DEIM Forum 2017 P6-2

Collaborative Hotel Recommendation based on Topic and Sentiment of Review Comments

Zhan ZHANG† and Yasuhiko MORIMOTO†

† Graduate School of Engineering, Hiroshima University
Hiroshima University Higashi-Hiroshima 739-8521 Japan
E-mail: †{m154277,morimo}@hiroshima-u.ac.jp

Abstract We considered a method for recommending hotels in this paper. Most of conventional hotel recommendation systems use a vector space model to represent hotels. A hotel recommendation system decides attributes such as service, location, price, etc. in the vector space. In such system-defined fixed vector space, it is difficult to capture personal preference and evaluation viewpoint well, since each user has her/his preferences and viewpoints that are different from axis of the system-defined vector space. Therefore, we proposed a method that can automatically decide attributes of vector space. In most of systems, users can submit natural language comments, in which users can express various impressions about hotels. Therefore, in our method, we have utilized natural language comments that have submitted by users. We, first, analyzed texts in comments by using Latent Dirichlet Allocation (LDA) and extracted representative topics about hotels from the tex ts automatically. We further analyzed the texts to find sentiment for each extracted topic for each hotel. By using the topic and sentiment information, we generated our recommendations.

Key words Hotel recommendation(ホテル推薦),Collaborative Filtering(協調フィルタリング), Sentiment Analysis(センチメント分析),LDA(LDA).

1. INTRODUCTION

Recently, many travelers use travel websites when they select a hotel. Most of travel websites have functions for recommending hotels for travelers who are looking for a hotel. In this paper, we consider a hotel recommendation method that takes into account sentiments of review comments as implicit ratings for hotels.

There are several approaches of hotel recommendation. One of the most successful approaches is Collaborative Filtering (CF). There are two kinds of CF. One is“ user-based ” CF. The other is “ item-based ” CF. Given a user, the user-based CF first finds similar users of the user and then recommends items that the similar users like to the user. On the other hand, item-based CF computes similar items that the user liked and recommends the similar items.

Most of conventional hotel recommendation systems, including CF, use a vector space model to represent hotels. In a vector space model, a hotel is a record of an m-dimensional table where m is the number of attributes such as price, rating, distance, and so on. In the vector space model, the schema of the model is defined by a recommendation system. For example, each travel web site provides fixed hotel profiles, which include hotel's demographic information, such as age, price, star rating, and so on, and other subjective values, such as service quality, food, that are comes from users' evaluation sheets. In such system-defined fixed vector space, it is difficult to capture personal preference and evaluation viewpoint well, since each user has her/his preferences and viewpoints that are different from axis of the system-defined vector space. Therefore, we proposed a method that can automatically decide attributes of vector space.

In order to capture each users' personal preferences and viewpoints, we use natural language comments of users. We, first, analyzed texts in comments by using Latent Dirichlet Allocation (LDA) and extracted representative topics about hotels from the texts automatically. We further analyzed the texts to find sentiment for each extracted topic for each hotel. By using the topic and sentiment information, we generated our recommendations.

In more details, we propose a two-phase scheme which generates hotel vectors for each hotel automatically.

In the first phase, the hotel vector space is defined via LDA. LDA is a simplest topic model, where is used to discover a set of “ topics ” from a large collection of documents and infer topics for unclear document. In our purposed method, LDA extracts latent topics from user's review comments given in TripAdvisor. We regard latent topic space as hotel vector space to represent hotels.

In the second phase, the hotel vectors are generated for each hotel by referring the sentiment of each topic in its review comments. A hart work for us is to assigns sentiment to topics respectively in a review. In general, the sentiment might differ in a review. To assign sentiment to topics correctly, it is necessary to split review into some smaller parts. Based on our observation, one sentence tends to represent one topic and one sentiment in reviews. Jo Y and Oh AH [4] developed a review analysis model based on sentiment and aspect for online review. They assume all words in a single sentence are generated from one aspect and demonstrate that such assumption holds up well in practice. Therefore, In the review analysis step, we infer the topic and analysis the sentiment to generate topic-sentiment pair for each sentence. Aggregate such sentiment and topic pair as the result of review analysis. Finally, hotels are represented as a vectors based on the result of its reviews analysis.

The performance of proposed scheme is evaluated via experiment. In the experiment, we use a data set collected from TripAdvisor. the result of experiment indicates that the performance of our proposed method is better than Item -based CF and User-based CF.

The remainder of this paper is organized as follows. Section 2 overviews related works. Section 3 describes the details of the proposed scheme. Section 4 shows Implementation and Evaluation. Finally, concludes the paper with future work in section 5.

2. RELATED WORKS

2. 1 Latent Dirichlet Allocation

Topic modeling provides a suite of statistical tools to discover latent semantic structures from complex corpora, with latent Dirichlet allocation (LDA) as the most popular one. It is an example of a topic model and was first presented as a graphical model as Figure 1 for topic discovery by David Blei in 2003[3]. After training topic model, LDA represents documents as mixtures of topics that spit out words with certain probabilities. Based on such mixtures, it is capable of inferring topics of document. Nowadays, LDA is a widely used topic model in NLP. It has found many applications in text analysis, data visualization, recommendation systems, information retrieval and network analysis.

2. 2 Stanford CoreNLP tool

The Stanford CoreNLP tool provides a set of natural language analysis tools, it is including the part-of-speech (POS) tagger, the named entity recognizer (NER), the parser, and the sentiment analysis tools.

Stanford CoreNLP provides a new deep learning model to analyze the sentiment. It actually builds up a representation of whole sentences based on the sentence structure, computes the sentiment based on how words compose the meaning of longer phrases. The performance is better than all previous methods is indicated by several experiments in EMNLP paper.

2. 3 Hotel Recommendation systems

Hotel Recommendation systems has attracted attention in the past few years. The CF has been explored in travel [9] with the hotels of making recommendations to users. The similar users or items are identified using a similarity metric [10]. There are two methods used for make recommendation: Topic-K [11] and similarity thresholding [12].

A lot of studies use a vector space model to represent hotels, and use the text of reviews as its main data. Such as [1], The attributes of hotel vector space are manually denoted into several aspects such as food, services, sleep and so on. After that, they find representative words form reviews and classify the words into each aspect. Finally, Hotel vector are generated based on the result of such textual analysis of reviews.

Some studies extract the sentiment form reviews to represent hotels as a vector. In existing work [8], They propose recommendations using a combination of features. the built hotel aspects and extracting sentiment assign to each of them, by using additional knowledge such as nationality or purpose of trip built the profile of users. [13] build user profiles from users' review texts and use these profiles to filter other review texts with the eyes of this user.

3. PROPOSED METHOD

In this section, we describe the detail of proposed method to provide a recommendation of hotel for a user based on review comments. The proposed method consists of two parts, one is hotel vector generation, the other one is Item-based Collaborative Recommendation.

The main feature of our proposed method is that it generates hotel vectors for each hotel automatically. Hotel vector space definition and hotel vector generation is represented in section 3.1 and 3.2 respectively. The detail of Item-based Collaborative Recommendation is represented in section 3.3.

3. 1 Hotel Vector Space definition

Let $H = \{h_1, h_2, \ldots, h_N\}$ denote a set of hotels to be covered by the proposed method. We defined a Hotel Vector Space with $|K|$-dimensions. Given a collection of reviews associated with set H, the Hotel Vector Space is designed as follows.

At first, we set the parameter of topic's number as K. Let $T = \{t_1, t_2, \ldots, t_K\}$ denote a set of topics are automatically extracted by LDA in reviews. We regarded such topics as aspects of hotels and defined that each topic is corresponding to a dimension of our hotel vector space to represent hotel reasonably.

3.2 Hotel vector generation

This subsection shows how to map hotel into a point in the Hotel Vector Space. For a given hotel $h_n \in H$, we denote the vector of hotel h_n as $\overrightarrow{V}^{h_n} = \{\overrightarrow{s_1}, \overrightarrow{s_2}, \ldots, \overrightarrow{s_K}\}$, where $\overrightarrow{s_k}$ denotes the value of dimension in Hotel Vector Space. In the paper, the value $\overrightarrow{s_k}$ is calculated by referring the sentiment of each topic in its review comments. At first, we extract hotel h_n's review comments $r_n = \{r_{n1}, r_{n2}, \ldots, r_{nM}\}$ from TripAdvisor, let M be the number of comments and r_{nm} be a review which belongs to h_n . For a comment $r_{nm} \in r_n$, the key issue for us is to assigns sentiment to topics respectively. we split comment r_{nm} into sentence. For each sentence S_r, we generate topic-sentiment pair as $\langle t^{s_r}, S\rangle$, where t^{s_r} is the topic and S is the sentiment of sentiment S_r. The sentiment of each topic in comment r_m is calculated by aggregation of such topic-sentiment pair.

Although LDA is powerful to infer the topics of reviews, it is weak to infer the topic of a sentence since the size of sentence is too small (i.e., lack of words). Therefore, we cannot infer the topic for each sentence by LDA directly.

Recall that K denotes the number of topics, and T denotes the set of topics extracted by LDA. Let $P(t_k|r_m)$ be the probability of a review r_m concerns with topic t_k. With the above notions, Similar with the topic prediction process in [4,6], we infer the topic of given sentence S_r by following equation:

$$P(S_r|t_k) = \prod_{W_i \in S_r} P(W_i|t_k) \cdot P(t_k|r_m) \cdot P(r_m) \tag{1}$$

where

$P(S_r|t_k)$ denotes the probability that the sentence S_r belongs to Topic t_k.

$\prod_{W_i \in S_r} P(W_i|t_k)$ denotes the probability that the words W_i of S_r belong to Topic t_k, and $P(t_k|r_m)$ denotes the probability that Topic t_k belongs to review r_m.

$P(r_m)$ is the probability that review is chosen from the set of reviews, which is a constant.

Since we assume that, all words in a single sentence are generated from one topic, therefore, we calculate probability $P(W_i|t_k)$ of each topic $t_k \in \{t_1, t_2, \ldots, t_K\}$, and choose the one which has highest probability value of $P(S_r|t_n)$ as the topic of sentence t^{s_r}. t^{s_r} is determined as follows:

$$t^{S_r} = argmax\{P(S_r|t_k)|t_k \in \{t_1, t_2, \ldots, t_K\}\} \tag{2}$$

Since $P(r_m)$ is a constant, we ignore $P(r_m)$ and rewrite the Formula (3) as follows:

... 2. We went there before seeing Funny Girl at The Savoy Theatre. 3.Dinner was delightful and delicious. 4.I had a salmon that was so moist and wonderful.....

Figure 1 Example review from TripAdvisor.com

$$P(S_r|t_k) =\propto \prod_{W_i \in S_r} P(W_i|t_k) \cdot P(t_n|r_k) \tag{3}$$

Where $P(W_i|t_k)$ is given by Topic-word frequency-based matrix from LDA, and $P(t_k|r_m)$ is also calculated by LDA as well.

Example. Figure 2 depicts an example to illustrate the topic inference phrase. Figure 2 gives a review which consists several sentence.To infer the topic of sentence 3 which is *Dinner was delightful and delicious.* For each topic, we compute $P(S_r|t_k)$ by Formula (4.3) to calculate the probability.

For topic 0, we compute $P(S_r|t_k)$ as follows:

$P(S_r|t_k) = P(W_Dinner|t_0) \cdot P(W_delightful|t_0) \cdot P(W_delicious|t_0) \cdot P(t_0|r_m)$

$= 0.08 \cdot 0.02 \cdot 0.09 \cdot 0.425 = 0.00068$

Similarly, we calculate the probability with condition of each topic. After that, according to (4.2), we choose the highest one as t^{s_r} (e.g., the topic of sentence S_r). In this case, the topic of sentence 3 is topic 0.

We calculate S of topic-sentiment pair $\langle t^{s_r}, S\rangle$ for each sentence S_r by Stanford CoreNLP tool. In Stanford CoreNLP tool, the sentiment is represented by a label which belongs to *very negative, negative, general, positive, very positive*. Here we transform such label into a value S in $\{1, 5\}$, e.g., *very negative* to be 1, *negative* to be 2, etc.

For given hotel h_n, we denote the vector of comment $r_{nm} \in r_n$ as $\overrightarrow{V}^r = \{s1, s2, \ldots, sK\}$, where s_k denotes the value of dimension of comment vector $\overrightarrow{V}^r$, it is calculated by average of topic-sentiment pair $\langle t^{s_r}, S\rangle$. we group $\langle t^{s_r}, S\rangle$ by $t^{s_r} \in \{1, 2, \ldots, K\}$, and s_k is calculated by average of S in each group. Note that: s_k is defined as 0, If there are not mention about topic t_k in the review.

Example: Figure 3 give us several topic and sentiment pair in a review. The example shows how to calculate $\overrightarrow{V}^r$

$s_1 : (3 + 5 + 4) \div 3 = 4$

$s_2 : 3 \div 1 = 3$

$s_3 : 3 \div 1 = 3$

$\ldots$

$\overrightarrow{V}^r = \{4, 3, 3, \ldots, s_K\}$

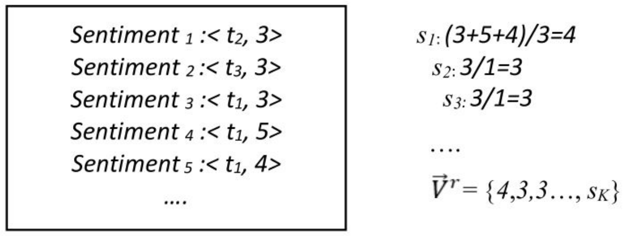

Figure 2 Fig.3. Example for aggregate topic and sentiment pair

Recall that, $\overrightarrow{V}^{h_n} = \{\overrightarrow{s_1}, \overrightarrow{s_2}, \ldots, \overrightarrow{s_K}\}$ denotes the hotel vector of hotel $h_n \in H$, r_n denotes the related review of hotel h_n. hotel vector $\overrightarrow{V}^{h_n}$ is calculated by aggregate of review vector $\overrightarrow{V}^r$ which are belong to r_n. The value of each dimension $\overrightarrow{s_k}$ is compute by following equation,

$$\begin{aligned} \overrightarrow{s_k} &= \sum_{\overrightarrow{V}^r \in R} s_k / I(if I \neq 0) \\ \overrightarrow{s_k} &= 0(if I = 0) \end{aligned} \tag{4}$$

where

s_k is the Sentiment value of Topic t_k in $\overrightarrow{V}^r$.

I denote the frequency of t_k appearance in reviews set r_n.

3.3 Item-based Collaborative Recommendation

In this section, we introduce how to make recommendation for a user. In order to perform Top-N recommendation for user u, we compute the estimated rating for every unrated hotel. It consists of two phases. In the first phase, we compute the similarity of hotels based on hotel vector. In the second phase, we predict the rating of unvisited hotel to user.

The similarity between hotel h_i and h_j is measured by cosine of the angle between hotel vectors $\overrightarrow{V}^{h_i}$ and $\overrightarrow{V}^{h_j}$. The similarity is denoted by $sim(h_i, h_j)$ and it is calculated by following equation,

$$Sim(h_i, h_j) = cos(\overrightarrow{V}^{h_i}, \overrightarrow{V}^{h_j}) = \frac{\overrightarrow{V}^{h_i} \cdot \overrightarrow{V}^{h_j}}{||\overrightarrow{V}^{h_i}||^2 \cdot ||\overrightarrow{V}^{h_j}||^2} \tag{5}$$

where $''\cdot''$ denotes the dot-product of the two vectors. The value of similarity is more near to 1, the hotels is more similar.

Let $\widehat{r}$ denote the prediction of User u_j to unvisited hotel h_i. LetH_j^u be the set of hotels which is rated by u_j. The prediction of rating $\widehat{r}$ of u_j to unvisited hotel h_i is given by:

$$\widehat{r} = \frac{\sum_{h_j^u \in H_j^u} sim(h_j^u, h_i) \cdot r_{u_j, h_j^u}}{\sum_{h_j^u \in H_j^u} (|sim(h_j^u, h_i)|)} \tag{6}$$

Where

h_j^u denote a hotel belongs to H_j^u.

Table 1 The part of LDA topic model

(topic)no.	words
1	check,call, desk, arrive, phone, charge, service ...
2	Guest, claim, problem, issue, service, clean ...
3	Walk, view, restaurant, shopping, taxi, breakfast ...
4	Airport, car, shuttle, drive, night, parking, pay ...
5	Breakfast, food, buffet, dinner, meal, order, caf? ...
6	Bar, place, drink, wine, feel, enjoy, people ...
7	Suite, Bed, TV, kitchen, shower, tub, sink, door...
...	...

4. IMPLEMENTATION AND EVALUATION

In this section, we conducted three experiments to evaluate the performance of proposed scheme using actual Tripadviosr dataset which download from DAIS(注1). First as the implementation phase, we represent our apprehension about topic model generated by LDA and the details of our dataset. After that we examine the effect of topic's number to find the best parameter settings. In each experiment we explain our method effect to the recommendation evaluated by MAE and RMSE.

4.1 Dataset

We used from TripAdvisor as data source. TripAdvisor is an authoritative travel website providing hotel ranking and information. Ten thousands of members are share their reviews in this website every day. The content of our dataset is including rating and review for each record. Our dataset contains 2256 reviews. The data density is about 3.57%.

4.2 Implementation

Preprocessing include Tokenizing, stemming, removing stop words. We can easily perform preprocessing of reviews data by Stanford CoreNLP tool. To generate LDA topic model, we using the JGibbLDA which is a LDA open source for Java. Table1 shows the extracted topic when we set the topic's number is 30. As we can see that, the *topic 1* and *topic 2* is related to service, *topic 3* and *topic 4* is related to Location, *topic 6*and *topic 7* is related to room...

4.3 Parameter Settings

To examine the effect of topic's number for make prediction of rating. We use two common quantities: root-mean-square error (RMSE) and mean absolute error (MAE) with different parameter settings $K = \{10, 15, 20, 25, 30, 35, 40\}$.

MAE measure how close predictions are to the eventual outcomes, and the formula as follows:

(注1)：[7] http://www.dims.ne.jp/timelyresearch/2008/080908/

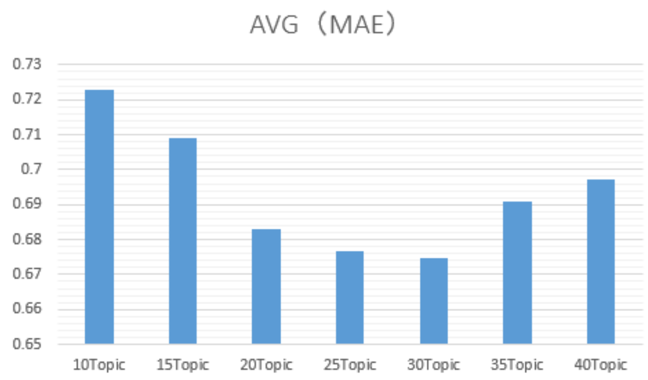

Figure 3 The avgerge of MAE

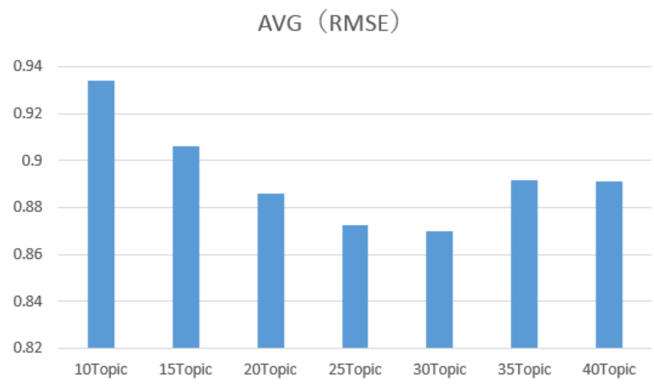

Figure 4 The avgerge of RMSE

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|r_i - \widehat{r}| \qquad (7)$$

Where, the N denote that the number of prediction. r_i denote the real rating and $\widehat{r_i}$ denote the prediction of rating.

RMSE measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed, the formula as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{N} e_j^2} \qquad (8)$$

We divide data into two sets, one is train data set and the other one is test data set. Train set contains 80% of all data and remaining data are used for test data set. For each trial, we implement 5 times and compute the averages of them to make sure the result is believable. Figure 3 shows the overall performance with different $K = 5, 10, \ldots 40$. For our method, we pick k=30; Figure 4 shows the performs when we fix K=30 compared with user-based CF and item-based CF.

In figure 3 we study the effect of the parameter K, and the improvement is best when the K=30. The reason is as follows. Too little cannot infer the topic of reviews. However, when K becomes large, one topic may be divided in to several topics more than one. That will raise an error too.

4.4 Evaluations

To evaluate the performance of our proposed method, we compare MAE and RMSE of our method with Item-based collaborative filtering and User-based filtering which are popular techniques of the Recommendation Systems. We use ItemKNN[11] to implement the item-based CF method, similarly use UserKNN to implement the user-based CF method.

As we see from Figure 4, the performance of our proposed our proposed method is better than both item-based CF and user-based CF. In more detail, MAE of our proposed method is about 0.674798202 and Item-based KNN is about

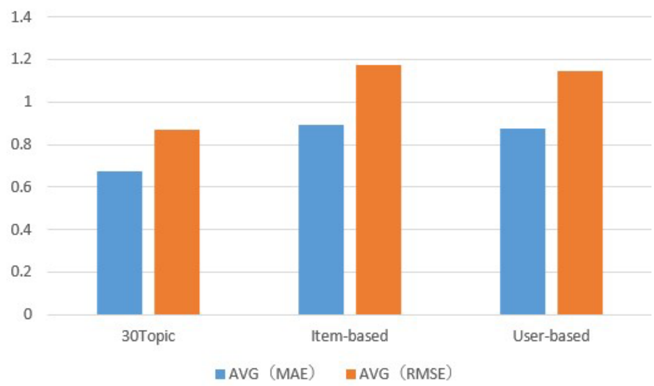

Figure 5 The result of Evaluation

0.891919, User-based KNN is about 0.87518. RMSE of our proposed method is about 0.869774758 and Item-based KNN is about 1.175592, Item-based KNN is about 1.148296.

With the experimental result, it is demonstrated that our method works effectively for providing recommendation.

5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a hotel recommender system that takes into account sentiments of review comments as implicit ratings for hotels. The first experiment showed that topic's number have an effect on accuracy of recommendation. The result of the second experiment indicated that our method works effectively for hotel recommendation.

As future work, we will try to find better method to infer topic for a sentence. Also with an improvement of recommendation method, we wish that more accurate results can be obtained in later work.

6. ACKNOWLEDGMENT

Firstly, I would like to show my deepest gratitude to my supervisor, Dr. Morimoto Yasuhiko. Also, my thanks go to laboratory member. And I am also grateful to my boyfriend who gives me a lot of supports both on lift and study.

References

[1] Hu Y H, Lee P J, Chen K, et al, "HOTEL RECOMMENDATION SYSTEM BASED ON REVIEW AND CONTEXT INFORMATION: A COLLABORATIVE FILTERING APPRO," 2016.

[2] Zhang J D, Chow C Y, Zheng Y, "ORec: An opinion-based point-of-interest recommendation framework," Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1641–1650. 2015.

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

[4] Jo Y, Oh A H, "Aspect and sentiment unification model for online review analysis," Proceedings of the fourth ACM international conference on Web search and data mining,ACM pp.815-824. 2011.

[5] Socher R, Perelygin A, Wu J Y, et al. "Recursive deep models for semantic compositionality over a sentiment treebank," Proceedings of the conference on empirical methods in natural language processing (EMNLP). pp. 1631–1642. 2013.

[6] Arora R, Ravindran B., "Latent dirichlet allocation based multi-document summarization," Proceedings of the second workshop on Analytics for noisy unstructured text data. ACM, pp: 91-97. 2008

[7] http://www.dims.ne.jp/timelyresearch/2008/080908/.

[8] Levi A, Mokryn O O, Diot C, et al, "Finding a Needle in a Haystack of Reviews: Cold Start Context-Based Hotel Recommender System," 2012.

[9] Zheng, Y., Burke, R., & Mobasher, B, "Differential Context Relaxation for Context-Aware Travel Recommendation. In Artificial Intelligence Review," pp. 88?99.2012

[10] Pappas, N., & Popescu-Belis, A, "Combining content with user preferences for non-fiction multimedia recommendation: a study on TED lectures. Multimedia Tools and Applications," 2015.

[11] Sun, J., Wang, G., Cheng, X., & Fu, Y, "Mining affective text to improve social media item recommendation. Information Processing & Management,"2014.

[12] Zheng, Y., Burke, R., & Mobasher, B, "A hotel recommendation system based on collaborative filtering and Rankboost algorithm," 2010 International Conference on MultiMedia and Information Technology, MMIT, pp.317?320. 2010.

[13] Zhang Y, Lai G, Zhang M, et al, "Differential Context Relaxation for Context-Aware Travel Recommendation. In Artificial Intelligence Review," pp. 88?99.2012

[14] Zhang Y, Lai G, Zhang M, et al, " Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. ACM, pp.83–92. 2014.