The Point Of Interest (POI) Recommendation for Mobile Digital Culture Heritage (M-DCH) Based on the Behavior Analysis Using the Recurrent Neural Networks (RNN) and User-Collaborative Filtering

Chung-Ming Huang, Chen-Yi Wu

Dept. of Computer Science and Information Engineering, National Cheng Kung University, Taiwan
huangcm@locust.csie.ncku.edu.tw, wucy@locust.csie.ncku.edu.tw

Abstract

Many Point Of Interest (POI) recommendation systems need to collect past users' scoring to generate the recommendation for current users. It always results in the not so precise recommendation because not a lot of users are willing to do the scoring. With the advanced deep learning technique, this work proposes the POIs' recommendation method that doesn't require a scoring mechanism to have the great precision, recall and diversity. The proposed POIs' recommendation method utilizes the deep learning model to analyze user's operational behaviors and then judge the user's preference. As a result, the proposed POI's recommendation method (i) can be built in an environment without a scoring mechanism because it can catch the user's preferences by analyzing his operational behaviors and (ii) considers similar users' historical data to make the recommended results more diversity. The performance evaluation shown that the precision, recall, f1-score and the next POI predicted rate of the proposed method is better than that of the Multi-Layer Perceptrons (MLPs) and the Long Short-Term Memory (LSTM) models. The diversities of the proposed method's results are better than that of the LSTM model. Therefore, the proposed method balances the precision, recall and diversities.

Keywords: Deep learning, Recommendation system, Collaborative filtering, Recurrent Neural Networks (RNN), Point Of Interest (POI)

1 Introduction

The recommendation system is devised to make one find information more quickly and more precisely [1-2]. The recommendation system brings a lot of convenience to many fields, such as the recommendation of movies, music, merchandises, Point of Interests (POIs) for touring, etc. For the POI recommendation system, it can help people quickly and more precisely to find the places in which they are interested and solve the problems for those people who often worry about where to travel. In recent years, with the continuous advancement of the wireless mobile IT technology, the recommendation system has evolved from the traditional rule-based method to the machine learning-based method, and then to the deep learning-based method [3-5]. The design of a deep learning model often requires experienced accumulation and compliance with current conditions. The deep learning models tend to differ depending on the developed ideas and the data. Thus, the deep network models that were designed by different people are always different. Similarly, due to differences in data and requirements, the deep learning model of the recommendation system should have different architectures.

There are three ways to implement a recommendation system: (1) Content-based filtering recommendations [6], (2) Collaborative filtering recommendation [7] and, (3) Hybrid recommendation [8-9]. The content-based filtering method uses the attributes of the item and the attributes of the user's preference to recommend items to the user. The recommended objects of the content-based filtering are usually similar to the ones that a user is interested in the past. The collaborative filtering method is the recommended method that uses the similarity function to get similar objects of the target object and then recommends these objects to users. The hybrid method is based on combining several recommended methods to come out a new recommendation system.

This work concentrates on the recommendation system for tourism, for which AI's deep learning technique is adopted to make users to get what they are interested more conveniently and precisely. The target application is the DEH platform (http://deh.csie.ncku.edu.tw) [10], which was developed by our lab. The DEH platform is an open platform that (1) allows users to create Point Of Interests (POIs) of Mobile Digital Culture Heritage (M-DCH) and then upload these POI's contents to the DEH platform and (2) allows

*Corresponding Author: Chung-Ming Huang; E-mail: huangcm@locust.csie.ncku.edu.tw
DOI: 10.53106/160792642021072204010

users to download POIs' contents, based on the user's current location or the location that the user specifies in the handled device, which is done using the Google map. The goal of this work is to design and develop a recommendation system for DEH users. Since the system doesn't have the scoring mechanism, it needs to analyze users' behaviors to get the user's preference of POIs. It can deliver those POIs' contents that a user may be interested based on his past record and other DEH users' past records, which is done through users' behaviors based on the characteristics of DEH's metadata.

Our recommending function mainly uses the Recurrent Neural Network (RNN) [11] and the self-attention mechanisms [12]. RNN is adopted to let machine analyze the time series data and the self-attention mechanism that can assist RNN to let the weights of important attributes be higher so that the model can focus on important POIs with which to predict. Two layers of the proposed model are (1) the Bi-directional Long Short-Term Memory (Bi-LSTM) [13] layer and (2) the Bi-LSTM with the self-attention layer. The former layer analyzes the behaviors for each POI, which are historical data of the users. The later layer analyzes the output from the former layer, which contains series of POIs. Additionally, the collaborative filtering-like method is used to get the POIs from other users, whose behaviors are similar to the target user. Then, it combines the POIs derived from the deep learning model and the collaborative filtering-like method to get the final recommended results. The scenario is shown in Figure 1.

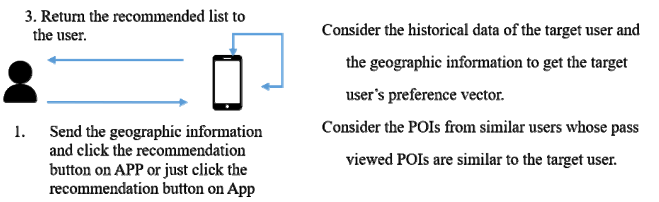


Figure 1. The APP's usage scenario of the proposed method

The rest of this paper is organized as follows. Section 2 introduces the overview of the DEH platform. Session 3 introduces related research works. Section 4 presents the architecture and the main technical issues of the recommending sub-system. Section 5 describes the input data preprocessing the sub-system. Section 6 describes the proposed recommending function that can be used in the DEH platform. Section 7 presents the performance evaluation for the recommending sub-system. Finally, Section 8 has the conclusion remarks.

2 The Overview of The DEH Platform

The target application of this work is the DEH platform (http://deh.csie.ncku.edu.tw), which was developed by our lab [10]. The main systems in the DEH platform are (1) the web system, which contains the exploring sub-system and the authoring sub-system, and (2) APPs, which contain the exploring APP and the authoring APP. Figure 2 shows the abstract architecture of the DEH platform.

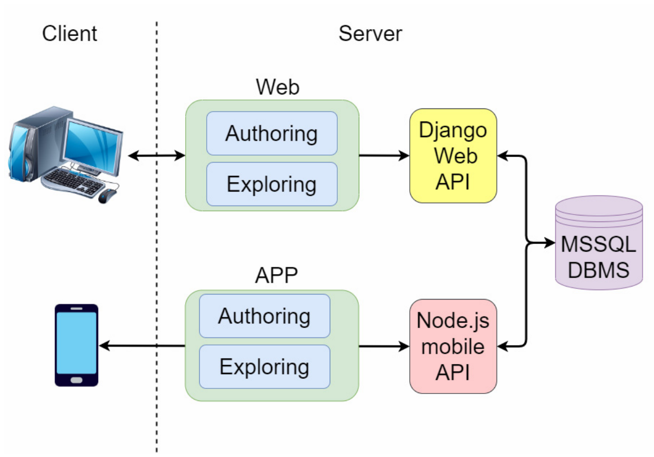


Figure 2. The abstract architecture of the DEH platform

This work aims to enhance the exploring mechanism, which is primarily used to provide POIs' contents to users. DEH's POIs have many attributes, which are "Region", "Subject", "Category", "Type", "Format" and "Media Type". These attributes can be seen on the filtering interface of DEH's Web system. Figure 3 shows the filtering interface in DEH's Web system.

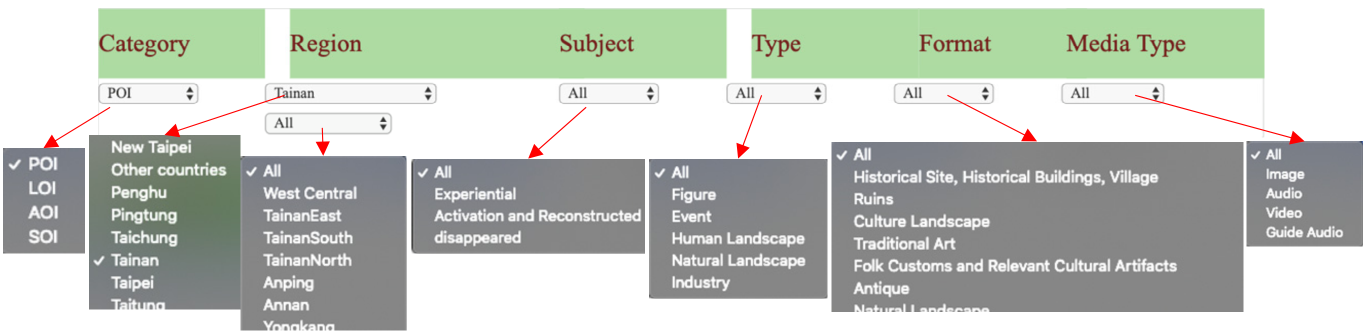
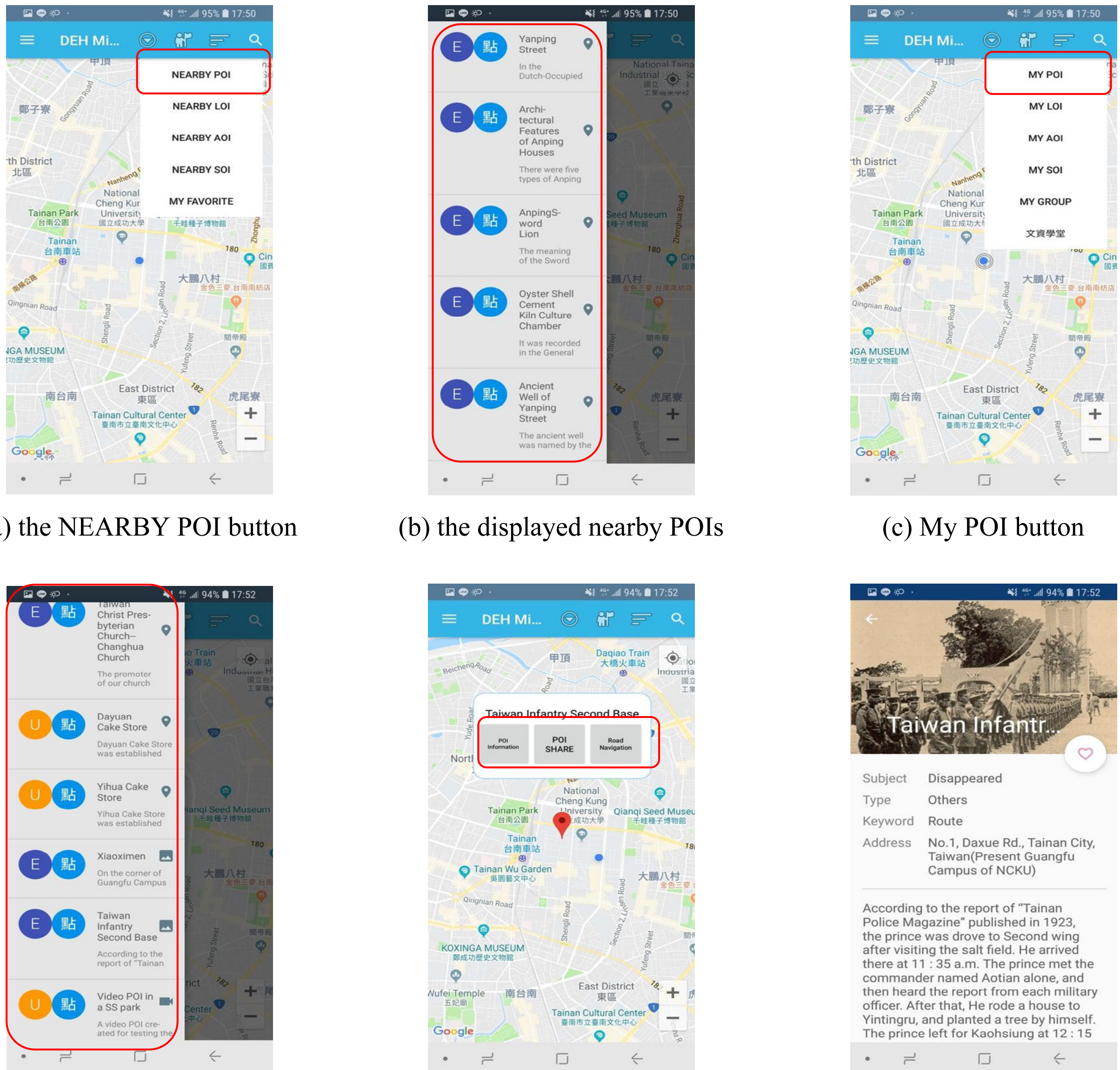


Figure 3. The filtering interface in DEH's Web system

The recommendation system was mainly implemented on the DEH's exploring APP, i.e., DEH Mini II. Figure 4 shows illustrated examples of using DEH's exploring APP. Referring to Figure 4(a), a user can click the button at the far right of the toolbar to see an option called "nearby POIs". When the user clicks "nearby POI", DEH Mini II can output the neighboring k POIs' names in the n-kilometer's range of the user's current location or the location that the user pinpoints in the Google Map of DEH Mini II: (1) If the number of nearby POIs is greater than k in the n kilometers, then the nearest k POIs are selected and listed; (2) if the number of nearby POIs is equal to or smaller than k in the n kilometers, then all of these k POIs are selected and listed. Then, the user can view these nearby POIs' names that were attributed as public and have been successfully verified. Figure 4(b) shows an example. The other option is to search those POIs that were made by the user himself/herself. Referring to Figure 4(c), if the user U logs in the DEH platform with U's registered account and password, then U can click the third button on the toolbar and then select "my POIs"; after that, DEH Mini II displays those POIs' names that were made by U and satisfy the filtering criteria that U set, for which Figure 4(d) shows an example. If the user is interested in the content of a displayed POI, then he can click the POI, and then he can see three buttons, which are the "POI information" button, the "share" button and the "navigation" button, and POI's location in Google Map, for which Figure 4(e) shows an example. If the user wants to see the POI's information, he/she can click the "POI information" button, and then he/she can see the POI's content. Figure 4(f) shows an illustrated example.



(a) the NEARBY POI button (b) the displayed nearby POIs (c) My POI button

(d) the displayed POIs that were authored by the user (e) the POI information button, navigation button, share button (f) the POI's description

Figure 4. Illustrated examples of using DEH Mini II

3 Related Work

This Session presents related works, including POI recommendation systems and the collaborative filtering.

3.1 POI Recommendation Systems

In [14], the authors proposed a method of POI recommendation with the matrix factorization. At first, it used a Latent Dirichlet Allocation (LDA) model to catch the topic information of each POI. Then, it used user's comments to generate POI's probability distribution. Next, it divided each user's check-in data into several slices. The proposed method presented the tensor to represent all users' preferences based on connecting the probability distribution of each user's viewed POIs. Finally, it used the higher-order singular value decomposition (HOSVD) algorithm to decompose the matrix, which contains the features of the user, POIs and the time, to get the preference vector for each user. The results of the performance shown that the proposed method can let the accuracy of the POI recommendation be better.

In [15], the authors proposed a POI recommendation method, which contains the check-in model and the other auxiliary information model. The proposed method used the one multinomial distribution to model the preference distribution of all users and one multinomial distribution to model the impact of other auxiliary information for the user. Then, it used the check-in model to build a unified model. This method had the good precision and recall for the Foursquare and Gowalla data sets.

The difference between the two aforementioned methods and our work is that these two works mainly use the machine learning algorithm for recommending, but our work concentrates on using the deep learning model to recommend POIs.

In [16], the authors a proposed POI recommendation framework, which uses the simple deep learning model. The proposed model considered the LBSNs' features and learned the influence on user behaviors. The model also considered the categorical, geographic and co-visiting impact to mitigate the sparse data issue. The inputs of the proposed model are two vectors, which are a user feature vector and a POI feature vector. The results of the performance shown that the proposed method has great results in precision and recall.

The difference between the aforementioned methods and our work is that these three methods have no attention mechanism to adjust their model, but our proposed model has.

In [17], the authors proposed an autoencoder-like model. The encoder part used the multi-head self-attention mechanism to let the machine analyze more accurately. The decoder part considered the geographical information and the results from the encoder to get the POI recommendation results. The decoder used the inner product of POI vectors to implement. The authors used three real datasets to evaluate the proposed model. The results of the performance shown that the proposed model is effective.

The difference between the aforementioned method and our work is that our work mainly considers the series of the operational behaviors and the series of the viewed POIs, but the aforementioned method didn't consider the sequential feature. That is, our work uses the Bi-LSTM with the self-attention but the aforementioned method uses just the self-attention.

3.2 Collaborative Filtering

In [18], the authors proposed a general framework called Neural network-based Collaborative Filtering (NFC). NCF is generic and can be used to express and generalize matrix decomposition. To use the nonlinear enhanced NCF modeling, the authors recommended using a multi-layer perceptron, which is simple but effective, to learn user-item interaction capabilities. Empirical evidence suggests that using deeper neural networks can provide better recommendations but each application may adopt a different architecture of the NCF.

In [19], the authors proposed a neural model for CF, named User-Item co-Autoregressive model (CF-UIcA), which considers the autoregression in the domains of both users and items, so as to build the model for User-User Correlations (UUCs) and Item-Item Correlations (IICs). The co-autoregression allows extra desired properties to be incorporated for different tasks. Furthermore, the authors also developed an efficient stochastic learning algorithm to handle large scale datasets. Two popular benchmarks that are used to evaluate the proposed method are (i) MovieLens 1M and (ii) Netflix. The proposed method shown that it has good Hit Ratio (HR) and Normalized Discounted Cummulative Gain (NDCG).

The difference between the aforementioned two methods and our work is that the aforementioned two methods concentrated on collaborative filtering, but our method also considered individual information. Collaborative filtering (CF) often makes recommendations more complete and rich. For the POI recommendation issue, it is usually not possible to only use collaborative filtering. For our work, we use collaborative filtering as an additional function to make the result more diversity.

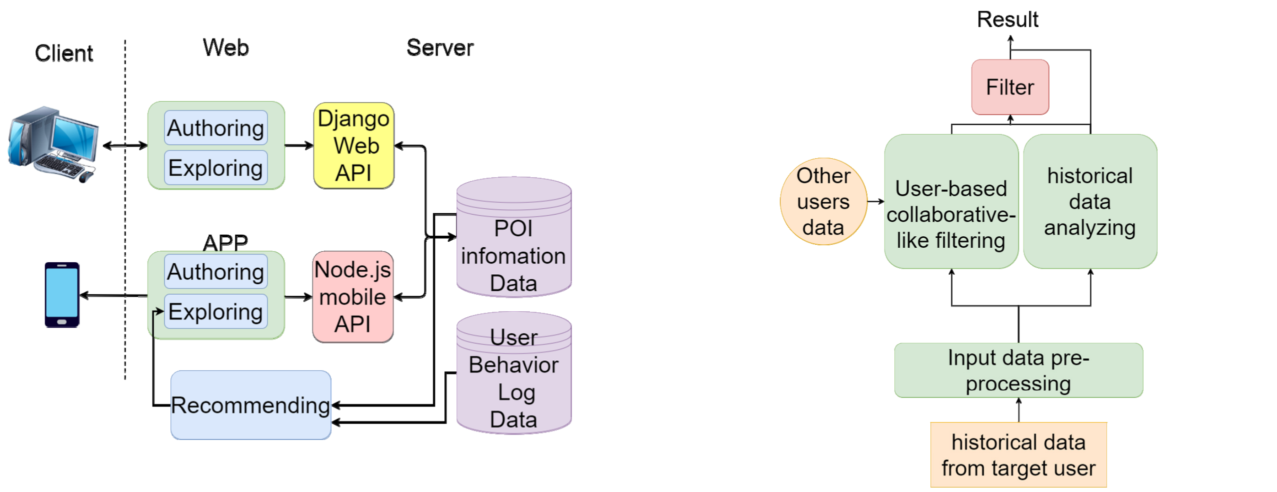
4 System Architecture And Main Technical Issues

Two main concerns for having the recommendation mechanism in the DEH platform are as follows: (1) in the DEH platform, users can't see others' evaluation of

POIs because there is no mechanism for users' scoring. Thus, it is difficult for users to have a criterion of distinguishing the preference of POIs. (2) The recommendation function often refers to the historical data of the target user and then makes the final decision, which often limits the diversity of the recommended items. For the first concern, since there is no scoring mechanism, this work proposed a mechanism called historical data analysis mechanism to judge the user's preference, which is done by analyzing each user's operational behaviors for POIs to get the POIs in which the target user may also be interested. For the second concern, this works proposed a mechanism called the user-collaborative filtering, which considers those users whose historic preferences on POIs have the similar characteristics as the similar users, and then refer to the historical data of these similar users to get the POIs in which the target user may also be interested.

This work needs two data tables, which are named as the POI information data table and the user behavior log data table, to store POI contents and user operational behaviors' records respectively. The POI contents table contains all of the POI's attributes, such as POI's ID, POI's title, POI's description, and POI's subject, etc. The user behavior log data table contains all of the user behaviors' records on each POI using DEH Mini II. The behavior includes "using a filter", "clicking the POI" and "clicking the Share button", etc. Figure 5(a) shows the abstract architecture of the enhanced DEH platform with the proposed recommendation function.

The overview of the recommending sub-system is depicted in Figure 5(b). At first, the historical data of the target user is transformed to vectors using autoencoder, word embedding and one-hot encoding in the input data pre-processing phase. Then, the historical data vectors of the target user, which are generated in the input data pre-processing phase, is taken as the input of the deep network model of the historical data analyzing to analyze and generate the POIs in which the target user may be interested. The POIs, which are generated from the historical data analysis module, have similar characteristics to the historical data of the target user. The preference vector of the target user, which is generated from the input data pre-processing phase, is used as the input of the user-collaborative filtering method. Calculating the similarity of the preference vector of the target user and those of the other system users, it can get the similar users with high similarity. Then, referring to the historical data of similar users to analyze and get the POIs in which the target users are most likely to be interested. After that, combining the results of the historical data analyzing module and the user-collaborative filtering method to get at the final result.



(a) The abstract architecture of the enhanced DEH platform with the proposed recommendation function

(b) The overview of the recommending sub-system

Figure 5.

5 The Input Data Pre-Processing

In the input data pre-processing phase, the main work is to transform historical data of target users, which includes historically viewed POIs, operational behaviors, etc., into vectors. The methods of transforming to vectors that this work uses are (1) one-hot encoding, (2) autoencoder [20] and (3) word embedding [21].

5.1 One-hot Encoding

Since there are many attributes that need to be digitized in the proposed method, three vectors, called (1) POI content vector, (2) POI ID vector, and (3) POI behavior vector, which can be transformed with one-hot encoding, are defined. The POI content vector is used to represent a POI's obvious characteristics, which have been defined in the DEH platform. The POI ID vector is used to distinguish all POIs. The POI behavior vector is used to represent the user's possible/allowed behaviors on each POI.

The POI content vector's dimension is 44 and is divided into five parts, (1) Region, (2) Subject, (3) Type, (4) Format and (5) Media Type. The Region vector contains 23 elements, which includes 22 administrative districts in Taiwan and 1 object, which represents other countries. The size of the Subject vector, Type vector, Format vector and Media Type vector are defined in the DEH platform. The Subject vector contains 3 elements. The Type vector contains 5 elements. The Format vector contains 9 elements. The Media Type vector contains 4 elements. Then, it can get the POI content vector by combining a Region vector, a Subject vector, a Type vector, a Format vector and a Media Type vector.

There are 20472 POIs in the DEH Platform, and each POI has its own ID. Thus, the POI ID vector contain 20472 elements. Using the one-hot encoding method to let each POI's ID be transformed to a vector. Then, combining the POI ID vector and the POI content vector to get the POI static one hot vector, which expresses the POI static attributes.

There are 6 types of behaviors that users are allowed to execute in the DEH platform. Thus, these 6 types of behaviors become the 6 elements of the behavior vector. These 6 elements of the behavior vector are as follows: (1) using a filter, (2) clicking the POI, (3) clicking the button to get the video and audio multimedia content of the POI, (4) clicking the button to listen to the guided speech's explanation, (5) clicking the "navigation" button and (6) clicking the "share" button.

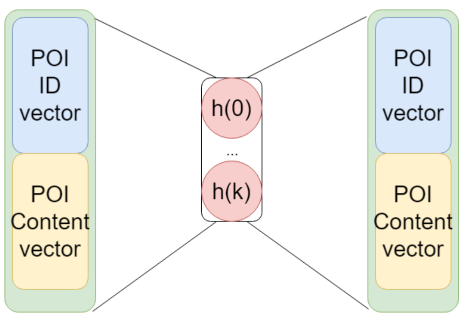
5.2 Autoencoder

Since using one-hot code to represent vectors is too wasteful, autoencoder is used to solve this problem. In order to improve the training efficiency, an autoencoder can be adopted to transform sparse vectors to dense vectors. As a result, it reduces the amount of parameters of the historical data analysis model. In our autoencoder, both the input and the output are the POI static one hot vectors. The input size and the output size is 20516, because there are 20472 elements in the POI ID vector and 44 elements in the POI content vector. The autoencoder is mainly used to reduce the input vector of the proposed model such that the training efficiency of the proposed method is better. The hidden layer's size of our autoencoder is set to 32, that is, using 32 elements to represent the information of POI's ID and the obvious characteristics of POIs. Figure 6(a) shows the autoencoder for our data preprocessing.

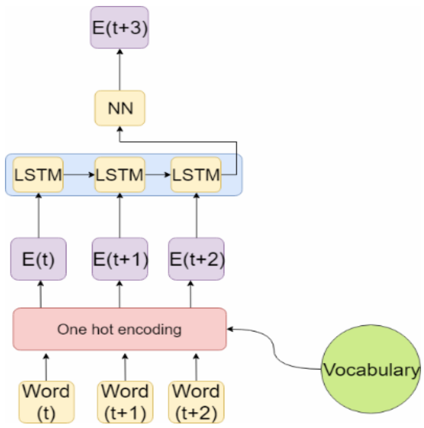
The training concept of the autoencoder is to compress the sparse vector to the dense vector, then it is still able to restore back to the sparse vector. The dense vector can be represented as the important characteristics of the sparse vector. Our autoencoder's training parameters are as follows. It sets the learning rate to 0.005, total epochs to 12 and the batch size to 64. It uses the mean square error (MSE) as the loss function and the stochastic gradient descent (SGD) as the optimizer.

5.3 Word Embedding

In order to refer to the POIs' descriptions, it uses the word embedding model to transform all words of POIs' description to vectors. Using the concept of n-gram, which tries to use the previous n-1 words (w(1), w(2), …, w(n-1)) to predict the next word w(n), through the model training, it can make the difference between the predicted result and the word w(n) as small as possible. The hidden size of our word embedding model is set to 64 and the n of the n-gram is set to 4. Figure 6(b) shows the architecture of the proposed n-gram model.



(a) The architecture of the autoencoder for transforming the combination of the POI ID vector and the POI content vector to the POI static vector



(b) The architecture of the proposed n-gram model

Figure 6.

The training concept is to compress the sparse vector to the dense vector, and let the dense vector get the relation between word and word. The training parameters of our word embedding model is as follows. It sets the learning rate to 0.003, total epochs to 30 and the batch size to 128. It uses the cross-entropy loss as the loss function and the stochastic gradient descent (SGD) as the optimizer.

It needs to transform the POI's description, which is composed of several words, to a vector. Therefore, it sums up all word vectors in POI's description and gets the average, and then gets the vector, named POI description vector, to represent the POI's description.

6 The Recommending Function

The proposed recommendation function is mainly composed of two parts: (1) the historical data analyzing and (2) the user-collaborative filtering. The final recommended POIs are derived based on both results from the historical data analyzing and the user-collaborative filtering.

6.1 The Historical Data Analyzing

Since the DEH platform does not have a scoring mechanism, it can't intuitively get the preference of each user's viewed POI. In this situation, it is expected to judge the user's preference by analyzing the operational behaviors of her/his viewed POIs. When the user has more operational behaviors than the basic operation behavior, which just includes the "Click the POI" behavior for the POI, it is assumed that the user is interested in the POI. In other words, if the user only has the basic operation on a POI, then it is assumed that the user isn't interested in the corresponding POI.

The user may have more than one type of behavior for each POI and the order of the behaviors also represents different meaning. For example, clicking the "share button" after clicking the "navigation button" is different from clicking the "navigation button" after clicking the "share button". The former case may represent that the user has arrived at the destination and feels that the POI is great, and then clicks the "share button" to share the POI. The later case may represent that the user has not arrived at the destination yet, just shares the POI.

Figure 7 shows the architecture of the deep learning model for the historical data analyzing. In order to consider the time series of the behaviors, it uses the Bi-LSTM to analyze the operational behaviors of each viewed POI. That is, it uses the series of POI behavior vectors of each viewed POI for BI-LSTM's input and getting the POI behavior abstract vector to represent the summary of the behaviors for each viewed POI. Next, in order to get the vector that represents the POI features, it can combine the POI static vector, which is from the autoencoder's embedding table, and the POI description vector to get the POI feature vector, which contains all of the POI's static features. Then, it combines the POI feature vector and POI behavior abstract vector to get the POI information vector. Obviously, even if users have viewed the same POI, since their operational behaviors are different, there will be different POI information vectors. Finally, since different orders of viewed POIs should get different results from the proposed module, i.e., the proposed module needs to consider the order of the viewed POIs, it uses the Bi-LSTM, which is the neural network with the time series characteristic, to consider the orders of the viewed POIs. However, since each viewed POI may have different influence for the current recommendation, the proposed module adds the self-attention mechanism on Bi-LSTM. That is, the proposed module not only considers the order of the viewed POIs but also the influence for each viewed POI.

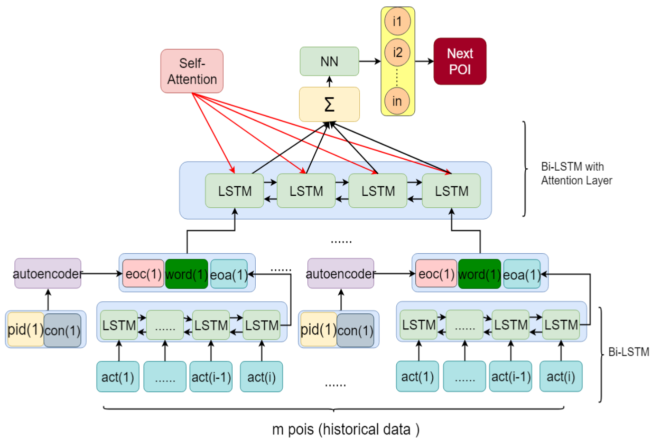


Figure 7. The architecture of the historical data analyzing

How to let the model know in which POIs the user is interested? Indeed, since the current model only produces a summary of the operational behaviors, the model doesn't know which viewed POIs are positive or negative for the user. Thus, it needs to filter the positive POIs, which denotes the one in which the user is interested, and the negative POIs, which denotes the one in which the user may not be interested, with the behavior labeling. The operational behavior labeling the viewed POIs that have the behaviors of "click the button to get the video or audio multimedia content of the POI", "click the button to listen to the guided speech's explanation", "click the navigation button" or "click the share button" to be positive POIs; others they are labeled as the negative POIs.

In order to let the model learn how to distinguish the positive POIs and the negative POIs, some designs were made for the loss function, which is the evaluation function for the model. The two loss functions that are used in the proposed model are (1) Bayesian Personalized Ranking (BPR) and (2) cross-entropy loss. The Bayesian Personalized Ranking is mainly used to calculate the distance between the predicted result and the positive sample $\hat{r}_{s,i}$ and the distance between the predicted result and the negative sample $\hat{r}_{s,j}$. Then, the total distance divides by N_s, which is the total pair of the $\hat{r}_{s,i}$ and $\hat{r}_{s,j}$. Using this loss function for training the model can let the difference between $\hat{r}_{s,i}$ and $\hat{r}_{s,j}$ larger and larger and then the model can learn which viewed POIs have the higher degree of influence for the current recommendation. The formula of Bayesian Personalized Ranking is as follows:

$$L_s = \frac{1}{N_s} \cdot \sum_{j=1}^{N_s} \log(\sigma(\hat{r}_{s,i} - \hat{r}_{s,j})) \quad \textbf{(1)}$$

The training method of the proposed model is to take m viewed POIs as input and use the next viewed POI for a target. It is just like inputting m features and outputting an item's class. Since the concept of the classification for our training is used, the cross-entropy loss, which is commonly used to quantify the difference between two probability distributions y and p, is used to tackle the classification problem. The formula of cross-entropy loss is as follows:

$$H = \sum_{c=1}^{C} \sum_{i=1}^{n} -y_{c,i} \log_2(p_{i,j}) \quad \textbf{(2)}$$

The weights of cross-entropy loss H and BPR loss L_s, α and β, are the important parameters for the loss function. After many attempts, it is found that α = 0.3 and β = 0.65 have the good hit rate for the proposed model. Thus, the loss function of the proposed model is as follows:

$$C = 0.3 * H + 0.65 * L_s$$

The training parameters of our proposed model are as follows. It sets the total epochs to 10, the learning rate to 0.0001 for previous 3 epochs and 0.00003 for last 7 epochs and the batch size to 64. It uses the cross-entropy loss and bayesian personalized ranking (BPR) as the loss function and the stochastic gradient descent (SGD) as the optimizer.

6.2 The User-Collaborative Filtering

Since the historical data analyzing only considers the viewed POIs for the target user, the result from the historical data analyzing model is very likely to have high similarity of the viewed POIs that the target user viewed in the past. In order to make the recommendation more diverse, it refers to the viewed POIs of similar users, who have the similar operational behaviors to the target user.

Since the number of POIs in the DEH system is much more than the number of users, using the user-based collaborative filtering is better than item-based collaborative filtering. The result of the user-based collaborative filtering is more diverse than that of the item-based collaborative filtering because item-based collaborative filtering only considers items that are similar to the historical ones of the target user. Figure 8 shows the architecture of the adopted user-collaborative filtering.

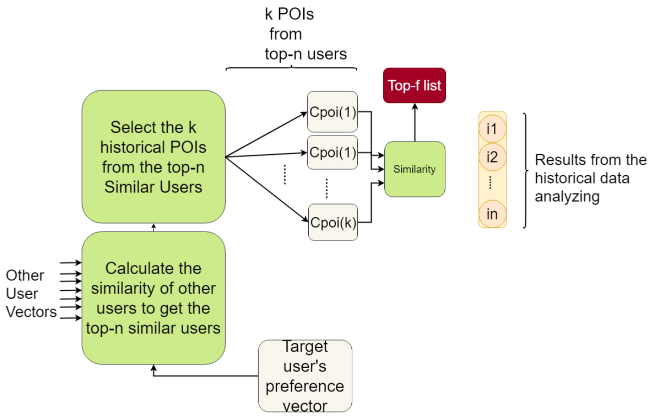


Figure 8. The architecture of the user-collaborative filtering

At First, calculating the preference vector for each user by summing up all of her/his own viewed POIs' POI static vectors and then getting the average in the offline. Next, it uses the cosine similarity function, which can get the similarity of two vectors, to get the top n users as the similar users, whose preference vectors have the high cosine similarity with the target user's preference vector. Finally, saving the result to the database and updating the database at 00:00 every day. The cosine similarity is as follows:

$$s(x,y) = \frac{\sum_{i=1}^{n} A_i \times B_i}{(\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2})} \quad \textbf{(3)}$$

In the runtime, the target user's user ID will be sent to the server. Then, using the user ID to search n similar users in the database. Next, get the d viewed POIs of each similar user and the result vector of the historical data analyzing. The selection of these d POIs has the following two restrictions: (1) it is not in the target user's historical data. (2) it has more than one operational behavior. Then, transforming these d viewed POIs to POI ID vectors. Next, calculating the cosine similarity between the POI ID vectors and the resulted vector, which is derived from the historical data analyzing, to get top k similar viewed POIs. These k similar viewed POIs have the high cosine similarity with the resulted vector that is derived from the historical data analyzing. Finally, recommend these k viewed POIs and the recommended POIs, which are from the historical data analyzing, to the target user.

7 Performance Evaluation

The proposed recommendation function is mainly composed of two parts: (1) the historical data analyzing and (2) the user-collaborative filtering. The final recommended POIs are derived based on both results from the historical data analyzing and the user-collaborative filtering.

7.1 Evaluation Environment and Method

The data set, which is stored in the DEH database, includes the users' operational behavior records and POIs' attributes. The total number of POIs are 20,472, users are 4,786 and the operational behavior records are 1,203,971. Among the 4786 users, 100 users are the testing users and 3,786 users are the training users. That is, it uses the training users' operational behavior records as input to train the model, and then uses the testing users' operational behavior records for evaluating the model. There are six evaluation items, which are precision, recall, F1-score, next POI predicted rate, POI-POI similarity and POI-User similarity.

It needs to define True Positives (TP), False Negatives (FN), False Positive (FP) and True Negative (TN) before defining precision, recall and f1-measure. Table 1 is used to explain these four terms. Take a binary classification for example, true positive means that the model predicts the object as belonging to "True" and it is actually "True". The false positive means that the model predicts the object as belonging to "True" but it is actually "False". The true negative means that the model predicts the object as belonging to "False" and it is actually "False". The false negative means that the model predicts the object as belonging to "False", but it is actually "True".

Table 1. The explanation of the True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN)

	Real True	Real False
Predict True	True Positive (TP)	False Positive (FP)
Predict False	False Negative (FN)	True Negative (TN)

The concept of the precision is how many objects that the model predicts "True", and they are actually "True". The concept of the recall is how many objects that are actually "True", and the model also predicts "True". The formula of the precision and recall are as follows:

$$\text{Precision} = \frac{TP}{(TP+FP)} \tag{4}$$

$$\text{Recall} = \frac{TP}{(TP+FN)} \tag{5}$$

In the top-k recommended problem, it supposes that u is the user, R(u) is the recommended list from the model for user u, and T(u) is u's testing data set. It defines that $|R(u) \cap T(u)|$ is the true positive. Then, it can get the formula of the precision and recall in the recommended problems as follows:

$$\text{Precision}_{\text{recom}} = \frac{\Sigma_{u \in U} |R(u) \cap T(u)|}{\Sigma_{u \in U} |R(u)|} \tag{6}$$

$$\text{Recall}_{recom} = \frac{\Sigma_{u \in U} |R(u) \cap T(u)|}{\Sigma_{u \in U} |T(u)|} \tag{7}$$

The F1 score is the harmonic mean of the precision and the recall. If the F1 value closes to 1, it represents that the recommendation system is perfect. If the F1 value closes to 0, it represents that the recommendation system is not good. The formula of the F1 score is as follows:

$$\text{F1} = \frac{2 \times \text{Precision}_{\text{recom}} \times \text{Recall}}{\text{Precision}_{\text{recom}} + \text{Recall}_{\text{recom}}} \tag{8}$$

A simple example is as follows. Let the number of products in the testing set for the three users be 10, 7 and 9, respectively. Among the top 5 recommended lists, the model has 2, 4, and 3 in the testing set. The value of the precision is (2 + 4 + 3) / (5 + 5 + 5) = 0.6. The value of the recall is (2 + 4 + 3) / (10 + 7 + 9) = 0.346. The value of the F1 score is (2×0.346×0.6)/ (0.346+0.6)= 0.4389.

The next POI predicted rate is the probability that the recommended POI list R(u) contains the next POI n(u) in the future. The formula of the next predicted rate is as follows:

$$\text{Next POI Predicted Rate } = \frac{1}{|U|} \times \Sigma_{u \in U} K(u) \tag{9}$$

$$K(u) = \begin{cases} 1, \text{ if } n(u) \subseteq R(u) \\ 0, \text{ otherwise} \end{cases}$$

In order to evaluate POI-User diversity, we first sum up the POI feature vectors for each user's viewed POIs, and then average them to get the diversity baseline of each user. Next, sum up the POI feature vectors of the recommended POIs and then get the average to get the diversity vector of each model. Finally, calculate the cosine similarity of baseline and the diversity vector of each model to compare the diversity. For the POI-POI diversity, the cosine similarity between POIs in the recommended POI list is calculated. Then, get the average of these cosine similarities to represent the model's diversity. Since these two diversities, i.e., the POI-POI similarity and the POI-User similarity, use the similarity for the diversity, the higher value means the less diversity.

The comparison objects are a two-layer Multi-Layer Perceptron (2-layer MLPs) model, a three-layer Multi-

Layer Perceptron (3-layer MLPs) model, a 2-layer Long Short-Term Memory (2-layer LSTM) model, and the proposed method without the user-collaborative mechanism. The first layer of each model uses the operational behaviors as input to get the POI behavior abstract vector. The second layer of each model uses the POI information vectors as input and gets the final results through the one or two layers.

7.2 Performance Evaluation Results

Since 2-layer MLPs, 3-layer MLPs, 2-layer LSTM and 2-layer Bi-LSTM with the attention mechanism just consider the historical data, they can use the results of the deep learning model directly. However, since our proposed method has two sets of results, in which one is from the historical data analyzing and the other one is from the user-collaborative filtering, we let the historical data analyzing provides 80% of the total recommended POIs and the remaining 20% is provided by the user-collaborative filtering.

Figure 9 shows the precision of these five methods. The x-axis denotes the total number of the recommended POIs and the y-axis denotes the precision values. The precision can indicate how many POIs in the recommended POI list match the POIs that the user will click on the APP in the future. It is observed that the precision of the proposed method is a little less than the precision of the 2-layer Bi-LSTM with the attention mechanism. It is because the proposed method has some recommended POIs from the user-collaborative filtering in order to increase the diversity and thus it must sacrifice some precision. Since the 2-layer Bi-LSTM with the attention mechanism just considers the individual data, all of its recommended POIs should match the user's naturally. Since the proposed method's deep learning model, which is the 2-layer Bi-LSTM with the attention mechanism, can accurately analyze the user's preference and then recommend the POI list, the proposed method sacrifices a few recommended contents to increase diversity. However, the precision of the proposed method is still better than other methods.

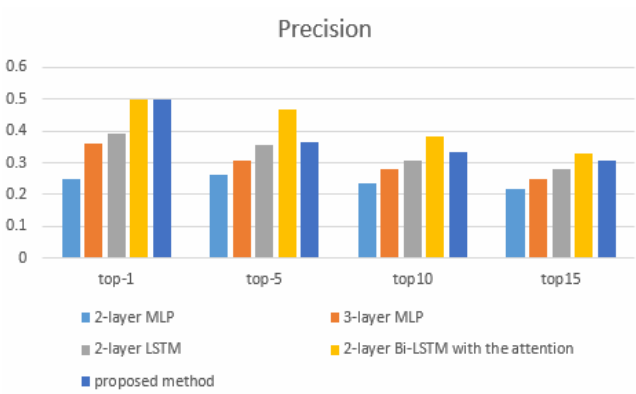


Figure 9. The precision of the 2-layer MLPs, the 3-layer MLPs, the 2-layer LSTM, the 2-layer Bi-LSTM with the self-attention mechanism and the proposed method

Figure 10 shows the comparison of the recall. The recall can indicate how many future POIs match the POIs in the recommended POI list. Since the proposed method has some recommended POIs from the user-collaborative filtering in order to increase diversity, of course, the recommended results would be slightly worse than that without the user-collaborative filtering. The proposed method's deep learning model considers the sequential features and focuses on POIs, which are important for the current recommendation. The proposed method's deep learning model analyzes individual data and achieves the greatest results. Therefore, the proposed method sacrifices a few recommended contents to increase diversity. However, the recall of the proposed method is still better than that of other methods.

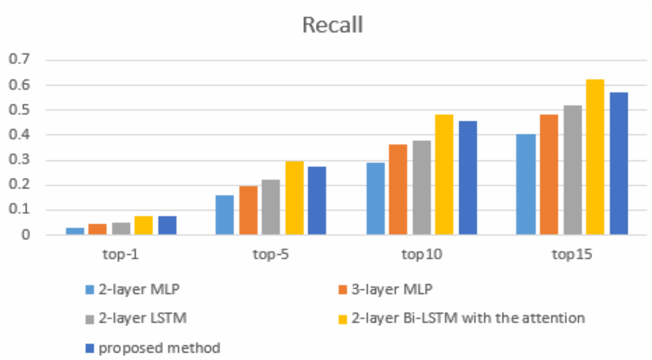


Figure 10. The recall of the 2-layer MLPs, the 3-layer MLPs, the 2-layer LSTM, the 2-layer Bi-LSTM with the self-attention mechanism and the proposed method

Figure 11 shows the comparison of the F1-score. The F1-score can indicate the summary of the precision and the recall. If it has a high recall or high precision, it will have high F1-score. Since the precision and recall of the proposed method's deep learning model is higher than that of other methods, the F1-score of the proposed method is also higher.

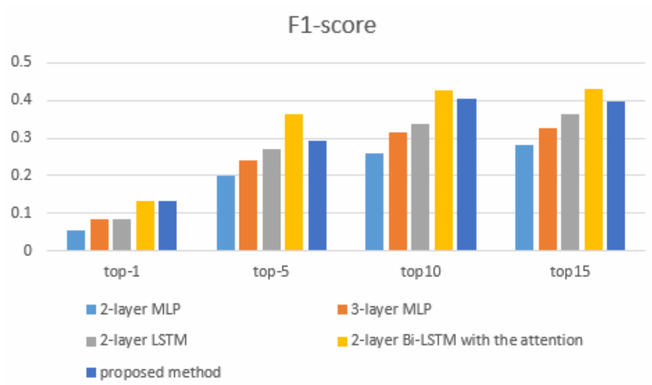


Figure 11. The F1-score of the 2-layer MLPs, the 3-layer MLPs, the 2-layer LSTM, the 2-layer Bi-LSTM with the self-attention mechanism and the proposed method

Figure 12 shows the next POI predicted rate. The next POI predicted rate shows how much probability that the recommended POI list contains the next POI. For the top-1 case, all of the methods don't have good results on the next POI predicted rate, and the proposed method, which has 2%, is higher than the others. For the situations of top-5 to top-15, it can be observed that the next POI predicted rate of each method grows rapidly. It means that all of these methods have gradually caught the user's preferences, but the degree of catching is different. The proposed method's deep learning model has the highest next POI predicted rate. Its next POI predicted rate is up to 65%. The proposed method, which sacrifices some next POI predicted rate, has 59%. The next POI predicted rate of the proposed method is still greater than the next POI predicted rates of 2-layer MLPs, 3-layer MLPs and 2-layer LSTM.

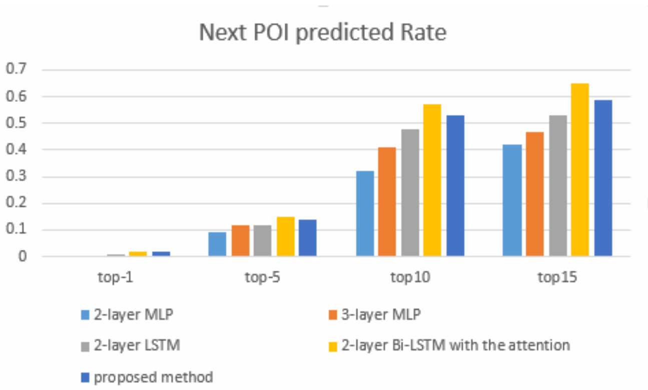


Figure 12. The next POI predicted rates of the 2-layer MLPs, the 3-layer MLPs, the 2-layer LSTM, the 2-layer Bi-LSTM with the self-attention mechanism and the proposed method

Figure 13 shows the comparison of the POI-POI similarity. The POI-POI similarity shows the difference of the POIs in the recommended POI list. Since the proposed method considers similar users' historical data, its POI-POI similarity is lower than that of the 2-layer Bi-LSTM with the self-attention mechanism. Since it has a high probability that the characteristics of the similar user's viewed POIs are not necessarily similar to the target user's preference, it can decrease the POI-POI similarity. The method that has the lowest cosine similarity is the 2-layer MLPs and the second one is the proposed method. Even if the proposed method is not better than 2-layer MLPs, the precision and the recall of the proposed method are much better than that of 2-layer MLPs.

Figure 14 shows the comparison of the POI-user similarity. The POI-user similarity shows the difference between the recommended POI list and the user's preference. Since the proposed method, which considers the POIs from similar users, can make the recommended POI list more diversity, the POI-user similarity will be lower. The POI-user similarity of the proposed method is higher than that of 2-layer MLPs

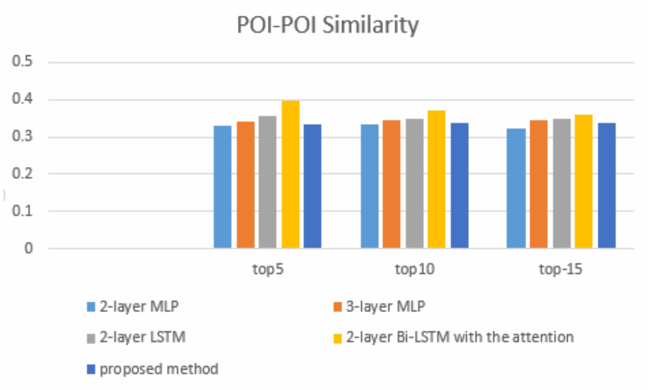


Figure 13. The POI-POI similarity of the 2-layer MLPs, the 3-layer MLPs, the 2-layer LSTM, the 2-layer Bi-LSTM with the self-attention mechanism and the proposed method

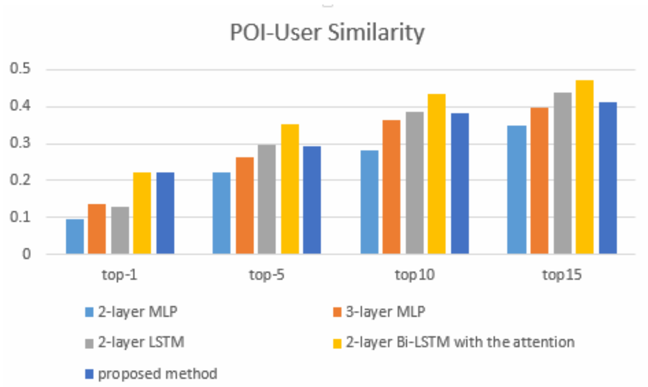


Figure 14. The POI-User similarity of the 2-layer MLPs, the 3-layer MLPs, the 2-layer LSTM, the 2-layer Bi-LSTM with the self-attention mechanism and the proposed method

and 3-layer MLPs, but the recall and the precision are higher than theirs.

Let the term m-n denote that m% results are provided by the deep learning model and n% results are provided by the user-collaborative filtering. The previous comparisons is based on the 80-20 configuration. How about the other configurations? Figure 15 shows the top-10 comparison of the 2-layer MLPs, the 3-layer MLPs, the 2-layer LSTM, the 2-layer Bi-LSTM with the self-attention mechanism, the 70-30 proposed method, the 80-20 proposed method and the 90-10 proposed method. The x-axis denotes all of the compared methods and the y-axis denotes the values of the precision, recall, next POI predicted rate, POI-POI similarity and POI-user similarity. Although the diversity of the 70-30 proposed method is definitely better than that of the 3-layer MLPs, 2-layer LSTM, the 80-20 proposed method and the 90-10 proposed method, i.e., the POI-POI similarity and the POI-user similarity of the 70-30 proposed method are lower than that of the 3-layer MLPs, 2-layer LSTM, the 80-20 proposed method and the 90-10 proposed method, the recall of the 70-30 proposed method is lower than that of the 2-layer LSTM and the next POI predicted rate of the 70-30 proposed method is just 1% better than that of 2-layer LSTM.

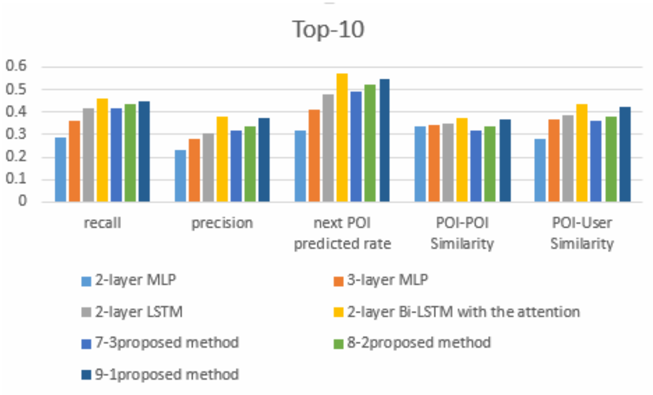


Figure 15. The POI-POI similarity of the 2-layer MLPs, the 3-layer MLPs, the 2-layer LSTM, the 2-layer Bi-LSTM with the self-attention mechanism and the proposed method

Since the 90-10 proposed method's recommended list only has one recommended POI from the user-collaborative filtering, the precision, recall and the next POI predicted rate of the 90-10 proposed method are only lower than that of the proposed method's deep learning model. Nevertheless, the POI-POI similarity and the POI-user similarity of the 90-10 proposed method are higher than that of the 2-layer MLPs, 3-layer MLPs, 2-layer LSTM, the 70-30 proposed method and the 80-20 proposed method, i.e., the POI-POI diversity of the 90-10 proposed method and POI-user diversity of the 90-10 proposed method are worse than that of the 2-layer MLPs, 3-layer MLPs, 2-layer LSTM, the 70-30 proposed method and the 80-20 proposed method. It can be observed that the precision, recall, next POI predicted rate of the 80-20 proposed method is also better than that of the 2-layer MLPs, 3-layer MLPs, 2-layer LSTM and the 70-30 proposed method; however, the POI-POI similarity of the 80-20 proposed method is only higher than that of the 2-layer MLPs and the POI-user similarity of the 80-20 propose d method is just higher than that of the 2-layer MLPs and the 3-layer MLPs. That is, the POI-POI diversity of the 80-20 proposed method is only worse than that of the 2-layer MLPs and the POI-user diversity of the proposed method is just worse than that of the 2-layer MLPs and the 3-layer MLPs. In other words, the 80-20 proposed method sacrifices a few precisions and recalls to let the recommended results have more diversity. Therefore, the 80-20 proposed method is the feasible one that balances the precision, the recall and the diversity.

8 Conclusion

This work has proposed the recommendation system that uses the deep learning model and the user-collaborative filtering to get precise and diverse results. Since many platforms don't have the scoring mechanism, our work uses the deep learning model, which includes the Bi-LSTM and the self-attention mechanism, to catch user's preference from analyzing user's individual data. The performance analysis has shown that the precision, the recall and the next POI predicted rate of the proposed method are a little bit worse than only using the proposed method's deep learning model. However, it is better than that of other methods, which are 2-layer MLPs, 3-layer MLPs and 2-layer LSTM. For the diversity, although the POI-POI similarity of the proposed method is just higher than that of 2-layer MLPs and 3-layer MLPs, the precision, the recall and the next POI predicted rate are much better than that of 2-layer MLPs and 3-layer MLPs. Therefore, the proposed method balances the precision, the recall, the next POI predicted rate and the diversity to make the recommendation system more comprehensive. For the future work, it can combine the question answering function to let the recommendation system understand the natural language and then recommend the POIs to users more precisely.

Acknowledgements

This work was supported by the Ministry Of Science and Technology (MOST), Taiwan (R.O.C.) under the grant number MOST 108-2221-E-006-100-MY3.

References

[1] F. Ricci, L. Rokach, B. Shapira, Introduction to Recommender Systems Handbook, in: F. Ricci, L. Rokach, B. Shapira, P. Kantor (Eds.), *Recommender Systems Handbook*, Springer, 2011, pp. 1-35.

[2] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender Systems Survey, *Knowledge-Based System*, Vol. 46, pp. 109-132, July, 2013.

[3] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep Learning based Recommender System: a Survey and New Perspectives, *ACM Computing Surveys*, Vol. 52, No. 1, p. 1-38, February, 2019.

[4] B. T. Betru, C. A. Onana, B. Batchakui, Deep Learning Methods on Recommender System: a Survey of State-of-the-Art, *International Journal of Computer Applications*, Vol. 162, No. 10, pp. 17-22, March, 2017.

[5] A. Patel, A. Thakkar, N. Bhatt, P. Prajapati, Survey and Evolution Study Focusing Comparative Analysis and Future Research Direction in the Field of Recommendation System Specific to Collaborative Filtering Approach, *Proceedings of the International Conference on Information and Communication Technology for Intelligent Systems (ICTIS2018)*, Ahmedabad, India, 2018, pp. 155-163.

[6] M. J. Pazzani, D. Billsus, Content-based Recommendation Systems, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), *The adaptive web: Methods and Strategies of Web Personalization,*

Lecture Note in Computer Science, Vol. 4321, 2007, pp. 325-341.

[7] X. Su, T. M. Khoshgoftaar, A Survey of Collaborative Filtering Techniques, *Advances in Artificial Intelligence*, Vol. 2009, pp. 1-19, October, 2009.

[8] E. Çano, M. Morisio, Hybrid Recommender Systems: a Systematic Literature Review, *Intelligent Data Analysis*, Vol. 21, No. 6, pp. 1487-1524, January, 2017.

[9] R. Logesh, V. Subramaniyaswamy, Exploring Hybrid Recommender Systems for Personalized Travel Applications, *Proceedings of the 2nd International Conference on Cognitive Informatics and Soft Computing (CISC 2019)*, Rajampet, India, 2019, pp. 535-544.

[10] C. M. Huang, Y. A. Guo, The Demodulating and Encoding Heritage (DEH) Platform for Mobile Digital Culture Heritage (M-DCH), *Journal of Internet Technology (JIT)*, Vol. 21, No. 3, pp. 765-782, May 2020.

[11] J. L. Elman, Finding Structure in Time, *Cognitive Science*, Vol. 14, No. 2, pp. 179-211, April-June, 1990.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, *Proceedings of the 31th International Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, U.S.A., 2017, pp. 5998-6008.

[13] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, November, 1997.

[14] G. Liao, S. Jiang, Z. Zhou, C. Wan, X. Liu, POI Recommendation of Location-based Social Networks using Tensor Factorization, *Proceedings of the 19th IEEE International Conference on Mobile Data Management (MDM)*, Aalborg, Denmark, 2018, pp. 116-124.

[15] H. Wang, W. Ouyang, H. Shen, X. Cheng, ULE: Learning User and Location Embeddings for POI Recommendation, *Proceedings of the 3rd IEEE International Conference on Data Science in Cyberspace (DSC)*, Guangzhou, China, 2018, pp. 99-106.

[16] R. Ding, Z. Chen, RecNet: a Deep Neural Network for Personalized POI Recommendation in Location-based Social Networks, *International Journal of Geographical Information Science*, Vol. 32, No. 8, pp. 1631-1648, August, 2018.

[17] C. Ma, Y. Zhang, Q. Wang, X. Liu, Point-of-Interest Recommendation: Exploiting Self-Attentive Autoencoders with Neighbor-Aware Influence, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Turin, Italy, 2018, pp. 697-706.

[18] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural Collaborative Filtering, *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia, 2017, pp. 173-182.

[19] C. Du, C. Li, Y. Zheng, J. Zhu, B. Zhang, Collaborative Filtering with User-Item Co-autoregressive Models, *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, U.S.A., 2018, pp. 2175-2182.

[20] A. Ng, Sparse Autoencoder, *CS294A Lecture Notes*, Stanford University, pp. 1-19, 2011.

[21] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, Scottsdale, Arizona, U.S.A., 2013, pp. 1-12.

Biographies

Chung-Ming Huang received the Ph.D. degree in computer and information science from The Ohio State University on 1991/6. He is currently a Distinguished Professor of Dept. of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C. He a senior member of IEEE (SM'07) and ACM (SM'12).

Chen-Yi Wu received the Master degree in Computer Science and Information Engineering from National Cheng Kung University, Tainan, Taiwan, on 2019/8. His research interests include Machine Learning and Deep Learning.