The 2016 Conference on Computational Linguistics and Speech Processing
ROCLING 2016, pp. 229-243

Facebook 活動事件擷取系統

Facebook Activity Event Extraction System

林圓皓　Yuan-Hao Lin
國立中央大學資訊工程學系
Department of Computer Science and Information Engineering
National Central University
luff543@gmail.com

張嘉惠　Chia-Hui Chang
國立中央大學資訊工程學系
Department of Computer Science and Information Engineering
National Central University
chia@csie.ncu.edu.tw

摘要

社群網路的普及使得不少人以 Facebook 為媒介來宣傳活動，因此本論文的目的即是建立一個 Facebook 的活動事件擷取系統，以幫助使用者快速地掌握活動的資訊。我們改善了黃等人的 Web NER Model Generation 工具[1]，藉以建立活動名稱及地點擷取模型，再利用序列樣版探勘找出活動的起始、結束日期。此外，我們也嘗試以大量的 Facebook 打卡地點來改善地點辨識準確率。實驗測試了 1,300 篇人工標記答案的貼文，以評斷系統擷取活動事件的效能和命名實體辨識的效能，並將擷取出來的活動地點實際投射到經緯度座標上，以評估預測活動實際位置的準確度。實驗結果顯示活動名稱、地點以及開始、結束日期擷取的F_1-score 分別為 0.727, 0.694 及 0.865, 0.72，活動事件整體辨識率為 0.708，顯示藉由此系統來統整 Facebook 上的活動事件並定位出事件發生的地點是相當可行的。

Abstract

The popularity of social networks has made them a perfect medium for activity or advertising campaign promotion. Therefore, many people use Facebook pages to announce their advertising campaign. The purpose of this study is to extract activity events by constructing two named entity recognition models, namely activity name and location, via a Web NER model generation tool [1]. We enhance the tool by improving the tokenizer and alignment technique. In addition, we also use a large database of FB checkin places for location name recognition improvement. For entity relation extraction, we apply sequential pattern mining to

find rules for start date, end date, and location coupling. We use 1,300 posts from Facebook to test the activity event extraction performance. The experimental results show 0.727, 0.694 F_1-score for activity name and location recognition; and 0.865, 0.72 F_1-score for start and end date extraction. Overall, the extraction performance for activity event extraction is 0.708.

關鍵詞：活動事件擷取，命名實體辨識，社群媒體事件

Keywords: Activity Event Extraction, Named Entity Recognition, Social Media Event

一、緒論

網際網路發展改變了人們獲得活動訊息的習慣，在過去是藉由電視、平面媒體、廣播媒體的宣傳來得知活動資訊，但隨著網路的進步和社群平台的蓬勃發展，商家有了新的傳播途徑，不僅增加了與客戶間的互動性，也可以更快回應客戶的問題。

得知活動或優惠的訊息是日常生活及旅遊休閒規劃重要的一環，若能取得較多的活動資訊，就能方便人們更有效的規劃。Facebook 是目前台灣最流行的社群平台，每天有一半以上的台灣人在使用手機，這樣的現象使得不少商家、政府、組織會自己經營的粉絲頁，並在粉絲頁上發佈活動訊息或商家優惠，我們發現這些跟活動有關的訊息數量會比一些現有的活動公告網站（如 CityTalk 或活動通）來的更多和即時，因此本文的目的即為從 Facebook 擷取活動訊息。

圖一、活動訊息擷取及活動地圖服務的呈現

早期社群平台尚未蓬勃發展時，想要獲取活動訊息只能仰賴 CityTalk、活動通這類網站查詢活動事件資訊，否則就只能辛苦地瀏覽政府、學校網站公告來得知事件。而這樣的

現象往往會使得資訊缺少整合與和使用者的互動。如果能將不同管道的資訊如 CityTalk、活動通、政府、學校網站公告和社群媒體做結合，便可以依活動的受歡迎程度和討論程度，提供一個活動地圖服務（如圖一所示），像是活動的評論、活動剪影和活動的圖片/影片，這些對於了解活動進行和參與有很大的幫助。所以整合現有的活動公告網站，並和社群媒體做結合是本研究的目標。

在本研究當中，我們專注於 Facebook 活動事件的擷取，並提出方法從粉絲頁發文中擷取活動及其重要資訊，系統將擷取出來的活動事件結合電子地圖與時間軸，幫助使用者了解系統擷取出來的活動事件。

二、相關研究

根據 Sarawagi 的 Survey 常見的資訊擷取(Information Extraction)[2]任務包括實體(Entity)、關係(Relation)、屬性(Descriptor)、結構(Table, List, Ontology, etc.)等四類。而資訊擷取的方法有 Hand-coded 及 Learning-Based 兩種方式，可產生規則式(Rule-Based)或是統計式(Statistical Model)。

(一)社群媒體事件擷取系統

常見的事件擷取做法，主要是利用命名實體辨識(NER)的技術去識別文章中和事件相關的實體，並進行 association task 識別出實體間關係，接著透過人工建立的規則去擷取定義的事件。舉例而言，Tweets calendar 系統[3]的擷取的目標是 Twitter 上開放領域的事件，定義事件為(Entity, Event Phrase, Date, Type)，其目標是擷取人物、事件及日期等三種資訊，再將事件分類，組合成事件的 4-tuple 屬性得到如:(Steve Jobs, died, 10/6/11, DEATH)的事件資訊。其作法是標記完命名實體之後，利用卡方測定(Chi square)來做關係的驗證，以強化實體和時間關係並得出前 100、500、1,000 tuple，組合成完整的事件關係。換言之，事件必須被眾人多次提及，才有足夠資訊證明事件為真。

(二)新聞的事件擷取系統

另外，Wang[4]則提出一個應用於新聞媒體上的事件擷取系統，其目的是 5W1H 的語意層級的元素擷取系統，新聞事件的 5W1H 元素事件屬性定義如表一。方法是設計一些特徵(feature)從新聞標題中找出新聞中的主題句，接著透過語義角色標註，最後將擷取

出來的元素填入到 News Ontology Event model 供後續的利用。

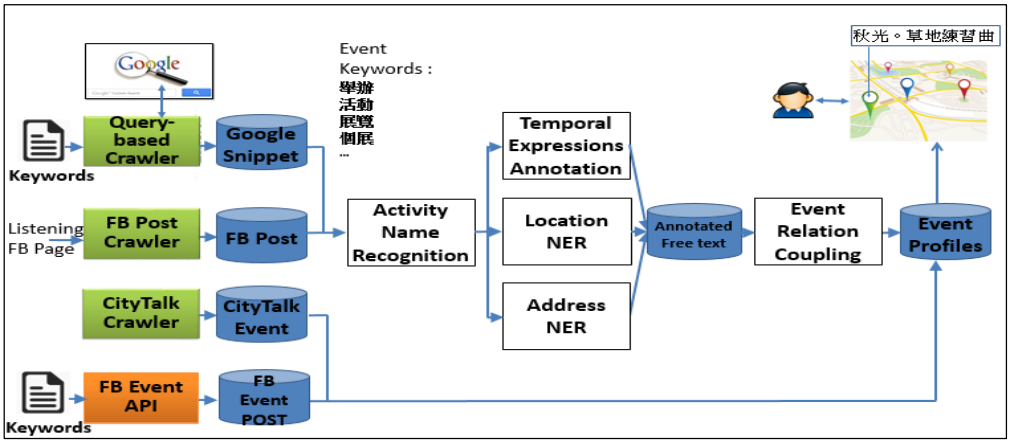
N. Kanhabua[5]提出的研究是跟公共衛生事件相關的事件擷取，主題是疾病爆發的事件擷取系統，疾病爆發的事件定義如表一，包括受害人、疾病、時間及地點，主要要解決的問題是找到疾病和疾病重要的時程表達式，定義這樣的問題為分類問題，並提出了相關排序(relevance ranking) 方法去判別重要的時程表達式。在不同分類方法中最好的效果是採用 J48，準確率(accuracy)能達到 0.65。

表一、新聞事件的 5W1H 元素事件屬性說明和疾病爆發的事件定義

5W1H	News 5W1H Event	Disease Event e: (v, m, l, t)
What	抵達	disease ***m***
When	8 日	time ***t***
Where	渥台華 加拿大	location ***l***
Who	中國國家主席胡錦濤	victim ***v***
Whom	加拿大首都渥台華	
How	中國國家主席胡錦濤抵達加拿大首都渥太華進行軍事訪問	

三、系統架構

本研究的系統架構如圖二，系統首先進行資料的蒐集，包括 CityTalk 和 FB 的發文，接著透過活動相關的關鍵字取得跟活動較為相關的發文，並利用活動名稱辨識的模型進行標記，只有包含活動名稱的貼文才會利用時程表達式、地點、地址辨識的模型進行標記。最後利用事件關係耦合的模組將發文的活動事件的關係找出來，並放到事件的資料庫，並提供介面給使用者查看擷取出來的活動資訊。



圖二、系統架構圖

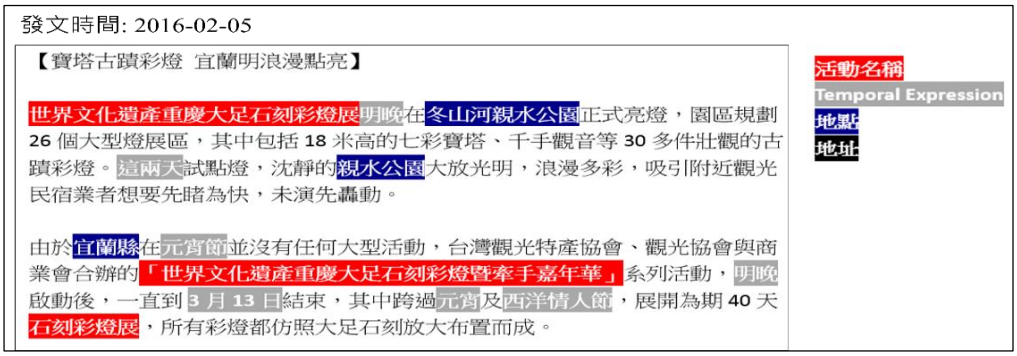
(一) 活動事件定義

活動事件的定義可由活動名稱、開始、結束時間、地點（或地址）四個基本元素組成，由於 FB 大部分提及事件的發文都只提到單一事件，因此本研究活動事件擷取任務即從每篇貼文中先行辨識活動名稱，再擷取活動的日期及地點。以圖三貼文為例，表二即為

活動事件擷取的輸出。

表二、活動事件的關係

Activity Name	Start Date	End Date	Location/Address
「世界文化遺產重慶大足石刻彩燈暨牽手嘉年華」	明晚 (2016-02-06)	3 月 13 日 (2016-03-13)	冬山河親水公園



圖三、事件擷取範例說明

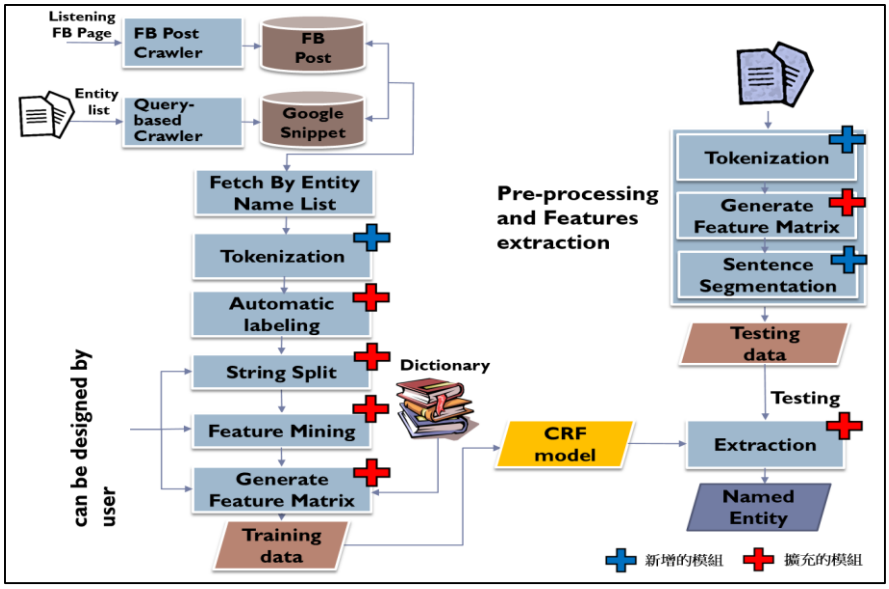
(二) 資料收集

在資料收集方面，系統主要會收集 FB、CityTalk 及搜尋引擎的活動關鍵字查詢結果。FB 網站上的資訊，包括打卡地點、台灣公開粉絲專頁、和 FB 事件的發文，其中打卡地點包括 245 萬 Places。由於 FB 沒有像 Twitter 提供對整個 FB 網站發文的搜尋 API，因此我們只能利用 FB Graph API 對 22 萬個台灣公開粉絲專頁(透過解析 1,400 萬筆 FB Object id 所得)，分別監聽關注的粉絲網頁取得發文資料。在 2015/9~2016/8 月間發文蒐集模組共收集粉絲頁的發文 2,947 萬篇以及和該發文的前 100 篇回應。另外系統也收集 FB 事件的發文，並利用爬蟲程式抓取 CityTalk 網站上的活動事件。

(三) 活動名稱實體的識別

活動名稱的模型是根據 CityTalk 網站蒐集回來的活動名稱做查詢詞，對搜尋引擎詢問結果，並經由自動標記得到訓練文本，活動名稱屬於長命名實體，所以歧義性等問題發生機會比較少，因此我們可以進行自動標記獲得大量標記的訓練文本。完整訓練過程和使用 Huang[1]的模型和改進部分可以參考下圖四，訓練模型的部分首先將我們爬蟲程式撈回來的文本透過 tokenization 模組將 token 做較好的切割，接著透過自動標記模組自動標記實體，標記出實體的文本我們不會整段使用而是透過 String split 模組只留 entity 前後固定長度的範圍，接著透過 Feature Mining 模組得到字典檔，透過這些字典檔經由 Generate Feature Matrix 模組產生 CRF ++訓練格式並訓練產生 CRF Model。在任務中改善 Huang[1]的排比（Alignment）方法，加入可自訂義的 tokenizer 模組，允許對 token 做

去詞幹設定。排比方式是參考 T.-S. Chen[6]提出的 Global alignment，針對長度大於 k 的活動名稱排比標記條件包括(1)不允許兩排比序列中的字元 mismatch 的對應，(2)兩相鄰 matched token 之間出現 Gap 數至多為 MaxG，且(3)重覆的 token 比例必須大於門檻值 r，滿足以上三者條件的比對系統即將其標記為出現範例。



圖四、活動名稱辨識模型建立和更新 Huang 工具的說明

(四) 工具的擴充

工具的擴充是為了改進原先 Huang[1]工具在 FB 文本效能不好的問題，除了上述改善長實體 Uni-Labeling 模組標記的準確性之外，並新增長實體排比 Full-Labeling 模組。較大的改變是擴充框架的可擴展性(scalability)，新增(1)不同資料來源的多線程(multithreading)標記和標記結果的倉儲、(2)支援 word-based 方法的標記（如表三）、(3)斷句模組，和(4)擷取出的實體精準度。

為了整併 word-based 和 character-based 方法並提供更豐富前處理功能新增 Tokenizer 模組，實作透過移植了 Lucene 的 Analyzer，可自行替換根據任務所需並自行設定詞幹設定。當採用 character based 時，我們使用 Jflex 這套工具定義常見的 token 型態(包括 money、alphanum、Chinese or Japan、URL、E-mail 等共 20 個)，而當採用 word based 時則預設使用 IK Analyzer，不過由於 IK Analyzer 會濾除未定義的 token，造成 token 遺失，因此我們改寫部分程式移除這樣的設定，並添加應用庖丁斷詞和 MMSeg 自定義的詞庫，如果要使用其它中文斷詞，可自行封裝斷詞來取代預設。標記精準度是指識別出

的實體和原文是一致的。我們要保證所有對文本做的前處理都不會影響識別出的實體。

表三、不同方法 tokenizer 自動產生的活動名稱辨識特徵值範例

ID	說明	長	character based	word based
…	…	…	…	…
10	常見於活動名稱前方的 token	1	到、在	舉辦、參加
11	常見於活動名稱前方的 token	2	舉辦、推出	推出/「、一年一度/的
12	常見於活動名稱前方的 token	3	活動：、參加「	活動/名稱/:、節目/名稱/:
13	常見於活動名稱後方的 token	1	】、」	即日起、開幕式
14	常見於活動名稱後方的 token	2	起跑、本次	表演/活動、明日/登場
15	常見於活動名稱後方的 token	3	來囉~	熱烈/開跑/囉、 /開幕/盛況
…	…	…	…	…

(五) 地點實體的識別

錯誤標記的地點會影響判定活動地點，對於地點辨識模型我們更看重精確率(precision)和實體邊界，精確率是不要錯誤地識別一些地點，採 CRF 方法召回率很高但識別錯誤例子也很多例如:“免費”識別成地點，另外識別出來的地點有時候邊界不是那麼完整如: “三灣鄉五穀廟前廣場”被識別為“三灣”和“廟前”，這些原因都可能造成地點轉 GPS 錯誤。另外還有原因是地點 NER 採 CRF 方法自動標記負擔太大。為此另外實做了配合 FBPlaceDB 標記方法的地點 NER 模組解決上述提到的問題。方法參考 Facebook Deduplicating a places [7]想法去實現地點 NER 模組。由於 FB Place 資料庫收錄的名稱有 245 萬，因此利用 Apache Solr 將打卡地點建成倉儲，同時為加快對句子的標記我們將句子切割成 n-gram (n=4~10)，分別查詢最相關的 k 個地點名稱 place。另外建立兩個跟地點及商家名稱有關的字典檔 CoreDic 及 LocBgDic：核心字典檔 CoreDic 是利用 MSRA 訓練出的實體(包含人名、地名、組織名)辨識模型，標記 92 萬個黃頁商家名稱中出現的實體，並刪掉出現頻率少於 6 次的實體名稱，做為核心字典檔(7,993 詞)；而地點商家背景字典檔 LocBgDic 收集方式則是將商家名稱經過中文斷詞後，取詞頻大於 500 且存在庖丁斷詞和 MMSeg 預設詞庫中但不屬於 CoreDic 的字詞(1,361 詞)。

利用這兩個字典檔，我們可以對每一個句子中的 n-gram 及其查詢到的 place 評分。給分原則為(1)避免跟 Location 或商家無關的字會得到分數。(2)當要標記的對象和資料庫的地點相關度判定分數超過門檻值，我們認為其實可視為相關，即可以地點標記句子。(3)比較好的實體邊界應取得較高的分數，如 n-gram1=“*清華大學旺宏館*”與 n-gram2=“*清華大學旺*”分別和 place=“*清華大學旺宏館*”匹配所得的分數，前者較後者為

高。完整的給分方式定義如下圖五。

其中 CoreFind 目的為找出輸入字串中可能出現的核心字詞集合(利用 MSRA NER 模型標記的實體加上 CoreDic 標記的字詞)，BgFind 透過中文斷詞並排除核心字，與 LocBgDic 交集得到的背景字詞集合，剩餘不屬於核心字和背景字的集合我們將其定義為描述字。CoreFind, BgFind 及 Descriptor 三個模組聯集所得的字詞數即是 Count 函數回傳值。演算法主要是計算出 n-gram 及 $place_i$ 共同的實體核心詞 CoreSet(去掉沒有出現在 $place_i$ 中核心詞)，以及共同背景詞 BgSet，並依 Eq.(1)計算 n-gram 與 $place_i$ 的相似度，若相似度大於門檻值，則用該地點 $place_i$ 進行 Partial Alignment 標記這個句子，進行最完整標記。

For each n-gram (n=4 to 10) in a sentence *s*

- Query FBPlaceDB to obtain top k place names
- For each $place_i$ from top k place names
  1. $CoreSet = CoreFind(place_i) \cap CoreFind(n\text{-}gram)$
  2. Find the Segment Core Entity based on common core, core length and frequency in CoreDic
  3. $BgSet = BgFind(n\text{-}gram) \cap BgFind(place_i)$
  4. Compute $Descriptor(n\text{-}gram)$ and $Descriptor(place_i)$, respectively
  5. $$NCBSim(n\text{-}gram, place_i) = \frac{\beta \times |CoreSet| + (1-\beta) \times |BgSet|}{\beta \times min(|Count(n\text{-}gram)|, |Count(place_i)|)} \quad \text{Eq. (1)}$$
  6. If ($NCBSim(n\text{-}gram, place_i)$ > threshold) then Label sentence *s* with $place_i$

圖五、FBPlaceDB 地點辨識標記演算法

舉例而言，句子 s1=「主辦單位：高雄市政府勞工局訓練就業中心」，利用不同的 n-gram 可查詢到“高雄市政府”、“新北市政府勞工局”、“勞工局訓練就業中心”、“斗六就業中心”、“高雄市政府勞工局勞工教育生活中心”、“高雄市政府勞工局訓練就業中心大寮職訓場域”等相關的 FB 打卡點，透過演算法最後句子中會標記到的地點為「高雄市政府勞工局訓練就業中心」，雖然這筆地點名稱不在 FBPlaceDB 中，但是因為與“高雄市政府勞工局訓練就業中心大寮職訓場域”經過排比，卻能完整的進行標記。

(六) 地址實體的識別和時程表達式標記模組

地址實體的識別模組主要應用是參考 2012 年 Chang 等人[8]提出的台灣地區地址擷取，使用 CRF 和配合表四 17 種地址特徵做訓練該模型，並配合極大分子序列演算法

（Maximal Scoring Subsequences）其F_1-score 約在 0.94 至 0.99 區間。另外本文研究主要文本擷取對象為 FB Post，但未來事件擷取的任務會擴充成 Web Data Extraction，為了相容HTML網頁(semi-structured)和FB Post (free text)採用Su [9]系統模組擷取台灣地址。

表四、地址擷取模型辨識特徵值

ID	Feature	ICCS	ID	Feature	ICCS
1	CountyCity	縣、市	**10**	ChineseNo	一、二、三
2	Township	鎮、鄉、區,	**11**	AllDigits	42011、0937137659
3	Village	村、里、鄰	**12**	DigitLen1	5、6
4	StreetRoad	道、路、街	**13**	DigitLen2	11、32
5	LaneAlley	段、巷、弄	**14**	DigitLen3	420、260
6	HouseNo	號	**15**	DigitLen4	5566、1234
7	Building	樓、室	**16**	DigitLen5	42011
8	ContactTag	地、址、電、話	**17**	DigitLong	327363、4227151
9	Punctuation	、、：、；			

活動時間在發文中會以比較口語的方式提及如: 本周六、明天下午等這些表達活動時間的陳述也是系統擷取的目標，標記的時間針對時程表達式進行標記，該模組參考Heideltime[10]，加入了口語的時間表達規則和正規化文字步驟（含全形字轉半形字），解決 FB 上書寫過於自由造成規則匹配失敗的問題，並利用匹配到的時間規則轉換成明確的年月日的日期格式，再修復跟週相關規則有關的臭蟲。

(七) 關係的耦合

這個步驟是將擷取出實體相關資訊，配對成完整的活動事件，我們採用規則式的方式，透過循序樣式探勘，快速找到和活動相關的文字特徵。由於透過系統標記活動名稱模組可能會得到多個活動名稱，我們將從中選擇系統認為最合適的活動名稱。挑選方法是針對每一個標記活動名稱 t 和系統標記的所有活動名稱 aSet 去排比，並得到共同交集比例最高的活動名稱做為標記的輸出。

雖然每個人介紹活動資訊的方式有很多種，但會有一些常見介紹活動資訊的語句，我們想法是找出這些語句出現的特定字，用其建立擷取活動時間的規則，我們的樣本數是包含時程表達式且是跟活動資訊有關的 40 萬句句子，接著透過循序樣式探勘(sequential pattern mining)，找出前 800 樣版，經由人工判定留下 79 Pattern，其中包括起始 51 個日期規則、14 個結束日期規則、以及 14 條 Date Confuse 規則，建立活動時間 Pattern 擷取規則。另外系統也手動建立一些輔助的規則以識別特殊的活動發文案例和識別活動地點，完整的制定規則在圖六。這些規則會有優先順序，對於比較完整的 Pattern 會有比

較高的優先權，如果同樣長度則看 Pattern 的觸發機率，計算方式是針對 Pattern s = $w_0 \dots w_{n-1}$在 40 萬時間相關句子去看 Pattern unigrams ，bigrams 的機率，如下 Eq. (2):

$$\text{Probability of s: } P(s) = P(w_0) \times P(w_1|w_0) \dots \times P(w_{n-1}|w_{n-2}) \qquad \text{Eq. (2)}$$

- Start Date rule
  - (r1)Prefix word: “於” | “活動時間” | “展期” | “活動日期” …
  - (r2)Suffix word: “起” | “舉辦” …
  - *(r5)Date Update: “活動改到” …*
- End Date rule
  - (r3)Prefix word: “截止日期” | “即日起至” …
  - (r4)Suffix word: “止” | “截止後” …
  - (r6)Date Update: “活動延長至” …
  - *(r11)Negative rule: “結標時間” | “原訂於” …*
- Special Date rule
  - (r7)Confuse rule: “販賣時間” | “徵件時間” …
- Location rule
  - *(r8)Prefix word: “於” | “假” …*
  - *(r9)Suffix word: “展出” | “舉行” …*
  - *(r10)LOC Update: “新活動地點” …*
- Negative Activity rule:
  - *(r12)Cancel: “活動取消” | “暫停舉辦” …*

註: 灰底是人工建立的規則

圖六、擷取活動時間、地點制定的規則

四、實驗

(一) 實驗資料集

實驗測試資料集透過隨機抽樣 FB 官方粉絲專頁的發文，挑選 1,300 篇文本中至少含有 1 個活動名稱的活動貼文。透過人工進行標記代表該篇文章最重要的活動事件關係，得到包含活動名稱、開始結束時間、地點或地址的活動事件關係資料集。另外將事件中提到的活動地點透過 FB Place 的資料庫對應到明確的 FB 上的打卡點取得人工判定的 GPS，或是透過 Google Map API 將活動地址轉成 GPS。另外我們也評估了系統命名實體模組效能，產生兩個估量命名實體的第一個資料集是將這 1,300 篇所有提到的活動名稱都進行人工標記，產生活動名稱辨識的資料集(2,132 Activity Name)，另外一個資料集則是將這 1,300 篇提到的地點和地址進行人工標記，產生地點辨識的資料集(2,015 Loc)，在此資料集我們會將組織和地點和地址做區隔，以反映出真實資料中三者實體重疊的情況。

(二) 命名實體辨識的效能

實體估量的算分方式，考量實體有時難以準確斷出邊界，以活動名稱舉例:「105 年基隆

市「主委盃」全國青少年 14 歲級網球錦 1 標賽（C-2）—基隆」可能只識別出部分「主委盃」全國青少年 14 歲級網球錦標賽」，參考 Huang 評估方法對於每個辨識到的命名實體 e 與正確答案的命名實體 a 進行命名實體辨識效能的評估，並定義 Eq. (3) *P(e,a)*、*R(e,a)*分數，加總分數後取平均值得到整體的 Eq. (4) Precision、Recall 與 Eq. (5) F_1-score。

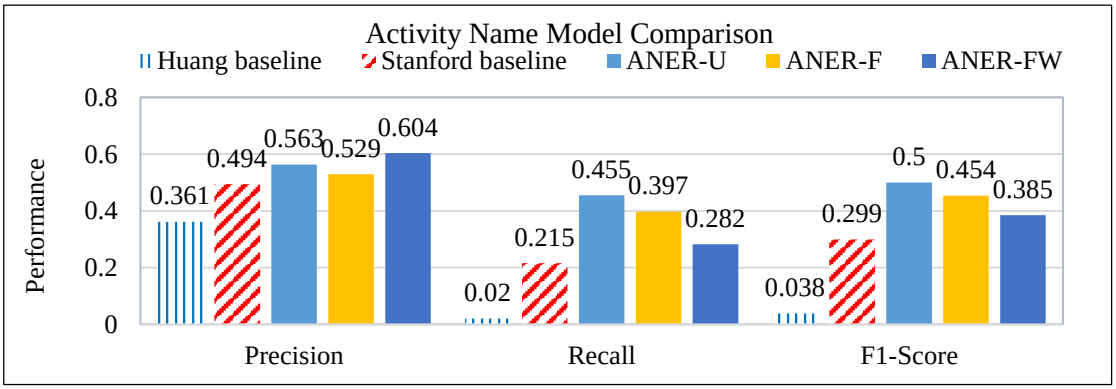
$$P(e,a) = \frac{|e \cap a|}{|e|}, \quad R(e,a) = \frac{|e \cap a|}{|a|} \qquad \text{Eq. (3)}$$

$$Precision = \frac{\sum P(e,a)}{|Identified\ entities|}, \quad Recall = \frac{\sum R(e,a)}{|Real\ entities|} \qquad \text{Eq. (4)}$$

$$F_1\text{-score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad \text{Eq. (5)}$$

(三) 活動名稱模型的評估

我們使用 CityTalk 11.7 萬個活動名稱(2015/12 前的活動)做為種子，查詢 Google 前 100 筆搜尋結果(絕大多數都不滿 100 筆且有可能找不到，造成 Google 自動將我們下的 Exact match 搜尋詞自動改成 Partial match)，得到約 67 萬個句子，利用 Uni-Labeling 標記活動名稱，提供給 Stanford 及 CRF++模型，利用活動名稱辨識的資料集評估結果如下圖七。



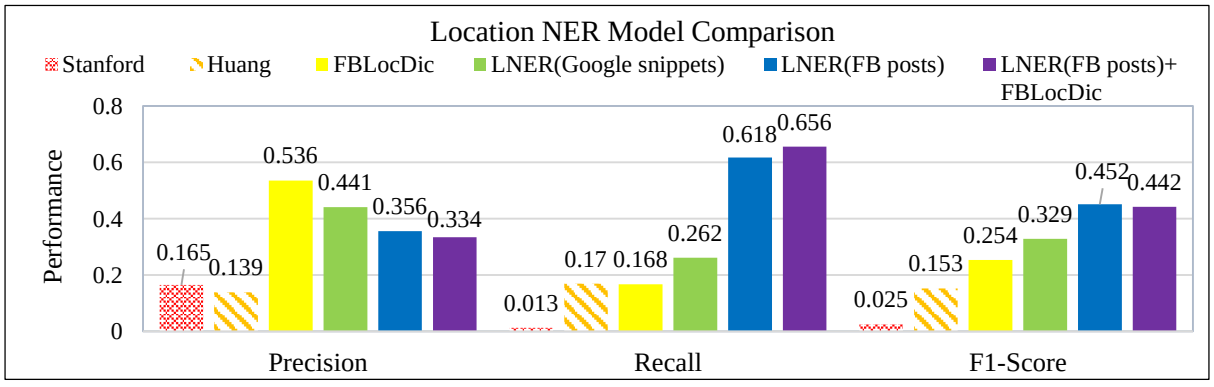
圖七、Activity Name NER Model 比較

分析發現給定相同的 Google snippets 文本進行自動標記，Huang 的工具在活動名稱(長實體採排比方式 Uni-Labeling)只能標記 2,221 個實體，而改進的排比方式 Uni-Labeling 則可標記 77,766 個實體，顯示透過新增 tokenizer 和重新實做的排比方式，能自動標記在拼音和非拼音的文本。並達到 0.5 的F_1-score，相對 Huang 與 Stanford 兩個 Baseline，均有大幅度的效能改進(0.038 及 0.299)。另外我們採 Full-Labeling 能標記到 195,400 個實體訓練文本並依 tokenizer 方法不同訓練兩個模型，比較 ANER-F 和 ANER-FW 這兩

個方法，word-based 有 Precision 較佳的優點，但在 Recall 較低。

(四) 地點模型的評估

我們使用打卡次數超過 1,000 次的 10.3 萬 FB 地點做為種子，查詢 Google 前 10 筆搜尋結果，以及 FB 貼文分別得到 17 萬，1,317 萬句子(FB 貼文搜尋採 Partial match 查詢，所以可能會得到查詢詞無關的文本，另外句子統計是查詢結果得到文本所含句子數量)，利用 Uni-Labeling 標記地點名稱，分別標記了 62,723 及 148,801 個地點實體，供給 CRF++ 訓練地點名稱辨識模型，其中 LNER(Google snippets)及 LNER(FB posts)是利用改善後的 Web NER Model Generation 工具配合 Google Snippets 和 FB 貼文的方法，FBLocDic 則為本文配合 FBPlaceDB 標記方法所提的方法。利用地點辨識的資料集評估結果如圖八所示，由於測試答案對地點和地址和組織的區分，且訓練種子檔有鄉鎮區和街道名組成的地點名稱，造成會標記到部分地址，普遍 Precision 都維持在 6 成以下，FBLocDic 精確率較高標記實體邊界也較準，但召回率過低是該方法的缺點。改進後工具以及 FB 貼文訓練所得的模型可以達到最好 0.618 的 Recall 以及最好的 0.452 的F_1-score。另外我們將兩者方法做結合的 LNER(FB posts)+FBLocDic 能將 Recall 提升到 0.656 且 Precision 僅有 0.02 下降。



圖八、Location NER Model 比較

(五) 活動事件擷取的效能

活動事件關係的估量方式先以每篇文章識別出個別實體進行評估，再定義整個活動事件的擷取效能。如前所述，由於 FB 上的活動貼文基本上是主述一個主要活動事件，因此給定一篇活動貼文及人工標記的活動、起始日期、結束日期以及地點四個答案(ActSet,

Start, End, Loc/Addr)[1]，根據答案含有 k 活動事件資訊，定義活動為 k-tuple (k=1 to 4)，若系統擷取出活動名稱 EAct、起始日期 EStart、結束日期 EEnd、以及地點 ELoc 或 EAddr，我們分別定義 Eq. (6)式個別實體擷取的 P, R 分數[2]。Eq. (7) 式為活動時間資訊判定是否正確的分數，我們將開始、結束時間分別估量。另外 Eq. (8) 為活動地點資訊正確的分數。再將這 k 個實體分數做平均便能分別得到 Eq. (9) EventPrecision 及 Eq. (10) EventRecall。活動事件關係評估定義如下:

$$P(ActSet, EAct)= \max_{a \in ActSet} \frac{|EAct \cap a|}{|EAct|}, \quad R(ActSet, EAct)= \max_{a \in ActSet} \frac{|EAct \cap a|}{|a|} \qquad \text{Eq. (6)}$$

$$I(Date, EDate) = \begin{cases} 1 & If \quad Date = EDate \; AND \; Date \neq NULL \\ 0 & If \quad Date \neq EDate \end{cases} \qquad \text{Eq. (7)}$$

$$MP(Loc, ELoc)= P(Loc, ELoc) \;, MR(Loc, ELoc)= R(Loc, ELoc) \qquad \text{Eq. (8)}$$

$$EventPrecision = (P(ActSet, EAct) + I(Start, EStart) + I(End, EEnd) + MP(Loc/Addr, ELoc/EAddr)) / k \qquad \text{Eq. (9)}$$

$$EventRecall = (R(ActSet, EAct) + I(Start, EStart) + I(End, EEnd) + MR(Loc/Addr, ELoc/EAddr)) / k \qquad \text{Eq. (10)}$$

$$EventF1 = 2 \times \frac{EventPrecision \times EventRecall}{EventPrecision + EventRecall} \qquad \text{Eq. (11)}$$

$$P(Act) = \frac{\sum P(ActSet, EAct)}{|Identified \; EAct|}, \quad R(Act) = \frac{\sum R(ActSet, EAct)}{|answer \; Act|}, \quad F1(Act) = 2 \times \frac{P(Act) \times R(Act)}{P(Act) + R(Act)} \qquad \text{Eq. (12)}$$

表五、不同 k 資料集個別的事件屬性和活動事件擷取效能

	F_1-score					Event		
attribute / F_1-score	#posts	Activity Name	Start Date	End Date	Loc/Add	Precision	Recall	F_1-score
1-tuple	45	0.766	NA	NA	NA	0.776	0.757	0.766
2-tuple	124	0.732	0.8587	NA	0.39	0.651	0.648	0.650
3-tuple	547	0.727	0.881	0.704	0.719	0.742	0.724	0.732
4-tuple	584	0.731	0.853	0.744	0.687	0.705	0.685	0.694
Total/Avg	1300	0.727	0.865	0.720	0.694	0.718	0.700	0.708

表六、個別的事件屬性和活動事件擷取效能

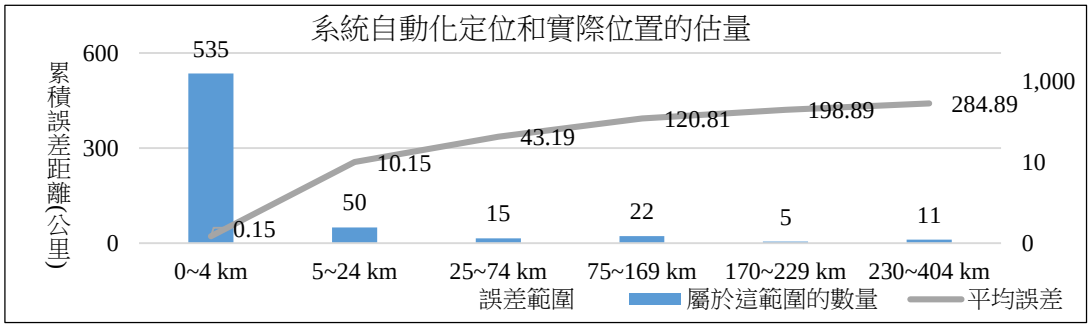
Item / Performance	Activity Name	Start Date	End Date	Loc/Add	Event
Precision	0.729	0.884	0.942	0.849	0.718
Recall	0.726	0.848	0.583	0.587	0.700
F_1-score	0.727	0.865	0.720	0.694	0.708

活動事件關係實驗結果表五、表六雖然在前一節中，個別實體名稱辨識效果只有 0.5 左右，但是由於活動貼文中可能以不同方式提及活動相關資訊，加以循序樣式探勘所得的

1 若文章無資料則標記為 NULL，於活動事件擷取效能時，不予計算。
2 目前系統預測事件屬性都只預測一個，取 max 為我們系統預測跟所有的答案標記去看能拿到的分數

擷取規則，因此我們整體的擷取效能可以達到 0.7 左右的F_1-score，對於起始日期更可達到 0.865 的F_1-score。

最後我們評估系統將活動地點投射到電子地圖上的位置，跟人工判定的真實座標的誤差。從 1,300 篇文章中，利用人工判定的活動位置 GPS 和系統自動化定位的 713 筆 GPS 進行實驗(目前系統每篇文章只會自動定位 1 個 GPS，自動定位 713 代表對 713 篇文章做自動定位)。扣除 75 筆系統進行自動定位、但無人工判定答案的 GPS，針對有答案的 638 筆 GPS 我們統計實際自動化定位和答案活動事件位置 GPS 的差距，將結果顯示在圖九。其中八成四的預測少於 4 公里(平均 0.15 公里)，與表六中的 0.849 的 Precision 相近。



圖九、活動事件預測活動位置(GPS)評估實驗

五、結論

本研究建構了一個 FB 事件擷取系統，並主動蒐集社群媒體資料，整合分散的粉絲頁發布的活動發文，擷取活動重要資訊，提供搜尋 FB 活動發文的功能，並將擷取到的活動事件在電子地圖上顯示方便使用者查看。在系統發展過程中，發現中文命名實體辨識模組仍有很大的進步空間，尤其是對於書寫較自由的 FB 發文，透過改善自動標記的方法，可以大幅改善標記的準確率和標記量，解決 Huang 在非拼音(中文)排比方法效能不好的問題。同時我們也加強文本前處理彈性以適應擴充和客製化。透過蒐集 FB 上的資料，可以有效的訓練我們的中文命名實體辨識模組。另外我們透過序列樣式探勘找出有用的特徵，輔助活動擷取的判斷，對提升準確率有很大的幫助。最後實驗透過人工標記的 1,300 篇發文評估命名實體辨識和事件擷取的效能。統計從 2015/6/23 截至 2016/8/8，針對 FB Post 進行活動事件擷取，系統在這段時間共截取了 11 萬 1,931 個活動事件(有同時提及活動名稱和活動時間抑或是同時提及活動名稱和活動地點/地址)。

目前系統對於每篇文章目前針對內文的單一事件去做擷取，但是如果文章提及多個事件，也需擷取文章中提到的多個事件。此外文章中提到的多個子活動也是擷取感興趣的目標，因為這些事件的子活動和子活動提及的描述和時間更能夠幫助人們快速了解該活動的詳情，如果能擷取這樣的資訊，就能提供人們活動排程功能和活動的推薦。此外像一些特殊情況發生例如颱風造成活動取消，系統應該註記活動因為何種原因取消，避免提供錯誤活動訊息。最後希望我們的任務能推廣到整個不只有社群媒體的 Web 上，從台灣的網站，政府、學校、售票網站公告自動化擷取活動公告，提供更完整豐富的活動訊息。

參考文獻

[1] Y. Y. Huang, C.H. Chung, "A Tool for Web NER Model Generation Based on Google Snippets," Proceedings of the 27th Conference on Computational Linguistics and Speech Processing, pp. 148–163, 2015.

[2] Sunita Sarawagi (2008), "Information Extraction," Foundations and Trends® in Databases, pp. 261-377, 2008.

[3] A. Ritter, O. Etzioni, and S. Clark, "Open domain event extraction from Twitter," Proc. SIGKDD, pp. 1104–1112, 2012.

[4] Wei, Wang "Chinese news event 5W1H semantic elements extraction for event ontology population," Proceedings of the 21st International Conference Companion on World Wide Web, pp. 197–202, 2012.

[5] N. Kanhabua, S. Romano, and A. Stewart, "Identifying relevant temporal expressions for real-world events," Proceedings of the SIGIR 2012 Workshop on Time-aware Information Access, 2012.

[6] T.-S. Chen, M.-C, Chen, and C.-H, Chang, "Efficient Page-Level Data Extraction via Schema Induction and Verification," Proceedings of the 15th Conference on Technologies and Applications of Artificial Intelligencester, 2014.

[7] N. Dalvi, M. Olteanu, M. Raghavan, and P. Bohannon, "Deduplicating a places database," Proceedings of the 23rd international conference on World Wide Web, pp. 409–418, 2014.

[8] C.H. Chung, C.-Y. Huang, and Y.-Y. Su, "On Chinese Postal Address and Associated Information Extraction," Proceedings of the 26th Annual Conference of the Japanese Society for Artificial Intelligence, 2012.

[9] Y.-S. Su, "Associated Information Extraction for Enabling Entity Search on Electronic Map," National Central University, 2012.

[10] J. Strötgen, M. Gertz, "Heideltime: High quality rule-based extraction and normalization of temporal expressions," Proceedings of the 5th International Workshop on Semantic Evaluation, 2010.