Progress Report on

An Intelligent Landmark Extraction and Route Recommendation System using Geo-tagged Images

Submitted by

Ashrith H C (11IT11)

Deeksha S D (11IT16)

Rohan Ashok Bansode (11IT71)

Under the Guidance of

Mrs. Sowmya Kamath S.

Department of Information Technology

National Institute of Technology Karnataka, Surathkal

2014-2015

Department of Information Technology

Major Project - II Progress Report

Course Code: IT499

Course Title: Major Project - II

Title of the project: *An Intelligent Landmark Extraction and Route Recommendation System using Geo-tagged Images*

Project Group:

Sl. No.	Name of the Student	Register No.	Signature with date
1	Ashrith H C	11IT11	
2	Deeksha S D	11IT16	
3	Rohan Ashok Bansode	11IT71	

Name of the Project Guide: Mrs. Sowmya Kamath S

Signature of the Project Guide:

Place: Surathkal

Date: 11th February, 2015

Abstract

Geo-tagged photos enable people to share their personal experiences at specific locations and times on social media like Flickr explicitly which can indicate the route taken by tourists. They can be employed to reveal the tourists preference on landmarks and routing of tourism. Most of existing works on routing searches are based on the trajectories of GPS-enabled devices users. We can assume that the collection of geo-tagged photos is a sequence of visited locations, photo-sharing sites are important sources for gathering the location histories of tourists. By following their location sequences, we can find representative and diverse travel routes that link key landmarks. We propose to build a recommendation system that provides users with the most popular landmarks as well as the best travel routings between the landmarks.

By using Flickr geo-tagged photos, the top ranking travel destinations in a city can be identified and then the best travel routes between the popular travel destinations are recommended. We apply a spatial clustering method to identify the main travel landmarks and subsequently rank these landmarks. Using machine learning techniques, we calculate the tourism popularity of the road in terms of relevant parameters, e.g., the number of users and the number of Point-of-Interests. These popularity assessments are integrated into the routing recommendation system. The routing recommendation system takes into consideration both the popularity assessment and the length of the road. The best route recommended to the user minimizes the distance while including maximal tourism popularity. Besides, the system offers user-generated semantic information for the recommended routes.

Keywords: *Recommender System, Travel route planner, Landmark Ranking, Social Media Analysis*

Contents

List of Figures

List of Tables

1 INTRODUCTION

Due to the proliferation of small digital cameras and mobile phone cameras, there has been great interest in online photo sharing services such as Flickr and Google Picasa. Web 2.0 technologies enable users of social media to make contributions or to communicate with each other. Among the various types of information contributed and shared by users on social media, the geographic one is called Volunteered Geographic Information (VGI).

The most common providers of VGI are Flickr, OpenStreetMap, Twitter, Facebook, YouTube, Wikimapia, Foursquare, etc. VGI has been used in research on tourism, disaster and crisis management, transportation, etc. These services allow users to upload photographs and attach informative tags to them, and they have succeeded in collecting large-scale sets of tagged photographs from huge numbers of users. Geotags indicate where the photographs were taken, and are automatically captured by the photo devices or location-aware devices, or alternatively are specified by the user. Geotags are powerful metadata for introducing spatial information into Web applications. Recent research efforts have shown the potential of geotags by developing various geotag associated applications such as geo-referenced image search [11], [12], automatic image geo-location [10], [3], and geo-referenced content browsing [8], [16].

As a typical photo-sharing provider, Flickr provides a platform where users can share their photos with metadata (e.g., size, time when photo was taken, location, camera type, etc.) and add textual information on the photos. Such textual information includes title, tag, etc. Geo-referenced photos in Flickr are associated with locational, temporal and textual information, which reflect the behaviors and activities of users, particularly tourists. For instance, spatial distribution of Flickr images can reveal the underlying process of tourists footprints.

With a combination of spatial and the temporal dimensions, Flickr images can be used to uncover the trajectories and movements of tourists. From this point of view, Flickr can be employed to provide users with travel recommendations based on the identification of tourists spatial and temporal patterns from images. In addition to the geo-referenced photos, geo-referenced posts in Twitter, geo-located check-ins in Foursquare, user-generated GPS trajectories, etc., have potentials of being leveraged for travel recommendations.

Location-based social networks (LBSN), classify travel recommendations into two types: generic and personalized. Basically, a generic travel recommendation system should provide users the top ranked landmarks, travel sequences, and travel experts in a specified region (e.g., a city). In the contrast, a personalized system is able to provide an individual user with a distinctive recommendation based on the travel preferences and histories of the users. Apart from the tourism attraction or landmark, routing is another important aspect of travel recommendation.

GPS history data is often used to track the footprint of the user, which becomes an indicator of the users trajectory and movement. Representative or typical routings of tourists can be mined from huge amounts of tourists trajectories and thus are recommended to the user. Some researchers take advantage of the trajectory similarity calculation to offer the individual user the most similar routing, which is acquired from the tracking histories of the other users, as the personalized routing. To enhance the real-time usability of the travel recommendation application, time constraint is also taken into account in some research work.

In general we aim to provide a good travel recommendation system that should provide the user with the most interesting (popular) landmarks including routing paths as the most people recommended. Additionally, for tourists, an optimal routing should take into account not only the distance but also the tourism popularity. The research results shows that tourists normally do not take the shortest path between landmarks. Tourists usually like popular streets where they can glimpse more touristic attractions (e.g., parks, squares, statues, memorials, etc.) or satisfy some personalized needs (e.g., eating, shopping, mailing post card, etc.) before reaching their destinations. Thus, to measure the popularity of a road, not only the images taken of the road itself, but also the images taken of the Points-of-Interest (POIs) on or near the road should be taken into consideration. Moreover, in addition to the popularity of the road itself, POIs on or near the road should be taken into account to recommend the best routing.

However, the previous research works generate recommended routings from the historic trajectories (or trajectory segments) of users directly. Unlike these previous research works, we have an alternative perspective on the routing generation, according to which a recommended routing is composed by a set of separate roads (or road segments) which are connected in road network.

Therefore, the advantage of our approach is that we are able to consider the tourism popularity of road as a new feature of roads popularity in a routing planning problem, and based on the values of features including road popularity as well as other attributes (e.g., POIs), we are able to find out the best road set constituting the best overall routing.

1.1 ***Motivation***

Before traveling to an unfamiliar location, most people have questions about how to plan their trips. For example:

- *"I will arrive at Agra on Jun. 3rd and plan to have a tour there. But I am not familiar with that city. Is there any travel route suggestion to visit the most famous places of interest in one day?"*
- *"I want to have a two-day trip in Mysore to visit and taste Mysore's best Dosa. I desperately need help in trip planning."*

Although users can search for related travel guide or ask questions in web-based communities, the process is generally not efficient and the results may not be customized. The most common way for current users to find answers for trip planning is probably to read travelogues one by one. However, as each travelogue only records individual footprints during a trip, it is very time-consuming for users to manually summarize tens of travelogues and find a proper travel route for his preference. Moreover, since the information provided by travelogues is usually unstructured and varies from person to person, from language to language, it is extremely hard for users to follow. In this case, an automatic and interactive travel route planning service is highly desired to plan a customized trip according to users preferences.

In practice, automatic trip planning is a very complex task, which depends on many factors, such as travel duration, travel cost, visiting time, tourists age and physical condition, and individual interests, out of which some are difficult to model and predict. As the first trial on this task, we target at dealing with the following preferences of users in this work, i.e., travel location and destination preference.

We propose a road-based travel recommendation paradigm combining the landmarks and the routing. The travel landmark can be identified by the geo-referenced images with a spatial clustering method. The best routings between the travel landmarks are recommended to users in terms of calculating the total value of the defined recommendation index. This defined recommendation index is constituted by the popularity and distance of the road as well as number of the POIs. The rest of the paper is organized as follows. Section 2 reviews related work on travel recommendation using geo-referenced photos. Section 3 introduces the proposed approach to recommending travel. Section 4 gives a detailed description of the work that has been carried out and Section 5 provides the results and analysis of the work done. Section 6 provides a conclusion of the project and gives future work of the project.

2 LITERATURE SURVEY

There are online travel guides available and some of the famous and most used ones include WikiTravel and the Yahoo Travel Guide. WikiTravel provides users with information such as climate of the destination, popular landmarks in that area, places to shop in the region, good restaurants and different kinds of food that are unique to that place and also information about how to get there. One major disadvantage is that if WikiTravellers do not add information about any particular destination, other users will not find information about that destination. Yahoo Travel Guide is different from WikiTravel in the fact that it provides users with an area based recommendation service. In each country, numerous important and popular cities are listed.

For example, if France is considered, the famous cities listed for that country are Paris, Nice, Bordeaux, St. Tropez, etc. For each city, a ranked list of landmarks are shown to users, associated with comments by other users. There are certain limitations with this travel assistant system as well and one of the major ones is that it is dependent on the critic opinion of travel editors and information about a limited number of cities is available.

The rapid development of camera technology has led to photos being associated with geo-tag information and nowadays popular media sharing websites like Panoramio, Flickr and YouTube are able to provide both tags and geo-tags. Yeran Sun et al [17] proposed a system where geo tag information was used to develop tag suggestion tools by finding tag correlation. These tags and geo-tags have also been used for other purposes like mining events. [13] [15]. In today's research related to tourism, the collection of geo-tagged images and GPS history data is used to identify hotspots and landmarks, mine trajectory and movement and also plan itinerary and recommend trips.

Very little work has been done in the area of automating the process of tour or trip planning. Hiroshi Kori et al [14] proposed to automatically generate multimedia tour guide from local blogs, but users preferences were not taken into account to automatically plan a trip. Similarly Evangelos Kalogerakis et al [10] considered users history to suggest routes.

Some work has also been done on landmark mining using user generated texts or photos. Rongrong Ji et al [9] mined city landmarks from blogs by using graphic models, while Yan-Tao Zheng et al [18] and Alexandar Jaffe et al [8] focused on visualizing, recognizing, describing and summarizing a scene or a landmark by making use of geo tagged photos. Lyndon S Kennedy et al [12] used the rich media available on the web to generate representative sets of images for landmarks. Yue Gao et al [6] proposed to use only the tags associated with images in order to extract potential landmarks and also rank them according to their popularity among tourists.

In [7], Tomoharu Iwata et al used clustering techniques to discover sub-trajectories and Yu Zheng et al [19] proposed a HITS (Hypertext Induced Topic Search)-based model to mine interesting locations and travel patterns using GPS trajectory and considering the relationship between the user and the location. Route suggestion has been addressed in [17], where Yeran Sun et al first use DB-SCAN clustering technique to determine landmarks from a collection of geotagged images and then provide a road based travel route between the extracted landmarks.

DB-SCAN clustering takes a lot of time to compute the clusters depending upon the density of the data. Since any algorithm can be made more efficient by using appropriate data structures, usage of R-tree was proposed [4].

Warehouses undergo changes everyday, every minute of the day. There are many insertions and deletions that need to happen and they are usually done periodically in a batch mode. After updates to the warehouse, data mining tasks have to be run again to update the previously obtained results. Because of huge volumes of data, it is advisable to perform these data mining operations incrementally. An incremental clustering approach was proposed by Martin Ester et al [5] which was typically intended for data warehouses that contain huge volumes of data.

2.1 *Outcome of Literature Survey*

All the existing work concentrate on area driven approaches for extracting popular landmarks from a collection of geo tagged photos. But we propose to concentrate on getting better and accurate results by using tag analysis before clustering to eliminate noise to a certain extent. Since an already filtered data-set is obtained, not a lot of processing goes into data mining techniques for extracting the landmarks. Since obtaining GPS data is a difficult task, the geo tagged photos can be used to suggest routes between landmarks because tourists generally make several stops between consecutive destinations, take a lot of pictures and share them on social networking sites. This pattern can be used to learn from tourist paths and suggest the best travel path using graph based representations and processing. Like Yeran Sun et al [17], we aim to provide the users with a road based travel route by considering the popularity of the road among tourists.

Since our data undergoes many changes everyday like the data warehouses, performing spatial clustering from scratch is not a good idea as spatial clustering algorithms take a lot of time to process the data. Since a web application has to be developed, the response time has to be low for the users to have a pleasant experience. Hence, we suggest to use incremental clustering approach as proposed by Martin Ester et al [5]. And to improve the time taken by clustering algorithms, multiple indexing schemes are compared to find the best one.

2.2 *Problem Statement*

To develop an interactive, personalized travel application for user location based popular landmark identification and route suggestion using machine learning and graph theoretic techniques.

2.3 *Objectives*

The defined objectives for the project are as follows:

1. To collect geo-tagged images from Flickr and analyze the tags for eliminating redundant and generic tags.
2. To develop techniques for eliminating noisy, unrelated images.
3. To incorporate clustering techniques to group images into clusters and calculate cluster attributes for eliminating noisy clusters based on cluster attributes.
4. Improve clustering time using appropriate indexing methods
5. To rank landmarks according to their popularity
6. Calculation of road popularity.
7. Set up hadoop for parallel processing of road network.

8. Calculate recommendation index for each road and generate the best route between two landmarks.
9. Development of the Web framework for interactive user involvement

3 METHODOLOGY

3.1 *System Architecture Diagram*

Figure 3.1 shows the System Architecture. It shows the flow of processes between the Landmark Recognition System and Route Recommendation System to give the desired output.

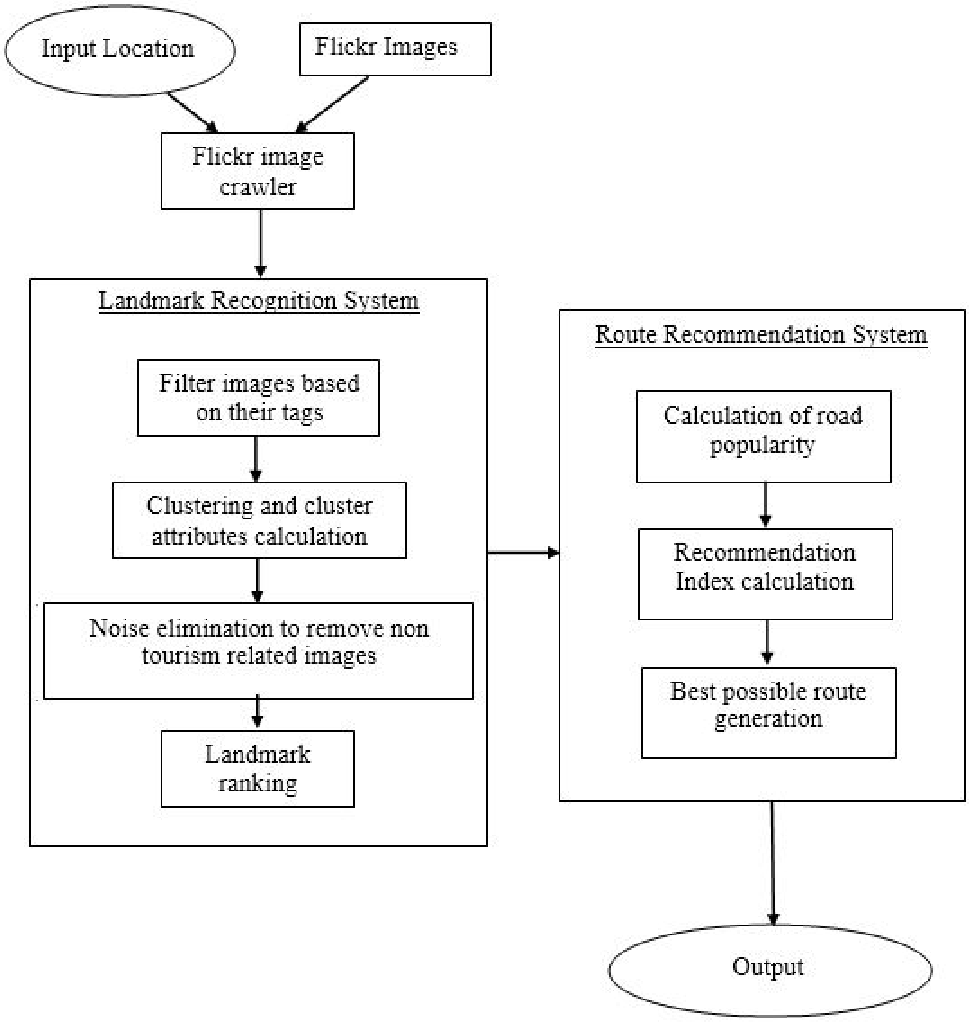

Figure 3.1: *System Architecture Diagram*

3.2 *Collecting geo-tagged images from Flickr*

Flickr exposes certain APIs to perform a limited number of operations like getting contacts, posting photos, get list of favorites, get photos from galleries etc. One of the services provided by Flickr that can be used to get some of the publicly available geo-tagged photos is `flickr.photos.search`. Along with the photos, the tags associated with the photos and some metadata is also obtained. The metadata includes information about the owner of the photo, the date on which the photo was taken, the location in which the photo was taken(geotag) etc. This constitutes our data set for the project.

3.3 *Dataset filtering for Image-relevant tags*

Tagging is done to describe images. These tags give us information about the photos like what kind of landmark it is, the name of the street on which the landmark is present on etc. Since the tags could contain noise and may be very generic in nature like "mySummer", "myTrip", "vacation" etc., we propose to use an NLP technique called TF-IDF which stands for term frequency - inverse document frequency. It is a statistical method that is used to find out how important a word is in a document or a collection of words.Once all the unimportant tags for each photo are removed, the remaining tags can be used to extract information about the landmarks extracted after the clustering process.

3.4 *Clustering and Calculation of cluster attributes*

Clustering algorithms are used to group the images into clusters based on their geo tags. These clusters are further filtered using cluster attributes which are number of users, content score for each user and overall content score for the cluster. These cluster attributes are calculated using the method proposed by Yue Gao et al [6].

- Number of users (N_{user}) - For each cluster, the number of users (N_{user}) is the number of users whose images are in that cluster. Greater the number of users, greater is the popularity of the region.

- Content score for each user (IC_{user}) - This describes how much content each user provides to the cluster. Let IC_{user}(k) denote the content score for k^{th} user and N_{photo}(k) denote the number of photos provided by the k^{th} user. IC_{user}(k) is calculated using the formula (1).

$$IC_{user}(k) = log(N_{photo}(k) + 1) \tag{1}$$

- Overall content score for the cluster (CS) - This describes the overall content provided by all users in the cluster. It is the sum of all IC_{user} values from all the users. It is calculated as given by the formula (2).

$$CS = \sum_{i=1}^{N-users} IC_{user}(k) \tag{2}$$

3.5 *Noise elimination*

3.5.1 *Entropy filtering method*

The obtained set of images may contain noise like photos that have been taken and uploaded by residents of that location and not tourists. Such photos have to be removed from the data set for accurate results. This is done using an entropy filtering method which is based on the fact that tourists stay in the city for a short period of time and upload photos that were taken within that span of time where as residents upload images which have been taken through out the year. Using this method, residents can be distinguished from the tourists among the users of the image set.

3.5.2 *Noise eliminated using cluster attributes*

Using the cluster attributes that were calculated, a cluster can be retained as a landmark or not. A cluster may be considered as a landmark only if it has a high importance factor. T_{CS} is the threshold value used to determine if the cluster is a landmark or not. If the cluster $cluster_i$ satisfies the condition represented by equation (3, it is considered to be a landmark.

$$CS(Cluster_i) > T_{CS} \tag{3}$$

3.6 *Ranking the landmarks according to their popularity*

In order to rank the landmarks, their popularity has to be quantified. This can be done by considering the fact that tag frequency of social media follows power law distribution. Consider a landmark L and that m different users contribute images to this landmark. Let $U_L = \{\ u_1, u_2, ...u_m\ \}$ be the user set of images assigned to landmark L. Then the popularity of the landmark L, Pop(L) is calculated by using the equation (4) which is proposed in [17].

$$Pop(L) = \sum_{\forall u}(1 + log n_L(u)) = m + \sum_{\forall u}(log n_L(u)) \quad (4)$$

Pop(L) denoted the popularity of the landmark L and $n_L(u)$ denotes the number of images taken by user u in the region of landmark L.

3.7 *Calculation of road popularity*

If all the landmarks are considered as POIs (Points of Interest), the most popular set of roads has to be returned to the user that connects two landmarks. A road is considered to be popular if there are many POIs on it or a user has to use that road to get to the POI. But in order to find which POI is on which road, we have to assign the images to appropriate roads. This is difficult because not all images contain tags that provide information about the street that it is located on. There are two kinds of images. Type 1 is the kind where the images with tags providing information about the street or road names. Type 2 is the kind where the images dont contain road names explicitly as tags but still contribute to the road popularity of the road that they lie on. We use the method used by Yeran Sun et al [17] to calculate the popularity of roads by assigning images to appropriate roads.

3.7.1 *Image assignment*

Type 1 images and images with geo tags very close to the road can be assigned to roads without any difficulty. For type 2 images, if only the geo tag is considered and is assigned to the road it is closest to by calculating the distance, this might be wrong. There are situations where the image might be taken at a distance away from the entrance of the building that represents POI and this location is close to one road, but the entrance might be closer to another road. So the road it should be assigned to should be the road closer to the entrance. But the location of the entrance is not known to us. In order to solve this problem, we use two candidate roads that every image can be assigned to. One is the 1st closest road and second is the 2nd closest road.

Apart from the fact that image should be assigned to the closest road, it should also be kept in mind that if that road has no POIs or very few POIs compared to the 2nd closest road, the 2nd road is preferred because of the large number of POIs. So this image assignment problem becomes a binary classification problem with the classification category set as {0,1}, where 1 means the image is assigned to 1st nearest road and 0 means that the image is assigned to the 2nd nearest road.

3.7.2 *Classification features*

The assignment is based on the distances of the image to the 1st and 2nd nearest roads. Hence these two distances are considered as the classification features. Apart from these, the number of POIs on the roads are also considered as classification features as they also contribute to the road popularity as described before.

Prior to addressing the assignment problem, POIs have to be assigned to appropriate roads. If a POI has address information, street name is extracted from it and is assigned to that road, else it is assigned to the road closest to it. But there are different kinds of POIs and each kind contributes a different amount of popularity to the road than the others. Hence we consider three categories of POIs and appropriate weights are assigned to each category. Category 1 consists of tourism attractions like museums, palaces etc. Category 2 consists of eating and drinking places and Category 3 consists of other places like lodges, banks etc. The classification feature set is represented as follows:

$$(Dis_1, Dis_2, Num_1^1, Num_2^1, Num_1^2, Num_2^2, Num_1^3, Num_2^3)$$

where Dis_1 is the distance of the image to the 1st nearest road, Dis_2 is the distance of the image to the 2nd nearest road, Num_1^1 is the number of POIs of Category 1 on the 1st nearest road, Num_2^1 is the number of POIs of Category 1 on the 2nd nearest road, Num_1^2 is the number of POIs of Category 2 on the 1st nearest road, Num_2^2 is the number of POIs of Category 2 on the 2nd nearest road, Num_1^3 is the number of POIs of Category 3 on the 1st nearest road and Num_2^3 is the number of POIs of Category 3 on the 2nd nearest road.

3.7.3 *Road popularity*

Popularity of the road is calculated in the same way as landmark popularity. Let the images assigned to road r be contributed by m different users. $U_r = \{u_1, u_2, ...u_m\}$ is the user set of images assigned to road r. The number of images taken on road r by user u is

n_r(u). Then the popularity of the road is calculated as depicted in the equation (5).

$$Pop(r) = m + \sum_{\forall u} (log n_r(u)) \tag{5}$$

3.8 *Recommendation Index*

The recommendation index of each road is calculated, as given by [17], by considering the number of images assigned to that road, the number of POIs and the length of the road. The first two attributes add to the index whereas the length of the road decreases the recommendation index of the road. The overall popularity of the road is quantified by defining a recommendation index $Rec\_Index(r)$ which mainly consists of two parts - tourism popularity and the POI usability. The POI usability is given by Poi(r) and is calculated as shown in the equation (6).

$$Poi(r) = w_1 Num^1_{POI}(r) + w_2 Num^2_{POI}(r) + w_3 Num^3_{POI}(r) \tag{6}$$

$Num^1_{POI}(r)$, $Num^2_{POI}(r)$ and $Num^3_{POI}(r)$ are the number of POIs of category 1, 2 and 3 respectively. w_1, w_2 and w_3 are the weights of $Num^1_{POI}(r)$, $Num^2_{POI}(r)$ and $Num^3_{POI}(r)$ respectively and $w_1 + w_2 + w_3 = 1$.

The recommendation index $Rec\_Index(r)$ is calculated as given by the equation (7).

$$Rec\_Index(r) = \alpha_1 Pop(r) + \alpha_2 Poi(r) - \beta Leng(r) \tag{7}$$

Pop(r) is the popularity of the road and Leng(r) is the length of the road which is used as penalty for generating unpractical routing that is too long for the user. α_1, α_2 and β are weights of Pop(r), Poi(r) and Leng(r) respectively.

3.9 *Generating best routing between two landmarks*

The best possible route between the two landmarks can be generating using graphs. The popular Dijkstras algorithm is used and recommendation index of the road is used instead of the length of the road.

4 WORK DONE

4.1 *Data Collection*

Data collection was done using Flickr's ***flickr.photos.search*** service. All of Flickr's services are exposed through APIs and hence to use the ***flickr.photos.search*** service, its corresponding API was used. There are three request formats supported by Flickr and they are ReST, XML-RPC and SOAP. In our project, ReST was used as it is the simplest request format to use that just involves sending an HTTP GET or POST request.

The ReST based request's endpoint URI is `https://api.flickr.com/services/rest/`. The ***flickr.photos.search*** service can be invoked by using `https://api.flickr.com/services/rest/?method-flickr.photos.search` followed by all the arguments that are necessary for your request as a number of query parameters. The response is in ReST format which is simple XML. An example request and response is shown in figure 4.1.

Request

```
https://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=c55d1c61df96fe
dbc444f9fa487e2170&per_page=3&bbox=-122.490128,37.722134,-122.416657,37.789994
&extras=tags,geo,date_taken
```

Response


```
<?xml version="1.0" encoding="utf-8" ?>

<rsp stat="ok">

  <photos page="1" pages="70776" perpage="10" total="707752">

    <photo id="15736130272" owner="62256195@N08" secret="718644886a"
server="8636" farm="9" title="Howard Wiley in the house this week... talking about
#ColemanHawkins   , "Body & Soul" and keeping it Date-Night Friendly. #Hotplate next
Thursday! #sfjazz" ispublic="1" isfriend="0" isfamily="0" />

    <photo id="15115078603" owner="58189808@N03" secret="f5ca109431"
server="3937" farm="4" title="Taking BART to the ferry to Angel Island" ispublic="1"
isfriend="0" isfamily="0" />

    <photo id="15548927078" owner="117499805@N05" secret="1c283e5972"
server="3942" farm="4" title="Running around #SanFrancisco with @stephaniehaller &
@jelywi" ispublic="1" isfriend="0" isfamily="0" />

  </photos>

</rsp>
```


Figure 4.1: *Example usage of Flickr API*

In the request shown in the example, the query parameters have been separated using

&. Following describes the meaning of each of the arguments used in the request.

1. api_key: An application key is required in order to use the Flickr API. This key is used by Flickr to track the API usage.

2. per_page: This parameters represents how many results have to be included in every page.

3. bbox: A comma-delimited list of 4 values defining the Bounding Box of the area that will be searched. The 4 values represent the bottom-left corner of the box and the top-right corner, minimum_longitude, minimum_latitude, maximum_longitude, maximum_latitude. Only those photos will be returned that fall within the bounding box.

4. extras: This parameter is a comma-delimited list of extra information to fetch for each returned photo. Hence for the above example, the extra information retrieved are tags, geotags and date taken for each photo.

Since the collection of data was done using python, a Flickr API kit was used which provides a wrapper for invoking Flickr's services. The kit that was used is Beejś Python Flickr API.

4.2 *Filtering tags to remove unimportant tags*

The method used to remove unimportant tags in images is a popular NLP technique called as TF-IDF(term frequency-inverse document frequency). This technique is commonly used to find unimportant words in a collection of documents. Here, we consider photos as documents and tags for each photo as the words in the document.

The term frequency is calculated as either 1 or 0. It is one if a tag is present in a photo and 0 if it is not. Inverse document frequency is calculated using the equation 8. Tf-idf is computed as given in equation 9.

$$IDF(tag_i) = log_2(1 + \frac{D}{doc\_freq\{i\}}) \tag{8}$$

where tag_i represents the tag in consideration, D represents the total number of documents or photos and $doc\_freq\{i\}$ represents the number of photos that tag_i occurs in.

$$TF - IDF(tag_i) = TF(tag_i) * IDF(tag_i) \tag{9}$$

4.3 *Clustering and Noise Elimination using Cluster Attributes*

Spatial clustering algorithm DBSCAN was used to group the images into clusters according to their geo tags. DBSCAN algorithm requires two parameters - the neighbourhood distance and the minimum number of points. By trial and error, a distance of 100 meters was used as the neighbourhood distance and 20 was used used as the minimum number of points.

Since the distance between two images is not known to us, we used the geo locations of the images to compute the great circle distance between them using haversine formula which is shown in equations 10 - 11.

$$haversine(point1, point2) = (sin(\frac{lat2 - lat1}{2})^2) + (cos(lat1) * cos(lat2) * sin(\frac{lon2 - lon1}{2})^2) \tag{10}$$

$$distance(point1, point2) = 2 * radius * arcsin(\sqrt{haversine(point1, point2)}) \tag{11}$$

where *lat1, lon1, lat2, lon2* represent the latitudes and longitudes of point1 and point2 respectively and *radius* represents the radius of the earth which is 6367 km approx.

Cluster attributes were then calculated as described in 3.4 using equations 1 and 2. A threshold value of CS_{thresh} was used to remove noisy clusters.

4.4 *Clustering indexed data*

Since the number of images that have to be clustered is large, DBSCAN takes a longer time to complete the clustering process. This is because of performing range queries for all the images in the data set. A range query is invoked to return all the images that are within a specific range of a particular image. If the images are not indexed according to their geo locations, then a normal range query would require the program to traverse through all the images, compute their distances with respect to the given image and return those that satisfy the maximum radius condition. This process can be improved by indexing images based on their geo locations to make DBSCAN a lot faster than the traditional approach. The usual one dimensional indexes and hashes are not useful for our two dimensional data. In this project, we have explored two methods of indexing.

4.4.1 *R-tree index*

R-tree is a tree which is used for indexing of multi-dimensional data such as geographic co-ordinates. 'R' in R-tree stands for rectangle. The basic working of R-trees is that nearby data objects are grouped together and represented within their minimum bounding rectangle in the next higher level of the tree [2]. So all the leaves of the tree represent single data objects. The leaves are also rectangles but contain a single entity. R-trees are most efficient for retrieval of information.

They support bounding box retrieval, nearest neighbor search and containment queries. Bounding box queries are useful because they don't require the entire tree to be traversed. Based on the bounding box, sub trees are bypassed to move to the next sub-tree if they don't lie within the bounding box.

Using R-trees for indexing the Flickr images made clustering a lot faster. This is because most of the time during clustering is spent on performing range queries and indexing makes querying within a range makes faster.

4.4.2 *MongoDB 2dsphere index*

MongoDB provides various indexing schemes including indexing for multi-dimensional data. One such index is the 2dsphere index which supports queries involving calculations on a sphere. It supports all kinds of geospatial queries like range, bounding box and nearest neighbor queries. [1]

Using this indexing scheme gave interesting results as clustering took more time than what traditional DBSCAN took. This is because mongoDB takes a longer time for execution of 2dsphere index queries when compared to other queries.

4.5 *Ranking the landmarks according to their popularity*

The important clusters that can be considered as landmarks are extracted and the photos assigned to each cluster are analyzed. Information about the number of users contributing to each cluster and the number of photos contributed by different users is obtained by using appropriate database queries. The clusters are then ranked using equation 4 as explained in section 3.7.

4.6 *Plotting the landmark clusters on Google Maps*

Once all noisy clusters are eliminated using cluster attributes, the mean of all photos in each cluster was calculated to find the geo location of the landmarks. This geo location was reverse geocoded to convert the latitude and longitude of the landmark to an address. To do this, mapquest API which requires two input parameters.

- key: This is the API key which is used to track the API usage.
- location: This is a comma separated values for latitude and longitude of the location that you want to reverse geocode.

The service endpoint URL is `http://open.mapquestapi.com/geocoding/v1/reverse?key=YOUR_KEY_HERE&location=40.053116,-76.313603`. The response returned by this API is JSON. A sample response is shown in figure 4.2.


```
{
    "info": {
        "copyright": {
            "imageAltText": "\u00a9 2014 MapQuest, Inc.",
            "imageUrl": "http://api.mqcdn.com/res/mqlogo.gif",
            "text": "\u00a9 2014 MapQuest, Inc."
        },
        "messages": [],
        "statuscode": 0
    },
    "options": {
        "ignoreLatLngInput": false,
        "maxResults": -1,
        "thumbMaps": true
    },
    "results": [
        {
            "locations": [
                {
                    "adminArea1": "US",
                    "adminArea1Type": "Country",
                    "adminArea3": "PA",
                    "adminArea3Type": "State",
                    "adminArea5": "Lancaster",
                    "adminArea5Type": "City",
                    "displayLatLng": {
                        "lat": 40.053234,
                        "lng": -76.313268
                    },
                    "dragPoint": false,
                    "geocodeQuality": "ADDRESS",
                    "geocodeQualityCode": "L1AAA",
                    "latLng": {
                        "lat": 40.053116,
                        "lng": -76.313603
                    },
                    "linkId": 0,
                    "mapUrl": "http://open.mapquestapi.com/staticmap/v4/getmap?key=Fmjtd|luurn96z20,7x=o5-9w8a5u&type=map&size=225,160&pois=purple-1,40.053116,-76.313603,0,
                    0|&center=40.053116,-76.313603&zoom=15&rand=1634589923",
                    "postalCode": "17603",
                    "sideOfStreet": "N",
                    "street": "254 Lincoln Avenue",
                    "type": "s"
                }
            ],
            "providedLocation": {
                "latLng": {
                    "lat": 40.053116,
                    "lng": -76.313603
                }
            }
        }
    ]
```


Figure 4.2: *Example response of mapquest API*

Before plotting the clusters on Google Maps, each cluster is assigned a different color and then Google Maps API for plotting markers is used to plot the clusters. The reverse geocoded address of each landmark is displayed when any cluster is clicked.

5 RESULTS AND ANALYSIS

Flickr was queried for all the public photos for 6 different locations - Sydney, Paris, London, Singapore, New York and San Francisco. A total of 46160 photos were collected. Among these 46160 photos, 1865 photos were eliminated as photos taken by residents. A summary of all the photos from the 6 locations is given below.

- Total number of locations: 6
- Total number of photos: 46160
- Total number of users: 4324
- Total number of tourists: 4301
- Total number of residents: 23
- Total number of photos taken by residents: 1865
- Total number of photos taken by tourists: 44295

For distinguishing tourists from residents, an entropy filtering method as described before was used. A threshold of 1.5 was used for the entropy. If the computed entropy was greater than this threshold, the user is labeled as a resident. Tf-idf was then used to eliminate common tags by using a threshold value of 2. So, all the tags that occur in more than one-fourth of the total photos will be discarded. After removing photos based on tag analysis, DBSCAN algorithm was used to group the photos into clusters and cluster attributes were calculated. The threshold content score of cluster that was used to eliminate noisy clusters is 8. A detailed statistics of the number of photos resulting after each step is shown in table 1.

Table 1: *Statistics of data preprocessing*

Location	No of photos collected	No of photos taken by tourists	No of photos of tourists containing tags	No of photos remaining after tf-idf elimination	No of clusters formed	No of clusters considered	No of photos in all the clusters considered
Sydney	6249	6096	4775	4666	24	9	934
Paris	6600	6543	4248	4191	31	12	433
London	8050	7842	5889	5877	39	17	1084
Singapore	5656	4553	2726	2241	12	3	77
New York	8173	8165	5235	5234	17	5	123
San Francisco	11432	11096	8145	7785	62	22	1003

The two indexing schemes for clustering yielded different results when compared to clustering on non indexed data. These results are shown in table 2.

Table 2: *Comparison of indexing methods*

Location	Number of photos	Number of clusters	DBSCAN on non indexed data	DBSCAN on R-tree indexed data	DBSCAN on 2dsphere indexed data
Sydney	4849	24	67.69 seconds	24.38 seconds	95.76 seconds
Paris	4275	31	53.44 seconds	18.89 seconds	69.91 seconds
London	5889	39	115.61 seconds	37.97 seconds	102.96 seconds
Singapore	3199	12	25.71 seconds	3.56 seconds	28.68 seconds
New York	5243	17	71.81 seconds	14.21 seconds	32.44 seconds
San Francisco	8321	62	195.03 seconds	57.21 seconds	223.61 seconds

From the above table, it can be seen that greater the number of photos to be clustered, greater is the clustering time. For cities that have similar number of photos, if the number of clusters is more, the time taken for DBSCAN is more. So we can conclude that clustering time depends on the size of the data set and the number of clusters that can be formed. R-tree appears to be the most efficient indexing method for our data-set. A comparison of clustering time is shown in figure 5.1.

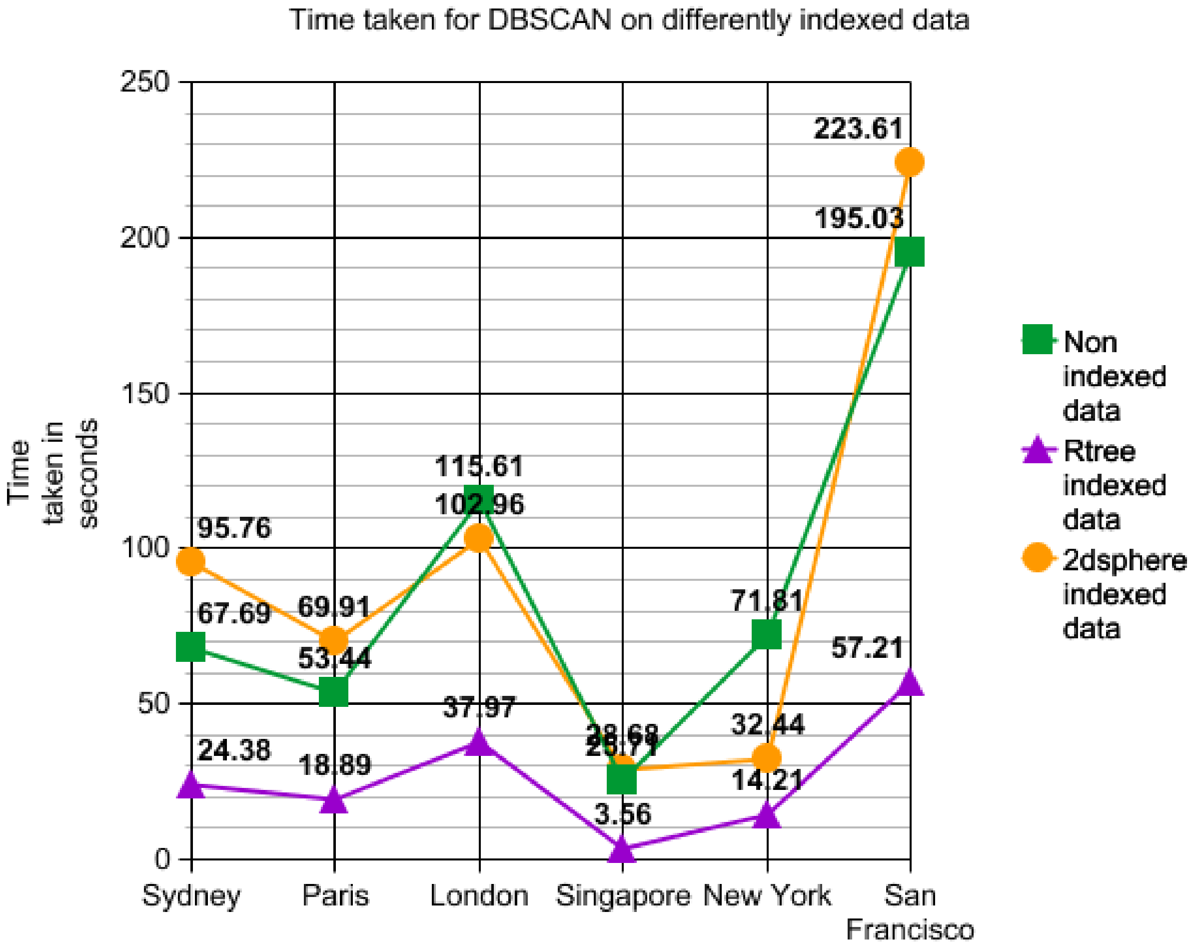

Figure 5.1: *Clustering time analysis for different indexes*

Each of the filtered clusters are plotted on the world map using Google Maps API. All the landmarks of the 6 locations have been plotted on the world map and is shown in figure 5.2. Figure 5.3 and 5.4 show the clusters that have been plotted for San Francisco and London respectively.

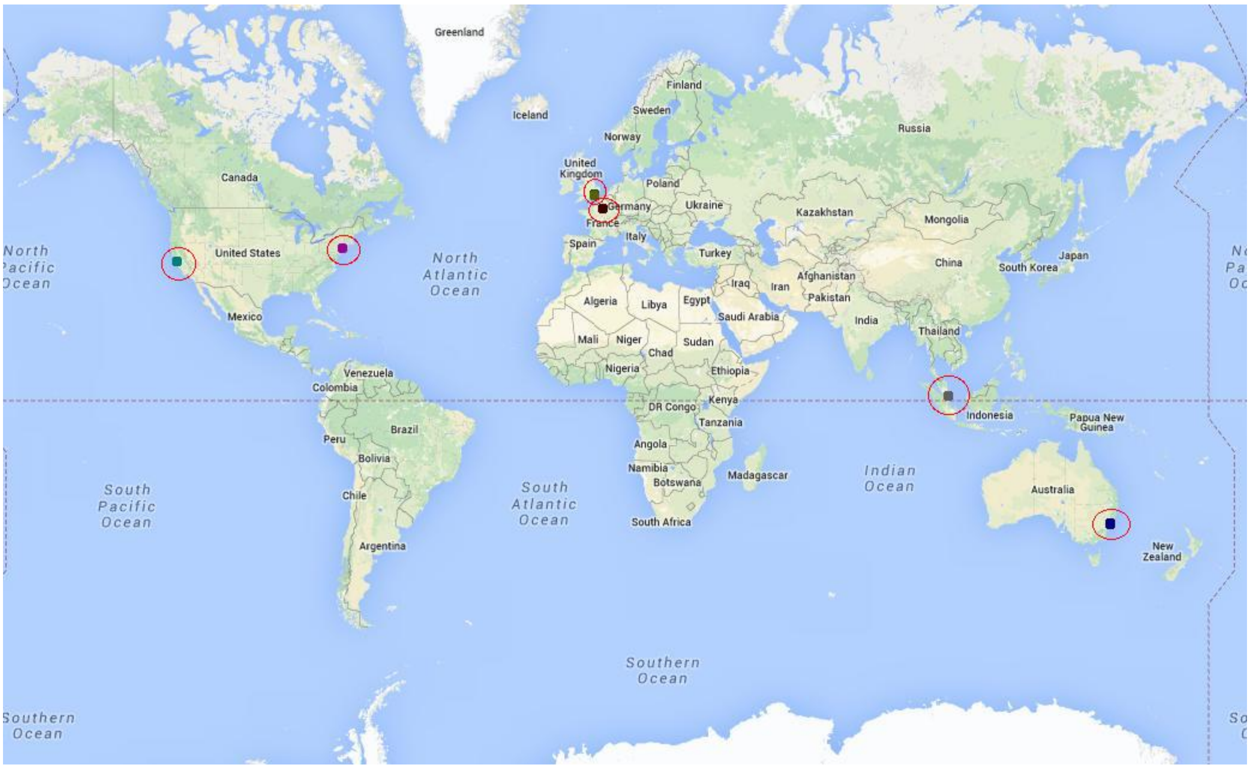

Figure 5.2: *Landmarks of all 6 locations on world map*

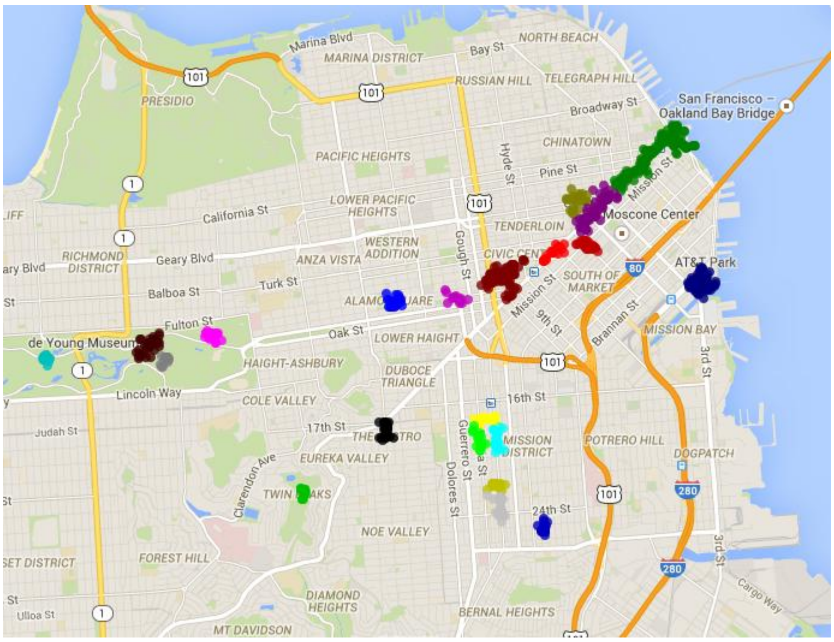

Figure 5.3: *Landmarks in San Francisco*

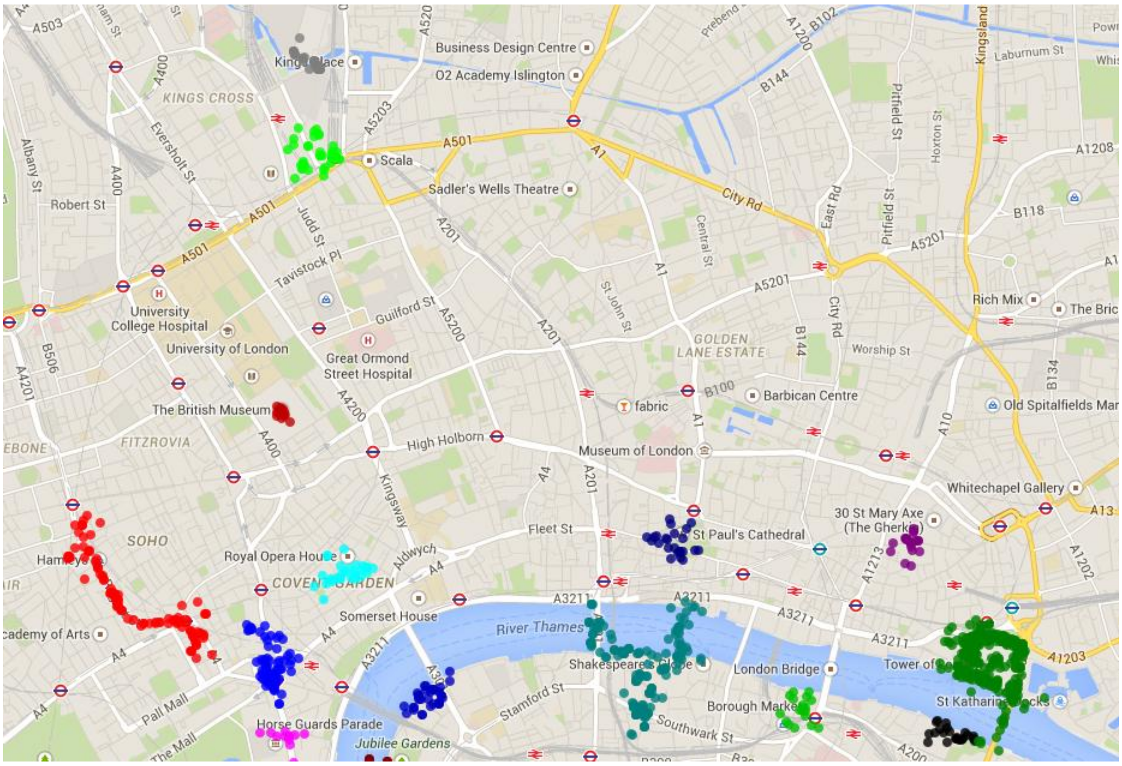

Figure 5.4: *Landmarks in London*

6 CONCLUSION AND FUTURE WORK

In this report, we have proposed to develop an online application that accepts a location as input and provides the user with the top landmarks in that location and further allows the user to choose two of the landmarks as endpoints and provides the best route between them. This will be done with the help of data mining techniques, machine learning, graph theory and Natural Language Processing.

We have collected meta data about photos for 6 locations from Flickr using python toolkit for Flickr's API. These photos were analyzed to distinguish tourists from residents and all the photos uploaded by residents were discarded as noise. The tags of the remaining photos were then analyzed using TF-IDF to eliminate common tags that occur in most of the photos. DBSCAN clustering was then applied to the photos remaining after tag analysis and cluster attributes were calculated for each of the clusters. R-tree was used the index data structure to make retrieval faster and hence reduced the time taken for clustering. The clusters that satisfied minimum requirements were considered as landmarks and plotted on a world map using Google Maps API.

In future, we propose to use incremental clustering approach in order to avoid redundant clustering. Then using digital maps and graph theory, best possible routes between two landmarks is generated and shown to the user. We plan to optimize and parallelize the techniques used and reduce the load on the server since it is an online application and the response time should be as low as possible.

7 PROJECT TIMELINE

Figure 7.1 shows the Proposed Project timeline.

Objectives	Jul 2014 - Aug 2014	Sept 2014 - Nov 2014	Dec 2014	Jan 2015 - Feb 2015	Mar 2015 - Apr 2015
Project Proposal					
Crawl Flickr to collect images for dataset					
Filter the images based on tags					
Clustering and calculation of cluster attributes					
Noise elimination					
Improve clustering using indexed data					
Landmark ranking					
Calculation of road popularity					
Set up hadoop for parallel processing of road network					
Calculation of recommendation index for each road					
Generate best route between 2 landmarks					
Development of the Web framework for interactive user involvement					

Figure 7.1: *Proposed Project Timeline*

References

[1] 2dsphere indexes. `http://docs.mongodb.org/manual/core/2dsphere/`.

[2] R-tree. `http://en.wikipedia.org/wiki/R-tree`.

[3] Liangliang Cao, Jie Yu, Jiebo Luo, and Thomas S Huang. Enhancing semantic and geographic annotation of web images via logistic canonical correlation regression. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 125–134. ACM, 2009.

[4] Min Chen, Xuedong Gao, and HuiFei Li. Parallel dbscan with priority r-tree. In *Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on*, pages 508–511. IEEE, 2010.

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Michael Wimmer, and Xiaowei Xu. Incremental clustering for mining in a data warehousing environment. In *VLDB*, volume 98, pages 323–333, 1998.

[6] Yue Gao, Jinhui Tang, Richang Hong, Qionghai Dai, Tat-Seng Chua, and Ramesh Jain. W2go: a travel guidance system by automatic landmark ranking. In *Proceedings of the international conference on Multimedia*, pages 123–132. ACM, 2010.

[7] Tomoharu Iwata, Shinji Watanabe, Takeshi Yamada, and Naonori Ueda. Topic tracking model for analyzing consumer purchase behavior. In *IJCAI*, volume 9, pages 1427–1432, 2009.

[8] Alexandar Jaffe, Mor Naaman, Tamir Tassa, and Marc Davis. Generating summaries and visualization for large collections of geo-referenced photographs. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 89–98. ACM, 2006.

[9] Rongrong Ji, Xing Xie, Hongxun Yao, and Wei-Ying Ma. Mining city landmarks from blogs by graph modeling. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 105–114. ACM, 2009.

[10] Evangelos Kalogerakis, Olga Vesselova, James Hays, Alexei A Efros, and Aaron Hertzmann. Image sequence geolocation with human travel priors. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 253–260. IEEE, 2009.

[11] Lyndon Kennedy, Mor Naaman, Shane Ahern, Rahul Nair, and Tye Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. In *Proceedings of the 15th international conference on Multimedia*, pages 631–640. ACM, 2007.

[12] Lyndon S Kennedy and Mor Naaman. Generating diverse and representative image search results for landmarks. In *Proceedings of the 17th international conference on World Wide Web*, pages 297–306. ACM, 2008.

[13] Slava Kisilevich, Milos Krstajic, Daniel Keim, Natalia Andrienko, and Gennady Andrienko. Event-based analysis of people's activities and behavior using flickr and panoramio geotagged photo collections. In *Information Visualisation (IV), 2010 14th International Conference*, pages 289–296. IEEE, 2010.

[14] Hiroshi Kori, Shun Hattori, Taro Tezuka, and Katsumi Tanaka. Automatic generation of multimedia tour guide from local blogs. In *Advances in Multimedia Modeling*, pages 690–699. Springer, 2006.

[15] Tye Rattenbury, Nathaniel Good, and Mor Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 103–110. ACM, 2007.

[16] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM transactions on graphics (TOG)*, 25(3):835–846, 2006.

[17] Yeran Sun, Hongchao Fan, Mohamed Bakillah, and Alexander Zipf. Road-based travel recommendation using geo-tagged images. *Computers, Environment and Urban Systems*, 2013.

[18] Yan-Tao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, T-S Chua, and Hartmut Neven. Tour the world:

building a web-scale landmark recognition engine. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1085–1092. IEEE, 2009.

[19] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009.