Applied Soft Computing Journal 96 (2020) 106536

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Efficient point-of-interest recommendation with hierarchical attention mechanism

Guangyao Pang [a,b,c], Xiaoming Wang [a,b,*], Fei Hao [a,b], Liang Wang [a,b], Xinyan Wang [a,b]

[a] Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an, 710062, China
[b] School of Computer Science, Shaanxi Normal University, Xi'an, 710119, China
[c] School of Data Science and Software Engineering, Wuzhou University, Wuzhou, 543002, China

ARTICLE INFO

Article history:
Received 2 November 2019
Received in revised form 2 July 2020
Accepted 5 July 2020
Available online 1 August 2020

Keywords:
POI recommendation
Hierarchical attention mechanism
Deep-learning
Semantic similarity
Sequential prediction

ABSTRACT

Personalized Point-of-Interest (POI) Recommendation is very important for application platforms based on Location Based Social Networks (LBSNs). It can assist users in making decisions to alleviate the problem of information overload, and can also improve the user experience of these platforms and advance platform operators achieve personalized and accurate advertising. However, there exist some problems of data sparseness and cold start for a single user, and it is also difficult to mine valuable long-tailed POIs, although the size of the check-in data is large. Therefore, in order to address the above problems, we propose a personalized POI Recommendation approach based on Hierarchical Attention Mechanism (HAM-POIRec) which can effectively increase data utilization. Firstly, we define the concepts of explicit features and implicit features, which pave the ideas of selecting data and computational models for POI recommendation based on machine learning. Secondly, we propose a hierarchical attention mechanism with the structure of local-to-global, which extracts contributions and mines more hidden information from individual features, combination features, and overall features. Finally, we present the Natural Language Processing (NLP)-based "User-POI" matching mechanism for the first time in the field of POI recommendation to improve the recommendation accuracy by fine-tuning the POIs predicted by the recommendation system. Extensive experiments are conducted for demonstrating that the HAM-POIRec method outperforms state-of-the-art DeepPIM method and the other comparison methods (SAE-NAD, MGMPFM and LRT), especially in predicting sequence POIs and solving cold start problem.

1. Introduction

With the rapid development of mobile devices and location acquisition technologies, it has become more and more convenient for people to obtain real-time location-based services [1–3]. In addition, the related platforms of Location Based Social Networks (LBSNs) including *Yelp*,[2] *Foursquare*,[3] *Dianping*[4] and *Mafengwo*[5] have also developed rapidly. On these platforms, the users share their locations and experiences in places they visit (such as tourist attractions, restaurants and shops, etc.), which generates a lot of check-in data [4,5]. These places, the users like and have visited, are named as Point-of-Interest (POI) [6,7]. A large number of interactive data between users and POIs bring potential data wealth to these mobile internet platforms based on LBSNs, but it causes the problem of information overload [8]. POI recommendation helps users to select their favorite POIs from the overloaded information. That is to say, on the one hand, the POI recommendation can understand the user's personalized requirements to reduce the pressure of filtering information, and help the users understand their surrounding environment to assist them in making decisions. On the other hand, it can help platform operators to implement intelligent advertising services, and can improve the user experience of the platform while increasing its advertising revenue [9].

POI recommendation faces more challenges than general recommendation (such as product recommendation and movie recommendation) [10]. Firstly, users' preferences for POIs are affected by geographic distance: Usually they visit a small number of POIs near their home or school. Second, the users may visit the same POIs (eg, workplace, breakfast shop, etc.) every day. Third, users' preferences depend on time series, for example, the restaurants they visited in the early morning and late at night

* Corresponding author at: School of Computer Science, Shaanxi Normal University, Xi'an, 710119, China.
E-mail addresses: pangguangyao@snnu.edu.cn (G. Pang), wangxm@snnu.edu.cn (X. Wang), fhao@snnu.edu.cn (F. Hao), wangliang@snnu.edu.cn (L. Wang), wangxinyan@snnu.edu.cn (X. Wang).

2 https://www.yelp.com.
3 https://foursquare.com.
4 https://www.dianping.com.
5 https://www.mafengwo.cn.

https://doi.org/10.1016/j.asoc.2020.106536
1568-4946/

are different. Fourth, the users' preferences are affected by social relationships, for example, he is affected by the preferences of his friends. Finally, data such as comments and descriptions of POIs can also influence user's preferences. As shown in Table 1, in order to solve these challenges, POI recommendation has gradually formed a social network-based POI recommendation method and a POI recommendations based on check-in data context and topic content.

In the social network-based POI recommendation, collaborative filtering based on social relationships is a widely used method. This method implements recommendations based on the social relationships between two friends and the similarity of check-ins [11]. It does not consider all users, so the calculation speed is faster, but it also results in lower accuracy [12]. Immediately after, there appeared the Probabilistic Matrix Factorization with Social Regularization (PMFSR) for integrating social influences to learn the user's potential preferences or potential characteristics using a matrix factorization method [13,14]. However, none of these methods can handle large-scale data. Fortunately, the methods based on deep neural networks can solve this problem. For example, the self-attention autoencoders put forward a multi-dimensional attention mechanism to improve the accuracy of feature extraction by integrating geographic location, so that the model can obtain more hidden features to alleviate the problem of data sparsity [15]. However, this method still has the problem of being unable to mine high-quality long-tail POIs (that is, the cold start problem). Taking Foursquare and Twitter as an example, only about 20% of friends have checked in at the same POIs, which means that about 80% of friends do not have any POIs in common [16].

The POI recommendations based on check-in data context and topic content is the current mainstream method, which can alleviate the problem of sparse data and help the users to mine high-quality long-tail POIs [12,17]. The core of this type of method is to extract more latent features from unstructured information such as rating, classification labels, and description text to describe users and POIs, and build their associations based on these latent features [18–20]. Previously, referring to the user-based and item-based collaborative filtering on the use of ratings to complete recommendations, researchers have used ranking-based geographic factorization machine to complement their potential preferences based on geographic location and check-in context data to recommend unvisited POIs for users [13], and use a collaborative POI recommendation based on naive bayes to model user check-in behavior using power law distribution to predict user access probability to new POIs [21]. However, the above methods usually model user preferences based on the weighted average of the ratings or the inner product of latent factors. It is the calculation of the inner product that limits the expressive power of the POI recommendation, which makes the method unable to process large-scale data [12,22,23]. As shown in Table 1, researchers have attempted to extract more latent features from structured and unstructured data by using basic Recurrent Neural Network (RNN), and achieved good results, but with problems such as gradient disappearance [24]. Hence, the researchers put forward two solutions: one is to introduce attention mechanism to extract more useful expression features from limited data [25,26], the other is to add unstructured data to mine more valuable potential features [18–20,25] [26]. However, these methods still ignore the latent features of the combination features and overall features (including the different contributions of these features to the recommendation). In addition, no researchers has yet attempted to extract the "User-POI" matching degree from text similarity to improve recommendation performance.

For the sake of solve the above problems, extracting more expression features from the sparse data of individual users to complete accurate personalized POI recommendation, and mine high-quality long-tail POIs for users, we propose a POI Recommendation method based on Hierarchical Attention Mechanism termed HAM-POIRec. First of all, in addition to the review text shown in Table 1, we have also added other unstructured data including POI description text and pictures, to further refine the concept of explicit features extracted from structured features and implicit features from unstructured data. Secondly, in the input data of the neural network, there are differences in the contribution of individual features, combination features and overall features to the recommendation, and it contains a lot of useful hidden information. Therefore, we propose a hierarchical attention mechanism of the local-to-global structure to extract more hidden information for sparse data and improve the utilization of high-quality information. Finally, we extract the potential association of "User-POI" from user comment (text) and POI description (text) through Natural Language Processing (NLP), and use this association as weights to fine-tune predictors built from the users and POIs features to improve recommendation performance. Overall, the major contributions of this paper are as follows:

- In order to analyze more representative features in complex check-in data by POI recommendation methods based on machine learning, we define the concept of explicit features and implicit features. These concepts provide some ideas to collect data and select computational models for machine learning based methods.
- In order to obtain more effective information from sparse data and to make use of high-contribution information as much as possible, we propose hierarchical attention mechanism with the structure of "local–global". This mechanism focuses on the contribution degree of individual features to POI recommendation in a local way, and excavates the contribution degree and hidden information of POI recommendation from the combination features and overall features as a whole.
- We first propose an NLP-based "User-POI" matching mechanism in the POI recommendation field. This mechanism mines "User-POI" *matching degree*[6] from the semantic similarity of user comments and POI description. Then, we propose a fine-tuning function based on "User-POI" matching degree, which used "User-POI" matching degree to fine-tune the POIs predicted by the system to obtain more accurate POIs.
- Considering the challenges of large-scale data operations, heavy user relationships maintenance, and cold start of new users, we constructed 3 data sets and several different evaluation scenarios to demonstrate the effectiveness of the hierarchical attention mechanism, "User-POI" matching degree, and the use of unstructured data. The experimental results show that the recommended performance of HAM-POIRec is optimal when compared with methods such as DeepPIM, SAE-NAD, MGMPFM and LRT, especially in predicting sequence POIs and cold-start problems.

The organization of this paper is as follows: Section 2 briefly reviews the literature on POI recommendation system. Section 3 describes the problem formulation. Section 4 describes the framework and implementation details of the HAM-POIRec. Section 5 reports the results of experiments followed by the conclusions and future work in Section 6.

[6] Matching degree indicates the possibility that the user likes the POI.

Table 1
Analysis of related work on POI recommendation.

Category	POI recommendation	Data	Advantages	Disadvantages
Social network	Collaborative filtering based on social network	Structured data: social relationships	These algorithms is simple and highly available;	(1) Low accuracy; (2) Inability to process large-scale data; (3) Data sparsity; (4) Cold-start problem for new users and new POIs.
	Matrix factorization based on social regularization			
	Self-Attentive Autoencoders[a]	Structured data: social relationships, geographical influence, temporal context	(1) Ability to handle large-scale data; (2) Ability to model the impact of social relationships and sequential pattern of user.	(1) Data sparsity; (2) New users and new POIs cold-start problem.
Check-in data context and topic content	User-based and item-based collaborative filtering	Structured data: rating data	The algorithm is simple and highly available;	(1) Low accuracy; (2) Inability to process large-scale data;
	Geographic factorization machine	Structured data: geographical influence, temporal context.		
	Collaborative filtering based on Naive Bayes			
	Basic RNN[a]	Structured data: geographic distance, temporal context, rating data, attributes of user and POI. Unstructured data: comments	(1) Mitigated the cold-start problem; (2) Ability to handle large-scale data; (3) Automatically extract user and POI features; (4) High scalability.	Problems with gradient disappearance, etc.
	Attention-based RNN[a]			These methods can further mine more hidden features and improve the accuracy of feature extraction.

[a]Indicates that the algorithm uses deep learning.

2. Related works

In the past few years, many researchers put forward the solution for recommending new locations to the users into practice, which has gradually evolved into the research field of POI recommendation [27]. The POI recommendation methods mainly include two categories: One is a POI recommendation-based on Social network; the other is a recommendation method based on check-in data context and topic content, which has recently started to use the deep learning methods based on unstructured data.

2.1. POI recommendation-based on social network

Traditional POI recommendations are based on social network methods. It mainly considers the common interests and the social relationships among friends [28–31]. For example, Ma et al. [11] found that social relationships are very beneficial to the POI recommendation system. Ye et al. [32] and Cheng et al. [13] proposed a POI recommendation method based on collaborative filtering for User-POI matrix, POI location information and social networks between the users. These methods search for the users' friends in social networks and recommend POIs that their friends have visited, that is, friends will like the same POIs. Si et al. [33] proposed a Collaborative Filtering (CF) method for the active users and the inactive users, which takes into account the user's check-in attributes and time attributes. In addition, Matrix Factorization (MF) model is a flexible model containing various potential factors, which can be used for POI recommendation. Lian et al. [14] proposed the Joint Geographical Modeling and Matrix Factorization (GeoMF) and Two-Dimensional Kernel Density Estimation (TDKDE), which alleviated the matrix sparsity problems existing in collaborative filtering. These methods pursue simple structure and high availability, but the accuracy is not high enough. The inner products that they rely on also limit their inability to handle large-scale data. To this end, the latest research applies a deep neural network based on attention mechanism, such as Chen Ma et al. [15] proposed a self-attentive autoencoders based on attention mechanism. This model can use social information for POI recommendation in a big data environment, but it relies on social network information cannot help the users mine high-quality long-tail POIs.

2.2. POI recommendations based on check-in data context and topic content

In recent years, researchers have begun to utilize the context and content of the check-in data in POI recommendations. This method mainly extracts latent features from auxiliary information such as text, and constructed the association between the users and POIs from the content matching of information. For example, Li et al. [17] proposed a new matrix factorization with geographical based on sorting that using geography and temporal context to recommend POIs. Li et al. [34] introduced the topic classification of POI into the computational model and proposed a unified POI recommendation method, which has higher recommendation performance than the model that uses the geographic information or social relationships alone. However, the recommendation accuracy of these methods are not high, and they cannot process massive data. Therefore, researchers have recently started to introduce deep learning into the field of POI recommendation. For example, Feng et al. [35] proposed a POI embedding model (POI2Vec), which trains the check-in data including geographic influences into the feature vector. The structure of the POI2Vec is similar to the word embedding in machine learning, which promotes deep learning model to improve POI recommendation performance by improving data utilization. For the context of check-in data, Liu et al. [12] introduced the RNN model, and Ma et al. [15] proposed a attention mechanism to extend the RNN model, and their work greatly improved the POI recommendation performance. In addition, Liu et al. [19] considered user text information to enrich the POI expression features to improve recommendation accuracy. In order to improve the recommendation performance of POI, Chang et al. [20] proposed a neural network based on single-layer attention mechanism, and used structured and unstructured data (text and image). These methods alleviate the cold start problem in POI recommendations to a certain extent.

On the whole, in the latest POI recommendations, researchers not only introduce the deep learning method and attention mechanism, but also begin to consider the use of unstructured data, which greatly promote the development of POI recommendation field. However, we observe that there is still a lot of room for improvement in the POI recommendation field. For instance, the "User-POI" relationships can be found from the similarity between user text and POI text.

3. Problem formulation

One of the most difficult challenges for mobile internet platforms based on LBSNs is how to accurately predict the next POI. As shown in Fig. 1, in the process of solving this challenge, the researches are using social network methods that use information such as relationship, time and distance have become common, and the deep learning method based on text information has emerged. Although this information contains many hidden features that improve the accuracy of POIs prediction, we observe that the similarity between user comments and POI descriptions can widely describe the relationships between users and POIs. Thus, the challenge from the accurate prediction of POI can be refined into: How to extract more representative explicit features and implicit features from structured data and unstructured data respectively? How to analyze the "User-POI" matching degree from the similarity between the users' comments and the POI description to improve the prediction accuracy?

To facilitate presentation, we formalize related definitions, symbols, and questions as follows:

Definition 1. Each POI P consists of POI index l and POI description g, which is represented as a tuple $P = \{l,\ g\}$. Besides, u is denoted as the user index and U represents the set of users, then $u \in U$.

Definition 2. The check-in data is a record to show the users' checking-in the POI at time t. We denote the check-in data at time t as a tuple $C_t = \{p,\ u,\ r,\ l,\ g,\ m,\ w,\ h\}$, where p represents the number of the check-in data, r denotes the comment of user u for POI l, and m, w and h represent the month, week and hour of check-in data respectively.

Problem. We define features extracted from structured data as explicit features, denoted as $C_e = \{p,\ u,\ l,\ m,\ w,\ h\}$, and features extracted from unstructured data as implicit features, expressed as $C_i = \{r,\ g\}$. So, the tuple C_t can be expressed in another form: $C_t = \{C_e,\ C_i\}$. The above problems can be formalized as:

$$\begin{aligned} &Input: C_e = \{p,\ u,\ l,\ m,\ w,\ h\}, C_i = \{r,\ g\} \\ &Output: TopK \\ &Constraints: \begin{cases} H_e \leftarrow C_e,\ H_i \leftarrow C_i \\ R = f(H_e,\ H_i) \\ S := \arg\max\ (Sim(r,\ g)) \\ TopK \leftarrow g(R,\ S). \end{cases} \end{aligned} \tag{1}$$

Specifically, we first extract their explicit features H_e and implicit features H_i from structured data C_e and unstructured data C_i, respectively, and use the prediction function $f()$ to calculate the POIs R that the users will visit; Secondly, *Sim()* calculates the similarity S of the user text r and the POI text g, and finally uses the function $g()$ to fine tune the predicted POIs R to obtain more accurate top k POIs, namely *TopK*.

4. Proposed approach

In order to solve the problems as shown in Eq. (1), we propose a HAM-POIRec method based on the hierarchical attention mechanism. For the sake of obtaining more accurate auxiliary information from structured data and unstructured data, this method proposes a hierarchical attention mechanism to obtain more latent features, and use text similarity to strengthen and improve the accuracy of recommendations. For example, for a user who often works overtime late at night, the model can analyze the coffee taste he likes based on the his previous evaluation of the coffee shop, and finally recommend a newly coffee shop that matches this taste to the user, although this newly coffee shop is in a humble alley. As shown in Fig. 2, the overall framework of the HAM-POIRec method mainly includes the following modules:

- **Explicit feature extraction model**: We propose a hierarchical attention mechanism based on Encoder–Decoder (as shown in Section 4.2), which extracts display features from structured data.
- **Implicit feature extraction model**: We propose an NLP-based attention mechanism (as shown in Section 4.3) that extracts implicit features from unstructured data.
- **"User-POI" matching mechanism**: We first propose the "User-POI" matching mechanism in the field of POI recommendation (as shown in Section 4.4), which can mine the "User-POI" matching degree from text similarity.
- **POIs predictor**: The POIs predictor (as shown in Section 4.5) uses the *softmax* function to predict POIs from explicit features and implicit features, and then uses the "User-POI" matching degree to fine-tune the POIs to obtain the final predicted result.

To better understand the HAM-POIRec method, we first introduce the pre-knowledge (i.e., the LSTM-based "Encoder–Decoder" as shown in Section 4.1), and then introduce the above modules in the order of execution.

4.1. LSTM-based encoder–decoder

The POIs we visited are coherent in time and geography. For example, on weekdays, we first visit the breakfast shop, the subway station, and the office in turn, all of which are in the same geographical areas. In addition, the highlight of our model is the analysis of texts such as descriptions and comments in the check-in data, which have the sequence characteristics of words in these texts. Therefore, our model needs to build a function that it is able to capture the temporal characteristics of these data. In common neural network models, Autoencoders (AE), Restricted Boltzmann Machines (RMB), and Convolutional Neural Networks (CNN) have the flaw that they cannot handle sequence problems [36,37]; Deep Belief Networks (DBN) can only capture short-term sequence features [38]; although RNN has better performance than DBN in extracting sequence features, it still cannot process long-term sequence features. That is to say, none of these basic models can extract long-term sequence features (i.e. contextual semantics) from text data. Fortunately, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models (i.e. two variants of RNN) can overcome the problem of excessive information transmission interval in RNN, so as to capture long-term sequence features [39]. Hence, we choose LSTM and GRU as the calculation unit of Encoder–Decoder structure.

The POIs sequence length of each user is different, and the text length is also different. That said, in addition to dealing with sequence prediction problems, we also face a bigger challenge: the input sequence and the output sequence have different lengths. Therefore, we build an LSTM-based Encoder–Decoder framework to address this challenge [40]. The framework uses an end-to-end computing model. Its operation is that the encoder maps the input sequence data to a fixed-length intermediate state vector, and then the decoder maps the intermediate state vector to expression characteristics of the users and POIs.

We first use the encoder to encode the input sequence into fixed-length feature vectors, which are expressed as follows:

$$\tilde{c}_t = tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{2}$$

where $\tilde{c}_t$ denotes the candidate state obtained by the nonlinear function, h_{t-1} represents the external state obtained after the LSTM model outputs information to the hidden layer at time

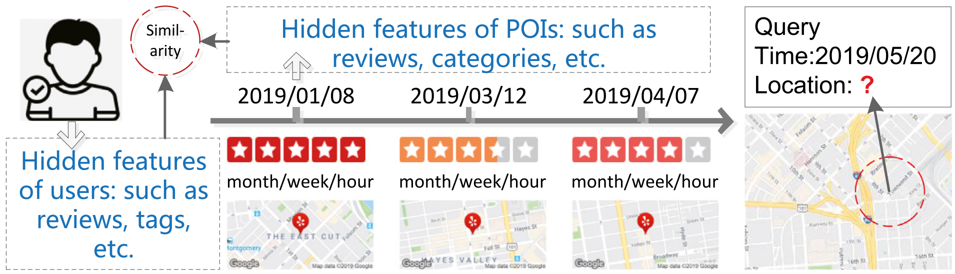

Fig. 1. Application scenario of POI recommendation.

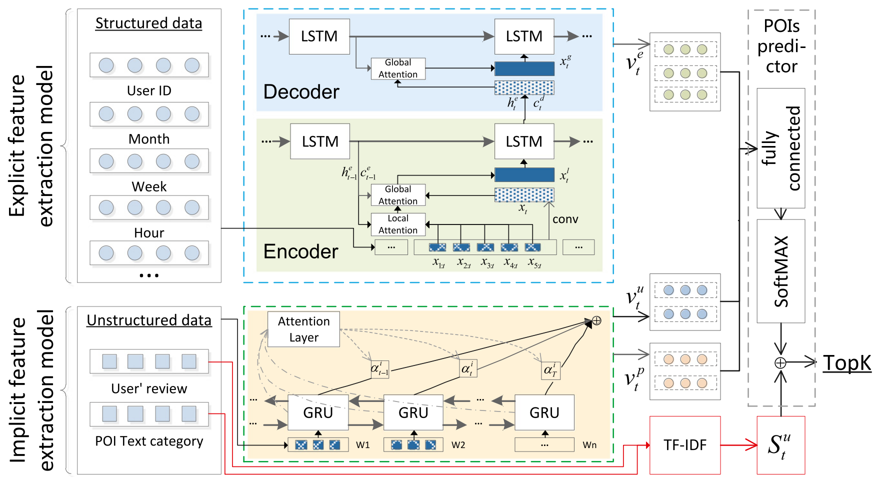

Fig. 2. The architecture of the proposed approach.

$(t-1)$, x_t denotes the training data. W_c, U_c, and b_c represent the learnable network parameters, and *tanh* [18] is an activation function that increases the nonlinearity of the neural network.

The LSTM model introduces a gate mechanism to control the path of information transfer. As shown in Eqs. (3) and (4), the three gates of the LSTM model are represented as input gate i_t, forgetting gate f_t, and output gate o_t, respectively. Assuming that c_t^d represents a new internal state generated by the LSTM model at time t, then the internal state c_t^d is calculated by forgetting gate f_t and input gate i_t as:

$$c_t^d = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{3}$$

where the c_t^d specializes in linear circular information transmission, and $\odot$ is expressed as the product of vector elements.

Assuming that h_t^e represents the external state obtained after the LSTM model outputs information to the hidden layer at time t, then the internal state c_t^d is calculated together with input gate i_t to get external state h_t^e as:

$$h_t^e = o_t \odot tanh\left(c_t^e\right) \tag{4}$$

where the external state h_t^e is used as the input state of the time $(t+1)$ to form a recurrent neural network.

Second, a decoder is utilized to decode the fixed-length feature vector into a prediction sequence, which is expressed as follows:

$$P\left(y_t \left| \{y_1,\ y_2,\ \ldots,\ y_{t-1}\}, h_t^e\right.\right) = g\left(y_{t-1},\ c_t^d,\ h_t^e\right) \tag{5}$$

where $g\left(\right)$ is the output function of the decoder (the commonly used function is *softmax* [41]), y_t represents the probability of the target vector output by $g\left(\right)$ function, and $P\left(\right)$ is the probability function.

4.2. Explicit feature extraction model

Obviously, structured attributes (i.e. features) can directly influence the recommend effect, and the influence is multi-level. Firstly, the individual features of users and items not only have a direct impact on recommend effect, but also have different contributions to recommendation. For example, in a scenario where a user selects a food store, features (such as time, location, taste, etc.) all affect the user's decision-making on the food store, and it is easy to see that the location is more important than other features. Secondly, between features (i.e. combination features) will also affect the recommend effect, so much so that the wrong combination will cause the recommend effect to deteriorate. That is to say, users who do not like to get up early will abandon the food store that only provides breakfast (that is, only considering individual features will lead to inaccurate recommendation results). In this case, if the time feature is ignored, and the combination of location and taste is selected by mistake, the system will not be able to recommend a suitable food store. Finally, from the overall point of view, we take into account the different weights of the overall features and combination features, so as to adjust the recommended prediction results. In other words, considering the overall effect of features(such as time, place, taste, etc.), increasing the weight of effective combination features and reducing the weight of invalid combination features, these ways enable the system to recommend a comprehensive favorite food store for users.

As a whole, we found that the multi-level impact of structured attributes on recommendations includes the influence of individual features, combination features and overall features. Therefore, as shown in the green part of Fig. 2, on the basis of LSTM-based "Encoder–decoder", we propose a hierarchical attention mechanism of "local–global" structure, forming a new explicit feature extraction model. The model can analyze the vectorized structural attributes, mine the auxiliary information contained in the individual features, combination features and overall features of the user (or item), and mine the different contributions of these auxiliary information to the recommendation to improve the accuracy recommended.

Local attention mechanism: In different scenarios, not all features can represent the user's willingness to visit a POI. Thus, we propose a local attention mechanism to find important features from the historical structured features, so that the HAM-POIRec method can improve the utilization of important features. This mechanism calculates the contribution degree (i.e. attention distribution) of user's each feature to the prediction POI at time t by weighted aggregation calculation. Assuming that $x_{i:t}$ is the ith feature at time t, the x_t is then defined as follows:

$$x_t = (x_{1:t},\ x_{2:t},\ \ldots,\ x_{i:t})\,. \tag{6}$$

Firstly, we input the additive model [42] with the feature vectors $x_{i:t}$ at the time t, the internal state c^d_{t-1} and the external state h^e_{t-1} of the encoder at the time $t-1$, and then obtain the attention score $a^l_{i:t}$. The specific process can be shown as follows:

$$a^l_{i:t} = V_i^T tanh\left(W_i(h^e_{t-1} \oplus c^e_{t-1}) + U_i x_{i:t} + b_i\right) \tag{7}$$

where V_i, b_i, W_i and U_i represent the network parameters that can be learned, and T denotes the inversion of a matrix.

Secondly, we use the *softmax* function to regularize attention score $a^l_{i:t}$ to the attention weights α^l_t (also known as the attention distribution) as:

$$\alpha^l_t = \frac{\exp\left(a^l_{i:t}\right)}{\sum_{i=0}^{k} \exp\left(a^l_{i:t}\right)},\ \ 1 \le i \le k \tag{8}$$

where the feature number k denotes the dimension of x_t.

At last, we use α^l_t and $x_{i:t}$ to calculate the local feature vector with attention weights by using the following equation:

$$x^l_t = \sum_{i=1}^{k} \alpha^l_{i:t} \cdot x_{i:t}\,. \tag{9}$$

Global attention mechanism: In the user's historical experience, there are differences in their experience of visiting POI every time, and even some POI visiting experience may be unpleasant. Besides, in each visiting experience, the contributions of different combinations of features to the recommended prediction are not the same. Therefore, by using weighted aggregate computing, we propose a global attention mechanism to extract the contribution degree (i.e. attention distribution α^g_t) of global features (including combination features) to the prediction POIs at time t. Similarly the operation of the local attention mechanism, we first use the additive model to construct the following attention scoring function, as:

$$a^g_t = V_j^T tanh\left(W_j(h^e_{t-1} \oplus c^e_{t-1}) + U_j x^l_t + b_j\right) \tag{10}$$

where V_j, b_j, W_j, and U_j denote the network parameters that the model can learn, a^g_t is expressed as a attention score, and we use CNN to reduce and merge x_t to obtain the input data x^l_t of the global attention mechanism.

Secondly, we use the *softmax* function to obtain the attention distribution α^g_t. The specific process can be shown as follows:

$$\alpha^g_t = \frac{\exp\left(a^g_t\right)}{\sum_{t=1}^{T} \exp\left(a^g_t\right)},\ \ 1 \le t \le T \tag{11}$$

where T denotes the maximum number of POIs accessed by the users.

Finally, we utilize α^g_t and x^l_t to obtain the global feature vector x^g_t with attention weights by using the following equation:

$$x^g_t = \sum_{j=1}^{T} \alpha^g_t \cdot x^l_t\,. \tag{12}$$

As shown in the blue part of Fig. 2, in order to balance the prediction accuracy and the computation speed, we only employ the global attention mechanism in the decoder. The output data of the encoder is the input data to the decoder, so the input data to the decoder is expressed as x^g_t. Also, assume that the internal and external states of the decoder are c^*_t and h^*_t, respectively. Then we substitute x^g_t, c^*_t and h^*_t into Eqs. (10)–(12) to obtain the explicit feature vector v^e_t of the unstructured data with attention weight.

4.3. Implicit feature extraction model

At present, the check-in data of the application platform based on LBSNs also contains abundant user comments and POI descriptions. These texts contain not only the words associated with the users' portraits and the POI portrait, but also the context in the checks-in data. In addition, words in user comments have a fixed order, and changes in word order can lead to changes in sentence meaning. Thus, we need to capture the hidden features of the check-in data chronologically. Based on the "Encoder–Decoder" structure which similar to Eqs. (3) and (4), we use a GRU-based Encoder–Decoder model [43]. We can easily obtain the state expression vector h^i_t of GRU cell shown in the dotted box at the green part of Fig. 2, which is the hidden state of encoder output data and the hidden state of decoder input data.

As shown in the orange part of Fig. 2, for the sake of calculating the importance of each word to user features or POI features, we propose an attention mechanism for NLP. The mechanism firstly uses the fully connected layer to extract the hidden feature a^i_t from the context of the user features or POI features (as shown in Eq. (13)), and then uses the *softmax* function to calculate the attention weight α^i_t (as shown in Eqs. (14) and (15)), and finally converts the input user comments and POI descriptions into text feature vectors v^u_t with attention weights, as follows:

$$a^i_t = tanh\left(Wh^i_t + b\right) \tag{13}$$

$$\tilde{a}^i_t = W^* a^i_t + b^* \tag{14}$$

$$\alpha^i_t = \frac{\exp\left(\tilde{a}^i_t\right)}{\sum_{t=0}^{T} \exp\left(\tilde{a}^i_t\right)},\ \ 1 \le t \le T \tag{15}$$

$$v^u_t = \sum_{t=1}^{T} \alpha^i_t \cdot h^i_t \tag{16}$$

where W, b, W^*, and b^* denote the network parameters that can be learned by the model, and T denotes the maximum length of the text.

In this paper, the attention weights of the user text and the POI text are calculated in the same way. Hence, the expression feature vector v^p_t of the POI text with the attention weights is obtained in the same way.

4.4. User and POI matching mechanism

Both structured and unstructured data have been considered in the latest POIs recommendation process, which has improved data utilization and improved POI prediction to some extent [19]. But they only considered extracting features from a single data item, ignore the associations between the data. We observe that the textual similarities between user comments and POI descriptions in unstructured data implies the degree to which they like each other. For example, if a user comment has an evaluation of "*coffee*", then he may be interested in a POI containing a description of the "*coffee*".

Therefore, as shown in the red part of Fig. 2, we first propose a mechanism to calculate the "User-POI" matching degree by using text similarities, which consists of two modules: word weight calculation and text similarity calculation, respectively. Firstly, we use the TF-IDF [44] algorithm to calculate the weight $W\left(r_i^t, g^n\right)$ of the ith word r_i^t of the user comment r^t at time t in the nth POI description g^n. The process can be expressed as follows:

$$W\left(r_i^t, g^n\right) = \frac{TF\left(r_i^t,\ g^n\right)}{DF\left(r_i^t, \left(g^1,\ g^2, \ldots, g^n, \ldots,\ g^N\right)\right)} \tag{17}$$

where $(g^1, g^2, \ldots, g^n, \ldots, g^N)$ denotes the POI description sets, and N denotes the number of POIs. We express the Term Frequency (TF) function as $TF\,()$, which is used to count the number of words of the user comment r^t in the POI description g^n. The Inverse Document Frequency (IDF) function is denoted as $DF\,()$, which is used to count the number of words of the user comment r^t in all POI descriptions.

Second, we take the first q representative keywords of r^t and g^n, denoted as r_q^t and g_q^n, respectively. We use the cosine similarity function to operate on r_q^t and g_q^n to obtain the similarity between r^t and g^n. The process can be shown as follows:

$$Sim\left(r^t, g^n\right) = \frac{\sum_{q=1}^{n} r_q^t g_q^n}{\sqrt{\sum_{q=1}^{n}\left(r_q^t\right)^2}\sqrt{\sum_{q=1}^{n}\left(g_q^n\right)^2}}. \tag{18}$$

After the operations of Eqs. (17) and (18), we obtain the similarities between the tth comment r^t of the user u and the nth POI description g^n. On this basis, after traversing all the comments of user u and the similarity operations of all POI descriptions, we obtain the similarity set S_u of user u for all POIs.

4.5. POIs predictor

We combine the explicit features v_t^e and implicit features (user text features v_t^u and POI text features v_t^p) into the overall features vector v as:

$$v = v_t^e \oplus v_t^u \oplus v_t^p. \tag{19}$$

Then, we enter the overall features vector v into the ReLU unit [45] of fully connected layer to get the weighted feature vector v', shown as follows:

$$v' = ReLU\left(vW' + b'\right) \tag{20}$$

where W' and b' denote the network parameters of the fully connected layer. We use the *softmax* function to normalize the vector v' for obtaining the POIs probability distribution y_i' as:

$$y_i' = tanh\left(v'W'' + b''\right),\ i = 1,\ 2,\ \ldots,\ N \tag{21}$$

where W'' and b'' denote the network parameters that the model can learn, and N is the maximum length of POIs. After sorting y_i', we get the best POIs recommendation list.

Further, we need to utilize similarity set S_u of user u for all POIs as the influence weight to fine-tune y_i'. That is, we use a nonlinear method to map S_u to a suitable threshold W^s as:

$$W^s = tanh\left(S_u W^o + b^o\right). \tag{22}$$

Then, we use the threshold W^s to fine tune y_i' with the addition to obtain the predicted POI probability distribution $\widehat{y}$, which can be shown as follows:

$$\widehat{y} = y' + W^s,\ y' = \left[y'_1,\ \ldots,\ y'_i,\ \ldots,\ y'_N\right]. \tag{23}$$

Table 2
The statistics of the experimental data sets.

Data sets	Filter criteria				Data description		
	N_t	N_c	N_p	N_u	C_u	C_p	C_c
yelp-b	100–1200	20	80	2	3957	37 684	630 960
yelp-c	100–1200	100	4	3	14 983	8452	17 270
yelp-h	100–1200	40	50	40	4195	2617	18 5061

In the end, we sort the predicted POIs probability distribution $\widehat{y}$, so the first k POIs are the best k POIs predicted by the system for user u at the time t.

In addition, we use Cross Entropy (CE) [20] as loss function to train our model, as shown below:

$$Loss = -\sum_i y_i \log \widehat{y}_i \tag{24}$$

where the predicted POIs is expressed as $\hat{y}_i$ and y_i denotes the POI probability distribution. We gradually optimize our model according to the loss function in the training phase.

5. Evaluation

5.1. Experimental settings

5.1.1. Datasets

We utilize the Yelp Dataset Challenge Round 13[7] of the world's largest review site. For the sake of verifying the comprehensive performance of the algorithm, as shown in Table 2, we construct a large-scale data set *yelp-b*, a cold-start user data set *yelp-c*, and a heavy user data set *yelp-h* (the heavy user refers to the old users who reuse a product or service) according to different screening conditions. Among them, N_t represents the length of the string, N_c is the maximum number of check-in data for a single user, and the maximum number of the users visiting a single POI is N_p, the maximum number of POIs visited by a single user is N_u; the number of all users in the dataset is C_u, and the number of all POIs in the dataset is C_p, and C_c represents the total number of check-in data in the dataset. Besides, in this paper, POI description uses business categories, and our training dataset, validation dataset, and test dataset ratio is 8:1:1.

5.1.2. Compared methods

To fully validate the recommended performance of our proposed approach, we select two traditional POI recommendation methods (LRT and MGMPFM) and two new machine learning-based POI recommendation methods (SAE-NAD and DeepPIM). Among them, in the machine learning-based recommendation method, SAE-NAD utilizes structured data, and DeepPIM employs structured data and unstructured data.

Location recommendation framework with Temporal effects (LRT) [46] is a location recommendation framework based on matrix decomposition model. It can analyze the user's time characteristics from the LBSN dataset, that is, it uses the strong correlation between the check-in time and the check-in location to improve the recommendation performance.

POI recommendation based on Multi-center Gaussian Model and Poisson Factor Model (MGMPFM) [13] is a POI recommendation model based on Poisson Factor Model (PFM), which integrates the geographic influence and social information captured by Multi-center Gaussian Model (MGM), and its recommendation performance is significantly higher than the traditional Matrix Factorization (MF) Model.

[7] https://www.yelp.com/dataset.

Table 3
The statistics of the data items used in each compared methods.

Model	Structured data	Unstructured data
LRT MGMPFM SAE-NAD	u, l, t, poi category, longitude, latitude, social relations (i.e. tuple $\{u_i, u_j\}$), number of u visits to l	-
DeepPIM DeepPIM-P DeepPIM-P-V	$C_e = \{p,\ u,\ l,\ m,\ w,\ h\}$	r r, g r, g, visual feature
HAM-POIRec HAM-POIRec-V	$C_e = \{p,\ u,\ l,\ m,\ w,\ h\}$	r, g r, g, visual feature

POI recommendation based on exploiting Self-attentive AutoEncoders with Neighbor-Aware Influence (SAE-NAD) [15] is a POI recommendation method, which consists of a Self-Attentive Encoder (SAE), a Neighbor-Aware Decoder (NAD) and a multi-dimensional attention mechanism. This method uses multidimensional attention to consider geographic context information to improve the performance of POI recommendations.

Deep neural Point-of-interest Imputation Model (DeepPIM) [20] is a POI recommendation based on machine learning. This method is implemented by using the Encoder–Decoder model based on the attention mechanism, which considers both structured and unstructured data (i.e. image and text information).

Our proposed HAM-POIRec is a POI recommendation method based on machine learning. The method proposes a hierarchical attention mechanism and a "User-POI" matching mechanism, and utilizes structured and unstructured data in the training process.

In the above comparison methods, the LRT, MGMPFM, and SAE-NAD rely on social network information to study POI recommendation. Besides, the DeepPIM and HAM-POIRec investigate POI recommendations based on check-in data context and topic content. The implementation principles of these two methods are different, which result in different data used by their training models. As shown in Table 3, the social network-based POI recommendation method mainly uses social network information (such as the relationships between all users who have visited a POI), and the topic and content-based POI recommendation mainly uses content and topic information (such as text and image). Among them, we use the VGG16[8] neural network to extract the "*visual features*" from the pictures of POI. u_i and u_j represent the ith user and the jth user, respectively.

5.1.3. Evaluation metrics

We found that in practical application scenarios, users visit the favorite POIs in a temporal manner. Therefore, we choose Mean Reciprocal Rank (MRR), Mean Average Precision (MAP@K) and Intersection over Union (IoU@K) [47] as our evaluation indicators. Tok-K is the first K POIs recommended by the system for a user.

MRR is an international common mechanism for evaluating search algorithms. The MRR can evaluate the performance of POI recommendation algorithm in time series prediction, which can be expressed as:

$$MRR = \frac{1}{|D_{test}|}\sum_{i=1}^{|D_{test}|}\frac{1}{rank_i} \tag{25}$$

where $rank_i$ is the rank of the first correct answer for the ith POI, D_{test} is the test set, and $|D_{test}|$ is the number of data sets.

MAP is a comprehensive recommendation evaluation index that focuses on sequence weights. The *Top-K* of MAP can be expressed as:

$$MAP@K = \sum_{u=1}^{M}\left(\frac{1}{K}\sum_{i=1}^{K}\frac{tp_u}{tp_u + fp_u}\right) \tag{26}$$

where M is the number of all users, i is the first i POIs of user u, the number of POIs visited by the user u is represented as tp_u, and the number of POIs that the user u has not visited is expressed as fp_u.

IoU is a comprehensive indicator for visually evaluating the performance of POI recommendation. IoU@K can be expressed as:

$$IoU@K = \frac{tp_u}{tp_u + fp_u + tn_u} \tag{27}$$

where the number of POIs visited by user u is expressed as tp_u, the number of POIs not visited by user u is expressed as fp_u, and tn_u is the number of POIs visited by user u but not within the range of *Top-K*.

5.1.4. Parameter optimization

For the LRT and MGMPFM in the traditional POI recommendation method, we have found through many experiments that the parameters of the Ref. [48] have reached the best. Namely, the hyperparameters of LRT are $K = 100$, $\lambda = 1.0$, $\beta = 2.0$, $\alpha = 2$, $T = 24$; the hyperparameters of MGMPFM consists of two parts: PFM and MGM, in which the hyperparameters of PFM is $K = 30$, $\alpha = 20.0$, $\beta = 0.2$ and the hyperparameters of MGM is $\alpha = 0.2$, $\theta = 0.02$, $d = 15$.

We provide the optimal hyperparameter settings for each comparison method. As shown in Fig. 3, we take several important hyperparameters as examples: Dropout Rate [49] is the retention rate of neuron in the training process, which is used to reduce the over-fitting problem of neural network. Its value depends on the number of neurons. Specifically, if the number of neurons is large, the value is initialized to *0.8*, otherwise it is initialized to *0.5*, and then the optimal value is searched in the range of *0.05* or *0.1*. The experimental results show that when the dropout rate of the SAE-NAD is *0.2*, and the dropout rate of DeepPIM and HAM-POIRec methods is *0.6*, their MAP values are best. Learning Rate is a hyperparameter that guides us how to adjust the weight of the network through the gradient of the loss function. Its value is based on *5–10* training iterations, the exploration range is $0.1 - 10^{-8}$, and the range of each exploration is expanded or contracted by *10* times. When the learning rate of all methods is *0.01*, the MAP value is best. Regularization parameter (L2) is a regularization parameter for the balance between data fitting and preventing overfitting. Its value starts with a very small value, and is explored with a *5–10* times zoom. After getting a suitable magnitude, it is adjusted with a very small value (such as *0.01*). The DeepPIM and HAM-POIRec methods are best when the value of the regularization parameter (L2) is *0.0001*. At the same time, the number of ReLU units is explored in units of *100*. When the number of ReLU units is *500*, the MAP value is best. Furthermore, for machine learning-based methods such as SAE-NAD, DeepPIM, and HAM-POIRec, the loss function tends to be stable at *40* (*i.e. epoch* = 40) training iterations.

5.2. Performance evaluation under different data sets and computational components

The basic DBN consists of multiple layers of RBM [50], which can only process one-dimensional data and is difficult to capture spatial correlation features, while CNN can process high-dimensional data to capture spatial correlation features [38]. Furthermore, it can be known from Ref. [51] that CNN can effectively solve the gradient dispersion problem compared with RBM, DBN, and AE, which has fewer parameters, better convergence performance, and extract richer expression features. Therefore, we use CNN-based Encoder–Decoder as a comparative experiment. As shown in Fig. 4, in the basic architecture of the HAM-Rec approach, the values of various indicators of the LSTM-based

8 https://github.com/machrisaa/tensorflow-vgg.

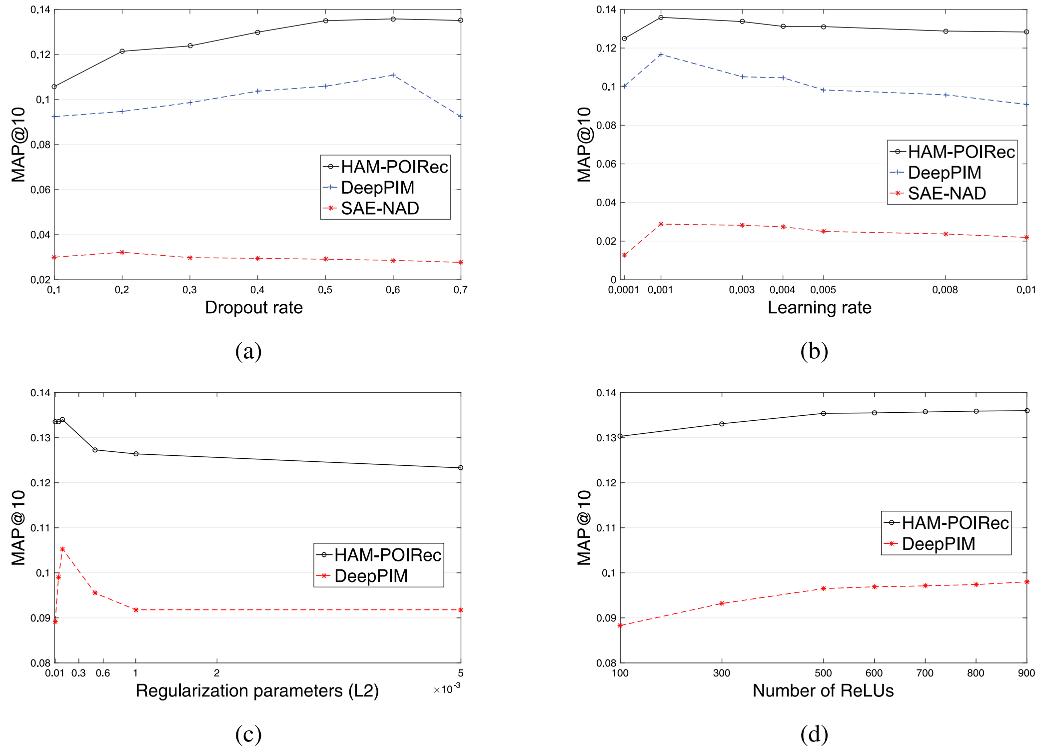

Fig. 3. Parameter optimization process in yelp-h dataset.

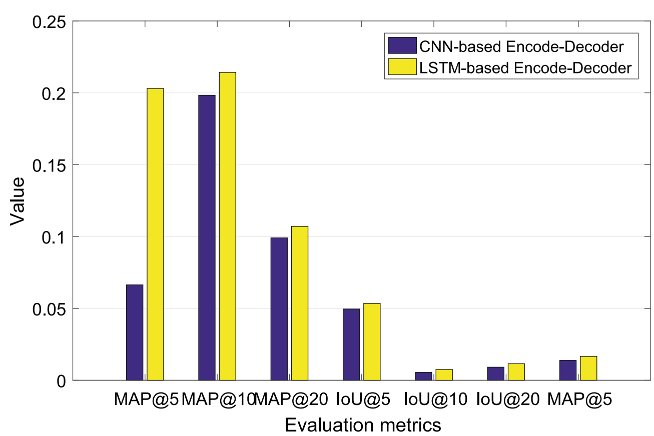

Fig. 4. Performance comparison of different Encoder–Decoder architectures.

Table 4
Performance comparison of our model under different conditions on yelp-h data set. *E*, *U*, *P*, and *S* denote the explicit feature, user text, POI text, and similarities between the users and POIs respectively. $-n$ denotes data are not computed by neural network. $-a$ means no attention mechanism is used.

Model	*MRR*	*MAP@5*	*MAP@10*	*IoU@5*	*IoU@10*
$E^{-n-a}+U$	0.1279	0.1835	0.0918	0.0062	0.0101
$E^{-a}+U$	0.207	0.2287	0.1143	0.0078	0.0118
$E+U$	0.2886	0.2375	0.222	0.01	0.0144
$E+U+P$	0.941	0.2677	0.1339	0.0158	0.0168
$E+U+P+S$	**0.9424****	**0.2691****	**0.1346***	**0.0159***	**0.0169***

*Denote the statistical significance for $p < 0.05$, compared to the best baseline.
**Denote the statistical significance for $p < 0.01$, compared to the best baseline.

Encoder–Decoder framework are significantly better than the CNN-based Encoder–Decoder. Experiments show that it is correct to use LSTM as the unit of Encoder–Decoder, which can capture temporal characteristics, and can improve the recommended precision of POI.

For purpose of analyzing the impact of each data item in the unstructured data on the recommended effect, and the advantage of each computing component to the recommendation, we design an experiment to show the improvement of recommendation performance of unstructured data and various computational models by gradually adding unstructured data and computational models. Among them, the added unstructured data are user comments and POI descriptions, and the computational models include LSTM-based Encoder–Decoder, hierarchical attention mechanism, and "User-POI" matching mechanism. As shown in Table 4, after adding the LSTM-based Encoder–Decoder and attention mechanism respectively, the MRR value is gradually increased. The recommended performance is significantly improved after the POI description is increased, and the MRR value is increased from 0.2886 to 0.941. The performance improvement of adding POI description is much greater than adding user comments. It also shows that the text written by the store or professional accurately describes the characteristics of POI, while the user's comments are casual, and may even contain other interference information unrelated to interest. That means that the more accurate the collected text information, the better the recommended performance of the model. In the end, after fine-tuning the POIs with the "User-POI" matching degree, the recommended performance is the best, that is, the MRR value is 0.9424. Therefore, it is verified that the "Encoder–Decoder" model based on attention mechanism is the most stable measure to improve the recommendation accuracy. At the same time, it also shows that our ideas of adding unstructured data and using "User-POI" matching degree extracted from text similarity to fine-tune POIs are very effective ways to improve the prediction performance.

Table 5
Performance comparison results of various methods on big data set.

Model	*MAP@5*	*MAP@10*	*MAP@20*	*IoU@5*	*IoU@10*	*IoU@20*
LRT	0.0184	0.0096	0.006	0.0035	0.0048	0.0062
MGMPFM	0.0053	0.0027	0.0017	0.00072	0.0012	0.0018
SAE-NAD	0.0319	0.0344	0.0387	0.0218	0.0223	0.0208
DeepPIM	0.1004	0.0515	0.0292	0.0134	0.0163	0.0184
DeepPIM-P	0.2184	0.1172	0.0761	0.045	0.0371	0.029
HAM-POIRec	**0.2256****	**0.1209****	**0.0786****	**0.046****	**0.0376***	**0.0294***

*Denote the statistical significance for $p < 0.05$, compared to the best baseline.
**Denote the statistical significance for $p < 0.01$, compared to the best baseline.

Table 6
Performance comparison results of various methods for Heavy User on yelp-h data set.

Model	*MAP@5*	*MAP@10*	*MAP@20*	*IoU@5*	*IoU@10*	*IoU@20*
LRT	0.0659	0.0329	0.0165	0.0023	0.0045	0.0071
MGMPFM	0.0247	0.0124	0.0062	0.0016	0.0025	0.0032
SAE-NAD	0.0671	0.0457	0.0438	0.0341**	0.046**	0.0548**
DeepPIM	0.1877	0.0938	0.0469	0.0060	0.0098	0.0149
DeepPIM-P	0.1912	0.0956	0.0478	0.0061	0.0099	0.0151
HAM-POIRec	**0.2699***	**0.1349***	**0.0673***	0.0159	0.0169	0.018

*Denote the statistical significance for $p < 0.05$, compared to the best baseline.
**Denote the statistical significance for $p < 0.01$, compared to the best baseline.

5.3. Comparison of various methods in different environments

The POI recommendation system is facing three of the most important challenges: large-scale data operations, relationship maintenance for older users (i.e. heavy users), and cold start issues for new users. Hence, we design the following three comparative experiments:

5.3.1. Evaluation on the large-scale dataset

Table 5 shows the experimental results of all POI recommendation methods in a large-scale data environment. It shows that the recommended performance of the machine learning based methods (HAM-POIRec, DeepPIM and SAE-NAD) is significantly better than those of traditional recommended methods (MGMPFM and LRT methods). The most likely reason is that the model fitted by large-scale data is more expressive. Furthermore, the recommended performance of the recommended methods based on data of unstructured and structured (HAM-POIRec and DeepPIM methods) is significantly better than the recommendation based only on structured data (SAE-NAD, MGMPFM, and LRT). Especially under the same single-layer attention mechanism, the recommendation effect of the DeepPIM is significantly higher than that of the SAE-NAD, which indicates that unstructured data is helpful enough to improve POI recommendation performance. It also proves that our HAM-POIRec method is very meaningful for improving the utilization rate of unstructured data. Although both HAM-POIRec and DeepPIM methods use attention-based neural networks to process unstructured data and structured data, HAM-POIRec achieves best recommendation performance because it uses hierarchical attention mechanism and "User-POI" matching to fine-tune POIs.

5.3.2. Evaluation on relationship maintenance for heavy users

As shown in Table 6, in a heavy user environment, the IoU values that focus on recommendation accuracy without regard to the POIs visit sequence show that the SAE-NAD is slightly better than the HAM-POIRec and DeepPIM, and the MAP values which focus on the POIs visit sequence show that the recommended effect of the HAM-POIRec is slightly better than the SAE-NAD and the DeepPIM. This indicates that multi-dimensional attention mechanism can predict more POIs with rich structured data, but if we want to predict more accurate POIs visit sequences, we need to consider obtaining more auxiliary information from unstructured data. In other words, in addition to unilaterally increasing the utilization rate of unstructured data, how to propose an effective comprehensive algorithm to mine more useful auxiliary information from unstructured data is the correct research idea.

5.3.3. Evaluation on cold start issues

Fig. 5 is measurement of the POI recommendation performance for a cold start of a new user. Obviously, in the case of highly sparse data, the performance of the POI recommendation method based on structured data is significantly degraded, and the data level is lower than *0.001*, which tends to fail. However, the recommended effects of the HAM-POIRec and DeepPIM methods are within acceptable limits, indicating that unstructured data is very helpful for POI recommendations in cold-start environments. In this environment, HAM-POIRec's recommended performance is significantly better than other methods, in which HAM-POIRec's MAP@5 value has been increased from 0.0004 of DeepPIM-P's MAP@5 value to 0.007. This shows that the hierarchical attention mechanism considers the single feature, the combination features and the overall features to improve the utilization of structured data, and it also shows that extracting "User-POI" matching degree from text can indeed mine more implicit information of unstructured data. That is to say, the hierarchical attention mechanism and the "User-POI" matching degree can alleviate the problem of data sparsity to some extent.

5.4. Experimental results on the effects of visual features on our models

The DeepPIM-P-V method not only uses text data (i.e. user text and POI text) but also image data. Therefore, as shown in Fig. 6, we add image data to the HAM-POIRec method to form the HAM-POIRec-V method. The experimental results show that the IoU@5 value of the HAM-POIRec-V method is 1.3% higher than the HAM-POIRec method. It explains that image features can directly improve the accuracy of the recommendation, although we only use the method of feature merging to add image features. Under the same data conditions, the IoU@5 value of the HAM-POIRec-V method is 3% higher than that of the DeepPIM-P-V method, indicating that the hierarchical attention and "Uer-POI" matching mechanism proposed in the HAM-POIRec method is effective. In addition, it also shows that unstructured data does contain hidden features that promote recommendation performance.

5.5. Performance evaluation under different text lengths

In order to verify the influence of the users and POIs text length on "User-POI" matching degree, we design a comparison experiment under different text length conditions. In the experiment, taking into account the elimination of adverse effects of other factors, we divided the data set into four groups according to different string lengths (50–300, 300–550, 550–900 and 900–1600) under the conditions of $N_t = 20$, $N_p = 10$ and $N_u = 20$. That is, the size of each group of data is about 70,000 (the sizes of the four data sets are 70 393, 69 868, 71 702 and 76 605, respectively). As shown in Fig. 7, the MRR and MAP@10 values of the long text are both higher than the those values of the short text. This shows that the length of the text is longer, the information is richer, and the "User-POI" matching degree is higher; Therefore, in the recommended practical application scenarios, we should collect long text training data as much as possible.

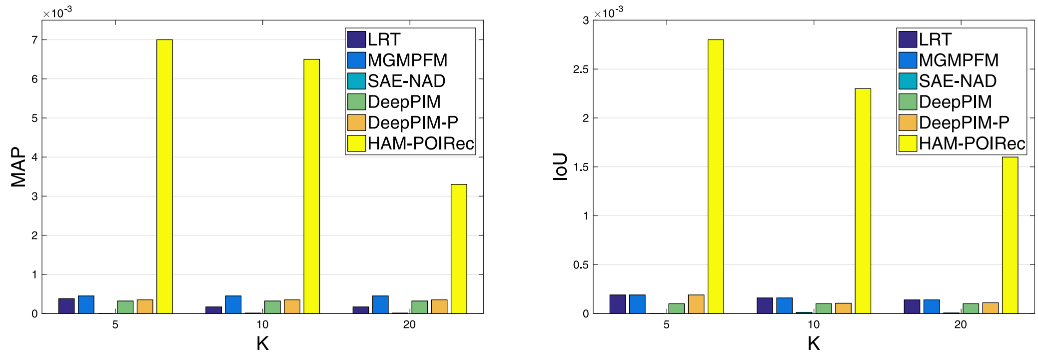

Fig. 5. The experimental results for cold start users on yelp-c data set.

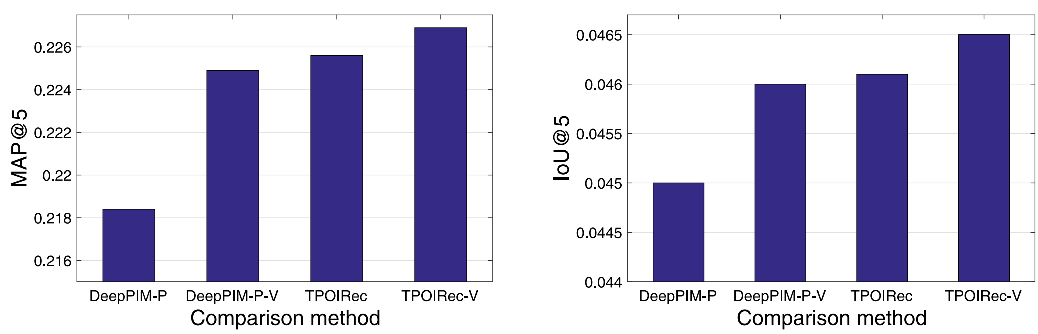

Fig. 6. Experimental results on the effects of visual features on our models.

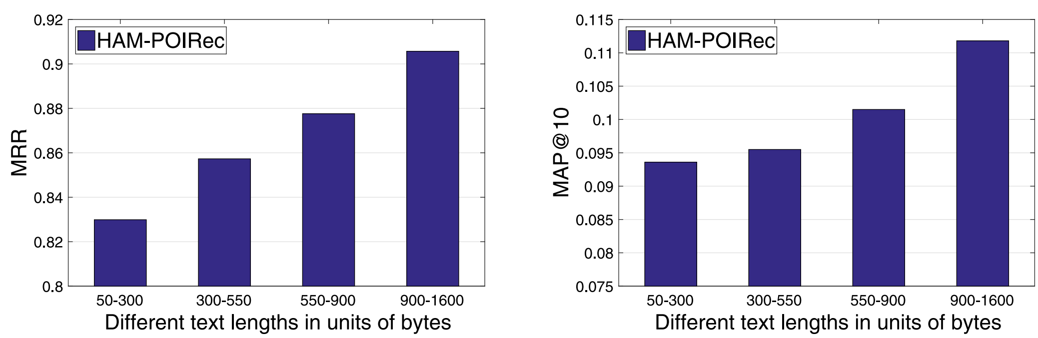

Fig. 7. Experimental results under different text lengths.

6. Conclusions

In this paper, we propose a personalized POI recommendation based on the hierarchical attention mechanism. Firstly, the approach integrates structured and unstructured data from Yelp, and on this basis we refined the concepts of explicit features and implicit features to guide the model to design different computing components based on different data. Then, we propose a hierarchical attention mechanism that extracts contributions and more hidden information from individual features, combination features, and overall features. Finally, we propose a method to extract "User-POI" matching degree from text similarity for the first time to optimize POI recommendation. Experimental results show that our proposed approach is superior to other methods in large-scale data sets, cold-start data sets, and heavy user data sets, especially in the cold-start environment. Besides, with the addition of image features, our proposed approach is 5% higher than the state-of-the-art DeepPIM. In other words, our proposed approach is better able to handle large-scale data and mitigate cold-start problems, and better excavate high-quality long-tail POIs that users like. It also shows that using hierarchical attention mechanism and extracting more hidden information from text similarity are effective ways to improve the data utilization of sparse data.

The text similarity calculation model used in our proposed HAM-POIRec approach is a rule-based non-machine learning method, which still has much room for improvement in accuracy and operation speed. Therefore we are interested in proposing an unsupervised model based on deep learning to solve the text similar in the future. Furthermore, for the purpose of increasing the resolution of processing sequence problems, we expand neural networks (i.e. RBM, DBN, and CNN) with high computation speeds to replace slower LSTMs (or GRUs), that also is an effective way to increase the computing speed of POI recommendations.

CRediT authorship contribution statement

Guangyao Pang: Conceptualization, Methodology, Software, Validation, Investigation. **Xiaoming Wang:** Resources, Supervision, Project administration, Funding acquisition. **Fei Hao:** Formal analysis, Writing - original draft. **Liang Wang:** Formal analysis, Writing - original draft. **Xinyan Wang:** Writing - original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61872228, 61702317, 61562074, 61862056), the National Key R&D Program of China (Grant No. 2017YFB1402102), the Shaanxi Provincial Key R&D Plan of China (Grant No. 2020ZDLGY10-05), the Natural Science Basic Research Plan in Shaanxi Province of China (Grant No. 2019JM-379, 2018JQ6048), the Guangxi Science and Technology Planning Project, China (Guike AB16380273), the Guangxi Innovation-Driven Development Special Fund Project, China (Guike AA18118036), the Project of Basic Ability Improvement of Young and Middle-aged Teachers of Universities in Guangxi Province, China (Grant No. 2020KY17018), the China Postdoctoral Science Foundation (Grant No. 2016M600761), the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie (Grant No. 840922).

References

[1] Y. Wang, G. Yin, Z. Cai, Y. Dong, H. Dong, A trust-based probabilistic recommendation model for social networks, J. Netw. Comput. Appl. 55 (2015) 59–67.
[2] Y. Lin, Z. Cai, X. Wang, F. Hao, Incentive mechanisms for crowdblocking rumors in mobile social networks, IEEE Trans. Veh. Technol. 68 (9) (2019) 9220–9232.
[3] Y. Wang, Z. Cai, Z.-H. Zhan, Y.-J. Gong, X. Tong, An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing, IEEE Trans. Comput. Soc. Syst. (2019).
[4] J. Manotumruksa, C. Macdonald, I. Ounis, A contextual attention recurrent architecture for context-aware venue recommendation, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2018, pp. 555–564.
[5] F. Hao, S. Li, G. Min, H.-C. Kim, S.S. Yau, L.T. Yang, An efficient approach to generating location-sensitive recommendations in ad-hoc social network environments, IEEE Trans. Serv. Comput. 8 (3) (2015) 520–533.
[6] R. Gao, J. Li, B. Du, X. Li, J. Chang, C. Song, D. Liu, Exploiting geo-social correlations to improve pairwise ranking for point-of-interest recommendation, China Commun. 15 (7) (2018) 180–201.
[7] F. Hao, Z. Pei, D.-S. Park, V. Phonexay, H.-S. Seo, Mobile cloud services recommendation: a soft set-based approach, J. Ambient Intell. Humaniz. Comput. 9 (4) (2018) 1235–1243.
[8] Z. Yu, H. Xu, Z. Yang, B. Guo, Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints, IEEE Trans. Hum.-Mach. Syst. 46 (1) (2016) 151–158.
[9] J. He, X. Li, L. Liao, D. Song, W.K. Cheung, Inferring a personalized next point-of-interest recommendation model with latent behavior patterns, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.
[10] Y. Yu, X. Chen, A survey of point-of-interest recommendation in location-based social networks, in: Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[11] H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, ACM, 2011, pp. 287–296.
[12] Q. Liu, S. Wu, L. Wang, T. Tan, Predicting the next location: A recurrent model with spatial and temporal contexts, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.
[13] C. Cheng, H. Yang, I. King, M.R. Lyu, Fused matrix factorization with geographical and social influence in location-based social networks, in: Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.
[14] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, Y. Rui, GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 831–840.
[15] C. Ma, Y. Zhang, Q. Wang, X. Liu, Point-of-interest recommendation: Exploiting self-attentive autoencoders with neighbor-aware influence, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM, 2018, pp. 697–706.
[16] H. Gao, J. Tang, H. Liu, gSCorr: modeling geo-social correlations for new check-ins on location-based social networks, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, 2012, pp. 1582–1586.
[17] X. Li, G. Cong, X.-L. Li, T.-A.N. Pham, S. Krishnaswamy, Rank-geofm: A ranking based geographical factorization method for point of interest recommendation, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 433–442.
[18] G. Pang, X. Wang, F. Hao, J. Xie, X. Wang, Y. Lin, X. Qin, ACNN-FM: A novel recommender with attention-based convolutional neural network and factorization machines, Knowl.-Based Syst. 181 (2019) 1–13.
[19] D.Y. Zhang, D. Wang, H. Zheng, X. Mu, Q. Li, Y. Zhang, Large-scale point-of-interest category prediction using natural language processing models, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 1027–1032.
[20] B. Chang, Y. Park, S. Kim, J. Kang, DeepPIM: A deep neural point-of-interest imputation model, Inform. Sci. 465 (2018) 61–71.
[21] Q. Yuan, G. Cong, A. Sun, Graph-based point-of-interest recommendation with geographical and temporal influences, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, 2014, pp. 659–668.
[22] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.
[23] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, D. Estrin, Collaborative metric learning, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 193–201.
[24] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M.J. Chadwick, T. Degris, J. Modayil, et al., Vector-based navigation using grid-like representations in artificial agents, Nature 557 (7705) (2018) 429–433.
[25] J. Xie, X. Wang, X. Wang, G. Pang, X. Qin, An eye-tracking attention based model for abstractive text headline, Cogn. Syst. Res. 58 (2019) 253–264.
[26] R. Gao, J. Li, X. Li, C. Song, Y. Zhou, A personalized point-of-interest recommendation model via fusion of geo-social information, Neurocomputing 273 (2018) 159–170.
[27] J. Zeng, X. He, Y. Li, J. Wen, W. Zhou, A point-of-interest recommendation method using user similarity, in: Web Intelligence, Vol. 16, IOS Press, 2018, pp. 105–112.
[28] X. Wang, X. Wang, F. Hao, G. Min, L. Wang, Efficient coupling diffusion of positive and negative information in online social networks, IEEE Trans. Netw. Serv. Manag. (2019) http://dx.doi.org/10.1109/TNSM.2019.2917512.
[29] Y. Lin, X. Wang, F. Hao, Y. Jiang, Y. Wu, G. Min, D. He, S. Zhu, W. Zhao, Dynamic control of fraud information spreading in mobile social networks, IEEE Trans. Syst. Man Cybern.: Syst. (2019).
[30] Z. Cai, X. Zheng, A private and efficient mechanism for data uploading in smart cyber-physical systems, IEEE Trans. Netw. Sci. Eng. (2018).
[31] Y. Wang, Z. Cai, X. Tong, Y. Gao, G. Yin, Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems, Comput. Netw. 135 (2018) 32–43.
[32] M. Ye, P. Yin, W.-C. Lee, D.-L. Lee, Exploiting geographical influence for collaborative point-of-interest recommendation, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2011, pp. 325–334.
[33] Y. Si, F. Zhang, W. Liu, CTF-ARA: An adaptive method for POI recommendation based on check-in and temporal features, Knowl.-Based Syst. 128 (2017) 59–70.
[34] H. Li, Y. Ge, R. Hong, H. Zhu, Point-of-interest recommendations: Learning potential check-ins from friends, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 975–984.
[35] S. Feng, G. Cong, B. An, Y.M. Chee, Poi2vec: Geographical latent representation for predicting future visitors, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[36] Z. Chen, W. Li, Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network, IEEE Trans. Instrum. Meas. 66 (7) (2017) 1693–1702.
[37] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin, Convolutional sequence to sequence learning, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 1243–1252.
[38] X. Kong, C. Li, F. Zheng, C. Wang, Improved deep belief network for short-term load forecasting considering demand-side management, IEEE Trans. Power Syst. 35 (2) (2019) 1531–1538.
[39] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: A search space odyssey, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2017) 2222–2232.

[40] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, J.-Y. Nie, A hierarchical recurrent encoder-decoder for generative context-aware query suggestion, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 553–562.

[41] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[43] S.H. Park, B. Kim, C.M. Kang, C.C. Chung, J.W. Choi, Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 1672–1678.

[44] I. Yahav, O. Shehory, D. Schwartz, Comments mining with TF-IDF: The inherent bias and its removal, IEEE Trans. Knowl. Data Eng. 31 (3) (2019) 437–450.

[45] G. Lee, Y.-W. Tai, J. Kim, ELD-net: An efficient deep learning architecture for accurate saliency detection, IEEE Trans. Pattern Anal. Mach. Intell. 40 (7) (2018) 1599–1610.

[46] H. Gao, J. Tang, X. Hu, H. Liu, Exploring temporal effects for location recommendation on location-based social networks, in: Proceedings of the 7th ACM Conference on Recommender Systems, ACM, 2013, pp. 93–100.

[47] X. Li, T.-A.N. Pham, G. Cong, Q. Yuan, X.-L. Li, S. Krishnaswamy, Where you instagram?: Associating your instagram photos with points of interest, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 1231–1240.

[48] Y. Liu, T.-A.N. Pham, G. Cong, Q. Yuan, An experimental evaluation of point-of-interest recommendation in location-based social networks, Proc. VLDB Endow. 10 (10) (2017) 1010–1021.

[49] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[50] S.N. Tran, A.S.d. Garcez, Deep logic networks: Inserting and extracting knowledge from deep belief networks, IEEE Trans. Neural Netw. Learn. Syst. 29 (2) (2016) 246–258.

[51] J. Feng, J. Chen, L. Liu, X. Cao, X. Zhang, L. Jiao, T. Yu, CNN-based multilayer spatial–spectral feature fusion and sample augmentation with local and nonlocal constraints for hyperspectral image classification, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 12 (4) (2019) 1299–1313.