的方式，在系統發展的初期階段，分析系統擷取之資料與人工擷取之資料的比較。根據比較的差異數據，可以做為系統發展過程的參考與調整之依據。

實驗進行時，系統已完成並且可正常分析的資料來源項目如表 3-3 所示，當時的報章雜誌是分成新聞資料與期刊雜誌兩個項目的，且資料來源也較少，但已足夠讓我們檢視系統自動化 Altmetrics 分析與人工查詢的差距。

表 3-3：實驗進行時的資料來源與類型

資料來源	資料類型	搜尋引擎
課程大綱	被引與自引次數	Google
學位論文	被引次數	Google
	指導論文數量	Google
新聞資料	被引次數	Google
	著作篇數	Google
期刊雜誌	被引次數	Google
	著作篇數	Google
維基百科	被引與提及次數	Google

實驗過程是將系統分析某個學者的數據與人工查詢出來的數據做對比，表 3-4 是本研究中挑選的學者名單。經由系統分析與人工比對後，我們可以得出如表 3-5 的數據，並算出平均差距，只要能從平均差距觀察出系統與人工查詢的差別，就能大概得知分析成效。

表 3-4：實驗用學者名單與編號

編號	姓名	編號	姓名
E1	劉錦添	E5	王道一
E2	陳虹如	E6	王泓仁
E3	林明仁	E7	沈中華
E4	江淳芳	E8	陳旭昇

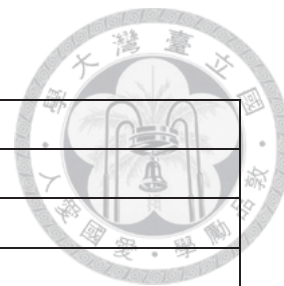


表 3-5：數據比對一覽

		新聞資料		期刊雜誌		課程大綱	學位論文		維基百科
		被引	著作	被引	著作	被引與自引	被引	指導	被引與提及
E1 劉錦添	人工	82	1	6	6	24	1076	43	0
	系統	36	0	1	1	43	570	37	0
	差距	-46	-1	-5	-5	19	-506	-6	0
E2 陳虹如	人工	1	0	0	0	0	72	9	0
	系統	0	0	1	1	201	80	12	0
	差距	-1	0	1	1	201	8	3	0
E3 林明仁	人工	8	0	1	1	15	33	13	1
	系統	37	0	4	4	13	51	20	0
	差距	29	0	3	3	-2	18	7	-1
E4 江淳芳	人工	0	0	1	1	0	10	2	0
	系統	0	0	0	0	138	10	3	0
	差距	0	0	-1	-1	138	0	1	0
E5 王道一	人工	3	0	0	0	18	2	4	1
	系統	51	0	0	0	310	6	11	0
	差距	48	0	0	0	292	4	7	-1
E6 王泓仁	人工	17	0	4	2	0	144	13	0
	系統	0	0	1	1	11	68	23	0
	差距	-17	0	-3	-1	11	-76	10	0
E7 沈中華	人工	377	82	9	8	488	3225	336	1
	系統	54	0	21	21	872	752	226	0
	差距	-323	-82	12	13	384	-2473	-110	-1
E8 陳旭昇	人工	3	0	0	0	28	433	11	0
	系統	0	0	0	1	313	281	14	0
	差距	-3	0	0	1	285	-152	3	0
人工總和		491	83	21	18	573	4995	431	3
系統總和		178	0	28	29	1901	1818	346	0
平均差距		-39.13	-10.38	0.88	1.38	166.00	-397.13	-10.63	-0.38

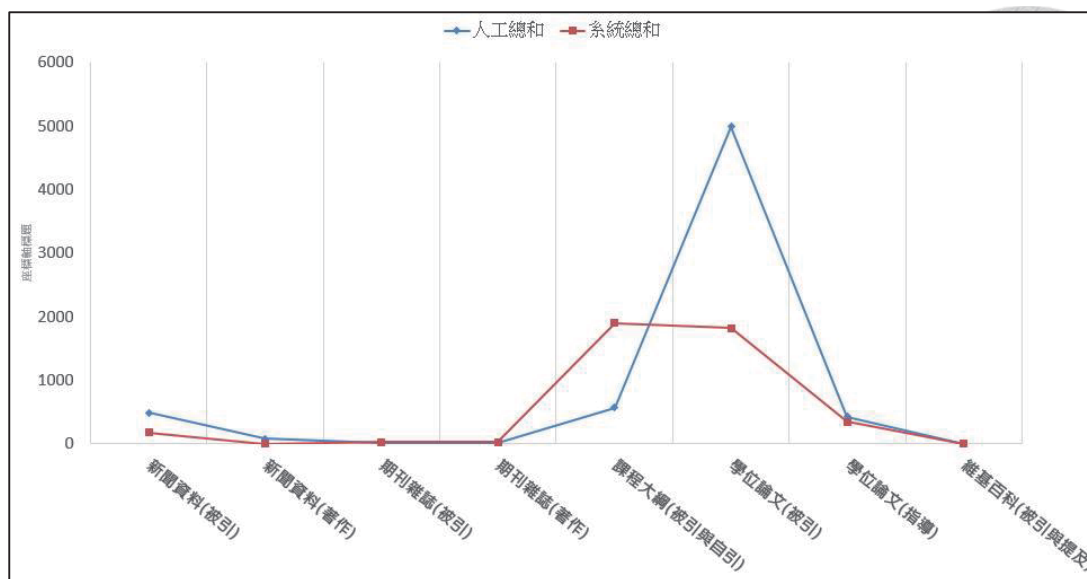


圖 3-2：人工總和與系統總和折線圖

由表 3-5 與圖 3-2 可見，課程大綱與學位論文的資料量特別大，其他資料來源則較小。主要原因是課程大綱與學位論文所編寫的收錄內容較明確、嚴謹，引用資料與格式也較完整，在搜尋上能夠精確的找尋到目標學者。以課程大綱來說，大多是由教授編寫，在內文提及與引用上會明確的標註，且各大專院校也會用統一格式收錄這些課程大綱，搜尋辨識更為容易。而學位論文採用的也是專業的論文索引資料庫，內容編寫有一定程度的品質，這方面使得擷取過程無論人工與系統都較容易，自然所得數據也較高。這是其他資料來源如維基百科、新聞與期刊平台等等比較不容易達到的程度。

但這兩個資料來源也因為資料量過大，其差距更為明顯。在差距比對上，課程大綱的人工擷取數量較系統擷取來得少，學位論文則人工擷取較系統擷取多。其中，課程大綱之所以人工擷取較少，是因為其採用 Google 搜尋引擎從各大專院校去搜尋資料，有較高的資料重複性。例如同一間學校的某個課程頁面，也許有多個不同的網址，在人工判讀時會自動剔除相同課程的資料，但系統不一定有能力分辨，可能會將重複的課程列入數據中。另外在多個學校都有不同格式的狀況下，人工判斷也容易出現遺漏，不如系統判斷精準。這可能是課程大綱的人工擷取數據較系統少的原因；學位論文則反過來，人工擷取數據

較系統擷取數據來得多，主要原因可能是系統查詢過程受限於資料來源存放在第三方平台上，不同的平台有查詢數量限制，當系統密集查詢時，會有部分結果無法返回給系統。而人工查詢較為緩慢且分散，不容易被系統認定為大量查詢而封鎖。

除了課程大綱與學位論文的數據差異較巨大以外，其他資料來源的差異較小，數據也較穩定，表示系統的分析結果與人工分析相差不遠，有一定可信度。隨著未來陸續新增不同的資料來源與採用的搜尋引擎，一定也會有部分資料來源也受到網路狀態、資料來源的特性與資料庫穩定狀態而導致數據差異較大，不過這是所有類似 Webometrics 以網路資料為基礎的分析研究都會面臨的狀況與挑戰，研究者與系統設計者的工作便是盡可能的用各種辦法降低網路與外部狀態不穩定所造成的數據波動。或是藉由演算法來證明資料的波動不影響可信度。整體而言，從現有數據來看，自動擷取的大量資料，仍然可以用來研究分析學者的活動狀態，有一定的參考價值。這也是目前所有網路計量與 Altmetrics 分析系統目前採用的方法。

第四章 系統實作



第一節 開發平台選擇

本研究中的整合分析系統由研究者獨立開發，為了達到較高的單人開發效率，讓研究過程中可以專注在系統流程的以及替代計量演算法的設計，而非複雜的開發工作，故在開發平台選擇上有以下考量：

1. 開發成本低，語法簡潔，且效能也不差的程式語言。
2. 架構完整的開發框架，不必關注底層細節。
3. 整合基本的 GUI 元件，不用自己開發使用者介面。
4. 良善的基礎平台與大量的使用者社群，取得支援容易。

以上的需求符合第二章所提到之 RAD 開發模式，目的是希望能以最快的速度完成基本功能雛形，接著藉由重複的測試與延展來讓系統更臻完整。在多方考量下，本研究選擇採用 Joomla CMS (<http://joomla.org>) 為開發平台，其有以下特色。

1. 採用 PHP 程式語言，是目前最流行的網頁語言，擁有高效能、使用族群廣大，語法簡潔易懂等優點，取得支援與日後維護皆較容易。
2. Joomla CMS 擁有一套開發框架平台，將基本的程式流程控管、資料庫存取、視圖渲染輸出皆預先準備好，並提供多組好用的函式庫，可大量降低重造輪子的機會。
3. CMS 軟體本身內建會員管理、基礎 GUI 介面元素，可以很快速的搭建出基本應用程式(Application)，無須費心於大量與應用邏輯無關的細節。
4. Joomla 同樣擁有龐大使用族群、經過優化的底層系統可以良好運作在多數 Linux 或 Windows 主機上，佈署方便，不受環境影響。

以上許多特點皆符合選擇開發平台的考量，兼顧開發速度與功能性，讓開發者可以專注在應用邏輯的編寫，是良好的 RAD 平台架構。圖 4-1 是 Joomla 官網。

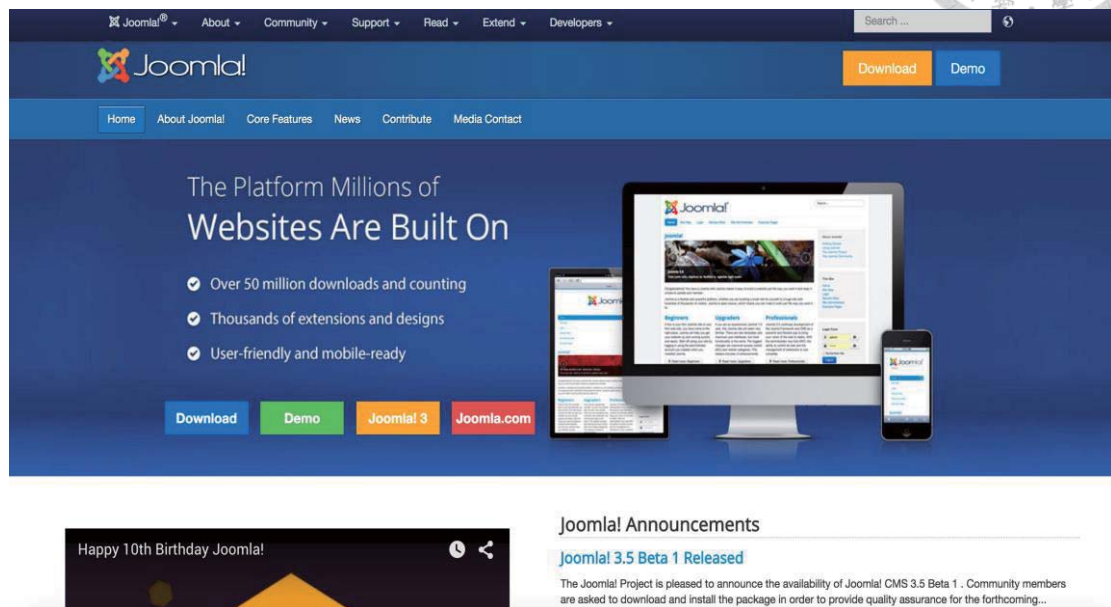


圖 4-1：Joomla.org 網站首頁

圖 4-2 則是 Joomla 自帶的管理 GUI 介面，只要根據官方提供的方式編寫程式，就能自動產出漂亮好用的操作介面。

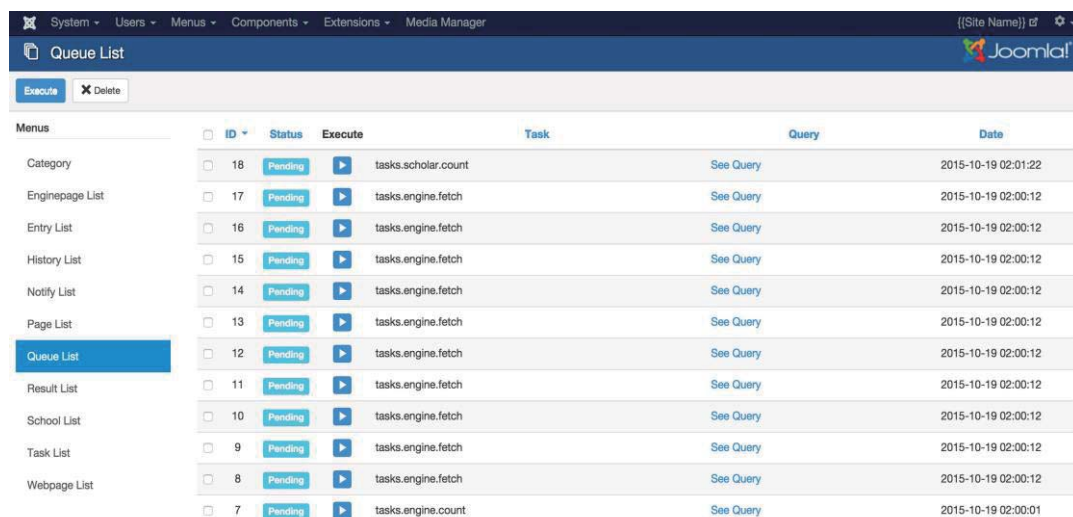


圖 4-2：Joomla CMS 後台操作介面

第二節 介面規劃

本研究規劃的 Altmetrics 系統是以學者姓名作為分析 (Author-Level)，因此介面設計上，需要提供一組表單，讓使用者輸入姓名來查詢。搜尋目標以國內學者為主，考量到參考文獻的編寫可能中英文皆有，系統需提供中英文兩種欄位，供使用者輸入正確的英文翻譯姓名或縮寫，以免找到錯誤的人選。

圖 4-3：查詢介面首頁

當人名資料送出後，我們會進入到一個整合的分析結果顯示介面，此介面會列出所有分析資料來源與資料類型的結果，如圖所示。

此系統最重要的設計在於，每一種資料來源都必須使用不同的邏輯來擷取所需的資料，無法以單一邏輯適用所有的資料來源與搜尋引擎。因此雖然介面很簡單，但內部邏輯較為複雜，將在下一節說明內部邏輯的流程與如何設計。

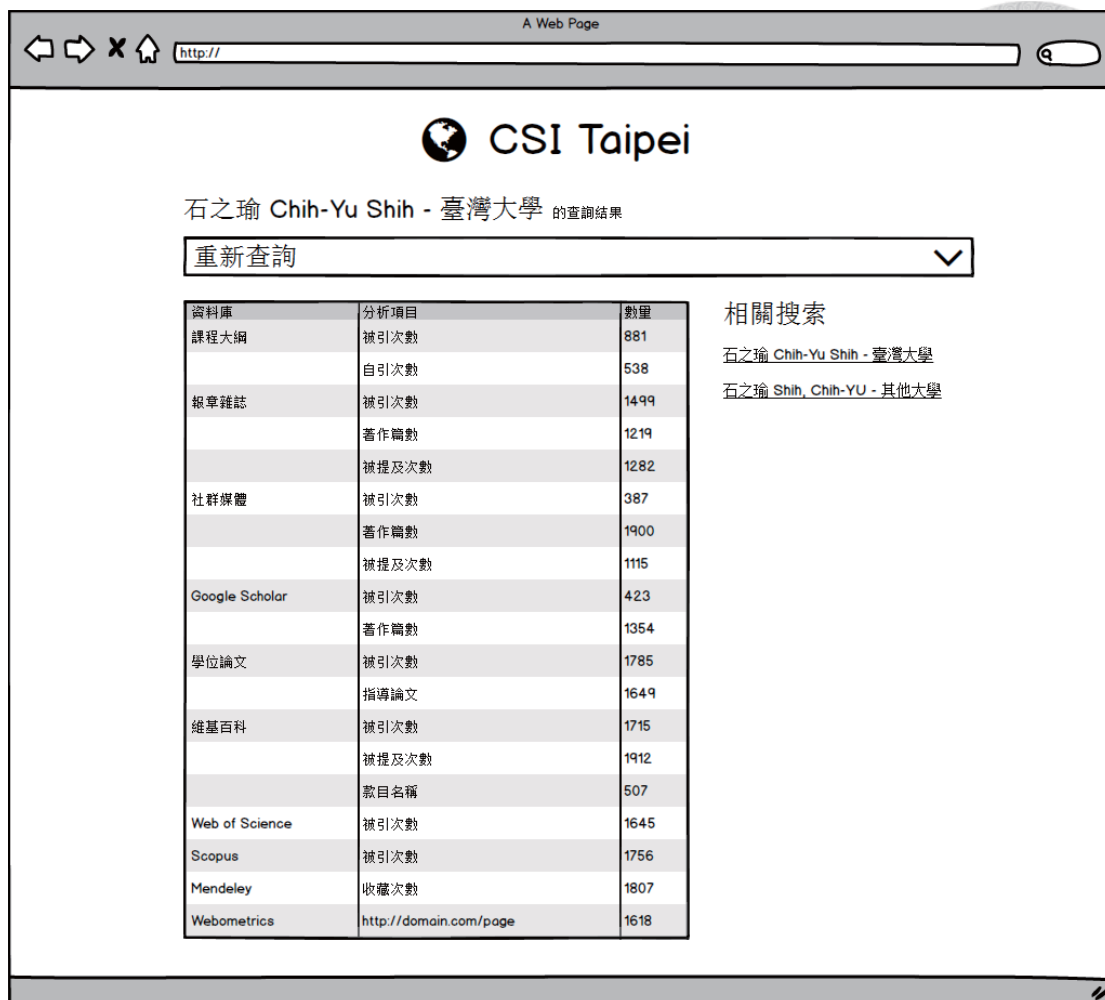


圖 4-4：查詢結果整合顯示介面

第三節 任務結構與流程

本節根據每一個不同的資料來源與資料類型，分別敘述系統任務流程，採用的關鍵字等，作為接下來程式設計時的基礎。以下每一個資料來源與使用的搜尋引擎都會附上一個英文的名稱，作為資料來源 ID。每個流程步驟也都將有一個英文 ID，在未來設計任務佇列時，可作為程式中辨認任務用的識別標記。



一、課程大綱

資料來源 ID：syllabus

搜尋引擎 ID：google

資料類型：被引、自引、是否為課程大綱

課程大綱採用 Google 為搜尋引擎，主要分析各大專院校有被索引的公開課程大綱中，是否有參考文獻引用此學者的著作。為了要能夠過濾出課程大綱，關鍵字組合如下，採用 Google 搜尋引擎的編寫規則：

("學者名字" OR "Last-Name First-Name" OR ...) AND (課程大綱 OR "Syllabus")

若本課程是學者自己開設的，則統計為自引。由於 Google 檢索結果是發散的，我們必須以簡單的邏輯來判斷該網頁是否為課程大綱，否則自引與被引皆不計入。

系統處理此分析的流程為：

1. 計算搜尋結果總數量，與搜尋總頁面。
2. 將搜尋引擎的每個頁面抓取下來暫存。
3. 抓取暫存的搜尋引擎頁面中的所有連結。
4. 依序將所有連結的目標網頁抓下來，並分析內容。

由於多數的搜尋引擎都會將搜尋結果分頁，以 Google 來說，每頁最大筆數為 100 筆，最多 10 頁，也就是 1,000 筆結果。因此我們必須先算出總頁數，再依序抓出每一頁之中的連結。最後陸續抓取每個搜尋引擎頁面所收錄的網頁連結，並計算該網頁的資料類型數據。

依據以上流程，可以先訂出以下的任務流程供後續程式開發時參考，ID 作為一個獨立的任務關鍵字，我們才能設計實際的程式流程：

表 4-1：課程大綱任務流程

順序	任務	ID	執行次數
1	計算搜尋結果總數量，與搜尋總頁面。	engine.count	1
2	將搜尋引擎的每個頁面抓下來暫存。	engine.fetch	1 - 10
3	抓取暫存的搜尋引擎頁面中的所有連結。	engine.parse	1 - 10
4	依序將所有連結的目標網頁抓下來，並分析內容。	page.analysis	0 - 1,000

在 page.analysis 階段，我們必須編寫課程大綱的分析邏輯，邏輯如下：

1. 網頁中需出現「課程大綱」、「課綱」、「Syllabus」等關鍵字才屬於課程大綱，否則結果一律回傳 0。
2. 學者姓名需出現在「參考文獻」、「參考書目」、「Reference」等關鍵字後方，才屬於「引用」，每出現在一行中計為一次引用。
3. 學者姓名出現在「教授」、「老師」等關鍵字一定範圍內，視為課程老師，則所有引用計為「自引」，反之則為「被引」。

最後結果會統一存進資料庫，在分析結果整合顯示面板中再取出加總後顯示於介面上。

二、報章雜誌

資料來源 ID：paper

搜尋引擎：google

資料類型：newspapers and general periodicals 的被引、作者、被提及

報章雜誌的分析流程與課程大綱相似，皆是依靠 Google 搜尋引擎來查找，

並且分析網頁內容，關鍵字如下，需要根據報章雜誌的特性包含一些相關字詞，並排除影音網站：

intext:"學者姓名" (intext:教授 OR intext:博士) AND intitle:"周刊 OR 週刊 OR 月刊 OR 雜誌 OR 簡訊 OR 通訊 OR 評論 OR 新聞 OR 報" AND -inurl:youtube AND (inurl:com OR inurl:gov)

分析流程設計是一模一樣的：

表 4-2：報章雜誌任務流程

順序	任務	ID	執行次數
1	計算搜尋結果總數量，與搜尋總頁面。	engine.count	1
2	將搜尋引擎的每個頁面抓下來暫存。	engine.fetch	1 – 10
3	抓取暫存的搜尋引擎頁面中的所有連結。	engine.parse	1 – 10
4	依序將所有連結的目標網頁抓下來，並分析內容。	page.analysis	0 – 1,000

而 page.analysis 階段的分析邏輯則如下，由於在關鍵字上已特別區隔出報章雜誌相關關鍵字，因此不會別判斷該網頁的類型，而是假設所有檢索到的網頁皆是報章雜誌：

1. 分析網頁 HTML 中 class 含有「author」字串的元素，若元素中的文字與學者中英文姓名相同，則視為是「作者」。
2. 若是作者，便忽視其他資料數據，若不是作者，則根據姓名出現次數計算被引或被提及。在「參考文獻」或「參考書目」等詞彙之前出現的次數記入被提及，之後出現的每行計入一次被引。

三、維基百科

資料來源 ID：wiki

搜尋引擎：google

資料類型：Wikipedia 之條目名稱、被引、被提及



維基百科的分析流程與前兩者相似，也是依靠 Google 搜尋引擎來查找，並且分析網頁內容，關鍵字如下：

```
("教授姓名" OR "First-Name Last-Name" OR ...)
site:zh.wikipedia.org/ inurl:"zh-tw" OR "wiki" AND -
inurl:template AND -inurl:category AND -inurl:talk
```

由於維基百科有些被索引到的頁面不是條目，例如 template、category 與 talk 等頁面，要從網址上去排除。而分析流程設計與課程大綱也是一模一樣的：

表 4-3：維基百科任務流程

順序	任務	ID	執行次數
1	計算搜尋結果總數量，與搜尋總頁面。	engine.count	1
2	將搜尋引擎的每個頁面抓下來暫存。	engine.fetch	1 – 10
3	抓取暫存的搜尋引擎頁面中的所有連結。	engine.parse	1 – 10
4	依序將所有連結的目標網頁抓下來，並分析內容。	page.analysis	0 – 1,000

而 page.analysis 階段的分析邏輯則如下，由於在關鍵字上已特別限定 site:wikipedia.org，因此搜尋到的一定是維基百科網頁：

1. 分析維基百科頁面的條目區域，文字是否與學者姓名一樣，若是的話則為條目名稱。

2. 根據姓名出現次數計算被引或被提及。在「參考文獻」或「參考書目」等詞彙之前出現的次數記入「被提及」，之後出現的每行計入一次「被引」。無論有無條目名稱，都照樣計算「被引」與「被提及」。



四、社群媒體

資料來源 ID：social

搜尋引擎：google

資料類型：作者、被引、被提及

社群媒體 ID：facebook / twitter / google-plus

社群媒體的分析流程也與前三者相同，差別在於需要搜尋 Facebook、Twitter 與 Google Plus 三個不同的平台，也因此需要分成三組不同的任務流程，每個任務流程有不同的關鍵字：

Facebook

("教授姓名" OR "First-Name Last-Name" OR ...) AND ("學校或機構名稱" OR ...) site:facebook.com

Google Plus

("教授姓名" OR "First-Name Last-Name" OR ...) AND ("學校或機構名稱" OR ...) site:plus.google.com

Twitter

("教授姓名" OR "First-Name Last-Name" OR ...) AND ("學校或機構名稱" OR ...) site:twitter.com

雖然關鍵字不同，但每一組任務流程都相同：

表 4-4：社群網站任務流程

順序	任務	ID	執行次數
1	計算搜尋結果總數量，與搜尋總頁面。	engine.count	3
2	將搜尋引擎的每個頁面抓下來暫存。	engine.fetch	3 – 30
3	抓取暫存的搜尋引擎頁面中的所有連結。	engine.parse	3 – 30
4	依序將所有連結的目標網頁抓下來，並分析內容。	page.analysis	0 – 3,000

page.analysis 階段會根據不同的社群媒體網站介面編寫不同的邏輯來作分析作者欄位，引用與被提及的判斷方式則與其他資料來源相似：

1. 根據不同社群媒體頁面抓取發文者姓名區域，判斷文字是否與學者姓名一樣，若是的話則為作者。
2. 根據姓名出現次數計算被引或被提及。在「參考文獻」或「參考書目」等詞彙之前出現的次數記入「被提及」，之後出現的每行計入一次「被引」。若是作者，則「被引」不計入，只計入「被提及」。

五、Google Scholar

資料來源 ID：scholar

搜尋引擎：google-scholar

資料類型：作者、被引

Google Scholar 的分析流程就與前面幾個資料來源不同了。首先 Google Scholar 是資料來源也是搜尋引擎，我們的目的是要分析上面所收錄的學者的著作。而 Google Scholar 本身提供直接用學者姓名檢索的功能，所以只要計算結果

數量即可得知著作篇數。運用 Google Scholar 自帶的作者搜尋，產生的關鍵字如下：

author:"學者姓名" OR author:"First-Name Last-Name" OR ...

而每條搜尋結果都包含「被引次數」的數據（如圖 4-5），所以系統也只要擷取這些數據加總即可，無須批次擷取所有網頁：

表 4-5：Google Scholar 任務流程

順序	任務	ID	執行次數
1	計算搜尋結果總數量，與搜尋總頁面。	engine.count	1
2	將搜尋引擎的每個頁面抓取下來暫存。	engine.fetch	1 – 10
3	從暫存的頁面抓取每個條目的被引次數數據，還有加總條目數量	scholar.count	1 – 10

Google Scholar search results for author:黃光國 OR author:Kwang-Kuo Hwang OR author:Hwang, Kwang. The results show three articles with their respective citation counts highlighted in red boxes:

- 文章 [引言] 中國人的權力遊戲
黃光國 - 1988 - ntur.lib.ntu.edu.tw
... 引文資訊. 資料載入中..... 請使用永久網址來引用或連結此文件: http://ntur.lib.ntu.edu.tw/handle/246246/228210. 題名: 中國人的權力遊戲. 作者: 黃光國. 日期: 1988-01. 上傳時間: 2011-02-15 16:59:09 (UTC+8). 出版者: 巨流. 顯示於類別: [心理學系] 專書. 文件中的檔案: ...
被引用 141 次 相關文章 全部共 2 個版本 引用 儲存
- [引言] 社會科學的理路
黃光國 - 2003 - 心理
被引用 65 次 相關文章 引用 儲存
- [引言] 《知識與行動: 中華文化傳統的社會心理詮釋》
黃光國 - 1995 - ntur.lib.ntu.edu.tw
... 資料載入中..... 請使用永久網址來引用或連結此文件: http://ntur.lib.ntu.edu.tw/handle/246246/228156. 題名: 《知識與行動: 中華文化傳統的社會心理詮釋》. 作者: 黃光國. 日期: 1995-01. 上傳時間: 2011-02-15 16:59:09 (UTC+8). 出版者: 心理出版社. 顯示於類別: [心理學系] 專書. ...
被引用 49 次 相關文章 全部共 2 個版本 引用 儲存

圖 4-5：Google Scholar 搜尋結果的被引用次數

由此處可發現，流程的前半段與其他資料來源是一樣的，但後第三步則改用 Google Scholar 自己所需要的邏輯。另外前兩個步驟流程雖然一樣，但因為是使用不同的搜尋引擎，其邏輯也會不同。為了要達到這種自動根據資料來源調整程式邏輯與任務流程的功能，採用「策略模式」整合「任務佇列」來進行就變成非常必要的手段，否則使用大量的 if...else 來手工控管程式流程會造成擴充性與可維護性變得非常低。詳細方式會在之後的章節繼續說明。

六、學位論文

資料來源 ID：thesis

搜尋引擎：artiti / ethesys

資料類型：被引次數、指導論文數

學位論文比較特殊的點在於一次用上了華藝線上圖書館（artiti）與 OAI 博碩士論文聯邦查詢系統（ethesys）兩個資料庫作為搜尋引擎，也是資料來源。任務會如同社群媒體一樣，會需要分兩線進行，被引次數與指導論文次數無法一次取得，也需要分成兩組任務。不過華藝並沒有被引的數據，只能取得指導論文次數。因此實際上會有三組任務分頭進行。關鍵字不必由我們來組合，只要根據查詢欄位將中英文姓名發出去給資料來源即可。

被引次數只搜尋博碩士論文聯邦查詢系統，由於系統本身支援直接以「參考文獻」欄位來搜尋，我們可以精確的取得「參考文獻」內含有學者著作的文獻，並直接抓取頁面上的結果數量即可，整個流程只需一個步驟：

表 4-6：學位論文被引次數任務流程

順序	任務	ID	執行次數
1	送出搜尋關鍵字，抓取頁面上的結果數量	thesis.cited.analysis	1

另外，華藝與博碩士論文聯邦查詢系統皆支援直接搜尋學者的指導論文，所以我們也是一樣送出搜尋指令後抓取結果數量即可，參考圖 4-6 與圖 4-7。

表 4-7：學位論文指導次數任務流程

順序	任務	ID	執行次數
1	送出搜尋關鍵字，抓取頁面上的結果數量	thesis.advisor.analysis	2

在這裡會與 Google Scholar 一樣，相同的任務行為（thesis.advisor.analysis）要用在不同的搜尋引擎與資料來源上，所以我們也是要倚賴「策略模式」來執行不同的邏輯。



圖 4-6：博碩士論文聯邦查詢系統的搜尋筆數結果



圖 4-7：華藝的搜尋筆數結果

七、TCI HSS

資料來源 ID：tci

搜尋引擎 ID：tci

資料類型：被引次數、著作篇數

TCI-HSS 同樣本身也是搜尋引擎，我們可以用系統自帶的搜尋功能去取得查詢數據。被引次數著作篇數需使用的關鍵字與查詢功能並不一樣，必須分成兩組任務個別執行。關鍵字與學位論文一樣，只需送出中英文姓名，無須組合搜尋字串。

被引次數需要兩個步驟，先抓取所有的搜尋結果頁面，再一一計算每個條目旁顯示的被引次數，然後加總。流程如下，一開頭的 engine.count 與之前的 Google 搜尋引擎是共用的：

表 4-8：TCI 被引次數任務流程

順序	任務	ID	執行次數
1	計算搜尋結果總數量，與搜尋總頁面。	engine.count	1
2	計算每個條目旁的引用次數，依序加總，參考圖 4-8。	tci.cited.analysis	1 – N

至於著作篇數也是可以在一個步驟內完成，只要送出查詢，並抓取頁面上的結果數量即可。

表 4-9：TCI 著作篇數任務流程

順序	任務	ID	執行次數
1	送出搜尋關鍵字，抓取頁面上的結果數量，參考圖 4-9。	tci.author.analysis	1



圖 4-8：TCI 每筆搜尋結果的被引用次數顯示



圖 4-9：TCI 搜尋結果總筆數

八、Web of Science

資料來源 ID：wos

搜尋引擎 ID：wos

資料類型：被引次數

Web of Science (WOS) 使用該平台所提供的 API (http://wokinfo.com/products_tools/products/related/webservices/) 來計算學者的被引次數。WOS 平台並未收入學者的中文姓名，皆是以英文為主，我們要給 WOS 的搜尋字串，參考 API 文件標準編寫會是：

```
AU=("First-Name" OR "Last-Name")
```

由於 WOS 提供的 cited 數據也是 Article-Level 的，我們需要先找出學者的所有文章清單，再一項一項的用該文章的 DOI 去抓取 cited 次數。

WOS API 的限制是一次只能取得 100 筆，所以當第一步以學者姓名搜尋著作時，我們需要先獲取總筆數，並除以 100 來得知應該要分成幾次分段查詢。接著依次查詢取得該學者的所有文章的 DOI 並存進資料庫，第三步則使用這些 DOI 去取得每篇著作的被引次數。

表 4-10：WOS 任務流程

順序	任務	ID	執行次數
1	以學者姓名搜尋，取得結果總筆數，並換算出總頁數。	wos.engine.records	1
2	將每頁查到的條目 DOI 依序存起來。	wos.engine.count	1 – N
3	以 DOI 取得每個條目引用次數，依序加總。	wos.cited.analysis	0 – N

九、Scopus

資料來源 ID：scopus

搜尋引擎 ID：scopus

資料類型：被引次數

Scopus 與 WOS 一樣使用該平台所提供的 API (<http://dev.elsevier.com/>) 來計算學者的被引次數。同樣也是 Article-Level 的情況下，需要根據作者名稱查詢著作列表並加總被引次數。根據文件說明，搜尋關鍵字如下：

AUTH(First-Name Last-Name)

由於 Scopus 的著作列表就已經包含了 cited-count 的資訊，所以可以比 WOS 少一個步驟即加總出被引次數。

Scopus API 的單頁條數限制是 25 筆，所以雖然步驟較少，但也會產生不少任務需要執行。

表 4-11：Scopus 任務流程

順序	任務	ID	執行次數
1	以學者姓名搜尋，取得結果總筆數，並換算出總頁數。	scopus.engine.records	1
2	將每頁查詢結果的條目被引數據加總起來。	scopus.cited.analysis	1 – N

十、Mendeley

資料來源 ID：mendeley

搜尋引擎 ID：mendeley

資料類型：被引次數

Mendeley 的查詢方式與 WOS 一樣，皆是靠官方提供的 API (<http://dev.mendeley.com/>)，查詢學者的著作，並加總每篇著作的被引次數，成為該學者的被引次數。Mendeley 無須組合關鍵字只要把英文姓名送出即可，由於 API 的限制最大只有前 100 筆查詢結果，因此不用分頁，一個步驟就能計算出結果（但也可能使得結果不夠精確）。

表 4-12：Mendeley 任務流程

順序	任務	ID	執行次數
1	取得每個條目引用次數，依序加總。	mendelet.cited.analysis	1

十一、總結

由以上的流程規劃，可發現因為各資料來源與搜尋引擎的特性皆不同，系統需要採用不同的策略與流程來抓取資料並分析。而有些資料來源因為共用搜尋引擎，又有部分流程相似可以共用。另外，根據資料來源或搜尋引擎的特性，有的任務只要執行一次即可取得足夠的數據，有的需要遍歷所有搜尋結果計算加總分數。故需要採用任務佇列模式，根據每個不同的資料來源，派發不同任務流程，才能滿足如此高彈性的需求，避免在程式中採用傳統的 if...else 方式，以提高開發與維護的彈性。

第四節 任務流程實作

一、任務佇列

任務佇列的進行方式，會是在每一次的任務結束後，根據當下的分析資料類型或資料來源，派發下一次的任務。由於在初始搜尋的時候，系統並不知道該關鍵字所搜尋出來的結果有多少筆，任務應該要執行幾次。因此每一次的任務結束時，都會根據這次任務的回傳結果，派發一定數量的新任務，丟入佇列中等待執行。

以「課程大綱 (syllabus)」為例，使用者輸入關鍵字完成之後，系統會根據課程大綱的需求先產生搜尋關鍵字，接著第一項任務 `engine.count` 會先送出查詢指令，計算 Google 中有多少個搜尋結果頁面，每頁 100 筆，最多 1000 筆，故最高可能有 10 個頁面，產生最多 10 個 `engine.fetch` 任務。

`engine.fetch` 任務是把搜尋引擎的頁面抓下來暫存，之後會再產生一個 `engine.parse` 任務，將頁面上的所有連結抓出來存進資料庫。因此每一個 `engine.fetch` 任務皆會產生 1 個 `engine.parse` 任務。

engine.parse 任務抓下搜尋結果頁面後，還要再產生 page.analysis 任務，實際去將搜尋到的外部網頁抓回系統中計算，由於每個頁面最多有 100 個連結，最多可能產生 100 個 page.analysis 任務。10 個 engine.parse 任務最多有可能產生 1000 個 page.analysis 任務。

在整個 Syllabus 的分析週期中，最多可能產生 1021 個任務在佇列中依次執行。最少則至少有 1 個任務會被執行（假設搜尋無結果），見表 4-13。

表 4-13：課程大綱任務流程

順序	任務	ID	執行次數
1	計算搜尋結果總數量，與搜尋總頁面。	engine.count	1
2	將搜尋引擎的每個頁面抓取下來暫存。	engine.fetch	1 - 10
3	抓取暫存的搜尋引擎頁面中的所有連結。	engine.parse	1 - 10
4	依序將所有連結的目標網頁抓下來，並分析內容。	page.analysis	0 - 1,000

這樣的任務流程確保了系統開發上的彈性與效能，當我們分析某個資料來源時，系統根據搜尋結果的多寡，產生的任務數量也不同，當搜尋結果較少時，不會每次都要跑完 1021 次任務流程，可以用相對較少的時間完成一項資料類型的分析。當分析不同的資料來源時，同樣是某個任務執行完成，也能夠依照不同的類型來派發適合的新任務。例如分析「社群網站 (social)」時，一開始 engine.count 就會區隔成三組分別執行 (Facebook, Google plus, Twitter)，每一組 engine.count 會根據其所負責的目標對象產生不同的關鍵字，並派發不同的任務流程。或者分析「Google Scholar」時，因為所需資料在搜尋引擎介面就能抓到，engine.fetch 任務完成之後，可以立即轉向 scholar.count 任務，直接計算出結果，省略後續 1000 次的資料分析。

二、策略模式

上一小節提到，任務佇列中的每個任務可能被不同資料類型共用，但又可以根據資料類型或搜尋引擎執行不同的分析行為，例如 Google、Google Scholar、TCI HSS 搜尋引擎共用 engine.count 任務，但實際上三者是不同網站，系統無法用同樣的程式去抓三個網站的資料，實作上必然不同。又 Syllabus 與 Google Scholar 前兩個任務相同，但在 engine.fetch 任務之後的流程就完全不同了。若我們要让每個任務執行的邏輯稍有不同，結束後派發的任務也有所不同，就要使用到設計模式中的「策略模式 (Strategy Pattern)」。

策略模式是一種程式中物件導向的設計法則，意味者程式有某個行為，在面對不同情境時，該行為有不同的演算法 (Wikipedia, 2015)。藉由策略模式，我們可以在課程大綱 (syllabus) 分析過程中，告訴 engine.count 任務採用抓取 Google 搜尋引擎的演算法。在 TCI HSS 的分析過程中，則告訴 engine.count 任務採用 tci 搜尋引擎的演算法。由以上規則可知，任何資料來源若有共用的任務行為，皆可用策略模式告知該任務使用不同的演算法，或告知該任務結束後應產生哪些新的子任務。這使得任務行為可以被重複使用，降低程式開發的成本，增加系統擴充彈性。

為了讓任務能夠知道應該使用何種策略執行工作，每個任務必須帶有「訊息 (Message)」，來讓任務行為能夠決策應該採行的策略。例如課程大綱 engine.fetch 的訊息可能就包含關鍵字、搜尋引擎、資料來源 ID 等等的資訊，來讓任務調整行為，這種佇列任務中帶有訊息的模式也稱為 Message Queue，可以讓每個任務自帶足夠的訊息讓系統執行。表 4-14 是本系統中會用到的訊息參數一覽：

表 4-14：佇列儲存的訊息參數

訊息名稱	訊息說明	範例
queue_id	這個任務在佇列中的編號	12157
task	任務 ID	engine.fetch
keyword	本次送入搜尋引擎的關鍵字	略
engine	本任務採用的搜尋引擎 ID	google
database	本任務採用的資料來源 ID	syllabus
page	本次 engine.fetch 要抓取的搜尋引擎頁面編號	6

三、任務執行過程

每一個任務執行過程會根據資料來源與搜尋引擎採用不同的演算法來處理，若我們要實作這種策略模式，最常用的就是建立多種物件分工的方式來處理。

當 12157 號任務開始執行時，此任務的 ID 為 engine.fetch，在程式中可能有一個對應的控制器（Controller）物件名為 EngineFetchController，此物件執行時，會將 keyword 送入搜尋引擎，抓取網頁頁面回來存放。Controller 是程式中一種控制主要邏輯的物件，通常是一個任務的主要入口點（Entry）。在我們的程式中，可以簡單的將所有任務都對應到一個 Controller 物件，例如 page.analysis 就對應到 PageAnalysisController，方便系統管理與維護。

搜尋引擎的演算法由 GoogleEngine 物件處理，這個物件會根據 google 搜尋引擎的特性，送出關鍵字查詢，並抓回 Google 的搜尋結果。我們告訴 GoogleEngine 這次的 keyword 並且要抓取的是 page 6，GoogleEngine 就根據這組參數去向搜尋引擎發出請求（request）。由於每個搜尋引擎的請求方式、參數名稱都不太一樣，故都會有一個對應的物件來處理，例如 TciEngine 或

GoogleScholarEngine 等。將搜尋引擎名稱直接對應在物件名稱上也能夠方便我們作管理。

關鍵字(keyword)會根據資料來源有所不同，抓下來的頁面分析邏輯也不同，所以每個資料來源也會有一個相關物件來處理，例如課程大綱的對應物件就是 SyllabusDatabase，專職用來提供 syllabus 的關鍵字組合以及分析結果頁面是否為課程大綱等等。Database 物件也會負責處理派發下個新任務的工作，因為任務的執行流程會依資料來源有所不同。

系統最終跑完所有任務後，所計算出的 altmetrics 數據，需要一個整合面板來顯示，如圖 4-10 規劃。

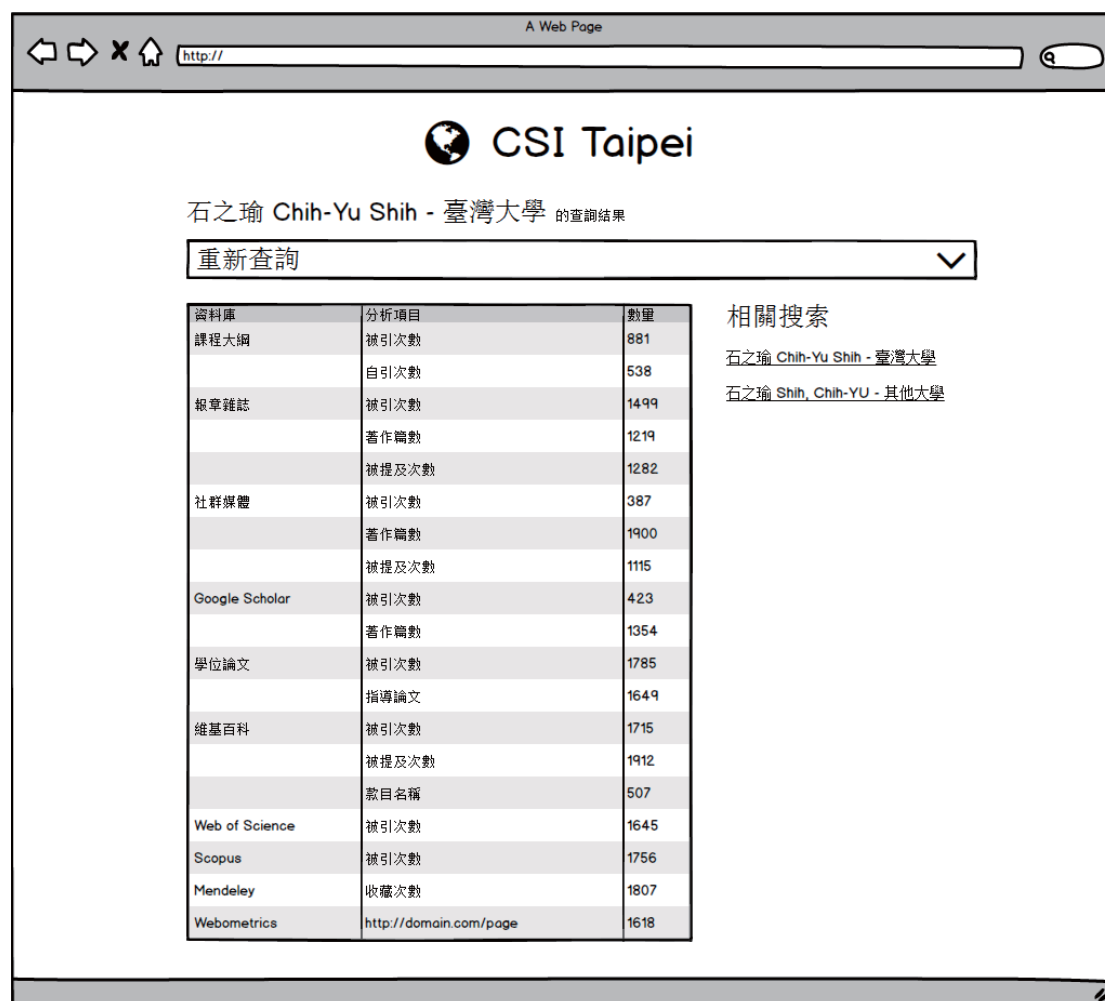


圖 4-10：查詢結果整合顯示介面

每組資料來源都有 2 到 3 種資料類型數據需要顯示，以 SyllabusDatabase 來說，每個頁面分析完成之後，就得出了 is_syllabus, cited 與 self_cited 三個數據，可以分別存入資料庫中，成為三筆紀錄。但不同的資料來源要顯示的數據都不同，在資料庫內的 ID 名稱也不一樣。因此最終的整合顯示面板，也需要對應的 Result 物件來處理。SyllabusResult 會抓出 is_syllabus, cited 與 self_cited 三組數據，GoogleScholarResult 則會抓出 author 與 cited 兩組數據。

綜合以上敘述，任務流程部分主要分成三層架構，相互支援。Controller 物件處理任務的進行，Engine 物件處理搜尋引擎的查詢與資料擷取方式，而 Database 責任較重，負責提供關鍵字與處理最終分析數據內容的演算法，還有任務派發。在每個資料來源的任務流程都不同的情況下，三組物件會交錯進行，分工處理不同的邏輯。這使得系統的開發維護變得更簡單，也更容易擴充新的資料來源。假設今天要新增一組新的資料來源 A，並使用搜尋引擎 B。則我們只要確認任務流程後，建立 BEngine 來抓取資料，與 ADatabase 來提供關鍵字與任務派發，就能自動形成一整組新的資料來源。若需要用到獨立的特殊任務模式 C，也可以額外建立 CController 來執行新任務。而最後在顯示結果時，還會有第四層 Result 來負責顯示計算結果，才能處理不同資料來源有不同的數據類型的狀況。

四、系統設計

依照前面所述之規則，可將每個資料來源的工作流程做出詳細設計。以下表 4-15 至表 4-23 列出各資料來源的完整任務流程，以及在系統中對應的程式處理物件。雙冒號「::」代表物件之下執行某項工作的函式（function），也稱方法（method）。

(1) 課程大綱 (syllabus)

表 4-15：課程大綱系統設計表

順序	任務 (控制器)	物件::方法	說明
1	submit	SyllabusDatabase::afterSubmit()	使用者送出查詢，派發 engine.count 任務
2	engine.count	SyllabusDatabase::getKeyword()	建立關鍵字
3	EngineCountController	GoogleEngine::getPage()	取得搜尋頁面
4		GoogleEngine::countPage()	計算有多少頁數
5		SyllabusDatabase::afterEngineCount()	根據頁數派發等量 engine.fetch 任務
6	engine.fetch EngineFetchController	GoogleEngine::getPage()	查詢對應的搜尋頁面，並抓取搜尋頁面 HTML
7		SyllabusDatabase::afterEngineFetch()	派發 engine.parse 任務
8	engine.parse EngineParseController	GoogleEngine::getLinks()	抓取搜尋結果頁面上的每一個連結
9		SyllabusDatabase::afterEngineParse()	根據連結數量派發 page.analysis 任務
10	page.analysis PageAnalysisController	SyllabusDatabase::analysis()	下載連結指向的目標網頁，並分析內容
整合顯示介面			
	result.view	SyllabusResult::isSyllabus()	顯示 is_syllabus 數據
		SyllabusResult::cited()	顯示 cited 數據
		SyllabusResult::selfCited()	顯示 self_cited 數據

(2) 報章雜誌 (paper)

表 4-16：報章雜誌系統設計表

順序	任務 (控制器)	物件::方法	說明
1	submit	PaperDatabase::afterSubmit()	使用者送出查詢，派發 engine.count 任務
2	engine.count	PaperDatabase::getKeyword()	建立關鍵字
3	EngineCountController	GoogleEngine::getPage()	取得搜尋頁面
4		GoogleEngine::countPage()	計算有多少頁數
5		PaperDatabase::afterEngineCount()	根據頁數派發等量 engine.fetch 任務
6	engine.fetch	GoogleEngine::getPage()	查詢對應的搜尋頁面，並抓取搜尋頁面 HTML
7	EngineFetchController	PaperDatabase::afterEngineFetch()	派發 engine.parse 任務
8	engine.parse	GoogleEngine::getLinks()	抓取搜尋結果頁面上的每一個連結
9	EngineParseController	PaperDatabase::afterEngineParse()	根據連結數量派發 page.analysis 任務
10	page.analysis	PaperDatabase::analysis()	下載連結指向的目標網頁，並分析內容
整合顯示介面			
	result.view	PaperResult::cited()	顯示 cited 數據
		PaperResult::author()	顯示 author 數據
		PaperResult::mentioned()	顯示 mentioned 數據

(3) 維基百科 (wiki)

表 4-17：維基百科系統設計表

順序	任務 (控制器)	物件::方法	說明
1	submit	WikiDatabase::afterSubmit()	使用者送出查詢，派發 engine.count 任務
2	engine.count	WikiDatabase::getKeyword()	建立關鍵字
3	EngineCountController	GoogleEngine::getPage()	取得搜尋頁面
4		GoogleEngine::countPage()	計算有多少頁數
5		WikiDatabase::afterEngineCount()	根據頁數派發等量 engine.fetch 任務
6	engine.fetch EngineFetchController	GoogleEngine::getPage()	查詢對應的搜尋頁面，並抓取搜尋頁面 HTML
7		WikiDatabase::afterEngineFetch()	派發 engine.parse 任務
8	engine.parse EngineParseController	GoogleEngine::getLinks()	抓取搜尋結果頁面上的每一個連結
9		WikiDatabase::afterEngineParse()	根據連結數量派發 page.analysis 任務
10	page.analysis PageAnalysisController	WikiDatabase::analysis()	下載連結指向的目標網頁，並分析內容
整合顯示介面			
	result.view	WikiResult::entry()	顯示 entry 數據
		WikiResult::cited()	顯示 cited 數據
		WikiResult::mentioned()	顯示 mentioned 數據

(4) 社群媒體

表 4-18：社群媒體系統設計表

順序	任務 (控制器)	物件::方法	說明
1	submit	SocialDatabase::afterSubmit()	使用者送出查詢，派發 engine.count 任務 (此時即分流出三個不同的社群媒體，會有三個 engine.count 任務)
2	engine.count	SocialDatabase::getKeyword()	建立關鍵字
3	EngineCountController	GoogleEngine::getPage()	取得搜尋頁面
4		GoogleEngine::countPage()	計算有多少頁數
5		SocialDatabase::afterEngineCount()	根據頁數派發等量 engine.fetch 任務
6	engine.fetch EngineFetchController	GoogleEngine::getPage()	查詢對應的搜尋頁面，並抓取搜尋頁面 HTML
7		SocialDatabase::afterEngineFetch()	派發 engine.parse 任務
8	engine.parse EngineParseController	GoogleEngine::getLinks()	抓取搜尋結果頁面上的每一個連結
9		SocialDatabase::afterEngineParse()	根據連結數量派發 page.analysis 任務
10	page.analysis PageAnalysisController	SocialDatabase::analysis()	下載連結指向的目標網頁，並分析內容
整合顯示介面			
	result.view	SocialResult::author()	顯示 author 數據
		SocialResult::cited()	顯示 cited 數據
		SocialResult::mentioned()	顯示 mentioned 數據

(5) Google Scholar

表 4-19：Google Scholar 系統設計表



順序	任務 (控制器)	物件::方法	說明
1	submit	GoogleScholarDatabase::afterSubmit()	使用者送出查詢，派發 engine.count 任務
2	engine.count	GoogleScholarDatabase::getKeyword()	建立關鍵字
3	EngineCountController	GoogleScholarEngine::getPage()	取得搜尋頁面
4		GoogleScholarEngine::countPage()	計算有多少頁數
5		GoogleScholarDatabase::afterEngineCount()	根據頁數派發等量 engine.fetch 任務
6	engine.fetch EngineFetchController	GoogleScholarEngine::getPage()	查詢對應的搜尋頁面，並抓取搜尋頁面 HTML
7		GoogleScholarDatabase::afterEngineFetch()	派發 engine.parse 任務
8	scholar.count ScholarCountController	GoogleScholarDatabase::count()	計算頁面的被引，還有加總條目數量
整合顯示介面			
	result.view	GoogleScholarResult::author()	顯示 author 數據
		GoogleScholarResult::cited()	顯示 cited 數據

(6) 學位論文

表 4-20：學位論文系統設計表

順序	任務 (控制器)	物件::方法	說明
1	submit	ThesisDatabase::afterSubmit()	使用者送出查詢，派發 thesis.cited.analysis 與 thesis.advisor.analys is 任務
被引次數			
2	thesis.cited.analysis	EthesysEngine::getCitedKeyword()	建立關鍵字
3	ThesisCitedAnalysisController	EthesysEngine::getCitedPage()	取得搜尋頁面
4		EthesysEngine::countCited()	從上一個動作中下載的 HTML 提取搜尋結果數量
指導論文			
2	thesis.advisor.analysis ThesisAdvisorAnalysisControll er	EthesysEngine::getAdvisorKeyword (ArtitiEngine::getAdvisorKeyword()	建立關鍵字 (由 queue 內儲存的 message 決定使用哪一個 Engine 物件)
3		EthesysEngine::getAdvisorPage() ArtitiEngine::getAdvisorPage()	取得搜尋頁面
4		EthesysEngine::countAdvisor() ArtitiEngine::countAdvisor()	從上一個動作中下載的 HTML 提取搜尋結果數量
整合顯示介面			
	result.view	ThesisResult::advisor()	顯示 advisor 數據
		ThesisResult::cited()	顯示 cited 數據

(7) TCI HSS

表 4-21：TCI HSS 系統設計表

順序	任務 (控制器)	物件::方法	說明
1	submit	TciDatabase::afterSubmit()	使用者送出查詢，派發 engine.count 與 tci.author.analysis 任務
被引次數			
2	engine.count	TciEngine::getCitedKeyword()	建立關鍵字
3	EngineCountController	TciEngine::getPage()	取得搜尋頁面
4		TciEngine::countPage()	計算有多少頁面
5		TciDatabase::afterEngineCount()	派發 tci.cited.analysis 任務
6	tci.cited.analysis	TciEngine::getPage()	下載搜尋結果頁面。
7	TciCitedAnalysisController	TciDatabase::countCited()	抓取每個條目旁的 cited 數據加總
指導論文			
2	tci.author.analysis	TctEngine::getPage()	下載 author 搜尋頁面
3	TciAuthosAnalysisController	TciDatabase::countAuthors()	直接取得筆數總計
整合顯示介面			
	result.view	ThesisResult::author()	顯示 author 數據
		ThesisResult::cited()	顯示 cited 數據

(8) Web of Science

表 4-22：Web of Science 系統設計表



順序	任務 (控制器)	物件::方法	說明
1	submit	WosDatabase::afterSubmit()	使用者送出查詢，派發 wos.engine.records 任務
2	wos.engine.records	WosDatabase::getAuthorKeyword()	建立作者名稱關鍵字
3	WosEngineRecordsController	WosEngine::search()	用 API 取得作者著作搜尋 結果
4		WosEngine::countPages()	計算結果總數與頁面數
5		WosDatabase::afterEngineCount()	根據頁數派發等量 wos.engine.count 任務
6	wos.engine.count EngineCountController	WosEngine::getPage()	一頁一頁的將搜尋結果與 條目 DOI 儲存起來
7		WosDatabase::afterEngineCount()	派發 wos.cited.analysis 任務
8	wos.cited.analysis WosCitedAnalysisController	WosEngine::getCitedByDOI()	依前面所儲存的著作 DOI 去取得 cited 數據，加總 起來
整合顯示介面			
	result.view	WosResult::cited()	顯示 cited 數據

(9) Scopus

表 4-23：Scopus 系統設計表

順序	任務 (控制器)	物件::方法	說明
1	submit	ScopusDatabase::afterSubmit()	使用者送出查詢，派發 scopus.engine.records 任務
2	wos.engine.records	ScopusDatabase::getAuthorKeyword()	建立作者名稱關鍵字
3	WosEngineRecordsController	ScopusEngine::search()	用 API 取得作者著作搜尋 結果
4		ScopusEngine::countPages()	計算結果總數與頁面數
5		ScopusDatabase::afterEngineCount()	根據頁數派發等量 scopus.cited.analysis 任務
6	wos.engine.count	ScopusEngine::getPage()	取得每一頁搜尋結果
7	EngineCountController	ScopusDatabase::countCited()	將搜尋到的條目被引次數 加總
整合顯示介面			
	result.view	ScopusResult::cited()	顯示 cited 數據

(10) Mendeley

表 4-24：Mendeley 系統設計表

順序	任務 (控制器)	物件::方法	說明
1	submit	MendeleyDatabase::afterSubmit())	使用者送出查詢，派發 mendeley.cited.analysis 任務
2	mendeley.cited.analysis MendeleyCitedAnalysisController	MendeleyDatabase::getKeyword()	建立關鍵字
3		MendeleyEngine::search()	用 API 取得作者著作搜尋結果
4		MendeleyDatabase::countCited()	加總 cited 數據
整合顯示介面			
	result.view	MendeleyResult::cited()	顯示 cited 數據

第五節 實際畫面導覽

以下將一一展示實際開發後的系統畫面預覽，並介紹實作過程中多出來的一些協助操作者更容易管理系統的額外功能項目。

一、前台畫面

圖 4-11 是系統的首頁，由於本系統包含了被引用的統計資料，以及社群媒體上的影響力係數，因此稱之為 Citation and Social Impact，簡稱為 CSI Taipei 系統，可以從首頁的標題中看見。首頁主要是查詢的入口，使用者可以在此輸入學者的中英文名稱與學校名稱（皆為必填欄位，英文至少要填一組）。接著可以勾選要查詢的資料來源，若勾選了 Webometrics，下方會出現個人網站網址的欄位，也是必填，且最多三個網址。

CSI Taipei

Home

Login

帳號

密碼

登入

註冊

CSI Taipei (citation and social impact)

請輸入檢索字串

* Chinese Name

石之瑜

* English Name

Chih-Yu

Shih

First Name

Last Name

First Name

Last Name

* School

台大

請選擇檢索項目

☒ 課程大綱
 ☒ 報章雜誌
 ☒ 社群媒體
 ☒ Google Scholar
 ☒ 學位論文
 ☒ 維基百科
 ☒ Web of Science
 ☒ Scopus
 ☒ Mendeley
 ☒ Webometrics

個人網站網址

http://politics.ntu.edu.tw/?p=143

Enter URL

圖 4-11：系統首頁

CSI Taipei

Home

Login

帳號

密碼

登入

註冊

您現在搜索的是:

石之瑜;Chih-Yu Shih::國立臺灣大學;臺大

重新搜尋 ▼

相關搜索

檢索結果

資料庫	分析項目	數量
課程大綱	被引次數	142
	自引次數	21
報章雜誌	被引次數	0
	著作篇數	3
	被提及次數	4078
社群媒體	被引次數	0
	著作篇數	0
	被提及次數	0

圖 4-12：搜尋結果頁面

資料庫	分析項目	數量
課程大綱	被引次數	142
	自引次數	21
報章雜誌	被引次數	0
	著作篇數	3
	被提及次數	4078
社群媒體	被引次數	0
	著作篇數	0
	被提及次數	0
Google Scholar	被引次數	1051
	著作篇數	548
學位論文	被引次數	279
	指導論文	107
維基百科	被引次數	2
	被提及次數	47
	數目名稱	1
Web of Science	被引次數	131
Scopus	被引次數	20
Mendeley	收藏次數	0
Webometrics	http://politics.ntu.edu.tw/?p=143	48

圖 4-13：詳細的搜尋結果列表

圖 4-12 是系統的搜尋結果頁面，會顯示目前搜尋的是哪個學者，以及其搜尋的目的學校（避免不同學校的同名學者造成誤判），學校與學者名稱會經過正規化正確的名稱，因此與使用者輸入的內容不會一模一樣。重新搜尋的下拉方塊可以點擊後拉開，會出現與首頁一樣的搜尋表單，可以重新再搜尋一次其他學者的資料。圖 4-13 顯示了完整的搜尋結果列表（系統完成後測試時，TCI 因故無法連線取得資料，因此沒有顯示 TCI 的數據）。

其中，有四個資料來源，分別是「課程大綱」、「報章雜誌」、「社群媒體」與「維基百科」，因為是用搜尋引擎抓取大量網頁再將分析數據加總起來的，因此這四個資料來源的頁面資料可以被瀏覽，並經過人工檢視後可以調整數據。圖

4-14到圖 4-17 分別是以以上四個資料來源的頁面總覽與調整頁面數據的控制項(每個資料來源的控制項都不同)。



CSI	Home	Login	登出
-----	------	-------	----

頁面總覽 石之瑜;Chih-Yu Shih::國立臺灣大學;臺大 [syllabus]

搜尋 搜尋工具 清除 排序表格依照: 100

☐	標題	檔案類型	被引	自引	是課程大綱	詳情	ID
☐	政治心理學 - 課程大綱 https://nol.ntu.edu.tw/nol/coursesearch/print_table.php?course_id=302 1090 o&class=&dpt_code=3020&ser_no=77937&semester=102-1	html	▼ 2 ▲	▼ 0 ▲	✓	詳情	16593
☐	政治心理學 - 課程大綱 https://nol.ntu.edu.tw/nol/coursesearch/print_table.php?course_id=302 1090 o&class=&dpt_code=3023&ser_no=4773&semester=96-1	html	▼ 0 ▲	▼ 1 ▲	✓	詳情	16594
☐	政治心理學 - 課程大綱 https://nol.ntu.edu.tw/nol/coursesearch/print_table.php?course_id=302 1090 o&class=&dpt_code=3023&ser_no=4773&semester=96-1	html	▼ 0 ▲	▼ 1 ▲	✓	詳情	16595
☐	思想與中國專題研究 - 課程大綱 https://nol.ntu.edu.tw/nol/coursesearch/print_table.php?course_id=322 D44 o&class=&dpt_code=3220&ser_no=73806&semester=103-1	html	▼ 0 ▲	▼ 1 ▲	✓	詳情	16596
☐	政治心理學 - 課程大綱 https://nol.ntu.edu.tw/nol/coursesearch/print_table.php?course_id=302 1090 o&class=&dpt_code=3021&ser_no=4987&semester=97-1	html	▼ 0 ▲	▼ 1 ▲	✓	詳情	16597
☐	政治心理學 - 課程大綱 https://nol.ntu.edu.tw/nol/coursesearch/print_table.php?course_id=302 1090 o&class=&dpt_code=3020&ser_no=22434&semester=100-1	html	▼ 0 ▲	▼ 1 ▲	✓	詳情	16598

圖 4-14：課程大綱頁面總覽

CSI	Home	Login	登出
-----	------	-------	----

頁面總覽 石之瑜;Chih-Yu Shih::國立臺灣大學;臺大 [paper]

搜尋 搜尋工具 清除 排序表格依照: 100

☐	標題	檔案類型	作者	被引	被提及	詳情	ID
☐	自由共和國》石之瑜 蔡英文千萬當心朱立倫- 自由電子報自由評論網 http://talk.ltn.com.tw/article/paper/922960	html	✗	▼ 0 ▲	✓	詳情	16817
☐	石之瑜文化暴發戶- 自由電子報自由評論網 - 自由時報 http://talk.ltn.com.tw/article/paper/87789	html	✗	▼ 0 ▲	✓	詳情	16818
☐	自由共和國》石之瑜憑什麼批評李登輝? - 自由電子報自由評論網 http://talk.ltn.com.tw/article/paper/90348	html	✗	▼ 0 ▲	✓	詳情	16819
☐	自由共和國》石之瑜國民黨應放棄2016年提名- 自由電子報自由評論網 http://talk.ltn.com.tw/article/paper/859194	html	✗	▼ 0 ▲	✓	詳情	16820
☐	自由共和國》石之瑜朱習會後旗號問題將惡化- 自由電子報自由評論網 http://talk.ltn.com.tw/article/paper/879295	html	✗	▼ 0 ▲	✓	詳情	16821
☐	石之瑜北京對海峽中線切實關? - 自由電子報自由評論網 - 自由時報 http://talk.ltn.com.tw/article/paper/848698	html	✗	▼ 0 ▲	✓	詳情	16822
☐	依附美日台灣將領失戰略空間 民意論壇 評論 聯合新聞網 http://udn.com/news/story/7339/1274461-依附美日-台灣將領失戰略空間	html	✗	▼ 0 ▲	✓	詳情	16823

圖 4-15：報章雜誌頁面總覽

CSI Home Login

登入

頁面總覽 石之瑜;Chih-Yu Shih::國立臺灣大學;臺大 [social]

搜尋

搜尋工具

清除

排序表格依照：

100

標題	檔案類型	作者	被引	被提及	詳情	ID
臺大校友聯誼會-【儒家與民主】--石之瑜教授演講... - Facebook https://zh-tw.facebook.com/alumni.NTU/posts/105197378439294	html		0		詳情	17535
海峽對話沙龍-釣魚台與東海問題專題... - Facebook https://zh-tw.facebook.com/APYPW/posts/1066156753133	html		0		詳情	17536
動感時報相片- 臺大論壇-跨科際論壇平臺北區論壇 Facebook https://zh-tw.facebook.com/forum.ntu/photos/a.1419301026141330126.140940839379736/169387469848406/	html		0		詳情	17537
亞太青年公益協會(籌建中) - 海峽對話沙龍-釣魚台與東海問題專題... https://www.facebook.com/APYPW/posts/1066156753133?stream_ref=5	html		0		詳情	17538
亞太青年公益協會(籌建中) - 海峽對話沙龍-釣魚台與東海問題專題... https://zh-cn.facebook.com/APYPW/posts/1066156753133?stream_ref=10	html		0		詳情	17539
亞太青年公益協會(籌建中) - Facebook https://zh-tw.facebook.com/APYPW/?filter=2	html		0		詳情	17540
退休人員聯誼會/臺大教職員工文康會 https://www.facebook.com/retireepage/posts/48568268839837	html		0		詳情	17541

圖 4-16：社群媒體頁面總覽

CSI Home Login

登入

頁面總覽 石之瑜;Chih-Yu Shih::國立臺灣大學;臺大 [wiki]

搜尋

搜尋工具

清除

排序表格依照: 100

標題	檔案類型	被引	被提及	款目名稱	詳情	ID
石之瑜- &#32500;基百科, 自由的百科全&#20070; https://zh.wikipedia.org/wiki/石之瑜	html	0			詳情	17628
臺灣知名外省人列表- &#32500;基百科, 自由的百科全&#20070; https://zh.wikipedia.org/wiki/臺灣知名外省人列表	html	0			詳情	17629
費希平- &#32500;基百科, 自由的百科全&#20070; https://zh.wikipedia.org/wiki/費希平	html	0			詳情	17630
胡佛(臺灣)- &#32500;基百科, 自由的百科全&#20070; https://zh.wikipedia.org/wiki/胡佛_(臺灣)	html	0			詳情	17631
國家講座- &#32500;基百科, 自由的百科全&#20070; https://zh.wikipedia.org/wiki/國家講座	html	0			詳情	17632
反中- &#32500;基百科, 自由的百科全&#20070; https://zh.wikipedia.org/wiki/反中	html	0			詳情	17633
陳良基- &#32500;基百科, 自由的百科全&#20070; https://zh.wikipedia.org/wiki/陳良基	html	0			詳情	17634

圖 4-17：維基百科頁面總覽

列表上的標題點擊進去後，會進到單一頁面詳細資料，可以預覽該頁面的內容，也可以觀察數據調整的歷史紀錄，下右下方可以留言討論（採用 Facebook 留言系統，需要有 Facebook 帳號才能發言），見圖 4-18。

頁面詳情

政治心理學 - 課程大綱

◀ 回前頁

搜尋資訊

搜尋結果 石之瑜;Chih-Yu Shih::國立臺灣大學;臺大

URL https://nol.ntu.edu.tw/nol/coursesearch/print_table.php?course_id=302.20900&class=&dpt_code=3020&ser_no=77937&semester=102-1

頁面預覽

課程資訊	
課程名稱	政治心理學 Political Psychology
開課學期	102-1
授課對象	社會科學院 政治學系
授課教師	黃曼華
課號	PS3109
課程識別碼	302.20900
班次	

修改紀錄

- admin 將此頁面的 被引 認為 是 - 2015-12-28 16:22:22
- admin 將此頁面的 被引 認為 是 - 2015-12-28 16:22:29
- admin 將此頁面的 是課程大綱 認為 否 - 2015-12-28 16:22:22
- admin 將此頁面的 是課程大綱 認為 是 - 2015-12-28 16:22:25

留言

0則回應 排序依據 最新 ▼

新增回應.....

Facebook Comments Plugin

圖 4-18：單一頁面詳細資料

二、後台畫面

後台沒有太多實際操作功能，主要是給管理員監控系統運行狀態，方便移除錯誤或過期條目用的。

Entry List

New
Edit
Duplicate
Publish
Unpublish
Check In
Delete
Batch

Menus

- Category
- Enginepage List
- Entry List**
- History List
- Notify List
- Page List
- Queue List
- Result List

Search

Search tools ▼

Clear

	Title
☐	蔡怡君;Yi-Chun Tsai
☐	黃光國;Kwang-Kuo Hwang
☐	石之瑜;Chih-Yu Shih::國立臺灣大學;臺大

圖 4-19：搜尋條目管理

沒有被搜尋過，就
要採用該名稱的搜

Joomla!

Duplicate

☒ Publish

☐ Unpublish

☒ Check In

☒ Delete

☒ Options

Search

Search tools ▾

Clear

Sort Table By: ▾

100 ▾

☐	Entry ID	Engine	Title	Keyword	ID ▴
☐	11	wos	黃光臨 Kwang-Kuo Hwang [wos]	All("Kwang-Kuo Huang" OR "Huang, Kwang-Kuo")	20
☐	17	google	石之瑜 Chih-Yu Shih. 國立臺灣大學 臺大 [syllabus]	("石之瑜" OR "Chih-Yu Shih" OR "Shih, Chih-Yu") AND (課程大綱 OR "Syllabus")	45
☐	17	google	石之瑜 Chih-Yu Shih. 國立臺灣大學 臺大 [paper]	(intext: "石之瑜" OR "Chih-Yu Shih" OR "Shih, Chih-Yu") (intext: 教授 OR intext: 博士) AND intitle: "選刊OR選書OR月刊OR雜誌OR期刊OR通訊OR評論OR新聞OR經" AND -inurl:youtube AND -inurl:roodo AND -inurl:blog AND -inurl:pixnet AND -inurl:books AND -inurl:blogspot AND -inurl:myblog AND -inurl:wordpress AND -inurl:angle AND -inurl:sharing AND -inurl:waterlike AND -inurl:bid AND -inurl:taaze AND -inurl:pchome AND -inurl:ptt AND -inurl:forum AND (inurl:"com" OR "gov")	46
☐	17	google	石之瑜 Chih-Yu Shih. 國立臺灣大學 臺大 [social]	{"facebook.com": "(\"u77f3\u04e4bu745c\" OR \"Chih-Yu Shih\" OR \"Shih, Chih-Yu\")\" (\\"u570bu7acb\u081fa\u0706\u05927\u05b78\" OR \"\u081fa\u05927\") site:facebook.com\", \"plus.google.com\": \"(\\"u77f3\u04e4bu745c\" OR \"Chih-Yu Shih\" OR \"Shih, Chih-Yu\")\" (\\"u570bu7acb\u081fa\u0706\u05927\u05b78\" OR \"\u081fa\u05927\") site:plus.google.com\", \"twitter.com\": \"	47

圖 4-20：任務管理

Components
Extensions
Media Manager

CSI
Joomla!

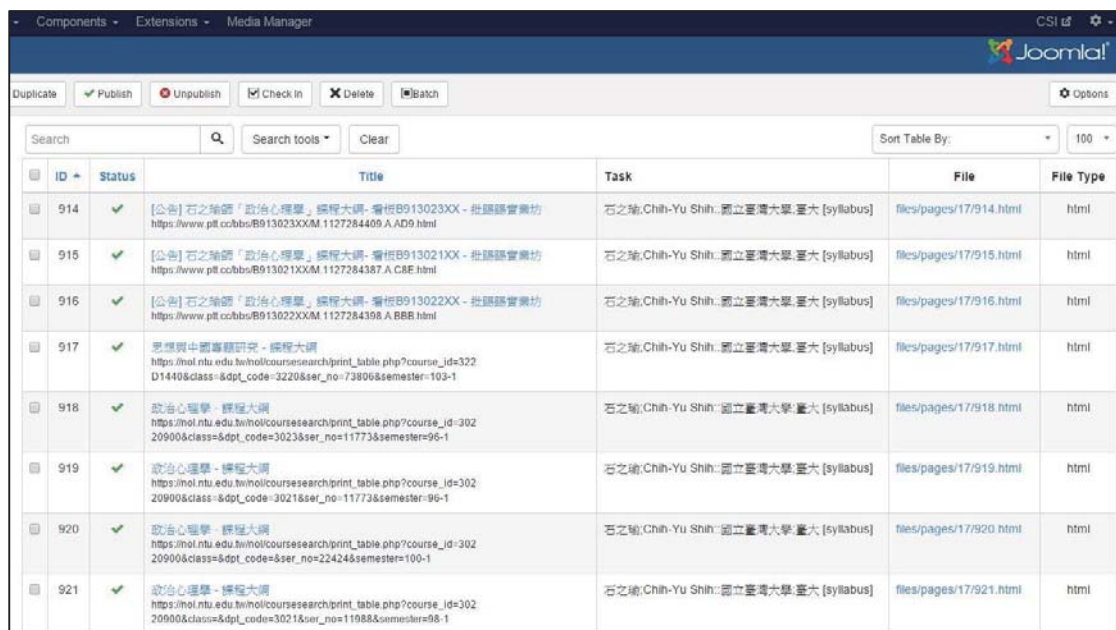
Duplicate
Publish
Unpublish
Check In
Delete
Batch
Options

Search
Search tools
Clear
Sort Table By:
100

	Status	Entry ID	Task ID	Entry	Task	URL	File	Total	Ordering	ID
	✓	17	45	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大	syllabus	https://www.google.com.tw/sear...	files/enginepages/17/24.html	7	1	24
	✓	17	45	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大	syllabus	https://www.google.com.tw/sear...	files/enginepages/17/25.html	7	2	25
	✓	17	45	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大	syllabus	https://www.google.com.tw/sear...	files/enginepages/17/26.html	7	3	26
	✓	17	45	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大	syllabus	https://www.google.com.tw/sear...	files/enginepages/17/27.html	7	4	27
	✓	17	45	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大	syllabus	https://www.google.com.tw/sear...	files/enginepages/17/28.html	7	5	28
	✓	17	45	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大	syllabus	https://www.google.com.tw/sear...	files/enginepages/17/29.html	7	6	29
	✓	17	45	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大	syllabus	https://www.google.com.tw/sear...	files/enginepages/17/30.html	7	7	30
	✓	17	46	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大	paper	https://www.google.com.tw/sear...	files/enginepages/17/31.html	1	1	31
	✓	17	47	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大	social	https://www.google.com.tw/sear...	files/enginepages/17/32.html	2	1	32
	✓	17	47	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大	social	https://www.google.com.tw/sear...	files/enginepages/17/33.html	2	2	33
	✓	17	47	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大	social	https://www.google.com.tw/sear...	files/enginepages/17/34.html	1	1	34

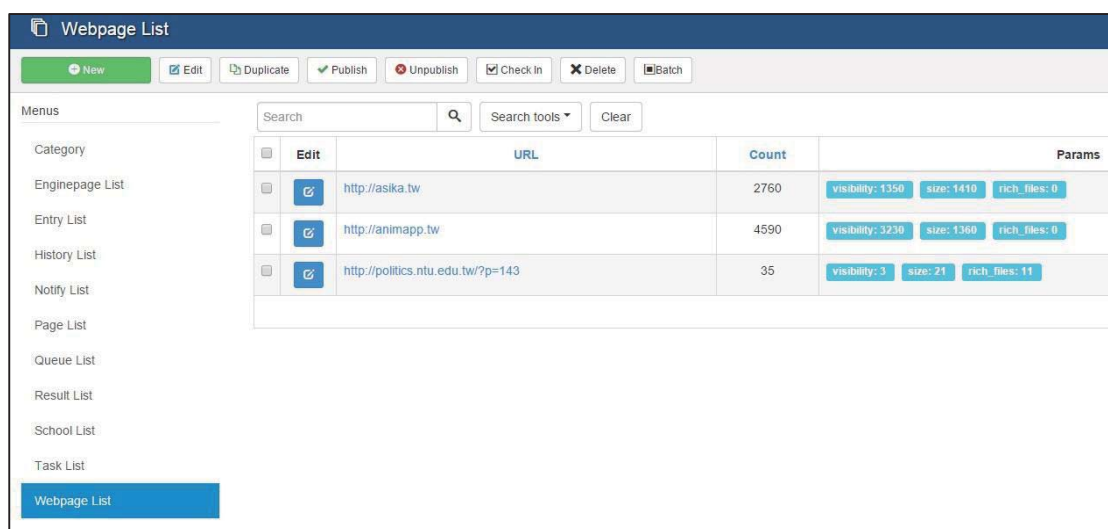
圖 4-21：搜尋引擎頁面管理

圖 4-21 與圖 4-22 分別是搜尋引擎頁面與結果頁面的列表，僅只是快取起來供查閱而已。圖 4-23 則是 Webometrics 的網址快取，由於 Webometrics 每一次都要發出多組查詢來抓取結果數量，系統會將相同的網址做快取，立刻就能 Webometrics 的計算結果。若一個網址快取超過 30 日則重新計算一次。



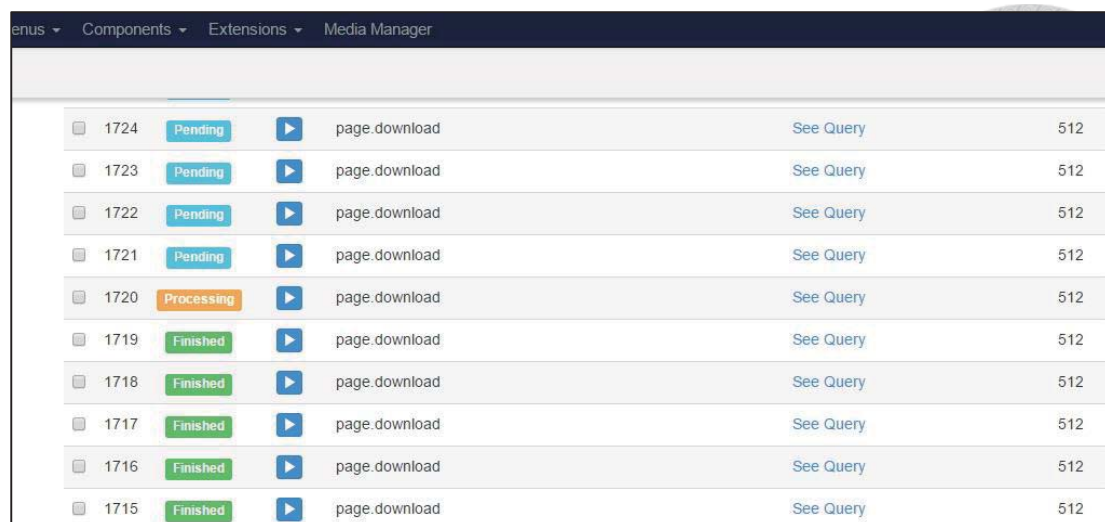
ID	Status	Title	Task	File	File Type
914	✓	[公告] 石之瑜師「政治心理學」課程大綱 - 增修B913023XX - 批語語彙彙訪 https://www.ptt.cc/bbs/B913023XXM.1127284409.A.AD9.html	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大 [syllabus]	files/pages/17/914.html	html
915	✓	[公告] 石之瑜師「政治心理學」課程大綱 - 增修B913021XX - 批語語彙彙訪 https://www.ptt.cc/bbs/B913021XXM.1127284387.A.C8E.html	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大 [syllabus]	files/pages/17/915.html	html
916	✓	[公告] 石之瑜師「政治心理學」課程大綱 - 增修B913022XX - 批語語彙彙訪 https://www.ptt.cc/bbs/B913022XXM.1127284398.A.BBB.html	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大 [syllabus]	files/pages/17/916.html	html
917	✓	思想與中國專題研究 - 課程大綱 https://mol.ntu.edu.tw/mol/coursesearch/print_table.php?course_id=322 D1440&class=&dpt_code=3220&ser_no=73806&semester=103-1	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大 [syllabus]	files/pages/17/917.html	html
918	✓	政治心理學 - 課程大綱 https://mol.ntu.edu.tw/mol/coursesearch/print_table.php?course_id=302 20900&class=&dpt_code=3023&ser_no=11773&semester=96-1	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大 [syllabus]	files/pages/17/918.html	html
919	✓	政治心理學 - 課程大綱 https://mol.ntu.edu.tw/mol/coursesearch/print_table.php?course_id=302 20900&class=&dpt_code=3021&ser_no=11773&semester=96-1	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大 [syllabus]	files/pages/17/919.html	html
920	✓	政治心理學 - 課程大綱 https://mol.ntu.edu.tw/mol/coursesearch/print_table.php?course_id=302 20900&class=&dpt_code=&ser_no=22424&semester=100-1	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大 [syllabus]	files/pages/17/920.html	html
921	✓	政治心理學 - 課程大綱 https://mol.ntu.edu.tw/mol/coursesearch/print_table.php?course_id=302 20900&class=&dpt_code=3021&ser_no=11988&semester=98-1	石之瑜:Chih-Yu Shih::國立臺灣大學:臺大 [syllabus]	files/pages/17/921.html	html

圖 4-22：搜尋結果頁面管理



Edit	URL	Count	Params
	http://asika.tw	2760	visibility: 1350 size: 1410 rich_files: 0
	http://animapp.tw	4590	visibility: 3230 size: 1360 rich_files: 0
	http://politics.ntu.edu.tw?p=143	35	visibility: 3 size: 21 rich_files: 11

圖 4-23：Webometrics 網址快取管理



ID	Status	Action	Details	Count
1724	Pending	▶	page.download	512
1723	Pending	▶	page.download	512
1722	Pending	▶	page.download	512
1721	Pending	▶	page.download	512
1720	Processing	▶	page.download	512
1719	Finished	▶	page.download	512
1718	Finished	▶	page.download	512
1717	Finished	▶	page.download	512
1716	Finished	▶	page.download	512
1715	Finished	▶	page.download	512

圖 4-24：佇列管理

最後還有佇列管理，見圖 4-24。由於系統查詢過程會產生超過上千筆任務佇列，佇列管理可以幫助管理員確認佇列的執行狀態、完成度、優先權、除錯訊息等等，也可以指定某個任務重新執行來確認分析結果，當系統開始全自動跑任務流程時，管理員只有依賴佇列管理介面才能夠掌握系統的狀況。

第五章 結論與未來展望



第一節 結論與貢獻

本研究以 Altmetrics 替代計量學為基礎設計了分析用的系統，從各個不同的資料來源擷取資料，並分析學者在網路與非正式學術交流中的活動狀況。由於 Altmetrics 的資料來源不像傳統文獻計量學直接來自各大資料庫，因此系統必須如爬蟲一般適應各大搜尋引擎與資料來源網站，其任務執行流程差異非常巨大，編寫上並不容易。引入任務佇列（Task Queue）的優點在於更容易設計每個資料來源網站的操作流程，由前一個任務指派下一個任務，而非傳統的 if...else 的流程控制。讓系統能夠像人類一樣，根據不同的網站，進入特定網址輸入關鍵字，並抓取頁面上的特定數據。任務佇列的缺點在於產生的任務數量極為龐大，開發過程一旦系統開始一項分析流程，要抓到中間出錯的某一個任務佇列，就要花費較多人工去檢視，但整體而言在一個合適規模的團隊下，將能夠輕鬆地掌握這種任務執行模式。採用任務佇列的另一個好處，在於未來更容易擴增資料來源與類型。由於 Altmetrics 通常以當下網路上流行的參考數據，例如被收藏、被分享等等資訊做為資料類型。雖著網路科技發展，資料來源與類型肯定會越來越多元化。良好的系統架構，將能夠協助研究者與開發者，在最低成本下，擴增新的資料來源與類型，而不必大幅修改系統程式碼，以免埋入不穩定因子造成系統錯誤。

資料分析結果在第三章第四節進行過前導研究，新聞資料、期刊雜誌與維基百科等資料來源分析的數據與人工查詢結果相差不大，雖然課程大綱與學術論文受限於資料來源的特性，有較高的誤差值，但這是目前各類 Altmetrics 與 Webometrics 系統常見的狀況，整體而言還有一定程度參考價值，可供研究分析採用。以系統自動進行的分析必定與人工有所差異，無法達到 100% 相同或 100% 精準，其真正用途在於適度取代大量的人工分析工作，可以用較快的速度

或者較低的成本取得可供參考的資訊。但計算學者貢獻值這件事情無論是系統還是人工，本來就就不精準之處，也不應該作為絕對的判斷或比較基準。更尤其 Altmetrics 作為一個「替代性」計量學，其所提供的數據價值在於傳統分析所無法提供的非學術管道交流活動與網路資源的貢獻程度。應該作為傳統計量學的補充資訊，相輔相成，而非作為評斷學者貢獻的單一標準。

也因為 Altmetrics 的出現，提供了傳統計量學之外的全新視野，讓我們能夠將正式文獻以外的學術行為列入分析數據中，使得許多砸下心思在教學、程式、非正式刊物與文章等等的學者，其努力有被眾人看到的一天。若 Altmetrics 能被善用，或者未來的各大資料庫與索引系統都能優先引入 Altmetrics 思想，將書籤、分享、下載等各類數據預先做整合，便能讓民眾在不需刻意查詢 Altmetrics 系統的情況下，知道這些文獻在網路與非正式學術行為中所受到的評價，也讓學者們更有動力在網路上與全世界進行更密切的學術交流。

第二節 建議

本次研究中，所採用的某些資料類型，如引用數量，著作篇數等資料類型，在面對大量網路上的非結構化資料時，判斷較為困難，錯誤機率也高。例如是否為某篇網頁的作者、被引用的格式非傳統 APA 時、或者縮寫方式不同等等。未來在設計相關系統時，除了單純的擷取資料以外，可以考慮引入更多演算法來優化資料的判斷。例如中英文斷詞、權威控制、文字探勘（Text Mining）或資料探勘（Data Mining）等技術，加強資料查詢時的求全與求準率。也可考慮採用機器學習（Machine Learning）或深度學習（Deep Learning）等方式，訓練電腦由上下文結構判斷內容意涵，將能使系統分析結果更有可信度，並且未來有越來越多新的資料來源加入時，也可以在短期內訓練系統熟悉新資料來源的特性，取代人工編寫判斷規則。

系統的資料來源有多數是採用第三方搜尋引擎或 API，如 Google 或 Web of Science 等，在資料的擷取上受到搜尋引擎與平台的大量限制。例如 Google 有短期查詢數量限制，系統在抓取搜尋結果時，常常被封鎖，導致返回的數據結果不夠準確。Mendeley 則是限制 API 搜尋的最大結果為 100 筆，因此更高的搜尋結果都不會顯示。其他各個平台與搜尋引擎，也會遇到結果返回不準確，過多或過少等狀況，也可能太過模糊，包含不正確的資料。雖然在與人工比對下差異並不大，但在未來設計相關系統時，若需要更高的準確度，可考慮直接與資料庫合作取得資料來源，但必須確保資料的即時性，以日或週為單位的更新分析數據是 Altmetrics 的重要特色。或者在平台資料擷取上，必須設計分散式架構，藉由不同的 Proxy 來分散流量，避免異常流量遭到平台阻擋。只有準確的資料量才能確保分析的可信度，本研究所設計之系統提供了搜尋流程的運作模型，未來還可以在這個基礎上加入分散式節點來確保系統可用性，藉由不停的更新迭代來完善系統功能。

最後，本研究建立在 Cave 等學者的資料類型上進行 Altermetrics 分析，以進行與傳統文獻計量相異的網路計量。隨著網路科技發展，可能出現的媒體、媒介還會越來越多，資料來源與類型也將會持續發展增加。屆時，現有的資料類型與來源可能也必須擴增。而針對學者的學術交流等活動行為，本研究目前也還集中在少數較傳統的國內引文平台，但實際上還有許多新興平台存在有價值的數據資料，例如 SlideShare (<http://www.slideshare.net>) 就包含了不少學者出席研討會所製作的簡報，類似的平台還有 SpeakDeck (<https://speakerdeck.com>) 與 Slides (<https://slides.com>) 等，人們在網路上的活動足跡不停的增加，會有越來越多過去不曾受到注目的網站成為有價值的平台。未來的研究還可以試圖挖掘更多有價值的資料來源，並根據當下的科技發展與時空背景，設計出更多合適的資料類型與計算方式，讓學術交流行為在網路空間更加多元化。

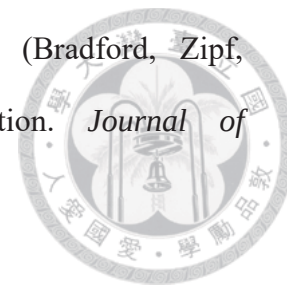


參考文獻



- Agile software development. (2015). Retrieved from
https://en.wikipedia.org/wiki/Agile_software_development
- Altmetric.com. (2015). Retrieved from <http://altmetric.com>
- Altmetrics. (2015). Retrieved from <http://www.altmetric.com/badges.php>
- Beck, M. T. (1978). Editorial Statment. *Scientometrics*, 1(1), 1-2.
- Björneborn, L., & Ingwersen, P. (2004). Toward a Basic Framework for Webometrics. *Journal of the American Society for Information Science and Technology*, 55(14), 1216-1227.
- Brookes, B. C. (1990). Biblio-, Sciento-, Infor-metrics??? What Are We Talking About? *Informetrics 89/90: selection of papers submitted for the second International Conference on Bibliometrics, Scientometrics, and Informetrics, London, Ontario, Canada, 5-7 July 1989* (pp. 35). Amsterdam: Elsevier.
- Buschman, M., & Michalek, A. (2013). Are alternative metrics still alternative? *Bulletin of the Association for Information Science & Technology*, 39(4), 35-39.
- Cave, R. (2012). *Overview of the altmetrics landscape*. Paper presented at the Proceedings of the Charleston Library Conference, West Lafayette.
- Cole, F. J., & Eales, N. B. (1917). The History of Comparative Anatomy. *Science Progress*, 1, 578-596.
- Dobrov, G. M., & Karennoi, A. A. (1969). The Informational Basis of Scientometrics. *On Theoretical Problems of Informatics*, 435, 165-191.

Fairthorne, R. A. (1969). Empirical Hyperbolic Distribution (Bradford, Zipf, Mandelbrot) for Bibliometric Description and Prediction. *Journal of Documentation*, 25(4), 319-343.



Fourth-generation programming language. (2015). Retrieved from https://en.wikipedia.org/wiki/Fourth-generation_programming_language

Hoffer, J. A., Georg, J. F., & Valacich, J. S. (1996). *Modern systems analysis and design*: Reading, Mass.: Benjamin/Cummings Pub.

Howard, J. (2013). Rise of 'Altmetrics' Revives Questions About How to Measure Impact of Research. Retrieved from <http://chronicle.com/article/Rise-of-Altmetrics-Revives/139557/>

Hulme, E. W. (1923). *Statistical Bibliography in Relation to the Growth of Modern Civilization*. London: Grafton.

Impactstory. (2015). Retrieved from <https://impactstory.org>

Ingwersen, P. (1998). The Calculation of Web Impact Factors. *Journal of Documentation*, 54(2), 236-243.

Konkiel, S. (2013). Altmetrics: A 21st-century solution to determining research quality. *Online Searcher*, 37(4), 11-15.

Lagace, N. (2013). NISO awarded Sloan Foundation grant to develop standards or recommended practices for altmetrics. *Information Standards Quarterly*, 25(2), 40.

Lin, J., & Fenner, M. (2013). Altmetrics in evolution: Defining & redefining the ontology of article-level metrics. *Information Standards Quarterly*, 25(2), 20-26.

Manifesto for Agile Software Development. (2001). Retrieved from <http://agilemanifesto.org/>

Mellor, S. J. (2005). Adapting Agile Approaches to Your Project Needs. *IEEE SOFTWARE*, May/June, 17-20.

Morales, M. (1985). Informetrics and Its Importance. *International Forum of Information Documentation*, 10(2), 15-21.

Moravcsik, M. J. (1985). Address at the Presentation of First Derek de Solla Price Award to Eugene Garfield on December 20. *Scientometrics*, 7, 143-144.

Nacke, O. (1979). Informetrie: Ein Neuer Name für Eine Neue Disziplin. *Nachrichten für Dokumentation*, 30, 219-226.

Nicholas, D., & Ritchie, M. (1978). *Literature and Bibliometrics*. London: Clive Bingley.

PaperCritic. (2015). Retrieved from <http://www.papercritic.com/>

Piwowar, H. (2013). Altmetrics: Value all research products. *Nature*, 493(7431), 159.

Piwowar, H. (2013). Introduction altmetrics: What, why and where? *Bulletin of the American Society for Information Science and Technology*, 39(4), 8-9.

PLoS. (2014). Overview: Article-level metrics measure the dissemination and reach of published research articles. Retrieved from <http://article-level-metrics.plos.org/alm-info/>

PLoS-ALM-Reports. (2015). Retrieved from <http://almreports.plos.org/>

PLUMX. (2015). Retrieved from <https://plu.mx/>

Priem, J. (2010). I like the term #articlelevelmetrics, but it fails to imply *diversity* of measures. Lately, I'm liking #altmetrics. Retrieved from <https://twitter.com/jasonpriem/status/25844968813>

Pritchard, A. (1969). Statistical Bibliography of Bibliometrics. *Journal of*



Documentation, 25(4), 348-349.

Publish-or-Perish. (2015). Retrieved from <http://www.papercritic.com/>

Rajan, T. N. (1985). Informetrics: The Concept and Ramifications. *FID/IM Newsletter*, 1(1), 1-2.

Rapid application development. (2015). Retrieved from
https://en.wikipedia.org/wiki/Rapid_application_development

Royce, W. W. (1970). Managing the Development of Large Software Systems. *IEEE WESTCON*, August, 1-9.

Sengupta, I. N. (1985). The Growth of Biophysical Literature. *Scientometrics*, 8, 365-375.

Software engineering. (2015). Retrieved from
https://en.wikipedia.org/wiki/Software_engineering

Stevens, R. E. (1983). Characteristics of Subject Literature. *American College and Research Library Monograph Series*, 7, 10-12.

Thelwall, M. (2008). Bibliometrics to webometrics. *Journal of Information Science*, 34(4), 605-621.

Thelwall, M., Vaughan, L., & Björneborn, L. (2005). Webometrics. *Annual Review of Information Science and Technology*, 39(1), 81-135.

TotalImpact. (2015). Retrieved from <https://github.com/total-impact/impact>

Waterfall model. (2015). Retrieved from
https://en.wikipedia.org/wiki/Waterfall_model

王鈞湄 (2013)。適用於人文社會學者之非傳統學術指標研究 (碩士論文)。國立臺灣大學，圖書資訊學研究所，臺北市。



王睿，胡文靜與郭瑋（2014）。高 Altmetrics 指標科技論文學術影響力研究。《圖書情報工作》，58（21），頁 92-98。

王睿，胡文靜與郭瑋（2014）。常用 Altmetrics 工具比較。《現代圖書情報技術》，235（12），頁 18-26。

快速應用開發（2015）。取自 <http://wiki.mbalib.com/zh-tw/%E5%BF%AB%E9%80%9F%E5%BA%94%E7%94%A8%E5%BC%80%E5%8F%91>

沈小玲、嚴尉中（2013）。網路科技論文學術影響力評價指標的選擇。《圖書情報工作》，57（3），頁 69-77。

邱均平、余厚強（2013）。替代計量學的提出過程與研究進展。《圖書情報工作》，57（19），頁 5-12。

敏捷軟體開發（2015）。取自 <https://zh.wikipedia.org/wiki/%E6%95%8F%E6%8D%B7%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91>

許培揚（2014）。替代計量分析（Altmetrics）的文獻類型與數據來源。取自 <http://blog.sciencenet.cn/blog-280034-825495.html>

軟體工程（2015）。取自 <https://zh.wikipedia.org/wiki/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B>

陳光華（2006）。文獻計量學的產生。取自 <http://www.lis.ntu.edu.tw/~khchen/course/bm/bm2006/bm200601.pdf>

陳亞寧（2013）。另一種指標 Altmetrics & ALMs 簡介。『創新華文知識服務：中文學術電子資源的發展與應用』研習會。台北市：科技政策研究與資訊中心 CONCERT；華藝數位股份有限公司。取自 <http://concert.stpi.narl.org.tw/Images/upload/2013/10/altmetric.pdf>

對於 Altmetrics 終於有了初淺的認識（2015）。取自 <http://libraryview.me/2015/04/07/9805/>

增加產能：快速應用程式開發。（2015）。取自 [https://msdn.microsoft.com/zh-tw/library/ms233948\(v=vs.90\).aspx](https://msdn.microsoft.com/zh-tw/library/ms233948(v=vs.90).aspx)

蔡明月（1997）。書目計量學、科學計量學與資訊計量學。《教育資料與圖書館學》，24（3），頁 268-284。

蔡明月（2002）。網路計量學 Webometrics。《圖書與資訊學刊》，42，頁 1-14。

蔡明月 (2003)。資訊計量學與文獻特性。台北市：華泰。

蔡明月 (2004)。論資訊計量學 An Introduction to Informetrics。圖書館學與資訊科學，30 (2)，頁 83-91。

蔡明月、曾苓莉 (2014)。網路計量學新指標 Altmetrics。教育資料與圖書館學，51，頁 91-120。

蘇義雄 (2013) 由瀑布式開發轉變至敏捷式開發對專案管理的影響 - 以 TDC 個案研究為例 (碩士論文)。國立台灣科技大學，管理研究所 EMBA 碩士在職專班，台北市。

顧立平 (2013)。開放數據計量研究綜述：計算網絡用戶行為和科學社群影響力的 Altmetrics 計量。現代圖書情報技術，234 (6)，頁 1-8。

