由史料中探勘職官年表:

以清聖祖實錄為例

闕伯丞、宋邡熏、沈錳坤、蔡銘峰

國立政治大學

資訊科學系

摘要

隨著史料數位化的發展，大量的數位史籍資料庫不僅提供歷史研究更方便快速的檢索工具，而且也提供了結合歷史與資訊科技跨領域研究的契機。資訊科技可以由數位史料中，發展歷史研究的工具，以有助於史學學者的研究。

本論文的研究目的是運用資料探勘技術，根據職官名稱，由史料中識別職官的人名與任期，以自動產生職官年表。我們研究利用頻繁區間項目集合探勘演算法來探勘常伴隨職官名稱出現之人名，以識別職官人名與任期。

進而根據史料中的仕途進退關連分析，建立社群網絡。由建立的社群網絡中，探勘分析歷史人物的角色與派系關係，並以視覺化分析顯示。

針對人名識別與職官任職資訊，針對仕途進退關連分析，我們由職官年表中探勘循序共現樣式，以探勘出官職陞貶的共現關係。接著根據所探勘出的循序共現樣式，建立職官的社會網絡。由社會網絡中，利用網路中心性與社群探勘演算法，分析權臣與派系。

本論文以《清聖祖實錄》為文本，實驗研發職官社會網絡探勘的工具，以提供歷史學者研究分析的工具。

關鍵詞：職官年表、資料探勘、清聖祖實錄、社群網絡

Mining Official Chronology from Historical Documents: By Example of Veritable Records of the Qing Kangsi

Bo-Cheng Que, Fang-Shiuh Song, Man-Kwan Shan, Ming-Feng Tsai

Department of Computer Science
Natioinal Chengchi University

Abstract

This research investigates the data mining approach to discover the official chronology from historical documents. We propose the named-entity extraction algorithm based on the frequent itemset mining with period to discover the names of chief officials and official chronology. Then we develop the algorithms to discover the promotion co-occurrence patterns from official chronologies. Then the social network is constructed based on the discovered promotion co-occurrence patterns, the event co-occurrence patterns and the discovered kinship mining from historical documents. The Veritable Records of the Qing Kangsi is taken as an example for experiments and the visualization analysis to demonstrate the proposed methods for historical research.

Keywords：Official Chronology, Data Mining, Veritable Records of Qing, Social Network

一、前言

自從 1984 年中央研究院的「史籍自動化計劃」迄今，中研院資訊所及史語所共同研發，已經先後建置完成《史記》、《漢書》、《後漢書》和《三國志》前四史的數位化，後來擴充至二十五史，更延伸至包括經、史、子、集的「漢籍電子文獻資料庫」。

古籍數位化在史學上，提供了豐富的數位史料資料庫。歷史學者在做歷史研究時，傳統的作法是在浩瀚的紙本史料中，皓首窮經地以人工的方式考證、分析、比較、歸納。史籍數位化後，不僅提供歷史研究更方便快速的檢索工具，而且也提供了歷史與資訊科技跨領域研究的契機。資訊科技可以由數位史料中，發展歷史研究的工具，以有助於歷史學者進行史學的研究。

資訊科技中，Information Retrieval 可以協助史料的檢索、Information Extraction 可以協助人名、地名等命名實體的辨別、Data Mining 可以協助探勘歷史事件的樣式，隨著 Social Network Mining 技術的發展，Social Network Mining 技術也可以協助歷史學者研究當時的人脈網絡、分析潛在的特徵。

人是構成歷史的主要因素，同時也是歷史的主體和動力，所有事件的背後都有人的存在。對於政治舞台上的歷史人物，除了其氏族背景之外，其仕途歷程扮演重要的角色。藉由仕途擔任的職官、任期以及其職官的變化，刻劃其政治舞台的發展。

舉例而言，錢實甫的《清代職官年表》[1]便是一部紀錄清代順治至宣統時期重要職官資訊的工具書。職官年表依按照時間順序記載重要職官歷任官員之姓名以及擔任的任期。《清代職官年表》主要以當代政治或社會上較具影響力的職官為主，例如大學士、尚書、將軍、總督、巡撫等。故職官年表是以中央職官佔最大部份，提供後人以便對於職官的人事動態之查詢。《清代職官年表》是錢實甫採取人工的方式，同時參考以《清實錄》為主之史料歷經多年心血編制完成。若能藉由資訊技術的輔助來協助產生歷史上不同時代的職官任職資訊，對歷史學者將有莫大的幫助。

職官任職資訊也可應用在社群網絡上。當我們翻開一本本厚重的歷史文獻時，不難發現每位歷史人物彼此之間的關係錯綜複雜，就似一張看似沒有條理但又有脈絡可尋的蜘蛛網。而歷史人物之間的互動、彼此間友好強弱關係正是構成這張網的主要元素，同時這樣的社群網路也正是歷史結構中一個很重要的模式。由歷史人物的社群網絡中，透過 Social Network Mining 的技術，我們可以協助歷史學者探勘分析歷史人物互動之間所形成的社群(Community)，也可以協助歷史學者探勘分析歷史人物在歷史舞台上所扮演的角色。

由史料中做社群網絡探勘，最基本的關鍵是必須先建立其社群網絡。而建立社群網絡，首要是人名的識別。由史料識別人名是重要且深具挑戰的研究議題，尤其是清代滿人的姓名。

例如慈禧太后姓葉赫那拉、名杏貞。又如乾隆期間的首席軍機大臣阿桂，姓章佳，在史書中卻皆以阿桂稱謂。即使辨識出人名，如何由史料中辨別任期資訊、如何分析歷史人物之間的互動關係，都是重要的研究議題。

因此本論文提出利用資料探勘技術，協助史學家由史料中探勘職官任職資訊，包括職官的人名與任期資訊。並提出職官任職資訊的應用，根據「一人得道，雞犬升天」的仕途進退關連分析，建立社群網絡。由建立的社群網絡中，探勘分析歷史人物的角色與社群關係，以提供歷史學者分析歷史人物仕途進退的工具。

二、文獻回顧

與我們的研究相關的史料數位典藏，除了中央研究院開發的「漢籍電子文獻資料庫」之外，還包括台大項潔教授主持的的「台灣歷史數位圖書館」(Taiwan History Digital Library, THDL, http://thdl.ntu.edu.tw/)[14]。THDL 為集合台灣史料之資料庫。主要內容包含 1388 到 1911 年間朝廷與台灣有關的「明清檔案」、1789 到 1895 年間地方政府的「淡新檔案」與「古契書」三大部分。明清檔案蒐集了明清時期與台灣有關之的官方行政公文。檔案內容來自於「明實錄」、「清實錄」、奏摺、軍機處檔案、諭旨、內閣大庫、 月摺檔、地方志等。「淡新檔案」則是清乾隆至光緒年間淡水廳、臺北府及新竹縣的行政與司法檔案。THDL 提供除了提供全文檢索的功能，最有特色的是結合歷史與資訊技術之研究，提供人名、地名與時間識別、清代官職表等研究參考工具。THDL 的人名地名識別是利用其所提出的詞夾子演算法。

目前有關中文命名實體識別 (Chinese Named Entity Recognition)的相關研究中，針對的文本主要為白話文，較少針對古文的處理。張敏[8]的研究分析了古文與白話文的異同，在大規模語料庫上，對古文進行統計分析。此研究主要針對古文進行統計分析，比較其與白話文的差異，但並未針對人命分析兩者間的區別。

在命名實體識別的研究裡，較常見的方法多為機器學習 (Machine Learning)的機率模型及結合機器學習和語料庫規則的方法。前者從文本中，統計姓氏及名字的用字機率及上下文等可用訊息，再配合相關的機率模型，進行人名的識別。後者則是在使用模型識別人名後，對前階段識別的結果分析，套用建立的語料庫規則，進行對結果的篩選或校正。例如中國的毛婷婷採用 Support Vector Machine 和機率統計模型結合的方式，進行中國人名的識別[4]。李中國與劉穎從訓練資料中，根據人名左右邊界的詞語及人名用字機率作為識別依據。台大的張尚斌之詞夾子演算法(Word-Clip Algorithm)，則是觀察 THDL 中的史料，發現經常出現具有特定樣板的文字[6]。例如：臣福康安跪奏、臣李侍堯跪奏，在「臣…跪奏」之間通常會是個人名。類似「臣…跪奏」這樣可以夾中人名的詞語稱之為詞夾子。在使用前，必須先給予樣本

人名。接著依照給予的樣本人名，找出與其相對應的詞夾子。然後，再使用這些詞夾子夾出更多的人名，如此反覆進行。

由古籍中探勘社群網絡的研究並不多。我們曾以清實錄中的乾隆時期為例，以詞夾子演算法為基礎，參考職官清單、地名、地名清單，提出改進的方法。並根據史料中人名在事件中的共現關係，建立官員之間的社會網絡，且從中探勘權臣及權力變化的消長[9, 14]。台灣大學項潔教授指導的碩士班學生廖儁凡於 2010 完成的碩士論文中，研究國古典小說中的社群網絡，由儒林外史中，利用詞夾子演算法找出出現的角色。接著根據角色間的對話關係，來建立社群網絡[10]。

關於非古籍社群網絡的建立，國內外學者也有相關的研究。現有的研究包括由 Enron 案中涉案人員之間的 email 往返建立社群網絡[18]、由國際學術會議與會學者及 DBLP 的 Co-authorship 建立社群網絡[17]、由電影畫面中角色的共現關係建立社群網絡。亞洲微軟研究院所開發的人立方關係搜索(http://renlifang.msra.cn)則是由中文網頁中自動擷取人名、地名、機構名稱及人名之間的關係，建立社群網絡，並提供六度分離搜尋的功能[27]。

三、人名識別與任職資訊探勘

本研究主要由史料中，根據官職名稱，探勘識別人名與職官任職資訊，自動產生職官年表。所謂的職官任職資訊是以人名為單位，條列此人歷任的職官及其任期。因此職官任職資訊包括人名、職官名稱、任期。由史料中擷取出職官的任職資訊並不容易。錢實甫即歷經數年的時間，整理清代古籍史料，才於 1980 年完成四大冊的清代職官年表。職官年表事實上就是職官任職資訊。只是職官年表是以職官為單位，條列此職官歷任的人名及其任期，表一所示即為康熙年間歷任湖廣總督的職官年表。

針對職官的任職資訊，一種作法是由史料中透過陞貶之動詞，配合人名識別來擷取出職官的陞貶資訊。以清朝為例，故宮博物院的清代檔案數位典藏計劃中，曾針對清代人名權威檔案，分析在史料中常見的職官進退動詞，就有陞、升、擢、拔、授、遷、晉、襲、轉、署、降、調、調補、降調、黜、貶、護理、起、復、攝、封、贈，以用、以...用、儘先...補用、儘先...推補、儘先...選用、以...委用、委、委用、委署、記名、充、兼、權等三十多種不同的用法。而且這些詞彙有可能因為不同的時期或者修纂官執筆風格之差異，有時甚至要對前後文進一步求證才能夠正確的判斷。甚至還有 10%至 20%的職官任職資訊，不易由前後文的解讀獲得解決。因此，故宮目前一萬多位的清代權威人名檔案是透過學者專家及著錄人員以人工的方式來完成。

表一：康熙年間歷任湖廣總督的職官年表。

任職人名	上任時間	卸任時間
張長庚	順治十八年二月	康熙七年十月
蔡毓榮	康熙九年四月	康熙二十一年正月
董衛國	康熙二十一年正月	康熙二十二年十一月
徐國相	康熙二十三年正月	康熙二十七年三月
丁思孔	康熙二十七年九月	康熙三十三年四月
吳琠	康熙三十三年四月	康熙三十五年六月
李輝祖	康熙三十五年七月	康熙三十八年六月
郭琇	康熙三十八年六月	康熙四十二年正月
喻成龍	康熙四十二年四月	康熙四十四年八月
石文晟	康熙四十四年八月	康熙四十六年五月
郭世隆	康熙四十六年六月	康熙四十九年十月
鄂海	康熙四十九年十月	康熙五十二年四月
額倫特	康熙五十二年四月	康熙五十五年三月
滿丕	康熙五十五年三月	康熙六十一年十一月

針對史料中的人名識別，台大項潔教授與張尚斌曾提出詞夾子演算法由史料中識別人名。詞夾子演算法利用人名前後經常出現的特定樣板文字來判斷人名。例如：湖廣總督丁思孔疏言，「湖廣總督」是人名的左詞夾子，「疏言」是人名的右詞夾子，在「湖廣總督...疏言」之間通常是人名。針對擔任職官的人名，職官名稱往往是常見的詞夾子。

表二為「清實錄」中出現湖廣總督的部分例句。第 1 句透過「湖廣總督」與「疏言」的詞夾子可以判斷「丁思孔」是人名。第 2 句透過「湖廣總督」的左詞夾子與頓號「、」的右詞夾子，也可以判斷「筆帖式克錫類」為人名。但第 3 句的例句，詞夾子演算法可能就無法正確地找出人名「宗室德沛」。因為「鎮國將軍」並不常出現在文本中，因此詞夾子演算法可能會誤判人名為「鎮國將軍宗室德沛」。第 4 句的人名「郭琇」，透過詞夾子可能會誤判為「郭琇等」。而第 5 句以左詞夾子「湖廣總督」與右詞夾子句號「。」可能會誤判「宜駐荊州」為人名。第六句的「審明」即使以人工判斷，亦無法確定是否為人名。換句話說，單單以詞夾子透過人名的前後文、搭配職官名稱，可能無法判斷出表二的例句 3, 4, 5 與 6 的人名。針對第 7 句，即使以詞夾子演算法準確地識別出人名為「董衛國」，仍必須判斷董衛國調任為湖廣總督。

針對編年體史料，本論文提出利用官職名稱，以資料探勘技術來探勘出官員任職資訊。在編年體史料中，大部份的史事會同時記載關於該事件的人員、職官、時間或地點。然而，若是以個別職官為單位，將包含此職官名稱的所有句子依序從史料中擷取出來，並按照時間排序來觀察，將可發現史料中職官名稱與其人名之間存在特殊的關係。例如表三所示即為《清

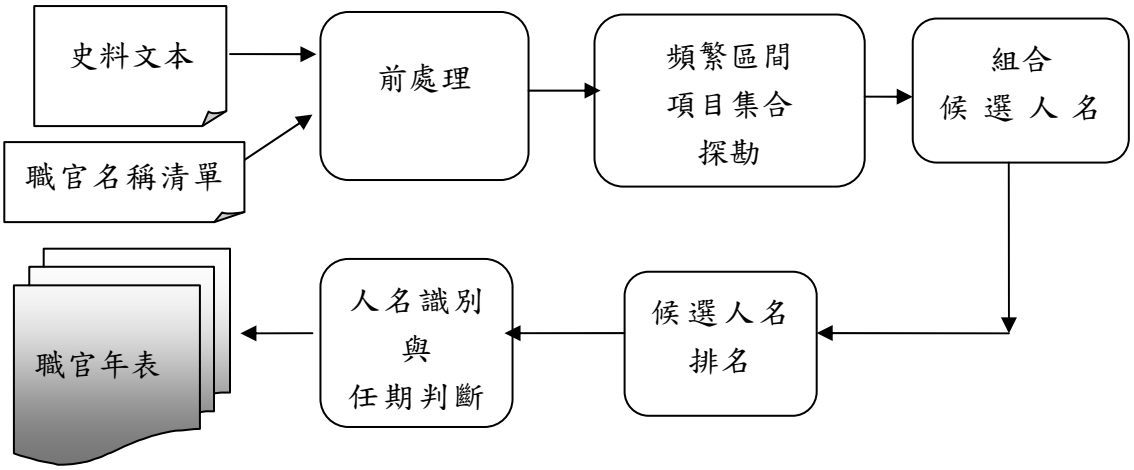
聖祖實錄》中康熙元年至康熙十年出現湖廣總督的片段(康熙九年更設為四川湖廣總督，康熙十三年四川另設總督，川湖總督又改稱湖廣總督)。其中，前 10 筆資料每當提及湖廣總督時，最常出現之人名為張長庚。從第 11 筆至第 16 筆中最常伴隨湖廣總督一起出現之人名為蔡毓榮。因此，我們可以猜測康熙元年二月至康熙五年之間，湖廣總督為張長庚。而康熙九年之後，則是蔡毓榮擔任湖廣總督。

表二： 《清聖祖實錄》中出現湖廣總督的部分例句。

1	兵部議覆、湖廣總督丁思孔疏言、武昌兵變…
2	先是、武昌兵變、湖廣總督筆帖式克錫類、探知賊有犯荊州之意自城溝內遁出…
3	…湖廣總督鎮國將軍宗室德沛奏、臣年五十。
4	九卿等議覆、湖廣總督郭琇等、遵旨詳議科事宜四疏嗣後直隸各省鄉試在京三品以上…
5	…吏部議覆、原任左都御史魏裔介疏言、湖廣總督宜駐荊州。
6	前經湖廣總督審明、收販船戶腳私七千二百餘包。
7	調江西總督董衛國、為湖廣總督。

表三： 《清聖祖實錄》中康熙元年至康熙十年出現湖廣總督的片段。

1	康熙元年二月	**湖廣總督**張長庚疏報、東鄉土司覃繩武、殺賊歸順。
2	康熙二年四月	**湖廣總督**張長庚疏報、提督董學禮、同各鎮將等、由魚剌坡、進抵長坪地方
3	康熙二年五月	**湖廣總督**張長庚疏報、忠建、高羅、木冊、三處土司、繳印投誠。
4	康熙三年元月	**湖廣總督**張長庚疏報、偽部院毛壽登、始為偽永曆朱由榔所倚恃、繼為眾逆寇所推重、今革面來歸、請敕優敘。
5	康熙三年二月	**湖廣總督**張長庚疏報、西山巨逆馬騰雲、黨守素、塔天寶等、率眾歸誠。
6	康熙三年八月	予故原任太子太保內弘文院大學士呂宮、原任**湖廣總督**太子太保兵部尚書李蔭祖、各祭葬如例。
7	康熙三年八月	**湖廣總督**張長庚疏報、靖西將軍都統穆裏瑪、定西將軍都統圖海等、率禁旅與三省綠旗兵、合剿西山巨逆郝搖旗、劉汝魁等、業經授首。
8	康熙三年十一月	兵部議覆、**湖廣總督**張長庚疏言、房縣、保康、竹山、竹溪、與興山巴歸一帶、皆楚省邊險之地。
9	康熙四年五月	戶部議覆、**湖廣總督**張長庚疏言、歸州、巴東、長陽、興山、房縣、保康、竹溪、竹山等州縣、久為巨逆盤踞、人民逃竄。
10	康熙五年十月	兵部議覆、**湖廣總督**張長庚疏言、殉難總兵官徐勇子襲阿思哈尼哈番職徐自貴、具呈情願效力、相應推用。
11	康熙七年十月	裁**湖廣總督**缺.
12	康熙八年九月	原任**湖廣總督**張長庚、著以原品隨旗行走。
13	康熙九年三月	復設四川**湖廣總督**一員、福建總督一員.
14	康熙九年四月	命吏部左侍郎管右侍郎事蔡毓榮、為四川**湖廣總督**
15	康熙十年六月	四川**湖廣總督**蔡毓榮疏言、蜀省有可耕之田、而無耕田之民。
16	康熙十年七月	兵部議覆、四川**湖廣總督**蔡毓榮疏言、川省沖要各營將弁缺出、照沿邊題補之例、令總督提督題補。



圖一：官員任職資訊探勘之流程。

因此，根據文本中職官與人名之間的關係，我們利用**頻繁區間(Frequent Period)**之概念，分析伴隨職官名稱與擔任之人名。其步驟如圖一所示。

（一） 前處理

首先以職官為單位，針對每個官職，由文本中將所有出現該職官名之句子根據年月依序擷取。接著針對每個句子做 Bigram 的處理。換句話說，將句子以兩個字一組的方式進行切割。每個句子表示為 Bigram 所組成的集合。之所以做 Bigram 的處理，因為史料中的人名最短為兩個字。經過 Preprocessing 的處理後，每個職官就對應到一個有時間先後順序的交易資料庫(Transaction Database)，其中每個 Bigram 視為一個項目(Item)，每個句子因此對應到一筆交易(Transaction)，也就是由 Bigram 所組成的 Itemset(項目集合)。

Time	Transaction ID	Transaction
January (Period 1)	T01	A, C, F
	T02	A, D, F
	T03	C, F
	T04	A, D
February (Period 2)	T05	B, C, D, F
	T06	C, F
	T07	B, C, D
	T08	A, B
March (Period 3)	T09	C, E, F
	T10	A, B, F
	T11	E, F
	T12	B
April (Period 4)	T13	E
	T14	A, B, E
	T15	A, B
	T16	A, B

Frequent Itemset	Frequent Period	Support
A, D	[1, 1]	2/4
A, F	[1, 1]	2/4
C, F	[1, 3]	5/12
B, C	[2, 2]	2/4
B, D	[2, 2]	2/4
C, D	[2, 2]	2/4
E, F	[3, 3]	2/4
A, B	[4, 4]	3/4
B, C, D	[2, 2]	2/4

圖二：TWAIN 之範例[15]。

（二） 頻繁區間項目集合探勘

為了探勘職官在時間區間內頻繁出現的候選人名，我們探勘區間內密集出現的頻繁項目集合(Frequent Itemset) [15]。原本關聯規則探勘(Association Rule Mining)中交易資料庫(Transaction Database)中之交易資料並沒有時間先後的關係，而且項目集合的支持度(Support)之定義是以整個交易資料庫總筆數為考量。但頻繁區間項目集合探勘(Frequent Itemset Mining with Period)則考量到時間區間的支持度。圖二所示即為 TWAIN 的範例，此交易資料庫共有 16 筆交易，切分成 4 個區間。若最小支持度(Minimum Support)為 30%，右方即為頻繁區間項目集合。例如{C, F}在 Period 1 到 3 之間為頻繁項目集合，Support 為 5/12。

我們以 J. W. Huang et al.所提出之 TWAIN (TWo-end AssocIation Miner)演算法為基礎。TWAIN 演算法之主要精神為區間頻繁的探勘，因此必須將資料依時間單位，切成多個區間，之後再運用這些區間資料進行頻繁區間項目集合之探勘。演算法的流程是從第一個 Partition 開始探勘，找出符合最小支持度之項目集合，並且延續至下一回合繼續探勘，每回合增加一個 Partition，下一回合將前一回合符合最小支持度之項目集合與本回合新增 Partition 的項目集合，一同判斷與否符合最小支持度。符合者繼續延續至下一個回合，若不符合則該項目集合便不再延續至下一回合。整個演算法一直執行到沒有可延續至下一回合之項目集合為止。因此頻繁項目集可分為兩種類型，第一種類型係只有在該回合才出現之頻繁項目集合，頻繁時間開始與結束時間都是在該回合；第二種類型係延續使用前一回合之頻繁項目集合，頻繁時間開始記錄為前一回，結束時間記錄為該回合。這兩種類型在計算支持度稍有不同，第一類型為最小支持度乘上該回合之資料筆數，而第二類型則要將最小支持度乘上所有聯集回合的總資料筆數。

此外，該演算法除了使用最小支持度來決定是否可將該回合延續至下一回合之外，還有另一個條件值：最大共同範圍。最大共同範圍之定義是項目集合重疊起來，重疊最早之 Partition 即為開始範圍，最晚之 Partition 為結束範圍。此條件門檻設定之目的在於判斷某些頻繁項目集合是否可以再延續下一回合之 Partition，但若從史料探勘自動產生職官年表的過程中，只需要針對是否符合最小支持度即可，因此沒有再加上最大共同範圍之限制條件。

（三） 組合候選人名

針對前一步驟所探勘出的頻繁區間項目集合，接著我們僅保留完全頻繁項目集合(Closet Frequent Itemset)。一個頻繁項目集合 *S* 稱之為完全頻繁項目集合，如果不存在任何頻繁項目集合 *S*’，使得頻繁項目集合 *S* 的支持度與 *S*’的支持度相同，且 $S \subset S'$。換句話說，如果存在符合以上這兩個條件的頻繁項目集合 *S*’，則 *S*’可以代表 *S*。例如如果頻繁項目集合 *S*={張長}的支持度是 10%，而頻繁項目集合 *S*’={張長, 長庚}的支持度也是 10%，則 *S* 不是完全頻繁項目集合。又如如果頻繁項目集合 *S*={張長, 長庚}的支持度是 15%，而頻繁項目集合 *S*’={張長, 長庚, 庚疏, 疏言}的支持度是 10%，且僅存在 *S*’，使得 $S \subset S'$，則 *S* 是完全頻繁項目集合。

對於每個完全頻繁項目集合，我們檢查其是否可以組合成候選人名。組合的條件是檢查完全頻繁項目集合中的項目，其所對應的 Bigram 是否彼此有字首(Prefix), 字尾(Suffix)的關係。例如若完全頻繁項目集合所對應的Bigram集合為{"那蘭", "性德", "蘭性" }，則此完全頻繁項目集合可以組合為候選人名"那蘭性德"。

（四） 候選人名排名

即使是經過頻繁項目集合探勘後的候選人名，仍然有可能是非人名。例如表四中，"疏報"的區間密度很高，很有可能被誤判為人名。

因此我們提出判斷人名可能性的排名指標，綜合這些指標來排名候選人名為人名的可能性。我們考慮的指標包括：

- 職官變動率：一般來說，在仕途進退中，職官異動之變動率不會太過頻繁。因此，一般來說，非人名在文本中所伴隨出現之職官，其變動率大於人名的職官變動率。
- 平均任期: 一般來說，非人名在文本中所伴隨出現的官職之平均任期，遠小於人名的平均任期。
- 職官品第之平均變化率：一般來說，每個人的職官品地不會有大幅度劇烈的變動。一般來說，因此，非人名在文本中所伴隨出現的官職之前後品第變化，遠大於人名所對應的官職品等之變化。
- 與職官名稱之平均字距：一般來說，非人名在文本中與所伴隨出現的官職名稱，其字距大於人名與對應職官名稱之字距。

因此，若有一候選人名，其綜合指標越高，那麼越可能是人名。此綜合指標將作為產生職官年表時判斷任期的優先順序。

表四：即使是經過頻繁項目集合探勘後的候選人名，仍然有可能是非人名。

十三年七月	江西總督董衛國**疏報**、閩賊侵犯 …
十三年八月	江西總督董衛國**疏報**、閩逆盤踞 …
十三年九月	江西總督董衛國疏言、龍泉…
十三年十月	江西總督董衛國**疏報**、賊犯湖口
十三年十一月	江西總督董衛國**疏報**、臣標右營遊…
十四年九月	安親王岳樂**疏報**、江西總督董衛國標下效用遊擊周志新

（五） 人名識別與任期判斷

經過前述五個步驟的處理後，將產生多個候選人名，每個候選人名都有排名的綜合指標及出現頻率高的頻繁區間中。最後，我們以職官為單位，將這些候選人名，根據其頻繁區間

一一填入此職官年表的時間軸中。在填入的過程中，我們根據候選人名的綜合指標的優先順序。換句話說，若遇到頻繁區間重疊的情形時，以綜合指標排名優先的候選人名為主。

四、職官任職資訊的應用

古代政治的最佳真實寫照，可以透過諺語：「一人得道，雞犬升天」來簡單詮釋。此出自於漢朝王充《論衡．道虛》：「淮南王學道，招會天下有道之人，傾一國之尊，下道術之士，是以道術之士並會淮南，奇方異術，莫不爭出。王遂得道，舉家升天，畜產皆仙，犬吠於天上，雞鳴於雲中。」相傳西漢淮南王劉安，喜好煉丹修道。有一天終於成功地煉成了仙丹，劉安吃了五顆後便成仙升天，而其他丹藥被其飼養的雞犬啄食後，也都跟著升天。後世遂將「一人得道，雞犬升天」暗喻一個人在官場得勢，其他的黨羽也一同沾光。史料中除了描述其家世淵源之外，最常著墨的莫過於其仕途狀況，其間充斥著裙帶與派系關係的攀附或結黨。

因此我們可以根據「一人得道，雞犬升天」的現象，透過職官任職資訊，探勘史料中所刻畫描寫的仕途進退陞貶關連，來建立社群網絡。並由建立的社群網絡中，探勘分析歷史人物的角色與社群關係。

(一) 陞官序列的產生

有個官職任職資訊後，我們根據官職品第的高低、官職權力的高低，來判斷官職的陞職。我們得出每位官員官職異動的陞職情形及異動時間後，將所有官員的官職異動的陞職情形及異動時間，根據時間的先後，組合出官員所形成的陞官序列。陞官序列中每個元素記錄了陞官的官員之人名。因此，如果將《清聖祖實錄》順治十八年元月到康熙六十一年十一月共七百六十六個月，以月份為單位，則可產生一條由七百六十六個集合所組成的陞官序列。每個集合代表對應的那個月陞官的歷史人物。舉例而言，陞官序列<{3,10}{5}{1,11,16}{2,7,9}{11}{1,15}{2,3}{7,5,9} {10}{11,18}>是一個由十個元素(集合)所組成的序列，其中每個集合代表陞官的歷史人物編號。例如第一個月時，編號 3 號與 10 號的歷史人物同時陞官。第二個月時，編號 5 號的歷史人物陞官。第三個月時，編號 1 號、11 號與 16 號的歷史人物同時陞官，以此類推。

陞官與否的判斷，最簡單的作法是根據每個朝代的官制品第。每個朝代都有其官制。例如清代官制將職官分為文官與武官，而職官的大小是以品等來區分，依序從最高品等的正一品、從一品到正九、從九品，共十八個品第。職官品位的大小可做為其重要性的一項參考指標。在本研究中，我們根據職官的品第關係，據以判斷職官的陞貶。

(二) 循序共陞樣式探勘

有了陞官序列，我們接著就可從中進行共陞樣式探勘，希望能夠探勘出職官人物之間存在之共陞現象，進而因此建立人脈網絡。我們修改資料探勘領域中的循序序列樣式探勘(Sequential Pattern Mining)來探勘循序共陞樣式(Sequential Co-occurrence Pattern)。

循序序列樣式探勘主要由多條由集合所形成的序列(Sequence of Itemsets)中，找出經常一起出現並且有相同先後順序的共同子序列。與循序序列探勘的資料來源不同，本研究所欲探勘之循序共陞樣式，其資料來源是一條由集合所形的陞官序列。我們利用資料探勘中 Apriori 的特性，來探勘符合最小支持數(Support Count)的循序共陞樣式[9]。舉例而言，<{1}{7,9}>在陞官序列<{3,10}{5}{1,11,16}{2,7,9}{11}{1,15}{2,3}{7,5,9}{10}{11,18}>中的支持數為 2，也就是說出現了 2 次。以陞官的說法，編號 1 號的歷史人物陞官後，編號 7 號與 9 號的人物也跟著陞官。這個現象出現了 2 次(第一次隔了 1 個月、第二次隔了 2 個月)。如果最小支持量設定為 2，則<{1}{7,9}>是其中一個所探勘出的循序共陞樣式。

傳統的循序序列樣式探勘允許加上 Maximum Gap 的限制。我們同樣可以在循序共陞樣式加上這個限制。換句話說，如果時間相隔愈久，此共陞關係來自於提拔的關係可能愈不顯著。例如<{3,10}{5}{1,11,16}{2,7,9}{11}{1,15}{2,3}{7,5,9}{10}{11,18}>中，如果設定 Maximum Gap 為 1、最小支持數為 2，那麼<{1}{7,9}>的支持數只有 1(第二次的出現相隔了 2 個月，超過了 Maximum Gap 設定為 1 的限制)。

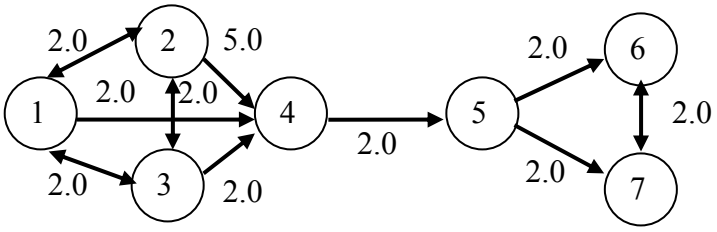
此外，我們也可以加上 Window Constraint 的限制。所謂的 Window Constraint 是指循序共陞樣式中第一個元素(集合)與最後一個元素(集合)的時間間隔。例如<{3,10}{5}{1,11,16}{2,7,9}{11}{1,15}{2,3}{7,5,9}{10}{11,18}>中，如果設定 Window Constraint 為 5，那麼<{1}{7,9}{11}>的支持數是 2。但 Window Constraint 如果設定為 3，<{1}{7,9}{11}>的支持數則為 1 (第二次的出現中，{1}與{11}間隔了 4 個月，超過了 Window Constraint 設定為 3 的限制)。

除了循序共陞樣式，我們也可探勘完全循序共陞樣式。一個樣式 *S* 稱之為封閉樣式，如果不存在任何樣式 *T*，使得樣式 *S* 的支持度與樣式 *T* 相同，且樣式 *S* 是樣式 *T* 則的子序列。換句話說，如果存在符合以上這兩個條件的樣式 *T*，則樣式 *T* 可以代表樣式 *S*。

(三) 社群網絡建立

探勘出循序共陞樣式後，我們將研究根據樣式中的人名、循序共陞樣式的支持度，建立社群網絡。換句話說，如果支持度越高，代表其在社群網絡中的連結越密切。例如：若探勘出循序共陞樣式為支持數為 2 的<{1, 2, 3}{4}{5}{6, 7}>及支持數為 3 的<{2}{4}>，則圖三所示即為根據此樣式所建立的社群網絡。首先，針對前後集合中的成對元素，在社群網絡中建

立有向的連結。舉例而言，支持數為 2 的循序共陞樣式<{1, 2, 3}{4}{5}{6, 7}>將會建立權重為 2 的 1→4, 2→4, 3→4, 4→5, 5→6, 5→7 這些有向連結。支持數為 3 的循序共陞樣式<{2}{4}>則會增加權重 3 至原本已經建立的連結 2→4，因此連結 2→4 的權重總和為 5。其次，同一個集合中的元素彼此之間則會建立雙向的連結。因為相同集合內的元素代表這些人物在同一個月份內同時升官。因此，同例將會建立權重為 2 的 1↔2, 2↔3, 1↔3, 6↔7 等雙向連結。



圖三：社群網絡建立之範例。

(四) 社群網絡探勘與視覺化

建立社群網絡後，接著我們就可利用社群網絡分析的技術來探勘

(1) Community Mining

社群網絡領域已研發包括 N-Clique, N-Clan, K-Core, K-Plex, KL-Algorithm, Modularity, Graph Partitioning, Bridge-Cut, Density-based Clustering, Stochastic Flows 等不同的 Community Mining 之方法。在本研究的實驗中利用 Modularity 演算法。

Modularity 的概念主要是利用一個衡量社群內部連結(Within-Community Edges)比例的指標。換句話說，如果一個社會網絡中，所有的連結都是社群內部的連結，那麼這就是一個最佳的社群偵測。Modularity 的原始定義如下：

$$\frac{\sum_{vw} A_{vw}\delta(c_v, c_w)}{\sum_{vw} A_{vw}}$$

其中，A_{vw} 為社群網絡的相鄰矩陣 A 中的元素。如果節點 v 與節點 w 相連，則 A_{vw} 為 1，反之則為 0。c_v, c_w 分別代表節點 v, 節點 w 群所屬的社群。如果 c_v, c_w 是相同的社群，則$\delta(c_v, c_w)$為 1，反之為 0。

(2) Role Mining

我們分析節點在社會網絡中的特性以及重要性評估，藉此找出人脈網絡中的權臣。節點重要性評估指標值可由 Degree Centrality, Closeness Centrality, Betweeness Centrality, Spectral Centrality 等不同的方法來觀察。本研究利用 Out-Degree Centrality。

Out-Degree Centrality 的是節點分支度之總合值。假設在一個社會網絡中的節點總數為 n，欲計算其中節點 v_i 與其他節點 v_j 相連結數，其中 $\forall j$，$1 \le j \le n$，Out-Degree Centrality 公式計算如下：

$$C_D(v_i) = \frac{\sum_{j=1}^{n} a < v_i, v_j >}{(n-1)}$$

$a < v_i, v_j >$代表的是 v_i與 v_j之間的連結關係。若節點 v_i有連結到節點 v_j，則$a < v_i, v_j >$代表其連結上的權重值，而(*n*-1)為每一個節點最多可以與(*n*-1)個節點相連。

用 Out-Degree Centrality 來分析歷史人物人脈網絡，可以衡量出哪位歷史人物與最多人建立連結，因為若與最多人之間都有關係之建立，亦即代表該人物越重要，也就越有可能為權臣的可能性較高。

(3) 網絡視覺化

我們利用視覺化軟體 NodeXL，將歷史人物之社群網絡以視覺化的方式呈現。並且將 Community Mining, Role Mining 之結果以顏色、大小的視覺化形式顯現。

四、文本與實驗評估

在實驗文本的取材上，本研究採用《清實錄》當中的《清聖祖實錄》作為實驗文本。清代距離現今年代較近，所保留的歷史古籍也較豐富且完整，且錢實甫的《清代職官年表》可供本研究的對照參考。清朝時代重要的史籍主要有《清史稿》與《清實錄》。《清史稿》屬於二十五史，在體裁上是紀傳體，是民國初年北洋政府設館編修清朝的正史。然而篇幅較為簡略，且先後參加的編寫者有一百多人，體制不一以至於存在一些編纂上的錯誤。《清實錄》在體裁上是編年體，內容為皇帝及中央政府為主的官書，專門記載清朝歷代皇帝統治時期的大事紀，包括政治、經濟、文化、軍事、外交及自然現象等，詳盡地記載了有清一代近三百年的用人行政和朝章國故。而清聖祖康熙是清朝歷史上在位時間最長的皇帝，也是中國歷史上在位時間最長的皇帝，故本研究以《清聖祖實錄》作為實驗文本。《清聖祖實錄》共計三百卷，每卷多為記錄二至四個月份的大事紀。《清聖祖實錄》由順治十八年元月份開始一直記錄到康熙六十一年十一月份，共七百六十六個月(其中二十三個閏月)，約三百多萬字。

本研究的職官資訊來源主要是以錢實甫《清代職官年表》上所記載的職官名稱為本，參考維基百科網站所記載的清朝官職表以輔助判斷職官的品位。

針對職官年表中職官的人名識別與任期判斷，本研究的實驗以正二品與從二品的部份職官為例，包括總督、巡撫及布政使，總計六十一個官職，包括：

總督：直隸總督、江南江西總督、山東總督、山西總督、直隸山東河南總督、四川陝西總督、福建總督、湖廣總督、四川總督、廣東廣西總督、雲南總督、雲南貴州總督、漕運總督、河道總督、江南總督、貴州總督、廣西總督、山西陝西總督、陝西總督、浙江總督、河

南總督。

巡撫：直隸巡撫、江寧巡撫、安徽巡撫、鳳陽巡撫、山東巡撫、山西巡撫、河南巡撫、陝西巡撫、延綏巡撫、甘肅巡撫、寧夏巡撫、福建巡撫、浙江巡撫、江西巡撫、南贛巡撫、湖廣巡撫、偏沅巡撫、四川巡撫、廣東巡撫、廣西巡撫、雲南巡撫、貴州巡撫、江蘇巡撫。

布政使：江蘇布政使、安徽布政使、甘肅布政使、湖北布政使、湖南布政使、河南布政使、山東布政使、山西布政使、陝西布政使、廣東布政使、廣西布政使、四川布政使、雲南布政使、貴州布政使、江西布政使、浙江布政使、福建布政使。

事實上，清朝時期正二品與從二品包含數百個官職，其中諸如武官、東官三少、侍郎暫時沒有列入本研究的實驗中。此乃因清朝重文清武，由總督或巡撫統轄數省或一省行政、經濟及軍事，也因此總兵、統領、副將等武官較少出現在史料中。例如正二品的左右翼前鋒統領在《清聖祖實錄》六十一年間分別只出現 8 次、9 次。而太子少師、太子少傅、太子少保東官三少多為兼任加封的榮譽職，實際上多擔任其他要職。而六部左右侍郎在清制中則是滿漢各設一人。易言之，六部中每部都有侍郎四名，在史料中每位皆有出現的可能。因此暫時沒有列入本研究的實驗中。

此外，部分職官或職官名稱在康熙期間屢有變動。有些是職官名稱的變動，有些職官則歷經合併、裁撤與復設。例如康熙元年時分別有廣東總督、廣西總督，但在康熙四年時裁廣西總督，廣東總督改為兩廣總督 (廣東廣西總督)。本研究的處理方式因此將廣東總督、廣東廣西總督視為不同名稱的相同官職，而廣西總督則在康熙四年時被裁撤。

更複雜的官職變化例如陝西總督、四川總督與湖廣總督三者之間。陝西總督在康熙四年改為山陝總督；康熙十一年又改回陝西總督；康熙十九年與四川合併為川陝總督；康熙五十七年改回陝西總督，另設四川總督；康熙六十年:裁陝西總督。康熙元年時，分別有湖廣總督與四川總督；在康熙七年時，裁湖廣總督，由四川總督兼管，改為川湖總督；康熙十三年時，川湖總督改稱湖廣總督，另設四川總督；康熙十九年時，裁四川總督。

本研究的處理方式因此將陝西總督(康熙元年至康熙四年、康熙十一年至康熙十九年)、山陝總督(康熙四年至康熙十一年)、川陝總督(康熙十九年至康熙五十七年)視為不同名稱的相同官職；同樣地，將湖廣總督(康熙元年至康熙七年、康熙十三年至康熙六十一年)、川湖總督(康熙七年至康熙十三年)視為不同名稱的相同官職；四川總督則視為康熙七年被裁撤、康熙五十七年時復設四川總督。

我們以 Information Retrieval 領域常用的精準度(Precision), 召回率(Recall)來評估職官人名識別與任期判斷的效果。我們所提出的系統對於官職 i 的精準度與召回率定義如下：

$$\text{官職}i\text{的精準度} = \frac{\text{系統正確判斷官職}i\text{的任期月數}}{\text{系統判斷官職}i\text{的任期總月數}}$$

$$\text{官職}i\text{的召回率} = \frac{\text{系統正確判斷官職}i\text{的任期月數}}{\text{官職}i\text{的正確任期總月數}}$$

如果系統判斷的任期總月數與正確的任期總月數相同的話，那麼精準度也就等於召回率。但往往系統判斷的月數與標準答案的月數不相同。例如錢實輔的《清代職官年表》中直隸總督歷經八十四個月，但我們的系統由《清聖祖實錄》判斷直隸總督卻只歷經七十九個月。此外，每個職官在康熙朝歷經的月數不同。有的職官歷經七百多月，有的職官只有數十個月。因此，我們也觀察以月數比例加權的加權平均精準度與加權平均召回率。表五所示分別為總督、巡撫、布政使的平均精準度、平均召回率、加權平均精準度、加權平均召回率，其中頻繁區間項目集合探勘的最小支持度設定為 60%。

表五：總督、巡撫、布政使的平均精準度與平均召回率（最小支持度 60%）。

	數量	平均 精準度	平均 召回率	加權平均 精準度	加權平均 召回率
總督	21	0.633	0.604	0.800	0.672
巡撫	23	0.767	0.592	0.905	0,678
布政使	17	0.614	0.211	0.657	0.220
合計	61	0.678	0.490	0.787	0.524

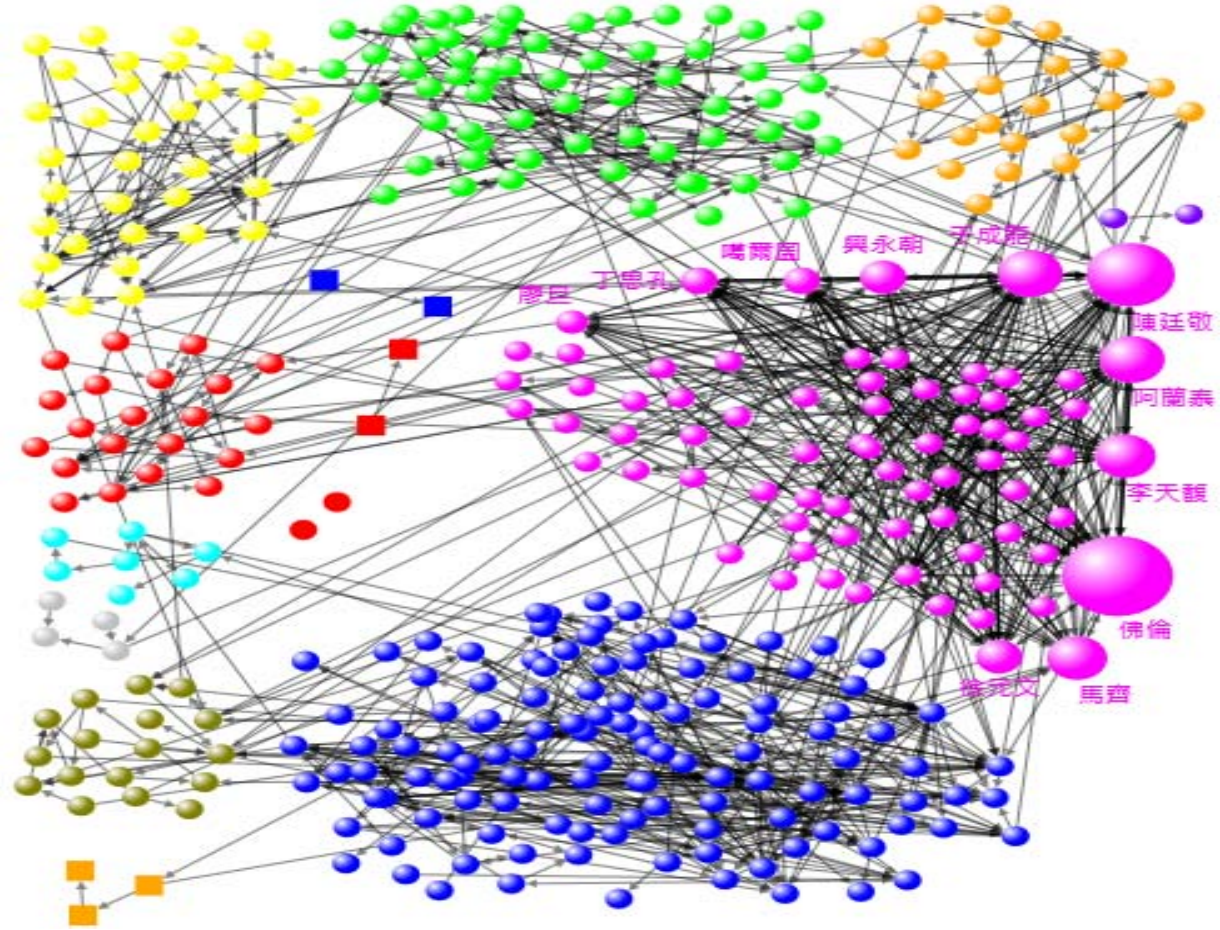
其錯誤原因包括

(一) 官職名稱出現次數太少：以延綏巡撫為例，其精準度與召回率都是 0。在《清聖祖實錄》中第一次出現是順治十八年，最後一次出現在康熙三十三年，但這三十三年間總共只出現七次。也因此本研究「任職期間在史料中出現次數頻繁」的原理對於出現次數太少的職官就不適用。

(二) 史料用字或用詞不一致：例如康熙初年先後擔任山西總督、直隸山東河南總督的白秉真，在康熙元年至康熙八年間共出現六次，但《清聖祖實錄》中有四次是白秉真，有二次是白秉貞。職官名稱也常有不同的用詞，雖然雲南總督與雲貴總督已經合併為雲南貴州總督，但在《清聖祖實錄》中有時候還是會出現雲南總督、雲貴總督、雲南貴州總督等不同的用詞。

（三）《清聖祖實錄》與《清代職官年表》的說法不一：例如在《清聖祖實錄》中康熙六十一年二月載有「直隸總督趙弘燮疏報...」，但《清代職官年表》卻非如此。又如《清聖祖實錄》中康熙元年六月載有「雲南總督趙廷臣疏言、曹滴司改土為流...」，但《清代職官年表》中此時的雲南總督是卞三元。

在職官任職資訊的應用方面，我們選用清朝官制裡正一品至從二品，一共四個品第的職官資料進行實驗。循序共陞樣式主要的參數有最小支持數(min_sup)、Maximum Gap(max_gap)及 Window Constraint(win_size)。Community Mining 是利用 Modularity 演算法、Role Mining 則是利用 Out Degree Centrality。圖五所示為康熙年間職官的人脈網絡、群組(派系)與角色(重要性)的視覺化。其中，相同的顏色代表相同的 Community，節點的大小代表 Centrality 的大小。針對 Centrality 較大的節點，我們標示出其人名。



圖四：視覺化人脈網絡(min_sup=3, max_gap=12, win_size=24)。

五、結論

在傳統的史學研究過程中，首先面臨之最大問題在於史籍文料數量相當繁多，若要從中彙集研究所需的資料，往往都必須投入大量的人力與時間，研究過程相當艱辛，所耗費的人力與時間成本更是難以估計。

有鑑於傳統史學研究之困難，本論文運用資料探勘技術，提出由史料中根據職官名稱，識別職官人名、探勘職官任期。在職官年表的應用方面，我們根據官場上仕途進退關係，提出循序共陞樣式探勘找出職官的共陞現象，並從中建立歷史人物的人脈網絡，以協助分析派系及權臣。希望能提供史學研究者更多基於仕途關係，其所內隱之人脈網絡關係資訊，供歷

史學者更易進行研究與分析。

在後續的研究上，針對職官人名識別，部分官職有一位以上的情形。例如清代協辦大學士滿人兩人、漢人兩人。目前我們的方法並沒有考慮這種情形。因此，也影響到循序共陞樣式的探勘結果。若加入考量，將可提升準確率。針對出現次處不頻繁的職官或歷史人物，若能輔以詞夾子演算法[6]，並搭配任職用語，將可提昇人名識別與任職時間的準確率。此外，詞夾子演算法乃透過樣本人名，找出與其相對應的詞夾子，進而找出更多的人名。本研究所找出的職官人名，也可提供詞夾子樣本人名，以提昇詞夾子的準確率。

針對社群網絡之建立，目前我們是根據職官共陞的關係。共貶關係也是判斷社群連結的線索。如果同時被貶官，也很有可能屬於相同的派系。而相同品第的職官，其實也有高低之分。例如兩廣總督與雲貴總督均為正二品職官，但兩廣總督駐廣州，廣州係清朝時期的對外通商口岸，坐擁豐富之稅收。若由雲貴總督調為兩廣總督應視為陞官。此外，職官的氏族關係與史料中事件的共現關係也是建立社群網絡的重要線索。歷史上家世與氏族在政治的舞台扮演重要的角色。尤其以魏晉南北朝時更為盛行。紀傳體史料中，世家與列傳就記錄了歷史人物的家世與氏族關係。透過紀傳體史料(例如清史稿)，我們正研究利用 Information Extraction 相關技術，來判斷歷史人物的氏族關係，並據以建立職官社群網絡的連結。史料事件中的共現關係也是社群網路的重要線索。如果兩個歷史人物出現在同一事件中，則在社群網絡中所對應的節點間彼此也應有連結相連。《清實錄》所收諭旨、奏疏等，均以時序排列，中間置以圈號 「○」來表示不同事件的間隔。我們也正研究根據文本中的圈號，來判斷歷史人物的共現場合，以更精確地描述歷史人物的社會網絡。

參考文獻

[1] 錢實甫編，清代職官年表 (共四冊)，中華書局出版社，北京，1980 年。

[2] 趙爾巽等纂修，清史稿 (共五冊)，博愛出版社，臺北，1983 年。

[3] 張敏與毛少平, 用於信息檢索的古文統計分析, 中文信息學報，第十五卷第六期，2001 年。

[4] 毛婷婷、李麗雙與黃德根，基於混合模型的中國人名識別，中文信息學報，第二十一卷，第二期，2007 年。

[5] 羅鳳珠，臺灣地區中國古籍數位化的現況與展望，第三次兩岸古籍整理研究學術討論會，2001 年 4 月。

[6] 張尚斌，詞夾子演算法在專有名詞辨識上的應用─以歷史文件為例， 臺灣大學資訊工程學系碩士論文，2005 年。

[7] 古鴻廷，清代官制研究，五南圖書出版社，臺北，2005 年。

[8] 李中國與劉穎，邊界模板和局部統計相結合的中國人名識別，《中文信息學報》，第二十卷，第五期，2006 年。

[9] 宋邡熏，由職官年表中利用循序共現樣式探勘人脈網絡，政治大學資訊科學系碩士論文，2010 年。

[10] 廖儁凡，中國古典小說中的社會網路關係：以儒林外史為例，台灣大學資訊網路與多媒體研究所碩士論文，2010 年。

[11] R. L. Breiger, The Analysis of Social Networks, In Handbook of Data Analysis, London: Sage Publication, 2004.

[12] A. Clauset, M. E. J. Newman, and C. Moore, Finding Community Structure in Very Large Networks, Physical Review E, Vol. 70, No. 6, 2004.

[13] S. P. Chen, J. Hsiang, H. C. Tu, and M. Wu, On Building a Full-Text Digital Library of Historical Documents. Proceedings of International Conference on Asian Digital Libraries, 2007.

[14] J. Gao, C. J. Chu, and M. K. Shan, Social Network Mining from Historical Documents: by Example during Qianlong's Reign, IICM Communication, Vol. 11, No. 4, 2009.

[15] J. W. Huang, B. R. Dai, and M. S. Chen, Twain Two-End Association Miner with Precise Frequent Exhibition Periods ACM Transactions on Knowledge Discovery from Data, Vol. 1, No. 2, 2007.

[16] K. T. Lua, and K. W. Gan, An Application of Information Theory in Chinese Word Segmentation, Journal of Computer Processing of Chinese and Oriental Language, Vol. 8, No. 1, pages115-124, 1994.

[17] Y. Matsuo, J. Mori, M. Hamasaki, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka, POLYPHONET: An Advanced Social Network Extraction System from the Web, Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 5, 2007.

[18] A. McCallum, X. Wang, and A. Corrada-Emmanuel, Topic and Role Discovery in Social Networks with Experiments on Enron and Academic Email, Journal of Artificial Intelligence Research, Vol. 30, 2007.

[19] S. Miller, H. Fox, L. Ramshaw, and R. Weischedel, A Novel Use of Statistical Parsing to Extract Information from Text, Proceedings of the 6th Applied Natural Language Processing Conference, pages 226-233, 2000.

[20] M. Newman, The Structure and Function of Complex Networks, SIAM Review, Vol. 45, No. 2, 2003.

[21] M. Newman and M. Girvan, Finding and Evaluating Community Structure in Network, Phys. Rev, 2004.

[22] J. Nieminen, On Centrality in a Graph, Scandinavian Journal of Psychology, Vol. 15, pages 332-226, 1974.

[23] W. D. Nooy, Exploratory Network Analysis with Pajek, Cambridge University Press, 2005.

[24] J. Shetty, and J. Adibi, Discovering Important Nodes through Graph Entropy: the Case of Enron Email Database, Proceedings of 3rd international workshop on Link discovery, LinkKDD, 2005.

[25] C. T. Tseng, Exploiting Search Techniques to Discover Sex Degree of Separation Between People from Web, Master Thesis, Department of Computer Science and Information Engineering, National Taiwan University, 2009.

[26] S. Wasserman, and K. Faust, Social Network Analysis: Methods and Applications, Cambridge University Press, 1994.

[27] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J. R. Wen, StatSnowball: A Statistical Approach to Extracting Entity Relationships. Proceedings of 18th International World Wide Web Conference, 2009.

[28] X. Zhu, M. Li, J. Gao, and C. N. Huang, Single Character Chinese Named Entity Recognition, Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing, pages 125-132, 2003.