Knowledge-Based Systems 233 (2021) 107521

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A multi-task learning approach for improving travel recommendation with keywords generation

Lei Chen [a], Jie Cao [a,b,*], Guixiang Zhu [b], Youquan Wang [b], Weichao Liang [a]

[a] College of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
[b] Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing, China

ARTICLE INFO

Article history:
Received 16 January 2021
Received in revised form 17 September 2021
Accepted 18 September 2021
Available online 21 September 2021

Keywords:
Recommendation system
Travel recommendation
Keywords generation
Deep learning
Multi-task learning

ABSTRACT

Travel recommendation is very critical to helping users quickly find products or services that they are interested in. The key to travel recommender systems is learning user shopping intentions, which are expressed through various supervision signals, such as the clicked products and their titles. Existing travel recommendation methods commonly infer user intentions from click behaviors on travel products. However, remarkable keywords in the product title, such as departure, destination, travel time, hotel, and transportation are paid less attention. To this end, we hypothesize that modeling click sequences and product keywords in title jointly would result in a more holistic representation of a product and towards more accurate recommendations. Thus, we propose a TRKG (short for **T**ravel **R**ecommendation with **K**eywords **G**eneration) model, which fulfills the travel recommendation and keywords generation tasks simultaneously. To generate explainable outputs, unlike most previous approaches that regard the product title as a hidden feature vector, TRKG regards keywords in the product title as an additional supervision signal. Meanwhile, TRKG integrates the long-term and short-term user preferences in the travel recommendation component and the keywords generation component. To evaluate the proposed model, we constructed datasets from a large tourism e-commerce website in China. Extensive experiments demonstrate that the proposed method yields significant improvements over state-of-the-art methods.

1. Introduction

Tourism is a popular leisure activity undertaken by more than 1.18 billion international tourists per annum [1]. Aware of this enormous business opportunity, many online travel agencies (OTA), such as Expedia, Booking.com, Travelocity, and Tuniu, have emerged to provide travel services. Tourists can use such a website or mobile app to search and book travel. However, with the explosion of travel products and services, tourists are overwhelmed by tremendous content. One working solution is travel recommendation [2], which facilitates tourists learning and purchasing travel products. In the travel recommender system, timely identifying users' intentions and recommending personalized products is still a challenge as users' purchase intentions are generally hidden in the noisy users' behaviors.

In a purchase session, users usually have specific click intentions. They browse the product titles displayed on the website homepage and decide whether to click them and go further into the detail page. As users' first impressions of travel products, the product title attracts users' interests and arouses users' desires to click and purchase the product. Meanwhile, different product titles have remarkable features in various aspects, such as departure, destination, travel time, hotel, and transportation. One or more interesting aspects may attract users to click. Thus, tracking users to understand their fine-grained click intentions via clicked product titles can help users make an informed decision and improve the likelihood of purchase. Recent work on this research tries to utilize the topic model or neural network-based model to embed the item title as the item feature vector. For instance, CTR [3] fuses Probabilistic Matrix Factorization (PMF) and Latent Dirichlet Allocation (LDA) into a unified framework and uses LDA to model the item textual content. CDL [4] proposes a hierarchical Bayesian model that jointly utilizes deep representation learning for item textual content and collaborative filtering for the user–item interactions. CRAE [5] replaces stacked denoising auto-encoder with recurrent auto-encoder to learn the hidden feature vectors from item text. However, these research works commonly regard textual content as a hidden feature vector rather than an explainable generative output. Moreover, they only focus on users' long-term static preferences while ignoring their short-term transactional patterns.

* Corresponding author at: Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing, China.
E-mail addresses: chenleinjust@gmail.com (L. Chen), caojie690929@163.com (J. Cao).

https://doi.org/10.1016/j.knosys.2021.107521
0950-7051/© 2021 Elsevier B.V. All rights reserved.

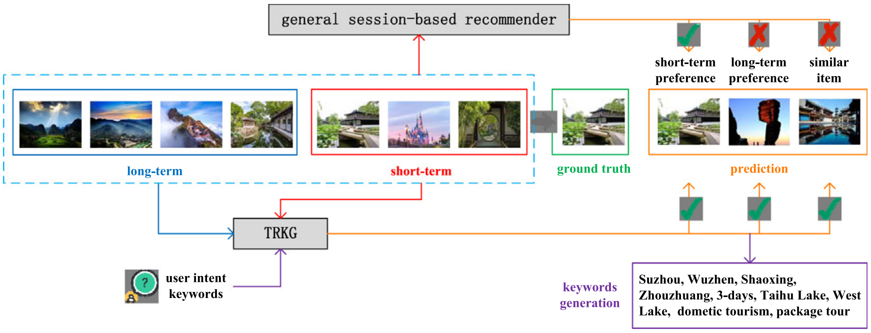


Fig. 1. The difference between our proposed TRKG model and the general session-based recommendation approach.

Many session-based methods have been designed for recommendation scenarios to learn user transactional behavior patterns and capture user preference shift. Unfortunately, most existing session-based approaches, such as GRURec-TopK [6], STAMP [7], RCNN [8], and SR-GNN [9], only consider the current session and treat it as a short sequence. They generally employ RNN or its variants with an attention mechanism to model short-term interests but ignore the time intervals between each interaction and long-term user interests. Furthermore, as shown in Fig. 1, most existing works only take the last behavior (*i.e.*, click, buy) in the current session as the supervision signal (*i.e.*, ground truth). However, many relevant items that meet the need of users are neglected. This is because different items with different identities may share the same intentions. In addition, in the absence of supervision of intention, noisy or irrelevant parts in the current session may degrade the recommendation accuracy. Since keywords in the product title can explicitly reveal users' intentions, we hypothesize that keywords could be helpful to item recommendation and consider keywords as additional supervision to address the above issues.

In this paper, we propose a travel recommendation model with keywords generation (TRKG). The TRKG model contains two critical tasks: travel recommendation and keywords generation. To be specific, TRKG learns user embedding from the long-term behavior sequence and the short-term behavior sequence with time intervals. In addition, TRKG also learns another user embedding from the clicked product titles to capture their shopping intentions. TRKG combines both user embeddings into a unified user representation and trains it using two supervision signals, *i.e.*, the purchased item and its keywords in the title. To sum up, our main contributions can be summarized as follows:

- Unlike content-based recommendations that regard textual content as a hidden feature vector, we treat keywords in the product title as an additional supervision signal in travel recommendation and generate explainable output.
- For both travel recommendation task and keywords generation task, we learn relevant long-term user preferences from historical sessions, learn short-term user preferences from the current session, and fuse these two types of user preference. Meanwhile, we explicitly model the timestamps of interactions in the short-term user preferences.
- TRKG is evaluated on real-world datasets. The results demonstrate that TRKG significantly outperforms the state-of-the-art approaches on travel recommendation.

The remainder of this paper is organized as follows. Section 2 reviews the work related to tourism-oriented recommendation and recommendation with textual information. In Section 3, we begin by describing the background and the problem that we study in this article, and then propose the TRKG model to recommend travel products with keywords generation. We exhibit the experimental results in Section 4. Section 5 discusses the interests and limitations of the proposed contribution in relation to related work. Finally, in Section 6, conclusions and perspectives for future research are presented.

2. Related work

This section surveys the relevant literature in two streams of research: tourism-oriented recommendations and recommendations with textual information.

2.1. Tourism-oriented recommendations

Usually, the tourism-oriented recommendations can roughly be categorized into two types. The first one is the POI recommendation, which predicts the next POI according to the user's latent behavior pattern [10–13]. Since the sequence of POI visits can be considered as a session, the session-based recommendation method is used to predict next POI. These methods utilize the Markov Chain to model the sequential influence or adopt the factorization model to model the sequential transition [14,15]. However, traditional methods only model users' static preferences. More recently, deep learning-based approaches pay more attention to users' dynamic preference modeling [16,17]. Inspired by RNN, Liu et al. [18] extended RNN to incorporate spatial and temporal contextual information. Manotumruksa et al. [19] captured both the users' dynamic and static preferences via a gate mechanism. Feng et al. [20] proposed a multi-modal RNN to learn the complicated sequential transitions. Gao et al. [21] proposed a time-aware item recommendation to capture the evolution of both user's interests and item's contents information via topic dynamics.

Although a wide array of studies fall within the field mentioned above, this paper is highly-related to the second substream: travel product or package recommendation. Related literature [22–24] collects offline datasets provided by travel companies, containing plenty of click data and purchase records. Unlike traditional products, travel products have several unique characteristics, such as extremely sparse and cold-start users. To address this challenge, some researchers [22,25] try to integrate multi-auxiliary content information with latent factor models to improve recommendation accuracy. Consider users' sequential behaviors, Zhu et al. [26] presented a recommendation method based on sequential topic patterns. However, most of these studies either ignored the historical sessions or time intervals between each interaction, which motivates us to utilize time information, long-term and short-term user interests together to make more accurate recommendation. Furthermore, we regard textual information as an additional supervision signal to capture user click intention.

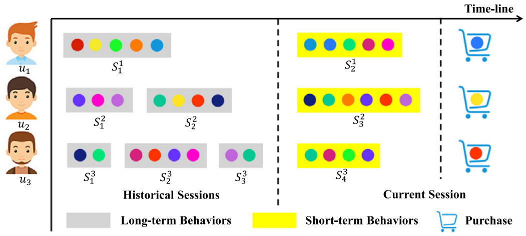


Fig. 2. Long-term and short-term behaviors with a cut time line.

2.2. Recommendations with textual information

A great number of recommendation algorithms have been proposed to leverage side information of users or items (*e.g.*, social network [27–30], item category [16], and user profile [21]) to address data sparsity and cold start problems. Recently, many approaches seek to model textual information to supplement sparse user–item interactions. According to different subjects, textual information can be classified into: item specifications and user reviews. Item specifications describe the attributes of the items, such as product description and paper abstract. Much of the literature applies topic modeling to item specifications and relates its topic distribution to the item latent factor. For instance, CDL [4] replaces the topic model with the auto-encoder to learn the hidden feature vectors from item specifications. CRAE [5] incorporates the item content information and further considers item sequence information. These neural network-based methods can make more accurate recommendations but are highly non-interpretable. The second type is user reviews, which are given by users according to their usage experiences. Some recent work exploits user reviews to generate text as explanations while providing recommendations. Lin et al. [31] jointly provided outfit recommendation and comment generation by combining attentive CNN network with RNN-based generation models. Lin et al. [32] jointly predicted ratings and generated opinionated content, expressing users' sentiment while reviewing an item. However, cold-start items have few user reviews, and user reviews are usually too long and suffer from noise.

Our work distinguishes itself from works mentioned above based on the following facts: (1) We model item textual content (*i.e.*, keywords of product titles), rather than user reviews, which are not available due to user privacy problems in some cases. (2) We seek to generate a piece of keywords text as an output to capture user intents instead of feature vectors.

3. The TRKG model

3.1. Problem formulation

This section introduces several preliminaries and formalizes the problem of travel recommendation with keywords generation.

Definition 1 (*Session*)**.** Session S^u is a clickstream of travel product views for a single user u, which refers to the sequential travel products browsed/purchased by user u during a certain period.

Sessions are divided once the time gap between two clicks is larger than 30 minutes [7]. Similar to [33], we reformulate the new session generation rules: (1) Interactions with the same session ID recorded by the backend system belong to the same one. (2) The maximum length of a session is set to 50, which means a new session will begin when the session length exceeds 50.

For user $u \in \mathcal{U}$, his/her interaction sequence $\{S_1^u, S_2^u, \ldots, S_L^u, S_{L+1}^u\}$ whose elements are arranged in chronological order can be obtained, where S_l^u denotes the lth session of user u. Each session S_l^u of user u can be represented as $S_l^u = \{v_1, v_2, \ldots, v_{|S_l^u|}\}$, where $|S_l^u|$ is the number of travel products in session S_l^u, v_j denotes the jth travel product that user u operates (such as clicking or purchasing). As shown in Fig. 2, the latest session S_{L+1}^u that before the latest purchase of user u is regarded as the short-term behaviors, while the rest of session of user u is regarded as the long-term behaviors $\mathcal{L}^u = \{S_1^u, S_2^u, \ldots, S_L^u\}$. The setting of the parameter L will be discussed in Section 4.5.

Example 1. As shown in Fig. 2, we divide user behaviors into long-term behaviors and short-term behaviors. For user u_2, his long-term behaviors consist of historical sessions S_1^2 and S_2^2, and his short-term behaviors correspond to current session S_3^2.

Definition 2 (*Keywords*)**.** Given a travel product title, we utilize Jieba,[1] a Chinese word segmentation module, to break it into words, and then delete the stop-words from the segmentation results. Finally, we take the preprocessed words as the keywords of the travel product.

Example 2. Fig. 3 gives examples of keywords of the clicked travel products. Keywords are extracted from product titles after Chinese word segmentation and deleting stop word participles.

Problem Definition 1 (*Travel Recommendation*)**.** *Given a user u, let $\mathcal{S}^u = [(v_1, t_1), (v_2, t_2), \ldots, (v_D, t_D)]$ be his/her short-term behaviors, where v_i is a unique identifier for the clicked item, t_i is the corresponding timestamp when this behavior occurred, and $D = |\mathcal{S}^u|$ is the number of travel products in short-term behaviors. The long-term behaviors is denoted as $\mathcal{L}^u = [v_1, v_2, \ldots, v_T]$, where $T = |\mathcal{L}^u|$ is the number of travel products in long-term behaviors. The travel recommendation task is to predict the item v_p that the target user u most likely to purchase in his/her next visit. This task is to fit a Model M as follows:*

$$\mathbf{z} = M(\mathcal{S}^u, \mathcal{L}^u), \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^I$ is a purchasing probability distribution calculated from the long-term and short-term behaviors $\mathcal{S}^u$ and $\mathcal{L}^u$.

Problem Definition 2 (*Keywords Generation*)**.** *Similarly, we can formulate the keywords generation task in the travel recommendation scenarios. By extracting a certain number of keywords (N) based on the word frequency in each session, we can get the keyword sequences of short-term and long-term behaviors. The keyword sequences of short-term and long-term behaviors are denoted as $\mathcal{W}^u = [w_1^s, w_2^s, \ldots, w_N^s]$ and $\mathcal{R}^u = [w_1^l, w_2^l, \ldots, w_M^l]$, where $M = N * L$ and L is the number of sessions in long-term behaviors. The keywords generation task is to predict the keywords $\mathbf{y}$ of the purchased item via a model M':*

$$\mathbf{y} = M'(\mathcal{W}^u, \mathcal{R}^u). \tag{2}$$

Example 3. As shown in Fig. 3, we extract keywords from users' long-term behaviors based on the word frequency in historical sessions. This example includes only one historical session, and the keywords length (M) is set to 30.

Problem Definition 3 (*Travel Recommendation with Keywords Generation*)**.** *Travel recommendation task and keywords generation task can be fused into a multi-task learning framework M'', which*

[1] https://pypi.org/project/jieba/.

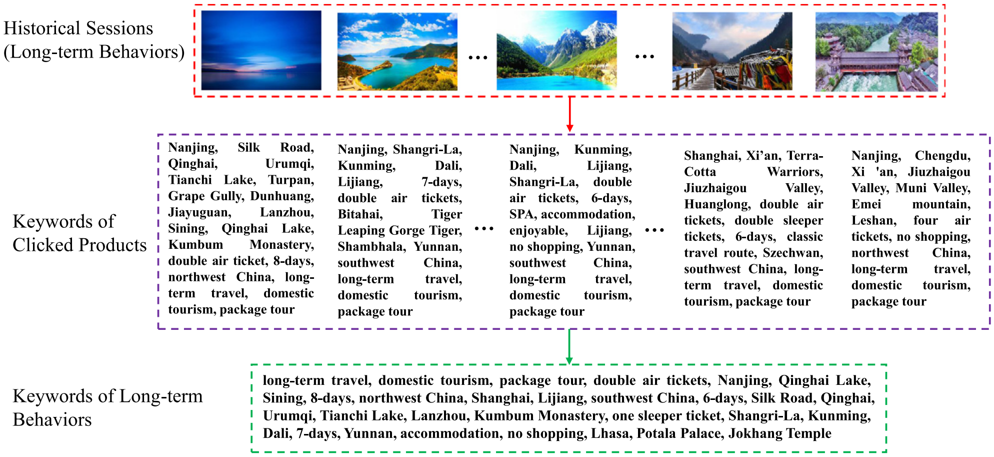

Historical Sessions (Long-term Behaviors)
Keywords of Clicked Products
Nanjing, Silk Road, Qinghai, Urumqi, Tianchi Lake, Turpan, Grape Gully, Dunhuang, Jiayuguan, Lanzhou, Sining, Qinghai Lake, Kumbum Monastery, double air ticket, 8-days, northwest China, long-term travel, domestic tourism, package tour
Nanjing, Shangri-La, Kunming, Dali, Lijiang, 7-days, double air tickets, Bitahai, Tiger Leaping Gorge Tiger, Shambhala, Yunnan, southwest China, long-term travel, domestic tourism, package tour
Nanjing, Kunming, Dali, Lijiang, Shangri-La, double air tickets, 6-days, SPA, accommodation, enjoyable, Lijiang, no shopping, Yunnan, southwest China, long-term travel, domestic tourism, package tour
Shanghai, Xi'an, Terra-Cotta Warriors, Jiuzhaigou Valley, Huanglong, double air tickets, double sleeper tickets, 6-days, classic travel route, Szechwan, southwest China, long-term travel, domestic tourism, package tour
Nanjing, Chengdu, Xi 'an, Jiuzhaigou Valley, Muni Valley, Emei mountain, Leshan, four air tickets, no shopping, northwest China, long-term travel, domestic tourism, package tour
Keywords of Long-term Behaviors
long-term travel, domestic tourism, package tour, double air tickets, Nanjing, Qinghai Lake, Sining, 8-days, northwest China, Shanghai, Lijiang, southwest China, 6-days, Silk Road, Qinghai, Urumqi, Tianchi Lake, Lanzhou, Kumbum Monastery, one sleeper ticket, Shangri-La, Kunming, Dali, 7-days, Yunnan, accommodation, no shopping, Lhasa, Potala Palace, Jokhang Temple

Fig. 3. Extracting keywords from users' long-term behaviors.

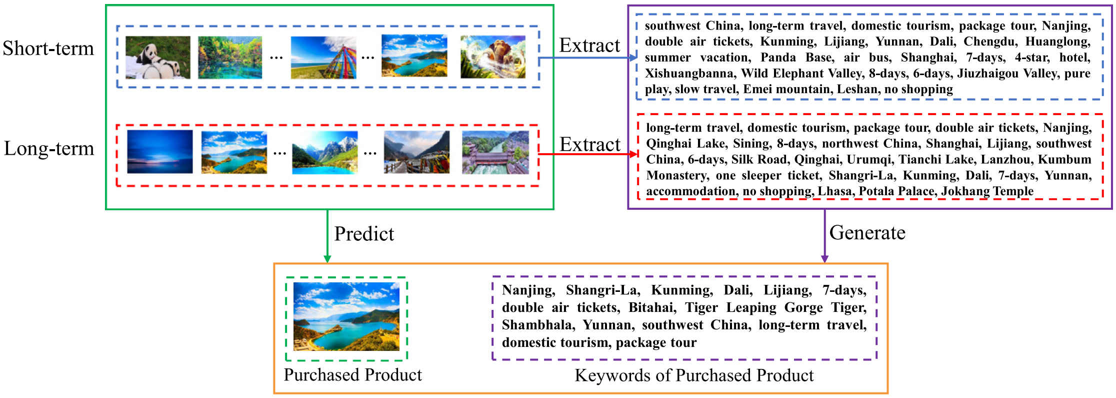

Short-term
Long-term
Extract
Extract
southwest China, long-term travel, domestic tourism, package tour, Nanjing, double air tickets, Kunming, Lijiang, Yunnan, Dali, Chengdu, Huanglong, summer vacation, Panda Base, air bus, Shanghai, 7-days, 4-star, hotel, Xishuangbanna, Wild Elephant Valley, 8-days, 6-days, Jiuzhaigou Valley, pure play, slow travel, Emei mountain, Leshan, no shopping
long-term travel, domestic tourism, package tour, double air tickets, Nanjing, Qinghai Lake, Sining, 8-days, northwest China, Shanghai, Lijiang, southwest China, 6-days, Silk Road, Qinghai, Urumqi, Tianchi Lake, Lanzhou, Kumbum Monastery, one sleeper ticket, Shangri-La, Kunming, Dali, 7-days, Yunnan, accommodation, no shopping, Lhasa, Potala Palace, Jokhang Temple
Predict
Generate
Purchased Product
Nanjing, Shangri-La, Kunming, Dali, Lijiang, 7-days, double air tickets, Bitahai, Tiger Leaping Gorge Tiger, Shambhala, Yunnan, southwest China, long-term travel, domestic tourism, package tour
Keywords of Purchased Product

Fig. 4. An example of travel recommendation with keywords generation.

makes recommendations and generates keywords. Thus, Eqs. (1) and (2) *are combined into:*

$$\mathbf{z}, \mathbf{y} = M''(\mathcal{S}^u, \mathcal{L}^u, \mathcal{W}^u, \mathcal{R}^u), \tag{3}$$

where $\mathbf{y}$ is the recommendation result, and $\mathbf{z}$ is the generated keywords. Supervised by both the purchased product and its keywords, M'' outputs the recommendation list $\mathbf{y}$ and the keywords $\mathbf{z}$ simultaneously.

Example 4. Fig. 4 gives an example of travel recommendation with keywords generation. The keywords of product titles are used as additional supervision signals to improve the performance of travel recommendation. Both travel recommendation and keyword generation take into account users' long-term behaviors and short-term behaviors.

Next section will propose the TRKG model to solve the problem mentioned above, consisting of three key modules: travel recommendation, keywords generation and multi-task learning. The architecture of TRKG is shown in Fig. 5.

3.2. Travel recommendation

For travel recommendation, we propose a time-aware self-attention network to encode user short-term preferences and learn long-term user preferences via a self-attention mechanism [34]. Then we fuse them via a gated-fusion operation to obtain the representation of user preferences. Finally, a predictor transforms this representation into recommendation results.

3.2.1. Short-term preference encoder

Two sessions that contain the same item sequence but with different time intervals among items may reflect different short-term preferences of the user. Considering that the short-term preferences tend to change frequently over time, we propose a time-aware self-attention mechanism to learn the short-term preferences of the target user u from his/her short-term behaviors $\mathcal{S}^u$. In particular, given a time sequence $t = [t_1, t_2, \ldots, t_D]$ corresponding to short-term behaviors of user u, $r_{ij} = |t_i - t_j|$ is the time interval between two items v_i and v_j. Following [35], we clip r_{ij} into a proper range to avoid too large or too small time intervals. Then, we take r_{ij} as an index to get the time interval encoding $\mathbf{r}_{ij}$ via a fixed set of sinusoid functions as the basis [36].

$$\begin{aligned} Base\left(r_{ij}, 2k\right) &= sin\left(r_{ij}/10000^{2k/d}\right), \\ Base\left(r_{ij}, 2k+1\right) &= cos\left(r_{ij}/10000^{2k/d}\right), \\ \mathbf{r}_{ij} &= f\left(Base\left(r_{ij}\right)\right), \end{aligned} \tag{4}$$

where f is a linear project function.

Similarly, the position encoding $\mathbf{p}_j$ can also be calculated by sinusoid function following [37]. With the time interval encoding $\mathbf{r}_{ij}$ and the position encoding $\mathbf{p}_j$, the output $\mathbf{e}_i$ is calculated as a weighted sum of linearly transformed embeddings of items:

$$\mathbf{e}_i = \sum_{j=1}^{D} \alpha_{ij} \left(\mathbf{W}_s^V \mathbf{v}_j + \mathbf{r}_{ij} + \mathbf{p}_j\right), \tag{5}$$

where $\mathbf{W}_s^V$ is projection parameters and $\mathbf{v}_j$ denotes the ID embedding of item v_j. The weight coefficient α_{ij} is learned by the

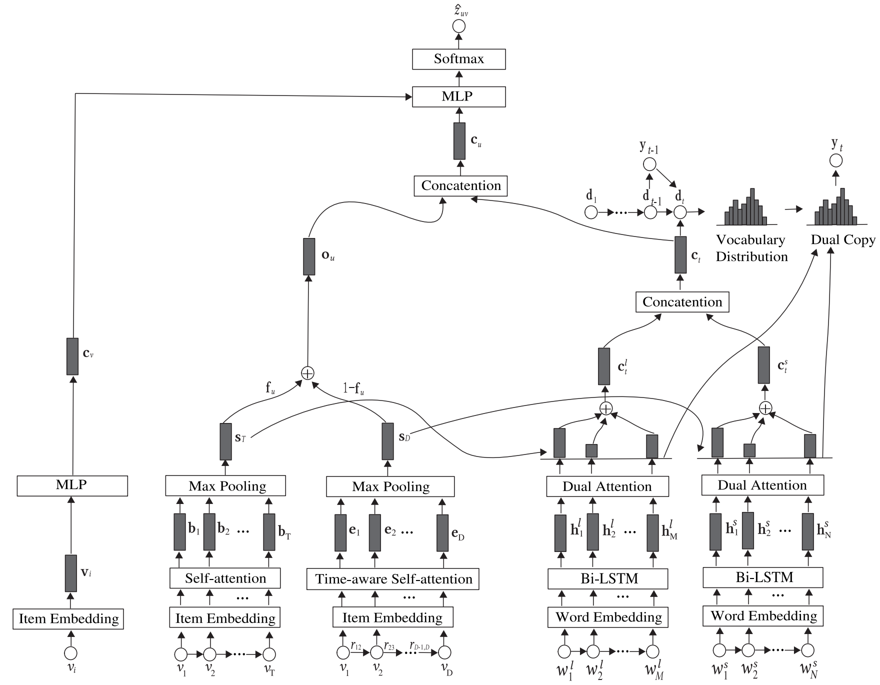


Fig. 5. Illustration of our proposed TRKG model.

time-aware self-attention network:

$$\begin{aligned}\alpha_{ij} &= \frac{\mathbf{W}_s^Q \mathbf{v}_i \left(\mathbf{W}_s^K \mathbf{v}_j + \mathbf{r}_{ij} + \mathbf{p}_j\right)^{\top}}{\sqrt{d}}, \\ \alpha_{ij} &= \frac{\exp(\alpha_{ij})}{\sum_{k=1}^{D} \exp(\alpha_{ik})},\end{aligned} \tag{6}$$

where $\mathbf{W}_s^Q$ and $\mathbf{W}_s^K$ are linear transformation matrices for a query and key respectively, and the scale factor $\sqrt{d}$ is to avoid exceedingly large inner products. We simply use a max pooling function to generate the final embedding vector of short-term preferences:

$$\mathbf{s}_{uj} = \max_{1 \le j \le d} \left(\mathbf{e}_{1j}, \mathbf{e}_{2j}, \ldots, \mathbf{e}_{Dj}\right). \tag{7}$$

3.2.2. Long-term preference encoder

Unlike the short-term behavior sequence, long-term behaviors are usually stable. Thus, we only take the position encoding into account when learning user long-term preferences. Similar to the short-term preference module, we obtain the embedding vector of long-term preference $\mathbf{l}_u$ from user long-term behaviors $\mathcal{L}^u$ via self-attention mechanism and the max pooling as follows:

$$\begin{aligned}\mathbf{b}_i &= \sum_{j=1}^{T} \gamma_{ij} \left(\mathbf{W}_l^V \mathbf{v}_j + \mathbf{p}_j\right), \\ \gamma_{ij} &= \frac{\mathbf{W}_l^Q \mathbf{v}_i \left(\mathbf{W}_l^K \mathbf{v}_j + \mathbf{p}_j\right)^{\top}}{\sqrt{d}}, \\ \gamma_{ij} &= \frac{\exp(\gamma_{ij})}{\sum_{k=1}^{T} \exp(\gamma_{ik})}, \\ \mathbf{l}_{uj} &= \max_{1 \le j \le d} \left(\mathbf{b}_{1j}, \mathbf{b}_{2j}, \ldots, \mathbf{b}_{Tj}\right),\end{aligned} \tag{8}$$

where $\mathbf{W}_l^V$, $\mathbf{W}_l^Q$ and $\mathbf{W}_l^K$ are learnable parameters. $\mathbf{b}_i$ is the embedding of item in the long-term behavior sequence and $\mathbf{l}_u$ represents the embedding vector of the long-term preferences.

3.2.3. Long-term and short-term preferences fusion

To combine the long-term preferences with short-term preferences, we employ a simple gated-fusion operation, same as the previous work [38]. Based on this scheme, the gate vector $\mathbf{f}_u$ is designed to control the importance of long-term and short-term preferences:

$$\mathbf{f}_u = sigmoid\left(\mathbf{W}_m \mathbf{l}_u + \mathbf{W}_n \mathbf{s}_u + \mathbf{W}_u \mathbf{q}_u + \mathbf{b}_u\right), \tag{9}$$

where $\mathbf{W}_m$, $\mathbf{W}_n$, $\mathbf{W}_u$ and $\mathbf{b}_u$ are parameters, $\mathbf{l}_u$ is long-term preferences, $\mathbf{s}_u$ is short-term preferences, $\mathbf{q}_u$ is ID embedding vector of user u. The final representation of user preferences is calculated by:

$$\mathbf{o}_u = \mathbf{f}_u \odot \mathbf{l}_u + (1 - \mathbf{f}_u) \odot \mathbf{s}_u, \tag{10}$$

where $\odot$ is element-wise multiplication.

3.2.4. Predictor

Following [39], we perform a bi-linear decoding operation to transform the user representation $\mathbf{o}_u$ into the probability vector. The purchase probability of item v for target user u is computed as:

$$p\left(\hat{z}_{uv}\right) = Softmax\left(\mathbf{c}_v^{\top} \mathcal{B} \mathbf{o}_u\right), \tag{11}$$

where $\mathcal{B}$ is the parameter matrix, $\mathbf{c}_v$ is the item embedding obtained by ID embedding layer and multi-layer perceptron (MLP) operation.

Lastly, the loss function of the travel recommendation task is defined as the cross-entropy of the prediction probability and the ground truth:

$$\mathcal{L}_z = -\sum_{v=1}^{I} z_{uv} log\left(p\left(\hat{z}_{uv}\right)\right), \tag{12}$$

where I is the number of travel products, $p(\hat{z}_{uv})$ is the prediction probability of item v and z_{uv} is the corresponding ground truth label.

3.3. Keywords generation

Owing to the success of the pointer generator network [40] in natural language processing, we extend it and propose a multi-source pointer generator network. It encodes the short-term keyword sequence as well as the long-term keyword sequence at the same time. In decoding, the decoder generates keywords from the short-term keywords encoder and the long-term keywords encoder.

3.3.1. Long-term and short-term keywords encoder

To encode long-term and short-term keyword sequences, we employ the bidirectional LSTM (Bi-LSTM). Bi-LSTM takes word embedding $\mathbf{w}_i$ as an input vector and outputs two hidden states $\overrightarrow{\mathbf{h}_i} = \overrightarrow{\mathrm{LSTM}}(\mathbf{w}_i, \overrightarrow{\mathbf{h}_{i-1}})$, $\overleftarrow{\mathbf{h}_i} = \overleftarrow{\mathrm{LSTM}}(\mathbf{w}_i, \overleftarrow{\mathbf{h}_{i+1}})$ from the forward and backward LSTMs, respectively. Then, Bi-LSTM concatenates the two hidden state vectors as the word representation vector $\mathbf{h}_i = [\mathbf{h}_i^{(f)} : \mathbf{h}_i^{(b)}]$. After the above operation, we can get the hidden state of the long-term keyword sequence $[\mathbf{h}_1^l, \mathbf{h}_2^l, \ldots, \mathbf{h}_M^l]$ and the hidden state of short-term keyword sequence $[\mathbf{h}_1^s, \mathbf{h}_2^s, \ldots, \mathbf{h}_N^s]$.

3.3.2. Dual-attention generator

Both the long-term keyword sequence and short-term keyword sequence play an important role in generating keywords of the final purchased item. We believe that they can benefit from sharing parameters to promote the capacity of capturing the user purchase intention. Thus, we employ a dual-attention mechanism to generate the context vector based on attention over long-term and short-term keywords.

Specifically, we transform the final hidden states $\mathbf{h}_N^l$ and $\mathbf{h}_N^s$ into the initial state $\mathbf{d}_0$ of the decoder using a rectified layer:

$$\mathbf{d}_0 = ReLU(\mathbf{W}_d[\mathbf{h}_M^l, \mathbf{h}_N^s]), \tag{13}$$

where $\mathbf{W}_d$ is the learnable parameter.

For the long-term keywords encoder and short-term keywords encoder, we calculate the corresponding attention distribution. The attention distribution can be seen as a probability on the source words, which tells the decoder where to look to generate the next word. The calculation formula is below:

$$\begin{aligned} \mathbf{e}_{ti}^l &= \mathbf{v}_l^{\mathrm{T}}(\mathbf{W}_l[\mathbf{d}_t, \mathbf{h}_i^l, \mathbf{h}_N^s] + \mathbf{b}_l), \\ \mathbf{e}_{ti}^s &= \mathbf{v}_s^{\mathrm{T}}(\mathbf{W}_s[\mathbf{d}_t, \mathbf{h}_i^s, \mathbf{h}_M^l] + \mathbf{b}_s), \\ \boldsymbol{\beta}_t^l &= softmax(\mathbf{e}_t^l), \quad \boldsymbol{\beta}_t^s = softmax(\mathbf{e}_t^s), \end{aligned} \tag{14}$$

where $\boldsymbol{\beta}_t^l$ is attention distribution for long-term keywords encoder, $\boldsymbol{\beta}_t^s$ is attention distribution for short-term keywords encoder, $\mathbf{v}_l$, $\mathbf{v}_s$, $\mathbf{b}_l$ and $\mathbf{b}_s$ are learnable parameters. $\mathbf{d}_t$ is decoder hidden state at time step t, we compute it as follows:

$$\mathbf{d}_t = LSTM(\mathbf{d}_{t-1}, \mathbf{y}_{t-1}, \mathbf{c}_{t-1}^l, \mathbf{c}_{t-1}^s), \tag{15}$$

where $\mathbf{d}_{t-1}$ is the decoder state at step $t-1$, $\mathbf{y}_{t-1}$ is the input of the decoder at step t. $\mathbf{c}_{t-1}^l$ and $\mathbf{c}_{t-1}^s$ are context vectors produced by a weighted sum of the encoder hidden states:

$$\begin{aligned} \mathbf{c}_t^s &= \sum_i \beta_{ti}^s \mathbf{h}_i^s, \\ \mathbf{c}_t^l &= \sum_i \beta_{ti}^l \mathbf{h}_i^l. \end{aligned} \tag{16}$$

3.3.3. Gated fusion

We introduce a fusion gate vector to combine two context vectors $\mathbf{c}_t^s$ and $\mathbf{c}_t^l$ as the final context vector.

$$\begin{aligned} \mathbf{g}_t &= sigmoid(\mathbf{W}_g[\mathbf{c}_t^l, \mathbf{c}_t^s]), \\ \mathbf{c}_t &= \mathbf{g}_t \odot \mathbf{c}_t^s + (1 - \mathbf{g}_t) \odot \mathbf{c}_t^l, \end{aligned} \tag{17}$$

where $\mathbf{g}_t$ is the fusion gate vector and $\mathbf{W}_g$ is the learnable parameter.

Finally, the probability distribution over all words in the vocabulary for timestep t is calculated from the decoder state $\mathbf{d}_t$ and the context vector $\mathbf{c}_t$ as follows:

$$p_{vocab}(w) = softmax(\mathbf{W}_h[\mathbf{d}_t, \mathbf{c}_t] + \mathbf{b}_h), \tag{18}$$

where $\mathbf{W}_h$ and $\mathbf{b}_h$ are learnable parameters.

3.3.4. Dual-copy pointer

To handle the problem of out-of-vocabulary (OOV) words and improve accuracy, we propose a dual-copy pointer that copies words from both the short-term keywords and long-term keywords as below:

$$\begin{aligned} p_{copyl}(w) &= \sum_{i:w_i=w} \beta_{ti}^l, \\ p_{copys}(w) &= \sum_{i:w_i=w} \beta_{ti}^s. \end{aligned} \tag{19}$$

The additional generation probability λ_t for timestep t is designed to decide whether to generate the word from the vocabulary distribution p_{vocab} or copy one from the vocabulary distribution p_{copyl} and p_{copys}. We obtain the following probability distribution over the extended vocabulary:

$$\begin{aligned} p(w) =& \lambda_t p_{vocab}(w) + \frac{1}{2}(1 - \lambda_t)(p_{copyl}(w) \\ &+ p_{copys}(w)), \end{aligned} \tag{20}$$

$$\lambda_t = softmax(\mathbf{W}_t[\mathbf{d}_t, \mathbf{y}_{t-1}, \mathbf{c}_t] + \mathbf{b}_t). \tag{21}$$

In the training phase, we use the negative log-likelihood as the loss function:

$$\mathcal{L}_g = -\sum_{t=1}^{N_t} \sum_{s=1}^{N_s} \mathbf{y}_t^s logp(\hat{\mathbf{y}}_t^s), \tag{22}$$

where N_t is the length of keywords y and N_s is the extended word vocabulary size.

3.4. Multi-task learning

The travel recommendation task can learn the representation of sequential behaviors and the keywords generation task can capture the semantic information. Therefore, we use a multi-task learning setup [41] to jointly optimize travel recommendation and keywords generation via parameter sharing.

In detail, following [39], we combine the preference representation $\mathbf{o}_u$ and hidden state $\mathbf{c}_0$ into $\mathbf{c}_u$ via a linear transformation and replace $\mathbf{o}_u$ with $\mathbf{c}_u$ in Eq. (11) as follows:

$$\begin{aligned} p\left(\hat{z}_{uv}\right) &= softmax\left(\mathbf{c}_v^{\mathrm{T}} \mathcal{B} \mathbf{c}_u\right), \\ \mathbf{c}_u &= \mathbf{W}_u'[\mathbf{o}_u, \mathbf{c}_0]. \end{aligned} \tag{23}$$

Moreover, similar to [42], we view the attention distribution of keywords as a probability distribution over the source words and user preference, which tells the decoder which word to generate.

$$\begin{aligned} \mathbf{e}_{ti}^l &= \mathbf{v}_l'^{\mathrm{T}}(\mathbf{W}_l'[\mathbf{d}_t, \mathbf{h}_i^l, \mathbf{h}_N^s, \mathbf{l}_u] + \mathbf{b}_l'), \\ \mathbf{e}_{ti}^s &= \mathbf{v}_s'^{\mathrm{T}}(\mathbf{W}_s'[\mathbf{d}_t, \mathbf{h}_i^s, \mathbf{h}_M^l, \mathbf{s}_u] + \mathbf{b}_s'), \end{aligned} \tag{24}$$

where $\mathbf{v}_l'$, $\mathbf{v}_s'$, $\mathbf{W}_l'$, $\mathbf{W}_s'$, $\mathbf{b}_l'$ and $\mathbf{b}_s'$ are learnable parameters.

Finally, the loss function of the multi-task learning framework is denoted as:

$$\mathcal{L} = \eta \mathcal{L}_z + (1 - \eta)\mathcal{L}_g, \tag{25}$$

where $\mathcal{L}_z$ is the travel recommendation loss in Eq. (12), and $\mathcal{L}_g$ is the loss of keywords generation in Eq. (22). $\eta \in (0, 1)$ is the weight of the travel recommendation task.

4. Experiments

In this section, we begin by introducing datasets, evaluation metrics and baseline methods, then we conduct experiments on real-world datasets to answer the following five research questions:

- **RQ1:** How does the TRKG approach perform compared with state-of-the-art recommendation methods?
- **RQ2:** What are the impacts of some of the design choices of TRKG? Can the keywords generation improve the recommendation performance of TRKG?
- **RQ3:** How do the weights in multi-task learning affect the performance of the TRKG method?
- **RQ4:** Do the number of historical sessions in long-term behaviors and lengths of keywords in each session have impacts on the performance of the travel recommendation?
- **RQ5:** What is the quality of the keywords generated by TRKG?

4.1. Experimental settings

Datasets. The dataset is provided by Tuniu,[2] a large tourism e-commerce company in China. In particular, we collect two months' logs of user click and purchase activity from July 1st to August 31st in 2013 and divide it into two experimental datasets (*i.e.,* D_1 and D_2). Then, we remove all sessions of length 1 and items that appear less than 2 times, and filter out items without product title. After that, for both datasets, we use the first 28 days' logs for training, logs of the 29th day for validation, and logs of the last two days for testing. Detailed statistics of datasets can be found in Table 1.

Evaluation Scheme. We evaluate the recommendation performance of all methods with hit ratio (HR) and mean reciprocal rank (MRR). Consider that only one item is purchased in each test case, HR@k is equivalent to Recall@k and proportional to Precision@k. This experiment reports HR and MRR with $k = 5, 10, 15, 20$.

HR@k measures the proportion of the cases that have correctly recommended items among the top-k items in all test cases. It is defined as follows:

$$\text{HR@}k = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} \mathbb{I}(R_{u,g_u} \le k), \tag{26}$$

where g_u is the item purchased in the current session for user u, R_{u,g_u} is the generated rank for item g_u and user u, and $\mathbb{I}$ is an indicator function.

MRR@k is the average of reciprocal ranks of the purchased items R_{u,g_u}:

$$\text{MRR@}k = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} \frac{1}{R_{u,g_u}}, \tag{27}$$

where MRR@k does not consider the rank that is larger than k.

To evaluate the quality of keywords generation, we use Rouge [43], a classic evaluation metric in the field of text summarization, as our evaluation metric. Rouge measures the keywords generation quality by counting the overlapping units between the generated keywords $\hat{s}$ and the ground truth keywords s. The Rouge-N score for $\hat{s}$ is defined as follows:

$$\text{ROUGE} - N(\hat{s}) = \frac{\sum_{g_n\in\hat{s}} C_m(g_n)}{\sum_{g_n\in\tilde{s}} C(g_n)}, \tag{28}$$

where $C(g_n)$ is the number of n-grams in $\tilde{s}$ ($\hat{s}$ or s), $C_m(g_n)$ is the number of n-grams co-occurring in $\hat{s}$ and s. When $\tilde{s} = s$, we can get ROUGE$_{recall}$, and when $\tilde{s} = \hat{s}$, we can get ROUGE$_{precision}$. In the experiment, We use Recall, Precision and F-measure of Rouge-1, Rouge-2 and Rouge-L to evaluate the quality of the generated keywords.

Baselines. To evaluate the performance of purchase prediction, we compare TRKG with three traditional recommendation approaches (*i.e.,* POP, Item-KNN and BPR), six widely-applied neural-based recommendation methods (*i.e.,* CDL, GRURec-TopK, Time-GRURec, STAMP, SR-GNN and ESRM-KG), and four variants of TRKG (*i.e.,* TRKG-NoG, TRKG-NoL, TRKG-NoT and TRKG-NoP) for ablation study.

- **POP**. POP simply selects popular items according to purchase frequency in the training data.
- **Item-KNN** [44]. The standard item-based collaborative filtering method with cosine similarity as a similarity measure is used here.
- **BPR** [45]. BPR adopts a stochastic gradient descent framework that contrasts pairs of positive and negative samples.
- **CDL** [4]. CDL is a hierarchical deep Bayesian model for joint learning of the auto-encoder and matrix factorization. In CDL, the auto-encoder is used to learn the hidden feature vectors from the auxiliary information of items.
- **GRURec-TopK** [6]. GRU-TopK is an RNN-based recommendation model that iteratively reads session items predicting the next one through a ranking-based loss function.
- **Time-GRURec** [46]. Time-GRURec is a variant of the GRU architecture that incorporates time intervals to track user interests shifting.
- **STAMP** [7]. STAMP considers both short-term interests and long-term preferences by using an attention mechanism to combine both of them.
- **SR-GNN** [9]. SR-GNN constructs a graph for session sequences based on the transition of items and predicts the next action with the hidden state of users learned from the graph neural network.
- **ESRM-KG** [39]. ESRM-KG is a hybrid approach that fuses the keywords generation into the session-based recommendation to enhance the performance.
- **TRKG-NoG**. TRKG-NoG removes the keywords generation part from the model.
- **TRKG-NoL**. TRKG-NoL means entirely removing the long-term preference encoder from both keywords generation and purchase prediction.
- **TRKG-NoT**. TRKG-NoT means neglecting the effect of time intervals in the current session in the TRKG model.
- **TRKG-NoP**. TRKG-NoP means that TRKG removes the purchase prediction part and is only used for keywords generation task.

Parameters Settings. For all comparison methods, we carry out experiments under the optimal parameter settings. For our proposed method, the embedding dimensions of items and words both are set to 128. The batch size is searched in [64, 128, 256, 512], and 128 is selected for both datasets. The learning rate is tuned amongst [0.0001, 0.0005, 0.001, 0.005], and the validation results show that 0.0005 and 0.001 are better for datasets D_1 and D_2, respectively. We used Adam optimizer for all gradient-based methods. We apply the dropout strategy to avoid overfitting and dropout ratio is set 0.2 for both datasets. In the test phase, the keywords are generated using beam search with beam size 5. The number of historical sessions in long-term behaviors (L) is set to 3, and the length of keywords in each session (N) is set to 30. The TRKG and all the compared neural-based models are defined and trained on a Windows server with 3.60 GHz Intel I9-9900k CPU

[2] http://www.tuniu.com.

Table 1
Description of datasets.

Datasets	Data type	Time interval	#Users	#Items	#Records	#Sessions	A.Len	#Purchased items
D_1	Train	1 to 28 Jul.	18,825	23,374	296,697	49,076	6.0457	6,953
	Validation	29 Jul.	1,193	4,096	10,613	1,645	6.4517	755
	Test	30 to 31 Jul.	1,817	5,882	19,762	2,944	6.7126	1,318
D_2	Train	1 to 28 Aug.	24,068	26,440	357,858	58,018	6.1681	8,650
	Validation	29 Aug.	923	3,414	8,064	1,279	6.3049	706
	Test	30 to 31 Aug.	1,186	4,442	11,557	1,715	6.7388	1,077

Note: (1) "#" indicates the number of some object;
(2) "A.Len" indicates the average length of clickstream in a session.

Table 2
Performance comparisons of travel recommendation on dataset D_1.

Methods	HR@5	MRR@5	HR@10	MRR@10	HR@15	MRR@15	HR@20	MRR@20
POP	0.0180	0.0095	0.0350	0.0117	0.0450	0.0125	0.0530	0.0129
Item-KNN	0.0985	0.0349	0.1333	0.0390	0.1581	0.0408	0.1766	0.0418
BPR	0.0741	0.0582	0.0966	0.0604	0.1098	0.0610	0.1194	0.0611
CDL	0.1302	0.0975	0.2152	0.1140	0.2736	0.1209	0.3218	0.1251
GRURec-TopK	0.3645	0.2292	0.4520	0.2410	0.5033	0.2450	0.5364	0.2469
Time-GRURec	0.3747	0.2303	0.4577	0.2627	0.5128	0.2871	0.5478	0.2991
STAMP	0.3255	0.2053	0.4005	0.2153	0.4486	0.2191	0.4794	0.2206
SR-GNN	0.4467	0.3506	0.4900	0.3564	0.5172	0.3585	0.5322	0.3593
ESRM-KG	0.4820	0.3579	0.5180	0.3630	0.5290	0.3639	0.5380	0.3644
TRKG-NoG	0.4999	0.3827	0.5428	0.3900	0.5781	0.3964	0.5822	0.3989
TRKG-NoL	0.5024	0.3833	0.5478	0.4031	0.5899	0.4104	0.5936	0.4063
TRKG-NoT	0.5037	0.3976	0.5570	0.3960	0.5781	0.4043	0.5989	0.4042
TRKG	**0.5075**	**0.4038**	**0.5665**	**0.4116**	**0.5910**	**0.4136**	**0.6085**	**0.4145**

Table 3
Performance comparisons of travel recommendation on dataset D_2.

Methods	HR@5	MRR@5	HR@10	MRR@10	HR@15	MRR@15	HR@20	MRR@20
POP	0.0140	0.0039	0.0300	0.0060	0.0450	0.0072	0.0530	0.0076
Item-KNN	0.1109	0.0395	0.1497	0.0441	0.1718	0.0457	0.1883	0.0466
BPR	0.0796	0.0621	0.0988	0.0642	0.1107	0.0647	0.1185	0.0648
CDL	0.1357	0.0978	0.2174	0.1134	0.2787	0.1203	0.3269	0.1242
GRURec-TopK	0.3608	0.2210	0.4424	0.2320	0.4847	0.2353	0.5184	0.2372
Time-GRURec	0.3750	0.2229	0.4542	0.2349	0.4974	0.2483	0.5258	0.2499
STAMP	0.3465	0.2198	0.4276	0.2307	0.4705	0.2341	0.4960	0.2355
SR-GNN	0.4441	0.3409	0.4878	0.3471	0.5041	0.3484	0.5218	0.3494
ESRM-KG	0.4790	0.3565	0.5190	0.3622	0.5340	0.3633	0.5490	0.3642
TRKG-NoG	0.4984	0.3879	0.5542	0.4010	0.6070	0.3949	0.6315	0.3877
TRKG-NoL	0.4942	0.3842	0.5554	0.3975	0.6057	0.3987	0.6268	0.4050
TRKG-NoT	0.5025	0.3895	0.5771	0.3926	0.6136	0.3940	0.6338	0.3927
TRKG	**0.5095**	**0.3920**	**0.5785**	**0.4015**	**0.6200**	**0.4048**	**0.6455**	**0.4063**

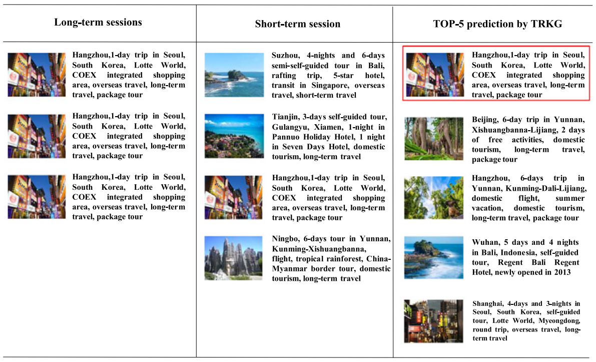

Long-term sessions
Short-term session
TOP-5 prediction by TRKG
Hangzhou,1-day trip in Seoul, South Korea, Lotte World, COEX integrated shopping area, overseas travel, long-term travel, package tour
Hangzhou,1-day trip in Seoul, South Korea, Lotte World, COEX integrated shopping area, overseas travel, long-term travel, package tour
Hangzhou,1-day trip in Seoul, South Korea, Lotte World, COEX integrated shopping area, overseas travel, long-term travel, package tour
Suzhou, 4-nights and 6-days semi-self-guided tour in Bali, rafting trip, 5-star hotel, transit in Singapore, overseas travel, short-term travel
Tianjin, 3-days self-guided tour, Gulangyu, Xiamen, 1-night in Pannuo Holiday Hotel, 1 night in Seven Days Hotel, domestic tourism, long-term travel
Hangzhou,1-day trip in Seoul, South Korea, Lotte World, COEX integrated shopping area, overseas travel, long-term travel, package tour
Ningbo, 6-days tour in Yunnan, Kunming-Xishuangbanna, flight, tropical rainforest, China-Myanmar border tour, domestic tourism, long-term travel
Hangzhou,1-day trip in Seoul, South Korea, Lotte World, COEX integrated shopping area, overseas travel, long-term travel, package tour
Beijing, 6-day trip in Yunnan, Xishuangbanna-Lijiang, 2 days of free activities, domestic tourism, long-term travel, package tour
Hangzhou, 6-days trip in Yunnan, Kunming-Dali-Lijiang, domestic flight, summer vacation, domestic tourism, long-term travel, package tour
Wuhan, 5 days and 4 nights in Bali, Indonesia, self-guided tour, Regent Bali Regent Hotel, newly opened in 2013
Shanghai, 4-days and 3-nights in Seoul, South Korea, self-guided tour, Lotte World, Myeongdong, round trip, overseas travel, long-term travel

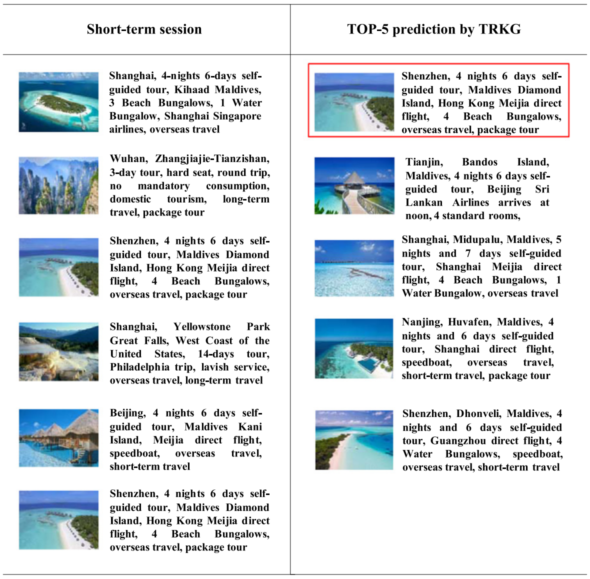
Fig. 6. Example 1 of input and output of TRKG. We mark the ground truth with a red rectangular box.

and 11 GB Nvidia GeForce RTX 2080 Ti GPU, and implemented in Pytorch.[3]

4.2. Overall performance comparison (RQ1)

This section analyzes the results of TRKG and baseline methods in terms of the metrics HR and MRR. The results on both datasets are presented in Tables 2 and 3, where the best result for the corresponding metric is highlighted in bold type. From the result, there are several observations. First and foremost, TRKG significantly outperforms benchmark methods indicated by all of the evaluation measures. For example, it outperforms the second-best performer ESRM-KG 13.1% and 13.7% in terms of HR@20 and MRR@20 on dataset D_1, respectively. Second, neural-based recommendation methods (*i.e.*, CDL, GRURec-TopK, Time-GRURec, STAMP, SR-GNN and ESRM-KG) outperform traditional recommendation approaches (*i.e.*, POP, Item-KNN and BPR) in all instances. This is probably because user–item interaction is so sparse that traditional methods are not suitable for real-world travel recommendation scenarios. This also demonstrates the superiority of neural networks, especially the great ability in modeling the high-order interactions between users and items. Third, we observe that Time-GRURec achieves better performance than GRU-TopK, because Time-GRURec takes time intervals into account which can help to capture the evolution of the purchase intentions of users. However, GRU-TopK ignores the impact of

[3] https://pytorch.org/.



Short-term session	TOP-5 prediction by TRKG
Shanghai, 4-nights 6-days self-guided tour, Kihaad Maldives, 3 Beach Bungalows, 1 Water Bungalow, Shanghai Singapore airlines, overseas travel	Shenzhen, 4 nights 6 days self-guided tour, Maldives Diamond Island, Hong Kong Meijia direct flight, 4 Beach Bungalows, overseas travel, package tour
Wuhan, Zhangjiajie-Tianzishan, 3-day tour, hard seat, round trip, no mandatory consumption, domestic tourism, long-term travel, package tour	Tianjin, Bandos Island, Maldives, 4 nights 6 days self-guided tour, Beijing Sri Lankan Airlines arrives at noon, 4 standard rooms,
Shenzhen, 4 nights 6 days self-guided tour, Maldives Diamond Island, Hong Kong Meijia direct flight, 4 Beach Bungalows, overseas travel, package tour	Shanghai, Midupalu, Maldives, 5 nights and 7 days self-guided tour, Shanghai Meijia direct flight, 4 Beach Bungalows, 1 Water Bungalow, overseas travel
Shanghai, Yellowstone Park Great Falls, West Coast of the United States, 14-days tour, Philadelphia trip, lavish service, overseas travel, long-term travel	Nanjing, Huvafen, Maldives, 4 nights and 6 days self-guided tour, Shanghai direct flight, speedboat, overseas travel, short-term travel, package tour
Beijing, 4 nights 6 days self-guided tour, Maldives Kani Island, Meijia direct flight, speedboat, overseas travel, short-term travel	Shenzhen, Dhonveli, Maldives, 4 nights and 6 days self-guided tour, Guangzhou direct flight, 4 Water Bungalows, speedboat, overseas travel, short-term travel
Shenzhen, 4 nights 6 days self-guided tour, Maldives Diamond Island, Hong Kong Meijia direct flight, 4 Beach Bungalows, overseas travel, package tour	

Fig. 7. Example 2 of input and output of TRKG. We mark the ground truth with a red rectangular box.

time on the session-based recommendation. Besides, ESRM-KG achieves better performance than other baseline methods. This is mainly because ESRM-KG enhances the item embedding to reinforce the recommendation task by keywords generation task. Last but not least, although CDL utilizes item title information, it performs worse than ESRM-KG and TRKG, indicating that RNN-based methods have better abilities to dynamically learn users' preferences than matrix factorization based models.

Case Study on Recommendation. To better understand what can be recommended by the TRKG model, we show two typical examples in Figs. 6 and 7, one of which contains long-term preferences while the other does not. From these two examples, we summarize the following three features of the TRKG model: (1) TRKG can learn the effect of long-term preferences on purchase behaviors in the current session (short-term session). For instance, in Example 1, a user frequently clicks a tourism product related to South Korea in the long-term sessions and in the current session, he/she again browses this product as well as other three travel products related to Bali, Xiamen, Yunnan and South Korea. TRKG regards "South Korea" as the purchase intention in the current session after integrating long-term and short-term preferences, and successfully predicts the purchased product at the first position in the recommendation list. (2) TRKG can capture the purchase intentions of users in the current session. In Example 1, we can see that TRKG recommends tourism products relative to Bali, Xiamen and Yunnan. This validates the effectiveness of TRKG in capturing the intention. (3) Accidental clicks in the current session can be filtered by the TRKG model. For instance, in Example 2, TRKG only recommends travel products relative to Maldives in the top 5 of the recommendation list and ignores accidental clicks relative to Zhangjiajie and Yellowstone.

4.3. Ablation study (RQ2)

To further understand the impacts of the critical components of TRKG on recommendation quality, we perform some ablation studies. Tables 2 and 3 show the results of three simplified variants. We observe that TRKG significantly outperforms TRKG-NoG, TRKG-NoL and TRKG-NoT on both datasets with respect to both metrics. This demonstrates that these components that make up TRKG are beneficial to model user preferences, and combining them leads to better performance. Moreover, TRKG-NoG performs worse than TRKG-NoL and TRKG-NoT on both datasets. This reveals that keywords generation part has a larger impact in learning user purchase intentions in our proposed method TRKG.

4.4. Impact of different weights η on multi-task learning (RQ3)

The selection of weight η in multi-task learning is essentially the tradeoff between recommendation task and keywords generation task. We vary η from 0.1 to 0.9 with a step size 0.2 in the experiment. The results are reported in Fig. 8. From Fig. 8, we can observe that: First, a very high η (*i.e.*, η=0.9) leads to the uselessness of supervision provided by the keywords and result in poor recommendation accuracy. Second, a very low η (*i.e.*, η=0.1) does not fully utilize the user–item interactions and gets the worse result. Third, different η (except for 0.9 and 0.1) have a slight impact on purchase prediction results in terms of HR and MRR, and can obtain satisfactory performance. This fact confirms that just considering user–item interactions or keywords may be insufficient to make recommendations.

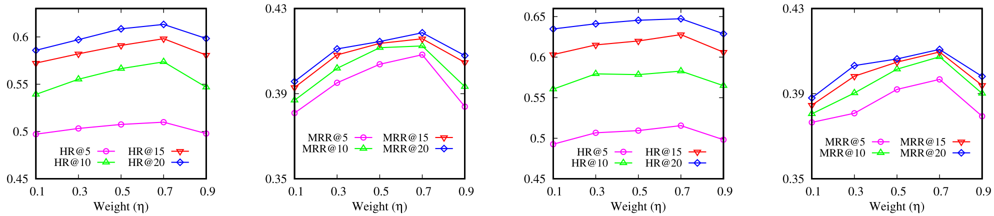
4.5. Parameter analysis (RQ4)

We vary the number of historical sessions in long-term behaviors (L) from 1 to 5 with a step size 1, and the length of keywords in each session (N) from 10 to 50 with a step size 10, to study the effect on the performance of travel recommendation. The results on datasets D_1 and D_2 are shown in Figs. 9 and 10. From these figures, we make the following observations: (1) In general, when L is less than 3, the performance of TRKG becomes better and better with the increasement of L, because long-term behaviors provide more information about user purchase intentions and the results of recommendation are more accurate. When the number of historical sessions increases again, the recommendation performance has lesser improvements or even decreases. We argue that this is mainly because historical behaviors that have passed for a long time have little effect on current purchase behaviors. (2) Similarly, when we increase the length of keywords in each session (N), the performance tends to first increase rapidly and then increase slowly or even decrease. (3) TRKG shows relatively good recommendation performance when $N = 30$ and $L = 3$, which is the default setting for TRKG.

4.6. Quality of generated keywords (RQ5)

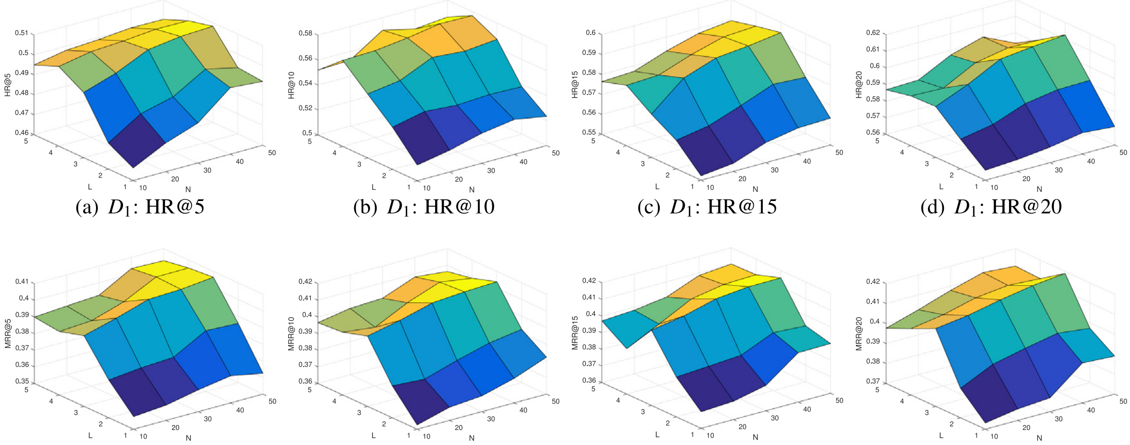
This section presents the comparison of keywords generation performance among TRKG, ESRM-KG and TRKG-NoP, where other baseline methods are not reported, since they do not contain keywords generation module. The comparison results on both datasets are shown in Table 4, where the best results are highlighted in bold type. As shown in Table 4, we can find that TRKG outperforms ESRM-KG in terms of Recall but performs worse than ESRM-KG in terms of Precision and F1-measure. It is worth noting that Recall is more important than Precision in the text generation task. Moreover, TRKG-NoP is better than TRKG in terms of all the metrics. In other words, the keywords generation task enhances the performance of the recommendation task, but the recommendation task degrades the performance of the keyword generation task. We argue that this is mainly because although the predicted product is very similar to the purchased item, it will still be treated as negative samples in the recommendation task. The keywords generation task is subjected to this noise from the recommendation task, which makes the performance of keyword generation decreases.

Case Study on Keywords Generation. To gain an intuitive sense of the generated keywords by TRKG, we illustrate four examples



(a) D_1: HR (b) D_1: MRR (c) D_2: HR (d) D_2: MRR

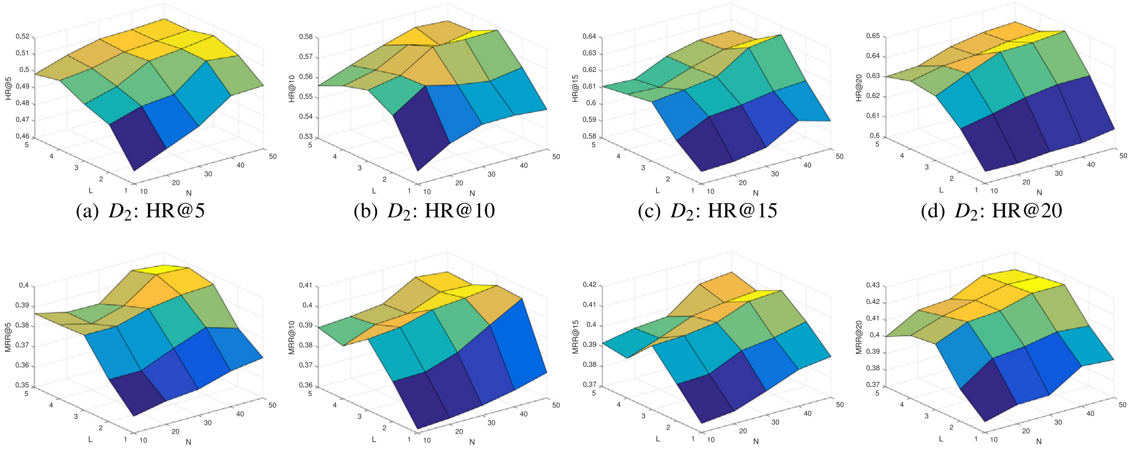
Fig. 8. Impact of weight η.



(a) D_1: HR@5 (b) D_1: HR@10 (c) D_1: HR@15 (d) D_1: HR@20

(e) D_1: MRR@5 (f) D_1: MRR@10 (g) D_1: MRR@15 (h) D_1: MRR@20

Fig. 9. Parameter analysis on dataset D_1.



(a) D_2: HR@5 (b) D_2: HR@10 (c) D_2: HR@15 (d) D_2: HR@20

(e) D_2: MRR@5 (f) D_2: MRR@10 (g) D_2: MRR@15 (h) D_2: MRR@20

Fig. 10. Parameter analysis on dataset D_2.

Table 4
Performance comparisons of travel recommendation on two datasets.

Methods	D_1									D_2								
	Rouge-1			Rouge-2			Rouge-L			Rouge-1			Rouge-2			Rouge-L		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
TRKG	.5764	.6659	.6168	.2290	.3150	.2732	.4452	.5057	.4765	.6071	.6484	.6290	.2469	.3038	.2815	.4728	.4891	.4878
ESRM-KG	**.6661**	.5947	.6206	**.2914**	.2905	.2866	**.5372**	.4805	.4878	**.6739**	.6014	**.6371**	**.2926**	.2928	.2879	**.5420**	.4845	.4906
TRKG-NoP	.5986	**.7067**	**.6253**	.2897	**.3307**	**.3016**	.4753	**.5598**	**.5048**	.6012	**.6910**	.6364	.2851	**.3203**	**.2937**	.4794	**.5376**	**.5011**

Note: P, R and F indicate Precision, Recall and F-measure, respectively.

in Table 5, where the left column is the keywords generated by TRKG, and the right column is the keywords corresponding to the purchased item. We have several interesting observations on these four examples, which are summarize below (1) The

Table 5
Examples of generated keywords.

TRKG	Ground truth
Tianjin, 2-days, package tour, Badaling Great Wall, the Palace Museum, the Old Summer Palace, Summer Palace, Beijing-Tianjin, Eastern China, the Great Wall, 3-days	Tianjin, 2-days, package tour, Badaling Great Wall, the Old Summer Palace, the Palace Museum, Summer Palace, capital, the Great Wall
self-guided tour, Samui island, Bangkok, 5-nights, 7-days, overseas travel, Thailand, round trip, Southeast Asia, Maldives, 2-nights	self-guided tour, Samui island, Bangkok, 5-nights, 7-days, romantic, dreamy, flight, multiple choices, Beijing, overseas travel, Thailand, Koh Samui, short-term travel
Beijing, Hangzhou, 4-days, 5-star, hotel, long-term travel, Science and Technology Museum, study tour, popular science, Chang'an Street, 4-star, self-funded, family trip, domestic tourism, package tour	Beijing, Hangzhou, domestic tourism, package tour, high-speed train, 4-days, summer vacation, family trip, lowest price, upgrade package, 5-star, hotel, Science and Technology Museum, Peking University, Tsinghua, North China, long-term travel
Fujian, Xiamen, Gulangyu Island, domestic tourism, long-term travel, Mid-Autumn Festival, exclusive product, package tour, 4-days, domestic flight, 2-nights, hotel, free time, distinctive, movie, theme hotel	Fujian, Xiamen, Shanghai, domestic tourism, package tour, Yunshuiyao, domestic flight, 4-days, 2-nights, sea-view room, Gulangyu Island, long-term travel, exclusive product

keywords generated by the TRKG model are of high quality. TRKG can capture the meaningful words and skip the nonsense words, *e.g.*, “romantic" and “dreamy" in Example 2. (2) Different from other text generation, word order in keywords generation is less important. For instance, from Example 3, we can find that the order of the generated keywords is inconsistent with that of keywords in the purchased item, but we can still understand the meaning of keywords. (3) Some contradictory keywords are generated by TRKG, such as “2-days" and “3-days" in Example 1, “5-nights" and “2-nights" in Example 2, which probably because TRKG captures diverse purchase intentions of the user but the user purchases only one item in the current session. (4) The TRKG model also generates some wrong words, such as “popular sciences" and “Chang'an Street" in Example 3, “movie" and “theme hotel" in Example 4. These keywords may also represent the user's purchase intention, but the user does not purchase related products in the end.

5. Discussion

Most session-based recommendation methods [6,7,46] learn a user's implicit intention by only taking the last behavior (*e.g.*, purchase) as the supervision signal. These studies cannot solve the low inclusiveness problem in recommendation scenarios, *i.e.*, many relevant products that satisfy the user's shopping intention are neglected by recommendation methods. Moreover, most existing approaches on session-based recommendation [9,47] merely consider the current session and regard it as a short sequence. They mainly use the recurrent neural network (RNN) or its variants and attention mechanism to characterize short-term user preferences. An obvious drawback of these methods is that they ignore long-term user preferences, *i.e.*, the effect of historical sessions on the current session.

In this paper, we regard keywords of product titles as a soft supervision signal, and combine them with the hard supervision signal by a multi-task learning mechanism. Keywords can explicitly reveal the user's intention. Different products may provide similar information for keywords generation. Therefore, even suffering from the penalty of the target product, the model may still recommend the satisfactory product with the help of multi-task learning. Meanwhile, the proposed method TRKG well adopts both long-term and short-term preferences to improve the performance of travel recommendation. TRKG considers the dynamics and evolutions of users' interests. The experimental results demonstrate that TRKG substantially improves the recommendation accuracy compared with the state-of-the-art methods.

Although TRKG achieves good results in the travel recommendation scenarios, it has some limitations. Travel recommendation may not enhance the performance of keywords generation. For the travel recommendation, even the recommended product is quite similar to the target product, it will still be considered as a negative example. Thus, keywords generation may suffer “noise" from the travel recommendation, resulting in a decrease in the performance of keyword generation.

6. Conclusion and future work

In this work, we regard keywords generation as an additional supervision signal in travel recommendation, and design the TRKG model, which jointly models travel recommendation and keywords generation by incorporating a multi-layer perceptron and a multi-source pointer-generator network. Specifically, we first learn user embedding from the long-term and the short-term behavior sequence with time intervals. Meanwhile, we also learn another user embedding from the clicked product titles in the keywords generation task. After that, we combine both user embedding into a unified user representation and train them together for better travel recommendation. To the best of our knowledge, this work is the first model to investigate travel recommendation via fusing keywords in product titles with long-term and short-term user click sequences provided by the OTA platform. Extensive experiments show that TRKG can significantly outperform state-of-the-art methods.

In the future, we will extend our work in the following aspects. First, we plan to consider multiple modalities of inputs (*e.g.*, image information of product) for further improve the accuracy of keyword generation. Second, we will consider multiple types of user behaviors (*e.g.*, reading the travel strategies, asking questions, and ordering) to capture more accurate user purchase intentions.

CRediT authorship contribution statement

Lei Chen: Conceptualization, Data curation, Software, Writing – original draft. **Jie Cao:** Resources Project, Administration, Funding Acquisition. **Guixiang Zhu:** Conceptualization, Methodology, Validation, Writing – review & editing. **Youquan Wang:** Methodology Formal Analysis, Supervision, Writing – review & editing. **Weichao Liang:** Visualization, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is partially supported by the National Key Research and Development Program of China under grant 2017YFD0401001, in part by the Key Program of National Natural Science Foundation of China under grant 92046026, in part by the National Natural Science Foundation of China under grant 72172057 and 71701089, in part by the Jiangsu Provincial Key Research and Development Program, China under grant BE2020001-3.

References

[1] J.G. Brida, D.M. Gómez, V. Segarra, On the empirical relationship between tourism and economic growth, Tour. Manag. 81 (2020) 104131.

[2] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: a survey, Decis. Support Syst. 74 (2015) 12–32.

[3] C. Wang, D.M. Blei, Collaborative topic modeling for recommending scientific articles, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011, pp. 448–456.

[4] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1235–1244.

[5] H. Wang, X. Shi, D.-Y. Yeung, Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks, Adv. Neural Inf. Process. Syst. 29 (2016) 415–423.

[6] B. Hidasi, A. Karatzoglou, Recurrent neural networks with top-k gains for session-based recommendations, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 843–852.

[7] Q. Liu, Y. Zeng, R. Mokhosi, H. Zhang, STAMP: Short-term attention/memory priority model for session-based recommendation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 1831–1839.

[8] C. Xu, P. Zhao, Y. Liu, J. Xu, V.S.S. S. Sheng, Z. Cui, X. Zhou, H. Xiong, Recurrent convolutional neural network for sequential recommendation, in: Proceedings of the 2019 World Wide Web Conference, 2019, pp. 3398–3404.

[9] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, T. Tan, Session-based recommendation with graph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 346–353.

[10] P. Zhao, A. Luo, Y. Liu, F. Zhuang, J. Xu, Z. Li, V.S. Sheng, X. Zhou, Where to go next: A spatio-temporal gated network for next poi recommendation, IEEE Trans. Knowl. Data Eng. (2020).

[11] K. Zhao, Y. Zhang, H. Yin, J. Wang, K. Zheng, X. Zhou, C. Xing, Discovering subsequence patterns for next POI recommendation, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020, pp. 3216–3222.

[12] G. Zhao, P. Lou, X. Qian, X. Hou, Personalized location recommendation by fusing sentimental and spatial context, Knowl.-Based Syst. 196 (2020) 105849.

[13] M. Shi, D. Shen, Y. Kou, T. Nie, G. Yu, Attentional memory network with correlation-based embedding for time-aware POI recommendation, Knowl.-Based Syst. 214 (2) (2021) 106747.

[14] R. He, J. McAuley, Fusing similarity models with markov chains for sparse sequential recommendation, in: 2016 IEEE 16th International Conference on Data Mining, IEEE, 2016, pp. 191–200.

[15] S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Factorizing personalized markov chains for next-basket recommendation, in: Proceedings of the 19th International Conference on World Wide Web, 2010, pp. 811–820.

[16] L. Liu, L. Wang, T. Lian, CaSe4SR: Using category sequence graph to augment session-based recommendation, Knowl.-Based Syst. (2020) 106558.

[17] X. Ma, J. Wu, S. Xue, J. Yang, Q.Z. Sheng, H. Xiong, A comprehensive survey on graph anomaly detection with deep learning, 2021, CoRR arXiv:2106.07178.

[18] Q. Liu, S. Wu, L. Wang, T. Tan, Predicting the next location: A recurrent model with spatial and temporal contexts, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2016, pp. 194–200.

[19] J. Manotumruksa, C. Macdonald, I. Ounis, A deep recurrent collaborative filtering framework for venue recommendation, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1429–1438.

[20] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, D. Jin, Deepmove: Predicting human mobility with attentional recurrent networks, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1459–1468.

[21] L. Gao, J. Wu, C. Zhou, Y. Hu, Collaborative dynamic sparse topic regression with user profile evolution for item recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, 2017.

[22] L. Chen, Z. Wu, J. Cao, G. Zhu, Y. Ge, Travel recommendation via fusing multi-auxiliary information into matrix factorization, ACM Trans. Intell. Syst. Technol. (TIST) 11 (2) (2020) 1–24.

[23] G. Zhu, Y. Wang, J. Cao, Z. Bu, S. Yang, W. Liang, J. Liu, Neural attentive travel package recommendation via exploiting long-term and short-term behaviors, Knowl.-Based Syst. 211 (2021) 106511.

[24] Y.-T. Wen, J. Yeo, W.-C. Peng, S.-W. Hwang, Efficient keyword-aware representative travel route recommendation, IEEE Trans. Knowl. Data Eng. 29 (8) (2017) 1639–1652.

[25] Y. Ge, H. Xiong, A. Tuzhilin, Q. Liu, Cost-aware collaborative filtering for travel tour recommendations, ACM Trans. Inf. Syst. 32 (1) (2014) 1–31.

[26] G. Zhu, J. Cao, C. Li, Z. Wu, A recommendation engine for travel products based on topic sequential patterns, Multimedia Tools Appl. 76 (16) (2017) 17595–17612.

[27] C.-Y. Liu, C. Zhou, J. Wu, Y. Hu, L. Guo, Social recommendation with an essential preference space, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.

[28] L. Gao, J. Wu, Z. Qiao, C. Zhou, H. Yang, Y. Hu, Collaborative social group influence for event recommendation, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 1941–1944.

[29] F. Liu, S. Xue, J. Wu, C. Zhou, W. Hu, C. Paris, S. Nepal, J. Yang, P.S. Yu, Deep learning for community detection: progress, challenges and opportunities, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020, pp. 4981–4987.

[30] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, Q.Z. Sheng, P.S. Yu, A comprehensive survey on community detection with deep learning, 2021, CoRR arXiv:2105.12584.

[31] Y. Lin, P. Ren, Z. Chen, Z. Ren, J. Ma, M. De Rijke, Explainable outfit recommendation with joint outfit matching and comment generation, IEEE Trans. Knowl. Data Eng. (2019).

[32] N. Wang, H. Wang, Y. Jia, Y. Yin, Explainable recommendation via multi-task learning in opinionated text data, in: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, 2018, pp. 165–174.

[33] F. Lv, T. Jin, C. Yu, F. Sun, Q. Lin, K. Yang, W. Ng, SDM: Sequential deep matching model for online large-scale recommender system, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2635–2643.

[34] C. Yang, L. Miao, B. Jiang, D. Li, D. Cao, Gated and attentive neural collaborative filtering for user generated list recommendation, Knowl.-Based Syst. 187 (2020) 104839.

[35] M. Ruocco, O.S.L.l. Skrede, H. Langseth, Inter-session modeling for session-based recommendation, in: Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems, 2017, pp. 24–31.

[36] Z. Hu, Y. Dong, K. Wang, Y. Sun, Heterogeneous graph transformer, in: Proceedings of the Web Conference 2020, 2020, pp. 2704–2710.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[38] H. Wu, H. Zhang, X. Zhang, W. Sun, B. Zheng, Y. Jiang, DeepDualMapper: A gated fusion network for automatic map extraction using aerial images and trajectories, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01, 2020, pp. 1037–1045.

[39] Y. Liu, Z. Ren, W.-N. Zhang, W. Che, T. Liu, D. Yin, Keywords generation improves e-commerce session-based recommendation, in: Proceedings of the 2020 International Conference on World Wide Web, 2020, pp. 1604–1614.

[40] A. See, P.J. Liu, C.D. Manning, Get to the point: Summarization with pointer-generator networks, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1073–1083.

[41] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, E.H. Chi, Modeling task relationships in multi-task learning with multi-gate mixture-of-experts, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 1930–1939.

[42] P. Avinesh, Y. Ren, C.M. Meyer, J. Chan, Z. Bao, M. Sanderson, J3R: Joint multi-task learning of ratings and review summaries for explainable recommendation, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2019, pp. 339–355.

[43] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, 2004, pp. 74–81.

[44] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, 2001, pp. 285–295.

[45] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, UAI (2012).

[46] V. Bogina, T. Kuflik, Incorporating dwell time in session-based recommendations with recurrent neural networks, in: RecTemp@ RecSys, 2017, pp. 57–59.

[47] Y. Zhu, Q. Lin, H. Lu, K. Shi, P. Qiu, Z. Niu, Recommending scientific paper via heterogeneous knowledge embedding based attentive recurrent neural networks, Knowl.-Based Syst. 215 (2021) 106744.