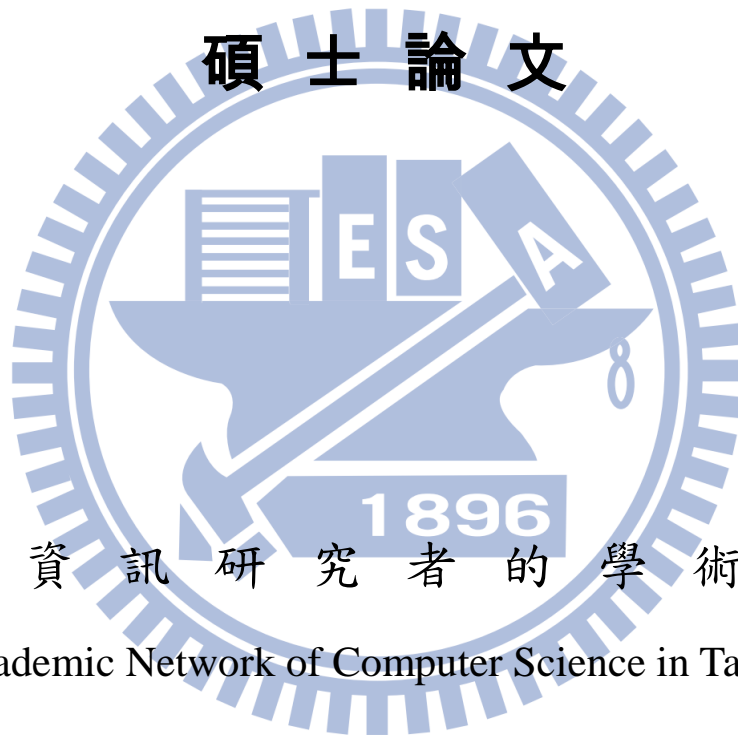


國立交通大學

多媒體工程研究所

碩士論文



台灣資訊研究者的學術網路

Academic Network of Computer Science in Taiwan

研究生：李羿賢

指導教授：梁婷 教授

中華民國 一 百 年 九 月

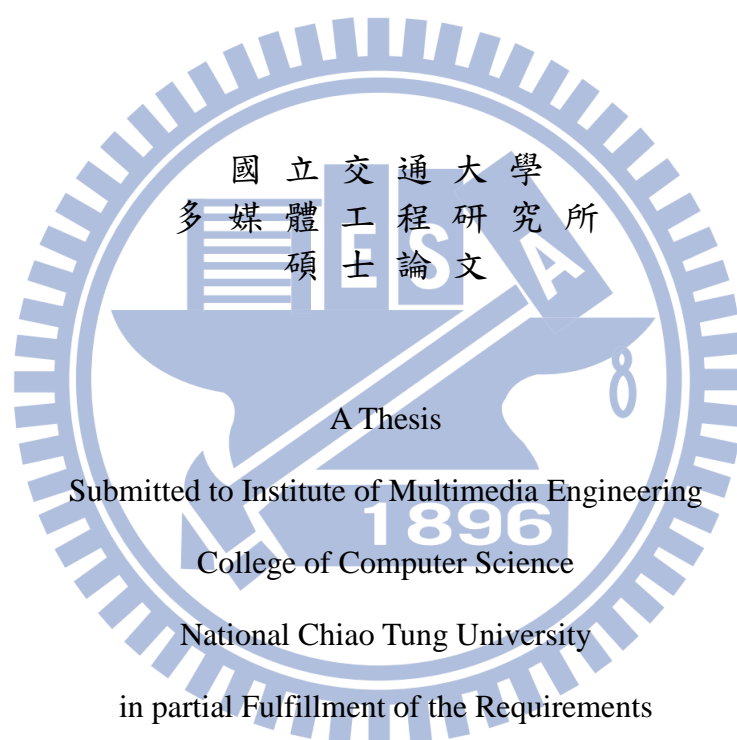
台灣資訊研究者的學術網路
Academic Network of Computer Science in Taiwan

研究生：李羿賢

Student：Yi-Hsien Li

指導教授：梁婷博士

Advisor：Dr. Tyne Liang



in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

September 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年九月

台灣資訊研究者的學術網路

研究生：李羿賢

指導教授：梁 婷 博士

國立交通大學

多媒體工程研究所

摘要

隨著 Web2.0 網路應用成功的發展，社交網路不再只是一種概念，而是能夠實際存在網路上的一種社會結構。在現實世界中，存在許多不同種類的社群網路，但是現今網路上的社群網路平台大部分都建立於朋友關係之上，例如：Facebook、Twitter、Google+。目前沒有一個學術網路可以提供研究者探索及了解其他研究者的學術網路。因此，這篇論文的目的是建立一個台灣資訊領域的學術網路，提供使用者能更有效地了解研究者和他們之間的關聯。在我們的研究方法中，首先收集 266 學者發表的 16,981 篇著作篇名，利用維基百科電腦科學內文建立 LSA 分類器，將著作篇名透過 LSA 分類器對應至所屬領域，從中取得學者在各研究領域的成果。最後透過研究者之間的研究領域、合作關係、機構單位建立台灣資訊學術網路。

Academic Network of Computer Science in Taiwan

Student : Yi-Hsien Li

Advisor : Dr. Tyne Liang

Institute of Multimedia Engineering
National Chiao Tung University

ABSTRACT

As web 2.0 applications are more and more successful, social network is no longer a concept, but a real social structure established in the internet. There are many kinds of social networks exist in the real world. However, most of the social network applications, such as Facebook, Twitter, Google+, are constructed based on friendship. There is no academic network for researchers to explore the research society and get acquainted with other researchers. In this thesis, we focus on constructing an academic network of computer science researchers in Taiwan. First, we collect 16,981 paper titles of 266 researchers. Then we use the context of computer science categories of Wikipedia as training corpus to establish an LSA classifier. This LSA classifier can help us classify paper titles into 68 predefined categories. Then we identify the relation of research area, research cooperation and research organization between researchers. In the end, we combine the three relations as the connection strength of the academic network.

誌謝

首先要感謝的是我的指導教授 梁婷老師，在這兩年的碩士生活中，教導我很多做研究、處理事情的方法，對於我的碩士論文給許多建議及想法，讓我可以順利地完成這一篇碩士論文。梁婷老師把我們當作自己的兒女一樣關心，時常問我們身體健康、生活起居...等，讓實驗室充滿到一股親切感。接著要感謝我的口試委員，鄭卜壬教授、楊武教授，感謝給予我許多寶貴建議，讓本論文可以更加完善。

其次我要感謝冠熙學長，他花了很多時間幫我修改論文紙本，在做碩論的其間，冠熙學長幫忙提出很多方法，讓論文更加完善。只要有問題也都去請教他，但他都不厭其煩地教導著我，是一位非常好的學長。還有陪伴研究所生活的鴻達、荃權，幸虧有你們生活才不至於苦悶，一起修課、打球、練嘴上功夫，當然還有課業上的交流、協助，真的很感謝你們一路互相幫忙、分享心情，讓我有動力可以完成這一篇碩士論文。

最後要感謝我的家人，無怨無悔地協助我完成學業，當我疲倦時，家總是一個可以充電的庇護站，讓我一直有動力去完成這碩士論文。謝謝一路研究所生活相伴的你們。讓我面對未來也更有勇氣，努力往自己的目標前進，不會辜負你們的期望。

羿賢
新竹

2011/09/19

目錄

摘要	i
ABSTRACT.....	ii
誌謝	iii
目錄	iv
表目錄	v
圖目錄	vi
第一章 緒論	1
1.1 研究動機與目的.....	1
1.2 問題定義.....	1
1.3 方法簡介.....	1
1.4 論文架構.....	2
第二章 相關研究	3
2.1 主題-關鍵詞.....	3
2.2 主題模型.....	4
第三章 研究方法	6
3.1 語料處理.....	7
3.1.1 單字詞.....	9
3.1.2 名詞片語.....	11
3.2 學術研究關係處理.....	12
3.3 學術合作.....	19
3.3.1 論文合作.....	19
3.3.2 教育合作.....	20
3.4 機構關係.....	22
3.5 關係結果處理.....	22
第四章 實驗與實驗分析	24
4.1 分類實驗.....	24
4.2 統計結果.....	24
4.3 分群實驗.....	25
第五章 結論	37
參考文獻	38
附錄	40

表目錄

表 1:相關研究比較	5
表 2:收集語料統計	9
表 3:過濾器處理結果	9
表 4:詞幹處理結果	10
表 5:單字詞前置處理結果	10
表 6:著作篇名的詞彙個數	10
表 7:名詞片語統計	11
表 8:名詞片語數量統計	11
表 9:梁婷教授研究領域	14
表 10:學術研究關係計算範例	14
表 11:共同研究領域矩陣	15
表 12:關係矩陣	17
表 13:學術研究關係	18
表 14:研究學者資訊	22
表 15:共同性結果	24
表 16:16,981 篇著作的發展	25
表 17:學者分群	28
表 18:學術研究與學術合作分群結果	31
表 19:維基百科電腦科學分類	40
表 20:STOPWORDS	42
表 21:LSA 主題詞彙	43

圖目錄

圖 1: LDA 模型	4
圖 2:關係處理流程圖	6
圖 3:維基百科資訊科學分類	7
圖 4:維基百科內文	8
圖 5:個人著作篇名	8
圖 6:單字詞前處理流程圖	9
圖 7:研究領域網頁	12
圖 8:學術研究處理流程圖	13
圖 9:論文合作關係圖	20
圖 10:共同指導教授.....	21
圖 11:教育合作關係圖.....	21
圖 12:13 群分析圖	27
圖 13:研究領域分群分析圖	30
圖 14:研究領域與合作關係分群分析圖	31
圖 15:研究領域、合作關係及機構關係分群分析圖	34
圖 16:自然語言處理相關學者	35
圖 17:學者分群展示	35
圖 18:交通大學網路研究所學者	36
圖 19:系統顯示交通大學網路研究主題學者	36

第一章 緒論

1.1 研究動機與目的

網際網路的快速發展下，人們的生活型態逐漸因為網路的發展有所改變。例如：資訊查詢、購物...等。當今人類的社交活動不再局限於直接面對面交談或電話溝通，人們的社交活動漸漸透過各式各樣的網路平台認識在世界各地中與自己有相同嗜好或興趣的人，例如：Facebook、無名小站...等。學術網路是藉由數個研究學者建立的社會結構，其中每位研究學者有各自的研究興趣、參與的計畫案、畢業學校、參訪組織、授課單位、指導學生、同窗好友及同事...等。藉由每位研究學者之間研究共同性建立起研究學者的關係。例如：張俊盛教授與梁婷教授對自然語言處理有專業研究。梁婷教授與曾新穆教授有共同著作 Efficient Mining of Temporal High Utility Itemsets from Data streams。表示在學術網路中梁婷教授與張俊盛教授有共同研究領域關係，曾新穆教授有共同合作關係。透過學術網路可以讓研究者找到與自己具有共同研究領域或畢業於相同機構的相關研究者。

本篇論文中我們專注探討存在於台灣資訊領域學者之間在學術上的關聯所建立的學術網路。透過每位研究學者發表的著作篇名及自訂的研究領域，從中我們可以發現台灣學術界在資訊研究領域中發展情況。例如：各個研究主題中有多少位研究學者及論文著作。各研究機構在資訊領域發展的情況。此外也可透過研究學者共同指導學生及共同發表著作，探討研究學者彼此間合作關係的密切度。透過此學術網路使用者可以查詢台灣資訊研究者在各資訊領域的發展情形。另外我們將著作篇名利用維基百科¹內文所建立的 LSA 分類器分成六十八類，讓使用者可以直接查詢各領域的論文著作及相對應的作者。

1.2 問題定義

學術網路的節點與關係的定義是非常重要的議題，在本論文中我們以台灣資訊領域的研究者為節點，另外關係的建立是以每位研究者的研究領域、學術合作以及機構單位建立關係。透過此學術網路可以讓使用者更快速瞭解台灣資訊領域的發展以及各領域的研究學者。

1.3 方法簡介

本論文中，我們從台灣資訊學術領域的學者中抓取著作篇名，並與維基百科內文做相識度判斷，已得到每篇著作屬於維基百科資訊領域的哪一類別。資訊領域的分類，我們採用維基百科所定義的 13 大類，再將其細分為 68 小類。並透過

¹ <http://zh.wikipedia.org/>

維基百科對上述 68 類的定義及描述來建立分類器。藉由維基百科內文我們半自動化產生各領域的關鍵詞彙。透過每位研究者歷年出版的著作篇名及研究領域判定各研究者的研究方向及比例。在先前的論文中，僅止於在各研究領域中出現的學者姓名當作唯一的關係建立。我們希望每位學者之間不僅是只有在研究領域上有所關聯，可以更進一步檢視研究領域上著重比例。另外探討研究學者之間學術合作關係以及機構單位。利用共同著作以及共同指導學生建立學術合作關係。機構單位依據研究學者服務機構以及畢業學校建立關係。最終統計每位學者在各領域的研究範圍。針對不同資訊領域列出所有相關的台灣研究學者，並且建立起每位學者之間的關係強度。

1.4 論文架構

本論文章節架構如下：第二章為學術網路相關研究；第三章介紹系統架構與研究方法。首先介紹語料的來源，利用這些語料收集各領域關鍵詞彙且分類，利用標記論文題目，來統計各研究者的研究方向，最後利用收集領域共同性及詞彙，來建立台灣資訊領域學術網路。在第四章中我們對單字詞及名詞片語在 LSA 分類器的結果進行比較，並且探討研究學者分群結果。第五章敘述結論與未來的發展方向。



第二章 相關研究

文本分類研究者主要探討分類模型及文本表示。分類模型相關研究以實用機器學習領域相關分類模型。文本表示主要探討文本與關鍵詞的權重關係與表示方式。我們分別對主題-關鍵詞的權重計算以及文本表示方式進行探討。

學術網路是社群網路的一種型態，主要探討學術領域中的學者彼此之間的關聯性。Shou-de Lin and Hans Chalupsky[2003]利用學者發表著作中的引述論文，從中利用論文彼此之間的引述關係探討學者彼此間的合作關係以及主題。Rosen et al, [2004] 提出作者-主題模型，透過詞彙與作者之間的關聯性從中找出學者的研究主題。每位學者之間可以依據詞彙與主題之間建立關係。Tang et al. [2008] 利用共同作者及研討會議提出 Author-Conference-Topic 模型。每位學者彼此之間可以透過主題以及研討會關係建立起一個小型網路關係。

2.1 主題-關鍵詞

主題-關鍵詞是指各個主題中具有代表性的詞彙。以資訊擷取(information retrieval)為例，關鍵詞有資訊(information)、擷取(retrieval)、索引(indexing)、查詢(query)...等。相對於解碼器(decoder)、像素(pixel) 壓縮(compression)...等更具代表性。主題關鍵詞的表示方式分為向量空間表示及機率模型。

向量空間表示中 Deerwester et al. [1990]利用以向量空間模型為基礎的 Latent Semantic Analysis(LSA)來表示主題與關鍵詞，其中主題與關鍵詞利用每個關鍵詞在各主題出現的次數表示，接著利用 Singular Value Decomposition(SVD) 降低向量維度，去除主題中不相關的詞彙。最後將文檔與關鍵詞權重利用向量空間表示主題與關鍵詞的關係。Hofmann[1999]提出 PLSI(probabilistic Latent Semantic Indexing)，此方法以 Deerwester et al. [1990]提出的 LSA 為基礎。有別於 LSA 使用關鍵詞出現在各主題的頻率次數，PLSI 使用 TF-IDF (Term Frequency-Inverse Document Frequency) 計算關鍵詞對於主題的重要性，最後利用 random mapping method 決定使用分群或投影以得到主題關鍵字關係。

在機率模型中，Blei et al. [2003]利用 Jensens 不等式建立機率模型，模型中有參數值和下限值。再藉由 Variational EM 演算法得到最佳主題與關鍵詞近似機率。在步驟 E(Expectation)計算出最佳的參數值。在步驟 M(Maximization)中算出最大的下限值。藉由上述方法不斷計算得到最佳的 主題與關鍵詞機率分佈。Griths and Steyvers[2004]利用 Gibbs sampling 算出文檔中的主題多項式分佈。其中利用 Markov chain Monte Carlo(MCMC) 近似反覆計算主題關鍵詞機率，從收斂的 Markov chain 中取得主題與關鍵字的近似機率分佈樣本。利用公式(1)算出主題與關鍵詞分佈。 C_{mj}^{WT} 表示詞彙 m 分配給主題 j 的頻率數； β 表示每個主題與詞彙的機率分佈。

$$\phi_{mj} = \frac{C_{mj}^{WT} + \beta}{\sum_m C_{mj}^{WT} + V\beta} \quad (1)$$

2.2 主題模型

主題模型是指在一個具有詳細且具有意義的文檔中找出文檔所表示的主題和可能隱含的相關主題。主題模型分為監督式以及非監督式兩種，監督式方法需要人工標記，藉由事先主題關鍵詞以產生訓練語料，由於需要大量人工處理，在不同領域的詞彙，也須要重新標記，所以處理費時，但其正確率較高。非監督式方式不需要大量人工標記，易於應用在不同領域，但正確率較低。

Yiming Yang[1999]利用人工標記每篇文檔的主題，再計算每個詞彙出現在各文檔的次數，得到個文字所屬的主題可能性，最後依據各詞彙的分佈判斷測試語料所對應的主題。在實驗中，使用 7,789 篇路透社新聞為訓練語料，對 3,309 篇測試語料判斷主題，得到正確率為 93%。非監督式主題辨識方法中，利用向量空間模型(Vector Space Model)來探討的主題模型是很常用的表示方法。將主題與詞彙表示為向量，再藉由向量空間的轉換以找出主題與詞彙的分群結果。Lagus et al.[1999]將詞彙以向量空間表示，利用 random mapping method [Kaski, 1998]不斷將高維度空間降維至低維度空間，最後藉由低維度空間對應到主題。Berry et al.[1998]利用向量空間投影方式，首先建立主題與詞彙向量空間，使用 LSA 降低向量空間維度，在將詞彙投影至各個主題空間，找出投影後所對應的主題。LSA 分類再[Schutze et al, 1995] [Chen L et al, 2003]中應用得到深入的研究。

非監督式主題模型除了以向量空間模型為基礎外，另外還有被廣泛應用的生成機率模型(generative probabilistic model)。其中以 LDA(Latent Dirichlet Allocation)為代表的系列模型。LDA 是一個藉由潛在主題生成文檔與詞彙的過程。的如下圖 1 所示。 α 表示文檔集中隱含主題間的相對強弱， β 表示所有隱含主題的機率分佈， θ 表示隱含主題的比重， z 表示目的檔分配在每個詞彙上的隱含主題， w 表示目的檔的詞彙向量表示法。 α 及 β 利用 Dirichlet distribution 計算分佈情形。

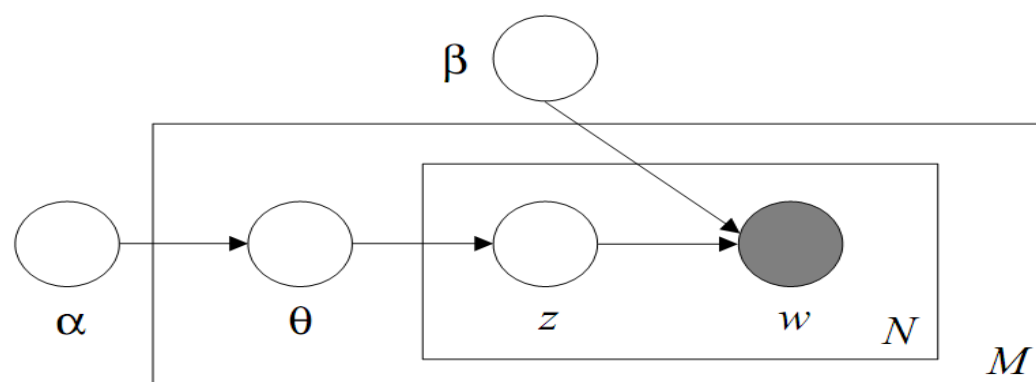


圖 1: LDA 模型

Blei et al. [2003]使用 Dirichlet distribution 計算主題-關鍵詞以及文檔-主題多項式分佈，再從主題-關鍵詞以及文檔-主題多項式分佈中找出關鍵詞與主題。再

藉由 Variational EM 演算法的到最佳近似機率。由於使用 Variational EM 會發生局部極值的問題，所以在後來的研究中都是使用 Gibbs sampling 已取得最佳近似解。其中利用觀察詞 w 在主題 z 上的抽樣機率 $p(w|z)$ 當作近似解。藉由 Gibbs sampling 不斷抽樣取出，最終取出主題-關鍵詞及文檔-主題分佈。Rosen et al. [2004] 提出作者-主題模型，首先利用 LDA 計算詞彙與主題對應關係，接著利用作者-詞彙的機率分佈考慮每個詞彙所對應的作者。最後利用上述兩個模型分別對文檔中各個詞彙找出對應的主題與作者建立作者-主題模型。Tang et al. [2008] 利用共同作者及研討會議提出 Author-Conference-Topic 模型。首先利用共同作者對主題機率分佈在透過主題與研討會分佈及主題與關鍵詞分佈找出作者與研討會的機率對應。在利用作者與研討會共同找出主題機率。最後利用上述的模型建立 Author-Conference-Topic 模型。

表 1: 相關研究比較

	[Lagus et al, 1999]	[Shiau Yang et al, 2010]	[Rosen et al, 2004]	[Jie Tang et al, 2008]	本研究
詞彙-文檔	TF-IDF	機率	機率	機率	TF-IDF
表示方式	向量空間	機率模型	機率模型	機率模型	向量空間
Stopwords	有	有	有	有	有
Stemming	有	無	無	無	有
索引詞	單字詞	單字詞	單字詞	單字詞	單字詞/ 名詞片語
方法	將高維度的向量空間利用 SVD 降至低維度空間。利用低維度空間對應至主題。	將文檔分割為句子，利用 ProbGemSum 和 SentGenSum 計算句子權重。從排序的句子中取出平均句子個數判斷文檔主題。	作者與詞彙分佈，主題與詞彙分佈。作者-主題模型。	共同作者與主題分佈，作者-研討會對應於主題分佈。作者與主題配對。	利用維基百科內文找出主題關鍵詞彙。在相識度計算判斷主題。利用著作篇名及研究領域探討作者研究主題。
實驗語料	科學文章 3,000 篇	DUC2002 590 篇文摘	1700 論文 CiteSeer 160,000 篇 摘要	研討會論文 10,716 篇	著作篇名 16,981 篇

第三章 研究方法

我們以台灣資訊領域學者建立學術網路，其中研究關係辨識依據研究者著作篇名、個人研究領域的共同性來建立。我們利用維基百科對資訊領域²的分類以及維基百科內文的詞彙來判定各篇論文題目所對應的分類，以及個人研究領域的分類。首先利用過濾器將語料中不重要的詞彙去除，再透過詞幹處理取出詞彙中詞幹相同的詞彙。最後利用 LSA 分類各著作篇名及研究領域類別。合作關係辨識利用學者之間同著作以及共同指導學生關係建立。機構關係透過學者目前服務單位以及最高學歷畢業學校組成。最後依據彼此之間共同研究領域、合作關係及機構單位建立關係。關係處理流程如圖 2 所示。

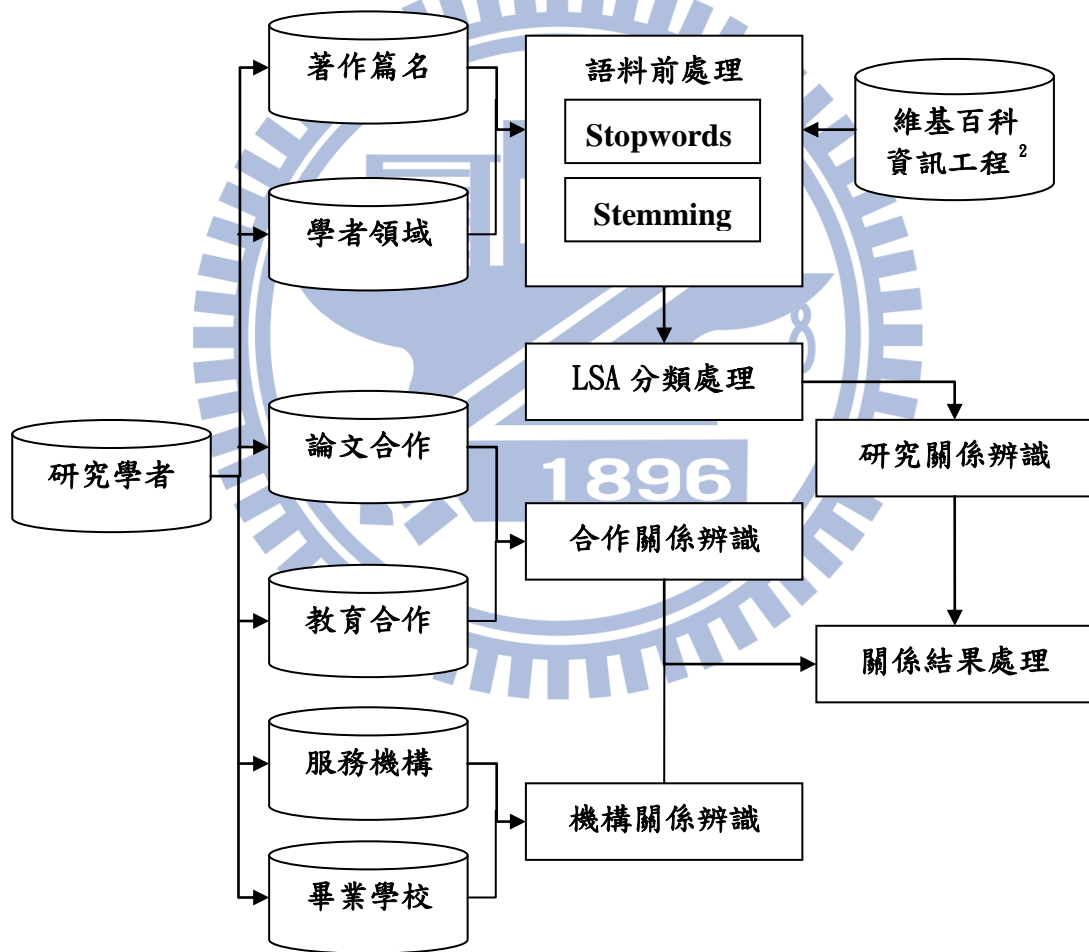


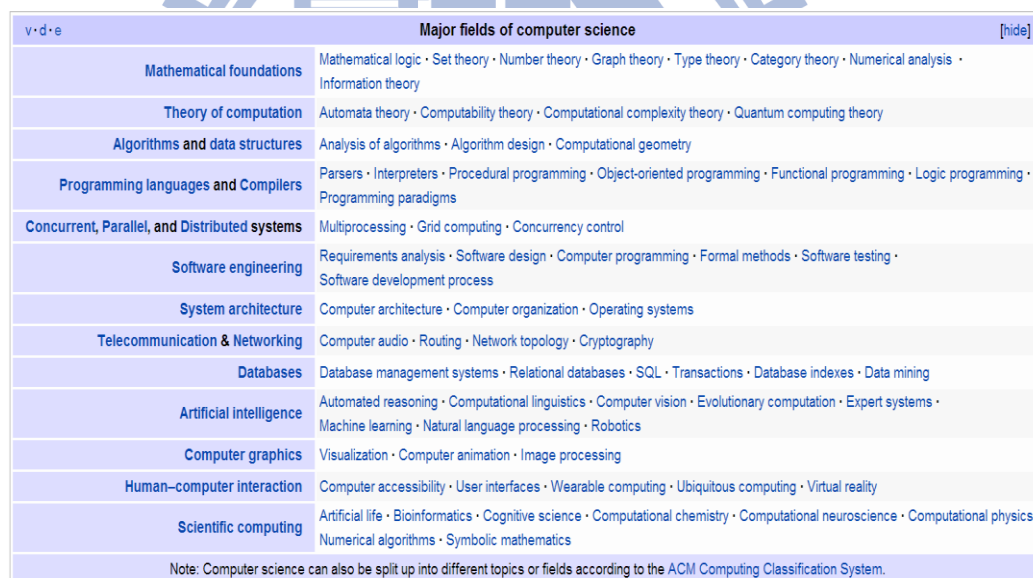
圖 2:關係處理流程圖

² http://en.wikipedia.org/wiki/Computer_science

3.1 語料處理

語料分為維基百科內文及著作篇名。我們利用維基百科內文建立詞彙對各研究領域的重要性並透過向量空間表示，在將著作篇名中的詞彙對應至向量空間並計算著作篇名與各領域的相識度。維基百科內文及著作篇名中充斥著許多不重要或具相同意義的詞彙，因此我們對語料進行過濾及詞幹處理。

維基百科是近年來快速發展的超文字系統，也可以解讀為屬於人類知識的網路系統。維基百科透過使用者編輯各個不同領域的專業知識，漸漸地成為具有豐富內文的語料。藉由此具有詳盡描述與維護的語料中，我們利用維基百科在電腦科學領域中的描述及介紹作為分類的依據。維基百科對資訊工程領域分成十三大類，再將十三大類分為六十八小類。如圖 3 所示。我們擷取維基百科²中對這六十八類的描述，如圖 4 所示。維基百科中對各類別的描述包含定義、問題、發展、延伸議題...等。



Major fields of computer science [hide]	
Mathematical foundations	Mathematical logic · Set theory · Number theory · Graph theory · Type theory · Category theory · Numerical analysis · Information theory
Theory of computation	Automata theory · Computability theory · Computational complexity theory · Quantum computing theory
Algorithms and data structures	Analysis of algorithms · Algorithm design · Computational geometry
Programming languages and Compilers	Parsers · Interpreters · Procedural programming · Object-oriented programming · Functional programming · Logic programming · Programming paradigms
Concurrent, Parallel, and Distributed systems	Multiprocessing · Grid computing · Concurrency control
Software engineering	Requirements analysis · Software design · Computer programming · Formal methods · Software testing · Software development process
System architecture	Computer architecture · Computer organization · Operating systems
Telecommunication & Networking	Computer audio · Routing · Network topology · Cryptography
Databases	Database management systems · Relational databases · SQL · Transactions · Database indexes · Data mining
Artificial intelligence	Automated reasoning · Computational linguistics · Computer vision · Evolutionary computation · Expert systems · Machine learning · Natural language processing · Robotics
Computer graphics	Visualization · Computer animation · Image processing
Human-computer interaction	Computer accessibility · User interfaces · Wearable computing · Ubiquitous computing · Virtual reality
Scientific computing	Artificial life · Bioinformatics · Cognitive science · Computational chemistry · Computational neuroscience · Computational physics · Numerical algorithms · Symbolic mathematics
Note: Computer science can also be split up into different topics or fields according to the ACM Computing Classification System.	

圖 3:維基百科資訊科學分類

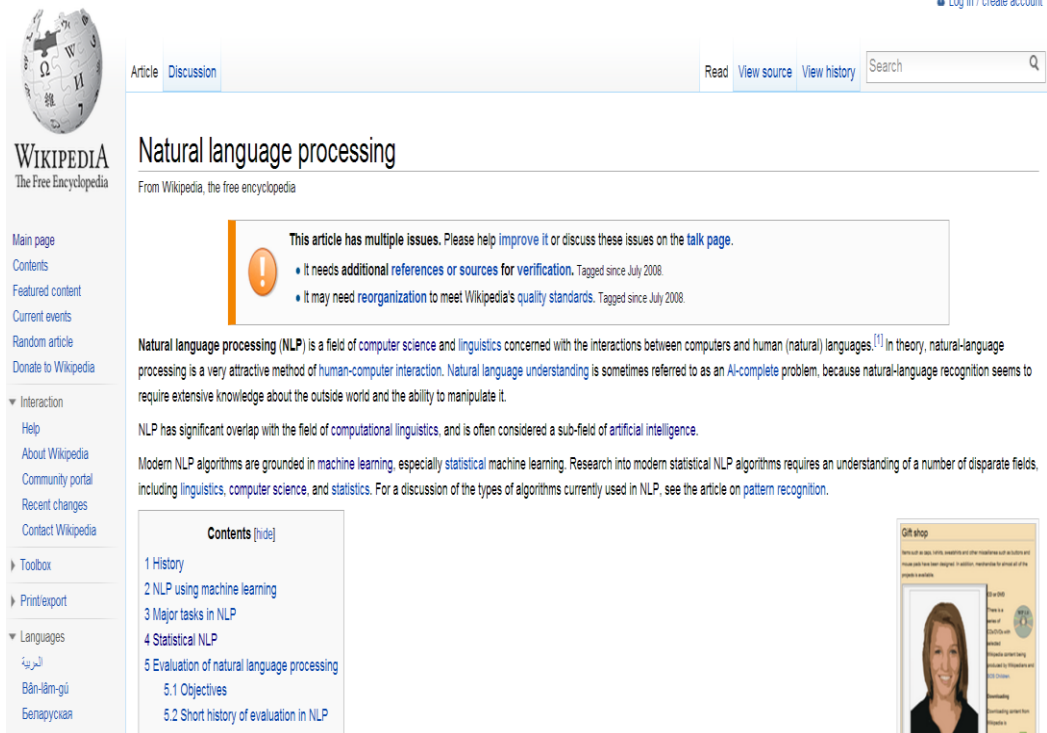


圖 4:維基百科內文

我們收集台灣大學、清華大學、交通大學、成功大學、中央大學、中央研究院，每位資訊領域研究者(共 266 位)，我們從各研究者個人網頁中擷取歷年著作篇名(共 16,981 篇)，如圖 5 所示。

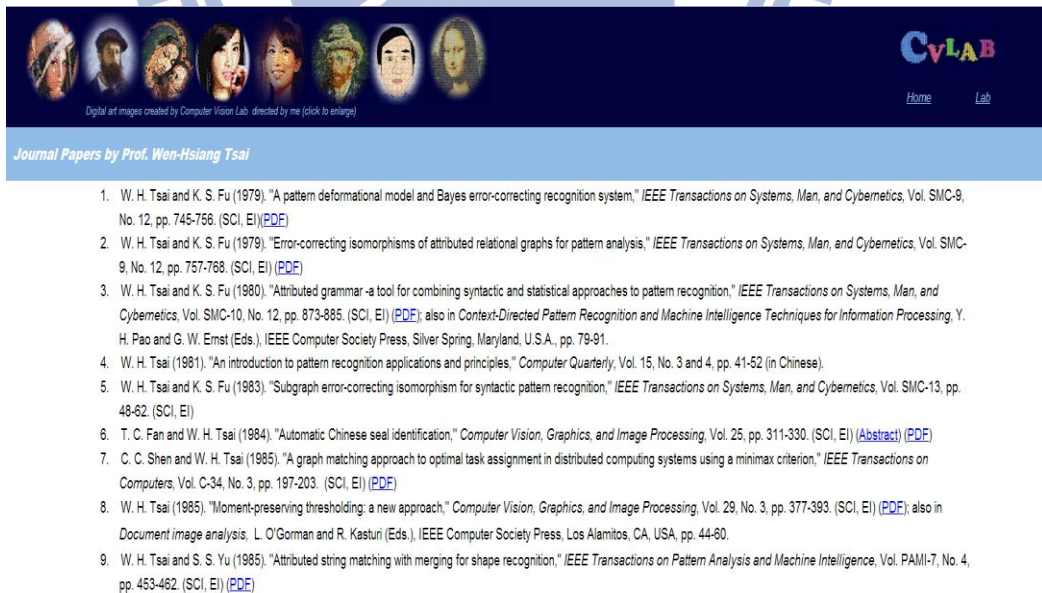


圖 5:個人著作篇名

著作篇名及研究領域由數個名詞片語或單字詞組成，為了有效找出著作篇名與研究領域所屬類別。我們將語料分別利用單字詞及名詞片語對領域類別進行實驗。

3.1.1 單字詞

我們收集六十八篇維基百科內文，其中包含 102,652 個詞彙，扣除重複詞彙共有 10,781 個詞彙，以及 16,981 著作篇名，其中包含 159,088 個詞彙，扣除掉重複詞彙共有 13,078 個詞彙。收集語料數據如表 2 所示。由於語料中包含許多無意義或是具有相同詞義的詞彙。所以，我們對收集的語料，進行前置處理，其中包含過濾語料中不重要的詞彙，以及針對詞彙的詞幹擷取。其流程如圖 6 所示。

表 2:收集語料統計

類別	詞彙個數	詞彙不重複個數
維基百科內文(68 篇)	102,652	10,781
著作篇名(16,981 篇)	159,088	13,078

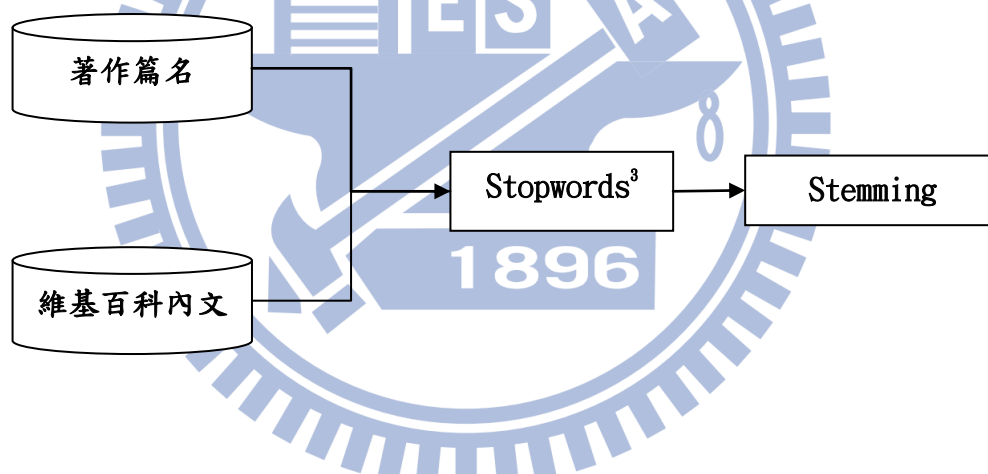


圖 6:單字詞前處理流程圖

過濾器目的在於去除語料中不重要的詞彙，例如:連接詞(and)、輕動詞(be)...等。我們使用 Stopwords³資料庫(共 571 個詞彙)過濾掉通用詞彙。最後六十八篇維基百科內文，其中包含 101,917 個詞彙，扣除重複詞彙共有 10,739 個詞彙，以及 16,981 著作篇名，其中包含 118,926 個詞彙，扣除掉重複詞彙共有 13,368 個詞彙。過濾結果如表 3 所示。

表 3:過濾器處理結果

類別	詞彙個數	詞彙不重複個數
維基百科內文(68 篇)	101,917	10,739

³ <ftp://ftp.s.cornell.edu/pub/smart/english.stop>

著作篇名(16,981 篇)	118,926	13,368
----------------	---------	--------

詞幹處理目的在於針對不同詞彙但其詞幹所表示是相同的詞彙。例如:stemming、stemmer、stemmed 三個詞彙的詞幹都是 stem。經過詞幹處理後都會被表示為詞幹。我們使用 Porter stemming algorithm[Porter,1980]處理詞幹問題。最後六十八篇維基百科內文,其中包含 101,917 個詞彙,扣除重複詞彙共有 10,672 個詞彙,以及 16,981 著作篇名,其中包含 118,926 個詞彙,扣除掉重複詞彙共有 10,126 個詞彙。詞幹處理結果如表 4 所示。

表 4:詞幹處理結果

類別	詞彙個數	詞彙不重複個數
維基百科內文(68 篇)	101,917	10,672
著作篇名(16,981 篇)	118,926	10,126

經由前置處理後,我們收集六十八篇維基百科內文,其中將 102,652 個詞彙中取出 101,917 個詞彙,扣除原本 0.7%的詞彙量,另外在不重複詞彙中將 10,781 個詞彙中取出 10,672 個詞彙,扣除原本 1.0%的詞彙量。另外在 16,981 著作篇名,其中將 159,088 個詞彙中取出 118,926 個詞彙,扣除原本 25.2%的詞彙量,另外在不重複詞彙中將 13,078 個詞彙中取出 10,126 個詞彙,扣除原本 22.6%的詞彙量。單字詞前置處理結果如表 5 所示。

表 5:單字詞前置處理結果

類別		處理前	處理後
維基百科	詞彙個數	102,652	101,917
	詞彙不重複個數	10,781	10,672
著作篇名	詞彙個數	159,088	118,926
	詞彙不重複個數	13,078	10,126

最後,經由前置處理後,有 52.49%的著作篇名,詞彙個數介於六至八個字。每篇著作篇名包含的詞彙個數如表 6 所示。

表 6:著作篇名的詞彙個數

詞彙個數	著作篇名篇數(比例)
≤4	444(0.03%)
5	1,375(8.10%)
6	2,555(15.05%)
7	3,265(19.23%)
8	3,092(18.21%)
9	2,509(14.78%)

10	1,677(9.88%)
11	1,051(5.98%)
≥12	1,049(6.18%)

3.1.2 名詞片語

我們將維基百科內文、論文著作篇名，利用 Stanford parser⁴取出句子中的名詞片語。經由 Stanford parser 的詞性標記、樹狀結構，從中取出(NP)。

例:Intent Boundary Detection in Search Query Logs.

<p>詞性標記: Intent/NNP Boundary/NNP Detection/NNP in/IN Search/NNP Query/NNP Logs/NNPS ./.</p> <p>樹狀結構:(ROOT (NP (NP (NNP Intent) (NNP Boundary) (NNP Detection)) (P (IN in) (NP (NNP Search) (NNP Query) (NNPS Logs))) (. .)))</p>

從上述的樹狀結構中，我們取出兩個名詞片語 Intent Boundary Detection 以及 Search Query Logs。

我們將六十八篇維基百科內文以及著作篇名，從中取出所有名詞片語。最後結果如表 7 所示。

表 7:名詞片語統計

類別	名詞片語個數	名詞片語不重複個數
維基百科內文	49,324	25,697
著作篇名	66,777	31,273

每篇著作篇名，經由 Stanford parser 取出名詞片語後。有超過七成的著作篇名含有三或四個名詞片語。每篇著作篇名包含的名詞片語個數如表 8 所示。

表 8:名詞片語數量統計

名詞片語個數	≤2	3	4	5	≥6
著作篇名篇數	927	5,794	6,739	2,817	704
(比例)	(5.46%)	(34.12%)	(39.69%)	(16.59%)	(4.15%)

⁴ <http://nlp.stanford.edu/software/lex-parser.shtml>

3.2 學術研究關係處理

在眾多的研究領域中，每位研究學者皆有各自專精的研究領域。藉由學者在各自研究領域的發展，我們從中透過研究領域的共同性，建立研究學者之間的關係。在學術網路中，每位學者的研究主題是建構彼此關聯的重要橋樑，我們收集每位學者在個人網頁中所提供的著作篇名和研究領域。透過著作篇名可以發現每位學者曾經在各領域上的研究成果。在研究領域的部分，直接說明每位學者自己目前著重的研究領域並更新於自己的首頁，如圖 7 所示。我們將著作篇名與研究領域對應至維基百科電腦科學的六十八類，流程如圖 8 所示。



圖 7:研究領域網頁

每位研究者的著作篇名及研究領域，再經由前置處理後，再經由 LSA 分類處理，將每一篇著作篇名及研究領域對應到維基百科中的電腦科學分類。最後，每一位學者依據著作篇名分類的統計，得到學者在六十八類的研究比例。利用每位研究者在六十八類的比例，計算彼此間在學術研究上的關係。

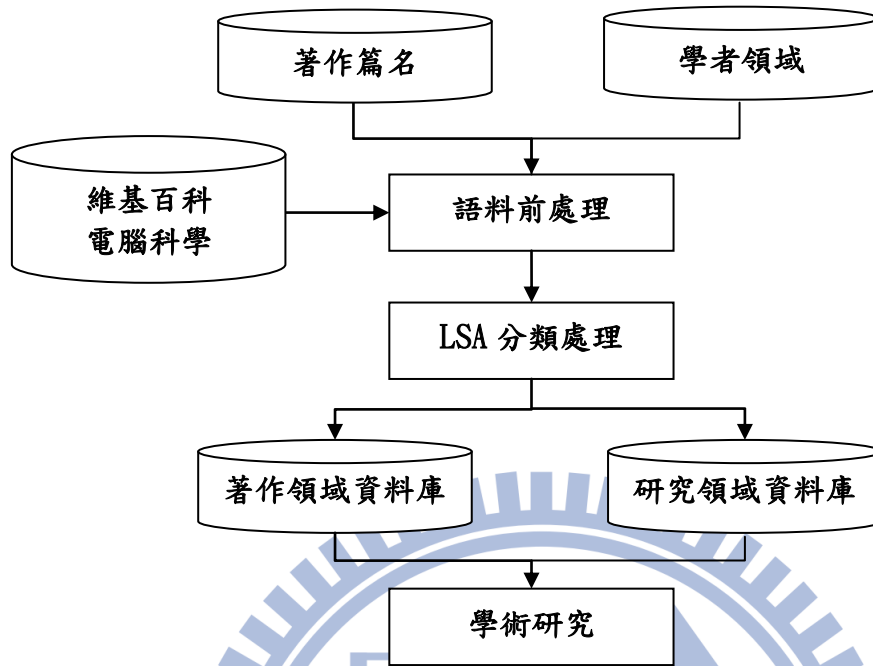


圖 8:學術研究處理流程圖

Deerwester et al.[1990]所提出的 LSA 是一種索引方式。LSA 將詞和文檔對應到潛在語意空間，從中去除原始向量空間中的雜訊。最後利用向量間的關係來判斷詞與文檔間的關係。

我們使用經由前處理的維基百科內文，共六十八個文檔，10,672 個單字詞。文檔與單字詞間使用 TF-IDF，計算每個詞彙對於各文檔中的重要性。TF 代表詞彙在文檔出現的頻率，IDF 涵義為當詞彙在多個文檔中出現，表示該詞彙對於區分文檔的意義變低。TF-IDF 如公式 2。 $N_{i,j}$ 表示詞彙 i 出現在文檔 j 的次數。 D 表示文檔數量，亦即六十八個文檔。 t_i 表示詞彙 i 。

$$M[i,j] = \left(\frac{N_{i,j}}{\sum_k N_{k,j}} \right) \times \log \frac{|D|}{|\{d: N_{i \in d}\}|} \quad (2)$$

經由上述公式計算後，可以得到一個 $68 \times 10,672$ 的矩陣 M 。由於詞彙當中有許多雜訊，亦即不重要的詞彙。接著將矩陣 M 透過 SVD 降維，降維的目的可以將不相關的詞彙刪減。透過 SVD 計算，最後得到 5,192 個詞彙。將 M 表示成三個矩陣相乘，如公式 3 表示。

$$M = U \times S \times V_t \quad (3)$$

我們將經由前置處理後的著作篇名當作輸入，並使用矩陣 V_p 表示。矩陣 V_p 計算輸入詞彙對應到矩陣 M 中詞彙的次數。表示如方式 4 所示。

$$V_p = \begin{bmatrix} N_{t_j} \end{bmatrix} \quad (4)$$

最後，將矩陣 V_p 對應到矩陣 M 中。如公式 5 所示。

$$P = V_p^T \times U \times S^{-1} \quad (5)$$

利用餘旋相似度計算輸入詞彙與哪一個文檔最相似。如公式 6 所示。取出最

大相似度最為該著作篇名所屬的類別。

$$\text{Sim}(P, D_i) = \frac{\sum_{i=1}^{68} W_{P,j} \times W_{i,j}}{\sqrt{\sum_j W_{P,j}^2} \times \sqrt{\sum_i W_{i,j}^2}} \quad (6)$$

範例:

著作篇名: Learning a Merge Model for Multilingual Information Retrieval

前置處理後: learn merg model multilingu inform retriev

相似度計算結果: 49, 7, 50, 26, 13, 33, 64, 30, 56, 62, ...

在相似度排序中，取出最高的相似度表示該篇論文主題的分類。

分類結果(小類): Machine learning

分類結果(大類): Artificial intelligence

依照上述方法，統計每位學者在各領域的研究比例。以交通大學梁婷老師為例，如表 9 所示。藉由研究者的著作篇名及研究領域，可以推測梁婷老師前三個主要研究領域為: Data mining、Network topology、Natural language processing。其餘未顯示的領域，皆為預設值零。表示該研究者在這些領域上沒有相關的研究。

表 9: 梁婷教授研究領域

領域	比例
Data mining	23.00%
Network topology	14.67%
Natural language processing	11.85%
Database management systems	9.74%
Information theory	9.04%

學術研究關係(research)，由研究者的著作領域與研究領域組成。每位研究者之間藉由下列公式 7 計算出彼此間關係強度。公式如下所示。 \vec{P}_i 表示研究者 i 的著作領域向量。如表 10 所示。 \vec{P}_{i_k} 表示研究者 i 的著作領域向量中第 k 項。 $N(\vec{P}_{i_k} = \vec{P}_{j_k})$ 表示 \vec{P}_{i_k} 與 \vec{P}_{j_k} 相同的個數。

$$\text{research}(\vec{P}_i, \vec{P}_j) = \frac{1}{2} \left(\frac{N(\vec{P}_{i_k} = \vec{P}_{j_k}) \times \sum_k (\alpha \times \cos(\vec{P}_{i_k} \times \vec{P}_{j_k}))}{\min\{|\vec{P}_i|, |\vec{P}_j|\}} + \frac{|\vec{P}_i \cap \vec{P}_j|}{|\vec{P}_i \cup \vec{P}_j|} \right) \quad (7)$$

$$\alpha = N(\vec{P}_{i_k} = \vec{P}_{j_k}) - k \quad (8)$$

研究者彼此間如果有共同的研究領域，表示研究者之間有共同的研究興趣。因此，利用基本集合運算，交集(彼此間共同研究領域)除以聯集(彼此間所有研究領域)。透過此計算，彼此間共同的研究領域越多，研究者之間關係強度越顯著。關係越強，表示研究者之間有越多共同的研究領域。相反的，關係強度越弱，表示研究者之間共同的研究領域越少。

表 10: 學術研究關係計算範例

著作領域向量	T_1	T_2	T_3	T_4

\vec{P}_i	0.7	0.2	0.1	0
\vec{P}_j	0.6	0	0.3	0.1
\vec{P}_k	0.5	0	0.3	0.2
\vec{P}_l	0.2	0	0.3	0.5

$\vec{P}_i = [0.7, 0.2, 0.1, 0]$ 表示研究者 i 對研究領域 $T_1 \sim T_4$ 的比重為 70%、20%、10%、0%。

\vec{P}_i 與 \vec{P}_j 共同研究領域為 T_1 與 T_3 ，即 $|\vec{P}_i \cap \vec{P}_j| = 2$ 。 \vec{P}_i 與 \vec{P}_j 所有研究領域為 T_1 、 T_2 、 T_3 與 T_4 ，即 $|\vec{P}_i \cup \vec{P}_j| = 4$ 。故 \vec{P}_i 與 \vec{P}_j 關係強度為 $|\vec{P}_i \cap \vec{P}_j| / |\vec{P}_i \cup \vec{P}_j| = 0.5$ 。

\vec{P}_j 與 \vec{P}_k 共同研究領域為 T_1 、 T_3 與 T_4 ，即 $|\vec{P}_j \cap \vec{P}_k| = 3$ 。 \vec{P}_j 與 \vec{P}_k 所有研究領域為 T_1 、 T_3 與 T_4 ，即 $|\vec{P}_j \cup \vec{P}_k| = 3$ 。故 \vec{P}_j 與 \vec{P}_k 關係強度為 $|\vec{P}_j \cap \vec{P}_k| / |\vec{P}_j \cup \vec{P}_k| = 1$ 。

依據上述計算，得到共同研究領域矩陣如表 11 所示。

表 11: 共同研究領域矩陣

關係強度	\vec{P}_i	\vec{P}_j	\vec{P}_k	\vec{P}_l
\vec{P}_i	1	0.5	0.5	0.5
\vec{P}_j	0.5	1	1	1
\vec{P}_k	0.5	1	1	1
\vec{P}_l	0.5	1	1	1

從矩陣內容可發現， \vec{P}_i 與 \vec{P}_j 、 \vec{P}_k 、 \vec{P}_l 關係強度較低於 \vec{P}_j 、 \vec{P}_k 、 \vec{P}_l 之間的關係強度。表示在共同研究領域中， \vec{P}_j 、 \vec{P}_k 、 \vec{P}_l 包含較多的共同研究領域。

在上述範例中，雖然 \vec{P}_j 、 \vec{P}_k 、 \vec{P}_l 具有相同的研究領域，但在研究比例次序上並

不相同。 \vec{P}_j 、 \vec{P}_k 研究領域次序分別為 T_1 、 T_3 與 T_4 。 \vec{P}_j 、 \vec{P}_k 應該有較強的關係強度，相對於 \vec{P}_l 研究領域次序為 T_4 、 T_3 與 T_1 。 \vec{P}_j 、 \vec{P}_k 除了有相同的研究領域，在這些共同的研究領域上， \vec{P}_j 與 \vec{P}_k 對於研究領域 T_1 具有最高興趣，次之為研究領域 T_3 ，最後為研究領域 T_4 。因此 \vec{P}_j 與 \vec{P}_k 的關係強度越接近。利用下列公式計算研究者間關係強度。

$$(\vec{P}_i, \vec{P}_j) = \frac{N(\vec{P}_i = \vec{P}_j) \times \sum_k (N(\vec{P}_i = \vec{P}_k) - k) \cos(\frac{\vec{P}_i \times \vec{P}_j}{|\vec{P}_i| |\vec{P}_j|}), \text{ if } \vec{P}_i = \vec{P}_j \times [1 - |\vec{P}_i - \vec{P}_j|]}{\min\{|\vec{P}_i|, |\vec{P}_j|\}} \quad (9)$$

\vec{P}_j 與 \vec{P}_k 的研究領域次序皆為 T_1 、 T_3 與 T_4 ，故 $N(\vec{P}_j = \vec{P}_k) = 3$ 。

$$k = 0: (3 - 0) \times (0.6 \times 0.5) = 0.9$$

$$k = 1: (3 - 1) \times (0.3 \times 0.3) = 0.18$$

$$k = 2: (3 - 2) \times (0.1 \times 0.2) = 0.02$$

$$\begin{aligned} (\vec{P}_j, \vec{P}_k) &= \frac{3}{3} \\ &\times \frac{0.9 + 0.18 + 0.02}{\sqrt{3 \times (0.6)^2 + 2 \times (0.3)^2 + 1 \times (0.1)^2} \times \sqrt{3 \times (0.5)^2 + 2 \times (0.3)^2 + 1 \times (0.2)^2}} \\ &\times (1 - (0.6 - 0.5)) \times (1 - (0.3 - 0.3)) \times (1 - (0.2 - 0.1)) \\ &= 0.908 \end{aligned}$$

\vec{P}_j 與 \vec{P}_l 的研究領域次序中只有第二項 T_3 相同，故 $N(\vec{P}_j = \vec{P}_l) = 1$ 。

$$k = 0: (1 - 0) \times (0.3 \times 0.3) = 0.09$$

$$\begin{aligned} (\vec{P}_j, \vec{P}_l) &= \frac{1}{3} \\ &\times \frac{0.09}{\sqrt{1 \times (0.3)^2} \times \sqrt{1 \times (0.3)^2}} \\ &\times (1 - (0.3 - 0.3)) \\ &= 0.333 \end{aligned}$$

$$\begin{aligned}
 (\vec{P}_i, \vec{P}_j) &= \frac{1}{3} \\
 &\times \frac{(1-0) \times (0.7 \times 0.6)}{\sqrt{1 \times (0.7)^2} \times \sqrt{1 \times (0.6)^2}} \\
 &\times (1 - (0.7 - 0.6)) \\
 &= 0.3
 \end{aligned}$$

$$\begin{aligned}
 (\vec{P}_i, \vec{P}_k) &= \frac{1}{3} \\
 &\times \frac{(1-0) \times (0.7 \times 0.5)}{\sqrt{1 \times (0.7)^2} \times \sqrt{1 \times (0.5)^2}} \\
 &\times (1 - (0.7 - 0.5)) \\
 &= 0.267
 \end{aligned}$$

$$(\vec{P}_i, \vec{P}_l) = 0$$

$$\begin{aligned}
 (\vec{P}_k, \vec{P}_l) &= \frac{1}{3} \\
 &\times \frac{(1-0) \times (0.3 \times 0.3)}{\sqrt{1 \times (0.3)^2} \times \sqrt{1 \times (0.3)^2}} \\
 &\times (1 - (0.3 - 0.3)) \\
 &= 0.333
 \end{aligned}$$

對角線部分，皆為預設值 1。最後可得關係矩陣如表 12 所示。

表 12: 關係矩陣

關係強度	\vec{P}_i	\vec{P}_j	\vec{P}_k	\vec{P}_l
\vec{P}_i	1	0.3	0.267	0
\vec{P}_j	0.3	1	0.908	0.333
\vec{P}_k	0.267	0.908	1	0.333
\vec{P}_l	0	0.333	0.333	1

在具有共同研究領域集合的研究者 \vec{P}_j 、 \vec{P}_k 與 \vec{P}_l ， \vec{P}_j 與 \vec{P}_k 的強度關係高於 \vec{P}_j 與

\vec{P}_i 。原因在於， \vec{P}_j 與 \vec{P}_k 在研究領域次序上完全相同，皆為 T_1 、 T_3 、 T_4 。 \vec{P}_j 與 \vec{P}_i 在研究領域次序上只有第二個研究領域相同(即 T_3)。故在考慮研究次序時， \vec{P}_j 與 \vec{P}_k 在研究領域著重次序較優於 \vec{P}_j 與 \vec{P}_i 。

藉由共同研究領域矩陣，以及次序性的研究領域矩陣。將兩矩陣加總，表示研究者之間的學術研究關係。學術研究關係如表 13 所示。

表 13: 學術研究關係

學術研究	\vec{P}_i	\vec{P}_j	\vec{P}_k	\vec{P}_l
\vec{P}_i	1	0.4	0.383	0.25
\vec{P}_j	0.4	1	0.954	0.667
\vec{P}_k	0.383	0.954	1	0.667
\vec{P}_l	0.25	0.667	0.667	1

從表中發現， \vec{P}_i 與 \vec{P}_j 、 \vec{P}_k 、 \vec{P}_l 學術關係較低，原因在於， \vec{P}_i 與其他三者間只有兩個共同研究領域(T_1 、 T_3)。此外， \vec{P}_i 與 \vec{P}_l 學術關係又較低於 \vec{P}_i 與 \vec{P}_j 以及 \vec{P}_i 與 \vec{P}_k 的學術關係。起因為， \vec{P}_i 的研究領域次序為 T_1 、 T_2 、 T_3 ， \vec{P}_j 以及 \vec{P}_k 的研究領域次序為 T_1 、 T_3 、 T_4 ， \vec{P}_l 的研究領域次序為 T_4 、 T_3 、 T_1 。 \vec{P}_i 與 \vec{P}_j 以及 \vec{P}_i 與 \vec{P}_k 在研究領域次序中，第一個研究領域皆為 T_1 。而 \vec{P}_i 與 \vec{P}_l 研究領域次序上，沒有任何相同的研究領域。所以，在計算研究領域次序時， \vec{P}_i 與 \vec{P}_j 以及 \vec{P}_i 與 \vec{P}_k 關係較高於 \vec{P}_i 與 \vec{P}_l 。故 \vec{P}_i 與 \vec{P}_l 學術關係又較低於 \vec{P}_i 與 \vec{P}_j 以及 \vec{P}_i 與 \vec{P}_k 的學術關係。 \vec{P}_j 與 \vec{P}_k 學術關係較高於 \vec{P}_j 與 \vec{P}_l 。研究者 \vec{P}_j 、 \vec{P}_k 與 \vec{P}_l 三者，在共同研究領域中皆為 T_1 、 T_3 、 T_4 。 \vec{P}_j 與 \vec{P}_k 最有興趣的研究領域為 T_1 ，次之為 T_3 ，最後為 T_4 。然

而 $\overrightarrow{P_i}$ 最有興趣的研究領域為 T_4 ，次之為 T_3 ，最後為 T_1 。因此在學術關係中， $\overrightarrow{P_j}$ 與 $\overrightarrow{P_k}$ 學術關係較高於 $\overrightarrow{P_j}$ 與 $\overrightarrow{P_l}$ 。 $\overrightarrow{P_k}$ 與 $\overrightarrow{P_l}$ 學術關係意同於 $\overrightarrow{P_j}$ 與 $\overrightarrow{P_l}$ 。

3.3 學術合作

在眾多研究領域之中，不管是相同研究領域或是跨領域研究，各研究領域之間有著些許相關性，例如：資料庫系統、資料探勘、自然語言處理，三者間含有些許的相關性。藉由研究領域的相關性，研究者之間不論是相同研究領域，或是相關研究領域上，存在某些學術上合作關係的可能性。我們利用論文合作(co-author)以及教育合作(advisor)，建立學術合作關係。

3.3.1 論文合作

論文合作代表著研究者之間透過相關研究領域或合作計畫案而共同發表的論文著作，因此在著作中包含兩位以上研究學者姓名，表示這幾位學者在論文合作上有合作關係。我們從 16,981 篇著作的作者姓名中，找出含有在學術網路中的研究者姓名。若著作作者中包含兩位以上學術網路的學者姓名，並且學者著作列表中都含有此篇著作篇名，表示研究學者之間有論文合作關係。此學術網路上，將這幾位學者間建立論文合作關係。

論文合作 co-author(P_i, P_j)處理流程如下：

P_i : 學者 i 的著作列表; P_j : 學者 j 的著作列表

1. 若 P_i 共同作者中含有 P_j ，紀錄該篇著作篇名 T_i 。
2. 若 P_j 共同作者中含有 P_i ，紀錄該篇著作篇名 T_j 。
3. 若 T_i 與 T_j 相同，co-author(P_i, P_j) 累計。

經由上述處理流程，共 129 位學者彼此之間有論文合作關係。在這 129 位學者中，共同合作的著作共有 852 篇。以交通大學的研究學者為例，顯示學者間共同著作關係。如圖 9 所示。黑色表示交通大學學者，紫色表示清華大學學者，綠色表示台灣大學學者，紅色表示成功大學學者，藍色表示中央大學學者，橘色表示中央研究院學者。

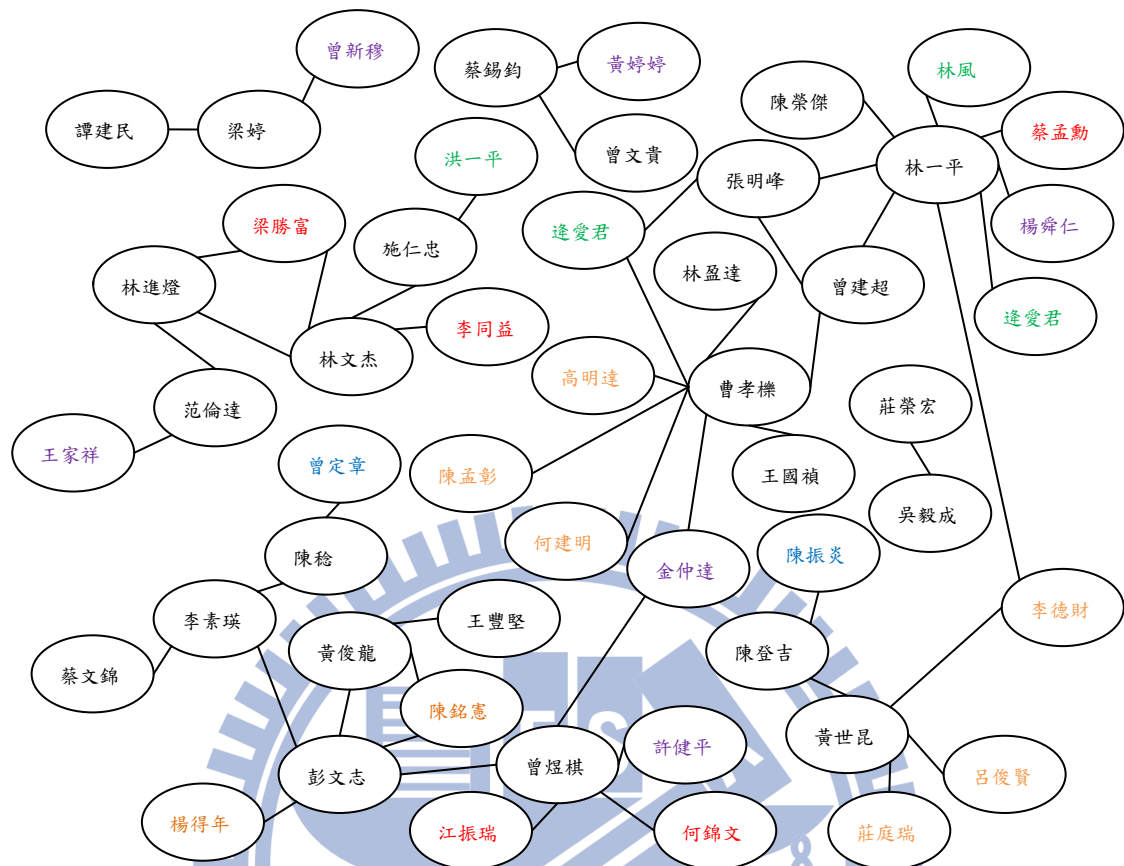


圖 9:論文合作關係圖

3.3.2 教育合作

教育合作表示學者之前曾經共同指導的學生，透過學者曾經指導的碩博士名單中，若有共同指導學生，表示學者間存在教育合作關係。我們從學者個人網頁中收集 5,936 位學生姓名，從中找出 459 位學生是由學術網路中的研究學者共同指導。本學術網路，將這幾位學者間建立教育合作關係。

教育合作 $\text{advisor}(P_i, P_j)$ 處理流程如下：

P_i : 學者 i 的指導學生列表; P_j : 學者 j 的指導學生列表

1. 若存在 P_k : P_i 與 P_j 共同學生姓名
2. 利用台灣博碩士論文知識加值系統
 - 2.1. 輸入學生姓名 P_k
3. 若存在指導教授 P_i 與 P_j (如圖 10 所示)， $\text{advisor}(P_i, P_j)$ 累計。



圖 10: 共同指導教授

以交通大學學者為例，顯示學者間共同指導學生關係如圖 11 所示。從圖中可發現具有共同指導學生的學者之間形成一個連通網路。

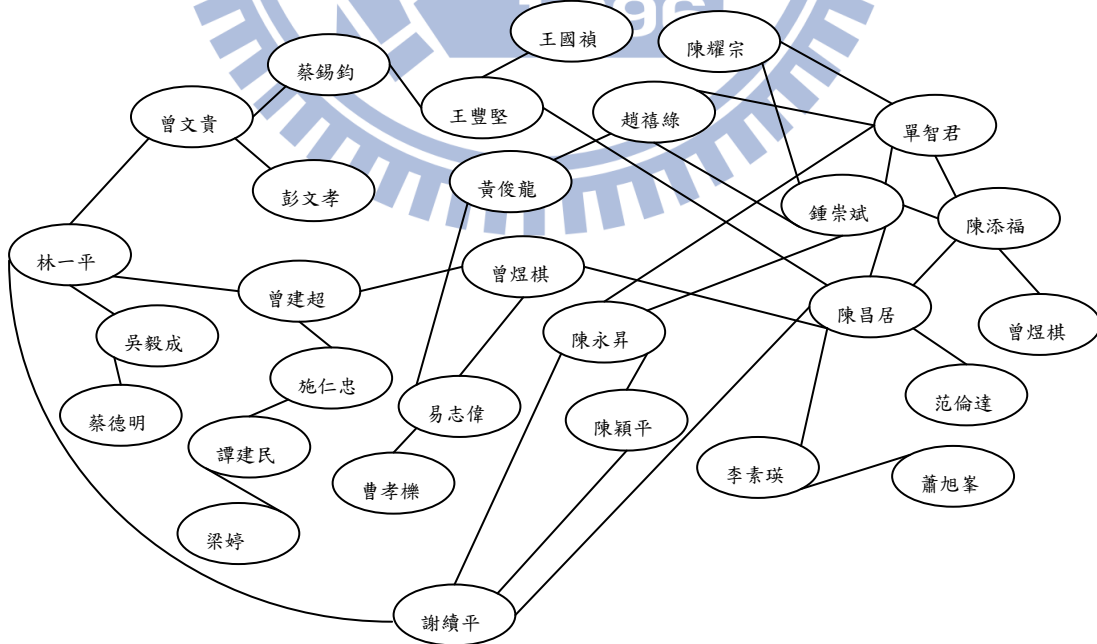


圖 11: 教育合作關係圖

3.4 機構關係

研究學者經歷的研究機構，表示學者之間可能存在同學或學長(姐)學弟(妹)關係。此外研究學者所服務的單位，也是彼此間某種理念上的共同性。本學術網路分別依據研究學者的畢業學校，以及目前服務單位，建立彼此間機構關係。

機構關係 $organization(P_i, P_j)$ 處理流程如下：

P_i : 學者 i 的最高學位畢業學校及服務單位

P_j : 學者 j 的最高學位畢業學校及服務單位

1. 若 P_i 與 P_j 畢業於相同學校且服務於相同機構

則 $organization(P_i, P_j)=2$

2. 若 P_i 與 P_j 畢業於相同學校或服務於相同機構

則 $organization(P_i, P_j)=1$

經機構關係處理後，266 位學者分別畢業於 66 所國內外大學。服務機構中，台灣大學學者 46 位，交通大學學者 75 位，清華大學學者 45 位，成功大學學者 31 位，中央大學學者 33 位，中央研究院學者 36 位。

3.5 關係結果處理

在學術網路中每位研究學者的資訊量如表 14 所示。我們希望利用學術研究(research)、學術合作(co-work)以及機構單位(organization)建立學術網路，如公式 10 建立學者之間的關聯性。研究者之間主要的關係建立於學術研究，透過共同研究領域，學者可以發現與自己共同研究主題的其他研究者。

表 14: 研究學者資訊

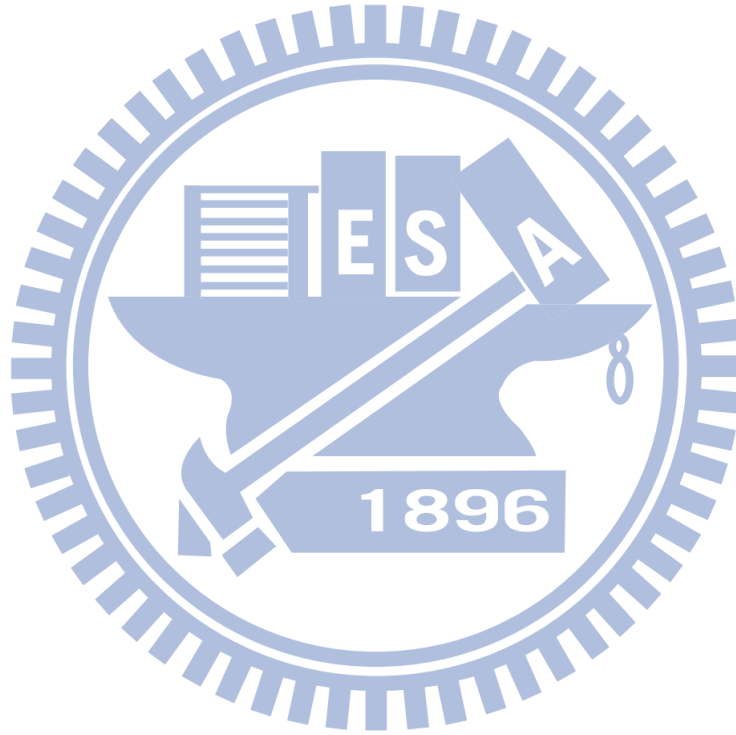
平均	論文著作	篇名長度	研究領域	指導學生
每位學者	63.84 篇	9.87 單詞	3.96 領域	39.56 位

$$\text{relation}(P_i, P_j) = W_i \times \text{research}(\bar{P}_i, \bar{P}_j) + W_j \times \text{cowork}(P_i, P_j) + W_k \times \text{organization}(P_i, P_j) \quad (10)$$

$$\mathbf{research}(\bar{P}_i, \bar{P}_j) = \frac{1}{2} \left(\frac{N(\bar{P}_i = \bar{P}_j) \times \sum_k (\alpha \times \cos(\bar{P}_i \times \bar{P}_j))}{\min\{|\bar{P}_i|, |\bar{P}_j|\}} + \frac{|\bar{P}_i \cap \bar{P}_j|}{|\bar{P}_i \cup \bar{P}_j|} \right) \quad (11)$$

$$\mathbf{cowork}(P_i, P_j) = \frac{1}{2} \left[\frac{\mathbf{co_author}(P_i, P_j)}{\mathbf{MAX}\{\mathbf{co_author}(P_i, P_j)\}} + \frac{\mathbf{advisor}(P_i, P_j)}{\mathbf{MAX}\{\mathbf{advisor}(P_i, P_j)\}} \right] \quad (12)$$

$$\mathbf{organization}(P_i, P_j) = \begin{cases} 1, & \text{若 } P_i \text{ 與 } P_j \text{ 畢業於相同學校且服務於相同機構} \\ 0.5, & \text{若 } P_i \text{ 與 } P_j \text{ 畢業於相同學校或服務於相同機構} \\ 0, & \text{其它} \end{cases} \quad (13)$$



第四章 實驗與實驗分析

在此章節中我們探討第三章著作分類結果的實驗分析。4.1 節中討論我們如何對著作分類進行正確率計算。4.2 節將分類結果透過統計顯示台灣資工學術領域在各領域發展的比例。4.3 節討論研究學者分群後的結果。

4.1 分類實驗

我們從經過分類處理的 16,981 篇著作篇名中，隨機抽取 424 篇著作篇名，其中含有作者提供的關鍵詞彙當作分類實驗。

論文著作中學者會依據該篇著作提供關鍵詞。關鍵詞包含數個單字詞及名詞片語，這些單字詞及名詞片語描述與該篇著作相關的研究主題，以及研究主題中相關的專業術語。有鑑於此，我們利用關鍵詞與各領域詞彙作餘旋相識度。再利用最高相識度與分類結果作比對。我們分別使用單字詞及名詞片語進行著作篇名分類實驗。

範例:

著作篇名: Learning a Merge Model for Multilingual Information Retrieval

分類結果: 49,7,50...

關鍵詞: Learning to merge; Merge model; MLIR

分類結果: 49,50,33...

小類: Machine learning

大類: Artificial intelligence

範例中，依據著作篇名分類出的結果為 Machine learning(49)。根據關鍵詞分類出的結果亦為 Machine learning(49)。因此判定共同性相同。最後，共同性結果如下表 15 所示。

表 15: 共同性結果

共同性	小類(68)	大類(13)
單字詞	61.88%	68.47%
名詞片語	39.15%	43.40%

LSA 可以有效處理隱含語意的特性，故在單字詞的共同性較高於名詞片語。名詞片語較少有隱含語意的特性，故從實驗中我們可以驗證。

4.2 統計結果

我們依據 16,981 著作篇名在各領域的分佈探討六所機構在學術領域的發展。

如表 16 所示。

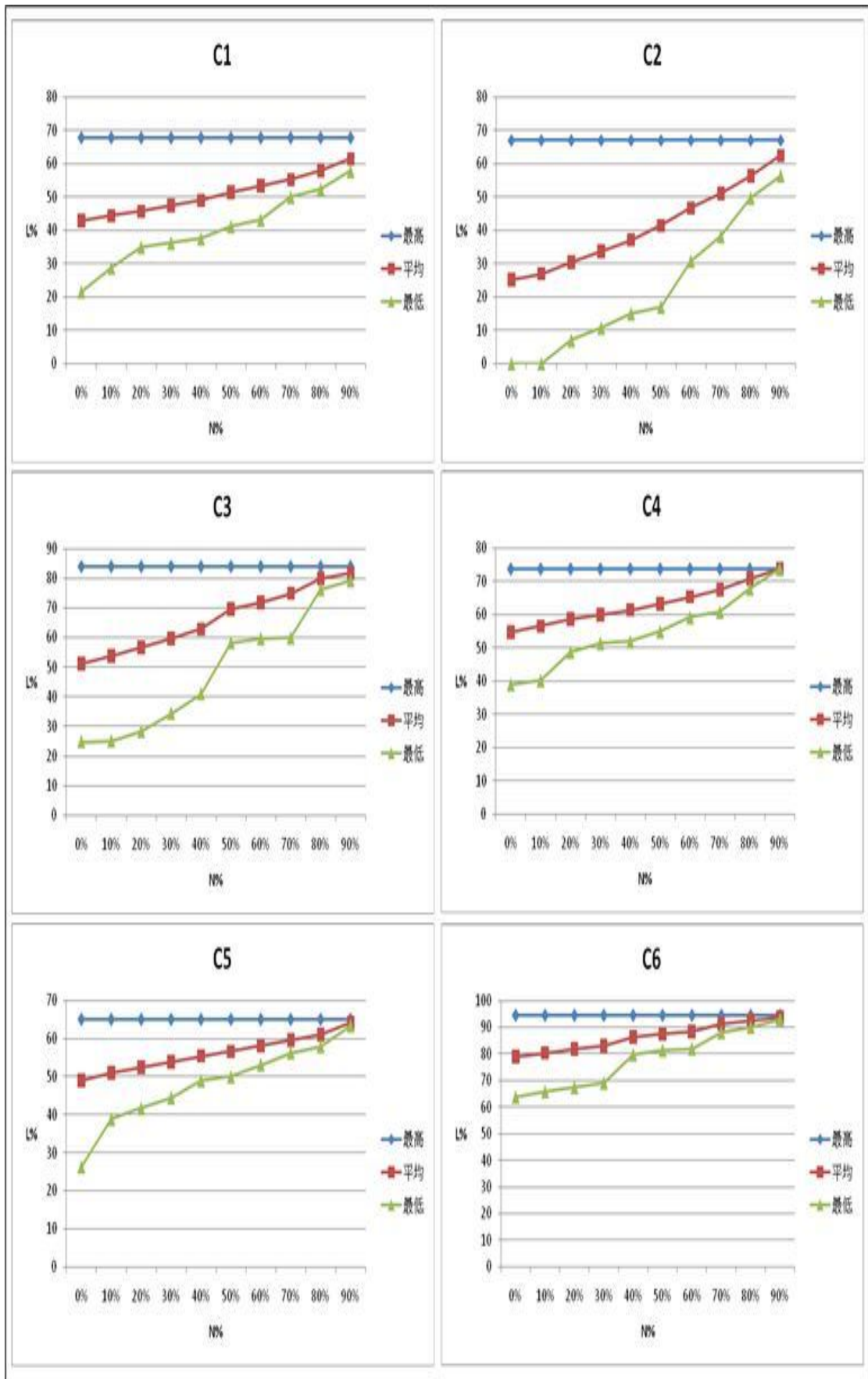
表 16:16, 981 篇著作的發展

領域	篇數	比例
Algorithms, data structures	3, 187	18. 77%
Mathematical foundations	2, 244	13. 21%
Artificial intelligence	1, 843	10. 85%
Telecommunication, networking	1, 803	10. 62%
System architecture	1, 491	8. 78%
Scientific computing	1, 361	8. 01%
Computer graphics	1, 264	7. 44%
Software engineering	1, 025	6. 04%
Programming language, compilers	776	4. 57%
Databases	772	4. 55%
Human-computer interaction	478	2. 81%
Theory of computation	398	2. 34%
Concurrent, parallel, distributed systems	339	2. 00%

六所機構學術領域發展中，前五項著重發展領域為 Algorithms, data structures、Mathematical foundations、Artificial intelligence、Telecommunication, networking、System architecture。

4.3 分群實驗

我們使用 k-means 將 266 位研究學者分群，從中探討分群結果的現象。維基百科將電腦科學定義為 13 個類別，因此我們將 266 位學者分為 13 群。每位學者平均有 3.96 個研究領域，因此取出每群中前 3 個最常出現的研究領域，計算各群學者在這 3 個研究領域的比例加總，取出最低值 L%，表示該群結果中每位研究者最少有 L% 的研究是屬於相同領域。由於分群有某部分比例的誤差，因此我們設立 N%，表示分群結果造成的誤差比例。實驗結果如圖 12 所示。13 群中的研究學者如表 17 所示。



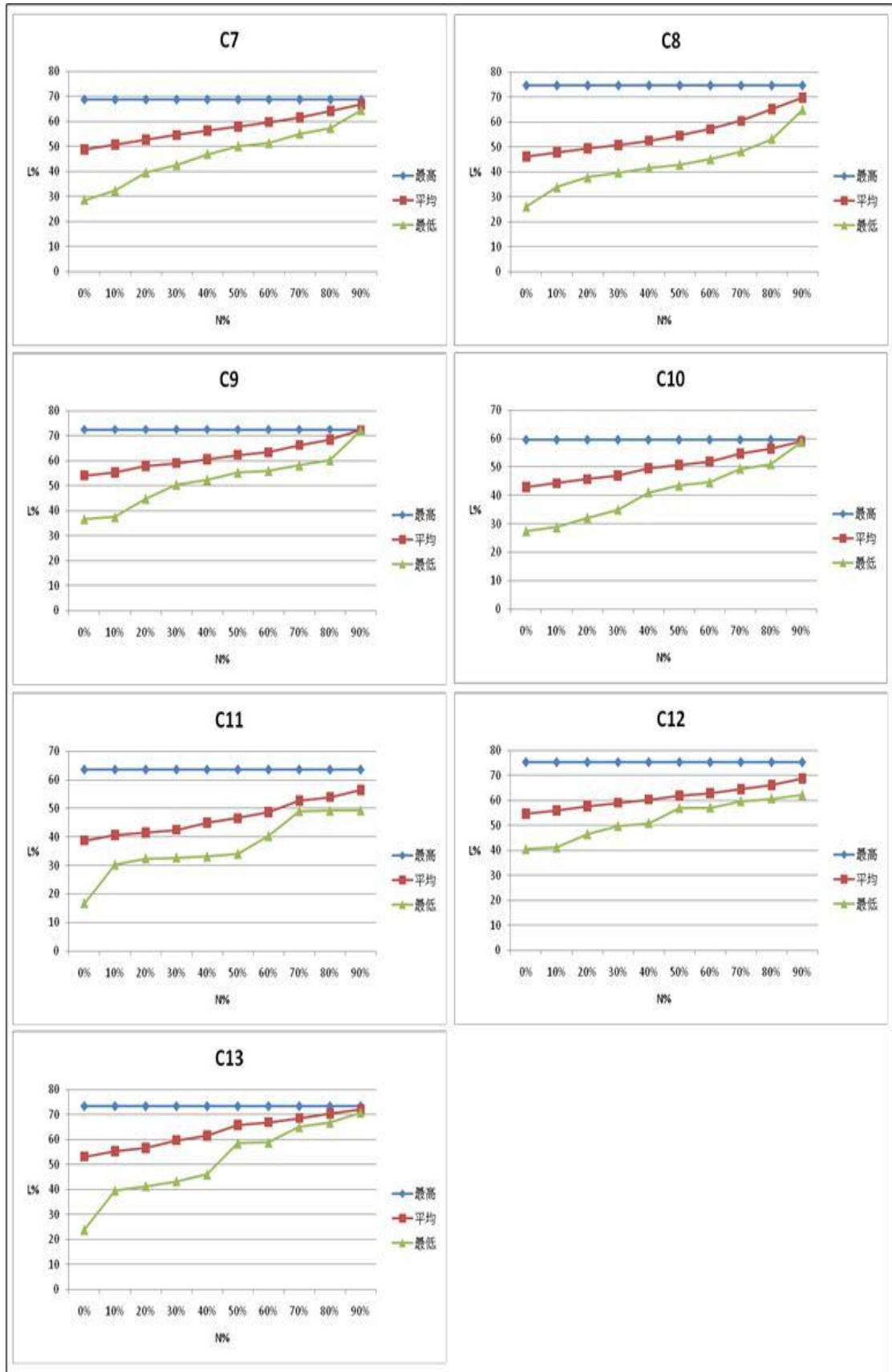


圖 12:13 群分析圖

表 17: 學者分群

群(人數)	領域	研究者
C1(23)	Algorithms (23) Telecommunication(8) Human_computer interaction(7)	趙坤茂,陳信希,朱浩華,劉邦鋒,逢愛君,蔡欣穆,李德財,黃國源,許騰尹,黃廷祿,曹孝櫟,黃婷婷,王炳豐,張俊盛,王廷基,麥偉基,李潤容,李強,楊中平,盧文祥,張貴雲,呂俊賢,李丕榮
C2(59)	Algorithms (20) Scientific computing(11) Databases(9)	陳俊良,洪士灝,李琳山,施吉昇,林寶樹,陳添福,荆宇泰,鍾崇斌,李嘉晃,林志青,孫春在,袁賢銘,胡毓志,單智君,蔡文能,陳健,陳俊穎,蕭旭峯,蕭子健,王才沛,陳文村,張世杰,石維寬,徐爵民,邱澗德,朱宏國,蘇文鈺,張大緯,范國清,曾黎明,洪炯宗,陳彥仰,鄭卜壬,薛智文,顧孟愷,劉長遠,曾宇鳳,楊佳玲,莊榮宏,傅心家,施仁忠,吳毅成,林正中,鍾葉青,陳宜欣,韓永楷,李哲榮,曾新穆,黃宗立,高宏宇,林英超,蘇木春,張嘉惠,陳慶瀚,鄭永斌,葉士青,吳真貞,林仲彥,陳銘憲
C3(12)	Software engineering(12) Mathematical foundations(5) Algorithms (4)	蔡德明,劉啟民,李政崑,張克正,周百祥,黃慶育,黃稚存,朱治平,李允中,陳振炎,黃為德,王柏堯
C4(10)	Mathematical foundations(10) Computer graphics(10) Algorithms(7)	張瑞峰,歐陽明,蔡文祥,黃世強,張隆紋,陳煥宗,李同益,郭淑美,鄭旭詠,黃文良
C5(20)	Computer graphics(20) Artificial intelligence(13) Algorithms(12)	陳炳宇,徐宏民,李明穗,陳玲慧,陳稔,莊仁輝,陳永昇,林奕成,林文杰,賴尚宏,陳朝欽,許秋婷,孫永年,連震杰,王士豪,曾定章,蘇柏齊,施純傑,張復,陳祝嵩
C6(13)	Telecommunication (13) Algorithms (11) Mathematical foundations(4)	陳耀宗,林盈達,易志偉,趙禧綠,黃能富,林華君,蔡明哲,高榮駿,許靜芳,蘇銓清,周立德,孫敏德,陳伶志

C7(31)	Mathematical foundations(31) Algorithms (31) System architecture(13)	呂育道,洪一平,陳文進,呂學一,吳家麟,劉文泰,陳榮傑,陳昌居,彭文志,彭文孝,蔡淳仁,范倫達,吳育松,蔡仁松,鄭復華,潘雙洪,郭耀煌,陳培殷,梁勝富,何錦文,江振瑞,王尉任,王家慶,鄭伯順,呂及人,徐讚昇,高明達,許聞廉,陳克健,楊柏因,葉彌妍
C8(31)	Artificial intelligence(31) Scientific computing(16) System architecture(12)	許永真,莊永裕,傅楸善,傅立成,高成炎,林智仁,林軒田,歐陽彥正,蘇雅韻,王傑智,林進燈,陳穎平,林永隆,蘇豐文,王俊堯,吳宗憲,蔣榮先,簡仁宗,陳國棟,楊鎮華,施國琛,陳德懷,楊接期,黃武元,王新民,許鈞南,陳郁方,廖弘源,劉庭祿,劉進興,蔡懷寬
C9(16)	Telecommunication (16) Algorithms (16) Mathematical foundations(13)	林風,周承復,林守德,謝續平,曾建超,王協源,李毅郎,黃俊龍,王家祥,楊舜仁,吳尚鴻,黃崇明,藍崑展,吳曉光,何建明,陳昇璋
C10(13)	Programming languages (13) Mathematical foundations(9) Algorithms (7)	項潔,廖世偉,陳登吉,徐慰中,王豐堅,游逸平,蔡孟勳,梁德容,王建民,莊庭瑞,游本中,廖純中,穆信成
C11(13)	Telecommunication (13) Algorithms (7) System architecture(6)	張明峰,簡榮宏,曾煜棋,王國禎,楊啟瑞,黃世昆,邵家健,許健平,張智星,鄭憲宗,張燕光,何宗易,許富皓
C12(11)	Mathematical foundations(11) Algorithms (8) Telecommunication (5)	郭大維,譚建民,蔡錫鈞,蔡文錦,張立平,金仲達,李端興,謝孫源,黃興燦,顏嵩銘,王大為
C13(14)	Mathematical foundations(14) Telecommunication(14) Algorithms(8)	陳健輝,賴飛熊,林一平,陳志成,李素瑛,曾文貴,楊武,梁婷,孫宏民,張適宇,蕭宏章,宋定懿,陳孟彰,楊得年

我們另外利用 HAC(Hierarchical Agglomerative Clustering)將 266 學者分群，從中探討分群結果的現象。HAC 每一回將學者合併成群時，計算該群學者在各自領域中最有興趣的研究領域在該群的比例 M%。另外，計算 F% 表示該群學者

各自領域中前 4 個研究領域最常出現的研究領域比例。T% 表示群的成員中在各自研究領域中前 4 個研究領域與彼此之間的比例。我們首先透過 research 關係建立分群，圖 13 顯示分群結果。

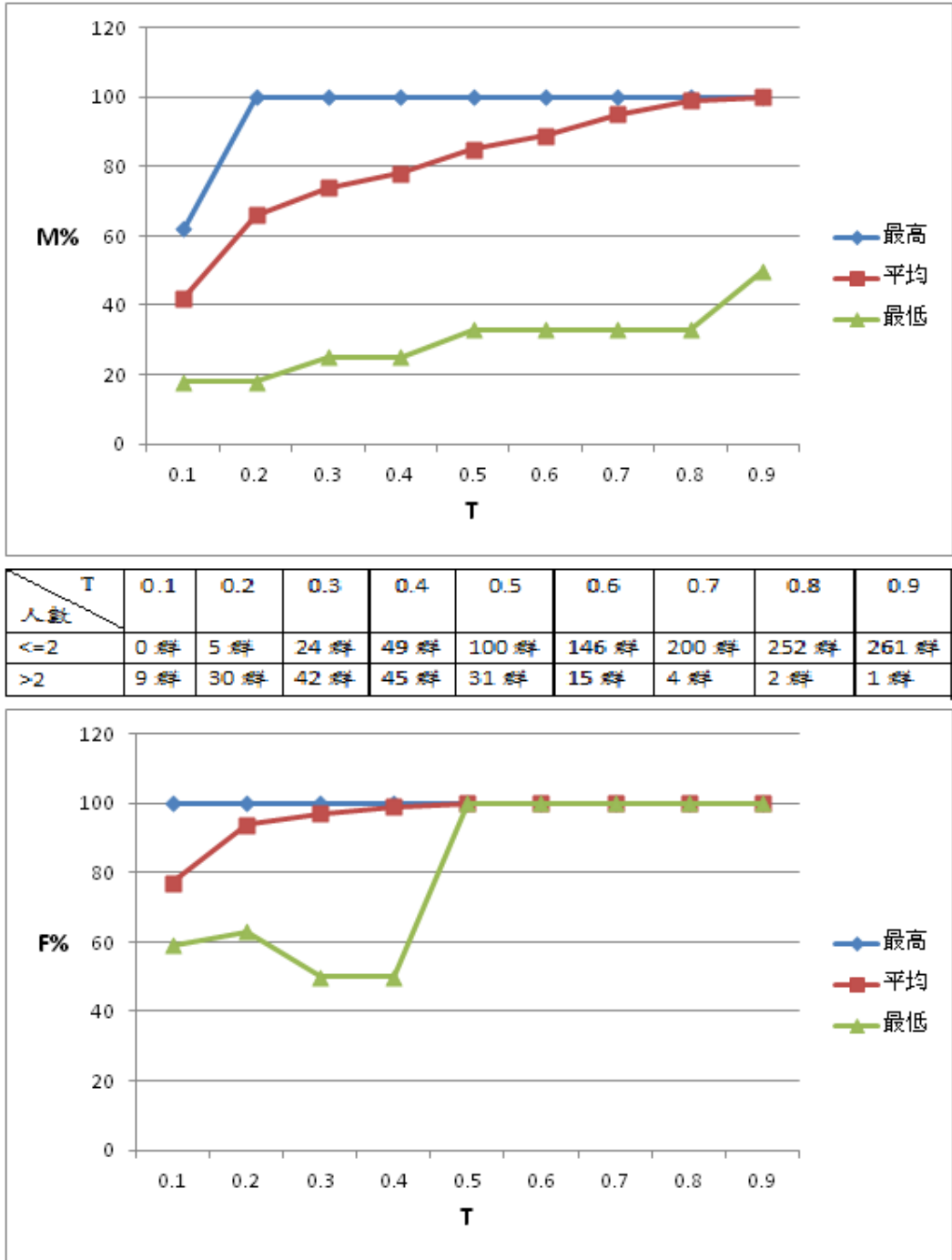
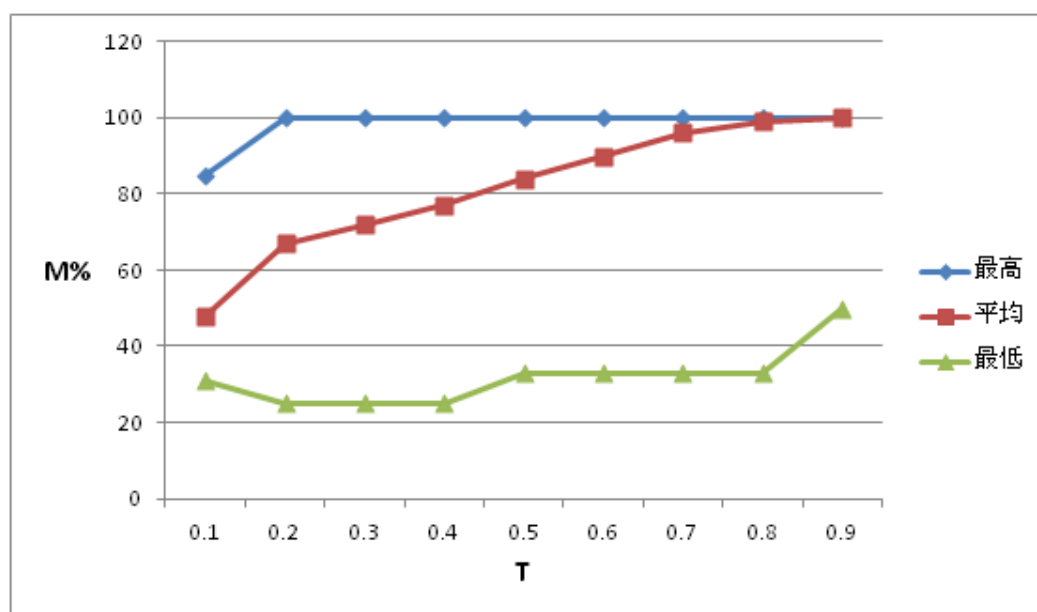


圖 13:研究領域分群分析圖

在 T=0.9 時存在一群人數超過兩人，此群中的學者分別為王炳豐教授、王廷

基教授及麥偉基教授。此群中的學者前三項主要研究主題次序完全相同分別為 Algorithm design、Automated reasoning 及 Analysis of algorithms。



T \ 人數	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
<=2	0 群	4 群	20 群	44 群	96 群	148 群	204 群	252 群	261 群
>2	9 群	33 群	44 群	48 群	32 群	14 群	3 群	2 群	1 群

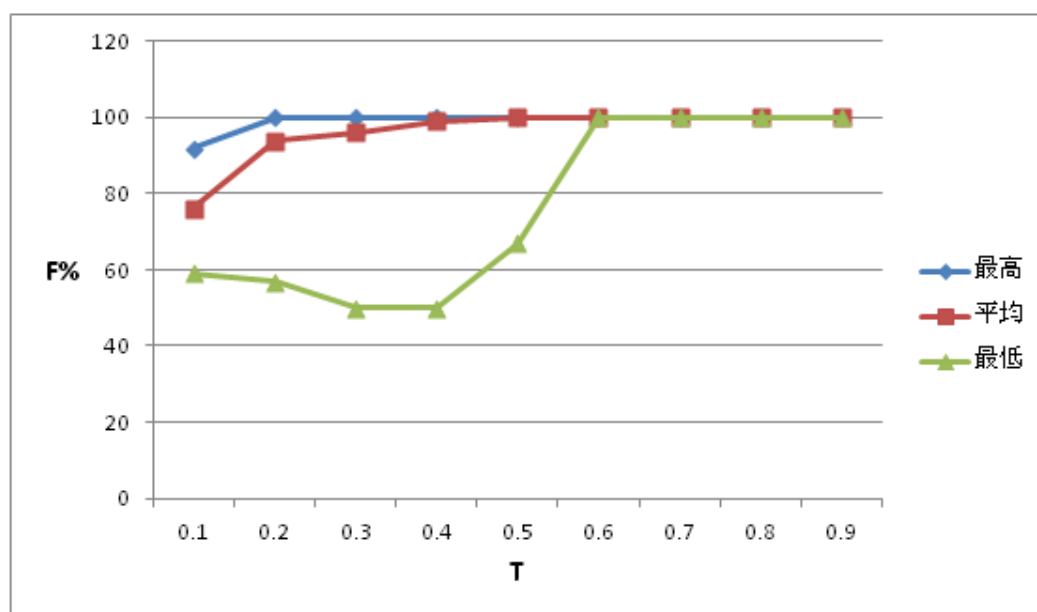


圖 14:研究領域與合作關係分群分析圖

圖 14 顯示利用研究領域關係與合作關係建立的分群結果。我們將 T=0.2 時將學者之間的分群結果如表 18 所示。

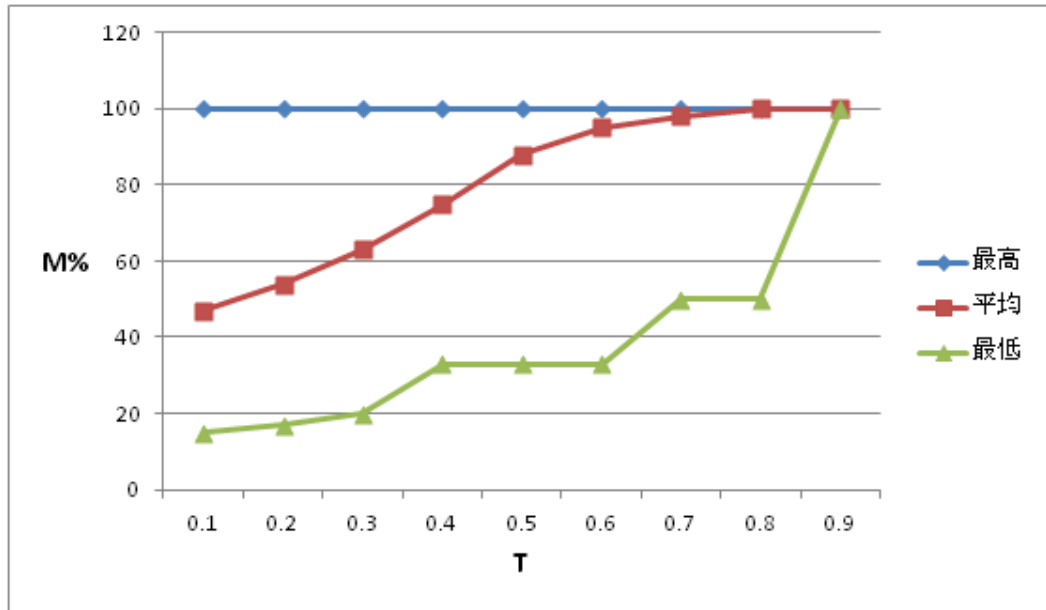
表 18:學術研究與學術合作分群結果

呂育道 陳信希 歐陽彥正 陳榮傑 蔡錫鈞 彭文志 黃俊龍 洪炯宗 何建明 蔡懷寬
呂學一 莊榮宏 鄭復華 潘雙洪 朱宏國 謝孫源 何錦文 顏嵩銘 王大為 呂及人 宋定懿 徐讚昇 高明達 許聞廉 楊柏因
曾文貴 邵家健 孫宏民 黃宗立
趙坤茂 逢愛君 李德財 李毅郎 王炳豐 王廷基 麥偉基 韓永楷 李丕榮
吳毅成 陳穎平 王家祥 石維寬 楊舜仁 吳真貞
劉邦鋒 黃廷祿
陳健輝 周承復 林寶樹 陳志成 陳耀宗 林盈達 王協源 易志偉 趙禧綠 陳健黃能富 林華君 張適宇 高榮駿 許靜芳 蘇銓清 周立德 吳曉光 孫敏德 陳伶志 楊得年
林一平 謝續平 曾建超 楊武 金仲達 李端興 鄭憲宗 黃崇明 蕭宏章 曾黎明 許富皓 陳孟彰
張明峰 簡榮宏 曾煜棋 王國禎 楊啟瑞 蔡文能 許健平
蔡欣穆 許騰尹 蔡明哲 陳昇璋
廖世偉 陳文村 黃興燦
陳彥仰 鄭卜壬 譚建民 梁婷 陳宜欣 吳尚鴻 曾新穆 李強 高宏宇 張嘉惠 陳銘憲 葉彌妍
荊宇泰
洪士灝 陳登吉 陳昌居 藍崑展
賴飛羆 陳添福 王豐堅 袁賢銘 蔡淳仁 游逸平 鍾葉青 蔡仁松 郭耀煌 楊中平 梁德容 王家慶
周百祥 王俊堯 黃稚存 朱治平 林英超 鄭永斌
顧孟愷 黃世昆 曹孝櫟 范倫達 李政崑 黃慶育 李允中 陳振炎
高成炎 陳俊穎 陳培殷
黃為德
鍾崇斌 徐慰中 單智君 李哲榮 張燕光 王建民 游本中
洪一平 傅楸善 薛智文 李琳山 楊佳玲 劉文泰 劉啟民 黃婷婷 楊接期 黃文良
吳家麟 林正中 李潤容 王尉任 呂俊賢
林風 蔡文錦 莊庭瑞 穆信成
張瑞峰 林守德 曾宇鳳 吳育松 張俊盛 盧文祥 張貴雲 鄭伯順 陳克健 陳郁方
陳俊良 陳文進 郭大維 施吉昇 蘇雅韻 張立平 張大緯 蔡孟勳 江振瑞 林仲彥
陳炳宇 莊永裕 歐陽明 施仁忠 林奕成 林文杰 黃世強 李同益
徐宏民 陳玲慧 林志青 陳朝欽 許秋婷 蘇柏齊

蔡文祥 陳稔 莊仁輝 陳永昇 賴尚宏 陳煥宗 曾定章
張隆紋 蘇文鈺
黃國源 孫永年 王士豪
李明穗 李素瑛 彭文孝 郭淑美 梁勝富 鄭旭詠
王才沛 邱澗德 連震杰 范國清 蘇木春 陳慶瀚 搭荀儔 陳祝嵩 劉庭祿
朱浩華 傅立成 王傑智 林進燈 李嘉晃 葉士青 劉進興
林智仁 林軒田 胡毓志 蘇豐文 蔣榮先 陳國棟 楊鎮華 施國琛 陳德懷 黃武元 許鈞南
劉長遠 蔡德明 傅心家 孫春在 張智星 吳宗憲 簡仁宗 王新民
許永真 項潔 蕭旭峯 林永隆 徐爵民 張克正 王柏堯 廖純中
蕭子健 張世杰 何宗易 施純傑 廖弘源

最後利用學者之間學術研究、合作關係及機構關係進行分群。圖 15 顯示利用上述三種關係所建立的分群結果。





人數 \ T	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
<=2	5 群	19 群	43 群	107 群	175 群	212 群	243 群	260 群	263 群
>2	16 群	39 群	54 群	32 群	8 群	5 群	1 群	1 群	1 群

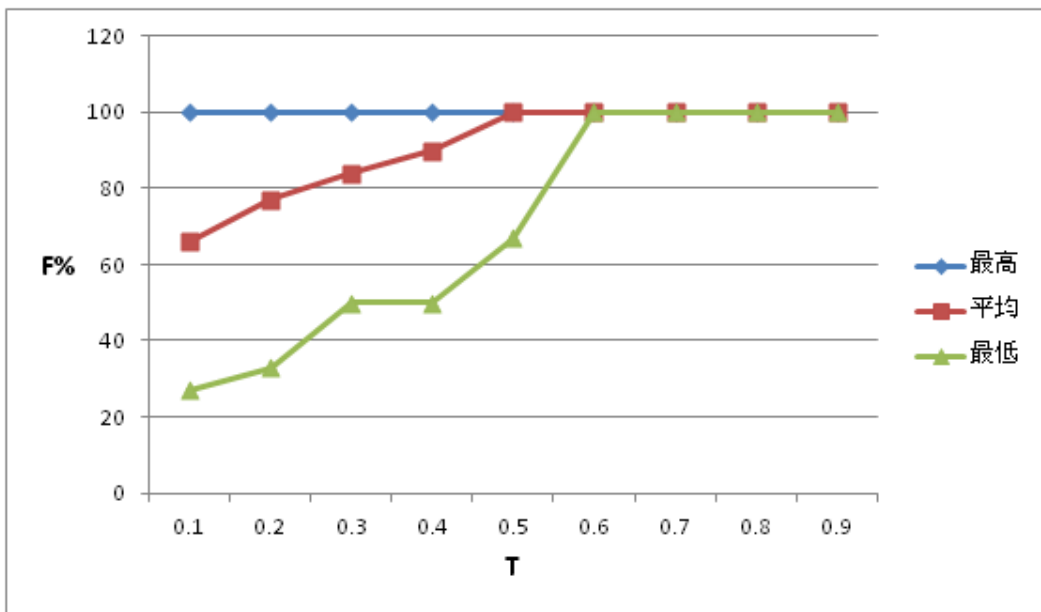


圖 15:研究領域、合作關係及機構關係分群分析圖

從分析圖中可以發現利用研究領域與合作關係的分群結果與單一使用研究領域的分群結果有較高的共同性。其中意味著有相關研究領域的學者在論文著作與指導學生中有較高的合作關係，再透過研究領域、合作關係與機構關係的分群分析圖可以發現各群中的研究領域共同性較低。

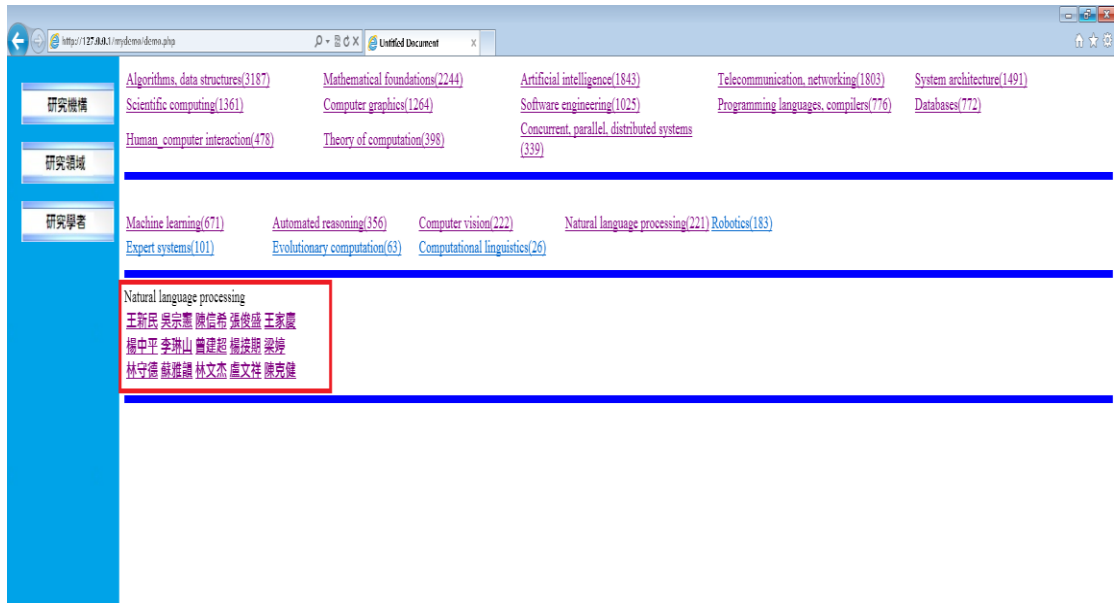


圖 16:自然語言處理相關學者

圖 16 顯示自然語言處理主題中的相關研究者學者。其中王家慶教授的研究領域語音/音訊處理被本系統誤判為自然語言處理，另外楊中平教授、曾建超教授、蘇雅韻教授與林文杰教授的研究領域在本系統中皆有誤判的情形。

name	eo_field	top_field
呂育通 傅維善 徐宏民	Algorithm design	Image processing
羅坤茂 陳信奇 李明棟 劉長輝 呂學一 歐陽志正 曾宇鳳 李德財	Analysis of algorithms	Algorithm design
林鳳 劉打鐘 褚秉君	Algorithm design	Algorithm design
周承強 林守謙 蔡啟輝	Analysis of algorithms	Analysis of algorithms
洪一平 陳俊良 陳文進 莊永裕 薛智文 鄭孟怡 鄧大維 賴瑞麗 李琳山 施吉兒 吳家麒 楊佳玲	Analysis of algorithms	Analysis of algorithms
陳彥仰 朱流慧 傅立成 洪士淵 林賢仁 林軒田 蘇雅韻	Ubiquitous computing	Machine learning
許承真 項潔 廖世偉	Operating systems	Logic programming
張成炎	Bioinformatics	Bioinformatics
潘瑞峰 鄭伯康	Operating systems	Image processing
王樹賢 劉進興	Robotics	Robotics
王建民 王新民 林仲彥 陳昇瑞 蔡懷真	Analysis of algorithms	Object-oriented programming
陳仰方 廖鋒中	Automated reasoning	Logic programming
陳銘華 羅雅斌	Database management systems	Data mining
呂俊賢 陳克健 鍾信成	Computer programming	Analysis of algorithms
李不樂 符錦發 馮徵 莊啟瑛 陳祝壽 高文良 曹弘源 劉啟輝	Image processing	Image processing
王大為 何建明 吳真真 呂及人 宋定誠 徐錦昇 高明達 許麗麗 楊柏因	Graph theory	Graph theory
王培培	Formal methods	Formal methods
蔣本中	Computer architecture	Computer architecture
陳明宇 歐陽明 林遠煌 蔡文祥 陳松 莊仁輝 莊樂宏 黃雨濼 林志清 蔣仁忠 林奕成 黃世雅 吳育松	Image processing	Image processing
顧文豪 陳玲玲 李素琪 賴啟民 許麗尹 黃世淵 梁梓 彭文亨	Analysis of algorithms	Image processing
陳建民 曹文欣 鄧家謙	Network topology	Graph theory
陳會良 陳添福 羅崇軍 徐蝦中 王麗駟 蔡賢儀 陳昌熾 單智君 陳復康 蕭怡? 蔡淳仁 游逸平	Type theory	Computer architecture
陳德偉 蔡新鈞 林文杰	Graph theory	Set theory
蔡德明 吳毅成 陳詠平 黃廷禧 李振郎	Algorithm design	Algorithm design
傅心家 李麗麗 吳有在 胡毓志	Artificial life	Artificial life
簡宇宗 林正中	Concurrency control	Concurrency control
林寶樹 陳志成 陳顯宗 林益達 王遠源 易志偉 趙儒鋒 陳健 黃崇明 馬立德 吳榮光 張敏德	Network topology	Network topology
林一平 馮明鋒 葉樂宏 謝廣平 曾建超 曾煥儀 王國祜 楊凱 楊啟瑛 蔡文龍 許健平 曾黎明	Network topology	Network topology
蘇發清 陳怡志 陳孟彰 楊得年	Network topology	Network topology
彭文豪 曹孝輝 潘立平 黃俊龍 范偉達 洪偉宗 楊嘉豐	Algorithm design	Software design
鄭卜王 蔡文錦 曾新穆 吳宗憲 盧文祥 梁勝富 張大緯 蔡孟勳	Type theory	Type theory
鄧憲宗 李培 朱治平 蘇文茂 藍冠康 林英超	Type theory	Analysis of algorithms
郭耀煌 李阿益 龍瑞源 陳培毅 潘清光	Software design	Image processing

圖 17:學者分群展示

圖 17 顯示利用學者彼此之間的學術研究、合作關係及機構關係所建立的分群結果。以鄭卜王教授、蔡文錦教授、曾新穆教授、吳宗憲教授、盧文祥教授、梁勝富教授、張大緯教授、蔡孟勳教授的主要研究領域及共同研究領域皆被判定為 type theory，主要原因在於該群學者中有較多學者的主要領域為 type theory，此外這些學者也有些許的研究著作被本系統判定為 type theory。

第五章 結論

本論文實作一個台灣資訊學術網路，利用研究領域、著作篇名、指導學生、共同作者、服務機構以及畢業學校建立研究學者之間的關係。本論文的主要貢獻如下：

1. 建立 266 位研究學者資料庫，資料庫包含學術研究、學術合作和機構單位。
2. 領域分類中利用維基百科資訊領域內文自動對著作篇名分類。
3. 研究學者共同研究領域關係。利用研究領域及著作篇名探討研究者之間在研究領域上的關係。
4. 分類 16,981 篇著作篇名對應至維基百科資訊工程領域類別。
5. 比較單字詞與名詞片語在分類的相似性。
6. 研究學者之間共同合作關係，共同著作及共同指導學生。

在本論文的未來研究，有下列幾個方向：

1. 研究人員資訊自動擷取。
2. 維基百科內文依據結構給予不同權重。例如：超連結、infobox…等。
3. 研究人員關係擴充。例如：學術活動、研究計畫…等。
4. 考慮時間次序問題。例如：在任一時間點資訊研究領域的發展。

參考文獻

- [1] Deerwester, S. , Dumais, S. T. , Furnas, G. W. , Landauer, T. K. , & Harshman, R. (1990) , "Indexing By Latent Semantic Analysis", Journal of the American Society For Information Science,41, 391-407. 10
- [2] Griths T. L, and Steyvers M. (2004) , "Finding scientific topics", Proceedings of the National Academy of Sciences,101 (suppl. 1), 5228-5235.
- [3] Blei D. M., Ng A. Y., and Jordan M. I. (2003) , "Latent Dirichlet Allocation",Journal of Machine Learning Research, 3 : 993-1022.
- [4] Y. Yang, (1999) , "An evaluation of statistical approaches to text categorization", Information Retrieval, 1, pp. 69-90.
- [5] Lagus, K, Honkela, T., Kaski, S., and Kohonen, T. (1999) , "WEBSOM for textual data mining", Artificial Intelligence Review, 13 (5{6), pp. 345-364.
- [6] Kaski, S. (1998) , " Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering", In Proceedings of IJCNN '98, International Joint Conference on Neural Networks 1 413–418. IEEE Service Center: Piscataway, NJ.
- [7] Berry, M.W., Dumais, S. T., and O'Brien, G. W. (1995) , "Using linear algebra for intelligent information retrieval", SIAM Review, vol. 37, no. 4, pp. 573-595.
- [8] Hofmann, T. (1999) , "Probabilistic latent semantic indexing", in Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99)
- [9] Rosen-Zvi, M., Griths, T., Steyvers, M., Smyth, P. (2004), "The author-topic model for authors and documents", Proceedings of the 20th UAI Conference
- [10] Tang J., Zhang J., Yao L., Li J., Zhang L., and Su. Z. (2008), "Arnetminer: Extraction and mining of academic social networks". In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'08), pages 990–998.
- [11] 楊瀟，馬軍，楊同峰，杜言琦，邵海敏(2010)，"主題模型LDA的文檔自動文摘"，智能系統學報
- [12] 李文波，孫樂，黃瑞紅，馮元勇，張大鯤(2008)，"基於Labeled-LDA模型的本文分類新算法"，第三屆全國信息檢索與內容安全學會會議
- [13] Porter M(1980), Porter stemming algorithm, software available at <http://nltk.googlecode.com/svn/trunk/doc/api/nltk.stem.porter-module.html>
- [14] Chris Manning, Dan Jurafsky(2010), Stanford parser, software available at <http://nlp.stanford.edu/software/lex-parser.shtml>
- [15] Schutze, H., Hull, D. A, et.al. "A comparison of classifiers and document representations for the routing problem", In Proceedings of SIGIR-95, 1995.

229–237.

- [16] L Chen, N Tokuda, A Nagai (2003). "A new differential LSI space-based probabilistic document classifier", *Information Processing Letters* 2003,88(5):203-212
- [17] Steyvers M., Smyth P., and Griffiths T. (2004). "Probabilistic author-topic models for information discovery. ", In Proc. of SIGKDD'04.
- [18] Griffiths T.L., Steyvers M., Blei D., and Tenenbaum J.B. "Integrating topics and syntax. " In *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.
- [19] McCallum A., Corrada Emmanuel A., and X. Wang. "The author-recipient-topic model for topic and role discovery in social networks". Technical Report UM-CS-2004-096, Department of Computer Science, University of Massachusetts, 2004.
- [20] Buntine W., Lofstrm J., Perki J., Perttu S., Poroshin V., Silander T., Tirri H., Tuominen A., and Tuulos. V."A scalable topic-based open source search engine". In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 228–234, 2004.
- [21] Blei D. and Lafferty J.. "Correlated topic models". In *Neural Information Processing Systems*, volume 18,2006.
- [22] Shou-de Lin and Hans Chalupsky ,”using unsupervised link discovery methods to find interesting facts and connections in a bibliography dataset”, in a bibliography dataset. *SIGKDD Explorations*, 5(2) 173-178, December 2003
- [23] Shou-de Lin and Hans Chalupsky. Unsupervised Link Discovery in Multi-relational Data via Rarity Analysis. In *Proceedings of the Third IEEE International Conference on Data Mining*. Melbourne, Florida. 2003

附錄

表 19: 維基百科電腦科學分類

大類	小類
Mathematical foundations	<ul style="list-style-type: none"> Mathematical logic · Set theory Number theory Graph theory Type theory Category theory Numerical analysis Information theory
Theory of computation	<ul style="list-style-type: none"> Automata theory Computability theory Computational complexity theory Quantum computing theory
Algorithms, data structures	<ul style="list-style-type: none"> Analysis of algorithms Algorithm design Computational geometry
Programming language, compilers	<ul style="list-style-type: none"> Parsers Interpreters Procedural programming Object-oriented programming · Functional programming · Logic programming · Programming paradigms
Concurrent, parallel, distributed systems	<ul style="list-style-type: none"> Multiprocessing · Grid computing · Concurrency control
Software engineering	<ul style="list-style-type: none"> Requirements analysis · Software design · Computer programming · Formal methods · Software testing · Software development process
System architecture	<ul style="list-style-type: none"> Computer architecture · Computer organization ·

	Operating systems
Telecommunication, networking	Computer audio · Routing · Network topology · Cryptography
Databases	Database management systems · Relational databases · SQL · Transactions · Database indexes · Data mining
Artificial intelligence	Automated reasoning · Computational linguistics · Computer vision · Evolutionary computation · Expert systems · Machine learning · Natural language processing · Robotics
Computer graphics	Visualization · Computer animation · Image processing
Human–computer interaction	Computer accessibility · User interfaces · Wearable computing · Ubiquitous computing · Virtual reality
Scientific computing	Artificial life · Bioinformatics · Cognitive science · Computational chemistry · Computational neuroscience · Computational physics · Numerical algorithms · Symbolic mathematics

表 20: STOPWORDS

a	a,a's,able,about,above,according,accordingly,across,actually,after,afterwards,again,against,ain't,all,allow,allows,almost,alone,along,already,also,although,always,am,among,amongst,an,and,another,any,anybody,anyhow,anyone,anything,anyway,anyways,anywhere,apart,appear,appreciate,appropriate,are,aren't,around,as,aside,ask,asking,associated,at,available,away,awfully
b	b,be,became,because,become,becomes,becoming,been,before,beforehand,behind being,believe,below,beside,besides,best,better,between,beyond,both,brief,but,by
c	c,c'mon,c's,came,can,can't,cannot,cant,cause,causes,certain,certainly,changes,clearly co,com,come,comes,concerning,consequently,consider,considering,contain containing,contains,corresponding,could,couldn't,course,currently
d	d,definitely,described,despite,did,didn't,different,do,does,doesn't,doing,don't,done down,downwards,during
e	e,each,edu,eg,eight,either,else,elsewhere,enough,entirely,especially,et,etc,even,ever every,everybody,everyone,everything,everywhere,ex,exactly,example,except
f	f,far,few,fifth,first,five,followed,following,follows,for,former,formerly,forth,four,from further,furthermore
g	g,get,gets,getting,given,gives,go,goes,going,gone,got,gotten,greetings
h	h,had,hadn't,happens,hardly,has,hasn't,have,haven't,having,he,he's,hello,help, hence,her,here,here's,hereafter,hereby,herein,hereupon,hers,herself,hi,him, himself,his,hither,hopefully,how,howbeit,however
i	i,i'd,i'll,i'm,i've,ie,if,ignored,immediate,in,inasmuch,inc,indeed,indicate,indicated Indicates,inner,insofar,instead,into,inward,is,Isn't,it,it'd,it'll,it's,its,itself
j	j,just
k	k,keep,keeps,kept,know,knows,known
l	l,last,lately,later,latter,latterly,least,less,lest,let,let's,like,liked,likely,little,look Looking,looks,ltd
m	m,mainly,many,may,maybe,me,mean,meanwhile,merely,might,more,moreover most,mostly,much,must,my,myself
n	n,name,namely,nd,near,nearly,necessary,need,needs,neither,never,nevertheless new,next,nine,no,nobody,non,none,no one,nor,normally,not,nothing,novel,now nowhere
o	o,obviously,of,off,often,oh,ok,okay,old,on,once,one,ones,only,onto,or,other,others otherwise,ought,our,ours,ourselves,out,outside,over,overall,own
p	p,particular,particularly,per,perhaps,placed,please,plus,possible,presumably probably,provides
q	q,que,quite,qv

r	r,rather,rd,re,really,reasonably,regarding,regardless,regards,relatively,respectively right
s	s,said,same,saw,say,saying,says,second,secondly,see,seeing,seem,seemed,seeming seems,seen,self,selves,sensible,sent,serious,seriously,seven,several,shall,she,should shouldn't,since,six,so,some,somebody,somehow,someone,something,sometime sometimes,somewhat,somewhere,soon,specified,specify,specifying,still,sub,such sup,sure
t	t,t's,take,taken,tell,tends,th,than,thank,thanks,thanx,that,that's,that's,the,their,theirs them,themselves,then,thence,there,there's,thereafter,thereby,therefore,therein theres,thereupon,these,they,they'd,they'll,they're,they've,think,third,this,thorough thoroughly,those,though,three,through,throughout,thru,thus,to,together,too,took toward,towards,tried,tries,truly,try,trying,twice,two
u	u,un,under,unfortunately,unless,unlikely,until,unto,up,upon,us,use,used,useful uses,using,usually,uucp
v	v,value,various,very,via,viz,vs
w	w,want,wants,was,wasn't,way,we,we'd,we'll,we're,we've,welcome,well,went,were weren't,what,what's,whatever,when,whence,whenever,where,where's,whereafter whereas,whereby,wherein,whereupon,wherever,whether,which,while,whither,who who's,whoever,whole,whom,whose,why,will,willing,wish,with,within,without,won't wonder,would,would,wouldn't
x	x
y	y,yes,yet,you,you'd,you'll,you're,you've,your,yours,yourself,yourselves
z	z,zero

表 21:LSA 主題詞彙

主題	主題詞彙
Mathematical logic	logic, axiom, firstord, theori, proof, hilbert, godel, mathemat, zermelo, cardin, recurs, theorem, der, intuitionist, axiomat, formal, ed, edit, foundat, reprint
Set theory	theori, cardin, axiom, cantor, member, zermelo, set, zfc, axiomat, determinaci, edit, paradox, mathemat, zf, udayana, antinomi, invari, infin, infinit, membership
Number theory	prime, diophantin, theori, integ, equat, fermat, edit, arithmet, conjectur, quadrat, theorem, euler, indetermin, number, proof, pell, legendr, ibn, gauss, isbn
Graph theory	graph, subgraph, edg, edit, vertex, theori, vertic, color, matrix, adjac, incid, cayley, draw, weight, minor, harari, graphtheoret, induc, conjectur, enumer

Type theory	russel, theori, axiom, type, quin, st, frege, edit, commentari, heijenoort, van, wiki, ramifi, cf, proposit, croppedsvg, church, variabl, introduct, logic
Category theory	morphism, functor, categori, theori, arrow, higherdimension, lane, topolog, algebra, isomorph, transform, eilenberg, commut, mathemat, edit, press, categor, topo, covari, mac
Numerical analysis	numer, equat, differenti, edit, iter, interpol, babylonian, error, decomposit, analysi, x1, x3, approxim, linear, method, lemonad, wellpos, discret, fx, sqrt2
Information theory	entropi, channel, mutual, shannon, theori, px, isbn, inform, compress, edit, bit, probabl, transmiss, code, capac, york, joint, ixi, hxi, transmit
Automata theory	automaton, automata, finit, transit, symbol, recogniz, word, nondeterminist, input, accept, q0, nondeterministicdeterminist, state, jump, languag, qn, qi, infinit, determinist, recogn
Computability theory	recurs, ture, enumer, degre, manyon, halt, post, priorit, reduc, oracl, automorph, theori, arithmet, set, noncomput, soar, truthtabl, reduct, theorem, infinit
Computational complexity theory	ture, np, problem, class, npcomplet, complex, reduct, fn, tn, decis, input, machin, integ, graph, polynomi, instanc, nondeterminist, bound, solv, determinist
Quantum computing theory	quantum, qubit, decoher, classic, probabl, threebit, grover, trap, shor, bqp, superposit, polynomi, spin, gate, algorithm, error, coeffici, string, speedup, suspect
Analysis of algorithms	ns, runtim, loop, run, growth, algorithm, size, step, rate, consum, t1t7, constant, on2, nanosecond, notat, worstcas, input, estim, outer, asymptot
Algorithm design	algorithm, sort, pli, kruskal, depthfirst, breadthfirst, editnot, decor, brook, dijkstra, quicksort, design, internet, packet, steven, son, algol, cobol, repositori, wiley
Computational geometry	geometri, queri, geometr, polygon, point, closest, computeraid, hull, space, search, convex, problem, curv, smallest, combinatori, dynam, partit, distanc, aircraft, triangul
Parsers	parser, pars, grammar, contextfre, topdown, token, lr, leftmost, lexic, rightmost, ll, dog, bottomup, linguist, sentenc, ambigu, bite, accommod, languag, syntact
Interpreters	compil, interpret, bytecod, ast, execut, code, justintim, runtim, jit, nativ, disadvantag, sourc, run, tree, translat, syntax, convert,

	card, intermedi, overhead
Procedural programming	procedur, imper, program, bn, b1, editcomparison, modular, subroutin, objectori, execut, showsolv, variabl, languag, argument, comparison, function, invoc, style, reus, paradigm
Object-oriented programming	oop, objectori, inherit, pattern, simula, subtyp, object, smalltalk, lisp, vbnet, oo, decoupl, class, reusabl, languag, polymorph, program, objectrel, rdbmss, prototypebas
Functional programming	imper, nonstrict, evalu, fibonacci, function, recurs, lambda, higherord, pure, languag, lazi, firstclass, program, argument, haskel, style, side, monad, fibrecurr, lisp
Logic programming	prolog, logic, claus, negat, kowalski, colmerau, predic, theoremprov, horn, planner, declar, failur, concurr, procedur, program, edinburgh, bn, b1, abduct, subgoal
Programming paradigms	paradigm, languag, multiparadigm, oz, objectori, efficaci, program, assembl, procedur, programm, support, oop, disallow, advoc, subroutin, smalltalk, pascal, algol, declar, lowlevel
Multiprocessing	multiprocess, mimd, processor, tightlycoupl, simd, misd, looselycoupl, instruct, cpu, smp, thread, stream, symmetri, multipl, multiprocessor, singl, cluster, xeon, sisd, opteron
Grid computing	grid, middlewar, infrastructur, saa, escienc, supercomput, resourc, servic, america, project, scaveng, ege, boinc, node, network, market, european, distribut, volunt, latin
Concurrency control	transact, serializ, concurr, abort, recover, ss2pl, commit, schedul, databas, distribut, optimist, recoveri, deadlock, lock, violat, global, correct, control, twophas, disappear
Requirements analysis	stakehold, elicit, requir, prototyp, interview, busi, session, nonfunct, analyst, user, jrd, crossfunct, analysi, document, mission, editstakehold, editrequir, endus, feasibl, engin
Software design	softwar, design, modul, compon, editdesign, stepwis, languag, architectur, plan, partit, usabl, packag, hierarchi, pattern, uml, decompos, horizont, concept, structur, user
Computer programming	punch, card, languag, programm, program, debug, readabl, instruct, highlevel, tabul, plugboard, loom, code, sourc, craft, lowlevel, debat, compil, assembl, invent
Formal methods	formal, proof, semant, lightweight, checker, vdm, backu, propon, notat, verifi, verif, critic, undetect, undertaken, postcondit, highinteg, csp, autom, correct, specif
Software testing	test, tester, box, defect, softwar, nonfunct, regress, assur, white,

	coverag, fault, team, sqa, qualiti, load, code, verifi, black, certif, target
Software development process	spiral, waterfal, risk, softwar, phase, agil, iso, cmmi, mainten, lifecycl, deploy, project, increment, document, iter, model, team, formal, test, develop
Computer architecture	isa, architectur, microarchitectur, instruct, processor, cpu, cach, latenc, clock, pin, uisa, macroarchitectur, consumpt, brake, hardwar, regist, editcomput, organ, brook, memori
Computer organization	pipelin, microarchitectur, instruct, cach, cpu, processor, stall, regist, multithread, isa, execut, risc, semiconductor, superscalar, fetch, cisc, architectur, branch, circuitri, decod
Operating systems	kernel, unix, os, window, interrupt, bsd, file, oper, multitask, mac, gnulinux, driver, mode, linux, appl, devic, unixlik, server, protect, run
Computer audio	music, improvis, composit, compos, musician, score, computergener, xenaki, koenig, synthesi, live, style, sound, midi, ircam, csirac, computeraid, soundtrack, omax, mozart
Routing	rout, node, path, destin, protocol, network, router, isp, linkstat, vector, distanc, editrout, distancevector, autonom, latenc, tabl, neighbor, ms, hop, advertis
Network topology	topolog, node, star, network, pointtopoint, hub, bu, transmiss, mesh, ring, cabl, physic, central, connect, peripher, endpoint, spoke, medium, layout, signal
Cryptography	cipher, cryptographi, encrypt, cryptograph, cryptosystem, attack, cryptanalysi, secur, publickey, key, plaintext, secret, hash, messag, rsa, decrypt, export, signatur, symmetrickey, diffi
Database management systems	dbm, databas, codasyl, data, attribut, sql, dbmss, ingr, codd, manag, queri, record, user, store, navig, file, im, access, transact, updat
Relational databases	tupl, attribut, databas, join, referenc, sql, relat, key, tabl, watermark, foreign, relvar, codd, constraint, queri, uniqu, store, normal, indic, surrog
SQL	sql, null, queri, row, claus, mytabl, tabl, sqlpsm, column, isoiec, vendor, databas, join, xml, transact, threevalu, statement, dbm, standard, sqlcli
Transactions	transact, transactionprocess, deadlock, commit, databas, restor, cic, portion, cancel, rollback, bank, roll, backup, rollforward, n1, jta, debit, acid, fail, compens

Database indexes	index, nonclust, row, tabl, indic, bitmap, cluster, column, emailaddress, pointer, databas, sql, key, block, file, reverseemailaddress, lastnam, editindex, dens, btree
Data mining	mine, data, miner, geograph, pattern, spatial, discoveri, datamin, privaci, custom, subjectbas, learnt, ethic, surveil, knowledg, email, associ, tap, supermarket, spss
Automated reasoning	autom, proof, theorem, prover, reason, light, principia, whitehead, mathematica, russel, newel, shaw, logic, prove, deduct, tptp, suttner, sutcliff, nqthm, lt
Computational linguistics	linguist, subdivis, translat, grammar, languag, tagger, deal, scientist, intellig, spoken, interdisciplinari, cognit, speech, devot, artifici, expert, lexicon, partofspeech, anthropologist, semiot
Computer vision	vision, imag, scene, vehicl, 3d, camera, sensor, autonom, medic, missil, motion, nois, detect, recognit, extract, robot, rover, signal, restor, task
Evolutionary computation	evolutionari, evolut, genet, rechenberg, ingo, recombini, optimis, mutat, surviv, schwefel, hanspaul, editevolutionari, popul, fit, nineti, artifici, algorithm, swarm, selforgan, chanc
Expert systems	expert, rule, confid, conclus, rulebas, knowledg, frog, infer, fritz, green, tax, claus, expertis, probabl, conclud, segment, confirm, cf, hop, certainti
Machine learning	learn, learner, neural, reinforc, cluster, train, netflix, network, unsupervis, prize, algorithm, confer, supervis, machin, input, observ, bayesian, acycl, svm, reward
Natural language processing	nlp, speech, evalu, sentenc, word, capit, tagger, linguist, text, task, corpora, chunk, campaign, learn, po, languag, recognit, handwritten, statist, rule
Robotics	robot, actuat, walk, motor, facial, leg, kinemat, wheel, human, unemploy, effector, apek, humanoid, muscl, fiction, balanc, snake, gripper, job, sensor
Visualization (computer graphics)	visual, scientif, draw, graph, graphic, analyt, ptolemi, diagram, anim, transfer, knowledg, editvisu, imag, chart, concentr, tuft, napoleon, minard, isosurfac, geographia
Computer animation	anim, movi, frame, render, film, facial, pyramid, avar, photorealist, motion, 3d, wall, charact, skelet, keyfram, computeranim, cgi, 2d, imag, goat
Image processing	imag, color, optic, recognit, composit, photograph,

	nonphotorealist, align, morph, bright, affin, departur, enlarg, registr, twodimension, photo, hash, lane, warn, raw
Computer accessibility	impair, disabl, access, hear, msaa, web, keyboard, sever, dexter, font, feedback, microsoft, peopl, reader, blind, illiteraci, caption, assessor, assess, motor
User interfaces	interfac, user, modal, hmi, mode, command, input, output, usabl, mmi, humanmachin, graphic, human, humancomput, gestur, gui, automobil, ergonom, batch, vehicl
Wearable computing	wearabl, eye, wrist, chord, brick, display, keyboard, radio, warwick, watch, xybernaut, symposium, panason, palmtop, mann, bankruptci, privat, wristwatch, mellon, carnegi
Ubiquitous computing	ubiquit, weiser, devic, pervas, mem, ubicomp, castel, everyday, parc, ambient, humancomput, tab, display, smart, wearabl, chief, dust, paradigm, food, network
Virtual reality	virtual, realiti, vr, immers, 3d, stori, fiction, recreat, therapi, therapist, laurel, lanier, heritag, glove, exposur, brenda, archaeolog, world, museum, movi
Artificial life	alif, life, dna, artifici, synthet, creatur, wet, simul, evolutionari, net, biolog, leg, agent, soft, tierra, hardwarebas, breve, biochemicalbas, aliv, organ
Bioinformatics	genom, protein, gene, bioinformat, sequenc, dna, biolog, microarray, proteinprotein, homolog, align, molecular, nucleotid, highthroughput, predict, cancer, acid, evolutionari, motif, dock
Cognitive science	cognit, brain, mind, psycholog, linguist, neurosci, intellig, percept, behavior, studi, spatial, neuron, human, neural, blood, innat, resolut, mental, acquir, scienc
Computational chemistry	chemistri, molecular, molecul, energi, initio, ab, semiempir, orbit, quantum, schroding, nuclei, hartre, chemic, stationari, electron, approxim, empir, densiti, wave, calcul
Computational neuroscience	neuron, neurosci, synaps, neural, brain, synapt, biophys, axon, biolog, dendrit, sensori, network, lapicqu, huxley, hodgkin, discrimin, hypotesi, plastic, oscil, psycholog
Computational physics	physic, plasma, solid, equat, fluid, eigenvalu, quantum, lattic, solv, gaug, astrophys, initio, condens, tune, eigenvector, differenti, ab, subdisciplin, chemist, teori
Numerical algorithms	numer, equat, differenti, interpol, error, iter, decomposit, babylonian, wellpos, approxim, linear, discret, extrapol, finit,

	guess, km, analysi, method, solut, fx
Symbolic mathematics	symbol, checker, algebra, manipul, simplif, proof, symbolicnumer, prover, noncomput, computerassist, substitut, express, quantiti, differenti, oppos, equat, theorem, approxim, mathemat, autom

