Tell Me Where to Go Next: Improving POI Recommendation via Conversation

Changheng Li[1], Yongjing Hao[1], Pengpeng Zhao[1(✉)], Fuzhen Zhuang[2,3], Yanchi Liu[4], and Victor S. Sheng[5]

[1] School of Computer Science and Technology, Soochow University, Suzhou 215006, China
{chlijq,yjhaozb}@stu.suda.edu.cn,ppzhao@suda.edu.cn

[2] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China

[3] Beijing Advanced Innovation Center for Imaging Theory and Technology, Academy for Multidisciplinary Studies, Capital Normal University, Beijing 100048, China
zhuangfuzhen@ict.ac.cn

[4] Management Science and Information Systems, Rutgers University, Piscataway, NJ, USA
yanchi.liu@rutgers.edu

[5] Department of Computer Science, University of Central Arkansas, Conway, USA
Victor.Sheng@ttu.edu

Abstract. Next Point-of-Interest (POI) recommendation estimates user preference on POIs according to past check-in history, suffering from the intrinsic limitation of obtaining dynamic user preferences. Conversational Recommendation System (CRS), which can collect dynamic user preferences through conversation, brings a solution to the above limitation. However, none of the existing CRS methods consider the spatio-temporal factors in the action selection phase, which are essential for POI conversational recommendation. In this paper, we propose a new Spatio-Temporal Conversational Recommendation System (STCRS) to fuse the spatio-temporal and dialogue information for next POI recommendation. Specifically, STCRS first learns the spatio-temporal information in the user's check-in history. Then reinforcement learning is used to decide which action (asking for an attribute or recommending POIs) to take at the next turn to achieve successful POI recommendation within as few turns as possible. Finally, our extensive experiments on two real-world datasets demonstrate significant improvements over the state-of-the-art methods.

Keywords: Conversational recommendation · Point-of-interest · Self-attention

C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12683, pp. 211–227, 2021.
https://doi.org/10.1007/978-3-030-73200-4_14

1 Introduction

Next Point-of-Interest (POI) recommendation systems are emerging as an essential means of facilitating user's information seeking in many scenarios, like Restaurants and Food (e.g., Yelp) and travel (e.g., Trip Advisor). However, existing methods cannot communicate with users and can only obtain passive feedback from users in the process of POI recommendations since they solely infer user preference on POIs from the historical spatio-temporal check-ins. Users can not actively express his/her immediate preferences, which are often drifting with time. For instance, a user may not be interested in the Great Wall initially, but once he/she happens to watch a video about it, he/she may like it and then become interested in POIs nearby. Such limitation makes it hard to obtain dynamic user preferences, preventing the system from providing accurate POI recommendation.

The Conversational Recommendation System (CRS), which is a recently emerging research topic, brings a solution to the limitation mentioned above. It allows a recommendation system to dynamically obtain user preferences through dialogue and make recommendations appropriately. As the conversational recommendation system became a hot topic, the community began to make great efforts to explore its various settings. Li et al. [8] recommended movies by focusing on natural language understanding and generation. Liao et al. [9] built a multi-modal dialogue systems which can capture rich semantics in the visual modality such as product images. Sun and Zhang [13] get the user attribute preferences by analyzing user utterances and feed them into a policy network. But it only handles single-round recommendation and does not consider the interaction between Conversational Component and Recommend Component. The single-round recommendation ends the conversation after only one recommendation and will not recommend again if the recommendation fails. In contrast, Lei et al. [6] proposed a method on multi-round recommendation setting and considered the interaction between the Conversational Component and the Recommend Component. Multi-round conversational recommendation will continue to ask or recommend after the recommendation is rejected, until the maximum conversation round is reached. Lei et al. [7] proposed a graph-based CRS, which reduces the space of candidate attributes and items by introducing the graph structure.

However, none of the existing methods consider POI conversational recommendation, where spatio-temporal information is essential. Integrating spatio-temporal information can benefit POI conversational recommendation, significantly reducing candidate attribute and item space and hence shortening interaction turns.

To this end, in this paper, we propose a novel POI conversational recommendation framework called Spatio-Temporal Conversational Recommendation System (STCRS), in which an agent can assist users in finding POIs interactively. The agent contains two components, i.e., Spatio-Temporal POI Recommendation module and Spatio-Temporal Policy Network module. Specifically, the Spatio-Temporal POI Recommendation module performs POI prediction via modeling the user's spatio-temporal sequential check-ins and immediate preference confirmed by the user in conversation; the Spatio-Temporal Policy Network module decides which action to take based on state vector with spatio-temporal information. The action may be to ask the user if they like a certain attribute or recommend a ranked list of POIs. We train a policy network with reinforcement learning, maximizing the reward based on the conversation state which integrates spatio-temporal information. Inspired by Lei et al. [6], recommendations will be updated with users' online feedback.

To validate the effectiveness of STCRS, experiments are conducted on CA[1] and Ant Financial[2] datasets. Performances are compared with state-of-the-art CRS methods [6], which also use the information of user, POI, and attribute but do not use spatio-temporal information. We analyze each method's properties under different settings, including binary question and enumerated question. The experimental results show that STCRS outperforms the existing approaches.

In summary, our contributions are listed as follows:

- To the best of our knowledge, this is the first work to investigate POI conversational recommendation systems.
- We propose the STCRS framework to integrate spatio-temporal information into a conversational recommendation system for next POI recommendation.
- We conduct extensive experiments on two real-world datasets. Our experimental results show the superiority and effectiveness of STCRS, comparing with the state-of-the-art methods via comprehensive analysis.

2 Related Work

In this section, we give a brief review of next POI recommendation and conversational recommendation system.

2.1 Next POI Recommendation

The goal of the next POI Recommendation is to recommend a ranked list for user based on his/her historical check-ins. The next POI to be visited by the user should have a higher ranking. Cheng et al. [2] combines the localized region constraints with personalized Markov chains and predicts next POI through the transition probability. However, Markov chains cannot learn complex transitions between POIs. With the development of deep learning, researchers began to try to use neural networks to solve this problem. Feng et al. [4] captures the user transition patterns by using a metric embedding method to embed users and POIs into the same latent space. Xie et al. [15] embeds relationship among POI, Region, Time and Word into a shared low dimensional space. And it uses the linear combination of inner products to compute the score of POIs. Zhang et al. [17] leverages the temporal dependency in user's check-in sequence to

[1] https://github.com/WeiqiXu/FoursquareData.

[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=58.

model user's dynamic preferences. With the continuous development of research, methods of extending existing neural networks have also been proposed. Liu et al. [10] is the first method to model spatio-temporal information for the next POI recommendation, it models spatio-temporal information by replacing the simple transition matrices of RNN with spatio-temporal transition matrices. Zhao et al. [18] is proposed to incorporate spatio-temporal gates to learn the spatio-temporal information of check-in sequences. Xu et al. [16] and Luo et al. [11] use self-attention network for recommendation. J. Ni et al. [12] use a decay function and self-attention block to model time and distance intervals for next POI recommendation. Although significant progress has been made, the intrinsic limitation of obtaining dynamic user preference cannot be avoided.

2.2 Conversational Recommendation

Conversational recommendation system makes it possible to obtain user explicit feedback. Users can interact with CRS using natural language. There are different settings for various problems.

Li et al. [8] built a system which recommend movies through sentiment analysis and movies mentioned in the dialogue. But it only uses mentioned movies for recommender and recommender cannot help generate better dialogue. Chen et al. [1] solved the above two defects by introducing knowledge graph. [9] built a multi-modal dialogue systems which can understand the user's intention more clearly by using the visual information. Zhang [13] built a single-round CRS and used supervised learning and reinforcement learning to train a policy network. But it does not consider the interaction between Conversational Component and Recommend Component. Subsequently, Lei et al. [6] proposed a method on multi-round recommendation setting and consider the interaction between the Conversational Component and the Recommend Component. Recently, Lei et al. [7] proposed a graph-based CRS, the policy network of it has a smaller action space, so it does not require pre-training as adopted in [6,13]. Zhou et al. [19] improved the conversational recommendation by integrating both item-based preference sequence and attribute-based preference sequence. But they don't consider problem about the spatio-temporal factors.

We believe that obtaining a user's dynamic preference and recommend POIs upon attribute feedback is the key to POI conversational recommender system. However, none of the existing CRS works have considered the spatio-temporal information into the conversation. We believe that utilizing the spatio-temporal information will help decrease interaction turns and accurate next POI recommendations.

3 Preliminaries

In this section, we discuss how to integrate spatio-temporal and dialogue information to improve the effect of next POI recommendation. Our framework has

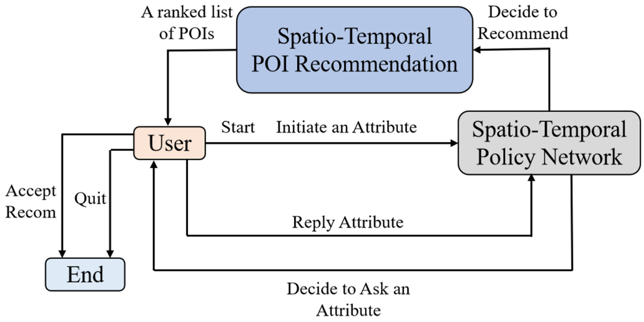

Fig. 1. The workflow of our Spatio-Temporal Conversational Recommendation System.

two components: the Spatio-Temporal POI Recommendation module and the Spatio-Temporal Policy Network module.

We now introduce the notation uesd to formalize our setting. Let u and U denote a user and the user set. v and V denote a POI and the POI set. Each POI v is associated with a set of attributes P_v, such as "School","Hotel", or "Theme Park" for businesses in CA. p and P denote a specific attribute and all attribute. The check-in records of each user are sorted into a sequence $L_u = (v_1^u, v_2^u, ..., v_{|L|}^u)$ by time. And each check-in record v_i^u is associated with its timestamp t_i^u and its geographic coordinates s_i^u.

STCRS aims to recommend POIs that users are interested in within as few turns as possible. The system asks questions based on L_u and the current time to determine the u's current preferences and makes personalized POI recommendations when appropriate to complete the conversation successfully.

Figure1 presents the workflow of our proposed STCRS framework. A CRS session is started with u's current preference attributes p^0, then the CRS removes POIs that do not contain attribute p^0 from the current POI candidate set V_{cand}. Then in each turn t, the STCRS needs to choose an action based on L_u: *recommend* or *ask*.

- If action is *recommend*, the Spatio-Temporal POI Recommendation module will rank the V_{cand}, and recommend a list. If the list contains the POI which user is interested in, user will accept the recommendation and this session ended successfully. Otherwise, user will reject the recommendation.
- If action is *ask*. User needs to clearly express whether he/she prefers the attribute selected by the Spatio-Temporal Policy Network (where the asked attribute is denoted as $p^t \in P$). If the feedback is positive, STCRS will keep the POIs which contain attribute p^t in the V_{cand} and add p^t into P_u (The user preferred attribute set determined through the dialogue). Otherwise, we only remove the POIs which contain attribute p^t.

A CRS session will continue as above until the POI is successfully recommended or the maximum number of turns is reached.

4 Proposed Methods

In this section, we will describe each module in our framework in detail. Figure 2 shows the architecture of STCRS, which contains the Spatio-Temporal POI Recommendation module and the Spatio-Temporal Policy Network module. STCRS executes a loop many times to complete the CRS session. This loop has the following steps:

First, the Spatio-Temporal POI Recommendation module models spatio-temporal information of L_u and scores candidate POIs. Second, the spatio-temporal information learned from L_u will be transformed into four Spatio-Temporal States (s_{st-ent}, s_{st-pre}, $s_{spatial}$ and $s_{temporal}$) and fed into the Spatio-Temporal Policy Network module. Third, the Spatio-Temporal Policy Network module decides which action (asking an attribute or recommending a POI list) to take based on the spatio-temporal and dialogue information. Then, the user simulator generates a reply to the action. Finally, using the information generated by the user simulator's reply to update the corresponding module. Specifically, if the user rejects the recommendation, we use the rejected POIs as negative sample to update the Spatio-Temporal POI Recommendation module. If the user gives feedback on attribute, we update the candidate set and dialogue state.

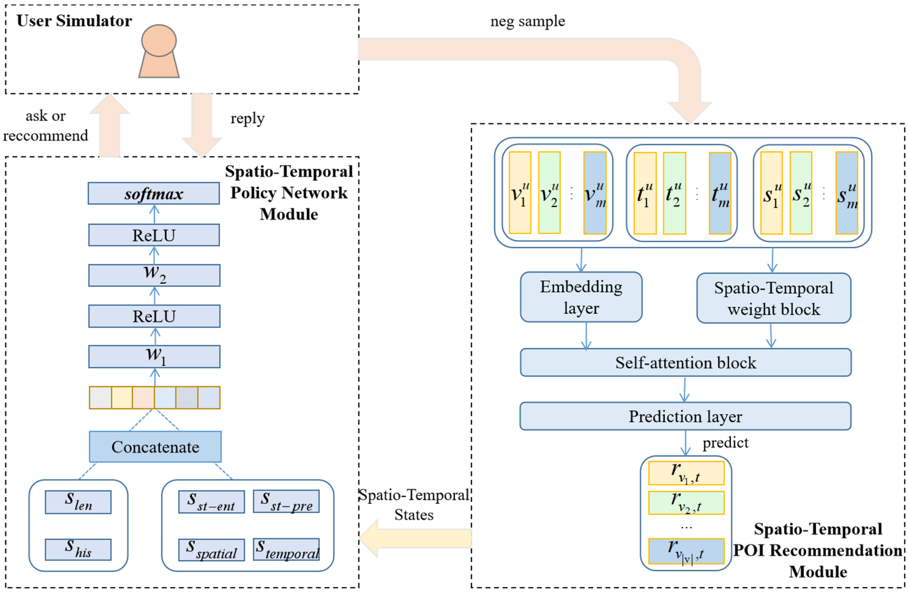

Fig. 2. The architecture of our proposed STCRS. The Spatio-Temporal POI Recommendation Module scores the candidate POIs and offers Spatio-Temporal States to the Spatio-Temporal Policy Network Module. The Spatio-Temporal Policy Network Module decides whether to ask or recommend at each turn. The user simulator responds to questions from the Spatio-Temporal Policy Network Module or recommendations from the Spatio-Temporal POI Recommendation Module. The reply of it is used to update the corresponding module.

4.1 Spatio-Temporal POI Recommendation Module

The goal of Spatio-Temporal POI Recommendation module is to calculate the scores of POIs that user may visit at next time step, given L_u.

To effectively merge spatio-temporal information in conversational recommendation, we use Spatio-Temporal Self-Attention Network (STSAN) to extract spatio-temporal information in L_u. STSAN consists of Embedding layer, Spatio-Temporal weight block, Self-Attention block and Prediction layer.

Embedding Layer: As the length of L_u is not fixed and the check-ins that are too early can not correctly reflect the u's current preferences, we only consider u's recent check-ins of a fixed length. Let m denote the fixed length. And we denote $\hat{L}_u = (v_1^u, v_2^u, ..., v_m^u)$ as the u's recent m check-ins. If the length of u's recent check-ins is less than m, we employ zero-padding to fill the left side of u's recent check-ins sequence until the sequence length is m. We create a POI embedding matrix $\mathbf{M} \in \mathbb{R}^{|V| \times d}$ to encode POI into a unique latent vector, where d is the latent dimension. And we create a positional matrix $\mathbf{P} \in \mathbb{R}^{m \times d}$ to encode m positional information in $\hat{L}_u$. As we mentioned above, each POI v in $\hat{L}_u$ has two embedding ($\mathbf{M}_v$ and $\mathbf{P}_i$, i is the position of v in $\hat{L}_u$). We add them to form the input matrix, the input matrix is defined as follows:

$$\mathbf{E} = \begin{bmatrix} \mathbf{M}_{v_1} + \mathbf{P}_1 \\ \mathbf{M}_{v_2} + \mathbf{P}_2 \\ \cdots \\ \mathbf{M}_{v_m} + \mathbf{P}_m \end{bmatrix} \tag{1}$$

Spatio-Temporal Weight Block: We calculate the temporal and spatial transition matrices $\mathbf{T}^u$ and $\mathbf{S}^u$ based on the temporal and spatial sequence associated with $\hat{L}_u$ (i.e., $(t_1^u, t_2^u, ..., t_m^u)$ and $(s_1^u, s_2^u, ..., s_m^u)$).

$$\mathbf{T}_{ij}^u = \begin{cases} \Delta t_{ij}^u, & i \geqslant j, \\ 0, & i < j, \end{cases} \tag{2}$$

$$\mathbf{S}_{ij}^u = \begin{cases} \Delta d_{ij}^u, & i \geqslant j, \\ 0, & i < j, \end{cases} \tag{3}$$

where Δt_{ij}^u and Δd_{ij}^u are the time and distance intervals between check-in v_i^u and check-in v_j^u. A decay function is used to convert Δt_{ij}^u and Δd_{ij}^u into a weight. Therefore the temporal and spatial weight matrix $\hat{\mathbf{T}}^u$ and $\hat{\mathbf{S}}^u$ can be calculated as follows:

$$\hat{\mathbf{T}}_{ij}^u = \begin{cases} g(\Delta t_{ij}^u), & i \geqslant j, \\ 0, & i < j, \end{cases} \tag{4}$$

$$\hat{\mathbf{S}}_{ij}^u = \begin{cases} g(\Delta d_{ij}^u), & i \geqslant j, \\ 0, & i < j, \end{cases} \tag{5}$$

where $g(x) = 1/log(e + x)$., we use $g(x)$ as the decay function. We utilize a weight factor ρ to balance the influence of the temporal and spatial information. We use the weight factor ρ as follows:

$$\mathbf{H} = \rho \cdot \hat{\mathbf{T}} + (1 - \rho) \cdot \hat{\mathbf{S}}, \tag{6}$$

where $0 < \rho < 1$. Finally we use linear transformation on $\mathbf{H}$:

$$\hat{\mathbf{H}} = \mathbf{WH} + \mathbf{b}, \tag{7}$$

Self-attention Block: We convert the input matrix $\mathbf{E}$ obtained from the embedding layer as follow:

$$\mathbf{W}_{SA} = softmax(\frac{\mathbf{EW}^Q(\mathbf{EW}^K)^T}{\sqrt{d}}), \tag{8}$$

$$\mathbf{F} = \hat{\mathbf{H}}\mathbf{W}_{SA}(\mathbf{EW}^V), \tag{9}$$

where $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V \in \mathbb{R}^{d \times d}$ are used to project $\mathbf{E}$ into three matrices and $\hat{\mathbf{H}}$ is the output of the Spatio-Temporal weight block. We use layer normalization and residual connection on $\mathbf{F}$. Finally, we feed $\hat{\mathbf{F}}$ into a two-layer fully-connected layer.

$$\hat{\mathbf{F}} = \mathbf{E} + LayerNorm(\mathbf{F}), \tag{10}$$

$$\mathbf{O} = ReLU(\hat{\mathbf{F}}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2. \tag{11}$$

Prediction Layer: We calculate the dot product of $\mathbf{M}_{v_i}$ and $\mathbf{O}_t$ to get a score $r_{v_i,t}$ of v_i. $\mathbf{O}_t$ is the t-th line of $\mathbf{O}$. The higher the score of v_i, the more likely v_i will be visited.

Network Training: We take the last POI of each user sequence L_u in the training set as a positive sample v^u_{pos} and perform negative sampling to form the positive and negative sample pairs (v^u_{pos}, v^u_{neg}). We optimize the network according to the following formula:

$$loss = -\sum_{u} \sum_{(v^u_{pos}, v^u_{neg})} [log(\sigma(r_{v^u_{pos},t})) + log(1 - \sigma(r_{v^u_{neg},t}))]. \tag{12}$$

4.2 Spatio-Temporal Policy Network Module

The goal of Spatio-Temporal Policy Network module is to learn a policy which selects an action based on the dialogue state at each turn, in order to accomplish successful POI recommendation within as few conversation rounds as possible.

We use the policy network in deep reinforcement learning as our Spatio-Temporal Policy Network. For more introduction about reinforcement learning, please see [14]. The structure of the policy network is shown in the lower left part in Fig. 2. The basic component of reinforcement learning is state, action, reward and policy.

State: The state s_t is the description of the current conversation session. It is composed of s_{st-ent}, $s_{st\text{-}pre}$, s_{his}, s_{len}, $s_{spatial}$ and $s_{temporal}$. $s_{st\text{-}ent}$, $s_{st\text{-}pre}$, $s_{spatial}$ and $s_{temporal}$ come from the Spatio-Temporal POI Recommendation module and contain spatio-temporal information. We believe that such a design can effectively use spatio-temporal information.

- $s_{st\text{-}ent}$**:** This vector encodes the attribute entropy of the top-k POIs in V_{cand}. The intuition is that using u's historical spatio-temporal information to score V_{cand} and obtaining attribute entropy of the top-k POIs. The attribute with the larger entropy is asked, the more information we can get.
- $s_{st\text{-}pre}$**:** We treat the POI score as the POI's attribute score. And we use the score of top-k POIs in V_{cand} to calculate the score of attributes. For CA dataset, we calculate the average score of each attribute; For Ant Financial dataset, we calculate the score of each first-level attribute by dividing the total score of second-level attributes by the number of second-level attributes that are not repeated. We use tanh to transform the attribute score. The intuition is that the attributes with higher scores are more likely to be user preference attributes.
- s_{his}**:** This vector records the dialogue history. Its size is the number of maximum turns. Specifically, we use -1 to represent recommendation is rejected, 0 to present u dislikes the attribute we asking, 1 to present u gives positive feedback to the attribute we asking, 2 to present make a successful POI recommendation.
- s_{len}**:** This vector is the binary code of the length of V_{cand}. The shorter the length of V_{cand}, the greater the probability of successful recommendation.
- $s_{spatial}$**:** We denote the average geographic coordinates of $\hat{L}_u$ as $mean_{pos}^{his}$, the average geographic coordinates of the top-k POIs in V_{cand} as $mean_{pos}^{cand}$, the variance of the geographic coordinates of $\hat{L}_u$ as var_{pos}^{his}, and the variance of the geographic coordinates of the top-k POIs in V_{cand} as var_{pos}^{cand}. Using $mean_{pos}^{his} \bigoplus \mathrm{mean}_{pos}^{cand} \bigoplus \tanh(var_{pos}^{his}/var_{pos}^{cand})$ as spatial information of the current dialogue round. $\bigoplus$ is used for vector concatenate. The intuition is that if the spatial information of V_{cand} and $\hat{L}_u$ is similar , a recommendation should be made.
- $s_{temporal}$**:** We assume that t_i^u is the time u seeked POI recommendations. According to the time period (morning or afternoon) when u visited v_i^u, count the attribute ratio of POIs during this time period in $\hat{L}_u$. We denote this ratio as f_{his}. And we count the attribute ratio of the top-k POIs in V_{cand}. We denote this ratio as f_{cand}. The cosine similarity of f_{his} and f_{cand} is denoted as cos_{his_cand}. Using $f_{his} \bigoplus f_{cand} \bigoplus cos_{his\_cand}$ as temporal information of the current dialogue round. The intuition is that if one attribute is visited multiple times by u in a period of time, u is more likely to prefer this attribute in the same period of time, and we should ask questions about it.

Action: The Spatio-Temporal Policy Network module needs to select an action a_t at time step t. Two kinds of actions can be selected. One is to make a POI recommendation. The other is to ask an attribute that the user may prefer in this CRS session. So the space of action is the number of attribute $|P| + 1$.

Reward: The reward follows [6], (1) r_{rec_suc}, we give a strongly positive reward when the POI recommendation is successful, (2) r_{rec_unsuc}, a slightly negative reward is given when the recommendation is rejected, (3) r_{ask_suc}, a strongly positive reward when the user accept the asked attribute, (4) r_{ask_unsuc}, a slightly negative reward, (5) r_{fail}, a strongly negative reward if the user quits the conversation, (6) r_{prev}, a slightly negative reward to avoid overly length conversations.

Policy: We denote the policy network as $\pi(a_t|s_t)$. It maps the current conversation state s_t into the action space. The Spatio-Temporal Policy Network module selects an action a_t according to the result of output layer and gets an immediate reward r_t at each turn. The goal of the Spatio-Temporal Policy Network module is to maximize the episodic expected reward of a CRS session. Policy Network will select high-value action after trial and error. The policy gradient method is used to optimize the network, formulated as follows:

$$\theta \leftarrow \theta - \alpha \nabla log \pi_\theta(a_t|s_t) R_t, \tag{13}$$

$$R_t = \sum\nolimits_{t'=t}^{T} \gamma^{T-t'} r_{t'}, \tag{14}$$

where θ and α are the parameters and learning rate of policy network respectively, γ is the discount factor. Note that if θ is initialized randomly, the learning can converge slowly or fail. To address this issue, we follow [6] to conduct the rule-based pre-training.

5 Experiments

In this section, we conduct experiments to evaluate our proposed STCRS framework on two real-world datasets. Our experiment are guided by the following Research Questions(RQs).

- **RQ1.** How does STCRS perform compared to the state-of-the-art methods for conversational recommendation?
- **RQ2.** Is the design of the state vector effectively utilize the spatio-temporal information to complete the POI conversational recommendation?
- **RQ3.** How does the hyper-parameters affect the method performance (e.g., the discount factor γ and the learning rate α of the Spatio-Temporal Policy Network module)?

Table 1. Datasets statistics.

Dataset	CA	Ant Financial
#Users	2389	2481
#POIs	9144	1481
#Check-ins	93598	26808
#Attributes	34	142

5.1 Settings

5.1.1 Datasets

For better comparison and make POI conversational recommendation, we conduct experiments on two datasets: CA and Ant Financial. CA is a Foursquare dataset from users whose homes are in California, collected from January 2010 to February 2011 and used in [5]. Ant Financial is an Internet financial services company in China and the dataset provides shop information and Alipay user's payment log and users' browsing log from 07.01.2015 to 10.31.2016 (except 2015.12.12). Follow [6], we remove the duplicated user-POI check-ins in our datasets and only keep the first check-in. The statistics of the two datasets are summarized in Table 1. We sort the check-ins of each user by time and take the early 70% of user's check-ins as the training data, the last 10% as the testing data, the remaining 20% as the validation data.

For better comparison, we follow [6] to conduct experiments on CA for binary question scenario and Ant Financial for enumerated question scenario. In binary question scenario, the user answers yes or no when the user is asked a question about attribute. In enumerated question scenario, we build a 2-layer taxonomy which includes 15 first-layer categories and 142 second-layer categories. For example, The first-level category "city" includes 88 second-level categories, and each second-level category represents a specific city.

5.1.2 User Simulator

Conversational recommendation is a process of continuous interaction with users. CRS needs to interact with user to obtain the dynamic user preferences and make POI recommendations. But CRS is too expensive to be applied to real users to train from scratch. To solve these problems, we follow [6] to build a user simulator. When the user simulator simulates one conversation session for a user-POI (u, v) check-in, it restricts u to only prefer the attributes in P_v and only accepts the recommendation containing v.

5.1.3 Training Details

We set the length of recommendation list as 10, maximum turn as 15 on CA dataset, and maximum turn as 6 on Ant Financial dataset. Maximum turn of CA follows [6] and the standard for setting maximum turn of Ant Financial is that the highest success rate of Max Entropy just exceeds 90%. Following [6,13], we perform two-stage training: (1) An offline training for Recommend Component. We use the training set to optimize Spatio-Temporal POI Recommendation module (Eq. (12)). The goal is to assign higher score to the check-in POI for each users. All hyper-parameters are tuned on the validation set: For CA dataset, the batch size is set as 64, the learning rate is 0.0001, the m is 10, the embedding size is 40, the dropout rate is 0.5, and the size of block and head is 1; For Ant Financial dataset, the learning rate is set to 0.001, the m is 9, the embedding size is 110, and the other are the same as the CA dataset. (2) An online training for Conversational Component. We use a user simulator to interact with STCRS to train the Spatio-Temporal Policy Network module using the validation set. The k is set as 100. The rewards are as follow: $r_{prev} = -0.01$, $r_{rec\_suc} = 1 + r_{prev}$, $r_{fail} = -0.3$, $r_{ask\_suc} = 0.1 + r_{prev}$, $r_{ask\_unsuc} = r_{prev}$. On CA dataset, $r_{rec\_unsuc} = r_{prev}$; On Ant Financial dataset, $r_{rec\_unsuc} = -0.1$. We use the AdamOptimizer to optimize the policy network.

5.1.4 Evaluation Metrics

To evaluation follows [6]. We use Success Rate (SR@t) [13] and Average Turns (AT) which is average conversation rounds for successful POI recommendations to measure the ratio of successful POI conversational recommendation and the effectiveness of conversation. Larger SR denotes better performance and shorter AT denotes more efficient conversation. In the offline training of Recommend Component, we use the NDCG@10 and HR@10 to find the best Recommend Component.

5.2 Baselines

To emphasize the importance of spatial-temporal information and the fairness of comparison, we compared our framework with the following CRS methods.

- **Max Entropy.** A ruled-based method. Generating random numbers based on the current number of candidate items to decide whether to ask or recommend. When asking a question, it chooses an attribute which has not been asked and has the maximum entropy in the candidate set. We use it for pre-training. Details can be found in [6].
- **Abs Greedy** [3]**.** This method only have a recommendation component. It only recommends items and updates itself when the recommendation is rejected, until it recommends the correct item or failed after reach the maximum number of turn.
- **CRM** [13]**.** This is a CRS method using reinforcement learning. It uses belief tracker to analyze the preferences expressed by user utterances. The output of belief tracker is fed into policy network for deciding which action should take at next step. We follow [6] to adapt it to the multi-round conversational recommendation scenario.
- **EAR** [6]**.** This method is based on multi-round conversational recommendation setting and emphasizes the interaction between conversation component

and recommendation component. Using BPR algorithm to update attribute-aware FM.

5.3 Performance Comparison (RQ1)

In this section, we compare our framework STCRS with four state-of-the-art baselines.

Table 2. Experimental results of STCRS and baselines (RQ1).

	CA				Ant Financial			
	SR@5	SR@10	SR@15	AT	SR@2	SR@4	SR@6	AT
Max Entropy	0.052	0.141	0.199	13.656	0.066	0.639	0.920	4.169
Abs Greedy	0.204	0.288	0.339	11.831	**0.381**	0.614	0.718	3.804
CRM	0.197	0.292	0.357	11.846	0.116	0.769	0.941	3.76
EAR	0.193	0.317	0.394	11.69	0.106	0.792	0.968	**3.673**
STCRS	**0.216**	**0.349**	**0.416**	**11.388**	0.058	**0.805**	**0.984**	3.879

Table 2 presents the statistics of method's performance. As can be clearly seen, our STCRS significantly outperforms the state-of-the-art baselines on various setting. This proves our hypothesis that considering spatio-temporal information in conversational recommendation can better make POI conversational recommendation. In order to better show the performance of STCRS and compare it with other baselines, we analyze the performance of STCRS in each round in Fig. 3.

Figure3 shows the Success Rate* (SR*) @t at different turns (t = 1 to 15 on CA and t = 1 to 6 on Ant Financial). SR* denotes the comparison of each method against the strongest baseline EAR, indicated as y = 0 in the figure.

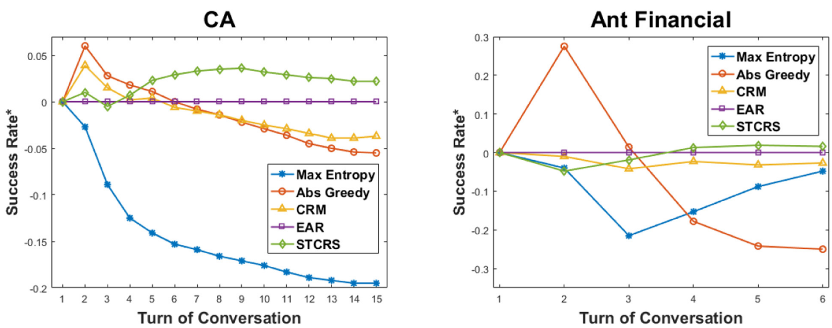

Fig. 3. Success Rate* of compared methods at different conversation turns on CA and Ant Financial (RQ1).

There is a common trend in two datasets. The performance of STCRS is weak at the beginning of a conversation, but it starts to grow and reach a stable state in the subsequent turns. The poor performance at first and the significant improvement afterward illustrate that it is difficult to successfully recommend the POI that the user is interested in by only using the Spatio-Temporal POI Recommendation module to extract the spatial-temporal information from the user's historical check-in records at the beginning of conversation, but the user's dynamic preferences obtained through the conversation can effectively improve the success rate of the POI recommendation. The subsequent excellent performance shows that introducing user's spatio-temporal information in the conversational recommendation can help the Spatio-Temporal Policy Network module choose the attributes that users are more likely to prefer to ask questions and improve the success rate of the POI conversational recommendation.

The performance of Max Entropy on Ant Financial gradually improve in turns 4–6. But the performance on CA continues to decline. The key reasons are that the POI in Ant Financial has more attribute information and the setting of Ant Financial is to ask enumerated question. User's response will sharply shrink the candidate POIs in this setting.

Comparing with EAR, STCRS has a greater advantage on CA. As we mentioned above, the POI in CA has fewer attributes and the setting of CA is to ask binary questions, so the performance of STCRS shows that the spatio-temporal information is helpful to choose the right attributes to ask questions and better complete POI conversational recommendations.

5.4 Ablation Studies on State Vector (RQ2)

In order to explore the effect of each part of the state vector, we remove or replace these parts one by one and check the change. Table 3 presents the statistics of our framework's performance on two conversation scenarios (binary questions and enumerated questions). s_{ent} and s_{pre} is the attribute entropy and attribute preference of all POIs in V_{cand}.

As can be clearly see, $s_{st\text{-}ent}$ is the most important part on two conversation scenarios. If we remove $s_{st\text{-}ent}$, although it obtains improvement at the beginning of conversation, SR@6 and SR@15 greatly suffers, due to the system makes POI recommendation before obtaining enough information. We replace $s_{st\text{-}ent}$ and $s_{st\text{-}pre}$ with s_{ent} and s_{pre}. Except for SR@2, the performance of other indicators have a decline. This shows that it is necessary to introduce spatio-temporal information in attribute entropy and attribute preference. Apart from $s_{st\text{-}ent}$ and $s_{st\text{-}pre}$, $s_{spatial}$ and $s_{temporal}$ also have a positive contribution to our framework. The spatio-temporal information is more important for CA (binary question). A reasonable explanation is that POI in CA has fewer attributes than POI in Ant Financial, so spatio-temporal information is more important to select attributes which users prefer at the current time.

Table 3. Performance of removing or replacing one component of state vector from STCRS (RQ2).

	CA				Ant Financial			
	SR@5	SR@10	SR@15	AT	SR@2	SR@4	SR@6	AT
-$s_{st\text{-}ent}$	0.218	0.296	0.35	11.664	**0.182**	0.629	0.836	4.024
-$s_{st\text{-}pre}$	0.229	0.344	0.404	11.352	0.088	0.779	0.976	3.911
-$s_{st\text{-}ent} + s_{ent}$	0.184	0.271	0.326	11.724	0.051	0.789	0.98	3.901
-$s_{st\text{-}pre} + s_{pre}$	**0.242**	0.342	0.4	**11.253**	0.069	0.778	0.973	3.921
-$s_{spatial}$	0.235	0.332	0.393	11.354	0.051	0.794	0.976	3.89
-$s_{temporal}$	0.201	0.295	0.352	11.839	0.074	0.794	0.977	**3.855**
STCRS	0.216	**0.349**	**0.416**	11.388	0.057	**0.813**	**0.982**	3.86

5.5 Sensitivity Analyses of Hyper-parameters (RQ3)

In this section, we explore the influences of the discount factor γ and the learning rate α in STCRS.

Influence of the Learning Rate α. We first fix optimizer of the policy network is Adam, γ is 0.6 on CA, γ is 0.8 on Ant Financial and vary α to explore the influence of α. We choose α in {0.0002, 0.0005, 0.001, 0.002}. As is shown in Table 4, when α is 0.0005, the performance of STCRS is best on two datasets. Although some indicators are better when α is set as other values, 0.0005 is the best value for the overall effect.

Influence of the Discount Factor γ. To explore the influence of γ, we fix the learning rate α as 0.001. And we search γ from {0.6, 0.7, 0.8, 0.9, 0.95, 0.99}.

Table 4. Influence of the learning rate α and the discount factor γ (RQ3).

		CA				Ant Financial			
		SR@5	SR@10	SR@15	AT	SR@2	SR@4	SR@6	AT
α	0.0002	**0.233**	0.333	0.404	11.351	0.051	0.794	0.978	3.883
	0.0005	0.228	**0.356**	**0.418**	**11.266**	0.051	**0.814**	**0.983**	3.923
	0.001	0.216	0.349	0.416	11.388	0.057	0.813	0.982	3.89
	0.002	0.198	0.29	0.338	11.885	**0.091**	0.779	0.97	**3.882**
γ	0.6	0.216	0.349	0.416	11.388	0.056	0.788	0.976	**3.867**
	0.7	0.224	0.331	0.395	11.45	**0.061**	0.801	0.982	3.917
	0.8	**0.23**	0.34	0.404	**11.372**	0.057	**0.813**	0.982	3.89
	0.9	0.164	0.319	0.385	11.847	0.058	0.805	0.984	3.879
	0.95	0.204	**0.352**	0.414	11.437	0.048	0.787	0.981	3.89
	0.99	0.169	0.337	**0.429**	11.645	0.036	0.801	**0.985**	4.038

From Table 4, we can see that the best γ for SR@2 is 0.7, the best γ for SR@5, SR@4 and AT on CA is 0.8, the best γ for SR@10 is 0.95, and the best γ for SR@15 and SR@6 is 0.99. From the perspective of successfully completing as many POI conversational recommendations as possible, the best γ is 0.99.

6 Conclusion

In this paper, we proposed a novel framework Spatio-Temporal Conversational Recommendation System (STCRS). We employed the Spatio-Temporal Self-Attention Network to extract the spatio-temporal information of user's check-in history, and used reinforcement learning to train a policy network to make decision at each turn. The state vector was designed carefully, which can build a bridge between Spatio-Temporal POI Recommendation module and Spatio-Temporal Policy Network module for communication. To the best of our knowledge, STCRS is the first method to use the spatio-temporal and dialogue information for next POI recommendation. We compared the Success Rate and the Average Turns of STCRS with CRS methods, and our experimental results show the improvement of our framework.

Acknowledgements. This research was partially supported by NSFC (No. 61876117, 61876217, 61872258, 61728205), ESP of the State Key Laboratory of Software Development Environment, and PAPD of Jiangsu Higher Education Institutions.

References

1. Chen, Q., et al.: Towards knowledge-based recommender dialog system. arXiv preprint arXiv:1908.05391 (2019)
2. Cheng, C., Yang, H., Lyu, M.R., King, I.: Where you like to go next: successive point-of-interest recommendation. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 2605–2611 (2013)
3. Christakopoulou, K., Radlinski, F., Hofmann, K.: Towards conversational recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 815–824 (2016)
4. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M.: Personalized ranking metric embedding for next new poi recommendation. In: Proceedings of the 24th International Conference on Artificial Intelligence, pp. 2069–2075 (2015)
5. Gao, H., Tang, J., Liu, H.: gSCorr: modeling geo-social correlations for new check-ins on location-based social networks. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1582–1586 (2012)
6. Lei, W., et al.: Estimation-action-reflection: towards deep interaction between conversational and recommender systems. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 304–312 (2020)
7. Lei, W., et al.: Interactive path reasoning on graph for conversational recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2073–2083 (2020)

8. Li, R., Kahou, S.E., Schulz, H., Michalski, V., Charlin, L., Pal, C.: Towards deep conversational recommendations. In: Advances in Neural Information Processing Systems, pp. 9725–9735 (2018)
9. Liao, L., Ma, Y., He, X., Hong, R., Chua, T.-S.: Knowledge-aware multimodal dialogue systems. In: Proceedings of the 26th ACM International Conference on Multimedia, pp. 801–809 (2018)
10. Liu, Q., Wu, S., Wang, L., Tan, T.: Predicting the next location: a recurrent model with spatial and temporal contexts. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
11. Luo, A., et al.: Collaborative self-attention network for session-based recommendation
12. Ni, J., et al.: Spatio-temporal self-attention network for next poi recommendation. In: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, pp. 409–423 (2020)
13. Sun, Y., Zhang, Y.: Conversational recommender system. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 235–244 (2018)
14. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2018)
15. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based poi embedding for location-based recommendation. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 15–24 (2016)
16. Xu, C., et al.: Long-and short-term self-attention network for sequential recommendation. Neurocomputing **423**, 580–589 (2021)
17. Zhang, Y., et al.: Sequential click prediction for sponsored search with recurrent neural networks. arXiv preprint arXiv:1404.5772 (2014)
18. Zhao, P., Zhu, H., Liu, Y., Li, Z., Xu, J., Sheng, V.S.: Where to go next: a spatio-temporal LSTM model for next poi recommendation. arXiv preprint arXiv:1806.06671 (2018)
19. Zhou, K., et al.: Leveraging historical interaction data for improving conversational recommender system. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2349–2352 (2020)