Next Point-of-Interest Recommendation with Temporal and Multi-level Context Attention

Ranzhen Li, Yanyan Shen, Yanmin Zhu
Department of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, China
{liranzhen, shenyy, yzhu}@sjtu.edu.cn

***Abstract*—With the prosperity of the location-based social networks, next Point-of-Interest (POI) recommendation has become an important service and received much attention in recent years. The next POI is dynamically determined by the mobility pattern and various contexts associated with user check-in sequence. However, exploring spatial-temporal mobility patterns and incorporating heterogeneous contextual factors for recommendation are challenging issues to be resolved. In this paper, we introduce a novel neural network model named TMCA (Temporal and Multi-level Context Attention) for next POI recommendation. Our model employs the LSTM-based encoder-decoder framework, which is able to automatically learn deep spatial-temporal representations for historical check-in activities and integrate multiple contextual factors using the embedding method in a unified manner. We further propose the temporal and multi-level context attention mechanisms to adaptively select relevant check-in activities and contextual factors for next POI preference prediction. Extensive experiments have been conducted using two real-world check-in datasets. The results verify (1) the superior performance of our proposed method in different evaluation metrics, compared with several state-of-the-art methods; and (2) the effectiveness of the temporal and multi-level context attention mechanisms on recommendation performance.**

***Keywords*-POI recommendation; attention mechanism; spatial and temporal; sequential prediction**

I. Introduction

Recent years have witnessed the proliferation of various location based social networks (LSBNs) such as Gowalla, Yelp, Foursquare, etc. As a result, a large number of user check-in activities are becoming available from the LSBNs, which facilitates a variety of applications. One of the most important applications is the next POI recommendation, which aims to predict the next POI based on the historical check-in activity sequence of a user, as illustrated in Fig. 1. In addition to user interests, the check-in sequences encode strong user mobility patterns and behavioral infingormation. For instance, according to our empirical analysis, most users have revisiting behaviors, i.e., visit the same POI repeatedly within a short time window; and the transition time and distance between consecutive check-in activities are relatively short indicating the spatial and temporal visiting localities.

A large number of efforts have been devoted to studying the problem of next POI recommendation. While matrix factorization (MF) method [1] have become the most popular solution adopted by recommender systems, they fail to take

Fig. 1. An illustration of next POI recommendation.

the sequential behaviors into consideration and hence achieve suboptimal performance in terms of the prediction accuracy. Prior works [2], [3] proposed the Markov Chain based methods for sequential recommendation, but they only considered the influence from the last check-in activity. Recently, inspired by the success of word2vec [4]–[6] in sequential problems, some researchers proposed to learn embedding vectors for POIs [7], [8] and perform recommendation based on the temporal POI embeddings. Some works [9]–[11] adopted the Recurrent Neural Network (RNN) to derive a deep representation for the check-in visiting trajectory for next POI prediction. While RNN-based methods achieved better performance, they suffer from the problem of vanishing gradient [12], [13].

On the other hand, the decision of the next POI to visit is typically determined by various context information associated with historical check-ins. Prior works [9], [14], [15] considered the effects of transition distances on next POI, while others [16] exploited the social network influence to improve recommendation accuracy. However, the existing solutions have two major limitations. First, the proposed models are delicately designed to capture a few contextual factors and can hardly be extended to incorporate other heterogeneous contexts for next POI recommendation in a unified manner. Second, they do not distinguish the importance of different spatial-temporal contexts on user preference over the POIs.

Inspired by the Dual-stage Attention-based RNN (DA-RNN) [17], recently proposed for multi-driving series prediction, we propose a novel neural network model named TMCA (Temporal and Multi-level Context Attention), for next POI recommendation. We identify heterogeneous contextual factors associated with check-ins and our model leverages an embedding layer to incorporate all kinds of contextual factors in a unified manner. In general, our model employs the encoder-decoder framework based on Long Short Term

978-1-5386-9159-5/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDM.2018.00144

Memory (LSTM) [18] to learn deep spatial-temporal dependence among historical check-in activities. Furthermore, we introduce two attention mechanisms to select relevant historical and contextual factors. First is the multi-level context attention which focuses on selecting important contextual factors with two levels, namely micro and macro. Second is the temporal attention which focuses on how to select historical relevant activities. Finally, we use a full-connect neural network to infer the probability of POIs for next visiting.

The major contributions of this paper are as follows.

- We propose a novel encoder-decoder neural network model named TMCA for next POI recommendation. TMCA leverages the embedding method to incorporate heterogeneous contextual factors in a unified way.
- The temporal and multi-level context attention mechanisms are integrated into TMCA, to identify relevant historical check-ins and contextual factors on next POI recommendation.
- Extensive experiments base on two real-world check-in data demonstrate (1) our TMCA model outperforms several state-of-the-art methods in different evaluation metrics, and (2) the two attention mechanisms are effective in improving the recommendation accuracy.

II. Preliminaries

A. Data Model

Let $\mathcal{U}$ and $\mathcal{P}$ denote the sets of users and POIs, respectively. Each POI $y \in \mathcal{P}$ is denoted by a pair of (lon, lat), indicating its geographical location. The basic input to the next POI recommendation problem is user historical check-in activity, as defined below.

Definition 1 (Check-in activity): A check-in activity is a triplet $r = (u, y, t)$, where $u \in \mathcal{U}$, $y \in \mathcal{P}$ and t is the absolute time when the check-in happens.

In this paper, we propose to leverage heterogeneous contexts associated with each check-in activity, to populate the semantics of check-ins and boost the recommendation performance. We mainly consider the following five categories of contexts.

Definition 2 (User, POI, activity and transition context): The *user context* x_u is a set of factors that describe the intrinsic attributes of the user u, such as the number of followers, etc. Similarity, the *POI context* x_p describes the properties of POI y, including its belonging city, state, category, etc. The *activity context* x_a includes the auxiliary information about the check-in, such as the average ratings, etc. For a check-in activity, we define two kinds of *transition context* x_t. The time-transition is the time difference between the current check-in and its previous check-in. The space-transition is the geographical distance between the current check-in and its previous one. It is worth mentioning that taking transition context into consideration is a novel attempt of this paper.

Henceforth, for any check-in $r = (u, y, t)$, we denote by $\bar{r} = (u, y, t, x_u, x_p, x_a, x_t)$ the *context-enhanced check-in activity* for r.

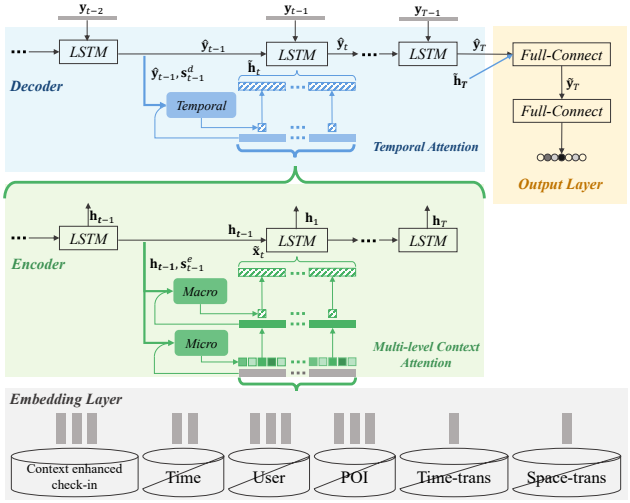


Fig. 2. TMCA architecture.

B. Problem Statement

We consider a sequence of context-enhanced check-in activities for a user u. Given the query time and location for recommendation, the goal of this paper is to produce a ranking list to recommend the next POI for u.

Definition 3 (Next POI recommendation): Given the set $\mathcal{P}$ of POIs, current query time Q_T and location Q_L, and a sequence $R = \{\bar{r}_1, \cdots, \bar{r}_{T-1}\}$ of the previous context-enhanced check-ins for a user $u \in \mathcal{U}$, the next POI recommendation problem is to predict the probability of each POI in $\mathcal{P}$ that would be visited by u at Q_t and Q_l, i.e.,

$$\Pr(y \in \mathcal{P} \mid R, u, Q_T, Q_L)$$

The output is a ranking list of k POIs with the highest predicted probabilities.

Generally, we aim to develop a neural network model to learn a function that measures the conditional probability distribution of the next POI for a particular user.

C. Framework Overview

Our framework mainly consists of three components. (1) Data acquisition. We collect the contexts from auxiliary information about users and POIs according to historical user check-in records, and derive context-enhanced check-in sequences in chronological order. (2) Model training. We sample a set of sequences of length T as the training examples, where the first $T-1$ check-ins together with the query time and location are input. The output is a distribution over all known POIs indicating the conditional probabilities for next visiting. The model parameters are learned iteratively through training error back-propagation. (3) Ranking. We sort POIs according to their predicted probabilities which can be acquired like training process, and report top k ranked POIs as the recommendation list.

III. Temporal and Multi-level Context Attentive Neural Networks

Fig. 2 depicts the architecture of our neural network model named TMCA, for next POI recommendation. At a high level, our model employs the embedding-based encoder-decoder framework. We use an embedding layer to cope with heterogeneous contexts in a unified manner. The encoder learns the representations of different contexts in the sequential check-in activities. The decoder absorbs the previous POI visiting sequence and selects relevant encoder hidden states for predicting the probability of each POI to be visited next. We introduce a temporal attention module to distinguish the importance of historical check-in activities over different time steps, and leverage a multi-level context attention module to adaptively select discriminative contexts for the final prediction. In the following subsections, we provide the details of each component in TMCA.

A. Embedding Layer

Given a sequence of such activities, we supply all pieces of information except the POI y to the encoder module. However, the factors in $\bar{r}$ are heterogeneous in terms of the data type. For instance, some factors like the number of reviews are numerical while other factors like POI's belonging type is categorical. The purpose of the embedding layer is to encode each factor in $(u, t, x_u, x_p, x_a, x_t)$ to a vector representation. The embeddings of different factors are obtained as follows.

Embedding check-in user and time. Users can be represented by one-hot representation and check-in time also can be represented by one-hot vectors by simple processing. We use two factors w and h to describe time, where $w = 0$ or 1 representing whether t belongs to weekday or weekend and $h \in \{0, 1, \cdots, 23\}$ is the hour time of t. Through full-connected embedding layer, we can get the latent vector.

Embedding numerical factors in contexts. Basically, for numerical factors in contexts, we directly use their original values as the scalar embeddings. A special case is the encoding of transition contexts x_t. According to the observed mobility locality, many consecutive check-ins have short distances and time intervals. We thus adopt the piecewise function to model transition contexts. Specifically, consider two successive check-ins with time transition x_{tt} and space transition x_{ts}, To distinguish time transitions with different intervals, we create three latent vectors $\mathbf{t}_\mathrm{s}, \mathbf{t}_\mathrm{m}, \mathbf{t}_\mathrm{l}$ representing short, middle and long intervals, respectively. Formally, the embedding $\mathbf{x}_{tt}$ of the time context x_{tt} is defined as follows.

$$\mathbf{x}_{tt} = \begin{cases} (1 - \frac{x_{tt}}{\theta_t})\mathbf{t}_\mathrm{s} + \frac{x_{tt}}{\theta_t}\mathbf{t}_\mathrm{m}, & \text{if } x_{tt} \leq \theta_t \\ \mathbf{t}_\mathrm{l}, & \text{otherwise} \end{cases} \tag{1}$$

Similarly, for the space transition x_{ts}, we leverage three latent vectors $\mathbf{d}_\mathrm{s}, \mathbf{d}_\mathrm{m}, \mathbf{d}_\mathrm{l}$ to encode short, middle and long distances, respectively. Given a distance threshold θ_d, we apply the following function to compute the embedding $\mathbf{x}_{ts}$ of x_{ts}.

$$\mathbf{x}_{ts} = \begin{cases} (1 - \frac{x_{ts}}{\theta_d})\mathbf{d}_\mathrm{s} + \frac{x_{ts}}{\theta_d}\mathbf{d}_\mathrm{m}, & \text{if } x_{ts} \leq \theta_d \\ \mathbf{d}_\mathrm{l}, & \text{otherwise} \end{cases} \tag{2}$$

Embedding categorical factors in contexts. For categorical factors in contexts, if the factor has a single label, we transform its one-hot representation to a latent vector through the embedding layer. For a factor with multiple labels (e.g. one POI can have multiple types), we retrieve the latent embedding for each label and compute the average vector as its embedding.

To summarize, the embedding layer transforms different factors in check-in activities into latent vector representations accordingly. Note that, because the context at step T is missing, we use the same default context for all sequences.

B. Encoder-Decoder Network

After supplying the original factors to the embedding layer, we concatenate the embeddings of all factors into a vector $\mathbf{x}_t$. For getting the latent vector to denote historical POIs, we use on-hot representation to encode all POIs, feed them to individual fully connected embedding layer. Formally, For the historical POI at step t, we denote by a latent vector $\mathbf{y}_t$. Both of $\mathbf{x}_t$ and $\mathbf{y}_t$ are fed to the encoder-decoder network at step t.

In order to derive the prediction result $\hat{\mathbf{y}}_T$ for next POI y_T, we employ the encoder-decoder architecture [18] with LSTM units [19]. Simply, the encoder and decoder LSTM are:

$$[\mathbf{h}_t; \mathbf{s}_t^e] = \mathrm{LSTM}_e(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{s}_{t-1}^e) \tag{3}$$

$$[\hat{\mathbf{y}}_t; \mathbf{s}_t^d] = \mathrm{LSTM}_d([\mathbf{h}_t; \mathbf{y}_{t-1}], \hat{\mathbf{y}}_{t-1}, \mathbf{s}_{t-1}^d) \tag{4}$$

where $\mathbf{h}_t, \mathbf{s}_t^e, \hat{\mathbf{y}}_t, \mathbf{s}_t^d$ are the hidden states and cell states of encoder and decoder LSTM, respectively; the superscript e, d represents the encoder and decoder. Specifically, $\hat{\mathbf{y}}_t$ can be regarded as the predicting representation of POI at step t.

Through the encoder-decoder network, we get the predicting representation of next POI using the following equation:

$$\hat{\mathbf{y}}_T = F(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T, \mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{T-1}) \tag{5}$$

In what follows, we present how to incorporate the two attention mechanisms into our encoder-decoder architecture. Since our model treats different factors equally, we denote by $\mathbf{x}(i, t)$ the embedding vector for the i-th factor in the t-th check-in activity, and we allow the dimensionality of factor embeddings to be different. For any check-in activity, we denote by $\{\mathbf{x}\}$ the output of the embedding layer, which will be fed to the encoder-decoder network presented in the next subsection.

C. Multi-level Context Attention

The multi-level context attention is used to adaptively select discriminative factors for the final prediction. As shown in green part of Fig. 2, for better learning the attention on different factors, we consider hierarchical attention with two levels: macro-context attention and micro-context attention. Specifically, each factor has its corresponding latent vector. The vector is macro-context and each value in this vector is micro-context.

Micro-context Attention module measures the importance of each value in the latent vector. In this module, each value

$\mathbf{x}(i,t)$ represents a micro-context. By collecting values at each time step, we get $\mathbf{x}(i,\cdot) \in \mathbb{R}^T$, which is defined as follows:

$$\mathbf{x}(i,\cdot) = [\mathbf{x}(i,1), \mathbf{x}(i,2), \cdots, \mathbf{x}(i,T)]^\top \tag{6}$$

The attention score of each dimension i at step t is calculated based on $\mathbf{h}_{t-1}$ and $\mathbf{s}^e_{t-1}$ and the micro-context representation $\mathbf{x}(i,\cdot)$. We use the softmax function for normalization.

$$a_{\mathrm{i}}(i,t) = \mathbf{v}_{\mathrm{i}}^\top \tanh(\mathbf{W}_{\mathrm{i}}[\mathbf{h}_{t-1}; \mathbf{s}^e_{t-1}] + \mathbf{U}_{\mathrm{i}}\mathbf{x}(i,\cdot) + \mathbf{b}_{\mathrm{i}}) \tag{7}$$

$$\alpha_{\mathrm{i}}(i,t) = \frac{\exp(a_{\mathrm{i}}(i,t))}{\sum_{k=1}^{I} \exp(a_{\mathrm{i}}(k,t))}, 1 \le i \le I \tag{8}$$

where I is the dimension of x; $\mathbf{v}_{\mathrm{i}}, \mathbf{b}_{\mathrm{i}} \in \mathbb{R}^T$, $\mathbf{W}_{\mathrm{i}} \in \mathbb{R}^{T \times 2N_e}$, $\mathbf{U}_{\mathrm{i}} \in \mathbb{R}^{T \times T}$ are parameters to be learned. Then we multiply $\mathbf{x}(i,t)$ by $\alpha_{\mathrm{i}}(i,t)$, and get the micro-context attentive embedding vector:

$$\tilde{\mathbf{x}}_{\mathrm{i}}(\cdot,t) = [\alpha_{\mathrm{i}}(1,t)x(1,t), \cdots, \alpha_{\mathrm{i}}(I,t)x(I,t)]^\top \tag{9}$$

Macro-context Attention module measures the importance degrees of different factor vectors. In this attention, we collect latent vectors of a factor at each time step as denoted by $\mathbf{x} \in \mathbb{R}^{I \times T}$. Then the corresponding macro-context attention scores at step t are calculated based on the information from last step and the macro-context itself. We also use the softmax function for normalization, as defined below:

$$a_{\mathrm{a}}(\cdot,t) = \mathbf{v}_{\mathrm{a}}^\top \tanh(\mathbf{W}_{\mathrm{a}}[\mathbf{h}_{t-1}; \mathbf{s}^e_{t-1}] + \mathbf{U}_{\mathrm{a}}\mathbf{x}\mathbf{b}_{\mathrm{a1}} + \mathbf{b}_{\mathrm{a2}}) \tag{10}$$

$$\alpha_{\mathrm{a}}(\cdot,t) = \frac{\exp(a_{\mathrm{a}}(\cdot,t))}{\sum_{a_{\mathrm{a}} \in \{a_{\mathrm{a}}\}} \exp(a_{\mathrm{a}}(\cdot,t))}, 1 \le t \le T \tag{11}$$

where $\{a_{\mathrm{a}}\}$ is obtained by calculating a_{a} for each factor's vector in the embedding output set $\{\mathbf{x}\}$; $\mathbf{v}_{\mathrm{a}}, \mathbf{b}_{\mathrm{a1}}, \mathbf{b}_{\mathrm{a2}} \in \mathbb{R}^T$, $\mathbf{W}_{\mathrm{a}} \in \mathbb{R}^{T \times 2N_e}$, $\mathbf{U}_{\mathrm{a}} \in \mathbb{R}^{T \times I}$ are parameters to be leaned. Then we multiply micro context attentive embedding vector $\tilde{\mathbf{x}}_{\mathrm{i}}(\cdot,t)$ by $\alpha_{\mathrm{a}}(\cdot,t)$, and get the macro context attentive embedding vector $\tilde{\mathbf{x}}_{\mathrm{a}}(\cdot,t)$:

$$\tilde{\mathbf{x}}_{\mathrm{a}}(\cdot,t) = \alpha_{\mathrm{a}}(\cdot,t)\tilde{\mathbf{x}}_{\mathrm{i}}(\cdot,t) \tag{12}$$

We also concatenate each factor's latent vector which have been updated by attentive $\tilde{\mathbf{x}}_{\mathrm{a}}(\cdot,t)$. Then we use the multi-level context attentive vector $\tilde{\mathbf{x}}_t$ to replace $\mathbf{x}_t$.

D. Temporal Attention

As each historical check-in activity has different weight for the preference of next POI, we apply the temporal attention to adaptively select the relevant historical check-in activities towards better recommendation.

As shown in the blue part of Fig. 2, the temporal attention mechanism is used to dynamically decide relevant hidden states of encoder. Temporal attention assigns attention to each hidden state $\mathbf{h}_j$ of encoder by referring to last decoder LSTM cell state $\mathbf{s}^d_{t-1}$, hidden state $\hat{\mathbf{y}}_{t-1}$ and encoder hidden state self. We also normalize the attention scores via the softmax function. The equations are formally defined as follows.

$$b^j_{\mathrm{t}} = \mathbf{v}_{\mathrm{t}}^\top \tanh(\mathbf{W}_{\mathrm{t}}[\hat{\mathbf{y}}_{t-1}; \mathbf{s}^d_{t-1}] + \mathbf{U}_{\mathrm{t}}\mathbf{h}_j + \mathbf{b}_{\mathrm{t}}) \tag{13}$$

TABLE I
STATISTICS OF THE TWO DATASETS

Dataset	#User	#POI	#Check-in	Density	From-To
Gowalla	22,209	50,569	1,493,799	0.0013	2009.2-2010.10
Yelp	11,564	18,683	492,489	0.0023	2014.1-2017.6

$$\beta^j_{\mathrm{t}} = \frac{\exp(b^j_{\mathrm{t}})}{\sum_{k=1}^{T} \exp(b^k_{\mathrm{t}})}, 1 \le j \le T \tag{14}$$

where $\mathbf{v}_{\mathrm{t}}, \mathbf{b}_{\mathrm{t}} \in \mathbb{R}^{N_p}$, $\mathbf{W}_{\mathrm{t}} \in \mathbb{R}^{N_p \times 2N_d}$, $\mathbf{U}_{\mathrm{t}} \in \mathbb{R}^{N_p \times N_e}$ are parameters to be learned. The temporal attention score β^j_{t} indicates how important step j is. Then we multiply $\mathbf{h}_j$ by each score β^j_t and calculate summation over all the multiplications. By doing this, we obtain the temporal attentive hidden state $\tilde{\mathbf{h}}_t$ at step t to replace $\mathbf{h}_t$:

$$\tilde{\mathbf{h}}_t = \sum\nolimits_{j=1}^{T} \beta^j_{\mathrm{t}} \mathbf{h}_j \tag{15}$$

Both multi-level context attention and temporal attention module are feed forward networks that can be jointly trained with the encoder and decoder LSTM units.

E. Learning and Optimization

Through the previous attention-based encoder-decoder network, we obtain the predicting representation of next POI $\hat{\mathbf{y}}_T$. We use two fully connected layers to produce the conditional probability distribution of next POI. The first fully connected layer considers temporal attention $\tilde{\mathbf{h}}_T$:

$$\tilde{\mathbf{y}}_T = \mathbf{W}_T[\hat{\mathbf{y}}_T; \tilde{\mathbf{h}}_T] + \mathbf{b}_T \tag{16}$$

where $\mathbf{b}_T \in \mathbb{R}^{N_p}$, $\mathbf{W}_T \in \mathbb{R}^{N_p \times (N_d + N_e)}$ are parameters to be learned. The second fully connected layer produces the conditional probability distribution of next POI:

$$\Pr(\cdot|R,u,Q_T,Q_L) = softmax(\mathbf{W}_o\tilde{\mathbf{y}}_T + \mathbf{b}_o) \tag{17}$$

where $\mathbf{b}_o \in \mathbb{R}^{|\mathcal{P}|}$, $\mathbf{W}_o \in \mathbb{R}^{|\mathcal{P}| \times N_p}$ are parameters to be learned. The objective function is formulated by a log likelihood function as follows:

$$\mathcal{O} = \mathrm{argmax}_\Theta \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} \log(\Pr(y_T|R,u,Q_T,Q_L)) \tag{18}$$

where $\mathcal{Q}$ is the set of $\{R,u,Q_T,Q_L\}$.

The source code of our proposed method is available online[1].

IV. Experimental Evaluation

A. Experimental Settings

Datasets. We experiment with two real-world datasets: Gowalla[2] and Yelp[3]. Note that, the category information of Gowalla dataset are provided by Yang et al. [16]. Gowalla has obvious revisiting behaviors while Yelp has rich contexts. Table I gives their basic statistics. We preprocess both datasets by removing inactive users who visit less than 10 POIs, and unpopular POIs that are visited by less than 10 users. For

[1] https://github.com/zhenql/TMCA
[2] http://snap.stanford.edu/data/loc-gowalla.html
[3] https://www.yelp.com/dataset/challenge

insuring the sequence length, we further remove the user who visit less than 20 times. For two datasets, former 70% check-ins of the history for each user are selected as train set, last 20% as test set and rest 20% as validation set. Note that, we don't use review content and social relationship.

Compared Methods. We compare with the following state-of-the-art algorithms:

- **Popu** recommend the most popular POIs through using the *User Count* location ranking scheme [20].
- **FPMC** is the Factorized Personalized Markov Chain method [3] which considers the impact of user-item, last item-item and last item-user interaction with Markov Chain framework.
- **PRME** is the Personalized Ranking Metric Embedding method [15] which jointly considers sequence transition and user preference and uses euclidean distance to rank POIs with Markov Chain framework.
- **RNN** is a standard RNN model for series problem.
- **LSTM** is a variant of RNN model with LSTM cell [19].

To investigate the effects of attention, we also consider the following variants:

- **TMCA-na** uses the same framework of the TMCA but the two attention structure are replaced by an average attention processing. Specifically, the multi-level context and temporal attention scores are calculated by:

$$\bar{\alpha}_{\mathrm{i}}(i,t) = \frac{1}{I};\ \bar{\alpha}_{\mathrm{a}}(\cdot,t) = \frac{1}{|\{\mathbf{x}\}|};\ \bar{\beta}_{\mathrm{t}}^{j} = \frac{1}{T} \qquad (19)$$

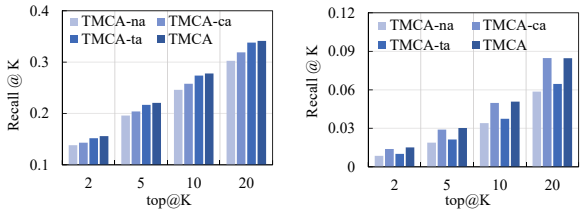
- **TMCA-ca** is similar to TMCA-na except that it only uses multi-level context attention module.
- **TMCA-ta** is similar to TMCA-na except that it only uses temporal attention module.

Evaluation Metrics. We experiment with several evaluation metrics. Recall@K and Normalized Discounted Cumulative Gain (NDCG@K) are two popular metrics for measuring the performance of the top@K personalized ranking recommendation list. We report recall@K and NDCG@K with K$\in \{2, 5, 10\}$.

Parameter Settings. We provide the optimal hyper-parameters setting for each compared method. For all methods, the dimensions of latent factor $\mathbf{x}_u, \mathbf{y}_p$ are set in $\{20, 40, 60\}$. For methods which based on RNN, we set the number of LSTM hidden units in $\{20, 40, 60\}$ and set the optimal learning rate from 0.0001 to 0.01. We adopt Adam optimizer to update all variables and set batch size of 256 and 512 for RNN, LSTM, TMCA and its variants. The Stochastic Gradient Descent algorithm is used to update the parameters in FPMC and PRME. In addition, we set threshold of time interval θ_t = 3hour, distance θ_d = 10km, and set the dimension of other latent vectors by 20.

B. Comparison of Various Approaches

The experimental results of state-of-the-art methods are shown in Table II. The length of check-in sequence is set to 8. Among these various approaches, TMCA achieves the best performance while Popu gets the worst performance.



(a) Gowalla (b) Yelp

Fig. 3. Results of different TMCA variants.

Comprising between two datasets, the results of Yelp are worse than Gowalla which has obvious revisiting behaviors. All the methods except Popu utilize sequential relationship so that they get better results in Gowalla. Furthermore, in terms of the metric of recall@2, TMCA achieves 5% improvements compared with the best result of others methods (i.e. RNN) in Gowalla, while achieving 148% (vs. FPMC) in Yelp. The improvement of TMCA in Yelp is far more than Gowalla, because the kinds of contexts information of Yelp is far more than Gowalla. We only use category and location contexts of POI, time context of check-in and transition context for Gowalla while using total 22 contexts for Yelp. The results not only can verify the performance of TMCA, but also show that, more information can get better result.

We conduct the experiments to evaluate the performance of the TMCA variants and the results are shown in Fig. 3. For both datasets, TMCA-na achieves the worst performance while TMCA achieves the best performance. It is worth noting that TMCA-ta achieves better performance in Gowalla while TMCA-ca achieves better in Yelp. The difference between two datasets exactly reflect their distinct characteristics. Gowalla has obvious revisiting behaviors so that the improvement from the temporal attention mechanism is significant, while Yelp has more contexts so that multi-level context attention mechanism is more suitable for Yelp. The performance of TMCA is close to the best variant of TMCA. It is promising that TMCA achieves better results in the situation: obvious revisiting behaviors and rich contexts.

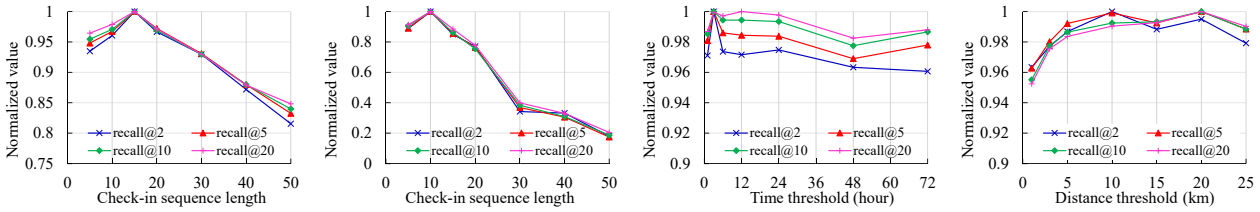
C. Effect of Different Hyperparameter Settings

As shown in Fig. 4, we try to use different hyper parameters to test the performance of TMCA. We first investigate the effect of sequence length. Fig. 4a and Fig. 4b plot the results of recall@K with different check-in sequence length T from 5 to 50, where θ_t = 3hour and θ_s = 10km on two dataset. The best length for Gowalla is 15, while the best length for Yelp is 10. Note that the tendency of sequence length T on Gowalla is more stable than on Yelp.

Then we investigate the effect of time interval and distance threshold by finding the best parameters. For Gowalla, we conduct experiments by searching best time interval threshold θ_t and distance threshold θ_d. As shown in Fig. 4c and Fig. 4d, the best time threshold is 3hour and the best distance threshold is 10km. In addition, the tendency of threshold is stable that the worst result can achieve 96% of the best performance.

TABLE II
RESULTS OF DIFFERENT METHODS

Dataset	Method	recall@2	recall@5	recall@10	NDCG@2	NDCG@5	NDCG@10
Gowalla	Popu	0.821%	1.544%	2.812%	0.466%	1.022%	1.427%
	FPMC	8.800%	13.008%	17.217%	7.815%	9.695%	11.053%
	PRME	11.334%	17.479%	22.487%	9.783%	12.548%	14.163%
	RNN	14.623%	20.887%	26.057%	12.981%	15.802%	18.337%
	LSTM	13.979%	19.902%	24.805%	12.440%	15.108%	16.693%
	TMCA	**15.404%**	**21.926%**	**27.725%**	**13.796%**	**16.726%**	**18.597%**
Yelp	Popu	0.115%	0.287%	0.559%	0.057%	0.167%	0.254%
	FPMC	0.548%	1.271%	2.230%	0.451%	0.771%	1.099%
	PRME	0.366%	1.061%	2.101%	0.275%	0.581%	0.915%
	RNN	0.502%	1.099%	2.045%	0.414%	0.676%	0.981%
	LSTM	0.490%	1.097%	1.947%	0.402%	0.669%	0.943%
	TMCA	**1.361%**	**2.870%**	**4.809%**	**1.142%**	**1.809%**	**2.430%**



(a) Gowalla (b) Yelp (c) Gowalla (d) Gowalla

Fig. 4. Results of different hyper parameters.

V. CONCLUSION

In this paper, we have studied the problem of next POI recommendation. We propose an encoder-decoder based neural network model named TMCA to capture the complex spatial and temporal dependencies among historical check-in activities automatically. Our model leverages the embedding method to incorporate heterogeneous contextual factors to boost recommendation performance. Furthermore, we introduce the temporal and multi-level context attention mechanisms to dynamically select the relevant check-ins and discriminative contextual factors for predicting the preferences over POIs to visit next. The experimental results on two real-world check-in datasets show that TMCA outperforms the state-of-the-art methods and both attention mechanisms are effective for improving recommendation performance.

ACKNOWLEDGEMENTS

This research is supported by the National Key Research and Development Program of China (No. 2018YFC0831604), NSFC (no. 61602297, 61772341, 61472254, 61572324, 61170238, and 61472241), Singapore NRF (CREATE E2S2). Yanmin Zhu is also supported by the Program for Changjiang Young Scholars in University of China, and the Program for Shanghai Top Young Talents. Yanyan Shen is the corresponding author.

REFERENCES

[1] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation," in *SIGKDD*, 2014, pp. 831–840.

[2] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King, "STELLAR: spatial-temporal latent ranking for successive point-of-interest recommendation," in *AAAI*, 2016, pp. 315–322.

[3] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010, pp. 811–820.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[5] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014.

[6] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *TACL*, vol. 5, pp. 135–146, 2017.

[7] X. Liu, Y. Liu, and X. Li, "Exploring the context of locations for personalized location recommendations," in *IJCAI*, 2016, pp. 1188–1194.

[8] S. Zhao, T. Zhao, I. King, and M. R. Lyu, "Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation," in *WWW*, 2017, pp. 153–162.

[9] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *AAAI*, 2016, pp. 194–200.

[10] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *CoRR*, vol. abs/1511.06939, 2015.

[11] C. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *WSDM*, 2017, pp. 495–503.

[12] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *ICML*, 2013, pp. 1310–1318.

[13] Y. Bengio, P. Y. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[14] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *IJCAI*, 2013, pp. 2605–2611.

[15] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new POI recommendation," in *IJCAI*, 2015, pp. 2069–2075.

[16] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation," in *SIGKDD*, 2017, pp. 1245–1254.

[17] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *IJCAI*, 2017, pp. 2627–2633.

[18] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *SSST*, 2014, pp. 103–111.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] J. Ye, H. Cheng, and Z. Zhu, "What's Your Next Move: User Activity Prediction in Location-based Social Networks," in *SIAM*, 2013, pp. 171–179.