A Next Location Predicting Approach Based on a Recurrent Neural Network and Self-attention

Jun Zeng(✉), Xin He, Haoran Tang, and Junhao Wen

School of Big Data and Software Engineering, Chongqing University, Chongqing, China
{zengjun, hexin, tanghaoran, jhwen}@cqu.edu.cn

Abstract. On most location-based social applications today, users are strongly encouraged to share activities by checking-in. In this way, vast amounts of user-generated data can be accumulated, which include spatial and temporal information. Much research has been conducted on these data, which enables heightening the understanding of human mobility. Therefore, the next location problem has attracted significant attention and has been extensively studied. In this paper, we propose a next location prediction approach based on a recurrent neural network and self-attention mechanism. Our model can explore sequence regularity and extract temporal feature according to historical trajectories information. We conduct our experiments on the location-based social network (LBSN) dataset, and the results indicate the effectiveness of our model when compared with the other three frequently-used methods.

Keywords: Trajectory patterns · Next location prediction · Self-attention · Neural network

1 Introduction

In recent years, user can share their activities whenever and wherever, and it owes to a diversity of location-based services in social media applications. Meanwhile, vast amounts of user-generated data can be accumulated, which include spatial and temporal information. For a given user, the check-in data can reflect the underlying patterns which govern his behavior. Finding these patterns can help us to build a model to predict the next location he may visit. Previous studies [1–5] demonstrated that human's behaviors have the features of random and variation. Researchers also exhibit that human mobility follows reproducible [6] under the influence of social and geographic information. Understanding human trajectory patterns and making next location prediction have been widely applied to various tasks, including urban computing [5], the spread of disease [7], traffic congestion prediction [8], personalized recommendation [9], abstractive summarization [10] and semantic role labeling [11].

Nowadays, the next location prediction problem has attracted significant attention and has been extensively studied. Many methods have been proposed in the last ten years. The most commonly used approaches include enhanced Markov model [1], MF

X. Wang et al. (Eds.): CollaborateCom 2019, LNICST 292, pp. 309–322, 2019.
https://doi.org/10.1007/978-3-030-30146-0_21

model [13], Recurrent Neural Network [14], and Bayesian model [15]. Spurred on these approaches, some researchers [1, 16, 17] considered temporal and spatial regularity when making a prediction. However, it still fails to address the two significant challenges well in predicting. Firstly, users are always willing to check in the locations which they are most interested in [12]. However, it can be difficult to identify consistent human mobility patterns according to the sporadic check-ins. In addition, human behavior is affected by several contextual factors like time, weather, emotion, and other factors.

In this paper, we propose a next location prediction approach based on a recurrent neural network and self-attention mechanism. An embedding architecture is used to convert sparse data (e.g., timestamp, location ID, user ID) into dense latent representations. Then, these latent representations are fed into a recurrent neural network to model complicated and long-time dependencies in trajectory sequences. In this way, discrete user check-in data are combined into a continuous sequence, and the relationship among check-ins which are separated by long periods can be extracted.

The output of the recurrent neural network will process into "Self-Attention" mechanism, with the aim of "understanding" the inner relationship of the original sequence. "Self-Attention" mechanism was proposed by Vaswani [18] at 2017 and was initially used to solve Machine Translation problem. Meanwhile, the importance of different contextual factors which influence the transition laws of human behavior can be captured. Then, combined with historical trajectories information [19], the next location is predicted. We conduct our experiments on a real-world dataset, and the results indicate the effectiveness of our model when compared with the other three frequently-used methods.

The rest of the paper is organized as follows. Section 2 summarizes the related work which is highly relevant to our research. Section 3 describes the preliminaries, which includes the definition of the next location prediction problem. Section 4 presents experiments and the results. Section 5 concludes this paper and outlines prospects for future work.

2 Related Work

Location-based services collect massive user-generated data, which contains detailed geo-location information. Researchers attempted to find out whether some basic laws governing human mobility or not. Gonzalez [6] suggested human mobility followed significant regularity like reproducible pattern and believed trajectory could roughly reflect human behavior during a fixed time interval. Zheng [16] introduced trajectories can be transformed into graphs, tensors, matrices, or other data formats. After processing, more data mining and deep learning methods can be applied to extract underlying patterns, which will help researchers analyze human behavior more accurately. In recent years, researchers do not only focus on mining human trajectory but also attempt to predict the next location according to the mined patterns.

Meanwhile, researchers realized the mined patterns could help heighten the understanding of human mobility, and it has a great significance in urban computing, the spread of disease, and personalized recommendation. Chen [20] discovered human movement patterns have a periodic feature, and considered user and collective mobility periodic patterns simultaneously. Liu [21] extracted stay points according to the mined trajectories of users, leveraging the Hidden Markov model predict the next location, and make recommendations. Yao [4] profiled temporal patterns of point-of-interest (POI) and modeled temporal matching user-POI pair to improve the prediction accuracy of POI recommendation.

Nowadays, neural network technology has matured development, and traditional methods are gradually being replaced. Researchers utilized neural networks for predicting which achieved better performance compared with the traditional methods. Kim [22] employed Deep Belief Network (DBN) and Deep Neural Network (DNN) to mine the relationship between human personality and mobility information, then they took this relationship into account when analyzing human mobility and predicting human mobility patterns. Song et al. [23] employed DBN for learning the latent feature representation of heterogeneous data and introduced a DeepMob model to predict the next locations more accurately. Yang [24] demonstrated trajectory records were meaningful for understanding human mobility and presented a neural network model which combined mobility trajectories and social networks. To characterize short-term sequential contexts and long-term sequential contexts, they employed the recurrent neural network (RNN) and Gated Recurrent Unit (GRU) to capture the relationship among sequences from short-term and long-term aspects.

In our paper, we follow these simple patterns that researchers have mined, predicting next locations with RNN and self-attention. Especially, GRU is applied as a methodology to capture complex information of sequences.

3 Preliminaries

In this section, we shed light on problem formulation and introduce GRU. Then Self-attention mechanism is described. Next, the deep connection and positional encoding method are described in detail. Finally, the framework of our model is illuminated.

3.1 Problem Formulation

This section will present the concepts which are referred to in this paper and introduce the research objective.

Definition 1: (check-in). It refers to an event that user records a particular location via location-based services. Each check-in record is unique and includes user ID, timestamp, location information geocoded by <longitude, latitude>. For example, the *m-th* check-in record c at time t in location l of a given user can be described as $c_m = (t, l)$

Definition 2: (check-in sequence). A user u generates lots of check-ins, and these records can be described as a time-ordered sequence: $S_u = c_1 c_2 c_3 \ldots c_n$

Definition 3: (trajectory). Given a time window w, check-in sequence S_u can be divided into subsequences: $S_u = s_{w1} s_{w2} \ldots s_{wk}$, each subsequence s_{wi} ($i \in \{1,2,3\ldots k\}$): $s_{w1} = c_1 c_2 c_3 \ldots c_j$, $s_{w2} = c_{j+1} c_{j+2} \ldots c_{j+m}$ $(1 < j < j + m < n)$ is defined as a trajectory including all the check-ins during the fixed time window w. The window w is the time interval between two subsequences, and its value can be set as an hour, one day, one week or any other threshold which depends on demands.

Goal: (next location prediction). Given check-in sequence $Su = c_1c_2c_3...c_n$ of a given user, the goal is to discover a location where he may visit. That is, given S_u, to obtain a ranked list includes c_{n+1}, c_{n+2} or more locations that user u would like to visit next.

3.2 Gated Recurrent Unit: Extracting the Relationship Among Each Location of Check-in Sequence

In order to mine users' behavior patterns and conduct accurate predicting, a large number of check-in data are necessary. User would like to check in the location where they may be interested in. The sporadic check-ins result in data sparsity. It is difficult to identify the relationship among check-in locations with more extended time intervals. Each location a user will go to is relevant to other locations he visited. RNN is an effective way to extracting the relationship among each location.

GRU and Long Short-Term Memory (LSTM) are the most popular variants of RNN. Different from LSTM, GRU [25] combines *forget gate* and *input gate*, and forms a *reset gate*. Meanwhile, the network no longer gives an extra memory state c_t but regards the output result h_t as the memory state in a continuous backward loop. In this way, GRU can extract the relationship among check-ins that are separated by long periods. The relationship can reflect a user's preference to some extent. For example, a user would like to go to a bookstore after lunch in a restaurant or to a cafe after shopping in a mall. The calculation of GRU is shown as follows:

$$z_t = \sigma(W_t \cdot [h_{t-1}, x_t]) \tag{1}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{2}$$

$$c_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{3}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * c_\mathrm{t} \tag{4}$$

where x_t is the input at timestamp t, h_{t-1} is the network output at time $t - 1$, W is weight matrix which is learned. z_t and r_t are updated gate and reset gate, respectively. σ is a logistic sigmoid function. c_t is a new hidden state, and h_t means the output of the network.

3.3 Self-attention: "Understanding" the Relationship of Each Location Under the Influence of Different Contextual Factors

Human mobility is affected by weather, emotion, and other contextual factors. User may not record the contextual factors when checks. Though GRU can mine the relationship among check-in locations with more extended time intervals. However, it cannot capture the relationship between user behavior and different contextual factors.

Self-attention, which was initially used to improve the accuracy of machine translation, is a particular type of attention mechanism. In this paper, it is utilized for "understanding" the relationship of each location under the influence of different contextual factors.

Vaswani et al. introduced the multi-head attention mechanism and chose a particular attention calculating method called "Scaled Dot-Product Attention". Firstly, we will detail the Scaled Dot-Product Attention. Its formulation is as Eq. (5)

$$Attention(Q, K, V) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

where d_k is the dimension of hidden units of our neural network. Q, K, V are queries, keys, and values, respectively. They are the outputs of the previous layer in the neural network.

Multi-head attention allows the model to focus on information from different representations of subspaces at different positions. The output of GRU can be divided into eight parts and each part calculates the score of each location under the influence of eight different factors. The formulation of multi-head attention is shown as Eq. (6):

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{6}$$

where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d\text{model}\times d_k}$, $W_i^K \in \mathbb{R}^{d\text{model}\times d_k}$, $W_i^V \in \mathbb{R}^{d\text{model}\times d_k}$ which are denoted as the learned linear maps for the *i-th* ($i \in \{1, 2, 3\ldots n\}$) head.

$$\begin{aligned} &score_{att}(Q, K, V) = \\ &Concat(head_1, head_2, \ldots, head_n)W^o \end{aligned} \tag{7}$$

All the vectors that are produced by parallel heads are concatenated together to form a single vector. Then, different channels from different heads are sent to a linear transformation. The output of multi-head attention has the same shape as the input. Finally, dropout, residual connections, and layer normalization strategies are employed on our network to achieve better performance.

3.4 Positional Encoding

Machine Translation task only requires to explore the relationship among every word. However, check-in sequence demonstrates temporal features. The self-attention mechanism cannot distinguish the connection among locations involved time in different trajectories. It is crucial to consider the time when encoding positions of each location in a given check-in sequence. Vaswani et al. proposed a position encoding method to encode positions of each input word for Machine Translation task. In the process of predicting the next location, the accuracy not only associates with locations' position in a fixed sequence but also correlates with the temporal feature. Though positional encoding works well on Machine Translation task, it is to a little avail on next locations prediction because of check-in sequence involving temporal feature.

Feng et al. proposed a method considering historical trajectories to predict human mobility with an attentional recurrent network, and the core idea of this method is suitable for position encoding at the time level. In this paper, we utilize this idea to explore the relationship among locations involved temporal feature, and $Score_{att}$ can be calculated in this way.

3.5 Framework

In this apart, we will introduce the framework of our model in detail. Figure 1 shows the main architecture of our proposed model.

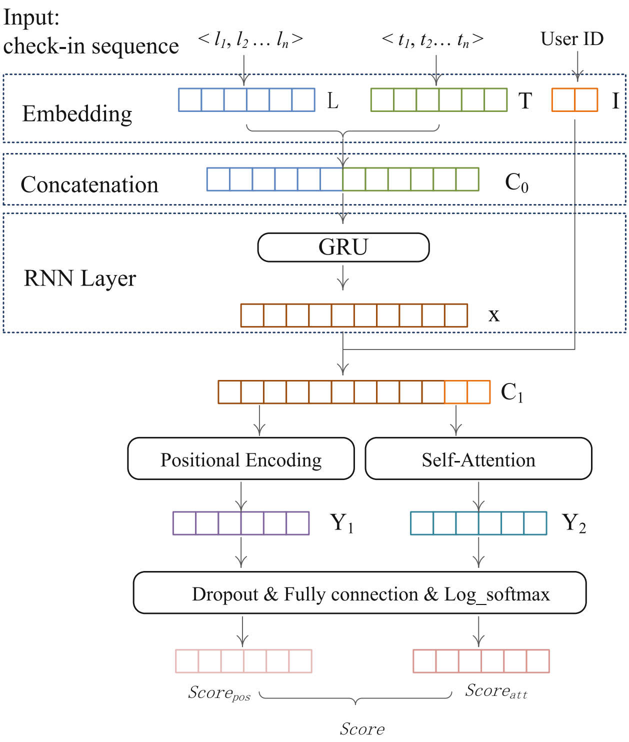


Fig. 1. The main architecture of our proposed mode

Data Processing and Get the Input Data. Check-in sequence of a fixed user has already divided into some trajectories. Each trajectory includes user ID, geographic location information <longitude, latitude> and timestamp. In this paper, geographic location information will be numbered as location ID, using l_i ($i \in \{1, 2, 3\ldots n\}$) to represent each location. In check-in sequence of a given user, location ID, the timestamp, and user ID have been encoded latent vectors representation, then are processed by Embedding layer.

Embed Each Feature Into Latent Representation. Check-in record has three kinds of context information, the timestamp, location ID, and user ID. The timestamp reflected when the user checked. Location ID manifested where the user visited. User ID was the unique identification of a user. Timestamps (t_1, $t_2...t_n$), locations (l_1, $l_2...l_n$) and user ID can be embedded into real-valued vectors *L, T, I* respectively. For a given user, these vectors include information about his behaviors over a while.

Extract the Relationship Among each Location of Check-in Sequence. As Fig. 1 shows, we concatenate the embedding vectors *L* and *T* into C_0. Then C_0 includes the information what the location is and the time when the user visits it. GRU processes the real-valued vectors C_0, aiming to capture the sequential and structure information of a given user's check-ins sequence. The pseudo-code of how GRU works in our method is described in Table 1.

Table 1. The pseudo-code of how GRU works in our method

Input: The real-valued vectors C_0 which is a concatenation of *L* and *T*
Output: The vectors which reflect the relationship among each location of check-in sequence
$L \leftarrow$ embedding vectors of locations
$T \leftarrow$ embedding vectors of timestamps
$C_0 \leftarrow$ embedding vectors which are a concatenation of *L* and *T*
$W_t \leftarrow$ the weight matrices from input to update gate *z*, $\forall\, W_t \in (0,1)$
$W_r \leftarrow$ the weight matrices from input to reset gate *r*, $\forall\, W_r \in (0,1)$
$W \leftarrow$ the weight matrices from input to new hidden state c_t, $\forall\, W \in (0,1)$
for all C_0:
$z_t = sigmoid\,(\,W_t \cdot [\,h_{t-1}, C_0\,]\,)$
$r_t = sigmoid\,(\,W_r \cdot [\,h_{t-1}, C_0\,]\,)$
$c_t = tanh\,(\,W \cdot [r_t * h_{t-1}, C_0\,]\,)$
$h_t = (1 - z_t) * h_{t-1} + z_t * c_t$
end until all C_0 are processed

"Understanding" the Relationship of Each Location Under the Influence of Different Contextual Factors. As Fig. 2 shows, *X* includes the relationship among each location of check-in sequence, and *I* reflects the unique identification of a user.

We concatenate *I* and *X* into C_1 to add the user feature. The vectors C_1 are regarded as the input of self-attention, and C_1 is a concatenation of the output vectors *X* of GRU and embedded vectors *I*. C_1 not only can reflect the relationship among each location of check-in sequence but also include user information in terms of the user ID that embedded vectors. Multi-head mechanism split the input vectors into eight parallel parts in this paper, and each part called $head_i$ ($\{i \mid 1 \le i \le 8, \mathrm{i} \in \mathrm{N}^*\}$).

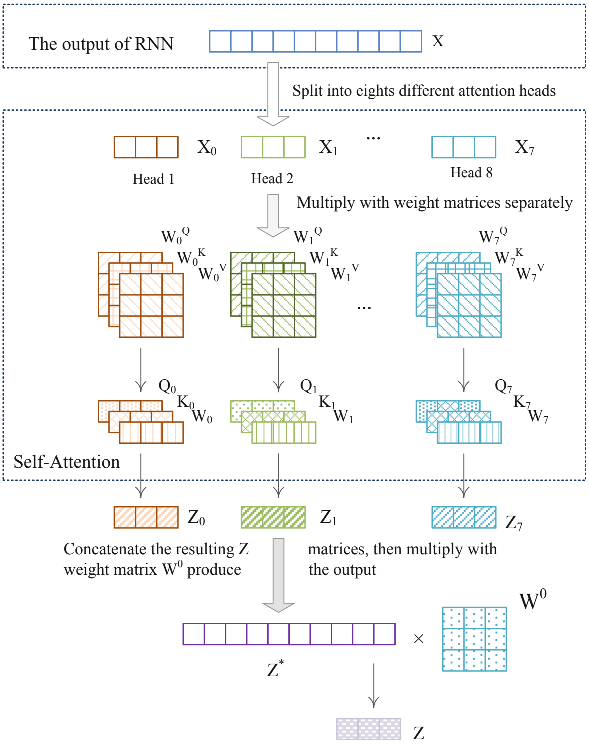


Fig. 2. How Self-attention mechanism works in our proposed model

Each part will multiply with different weight matrices respectively. Users who went to one location could be affected by other locations where he has already gone. Additionally, their behaviors are also influenced by weather, emotion, and other factors. These weight matrices can help the network learn the importance of each location for a given user from different representation subspaces. The path length of the self-attention mechanism is 1. The path length is short enough, which is much easier to learn long-range dependencies and achieve much powerful performance. The output of this step indicates the latent representation of one given location to other locations in the sequential level. The pseudo-code of Self-attention is described in Table 2.

Table 2. The pseudo-code of how self-attention works in our method

Input: The output vectors X and the embedding vectors I
Output: The vectors $Score_{att}$ which the relationship of each location under the influence of different contextual factors
$X \leftarrow$ the output vectors of GRU
$I \leftarrow$ embedding vectors of user ID
$C_1 \leftarrow$ embedding vectors which are a concatenation of X and I
split C_1 into 8 equal parts ($head_1, head_2 \ldots head_8$)
queries/Q, keys/K, values/V $\leftarrow C_1$
$W_i^Q, W_i^K, W_i^V, W^o \leftarrow$ the weight matrices of each $head_i$, $\forall\, W_i^Q, W_i^K, W_i^V, W^o \in (0,1)$, $\{i \mid 1 \le i \le 8, i \in N^*\}$
for all $head_i$:
$head_i = attention(QW_i^Q, KW_i^K, V W_i^V)$
end until all $head_i$ are processed
$head_1, head_2 \ldots head_8$ are concatenated and multiply with W^o
output $Score_{att}$

Positional Encoding. Self-attention was initially used to solve Machine Translation task, and it can capture long-range dependencies among each word. Machine Translation task is essentially a sequence problem. Until now, a given user's check-in data have processed into a sequence. Therefore, it can be solved as a sequence like Machine Translation task. Machine Translation does not have a temporal feature compared with the next location prediction. It is necessary to add positional encoding with temporal feature into prediction problem.

Positional Encoding mechanism will extract human mobility regularities in time. The output of positional encoding indicates the latent representation of one given location to other locations. The DeepMove model proposed by Feng et al. can capture historical trajectories information. We take the main idea of this method to replace the Positional Encoding mechanism proposed by Vaswani. The original Positional Encoding mechanism aims to solve machine translation task. However, it only considers the position of each word. Next locations prediction should take temporal factor into account. Moreover, historical trajectories with time factor can help us to measure the importance of each location.

Dropout and Full Connection and Log-softmax. By taking the concatenation of Self-attention and user ID embedding as the input, dropout strategy can prevent overfitting, and full connection can synthesize the extracted features. Then, the output is processed by log-softmax function to get $score_{att}$. It indicates the transition probabilities from one given location to another in a sequential level. Similarly, the output of Positional Encoding is also projected in the same way to get $Score_{pos}$.

We set *Score* to represent the final result which is composed of two parts: $Score_{att}$ calculated by Self-attention mechanism and $Score_{pos}$ calculated by Positional mechanism. Its calculation formula is as Eq. (8):

$$Score = \alpha \cdot Score_{att} + \beta \cdot Score_{pos} \tag{8}$$

4 Experiments and Results Analysis

4.1 Experimental Objective

1. Set the different value of α and β, to explore the relationship between two kinds of methods, finding the (α, β) pair that maximizes the Score value, and optimizing the performance of the model.
2. Compared with our proposed method and three frequently-used methods, it aims to measure the predicting performance of our method.

4.2 Dataset

The dataset we choose is Foursquare, which has amassed check-ins in New York City, and these data were collected for about ten months from 12 April 2012 to 16 February 2013. It contains 227,428 check-ins generated by 1083 users. Each check-in is associated with its timestamp, and its GPS coordinates consist of <longitude, latitude> and the unique user ID which represents the corresponding user. We remove users with fewer than 10 check-in records. The time difference between two neighbor trajectories is set as 72 h based on the practice. We also drop the trajectories with fewer than 5 records and users who have trajectories less than 5.

4.3 Experimental Setup

Firstly, we take 70% of each user's trajectory data as a training set and the rest of the data as a testing set.

To evaluate our proposed model on the accuracy of the next locations prediction, we compared our model with some popular methods:

1. **Markov.** It is a widely used method for the next locations prediction. Specifically, the Markov model regards a fixed location as a state, then calculates state transition matrix which corresponds to moving from one given location to other locations.
2. **RNN.** In recent years, RNN attracts increasing attention and is used to predict the next locations. The variants of RNN can capture relationship among sequences at short-term or long-term aspects.

3. **RNN with Attention.** Attention mechanism derives from Computer Vision and Pattern Recognition. Feng et al. proposed an attention-based RNN model called DeepMove to predict human mobility with historical information.

4.4 Experimental Results and Analysis

Here we introduce prediction accuracy to evaluate the performance of our proposed model, which can be described as Eq. (9):

$$prediction\ accuracy = \frac{number\ of\ correct\ prediction\ locations}{total\ number\ of\ prediction\ locations} \tag{9}$$

where *total number of prediction locations* means how many locations or top-k locations that we predict next time interval, e.g., 1/top-1 location, 5/top-5 locations, or more locations/top-n depend on practical demand.

In this paper, the final score reflects the probabilities from one given location to other locations is described as Eq. (8). $Score_{att}$ indicates the latent representation of one given location to other locations in trajectory level; $Score_{pos}$ indicates the same latent representation in time level. Here we set the different value of α and β $(\beta = 1 - \alpha)$ to explore the relationship between two kinds of methods. It also means the importance of discovering the sequence regularity and temporal regularity.

We can conclude Fig. 3 when α equals 0.4 and β equals 0.6, then the model achieves the best prediction performance. According to the experimental results, we consider the importance of sequence regularity takes account forty percent approximately. In previous studies, many researchers indicated human mobility patterns not only associated with temporal information but also related to sequence regularity. In this paper, we explore sequence regularity using Self-attention mechanism and extract temporal feature based on the method proposed by Feng. The experimental results precisely accord with that viewpoint.

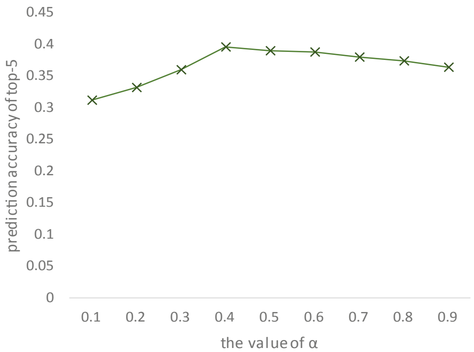
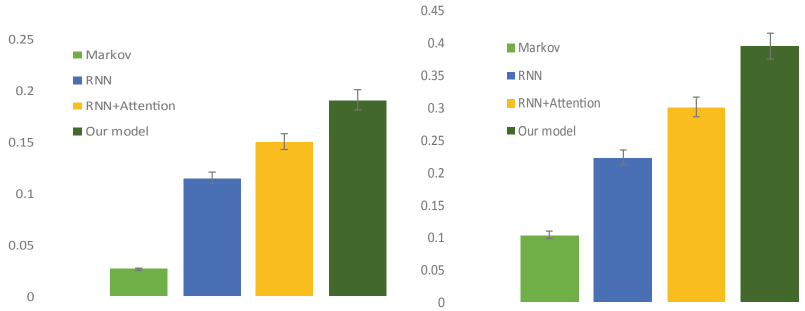


Fig. 3. Prediction accuracy of different α and β value

Figure 4 shows the prediction accuracy of top-k, and k equals to 1 and 5, respectively.



(a) prediction accuracy of top-1 (b) prediction accuracy of top-5

Fig. 4. Prediction accuracy

Figure 4 also shows the experimental results on prediction accuracy and illustrates the proposed model achieves better performance compared with other popular methods. Base on the knowledge of previous researches, though we explore traditional Markov chain can build the transition matrix by taking the sequence of locations a user last visited, it ignores the temporal features. RNN can capture temporal feature compared with the Markov model, and its recurrent units present a solution for long-range dependencies. However, it may pay little attention to historical information. Attention-based RNN model extends RNN with historical information, which illuminates human periodical pattern from long length historical trajectories, and an attention mechanism can capture the information a user pays more attention to some locations. The model we proposed utilizes historical trajectories information to capture the relationship between temporal feature and locations a user has already visited. The Self-attention mechanism helps us to explore the inner relationship of trajectories from different contextual factors. In conclusion, our model achieves better performance on prediction accuracy when compared with the other three frequently-used methods.

5 Conclusion and Future Work

We propose a next location prediction approach based on a recurrent neural network and self-attention mechanism. Exploring the underlying laws behind historical trajectories information of a given user, it can explore sequence regularity and extract temporal feature that governs human mobility. We conduct our experiments on a real-world dataset, Foursquare NY. The experimental results indicate our model outperforms other frequently-used methods significantly.

Future work contains two parts. Firstly, local attention and global attention can be considered, because the mobility pattern of a user will change with time goes by. Secondly, semantic information like reviews can reflect a user's preference. Therefore, we consider predicting the following locations with semantic information, then recommend locations where a user has not visited before while he may be interested in.

Acknowledgment. This research is supported by the National Natural Science Foundation of China (Grant No. 61502062, Grant No. 61672117 and Grant No. 61602070), the China Postdoctoral Science Foundation under Grant 2014M560704, the Scientific Research Foundation for the Returned Overseas Chinese Scholars (State Education Ministry), and the Fundamental Research Funds for the Central Universities Project No. 2015CDJXY.

References

1. Qiao, Y., Si, Z., Zhang, Y., et al.: A hybrid Markov-based model for human mobility prediction. Neurocomputing **278**, 99–109 (2017)
2. Chen, X., Kordy, P., Lu, R., Pang, J.: MinUS: mining user similarity with trajectory patterns. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI), vol. 8726, pp. 436–439. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44845-8_29
3. Chen, C., Kuo, C., Peng, W., et al.: Mining spatial-temporal semantic trajectory patterns from raw trajectories. In: International Conference on Data Mining, pp. 1019–1024 (2015)
4. Yao, Z.: Exploiting human mobility patterns for point-of-interest recommendation. In: Web Search and Data Mining, pp. 757–758 (2018)
5. Altomare, A., Cesario, E., Comito, C., et al.: Trajectory pattern mining for urban computing in the cloud. IEEE Trans. Parallel Distrib. Syst. **28**, 586–599 (2017)
6. Gonzalez, M.C., Hidalgo, A.R., Barabasi, A., et al.: Understanding individual human mobility patterns. Nature **453**, 779–782 (2008)
7. Ajelli, M., Litvinova, M.: Estimating contact patterns relevant to the spread of infectious diseases in Russia. J. Theor. Biol. **419**, 1–7 (2017)
8. Gao, J., Sun, Y., Liu, W., et al.: Predicting traffic congestions with global signatures discovered by frequent pattern mining. In: Green Computing and Communications, pp. 554–560 (2016)
9. Gao, H., Tang, J., Liu, H., et al.: Personalized location recommendation on location-based social networks. In: Conference on Recommender Systems, pp. 399–400 (2014)
10. Paulus, R., Xiong, C., Socher, R., et al.: A Deep reinforced model for abstractive summarization. In: International Conference on Learning Representations (2018)
11. Tan, Z., Wang, M., Xie, J., et al.: Deep Semantic role labeling with self-attention. In: National Conference on Artificial Intelligence, pp. 4929–4936 (2018)
12. Joseph, K., Tan, C.H., Carley, K.M.: Beyond local, categories and friends: clustering foursquare users with latent topics. In: ACM Conference on Ubiquitous Computing, pp. 919–926. ACM (2012)
13. Wang, Y., Shang, W., Li, Z.: The application of factorization machines in user behavior prediction. In: International Conference on Computer and Information Science, pp. 1–4. IEEE (2016)
14. Liu, Q., Wu, S., Wang, L., et al.: Predicting the next location: a recurrent model with spatial and temporal contexts. In: Thirtieth AAAI Conference on Artificial Intelligence. AAAI Press (2016)
15. Jia, Y., Wang, Y., Jin, X., et al.: Location prediction: a temporal-spatial Bayesian model. ACM Trans. Intell. Syst. Technol. 1–25 (2015)
16. Zheng, Y.: Trajectory data mining: an overview. ACM (2015)

17. Zhenjiang, D., Jia, D., Xiaohui, J., et al. RTMatch: real-time location prediction based on trajectory pattern matching. 84–95 (2017)
18. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Neural Information Processing Systems, pp. 5998–6008 (2017)
19. Feng J.: DeepMove: predicting human mobility with attentional recurrent networks. In: The Web Conference, pp. 1459–1468 (2018)
20. Chen, M., Yu, X., Liu, Y.: Mining moving patterns for predicting next location. Inform. Syst. **54**, 156–168 (2015)
21. Liu, H., Wu, G., Wang, G.: Tell me where to go and what to do next, but do not bother me, pp. 375–376 (2014)
22. Kim, D.Y., Song, H.Y.: Method of predicting human mobility patterns using deep learning. Neurocomputing **280**, 56–64 (2017)
23. Song, X., Shibasaki, R., Yuan, N.J., Xie, X., Li, T., Adachi, R.: DeepMob: learning deep knowledge of human emergency behavior and mobility from big and heterogeneous data. ACM Trans. Inform. Syst. **35**, 1–19 (2017)
24. Yang, C., Sun, M., Zhao, W.X., Liu, Z., Chang, E.Y.: A neural network approach to joint modeling social networks and mobile trajectories. ACM Trans. Inform. Syst. (2016)
25. Cho, K,. Van Merrienboer, B., Gulcehre, C., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. Comput. Sci. 1724–1734 (2014)