以詞彙擴展和地域性社群網絡為基礎之見解探勘方法

An Opinion Mining Approach Based on Localized Social Group and Query Expansion

Ming-Wei Kat, Nien-Lin Hsueh, Yi-Chung Chen and Shi-Chuen Hwang
Feng Chia University
Department of Information Engineering and Computer Science
Taichung city, Taiwan
Email: nlhsueh@mail.fcu.edu.tw

Abstract—隨著社群網絡不斷的發達，許多研究在探討如何從這些社群網絡中找出有用的資訊。其中更有研究指出，利用社群網絡分析事件的發生，會比政府單位公告的時間更快[1]，且範圍更廣闊[2]。因此，我們設計了一套能即時性廣泛的收集社群網絡上的發文資料，且能任意切換所偵測的事件，並針對不同的事件，找出對於該事件最佳的資料收集源、關鍵字列表和特徵值組合的方法。所產生的關鍵字列表也會隨著所偵測到的事件而更新，會隨著時間而成長。從水災事件的實驗結果來看，系統所偵測出發生事件的時間點也的確的發生了水災事件。對於發文內期望取得的時間、地點和災害程度，實際上難以從單一的發文上取得。因此這些資料可以用系統的特性和偵測結果來彌補，讓分析結果能被更好的利用。

I. Introduction

根據網站 Statista 的統計，著名的社群網絡軟體 Twitter 在2015年時全球活躍的使用者數量已達到3.07億[3]，同一時期 Facebook 的活躍使用者數量更是它的5倍，已達到15.45億[4]，而且數字還在不斷的成長中。因此在過去也有不少的學術論文在探討如何從這些社群網絡中找出有用的資訊，比較常見的是用於股市的預測[5],疫情預測[2][6]和自然災害預測[1] 等。根據 Sakaki 等人[1]的研究，利用社群網絡進行即時性的地震偵測，能在政府單位公佈災難警報前6分鐘就偵測出事件的發生，而這關鍵時刻的6分鐘，對於即將受災的人們來說是非常珍貴的，如善用這數分鐘的時間可避免許多損失。

台灣是島嶼型的國家，常年受各種水患的影響造成各種的經濟損失，例如1996年的賀伯颱風及2009年的莫拉克颱風，總共造成了近500億台幣的經濟損失，其中莫拉克颱風更造成了六百多人死亡。傳統的水患預測方式是採用水位感測器，但台灣各地區的地勢不一，加上成本上的考量感測器無法全面的遍布各地。因此水災造成的影響的區域難以預測，如何能夠較為精準地劃定淹水區域，並依據當時之資源進行各種決策行為，一直是困擾台灣政府的課題。

因此台灣科技部 MOST (Ministry Of Science and Technology) 推動了一個整合型計劃，計劃中會結合感測器和社群網絡上分析所得的資料，互補不足之處去預測和判斷發生水患的地區。在計劃分成多個子計劃進行，其中在社群網絡資料分析的部分，Zong-Han 等人[7]在分析了一般的社群網絡資料分析架構後，針對社群網絡的資料分析進行了軟體框架的設計。但經過一年的實際執行和資料分析以後，發現系統的各部分還有可改善之處。

- 首先是資料分析所採用的資料源。該計劃提供了一個社群網絡軟體 Facebook 上的社群為平台，讓民眾在發生水患時，可以到這平台去公告發生水患的時間、地點、淹水水位等資訊，系統會收集這些資訊去進行分析和整理，找出發生水患的地區。但經過大約一年的資料收集以後發現，當水患或其他災害發生時，的確還是會有民眾發布留言公告，希望能通知身邊的人，讓他們能做出相對應的措施。但他們並不會刻意的到某個社群網絡平台去發表留言公告，一般上也只會發布在自己的個人頁面，或是自己經常瀏覽的社群。
- 第二點是關鍵字列表。系統採用的是關鍵字的擷取的分類方法，因此分類器能分析出什麼災害事件取決於關鍵字列表。如果關鍵字列表內的關鍵字與地震有關，則能分析出發文是否為地震相關的發文，如果列表內的關鍵字與水災或其他事件有關亦是如此。關鍵字列表除了能決定所分析的事件外，列表內的關鍵字與事件的相關度也大大的影響著分類器的精準度。在過去的方法中該列表是透過人工的方式去決定列表的內容和權重，但這種方式太過於主觀，而且列表的內容會是屬於靜態的。如果最近將要有某個颱風靠近台灣，那麼對於該事件來說，這個颱風的名字可能是判斷是判斷發文的重要依據。但如果列表是靜態的資料，那麼該颱風的名字是不會出現在列表內的。而且過往的方式也不容易進行事件的切換，如要偵測水災以外的事件，則需再手動的建立事件的關鍵字列表。
- 第三點是分類器採用的特徵值。由於資料收集的方式改變了，我們對於精準度的要求也跟著提高了。所以只採用關鍵字為分類的特徵的方式已經不能滿足系統的需求了，另外系統也擴充成支援不同事件的偵測，因此也需針對採用的特徵值進行改進。

因此本論文提出的方法改善了以上的問題，為了驗證本論文的方法，我們實際的建立了該系統並進行了一些實驗。要進行事件偵測，使用者只需在一開始給系統一個或多個關鍵字，然後在系統上進行發文與事件相關性的判斷，過程中會慢慢的優化關鍵字列表、同時也會找出對於事件最佳的社群網絡資料收集平台、和產生分類器所需要的訓練資料。然後藉由這些產出去進行分類器的訓練，並找出最佳的特徵值組合。在這時候系統的模型就已經建立好了，只需再把即時性收集的的社群網絡發文群資料，交由已經訓練完成的分類器進行分類，判斷出發文是否與事件有關。如短時間內偵測出比較大量和事件相關的發文，這意味著目前該事件正在、將要或剛發生的可能性很高，就會透過使用者介面提醒使用者。另外，我們也把該系統應用在 MOST 的水災整合型計劃底下，因此該資料也會即時性的通知該計劃底下的其他子計劃，經過資料整合確認事件的發生以後，會進行相對應的災害規劃。為了證實系統的有有效性，我們也針對系統上的各個部分進行了多項實驗測試。

本論文的其它章節安排如下，第二章我們整理了其他學術論文在本系統各個部分上的做法。第三章會介紹系統各個模組的設計概念，然後在第四章介紹進行的案例和實驗，並在最後進行結論。

II. 相關研究

A. 社群網絡資料分析

隨著社群網絡日益的發展，有不少的學術論文在探討如何從這些社群網絡中找出有用的資訊，其中比較受人關注的是含有巨大商機的股市預測，Bollen 等人[5]透過每天收集的 Twitter 發文，利用 OpinionFinder 和 Google-Profile of Mood States (GPOMS) 從中分析出群眾的正向與負向情緒，以及把發文內容分成冷靜、警惕、確信、活力、友善和幸福。然後再透過現實中實際發生的事情與分析結果進行比對，從而找出相關性來預測股市。

另外，會危害人們生命安全的議題也是常被學術論文討論的議題。2010年 Sakaki 等人[1]提出的論文，他們利用關鍵字對Twitter上全部的內容進行搜尋，再從中找出出現地震字眼的發文。如果某段時間偵測出大量的發文，系統就會判斷為有地震事件的發生，然後再透過發文內容預測出地震的震央。根據作者的描述，如果在日本一端偵測到地震的發生，如果即時性的通知另一端的民眾，他們將會有大約6分鐘的時間進行避難。而且這資訊的接收速度，會比政府單位在電視上公告的避難通知更快。

雖然有許多研究在進行事件的分析，但社群網絡上能分析到什麼樣的事件資訊是屬於未知的問題，因此除了比較針對特定事件的偵測外。2015年 Nutzel 等人[8]利用了 Burst Based 的方式對社群網絡資料進行分析。他們利用的同樣是 Twitter 上的發文資料進行分析，他們以一個小時為測試基準，收集某測試地區的 Twitter 使用量，再以此為基準，判斷該地區需要收集多久的資料才能足夠形成參考語料庫。例如 Twitter 使用者非常多的美國或許只需收集幾天即可形成參考語料庫，而使用者比較少的地區則需收集一個月或更久的資料來形成參考語料庫。這動作的目的是為了解決 Twitter 使用人數不多的地區，收集資料上的問題。另外作者也認為不同地區的人們，習慣性使用的字詞也不一樣，所以比較合適的方式是針對不同的地區使用不同的參考語料庫。形成語料庫以後他們會以此為基準，判斷當下是否有出現頻率突然暴增的字詞。關聯詞在一般的發文中就常出現，所以參考語料庫中它也有著一定的出現次數，因此即使它們是常出現的字詞，但也不會被判斷為是突然暴增的字詞。反而是一些比較冷門的字詞，例如“登革熱”、“世界杯”、“地震”等等，容易出現突然暴增的表現，這表示這是社群網絡上突然興起的議題。因此，這部分除了可以作為社群網絡資料分析的參考以外，我們也會把這方法應用在關鍵字的擴充上。

從 Sakaki 等人[1]和 Nutzel 等人[8]等人的論文中我們可以得知，當某些特別的事件發生時，社群網絡上使用者的發文數量會提升，更精確的說是含有與事件相關資訊的發文數量會增多。Bollen 等人[5]的論文也讓我們知道，如果從社群網絡中所分析出在過去的結果能與現實發生的情況對應，那麼如果當下分析出的結果，也很大的可能在現實中正在發生，所以他們的方法才能用於預測股市。對於台灣地區這些學術論文在社群網絡資料分析的方法並無法直接套用，但取各家的特色概念，並加以改善以後會是不錯的社群網絡資料分析方法。

B. 關鍵字

用關鍵字判斷發文與事件相關性常是常用的方式。搜尋引擎巨頭 Google 公司，證實了利用人們在搜尋引擎上的搜尋流感，以及與其相關的關鍵字資料，就能能用於預測流感的爆發地點。這套方法名為 Google Flu Trends (GFT) 他們也把這些資料整理後發布在網頁[2]上讓人們進行研究。但採用單獨的關鍵字一般上無法有效的進行事件的發生，就如 GFT 的研究如果單純以“流感”為關鍵字還難以確定事件的發生，還需要搭配其他的一些關鍵字，例如“治療”、“預防”、“應急措施”等等會比較能確定事件的發生。

在系統的分類器上，我們會需要針對事件的關鍵字列表，關鍵字列表是由一組和事件相關的關鍵字以及權重組成的列表。為了解決不同事件需要不同關鍵字列表的問題，可以採用關鍵字的擴充處理，該議題早在多年前就已經有學術論文在討論，學術名稱為詞彙擴展 (Query Expansion)。他的核心做法是給定一個關鍵字以後，依據一些參考的資料，把該關鍵字擴充成多個和該關鍵字相關的關鍵字。但會依據所使用的方法不同，所產生出的關鍵字也會有著不同意義上的相關性，這技術目前廣泛的被運用在搜尋引擎上。例如2009年 Biancalana 等人[9]提出的論文是當使用者給出一個關鍵字查詢網頁時，就會根據使用者過去的瀏覽記錄，再結合這些網頁上的標籤把使用者的關鍵字擴充成更多符合使用者所想的關鍵字。2008年 Wang 等人[10]提出的論文同樣的是利用網頁上的標籤進行使用者關鍵字查詢的擴充。

也有研究把詞彙擴展應用在社群網絡上，例如2009年 Bertier 等人[11]相信，使用者的喜好和它朋友的喜好會

非常的接近，所以他們利用社群網絡軟體上使用者朋友的標籤作為參考，擴充和推薦使用者的查詢關鍵字。2013年 Roque López 等人[12]的論文在建立訓練資料時就是直接的使用社群網絡上的發文資料。另外2015年 Nutzel 等人[8]的論文也認為每個國家的人民，在社群網絡上發文所使用的字詞是類似的，所以他們的論文中使用的方法也會收集社群網絡上的發文作為訓練資料。由此可見同個地區的發文內容會具有一定的參考價值，例如說使用者的關鍵字是“政治”，那麼用該地區的發文資料或許會得到該地區的一些政治人物名稱。如果是在常淹水到屋頂的地區收集發文資料，或許用“淹水”為關鍵字時可以收集到“屋頂”這關鍵字。

III. 研究方法

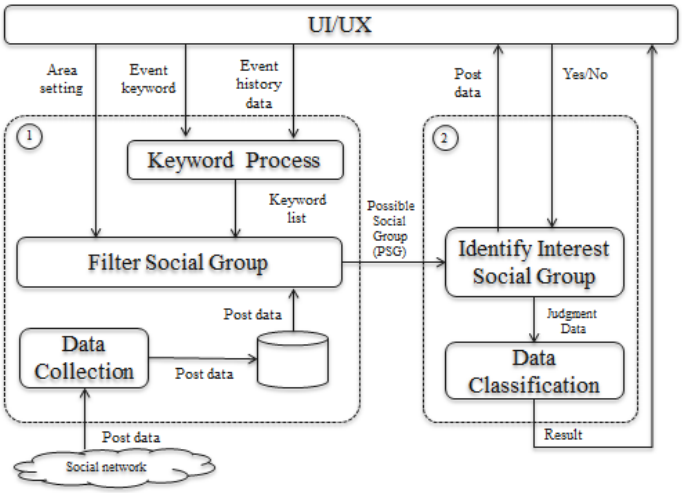

圖 1. 系統架構

本論文提出的方法是能任意切換事件分析的系統，系統的架構設計如圖1，系統的執行可分成訓練階段和實際執行階段。訓練階段主要的目的是產生和確定架構中的資料和模型，所使用的資料是過去所有收集的發文資料。我們會先以擴散的方式，盡可能的讓Data Collection 模組收集所有社群網絡平台上的資料。然後經過Filter Social Group 模組的過濾出(Possible Social Group – PSG)，並在Identify Interest Social Group 模組的功能底下讓使用者判斷發文是否與事件相關，過程中會慢慢的篩選出適合該區域和該事件的平台 (Interest Social Group – ISG)，我們相信不同的地區針對不同的事件需要從不同的平台收集資料。Keyword Process 是詞彙擴展的模組，在關鍵字的處理上只需給定系統數個和事件相關的關鍵字，系統會透過詞彙擴展的方式自動生成關鍵字列表，但這時的關鍵字列表還是屬於比較粗糙的，所以Identify Interest Social Group中使用者的發文判斷也會回饋給這模組，讓系統慢慢的去調整該關鍵字列表。另外系統也會自動的透過 Burst Based 的方式，自動的更新最新可能和事件有關的關鍵字，讓關鍵字列表能隨著時間一起成長。

Data Classification 是分類器的處理模組，我們使用的分類器是SVM（Support Vector Machine）。在特徵值的處理上，由於我們希望系統有更好的通用性可以方便的進行的事件切換，所以系統並不會固定所採用的特徵值。而是在訓練階時計算出各種特徵值組合下的精準度後，再決定採用那一組的特徵值。因為我們相信不同的特徵值，在不同的事件分析上有著不同的效果。

模型訓練完成以後，系統就會進入實際執行階段，這時Data Collection 模組的功能就會轉換成即時性的發文資料收集，讓Data Classification模組利用已完成訓練的模型進行即時性的發文分類，這時整體系統就能進行及時性的災害事件預測。

A. Keyword Process

詞彙擴展 （Query Expansion） 是常被使用在搜尋引擎上的技術，它的基本精神是認為使用者給予的關鍵字並不完全代表他所想要搜尋的事情。所以利用各種其他的參考資料，找出其他與使用者輸入的關鍵字相關，且接近使用者所想的關鍵字。透過額外加上的這些關鍵字，讓搜尋結果更接近使用者所想要搜尋的結果。在系統中我們希望透過使用者輸入的幾個關鍵字，自動產生出和事件相關的關鍵字列表，這部分正好可以套用詞彙擴展的概念。

該模組的功能可參考圖2，它會從使用者平台上接收使用者給出的任意數個和事件相關的關鍵字關鍵字例如“水災”、“淹水”、“颱風”等，然後依據該關鍵字進行詞彙擴展。擴展的方式可以分成三個部分。第一部分是基礎資料，這部分設計成可結合多個資料源進行，目的在於先建立基本的關鍵字列表。在這部分可以使用的資料源可以依據事件的特性再進行擴充，目前系統目前使用的是網絡上的同義詞和近義詞API、維基百科API和Yahoo 關鍵字搜尋為基本的資料源。

在第一部分以後就先建立了基本的關鍵字列表，但由於我們所要判斷的是使用者的發文，所以關鍵字也應該從使用者的發文中取得。因此第二部分我們借助該事件過去發生歷史資料為參考，從資料中收集所有該時間和地點的發文。經過字詞的切割以後再透過 Burst Based 的方式，找出該段時間突然出現頻率暴增的字詞，以此為新的關鍵字，權重則以暴增頻率的倍數為基準。而不同的事件在資料收集的時間上也需要再進行調整，例如說地震事件是屬於比較突然且危急的，所以在資料收集上單獨利用歷史資料中災害事件發生的當天，或精確到以小時為單位即可。但對於水災事件，由於該事件是需要醞釀期的，所以該類事件所需參考的除了歷史資料中的當下以外，還需要再參考前後的一段時間。Burst Based 的另一項議題是如何定義那些字詞是屬於出現頻率突然暴增的，我們採用的是以歷史資料中事件發生的時段前的x天中的資料為基準，參數的設置可考慮該地區的平均發文數量來設置。假設x設置為30天，那麼系統就會把該地區30天內所有的發文進行字詞的切割，並計算出所有字詞出現的次數，以此為基準去判斷發生災害事件時，那些字詞的出現頻率是屬於暴增的現象。

經過了第一和第二部分的關鍵字擴充以後，目前的關鍵字列表和權重還是屬於一個比較粗糙的情況。因此第三部

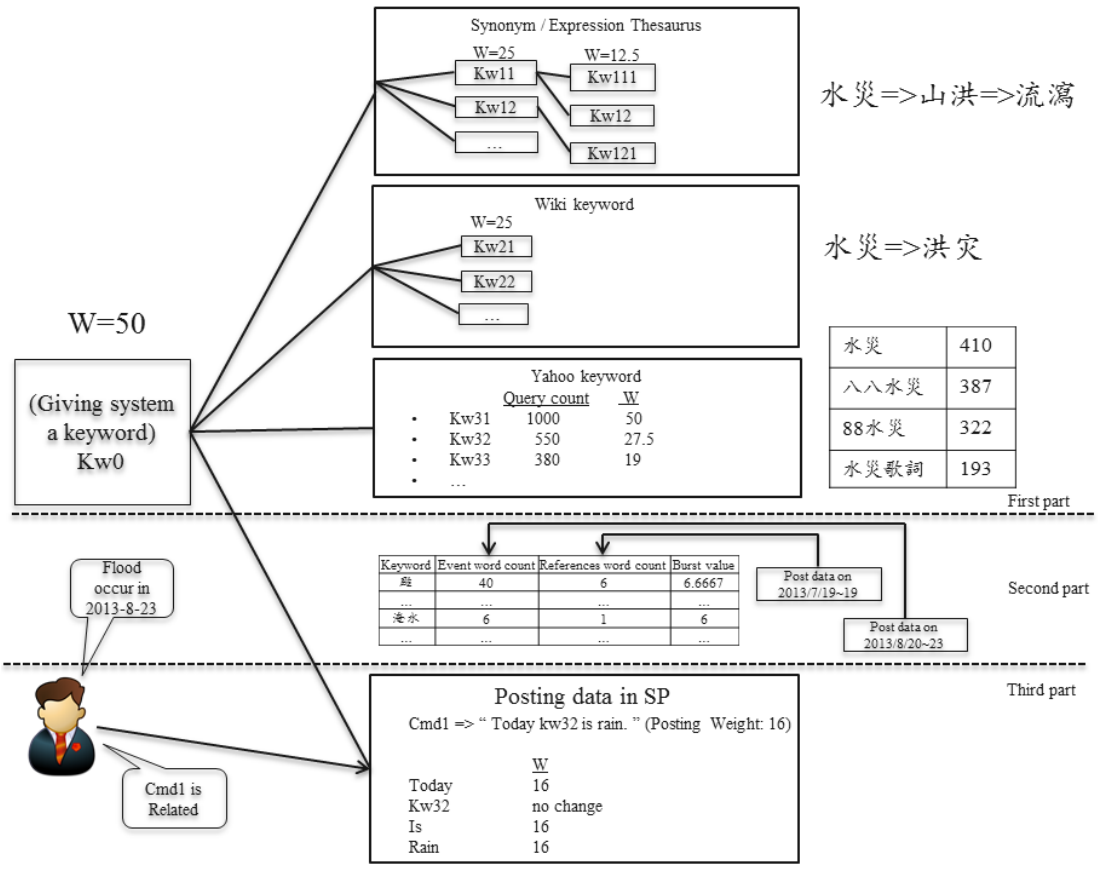

圖 2. 詞彙擴展架構

分是透過 Identify Interest Social Group 模組和使用者進行互動，以目前的關鍵字列表的內容，從所有發文中找出出現該關鍵字的發文，讓使用者進行是否與事件相關的判斷。如果利用該關鍵字找出的發文與事件無關則減低該關鍵字的權重，那麼慢慢的該關鍵字就會因為權重不高而失去代表意義。相反的如果與事件有關則增加該關鍵字的權重，另外如果該發文與事件有關，該發文的內容也會進行字詞的切割，並新增到關鍵字列表中，權重則以該篇發文的權重值設置。

另外，系統在實際執行階段時，如有偵測到確定性的災害事件發生時，也會以當下的時間為基準，利用第二部分的詞彙擴展更新目前的關鍵字列表。好處在於關鍵字列表不會是屬於靜態的資料，如果最近將要有某個颱風靠近台灣，那麼對於該事件來說，這個颱風的名字可能是判斷是判斷發文的重要依據。但如果列表是靜態的資料，那麼該颱風的名字是不會出現在列表內的。

B. Data Collection

該模組底下有一隻常駐程式，全天候相隔特定時間就到社群網絡軟體/論壇上去抓取最新更新的資料。台灣常用的社群網絡軟體是 Facebook 、論壇、BBS 和其他網頁社群網絡軟體，除了Facebook以外一般的社群網絡軟體都沒有提供能進行資料收集的 API 。因此需要透過網頁爬蟲[13]、關鍵字萃取[14]等技術，從網頁上找出我們所要的資訊。但經過實際的資料收集以後發現這些社群網絡軟體的資料收集效果並不佳，主要原因在於難以判斷發文相關的區域，因此目前只考慮Facebook的發文資料。

一般的社群網絡軟體 API (例如 Twitter、Facebook) 都有提供地理位置搜尋的功能，使用的方式是給予一個中心點和半徑的距離為參數。所以在我們的系統上，收集資料時，是以行政區域為單位，先找出各行政區域的中心坐標，和該區域的面積大小，推算出半徑以此為參數進行搜尋。然後再透過收集到社群資料上的地理坐標，去調整該社群所屬的行政區域。

透過地理位置的方式只能收集到 Twitter 和 Facebook 上部分的社群網絡社群資訊，而且這方法不適用於一般的論壇，所以在資料收集上我們還利用了另外一套方式。Facebook 上還提供了透過社群名稱搜尋社群的功能，因此有效率的透過名稱搜尋找出和該地區有關的社群就成為這部分資料收集的關鍵。社群網絡軟體Facebook 的發展是起始於美國的哈佛大學，後來在學生族群內不斷的擴散，由此可見 Facebook 是深受學生族群歡迎的社群網絡 ，所以我們在名稱搜尋時，就以學校名稱為出發點去考慮。行政地區所屬的學校資料在一般上在國家的教育部網站上就可取得，而且常附有該學校的各種地理資訊，例如坐標

地址等。從結果來看，所收集到的一般上是該學校的各種社團、班級、系所的社群，但很多該地區發生的事件例如停電、水災等，都會被公告在這些社群內。 除了學校名稱外，該行政區的名稱也非常適用於名稱搜尋，因為許多的社群名稱都會加上行政區，例如“台中市西屯區活動中心”、“台中市西屯區衛生所”等。另外如果知道該地區的一些名勝地、學校別稱、公司行號、商店等也可考慮其中，因為會影響營業的災害事件，許多商家業者也相當的關注。另外，使用名稱搜尋的方式也可用於一般的論壇資料收集，許多的論壇在板塊名稱上都會直接以行政區域、學校名稱命名，而在該版塊上的資訊也會和該行政區域有關。

透過以上兩種方式可以收集到一定數量的社群，但並不是所有的社群都保持在活躍的狀態，例如說某大學的班級社群會因為大家都已經畢業了，所以該社群的活躍程度會逐漸下降。而且也因為所收集到的社群數量不少，不斷的花時間在不活躍的社群上，會讓系統的資料無法保持即時性。經過資料的觀察以後，發現活躍的社群大約只有20%左右。因此我們借用了2014年 Luo 等人[15]在多租戶資料庫表格上的管理方式，把所有的社群根據活躍程度分到兩個列表上，名為 AGL(Active Group List)和 NAGL(Non-Active Group List)，AGL 儲存的是活躍和 ISG 上的社群，判斷活躍的方式是最後發文時間和發文總數，AGL 每隔數分鐘就會到 Facebook 去取得最新的發文資訊，而其大小由門檻值決定，該值可依據硬體系統的效能和網絡狀況決定，一般建議在總群組數量的20%。NAGL 儲存的是 AGL 以外所有的社群，為了維持兩個列表之間的平衡，NAGL 同樣也會到 Facebook 上去取得發文資料，但時間間隔會比較久，例如數小時或每天一次，而且因為社群的數量比較多其執行時間也會比較久。兩個List列表內也會依據活躍程度有優先順序，執行時會根據這順序執行，另外每次執行完後 NAGL 內最活躍的數個群組會被加到 AGL 內，然後再根據 AGL 的優先次序，把其中最不活躍的數個社群移到 NAGL 內，因此逐漸不活躍或逐漸活躍的社群就能被放在合適的列表內。

C. Filter Social Group

由於 Data Collection 模組所收集的資料是龐大且包含所有時間與地區的，但使用者在不同的時候，對於所需的資料會有不同的需求。所以該模組的主要功能是過濾和排除完全不相關的資料，篩選出PSG讓下一模組使用，篩選條件包括地理位置（國家、區域、城市），時間等。另外除了依據使用者給予的基本條件過濾資料外，也會依據 Keyword Process 模組所整理出的關鍵字，篩選出含有關鍵字的發文內容。因為系統是採用關鍵字的擷取的方式進行分類，如果完全不含有關鍵的發文，是無法分析出與事件有關的資訊，在一開始的步驟就把這些發文過濾，可大大的減少往後的計算量。

D. Identify Interest Social Group

該模組為系統的核心模組，透過和使用者的互動慢慢的調整關鍵字列表的內容和權重，並在最後找出對於事件最佳的社群網絡社群資料收集源，以及產生訓練資料讓分類器進行模組的訓練。與使用者進行互動的動作會以每次少量資料，但執行次數多的方式執行。在每次的執行中系統都會找出，以目前現有的資料為基準，最有可能和事件有關的發文讓使用者進行判斷，然後依據結果調整關鍵字列表和權重，再重新找出新的發文。利用該方式的目的主要是以現實執行的層面去考量，使用者有耐心去進行發文判斷的時間並不會太長，所以使用者所判斷的每筆資料都是珍貴的。假設單純的以隨機的方式選出發文讓使用者判斷，出現與事件相關發文的機率並不高，利用全部與事件無關的資料進行訓練並沒有太大的意義。

這模組一開始會先隨機篩選出一定數量的的社群數量目前以50為例，然後會收集這些群組的所有發文，但考慮到這些群組的發文數量總和會太少，所以會用另外的設定值去設定最少發文總數目前以50萬為例。然後依據詞彙擴展後的關鍵字列表和初始設定的權重值，計算出全部發文的權重值。計算方式是以過濾重複出現的關鍵字以後，把關鍵字的權重值都累加的起來。例如“昨天下雨，今天又下雨”，假設“今天”的權重值是5，“下雨”的權重值是10，那麼這篇發文的權重值就是15。然後把所有發文的權重值進行排序，只取出前數名篇權重值最高的發文目前以10篇為例。然後把完整的發文和回文內容一併的找出再顯示在使用者介面上，然使用者決定該文章是否和他所想要的事件有關。以前面所舉的例子來看，如果它是和事件無關，那麼“今天”和“下雨”的權重值就會減1有關則加1。另外如果該發文是與事件有關的，那麼這發文會依據斷詞演算法進行切割，去除既有的關鍵字以後其餘的都會被當成是新的關鍵字。另外與事件相關發文的社群也會被記錄到社群列表中，該列表中的社群會優先被當作取出發文分析的社群，但如果該列表中的社群數量超過所需社群的80%時，當中發文與事件相關最少的群組會被剔除。目的在於保留20%的社群是利用隨機的方式選，因為或許有些相關性很高的社群在前面數次的隨機選取中沒有被選上，而列表的位置卻被中低相關的社群所佔據。

當使用者判斷完這10篇發文以後，就會依據新的群組選擇方式、關鍵字列表和權重值再次計算出10組發文。以漸進式小數量為單位重複計算發文的，目的是讓使用者更有效率的判斷發文，更高機率的出現與事件有相關的發文。以實驗結果來看，使用者所要的發文一般上都不會在第一次篩選就出現，而是在經過數次的篩選以後，不相關的發文慢慢被過濾之後才出現。這篩選的動作會不斷的重複進行，直到選出各地區的關鍵性社群以後才停止。但也考慮到有些較小，或是人煙稀少的區域，所以同時會參考該區域所收集到的群組數量作為停止條件。

篩選出足夠數量的群組以後，就會用目前所有已判斷的發文去計算出目前的準確度，並把統計數據顯示給使用者參考。假設準確度非常的低，使用者可再次進行判斷發文的動作來增加準確度。但如果使用者認為目前的準確度已經屬於可接受的範圍，那麼就會交由往後的模組進行全部發文的分析。發文的判斷有以下幾點的意義：

- 第一點是作為詞彙擴展的參考依據，在一般的詞彙擴展中都會利用一些資料作為參考的依據，例如過去的

歷史資料[9]、朋友的喜好[11] 等。而我們使用的方式則是該關鍵字是否出現在使用者認為相關的發文上。

- 第二點是解決事件的灰色地帶問題，以判斷“水災”事件為例，假設使用者認為他所想要的分析的事件是當下是否發生水災，那麼他在判斷發文時,對於通告性質的發文就會判斷成否，如果他只是想分析的事件是包含已發生的事件，那麼也會表現在發文的判斷上。
- 第三點是產生訓練資料，在機器學習的方法上在分析事件以前都必須先準備好訓練資料，而且針對不同的事件也需要準備不同的訓練資料，如果要建制一個可判斷多種事件的系統，則需要準備非常多組的訓練資料。
- 最後是迎合不同使用者的需求，不同的使用者對於準確度也有不同的要求，例如一個保險從業員或許他只是想大略的知道某地區常發生什麼自然災害，從而販售相關保險，那麼他所需要的事件種類會非常多，但精準度一般即可。這時他只需對不同的事件進行比較少數量的發文判斷即可。如果是在進行水災事件預防的使用者，那麼它只想要分析水災事件，但精準度卻要非常的高。這時的操作方式則是針對水災事件進行大量的發文判斷即可。

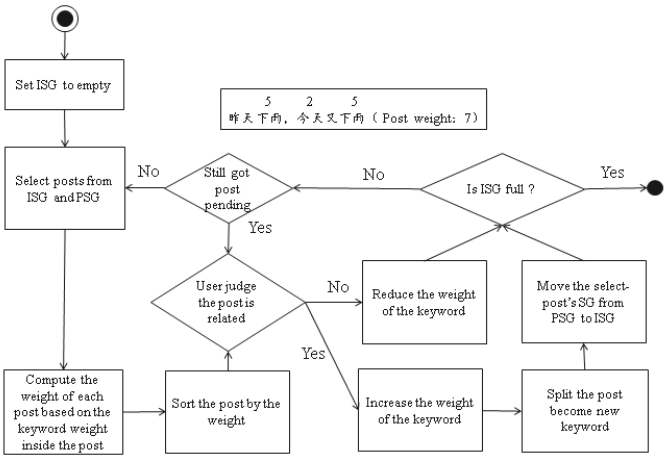

圖 3. Identify Interest Social Group流程圖

E. Data Classification

該模組的功能是處理使用者已經判斷的發文，把它們轉換成特徵值，並用分類器 SVM 判斷出發文是否與事件有關。我們相信不同的特徵值，在不同的事件分析上有著不同的效果。例如說所偵測的事件是比較緊急的像是地震，那麼發文內容一般上字數並不會太多。因為發文者會急著把訊息發布出去通知其他的人，但對於比較慢性的災害像是水災，發文長度或許會比較長。因此雖然系統上有著多項的特徵值，但系統並不會固定所採用的特徵值，而是依據使用者已經判斷的發文為基準，把所有已判斷資料隨機的分成8成的訓練資料和2成的測試資料。然後把這些資料計算出各種特徵值組合下的精準度後，再決定採準確率較高的一組特徵值。因為是以組合的方式進行精準度的比較，相同精準度的情況經常出現，尤其當訓練資料量不大的時候。因此面對這情況我們會採用特徵值使用得較少的組合，另外特徵值計算的難易度決定是決定因素之一。為了讓系統的通用性更高，能使用各種資料源例如 Twitter、 Facebook 、論壇等發文資料進行分析，以及能適合用於各種事件，所以我們利用了他們都共同擁有的特徵進行分析。

- 與事件相關程度（發文權重，以關鍵字權重計算）
- 文章長度
- 討論熱烈程度（發文和回文的次數加總）
- 文章種類（屬於發文或回文）

除了以上幾點針對發文的特徵值外，我們認為偵測事件很多情況下也會和時序性有關，例如水災事件，如果前一天有偵測到事件的發生，那麼今天還會偵測到的機率會很高。但不同的事件應該會有不同的時間曲線，例如說地震事件在一開始偵測到地震的發生以後，因為還有餘震的關係，所以之後或許會還是會斷斷續續的偵測到事件的發生。水災的時間曲線或許是比較連續但緩慢成長和降下的。另外，可能的特徵值是事件發生的地區，我們相信事件在不同的地區發生的頻率並不一樣，例如一些地勢較低的地區會比較容易淹水，近海邊的地區容易受風災影響等等。因此系統在訓練階段會進行兩階段的分類測試，第一階段先採用前面幾項對發文的特徵值的組合測試，找出目前最佳的特徵值組合。同時也會先以該特徵值進行全部現有資料發文的分類測試，找出以目前的特徵值為基準，各個時間和地區偵測出發文的數量。然後針對不同的事件，設置不同的參考單位和數量為特徵值。例如水災事件是參考前兩天偵測發文數量為特徵值，而昨天的偵測結果是10，前天是5，則該特徵值為15，對於地震事件或許是以前兩小時的偵測數量加總為特徵值。設定好後會再和之前各項特徵值進行組合特徵值的測試。

IV. 案例探討與實驗評估

為了驗證系統設計的有效性，我們以目前能取得的資料對系統進行了測試。在測試案例上我們主要以水災事件進行測試，然後以各個模組為單位去觀察輸入與輸出所造成的結果。

A. Data Collection

資料的收集我們以台灣台中市為例，雖然台灣人們常用的社群網絡軟體為 Facebook、論壇和BBS等。但經過試驗以後發現除了Facebook外，其他的社群網絡軟體的資料收集量並不多，且比較難以判斷所在的區域 ，因此我們也以Facebook為主要的資料收集源。台中市共有29個行政區域，因此我們在資料收集時就以這29個行政區域為出發點，找出各個行政區域的地理坐標資訊收集它們地點專頁上的發文資訊。然後我們再透過台灣教育局所發布的資料，整理出各個地區所有的學校名稱，加上該地區的名稱以後形成該地區的名稱搜尋關鍵字，再去收集所有的相關社群和發文。透過這種方式我們共收集到106,596個

Social Group 和當中的3,522,291筆發文資料，這些發文資料的時間介於2010年到2016年4月之間。

B. Keyword Process

因為測試案例是以水災為例，因此初始關鍵字我們採用了“颱風”、“淹水”、“大雨”、“水災”為初始關鍵字，經過第一部分的詞彙擴展以後得到91個介於權重值介於5到50之間的關鍵字。第二部分的詞彙擴展會以水災事件過去發生的時間和地點為參考去進行擴充。以2013年的蘇力颱風事件為例，該事件發出警報的時間共有3天（7/11～713）。我們會以天為單位去處理暴增的現象，過去出現次數為0的情況會視為1處理，以避免除以零的問題。例如7/12共找出334個關鍵字有一倍以上的暴增，我們只取前面10%的加入到關鍵字列表，範例如表I。第三部分是與使用者的互動進行關鍵字的擴充和調整，在經過約400次的資料判斷以後，共產生了3837筆的已判斷發文資料，其中只有約114筆是使用者認為與水災事件相關的。最後形成了769個關鍵字，權重則介於192至-36。而系統的設計上權重值少於5的都不會列入考量的關鍵字內，因此所考慮的關鍵字只有有143個。

表 I
詞彙擴展-蘇力颱風7/12擴充之詞彙

字詞	發生時出現次數（7/10～7/12）	過去出現次數（6/9～7/9）	Burst（暴增倍數）
蘇力	108	0	108.0
tw	100	0	100.0
防颱	86	0	86.0
停班	63	1	63.0
照常	113	2	56.5

C. Identify Interest Social Group

在系統中使用者會需要進行比較耗時間的判斷發文動作，因此使用者需要判斷多少發文才能有一定的準確度，會成為該系統是否可行的主要因素之一，另外也為了驗證判斷發文的數量是否會影響準確度。經多次的實驗測試以後，發現主要影響準確度的是已判斷發文中與事件相關的發文數量。因此我們把使用者已判斷的3932篇發文以判斷時間為排序次序，把所有資料分成5等分。以第一份資料為例，資料中與事件相關的發文為29篇，然後會再從剩下的資料中選取同等數量與事件無關的發文，形成合共58筆的資料為這部分的實驗資料。然後我們會把這些實驗資料隨機的分成8比2的訓練資料和測試資料，然後進行精準度的測試，實驗結果如圖4。從數據中我們可以看到如果有29筆與事件相關的資料，系統就能有83%的準確度，雖然準確度度會隨著訓練資料而成長，但成長的速度非常緩慢，第五階段時與事件相關的資料已成長3倍，但精準度卻只成長了8%左右。

D. Data Classification

分類器的模型訓練好以後，系統會針對所有已收集的發文進行分類，找出當中有可能與事件相關的發文。以2013年為例，全台中地區就有990筆發文被分類成與事件相關，把這些發文以天為單位去統計後，可以發現有四個時段所偵測到的發文數量會比平時高（圖5）。這些時段的整理數據如表II，由於我們無法直接取得台中市當時是否有淹水以及淹水嚴重的程度，因此我以台中市的預估降雨量以及台灣氣象局發布的颱風警報訊息??作為比較。在系統所偵測出有發生淹水事件的時間點上，其中三起事件（7/11 7/14、8/20 8/22、8/27 8/31）台中市確實降雨量也有提高。在9/19～9/22期間，雖然系統有偵測出水災事件，但台中市的降雨量並沒有提升。不過在同一時間上台灣卻有颱風警報，由此可以估計是當時台灣是有颱風事件的發生，雖然沒有影響到台中地區的降雨量增加，但也在社群網絡上被大家熱烈討論。另外颱風的警報訊息顯示的是全台灣發生颱風的警報，不完全代表台中地區。但系統也不完全會因為颱風事件而偵測到水災事件，在8/17～8/18和10/04～10/07這兩段時間，雖然都有發布颱風警報，但系統並沒有偵測出水災事件的發生，同時台中的降雨量也並沒有提高，估計是嚴重程度並不高，或是颱風所顯示的跡象並不嚴重所致。

另外，系統所偵測到事件的時間點，也比積水器所偵測到有大雨的時間點還要早。在7/12和8/20系統已經偵測出有水災事件的發生，但這時積水器的水位數值還維持在很低的水位上，而到了隔天才突然的上漲很多。但在8/28的這起水災事件上就沒有這種跡象，估計是由於水位並不算太高所致。

表 II
水災事件-系統結果與真實事件比較

日期	降雨量	颱風事件警報等級	本研究使用方法
2013-07-11	0	22	33
2013-07-12	2	22	118
2013-07-13	395	22	116
2013-07-14	1	22	12
2013-08-17	5	10	3
2013-08-18	3	10	2
2013-08-20	7	16	53
2013-08-21	179	16	109
2013-08-22	129	16	25
2013-08-27	0	20	4
2013-08-28	0	20	7
2013-08-29	122	20	32
2013-08-30	71	0	5
2013-08-31	116	0	0
2013-09-19	0	22	11
2013-09-20	0	22	40
2013-09-21	1	22	29
2013-09-22	4	22	2
2013-10-04	0	20	1
2013-10-05	2	20	5
2013-10-06	2	20	9
2013-10-07	0	20	2

V. 結論

本研究提出了一個能任意切換事件，並能針對不同的事件找出該事件最佳的資料收集平台、特徵值組合和關鍵字列表。對於不同的事件，最簡單的情況下只需給予一個或多個關鍵字，即可開始進行事件分析，過程中也只需在系統上判斷不同的發文是否與事件相關。另外，關鍵字列表

Time Group	All Data Size	All Yes Data Size	All No Data Size	Use Data Size	Train Size	Test Size	Train Yes	Train No	Test Yes	Test No	UY_SY (TP)	UY_SN (FN)	UN_SY (FP)	UN_SN (TN)	Accuracy
1	787	29	755	58	46	12	22	24	7	5	6	1	1	4	83.3333%
2	1574	34	1536	68	54	14	26	28	8	6	8	2	0	4	85.7143%
3	2361	50	2307	100	80	20	38	42	12	8	12	3	0	5	85.0000%
4	3148	84	3060	168	134	34	63	71	21	13	19	3	2	10	85.2941%
5	3936	116	3816	232	185	47	89	96	27	20	24	1	3	19	91.4894%

圖 4. 不同訓練資料量下的精準度

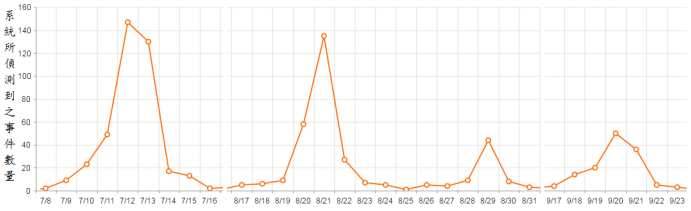

圖 5. 水災事件-以天統合結果

也會隨著所偵測到事件發生的情況，而自動更新關鍵字列表的內容。因為系統即時性的資料收集，所以使用者除了能從系統上得知該事件的歷史發生情況外，也能得知目前是否有事件發生以及發生的地區。預測結果也能透過資料的整理，共享到其他平台上。

在過去的研究中我們希望從發文內容中，取得與事件相關的詳細資料，例如事件發生的時間、事件的嚴重程度、事件發生的地點等資訊，所以所期待的發文的內容為“整個三民路與民權路口一帶淹水了”、“颱風過後，在靜宜大學往南 5 公里處淹大水”等等。但經過實際的資料觀察以後，發現甚少民眾會進行如此詳細的描述。因此在事件資料的整理上我們還是以發文內容描述的人、事、時、地、物等資訊為主，但如果當中有缺少的資料就會採用預估的方式彌補缺失的資料：

- 事件嚴重程度：該段時間所偵測出的發文數量
- 事件發生時間：發文時間
- 事件發生地點：發文收集的地區

A. 問題與限制

本系統主要透過關鍵字的擷取與分類來判斷事件的發生，雖然實驗證明了越多的訓練資料能讓分類器越精準，但受限於分類器本身的限制，精準度在後期的成長會越來越緩慢。而且從使用者實際操作的層面來思考，使用者也不太可能判斷非常大量的發文。所以系統在這方面的特性是以少量的發文判斷，卻能有一定程度的精準度，但卻不容易擁有非常高的準確度。要改善這問題可能可以採用其他的分類器或其他的方式去進行事件的判斷，但可能會因此而犧牲系統的事件簡易切換性和快速建立模型的特性。

Acknowledgement

本研究接受國科會編號： MOST 104-2627-M-035-007 研究計劃經費補助

References

[1] T. Sakaki, M. Okazaki, and Y. Matsuo, “Earthquake shakes twitter users: real-time event detection by social sensors,” in Proceedings of the 19th international conference on World wide web. ACM, 2010, pp. 851–860.

[2] A. F. Dugas, Y.-H. Hsieh, S. R. Levin, J. M. Pines, D. P. Mareiniss, A. Mohareb, C. A. Gaydos, T. M. Perl, and R. E. Rothman, “Google flu trends: correlation with emergency department influenza rates and crowding metrics,” Clinical infectious diseases, vol. 54, no. 4, pp. 463–469, 2012.

[3] Statista, “Number of monthly active twitter users worldwide from 1st quarter 2010 to 4th quarter 2015 (in millions),” http:// www.statista.com/ statistics/ 282087/ number-of-monthly-active-twitter-users/, accessed: 2016-03-02.

[4] ——, “Number of monthly active facebook users worldwide as of 4th quarter 2015 (in millions),” http://www.statista.com/ statistics/ 264810/ number-of-monthly-active-facebook-users-worldwide/, accessed: 2016-03-02.

[5] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” Journal of Computational Science, vol. 2, no. 1, pp. 1–8, 2011.

[6] M. Szomszor, P. Kostkova, and E. De Quincey, “# swineflu: Twitter predicts swine flu outbreak in 2009,” in Electronic Healthcare. Springer, 2010, pp. 18–26.

[7] W. Zong-Han, H. Nien-lin, and L. Feng-Cheng, “A framework for web comments-based opinion mining system,” in International Conference on Internet Studies (NETs 2016), 2016.

[8] J. Nutzel and F. Zimmermann, “Improved burst based real-time event detection using location dependent corpora,” in Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on. IEEE, 2015, pp. 681–686.

[9] C. Biancalana and A. Micarelli, “Social tagging in query expansion: A new way for personalized web search,” in Computational Science and Engineering, 2009. CSE’09. International Conference on, vol. 4. IEEE, 2009, pp. 1060–1065.

[10] J. Wang and B. D. Davison, “Explorations in tag suggestion and query expansion,” in Proceedings of the 2008 ACM workshop on Search in social media. ACM, 2008, pp. 43–50.

[11] M. Bertier, R. Guerraoui, V. Leroy, and A.-M. Kermarrec, “Toward personalized query expansion,” in Proceedings of the Second ACM EuroSys Workshop on Social Network Systems. ACM, 2009, pp. 7–12.

[12] R. Lopez, J. Tejada, and MikeThelwall, “Spanish sentistrength as a tool for opinion mining peruvian facebook and twitter,” in ITHEA, 2013, pp. 82–85.

[13] Nadeau, “Php tip: How to extract urls from a web page,” http:// nadeausoftware.com/ articles/ 2008/01/ php_tip_how_extract_urls_web_page, accessed: 2016-01-03.

[14] W.-t. Yih, J. Goodman, and V. R. Carvalho, “Finding advertising keywords on web pages,” in Proceedings of the 15th international conference on World Wide Web. ACM, 2006, pp. 213–222.

[15] Y. Luo, S. Zhou, and J. Guan, “Layer: a cost-efficient mechanism to support multi-tenant database as a service in cloud,” Journal of Systems and Software, vol. 101, pp. 86–96, 2015.