國立臺北科技大學

資訊工程系碩士班

碩士學位論文

社群媒體熱門主題偵測研究_以批踢踢實業坊為例

Popular Topic Detection in Social Media-using PTT as an Example

研究生：蔡季霖

指導教授：王正豪 博士

中華民國一百零四年七月

摘要

論文名稱：社群媒體熱門主題偵測研究_以批踢踢實業坊為例

校所別：國立臺北科技大學資訊工程系碩士班　　頁數:61 頁

畢業時間：一百零三學年度第二學期　　學位:碩士

研究生：蔡季霖　　指導教授:王正豪

關鍵詞：主題偵測, 斷詞, 短文分群

社群網站蓬勃發展，國外已經有許多熱門主題偵測的相關研究與追蹤(TDT)的技術，可以從大量文字資料中擷取出主題。然而現有 TDT 的技術大多侷限於英文新聞資料集，但是社群網站上不同作者的用字遣詞與文章結構都截然不同，因此現有技術無法適用於中文的社群文章。

本論文採用的方法能夠對任何時段區間的文章斷詞，得到文章的特徵向量，再使用分群的方法，將相關的文章聚在一起。最後，藉由每一個群裡分數最高的關鍵字代表該群的相關主題，如此一來就可以偵測任何時段區間的熱門主題，並追蹤熱門主題在不同時段的變化。

實驗資料使用批踢踢實業坊八卦版 2015 年二月到六月的文章。結果顯示，本論文能夠有效偵測到該時間區間的熱門新聞以及大眾所熱烈討論的主題，能為新聞工作者與一般網路使用者只想關注熱門主題的需求帶來更好的便利性。

ABSTRACT

Title：Popular Topic Detection in Social Media using PTT Pages: 61

School：National Taipei University of Technology Degree: Master

Department: Computer Science and Information Engineering Advisor: Jenq-Haur Wang

Time：July, 2015

Researcher：Chi-Lin Tsai

Keywords：Topic Detection, Word Segmentation, Text Clustering

A popular topic is defined as a seminal event or activity along with all directly related events and activities. Topic Detection and Tracking (TDT) techniques could automatically capture the topics from large quantity of the words; however, most of the existing TDT techniques were limited to English news dataset, but differences in languages and post styles makes TDT techniques less applicable to posts on social networking sites.

In this paper, we propose a popular topic detection method in social media. With word segmentation to any posts during a period of time, the proposed method used in this paper could achieve their feature vectors and thus cluster similar posts altogether. Key terms with highest scores are chosen to represent the topic of a particular cluster, making it able to detect popular topic during any time, tracking variation of topics along time.

The test data we collected were posts from Gossiping Broad of PTT between Jan. 2015 and June. 2015. The results show that the proposed method could effectively detect popular news and popular topics shared discussed by the public during a time interval, bringing more convenience for journalists and all Internet users who want to focus on only the most discussed topics.

誌謝

人生第一份論文，許多驚奇在這過程發生。一路上要感謝王正豪教授不辭辛勞地教導，感謝佳志、奕豪和黃驤三位學長在我碩一時給我很多想法和建議，還有在業界的經驗和觀念等等，讓我收穫良多。

一路上實驗室的夥伴們給了我很多幫助，宇辰幫忙負擔許多國科會繁多的瑣事，並且帶給我們實驗室很多歡樂，非常感謝；冠廷幫了我一大把，許多演算法、程式流程和觀念都是與他討論，在跟他討論的過程中我收穫良多，非常感謝；謝謝廷翰幫助我許多程式除錯的部分，還有很多演算法、研究方法的討論，謝謝他的幫忙，還有國中同學皓縈的幫忙，沒想到在研究所依然能夠一起努力，還有謝謝宗廷的加油打氣。

謝謝家人一路陪伴的鼓勵與支持，讓我能無後顧之憂並安心地將碩士學位取得，學生之路就要在 24 歲燙下一個休止符，但是學習的路程仍然會繼續。學生的生活轉眼間就這麼過去了，謝謝毓婷陪伴著我這一生最後大半的學生時光，雖然這份論文最後是自己完成，但謝謝妳一路上曾經陪伴我並且支持我。

這份論文得來不易，在許多個看日出的日子後，論文的果實也十分甜美，就是自己從零開始所孕育的種子，不斷地學習如何加入養分，不斷地讓自己學習新的知識，這些種種都對我的未來有非常大的幫助，希望能夠在職場上學以致用，製作更好的產品，提出更多創新的想法，這是我不斷對自己要求的，因為創新是非常有趣，並且很有成就感的。

最後，期許自己在未來的旅途中，能夠不斷地充實自己，面對失敗，接受失敗，改進自我，成為更好的自我。這篇論文謹獻給所有愛我的家人與朋友們，和曾經愛過我，陪伴在我身邊的人。

季霖 2015.07.31

目錄

圖目錄

表目錄

第一章 緒論

本章節將介紹本論文的研究背景與動機，描述研究目的與貢獻，並說明整篇論文的組織架構。

1.1 研究背景與動機

隨著網路帶來的便利性，BBS（Bulletin Board System）社群文章討論網站平台越來越活躍，如 Reddit（https://www.reddit.com/）[1]、PTT 等等，當越來越多使用者在社群文章討論平台上面發表文章，以及每天新聞媒體所播報的上千則新聞，討論的主題會越來越多，造成大多數的使用者要將所有文章和新聞全部讀完是非常困難的事情，因為使用者必須花大量的時間成本去閱讀。

大多數的使用者對於大眾們所關心的主題非常感興趣，也就是當許多人們都熱烈地參與某一個主題時，這個主題就可以稱為“熱門主題”。找出熱門主題可以幫使用者省下龐大的時間成本，過濾討論非常少的文章，熱門主題有可能具有潛在的商業契機，可以觀察大眾容易被什麼吸引，或是常常人們遇到的問題，種種討論都是成為熱門主題的因素。

為了偵測熱門主題，本論文從社群文章平台選取可能成為熱門主題的文章，並且加以分析，並藉由每一篇文章的特徵，將有可能是相關討論的主題聚在一起，再使用關鍵字擷取，使得每一個聚在一起的文章群成為一個主題，以達到熱門主題偵測的目的。

1.2 研究目的

因為社群文章討論的即時性，使得社群文章討論平台通常會在短時間內出現大量的文章，這讓使用者難以閱讀目前的熱門主題，因為大多數的使用者渴望關注哪些主題是目前被人們熱烈討論的。

在方法部分，過去研究將文章直接做分群，或是將關鍵字做分群來找出熱門主題，但大部分都是針對英文文章進行分析。因此本研究嘗試結合這兩種分法，並且針對中文的社群文章討論平台研究，我們希望從每天被大量發佈的文章當中，探勘哪些主題才是真正的熱門主題。

1.3 研究對象

批踢踢實業坊 （英語：PTT，本論文之後皆以-PTT 稱之）是一個 BBS 電子佈告欄系統，以學術性質為目的在網路上提供快速即時、免費、開放、自由與平等的言論空間。目前是由國立臺灣大學電子布告欄系統研究社管理，大部份的系統原始碼目前是由資訊工程學系的學生與校友進行維護，並且邀請法律專業人士擔任法律顧問[2]。PTT 社群平台與目前時下流行的社群媒體（Facebook、Twitter、Plurk…等）差別的地方在於 PTT 是一個公共的討論平台，裡面有許多討論區，如圖 1.1 所示，進行發文或是參與各式各樣的討論，討論的參與者是來自任何一個註冊 PTT 帳號的使用者。

圖 1.1 PTT 看版截圖

在台灣，PTT 具有相當大規模的社群，以及極佳的即時性互動，故在網路上發生的一些事情甚至足以影響現實社會，成為新聞媒體報導對象。有些組織、甚至政府單位，亦會參考 PTT 的輿論進行調整，以切合民意。例如：台鐵與高鐵：每日均會檢視相關看板之討論[1]。

目前在批踢踢實業坊註冊總人數約 150 萬人，尖峰時段超過 15 萬名使用者同時上線，八卦版甚至同時在線人數曾經高達 100084 人過，如圖 1.2 所示，圖 1.2 右上角的人氣是 PTT 八卦版當下版內的人數，擁有超過 2 萬個不同主題的看板，每日超過 2 萬篇新文章及 50 萬則評論（在 PTT 稱為推文）被發表 [2]。

[1] “跟上民意 雙鐵成立 PTT 監看小組”，聯合報（http://archive.is/Tyfac）

圖 1.2 同時 10 萬名使用者在八卦版截圖

因為某些主題被熱烈討論，常常有許多鄉民創造一些有趣的字彙，並且被報章雜誌媒體所引用，所以 PTT 站方建立了一個網站透過所有的 PTT 使用者一起新增與編寫，簡稱鄉民百科[2]，如圖 1.3 所示。

圖 1.3 鄉民百科截圖

[2] 來源：http://zh.pttpedia.wikia.com/wiki/PTT 鄉民百科

以下我們舉三個從鄉民百科查到的名詞：

1. 鄉民：現用於泛稱 PTT 的使用者。鄉民一詞最初的由來是周星馳電影《九品芝麻官》中狀師方唐鏡的台詞：「我是跟鄉民進來看熱鬧的，只是往前站了一點」而在 PTT 較廣泛使用該詞，是在 2004 年 5 月至 8 月間。
2. 馬路三寶：是 ptt 的流行用語，尤其流行於八卦板。其意思是鄉民們認為馬路上最會造成危險的三種人：女人、老人、老女人。該用法由來應不是起源自 ptt，但約在 2010 年之後在八卦版開始常常出現，甚至只要出現「三寶」直接大家就會直接想到是馬路三寶。
3. 銅鋰鋅：(同理心）或作”銅李星”、”捅李星”等等，是 PTT 的流行用語，在 2015 年 6 月 27 日八仙樂園派對粉塵爆炸事故之後，八卦板內文與推文大量出現。銅鋰鋅是"同理心"的諧音字，用來反應鄉民對於台灣社會的濫情理盲的另一種想法。一開始有些鄉民對於參加者或家屬的某些作為有些批評，另一些鄉民會以「沒同理心」「要有同理心」來回應。

我們對於同時大量的 PTT 使用者在八卦版非常感興趣，透過 PTT 的歷史版（PTT HISTORY）列出當天在八卦版討論的主要事件，如表 1.1 所示。

表 1.1 八卦版人數與主題

日期	人數	事件
2014/03/19	53494	反服貿佔領立法院事件
2014/03/20	42936	反服貿佔領立法院事件
2014/03/21	44480	反服貿佔領立法院事件
2014/03/22	44723	反服貿佔領立法院事件
2014/03/23	100084	反服貿佔領立法院事件
2014/03/24	45951	反服貿佔領立法院事件

2014/03/30	40546	反服貿凱達格蘭大道黑衫軍集會
2014/11/07	51340	台北市長候選人辯論會
2014/11/29	83538	九合一地方公職人員選舉日
2014/11/30	46502	九合一地方公職人員選舉後一日
2014/12/01	40946	九合一地方公職人員選舉後二日
2015/06/27	56212	八仙樂園粉塵爆炸事件

透過人工找出當天新聞主題的對照，我們可以很明顯地發現當台灣有重大主題或是新聞事件發生時，八卦版的人數就會暴增，非常多的使用者會湧入 PTT 討論時下熱門的話題。

1.4 研究貢獻

本論文進行大量的實驗，來證實中文主題偵測的方法，能夠應用在社群文章討論網站平台，並且成功偵測熱門主題，也能夠持續追蹤熱門主題，並且抓出熱門主題的討論核心。

1.5 章節概要

本論文第一章為緒論，描述研究背景、對象、目的與貢獻。第二章為相關研究探討，介紹近期的文獻與技術。第三章為研究方法，說明整個系統的架構流程，流程共分為五個部分：潛力文章篩選 (Potential Article Selection)、斷詞(Word Segmentation)、 分群(Clustering)、 關鍵字擷取(Key Term Extraction)、 事件偵測(Topic Detection)。第四章為實驗部分，驗證研究方法的可行性。第五章為結論與未來展望，總結本論文的有效性以及未來發展。

第二章 相關研究

在說明本論文提出的方法以及實驗之前，本章節會先介紹過去的相關研究，以存在的方法和技術，以及本論文所提方法的差異。

2.1 熱門主題相關研究

主題偵測與追蹤(TDT)的技術[3]，可以從大量文章中偵測出主題。然而過去 TDT 的技術專注於新聞文章，不適用於短文字、變化較多而且口語化的社群文章。熱門主題相關研究已經有很多[4][5][6]，有些研究使用語義分析，例如： 透過 Latent Semantic Analysis (LSA) [7][8]分析文章，將每篇文章做成特徵向量，再分群或是分類、而 Probabilistic Latent Semantic Analysis (PLSA) [9][10] 是改善 LSA，利用機率模型分析文章語意之間的關係，Zhang et al.[11] 三人就是參照這兩種研究方式進行。

Lee et al.[12]使用關鍵字擷取的方法，總結網路上的韓文新聞資料集並將主題呈現出來，他們使用六種改良的 TF-IDF 進行主題的追蹤。 Chen et al. [13] 提出一個名為 TSCAN 的模組，去分析主題，將主題的 themes、events 和 event summaries 擷取出來，並依照主題的時間順序，將不同主題的 themes、events 和 event summaries 建立關聯性，並將整個剖析的過程圖像化，方便了解各個主題的原委以及結果。

Wartena et al. [14] 提出使用 k-bisecting [15] 對關鍵字來偵測主題，他們實驗的對象是英文的維基百科。他們先將最常出現的關鍵字挑選出來，再對關鍵字進行分群，並這些分群後的關鍵字各別定義成一個主題。

Lan et al. [16] 對 Yahoo! BBS 進行主題偵測，認為一個熱門的主題要有許多文章、作者和相關性等等，他們將所有文章進行具有熱門主題潛力的篩選，篩選後使用 K-Means 分群，並使用 Back-Propagation Neural Network 的方法對主題進行熱門程度分類，然後再呈現給使用者。Ye et al. [17] 針對網站和 BBS 使用 TF-IDF 以及 Single-Link Incremental 的分群方法進行主題偵測。Li et al. [18] 從 BBS 中篩選可能成為熱門主題的文章，使用 Aging Theory 的模式去偵測主題，並採用 BBS 裡面內建前五名的熱門文章當作對照組去比對實驗結果。

從以上研究，我們知道偵測主題的方式常常使用 TF-IDF 以及分群，而以上研究沒有針對台灣的 BBS（本論文之後稱之中文的社群文章討論平台）進行熱門主題偵測，因此本論文參考相關研究的分群以及關鍵字擷取的方法在台灣的社群文章討論平台進行熱門主題偵測。

2.2 斷詞工具

中文斷詞在中文的自然語言處理上，是非常重要前置處理工作。許多中文的自然語言相關的領域，例如：自動摘要、文件檢索...等，都需要先處理中文斷詞，可見中文斷詞是相當基礎且重要的部分，中文斷詞的研究也以行之有年[19][20][21]，目前中文斷詞系統可以免費使用的分別有：Jieba [22]、中研院 [23] 、Stanford Word Segmenter [24]等，如圖 2.1 所示。

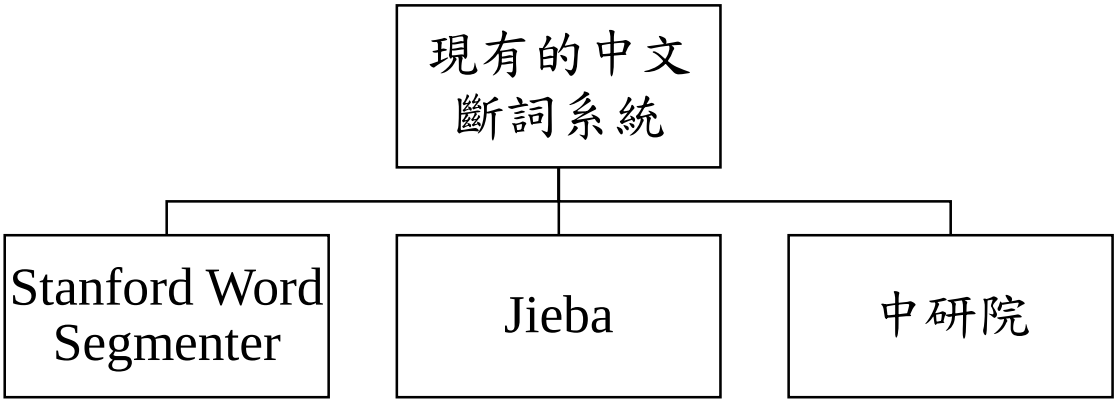

圖 2.1 現有的中文斷詞系統

由於中研院斷詞系統是需要將斷詞的內容藉由 XML 的方式傳至他們的系統端進行斷詞的運算，並不能在客戶端執行、不能一次送出大量的資料以及不能密集的傳送資料，對本論文來說大大降低了研究方便性，時間成本也隨之提高，程序也非常的繁複，所以本實驗並沒有採用此斷詞系統，如圖 2.2 所示。

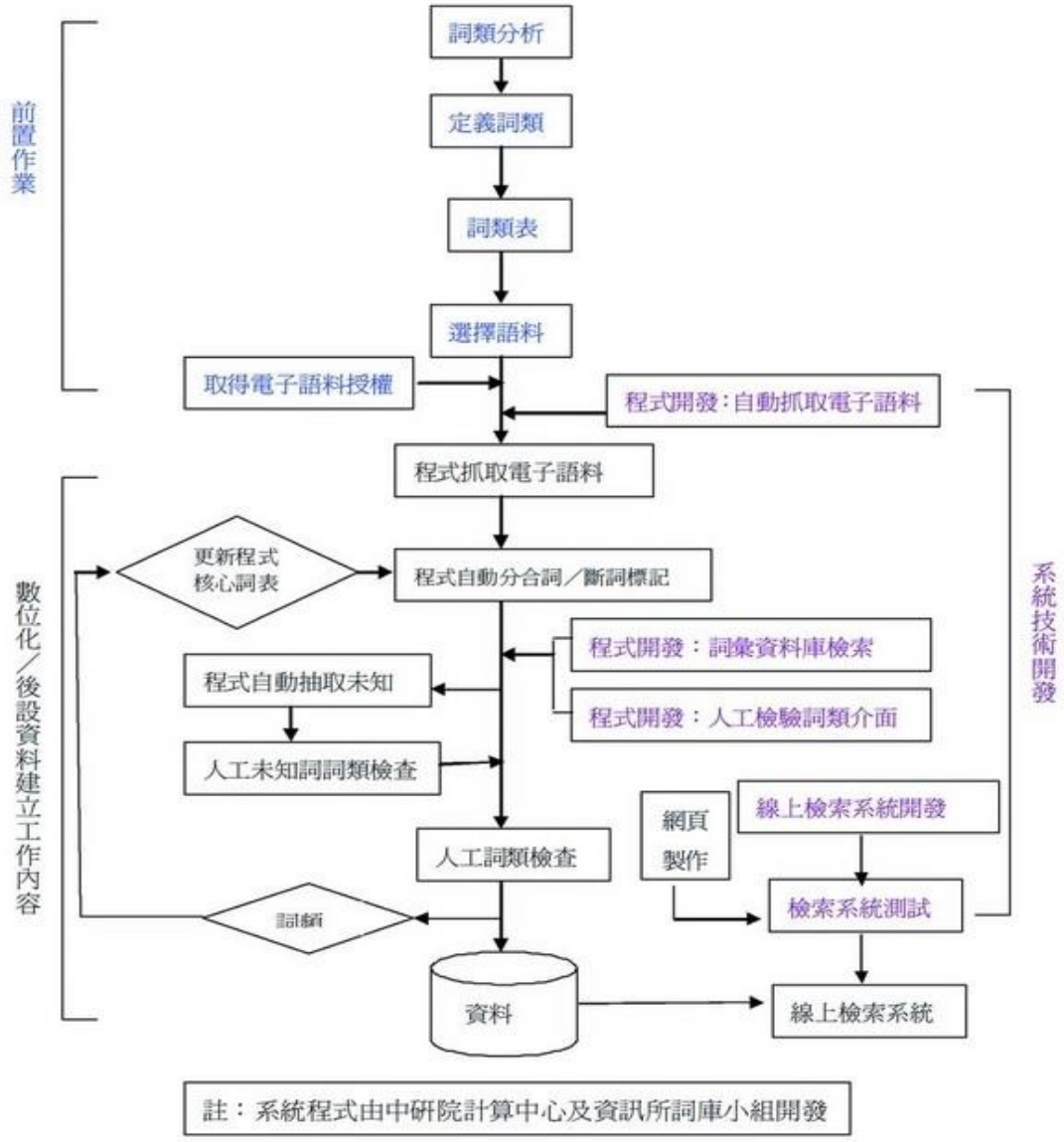

圖 2.2 中研院斷詞系統的使用流程

Stanford Word Segmenter 則是需要將斷詞的內容轉成簡體並傳至他們的系統端執行運算才會有較好的效果，也無法支援客戶端的運算，對實驗的時間成本以及便利性是一個負擔，所以我們並未採用。對於本論文來說，中文斷詞系統只是一個運算工具，我們希望使用的中文斷詞系統能夠在客戶端執行運算，並且系統能夠自動辨識新詞彙或是藉由人工添加自定義詞典，或是添加詞彙頻率的方式讓斷詞的結果更好。

Jieba 中文斷詞系統具有高度彈性的 API 可以使用，它可以應用在 7 種不同的程式語言當中，而本論文使用的 Python 系統正好是 Jieba 斷詞系統最先支援的程式語言，而且使用上非常的方便且易於上手，所以實驗部份的斷詞系統皆使用 Jieba 中文斷詞系統。

目前尚未存在一種中文斷詞系統能夠完美地處理文章內容，不過本論文的重點並不在於改善斷詞系統的運算效能，所以挑選在 Python 上能夠執行客戶端運算並且支援程度高的 Jieba 中文斷詞系統。

2.3 K-Means 分群演算法

K-Means Clustering 能夠追溯到 1957 年的 Hugo Steinhaus [25]，術語「*k*-均值」 於 1967 年才被 James MacQueen [26] 首次使用。標準演算法則是在 1957 年被 Stuart Lloyd 作為一種脈衝碼調制的技術所提出，但直到 1982 年才被貝爾實驗室公開出版 [27]。在 1965 年，E.W.Forgy 發表了本質上相同的方法，所以這一演算法有時被稱為 Lloyd-Forgy 方法。

K-Means 演算法的架構流程為先隨機挑選群組的初始中心點，接著將每一個資料點分到相似度最接近的群組，然後依據群組內的所有資料點重新計算每個群組的中心點，一直反覆執行分組與計算中心點的動作，直到所有的群組中心點不再改變為止。如圖 2.1 所示。

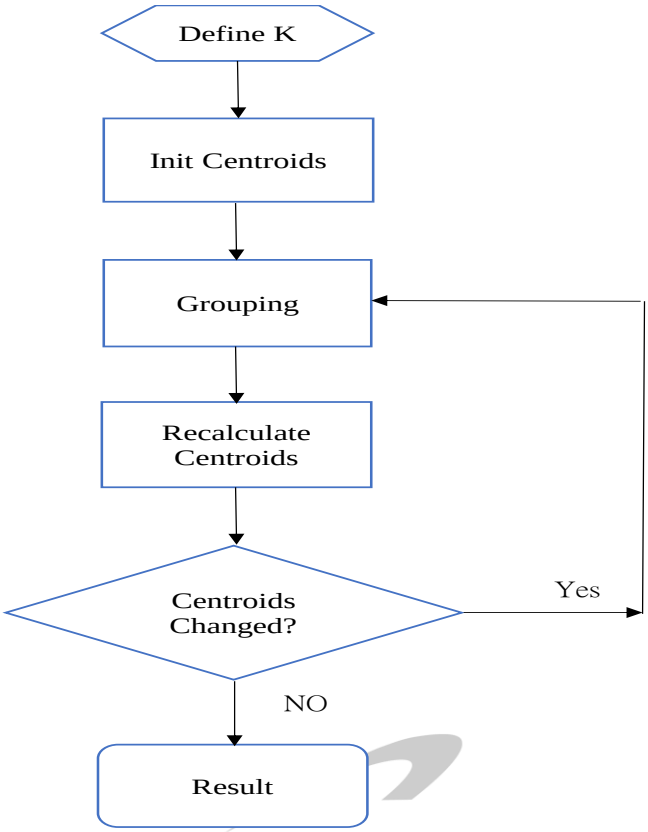

圖 2.3 K-Means 演算法架構流程圖

K-Means 分群演算法還存在一些問題，所以也有不少的相關研究在改良這些問題，像是如何自動決定群集數量 K 的問題，Jain et al. [28] 使用最小訊息長度 (MML) 的標準與高斯混合模型 (GMM) 相結合來估計 K 值。在初始中心點的挑選方式上，Ahmad et al. [29] 等人提出了以初始中心點改良 K-Means 分群演算法的方法，其方法為不隨機挑選初始中心點，而是先計算兩兩文件的距離，再根據一些條件將文件分成幾個群組，最後以這些群的中心點作為初始中心點。Adnan et al. [30] 等人提出了採用 Principal Component Analysis (PCA) 來產生初始中心點的方法來改良 K-Means 分群演算法的效率。

但是對於本論文來說，選取 K 的問題並不是研究的重點，在第四章時會藉由實驗資料來決定 K 值取多少對本論文的實驗是較適合的。

第三章 研究方法

本論文提出的方法可分為五大部分：潛力文章選取 (Potential Article Selection)、斷詞(Word Segmentation)、分群(Clustering)、關鍵字擷取(Key Term Extraction)、事件偵測(Topic Detection)。流程為取得資料後，先進行潛力文章選取，篩選可能熱門之文件，並將文章標題及內容做斷詞。再來就進行文章的特徵轉換，藉由斷詞的關鍵字製作每篇文章的特徵向量，並給予權重，並根據向量進行文章分群。最後進行主要的事件偵測。如圖 3.1 方法架構圖所示。

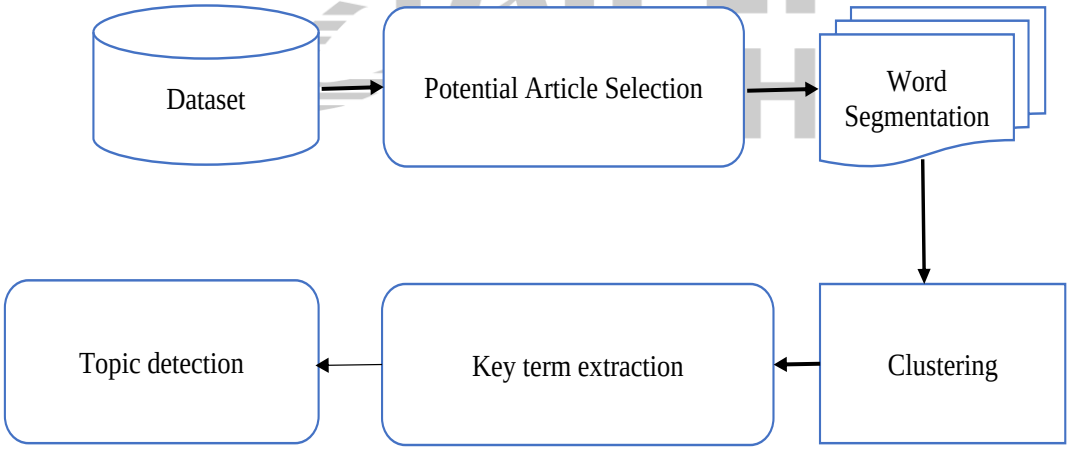

圖 3.1 方法架構圖

3.1 Potential Article Selection

在 PTT 裡,任何一個使用者都可以藉由其他使用者所發佈的文章,重新發佈一篇文章進行更進一步的討論，由於標題會一樣，所以這個新的文章就會算是原標題的其中一篇“回文”，將所有回文的文章加總後得出來的就是這個標題的”回文數”。

每天都有許多文章被使用者發佈並且進行討論，但並不是每一個話題都可以讓大多數的使用者感興趣，所以我們以”回文數”做為一個參考標準，將某些能讓使用者產生興趣的話題蒐集起來，因為這些文章很有可能就是具有熱門潛力的話題。為了篩選具有成為熱門主題潛力的文章，我們藉由回文數為基底（Reply-Based，本論文之後簡稱 RB）將文章過濾出來，作為 Word Segmentation 的輸入。

表 3.1 以回文數篩選 2015 年 4 月 11 日具有潛力的熱門文章範例

標題	回文數
［新聞］中捷意外 林欽榮:工程進度超前 中捷未趕	13
［新聞］林佳龍臉書疑隱藏「捷運趕工」動態？ 網	10
［爆卦］林佳龍偷刪「拆捷運圍籬」臉書貼文	10
［新聞］台中捷運意外 林佳龍強調無要求趕工	8
［新聞］北捷代辦中捷死傷　施工廠商為遠揚營造	7

如表 3.1 所示，回文數越多，代表這篇文章的內容引起 PTT 使用者的興趣，越來越多人參加討論，甚至是藉由此文章的內容再發一篇文章。

3.2 Word Segmentation

本論文藉由斷詞系統將文章執行斷詞的運算，將每一篇文章斷詞後，製作成該篇文章的特徵向量，並作為 Clustering 的輸入並進行分群的運算。

Jieba 斷詞系統是一個中文斷詞工具，有三種斷詞模式：

1. 精確模式：試圖將句子最精確地切開，適合文本分析。
2. 全模式：把句子中所有可能成為詞彙的詞語都掃描出來。
3. 搜索引擎模式：對較長的詞彙再次切分，提高召回率。

以”小明畢業於臺北科技大學電資科學院計算機所，後來在日本京都大學深造。”句子當作範例，如表 3.2 所示。

表 3.2 Jieba 模式斷詞範例

模式	將範例句子斷詞後的結果
精確模式 Default Mode	小明/畢業/於/臺北科技大學/電資/科學院/計算機所/，/後來/在/日本京都大學/深造/。
全模式 Full Mode	小/明/畢業/於/臺北/臺北科技大學/科技/大學/電/資/科學/科學院/學院/計算/計算機/計算機所/算機/所/後來/在/日本/日本京都大學/京都/京都大學/大學/深造/。
搜索引擎模式 Search Mode	小明/畢業/於/臺北/科技/大學/臺北科技大學/電資/科學/學院/科學院/計算/ 算機/計算機/計算機所/，/後來/在/日本/京都/大學/日本京都大學/深造/。

本論文會在第四章實驗哪一種斷詞模式比較適合我們的主題偵測，並在往後的實驗直接採用對本論文來說效果最好的模式。

Jieba 斷詞系統可以指定自己自定義的詞典，以便包含 Jieba 詞彙庫裡沒有的詞彙。雖然 Jieba 有新詞識別能力，但是自行添加新詞可以保證更高的正確率，如表 3.3 所示。

表 3.3 文字切割範例

是否加入自定義辭典	文字切割範例
預設（無加入）	**台/中**/捷運/日夜/趕工/拚/提前/兩年/上路/，**台/中**/市長/**林/佳/龍**/要求/提前/兩/年/完成。
加入自定義辭典	**台中**/捷運/日夜/趕工/拚/提前/兩年/上路/，**台中**/市長/**林佳龍**/要求/提前/兩/年/完成。

斷詞系統預設的情況下，會將“台中”分成“台/中”兩個詞彙，將“林佳龍”分成“林/佳/龍”。但是”台/中”這個斷詞結果對本論文來說不盡理想，”林/佳/龍”則是人名，有時候斷詞系統會無法分辨，這時候在字典檔中加入”台中”與”林佳龍”兩個字彙，Jeiba 斷詞系統就會學習並且校正，輸出較理想的結果而自定義加入的詞彙會越加越少，因為 Jieba 會透過自定義的詞彙自動學習，本論文總共增加 200 個詞彙。

3.3 Clustering

由於 K-Means 演算法的架構流程為先隨機挑選群組的初始中心點，接著將每一個資料點分到相似度最接近的群組，會因為隨機挑選的關係，有分群不均勻的情況，所以本論文將會在分群的這個部分執行一百次取平均以得到誤差較小的結果，公式 3.1 為 K-Means 的公式。

$$\operatorname{argmin} s \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 \quad (3.1)$$

由公式 3.1 得知，K-Means 不斷地改變中心，將距離近的特徵聚在一起。本論文直接採用 K-means 分群的方法對已經將文章轉成的特徵向量執行運算。藉由 K-means 的特性，將特徵相近的文章聚在一起，運算過後，每一群的文章都是屬於同一個主題討論範圍的，因為群裡的文章特徵非常相近。我們選取 2015 年 4 月 11 日部分分群結果當作範例，如表 3.4 所示。

表 3.4 2015 年 4 月 11 日部分分群結果範例

第一群
[問卦] 要怎麼檢舉警察?
[新聞] 飛彈「攔腰」攻台 共軍戰略曝光
[問卦] 紅衛兵與鄉民文化的異同處
……
第二群
[新聞] 林佳龍臉書疑隱藏「捷運趕工」動態(假新聞
[新聞] 林佳龍：北捷違反交通維持計畫 白天吊掛
[爆卦] 林佳龍偷刪「拆捷運圍籬」臉書貼文
……
第三群
[問卦] 班上的最後一名後來都怎樣了?
[問卦] 班上的第一名後來都怎麼樣了?

3.4 Key Term Extraction

在分群運算後，每群包含許多討論的文章，也就是每一群代表相關事件的集合，而事件通常包含幾個代表性的關鍵字，這些關鍵字我們又可以稱為事件的”核心”，所以在本節我們試著找出這些事件的”核心”。我們計算每一個關鍵字的 Term Frequency，並過濾 stop word，按照分數排名列出來，因為這些排名前面的關鍵字有利於我們定義主題，如公式 3.2 所示。

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \qquad (3.2)$$

公式 3.2 中的 $n_{i,j}$ 是關鍵字 j 在第 i 個群中出現的次數，而分母則是在群 i 中所有字詞的出現次數之和。參考相關研究 [18] 透過社群文章討論平台本身的前五名熱門主題當作對照組，本論文選取群 i 裡前五名的關鍵字做為群 i 的核心代表。

不使用 TF-IDF 的原因在於，我們已經將具有高度相關的文章聚在一起，此時若對每一群使用 TF-IDF，會將這個群的核心關鍵字濾除，因為這個群中常常出現的關鍵字正是主題討論的核心，但卻因為 TF-IDF 的特性被視為不重要的訊息而被過濾，反而非主題核心的關鍵字會被 TF-IDF 挑選出來。

選出來的關鍵字會與人工選出來的關鍵字做比對，測試系統的準確率。流程如圖 3.2 所示。

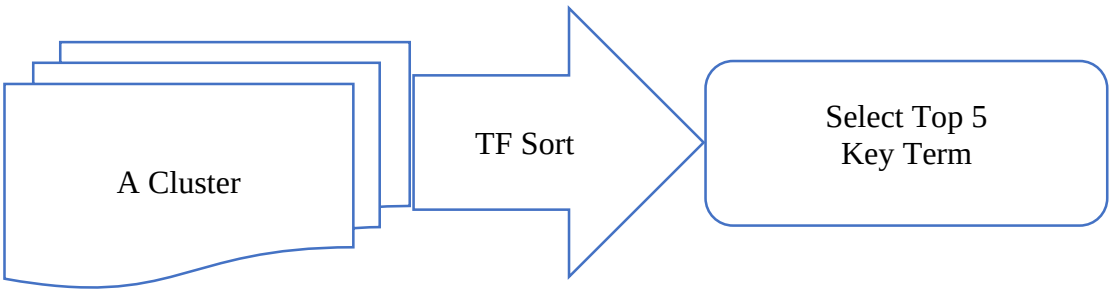

圖 3.2 代表關鍵字選取

在此我們選 2015 年 4 月 4 日的其中一群文章為範例，如表 3.5 所

示。

表 3.5 從群中擷取關鍵字詞

人工選取的關鍵字	系統計算出的關鍵字
捷運	捷運
林佳龍	林佳龍
施工	工程
工程	施工
意外	台中市

以表 3.5 來說，儘管人工選取關鍵字沒有名次之分，但本論文只考慮人工選出的關鍵字是否有出現在系統計算出的關鍵字當中，所以只要有出現就算正確，所以只有人工選取的”意外”以及系統選取的”台中市”沒有比對正確。

3.5 Topic Detection

藉由 Key Term Extraction 的輸出，透過人工檢視可以觀察出排名前五的關鍵字能夠讓我們對整個事件的重點部分有大致上了解。參考相關研究 [14] 主題的定義，我們將每一群前五名的關鍵字定義成一個主題。隨著時間的變化，關鍵字也會有所改變，有一些關鍵字會不斷的重複出現，更能夠肯定此關鍵字必定是事件的主要核心；有些關鍵字會慢慢變少被其他關鍵字取代，代表這個關鍵字的核心已經不再被大眾所討論，失去熱門的性質。

而有些主題會消失，代表這個主題已經不再被大眾所討論，取而代之有些新的主題會出現，因為熱門主題是具有時效性的，所以主題會不斷的改變。藉由觀察關鍵字沿著時間軸的變化，關鍵字會不斷的改變，本論文藉由關鍵

字排名的變化而動態地調整。如圖 3.3 所示。

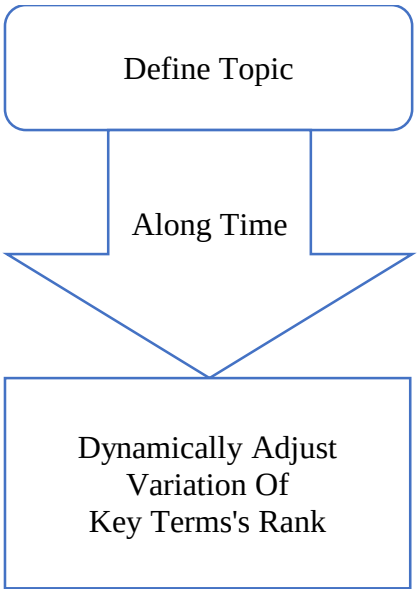

圖 3.3 主題偵測的流程圖

接下來我們很好奇大部分的熱門主題能夠持續多久，是一天還是一個禮拜，所以我們嘗試在網路和 PTT 去觀察大部分的熱門主題能夠持續討論幾天，如表 3.6 所示。

表 3.6 透過人工觀察部分熱門主題持續時間

主題	持續時間	持續天數
復興航空空難	02/05~02/09	5
高雄劫獄	02/12~02/14	3
慈濟園區變更案	03/17~03/19	3
藝人遊玩阿帕契	04/03~04/11	9
大巨蛋勒令停工	05/21~05/22	3
台大生爬山募款	06/22~06/24	3

透過人工觀察，我們發現大多數的熱門新聞或是熱門主題，持續的時間差不多都是三天或是三天以上，在此我們設計一個公式去追蹤每一個關鍵字的熱度，如公式 3.3 所示。

$$Popularity\ Score(i,j,n)$$

$$= \begin{cases} \dfrac{TF(i,j,n)}{N}, & TF(i,j,n) \neq 0 \\ \dfrac{2Score(i,j,n-1)}{3}, & TF(i,j,n) = 0,\ TF(i,j,n-1) \neq 0 \\ \dfrac{Scroe(i,j,n-2)}{3}, & TF(i,j,n) = 0,\ TF(i,j,n-1) = 0,\ TF(i,j,n-2) \neq 0 \end{cases}$$

公式(3.3)

$Popularity\ Score(i,j,n)$（以下簡稱 PLS）表示第 i 群的關鍵字 j 在第 n 天的分數，$TF(i,j,n)$ 則是第 n 天關鍵字 j 在群中出現的次數，N 為群中所有字的總數。如果第 n 天關鍵字 j 有出現在該群的前五名中出現，往後敘述關鍵字在前五名的話我們都稱為有出現，將此關鍵字藉由該群裡的總字數 N 做正規化，$Score(i,j,n)$ 的值便是使用 $\frac{TF(i,j,n)}{N}$ 來表示。若是第 n 天關鍵字 j 沒出現，但在 n-1 天時有出現時，則將 $\frac{TF(i,j,n-1)}{N} - \frac{TF(i,j,n-1)}{3N} = \frac{2Score(i,j,n-1)}{3}$ 賦予 $Popularity\ Score(i,j,n)$，雖然關鍵字 j 在第 n 天沒有出現，但是我們保留此關鍵字的熱度，也就是它的值，因為它還是有可能在往後的討論天數中再次被提及，畢竟這個關鍵字 j 曾經被熱烈討論過。若是在 n 天時，關鍵字 W 仍然沒有出現，就將 $\frac{TF(i,j,n-2)}{N} - \frac{TF(i,j,n-2)}{3N} = \frac{Score(i,j,n-2)}{3}$ 賦予 W_n，直到 W_n 遞減為 0 時，我們才認定這個關鍵字已經不再被大眾所討論了，因為我們已經保留這個關鍵字三天的熱度，並將分數做遞減直至 0。藉由表 3.6 的範例，我們相信大多數的主題可以持續三天的熱度，所以公式 3.3 分母是 3。

我們以 2015 年 4 月 9 日至 4 月 14 日的“台中捷運工程意外”主題進行關鍵字追蹤，如圖 3.4 所示。

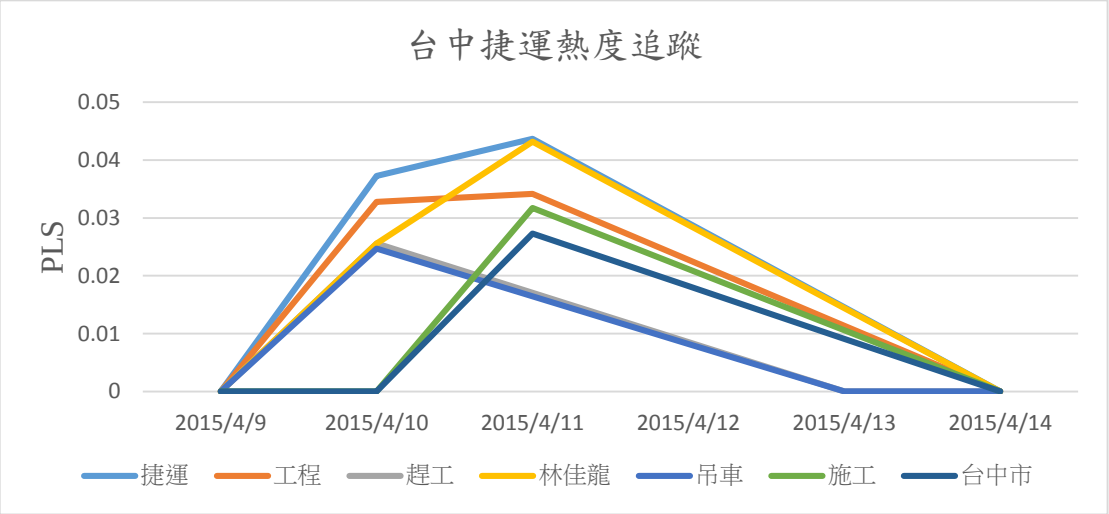

圖 3.4 追蹤關鍵字熱度範例

圖 3.4 裡的縱軸為透過公式 3.3 計算出的關鍵字熱度分數 Popularity Score，簡稱 PLS，橫軸為日期。我們可以得知，這個主題是在 4 月 10 日開始廣泛討論的，也就是主題從 10 日開始出現，在 4 月 11 日討論達到最高峰，12 日之後討論熱度持續下降，這時主題可能已經退燒，也可以發現從 12 日之後圖表的曲線非常的線性，是因為我們使用公式保留關鍵字的熱度，直到 4 月 14 日時這個主題已經不再被大眾所提及，我們在此定義這個主題持續的時間是 10 日至 13 日，總共 4 天。

我們可以再對這個主題的關鍵字做更進一步的篩選，將只出現過一天的關鍵字濾除，如圖 3.5 所示。

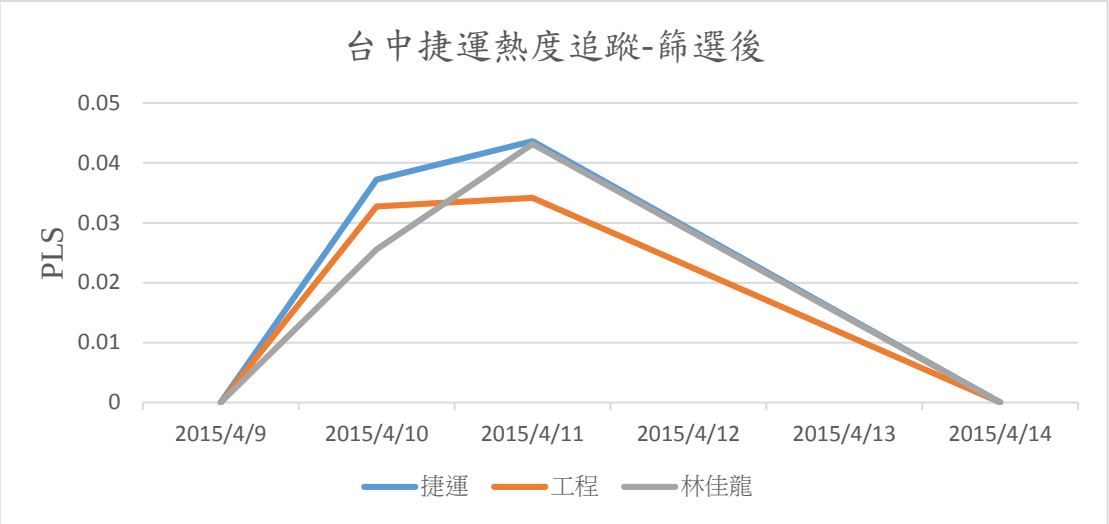

圖 3.5 對關鍵字熱度進行篩選

篩選的目的除了可以讓圖像化更直覺之外，我們也可以更清楚這個主題的核心。我們將每個主題的每個關鍵字分數各別在主題持續討論的期間做一個加總，如公式 3.4 所示。

$$Popularity\ \ Score\ Sum(i.j) = \sum_{n=1}^{n=k} Popularity\ Score(i,j,n) \quad \text{公式 (3.4)}$$

$Popularity\ Score\ Sum(i.j)$ 是第 i 個群的第 j 個關鍵字在主題持續期間裡的總和，簡稱 PLSS。$\sum_{n=1}^{n=k} Popularity\ Score(i,j,n)$ 代表從第一天到第 k 天所有關鍵字 $Popularity\ Score(i,j,n)$ 的分數加總，k 表示主題持續了幾天，以圖 3.5 來說，主題持續了三天，所以 k=3。透過這個公式，我們可以觀察這個主題持續的期間哪些關鍵字是比較重要的，如圖 3.6 所示。

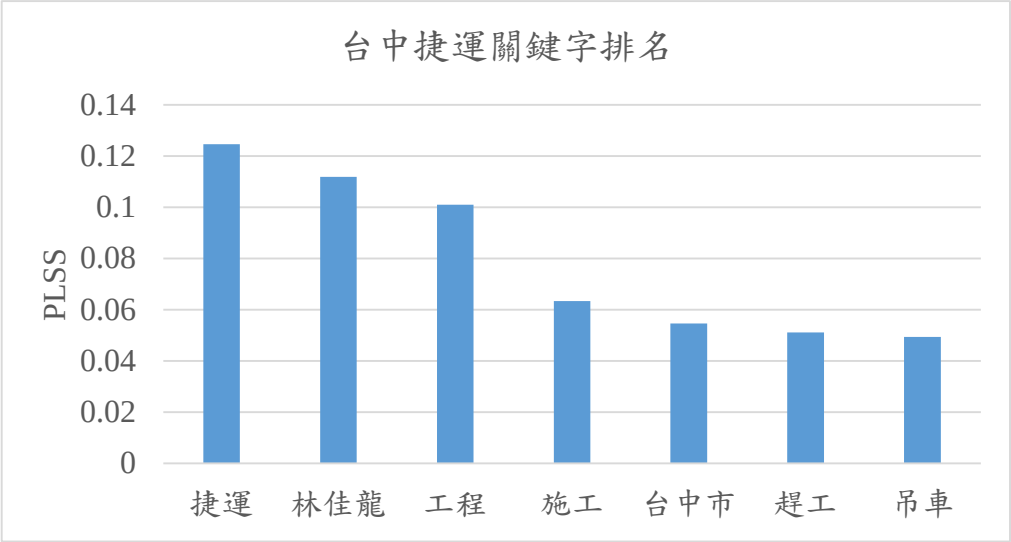

圖 3.6 主題關鍵字排名範例

我們可以發現”台中捷運”主題主要的關鍵字有”捷運”、”林佳龍”、”工程”、”施工”、”台中市”這五個關鍵字，因為這五個關鍵字的平均分數在主題討論的期間，加總的分數是最高的。

第四章 實驗與討論

本章節針對本論文所提出的方法進行實驗、分析與討論，最後將本論文提出事件偵測出來的結果，與蘋果日報和雅虎新聞所整理出來的熱門主題作比較，來驗證本論文所提出方法的有效性。

首先進行的是 K-Means 取 K 值的實驗，觀察文章數量和 K 值的關係，當分群能夠達到的最佳狀態。接著進行熱門主題偵測的測試，觀察事件隨著時間的變化並追蹤主題的熱度。

4.1 實驗架構

本論文實驗分為三個部分，第一部分為研究資料的潛力文章選取、斷詞系統參數的選擇以及分群參數調整與最佳分群的探討；第二部分為關鍵字擷取排名的實驗；第三部分為主題偵測與追蹤熱度的實驗。我們收集 2015 年 2 月 1 日至 2015 年 7 月 6 日 [至八卦版最新的文章，此論文撰寫的時間]期間 PTT 八卦版的所有文章，以中文文章為主要研究對象。

4.2 實驗環境

本論文系統開發與測試環境於 Mac OS 上以 Python 開發，如表 4.1 所示。

表 4.1 系統實驗環境列表

作業系統	OS X Yosemite 10.10.3
程式語言	Python 2.7
資料庫儲存	Solr 5.1.0

4.3 實驗資料

實驗資料來源是透過本實驗室寫出的 telnet 的爬蟲程式自動抓取下來，所有實驗使用批踢踢實業坊的八卦版裡的文章，時間從 2015 年 2 月 1 日至 2015 年 7 月 7 日[至八卦版最新的文章，此論文撰寫的時間]，總共收集 29834 篇文章，文章部分內容範例如表 4.2 所示，文章分為四大類別 (問卦、爆卦、新聞、公告)，本論文使用除了公告以外的其他三種類別，進行熱門主題偵測的測試與實驗。潛力的熱門文章篩選方式藉由 Reply-Based 的方法過濾。

表 4.2 批踢踢實業坊八卦版其中一篇文章部分內容

作者　godshibainu (神柴)　　看板 Gossiping 標題　[新聞] 李蒨蓉案再踢爆 荒唐飛官戴阿帕契頭盔跑趴 時間　Sat Apr　4 05:29:28 2015
1.媒體來源: 蘋果日報頭版 http://ppt.cc/BCY1 2.完整新聞標題: 李蒨蓉案再踢爆 荒唐飛官戴阿帕契頭盔跑趴 3.完整新聞內文: 【綜合報導】爛到骨子裡了！藝人李蒨蓉爽搭阿帕契攻擊直升機還 po 照，引發撻伐，軍方昨急懲陸軍航特部 5 名將、校軍官滅火，其中帶李蒨蓉搭阿帕契的作戰隊副隊長勞乃成，記大過調職並送法辦，其他 4 名將、校也被嚴懲。但《蘋果》調查發現，勞乃成不但帶親友進營區亂拍機密戰機，去年萬聖節還穿飛行裝、戴造價 200 萬元的阿帕契頭盔跑趴，把頭盔當私人物品向在場的李蒨蓉等炫耀，專家痛批勞「荒唐離譜至極」！ http://ppt.cc/kB5B （飛官勞乃成帶藝人李蒨蓉賞玩阿帕契引發軒然大波，又被發現去年萬聖節他戴阿帕契頭盔在家開趴。 翻攝 Joyce Chang 臉書）

4.4 分群實驗評估指標

本論文以四種評估方式來觀察分群結果，本論文使用資訊檢索領域常見的指標：精確率 (Precision)、召回率 (Recall)、Rand Index、F-measure。為了避免實驗誤差，每個實驗都分群執行 100 次後取平均值。

Rand Index[19]公式如(4.1)式

$$RI = \frac{(TP + TN)}{(TP+FP+FN+TN)} \quad (4.1)$$

這裡的 TP 表示同類別的兩文件分到同一群組，TN 表示不同類別的兩文件分到不同群組，FP 表示不同類別的兩文件分到同一群組，FN 表示同類別的兩文件分到不同群組。如表 4.3 所示。

表 4.3 Contingency table

	Same Cluster	Different Clusters
Same Class	TP	FN
Different Classes	FP	TN

Precision 為分到同一群組內的成員是同一類別的比率，公式如 (4.2) 式

$$\text{Precision} = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} = \frac{TP}{TP+FP} \quad (4.2)$$

Recall 為同一類別分到同一群組的比率，公式如 (4.3) 式

$$\text{Recall} = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} = \frac{TP}{TP+FN} \quad (4.3)$$

一般而言，Precision 高 Recall 就低，Precision 低 Recall 就高，為了計算一個合理的平均值，所以使用 F Measure，公式如 (4.4) 式

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R} \ where\ \beta^2 = \frac{1-\alpha}{\alpha} \quad (4.4)$$

這裡的 $\alpha \in [0,1]$ ， 因此 $\beta^2 \in [0,\infty]$。通常會讓 precision 和 recall 有相同權重，故 β 值取 1，即為 F_1。

4.5 K-Means 與 Jieba 斷詞的參數實驗

本節將進行 K-Means 與 Jieba 參數調整的相關實驗，以先提升 K-Means 分群結果的穩定性後，再進行後續的其他實驗。

4.5.1 Jieba 斷詞三種參數的設定與比較

實驗的測試資料選取 2015 年 4 月 4 日透過 RB 所篩選出來的文章進行實驗，總共 267 篇文章，如圖 4.1 所示。

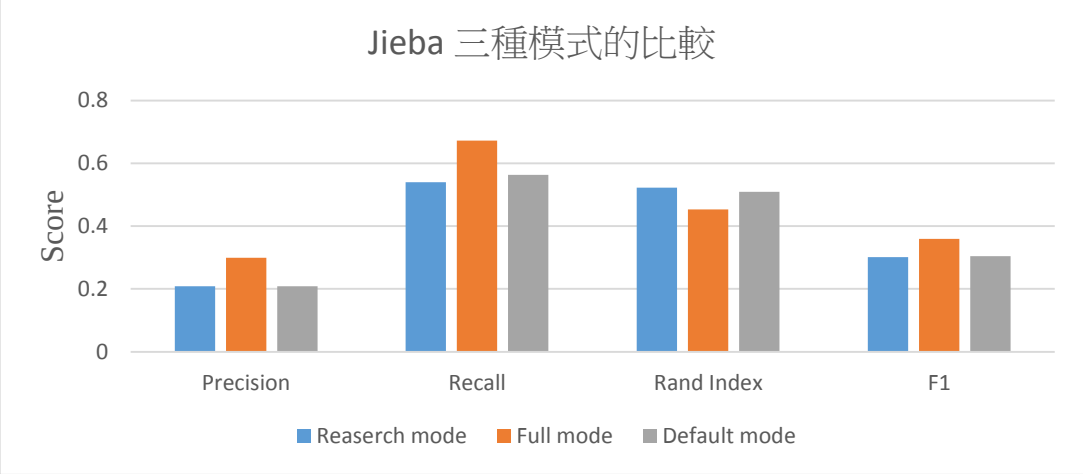

圖 4.1 Jeiba 斷詞系統的三種模式比較

如圖 4.1 所示，使用全模式的效果會比其他兩個模式表現較優，因為在 Recall 及 F1 的分數都較另外兩個模式來得好，所以本論文往後的實驗哩，斷詞模式皆使用全模式來執行斷詞運算。

4.5.2 如何決定群集數量 K

K-Means 分群演算法有個難題一直沒有很好的解決方法，就是如何決定群集 K 的數量，本論文的重點並不在解決如何取 K 值，本論文藉由調整 K 值的分群結果，比較對文章數量 N 來說，K 值取多少能夠得到最佳的結果後，就可以進行往後的實驗。

本實驗依據平均一百次的分群結果來評分，以降低隨機挑選初始中心點所造成的影響。

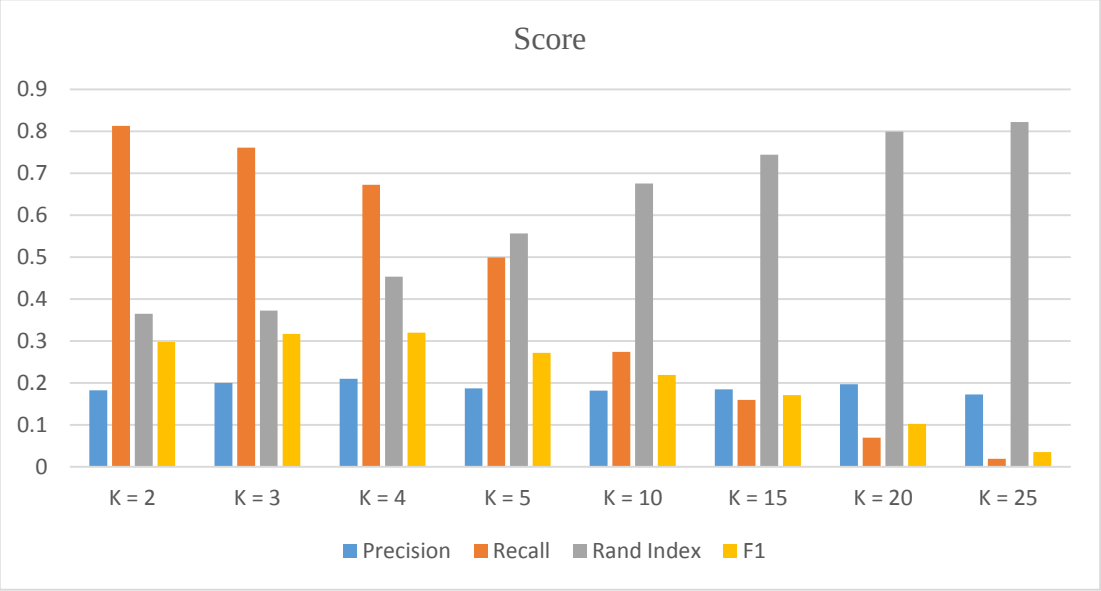

圖 4.2 K 值實驗結果

如圖 4.2 所示，Rand Index 的分數之所以隨著 K 值的增加而越來越高，是因為測試的文件數量只有 30 篇文章，所以分的群越多，會有越多群裡面只有一個文件標題的元素，對 Rand Index 來說，每群裡面只有一個文章算是正確的，所以 Rand Index 與 K 值成正比。但本論文所期望的分群結果是要兩兩文件互相比對，將有關聯的標題聚在一起。以本論文所要探討的熱門主題來說，藉由文章特徵將討論主題相似的文章分在同一群，同一群的至少一

篇文章以上才能夠構築成一個主題。所以 Rand Index 在本實驗對論文是沒有參考價值的。

我們可以發現 K 值取 3 與 4 時會有較好的表現，我們進行人工分群 30 篇以使用 RB 篩選文章時，大多數只能 4 群，與此實驗的結果相近，所以之後的實驗 K 值都取 4。

4.5.3 分群實驗

在 RB 的方式篩選文章後，我們藉由 Jeiba 斷詞系統，取得文章的特徵向量，並且進行分群，以下是分群實驗，以 2015 年 4 月 4 日所有文章進行實驗。如表 4.4 所示。

表 4.4 2015 年 4 月 4 日分群結果

第一群	
[爆卦] 政黑要選新版主了	10
[爆卦] 偉大的+long	10
[新聞] 號外！閃辭 18 天後　彭文正、李晶玉密談回	10
[新聞] F-18 降落台灣　美軍媒體：中國非常不爽	8
[問卦] 人生的意義到底是什麼？	8
[問卦] 設計一架戰鬥機有多難？	7
[新聞] 美 F-18 戰機離台時間未定　AIT：仍在維修	6
[問卦] 人類壽命還很難超過 100 歲的八卦	6
[問卦] 台灣男人吃魚喝茶的人真的很多嗎	5
[問卦] 為什麼突破不了光速??	5
[問卦] 有沒有太平天國的八卦?	5

第二群	
[問卦] 為何不正視女生不需服義務役的問題	27
[新聞] 正妹護航阿帕契姊　稱不知軍事重地禁拍照	14
[新聞] 〈快訊〉　李蒨蓉等人限制出境	14
[新聞] 軍地禁拍照打卡 陳艾琳為李蒨蓉緩頰:有誰	12
[新聞] 李蒨蓉案再踢爆 荒唐飛官戴阿帕契頭盔跑趴	9
[新聞] 【動新聞】扯！阿帕契荒唐飛官　遭爆戴	5
[新聞] 吳鳳感慨臺灣視野小 疾呼該關心大事	6
[問卦] 害人領不到月俸，這就是鄉民的正義？	5
[新聞] 「機棚非要塞」 軍方辯沒洩密	5
第三群	
[新聞] 中研院士：若柯 P 贊成亞投行　還有人吵嗎	8
[爆卦] 科 p 模式等於新加坡模式	5
[新聞] 年輕人還是租不起！ 柯 P 聯開宅最低租 8400 元	5
第四群	
[問卦] 女生愛身高 165 但年薪百萬的男人嗎？	14
[問卦] 正常台人被中資五倍挖角，會跳嗎？	14
[問卦] 35 歲月薪有 40k 還算魯蛇嗎?	11
[新聞] 就是不想生！六成五民眾怕養不起小孩	10
[問卦] 月薪領不到 3 萬的人真的這麼多?	9

4.6 關鍵字擷取

由於分群沒辦法達到百分之百的準確分群，所以本節將 4.5 節分群完的文章進行人工些微的調整，以免分群不佳的狀況影響後面的實驗結果，我們

對 2015 年 4 月 3 日以及 4 月 4 日兩天的實驗資料進行測試，並將一連串的主題偵測在下一節一併實現。

首先，我們對 2015 年 4 月 3 日所有文章，使用系統將每一群的前五名關鍵字擷取出來，並與人工選取的關鍵字做比較，如表 4.5 所示。

表 4.5 2015 年 4 月 3 日人工選取各群的關鍵字

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
ASUS	阿帕契	套房	小孩	美國
手機	李蒨蓉	8400 元	工作	戰機
郵政	航特部	聯開宅	薪水	AIT
科系	勞乃成	亞投行	女生	飛機
歷史	601 旅	柯 P	百萬	引擎

在此的人工選取關鍵字是以三個人分別對各群進行關鍵字選取，並將選取的答案做比較後取交集，使人工選取關鍵字的誤差降至最低。而人工選取沒有排名之分，只要這個關鍵字能夠代表這個群就選出來，以不考慮排名的方式對應至系統找出的關鍵字，只要關鍵字有出現，就算正確，如表 4.6 為系統選取關鍵字以及與人工選取的關鍵字比對後計算得來的準確率。

表 4.6 2015 年 4 月 3 日人工選取的關鍵字與系統選取的關鍵字比對

群	C1		C2		C3		C4		C5	
排名	關鍵字	PLS	關鍵字	PLS	關鍵字	PLS	關鍵字	PLS	關鍵字	PLS
1	老師	0.016	阿帕契	0.039	套房	0.027	小孩	0.023	美國	0.052
2	國民黨	0.010	李蒨蓉	0.026	8400	0.027	工作	0.021	戰機	0.046
3	歷史	0.010	601	0.026	亞投行	0.024	薪水	0.017	一架	0.045
4	問題	0.009	航特部	0.024	台北	0.024	問題	0.014	飛機	0.032
5	美國	0.008	陸軍	0.023	柯 P	0.022	公司	0.014	引擎	0.303
準確率	0.2		0.8		0.6		0.6		0.8	

以 C2 (第二群) 來說，只有“勞乃成”沒有出現在系統選取的關鍵字裡，其他關鍵字正確，所以準確率為 80%，其他群依此方法辦理。

在此我們觀察到除了第一群之外，其他群的準確率都在 60% 以上，這是因為每天批踢踢八卦版發佈的主題類型有百種以上，除了我們分群出來的話題之外，有些話題雖然有機會成為熱門主題，但是他並不夠熱門，網路的使用者雖然有討論，但是討論的熱度沒有辦法跟其他的話題所比較，我們稱這些話題是“分群後的雜訊”，我們將這些雜訊全部群聚至 C1 (第一群)，但仍然將第一群的關鍵字擷取出來。所以我們會發現第一群的跟標準答案比較的準確率會比較低，因為太多不同的話題都在這一群。接下來我們觀察 2015 年 4 月 4 日的關鍵字擷取與比對，如表 4.7 所示。

表 4.7 2015 年 4 月 4 日系統選取的關鍵字與人工選取的關鍵字比對

	C1		C2		C3		C4		C5		C6	
排名	關鍵字	PLS	關鍵字	PLS	關鍵字	PLS	關鍵字	PLS	關鍵字	PLS	關鍵字	PLS
1	美國	0.017	李蒨蓉	0.059	外省人	0.091	互聯網	0.027	日本	0.055	意義	0.054
2	小型	0.017	阿帕契	0.045	本省	0.071	工作	0.025	中國	0.034	人生	0.046
3	預算	0.016	拍照	0.028	外省	0.051	老師	0.015	飛彈	0.025	死亡	0.045
4	議員	0.014	勞乃成	0.024	基層	0.046	女生	0.013	美國	0.018	自己	0.036
5	電視	0.014	藝人	0.018	權貴	0.039	企業	0.013	統一	0.014	人類	0.025
準確率	0.2		0.8		0.6		0.6		0.6		0.6	

我們可以發現，表 4.7 比表 4.6 多了 C6，代表 4 月 4 日當天多了一個討論的話題。除此之外，由於我們將分群後的雜訊全部放至 C1 (第一群)，所以 C1 的關鍵字變動幅度相當的大。C2 我們可以發現與 4 月 3 日的關鍵字有相同之處，透過人工驗證得知話題有延續性，接下來 C3 的關鍵字跟前一天完全不同，這時可以說，表 4.6 的 C3 被另一個話題所取代，所以關鍵字的部份也都不相同了，C4 與 C5 皆與 C3 的狀況相同。

4.7 主題偵測

藉由上一節的關鍵字擷取，我們將每一群藉由前五名的關鍵字命名成一個主題。以上一節表 4.7 為例，如圖 4.3 所示。

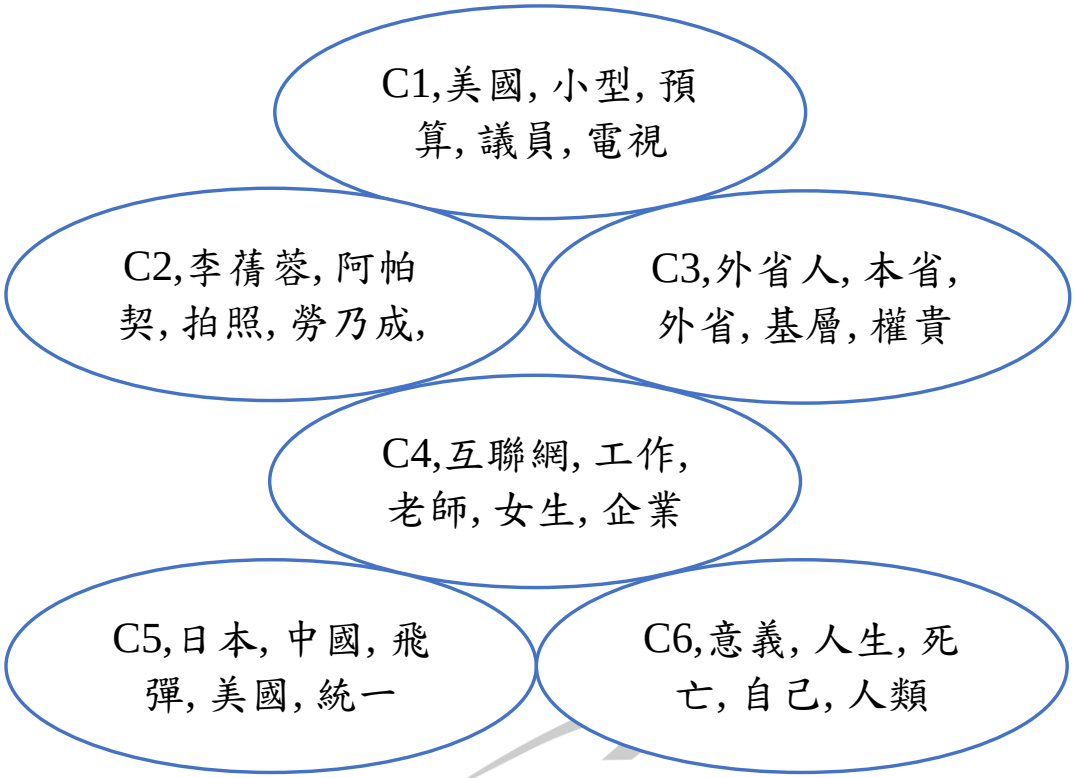

圖 4.3 主題偵測結果

圖 4.3 已經將 4 月 4 日的六大主題偵測出來，接下來本論文所關心的是，哪些主題忽然被大眾所討論，以及使用第三章所設計的方法去觀察主題的走向以及討論熱度的延續性。

4.7.1 主題熱度追蹤實驗

我們會針對多個主題去實驗公式 3.3 是否能夠有效的追蹤熱門主題。透過 4.6 節對 4 月 3 日以及 4 月 4 日關鍵字擷取，我們繼續使用公式 3.3 追蹤這個主題，如表 4.8 所示。

表 4.8 追蹤篩選後 Topic 2 阿帕契主題 PLS 變化的情況

日期 關鍵字	4/2	4/3	4/4	4/5	4/6	4/7	4/8	4/9	4/10	4/11	4/12	4/13
阿帕契	0	0.039	0.045	0.025	0.037	0.046	0.030	0.028	0.029	0.019	0.009	0
國軍	0	0	0	0.028	0.023	0.014	0.009	0.030	0.041	0.047	0.031	0.015
李蒨蓉	0	0.027	0.060	0.018	0.012	0.019	0.012	0.006	0.026	0.017	0.008	0
軍人	0	0	0	0	0	0	0.072	0.047	0.023	0.024	0.016	0.008
勞乃成	0	0	0.024	0.016	0.008	0.019	0.012	0.033	0.029	0.019	0.009	0
航特部	0	0.024	0.016	0.008	0	0	0.042	0.028	0.014	0	0	0
陸軍	0	0.024	0.015	0.007	0.032	0.021	0.010	0	0	0	0	0
601	0	0.026	0.017	0.008	0	0	0	0.02	0.013	0.006	0	0
懲處	0	0	0	0.016	0.027	0.018	0.009	0	0	0	0	0

由於每一個主題會出現非常多的關鍵字，為了方便表格以及圖表的觀察，所以我們將關鍵字只出現一次的關鍵字濾除掉，往後的主題追蹤實驗也是如此。我們可以發現，阿帕契主題從 4 月 3 日開始被大眾廣泛地討論，主題持續討論的途中，冒出許多未曾出現的關鍵字，代表主題討論的核心不斷地變化。舉例來說，”李蒨蓉” 這個關鍵字在 3 日出現，在隔日討論更加的熱烈，但是在 5 日以及 6 日 分數明顯的大幅下降，在 5 日下降的當天，出現新的關鍵字 “國軍” 以及 “懲處”，這剛好驗證了主題的核心會因為新的核心關鍵字出現而減少討論度以及同一主題的每個關鍵字並不會一直被討論。而是隨著主題的發展以及大眾關注主題核心的變化，牽動著關鍵字起伏，在此我們將表 4.8 圖表化，如圖 4.4 所示。

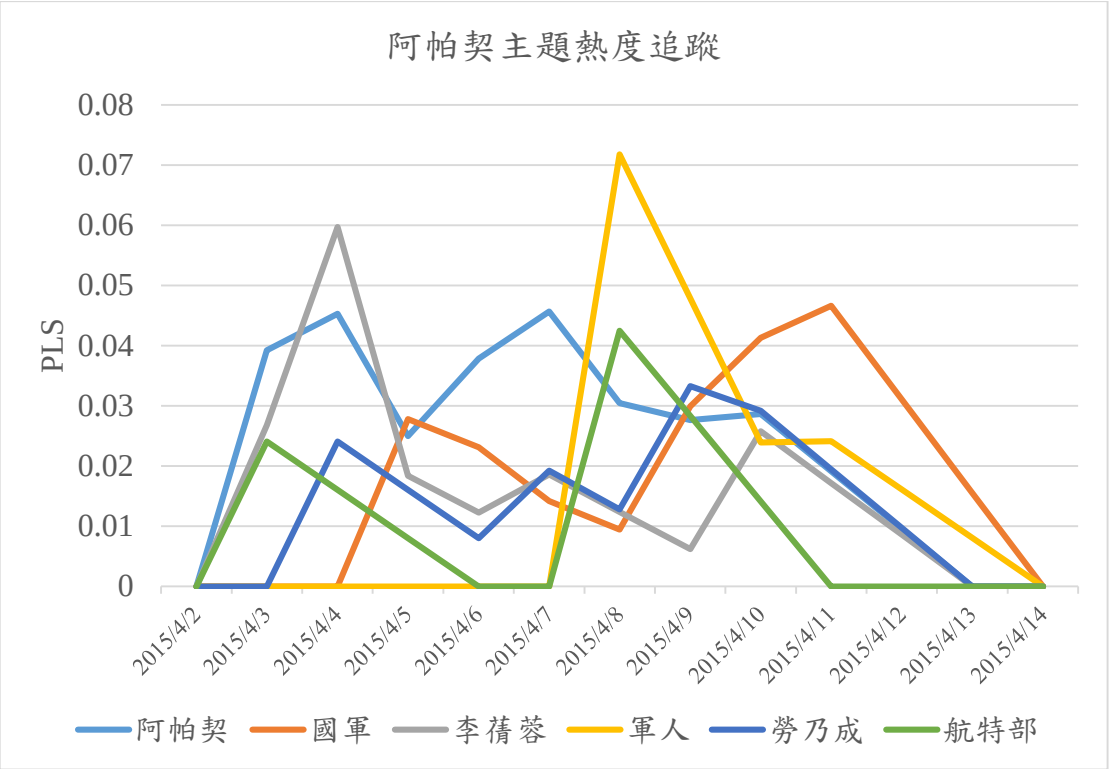

圖 4.4 阿帕契主題熱度追蹤

縱軸是關鍵字的 PLS 分數，橫軸是日期。經過圖像化，我們更加確定同一主題的核心關鍵字隨著時間，大眾們不斷改變討論方向，我們也可以觀察到，透過公式 3.3 保留關鍵字的熱度是具有意義的。”國軍”在 5 日之後討論度持續下降，但藉由公式 3.3 保留此關鍵字的熱度，直到 9 日時這個關鍵字又再被大眾所注意，一直到 11 日達到最高峰。

阿帕契主題在 3 日出現，一直到 11 日之後，大多關鍵字已經消失，討論曲線因為公式的關係所以線性的下降，我們可以知道，這個主題被大眾討論 3 日至 11 日，分數在 11 日之後線性的慢慢在 14 日時全部消失，這個主題總共被大眾關注 9 天。

在主題被偵測到的這幾天，我們對所有關鍵字的排名也非常地感興趣，藉由公式 3.4 我們可以將關鍵字圖像化，如圖 4.5 所示。

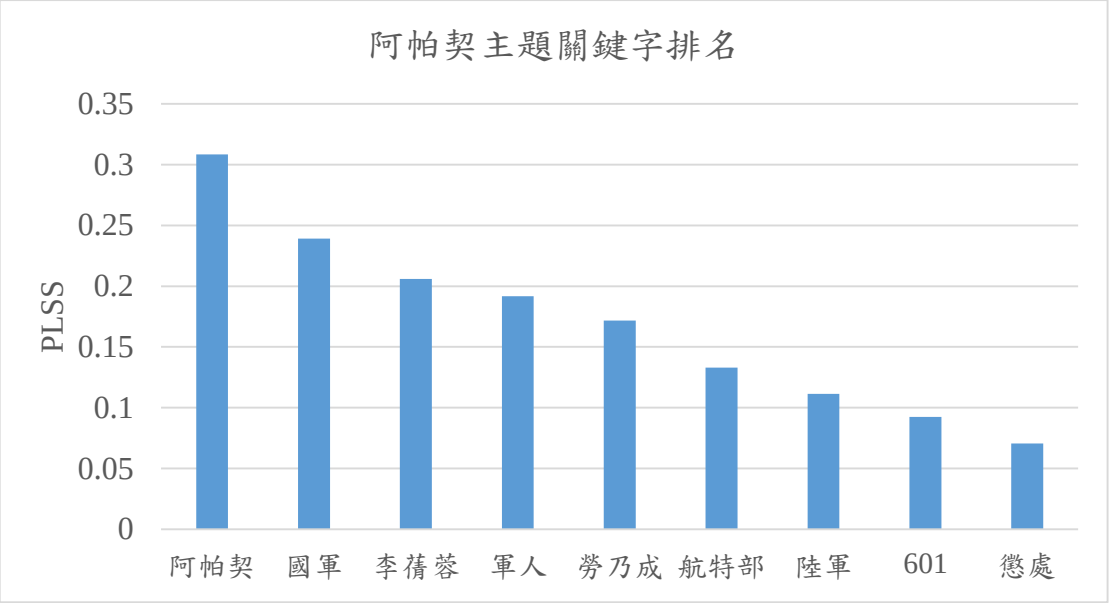

圖 4.5 阿帕契主題關鍵字排名

透過圖 4.5 我們可以將關鍵字再次收斂，阿帕契的主題核心分別是”阿帕契”、”國軍”、”李蒨蓉”、”軍人”、”勞乃成”，這五個關鍵字分別排行一到五名，這五個關鍵字是阿帕契主題最主要的核心關鍵字。接下來我們採用 4 月 9 日至 11 日的文章來繼續實驗。選取其中的一個主題“台中捷運工程意外”來觀察，如表 4.9 所示。

表 4.9 追蹤篩選後台中捷運主題 PLS 變化的情況

日期 / 關鍵字	4/9	4/10	4/11	4/12	4/13	4/14
捷運	0	0.037	0.044	0.029	0.0145	0
林佳龍	0	0.026	0.043	0.029	0.0143	0
工程	0	0.033	0.034	0.023	0.011	0

顯而易見地，這個主題關鍵字的數量相較“阿帕契主題”少了很多，有可能是因為持續天數沒有那麼的長。接下來我們將表 4.9 圖表化，如圖 4.6 所示。

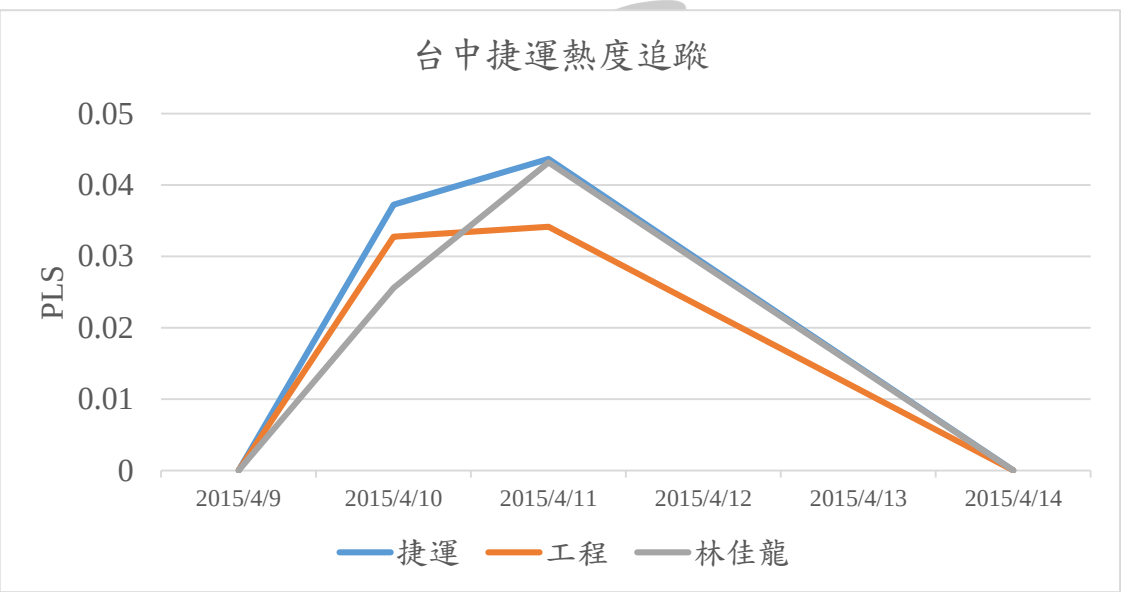

圖 4.6 台中捷運主題熱度追蹤

對此主題來說，主題圍繞在”捷運”、”工程”、”林佳龍”這三個關鍵字上，本節之後的關鍵字曲線圖皆使用篩選過後的圖來呈現，接下來讓我們用公式 3.4 的角度來觀察，如圖 4.7 所示。

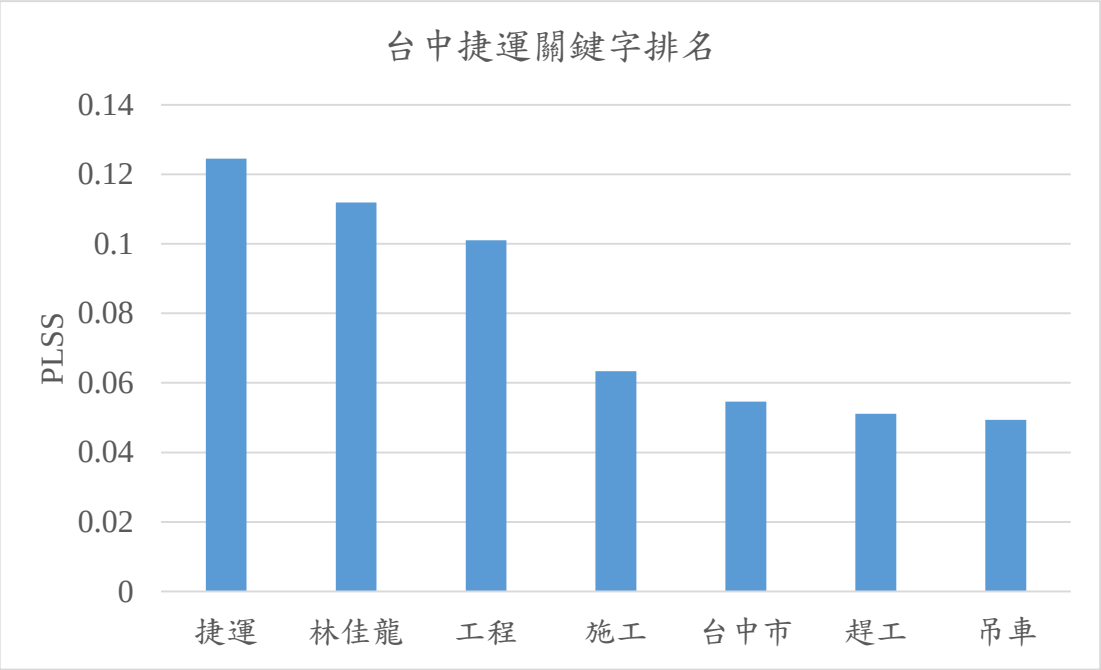

圖 4.7 台中捷運主題關鍵字排名

我們將此主題收斂至前五個關鍵字，分別是”捷運”、”林佳龍”、”工程”、”施工”、”台中市”，經過人工驗證，這五個關鍵字的確能代表這個主題真正的核心。接著我們對 4 月 22 日至 4 月 29 日的主題進行追蹤，如表 4.10 所示。

表 4.10 追蹤篩選後網路霸凌主題 PLS 變化的情況

日期 / 關鍵字	4/21	4/22	4/23	4/24	4/25	4/26	4/27
霸凌	0	0.098	0.093	0.05	0.085	0.057	0.028
網路	0	0.06	0.085	0.031	0.112	0.075	0.037
鄉民	0	0.021	0.0140	0.007	0.023	0.0154	0.007
楊又穎	0	0	0.026	0.027	0.017	0.008	0

接下來我們對表 4.10 進行圖表化，更進一步的觀察。

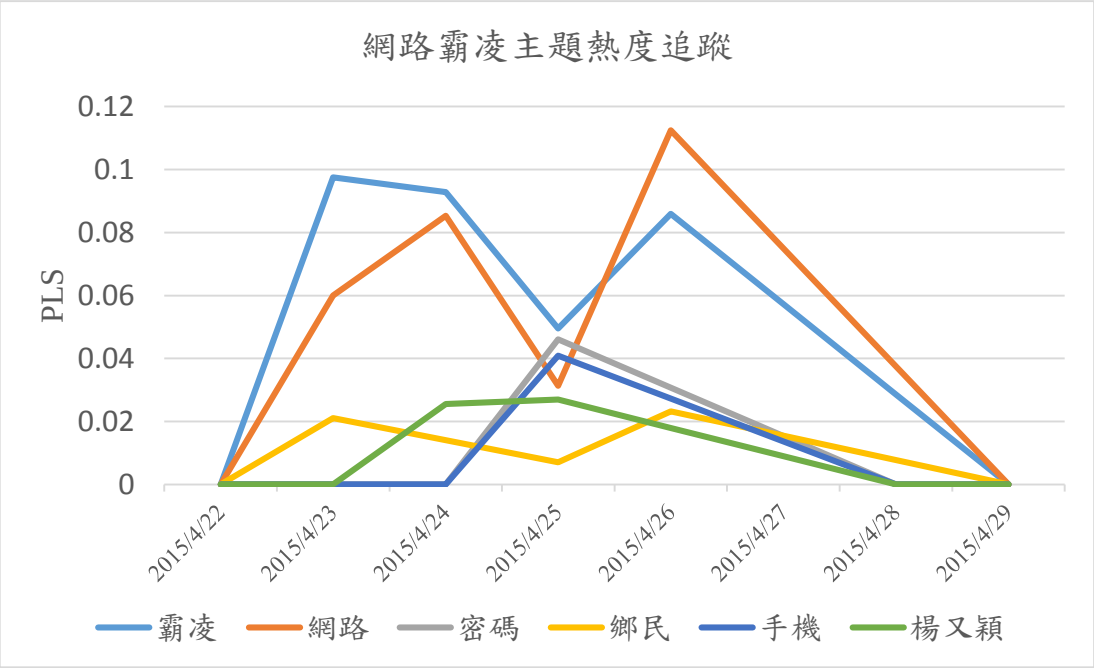

圖 4.8 網路霸凌主題熱度追蹤

由圖 4.8 得知，”網路”與”霸凌”這兩個關鍵字很明顯示這個主題的核心，因為他不斷地被大眾所討論。我們將這個主題持續期間的關鍵字作排名，如圖 4.9 所示。

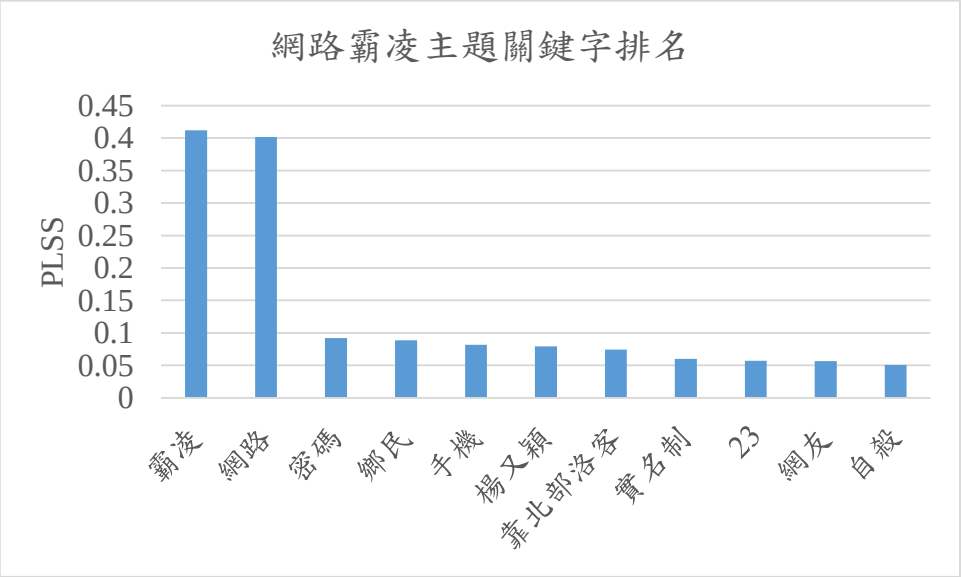

圖 4.9 網路霸凌主題關鍵字排名

這個主題幾乎都是圍繞著“霸凌”與“網路”兩個核心在討論，接下來是“密碼”、”鄉民”以及“手機”這三個關鍵字。

接下來我們對 6 月 19 日至 22 日的主題進行分析與追蹤。如表 4.11 所示。

表 4.11 追蹤篩選後學生募款主題 PLS 變化的情況

日期 / 關鍵字	6/21	6/22	6/23	6/24	6/25	6/26	6/27
募款	0	0.023	0.0150	0.056	0.037	0.018	0
社會	0	0.019	0.0127	0.026	0.017	0.008	0
領導	0	0.015	0.038	0.019	0.012	0.006	0
學生	0	0	0.057	0.035	0.023	0.011	0
25	0	0	0.03	0.022	0.014	0.007	0

這個主題討論的核心跟表 4.10 的主題相比，討論核心比較集中，因為我們發現兩個主題討論的時間長度差不多，但是“學生募款主題”的關鍵字不需藉由公式保持關鍵字的熱度，而是不斷提及已經出現的關鍵字，代表 PTT 的使用者們在此主題討論的核心比較集中。接著我們將表 4.11 圖表化，如圖 4.10 所示。

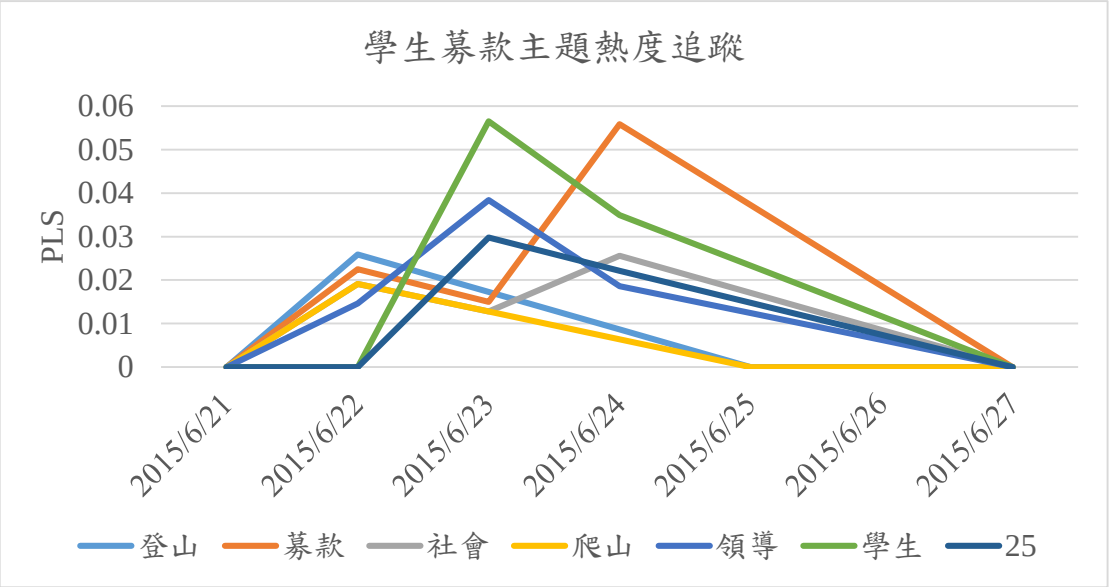

圖 4.10 學生募款主題追蹤

我們從圖 4.10 可以得知，”登山”與”爬山”這兩個詞表達雖然是相同的意思，但是斷詞系統無法分辨，所以還是算兩個詞彙。這個主題討論的關鍵字比較集中在”學生”和 ”募款”這兩個詞彙上，主題是從 22 日至接下來對主題持續時間的關鍵字作排名，如圖 4.11 所示。

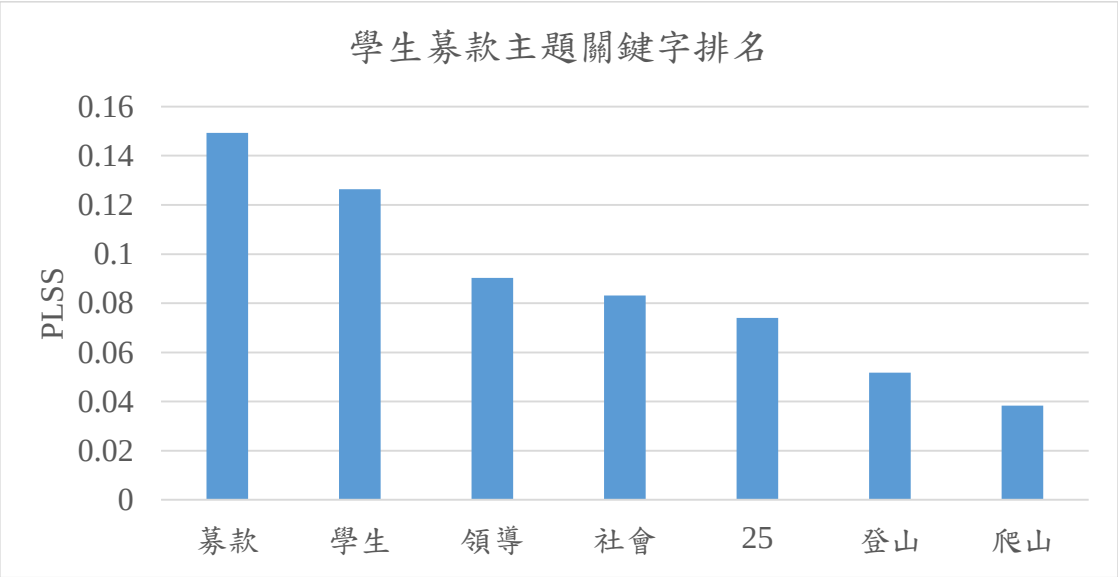

圖 4.11 學生募款主題關鍵字排名

這個主題的關鍵字排名與其他新聞主題一樣，透過人工驗證，前五名的關鍵字 “募款”、”學生”、”領導”、”社會”、”25” 能夠正確地表達主題的核心。接下來我們在對另一個 6 月 27 日至 6 月 30 日的主題進行分析，如表 4.12 所示。

表 4.12 同性婚姻主題關鍵字排名

日期 / 關鍵字	6/26	6/27	6/28	6/29	6/30	7/1
支持	0	0.041	0.039	0.025	0.012	0
同志	0	0.038	0.025	0.0128	0.004	0
KMT	0	0.027	0.017	0.008	0.002	0
婚姻	0	0.025	0.016	0.008	0.002	0
平權	0	0.021	0.014	0.007	0.002	0

在表 4.12 中，我們可以看到” 支持” 、” 同志” 、” KMT” 、” 婚姻” 、” 平權” 這幾個關鍵字討論的熱度是比較高的，圖表化後如圖 4.12 所示。

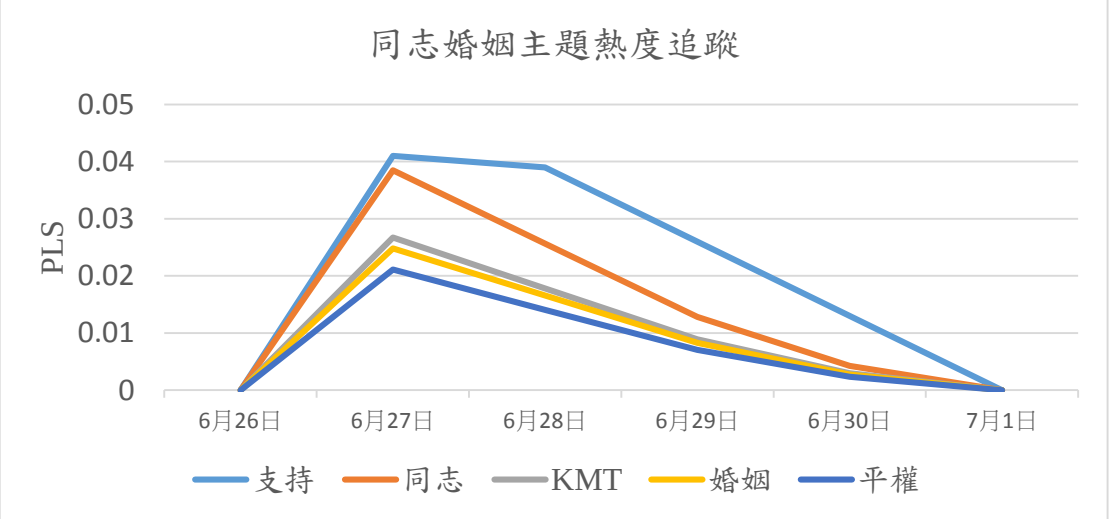

圖 4.12 同志婚姻主題追蹤

接下來我們對這個主題所有關鍵字進行排名，如圖 4.13 所示。

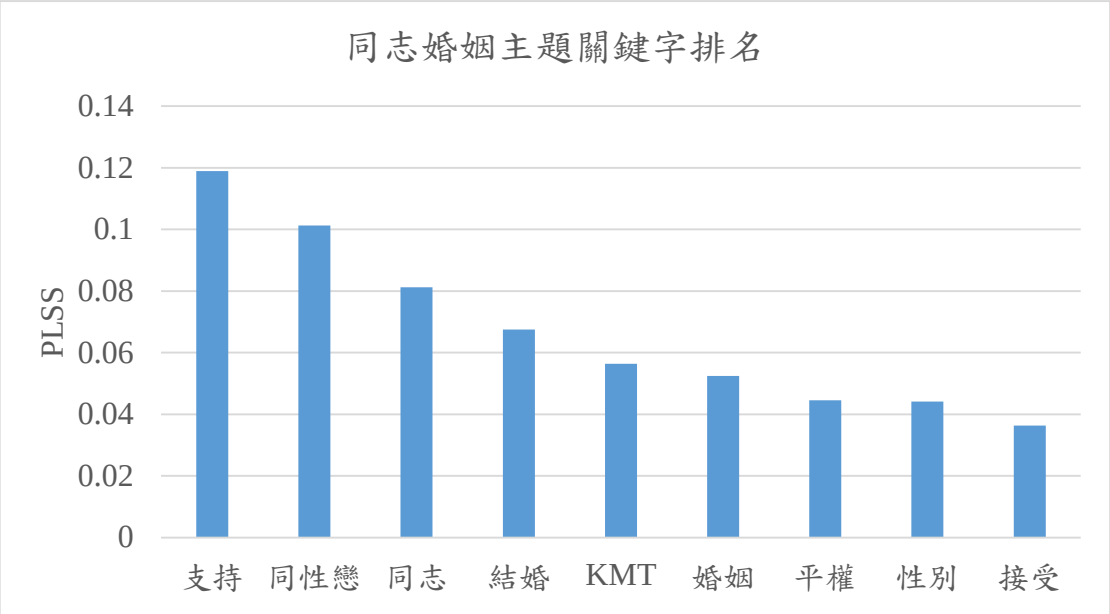

圖 4.13 同性婚姻主題-關鍵字排名

透過人工驗證，同性婚姻主題的前五名關鍵字能夠正確地代表這個主題的核心。接下來，我們再對 6 月 27 日另一個主題進行追蹤與分析，如表 4.13 所示。

表 4.13 八仙塵爆主題關鍵字

日期 / 關鍵字	6/26	6/27	6/28	6/29	6/30	7/1	7/2	7/3	7/4	7/5
塵爆	0	0.034	0.058	0.046	0.031	0.013	0.008	0.004	0	0
八仙	0	0.034	0.022	0.021	0.021	0.059	0.014	0.018	0.055	0.056
粉塵	0	0.028	0.055	0.045	0.030	0.015	0	0	0	0
彩色	0	0.019	0.012	0.006	0.004	0.002	0	0	0	0
病患	0	0.017	0.011	0.005	0.003	0.001	0.016	0.010	0.005	0
醫院	0	0	0	0.025	0.039	0.059	0.039	0.019	0	0
醫療	0	0	0	0.017	0.025	0.016	0.008	0.029	0.019	0.009
燒傷	0	0	0	0.017	0.022	0.014	0.007	0	0	0
家屬	0	0	0	0	0.018	0.012	0.006	0.025	0.016	0.008

我們初步可以知道，這個主題架構算是非常龐大，持續的時間久，關鍵字的數量多又雜，接下來將表 4.13 圖表化，如圖 4.14 所示。

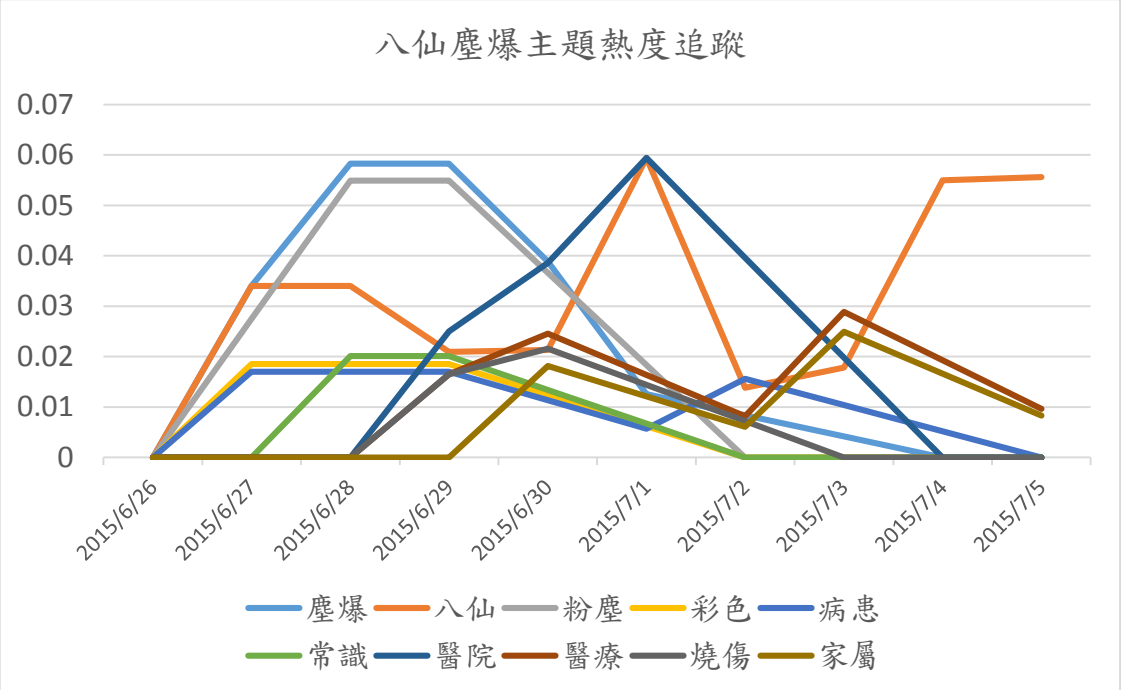

圖 4.14 八仙塵爆主題熱度追蹤

藉由觀察關鍵字的數量，可以了解這個八仙塵爆主題討論範圍非常的廣，大眾使用者討論的關鍵字不斷地轉變，每一天幾乎都有出現新的關鍵字，由於本論文撰寫時間的關係，我們並沒有 7 月 5 日以後的文章，但是經由人工驗證後得知此主題仍在繼續。“八仙”、”塵爆”、”醫療” 很明顯常被提及，接下來將主題期間的所有關鍵字作排名，如圖 4.15 所示。

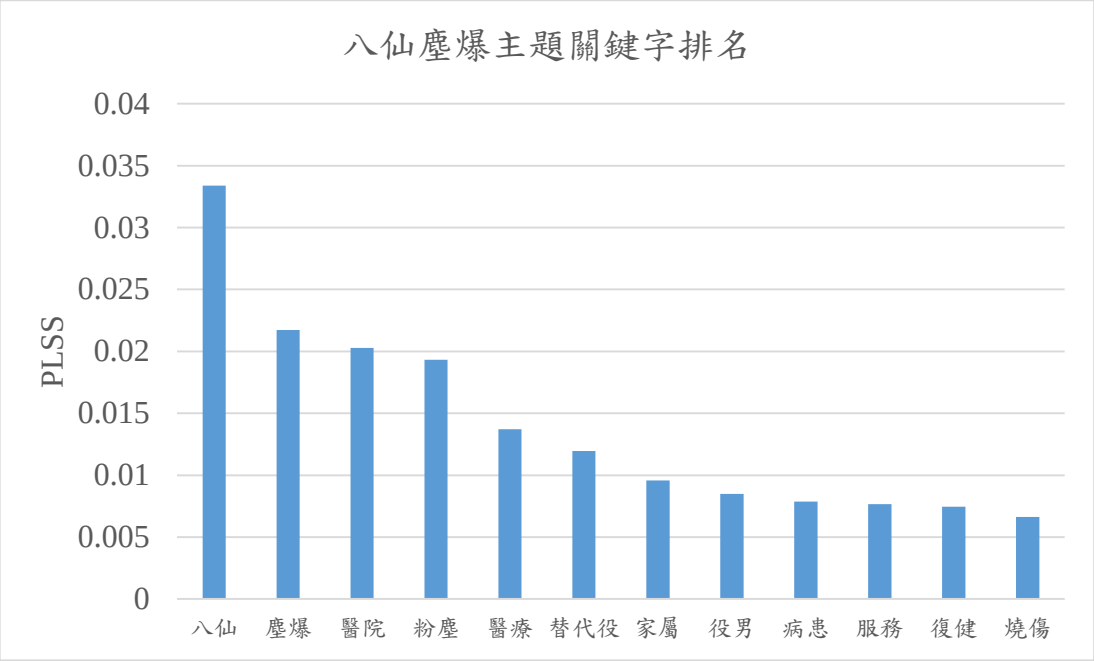

圖 4.15 八仙塵爆主題關鍵字排名

透過主題討論的期間對關鍵字作排名，經由人工驗證，”八仙”、”塵爆”、”醫院”、”粉塵”以及”醫療”都能夠十足地代表八仙塵爆主題。

4.7.1 節我們對大量的文章進行實驗，得出許多的主題，並且分析各種主題以及追蹤主題的熱度，在 4.7.2 節當中，我們將會透過網路蒐集本節的實驗主題對照的參考答案作比對。

4.7.2 主題命中率驗證

本節會將 4.7.1 所偵測出來的主題跟蘋果日報、中國時報以及自由時報這三個主要的報章頭條進行比對。以 2015 年 4 月 3 日 的三大報紙頭條與偵測出的主題作配對，如表 4.14 所示。

表 4.14 4 月 3 日三大報紙頭條命中率驗證

20150403	蘋果日報	中國時報	自由時報
Topic 1 老師、國民黨、歷史、問題、美國	NA	NA	NA
Topic 2 阿帕契、李蒨蓉、601、航特部、陸軍	● 白目李蒨蓉挺老公上酒店 ● 腦殘飛官帶女星闖禁地 李蒨蓉爽玩阿帕契 ● 出身將門 種子教官 凸槌中校曾赴美 ● 國軍性醜聞 將領帶頭作亂	● 李蒨蓉全家登阿帕契 尖叫喊酷 ● 洩軍情 李蒨蓉恐觸法 桃檢要辦 ● 李蒨蓉輕易登機 勞乃成中校妻牽線	● 軍紀渙散 飛官帶藝人李蒨蓉上阿帕契 ● 桃檢要辦人 李蒨蓉改口：我白目，我道歉 ● 李蒨蓉案 桃檢：涉違反要塞堡壘地帶法 ● 阿帕契頭盔 造價兩百萬
Topic 3 套房、8400、亞投行、台北、 柯 P	● 毛：亞投行 22 億不會成壁紙	● 毛揆打臉財長：絕不會變壁紙	● 台入亞投行 經部坦承：影響與美日關係 ● 《 加入亞投行 決策爭議大》台左維新：馬在愚人節開最大玩笑 ● 22 億入亞投行變壁紙？ 毛揆急滅火 ● 財長：寫上國名 對方連收件都不會收
Topic 4 小孩、工作、薪水、問題、公司	NA	NA	NA
Topic 5 美國、戰機、一架、飛機、引擎	● 維修 F-18 美軍送發動機抵台	● 華府專家：F-18 降落台灣	● F-18C 原本任務 掩護 EA-6B 飛往星國

		有政治意涵	
未命中	● 李全教賄選 1500 萬交保滾雪球 ● 再爆 3 胃藥摻工業鎂	● 吳釗燮：美盼蔡英文訪問成功 ● 阿茲海默症早知道徒增擔心？ ● 1cc 驗血幫你查出老人失智症 ● 台南議會賄選案 李全教 1500 萬交保	● 台師大獨步全球 抽血 就可檢測阿茲海默症
命中	6	5	9
未命中	2	4	1
命中率	0.750	0.555	0.900
平均命中率	0.735		

本論文將偵測出來的主題透過人工驗證與三大報紙焦點新聞進行比對，只要主題的關鍵字與新聞內容相關我們判定為命中，以蘋果日報來說，當天焦點新聞有八則，Topic 2 命中的是”白目李蒨蓉 挺老公上酒店”、”腦殘飛官帶女星闖禁地 李蒨蓉爽玩阿帕契”、”出身將門 種子教官 凸槌中校曾赴美”這三則新聞，Topic 3 命中的是”毛：亞投行 22 億不會成壁紙”，Topic 5 命中的是”維修 F-18 美軍送發動機抵台”，所以總共命中 5 則新聞，其他 3 則未命中。

在此我們定義命中率（Hit Rate）為：(命中的新聞／(命中的新聞 ＋ 未命中的新聞))，舉例來說，蘋果日報當天總共有 8 則焦點新聞，所以命中率

是 0.750。其他兩個報章媒體也是利用相同方法去驗證，中國時報分別是 0.5560，自由時報是 0.9000，將三大報的命中率平均取值得到 0.735 則是作為當天主題驗證的分數。

我們也注意到 Topic 1 沒有命中的狀況，原因是 Topic 1 裡面討論的內容非常複雜，因為“分群後的雜訊”都在這個主題之中，但是並不代表 Topic 1 沒有命中的機會，因為裡面還是由具有潛力成為熱門主題的文章。Topic 4 經過人工驗證，是 PTT 使用者在 PTT 發起的討論，沒有命中到任何新聞是因為這個主題是網路使用者發起的討論，並不是有這篇新聞所以大眾開始關注的。接下來讓我們驗證 4 月 11 日的主題與新聞的命中狀況，如表 4.15 所示。

表 4.15 4 月 11 日三大報紙頭條與偵測出的主題進行驗證

20150411	蘋果日報	中國時報	自由時報
Topic 1 警察、鄉民、日本、民生、紅衛兵	NA	NA	NA
Topic 2 國軍、軍人、廢掉、郁慕明、廢物	● 209 噸鋼梁墜 12 米 像地震	NA	● 大小眼？鴻禧山莊設供水站 民罵媚富
Topic 3 捷運、林佳龍、工程、施工、台中市	● 「沒法閃」婦轎車壓扁 枉死	● 中捷趕工惹禍 4 大致命疏失 ● 台中捷運工安意外 209 噸鋼	● 台中捷運施工 209 公噸鋼樑砸落 4 死 4 傷

	● 中捷 鋼梁砸死 4 人 玩命趕工 竟無… ● 吊臂歪斜支架也插進路面 ● 包商生前抱怨趕工壓力大 ● 趕工撤圍籬搶早 2 年完成	梁砸落 4 死 4 傷 ● 突見鋼梁掉鐵片…車閃撞民宅逃死劫	● 偏要白天施工？北捷：曲線樑密合難度高 ● 209 噸鋼樑天降 路過車瞬成鐵餅 ● 要命誤判 太早鬆開吊樑鋼索 ● 交管未完全封路 禍從天降
Topic 4 第一名、同學、班上、高中、倒數	NA	NA	NA
未命中		● 陳德銘喊話：大陸市場屬台灣同胞勿放棄	● 搶食中客 中國紅二代來台開旅館
命中	6	3	5
未命中	0	1	2
命中率	1	0.75	0.7142
平均命中率	0.8214		

我們可以藉由表 4.15 得知，Topic 2 與表 4.15 裡的 Topic 2 的關鍵內容相似，透過人工驗證去觀察主題裡面的文章，發現討論的內容是在同一主題，代表這個主題從 3 日繼續被 PTT 的網路熱烈地討論到 11 日，但 11 日並沒有命中任何新聞，討論熱門程度很明顯地退燒許多，從三大報章媒體蒐集而來的新聞也沒有出現有關 Topic 2 的報導。Topic 3 我們推測它是當天最熱門的主題，透過命中當天 14 則新聞，佔了所有新聞的 82.14 % 的新聞，也驗證我們偵測出來的主題的確被大眾所廣泛地討論。

接下來我們將 4 月 3 日至 4 月 11 日的主題驗證準確率計算出來，如圖 4.16 所示。

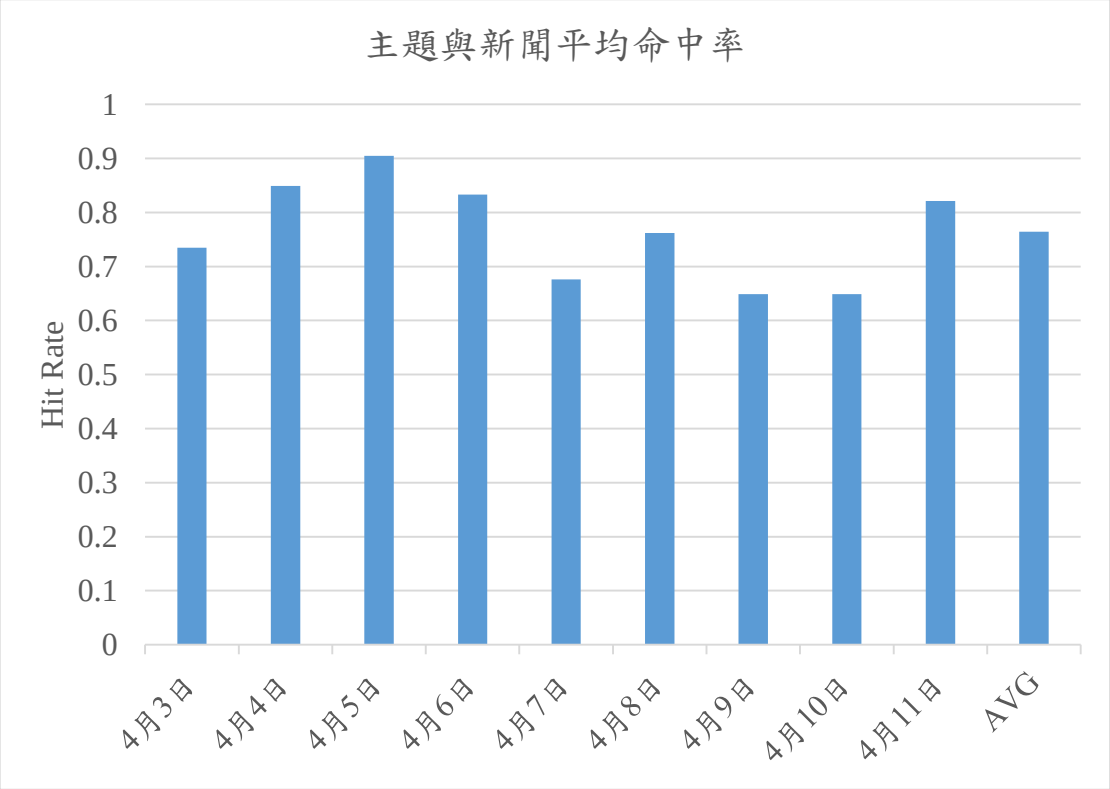

圖 4.16 4 月 3 日至 4 月 11 日的主題與新聞平均命中率

由圖 4.16 可以得知，透過本論文的研究方法， 4 月 3 日至 4 月 11 日的平均準確率為 76.43%。接下來我們來觀察 6 月 27 日至 7 月 5 日與新聞驗證的準確率，如圖 4.17 所示。

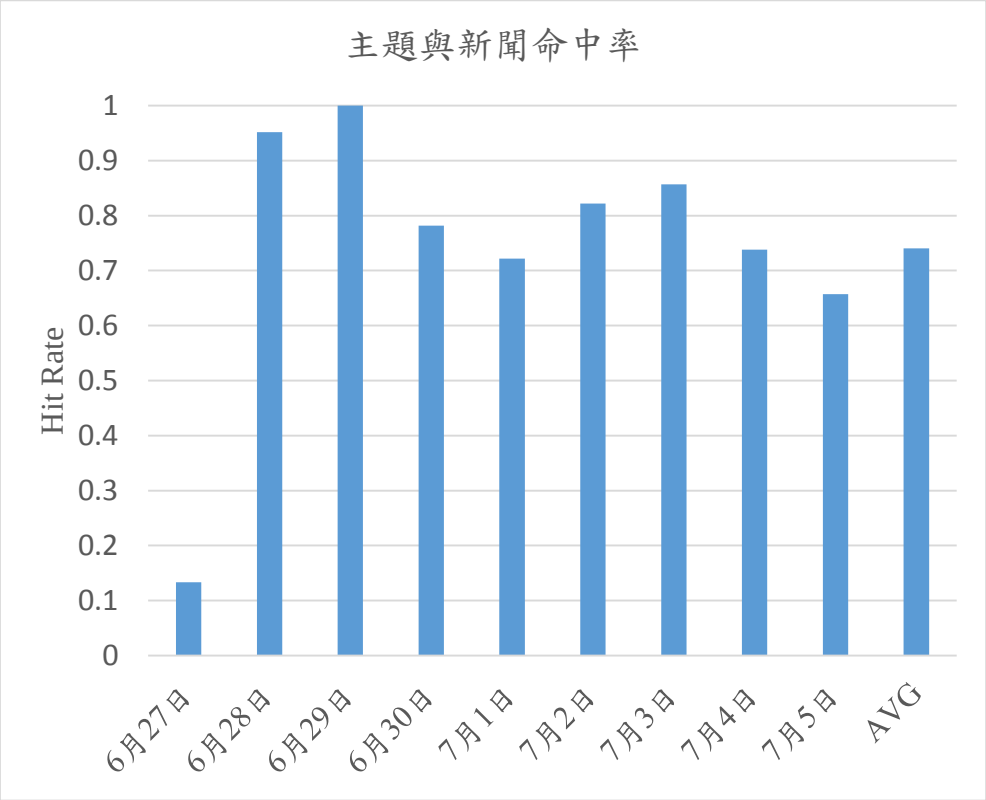

圖 4.17 6 月 27 日至 7 月 5 日的主題與新聞平均命中率

6 月 27 日的準確率非常低，原因是我們主要是從 PTT 偵測目前那些主題被多數的使用者們熱烈討論，討論的主題多半是當天新聞以及 PTT 使用者發起的討論，當天”八仙塵爆”主題是在晚上 8:50 時發生，這個主題在 3 個小時半的時間內熱列地被 PTT 使用者所討論，而媒體隔天才會大量報導有關此主題的新聞，所以與當天新聞驗證比較的分數不盡理想，因為事件發生在三大報頭版的時間之後，6 月 28 日分數非常的高，可以說明 27 日的”八仙塵爆”的主題變成 28 日的頭版與焦點新聞，而平均準確率也達到 74.03%，這也驗證了，使用本論文的研究方法能夠確實有效地偵測出熱門主題。

4.7.3 討論

表 4.15 的 Topic 4 是當天被偵測出的主題，但它並沒有命中 4 月 11 日任何一則新聞，而我們從網路收集一些新聞如表 4.16 所示。

表 4.16 Topic 4 成為新聞以及綜藝節目主題

日期	來源	URL
2015/04/11	蘋果日報	http://www.appledaily.com.tw/realtimenews/article/new/20150411/590660/
2015/04/14	自由時報	http://news.ltn.com.tw/news/society/breakingnews/1286563
2015/04/15	蘋果日報	http://www.appledaily.com.tw/realtimenews/article/new/20150415/592783/
2015/04/30	Youtube-大學生了沒	https://www.youtube.com/watch?v=ZJzkWG-WxwA

由表 4.16 得知，本論文偵測出 Topic 4 的當天，蘋果日報將它作為新聞，並且持續到 4 月 15 日。而自由時報也將此主題在 4 月 14 日時做成一篇新聞，甚至綜藝節目“大學生了沒”也因為這個主題受到大眾們的關注與討論所以在 4 月 30 日時以 Topic 4 製作相關的主題。由此也驗證了本論文除了能偵測 PTT 的使用者討論的熱門主題之外，也能從 PTT 偵測到非新聞的熱門主題，並且這個熱門主題也有機會成為一篇新聞。

第五章 結論與未來展望

5.1 結論

BBS 雖然是一個比較過時的社群討論平台，但對台灣來說，儘管目前新興的社群網站如 Facebook、Plurk 以及 Twitter 這些比 PTT 更容易使用且使用者介面更優化的情況下，PTT 反而藉由這些社群網站讓自己更加壯大，吸引大量的使用者參與，可能跟台灣本土文化風情有相關聯，也或許是 PTT 匿名與使用者的個人資訊極少的情況下，偏向匿名發文，所以有許多極具爭議性及討論性的話題被使用者們所討論，使得 PTT 成為本論文的研究對象。

本論文提出了一種方法，先選取具有熱門主題潛力的文章，並藉由斷詞系統擷取文章的特徵向量，再以分群將相關的討論文章聚在一起，進行熱門主題偵測，並更進一步地追蹤熱門主題，能夠有效地在 BBS 上實現。經過實驗證明，本次研究提出的方法能夠偵測到目前網路使用者正熱烈討論地熱門主題，準確率平均能達到 70% 以上。

本次研究提出的方法能夠在 PTT 偵測到的熱門主題幾乎都是時下熱門的新聞，代表當有重大主題發生時，PTT 的使用者會熱烈地討論；而有些熱門主題是從 PTT 發起，被新聞媒體關注後變成新聞，這在實驗結尾已經驗證一個例子，證明這個情況是會發生的，透過實驗過程，PTT 在台灣是具有高度影響力的社群文章討論平台也是有跡可循的。

5.2 未來展望

在實驗過程中，我們所提出的方法面臨仍有許多地方可以調整，如斷詞系統的精確度，若是斷詞的效果降低會影響分群的結果，進而影響熱門主題的偵測；也有可能因為分群效果不彰，影響主題偵測的準確率。

以下逐一敘述本論文在未來可以改進與研究的方向：

1. 除了使用文章的回文數的特性之外，可以嘗試加入具有高度影響力作者的考量，或是採用文章分數當作參考的基準。
2. 熱門主題的文章內通常會有大量的使用者參與討論並且留言，未來可以嘗試使用者意見分析，探討是否能改善熱門主題偵測的結果。
3. 斷詞系統的優化，本研究只有採用 Jieba 斷詞系統，若是使用其他城市語言去進行研究的話，是否能找到更好的斷詞方式，增加擷取文章特徵的精確度。
4. 未來可以嘗試更多更好的分群方法，降低人工些微調整的時間成本，改善分群的效率。
5. 本論文只有使用 PTT 這個平台所有的實驗資料做為案例，未來可以選擇其他相同類型的 BBS 進行分析，觀察本論文提出的方法在不同案例下的效果。

參考文獻

[1] Reddit, https://zh.wikipedia.org/wiki/Reddit （Viewed on 2015/05/05）

[2] Wikipedia, 批踢踢, https://zh.wikipedia.org/wiki/批踢踢（Viewed on 2015/05/05）

[3] Allan, James, et al. "Topic detection and tracking pilot study final report." (1998).

[4] Yu, H. O. N. G., Yu FAN ZHANG, and Ji-Li LIU Ting LI Sheng. "Chinese topic link detection based on semantic domain language model." Journal of Software 9 (2008): 010.

[5] Ponte J, Croft WB. Text segmentation by topic. In: Peters C, ed. Proc. of the European Conf. on Research and AdvancedTechnology for Digital Libraries. London: ECDL Press, 1997. 113−125.

[6] Nallapati R. Semantic language models for topic detection and tracking. In: Hearst M, ed. Proc. of HLT-NAACL2003

[7] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester,and R. Harshman. Using latent semantic analysis to improve access to textual information. In CHI ’88: Proceedings of the SIGCHI conference on Human factors in computing systems,pages 281–285, New York, NY, USA, 1988. ACM.

[8] T. Landauer, P. Foltz, and D. Laham. Introduction to latent semantic analysis. Discourse Processes, 25:259–284, 1998.

[9] T. Hofmann. Probabilisitic latent semantic analysis.In UAI99: Uncertainty in artificial intelligence, 1999.

[10] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. Machine Learning, 42(1-2):177–196, January 2001.

[11] L. Zhang, X. Wu, and Y. Yu. Emergent semantics from folksonomies:A quantitative study. 4090:168–186, 2006.

[12] Lee, Sungjick, and Han-joon Kim. "News keyword extraction for topic tracking."Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on. Vol. 2. IEEE, 2008.

[13] Chen, Chien Chin, and Meng Chang Chen. "TSCAN: a novel method for topic summarization and content anatomy." Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2008.

[14] Wartena, Christian, and Rogier Brussee. "Topic detection by clustering keywords." Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on. IEEE, 2008.

[15] F. Archetti, P. Campanelli, E. Fersini, and E. Messina. A hierarchical document clustering environment based on the induced bisecting k-means. In H. L. Larsen, G. Pasi, D. O.Arroyo, T. Andreasen, and H. Christiansen, editors, FQAS,volume 4027 of Lecture Notes in Computer Science, pages 257–269. Springer, 2006.

[16] You, Lan, et al. "BBS based hot topic retrieval using back-propagation neural network." Natural Language Processing–IJCNLP 2004. Springer Berlin Heidelberg, 2005. 139-148.

[17] Hui-min, Ye, Cheng Wei, and Dai Guan-zhong. "Design and implementation of on-line hot topic discovery model." Wuhan University Journal of Natural Sciences 11.1 (2006): 21-26.

[18] Zheng, Donghui, and Fang Li. "Hot topic detection on BBS using aging theory." Web Information Systems and Mining. Springer Berlin Heidelberg, 2009. 129-138.

[19] Chen, Keh-Jiann, and Shing-Huan Liu. "Word identification for Mandarin Chinese sentences." Proceedings of the 14th conference on Computational linguistics-Volume 1. Association for Computational Linguistics, 1992.

[20] Chen, Keh-Jiann, and Wei-Yun Ma. "Unknown word extraction for Chinese documents." Proceedings of the 19th international conference on Computational linguistics-Volume 1. Association for Computational Linguistics, 2002.

[21] Ma, Wei-Yun, and Keh-Jiann Chen. "A bottom-up merging algorithm for Chinese unknown word extraction." Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17. Association for Computational Linguistics, 2003.

[22] Jieba, [Online]. Available: https://github.com/fxsjy/jieba （ Viewed on 2015/01/02）

[23] 中研院斷詞系統, https://github.com/fukuball/CKIPClient-PHP（Viewed on 2015/01/02）

[24] Stanford word Segmenter, http://nlp.stanford.edu/software/segmenter.shtml （Viewed on 2015/01/02）

[25] Steinhaus, H.. Sur la division des corps matériels en parties. Bull. Acad. Polon. Sci. 1957, 4 (12): 801–804. MR 0090073. Zbl 0079.16403

[26] MacQueen, J. B.. Some Methods for classification and Analysis of Multivariate Observations, 1, Proceedings of 5th Berkeley Symposium on

Mathematical Statistics and Probability. University of California Press. 1967: pp. 281–297 [2009-04-07].

[27] Lloyd, S. P. Least square quantization in PCM. Bell Telephone Laboratories Paper. 1957. Published in journal much later: Lloyd., S. P. Least squares quantization in PCM. IEEE Transactions on Information Theory. 1982, 28 (2): 129–137 [2009-04-15]. doi:10.1109/TIT.1982.1056489

[28] Figueiredo, Mario AT, and Anil K. Jain. "Unsupervised learning of finite mixture models." Pattern Analysis and Machine Intelligence, IEEE Transactions on 24.3 (2002): 381-396.

[29] Usman, Ghousia, Usman Ahmad, and Mudassar Ahmad. "Improved K-Means Clustering Algorithm by Getting Initial Cenroids." World Applied Sciences Journal 27.4 (2013): 543-551.

[30] Alrabea, Adnan, et al. "Enhancing k-means algorithm with initial cluster centers derived from data partitioning along the data axis with PCA." Journal of Advances in Computer Networks 1.2 (2013): 137-142.