Research Article

POI Recommendation Method Using Deep Learning in Location-Based Social Networks

Yang Liu and An-bo Wu

School of Management, Xi'an University of Science and Technology, Xi'an, Shaanxi 710054, China

Correspondence should be addressed to Yang Liu; liuyang_xust@xust.edu.cn

Received 11 April 2021; Revised 4 June 2021; Accepted 25 June 2021; Published 9 July 2021

Academic Editor: Yi Sun

Copyright © 2021 Yang Liu and An-bo Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To solve the problems of large data sparsity and lack of negative samples in most point of interest (POI) recommendation methods, a POI recommendation method based on deep learning in location-based social networks is proposed. Firstly, a bidirectional long-short-term memory (Bi-LSTM) attention mechanism is designed to give different weights to different parts of the current sequence according to users' long-term and short-term preferences. Then, the POI recommendation model is constructed, the sequence state data of the encoder is input into Bi-LSTM-Attention to get the attention representation of the current POI check-in sequence, and the Top- N recommendation list is generated after the decoder processing. Finally, a negative sampling method is proposed to obtain an effective negative sample set, which is used to improve the calculation of the Bayesian personalized ranking loss function. The proposed method is demonstrated experimentally on Foursquare and Gowalla datasets. The experimental results show that the proposed method has better accuracy, recall, and F1 value than other comparison methods.

1. Introduction

With the maturity of internet technology and the widespread application of global satellite positioning systems, location-based social network (LBSN) has gradually received more and more attention and research. Recommending future potential points of interest (POI) to users by analyzing their existing historical check-in points has become a research hotspot in this field [1]. Specifically, personalized POI recommendation tasks can be divided into two types according to the recommendation basis. One is static POI recommendation. The basis of this recommendation is limited. It only collects the POIs that the user has checked in and does not consider the information before and after each check-in of the user [2, 3]. The second is the context-aware POI recommendation. This recommendation is based on the use of the user's check-in behavior and the scene of the user's context, dynamically learning the user's dynamic preferences in a certain scene, and generating accurate and timely private customized solutions for the user [4, 5]. In contrast, static POI recommendation focuses on learning users' static preferences and cannot consider users' dynamic and real-time needs.

Therefore, POI recommendation based on context awareness is the core of research in recent years [6].

At present, the POI recommendation system is more complicated than the classic news, movie, or online commodity recommendation system [7]. The first is the severe data sparsity problem. Among the massive POIs provided by LBSN, the POIs that can be accessed by users are almost negligible, resulting in extremely sparse datasets [8]. Secondly, there are three types of cold-start problems commonly encountered in recommended system tasks. A location that has never been visited is called a cold-start POI. Users who have never visited any location are called cold-start users. Users who move from one place to another unfamiliar place to live or travel will also encounter cold-start problems [9, 10]. Finally, there is the problem of user dynamic preferences, that is, user preferences will change with the passage of time and changes in the environment. Therefore, using check-in points to analyze user preferences will face great difficulties and challenges [11, 12].

In order to solve the above-mentioned problems of user dynamic preferences and data sparsity, a POI recommendation method based on deep learning in LBSN is proposed.

Compared with the traditional POI recommendation method, its innovations are summarized as follows:

- (1) Due to the temporal dynamics problem that the traditional recommendation method does not solve and the in-depth recommendation does not consider the attention problem. To this end, the proposed method (Bi-LSTM-Attention) embeds a user-location cross-attention mechanism layer in the neural network to capture effective contextual information
- (2) In order to better capture the long-term and short-term preferences of users in the POI sign-in sequence, the proposed method introduces a Bi-LSTM-Attention mechanism. The long-term and short-term preferences of the user are mined from the user's history and the current POI sign-in sequence. It solves the problem of inaccurate recommendation caused by users' dynamic preferences
- (3) Aiming at the problem of the lack of valid negative samples in the dataset, the proposed method combines the popularity sampling weight and the distance sampling weight in a weighted combination. In this way, negative samples are obtained for model learning, which improves the efficiency of POI recommendation

The rest of this paper is arranged as follows. The second section introduces the related research progress in this field. The third section introduces the proposed POI recommendation method based on the BiLSTM-Attention model. In the fourth section, the feasibility and optimality of the proposed method are simulated by comparing with the current POI recommendation method. The fifth section is the conclusion of this paper.

2. Related Research

In the POI recommendation system, the user's check-in record has only positive samples, there is lack of negative sample information, and the amount of data is sparse. Therefore, recommending a location is challenging [13]. The early POI recommendation was to recommend the most popular Top-K points of interest to users. Although it provides certain guidance for the user's choice, it lacks personalization and the recommendation accuracy rate is low. The current personalized POI recommendation system is constructed based on the early POI recommendation algorithm by fusing various characteristic information. The characteristic information includes user sign-in behavior, social relationship between users, location evaluation information, location description information, location latitude and longitude information, and time context information, thereby improving the accuracy of recommendation [14, 15].

Among the existing research results, most POI location recommendations use traditional collaborative filtering recommendation algorithms. For example, Reference [16] proposed a three-layer network structure POI recommendation

method based on LBSN. More tags and social and geographic information are separately modeled and integrated into a matrix factorization framework. This method has better accuracy than other methods, but the overall recommendation performance is still slightly insufficient. Reference [17] proposed a POI recommendation model based on user contextual behavior semantics. The metapath of the heterogeneous information network is used to represent the complex semantic relationship between the user and the POI, and the recommendation results are ranked by combining with the fusion method of learning ranking. The results show that its recommendation performance is ideal in simple scenarios. The aforementioned POI recommendation algorithm models all belong to the fusion of context information to improve recommendation accuracy. But how to better integrate contextual information into the model has become a difficult problem to be solved [18]. Reference [19] proposed a new POI recommendation framework based on hierarchical category conversion. Different levels of category conversion are used to model the preference conversion modes of users with different granularities. Effectively alleviate the cold-start problem, but the recommended accuracy is poor. Reference [20] recommends POI by integrating user preferences for geography, category, and attribute criteria with personalized weights. And through the collaborative filtering method, the opinions of similar users were merged under three conditions. This method has great advantages in processing large-scale data, but the accuracy and efficiency need to be improved.

Due to the powerful information processing capabilities of deep neural networks, many experts and scholars have applied it to the research of recommendation algorithms in recent years. Reference [21] proposed a real-time POI embedding model, which can mine real-time geographic information and learn potential information according to the corresponding geographic tagging posts. Among them, the convolutional neural network is used to mine the text information of the POI and learn the internal association. Although the universality of the model is better, it is less efficient when dealing with large-scale data. Reference [22] uses a recurrent neural network (RNN) to implement the next POI recommendation. It not only considers the location interests of similar users but also considers contextual information such as time, current location, and friends' preferences. Therefore, the recommendation accuracy and efficiency are relatively ideal, but the learning ability of the RNN single network needs to be improved. Reference [23] combined stacked denoising autoencoders (SDAE) and Bi-LSTM and proposed a new neural network model for context-aware citation recommendation. Use the attention information in the citation context to extend SDAE and enhance its learning ability. However, the huge amount of information data affects the recommendation efficiency. Reference [24] proposed a continuous POI recommendation method based on latent factors with the help of RNN. It integrates POI's sequential access and user preferences to recommend POIs. Although this method has higher accuracy than continuous POI, there is still room for improvement in the extraction of potential information.

FIGURE 1: The structure of the attention layer model.

3. POI Recommendation Based on Bi-LSTM-Attention Model

3.1. Problem Definition. For the POI recommendation problem, define U as a user set, P as a POI set, and T as a label set. And specify $q_u \in R^d$, $p_v \in R^d$, and $s_k \in R^d$ to represent the d -dimensional vector of the user u , the point of interest v , and the label s . Each point of interest v has its longitude and latitude information $\{x_v, y_v\}$ and label collection $\{s_1^v, s_2^v, \dots\}$. Each user u is bound with its label set $\{s_1^u, s_2^u, \dots\}$, friend list $\{f_1^u, f_2^u, \dots\}$, and historical visit record $\{p_{t_1}^u, p_{t_2}^u, \dots\}$, where $p_{t_i}^u$ indicates that the user u visited the point of interest p at time t_i . The historical behavior sequence of all users is denoted as $P^u = \{P^{u_1}, P^{u_2}, \dots\}$, given the user's historical behavior record information and related friend information. The task of the model is to predict the points of interest that the user is most likely to visit next time.

3.2. Attention Layer Model. The attention model was originally used in the field of machine translation and has now become an important concept in the field of neural networks. In the field of artificial intelligence, attention has become an important part of the neural network structure and has a large number of applications in natural language processing, statistical learning, speech, and computer fields [25]. In response to the POI recommendation problem, it is possible that only certain locations in the user's historical time series may be related to the predicted next location. Therefore, the attention mechanism can be used to capture some of the information that assists in the judgment of the next POI and ignore the unimportant information. The principle of using attention is to calculate the matching degree between the current input sequence and the output vector. The high degree of matching means that the relative score of the attention point is also high [26]. The matching degree weight calculated by attention is only limited to the current sequence team, rather than the weight obtained by the traditional neural network model as the overall weight. Attention is mainly used in natural language processing and less involved in POI recommendation problems. Therefore, a user-location cross-attention mechanism layer is embedded in the neural network, and this new attention mechanism is applied to the

next POI recommendation. The attention layer structure is shown in Figure 1.

Among them, l_1, l_2, l_n , and l_u are a set of time and location sequence of user U from p^{u_1}, p^{u_2} to p^{u_n} , respectively, and the output vector of user U after embedding layer and linear hidden layer. α_1, α_2 to α_n is the attention weight of the location output vector, which is the degree of importance for the model prediction, obtained by the transposition of the location output vector and the inner product of the user vector. $+$ represents the addition of the corresponding positions of the location vector. $<>$ represents the splicing of vectors. The calculation expression is as follows:

$$\begin{aligned} \alpha_1 &= l_1 \cdot l_u^T, \\ \alpha_2 &= l_2 \cdot l_u^T, \\ &\dots \\ \alpha_n &= l_n \cdot l_u^T, \end{aligned} \quad (1)$$

$$\alpha_1, \alpha_2, \dots, \alpha_n = \text{softmax}(\alpha_1, \alpha_2, \dots, \alpha_n),$$

$$A = \alpha_1 l_1 + \alpha_2 l_2 + \dots + \alpha_n l_n,$$

$$\text{Relu}(A <> l_u <> l_t).$$

3.3. Model Framework. In order to better capture the long-term and short-term preferences of users in the POI sign-in sequence, Bi-LSTM-Attention is introduced into the decoder. It can dig out the user's long-term preferences from all the historical POI sign-in sequences of the user, while focusing on the short-term preferences in the current POI sign-in sequence, and allows the decoder to assign different weights to different parts of the current sequence according to the user's long- and short-term preferences. The structure of the POI recommendation model that introduces this attention mechanism is shown in Figure 2.

In Figure 2, the encoder represents the part of the encoder structure excluding the dropout layer. Assume that the user has a total of m POI sign-in sequences, and the m th is the current POI sign-in sequence. The overall representation of the user's first $m-1$ historical POI check-in sequence is denoted as $\{h_1^1, h_2^1, \dots, h_{m-1}^1\}$. The hidden state obtained at each moment of the user's current sequence is denoted as $\{h_1^m, h_2^m, \dots, h_t^m\}$. h_ϕ^m is the attention representation of the user's current sequence. $\hat{y}_t$ represents the scoring prediction results for all items.

In Bi-LSTM, the input at the current moment depends not only on the previous video frame but also on the subsequent video frame. The combination of the two units fully considers the timing information before and after the video frame. The model structure is shown in Figure 3.

In Figure 3, $\omega_i (i=1, 2, \dots, 6)$ represents the weight from one unit layer to another. x_t represents the feature vector obtained after extracting deep features through the Inceptionv3 layer. ϕ represents the LSTM unit composed of the feature sequence input from front to back by $\{\dots, x_{t-1}, x_t, x_{t+1}, \dots\}$. ϕ' represents the LSTM unit composed of $\{\dots, x_{t+1}, x_t, x_{t-1}, \dots\}$ feature sequence input from back to front. o_t represents the corresponding output result after the feature

FIGURE 2: The overall architecture of the proposed model.

FIGURE 3: The structure of the Bi-LSTM model.

vector passes through the Bi-LSTM network. The mathematical calculation process of Bi-LSTM is as follows:

$$\begin{aligned}
 \varphi_t &= \text{sigmoid}(\omega_1 x_t + \omega_2 \varphi_{t-1} + b_t^{(1)}), \\
 \varphi'_t &= \text{sigmoid}(\omega_3 x_t + \omega_5 \varphi_{t+1} + b_t^{(2)}), \\
 o'_t &= \tanh(\omega_4 \varphi_t + b_t^{(3)}), \\
 o''_t &= \tanh(\omega_6 \varphi'_t + b_t^{(4)}), \\
 o_t &= \frac{(o'_t + o''_t)}{2}.
 \end{aligned} \quad (2)$$

In the formula, $b_t^{(1)}, b_t^{(2)}, b_t^{(3)}, b_t^{(4)}$ is the bias of the hidden unit of the t th feature vector in the Bi-LSTM network. o'_t and o''_t are the result of two LSTM units, respectively, processing the feature vector output by the Inceptionv3 layer at the corresponding time. The result of adding and averaging the two eigenvectors at the corresponding time is used as the output eigenvector o_t . Finally, the feature vector is sent to the attention mechanism to perceive the network weight [27].

The proposed model first puts the user's historical POI sign-in sequence into the encoder to encode the overall sequence. At the same time, the encoder is used to encode the hidden state at each moment of the user's current sequence. Then, input these data into Bi-LSTM-Attention to get the attention representation of the current POI check-in sequence. h_ϕ^m is calculated as follows:

$$h_\phi^m = \phi(\{h_t^1, h_t^2, \dots, h_t^{m-1}\}, \{h_1^m, h_2^m, \dots, h_t^m\}). \quad (3)$$

After calculating h_ϕ^m , splice it with h_t^m to obtain h_Ω^m . In order to prevent the model from overfitting, it is first put into the dropout layer, and then, the score of each candidate POI is obtained through the fully connected layer and the nonlinear activation function tanh. Finally, the candidate POIs are sorted according to their scores, and a Top-N recommendation list is generated.

$$\begin{aligned}
 \hat{y}_t &= \tanh(\omega_y h_\Omega^m + b_y), \\
 h_\Omega^m &= [h_\phi^m; h_t^m],
 \end{aligned} \quad (4)$$

where ω is the weight coefficient from the splicing layer to the output layer and b is the offset.

3.4. Loss Function and Optimization Algorithm. The basic task of the recommendation system is to sort the items in the system according to a series of related information. Although this task can be regarded as a classification task, it is still a sorting task in essence [28, 29]. The proposed method regards the task of generating the Top-N recommendation list as a sorting task. The specific purpose is to make the POI actually checked in by the user in the next step to rank higher in the generated recommendation list. In order to improve the ranking performance of the model, consider using Bayesian personalized ranking (BPR) loss [30, 31]. The definition of BPR is calculated as follows:

$$L_{\text{BPR}} = -\frac{1}{N_p} \sum_{j=1}^{N_p} \sigma(\hat{y}_{p_i} - \hat{y}_{p_j}), \quad (5)$$

where $\hat{y}_{p_i}$ represents the predicted score of the target POI_{p_i} . $\hat{y}_{p_j}$ represents the predicted score of the target POI_{p_j} . N_p is the number of negative samples.

In order to allow the model to learn more effectively, it is usually desirable to use more effective negative samples for learning, which can make BPR generate higher gradients. If the negative sample has no value, it will produce a gradient

close to zero. Due to the averaging operation in the BPR calculation, when there are many worthless negative samples, the gradient of the loss function will become very small. That is, the phenomenon of gradient disappearance occurs, so that learning can no longer continue [32, 33]. In order to avoid the above-mentioned problems of BPR, the ranking loss function BPRmax is used to measure the prediction error of the model. The calculation of BPRmax is as follows:

$$L_{\text{BPRmax}} = \log \sum_{j=1}^{N_p} c_j \sigma(\hat{y}_{p_i} - \hat{y}_{p_j}) \sum_{j=1}^{N_p} c_j y \wedge^2_{p_j}, \quad (6)$$

where c_j represents the likelihood that POI_{p_j} is similar to the target POI_{p_i} . The softmax function can be used to process the prediction score output by the model, and λ is the regular term coefficient.

For negative samples with low prediction scores, it is considered invalid to repeat learning for them, so they are given a lower calculation weight [34]. At the same time, in order to prevent the predicted score from being too extreme, BPRmax adds a regular term for the predicted score. These measures effectively solve the problem of gradient disappearance caused by the BPR loss function and improve the performance of the model [35].

Since implicit feedback data often only has positive feedback, and negative feedback is usually missing or difficult to judge, it becomes very necessary to generate a corresponding negative sample for each positive sample. If a user has not performed a check-in operation on multiple POIs, it is impossible to explicitly determine which POI is the user who did not perform the check-in operation because he did not like it. Only the POIs that users really do not like can make the model learn effective user preferences. Invalid negative samples will increase the noise in the data and affect the learning effect of the model. At the same time, in the recommendation system, the number of items is usually very large, and it is not realistic to calculate a score for each item. This will cause the calculation of the system to continue to expand as the product of the number of projects and the number of user-project interactions increases, making the system unusable. A large number of calculations will also cause the accuracy of the final result to deteriorate. For the above reasons, effective negative sampling in the process of model learning can not only make the training process of the model more efficient but also reduce the amount of model calculations.

The proposed method also takes into account the popularity of the target POI and the geographical distance between the target POI and the last time the user checked in to the POI. A negative sampling method that is more suitable for the characteristics of the POI recommendation task is proposed. The sampling method firstly calculates the sampling weight β_p of place popularity based on the statistics of the number of check-ins of POI. Then, the distance sampling weight d_p of all POIs is calculated according to the POI and POI distance matrix that the user signs in at time t . The specific calculation is as follows:

$$\begin{aligned} \beta_p &= \frac{\vartheta_p}{\vartheta_{\max}}, \\ d_p &= \frac{\text{mid}_d}{(\text{mid}_d + d_{pt,p})}, \end{aligned} \quad (7)$$

where ϑ_p is the number of check-ins for POI_p . $\vartheta_{\max}$ is the number of check-ins of the most popular POI. $d_{pt,p}$ is the distance between the user's sign-in POI_{pt} at time t and the target POI_p . mid_d is the adjustment parameter.

Then, the popularity sampling weight and the distance sampling weight are weighted and combined to obtain the overall sampling weight $w_{\beta,d}$ of the POI. The calculation is as follows:

$$w_{\beta,d} = \mu \cdot \beta_p + \eta \cdot d_p, \quad (8)$$

where μ and η are the adjustment factors of the two weights, which add up to 1.

Finally, according to the sampling weight, negative sampling is performed on all POIs except the last POI that the user signs in to obtain a set of negative samples used to calculate the loss function.

4. Experiment and Analysis

4.1. Dataset. The experiment uses two public datasets: Foursquare and Gowalla datasets. In order to show the spatial distribution of user check-in data more clearly, the geographic spatial distribution of the Foursquare and Gowalla datasets is shown in Figure 4.

As can be seen from Figure 4, the distribution of the check-in points of interest in the two datasets presents different geographic distribution characteristics. The spatial distribution of user check-ins in the Foursquare dataset is relatively more scattered, while the spatial distribution of user check-ins in the Gowalla dataset is concentrated in several different centers. The differences in user activity patterns in different geographic regions are explained. At the same time, the sign-in sequences of several users were randomly selected in the two datasets and visualized on the map, as shown in Figure 5.

It can be seen from Figure 5 that the movements of different users present different geographic distribution patterns. It proves that there are individual differences in the activity patterns and spatial preferences of different users. For example, in Figure 5(a), the geographical distances of users who check in continuously are large and scattered, while the geographical distances of users who check in continuously in Figure 5(b) are relatively small and concentrated. This also further illustrates the differences in the degree of preference of different users for geographic distance. In addition, from the perspective of the two datasets as a whole, users prefer to visit points of interest closer to them, and the visited points of interest show geographic clustering. Therefore, in the POI recommendation task, consider using continuous geographic distance information to model the user's personalized spatial preferences.

FIGURE 4: Geospatial distribution of two check-in datasets.

FIGURE 5: Distribution of sign-in sequences of different users.

TABLE 1: Basic statistics of check-in dataset.

	Foursquare	Gowalla
Number of users	2475	10839
Number of points of interest	5803	25172
Number of check-in	205317	599675
Average number of points of interest per user	46.75	30.96
Average number of visitors per POI	17.89	13.14
Data sparsity	99.19%	99.89%

For the two check-in datasets, in order to alleviate the sparseness of the sign-in data, sparse users who visit less than 5 points of interest are removed, and at the same time, the unpopular points of interest that visit less than 5 users are removed. After data preprocessing, the statistics of Foursquare and Gowalla datasets are shown in Table 1.

It can be seen from Table 1 that the user sign-in data is very sparse. The average number of points of interest visited by users in the two datasets is less than 1% of the total number of points of interest in each. The average number of visitors to each point of interest is also less than 1% of the number of users in their respective datasets. Therefore, the problem of data sparsity is a big challenge for the task of point of interest recommendation. With limited historical check-in data, fusing spatiotemporal context information can not only improve the accuracy of point-of-interest recommendation but also greatly alleviate the problem of data sparsity.

4.2. Evaluation Index. In order to accurately evaluate the performance of the proposed method on the next POI recommendation task, Precision is used to measure it, and the calculation is as follows:

$$\text{Precision} = \frac{\sum_{u=1}^{\psi} \zeta_u}{\psi}, \quad (9)$$

$$\zeta_u = \begin{cases} 1, & O \in \text{sort}_u^N, \\ 0, & O \notin \text{sort}_u^N. \end{cases}$$

Each user has one positive sample data and multiple negative sample data in the test set. Among them, O represents the positive sample data of user u in the test set. Both positive and negative sample data have a score, and the model predicts the score of the user's next destination. sort_u^N represents the set of N POI locations with the highest scores predicted

FIGURE 6: The impact of different moving sequence lengths on the performance of the Bi-LSTM-Attention.

by the model for the positive and negative samples of user u in the test set. If the location of the positive sample data of the user u is in the sort_u^N set, then ζ_i is recorded as 1, which means that the model predicts the user's POI location correctly; otherwise, it is recorded as 0. ψ represents the total number of users. Generally, the larger the value of N , the higher the accuracy of the model's prediction.

Aiming at the Top- N ranking problem of point-of-interest recommendation, the two most commonly used evaluation indicators are used, namely, recall and F1-score. They are also commonly used algorithm evaluation indicators in the field of machine learning and data mining. The two types of evaluation indicators are abbreviated as Recall@ N and F1@ N , respectively. For a given user set U , any user $u (u \in U)$ and the set of points of interest that the user has visited in the test set are P_u . The set of N points of interest recommended to the user through the algorithm is $R_u(N)$. Then, Recall@ N and F1@ N are calculated as follows:

$$\text{Recall@}N = \frac{|R_u(N) \cap T_u|}{|T_u|},$$

$$\text{F1@}N = \frac{2(|R_u(N) \cap T_u|/N)(|R_u(N) \cap T_u|/|T_u|)}{(|R_u(N) \cap T_u|/N) + (|R_u(N) \cap T_u|/|T_u|)}, \quad (10)$$

where T_u is the set of label points of interest of the user.

4.3. Parameter Discussion. In the two datasets, the optimal embedding dimension and spatial window width are maintained. By setting different moving sequence lengths $\{10, 20, \dots, 100\}$, observe the changes of the Bi-LSTM-Attention under the Recall@5 and Recall@10 evaluation indicators. The effect of different moving sequence lengths on the performance of the Bi-LSTM-Attention is shown in Figure 6.

It can be seen from Figure 6, whether it is Recall@5 or Recall@10, for the Foursquare dataset, when the sequence length is set to 80, the Bi-LSTM-Attention can obtain the best recommendation results. For the Gowalla dataset, when the sequence length is set to 60, the Bi-LSTM-Attention obtains the best recommended performance. The different results of the two datasets may be caused by the difference in the number of user check-ins in the two datasets.

4.4. Performance Comparison with Comparison Method. In order to demonstrate the recommended performance of the Bi-LSTM-Attention, multiple experiments were carried out using the evaluation indicators of Precision@2, Precision@4, Precision@6, and Precision@8. Compare it with the results in References [16, 17, 23, 24]. Pay special attention to the fact that the length of the sequence accessed by each user is different, so the test sequence length is greater than or equal to N parts. Therefore, the larger the N , the smaller the number of test datasets. The experimental results of the two datasets are shown in Figure 7.

It can be seen from Figure 7 that on the Foursquare and Gowalla datasets, the recommended performance of the Bi-LSTM-Attention is better than other comparison methods. When Precision@2, its recommended accuracy rates on the two datasets are 0.045 and 0.038, respectively. Because the attention layer is integrated into the Bi-LSTM model, different parts of the current sequence are given different weights according to the user's long-term and short-term preferences, thereby improving the accuracy of interest point prediction. And the effect of the Bi-LSTM-Attention on Foursquare compared to other methods is more obvious than that on Gowalla. Because the Foursquare dataset has a shorter average sequence length than Gowalla's dataset, the use of additional nonsequence information is more helpful in improving accuracy. This demonstrates that the Bi-LSTM-Attention has a better handling effect in dealing with the cold-start problem.

FIGURE 7: Comparison results of Precision@N of different methods.

FIGURE 8: Comparison results of Recall@N of different methods.

Similarly, the Recall@N comparison results of different methods on the Foursquare and Gowalla datasets are shown in Figure 8.

It can be seen from Figure 8 that as N increases, the recall value continues to increase. And the Recall@N value of the Bi-LSTM-Attention on the two datasets is better than other comparison methods. When N is 20, the Recall@N values of the Bi-LSTM-Attention are 0.33 and 0.36, respectively. Because it uses the Bi-LSTM-Attention mechanism to increase the weight of important location information and combines a variety of context-aware information, it greatly

improves the recommendation performance. Reference [16] integrates more label and social and geographic information into a matrix factorization framework. This method can achieve better recommendation accuracy, but the overall recommendation performance is still slightly insufficient. Reference [17] completes POI recommendation based on user context behavior semantics. But the method is more traditional, so the recommendation effect is not ideal. Reference [23] combines SDAE and Bi-LSTM to implement location recommendation for context awareness. Reference [24] uses an improved RNN for the next POI recommendation, and

FIGURE 9: Comparison results of F1@N of different methods.

both are better than RNN. Taking the Foursquare dataset as an example, the Recall@20 values of the two are 0.32 and 0.34, respectively. However, it lacks consideration of the user's long-term and short-term location preferences, so the overall recommendation performance is still lacking.

In addition, the F1@N comparison results of different methods on the Foursquare and Gowalla datasets are shown in Figure 9.

It can be seen from Figure 9 that the Bi-LSTM-Attention shows the best recommendation effect on the Foursquare and Gowalla datasets. Among them, the Bi-LSTM-Attention is significantly better than the traditional recommendation methods in References [16, 17]. Comparing Reference [17], it has an average increase of 51.58%, 47.24%, 57.12%, and 48.35% under the evaluation indicators of F1@5, F1@10, F1@15, F1@20, etc., of the two datasets, respectively. At the same time, the Bi-LSTM-Attention is also significantly better than the other two methods based on deep neural networks. Compared with Reference [23], the Bi-LSTM-Attention has an average increase of 11.24%, 23.17%, 37.18%, and 33.34% under the evaluation indicators of F1@5, F1@10, F1@15, F1@20, etc., respectively.

In summary, the Bi-LSTM-Attention is effective on the task of point-of-interest recommendation, mainly because it can more effectively integrate spatiotemporal context information, thereby more accurately modeling the user's sequence preferences and personalized spatiotemporal preferences, and to a certain extent alleviate the problem of data sparsity.

5. Conclusion

Points of interest recommendation, as an important branch of the location social network, can help users find the most

interesting personalized locations. However, POI recommendation is affected by issues such as data sparseness and dynamic changes in user preferences, and it is challenging to achieve high-accuracy recommendation. For this reason, a POI recommendation method using deep learning in LBSN is proposed. Enter the user's history and current POI sign-in sequence into the encoder to encode the entire sequence. And input these data into Bi-LSTM-Attention to get the attention representation of the current POI sign-in sequence. Afterwards, it is processed by the decoder to generate a Top-N recommendation list. Finally, the Bi-LSTM-Attention is demonstrated on foursquare and Gowalla datasets. The results show that the accuracy, recall, and F1 value of the Bi-LSTM-Attention are better than other comparison methods, and it has better recommendation performance. In addition, the attention layer is integrated into the Bi-LSTM model, which can give different weights to different parts of the current sequence according to the user's long-term and short-term preferences, further improving the accuracy of the recommendation.

Since the length of the time series of the input user of the Bi-LSTM-Attention is part of the historical POI location of the user, in the next study, the length of the sequence will be increased to improve the universality of the Bi-LSTM-Attention. And the Bi-LSTM-Attention also solves the cold-start problem to a certain extent, but the improvement is limited. In the future, we will need to find new ways to improve model performance.

Data Availability

The data included in this paper are available without any restriction.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Natural Science Foundation of China (No. 71673220) and the Special Scientific Research Plan of Education Department of Shaanxi Provincial Government-Humanities and Social Sciences (No. 20JK0232).

References

- [1] L. Tang, D. Cai, Z. Duan, J. Ma, M. Han, and H. Wang, "Discovering travel community for POI recommendation on location-based social networks," *Complexity*, vol. 2019, no. 2, 8 pages, 2019.
- [2] J. Jiao, X. Zhang, F. Li, and Y. Wang, "A novel learning rate function and its application on the SVD++ recommendation algorithm," *IEEE Access*, vol. 8, pp. 14112–14122, 2020.
- [3] Z. Huang, X. Lin, H. Liu, B. Zhang, Y. Chen, and Y. Tang, "Deep representation learning for location-based recommendation," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 3, pp. 648–658, 2020.
- [4] Y. S. Lu and J. L. Huang, "GLR: a graph-based latent representation model for successive POI recommendation," *Future Generation Computer Systems*, vol. 102, no. 6, pp. 230–244, 2020.
- [5] T. Qian, B. Liu, Q. V. Nguyen, and H. Yin, "Spatiotemporal representation learning for translation-based POI recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 2, pp. 1–24, 2019.
- [6] L. Hu, Y. Xing, Y. Gong, K. Zhao, and F. Wang, "Nonnegative matrix tri-factorization with user similarity for clustering in point-of-interest," *Neurocomputing*, vol. 363, no. 8, pp. 58–65, 2019.
- [7] K. Yang, J. Zhu, and X. Guo, "POI neural-rec model via graph embedding representation," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 76–86, 2021.
- [8] Y. Huo, B. Chen, J. Tang, and Y. Zeng, "Privacy-preserving point-of-interest recommendation based on geographical and social influence," *Information Sciences*, vol. 543, no. 3, pp. 202–218, 2021.
- [9] X. Xiong, S. Qiao, N. Han et al., "Where to go: an effective point-of-interest recommendation framework for heterogeneous social networks," *Neurocomputing*, vol. 373, no. 7, pp. 56–69, 2020.
- [10] P. Kefalas, P. Symeonidis, and Y. Manolopoulos, "A graph-based taxonomy of recommendation algorithms and systems in LBSNs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 604–622, 2016.
- [11] W. Feng, Q. Zhu, J. Zhuang, and S. Yu, "An expert recommendation algorithm based on Pearson correlation coefficient and FP-growth," *Cluster Computing*, vol. 22, no. 3, pp. 7401–7412, 2019.
- [12] T. Y. Qian, B. Liu, L. Hong, and Z. N. You, "Time and location aware points of interest recommendation in location-based social networks," *Journal of Computer Science and Technology*, vol. 33, no. 6, pp. 1219–1230, 2018.
- [13] Y. Si, F. Zhang, and W. Liu, "An adaptive point-of-interest recommendation method for location-based social networks based on user activity and spatial features," *Knowledge-Based Systems*, vol. 163, no. 6, pp. 267–282, 2019.
- [14] D. Li and J. Liu, "Uncovering the relationship between point-of-interests-related human mobility and socioeconomic status," *Telematics and Informatics*, vol. 39, no. 6, pp. 49–63, 2019.
- [15] Y. Liu, "An experimental evaluation of point-of-interest recommendation in location-based social networks," *Proceedings of the Vldb Endowment*, vol. 10, no. 10, pp. 1010–1021, 2017.
- [16] Z. Zhang, Y. Liu, Z. Zhang, and B. Shen, "Fused matrix factorization with multi-tag, social and geographical influences for POI recommendation," *World Wide Web*, vol. 22, no. 3, pp. 1135–1150, 2019.
- [17] D. Yu, K. Xu, D. Wang, T. Yu, and W. Li, "Point-of-interest recommendation based on user contextual behavior semantics," *International Journal of Software Engineering and Knowledge Engineering*, vol. 29, no. 11n12, pp. 1781–1799, 2019.
- [18] Z. Zhang, C. Zou, R. Ding, and Z. Chen, "VCG: exploiting visual contents and geographical influence for point-of-interest recommendation," *Neurocomputing*, vol. 357, no. 2, pp. 53–65, 2019.
- [19] L. Zhang, Z. Sun, J. Zhang, H. Kloeden, and F. Klanner, "Modeling hierarchical category transition for next POI recommendation with uncertain check-ins," *Information Sciences*, vol. 515, no. 3, pp. 169–190, 2020.
- [20] Y. Lyu, C.-Y. Chow, R. Wang, and V. C. S. Lee, "iMCRec: a multi-criteria framework for personalized point-of-interest recommendations," *Information Sciences*, vol. 483, no. 4, pp. 294–312, 2019.
- [21] P. Y. Hao, W. H. Cheang, and J. H. Chiang, "Real-time event embedding for POI recommendation," *Neurocomputing*, vol. 349, no. 7, pp. 1–11, 2019.
- [22] M. Chen, W. Z. Li, L. Qian, S. L. Lu, and D. X. Chen, "Next POI recommendation based on location interest mining with recurrent neural networks," *Journal of Computer Science and Technology*, vol. 35, no. 3, pp. 603–616, 2020.
- [23] T. Dai, L. Zhu, Y. Wang, and K. M. Carley, "Attentive stacked denoising autoencoder with bi-LSTM for personalized context-aware citation recommendation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, no. 3, pp. 553–568, 2020.
- [24] Y. S. Lu, W. Y. Shih, H. Y. Gau, K. C. Chung, and J. L. Huang, "On successive point-of-interest recommendation," *World Wide Web*, vol. 22, no. 3, pp. 1151–1173, 2019.
- [25] D. Cao, Y. Huang, H. Li, X. Zhao, H. Chen, and Y. Fu, "Text sentiment classification based on attention mechanism and decomposition convolutional neural network model," in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pp. 234–238, Dalian, China, 2020.
- [26] W. Chen, J. Xia, X. Wang, Y. Wang, J. Chen, and L. Chang, "Relation lines: visual reasoning of egocentric relations from heterogeneous urban data," *ACM transactions on intelligent systems*, vol. 10, no. 1, pp. 1–21, 2019.
- [27] R. Logesh, V. Subramaniaswamy, V. Vijayakumar, and X. Li, "Efficient user profiling based intelligent travel recommender system for individual and group of users," *Mobile networks & applications*, vol. 24, no. 3, pp. 1018–1033, 2019.

- [28] A. Likhvani, S. K. Bedathur, and P. Deepak, "Location-specific influence quantification in location-based social networks," *ACM transactions on intelligent systems*, vol. 10, no. 3, pp. 23.1–23. 28, 2019.
- [29] F. Wang, X. Meng, and Y. Zhang, "Context-aware user preferences prediction on location-based social networks," *Journal of Intelligent Information Systems*, vol. 53, no. 1, pp. 51–67, 2019.
- [30] Y. Wang, H. Sun, Y. Zhao, W. Zhou, and S. Zhu, "A heterogeneous graph embedding framework for location-based social network analysis in smart cities," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2747–2755, 2020.
- [31] L. W. Dietz, A. Sen, R. Roy, and W. Wörndl, "Mining trips from location-based social networks for clustering travelers and destinations," *Information Technology & Tourism*, vol. 22, no. 1, pp. 131–166, 2020.
- [32] Q. Cui, S. Wu, Y. Huang, and L. Wang, "A hierarchical contextual attention-based network for sequential recommendation," *Neurocomputing*, vol. 358, no. 9, pp. 141–149, 2019.
- [33] G. Zheng, H. Yu, and W. Xu, "Collaborative filtering recommendation algorithm with item label features," *International Core Journal of Engineering*, vol. 6, no. 1, pp. 160–170, 2020.
- [34] R. L. Rosa, G. M. Schwartz, W. V. Ruggiero, and D. Z. Rodriguez, "A knowledge-based recommendation system that includes sentiment analysis and deep learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2124–2135, 2019.
- [35] K. Song, M. Ji, S. Park, and I. C. Moon, "Hierarchical context enabled recurrent neural network for recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 3, pp. 4983–4991, 2019.