Personalized Tour Recommender through Geotagged Photo Mining and LSTM Neural Networks

Chieh-Yuan Tsai^{1,*}, Gerardo Paniagua¹, Yu-Jen Chen¹, Chih-Chung Lo² and Liguao Yao³

¹Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan City, 32003, Taiwan, R.O.C.

²Department of Applied Informatics, Fo Guang University, Yilan County 26247, Taiwan, R.O.C.

³Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University, Guiyang 550025, China

Abstract. In this study, a tour recommendation system based on social media photos is proposed. The proposed recommendation system can generate trip tours considering both the user's current location and interests. First, we exploited the geotagged photo dataset from social media websites, which includes photo related information such as user ID numbers, timestamps, hashtags, and GPS coordinates. With this information, the second step is to group photos and identify those places that could be considered relevant for travellers using clustering algorithms. The third step characterizes the resulting clusters by grouping them into different categories using latent dirichlet allocation (LDA) topic modelling approach. The last step is the generation of tours using a long-short term memory neural network (LSTM). The experiments show that the proposed system can be efficient to advise future travellers about the places they would be more likely to visit and arrange trips for them.

1 Introduction

Many social media websites, such as Instagram or Foursquare, serve as platforms to gather the data that their users share. With the help of these valuable data, it is possible to discover hidden travel patterns from the user's behaviour. The patterns can help us have insight about particular characteristics on different users, and make prediction or classification of future events under an acceptable margin of error [1]. Therefore, there has been an increasing interest and research in developing intelligent tour recommendation systems (or recommenders) to help travellers arrange their trips.

Typically, the recommenders can be distinguished into three main approaches: collaborative filtering, content-based filtering, and hybrid recommender systems. Among them, collaborative filtering is one of the most popular ones and has been applied in many tour recommenders with different purposes. [2] proposed a framework that takes the user's visit history to make recommendations after categorizing each tags included on each picture. [3] used user generated GPS traces in combination with a tree based hierarchical graph model in order to extract location histories and then propose a HITS (Hypertext Induced Topic Search)-based model to infer the interest level of a location and a user's travel experience. [4] used RFID sensors to train a recommender for tourists in a theme park to deliver recommendations for different areas, including the sequential orders to visit them. [5] collected several check-in datasets to make time sensitive tours

recommendations based on visit histories and timestamps in order to compute an estimate of the visiting time and transportation time from one place to another. [6] developed a collaborative filtering system based on contextual information of geotagged photos. Their framework considers 5 aspects: the user information, the geotags, the timestamps, the tags and visual information. They make use of these elements to calculate user similarity, attraction and popularity scores among users and locations as well as the most popular visiting times.

[7] proposed a trip planner by comparing the performance of different heuristic algorithms to optimize the time, the total travelled distance, and the overall satisfaction of users on an unfamiliar city. [8] established a topic based context-aware travel recommendation method based on geotagged photos. Their method is built on top of a topic model which mines the interest distribution of users, and is then used to build a user-user similarity model to make travel recommendations. [9] leveraged geotagged photos obtained from Flickr; they inferred landmarks by means of clustering and then generated recommended routing from separate roads instead of historic trajectories of users. [10] used clustering to extract landmarks and applied topic model to characterize these landmarks. The objective was of obtaining areas of interest and their corresponding landmarks. Furthermore, the landmark vector resulting from the topic model was used to calculate a subsequent user vector.

* Corresponding author: cytsai@saturn.yzu.edu.tw

Although the above researches have demonstrated themselves to be useful at recommending new places for people to visit in an unfamiliar city, we found some problems that they still need to solve. Firstly, most of them fail to transform their recommended venues from a list format to a tour. Even after getting a complete list with the best touristic attractions on a city, many users may find it tedious to plot an itinerary by themselves including the many point of interests (POI) to visit. Secondly, those lists are ranked based on the opinion of the majorities of the users. People have different preferences, hence not everyone will be satisfied with just the most popular recommendation. Thirdly, many of these recommenders ignore some important considerations such as the user's interests and current location.

To solve the above problems, a recommendation system based on a social media photo dataset is proposed. The proposed recommendation system can generate trip tours considering both the user's current location and interests. First, we exploited the geotagged photo dataset from social media websites. With this information, we can transform the users' travel log histories into trajectories and later use them to train a recommendation system to advise future travellers about the places they would be more likely to visit and arrange trips for them.

2 Methodology

2.1 Location-based recommendation systems

The framework of the proposed tour recommender is outlined in Figure 1 which includes four important steps: data collection, location clustering, landmark characterization, and tour recommendation.

Fig. 1. The proposed tour recommender framework.

2.2 Data collection

The development of this framework relies on building up a photo dataset to analyse the trajectories followed by the travellers and map the photos taken by the users in order to represent them as Point of Interest (POI). Each POI will constitute an attractive place for tourists such as monuments, museums, parks, and so on. In this study, we will build up the groups of candidates for these POI from social media websites and extract the content from the photo metadata such as geolocation, tags, number of views, timestamp, and encrypted user ID

numbers. All this information was stored to later cleaned it, pre-processed it, and fed it into the model we trained.

2.3 Location clustering

The second step is to group photos and identify those places that could be considered relevant for travellers using clustering algorithms. The clustering algorithm can help us to separate the area of a city into several small neighbouring areas according to every photo's geographic positions, their frequency density, and the distances among each point.

In this study, mean shift clustering algorithm is adopted for landmark extraction [11]. Mean shift clustering algorithm is a non-parametric algorithm can map the probabilistic distribution from a spatial dataset outlining landmarks based on clusters with higher photo density. This can be achieved by seeding this procedure from many initial points iterating this process until a convergence. The mean shift is defined as:

$$m_{h,G}(x) = \frac{\sum_{i=1}^n x_i ||(x-x_i)/h||^2}{\sum_{i=1}^n g ||(x-x_i)/h||^2} - x \quad (1)$$

where g is the weight assigned to each point regarded to the kernel function chosen, and h represents the bandwidth parameter. The given location is represented as x , and x_i represents the neighbouring points. For analysis involving landmark extraction, a Gaussian distribution is commonly set up in the kernel function G .

2.4 Landmark characterization.

The third step is to characterize the resulting clusters by grouping them into different categories. This was done by analysing the textual descriptions contained on the hashtags attached to each photo by using a topic modelling algorithm. The topic modeller applied in our work is Latent Dirichlet Allocation, or LDA in short. One of the LDA's most important characteristics is that it can assign one word to more than one group at the same time. Like in English, one single word can have different meaning depending on the context used. This method has been frequently used in natural language processing in application such as tag recommendations, topic inference, and document summarization.

LDA is a generative model that represents documents as being generated by a random mixture over latent variables called topics [12]. A topic is defined as a distribution over words. For a given corpus (a collection of documents) of D documents each of length N_d , the generative process for LDA is defined as follows:

1. For each topic k , draw a distribution over words $\phi_k \sim \text{Dirichlet}(\beta)$ with $k = \{1, 2, \dots, K\}$
2. For each document d :
 - a. Draw a vector of topic proportions $\theta_d \sim \text{Dirichlet}(\alpha)$
 - b. For each word i
 - i. Draw a topic assignment $z_{d,n} \sim \text{multinomial}(\theta_d)$ with $z_{d,n} \in \{1, 2, \dots, K\}$
 - ii. Draw a word $w_{d,v} \sim \text{multinomial}(\phi_{k=z_{d,n}})$ with $w_{d,v} \in \{1, 2, \dots, V\}$

Note: Only the words w are observed.

2.5 Tour recommendation

The last step in our framework is the generation of tours. In this study, a long-short term memory neural network (LSTM) is applied to make the route recommendations. LSTM is a recurrent neural network (RNN) architecture that has good performance and a wide range of applications like text generation and analysis of sequences [13, 14]. This method belongs to the family of machine learning models, more specifically to deep learning. RNN has the peculiarity that its signals can travel in both directions, back and forward. There are many variations of RNN models such as bidirectional RNN, gated recurrent units, recursive neural networks, long-short term memory neural network (LSTM).

Different to traditional RNN, the LSTM model introduces a new structure called a memory cell. A memory cell, as shown in Figure 2, is composed of four main elements: an input gate, a neuron with a self-recurrent connection (a connection to itself), a forget gate and an output gate. The self-recurrent connection has a weight of 1.0 and ensures that, barring any outside interference, the state of a memory cell can remain constant from one timestamp to another. The gates serve to modulate the interactions between the memory cell itself and its environment. The input gate can allow incoming signal to alter the state of the memory cell or block it. On the other hand, the output gate can allow the state of the memory cell to have an effect on other neurons or prevent it. Finally, the forget gate can modulate the memory cell's self-recurrent connection, allowing the cell to remember or forget its previous state, as needed.

Fig. 2. The long short-term memory cell.

The equations below describe how a layer of memory cell is updated at every timestamp. x_s is the input to the memory cell layer at timestamp s , W and U are weight matrices, b is a bias vector. First, the input gate i_s will be computed, and the candidate value $\tilde{C}_s$ is for the states of the memory cells at timestamp s :

$$i_s = \sigma(W_i x_s + U_i h_{s-1} + b_i) \quad (2)$$

$$\tilde{C}_s = \tanh(W_c x_s + U_c h_{s-1} + b_f) \quad (3)$$

where σ is a *sigmoid* activation function and $\tanh$ is a hyperbolic tangent activation function.

Second, the forget gate activation f_s will be computed at timestamp s :

$$f_s = \sigma(W_f x_s + U_f h_{s-1} + b_f) \quad (4)$$

Given the value of the input gate activation i_s , the forget gate activation f_s and the candidate state value $\tilde{C}_s$, the new cell state vector C_s can be computed at timestamp s :

$$C_s = i_s \times \tilde{C}_s + f_s \times C_{s-1} \quad (5)$$

With the new state of the memory cells, the value of their output gates and, subsequently, their outputs can be computed:

$$o_s = \sigma(W_o x_s + U_o h_{s-1} + b_o) \quad (6)$$

$$h_s = o_s \times \tanh(C_{s-1}) \quad (7)$$

A particular LSTM structure called sequence to sequence (seq2seq) model is implemented for event predictions by using touristic landmarks and travelling routes. This model structure is very useful since it offers flexibility at handling the length of both, inputs and outputs. With this particular architecture, it is easier to generate tours under a specified length.

3 Implementation and experiments

3.1 Flickr data collection and pre-processing

First, we built up our dataset by querying photos from Flickr using as a reference the latitude and longitude of the centre of the city of San Francisco, California, under a radius of 5km. The resulting dataset was equivalent to a total of 514,765 pictures containing descriptive tags and geographic coordinates belonging to 15,117 unique users. After we merged the data into one file, we removed the duplicates and analysed the characters on each tag in order to exclude those words containing non-English characters.

3.2 Mean shift clustering

With mean shift clustering algorithm, landmarks can be extracted by adjusting the value of the bandwidth until the average distance is short enough. The bandwidth, which work as a windows size, is an argument associated to the convergence rate, density estimation and hence it will influences the resulting number of clusters. For example, a lower bandwidth will generate a higher amount of clusters but it will also slow down the convergence. After using mean shift to analyse the set of latitudes and longitudes from our dataset we got the results shown on Figure 3.

Fig. 3. Number of clusters and mean distances by using mean shift clustering algorithm.

While a very small bandwidth value might result in too many clusters, a large value might result in very few, merging some of them. Therefore, a short average distance is desired while trying to keep the right amount of clusters. Hence, by using mean shift clustering we decided to choose 0.0150 as the best bandwidth value, generating with this a total of 160 clusters and a mean distance of 3.65 km. Figure 4 shows how the identified landmarks are well-distributed through entire area of San Francisco city.

Fig. 4. The 160 attractive landmarks by using mean shift clustering.

3.3 Topic modelling

Once potential landmarks have been identified by using clustering algorithms, it is possible to classify those places into different categories (topics) by applying LDA. The LDA method can assign each cluster to one or more topics by analysing the tags contained on every photo and by extracting the similarities among their textual content. LDA has been applied on our photo dataset several times by trying different number of topics in order to find the optimal one. The indicator used to estimate the performance of the results is the semantic coherence value.

The semantic coherence has been used on other similar analysis to determine whether the selected number of topics can arrange related words, coherently [15]. The larger the coherence value the more coherent the popular hashtags from a topics will be. This value can be calculated by applying the following formula:

$$C_t = \frac{\sum_{t=1}^{|N|} \sum_{j=2}^m \sum_{i=1}^{j-1} \log \frac{D(t, w_i, w_j) + 1}{D(t, w_i)}}{|K|} \quad (8)$$

where C_t represents the coherence value; $|N|$ is the amount of topics; m is the value of most frequent word in topic t ; $D(t, w_i, w_j)$ is the number of virtual documents containing w_i and w_j in topic t ; and $D(t, w_i)$ represents the amount of virtual documents containing the word w_i in the topic t . After we iterated LDA by changing the number of topics, we show the results we got on Figure 5.

Fig. 5. LDA coherence score by number of topics.

On Figure 5, the amount of topics we tried vary from 4 to 50 by steps of 2. The larger the coherence score the better the results. As it can be noticed, the coherence doesn't improve much when the model has more than 32 topics ($C_t = 0.5269$). We also compared this results by using LDA with different number of topics such as 30, 24, 20, but the most frequent words among their topics were not as coherent as those gotten from the LDA with 32 topics, then we decided to use 32 topics for the further analysis of this framework. For example, the ten most important words on topic number 30 and their weights are as follow: summer (17.0%), painting (5.8%), green (3.9%), sanfranciscobotanicalgarden (3.0%), potrerohill (2.7%), animal (2.6%), nature (2.2%), sport (2.0%), decoration (1.7%), pink (1.5%). Clearly, topic 30 could be defined as "nature" or "park" since most of these words are related to these terms.

After our LDA model was defined with the right number of topics, the next step consists on assigning every cluster or landmark obtained from our mean shift clustering model to these topics. It is good to remember that by using LDA, one landmark can be assign to one or more topics, which is good since in real life one place could fit into more than one category. In order to decide if a cluster belongs to a topic or not it was necessary to define a threshold parameter. If the probability of a cluster to belong to a specific topic is higher than this threshold, then we can say that it belongs to that topic. After several tries, we found that 0.015 was a threshold value capable to generate good results at distributing the right amount of clusters on each topic. When the threshold was higher than 0.015, very few clusters were assigned on each topics; and when we tried it with a lower threshold we got too many cluster assigned on every topics. Therefore, we choose a threshold value of 0.015 in to continue with the development of our framework.

3.4 Tour recommender

Next, users need to select one among the 32 topics as their interest so that the generated route will be related to this topic. For example, if the topic we select is about nature, probably the recommended landmarks will be associated to places like parks, beaches, garden, etc. For demonstation purpose, we randomly selected topic 02, which refers to urban related items, as our target.

The next step is to obtain the latitude and longitude of the user's current location and calculate its nearest centroid. This centroid will represent the starting point of the route recommendation. Once done with this, we trained a traveler behavior modeler to predict the most likely destination to be visited by a new tourist based on

the experience of previous travelers. In this modeller, we defined an amount of 125 neurons, a word embedded size of 100, a learning rate of 0.01, a batch size of 100, and 25 epochs.

Figure 6 shows the four route suggestions when we selected the topic 02 (urban), and a route length of 6 landmarks. The proposed LSTM model needs only less than 1 second to make a recommendation but takes about 3 minutes to finish its training process. Fortunately, the training process will be done just one time by every topic.

Fig. 6. Output with LSTM model from four different locations.

4. Conclusion

After analyzing the geographic coordinates on the photo dataset, we concluded that the mean shift algorithms can generate a suitable result for geospatial clustering. With LDA we successfully grouped related words into topics to later categorize landmarks into topics. Finally, we successfully generate tour recommendations, demonstrating that LSTM model can be used not only for text generation but also to model travelers' behavior.

In the future we may consider comparing this framework with the implementation of other modern variation of recurrent neural networks architectures in order to fix the redundancy of landmarks in the generated routes. In a similar way, since it was normal to encounter recommendations where the resulting trajectories went back and forth, we may also consider to combine our framework with heuristic methods with the purpose to replace the recommended routes with more efficient ones by implementing the traveling salesman problem. In this way, tourists will be able to visit more places in the same time, or visit the same number of places in less time.

References

1. H. Zhang, K. Mohammed, C. David, and L. Gretchen, Mining photo-sharing websites to study ecological phenomena, *Proceedings of the 21st International Conference on World Wide Web*, 749-758 (2012).
2. T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, Travel route recommendation using geotags in photo sharing sites, *Proceedings of the 19th ACM International Conference on*
3. Y. Zheng, and X. Xie, Learning travel recommendations from user-generated GPS traces, *Transactions on Intelligent Systems and Technology*, **2**(1), 2:1-2:29 (2011).
4. C. Y. Tsai, and S.H. Chung, A personalized route recommendation service for theme parks using RFID information and tourist behavior, *Decision Support Systems* **52**(2), 514-527 (2012).
5. H. Hsieh, C.T. Li, and S.D. Lin, Exploiting large-scale check-in data to recommend time-sensitive routes, *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, 55-62 (2012).
6. K. Jiang, H. Yin, P. Wang, and N. Yu, Learning from contextual information of geo-tagged web photos to rank personalized tourism attractions, *Neurocomputing*, **119**, 17-25 (2013).
7. D. Gavalas, K. Charalampos, M. Konstantinos, and P. Grammati, A survey on algorithmic approaches for solving tourist trip design problems, *Journal of Heuristics*, **20**(3), 291-328 (2014).
8. Z. Xu, L. Chen, and G. Chen, Topic based context-aware travel recommendation method exploiting geotagged photos, *Neurocomputing*, **155**, 99-107 (2015).
9. Y. Sun, H. Fan, M. Bakillah, and A. Zipf, Road-based travel recommendation using geo-tagged images, *Computers, Environment and Urban Systems*, **53**, 110-122 (2015).
10. C.Y. Sun, and A. Lee, Tour recommendations by mining photo sharing social media, *Decision Support Systems*, **101**, 28-39 (2017).
11. D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg, Mapping the world's photos, *Proceedings of the 18th International Conference on World Wide Web*, 761-770 (2009).
12. D.M. Blei, A. Y. Ng, M.I. Jordan, Latent dirichlet allocation, *Journal of Machine Learning Research*, **3**(4-5), 993-1022 (2003).
13. N. Tax, I. Verenich, M.L. Rosa, and M. Dumas, Predictive business process monitoring with LSTM neural networks, *International Conference on Advanced Information Systems Engineering*, 477-492 (2017).
14. T. Luong, *Building Your Own Neural Machine Translation System in TensorFlow*. Retrieved on May 25th, 2018 from: <https://ai.googleblog.com/2017/07/building-your-own-neural-machine.html> (2017).
15. D. Mimno, H.M. Wallach, E. Talley, M. Leenders, and A. McCallum, Optimizing semantic coherence in topic models, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 262-272 (2011).