Web Retrieval and Mining
Final Project Report
今晚吃什麼？

組員分工				
姓名	卓書宇	王智顥	歐政鷹	韓維
分工	使用者介面	Jieba 分詞、推薦評分	Google API、地區分析、Jieba 自定義詞典	trip advisor 爬蟲
簽名				

1. Introduction

台北大商圈餐飲檢索系統，因為每次在一群人出去吃飯時，總是不知道要吃些什麼，永遠都吃一樣的，所以我們決定做一個餐飲推薦系統，讓同學們可以不用在煩惱要吃什麼。

我們透過搜集網路上商家的各種評論，例如： google map、tripadvisor 上的評論，然後利用評論及地區資料，給出相關店家的排名。

2. Methodology

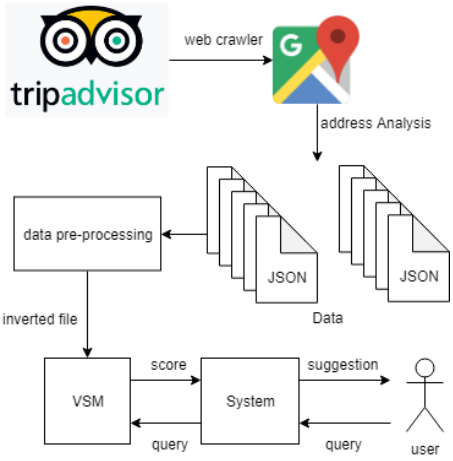

Fig.1 - 系統流程圖

在資料收集方面，資料均為 tripadvisor 網站上台北市中每一間餐廳的店家資訊，包括地址、分類、評分、圖片、價位等，及客人對店家的評論。透過對 tripadvisor 網站進行爬蟲，取得以上各種資訊，並進行處理。

因為 tripadvisor 的資料中只有店家地址，並沒有提供店家的所在地區，因此我們對店家地址進行地區分析，找出每一個店家的所在地區。若地址中沒有找出所在地區，則會利用爬蟲方式從 google map 對地址作搜尋，並從中分析出地址所在地區，讓使用者能夠找出特定地區的餐廳。

取得資料後，對所有評論的內容進行 jieba 中文分詞的處理，利用網絡上收集到不同有關飲食的字詞，建立一個供 jieba 使用的自定義詞典，以取得更好的分詞效果。

利用對評論分詞的結果，分別對每個 Term 以及所出現的餐廳的次數作統計，建立 inverted file，並實作 Vector Space Model (VSM)，透過 Term Frequency - Inverse Document Frequency(TF-IDF) 來計算出每個餐廳的分數。

針對 Term Frequency(TF) ，我們利用 Pivoted Length Normalization 方法對 TF Weighting 作 Documen Length Normalization 的處理，來減少 Document Length 差距所帶來的影響，從回傳結果上我們認為這個方法可以得到比 BM25/ Okapi 有更好的結果。

$$\frac{ln[1+ln[1+c(w,d)]]}{1-b+b*\frac{d}{avgd}} log\frac{M-df+0.5}{df+0.5}$$

Fig.2 - Pivoted Length Norm.

在計算相似度的部分，我們採用了 Dot Product similarity，由於在計算 TF Weighting 時已經有考慮到 Document Length 所帶來的影響，因此不用在計算相似度時重複考慮相同的問題，使用 Dot Product 已經足夠。

3. Experiments

```
餐廳名稱: {
    "address": 餐廳地址,
    "area": 地區,
    "rating": 綜合評分 ,
    "avg_price": 價格範圍,
    "category": [#類型
        "類型1", "類型2" , ...
        ],
    "comments": [ # 文章內容
      {
        "user_name": 使用者名稱,
        "message_title": 評論標題,
        "message": 評論內容,
        "rating": 使用者評分
      },
      ...
      ]
}
```

Fig.3 - 資料格式

3.1 - 資料處理

原先打算透個利用 Google Place API 取得餐廳資訊及顧客的評論，但經實作後發現 API 有流量限制問題，需要長時間來收集資料。同時 API 對搜尋結果的回傳數量有作限制，無法取得餐廳的所有評論。考慮到對 Google Map 的爬蟲不易實作，因此我們最後放棄了從取得 Google Map 中餐廳資訊及評論的做法，改成只利用 tripadvisor 的資料來進行推薦。

Fig.3 為對從 tripadvisor 中收集的資料作處理後的格式。

3.2 - 使用者模式

在這個系統中，總共有兩個使用者模式。第一種模式是針對已經有一些基本想法的使用者，讓他們根據他們基本的想法去搜尋餐廳，例如，使用者原本想吃「火鍋」，但不知道要吃哪一家，此系統會推薦一家火鍋給他。

第二種模式是針對毫無頭緒，沒有想法的使用者給予推薦，透過使用者回答一些我們設定的問題，推薦一些適合的餐廳，並且提供了完全隨機選項。

4. Conclusions

每天在想吃什麼，其實是非常浪費時間的，在這個分秒必爭的時代，造成了此系統的誕生。此系統不但可以作為一個餐廳的搜尋引擎，更可以幫沒有目標，漫無目的走在街上尋找餐廳的人。未來可以加上一些自然語言的技術，例如：「用 ELMo 或 Bert 來做 Word Embedding，讓每個字之間有一種關聯性」，來改進我們的模型。

5. References

I. 向量空間模型-VSM 的改進, https://reurl.cc/zRqqk
II. tripadvisor-貓途鷹, https://www.tripadvisor.com.tw/Restaurants