TRACE: Travel Reinforcement Recommendation Based on Location-Aware Context Extraction

ZHE FU, University of North Carolina
LI YU, Renmin University of China
XI NIU, University of North Carolina

As the popularity of online travel platforms increases, users tend to make ad-hoc decisions on places to visit rather than preparing the detailed tour plans in advance. Under the situation of timeliness and uncertainty of users' demand, how to integrate real-time context into dynamic and personalized recommendations have become a key issue in travel recommender system. In this article, by integrating the users' historical preferences and real-time context, a location-aware recommender system called TRACE (**T**ravel **R**einforcement Recommendations Based on Location-**A**ware **C**ontext **E**xtraction) is proposed. It captures users' features based on location-aware context learning model, and makes dynamic recommendations based on reinforcement learning. Specifically, this research: (1) designs a travel reinforcing recommender system based on an Actor-Critic framework, which can dynamically track the user preference shifts and optimize the recommender system performance; (2) proposes a location-aware context learning model, which aims at extracting user context from real-time location and then calculating the impacts of nearby attractions on users' preferences; and (3) conducts both offline and online experiments. Our proposed model achieves the best performance in both of the two experiments, which demonstrates that tracking the users' preference shifts based on real-time location is valuable for improving the recommendation results.

CCS Concepts: • **Information systems** → **Recommender systems;**

Additional Key Words and Phrases: Travel recommender system, reinforcement learning, location-aware context learning

ACM Reference format:
Zhe Fu, Li Yu, and Xi Niu. 2022. TRACE: Travel Reinforcement Recommendation Based on Location-Aware Context Extraction. *ACM Trans. Knowl. Discov. Data.* 16, 4, Article 65 (January 2022), 22 pages.
https://doi.org/10.1145/3487047

65

1 INTRODUCTION

With the development of information technology and wide use of smartphones, the collections of user data from tourism platforms become more and more available. The rich user data makes the high-quality and personalized recommendation possible. To deal with the massive and complex

This research is supported by National Science Foundation (NSF) (Award #1910696). We would like to thank NSF to make this research possible.
Authors' addresses: Z. Fu and X. Niu, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, North Carolina 28223-0001; emails: {zfu2, xniu2}@uncc.edu; L. Yu (corresponding author), Renmin University of China, No. 59 Zhongguancun Street, Beijing 100872, China; email: buaayuli@ruc.edu.cn.

1556-4681/2022/01-ART65 $15.00
https://doi.org/10.1145/3487047

data from users, many researches have made efforts on integrating various information including users' tour histories, comment texts, geographical locations, and so on to provide personalized attraction recommendations for tourists. On users' side, it is challenging for users to collect lots of information on attractions and to plan their traveling routes in advance. As a result, users are likely to make ad-hoc decisions on the visiting places depending on their preferences as well as their locations. So, a real-time recommender system is important to recommend and arrange travel plans based on real-time users' context. However, it is a challenging issue to promptly capture the real-time information and compute its impact of recommendation results in a recommender system.

To address with these challenges, many existing studies attempted to design a context-aware recommendation system based on machine learning algorithms. On one hand, some studies (e.g., Chen et al. [7]; Guo et al. [8]; Jiang et al. [12]; and Shi et al. [24]) focus on extracting the users' characteristics by analyzing their personal information as well as historical travel data, and then predicting the potential user interests, which help the system to effectively recommend appropriate attractions to users. On the other hand, there are also many studies (Aydin, & Telceken [4]; Kurashima et al. [15]; Sun et al. [26]; Xu et al. [34]; and Yamasaki et al. [35]) focusing on optimizing the travel routes of attractions. These works consider the relationship between the users' preference and attractions' locations, then attempt to find out the most economical or time-saving routes to users. Although these efforts have achieved significant performance gain, most of them avoid to track the users' real-time shift of preferences from users' dynamic interactions with systems and ignore the impact of users' current location context on attraction selection, missing the opportunity in maximizing user satisfactions.

In this article, we propose a recommender framework called **Travel Reinforcement Recommendation based on Location-Aware Context Extraction** (**TRACE**). First, the proposed method models the recommendation process of attractions as a **Markov Decision-Making Process** (**MDP**) in order to evaluate the long-term benefits of recommendation results rather than just calculating the short-term accuracy. Second, we construct a reinforcement recommendation framework for the modeled MDP based on the Actor-Critic framework to dynamically train and update the parameters. In addition, to help the TRACE model track the users' real-time context, we innovatively operationalized the meaning of travel context as travel sites distances and travel sites sequences and proposed a location-aware user context learning model. Finally, we also evaluate the performance of TRACE model using real-world data. There are three main research tasks in this study: user preference extraction based on users' historical behaviors, user context learning based on real-time location information, and recommender system development.

Our contributions mainly include the following:

- The proposed TRACE recommender system innovatively integrates attention mechanism, context-aware learning approach, and **reinforcement learning** (**RL**) techniques in travel recommendations, where users' needs are complex and dynamically changing.
- Our novel location-aware learning algorithm considers both travel sites distances and travel order popularity to update the user context real time and dynamically, which has special importance for the travel domain.
- The proposed TRACE was evaluated by both offline and online experiments from the computational and human-centered perspectives. Both experiments demonstrated the effectiveness of TRACE.

2 RELATED WORK

This work integrates three lines of research: travel recommender systems, user context learning, and RL. We will review each of these lines in the following subsections.

Table 1. Research on Travel Recommender Systems

Type	Study	Data Source	Method
Preference Prediction	Shi et al. [23]	Comment text	Collaborative Filtering
	Guo et al. [8]	Comment text	SVM, LDA
	Jiang et al. [12]	Comment text Travel photo	Topic Model
	Chen et al. [7]	Travel photo	Bayesian Probability Model
Route Optimization	Xu et al. [34]	Comment text	Monotone Ratio Scheduling Algorithm
	Aydin et al. [4]	Comment text Historical route	Ant Colony Algorithm
	Yamasaki et al. [35]	Historical route	Markov Model
	Sun et al. [26]	Historical route	Hyperchain Induced Topic Search Model
	Kurashima et al. [15]	Historical route	LDA, Markov Model
	Huang et al. [11]	Historical route	Multi-destination Route Planning Algorithm

2.1 Travel Recommender Systems

Nowadays, online travel platforms are growing rapidly. More and more travel information can be shared conveniently. It has become an important resource for tourists to make plans. Many studies are now focusing on optimizing travel recommendation services by analyzing the data from online travel communities based on machine learning algorithms. Table 1 summarizes the two types of those studies: preference prediction and route optimization.

Most of the studies prioritize on recommending attractions to tourists by analyzing their historical comments or routes information. For example, Shi et al. [23] proposed an ontology-driven recommendation strategy based on user context, which could integrate users' direct demands and potential preferences in recommendation context based on the proposed ontology. Jiang et al. [12] proposed user topic modeling and route topic modeling by mapping the description texts of users and routes. At the same time, they used two kinds of social media information to complement each other: travel diaries and photos. Chen et al. [7] detected the photos from the Flickr platform to automatically obtain the demographic characteristics and travel path of individuals or groups, which could improve the performance of personalized recommendations to users.

However, these studies did not incorporate heterogeneous and rich context sources of travel information such as location and routes, made available today with advances of technology. This study believes the context information has great potential in inferring users' preferences of travel attractions. Therefore, it will move forward with incorporating those contextual information sources into travel recommendations.

2.2 Recommendations Based on User Context

One of the most critical tasks in personalized recommendation is to comprehensively calculate users' preference representations based on user context. Traditional recommender systems aimed to recommend the items that can potentially be attractive to users based on their past preferences. However, more and more recent research has noticed that user context can help filter out irrelevant past preferences (Aliannejadi, & Crestani [3]and Zhou et al. [40]). Contextual information is a kind of extra information in addition to traditional users' ratings or behavior data. It is widely used to solve the cold-start problem and improve the users' satisfaction in recommender systems (Zeng et al. [36]). Recently, due to the explosive growth of e-commerce platforms, more and more attention has been drawn to integrating user context with user historical behavior, which has apparent advantages in extracting users' preferences. For example, Hong et al. [10] believed that users' interests and trust relations in mobile scenarios can be affected by context of time and location. Therefore, to improve the quality of the user experience, they proposed an efficient recommendation approach by combining the context-aware interests and context-aware trust values obtained from users' behavior. Unger et al. [27] designed a deep learning recommendation framework integrating structured contextual information, including time, locations, and user activities, with a **Neural Collaborative Filtering** (**NCF**) algorithm to improve the performance of recommender systems. Another attempt conducted by Alhamid and his colleagues [1, 2] on improving the user experience of multimedia content recommendation was leveraging contextual information associated with user-item interaction tuple to represent the latent preferences of users and items toward contexts. Moreover, Wu et al. [31] proposed a context-aware user-item representation learning model based on **Factorization Machines** (**FM**) for rating prediction, which extracts latent feature interactions between users and items as contextual information to better model the user's rating behavior.

However, the impacts of users' context in recommender systems are not static, and not all features of context have the same impacts on users. Therefore, many studies attempted at introducing attention mechanism (Bahdanau et al. [5]; Chen et al. [6]; and Xu et al. [33]) to adjust the weights of different types of context. Wang et al. [29] attempted to imitate the editors' behavior of filtering news by proposing a dual attention mechanism to adjust the news' context of timeliness and categories for each day. In the research of session-based recommender systems, Liu et al. [18] believed that the existing methods based on LSTM models could effectively capture users' general preferences from historical records, but failed to focus on the impacts of users' recent preference. Therefore, their research further focused on the users' last click actions and proposed a short-term attention/memory priority model to capture the short-term preferences from their last clicks. These efforts collectively show the effectiveness of context-aware recommendation on improvement of user satisfaction. In this article, we applied this context-aware idea into our research. More importantly, we innovatively operationalized the meaning of travel context as travel sites distances and travel sites sequences.

2.3 Recommendations Based on RL

Due to its adequate model capability and the support of dynamic feature capture, RL has achieved huge success in personalized recommendation. Compared with traditional deep learning algorithms, RL is able to quickly extract the user's preferences and maximize their benefits from user-system interactions. Therefore, more and more research efforts begin to integrate the RL and deep learning technologies to deal with real-time recommendation tasks. Zheng et al. [39] pointed out that users' preferences will change dynamically in the situation of news recommendations. In order to capture the user's real-time behavior, they proposed a **Deep Q-Network** (**DQN**) based RL framework, which captured the shift of preferences through the user-system interactions. Zhao et al. [38] found that most existing recommender systems based on deep learning models only focused on the current accuracy and ignored potential future rewards. To address the problem, they used the MDP to model the interaction process between users and system, and proposed a

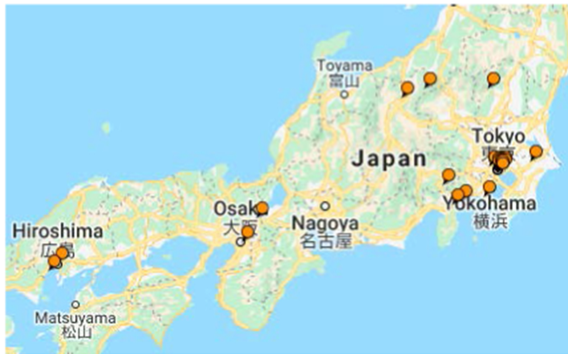

Fig. 1. Travel sites of an example user in Japan.

recommendation model based on a **Deep Deterministic Policy Gradient (DDPG)** framework. Similarly, Lei and Li [16] also formulated the problem of interactive recommendation for different users as MDP and proposed a DQN-based model to estimate optimal policies. Wang et al. [28] studied the problem of helping doctors to recommend prescriptions to patients. They proposed a supervised RL framework, which combined the advantages of deep learning and RL to deal with the complex relationship among three factors of drugs, diseases, and patients.

Although those recent studies have fully considered the users' preferences and the attractions' characteristics, they only calculated the best matched attractions with users' preferences through learning the history records, focusing on short-term matches without considering the user gain in a longer time span where users' preferences may shift and evolve. We believe RL is able to weigh the long-time benefits in addition to the short-term immediate accuracy of recommendation results. Therefore, this study incorporated a RL technique into the TRACE recommender system to overcome the shortcomings mentioned above.

3 USER BEHAVIOR ANALYSIS AND FEATURE CONSTRUCTION

In this section, we first analyzed the users' behavior from historical data and found the impact of real-time locations and the shift of preferences may affect the users' choices on attractions. Besides, to comprehensively represent the users' characteristics, we proposed a hybrid preference learning model, which can track both the users' long-term and short-term preferences.

3.1 User Behavior Analysis: An Example User of TripAdvisor

To comprehensively understand the behavior patterns and characteristics of real-world travelers, we collected the user data from TripAdvisor, one of the largest online travel platforms in the world with hundreds of millions of user records, as the use case for research.

3.1.1 User Preference Shifts. In reality, most of the tourists may not always choose the same type of attractions, and sometimes they would like to explore new attractions that may be different from their past preferences. The following is an example of a user in TripAdvisor traveled to Japan from March 15 to March 26, 2020, and left a series of trip records on the website each day (as shown in Figure 1).

After analyzing the types of different attractions visited by the user for each day, as shown in Figure 2, we found that in the first four days of the trip, the most visited attraction type was "landmark architecture", and from the 5th to the 7th day was mainly "natural landscape" sites,

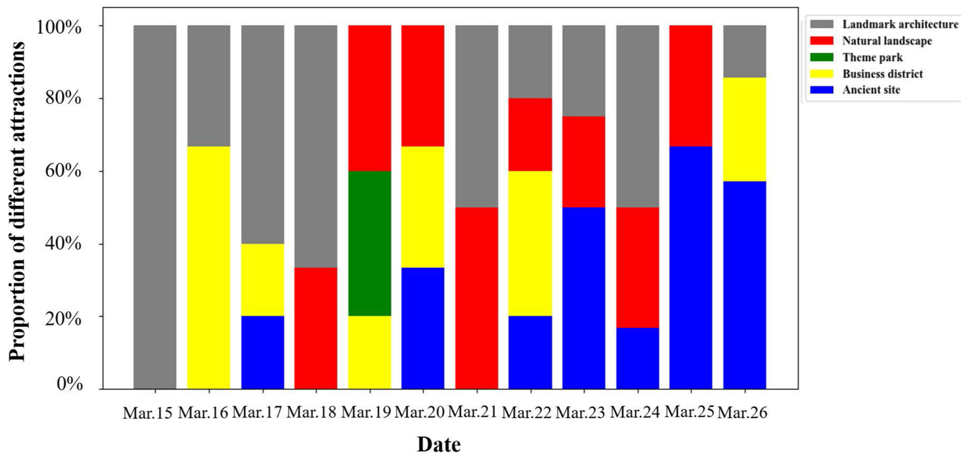

Fig. 2. Attraction types of the example user.

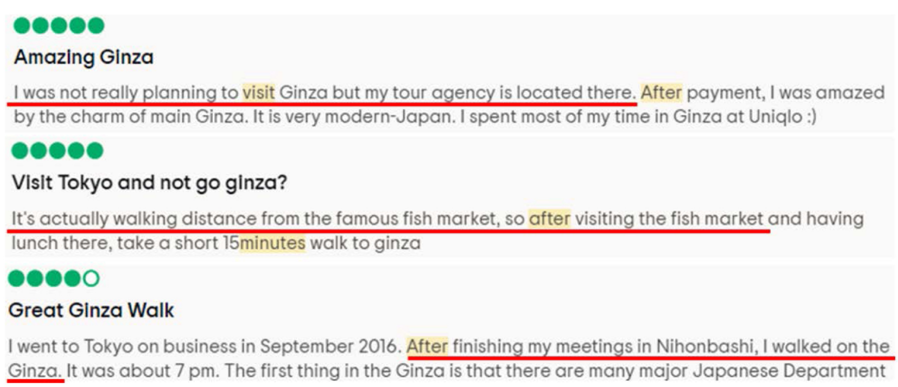

Fig. 3. User comments on Ginza.

while in the last four days was "ancient site" attractions. Therefore, even in a few-day trip, users may shift their preferences.

3.1.2 Influences of Location Context. In addition to individuals' preferences, users' selections may often be affected by the distance, traffic of attractions, or the experiences from other tourists. Travel distance is the reflection of the cost and time (Wei et al. [30]). Users may sometimes choose the attractions nearby rather than the attractions they preferred. For example, as shown in Figure 3, there are several comments of Ginza from the TripAdvisor website. These comments indicated the reasons of visiting "Ginza" was the proximity of its location to where the users were. As a result, we conclude that the users' current location plays an important role in users' next-round choice of attractions.

3.2 Preference Extraction

In this subsection, we introduce our preference extraction method, including long-term and short-term extraction models, which will be exploited to develop a comprehensive representation of users' preferences based on their historical travel records.

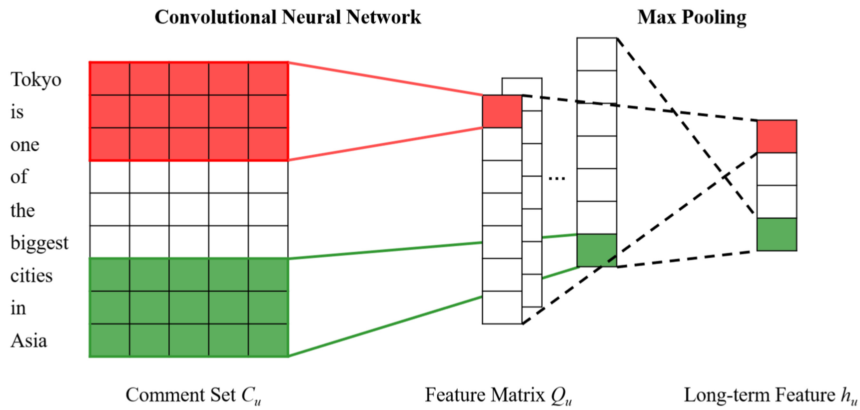

Fig. 4. Long-term feature extraction based on CNN.

3.2.1 Long-Term Feature Extraction. With the explosive growth of online travel platforms, tourists are now more and more familiar with sharing travel experiences and commenting on attractions. As a result, it is easy for us to obtain users' ratings and their traces of attractions from the online platforms. A user comment set is formed by merging this user's all comments on all the attractions, representing as $C_u = \{e_1, e_2, \ldots, e_n\}$, in which e_i represents the embedding of ith word generated by the skip-gram algorithm [19] in the comment set after removing the stop words. Compared with traditional textual modeling methods, such as TF-IDF, naïve Bayes and LDA, **conventional neural network** (**CNN**) has advantages in considering the semantic relatedness and similarity between different words, and effectively reducing a high-dimensional word embedding matrix to a low-dimensional feature matrix (Kim [13]; Zhang, & Wallace [37]). Therefore, we construct a word vector matrix M_u from user's comment set on all attractions C_u, and leverage a CNN-based feature extraction model to extract features from the user's comments, as shown in Figure 4. The dimension of word vector matrix M_u is n×dim, and it is user specific. The size of convolutional kernel K_j is h×dim.

$$q_j[i] = tanh(M_u[i : i + h - 1, *] \odot K_j + b_j), \tag{1}$$

where $i \in [1, n - h + 1]$, $j \in [1, J]$ denotes the number of convolutional kernels, and $\odot$ is Frobenius inner product. A new feature matrix $Q_u = [Q_1, Q_2, \ldots, Q_J]$ can be generated for each convolution kernel and the output vector h_u is computed by max pooling:

$$h_u = (Q_u[*, j]), j \in [1, J], \tag{2}$$

where h_u is the representation of the user u long-term feature extracted from his/her historical comment text.

At the same time, in the same way, the feature vector representation of an attraction h_p can also be obtained by forming a vector matrix M_p from the attraction's comment set C_p.

3.2.2 Short-Term Feature Extraction. Since there may be a large time span in users' travel history, it is necessary for recommender systems to focus on the most recent travel records and users' behavior. Therefore, in order to help systems more accurately capture the users' current preferences, a short-term feature extraction module using attention mechanism is proposed to improve the efficiency of recommender systems.

Specially, we compute a user's general preferences m_s based on the user's sequence of traveled attractions in time order $X_t = \{x_1, x_2, \ldots, x_t\}$, where x_i is the attraction visited by the user at

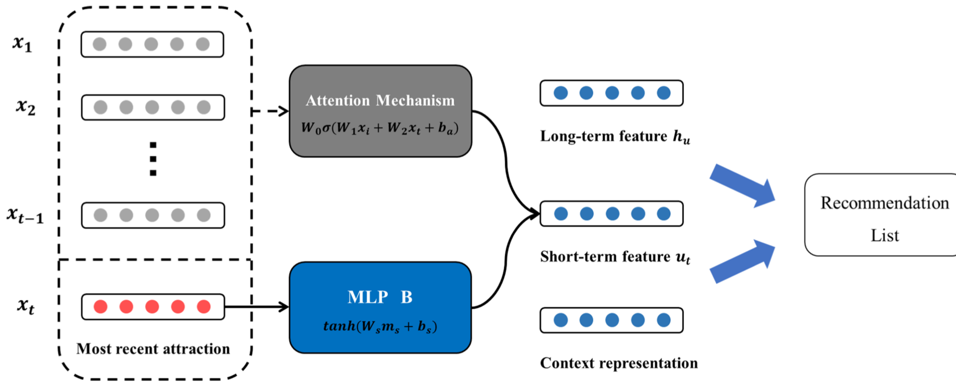

Fig. 5. Short-term feature extraction module based on attention mechanism.

time t. To extract the user's short-term preferences, the attention mechanism is used to increase weights of attractions that is similar to the user's the most recent visited attraction x_t and reduce the influence of attractions far back. The weight of each attraction is calculated as follows:

$$a_i = W_0\sigma(W_1x_i + W_2x_t + b_a), \tag{3}$$

$$\alpha_i = (a_i) = \frac{exp(a_i)}{\sum_i exp(a_i)}, \tag{4}$$

where W_0, W_1, W_2 are the weight matrixes, b_a is the bias term, and $\alpha_i (i \in [1, n])$ represents the weight of each attraction after the adjustment of attention mechanism. The general preference expression can be calculated as follows:

$$m_s = \textstyle\sum_{i=1}^{n} \alpha_i x_i, \tag{5}$$

Then, a **Multi-Layer Perceptron** (**MLP**) is leveraged to obtain the hidden vector of general preference h_s, which will be used to calculate the probability distribution of users' interests on attractions in the future. The hidden vector of the most recent attraction h_t based on x_t:

$$h_s = tanh(W_s m_s + b_s), \tag{6}$$

$$h_t = tanh(W_t x_t + b_t), \tag{7}$$

where W_s, W_t are the weight matrixes, b_s and b_t are the bias terms. Finally, the hidden vector of the general preference h_s and the hidden vector of the most recent attraction h_t are merged to generate the short-term preference expression of this user u at time t.

$$u_t = \frac{1}{2}h_s + \frac{1}{2}h_t, \tag{8}$$

The structure of short-term feature extraction module is shown in Figure 5.

4 LOCATION-AWARE USER CONTEXT LEARNING

Most of the existing travel recommender systems only focus on the preferences of users without considering the impact of real-time location, missing the valuable opportunities for systems to track the preferences shift of users. In this article, we propose a location-aware user context learning model to dynamically capture and update user' real-time context. The context learning model consists of two modules: the distance sensitive learning module and the order sensitive learning module.

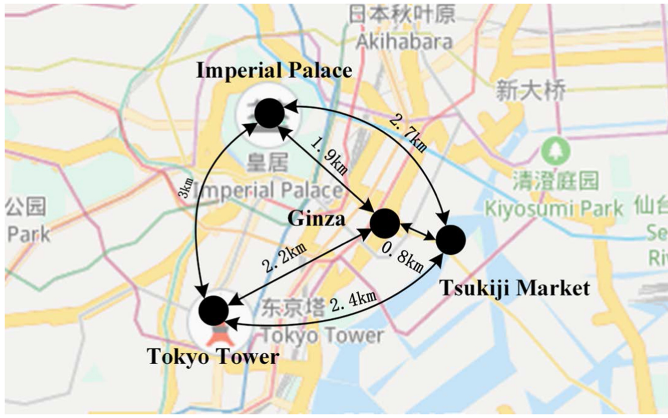

Fig. 6. Distance of attractions around Ginza.

4.1 Distance Sensitive Learning Module

The purpose of distance sensitive learning module is to integrate the information of location relationship between attractions and users, and recommend the most appropriate attractions for users. In reality, the distance of a user's current location from each attraction plays an important role in a user's selection of attractions. They could be more likely to choose a closer attraction to visit (Aydin & Telceken [4]; Sun et al. [26]). For example, if a tourist has just visited the Imperial Palace in Tokyo, Ginza would be highly likely to be his next visit site, which is closer to the Imperial Palace than Tokyo Tower and Tsukiji Market, as shown in Figure 6. Therefore, we propose a distance sensitive learning module. The principle of distance sensitive context learning module in this article is that the farther the distance between the user and the attraction, the less probability that the attraction will be selected.

First of all, we get the distance matrix $M_D = d_1, d_2, \ldots, d_n$ of all the attractions, where the vector $d_i = d_{i1}, d_{i2}, \ldots, d_{in}$ represents the distance between each attraction. This distance matrix M_D is a mask layer, which served as a look-up matrix to record pairwise attractions' distance (Wu et al. [32]). As shown in Equation (9), u_r is a one-hot vector which extracts the corresponding column from the look-up matrix M_D, and generates a distance vector d_r between the current location and other attractions.

$$d_r = M_D u_r, \tag{9}$$

Second, based on the distance vector d_r calculated in Equation (9), we set the values in vector d_r to be the reciprocals, and obtain the distance weight vector $a_d = \{\frac{1}{d_{r1}}, \frac{1}{d_{r2}}, \ldots, \frac{1}{d_{rn}}\}$.

Finally, the context representation of travel route at time t, g_loc_t, is calculated by summing up the production of a_d and attraction p.

$$\alpha_{ti} = Softmax(a_{di}) = \frac{exp(a_{di})}{\sum_i exp(a_{di})}, \tag{10}$$

$$g\_loc_t = \boldsymbol{\alpha h_p}, \tag{11}$$

where $\alpha_t i$, $i \in [1, n]$ represents the weight of each attraction on the user's location context. Since the users will update the current location information every time a recommendation request is made, the recommender system should also recalculate a new distance matrix M_D. It will lead to huge computational and time cost, and reduce the response speed of the recommender system. In order to simplify the calculation, this article assumes that users will generate new recommendation demands as soon as finishing the visit of the current attractions. Therefore, the users' real-time

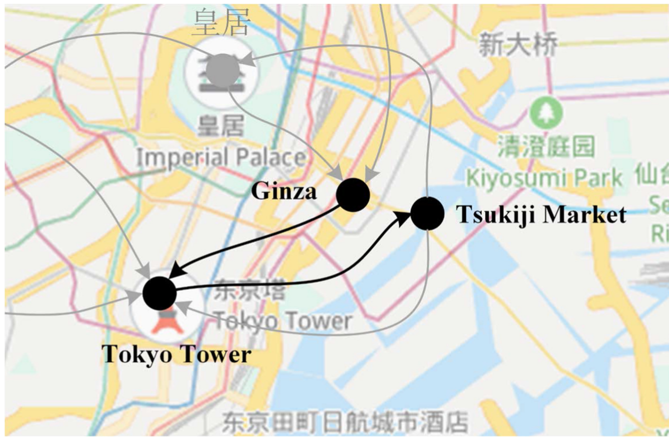

Fig. 7. Popular travel route around Ginza, Tokyo.

locations of a new recommendation request will be located near the current attractions. That way, the recommender system only needs to obtain the distance matrix of attractions M_D at the beginning and no longer needs to recalculate it in the following recommendation rounds. The algorithm of distance sensitive learning module is presented in Algorithm 1.

ALGORITHM 1: Distance Sensitive Learning Module

Input: u_r, M_D, h_p
Output: g_loc_t
1: **While** update u_r **do**
2: Select $d_r = M_D u_r$
3: Compute $a_d = \left\{\frac{1}{d_{r1}}, \frac{1}{d_{r2}}, \ldots, \frac{1}{d_{rn}}\right\}$
4: Compute $\alpha_{ti} = Softmax(a_{di}) = \frac{exp(a_{di})}{\sum_i exp(a_{di})}$
5: Generate $g\_loc_t = \boldsymbol{\alpha h_p}$
6: **Return** g_loc_t

4.2 Order Sensitive Learning Module

In addition to the distance of attractions, the travel order of attractions is believed to be another important factor when travelers make a travel plan. Often, travelers consult to other tourists' travel orders and the popularity of attractions (Song et al. [25]). For example, Tom just visited Ginza in Tokyo, and now he requests the system to recommend the next attraction for him. There are two candidate attractions: Tsukiji Market and Tokyo Tower. Tsukiji Market is far closer to Ginza than Tokyo Tower, as shown in Figure 6. However, according to the analysis of users' travel records, almost all of the tourists who have visited Ginza did not choose Tsukiji market as the next attraction, but went to Tokyo Tower instead, as shown in Figure 7.

In this article, we propose an order sensitive learning module based on the frequent sequences of visited attractions. The order sensitive context learning module sets one day as the time window and generates a Markov transition matrix (1st order) between attractions $M_V = \{v_1, v_2, \ldots, v_n\}$, where vector $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ represents the frequencies of the attraction i followed by other attractions. Similar to the algorithm of distance sensitive learning module, the directed graph matrix M_V is used as a mask layer, which generates relationship vectors between the current location and other attractions based on real-time location u_r. The relationship vector v_r indicates that the more popular the attractions are, the greater the impacts of those attractions having on users.

$$v_r = M_V u_r, \tag{12}$$

Finally, the context representation of travel route at time t g_way_t is calculated by summing up the production of v_r and attraction p.

$$\beta_{ti} = Softmax(v_{ri}) = \frac{exp(v_{ri})}{\sum_i exp(v_{ri})}, \tag{13}$$

$$g\_way_t = \boldsymbol{\beta h_p}, \tag{14}$$

where β_{ti}, $i \in [1, n]$ represents the weight of each attraction on a user's route context, $\boldsymbol{\beta} = (\beta_{t1}, \beta_{t2}, \ldots, \beta_{tn})$. Algorithm 2 shows the algorithm of order sensitive learning module.

ALGORITHM 2: Order Sensitive Learning Module

Input: u_r, M_V, h_p
Output: g_way_t
1: **While** update u_r **do**
2: Select $v_r = M_V u_r$
3: Compute $\beta_{ti} = Softmax(v_{ri}) = \frac{exp(v_{ri})}{\sum_i exp(v_{ri})}$
4: Generate $g\_way_t = \boldsymbol{\beta h_p}$
5: **Return** g_way_t

In summary, the purpose of the location-aware context learning model is to help the recommender system compute the expression of the users' real-time context by considering the travel orders of other tourists as well as the influence of hot travel routes.

4.3 User Context Update Mechanism

After obtaining the user's location context g_loc_t and route context g_way_t, we calculated the comprehensive context expression s_t at time t by integrating g_loc_t, g_way_t, the user's short-term feature vector u_t, and his/her long-term feature expression h_u.

$$s_t = \lambda(g\_loc_t + g\_way_t + u_t) + (1 - \lambda)h_u, \tag{15}$$

where λ is a hyperparameter to control the impact of two parts. At the same time, except for the user's long-term feature h_u, the location-aware context learning module and the short-term feature extraction module are both dynamically updated by the user's real-time travel condition. In order to improve the learning efficiency of our model, we simplified the updated location information as the position of the user's last visited attraction, in which the location-aware context learning and short-term feature extraction module shared the same updated information. As a result, we designed a user context update mechanism to help the system dynamically track the condition of the user. The algorithm is defined as Algorithm 3.

5 TRAVEL REINFORCEMENT RECOMMENDATION BASED ON LOCATION-AWARE CONTEXT EXTRACTION

In order to dynamically capture, the preferences from real-time interactions between users and recommender systems, a novel recommender framework called TRACE is proposed in this section.

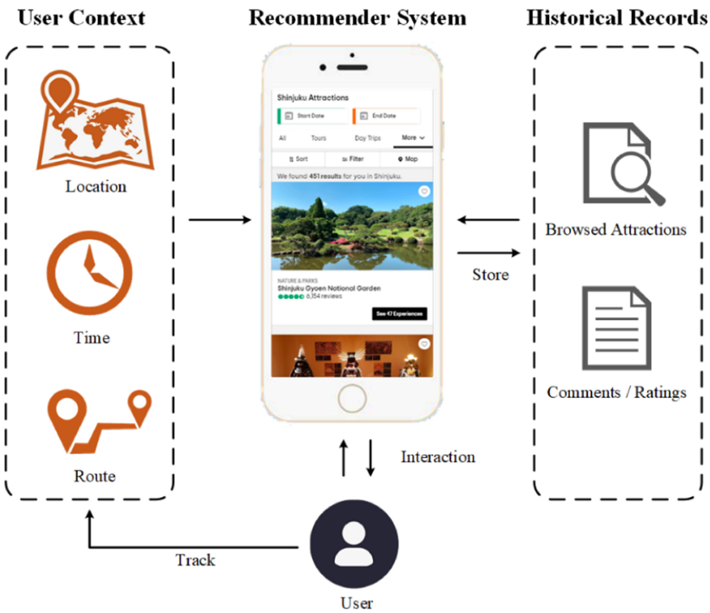

Fig. 8. Context-aware recommender system.

ALGORITHM 3: User Context Update Mechanism

Input: u_r, M_D, M_V, h_p, X_{t-1}, h_u
Output: s_t
While update user real-time location u_r **do**
  Reset x_t
  Remove x_1 from X_{t-1}
  Add x_t to the bottom of $X_{t-1} \rightarrow X_t$
  Compute u_t according to Equation (3)–(8)
  Compute g_loc_t and g_way_t according to Equation (9)–(14)
  Generate $s_t = \lambda(g\_loc_t + g\_way_t + u_t) + (1 - \lambda)h_u$
  Return s_t

5.1 TRACE Framework

As mentioned above, to model the users' dynamic demands, we aim at constructing a context-aware recommender system, which not only extracts user preferences from historical records, but also obtain the real-time states from user context (shown in Figure 8). Therefore, the problem definition of this study is given a user set C_u, an item set C_p, a distance matrix for attractions M_D, a transition matrix for attractions M_V, and users' real-time location u_r as the input, the recommender system will generate a list of recommended attractions A upon users' requests. However, most of the existing travel recommender systems only compute the correlation or similarity between the features of attractions and preferences of users, which fails to consider the current context of users and track the shift of preferences. Therefore, how to integrate the representation of real-time user state and user preference together is a big challenge in this research.

Inspired by an existing algorithm Actor-Critic framework (Konda, & Tsitsiklis [14]; Lillicrap et al. [17]), which can automatically learn the optimal strategies from users' real-time feedbacks, we propose a framework of TRACE to address the challenge of preference shifting in the process

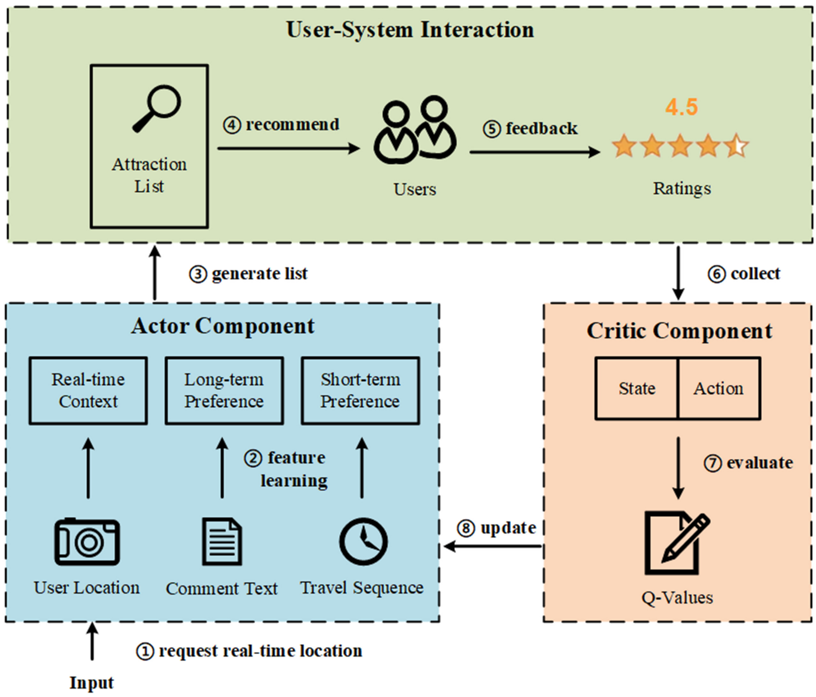

Fig. 9. Framework of TRACE, an Actor-Critic based recommender system.

of user-system interactions. In this article, the Actor component of strategy generating is replaced by an attraction recommendation module to find out the most suitable attractions based on the users' real-time context. The Critic component is used to calculate users' feedbacks on the recommendation list and evaluate the effect of the Actor component. At last, the feedback is defined as users' ratings on recommended attractions and will be treated as reward to update the parameters in both the Actor component and the Critic component, respectively.

Furthermore, to track the preference shifting in the process of user-system interactions, we model the recommendation process using the MDP and then deploy the recommender system in a RL framework. As shown in Figure 9, our TRACE model consists of two components: the Actor component and the Critic component. The real-time state of a user in the framework is defined as the user's comprehensive context expression s_t, which includes the user's preferences and location. The action options in the critic component is defined as the vector representation of attractions h_p, and the reward is defined as the user's selection of attractions and the ratings of the attractions r.

5.2 Actor Component

The function of the Actor component is to obtain the candidate action set from all attraction set based on the users' current states and reduce the computational complexity of the Critic component. In this article, we define the state of a user at time t as the user's comprehensive context representation s_t. As shown in Figure 9, the Actor component can be divided into three parts: CNN-based comments extraction model for user long-term preference, attention-based travel sequence learning model for user short-term preference, and location-aware context learning model for user real-time state.

Due to the dynamic demands from tourists, the travel recommender system needs to take the current state and short-term preference of users into account. Therefore, in order to recommend a suitable attraction to users, we calculate the similarity between the user's comprehensive state expression and the attraction vector expression. The similarity between attractions and the users' context is defined as follows:

$$score_t = \sigma(s_t \cdot h_p), \tag{16}$$

where $\sigma(\cdot)$ denotes the sigmoid function. Finally, we obtain the probability distribution z_t of the user's choice on attractions at time t by using the Softmax function to normalize the similarity score between users and attractions.

$$z_t = Softmax(score_t), \tag{17}$$

After calculating the probability of user's choice z_t on all attractions p, we select top-K attractions as a candidate action subset $A = \{a_t^1, \ldots, a_t^k\}$ and generate a recommendation list for the user. In this article, the Actor component in fact serves as a travel recommendation engine. After the users give feedback to the recommendation list generated by the Actor component, the Critic component will evaluate the rewards of all possible recommendations. Therefore, the training of the Actor component relies on the rewards value calculated by the Critic component. The loss function of the Actor component is defined as follows:

$$L_{actor} = -E_{s_t, a_t}[Q(s_t, a_t; W_c)], \tag{18}$$

where E denotes the expectation of Q-value function, $Q(s_t, a_t; W_c)$ denotes the rewards value obtained from the Critic component. The function of L_{actor} is to maximize rewards value $Q(s_t, a_t; W_c)$, so we set L_{actor} to the negative value.

5.3 Critic Component

The function of the Critic component is to evaluate the recommendation list generated by the Actor component by calculating the reward values of the user's actions. The key of the Critic component is to obtain the reward function $Q(s, a)$ for the Actor component based on the Behrman equations, and optimize the parameters in the Actor component to improve the recommendation efficiency. The reward function $Q(s, a)$ based on Bellman equations aims at calculating the maximum expected rewards by an optimal policy and is defined as follows:

$$Q(s_t, a_t) = E_{s_{t+1}}[r_t + \gamma Q(s_{t+1}, a_{t+1}) | s_t, a_t], \tag{19}$$

where r_t is the reward of the set of recommended attractions on users at current t time, γ is the reward discount for the future potential rewards $Q(s_{t+1}, a_{t+1})$. However, to compute the maximum future potential reward $Q(s_{t+1}, a_{t+1})$, we must at first calculate all the possible states s_{t+1} and candidate action a_{t+1} in the future, which has computational complexity of $O(n^2)$, a huge cost of calculation. However, since the Actor component has generated a list of candidate attractions, it provides a smaller-scale action subset for the Critic component and reduces the computational complexity to $O(n)$. As reward function $Q(s, a)$ is non-linear, we will use a neural network to calculate the loss function, defined as

$$L_{critic} = E_{s_t, a_t, r_t}\left[(y_t - Q(s_t, a_t; W_c))^2\right], \tag{20}$$

where y_t is the target network and shares the weights with the main network $Q(s_t, a_t; W_c)$, which can also be defined as

$$y_t = r_t + \gamma Q(s_{t+1}, a_{t+1}; W_c), \tag{21}$$

However, due to the user's real-time context s_t is updated dynamically, the reward function $Q(s, a)$ based on the Behrman equation will be extremely unstable and hard to converge in training. Therefore, to solve the problem, we will leverage the method of separate target network in DQN to stabilize the training process by fixing weights W_c' of y_t. The loss function of the Critic component is therefore defined as

$$L_{critic} = E_{s_t, a_t, r_t}\left[(r_t + \gamma Q(s_{t+1}, a_{t+1}; W_c') - Q(s_t, a_t; W_c))^2\right]. \tag{22}$$

5.4 Reward Update Mechanism

In our TRACE model, the mechanism of tracking users' feedback is the key of user-system interactions. After the recommender system generating a recommended list, the user will give feedback to the list, including whether to click or rate. In this article, the reward of click is defined as $r_{click} = \{0, 1\}$, and the reward of each rating is defined as $r_{rate} = \{1, 2, 3, 4, 5\}$. The total reward of users' feedback r_{click} and r_{rate} is defined as Equation (23).

$$r = r_{click} \cdot r_{rate}, \tag{23}$$

The system will store the users' feedbacks information as a tuple $((s_t, a_t) \rightarrow r)$. At last, the TRACE model will update the parameters of the Actor and Critic components based on the total reward r. The training mechanism of the TRACE model is shown in Algorithm 4.

ALGORITHM 4: Travel Reinforcement Recommendation Based on Location-Aware Context

Input: s_t, h_p
Output: recommendation list A, action-value Q
1: **While** update Q **do**
2: Compute $score_t = \sigma(s_t \cdot h_p)$
3: Compute $z_t = Softmax(score_t)$
4: Select top K actions as $A = \{a_t^1, \ldots, a_t^k\}$
5: Observe reward $r = r_{click} \cdot r_{rate}$
6: **if** reward $r > 0$ **do**
7: Update $s_t \rightarrow s_{t+1}$
8: **end if**
9: Compute $Q(s_t, a_t) = E_{s_{t+1}}[r_t + \gamma Q(s_{t+1}, a_{t+1}) | s_t, a_t]$
10: **Return** A, Q

6 EXPERIMENTS

In this section, we will introduce the evaluation method of our proposed TRACE model through online and offline experiments.

6.1 Dataset

TripAdvisor is one of the largest online travel platforms in the world with more than 7 million attractions and hundreds of millions of user records around the world. Therefore, due to the abundant data, TripAdvisor is an ideal platform for analyzing users' travel preferences. We crawled the travel data related to Tokyo from the online platform TripAdvisor, which contained users and attractions information between January 1, 2017 and March 31, 2019. The basic statistics of the crawled dataset is shown in Table 2.

Table 2. Statistics of the TripAvisor Dataset

Description	Amount
Number of TripAdvisor users	20,473
Number of registered user nationalities	53
Number of attractions in Tokyo	5,396
Number of user comments	437,310
Average number of words per comment	452
Average number of words per attraction]	36,632
Average number of words written by a user	9,655
Average number of attractions visited by a user	21

6.2 Experiment Design

6.2.1 Offline Experiment. We first conducted an offline experiment on the dataset collected from TripAdvisor and evaluated our proposed TRACE. Among the total 437,310 pieces of user comments, the training set and test set are separated by time order in the proportion of 4:1. The predictors of TRACE model includes users' comment vector matrix M_u, users' travel sequence X_t, users' real-time locations u_r, attractions' comment vector matrix M_p, attractions' distance matrix M_D, and attractions' directed graph matrix M_V. The target of the model is the normalized similarity score between users and attractions z_t. In this article, we set the dimension of word embedding 200 and the window size of the convolutional kernel 3×200. We trained the CNN-based feature extraction model for 200 epochs. Moreover, the K value in the recommendation list is set to be 5 or 10 in this article, which means the system will recommend the top 5 or 10 best matched attractions to users.

6.2.2 Online Experiment. In the real-time online experiment, we deployed the recommender system on the website of our laboratory, and invited the faculty and students as the experimental participants. The opening page of our online recommender system is shown in Figure 10 below. Users were able to choose whether to click or give ratings (range from 0 to 5) on the attractions in the radio boxes on the right. The recommended list would update if the user was not satisfied with any attraction in it.

6.3 Baselines

To investigate the performance of TRACE, the following seven baseline models, including conventional machine learning models and deep RL models, are selected to compare with TRACE in both the offline and online experiments.

- **FM:** is Factorization Machines, which combines advantages of **Matrix Factorization** (**MF**) and **Support Vector Machines** (**SVM**), and is a state-of-the-art context-aware recommendation method [21, 22]. In this article, the users' rating matrices on attractions are set as input.
- **LR:** is the Logistic Regression model, which is a linear regression method and often used to deal with the classification tasks. In this article, the logistic regression model takes the user's historical records as input.
- **NCF:** is Neural Collaborative Filtering, which is another state-of-the-art recommendation method based on deep neural network to capture user-item interaction feature [9], and takes the user's historical travel records as input.

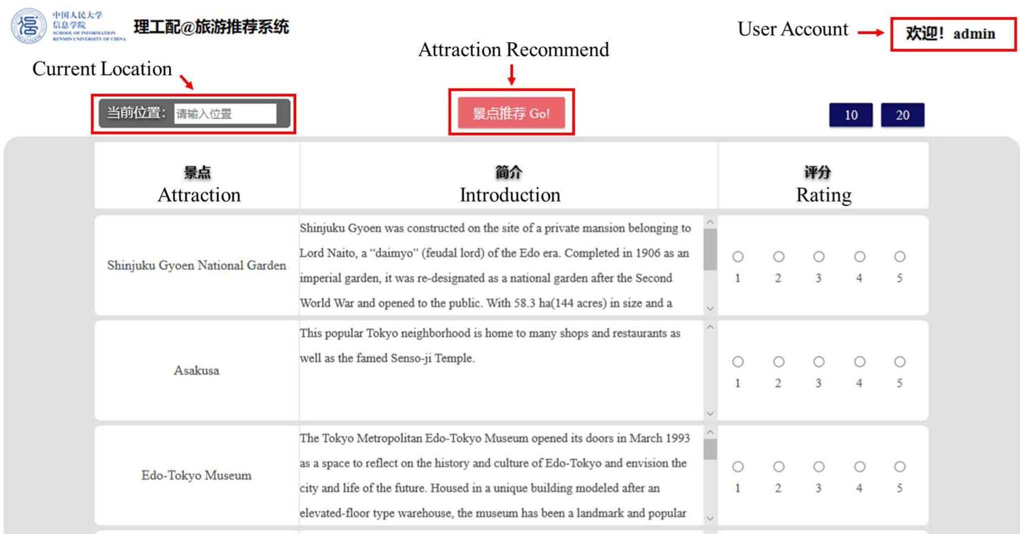

Fig. 10. Interface of TRACE.

- **RNNs:** is the Recurrent Neural Network, which is widely used in dealing with time-series data [28]. In this article, we set the users' historical travel record X_t as the input of the RNN model.
- **DQN:** is Deep Q Network, which combines the advantages of deep learning and Q-learning. It calculates the value of current iteration based on the state of users and is a state-of-the-art RL-based approach in recommendation problem [20, 38, 39].
- **w/o Attention:** is a simplified TRACE model without short-term feature extraction module. The w/o Attention model only uses the long-term feature extraction module and location-aware context learning mechanism to do the travel recommendations, and excludes the user's short-term preferences.
- **w/o Location:** is a simplified TRACE model without context learning mechanism using user's real-time location. The w/o Location model makes the recommendations only based on long-term and short-term feature extraction module and removes the real-time location information of users.

The parameter settings of the best performance are determined by gradient optimization. The detailed parameter setting of each model is shown in Table 3 below. It is noteworthy that the w/o Attention, w/o Location, and the TRACE model have the same parameter setting.

6.4 Evaluation Metrics

In this article, we use the metrics of **Average Recall** (**Ave-Recall**), **Mean Reciprocal Rank** (**MRR**), and **Average Ratings** (**Ave-Rating**) to evaluate the performance of each model. These metrics are widely used in the recommender research community.

- **Ave-Recall@K:** Represents the proportion of the number of correct recommendations from the top-K attractions of all the recommendations. Where n_{hit} donates the number of correct attractions recommended in the list, N donates the number of total recommendations.

$$Ave - Recall@K = \frac{n_{hit}}{N}, \tag{24}$$

Table 3. Parameter Setting

Model	Parameter	Setting
FM	Number of dimensions of FM	200
NCF	Number of hidden layers	128
RNNs	Number of units of RNN	50
DQN	Reward discount γ	0.4
	Kernel size of CNN	3
TRACE	Reward discount γ	0.4
	Impact factor λ	0.5

where n_{hit} donates the number of correct attractions recommended in the list, N donates the number of total recommendations.

— **MRR@K:** The value of it will be set to 0 if there is no matched attraction in the top-K recommendation list. MRR@K is used for all the experiments and is defined as:

$$MRR@K = \frac{1}{N} \sum \frac{1}{Rank(p_{hit})} \tag{25}$$

where p_{hit} represents the correct attractions recommended in the list. The higher the value of MRR@K, the higher quality of the ranking in the recommendation list.

— **Ave-Rating@K:** In the online real-time recommendation experiment, users offer ratings on the recommended attractions. Therefore, we calculate the Ave-Rating of users to measure the users' satisfaction on recommendation results. Ave-Rating@K represents the average ratings of the users on the top-K recommended attractions. The value of Ave-Rating@K will be set to 0 if the users does not select any attractions from the current recommendation list.

$$Ave - Rating@K = \frac{1}{n_{hit}} \sum r_{rate}, \tag{26}$$

where n_{hit} represents the number of correct attractions recommended in the list. From Equation (26), the higher the value of Ave-Rating@K, the greater the user satisfaction in recommendation service, which indicates the better performance of the recommender system.

6.5 Results

6.5.1 Offline Experiment. The Ave-Recall@K and MRR@K results of the offline experiment are shown in Table 4.

Our proposed TRACE model has the best performance in both Ave-Recall and MRR compared with the baseline models. At the same time, by further investigating the experimental results in Table 4, we can draw a conclusion that RNNs model based on a deep learning algorithm has a relatively higher Ave-Recall and MRR among all baseline models. The results show that the user travel data do have the time sequence nature, which affects users' selections on attractions. Moreover, the TRACE model has a better performance compared with w/o Location model on Ave-Recall. It indicates that the context of real-time location has impacts on the users' selections of attractions. The real-time location-aware context learning mechanism can better target users' interests. In addition, by comparing with the TRACE model and w/o Attention model, it can be found that w/o Location model has the lowest MRR. It demonstrates that the users' real-time location can not only improve the prediction accuracy, but also help the system optimize the ranking of the recommendation list.

Table 4. Ave-Recall and MRR while K = 5 & 10 (%)

	Ave-Recall@5	MRR@5	Ave-Recall@10	MRR@10
FM	44.31	27.14	54.12	28.45
LR	39.22	19.61	47.06	20.67
NCF	53.14	24.35	60.59	24.96
RNNs	54.31	27.67	63.53	29.80
DQN	45.10	17.65	54.90	23.91
w/o Attention	45.29	39.33	47.25	39.70
w/o Location	54.90	32.94	76.47	34.83
TRACE	**64.57**	**43.39**	**78.84**	**45.08**

*Denotes the difference is statistically significant compared to other models under paired t-test with $p \leq 0.05'$ below the table.

Table 5. Ave-Rating and MRR while K = 10 & 20

	Ave-Rating@10	MRR@10	Ave-Rating@20	MRR@20
FM	3.55	69.40%	3.58	70.55%
LR	3.33	45.83%	3.50	51.11%
NCF	3.70	51.67%	3.75	54.17%
RNNs	3.88	53.32%	3.91	62.22%
DQN	3.53	69.47%	3.56	66.65%
w/o Attention	3.67	78.92%	3.61	74.84%
w/o Location	3.83	68.26%	3.89	61.10%
TRACE	**4.08**	**86.00%**	**3.97**	**83.65%**

*Denotes the difference is statistically significant compared to other models under paired t-test with $p \leq 0.05'$ below the table.

6.5.2 Online Experiment. In the online recommendation experiment, we noticed that most of the users selected at least one attraction in the recommendation list and gave the ratings. Therefore, the metric of Ave-Recall is not suitable for evaluating the online experiment. We will leverage the Ave-Rating of users and MRR to evaluate the performance of recommendation results. Table 5 illustrates the Ave-Rating and MRR of each model in the online real-time experiment, in which K is set to be 10 and 20, respectively.

From Table 5, the proposed TRACE model outperformed all the baseline models in both Ave-Rating and MRR. Furthermore, the following conclusions can be also drawn from the online real-time recommendation experiment: First, compared with other baselines, NCF model and RNN model based on a deep learning algorithm have a lower MRR in online recommendation experiments. As the NCF model and RNN model only extract users' long-term preferences from historical records, these two methods have disadvantages in tracking users' real-time preference and can easily recommend repeated or similar attractions to users. Second, the Ave-Rating of w/o Attention model is much lower than the TRACE model. It demonstrates that users will be significantly affected by their most recent preferences when making the decision. At last, the MRR of w/o Location model is much lower than TRACE model. It confirms that the context of users' real-time location has a significant impact on the performance of travel recommendation.

To further investigate whether the context learning mechanism based on real-time location information optimizes the distance of recommended attractions to users, we observe the distance

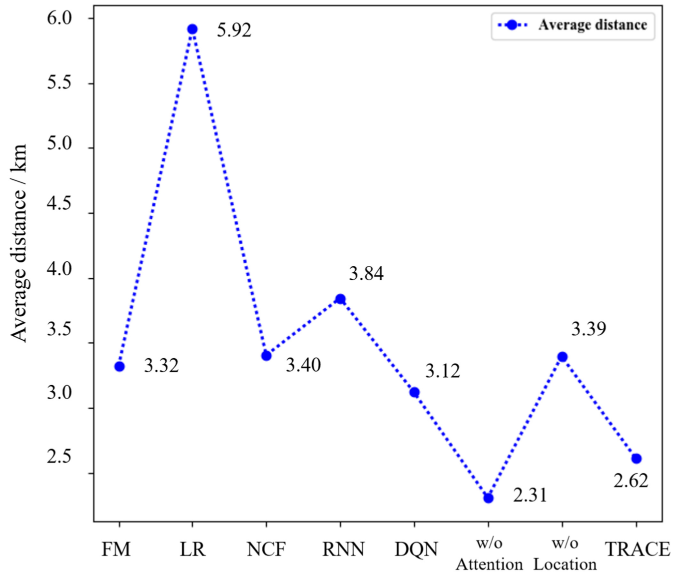

Fig. 11. Average distance of each model for K = 10.

between the recommended attractions and the users' current locations with a recommendation list of 10 attractions. Figure 11 shows the average distance between the recommended attractions of each model and the users' locations. As shown in Figure 11, the w/o Attention model and the TRACE model have an average distance less than 3 km, which is significantly smaller than other models. It shows that the user context learning mechanism based on real-time location information effectively addresses the relationship between attractions and users.

Above all, the offline simulation recommendation and online real-time recommendation experiments in this section shows that our proposed TRACE model can significantly improve the effectiveness of the travel recommendation system, since it has the best performance in Ave-Recall, MRR, and Ave-Rating.

7 CONCLUSION

In this article, we first analyzed the crawled dataset from the TripAdvisor platform, and found that (1) users' preferences of attractions have shifted over time, and (2) users' current location would affect the decisions on attractions to visit. According to the analysis of data, this article proposed and implemented a recommender system called TRACE. The contributions are (1) to address user preferences shifting, a short-term preferences learning model was added, which strengthened the weights of users' most recent records by using an attention mechanism; (2) to consider the impact of distance relationship between attractions and the relationship of travel order, the constructed location-aware user context learning mechanism based on real-time location information was incorporated; and (3) to track the short-term preferences in human–computer interaction and evaluate the long-term benefits in attraction recommendation, the Actor-Critic framework is leveraged. The offline and online experiments were conducted on real-world data and six effective baseline algorithms were implemented to compared with TRACE model. The results indicate that our proposed TRACE model achieves the best performance in average recall, mean reciprocal ranking and average score of attractions, which proves the effectiveness of the TRACE model. For the future work, on one hand, we will focus on optimizing the representation of user's context. In this article, while learning the user's context of real-time location, we only consider the location relationships between attractions and ignore the time of tour. Therefore, in the future, we will further consider the time cost, including the commuting time between attractions and the dwell time in each, which may optimize the performance of recommendation systems. On the other hand, in the actual model training process, we found that the model cannot optimize the weight parameters and obtain the preferences of users due to a number of users lacking feedback information. In order to solve this problem, we plan to add a module generating part of user feedback information to enrich the training data set and improve the recommendation effect of the model.

REFERENCES

[1] Mohammed F. Alhamid, Majdi Rawashdeh, Hussein Al Osman, M. Shamim Hossain, and Abdulmotaleb El Saddik. 2015. Towards context-sensitive collaborative media recommender system. *Multimedia Tools and Applications* 74, 24 (2015), 11399–11428.

[2] Mohammed F. Alhamid, Majdi Rawashdeh, Haiwei Dong, M. Anwar Hossain, and Abdulmotaleb El Saddik. 2016. Exploring latent preferences for context-aware personalized recommendation systems. *IEEE Transactions on Human-Machine Systems* 46, 4 (2016), 615–623.

[3] Mohammad Aliannejadi and Fabio Crestani. 2018. Personalized context-aware point of interest recommendation. *ACM Transactions on Information Systems* 36, 4 (2018), 1–28.

[4] Ahmet Aydin and Sedat Telceken. 2015. Artificial intelligence aided recommendation based mobile trip planner for Eskisehir city. In *Proceedings of the 2015 IEEE 10th Conference on Industrial Electronics and Applications*. IEEE, 1650–1654.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *the 3rd International Conference on Learning Representations. San Diego, CA, May 7-9, 2015*. ICLR.

[6] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 335–344.

[7] Yan-Ying Chen, An-Jung Cheng, and Winston H. Hsu. 2013. Travel recommendation by mining people attributes and travel group types from community-contributed photos. *IEEE Transactions on Multimedia* 15, 6 (2013), 1283–1295.

[8] Yue Guo, Stuart J. Barnes, and Qiong Jia. 2017. Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation. *Tourism Management* 59 (2017), 467–483.

[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.

[10] Liang Hong, Lei Zou, Cheng Zeng, Luming Zhang, Jian Wang, and Jilei Tian. 2015. Context-aware recommendation using role-based trust network. *ACM Transactions on Knowledge Discovery from Data* 10, 2 (2015), 1–25.

[11] Yu Huang, Josh Jia-Ching Ying, Philip S. Yu, and Vincent S. Tseng. 2020. Dynamic graph mining for multi-weight multi-destination route planning with deadlines constraints. *ACM Transactions on Knowledge Discovery from Data* 15, 1 (2020), 1–32.

[12] Shuhui Jiang, Xueming Qian, Tao Mei, and Yun Fu. 2016. Personalized travel sequence recommendation on multi-source big social media. *IEEE Transactions on Big Data* 2, 1 (2016), 43–56.

[13] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. ACL, 1746–1751.

[14] Vijay R. Konda and John N. Tsitsiklis. 2000. Actor-critic algorithms. In *Proceedings of the Advances in Neural Information Processing Systems*. Citeseer, 1008–1014.

[15] Takeshi Kurashima, Tomoharu Iwata, Go Irie, and Ko Fujimura. 2013. Travel route recommendation using geotagged photos. *Knowledge and Information Systems* 37, 1 (2013), 37–60.

[16] Yu Lei and Wenjie Li. 2019. Interactive recommendation with user-specific deep reinforcement learning. *ACM Transactions on Knowledge Discovery from Data* 13, 6 (2019), 1–15.

[17] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *the 4th International Conference on Learning Representations. Caribe Hilton, San Juan, Puerto Rico, May 2-4, 2016*. ICLR.

[18] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1831–1839.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*. PMLR, 3111–3119.

[20] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[21] Atsuyoshi Nakamura. 2015. A ucb-like strategy of collaborative filtering. In *Proceedings of the Asian Conference on Machine Learning*. PMLR, 315–329.

[22] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[23] Lin Shi, Feiyu Lin, Tianchu Yang, Jun Qi, Wei Ma, and Shoukun Xu. 2014. Context-based ontology-driven recommendation strategies for tourism in ubiquitous computing. *Wireless Personal Communications* 76, 4 (2014), 731–745.

[24] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, and Alan Hanjalic. 2014. Cars2: Learning context-aware representations for context-aware recommendations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 291–300.

[25] XY Song, HF Xu, and HL Sun. 2013. Short-term experience route search based on check-in data. *Chinese Journal of Computers* 36, 8 (2013), 1693–1703.

[26] Yeran Sun, Hongchao Fan, Mohamed Bakillah, and Alexander Zipf. 2015. Road-based travel recommendation using geo-tagged images. *Computers, Environment, and Urban Systems* 53 (2015), 110–122.

[27] Moshe Unger, Alexander Tuzhilin, and Amit Livne. 2020. Context-aware recommendations based on deep learning frameworks. *ACM Transactions on Management Information Systems* 11, 2 (2020), 1–15.

[28] Lu Wang, Wei Zhang, Xiaofeng He, and Hongyuan Zha. 2018. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2447–2456.

[29] Xuejian Wang, Lantao Yu, Kan Ren, Guanyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2051–2059.

[30] Zhang Wei, Han Linyu, Zhang Dianlei, Ren Pengjie, Ma Jun, and Chen Zhumin. 2017. GeoPMF: A distance-aware tour recommendation model. *Journal of Computer Research and Development* 54, 2 (2017), 405.

[31] Libing Wu, Cong Quan, Chenliang Li, Qian Wang, Bolong Zheng, and Xiangyang Luo. 2019. A context-aware user-item representation learning for item recommendation. *ACM Transactions on Information Systems* 37, 2 (2019), 1–29.

[32] Yuexin Wu, Yiming Yang, Hiroshi Nishiura, and Masaya Saitoh. 2018. Deep learning for epidemiological predictions. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1085–1088.

[33] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*. PMLR, 2048–2057.

[34] Ying Xu, Tao Hu, and Ying Li. 2016. A travel route recommendation algorithm with personal preference. In *Proceedings of the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*. IEEE, 390–396.

[35] Toshihiko Yamasaki, A. Gallagher, and Tsuhan Chen. 2013. Geotag-based travel route recommendation featuring seasonal and temporal popularity. In *Proceedings of the 9th International Conference on Information, Communications & Signal Processing*. IEEE, 1–4.

[36] Chunqiu Zeng, Qing Wang, Shekoofeh Mokhtari, and Tao Li. 2016. Online context-aware recommendation with time varying multi-armed bandit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2025–2034.

[37] Ye Zhang and Byron C. Wallace. 2017. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*. AFNLP, 253–263.

[38] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 95–103.

[39] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*. 167–176.

[40] Ningnan Zhou, Wayne Xin Zhao, Xiao Zhang, Ji-Rong Wen, and Shan Wang. 2016. A general multi-context embedding model for mining human trajectory data. *IEEE Transactions on Knowledge and Data Engineering* 28, 8 (2016), 1945–1958.

Received February 2021; revised June 2021; accepted September 2021