Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186–5140
Volume 4, Number 1, pages 21–26, January 2015

How to Extract Seasonal Features of Sightseeing Spots from Twitter and Wikipedia (Preliminary Version)

Guanshen FANG, Sayaka KAMEI and Satoshi FUJITA
Hiroshima University, Japan

***Abstract*—In this paper, we consider a tourism recommender system which can recommend sightseeing spots for users who wish to make a travel plan for a designated time period such as early autumn and Christmas vacation. A key issue in realizing such seasonal recommendations is how to calculate feature vector of each spot which would vary depending on the time of travel. We propose a two-phase scheme which generates seasonal feature vectors for each sightseeing spot. In the first phase, the basic feature vector is generated for each spot using the description of Wikipedia and the TF-IDF weights. In the second phase, seasonal feature vectors are generated for each spot by referring to the distribution of keywords contained in tweets associated with spots for each season. The performance of the scheme is evaluated via experiments using actual data set drawn from Wikipedia and Twitter.**

***Index Terms*—Tourism recommender system, seasonal feature vector, Wikipedia, Twitter.**

I. Introduction

According to the widespread of smart-phones and wireless communication services, we have become to easily acquire useful information from the web in a timely and seamless fashion. Such information includes the exposition of facilities (e.g., museum, temple and castle) and the schedule of attractions held in a designated time duration (e.g., Oktoberfest), which are generally called **tourism information** when the objective of the searcher is to make a travel plan in a particular region for a particular time period. Of course, such tourism information is useful even while traveling. For example, travelers can find an appropriate route to nearby facilities via online navigation services such as NAVITIME[1] and MapFan[2], and can acquire a detailed exposition of facilities via Wikipedia and the webpage of the facility.

In this paper, we consider tourism recommender system which can recommend sightseeing spots for users who wish to make a travel plan for a designated time period such as early autumn and Christmas vacation. A key issue in realizing such "seasonal" recommendations is how to calculate feature vector of each spot which would vary depending on the time of travel. For example, Mt. Fuji, which is a mountain in Japan registered as a World Heritage Site in 2013, shows a variety of appearances for each season. We can enjoy winter sports at the foot of the mountain from December to February and could enjoy mountain climbing from July to September. It would also be nice to enjoy the variety of scenery around Mt. Fuji, e.g., we can see beautiful cherry blossoms in April and can view scarlet maple leaves in November. As such, Mt. Fuji should be recommended to travelers who prefer to mountain climbing in summer vacation, while it should be recommended to travelers who prefer to hiking during spring and autumn.

As the source of such seasonal information, we will focus on Twitter which is a typical Social Network Service (SNS) widely used by many users in the world; e.g., it is reported that the number of monthly active users of Twitter exceeds 271 million in 2014 and the average number of daily tweets is 58 million. As will be described later, a certain portion of tweets issued in this SNS are relevant to some sightseeing spot, and due to the nature of SNS so that every tweet is tagged with a time stamp, it could effectively work as the source of seasonal information of sightseeing spots. In fact, Bannur and Olonso [1] demonstrated that the popularity of sightseeing spots certainly changes across seasons by analyzing check-in data of users in Twitter.

In this paper, we propose a concrete scheme to generate seasonal feature vectors for each sightseeing spot. The proposed scheme consists of two phases. In the first phase, the basic feature vector is generated for each spot using the description of Wikipedia and the TF-IDF weight of keywords contained in these documents. In the second phase, seasonal feature vectors are generated for each spot by referring to the distribution of keywords contained in tweets associated with spots

[1]http://www.navitime.co.jp
[2]https://www.mapfan.net

for each season. The performance of the scheme is evaluated via experiments using actual data set drawn from Wikipedia and Twitter. The experimental result indicates that: 1) seasonal feature vectors generated by the proposed scheme drastically changes from February to March, and 2) the result of recommendation certainly changes over time by using seasonal feature vectors generated by the proposed scheme.

The remainder of this paper is organized as follows. Section II overviews related works. Section III describes the details of the proposed scheme. Section IV shows the result of preliminary experiments conducted by using a prototype of the proposed tourism recommender system and a collection of real-world data drawn from Wikipedia and Twitter. Finally, Section V concludes the paper with future work.

II. Related Work

Tourism recommender systems proposed in the literature can be classified into three types, namely CF-based recommender systems, contents-based recommender systems, and their hybrid, where CF is an abbreviation of collaborative filtering.

CF-based tourism recommender systems can be further classified into user-based systems and item-based systems, where the former focuses on the similarity of users and the latter focuses on the similarity of sightseeing spots (items) in an appropriate metric space. In typical user-based systems, user A is recommended an item *which is positively evaluated by other users to have similar preference with* A. As for the way of calculating the similarity of preferences, several techniques have been proposed in the literature. Chen *et al.* [2] use the check-in history to mobile applications for the characterization of users. More concretely, they regard the check-in history containing text data as a document, and characterize the interest of users using the TF-IDF weight of keywords contained in the document (e.g., users who are interested in hot spring will be characterized by a larger weight of keywords relevant to hot spring since such keywords frequently appear in their check-in history). Ye *et al.* [3] use the friends relation in SNS in calculating the similarity of interests. They implemented a recommender system based on this idea, and demonstrated that the use of friends relation could improve the quality of recommendation of sightseeing spots. A challenging issue to be overcome in these CF-based tourism recommender systems is how to improve the quality of recommendation when the total number of evaluations is small, which is widely recognized as the cold-start problem in CF-based schemes.

In contents-based recommender systems, each user is recommended an item *which is the most fitted to the user*. The fitness between users and items is determined by the similarity in a metric space, which is calculated by extracting features of users and items by using statistical techniques [4]. Such an approach would work well if we could generate accurate feature vectors for the users and items. However, it is difficult to generate such vectors in actual tourism recommender systems since the feature of sightseeing spots varies for each season and the feature of users could not be accurately acquired without conducting a detailed questionnaire survey.

The above issues of CF-based and contents-based approaches could be overcome by taking a hybrid of them, i.e., the cold-start problem can be partially resolved with the aid of contents-based approach and the inaccuracy of feature vectors could be partially overcome with the aid of evaluations conducted by other users. Kurashima *et al.* [5] improve the accuracy of feature vectors of users in hybrid systems by identifying the user's scope and patterns of daily activity through the analyses of GPS logs. Meehan *et al.* [6] propose a hybrid tourism recommender system which considers several exterior factors such as the time and the weather of the destination. Such supplemental information could improve the accuracy of recommendation given by hybrid systems, but to the best of our knowledge, no conventional hybrid system supports the seasonal difference of sightseeing spots.

III. Proposed Method

A. Overview

The main feature of the proposed tourism recommender system is that it can distinguish the *feature of sightseeing spots for different seasons* which was not explicitly considered in conventional systems. The key idea is to introduce the notion of **seasonal feature vectors** which reflect the seasonal difference of each sightseeing spot. In the following, we explain how to generate such vectors for each sightseeing spot.

The proposed scheme consists of two phases1. In the first phase, **basic feature vectors** are generated from the description of Wikipedia. Let $O = \{o_1, o_2, \ldots, o_L\}$ denote the set of spots to be covered by the proposed system. For each $o_l \in O$, we extract document w_l relevant to "tourism" from Wikipedia and extract a set of keywords D_l from w_l through the morphological analysis using Mecab [7]. The basic feature vector of spot o_l is calculated as a vector of TF-IDF weights of keywords contained in document w_l. After that, the second phase calculates the frequency of the occurrences of keywords in tweets for each spot, and generates a feature vector

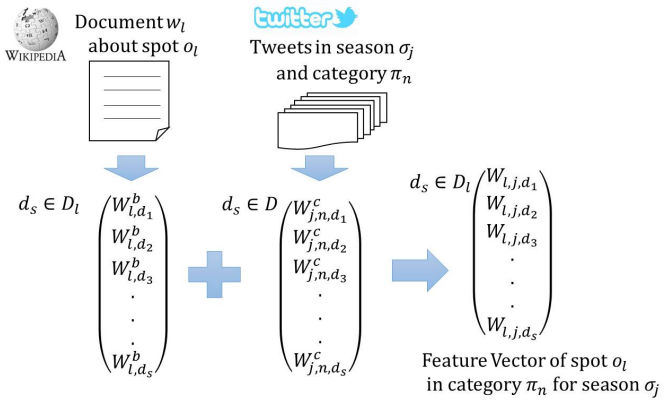

Fig. 1: Overview of the proposed system.

of the spot for each season. At last, we take a linear combination of the above two vectors (i.e., a vector generated from Wikipedia and another vector generated from tweets), to generate a seasonal feature vector of the spot.

B. Basic Feature Vector

Given a collection of documents associated with set O, the basic feature vector of each sightseeing spot is constructed as follows. Recall that w_l denotes the document associated with spot o_l and D_l denotes the set of keywords contained in document w_l. Let f_{l,d_s} be the number of occurrences of keyword d_s in document w_l, and M_{d_s} be the number of documents containing d_s. With the above notions, the TF-IDF weight W^b_{l,d_s} of keyword d_s in document w_l is calculated as follows:

$$W^b_{l,d_s} := \frac{f_{l,d_s}}{\sum_{d \in D} f_{l,d}} \cdot \log\left(\frac{L}{M_{d_s}}\right) \quad (1)$$

where $D \stackrel{\text{def}}{=} D_1 \cup \ldots \cup D_L$. The basic feature vector of spot o_l is the vector of TF-IDF weights W^b_{l,d_s} of keywords contained in D.

C. Seasonal Feature Vector

This subsection shows a concrete procedure for the second phase. At first, we divide the time period of one year into J seasons. Let $S = \{\sigma_1, \sigma_2, \ldots \sigma_J\}$ denote the resulting set of seasons. In the proposed scheme, we construct a seasonal feature vector of each sightseeing spot for each season in S. More specifically, J seasonal feature vectors of sightseeing spots are constructed from the basic feature vectors and a collection of tweets associated with the spots. The basic idea behind such a two-step approach is that: 1) typical keywords associated with spot o_l have already been contained in the Wikipedia document w_l, 2) the "typicalness" of such keywords will change over time, and 3) such a temporal distribution of the typicalness of keywords could be approximated by the frequency of the occurrences of keywords in tweets, if the number of relevant tweets is sufficiently large. The reader should note that by restricting the set of keywords to be contained in seasonal vectors to the keywords contained in Wikipedia documents, "meaningless words" contained in typical tweets could effectively be discarded.

Another important feature of the proposed method is that it explicitly introduces the notion of **category** to overcome the rareness of tweets associated with minor spots. Let $P = \{\pi_1, \pi_2, \ldots, \pi_N\}$ be the set of categories prepared by the system, where we assume that each spot is associated with several categories in P (in the current version of the system, we use five categories drawn from guidebooks, namely Culture, Nature, Shopping, Art and Entertainment.). Let P_l $(\subseteq P)$ denote the set of categories associated with spot o_l. With the above notions, the weight of keyword d_s concerned with "sightseeing spot o_l in season σ_j" is calculated as follows:

$$W_{l,j,d_s} := W^b_{l,d_s} + \alpha \sum_{\pi_n \in P_l} W^c_{j,n,d_s}, \quad (2)$$

where α is a parameter and W^c_{j,n,d_s} denotes the weight of keyword d_s concerned with "category π_n in season σ_j" which is formally defined in the next paragraph.

Let $T_{j,n}$ denote a set of tweets relevant to category π_n which are tweeted in season σ_j. Let f_{j,n,d_s} be the number of occurrences of keyword d_s in subset $T_{j,n}$ and $D'_{j,n}$ be the set of keywords contained in subset $T_{j,n}$. Note that $D'_{j,n}$ might contain keyword which does not appear in D, and vise versa. With the above notions, for each keyword $d_s \in D$, the weight W^c_{j,n,d_s} of keyword d_s in subset $T_{j,n}$ is calculated as follows:

$$W^c_{j,n,d_s} := \frac{f_{j,n,d_s}}{\sum_{d \in D} f_{j,n,d}} \cdot \log\left(\frac{J \times N}{R}\right) \quad (3)$$

if $d_s \in D_{j,n}$ and $W^c_{j,n,d_s} := 0$ otherwise, where R is the number of subsets containing d_s. The reader should note that in the above definition, words in $D'_{j,n} - D$ are simply omitted and words in $D'_{j,n} \cap D$ are given a TF-IDF weight by regarding that the set of tweets contained in $T_{j,n}$ is a document.

D. Implementation

We implemented a prototype of the proposed tourism recommender system in the following environment: OS: openSUSE 12.3, Language: Java, IDE: Eclipse, and Application Server: Tomcat. Figure 2 illustrates an overview

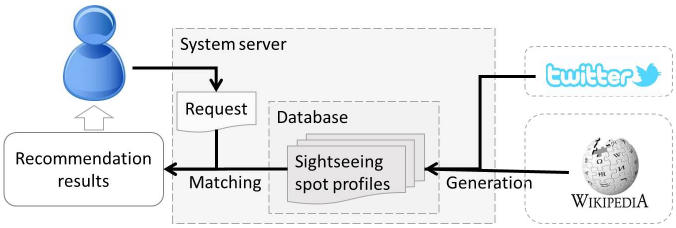

Fig. 2: Overview of the prototype system.

of the prototype system. This system consists of three main parts; namely Request Generator, Spot Profile Generator, and Decision Maker. Spot Profile Generator generates seasonal feature vectors for each sightseeing spot by using the method described in Section III. More concretely, we selected 1187 sightseeing spots in Japan (i.e., $L = 1187$) which contain many famous spots such as Mt. Fuji and Tokyo Disneyland, and extract documents and keywords from the corresponding Wikipedia pages. The keywords extraction is done using Mecab.

As for the set of tweets, we collect more than 5×10^7 Japanese tweets from October 2013 to April 2014 using Twitter Streaming API, and extract about 6×10^4 tweets relevant to tourism by using the name of sightseeing spots. The number of seasons is fixed to $J = 12$, so that each month corresponds to a season. To avoid the separation of important seasonal features extending over two months (e.g., cherry - blossom viewing from the end of March to the beginning of April), we take the following approach to generate feature vector for each season: 1) generate the feature vector $W_{l,j}$ of spot o_l in season σ_j by using Equation (2) for each j; and 2) take the average of vectors for the $(j-1)$st, j^{th} and $(j+1)$st seasons to generate the (refined) feature vector $W'_{l,j}$ of spot o_l in season σ_j, i.e., $W'_{l,j} := W_{l,j-1} + W_{l,j} + W_{l,j+1}$.

IV. Evaluation

We evaluate the performance of the proposed scheme by using data set drawn from Wikipedia and Twitter.

A. Change of Configuration of Feature Vectors

At first, we evaluate the impact of the proposed scheme to the configuration of seasonal feature vectors. In the proposed scheme, sightseeing spots sharing similar features in a given season will have similar feature vectors in the metric space, which implies that for each season, sightseeing spots sharing similar features organize a "cluster" and such a clustering will vary over time. We conduct the following experiments to verify this conjecture: 1) cluster seasonal feature vectors of L sightseeing spots by using k means method for each season with $k = 34$; and 2) examine the purity of each clustering with respect to another clustering constructed for a prior season. Let $\mathcal{C}_j = \{C_{1,j}, C_{2,j}, \ldots, C_{k,j}\}$ denote the clustering for season σ_j. The **purity** of clustering $\mathcal{C}_j$ with respect to a clustering $\mathcal{C}_{j'}$, where $j' < j$, is defined as

$$\Phi_{j,j'} \stackrel{\text{def}}{=} \sum_{i=1}^{k} \left(\max_{1 \le \ell \le k} \frac{|C_{i,j} \cap C_{\ell,j'}|}{L} \right).$$

Note that a large $\Phi_{j,j'}$ indicates that two clusterings $\mathcal{C}_j$ and $\mathcal{C}_{j'}$ are similar. Thus we could identify the season at which the configuration of feature vectors significantly changes by identifying the season at which the purity significantly decreases.

TABLE I: Purity of clusterings.

$j - j'$	Nov	Dec	Jan	Feb	Mar	Apr
1	0.81	0.73	0.73	0.76	0.47	0.83
2	-	0.69	0.63	0.71	0.42	0.41
3	-	-	0.58	0.59	0.42	0.35

Table I summarizes the results, where the purity of clustering for November (i.e., $j = 2$) with $j - j' \ge 2$ cannot be calculated since collected tweets start from October. We can observe from the table that: 1) the purity significantly reduces when $j = 6$ (i.e., March) and $j - j' = 1$, which indicates that the configuration of feature vectors drastically changes from February to March, and 2) the purity gradually degrades for $j = 4$ and 5 as the value of $j - j'$ increases, which indicates that the configuration changes even during winter although the amount of change per month is not very large.

B. Change of Similarity of Vectors

Next, we evaluate the impact of the proposed scheme to the result of recommendation. In the last experiment, the clustering for April contains a cluster of 31 sightseeing spots associated with the cherry blossom viewing. Let C^* denote the cluster. Members in C^* are associated with similar vectors in April since they are characterized by keywords relevant with April such as cherry blossom viewing, but in other seasons, the similarity of these spots is expected to degrade since they are also relevant with other attractions such as the promenade, rivers and so on. Thus, by evaluating the cosign similarity of vectors associated with those spots for each season, we can predict the impact of the proposed scheme to the result of recommendation, provided that the query is given at the point of spots.

Table II summarizes the result. In the table, Hi means "Hiikawa dike row of cherry blossom trees," San means

TABLE II: The change of similarity of five spots contained in cluster C^*.

	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Yama-Hyaku	0.2054	0.2091	0.2262	0.2192	0.2818	0.1298	0.5138
Yama-San	0.2966	0.2586	0.2739	0.2827	0.2639	0.3003	0.5841
Yama-Kuma	0.2661	0.2515	0.2545	0.2924	0.2561	0.1452	0.5401
Yama-Hi	0.1385	0.1261	0.1314	0.1484	0.1334	0.0847	0.4844
San-Kuma	0.3367	0.3094	0.2799	0.2924	0.2921	0.2108	0.7622
San-Hi	0.2391	0.2563	0.2934	0.3309	0.3263	0.2398	0.7321
San-Hyaku	0.1666	0.1619	0.1597	0.2314	0.2427	0.1607	0.7276
Kuma-Hi	0.1611	0.1529	0.1364	0.1989	0.2108	0.2072	0.7368
Kuma-Hyaku	0.2737	0.3224	0.3864	0.3766	0.4669	0.2794	0.7806
Hi-Hyaku	0.2911	0.2752	0.2657	0.3173	0.3187	0.2255	0.7206

TABLE III: Top 12 similar spots of "Yamaguchi Reservoir" in October, January and April.

Oct	Jan	Apr
Murayama Reservoir	Murayama Reservoir	Nippo seaside Scenic Byways
Earth pillars	Mt. Nokogiriyama	Chidori-ga-fuchi
Village Park of the Echizen narcissus	Hoheikan	Forest of Rokugo
Chidori-ga-fuchi	Wakasahyonosen Village	Row of cherry blossom trees of Yamakita $*$
Awa history culture way	Cedar Avenue of Nikko	Green Park of kyogamaru
Earth 33	Todai-ji Temple Daibutsu-den Hall	Ojagaike
Mt. Fuji Radar Dome Museum	Hashima Island	Chikiu Cape
Cedar Avenue of Nikko	Izumigamori	Kirigaki observatory
Inubosaki Lighthouse	Suigo-Tsukuba Quasi-National Park	Iyashi Motenashi Kamiyama Highway
Shurikinjochou stone pavement way	Musashiranzan	Unugase valley
Otoi	Kimimachisaka	Kumagaya Sakurazutsumi $*$
Tamadoubutsu Park	Awa history culture way	Sesshouseki

"Row of cherry blossom trees of Yamakita," Hyaku means "Hyakujuro sakura," Yama means "Yamaguchi Reservoir," and Kuma means "Kumagaya Sakurazutsumi." We can find from the table that the similarities of these five spots in April are the highest over all seasons. Table III shows the top 12 spots closest to "Yamaguchi Reservoir" (Yama) over L sightseeing spots for October, January and April. The reader should note that this result corresponds to the list of recommendations to the query given at the point of Yama in the metric space. We can see from the table that the ranking in April contains two sightseeing spots in cluster C^*, while in other seasons, none of the above five spots is contained in the ranking. This suggests that the seasonal feature vectors generated by the proposed scheme can certainly realize a seasonal recommendation of sightseeing spots.

V. Concluding Remarks

This paper proposes a scheme to generate seasonal feature vectors of sightseeing spots, which can be used as a part of tourism recommender system supporting seasonal recommendation of sightseeing spots to the users making a travel plan. The proposed scheme generates seasonal feature vectors using data sets drawn from Wikipedia and Twitter, and introduces the notion of category to overcome the rareness of tweets concerned with minor sightseeing spot. We conducted preliminary experiments using a prototype of the proposed system implemented using Java. The result of experiments indicates that many sightseeing spots' attractions and its popularity vary seasonally, and our proposed method can successfully characterize them into vectors to construct profiles in tourism recommender systems.

A future work is to propose a method to extract the feature of each user without conducting a lengthy questionnaire survey. We are trying to apply techniques proposed in our previous papers [8]. The evaluation of the proposed method as a tourism recommender system is another issue to be challenged.

References

[1] S. Bannur and O. Alonso, "Analyzing temporal characteristics of check-in data," in *Proc. Companion Publication of the 23rd International Conference on World Wide Web Companion*, 2014, pp. 827–832. [Online]. Available: http://dx.doi.org/10.1145/2567948.2579041

[2] C. Hongbo, C. Zhiming, M. S. Arefin, and Y. Morimoto, "Place recommendation from check-in spots on location-based online social networks," in *Proc. 3rd International Conference on Networking and Computing*, 2012, pp. 143–148.

[3] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proc. 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011, pp. 325–334. [Online]. Available: http://doi.acm.org/10.1145/2009916.2009962

[4] Q. Liu, H. Ma, E. Chen, and H. Xiong, "A survey of context-aware mobile recommendations," *International Journal of Information Technology and Decision Making*, vol. 12, no. 1, 2013.

[5] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura, "Geo topic model: Joint modeling of user's activity area and interests for location recommendation," in *Proc. 6th ACM International Conference on Web Search and Data Mining*, 2013, pp. 375–384. [Online]. Available: http://doi.acm.org/10.1145/2433396.2433444

[6] K. Meehan, T. Lunney, K. Curran, and A. McCaughey, "Context-aware intelligent recommendation system for tourism," in *Proc. IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2013.

[7] MeCab, http://mecab.sourceforge.net/.

[8] S. Hayashi, Y. Inoshita, and S. Fujita, "An efficient web page recommendation based on preference footprint to browsed pages," in *Proc. 5th International Workshop on Computational Intelligence & Applications (IWCIA 2009)*, 2009.