整合多種搜尋結果以提高 POI 搜尋的準確性

鄭仲庭
國立中央大學
glant19901021@gmail.com

莊秀敏
國立中央大學
showmin1205@gmail.com

張嘉惠
國立中央大學
chia@csie.ncu.edu.tw

摘要

隨著無線網路及行動裝置的普及，在地化服務(location-based services)逐漸受到重視，搜尋興趣點(Points of Interest, POI)已成為日常生活中常見的需求。雖然 Google Maps 是現今最廣泛使用的 POI 搜尋服務，但仍有許多 POI 在地圖上無法找到。因此，我們整合 Web 上多種搜尋來源結果，以有效地提高 POI 搜尋的準確性。

透過多種搜尋結果的整合，可以有效增加 POI 數量，即召回率提升。並且，依據 POI 與查詢詞的相關度排序結果可提高系統的準確性。在本篇論文中，我們整合了三種 POI 搜尋結果：(1)藉由 crawler 擷取 Web 中 POI，並利用 Solr 建構 POI 搜尋系統；(2)透過線上擷取模組，即時從 Google 搜尋 snippet 中辨識出 POI；以及(3)由 Google Place API 使 POI 數量更為豐富。我們對於排序結果考量了 POI 與查詢詞相關性，以及距離使用者位置遠近的兩種因素。另外，POI 與查詢詞不匹配以致檢索不到相關結果的問題，除了增加 POI 的相關描述來改善搜尋結果外，擴展查詢詞也是能提升召回率的方法。

本研究的系統架構區分兩個部分：第一部分為 POI 搜尋，藉由整合多種搜尋結果及 POI 相關模組進行排序。第二部分為查詢詞擴展，以 POI 相關資訊做為語料庫，經由主題模型分群詞彙，透過建立二分圖(bipartite graph)將詞彙和 POI 的標籤進行對應來擴展查詢詞。實驗結果顯示，本系統的 POI 搜尋效能優於 Wikimapia 與 What's The Number，並與 Google Maps 效能相近。

關鍵詞：興趣點搜尋、查詢詞擴展、學習排序。

1. 緒論

近年來 Web 蓬勃發展，電子地圖與黃頁網站已取代過去的紙本地圖與電話簿。在 GIR 領域上，OpenStreetMap[1]在 2004 年最早推出自由地理資料開放與分享協作的概念。接著，Google Maps 於 2005 年推出地圖搜尋服務。隨著行動網路與智慧裝置的普及，使得搜尋 POI 成為日常生活中常見的需求。

根據全球資訊網協會(WWW Consortium, W3C)對 POI[2]的定義，POI 是與位置相關聯的人造詞彙，包含的屬性如地址、經緯度、POI 名稱、類別、URL 及 POI 的描述，我們統稱為相關資訊。透過相關資訊使得 POI 在搜尋系統上更容易被索引。

目前已有不少熱門 POI 搜尋服務，如：Google Maps、食在方便、愛評生活通等，然而，仍有許多 POI 沒有被索引在這些服務中。使用者必須藉由搜尋引擎先找出包含有 POI 的網頁，剪貼其地址到地圖上再進行定位。這些繁瑣的動作對於使用者而言相當不便，也說明了目前的 POI 搜尋服務在數量上仍有不足，且集中於少數類別，如美食、旅遊。因此，為了提高 POI 搜尋結果的數量，我們整合了三種搜尋結果以改善 POI 搜尋的準確性。除了藉由 crawler 擷取網頁中 POI，並建構離線的 POI 搜尋系統[8][13]，我們也利用搜尋引擎找出相關搜尋結果後，利用線上擷取模組即時辨識出與查詢詞相關的 POI[6]，以補足離線資料庫的不足，並加入 Google Places API[3]查詢結果。然而，POI 搜尋必須考量搜尋範圍以及排序方式，甚至動態調整，因此是 POI 搜尋服務的一大挑戰。舉例而言，搜尋模式分為區域與全域搜尋，前者須給定搜尋範圍限制及使用者經緯度，以進行 POI 的空間搜尋(spatial search)，後者不須給定空間參數但無距離排序，因此，何時選擇區域或全域搜尋是我們要考量的問題。

POI 搜尋服務的挑戰在於辨識使用者的需求。當查詢詞為常見的 POI 類別或產品服務時，如餐廳、咖啡、百貨公司，由於符合的 POI 搜尋結果數量相對較多，為了讓使用者能快速找到符合需求的 POI，查詢詞基於使用者位置及搜尋範圍來檢索是有效的做法。另一種情況是當查詢詞為特定的 POI 時，使用者所在的位置和區域搜尋範圍內可能沒有相對應的 POI。例如：查詢詞為"西雅圖"，系統必須能自動擴大搜尋範圍，如"西雅圖美日語補習班"，或是"西雅圖咖啡"，抑或是美國的西雅圖，以提供使用者預期獲得的搜尋結果。另外，查詢詞通常相當簡短，且 POI 的相關資訊可能與查詢詞不相關或內容不足。為了提高系統檢索效能，我們設計了兩個方法來處理：POI 相關性預測模組以及查詢詞擴展的方法。

為了預測 POI 與查詢詞相關與否，我們將這個問題看作是二元分類問題來處理。為了訓練出高準度的分類模型，我們透過人工標記與使用者記錄來準備訓練資料，並利用查詢詞與 POI 名稱兩個文字向量之間的文字相似度做為特徵，訓練出 SVM 分類器以有效地過濾出與查詢詞相關的 POI 結果。

查詢詞擴展的目的在於提高搜尋結果的召回率。例如：給定查詢詞"小吃"，獲得建議詞如："黑輪"、"鹹酥雞"、"牛肉麵"。當 POI 具有相對應的標

[1] https://www.openstreetmap.org/#map=5/51.500/-0.100

[2] http://www.w3.org/2010/POI/wiki/Main_Page

[3] https://developers.google.com/places/webservice/search?hl=zh-tw

籤時，就能夠藉由查詢詞擴展被檢索出來。因此，為了更有效地提供使用者可能想要的搜尋結果，蒐集大量的使用者點擊記錄將能逐漸改善 POI 搜尋次數，並且結合查詢詞與所在的地理範圍，以提高使用者的搜尋滿意度。

在本篇論文，我們實作了 POI 搜尋系統服務，主要貢獻包含三項：(1)整合多種搜尋結果，並進行相關度與距離兩種排序方式以改善 POI 搜尋服務。(2)建構分類預測模型來辨識查詢詞和 POI 的相關度，實驗效能達到 0.932 (NDCG)。(3)利用 LDA 建構詞彙與標籤二分圖，將查詢詞擴展為數個相關標籤。

本篇論文的內容組織如下：第 2 章描述相關研究。第 3 章為系統架構，包含問題定義及方法設計。第 4 章為實驗設計與實驗結果。第 5 章提出結論和未來研究方向。

2. 相關研究

資訊檢索(IR)是探討如何自大量文件中找出符合使用者期望文件的方法，主要概念為計算查詢詞與各文件間的關聯程度，依據關聯程度由高至低排序[17]。傳統資訊檢索考量關鍵字於整體資料集中的稀疏程度，將查詢與文件表達成向量形式(Vector Space Model)再進行 cosine similarity 的加權計算。而 GIR 的研究一則可以資訊檢索領域的延伸來看待文件標的，一則是以地理位置為目標對象。Jones 與 Purves 發表在 GIS 期刊的研究中提出發展地理資訊檢索系統所要面對的幾項挑戰[12]，包括：(1)在使用者的查詢詞與文件集中偵測符合限制範圍的搜尋地點。(2)去歧異(disambiguating)地名。(3)解釋模糊的地理量詞。(4)索引與地理資訊相關文件以及非空間主題的內容。(5)搜尋結果的排序，對於主題相關性與地理範圍。(6)設計使用者搜尋介面，幫助使用者找到需求的資訊。(7)評估地理資訊檢索效能的方法。換句話說，以文件的角度來處理地理資訊檢索，更著重於理解查詢詞的限制、非結構化文件的內容擷取。

因此，GIR 與 IR 最大的差異在於 GIR 以地理位置為目標對象，搜尋與使用者查詢詞相關的地理位置及其相關資訊，排序結果除了依據查詢詞的相關程度外，也須考量使用者查詢詞的範圍限制與地理上的相關量詞。相對於 IR 僅傳回相關或部分相關的網頁內容，GIR 更需考量傳回符合條件的 POI 相關資訊，以及在地圖上標示實際位置。

相對於 GIR 的檢索單位為網頁，POI 搜尋是以 POI 為搜尋單位。Ahlers 與 Boll 基於地點的 Web 搜尋研究[2]，提出從網頁中擷取地點的系統架構。而 Chuang 等人[8]及 Kao[13]亦提出黃頁爬取以及基於查詢詞的策略來擷取 Web 中包含有地址的 POI，以建構離線資料庫。在線上即時擷取 POI 的部份，Chang 等人利用 Google 搜尋引擎及時蒐集關於查詢詞的前十筆 snippets[6]，再藉由地址擷取模型與店名辨識模型擷取出相關的 POI。實驗數據顯示，使用 188 個 POI 做為查詢詞進行前 50 筆搜尋引擎結果的驗證，其中 77%的 snippets 包含有正確的 POI，14%的 POI 不包括在 Google Maps 資料庫。這裡的 77%的涵蓋率有可能再提高，如：增加搜尋深度至前 100 筆結果、深入拜訪每筆搜尋結果的頁面進行 POI 擷取。另外[3]和[18]從 Wikimapia、旅遊網和 Twitter 爬取大量 Web 資料，將關鍵字和地點連結，藉由熱圖(heat map)呈現搜尋結果的相關度以提供互動式搜尋服務。相似地，[15]整合 Web 上多種資料如 Foursquare[4]、Facebook[5] Places、Google Maps[6]、DBpedia[7]和 LinkedGeoData，以提供使用者位置事件搜尋。

在檢索服務的方法上，近年來一些學者利用分類、學習排序或使用者點擊紀錄來改善檢索效能的研究[11]。L. Aleksandra 等人[1]利用 Yandex 搜尋引擎中大量使用者紀錄，過濾搜尋結果中可能回傳的不相關的文件。特徵設計包含了四種類型：數量、文字、連結、其他用來訓練分類器。主要目標為過濾掉不相關的文件以降低負例對於使用者影響。Costa 等人提出以學習排序問題解決 Web 中存在不同版本文件的檢索問題[9]。WAIR (Web Archive Information Retrieval)的資料集隨著時間的推移被索引，因此每份文件可能存在不同時間的多個版本。作者收集使用者紀錄並依照時間區段做切割，將這些基於時間的特徵資料用來訓練學習排序模型，並加入了時間權重，用來學習排序出最可能的搜尋結果。

Liu 等人提出建議查詢詞的研究[14]。由於使用者在 Web 搜尋時，可能使用不適當的查詢詞而無法搜尋到想要的結果。因此作者利用學習排序的方法進行查詢詞建議。利用分群方法取得與查詢詞相關的候選詞，接著透過 Bing 搜尋引擎蒐集查詢詞和候選詞的搜尋結果，擷取其特徵用來訓練學習排序模型。本篇論文與 Liu 等人的做法差異在於我們透過主題模型 LDA[4]獲得與主題相關的詞彙，並建立詞彙與標籤的關聯。藉由 POI 所對應的詞彙，再推薦給使用者相關的標籤。

推薦系統也常與 LBS 結合，用來推薦使用者在地化的資訊和服務[16]。Noguera 等人提出了依據使用者軌跡推薦 POI 的方法，智慧型行動裝置被用於推薦給遊客下個 POI 的選擇，以及感測使用者的移動用來改變候選 POI 的內容。他們將混合式推薦引擎和移動式三維的 GIS 架構整合，透過特徵分析並收集使用者的評分，以調整系統的整體效能。

3. 系統架構

本系統為有效改善 POI 搜尋服務，架構包括兩部分：第一部分為 POI 排序以及第二部分關鍵字與標籤關聯，如圖 1 所示。給定一查詢詞，系統先透過三個搜尋引擎(Solr、online search、Google Place

[4] https://foursquare.com/
[5] https://developers.facebook.com/docs/reference/api/search/
[6] https://developers.google.com/places/
[7] http://wiki.dbpedia.org/

API)檢索出相關結果，接著相關度模型預測排序出相關的 POI 結果。在查詢詞擴展部分，經由 POI 語料庫中的詞彙與 POI 的標籤所對應的二分圖，查詢出與查詢詞相關的前三組標籤。

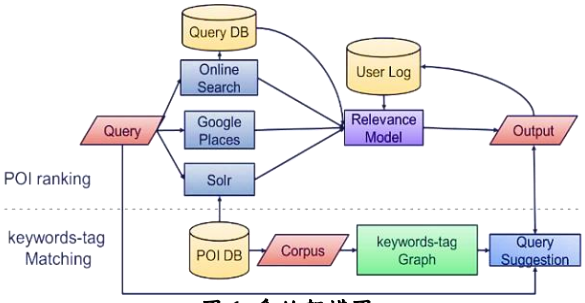

圖 1 系統架構圖

3.1 搜尋結果整合

為了有效提高 POI 搜尋結果的數量，以補充現有資料庫的不足，我們整合三個蒐尋引擎結果，包含：(1)Solr 檢索系統、(2)線上 POI 擷取模組，以及(3)Google Maps API 查詢結果。

Chuang 等人利用基於查詢爬蟲與黃頁爬蟲兩種策略進行 Web 上台灣 POI 資料的擷取[8]，目前資料庫中存有 140 萬筆 POI 資料。為有效索引 POI 資料庫，我們利用開放源碼的 Solr[8]全文檢索軟體以提供 POI 搜尋服務。Solr 4.0 在空間搜尋功能(spatial search)上提供區域及全域搜尋兩種模式，即給定查詢範圍限制為區域搜尋，反之為全域搜尋。區域搜尋排序依據給定的座標與查詢詞，找出相關的 POI 並進行距離排序，全域搜尋則依據查詢詞與 POI 的相關度排序而無距離排序。因此，對於 Solr 搜尋的挑戰在於區域搜尋與全域搜尋模式的選擇。為了有效地回傳 POI 搜尋結果並排序，我們會優先選擇區域搜尋；然而其限制在於當搜尋範圍內找不到符合的結果時，將擴大搜尋範圍以再次搜尋 POI。

為了有效獲取即時 Web 上相關的 POI 資料，Chang 等人提出利用線上擷取 POI 模組以彌補現有資料庫之不足[6]。線上擷取模組主要是透過 Google 搜尋引擎蒐集與查詢詞相關的前十筆搜尋結果。接著，利用地址擷取模組[7]與店家辨識模組[10]以擷取出相對應的 POI，再選擇出最可能的 POI 配對。最後，擷取出的 POI 地址透過 Google Geocoding API[9]轉換為地圖上的經緯度。這項搜尋來源的限制是同一 IP 不能頻繁地向 Google 搜尋引擎蒐集資料。另一項限制在於 Google Geocoding API 地址經緯度的轉換次數一天不能超過 2,500 次。

Google Maps 是現今最廣泛使用的 POI 搜尋服務。為有效取得 Google Maps 資料，我們透過 Google Places API 發送請求以取得 POI。此部分限制亦為一天請求次數為 2,500 次，且須給定搜尋範圍，即 API 不提供無限大的範圍搜尋。

當三種搜尋引擎結果回傳到系統時，去除重複的 POI 是一項重要的前處理工作(deduplication)。在本篇論文中，我們先將所有 POI 資料進行地址正規化，接著對於同一地址所對應的 POI 進行字串比對以去除重複的配對。在未來研究上，我們將分析更多去重覆的方法以解決複雜的 POI 配對問題。

3.2 POI 排序

為了有效地整合及排序多個 POI 搜尋結果，以及考量 Solr 的區域搜尋與全域搜尋模式限制，我們利用搜尋演算法對於多個搜尋引擎結果進行排序。並且，透過分類及排序的方法將預測 POI 相關度的問題定義為二元分類及學習排序問題，用以辨識出相關與不相關的 POI，進而排序搜尋結果。

3.2.1 搜尋演算法

由於三個搜尋引擎結果的整合上有其限制(詳如 3.1 節)，因此我們設計了搜尋演算法目的在於有效率地將多種搜尋引擎結果整合。在演算法 1 中，我們考量了 POI 與查詢詞的相關度，以及 Solr 的區域與全域搜尋模式的選擇。起初，給定參數包括：查詢詞 q、搜尋範圍 r、使用者座標 GPS、相關度門檻值δ以及累計次數 i，其中 i 給初始值，例如 5。接著，系統藉由三個搜尋引擎收集與查詢詞 q 相關的 POI 並取其聯集。利用相關度模型預測聯集中的 POI 與查詢詞的相關度。若 POI 相關度大於δ，則放入集合 C 中降冪排序；若存在多筆 POI 相關度相同，則依 POI 距離使用者遠近排序；若 C 為空集合，則擴大搜尋半徑 r 的三倍範圍再次搜尋，此時 i 累計次數減 1。當 i 為 1 時，則不再設定搜尋範圍，(即搜尋範圍為 3^5=243 公里，約全台灣)，此時即為 Solr 搜尋模式的全域搜尋；若 i 為 0 則演算法結束。演算法最後的輸出結果為預測排序後的 POI 列表。

Algorithm 1. ***Search*** (q, r, GPS, i, δ)

```
Input: user query q, user's GPS, search scope r
Output: POI list
Initial: i > 0, δ = 0.5
If (i = 0) EXIT
MS = Solr ∪ Google Place API ∪ Online search
C = Ranking (MS, δ)
If (C = null)
    Search(q, r×3, GPS, i-1)
Else
    C order by the confidence and the distance
```

對於多種POI搜尋引擎的結果，有時仍會找到不相關的結果或部分匹配的問題。例如Google Places API的搜尋結果，可能包含有不相關的POI。例如：查詢詞“理髮店”，會找到“屈臣氏”。分析其原因可能為crowdsourcing標記錯誤或是屈臣氏的描述中包含有“理髮”的敘述。對於Solr的搜尋結果，若查詢詞中包含有引號，如“牛肉麵”，我們可以找到完全匹配的POI，但數量可能是不足的。反之，若查詢詞中不包含有引號，則可能獲得大量的POI

[8] http://lucene.apache.org/solr/
[9] https://developers.google.com/maps/documentation/geocoding/intro

但不一定相關的POI。因為Solr對查詢詞的每個字分別比對查詢，即“牛”、“肉”、“麵”，所以相關度預測模型用來過濾掉不相關的POI搜尋結果。對於線上擷取模組而言，透過Google搜尋引擎將“查詢詞”與“城市”組合所找出的搜尋結果中，可能擷取出不正確或不相關的POI。因此，相關度預測模型的目的在於將不相關的POI過濾或排序到最後。

3.2.2 學習排序

我們將預測查詢詞與POI相關度的問題看成二元分類的問題以及學習排序的問題來解決。首先，給定查詢詞q與POI名稱T，判斷POI名稱T與查詢詞q是否相關，即 1 為相關，否則為 0 表示不相關。並且，我們利用資訊檢索的學習排序問題，判斷查詢詞q所對應的數個POI的排序問題。給定查詢詞q與多個POI，$p_1, p_2, \ldots, p_n$，根據POI與查詢詞q的特徵，排序出POI之間的關係。在本篇論文中，我們利用LibSVM[6]和RankSVM[11]工具來訓練二元分類模型與學習排序模型。

透過LibSVM所訓練的分類模型，除了預測查詢詞和POI是否相關外，並輸出各POI預測相關度的信心值，介於0與1之間。為了過濾掉不相關的POI，給定相關門檻值δ=0.5，當相關度低於δ的POI都不會出現在搜尋結果中。接著，依據POI信心值，將搜尋結果降冪排序，當多組POI信心值相同時，再依據使用者與POI間的距離由近至遠排序。

另一方面，利用RankSVM所訓練的學習排序模型，亦提供預測是否相關的信心值。RankSVM與LibSVM使用策略的差異在於與RankSVM所預測的信心值不是用來過濾相關度低的POI。換句話說，所有的POI搜尋結果依據POI信心值降冪排序。使得相關度高的POI排在較前面的位置，相關度低的POI排到後面的位置，因此保留了更多的搜尋結果。

在訓練資料的部份，我們準備了人工標記和使用者紀錄，並擷取查詢詞與POI名稱T的六組特徵(如 3.2.3 節)。在人工標記資料上，我們標記了 200 個查詢詞所對應的 2,000 個POI，即每個查詢詞q對應到 10 個POI，每對可以表示為pair(q, POI$_i$)。我們擷取其特徵做為訓練資料，再由人工標記每對查詢詞和POI的類別，若相關則標記為 1，否則為 0。對於使用者紀錄資料，我們選出 113 個使用者點擊紀錄，其中包含有 63 個查詢詞。將查詢詞與其POI搜尋結果進行對應，若POI搜尋結果中包含有使用者的點擊紀錄，則該配對視為相關，否則為不相關。總計資料對數有 270 對。

3.2.3 特徵擷取

為了有效地評估查詢詞q與 POI 名稱T的相關度，我們將q和T視為兩組文字向量，進行相似度計算。設計特徵如表一，包含了字串匹配、字串位置匹配、cosine 相似度、最長共同子字串(LCS)分別在查詢詞與 POI 的比例，以及使用者點擊次數。六個特徵詳述如下。

表一、訓練 POI 相關模型的特徵

id	Name	Descriptions
1	MatchWord	$\sum_{i=1}^{n} \frac{MatchW(q_i,T)}{log2(i+1)}$
2	MatchPosition	$\sum_{i=1}^{n} \frac{MatchP(q_i,T_i)}{log2(i+1)}$
3	Cosine(q,T)	$\frac{\sum_{i=1}^{\max(m,n)} V_{q_i} * V_{T_i}}{\sqrt{\sum_{i=1}^{n}(V_{q_i})^2} * \sqrt{\sum_{i=1}^{m}(V_{T_i})^2}}$
4	RatioLCS$_q$(q,T)	$\frac{LCS(q,T)}{length(q)}$
5	RatioLCS$_T$(q,T)	$\frac{LCS(q,T)}{length(T)}$
6	Click-through	$\frac{CT_{POI}}{CT_{Maximum}}$

(1)字串匹配：此特徵用於計量查詢詞 q 的字元出現在 POI 名稱 T 的個數，即估算查詢詞和 POI 的匹配程度。字元匹配函數定義為：

$$\text{MatchW}\ (q_i, T) = \begin{cases} 1, & q_i \subseteq T \\ 0 & \end{cases}$$

由於 POI 通常位於前面字的特殊性較高，例如：誠品書局的“誠品” 相對於“書局”更能凸顯匹配程度，因此，我們依據匹配字的位置給予不同的匹配分數。即匹配分數最高為 1，依位置比例遞減。

(2)字串位置匹配：第二個特徵與第一個特徵相似，差異在於估算查詢詞和 POI 名稱 T 的匹配字元是否位於相同的位置上。即字元匹配函數定義為：

$$\text{MatchW}\ (q_i, T_i) = \begin{cases} 1, & q_i = T_i \\ 0 & \end{cases}$$

(3)Cosine 相似度：將查詢詞與 POI 名稱表示為向量 V_q和V_T，用 cosine 相似度估算兩文字相似度距離。

(4)最長共同子字串在查詢詞 q 的比例：找出查詢詞 q 和 POI 名稱 T 的最長共同子字串，並計算最長共同子字串在查詢詞 q 中的比例。

(5)最長共同子字串(LCS)在 POI 名稱 T 的比例：與前者相似，差異在於計算最長共同子字串在 POI 名稱 T 中的比例。

(6)使用者點擊次數：根據使用者點擊紀錄，計算查詢詞 q 與 POI$_i$ 被使用者點擊的次數，並正規化使數值分布在 0 到 1 之間。

3.3 查詢詞與標籤的關聯

為了建構查詢詞擴展關係架構，我們需要獲得詞彙與標籤(即子類別)的對應關係，進而推薦新的查詢詞給使用者。如圖 2 所示，查詢詞 q 所對應的 POI，其對應到的標籤可做為新的查詢詞提供給使用者，使其找到更多相關的 POI 結果。因此在關係圖的建構上，查詢詞 q 對應到 POI 的關係，以及 POI 對應到標籤的關係成為擴展查詢的關鍵。

在本研究中，我們的詞彙來源為 POI 資料庫中每筆 POI 的描述句，經中文斷詞處理後，篩選名詞做為語料庫詞彙 W。在對應標籤上，由於 POI 資料庫中已給予每筆 POI 主類別及子類別的標籤，因此我們將 POI 的子類別定義為標籤 Tag。利用 W 做為

查詢詞，找出相對應的 POI，再檢索出 POI 所對應的類別標籤 *Tag*，即為詞彙和標籤的對應二分圖。

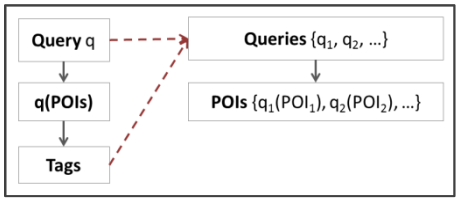

圖 2 查詢詞擴展的概念

3.3.1 詞彙生成

首先，我們將 POI 資料庫 29 個類別中的 POI 描述句，使用史丹佛自然語言處理工具[10]進行中文斷詞後篩選出名詞儲存為語料庫。接著，利用 LDA[4]將語料庫中的詞彙進行主題分群。LDA 的主題模型在於推估出文章/句子中所包含的主題分布機率。換句話說，相似概念的詞彙在相同主題下的機率就會較高，因此，利用 LDA 將相關主題或經常一起出現的詞彙進行分群。LDA 的主題數與參數的給定上，我們利用 perplexity 來評估不同主題數下機率分布的 entropy。perplexity 越小代表模型越好，因此我們用 perplexity 來調整 LDA 的 α、β 參數和主題數 k。對於 29 個類別的語料庫，我們各自訓練出最佳的主題數 k，以及對應的 α、β 值，$0 \leq \alpha, \beta \leq 1$。並且，從最佳的主題數中，選出每個主題下，前 100 個機率高的詞彙做為查詢詞進行 POI 搜尋。當查詢詞的主題類別與 POI 的類別相同時，詞彙與 POI 的對應關係就完成，否則不成立。如：“美食”語料庫中的“老店”為查詢詞搜尋到“十元老店”，因為該筆 POI 屬於“購物”類別，所以過濾掉。因此，詞彙與 POI 的對應關係如圖 3。

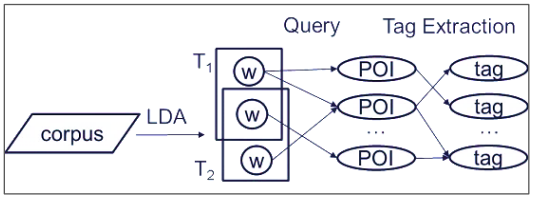

圖 3 詞彙與標籤對應流程

3.3.2 詞彙標籤二分圖建構

由於這些 POI 已具有類別及數個子類別標籤，所以我們利用上一節所生成的詞彙做為查詢詞，搜尋出所有 POI，再過濾掉和查詢詞屬於不同類別的 POI 後，將各筆 POI 的子類別列出。因此，詞彙與標籤對應關係二分圖透過 POI 搜尋完成對應關聯。

在詞彙標籤二分圖中，每個詞彙和標籤的對應關係被表示為一個具有權重的邊，其中權重值的給定為：不同主題下詞彙 *w* 對應到標籤 *tag* 的數量加總。換句話說，權重越高的邊，代表了詞彙 *w* 與標籤 *tag* 的顯著關聯。例如：用“漢堡”查詢到許多早餐或美食主題的 POI，因此對應到“速食”、“快餐”、“美式餐廳”等標籤，若對應的次數排序關係為“速食”>“美式餐廳”>“快餐”，則當查詢詞為“漢堡”時，系統會優先推薦“速食”給使用者。另外，對於使用者給定的查詢詞 Q，我們會先進行斷詞，即 $q_i \in Q$，當查詢詞 q_i 屬於詞彙 W 時，我們會推薦查詢詞 q_i 對應的前三項高權重的標籤給使用者。

本系統中查詢詞與標籤的對應關係為離線建構，即 POI 所對應到的相關標籤已事先處理。因此，查詢詞經過三種資料來源搜尋排序後的 POI 結果，再透過各 POI 所對應的標籤做為下一輪的查詢詞。當使用者點擊推薦詞的時候，表示使用者可能尚未找到滿意的 POI，因此系統將新的查詢詞推薦給使用者。由於相關主題下使用不同查詢詞所找出的 POI 結果不盡相同，因此查詢詞擴展的概念即為變更不同查詢詞以找出符合使用者需求的 POI。

4. 實驗

為了驗證本研究所提出的整合多個搜尋引擎能有效改善 POI 搜尋效能，實驗分為兩部分：第一個實驗是整合多搜尋引擎方法與其他搜尋服務的比較；第二個實驗是由使用者點擊次數驗證查詢詞擴展的效能。本實驗中所採用之環境參數如下表。

表二、實驗環境參數給定

符號	值	描述	符號	值	描述
r	3	搜尋半徑。	α	0.05	LDA 參數，α=0.05 收斂
i	5	搜尋次數，初始給定 5。	β	0.02	LDA 參數，β=0.02 收斂
δ	0.5	相關度門檻	*k*	48	LDA 主題數

4.1 資料集與評量方法

POI 搜尋結果(multiple sources, MS)包含三個來源的整合：分別是 Solr 檢索系統、線上 POI 擷取模組，以及 Google Place API。第一種搜尋引擎為 Chuang 等人[5]在 2013 年 7 至 8 月爬取中華黃頁[11]和愛評網[12]網站，POI 數量共計 140 萬筆，其中包含有 29 個主類別，主類別項下包含有 1,290 個子類別。

為了驗證 POI 搜尋結果在數量上及 POI 排序上的效能，我們採用常見的 POI 排序評量指標 NDCG (normalized discounted cumulative gain)進行評估。公式定義如下：

$$DCG@k = \sum_{i=1}^{k} \frac{2^{r(i)} - 1}{log_2(1+i)}$$

$r(i)$表示文檔在位置 i 的相關程度，當 POI 排序位置越後面代表相關分數越低。相關度的給分標準區分為兩個等級，即相關 1 與不相關 0。本實驗 k 為 10，即評估前十筆 POI 排序結果。最後正規化 DCG 值使數據分佈在 0 到 1 之間。

[10] http://nlp.stanford.edu/software/stanford-dependencies.shtml

[11] https://www.iyp.com.tw/

[12] https//www.ipeen.com.tw/

在查詢詞給定上，我們採用了與日常生活相關的 20 個一般查詢詞，例如：飯店、民宿、加油站，以及 20 個特定查詢詞，例如：麥當勞、星巴克。對於搜尋區域，我們選擇四個市區，分別為台北、桃園、台中、高雄的火車站，以及四個郊區，分別為暨南大學、中正、屏東科大、東華大學。在結果標記上，採用人工驗證搜尋結果，即找到的 POI 和查詢詞相關給予 1 分，否則 0 分。

4.2 POI 搜尋效能

為了比較我們所提出的整合多搜尋引擎的方法與其他 POI 搜尋服務的效能，三個比較對象分別為：Wikimapia、店家搜尋服務 What's The Number 及 Google Maps。我們分別進行 40 個查詢詞以及 8 個搜尋位置的實驗。實驗結果顯示，我們的系統效能遠優於 Wikimapia 和 What's The Number，並且與 Google Maps 效能相近，如圖 4。LibSVM 及 RankSVM 預測結果分別是 0.932 及 0.920，說明了我們 POI 搜尋結果的數量或是排序，能有效地擷取一般搜尋服務中所不足的 POI 數量，以及準確地排序 POI 結果。透過線上即時擷取的方法，使我們效能略超越 Google Maps，其原因在於我們能擷取 Web 中有提及的 POI，而沒有被標記在 Google Maps。

為了瞭解 POI 搜尋服務提升整體效能的原因，我們進行了資料整合與排序後的效能比較，如圖 5。四個比較資料分別為：爬蟲從網頁中離線擷取 POI、三種資料來源的整合、以 libSVM 預測整合後資料的相關度排序，以及使用 rankSVM 預測整合後資料的相關度排序。比較資料來源數量，整合多個資料來源後的平均準度(0.709)的確較單一網頁擷取(0.65)提升了 5%效能，然而搜尋結果並未真正提高 POI 搜尋準度；經過 libSVM 及 rankSVM 兩種方法分別預測相關度後排序的搜尋結果，則顯著地提升效能至 0.932 及 0.92。因此，從本實驗中可看出 POI 名稱與查詢詞的相關度為影響 POI 搜尋準度的主要原因。而 POI 名稱中若能與查詢詞有較多的相同字，則預測的相關度越高；反之，若 POI 名稱與查詢詞完全不同，則可能在搜尋結果中無法被找到或排序到較後面的搜尋結果。

4.3 查詢詞擴展效能

第二個實驗目的在於驗證查詢詞推薦的效能。圖 6 為 35 位使用者在 600 個查詢詞所推薦的的標籤的點擊率。藍色為各個標籤的點擊率，綠色為累積標籤點擊率。實驗結果顯示，第一個標籤的點擊率(38.04%)相較於第二標籤與(30.68%)及第三個標籤(24.79%)更貼近於使用者想要的結果，因此驗證了查詢詞標籤二分圖的建構，即權重越重的詞彙標籤關係顯示與查詢詞的顯著相關。從累積標籤點擊率的結果顯示，每次推薦三個標籤的總點擊率為 53.25%，這意味著使用者對於每次推薦的標籤有超過一半的機率點擊。相較於一般的推薦系統，我們的初步效能已讓使用者感到興趣。

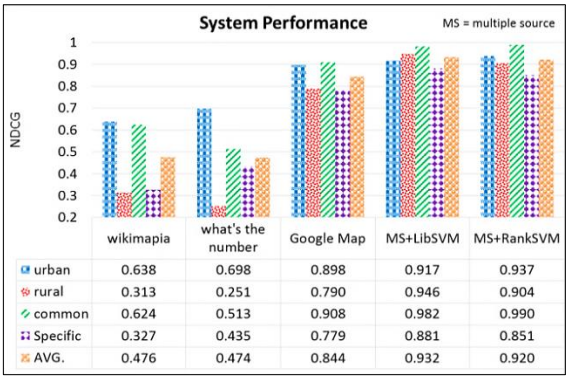

圖 4 POI 搜尋效能比較 (MS:多搜尋結果)

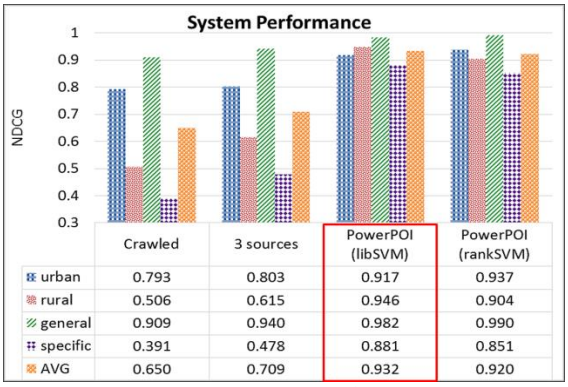

圖 5 POI 效能提升分析

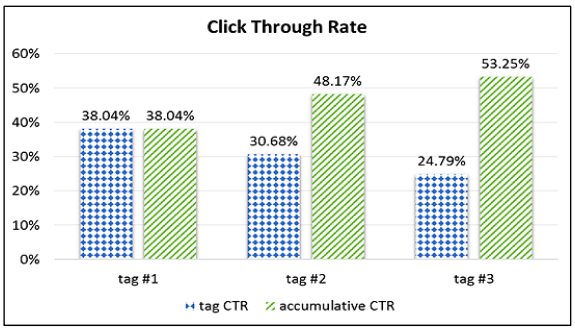

圖 6 查詢詞擴展的標籤點擊率

5. 結論

在本篇論文中，我們提出一個整合多個搜尋結果的 POI 搜尋架構以提高系統效能。藉由離線 POI 資料庫、線上 POI 擷取模組，以及 Google Place API 的搜尋結果整合及排序策略，提高了 POI 搜尋效能。我們利用分類器及學習排序方法，並設計查詢詞與 POI 相關的特徵來訓練預測模型，以有效地排序 POI 搜尋結果。實驗結果顯示，我們的系統效能(NDCG=0.932)比 Wikimapia 和 What's The Number 有更好的表現，並且與 Google Maps 有相近的效能。在查詢詞擴展上，推薦詞有超過一半的機率獲得使用者的同意。在未來研究方向上，社群媒體資料，如 Facebook 及 PTT，以獲得更多 POI 與在地化資訊，將能提供更多實務上有用的搜尋服務。另一方面，POI 資料存在許多歧異的別名(alias)，因此，去重覆性工作也是資料庫整合上一個具有挑戰性的議題。最後，無法在短時間內對於 Google 搜尋

引擎頻繁爬取，且 Google Places API 限制 2,500 次/天的免費查詢，因此提高系統效能將是一項挑戰。未來，本系統的查詢次數超過其限制後，也將結合廣告配置模式，使得本方法結合 Google 付費 API 取得資料來源的成本獲得平衡。

致謝

本研究由科技部計畫編號 MOST103-2221-E-008-094 部分贊助。

參考文獻

[1] L. Aleksandra, P. Nikita, S. Pavel, "Web search without 'stupid' results," *SIGIR*, pp.943-946, 2014.

[2] D. Ahlers and S. Boll, "Location-based Web Search." *The Geospatial Web*, pp.55-66, 2007.

[3] S. Auer, J. Lehmann and S. Hellmann, "Linkedgeodata: Adding a spatial dimension to the web of data," *ISWC*, *LNCS*, vol. 5823, pp. 731-746. Springer, Heidelberg, 2009.

[4] D. M. Blei, A.Y. Ng, M.I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, pp. 993-1022, 2003.

[5] C.-C. Chang, C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, 2(27), 1-27 2011.

[6] C.-H. Chang, H.-M. Chuang, C.-Y. Huang, Y.-S. Su, S.-Y. Li, "Enhancing POI Search on Maps via Online Address Extraction and Associated Information Extraction," *Applied Intelligence*, 2015.

[7] C.-H. Chang, C.-Y. Huang, and Y.-Y. Su, "On Chinese Postal Address and Associated Information Extraction," *JSAI*, 2012.

[8] H.-M. Chuang, C.-H. Chang and T.-Y. Kao, "Effective Web Crawling for Chinese Addresses and Associated Information," *ECWeb*, 2014.

[9] M. Costa, F. M Couto, M. J. Silva, "Learning temporal-dependent ranking models," pp. 757-766, *SIGIR*, 2014.

[10] Y.-Y. Huang, "A Tool for Web NER Model Generation Based on Google Snippets," *Master Thesis of National Central University*, 2015.

[11] T. Joachims, "Optimizing Search Engines using Clickthrough Data," 133-142, *SIGKDD*, 2002.

[12] C. B. Jones and R. S. Purves, "Geographical information retrieval," *International Journal of Geographical Information Science*, pp. 219-228, 2008.

[13] T.-Y. Kao, "Points of Interest Extraction from Unstructured Web," Master Thesis of National Central University, 2015.

[14] Y. Liu, R. Song, Y. Chen, J.Y. Nie, J.R. Wen, "Adaptive query suggestion for difficult queries," pp.15-24, *SIGIR*, 2012.

[15] T. Matuszka and A. Kiss, "Geodint: towards semantic Web-based geographic data integration," *ACIIDS*, *LNAI* 8397, pp.191-200, 2014.

[16] J. M. Noguera, M. J. Barranco, R. J. Segura, L. Martínez, "A mobile 3D-GIS hybrid recommender system for tourism," Vol 215, pp. 37-52, 2012.

[17] A. Singhal, "Modern Information Retrieval: A Brief Overview." *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*. vol. 24, pp.35-43, 2001.

[18] C. Stefano, G. Davide, L. D. Angelica, M. Amdrea and Maurizio, T., "Geo data annotator: a Web framework for collaborative annotation of geographical datasets," *WWW*, Italy, 2015.