An Attention-based Spatiotemporal LSTM Network for Next POI Recommendation

Liwei Huang, Yutao Ma, *Member, IEEE*, Shibo Wang, and Yanbo Liu

Abstract—Next point-of-interest (POI) recommendation, also known as a natural extension of general POI recommendation, is recently proposed to predict user's next destination and has attracted considerable research interest. It focuses on learning users' sequential patterns of check-in behavior and on training personalized recommendation models using different types of contextual information. Unfortunately, most of the previous studies failed to incorporate the spatiotemporal contextual information, which plays a critical role in analyzing user check-in behavior, into recommending the next POI. In recent years, embedding learning and recurrent neural network (RNN) based approaches show promising performance for modeling sequential patterns of check-in behavior in next POI recommendation. However, not all of the historical check-in records contribute equally to the next-step check-in behavior. To provide better next POI recommendation performance, we first proposed a spatiotemporal long and short-term memory (ST-LSTM) network. By feeding the spatiotemporal contextual information into the LSTM network in each step, ST-LSTM can model the spatial and temporal information better. Also, we developed an attention-based spatiotemporal LSTM (ATST-LSTM) network for next POI recommendation. By using the attention mechanism, ATST-LSTM can focus on the relevant historical check-in records in a check-in sequence selectively using the spatiotemporal contextual information. Besides, we conducted a comprehensive performance evaluation using large-scale real-world datasets collected from two popular location-based social networks, namely Gowalla and Brightkite. Experimental results indicated that the proposed ATST-LSTM network outperformed two state-of-the-art next POI recommendation approaches regarding three commonly-used evaluation metrics.

Index Terms—Next point-of-interest recommendation, location-based service, long short-term memory, attention, spatiotemporal embedding

1 Introduction

THE boom of the mobile Internet facilitates the widespread application of location-based social networks (LBSNs), such as Foursquare[1], Loopt[2], and Yelp[3], in human society. Users on LBSNs can find any points of interest (POIs), post their check-ins, and share their life experiences in the real world via mobile devices and location-based services (LBSs). A large number of users' check-in records have been used to improve user experience on LBSNs by accurate location prediction services. As an extension of general POI recommendation [1], [2], [3], [4], *next POI recommendation* (or called *successive POI recommendation*) has become an active research focus in the academic and industrial field [5]. Its primary goal is to predict the next POI a user may visit at a specific time point by mining the user's check-in records and other types of information available [6].

Fig. 1 illustrates an example of next POI recommendation. Given the first user's sequence of check-ins at the hotel (T_{t-5}), gym (T_{t-3}), and hotel (T_{t-1}), which POIs could be recommended to the user at time point T_t? General POI recommender systems may recommend a restaurant or a museum at T_t with the same probability because "restaurant" and "museum" appear with "hotel" and "gym" at the same frequency (see the other three similar users in Fig.1). In contrast, next POI recommender systems may recommend a restaurant for the first user because "restaurant" frequently appears after "hotel."

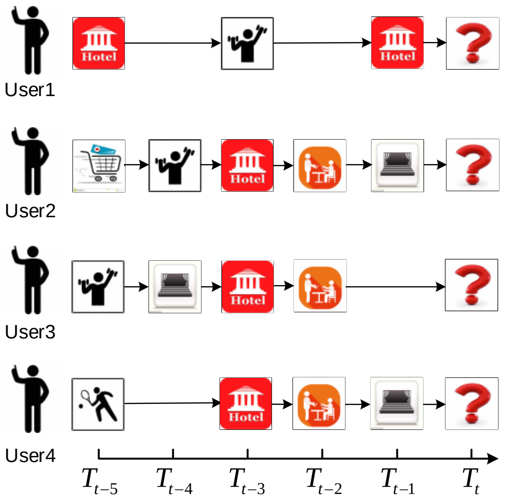


Fig. 1. An example of next POI recommendation.

Unlike items such as movies, music, and news in traditional context-free recommender systems, the interactions between POIs and a user (i.e., check-ins) require the user to visit those POIs in the physical world. Therefore, spatial contextual information, including latitude and longitude coordinates of places, would have a significant effect on

- L. Huang and Y. Liu are with the Beijing Institute of Remote Sensing, Beijing 100854, China. E-mails: {dr_huanglw, liuyanbonudt}@163.com.
- Y. Ma is with the School of Computer Science, Wuhan University, Wuhan 430072, China. E-mail: ytma@whu.edu.cn.
- S. Wang is with the Google, Mountain View, CA 94043, USA. E-mail: wsb1112@gmail.com.

[1] https://foursquare.com

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

[3] http://www.yelp.com

xxxx-xxxx/0x/$xx.00 © 200x IEEE Published by the IEEE Computer Society

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2019.2918310, IEEE Transactions on Services Computing

user's check-in behavior. Besides, time is also a crucial factor that affects human real-life check-in activities. For example, some people often go to the gym after work on weekdays, and they may go to cinemas at night on weekends. In brief, spatial and temporal contexts are critical to analyzing user behavior for better-personalized next POI recommendation. Owing to the significance and value of next POI recommendation in urban planning, business advertising, and service industry, researchers have proposed many approaches to enhance the quality of next POI recommender systems [6], [7], [8], [9]. However, how to accurately predict users' whereabouts at a given time point according to complex spatiotemporal contextual information is still a challenging issue [9].

Recently, recurrent neural networks (RNNs) (e.g., long short-term memory (LSTM) networks [10]) have been successfully used to mine and model sequential patterns of human check-in behavior for next POI recommendation [9], [11]. These previous studies suggest that a recently-visited POI always has a more significant impact on future check-in behavior than the ones before it in a check-in sequence. It is worth noting that human check-in behaviors often show periodicity, non-uniformness, and consecutiveness [4], implying that not all historical check-ins are meaningful for predicting next-step check-in behavior. Owing to the noises caused by irrelevant historical check-ins, using a global vector to represent the influence of historical check-ins of the RNN architecture may result in suboptimal results.

To address the problem mentioned above, in this study, we attempt to use an extended LSTM (ST-LSTM) network to model the spatiotemporal contextual information derived from LBSNs. Inspired by the artificial "attention mechanism" in neural networks [12], [13], [14], we further propose an attention-based spatiotemporal LSTM network (ATST-LSTM) for next POI recommendation, which can capture the most pertinent piece of a check-in sequence. Besides, we conduct an elaborate experiment on two publicly-available datasets, namely Gowalla[4] and Brightkite[5], to demonstrate the effectiveness of ATST-LSTM. Therefore, this work would help to predict individual and collective activities driven by human mobility more precisely, thus providing better location-based mobile recommendation and visual (personal) assistant services. In conclusion, the technical contributions of this work are three-fold.

1. By learning a general representation of complex dependencies between users' historical check-ins on LBSNs, we proposed an RNN-based network architecture, called ST-LSTM, to model the temporal-spatial contextual information collected from LBSNs jointly. ST-LSTM embeds the spatial and temporal contexts of user check-ins into a compact vector representation used to predict user's next-step check-in behavior more precisely.
2. Considering the successful applications of the attention mechanism in computer vision and natural language processing (NLP), we proposed an attention-based spatiotemporal LSTM (ATST-LSTM) network that first introduces the attention mechanism to ST-LSTM for next POI recommendation. More specifically, ATST-LSTM can automatically measure the relevance of various inputs to the network (e.g., POIs and the spatial and temporal contexts) at each step and then adjusts the attention weights for the inputs accordingly.
3. We compared ATST-LSTM with eight baseline approaches of next POI recommendation, and the empirical results on the datasets of Gowalla and Brightkite indicated that ATST-LSTM outperformed these competing baselines regarding the evaluation metrics. For example, compared with a state-of-the-art next POI recommendation model (ST-RNN [9]), the average *Precision*@5, *Recall*@5, and *F1-score*@5 values of ATST-LSTM were increased by 29.43%, 29.32%, and 29.35%, respectively, on the two datasets.

The remainder of this paper is organized as follows. Section 2 reviews the works related to the topics of POI recommendation and next POI recommendation. Section 3 introduces the preliminaries to our study. Section 4 introduces the details of the proposed attention-based spatiotemporal LSTM network, Section 5 describes the experimental setups, and Section 6 presents the experimental results. Finally, Section 7 concludes this paper and outlines our future work.

2 Related Work

2.1 POI Recommendation

So far, POI recommendation has attracted much more attention in the fields of LBS and recommender systems. Many previous studies learned user preference for POIs using collaborative filtering (CF) [15]. For example, the user-based CF technique was commonly used to recommend POIs for target users [16], [17]. Besides, other researchers employed the model-based CF technique such as matrix factorization (MF) [1], [2], [18], [19]. Unfortunately, CF always suffers from the data sparsity problem in the user-POI matrices, which contain a large number of inactive users and unpopular POIs.

The integration of different types of information into POI recommendation approaches has been proved to be useful to alleviate the data sparsity problem. To the best of our knowledge, researchers have exploited the social influence [1], [16], [20], [21], sequential influence [22], [23], [24], [25], geographical influence [1], [2], [17], and temporal influence [2], [8], [26] to improve the performance of POI recommender systems. Moreover, some recent works [8], [9] attempted to model users' implicit feedback by learning and ranking pairwise preference with the pairwise rank learning techniques, such as the Bayesian personalized ranking loss [27] and the online weighted approximate-rank pairwise loss [28].

2.2 POI Recommendation with Neural Networks

Deep learning has recently been applied to POI recom-

[4] https://en.wikipedia.org/wiki/Gowalla

[5] https://brightkite.com

mender systems, which may change their traditional recommendation architectures significantly and brings new opportunities to improve further user experience [29]. In particular, a few previous studies employed Word2vec [30] to model users' sequential patterns [26], [31], [32], [33]. For instance, Zhao *et al.* [26] proposed a POI embedding model for capturing the sequential correlations between two check-in behaviors under different temporal states. Zhou *et al.* [31] developed a distributed representation learning framework that incorporated multiple types of contextual information of trajectory data for location recommendation. By mining the contextual influence of each POI, Liu *et al.* [32] learned the individual representation of each POI using Word2vec. In [33], Feng *et al.* proposed a POI2Vec model, which learned the vector representation of POIs based on the geographical influence, to predict possible future visitors.

Also, some recent studies [34], [35] have applied multilayer perceptrons (MLPs) and convolutional neural networks (CNNs) to POI recommendation. In [34], Yang *et al.* proposed a deep neural network framework to predict user preferences to POIs and the contexts associated with users and POIs by learning complex embeddings of users and POIs. In [35], a CNN model was used as a feature extractor by Wang *et al.* to learn useful features from images, thus incorporating visual contents for more precise POI recommendation.

2.3 Next POI Recommendation

Aiming at the next POI recommendation problem, a few previous studies attempted to mine and utilize sequential patterns of users. Most of the existing studies usually employ the properties of a Markov chain to model the sequential influence [6], [22], [23]. For example, in [6], successive POIs were recommended to target users by the factorized personalized Markov chain (FPMC) model [36]. Similarly, Zhang *et al.* [22] developed an additive Markov chain model for predicting the sequential transitive probability, and Ye *et al.* [23] proposed a mixed hidden Markov model to learn the POI categories' transitive patterns of sequential user check-ins.

Meanwhile, matrix factorization models have also been used to characterize the personalized sequential patterns of users. For instance, in [7], a personalized ranking metric embedding (PRME) method was proposed to capture user preferences and POI sequential transitions. In [8], Zhao *et al.* proposed a pairwise tensor factorization method (STELLAR) for next POI recommendation, which was a ranking-based framework and could incorporate fine-grained temporal contexts. Liu *et al.* [37] proposed a "Where and When to gO" (WWO) recommender system to recommend POIs for a specific period, which was able to capture both the static user preferences and the dynamic sequential patterns in a unified framework.

Because RNNs can cope with sequentially ordered data of any kind, they have been used to model sequential correlations and temporal dynamics in next POI recommender systems [9], [11], [38]. For example, Liu *et al.* [9] proposed a spatial and temporal recurrent neural network (ST-RNN) for location prediction, which utilized an RNN architecture to learn the sequential transition. Assuming that sequential correlations in mobile trajectories have different levels, Yang *et al.* [11] employed the RNN and gated recurrent unit (GRU) models to characterize short-term and long-term sequential contexts separately. In [38], an RNN was used to learn to generate new user paths by modeling temporal correlations between POI categories for next stop-over prediction.

Unlike the above studies that failed to consider the spatiotemporal correlation between past check-in behaviors and the next-step user behavior using selective attention, our work performs an attention model on the time steps of LSTM units to selectively emphasize on those more relevant historical check-ins in each step. More specifically, an attention representation is learned and generated in ATST-LSTM to enhance the performance of next POI recommendation.

3 Preliminaries to This Study

3.1 Notations and Definitions

Table 1 presents the notations used in this paper.

TABLE 1. NOTATIONS USED IN THIS STUDY.

Symbol	Description
u, v, l_v, t	user, POI, location (latitude and longitude), time
U, V, L	set of users, set of POIs, set of POI locations
$l_v = (x_v, y_v)$	location coordinates of POI v
$v_{t_k}^u$	POI visited by user u at time point t_k
$c_{t_k}^u = (u, v_{t_k}^u, l_{t_k}^u, t_k)$	check-in activity performed by u on POI $v_{t_k}^u$ at t_k
C_u=$\{c_{t_i}^u\}$	set of check-in activities performed by u
D^u	set of check-in trajectory samples combining with negative POIs of u
C^U=$\{C_{u_i}\}$	set of check-in activities of all the users in U
S_i^u	ith check-in trajectory of u
$\mathbf{p}_u \in \mathbb{R}^d$	latent representation of u
$\mathbf{q}_v \in \mathbb{R}^d$	latent representation of v
$\mathbf{v}_{t_k}^u$	embedding vector of POI $v_{t_k}^u$
$\mathbf{l}_{t_k}^u$	the spatial feature vector of $v_{t_k}^u$
$\mathbf{t}_{t_k}^u$	the temporal feature vector of $v_{t_k}^u$
$\mathbf{h}_{t_k}^u$	the hidden vector of an LSTM unit
o_{t_{N+1},v_k}^u	predicted probability that u visits POI v_k at t_{N+1}
$\mathbf{i}_{t_k}^u, \mathbf{c}_{t_k}^u, \mathbf{f}_{t_k}^u, \mathbf{o}_{t_k}^u$	input gate vector, cell vector, forget gate vector, and output gate vector of LSTM units
$\mathbf{z}^u$	context vector of u
$\boldsymbol{\alpha}$	attention weight vector of ATST-LSTM
$\mathbf{r}$	weighted hidden representation of ATST-LSTM
σ	sigmoid function
$\{\mathbf{W}\}$	set of weight matrices for an LSTM network
$\{\mathbf{b}\}$	set of bias vectors for an LSTM network

Definition 1: POI. In LBSNs, a point of interest (POI) is a spatial item associated with a geographical location, such as a gym or a hotel.

Definition 2: Check-in activity. A user's check-in activity is a quadri-tuple $c_{t_k}^u = (u, v_{t_k}^u, l_{t_k}^u, t_k)$, which indicates that user u visits POI $v_{t_k}^u$ with location $l_{t_k}^u$ at time point t_k.

Definition 3: Check-in sequence. A check-in sequence of a user u is a set of check-in activities of the user, denoted

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2019.2918310, IEEE Transactions on Services Computing

by $C_u = \{c^u_{t_1}, c^u_{t_2}, \dots, c^u_{t_i}\}$. For simplicity, the historical check-ins of all users are denoted by $C^U = \{C_{u_1}, C_{u_2}, \dots, C_{u_{|U|}}\}$, where U is the set of users.

Definition 4: Check-in trajectory. A user's check-in trajectory is a set of consecutive check-ins, denoted by $S^u_i = \{c^u_{t_k}, c^u_{t_{k+1}}, \dots, c^u_{t_{k+N-1}}\}$, where N is the length of the check-in trajectory. S^u_i is a subset of C_u, i.e., $C_u = \cup_i S^u_i$. Note that, in this study, each check-in sequence wherever the time interval between any two successive check-in activities is more than six hours (a standard sliding window) is divided into different check-in trajectories, and all isolated check-ins are removed from the original dataset.

The primary goal of this study is to offer a target user a list of possible POIs that the user is likely to visit at the next time point by mining all users' check-in records. Here, we then formulate the problem of next POI recommendation as follows.

Definition 5: Next POI recommendation. Given all users' check-in sequences C^U, the goal of next POI recommendation is to predict the most likely location v_k that a user u will visit at a specific time point t_{N+1}, i.e., $\max o^u_{t_{N+1},v_k}$.

3.2 Long Short-Term Memory Units

The primary challenge of the next POI recommendation problem is learning personalized user preference for POIs and the sequential correlations between check-ins jointly and efficiently. As we know, a favorite choice is RNN architectures. However, the vanishing gradient problem or the exploding gradient problem exists in a standard RNN. LSTM units [10] were then proposed to address the problems mentioned above, thus making LSTM networks easier to have broad application prospects. As the standard LSTM architecture can capture long-range dependencies in a sequential pattern, in this study, we use it as a building block for next POI recommender systems.

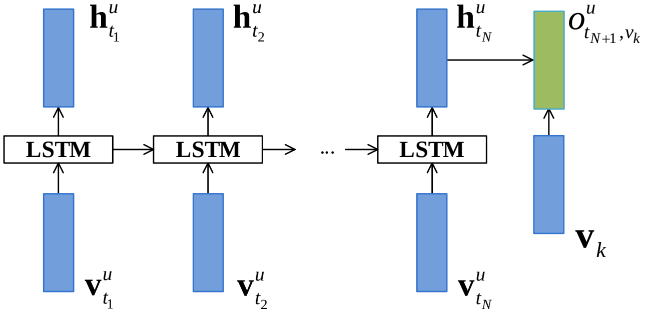


Fig. 2. An illustration of LSTM units used in this study.

Fig. 2 illustrates the standard architecture of LSTM units. At each time step, an LSTM unit takes an input vector $\mathbf{v}^u_{t_k}$ (i.e., the embedding vector of POI $v^u_{t_k}$) and outputs a hidden vector $\mathbf{h}^u_{t_k}$, using an input gate $i^u_{t_k}$, a memory cell $c^u_{t_k}$, a forget gate $f^u_{t_k}$, and an output gate $o^u_{t_k}$. The details of these parameters are introduced as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{h}^u_{t_{k-1}} \\ \mathbf{v}^u_{t_k} \end{bmatrix}, \quad (1)$$

$$\mathbf{f}^u_{t_k} = \sigma(\mathbf{W}_f \cdot \mathbf{X} + \mathbf{b}_f), \quad (2)$$

$$\mathbf{i}^u_{t_k} = \sigma(\mathbf{W}_i \cdot \mathbf{X} + \mathbf{b}_i), \quad (3)$$

$$\mathbf{o}^u_{t_k} = \sigma(\mathbf{W}_o \cdot \mathbf{X} + \mathbf{b}_o), \quad (4)$$

$$\mathbf{c}^u_{t_k} = \mathbf{f}^u_{t_k} \odot \mathbf{c}^u_{t_{k-1}} + \mathbf{i}^u_{t_k} \odot \tanh(\mathbf{W}_c \cdot \mathbf{X} + \mathbf{b}_c), \quad (5)$$

$$\mathbf{h}^u_{t_k} = \mathbf{o}^u_{t_k} \odot \tanh(\mathbf{c}^u_{t_k}), \quad (6)$$

where $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o, \mathbf{W}_c \in \mathbb{R}^{d\times 2d}$ are weight matrices and $\mathbf{b}_i$, $\mathbf{b}_f$, $\mathbf{b}_o$, $\mathbf{b}_c \in \mathbb{R}^d$ are bias vectors of LSTM units. Here, σ denotes the sigmoid function and $\odot$ represents the operation of element-wise multiplication.

Note that we regard the last hidden vector $\mathbf{h}^u_{t_N}$ as the representation of user u. Like MF-based next POI recommendation approaches, this study uses the inner product of user and POI representations to calculate the (predicted) probability that user u visits POI v_k at time point t_{N+1}:

$$o^u_{t_{N+1},v_k} = (\mathbf{h}^u_{t_N})^T \mathbf{v}_k, \quad (7)$$

where $\mathbf{v}_k$ indicates the embedding vector of POI v_k. Finally, for each target user u, our next POI recommender system will offer the top k POIs with the highest values of $o^u_{t_{N+1},v_k}$ to the user.

3.3 Attention Model

The attention mechanism was proposed based on the selective attention mechanism in the human visual system. The principle of *selective attention* is that we need to pay more attention to the most relevant information in a system rather than all available information. Inspired by the idea, various attention models in deep learning were developed by learning to focus on the specific components of the input data. Attention models do not constrain a neural network to encode the input sequence into one fixed-length vector, thus allowing it to refer back to the different parts in the input sequence. Moreover, attention models can model mutual correlations without regard to their path distance in the input or output sequence. Until now, attention models have successfully been employed in a wide variety of tasks, such as machine translation [14], speech recognition [39], and image caption [40].

In [14], Vaswani *et al.* defined an attention function to encode an input sequence into an output sequence using the attention mechanism. In particular, the attention function maps a query and a group of key-value pairs to a context vector, which is a weighted sum of all values. The input of the attention function includes queries, keys, and values. Although the queries and keys have the same dimension d_k, the dimension of the values is d_v ($d_v \neq d_k$). The queries, keys, and values are concatenated as matrices Q, K, and V_{val}, respectively.

In practice, the attention function can be computed on a set of queries simultaneously. For the output of the attention function, the weight assigned to each value is calculated by an alignment function (or called the compatibility function [14]), which measures how well the input query matches with the corresponding key. More specifically, Vaswani *et al.* [14] use the following equation to compute the matrix of outputs:

$$\text{Attention}(Q, K, V_{val}) = \text{softmax}(f(Q, K))V_{val}, \quad (8)$$

where $f(Q, K)$ denotes the attention function. As we know, *additive attention* [13] and *dot-product (multiplicative) attention* are the two most frequently used attention functions, and their definitions are described as follow:

$$f_{add}(Q, K) = \tanh(\mathbf{W}_Q Q + \mathbf{W}_K K), \quad (9)$$

$$f_{mul}(Q,K) = QK^T. \tag{10}$$

Generally speaking, the two attention functions have similar computation complexity in theory. On the one hand, additive attention utilizes a feed-forward neural network with a single hidden layer to calculate the alignment function. On the other hand, dot-product attention implemented using optimized matrix multiplication operation is much faster and more space-efficient in practice [14]. Considering the advantages of dot-product attention, in this work, we also calculate the attention weights using dot-product attention.

4 Attention-Based Spatiotemporal LSTM Network

This section consists of three sub-sections: (1) we introduce a spatiotemporal LSTM (ST-LSTM) network as our base network; (2) we then detail the proposed attention-based spatiotemporal LSTM network (ATST-LSTM); and (3) finally, we present the learning procedure of ATST-LSTM.

4.1 Spatiotemporal LSTM Network

Considering the effect of spatiotemporal contextual information on human real-world check-in activities, modeling the geographical influence and temporal dynamics is essential to predict the next destination of a user on LBSNs. To model such information more effectively, we propose a base network called spatiotemporal LSTM (ST-LSTM). The fundamental idea of ST-LSTM is to learn the non-linear dependency representation over POIs and the spatiotemporal contexts from historical check-in activities. Moreover, we define a spatial feature vector $\mathbf{l}^u_{t_i}$ and a temporal feature vector $\mathbf{t}^u_i$ to incorporate spatiotemporal contextual information.

We further employ the geographical distance $l^u_{t_i} - l^u_{t_{i-1}}$ and the time interval $t_i - t_{i-1}$ to define the spatial feature and the temporal feature, respectively. At time point t_i, we first embed POI IDs into a latent space. Then, ST-LSTM takes the embedded vector with the spatial and temporal features, a triple $(\mathbf{v}^u_{t_i}, \mathbf{l}^u_{t_i}, \mathbf{t}^u_i)$, as input at each time step. In this way, the output of ST-LSTM represents the cumulative influence of the information of POIs and spatiotemporal contexts from the past check-ins. Fig. 3 illustrates the architecture of ST-LSTM.

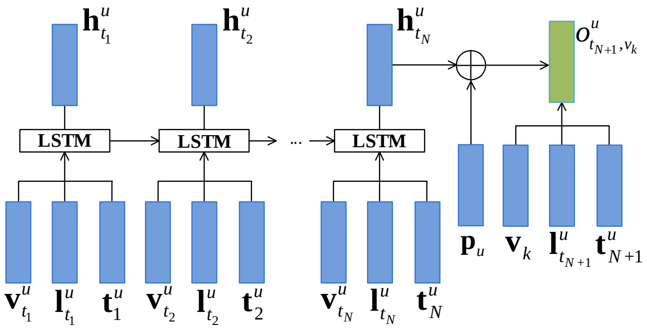

$\mathbf{h}^u_{t_1}$
$\mathbf{h}^u_{t_2}$
$\mathbf{h}^u_{t_N}$
$o^u_{t_{N+1},v_k}$
LSTM
LSTM
...
LSTM
$\mathbf{v}^u_{t_1}$ $\mathbf{l}^u_{t_1}$ $\mathbf{t}^u_1$
$\mathbf{v}^u_{t_2}$ $\mathbf{l}^u_{t_2}$ $\mathbf{t}^u_2$
$\mathbf{v}^u_{t_N}$ $\mathbf{l}^u_{t_N}$ $\mathbf{t}^u_N$
$\mathbf{p}_u$ $\mathbf{v}_k$ $\mathbf{l}^u_{t_{N+1}}$ $\mathbf{t}^u_{N+1}$

Fig. 3. The architecture of ST-LSTM.

In the hidden layer of ST-LSTM, we update each hidden vector $\mathbf{h}^u_{t_i}$ after receiving the current input and the memory $\mathbf{h}^u_{t_{i-1}}$ from the past check-in activities. In ST-LSTM, we have

$$\mathbf{h}^u_{t_i} = \text{LSTM}(\mathbf{W}_v\mathbf{v}^u_{t_i} + \mathbf{W}_l\mathbf{l}^u_{t_i} + \mathbf{W}_t\mathbf{t}^u_i, \mathbf{h}^u_{t_{i-1}}), \tag{11}$$

where $\mathbf{W}_v \in \mathbb{R}^{d\times d}$, $\mathbf{W}_l \in \mathbb{R}^{d\times d}$, and $\mathbf{W}_t \in \mathbb{R}^{d\times d}$ are transition matrices. The learned hidden vector $\mathbf{h}^u_{t_i}$ is a dynamic component of ST-LSTM and can be regarded as the representation of user u at time point t_i. In essence, it reflects dynamic user preferences for POIs under different spatial and temporal contexts.

Assuming that some components may encode essential fixed (or inherent) properties such as the profile and long-term preference of a user, in ST-LSTM, we also design a stationary component $\mathbf{p}_u$. Hence, in this study, the user interest is defined as a function of both dynamic state $\mathbf{h}^u_{t_i}$ and stationary state $\mathbf{p}_u$. We then recommend POIs for target users by calculating the dot-product of user and POI representations, which is similar to those previous studies using matrix factorization. Finally, the predicted probability that user u visits POI v_k at time point t_{N+1} can be obtained by the following operation:

$$o^u_{t_{N+1},v_k} = (\mathbf{W}_N\mathbf{h}^u_{t_N} + \mathbf{W}_p\mathbf{p}_u)^T(\mathbf{W}_v\mathbf{v}_k + \mathbf{W}_l\mathbf{l}^u_{t_{N+1}} + \mathbf{W}_t\mathbf{t}^u_{N+1}), \tag{12}$$

where $\mathbf{W}_N \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_p \in \mathbb{R}^{d\times d}$ are the parameters of the output layer, $\mathbf{W}_N\mathbf{h}^u_{t_N} + \mathbf{W}_p\mathbf{p}_u$ represents the user representation, and$\mathbf{W}_v\mathbf{v}_k + \mathbf{W}_l\mathbf{l}^u_{t_{N+1}} + \mathbf{W}_t\mathbf{t}^u_{N+1}$represents the POI representation. Note that $\mathbf{l}^u_{t_{N+1}}$ is determined by the locations of v_k and $v^u_{t_N}$.

4.2 Attention-Based Spatiotemporal LSTM Network

Intuitively speaking, not all historical check-ins are related equally to a user's next-step behavior. In other words, we need to pay more attention to the informative ones. However, the standard LSTM and ST-LSTM networks cannot detect which part of their inputs is critical to next POI recommendation. We design an attention mechanism and propose an attention-based spatiotemporal LSTM (ATST-LSTM) network to address the issue mentioned above. ATST-LSTM is designed to capture different correlations between past check-in behaviors regarding next-step user behavior. By using the attention mechanism, ATST-LSTM can also help to select the representative check-ins that characterize user preference, as well as to assign them different weights. Hence, we can integrate the representations of informative check-ins to describe user interest in an efficient way.

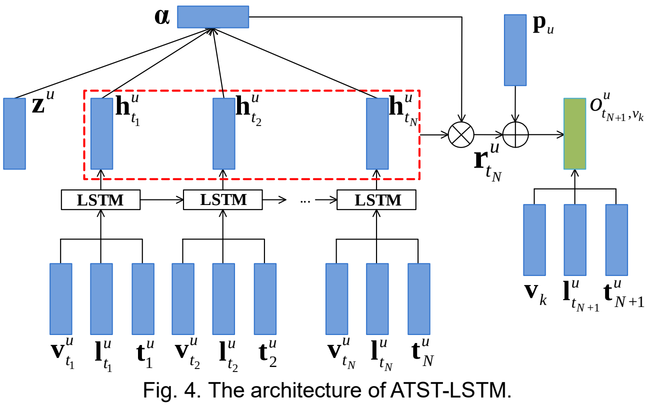

$\boldsymbol{\alpha}$
$\mathbf{p}_u$
$\mathbf{z}^u$
$\mathbf{h}^u_{t_1}$
$\mathbf{h}^u_{t_2}$
$\mathbf{h}^u_{t_N}$
$\mathbf{r}^u_{t_N}$
$o^u_{t_{N+1},v_k}$
LSTM
LSTM
...
LSTM
$\mathbf{v}_k$ $\mathbf{l}^u_{t_{N+1}}$ $\mathbf{t}^u_{N+1}$
$\mathbf{v}^u_{t_1}$ $\mathbf{l}^u_{t_1}$ $\mathbf{t}^u_1$
$\mathbf{v}^u_{t_2}$ $\mathbf{l}^u_{t_2}$ $\mathbf{t}^u_2$
$\mathbf{v}^u_{t_N}$ $\mathbf{l}^u_{t_N}$ $\mathbf{t}^u_N$

Fig. 4. The architecture of ATST-LSTM.

Fig. 4 illustrates the architecture of ATST-LSTM. Let $H \in \mathbb{R}^{d \times N}$ be a matrix which consists of all hidden vectors $\{\mathbf{h}^u_{t_1}, \mathbf{h}^u_{t_2}, \ldots, \mathbf{h}^u_{t_N}\}$ generated by ATST-LSTM, where d indicates the dimension of hidden vectors and N represents the length of the input check-in sequence. By using the attention mechanism, ATST-LSTM generates an attention weight vector $\boldsymbol{\alpha}$ and then aggregates hidden vectors of all check-ins $\{\mathbf{h}^u_{t_i}\}$ to produce a weighted hidden representation $\mathbf{r}$, described as below:

$$\mathbf{r}^u_{t_N} = \sum_{i=1}^{N} \alpha_i \, \mathbf{h}^u_{t_i}. \tag{13}$$

We then introduce how we obtain the attention weight vector in detail. For each $\mathbf{h}^u_{t_i}$, the corresponding weight α_i measures how well the ith historical check-in behavior matches with next-step check-in behavior. More specifically, we calculate this parameter using the following equation:

$$\alpha_i = \frac{\exp(f(\mathbf{h}^u_{t_i}, \mathbf{z}^u))}{\sum_{i=1}^{N} \exp(f(\mathbf{h}^u_{t_i}, \mathbf{z}^u))}, \tag{14}$$

where $f(\mathbf{h}^u_{t_i}, \mathbf{z}^u)$ is the attention function. As mentioned above, we use the dot-product attention as the attention function in this study. Only in the case of large d, the additive attention outperforms the dot-product attention. As with the work of Vaswani *et al* [14], we also define the attention function (see (10)) with a scale:

$$f\left(\mathbf{h}^u_{t_i}, \mathbf{z}^u\right) = \frac{\mathbf{h}^u_{t_i} (\mathbf{z}^u)^T}{\sqrt{d}}, \tag{15}$$

where $\mathbf{z}^u$ is a context vector of user u. Inspired by the idea used in memory networks [41], this parameter denotes the query of our attention model and can be deemed as an abstract representation of the query *"what is the informative check-in for the current behavior prediction"* over all historical check-ins. In addition, the context vector can be considered as a training parameter and is learned in the training process. Finally, the predicted probability that user u visits POI v_k at time point t_{N+1} is calculated by the following operation:

$$o^u_{t_{N+1}, v_k} = (\mathbf{W}_N \mathbf{r}^u_{t_N} + \mathbf{W}_p \mathbf{p}_u)^T (\mathbf{W}_v \mathbf{v}_k + \mathbf{W}_l \mathbf{l}^u_{t_{N+1}} + \mathbf{W}_t \mathbf{t}^u_{N+1}), \tag{16}$$

where the definitions of $\mathbf{W}_N$ and $\mathbf{W}_p$ refer to (12).

4.3 Network Training

The dataset used in this work consists of a set of triplets sampled from the user-POI data, each of which includes one user and a pair of POIs where one POI is positive (or called observed) and the other one is negative (or called unobserved). In this work, we choose the Bayesian Personalized Ranking (BPR) [27] instead of the point-wise loss used in CF, to define loss function for network parameter learning. Since BPR learns a pair-wise ranking loss to train recommender systems, it is capable of exploiting the unobserved user-POI data more effectively. Besides, BPR considers the relative order of the predictions for pairs of POIs, according to an underlying assumption that each user prefers the observed POI over all unobserved POIs.

We then use the maximum a posterior (MAP) estimation to learn the parameters of ATST-LSTM, described as follows:

$$p(u, t, v \succ v') = g(o^u_{t,v} - o^u_{t,v'}), \tag{17}$$

where v and v' represent a positive POI and a negative POI, respectively, and $g(\cdot)$ denotes a nonlinear function defined as

$$g(x) = \frac{1}{1+e^{-x}}. \tag{18}$$

By integrating the loss function and a regularization term, we can solve the objective function of our network for next POI recommendation as follows:

$$\begin{aligned} J &= -\sum_{(v,v')} \ln p(u, t, v \succ v') + \frac{\lambda}{2} \|\Theta\|^2 \\ &= \sum_{(v,v')} \ln(1 + e^{-(o^u_{t,v} - o^u_{t,v'})}) + \frac{\lambda}{2} \|\Theta\|^2. \end{aligned} \tag{19}$$

where λ is used to determine the power of regularization and Θ is the parameter set.

The output of ATST-LSTM is a set of scores for POIs, corresponding to their likelihood of being the next POI in each sequence. The BPR loss function requires the pairs of one score for the target item (i.e., the actual next POI) and the other score for a negative sample (i.e., any POI except the target item). It is often impractical to calculate scores for all pairs since this will make the network unscalable [42]. Therefore, we use a sampling mechanism and compute the scores for only a subset of POIs during the training process. Because the geographical information of POIs would have a significant impact on the analysis of user's check-in behavior, in this study, negative samples are sampled from the POIs located in the same city as positive samples. If negative samples are not enough to support the training process, we utilize the popularity-based sampling method [42] to generate the remaining negative samples further.

Algorithm 1: Training of ATST-LSTM

```
Input: Set of users $U$ and set of historical check-in sequences $C^U$
Output: ATST-LSTM model $\{M\}_u$
//construct training instances
Initialize $D = \cup_u D^u = \emptyset$;
For each user $u$ in $U$ do
    For each check-in trajectory $S^u_i$ in $C_u$ do
        Get the set of negative samples $\{v'^u_{t_k}\}$ by the method [42];
        For each check-in activity $c^u_{t_k}$ in $S^u_i$ do
            Compute the embedding vector $\mathbf{v}^u_{t_k}$ of POI $v^u_{t_k}$;
            Compute the corresponding vectors $\mathbf{l}^u_{t_k}$ and $\mathbf{t}^u_k$;
        End for
        Add a training instance $(\{(\mathbf{v}^u_{t_k}, \mathbf{l}^u_{t_k}, \mathbf{t}^u_k)\}, \{v'^u_{t_k}\})$ to $D^u$;
    End for
End for
//train the model
Initialize the parameter set $\Theta$;
While (exceed(maximum number of iterations)==FALSE) do
    For each user $u$ in $U$ do
        Randomly select a batch of instances $D^u_b$ from $D^u$;
        Find $\Theta$ minimizing the objective (19) with $D^u_b$;
    End for
End while
Output the learned ATST-LSTM model $\{M\}_u$;
```

The parameter set of a standard LSTM contains $\{\mathbf{W}_i, \mathbf{b}_i, \mathbf{W}_f, \mathbf{b}_f, \mathbf{W}_o, \mathbf{b}_o, \mathbf{W}_c, \mathbf{b}_c, \mathbf{W}_s, \mathbf{b}_s\}$. The dimension of $\mathbf{W}_i$, $\mathbf{W}_f$, $\mathbf{W}_o$, and $\mathbf{W}_c$ changes along with various models. Also, additional parameters in our proposed networks include:

1. ST-LSTM: Its parameter set takes account of the

transition matrices of the inputs $\{\mathbf{W}_l, \mathbf{W}_t, \mathbf{W}_v\}$ and the parameters of the output layer $\{\mathbf{W}_N, \mathbf{W}_p\}$ naturally.

2. ATST-LSTM: The set of its parameters includes the transition matrices of the inputs $\{\mathbf{W}_l, \mathbf{W}_t, \mathbf{W}_v\}$, context vector $\mathbf{z}^u$, and the parameters of the output layer $\{\mathbf{W}_N, \mathbf{W}_p\}$.

Since it has been proven that AdaGrad [43] can promote the robustness of stochastic gradient descent (SGD) remarkably in a distributed environment and has been widely used in large-scale learning tasks, we use AdaGrad to optimize the network parameters in this study. Also, AdaGrad can adapt the learning rate to different parameters. For example, the parameters that change infrequently have a higher learning rate than frequently-changing parameters. *Algorithm 1* in pseudocode outlines the whole training process of ATST-LSTM. We first construct the training instances from the original sequence data (see lines 1-11). Then, we train ATST-LSTM using backpropagation and AdaGrad (see lines 12-18).

5 Experimental Setups

5.1 Research Questions

Aiming at evaluating the effectiveness of the proposed ATST-LSTM network, our work attempts to answer the following two research questions:

RQ1: *Does ATST-LSTM improve the performance in next POI recommendation by using the attention mechanism?* In other words, does it perform better than baseline methods such as ST-RNN [9] and ST-LSTM?

RQ2: *Does ATST-LSTM perform better than other similar methods using attention models?* In other words, does it outperform baseline methods such as MANN [48]?

5.2 Data Collection and Preprocessing

We conducted a few experiments for evaluation based on two publicly-available LBSN datasets [44], i.e., Gowalla and Brightkite. Both the two datasets provide user check-ins. In this study, a check-in record is a quadri-tuple composed of a user, a POI, the geographical location of the POI, and the corresponding check-in timestamp. All the check-in records in these two datasets were treated as user sequences. As with the work of Cheng *et al.* [1], we divided each sequence wherever the time interval between any two successive check-in records was more than six hours into different check-in trajectories. Also, we performed a preprocessing step on both the two datasets to filter out inactive users and unpopular POIs. In particular, we removed all the users whose check-ins were fewer than twenty and all the POIs where check-ins were fewer than ten from the two datasets.

After the above preprocessing, the average numbers of trajectories per user in the two datasets are 2.2 and 9.1, respectively. It is worth noting that about 90% and 75% of users whose check-in sequences were less than five exist in the two datasets. Apparently, for those users who have few check-in sequences, it is difficult to predict their next move because there is a specific cold-start problem. We then removed all the users with fewer than five check-in trajectories from the two datasets and obtained 47,655 and 111,328 sub-trajectories, respectively. In the meantime, we augmented check-in trajectories using data augmentation to overcome the cold-start problem. Following the work of Tan *et al.* [45], we treated all the prefixes of the original input trajectories as new training trajectories and finally obtained 349,856 and 839,890 sub-trajectories, respectively, for the two datasets used in our experiments. Table 2 presents the statistics of the preprocessed datasets.

TABLE 2. Statistics of The Preprocessed Datasets.

Dataset	#Users	#Check-ins	#Locations	#Sub-trajectories
Gowalla	2,874	445,166	60,534	349,856
Brightkite	3,277	1,062,465	22,789	839,890

5.3 Baseline Approaches

To validate the effectiveness of ATST-LSTM, we compared it with ST-LSTM and the following eight competing baseline approaches:

1. *PMF* [14]: This method is designed based on conventional probabilistic matrix factorization over the user-POI matrix.
2. *FPMC-LR* [1]: It is a successive POI recommendation method that models personalized sequential transitions using Markov chains. Moreover, this method is an extension of FPMC [36] with the geographical constraint.
3. *PRME-G* [7]: This method embeds users and POIs into the same latent space to model transition patterns of users. Besides, it utilizes the geographical influence via a simple weighing scheme.
4. *Rank-GeoFM* [46]: This method is a ranking-based geographical factorization approach, which learns the embeddings of users and POIs by fitting the frequency of user check-ins. Also, it incorporates both the temporal context and geographical influence via a weighting scheme.
5. *RNN* [47]: It is an advanced temporal prediction approach, which has been applied in advertisements click prediction and word embedding successfully. In this study, we employ POI IDs to construct the input feature for this method.
6. *LSTM* [10]: Standard LSTM units are an extension of the RNN model. An LSTM unit has a (memory) cell and three multiplicative gates to allow long-term dependency learning.
7. *ST-RNN* [9]: This method is an RNN-based model for next POI recommendation. It incorporates both the temporal context and geographical information within the recurrent architecture.
8. *MANN* [48]: MANN is a new memory-augmented neural network combining with collaborative filtering for item recommendation. Besides, this approach also leverages the attention mechanism in memory networks. However, it is not designed for the next-POI recommendation scenarios.

Table 3 summarizes the ten different approaches used in our experiments. Generally speaking, they fall within the scope of four categories of commonly-used methods. First, standard CF recommendation approaches, such as

PMF. Second, sequential POI recommendation approaches using Markov chains (such as FPMC-LR) and RNNs (such as ST-RNN and ST-LSTM). Third, hybrid approaches that integrate spatial information or temporal information via a weighting scheme, such as PRME-G and Rank-GeoFM. Fourth, the state-of-the-art approaches that leverage the attention mechanism, such as MANN.

All the approaches under discussion predict the probability that a user will visit a POI at a time point via the calculation of the dot product between user representation and POI representation. In general, they differ in modeling user representation at different time points. Since it is difficult to analyze the time complexity of each approach directly, we estimate the complexity of calculating user representation in each approach so as to evaluate their efficiency. These approaches are assumed to have the same dimension of hidden variables (denoted by m) and the same size of samples (denoted by n). Because PMF uses a static user representation, its complexity is $O(nm)$. FPMC-LR and PRME-G use a combination of the latest POI representation and a static user representation to model users, and their complexities are $O(2nm)$. Rank-GeoFM utilizes the nearest k POIs of each check-in record to model user preference, and its complexity is $O((k+2)nm)$. For the remaining six approaches, they predict user preference by using l historical check-ins each time. Since MANN is essentially an extension of FPMC, its complexity approximates $O(nl(2m^2+10m)+2nm+nl)$. For RNN and ST-RNN, their complexities are $O(nl(2m^2+2m)+2nm)$ and $O(nl(3sm^2+2m)+2nm)$, respectively. Here, s is the length of each time window defined in ST-RNN. The complexity of LSTM is $O(nl(8m^2+13m)+2nm)$. Compared with LSTM, the complexity of ST-LSTM is equal to $O(nl(9m^2+17m)+2nm)$ because it integrates spatiotemporal information. Considering the time cost of calculating the attention weights, the complexity of ATST-LSTM increases compared with ST-LSTM, and the complexity value reaches $O(nl(9m^2+20m)+2nm)$. In summary, although ATST-LSTM is more complex than the baseline approaches, it has the same order of magnitude ($O(nlm^2)$) with the approaches based on RNNs and LSTMs in time complexity.

TABLE 3. ALL THE APPROACHES USED IN OUR EXPERIMENTS.

Property	PMF	FPMC-LR	PRME-G	Rank-GeoFM	RNN	LSTM	ST-RNN	ST-LSTM	MANN	ATST-LSTM
SE	×	✓	✓	×	✓	✓	✓	✓	✓	✓
SP	×	✓	✓	✓	×	×	✓	✓	×	✓
TE	×	×	×	✓	×	×	✓	✓	×	✓
AT	×	×	×	×	×	×	×	×	✓	✓
Time	$O(nm)$	$O(nm)$	$O(nm)$	$O(nm)$	$O(nlm^2)$	$O(nlm^2)$	$O(nlsm^2)$	$O(nlm^2)$	$O(nlm^2)$	$O(nlm^2)$

SE, SP, TE, and AT represent whether the given approach considers the sequential influence, spatial information, temporal information, and attention mechanism, respectively. For each of the ten approaches, we only present the order of magnitude of approximate time complexity due to space limitation. Here, n, m, l, and s denote the size of samples, the number of dimensions of hidden variables, the number of historical check-ins, and the length of time windows, respectively.

5.4 Evaluation Metrics

As we know, $P@k$ (short for *Precision@k*), $R@k$ (short for *Recall@k*), and $F_1@k$ (short for *F1-score@k*) are three favorite evaluation metrics for ranking learning. In this study, the three metrics are formally defined as

$$P@k = \frac{1}{N}\sum_{u=1}^{N} P_u@k = \frac{1}{N}\sum_{u=1}^{N}\frac{|S_u(k)\cap V_u|}{k}, \qquad (20)$$

$$R@k = \frac{1}{N}\sum_{u=1}^{N} R_u@k = \frac{1}{N}\sum_{u=1}^{N}\frac{|S_u(k)\cap V_u|}{|V_u|}, \qquad (21)$$

$$F_1@k = \frac{1}{N}\sum_{u=1}^{N} F_{1_u}@k = \frac{1}{N}\sum_{u=1}^{N}\frac{2\cdot P_u@k\cdot R_u@k}{P_u@k+R_u@k}, \qquad (22)$$

where $S_u(k)$ denotes the set of the top k POIs recommended to user u and V_u denotes the set of POIs that the user actually visited at the next time stamp in the test set. Note that we present the results of the three evaluation metrics with the setting of k = 5 and 10 in this paper.

5.5 Configurations

Our experiments were conducted on a Lenovo ThinkStation P910 Workstation with dual processors (2 x Intel Xeon E5-2660 v4, 2.0 GHz) and one graphics processing unit (GPU, NVIDIA TITAN X Pascal, 12GB). The operating system of the workstation was Microsoft Windows 7 (64-bit). The code used in our experiments was written in Python 3.5. In the meantime, we used TensorFlow[6] 1.2.0 as a machine learning framework for the experiments.

We adopted a single-layer LSTM architecture and set the sequence length of LSTMs to the maximum length of input check-in sequences. For each user, we divided the user's check-in sub-trajectories sorted in chronological order into a training set and a test set, i.e., the first 90% of sub-trajectories of each user were used as the training set and the remaining 10% as the test set, on both the two datasets. For each target POI, the number of negative samples was set to 500. The initial learning rate was set to 0.01, the size of each batch was set to 30, and the dropout rate was set to 0.2 in order to avoid overfitting. The source code of the proposed method is publicly available for download at https://github.com/drhuangliwei/An-Attention-based-Spatiotemporal-LSTM-Network-for-Next-POI-Recommendation.

6 RESULTS AND DISCUSSION

6.1 Comparison of Recommendation Performance

Table 4 presents a comparison of the ten methods' recommendation results on the two datasets. The numbers shown in bold in Table 4 represent the best performance of each column in this table.

[6] https://www.tensorflow.org

TABLE 4. RECOMMENDATION PERFORMANCE COMPARISON.

Methods \ Metrics	Gowalla						Brightkite					
	P@5	*R*@5	F_1@5	*P*@10	*R*@10	F_1@10	*P*@5	*R*@5	F_1@5	*P*@10	*R*@10	F_1@10
PMF	0.0167	0.0869	0.0280	0.0129	0.1342	0.0235	0.0219	0.1123	0.0367	0.0186	0.1895	0.0339
FPMC-LR	0.0214	0.1140	0.0360	0.0226	0.2323	0.0412	0.0370	0.1897	0.0619	0.0262	0.2674	0.0477
PRME-G	0.0326	0.1690	0.0547	0.0277	0.2796	0.0504	0.0459	0.2342	0.0768	0.0309	0.3123	0.0562
Rank-GeoFM	0.0341	0.1756	0.0571	0.0279	0.2864	0.0508	0.0483	0.2458	0.0807	0.0327	0.3321	0.0595
RNN	0.0322	0.1649	0.0539	0.0261	0.2686	0.0476	0.0460	0.2345	0.0769	0.0281	0.2875	0.0512
LSTM	0.0380	0.1901	0.0634	0.0274	0.2827	0.0500	0.0538	0.2732	0.0899	0.0306	0.3132	0.0558
ST-RNN	0.0592	0.2924	0.0985	0.0339	0.3314	0.0615	0.0832	0.4123	0.1385	0.0451	0.4537	0.0820
ST-LSTM	0.0697	0.3421	0.1158	0.0382	0.3714	0.0693	0.0920	0.4554	0.1531	0.0519	0.5143	0.0943
MANN	0.0423	0.2113	0.0704	0.0302	0.3023	0.0550	0.0590	0.2923	0.0982	0.0338	0.3334	0.0614
MANN+ST	0.0752	0.3713	0.1251	0.0392	0.3883	0.0712	0.0969	0.4823	0.1614	0.0539	0.5334	0.0979
ATST-LSTM	**0.0791**	**0.3902**	**0.1315**	**0.0416**	**0.4088**	**0.0755**	**0.1042**	**0.5162**	**0.1734**	**0.0594**	**0.5876**	**0.1079**

MANN+ST represents that we incorporated the spatial and temporal information of the experimental datasets into the original MANN method.

For both the two datasets, PMF was the worst performer regarding the three metrics because the user-POI matrices were very sparse (i.e., the data sparsity problem). Besides, PMF did not take into account additional information such as temporal context and geographical influence. Although PFMC-LR performed slightly better than PMF by integrating distance information, it did not consider other useful information like temporal information.

Compared with PMF and PFMC-LR, Rank-GeoFM incorporated both temporal context and geographical influence within their models, PRME-G incorporated both sequential influence and geographical influence, and they utilized different ranking-based optimization strategies. Therefore, these approaches alleviated the data sparsity problem to a certain extent by making use of unobserved data. Even so, ATST-LSTM (or even ST-RNN) outperformed them significantly, which suggests that the RNN architecture of ATST-LSTM can model user's spatial, sequential behaviors better.

It is worth noting that these hybrid approaches did not perform worse than RNN and LSTM. This result indicates that modeling temporal and spatial contexts is indeed useful for the task of next POI recommendation. In other words, a good network architecture is not enough to obtain excellent results, so that we have to take into account more spatial and temporal information of human check-in behavior. That is why ST-RNN and ST-LSTM outstripped RNN and LSTM.

Answer to RQ1: As shown in Table 4, ST-RNN and ST-LSTM are two best performers amongst the approaches without the attention mechanism. Compared with ST-RNN, the *P*@5, *R*@5, F_1@5, *P*@10, *R*@10, and F_1@10 values of ST-LSTM were increased, on average, by 14.16%, 13.73%, 14.09%, 13.88%, 12.72%, and 13.78%, respectively, on the two datasets. The performance improvements of ST-LSTM may be due to the advantage of LSTMs over RNNs, i.e., the primary goal of LSTMs is to alleviate the exploding or vanishing gradients problem. Compared with ST-LSTM, ATST-LSTM also yielded 13.49%, 14.06%, 13.58%, 8.90%, 10.07%, and 9.01% improvements in *P*@5, *R*@5, F_1@5, *P*@10, *R*@10, and F_1@10, respectively, on the Gowalla dataset. Also, for the Brightkite dataset, the performance improvements in the evaluation metrics are 13.26%, 13.35%, 13.28%, 14.45%, 14.25%, and 14.43%, respectively. The results mentioned above indicate that *leveraging the attention mechanism can indeed enhance the performance of next POI recommendation*. Moreover, the evidence that MANN outperforms RNN and LSTM also supports this conclusion.

Answer to RQ2: Considering that the original MANN method did not utilize the spatial and temporal information, we then compared ATST-LSTM and MANN+ST (i.e., a variant of MANN). In the MANN+ST method, we concatenated POIs and the corresponding temporal and geographical contexts to represent check-in records and embedded them into a compact vector representation to generate user memory embeddings. As shown in Table 4, our method also performs better than MANN+ST. The *P*@5, *R*@5, F_1@5, *P*@10, *R*@10, and F_1@10 values of ATST-LSTM are increased, on average, by 6.36%, 6.06%, 6.31%, 8.16%, 7.72%, and 8.12%, respectively, on the two datasets. Although MANN+ST can enrich the user representation by memory networks, it ignores the order of any check-in sequence as well as the interaction between historical check-in records and thus has limited ability to model the sequential patterns of user behaviors. Thus, *our method outperformed MANN, the state-of-the-art approach using the attention mechanism*.

6.2 Sensitive Analysis of Parameters

We then analyzed the effects of different model parameters on the performance of ATST-LSTM. Here, we focused on two critical parameters, i.e., the number of embedding dimensions and the number of negative samples. Because $|V_u| = 1$ in this study, our experiments on the two datasets indicated that *R*@*k* had a strong positive correlation with *P*@*k*. Besides, high recall is more useful than high precision in the next POI recommendation scenarios [6], [7], [8]. As with previous studies [7], [9], we analyzed the effects of the two parameters on *R*@*k* and F_1@*k* due to space limitation.

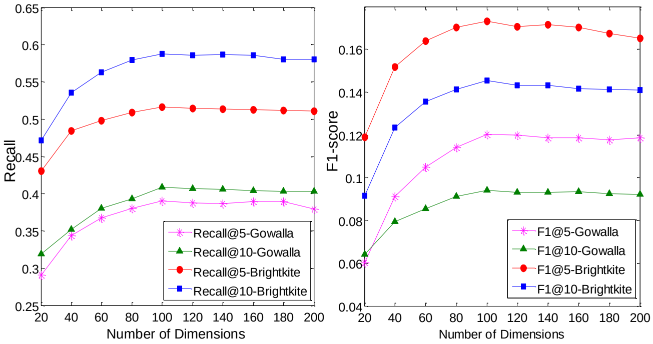
6.2.1 Number of embedding dimensions

The embedding dimension size is an essential factor to affect the performance of ATST-LSTM. A higher number of embedding dimensions suggests a stronger expressive ability, which also, probably, leads to over-fitting.

Fig. 5 presents the performance of the proposed ATST-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2019.2918310, IEEE Transactions on Services Computing

LSTM network with different embedding dimensions regarding *R@k* and *F*$_1$*@k*. It is apparent from this figure that the *Recall* and *F1-score* values of ATST-LSTM gradually increase with the size of embedding dimensions; moreover, the recommendation performance of our model becomes stable when the number of embedding dimensions varies from 100 to 200. Therefore, we set the number of embedding dimensions to 100 in our experiments.



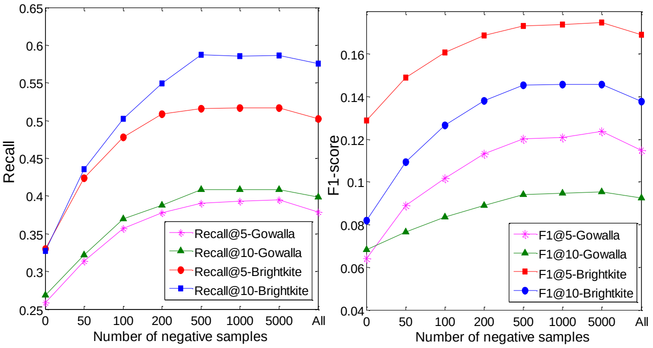
(a) *Recall@k* (b) *F1-Score@k*

Fig. 5. Performance tuning with different embedding dimensions.

6.2.2 Number of negative samples

The number of negative samples is also critical to the recommendation performance of ATST-LSTM. If the negative sample size is small, our model cannot make full use of unobserved data. However, adding more this type of samples will increase the computational complexity of the model. We then experimented with the two datasets to examine the effect of additional negative samples on recommendation performance.

Fig. 6 depicts the performance of ATST-LSTM with the BPR loss. Here, we evaluated its recommendation performance according to different sizes of negative samples. Note that "All" in this figure represents a specific case for the computation of all scores which were used as input to the BPR loss without POI sampling. As the size of negative samples grew, the overall performance of ATST-LSTM gradually increased and then became stable after the negative sample size reached 500. However, the *R@k* and *F*$_1$*@k* values of ATST-LSTM tended to decrease when all POIs in the training set were used (see the "All" in Fig. 6). Therefore, by considering the trade-off between effectiveness and efficiency, we set the number of negative samples to 500 in our experiments.



(a) *Recall@k* (b) *F1-Score@k*

Fig. 6. Performance vs. the number of negative samples.

6.3 Attention Visualization

To understand the attention mechanism used in ATST-LSTM better, we illustrated an example in Fig. 7. This example presented a qualitative analysis of a randomly-selected user's thirteen check-in records in two days. As shown in the upper part of Fig. 7, we visualized the trajectory of the user on a Google map. Moreover, we displayed the attention weights of all the check-ins in this trajectory using a heat map. The color depth of each check-in denotes the size of each weight in the vector of attention weights, i.e., a circle with darker colors indicates that the corresponding check-in behavior is more critical than those with a light color. Besides, the lower part of Fig. 7 gives another visualization of the attention weights. For each of the check-in in the user's trajectory, we showed the POI category and check-in time, as well as the corresponding attention weight represented by a vertical bar.

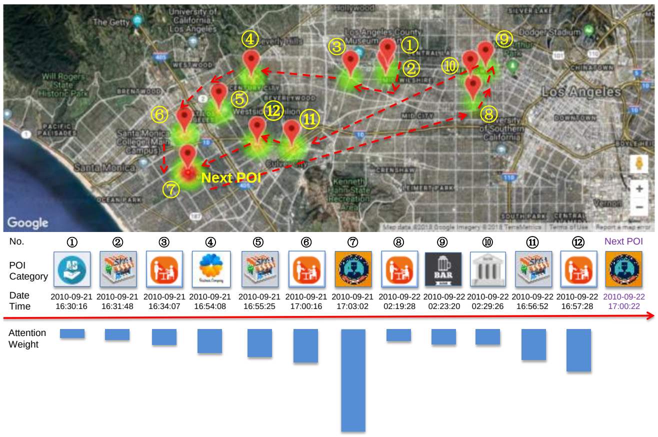


Fig. 7. An example of attention visualization.

According to Fig. 7, we find some interesting behavioral regularities of the user. First, most of the historical check-ins of the user had little impact on his thirteenth check-in (i.e., the next POI recommended by ATST-LSTM). Also, the sequential dependency of the next POI on any POIs visited by the user did not change with their relative positions in the trajectory monotonously. This finding indicates the randomness and uncertainty in an individual's behavior. Second, the seventh check-in had a higher weight value than the other check-ins on predicting the next POI visited by the user after almost one day. The result suggests that the user's check-in behavior appears to be periodic, and the same goes for the fifth and eleventh check-ins. We suspect that the periodic behavior of the user may stem from his habits and customs, as well as a regular study program. Third, some similar historical check-ins tended to have equal attention weights for the same user. For example, since the fifth and eleventh check-ins were alike in time, location, and POI category, they shared similar attention weights in ATST-LSTM. In brief, the results mentioned above suggest that ATST-LSTM can indeed model users' sequential patterns more effectively via the attention mechanism.

6.4 Threats to Validity

In this subsection, we discuss some potential threats to the validity of our study.

Internal validity is a form of experimental validity. The threats to the internal validity of our study include two main aspects: data selection and parameter setting.

Selection bias is one of the most common threats to internal validity. Considering the characteristics of next POI recommendation, we constructed user check-in trajectories after removing inactive users and unpopular POIs. We then augmented the obtained check-in trajectories using the data augmentation approach [45] to alleviate the cold-start problem. Besides, for user check-in sub-trajectories, we set the splitting proportion of training data to test data to 90:10. The primary objective of such data processing is to improve the recommendation performance of all the approaches under discussion. Also, ATST-LSTM used a popularity-based sampling method [42] when calculating the scores of the BPR loss function. It is a useful trick used to improve recommendation performance as well as efficiency further [49]. Therefore, we have to admit that the recommendation performance of our approach will decrease without an appropriate size of negative samples (see the analysis in 6.2.2).

In our experiment, we trained different types of baseline methods based on their default hyper-parameter settings. As we know, there are also several implicit tricks, such as fine-tuning, in the baseline approaches based on deep neural networks, even though the source code is freely available. Therefore, we cannot ensure that these methods can achieve the optimal performance stated in their original papers on the two datasets.

External validity refers to the extent of the generalizability of a study to other situations as well as to other people. The threats to the external validity of our study include three main aspects: new datasets, new baselines, and new application scenarios.

The recommendation performance of our method on other LBSNs (e.g., Twitter[7] and Sina Weibo[8]) is yet to be tested, which is one of the leading threats to the external validity of our study. Moreover, the scalability of ATST-LSTM up to large-scale datasets remains to be determined, although we estimated the time complexity of our method in theory. We expect that it can be used as an off-line recommendation system to recommend the next POI after further optimization.

In this study, we designed an elaborate comparison of ATST-LSTM with six commonly-used methods and two state-of-the-art approaches to next POI recommendation. Another primary threat to the external validity of our work is that the merit of AST-LSTM over the latest methods for general POI recommendation remains unknown. Even so, we argue that this threat will not affect the conclusion of this study because our work has its own research goal and questions distinct from those studies of general POI recommendation.

Last but not least, the primary goal of ATST-LSTM is to recommend the next POI for target users. In this study, we do not care whether the recommended next POI is new (or unvisited) or not. Therefore, the performance of our approach yet remains mostly unexplored in the scenario of recommending the next new POI. Because the next new POI recommendation problem is a far more challenging task, we plan to investigate it systematically in the future.

7 Conclusion

In recent years, next POI recommendation has attracted much attention in the fields of LBS and recommender systems. In this paper, we propose a novel Attention-based Spatiotemporal LSTM (ATST-LSTM) approach to tackle the next POI recommendation problem. More specifically, ATST-LSTM focuses on critical parts of a user check-in sequence by leveraging the attention mechanism, and it can model spatial and temporal contexts better by capturing various correlations between historical check-ins corresponding to the current situation. Besides, we experimented with two publicly-available LBSN datasets (i.e., Gowalla and Brightkite) to validate the effectiveness of ATST-LSTM. The experimental results indicate that ATST-LSTM outperforms the other two state-of-the-art methods (ST-RNN [9] and MANN [48]) for next POI recommendation regarding three commonly-used metrics, i.e., *Precision*, *Recall*, and *F1-score*.

It is worth noting that our work could be used in the application scenarios of location-based advertising and mobile recommendation for precision marketing [50]. Also, it may contribute to the development of visual (trip) assistants, each of which can create personalized user profiles by learning individual historical check-in records automatically and make a suggestion on some possible POIs in real time. To this end, our future work will enrich and optimize the model of ATST-LSTM by considering more information, such as semantic information to improve its performance and scalability. Besides, we plan to extend the proposed method to investigate further the next new POI recommendation problem.

Acknowledgment

This work was partially supported by the National Basic Research Program of China (No. 2014CB340404), National Key Research and Development Program of China (No. 2017YFB1400602), National Science Foundation of China (Nos. 61672387 and 61702378), and Wuhan Yellow Crane Talents Program for Modern Services Industry. Yutao Ma is the corresponding author of this paper.

References

[1] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks," *Proc. Twenty-Sixth AAAI Conf. Artificial Intelligence (AAAI '12)*, pp. 17-23, 2012, available at https://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/viewPaper/4748.

[2] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring Temporal Effects

[7] https://twitter.com

[8] https://weibo.com

for Location Recommendation on Location-Based Social Networks," *Proc. Seventh ACM Conf. Recommender Systems (RecSys '13)*, pp. 93-100, 2013, doi:10.1145/2507157.2507182.

[3] M. Ye, P. Yin, W.-C. Lee, and D. Lee, "Exploiting Geographical Influence for Collaborative Point-of-Interest Recommendation," *Proc. Thirty-Fourth Intl ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '11)*, pp. 325-334, 2011, doi:10.1145/2009916.2009962.

[4] S. Zhao, I. King, and M. R. Lyu, "A Survey of Point-of-Interest Recommendation in Location-Based Social Networks," *arXiv*, 2016, arXiv:1607.00647.

[5] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel, "Recommendations in Location-Based Social Networks: A Survey," *GeoInformatica*, vol. 19, no. 3, pp. 525-565, 2015, doi:10.1007/s10707-014-0220-8.

[6] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where You Like to Go Next: Successive Point-of-Interest Recommendation," *Proc. Twenty-Third Intl Joint Conf. Artificial Intelligence (IJCAI '13)*, pp. 2605-2611, 2013, available at http://www.ijcai.org/Proceedings/13/Papers/384.pdf.

[7] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized Ranking Metric Embedding for Next New POI Recommendation," *Proc. Twenty-Fourth Intl Joint Conf. Artificial Intelligence (IJCAI '15)*, pp. 2069-2075, 2015, available at http://www.ijcai.org/Proceedings/15/Papers/293.pdf.

[8] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King, "STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation," *Proc. Thirtieth AAAI Conf. Artificial Intelligence (AAAI '16)*, pp. 315-322, 2016, available at http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12249.

[9] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts," *Proc. Thirtieth AAAI Conf. Artificial Intelligence (AAAI '16)*, pp. 194-200, 2016, available at https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11900.

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997, doi:10.1162/neco.1997.9.8.1735.

[11] C. Yang, M. Sun, W. Zhao, Z. Liu, and E. Y. Chang, "A Neural Network Approach to Jointly Modeling Social Networks and Mobile Trajectories," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, pp. 36:1-36:28, 2017, doi:10.1145/3041658.

[12] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent Models of Visual Attention," *Proc. Ann. Conf. Neural Information Processing Systems 2014 (NIPS '14)*, pp. 2204-2212, 2014, available at http://papers.nips.cc/paper/5542-recurrent-models-of-visual-attention.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXiv*, 2014, arXiv:1409.0473.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," *Proc. Ann. Conf. Neural Information Processing Systems 2017 (NIPS '17)*, pp. 6000-6010, 2017, available at https://papers.nips.cc/paper/7181-attention-is-all-you-need.

[15] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," *Proc. Twenty-First Ann. Conf. Neural Information Processing Systems (NIPS '07)*, pp. 1257-1264, 2007, available at http://papers.nips.cc/paper/3208-probabilistic-matrix-factorization.

[16] J.-D. Zhang and C.-Y. Chow, "iGSLR: Personalized Geo-Social Location Recommendation: A Kernel Density Estimation Approach," *Proc. Twenty-First SIGSPATIAL Intl Conf. Advances in Geographic Information Systems (SIGSPATIAL/GIS '13)*, pp. 324-333, 2013, doi:10.1145/2525314.2525339.

[17] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning Geographical Preferences for Point-of-Interest Recommendation," *Proc. Nineteenth ACM SIGKDD Intl Conf. Knowledge Discovery and Data Mining (KDD '13)*, pp. 1043-1051, 2013, doi:10.1145/2487575.2487673.

[18] H. Gao, J. Tang, X. Hu, and H. Liu, "Content-Aware Point of Interest Recommendation on Location-Based Social Networks," *Proc. Twenty-Ninth AAAI Conf. Artificial Intelligence (AAAI '15)*, pp. 1721-1727, 2015, available at https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9560.

[19] D. Lian, Y. Ge, F. Zhang, N. J. Yuan, X. Xie, T. Zhou, and Y. Rui, "Content-Aware Collaborative Filtering for Location Recommendation Based on Human Mobility Data," *Proc. 2015 IEEE Intl Conf. Data Mining (ICDM '15)*, pp. 261-270, 2015, doi:10.1109/ICDM.2015.69.

[20] L. Huang, Y. Ma, and Y. Liu, "Point-of-Interest Recommendation in Location-Based Social Networks with Personalized Geo-Social Influence," *China Communications*, vol. 12, no. 12, pp. 21-31, 2015, doi:10.1109/CC.2015.7385525.

[21] L. Huang, Y. Ma, Y. Liu, and A. K. Sangaiah, "Multi-Modal Bayesian Embedding for Point-of-Interest Recommendation on Location-Based Cyber-Physical-Social Networks," *Future Generation Computer Systems*, 2017, doi:10.1016/j.future.2017.12.020.

[22] J.-D. Zhang, C.-Y. Chow, and Y. Li, "LORE: Exploiting Sequential Influence for Location Recommendations," *Proc. Twenty-Second ACM SIGSPATIAL Intl Conf. Advances in Geographic Information Systems (SIGSPATIAL/GIS '14)*, pp. 103-112, 2014, doi:10.1145/2666310.2666400.

[23] J. Ye, Z. Zhu, and H. Cheng, "What's Your Next Move: User Activity Prediction in Location-Based Social Networks," *Proc. 2013 SIAM Intl Conf. Data Mining (SDM '13)*, pp. 171-179, 2013, doi:10.1137/1.9781611972832.19.

[24] S. Zhao, I. King, and M. R. Lyu, "Capturing Geographical Influence in POI Recommendations," *Proc. Twentieth Intl Conf. Neural Information Processing (ICONIP '13)*, vol. 2, pp. 530-537, 2013, doi:10.1007/978-3-642-42042-9_66.

[25] R. He, W.-C. Kang, and J. McAuley, "Translation-Based Recommendation," *Proc. Eleventh ACM Conf. Recommender Systems (RecSys '17)*, pp. 161-169, 2017, doi:10.1145/3109859.3109882.

[26] S. Zhao, T. Zhao, I. King, and M. R. Lyu, "Geo-Teaser: Geo-Temporal Sequential Embedding Rank for Point-of-Interest Recommendation," *Proc. Twenty-Sixth Intl Conf. World Wide Web Companion (WWW (Companion Volume) '17)*, pp. 153-162, 2017, doi:10.1145/3041021.3054138.

[27] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback," *Proc. Twenty-Fifth Conf. Uncertainty in Artificial Intelligence (UAI '09)*, pp. 452-461, 2009, available at http://www.auai.org/uai2009/papers/UAI2009_0139_48141db02b9f0b02bc7158819ebfa2c7.pdf.

[28] J. Weston, C. Wang, R. J. Weiss, and A. Berenzweig, "Latent Collaborative Retrieval," *Proc. Twenty-Ninth Intl Conf. Machine Learning (ICML '12)*, p. 12, 2012, available at https://icml.cc/2012/papers/12.pdf.

[29] S. Zhang, L. Yao, and A. Sun, "Deep Learning based Recommender System: A Survey and New Perspectives," *arXiv*, 2017, arXiv:1707.07435.

[30] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *Proc. Twenty-Seventh Ann. Conf. Neural Information Processing Systems (NIPS '13)*, pp. 3111-3119, 2013, available at https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.

[31] N. Zhou, W. Zhao, X. Zhang, J.-R. Wen, and S. Wang, "A General Multi-Context Embedding Model for Mining Human Trajectory Data," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 1945-1958, 2016, doi:10.1109/TKDE.2016.2550436.

[32] X. Liu, Y. Liu, and X. Li, "Exploring the Context of Locations for Personalized Location Recommendations," *Proc. Twenty-Fifth Intl Joint Conf. Artificial Intelligence (IJCAI '16)*, pp. 1188-1194, 2016, available at http://www.ijcai.org/Proceedings/16/Papers/172.pdf.

[33] S. Feng, G. Cong, B. An, and Y. M. Chee, "POI2Vec: Geographical Latent Representation for Predicting Future Visitors," *Proc. Thirty-First AAAI Conf. Artificial Intelligence (AAAI '17)*, pp. 102-108, 2017, available at https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14902.

[34] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation," *Proc. Twenty-Third ACM SIGKDD Intl Conf. Knowledge Discovery and Data Mining (KDD '17)*, pp. 1245-1254, 2017, doi:10.1145/3097983.3098094.

[35] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu, "What Your Images Reveal: Exploiting Visual Contents for Point-of-Interest Recommendation," *Proc. Twenty-Sixth Intl Conf. World Wide Web (WWW '17)*, pp. 391-400, 2017, doi:10.1145/3038912.3052638.

[36] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing Personalized Markov Chains for Next-Basket Recommendation," *Proc. Nineteenth Intl Conf. World Wide Web (WWW '10)*, pp. 811-820, 2010, doi:10.1145/1772690.1772773.

[37] Y. Liu, C. Liu, B. Liu, M. Qu, and H. Xiong, "Unified Point-of-Interest Recommendation with Temporal Interval Assessment," *Proc. Twenty-Second ACM SIGKDD Intl Conf. Knowledge Discovery and Data Mining (KDD '16)*, pp. 1015-1024, 2016, doi:10.1145/2939672.2939773.

[38] E. Palumbo, G. Rizzo, R. Troncy, and E. Baralis, "Predicting Your Next Stop-over from Location-based Social Network Data with Recurrent Neural Networks," *Proc. Second Wksp Recommenders in Tourism co-located with Eleventh ACM Conf. Recommender Systems (RecTour@RecSys '17)*, pp. 1-8, 2017, available at http://ceur-ws.org/Vol-1906/paper1.pdf.

[39] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-Based Models for Speech Recognition," *Proc. Twenty-Eighth Ann. Conf. Neural Information Processing Systems (NIPS '15)*, pp. 577-585, 2015, available at https://papers.nips.cc/paper/5847-attention-based-models-for-speech-recognition.

[40] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," *Proc. Thirty-Second Intl Conf. Machine Learning (ICML '15)*, pp. 2048-2057, 2015, available at http://proceedings.mlr.press/v37/xuc15.html.

[41] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-To-End Memory Networks," *Proc. Twenty-Eighth Ann. Conf. Neural Information Processing Systems (NIPS '15)*, pp. 2440-2448, 2015, available at http://papers.nips.cc/paper/5846-end-to-end-memory-networks.

[42] B. Hidasi and A. Karatzoglou, "Recurrent Neural Networks with Top-k Gains for Session-based Recommendations," *arXiv*, 2017, arXiv:1706.03847.

[43] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121-2159, 2011, available at http://jmlr.org/papers/volume12/duchi11a/duchi11a.pdf.

[44] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and Mobility: User Movement in Location-Based Social Networks," *Proc. Seventeenth ACM SIGKDD Intl Conf. Knowledge Discovery and Data Mining (KDD '11)*, pp. 1082-1090, 2011, doi:10.1145/2020408.2020579.

[45] Y. K. Tan, X. Xu, and Y. Liu, "Improved Recurrent Neural Networks for Session-based Recommendations," *arXiv*, 2016, arXiv:1606.08117.

[46] X. Li, G. Cong, X. Li, T.-A. N. Pham, and S. Krishnaswamy, "Rank-GeoFM: A Ranking based Geographical Factorization Method for Point of Interest Recommendation," *Proc. Thirty-Eighth Intl ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '15)*, pp. 433-442, 2015, doi:10.1145/2766462.2767722.

[47] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu, "Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks," Proc. Twenty-Eighth AAAI Conf. Artificial Intelligence (AAAI 14), pp. 1369-1375, 2014, available at http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8529.

[48] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha, "Sequential Recommendation with User Memory Networks," *Proc. Eleventh ACM Intl Conf. Web Search and Data Mining (WSDM '18)*, pp. 108-116, 2018, doi:10.1145/3159652.3159668.

[49] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational Autoencoders for Collaborative Filtering," *Proc. Twenty-Seventh Intl Conf. World Wide Web (WWW '18)*, pp. 689-698, 2018, doi:10.1145/3178876.3186150.

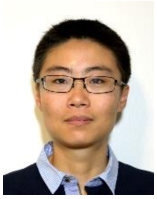
[50] S. Deng, L. Huang, H. Wu, W. Tan, J. Taheri, A. Y. Zomaya, and Z. Wu, "Toward Mobile Service Computing: Opportunities and Challenges," *IEEE Cloud Computing*, vol. 3, no. 4, pp. 32-41, 2016, doi:10.1109/MCC.2016.92.

Liwei Huang received his Ph.D. degree in Computer Science from the PLA University of Science and Technology, China, in 2014. He is currently an engineer at the Beijing Institute of Remote Sensing. Dr. Huang's research focuses on data mining and machine learning. He is now a member of the China Computer Federation (CCF).

Yutao Ma received his Ph.D. degree in Computer Science from Wuhan University, China, in 2007. He is now an associate professor in the School of Computer Science, Wuhan University. Dr. Ma was with the Institute of China Electronic System Engineering Corporation (Beijing) as a post-doctoral fellow and has been a visiting scholar in the Department of Electronic and Computer Engineering, Lehigh University, USA. His research focus is on the development and maintenance of large-scale software service systems. He has published more than fifty peer-reviewed papers and received two best paper awards at international conferences. In addition to a member of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2019.2918310, IEEE Transactions on Services Computing

the ACM and the IEEE, he is now a senior member of the CCF and a member of the CCF Technical Committee on Services Computing (TCSC).



Shibo Wang received her M.S. degree in 2013 and a Ph.D. degree in 2017 in Computer Science from the University of Rochester. She is now a software engineer at the Google. Her research interests include machine learning and computer architecture.

Yanbo Liu received her M.S. degree in Automation from the National University of Defense Technology, China. She is currently an engineer at the Beijing Institute of Remote Sensing. Her research interests include information automation and machine learning.