BJUT at TREC 2014 Contextual Suggestion Track: Hybrid Recommendation Based on Open-web Information

Hanchen Li, Zhen Yang*, Yingxu Lai, Lijuan Duan, Kefeng Fan
College of Computer Science, Beijing University of Technology, China
yangzhen@bjut.edu.cn

Abstract

In this paper we describe our efforts for TREC contextual suggestion task. Our goal of this year is to evaluate the effectiveness of: (1) Preference crawling method that as far as possible to obtain more candidate spots' information from open-web to model the users' interest profiles; (2) Automatic summarization method that leverages the information from multiple resources to generate the description for each candidate scenic spots; (3) Hybrid recommendation method that combing a variety of factors to construct a system of hybrid recommendation system. Finally, we conduct extensive experiments to evaluate the proposed framework on TREC 2014 Contextual Suggestion data set, and, as would be expected, the results demonstrate its generality and superior performance.

Introduction

In this year Contextual Suggestion (CS) Track, we main aims are two folds: (1) combing a variety of factors which are crawled from open-web to construct a system of hybrid recommendation system. (2) Explore a new description generation method which combines multiple aspects of information. Information recommendation is always a dilemma. It's a contradiction by generality and individuality. Recommend items need to make a compromise between popularity and user's personalized interest. First, the higher popularity of items tend that each user will like it, but it can't reflect users personalized interest. At the same time, recommending according to user's personalized needs the data describes the user's interest accurately. The data about spots crawled from open-web has sparseness problem, and it is difficult to truly reflect the personal interest of each user and reflects more of the spots' popularity.

In this sense, we crawled a variety of indirect information of scenic spots from the open-web such as: attractions, spots rank, reviews of spots, etc. using this information to reflect the quality of spots. Through analysis user profiles, we can get the interest preference of each user to each Category, and use spots in Example as the training dataset to train the SVM classifier for each user interest. Then, we use classifier to get the judgments about like or dislike for each user-spots pairs. Finally, we use the information crawled from website as the reflecting of spots' popularity, while use the user's interest which is analyzed from profiles as the reflecting of user's personalized interest. In the recommendation algorithm module, we combine the spots popularity and user personalized interest to generate two recommendation algorithms, eventually get BJUTa and BJUTb as two submitted results.

Our Method

System Framework

Figure 1 shows our system framework. It mainly consists of three parts: (1) Useful information gathering, (2) Examples labeling, (3) Profile Modeling and Interest classification, (4) Recommendation algorithm, (5) Description generation, (6) Results generation and checking. Figure 2 shows the legend of Figure 1.

- Useful information gathering component mainly crawls everything that we need to rank the candidate scenic spots.
- Examples labeling component determine the scenic spots' category in Examples through searching the internet and a small part of the manual scenic spots.
- Profile Modeling and Interest classification component mainly consists of two parts: (1) Modeling user profiles; (2) user-spots Interest classification. Statistical method is used to identify each user preferences for each category of spots. User-spots Interest classification use the spots in Examples as the training sample to train to the SVM classifier, and use it to classify the spots into two class, user like and user dislike.
- Recommendation algorithm component mainly consists of two parts: (1) for each user - context pair choose 50 candidate recommendation spots. (2) Sort the 50 candidate recommendation spots for each user - context pair.
- Description generation component mainly utilizes multi-resource information to generate spot's brief automatically. We also describe this part in details later this paper.
- Results generation and checking component get the recommend spots and spots briefly together, and use the official script to check the results and submit results to TREC.

Useful Information Gathering

The first step of solving the contextual suggestion problem is to gather useful information. Useful information contains not only the candidate scenic spots, but also their web

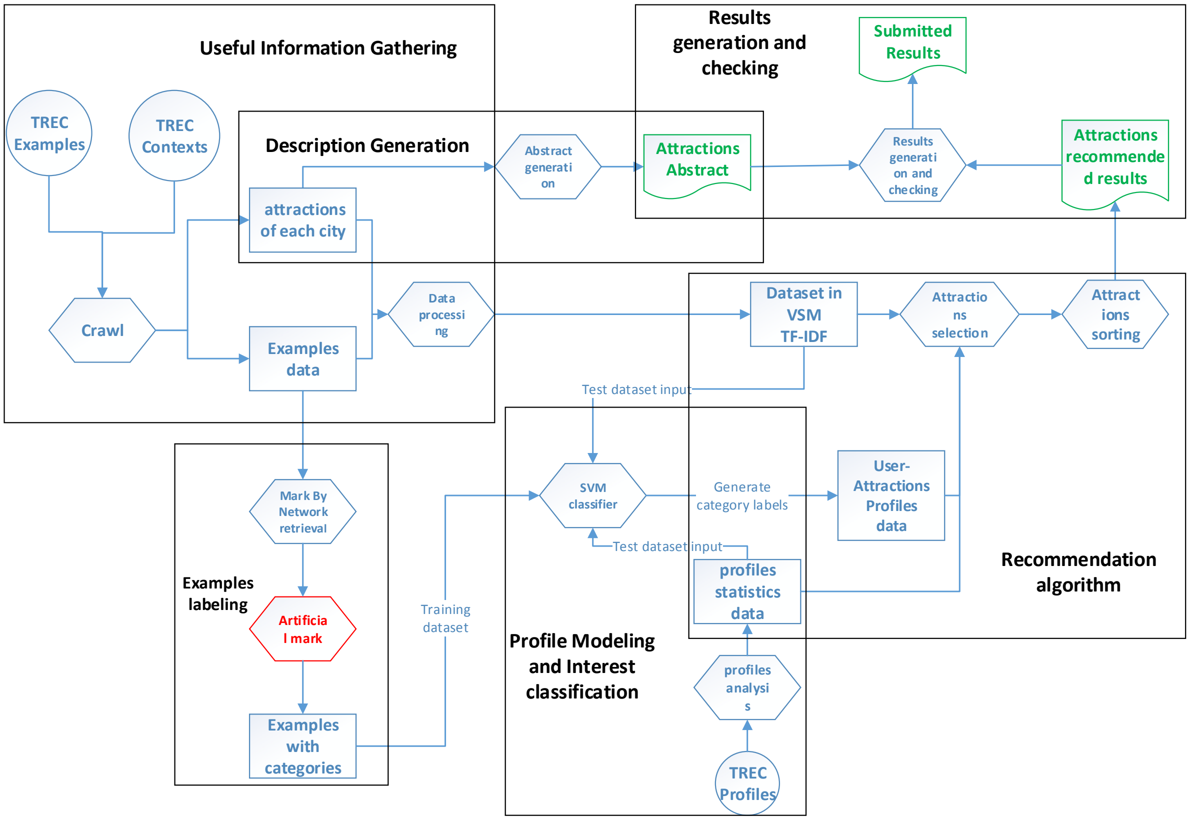

Figure 1: System Framework.

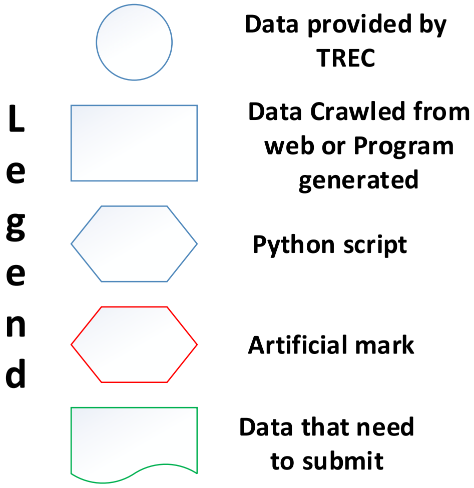

Figure 2: Legend of Fig. 1.

site, reviews associated with them and etc. For candidate scenic spots we use open web as they contain comprehensive information than ClueWeb12 does. We crawl the information such as candidate scenic spots, main page, categories and reviews of candidates exclusively from the website http://www.tripadvisor.com/ for all contexts. The crawling is done per category. All scenic spots are crawled in five categories from TripAdvisor, i.e., attractions, activities, restaurants, nightlife and shopping.

In order to make up for the spots data sparse problem, we crawl the indirect description as supplementary information for spots as much as possible, includes: rank on TripAdvisor, rate on TripAdvisor, reviews of spots, a review information includes: content, number, helpful vote number. We get the proportion of each category of the spots in Example according to the example label and crawl the spots according to the proportion for each context. Attraction, restaurant and activities crawl 120 as the top for each category, nightlife and shopping crawl 60 as the top for each category. For some context don't have 50 spots, according to the requirement of the location choice spots as the supplement which near the context city. Finally, we ensure each context has at least 70 spots.

All the crawlers are developed by python and the data is stored in a hierarchical relationship. Some data are normalized using the class structure of storage by cpickle module of python.

Profile Modeling and Interest classification

We need to analyze each user's interest in all kinds of spots categories. The meaning of user rating in Profiles is showed as follow:

- 4: Strongly interested
- 3 Interested
- 2: Neutral
- 1: Disinterested
- 0: Strongly disinterested
- -1: Website didn't load or no rating given

We take the website rate as the spot rate in profiles. Rate three and four points are considered as users likes the spot while zero and one points are considered as users don't like the spot. Point two is a neutral rating. First, in our Examples labeling module we classify each spot in Example, and then analyze the profiles to get the user interest to every Category. Finally, we get the probability of each user to every Category and the Users interested statistical Fig. 3.

We can see that the users of different categories of spots showed obvious interest orientation. For example: User for attraction and restaurant spots' like probability is higher than others and dislike probability is lower than others. Therefore, we can use the probability of user interest for each category as an important index of recommendation, to reflect the interest of each user. Through the analysis of the profiles, we classify spots in example into two categories, likes and dislikes for each user. Example spots are training set to train SVM classifier with RBF kernel function for each user. After that, each user will have a classifier which can classify a spot into like class or dislike class. So, we can use SVM classifier to classify every candidate spot for each user and label every spot with "like" or "dislike" for each user-spot pairs. We call this "user-spot favorite label". User-spot favorite label also reflects each user's personalized interest. User interest probability is aiming at a class scenic spots of user interest, while user-spot favorite label is aiming at a spot of user interest. They all reflect the personalized interest of the user and recommendation system.

Recommendation algorithm

We determine some criteria of the recommendation system according to the three evaluation standards of TREC: $P@5$, MRR and TBG. First of all, the recommended spots should conform to the requirements of location in context, so we will use location as a criterion of the recommendation system. Second, we will use the rank of spots on TripAdvisor and the rate of the reviews as the indicators of spots' quality, it embodies the commonness of recommendation system, while we use the probability of user interest for each category and the classification label of each user-spots pairs as the reflecting of the user personalized interest, it embodies the personality of recommendation system.

According to the requirement of TREC, we need recommend 50 spots for each user-context pair. In this sense, the whole recommended Algorithm is divided into two process: (1) choose 50 spots for each user-context pair (2) sort the spots of each user-context pair.

choose 50 spots for each user-context pair For 50 candidate spots, we choose the spots to follow the following principles. According to the priority order of each criterion, first, we choose the spots which location is meet the requirements of geographical of context. If all spots' location are conform to the requirements of geographical of context, location will not be a criteria of recommendation system. In order to guarantee the quality of TBG, the spots which conform the location requirement will be selected firstly. Then, we need determine the number of each category according to the probability of user's interests. We determine the number of each user-context pair by following formula:

$$number_i = sum_i \times p(like|category_i) \qquad (1)$$

where $i \in category\{attraction/activities/restaurant/shopping/nightlife\}$, sum_i is the spots number of $category_i$ in the context, $p(like|category_i)$ is the conditional probability.

There, $number_i$ is the upper bound of $category_i$.It is important to note that for the context that its spots don't fit the location requirement, we will prefer to select the spots which location fit with the requirements of context. If this strategy results in a certain number of spots of one category exceeded the upper bound, the spots of this category will not be selected any more. Other categories of spots will shrink their upper bound in Proportion of the category's upper bound.

After determine the spots' number of each category, our algorithm can be divided into two kinds:(1)Only-rank algorithm ,only using rank as a criteria, sort the spots of all categories together by rank and choose the spots in order. (2) like-rank algorithm, using rank and user-spot label (like or dislike) as criterion. We prefer to select the spot with like label. If some spots all have like label, choose the spot with higher rank. The result of Only-rank algorithm is BJUTa, while the result of like-rank algorithm is BJUTb.

sort spots of each user-context pair After choosing the 50 recommended spots, we need to sort this 50 spots to generate the final recommended list. Here, we used two different algorithms to sort the 50 spots, the difference between two algorithms is whether to use the user-spot label as a criteria in sortingalgorithm. Because the TREC using $P@5$ as one of the main evaluation indicators, in order to improve the quality of the first five recommended spots, each spot sorting algorithm can be divided into two parts: (1) first six spots sorting (2) rest of the spots sorting.

- First six spots sorting
  In order to guarantee the quality of $P@5$, we recommend the former six spots continue to separate selection and sorting. In first six spots, two of them are selected from the category which the probability of user like is the highest. Two of them are selected from the category which the probability of user dislike is the lowest. Two of them are selected from the category which has the largest number of spots in recommended 50 spots. After this step, we can know the number of each category in first six spots. We sort the spots in each category by rank and select spots according to the amount of this category in first six spots. Then we will get the first six spots of BJUTA result. When we select the first six spots, if we need all these spot has the user-spot label with like, we will get the first six spots of BJUTb.

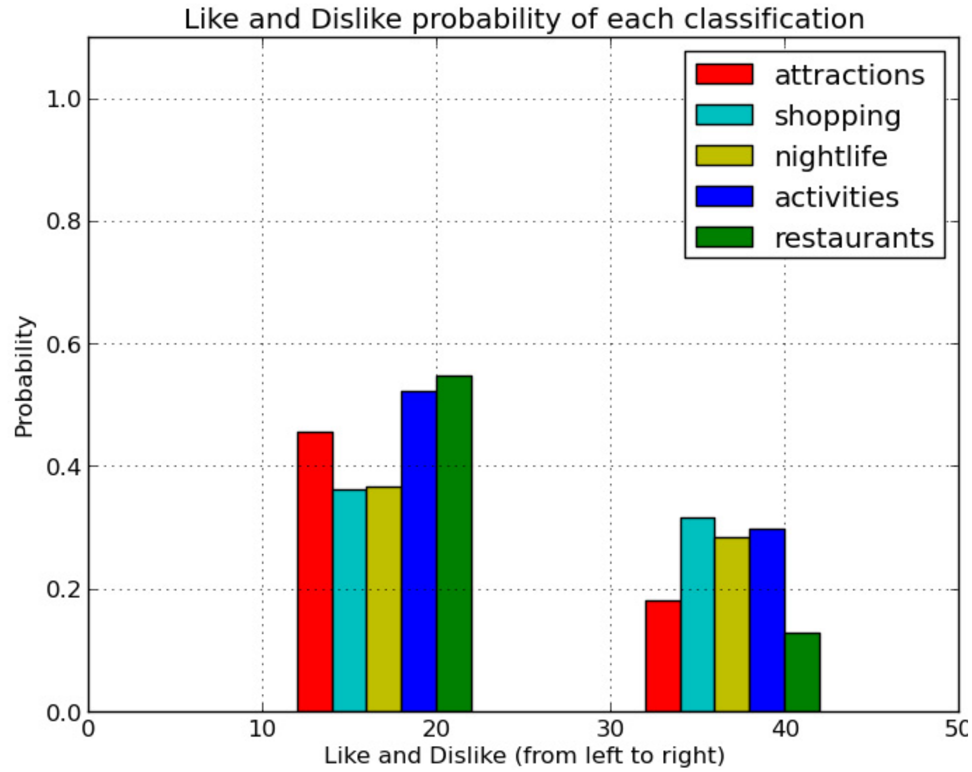

Figure 3: Users interested statistical.

- Rest of the spots sorting

  First of all, we sort the probability of user interest of dislike for each category in ascending way. We use 0.3 and 0.8 as two thresholds to filter each category. The categories which user dislike probability less than 0.3 are as a bunch of classes and give priority to recommend. While the categories which user dislike probability greater than 0.8 are recommended at last. For the categories which user dislike probability are in 0.3 to 0.8, we use treat the probability as one-dimensional vector and use k-means method to cluster the categories into two clusters. We use each category vector as the center of initial cluster and get the best initial cluster center. If only two categories are remained, they will be treated as a cluster. Finally, we can get some cluster which consists of some categories that its dislike probability is in different level. Then, we choose spots in the cluster which has a lower dislike probability of cluster center, sorting these spots by rank and add them to the recommended list. The cluster which has a higher dislike probability of cluster center will be recommended later. Follow this rule, and we can get the recommended list of remaining spots.

At last, combining the two results of (1) (2) and we will get the final submitted results, BJUTa and BJUTb.

Description Generation

There are mainly two parts of the data obtained from openweb by our method: less brief introductions and a great deal of consumer critics, and text information of websites of spots. Among the data lots of disordered information is included. For great evaluation such as TREC, characteristics of automatic description for each user are not an important thing. Researchers show that language and logic are the prime factors for description. That is to say, a description that is lucid always tends to a better evaluation. Therefore, extraction for a whole sentence and the compound of other information are our method to generate automatic description. The descriptions are gained by following things:

- The introduction information of the spot net
- Reviews has “helpful vote” or a good score to the scenic spot
- Some introduction information on TripAdvisor

Brief description of candidate spots are made following such a template: scenic spot’s name + introduction information on TripAdvisor + concrete introduction of spots. More detailed methods for description are done according to 3 levels.

- Level 1: description information on spot website
- Level 2: Critic information with helpful vote tag on critics
- Level 3: Critics whose score provided by consumers is better than average points

The sentences which have most number of words for each level are extracted. After all sentences for a higher level have been extracted, sentences at lower level can be extracted. Add extracted sentences until the brief description reaches 512 bytes. There, data structure similar to stack is used.

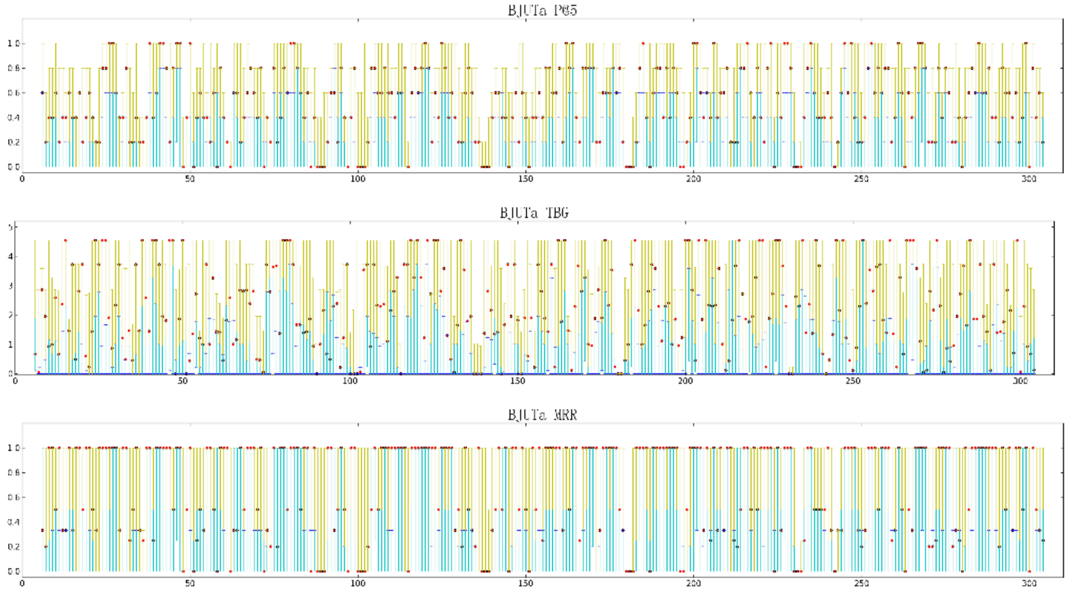

Figure 4: Detailed Results of BJUTa.

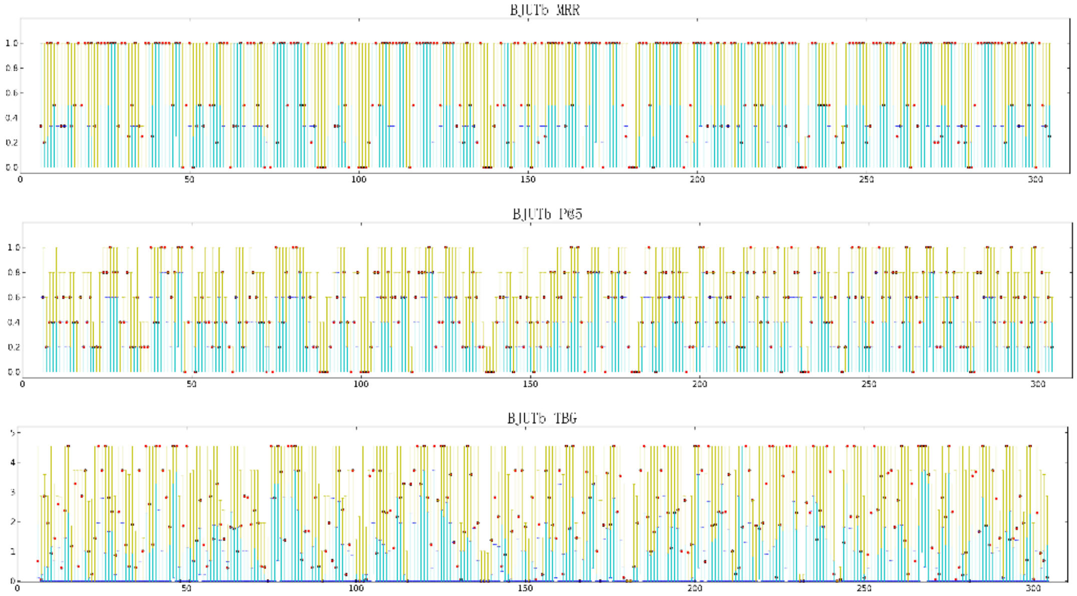

Figure 5: Detailed Results of BJUTb.

From top to bottom level, complete sentence are extracted and add to automatic description. The introduction information of spot website is obtained by statistics of word numbers of text in each html label and the number for words containing spot's name. These methods not only guarantee the fluency of the automatic description, but also make it best for description to have more information of spot advantages.

Examples labeling

The spots in the example are provided by TREC. Interests towards the example spots have been evaluated by users. In order to analyze interests of users, classification on spots of the example is needed to get interest information of users. Internet search is utilized to classify each spot, whose realization method for is: search at the TripAdvisor using spot name in the Example as the key, and the scenic spot's category which its name and location information are satisfied with the example will be returned. For the spots that are not

Table 1: Overall Mean Performances.

	BJUTa	BJUTb
P@5	0.5057	0.5037
MRR	0.6850	0.6700
TBG	2.1993	2.2003

found in TripAdvisor, we will label them by hands. There are 100 spots in the Example, among which 87 spots are searched out, 13 spots are marked by hands.

Submitted Runs and Experiment Results

We submitted two runs: BJUTa and BJUTb. BJUTa only uses spot rank and probability of user interest in each category as the basis for selection and sorting candidate's spots, while BJUTb uses spot rank and probability of user interest in each category and user favorite label of each spots as the basis for selection and sorting candidate's spots. For description, it combines the opening sentence, meta-description of the web site, the review which has helpful vote tag and the sentence from the reviews which are good reviews of the scenic. We use 10-fold cross-validation method for the parameters of RBF kernel function in SVM training process.

As for judgments, 304 profile/context pairs were sampled and judged. There are four criteria: geographic appropriateness, interestingness of website and description. Geographic appropriateness has 3 scales: 0, 1, 2, with 0 not geographically appropriate, 1 marginally geographically appropriate, and 2 geographically appropriate. Interestingness of website and description has 5 scales: 0, 1, 2, 3, 4, with 0 strongly uninterested, 1 uninterested, 2 neutral, 3 interested, 4 strongly interested. Moreover, geographic appropriateness was judged partially by NIST assessors and partially by users. Scores from NIST assessors were used if a profile/context pair was judged by both of them.

Three evaluation measurements were reported: $P@5$ (precision at 5), MRR (mean reciprocal rank) and TBG (time-biased gain). For $P@5$ and MRR, a suggestion is labeled as relevant only if it has geographic score of 1 or2, and description score of 3 or 4, and web site score of 3 or 4.

Figure 4 and Figure 5 show the performances of our two runs in terms of all evaluation measurements. The X axis consists of all profile/context pairs, ordered alphabetically in the format of "profile-context". One red point represents our result for that profile/context. Three bars corresponding to the best, median and worst results of that profile/context. The yellow line is the range which is above the average. We can see that our runs achieve some of the best results. Most of our results are equal or better than the median results, indicating the effectiveness of our proposed system. Table 1 shows the overall mean performances of our runs in terms of all evaluation measurements.

Conclusion

In TREC 2014 Contextual Suggestion Track, we submitted two runs. Both of them use the indirect description information of candidate spots and user interest information to select and sort the candidate spots. Indirect description information of candidate spots include: spot's category, location and rank. User interest information includes: probability of user interest in each category and user favorite label of each spots. We use these indicators to make recommendation algorithm. The spots category, location information, rank, and probability of user interest in each category are used to get the result of BJUTa, while the spots category, location information, rank, and probability of user interest in each category and user favorite label of each spots are used to get the result of BJUTb. Due to the open-web data sparseness problem, our recommendation algorithm does not depend on the similarity between two spots, but using a variety of indirect description of scenic spot from the open - the web which reflect the quality of spots and user profile which reflect the user interest to select and sort the candidate spots. We use a variety of information on the open-web with whole sentence extraction method to generate spots brief automatically.

The performances of our two submitted runs are in general better than the median performance. Some of the results are even best results, indicating the effectiveness of our proposed method.

References

1. Albadvi A, Shahbazi M. A hybrid recommendation technique based on product category attributes[J]. Expert Systems with Applications, 2009, 36(9): 11480-11488.
2. Sobecki J, Babiak E, S?anina M. Application of hybrid recommendation in web-based cooking assistant[C]//Knowledge-Based Intelligent Information and Engineering Systems. Springer Berlin Heidelberg, 2006: 797-804.
3. Shih Y Y, Liu D R. Hybrid recommendation approaches: collaborative filtering via valuable content information[C]//System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on. IEEE, 2005: 217b-217b.
4. Xu J A, Araki K. A SVM-based personal recommendation system for TV programs[C]//Multi-Media Modelling Conference Proceedings, 2006 12th International. IEEE, 2006
5. Wen H, Fang L, Guan L. A hybrid approach for personalized recommendation of news on the Web[J]. Expert Systems with Applications, 2012, 39(5): 5806-5814.
6. Joachims T. Text categorization with support vector machines: Learning with many relevant features[M]. Springer Berlin Heidelberg, 1998.