I • Travel

R01921050 向思蓉
B98901111 蔡維哲
B98901159 馬惟九
B99902114 莫文鈞

1. Introduction

In the recent years, traveling has been popular and prosperous all over the world. Moreover, most of people who are planning a trip will check some travel websites for information. Inspired by the trend, we aim to develop a travel notes recommendation system using some techniques of information retrieval and image processing.

2. Proposed Approach

2.1. Flowchart

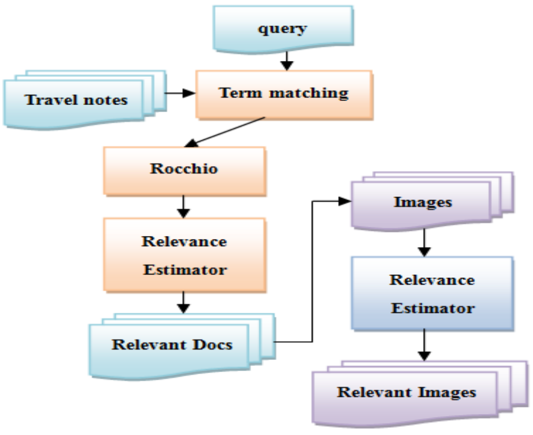

Figure 1: *Flowchart of our recommendation system.*

2.2. Rocchio — Pseudo Relevance Feedback

For some keywords or concepts with ambiguous word representation, such as "溫泉"or "湯" and "巴黎" or "Paris". To handle this problem, we introduce Rocchio pseudo relevance feedback (1) to our work.

$$\overrightarrow{Q_m} = a \times \overrightarrow{Q_0} + b \times \frac{1}{|D_r|} \times \sum_{D_j \in D_r} \overrightarrow{D_J} \qquad (1)$$

,where $\overrightarrow{Q_0}$ is the original query, $\overrightarrow{Q_m}$ is the expanded query and D_r is the relevant documents set.

2.3. Query-focused Page Rank

Different from the basic Page Rank algorithm, this proposed approach includes the similarity between documents and query. Taking advantages from this approach, we can retrieve documents by document importance and query similarity at the same time, which may be very helpful in recommendation system. The following is the modified formula for query-focused Page Rank:

$$p(s|q) = d \sum_{v \in C} \frac{sim(s,v)}{\sum_{z \in C} sim(z,v)} + (1-d) \frac{sim(s,q)}{\sum_{z \in C} sim(z,q)} \quad (2)$$

,where s is a document, q is a query, C is the documents set and d is damping ratio.

2.4. Image Relevance Scores Estimator

Images in travel notes are full of diversity, for example, images taken from close or distant shot, or images with people or not will make the features of images extremely different even though taken in the same scene. To alleviate this problem, we proposed a method that takes inter-documents images similarity into consideration. We estimate the images that have high similarity with other documents as relevant images. As the relevant documents have been retrieved, images in these documents are considered as candidate images. From the candidate images, images with high probability to have similar features to other images in the image candidate set should be estimated with high relevance score:

$$\mathrm{score}_{i,j} = \sum_{k \notin j} \max_l \mathrm{sim}(\mathrm{img}_{i,j}, \mathrm{img}_{l,k}) \qquad (3)$$

,where $\mathrm{img}_{i,j}$is the i-th image in the j-th relevant document and $\mathrm{score}_{i,j}$ is the relevance score of image $\mathrm{img}_{i,j}$.

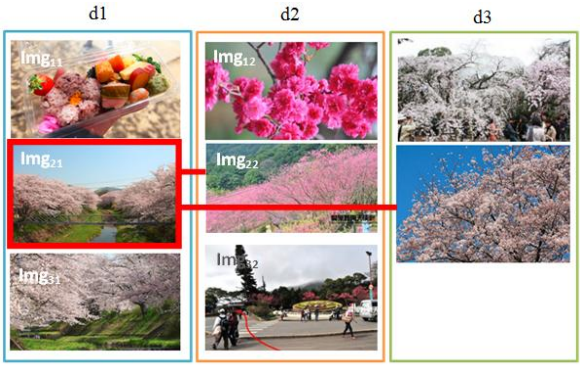

Figure 2:*Inter-documents image similarity.*

3. Features

3.1. Text Features

- Okapi / BM25

$$\mathrm{Score}(D,Q) = \sum_{i=1}^{n} \mathrm{IDF}(q_i) \cdot \frac{f(q_i,d) \cdot (1+k_1)}{f(q_i,d)+k_1 \cdot (1-b+b \cdot \frac{|D|}{avgdl})} \qquad (4)$$

,where q_i is the i-th term, d is the document, and k_1 and b are 1.5 and 0.75 respectively.

- Latent Dirichlet Allocation

Latent Dirichlet Allocation(LDA) is a generative model that allows sets of observations to be explained by unobserved groups that explain why some parts of data are similar. Using the LDA topic model on travel notes data, topics are well clustered by their region or popular part of a city, which may be distinguishable and useful in our recommendation system.

Table 1. *top 5 terms from 6 out of 100 topics generated by LDA*

1	2	3	4	5	6
巴黎	北京	澎湖	教堂	箱根	越南
法國	長城	點點	廣場	富士山	吳哥
鐵塔	中國	牧場	義大利	河口湖	柬埔寨
羅浮宮	胡同	馬公	羅馬	纜車	美金
美術館	皇帝	秘境	威尼斯	登山	寮國

3.2. Image Features

- SIFT

SIFT features is extracted from images for describing an interesting point or object. It is useful in calculation of images similarity and objects matching.

- Sparse representation

Sparse representations are representations that account for most or all information of a signal with a linear combination of a small number of elementary signals. Due to the difference size of images and different number of descriptors, it is hard to generate a fixed number of features for each image. With the techniques of sparse representation and max pooling, we can extract fixed number of features (in this case, 512) from each image.

- Gabor Texture

Using frequency and orientation representations of Gabor filter, texture features can be well extracted. We extract 48 dimensions of feature from each image.

- Color Histogram

Color histogram represents the number of pixels that have colors in a list of color range. In this program, we extracted 225 dimensions of color histogram from each image.

- Columbia 374

This model is trained on color histograms and Gabor textures by SVM, and is used to detect semantic concepts in images such as “Tree”, “People”, “Urban” and “Weather”. With the help of Columbia 374 (provided by Columbia University), we generated 748 features representing 374 semantic concepts.

4. Experiments

4.1. Datasets

There are 21437 text documents and 234663 images belonging to these text documents. First, we parsed PTT (with boards: Ind-travel, Japan_Travel, isLandTravel, EuropeTravel) and backpackers (背包客棧). Then we extracted all the out-links to blogs and saved the images in backpackers and blogs. Finally, we processed our data by CWS provided by Academia Sinica and removing stop words from the list downloaded on the internet. Fig.3 is our flowchart of data generation.

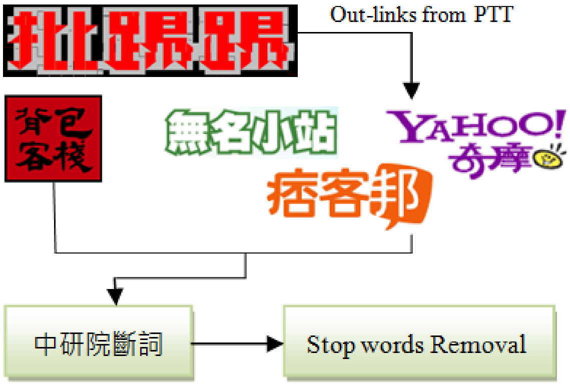

Figure 3: *Data collection and processing.*

Table 2. *text result using text queries*

Text feature	Okapi/BM25 (with no PRF)	Okapi/BM25	Okapi/BM25 +LDA
Acc 30	0.7733	0.8138	0.8074
Acc 20	0.8050	0.8416	0.8500
Acc 10	08900	0.9250	0.9250

Table 3. *image result using text queries*

Top K	3	5	10	15	20
Acc	0.7778	0.7250	0.6917	0.6833	0.6618

Table 4. *text result using image queries*

Top K	2	4	6	8	12
Acc	0.375	0.375	0.3958	0.3593	0.33

Table 5. *image result using image queries*

Top K	10	20	30	40	50	60
Acc	0.41	0.415	0.35	0.3475	0.334	0.3183

4.2. Results

In this project, we generated 100 text queries and 100 image queries. For the lack of travel notes label for each query, we only evaluated our system by accuracy. Table. 2, 3, 4 and 5 shows the results of some experiments. The experiment result is calculated under condition returning different number K documents or images.

5. Demo System

In this project, we built up a demo system for travel notes recommendation system. The web page we presented is coded by Javascript and HTML, and the server processing data and result is coded by Python. In this system, user can type in some queries in the text area or select an image from the webpage to get relevant documents and images.

6. Discussion

In this work, most of error occurred when noise (for example: advertisements, recommendation links written by the blog owners) is included in the travel notes we parsed. Besides that, some of error occurred when the query terms having ambiguous or multiple meanings; for instance, “米開朗基羅” means an famous artist or a café shop in Japan, or “購物” appears in “購物中心” and “東森購物” has different meaning in each case. In this program, we used LDA to capture terms with similar meanings; however, we didn’t do any effort to process the same words with different meanings. For the future work, we should remove the irrelevant part of the travel notes and consider the ambiguous meaning of terms.

7. Work Assignment

向思蓉： parse backpackers and blogs, algorithm, image features extraction, CWS, LDA, report.
馬惟九： Okapi/BM25, data processing, PLSA
蔡維哲： parse PTT, Demo system, data processing, PLSA

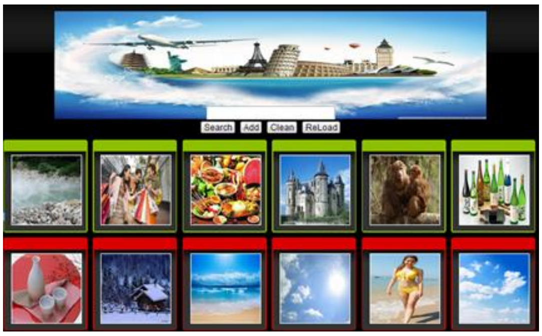

Figure 4: *main page of our demo system.*