Fusing User Reviews into Heterogeneous Information Network Recommendation Model

XU CHEN¹, JINGJING TIAN¹, XINXIN TIAN¹, AND SHUDONG LIU¹

¹School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China

Corresponding author: ShuDong Liu (e-mail: bupt.mymeng@gmail.com).

This research was funded by National Funds of Social Science(21BXW073), Philosophy and Social Science Research Project of Hubei Provincial Department of Education (20G026) and Innovation Research of Young Teachers of Central Universities in 2021(2722021BZ040).

ABSTRACT With the advent of the information epoch and the development of Big Data, users are constantly overwhelmed in massive information online. As an effective method to deal with this dilemma of information overload, recommendation system has become a popular area of research for the past few years. However, most of the current recommendation models fail to make the most of the ample resources hidden behind auxiliary data and user reviews. For this reason, we put forward the FHRec model. We try to combine heterogeneous information network with deep learning technology to improve recommendation performance. The working mechanism is to represent the rich auxiliary data by using heterogeneous information network, learning the features of entities through network embedding; in the meantime, by adopting deep learning technology to mine the features of entities from review texts, using the attention mechanism to fuse entity feature vectors, inputting them into the neural network for recommendation; the last step is that we use Yelp dataset and Douban movie dataset to verify the FHRec model. The experimental results show that the FHRec behaves better than the traditional comparison algorithm on the Yelp dataset and Douban movie dataset.

INDEX TERMS Recommendation System; Reviews; Network Embedding; DeepCoNN Model; Neural Networks

I. INTRODUCTION

Today, with the advent of the information epoch and the development of Big Data, information is expanding everywhere. People are facing large amounts of redundant messages, which will interfere with their judgment and selection for useful information. Therefore, the emergence of recommendation system, which helps users pick out the information they are in need of, can alleviate the current issue of information overload to a certain extent.

The application of recommendation algorithm can be seen everywhere on the website we browse. For example, when we browse e-commerce websites, there will be a "guess your favorite" function on the front page. There are many other websites that have applied recommendation algorithms to their own websites as well, including movies, videos, news, books, takeout, map and so on. And among existing recommendation algorithms, collaborative filtering (CF) algorithm [1] is one of the most popular of late years. However, CF has the problem of sparsity. When the users have little historical behavior data, the co-occurrence matrix between users and items will be very sparse, which will affect finding the users who are similar to the current user or the items that are similar to the current user’s preferences, and it will eventually lead to large errors in the recommendation results. Matrix factorization algorithm (MF) [2] strengthens the ability of the recommendation model to process sparse matrices. By decomposing the rating matrix into the user’s and item’s potential factor matrix, the sparsity of data can be effectively reduced. However, MF algorithm is only based on the rating matrix, as a result of which it is difficult to integrate other features that may affect results into recommendation system in the training process, such as friendship networks of the target user, commodity evaluation, store location, etc. Factorization machine (FM) [3] can be regarded as the generalization of basic matrix factorization, designed to solve the problem of feature combination under large-scale sparse data, and
can well integrate multi-dimensional features. However, FM only considers the interaction between low-order features and ignores the interaction between high-order features.

With the emergence of heterogeneous information network (HIN), it opened up a new development direction for the evolution of recommendation systems. As we all know, the real world is heterogeneous, with a large quantity of entities belonging to the different categories and a variety of complex relationships between these entities, so the recommendation system can use heterogeneous information network to characterize rich auxiliary data. By fully mining and utilizing the information in the HIN, the recommendation results will be more accurate. Over recent years, the research of recommendation algorithms based on HIN has been on an increase. For example, Chen et al. [5] combined model-based CF with heterogeneous information networks, and applied user’s feedback and the similarities with different semantics measured by PathSim [6] algorithm to the model. Shi et al. [7] proposed SemRec based on semantic path, this model can flexibly integrate heterogeneous information through meta-paths, can also obtain priority and personalized weight preferences on multiple meta-paths.

With the rise of network representation learning for the past few years, network representation learning technology has been put into use in the research of recommendation algorithms. For instance, Shi et al. [8] put forward the HERec model, which generated a node sequence by random walk based on meta-path, learnt the embedding representation of nodes by network embedding technology, and finally integrated them into the MF model for the recommendation. Zhou et al. [9] put forward a scalable graph embedding method, which can maintain the asymmetric similarity between node pairs. This method has been tested on an industrial scale large graph with hundreds of millions of nodes. Hu et al. [10] put forward a unified model to integrate direct interaction information and generalize interaction information for top-N recommendation.

Later, more researchers employ deep learning technology in recommendation systems with the boom of this technology. For example, Hu et al. [11] proposed a new DNN with co-attention model, taking advantage of rich contexts based on meta-path for Top-N recommendation. Wang et al. [12] put forward the NGCF model, which used the high-order connection information between user and item to encode the interaction information, and effectively injects the collaborative signal of user and item into the embedding process in an explicit way.

However, most of these existing recommendation algorithms fail to consider the effective combination of auxiliary information, user ratings and user reviews in the recommendation system. Auxiliary information contains user’s attributes and item’s attributes, which belongs to implicit feedback. Therefore, the implicit features for entities can be mined from auxiliary information. User ratings can intuitively reflect a user’s preference for an item, but meanwhile, it should be considered that rating habits may vary among users. For example, some users are usually used to giving high ratings, while others may have high requirements, and the ratings given are generally low. In this case, it is difficult to evaluate the quality of the item only by the user ratings. In addition, even if both users give a high rating, it does not mean that they have the same sentiment tendencies towards the item, there may be a user who likes the item very much, and the other user is just because of the rating habits, he is accustomed to giving a relatively high rating for any item that is not too bad. To effectively solve these problems, we will apply user reviews to the recommendation system. Both user reviews and user ratings are explicit feedback, but user reviews are more subjective than user ratings and can better reflect users’ sentiment tendency.

Auxiliary information, user ratings and user reviews are very important for improving the performance of the recommendation system, and the effective combination of them can enhance the recommendation effect of recommendation system and alleviate the problem of sparsity to a certain extent, but some recommendation algorithms only take into account the user ratings and some auxiliary information in the recommendation system, ignoring the importance of reviews; or only consider user reviews, ignoring the role of user ratings and auxiliary information. Therefore, in this paper, we applies auxiliary information, user ratings and user reviews to the recommendation system. We use HIN to represent rich auxiliary data and fuse user reviews into HIN based recommendation system.

II. PRELIMINARY
A. INFORMATION NETWORK
Definition 1. Information network [13]. A directed graph \( G = (V, E) \) is used to represent the information network, \( V \) is the collection of object nodes in the network, \( E \) is the set of edges between different object nodes in the network, and the node type \( V \) needs to meet the object mapping relationship \( \Phi: V \to A \), which means that the type of each node \( v \in V \) belongs to object type set \( A \). Edge type \( E \) needs to meet the link mapping relationship \( \Psi: E \to R \), which indicates that the type of each edge \( e \in E \) belongs to the link type set \( R \).

If \( \mid A \mid > 1 \) and \( \mid R \mid > 1 \), indicating the information network is a HIN. As is shown in Figure 2, (a), (b) respectively show the HIN built based on Yelp website and the movie systems. We take the HIN built on a movie system as an example, the HIN consists of three kinds of nodes belonging to different types, that is, three types of objects, including director, movie and actor. The edge connecting the two nodes represents a certain relationship between the two nodes. For example, the edge connecting the director and the movie indicates that the director made the movie, and the edge connecting the movie and the actor indicates that the actor appeared in this movie.

B. NETWORK SCHEMA
Definition 2. Network schema [4]. It can be defined as \( TG = (A, R) \), which is a directed graph defined on \( A \) and \( R \). As is shown in Figure 3, the network schemas are more abstract
representation of Figure 2, and describe different types of objects contained in Figure 2, and the relationships between various object types.

C. META-PATH

Definition 3. Meta-path [4]. It is defined on TG (A, R) and can be expressed in the form of \( A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} A \xrightarrow{R_3} \cdots \xrightarrow{R_s} A_{s+1} \). The combination relationship \( R \) between \( A_1 \) and \( A_{s+1} \) can be expressed as \( R = R_1 \circ R_2 \circ R_3 \circ \cdots \circ R_s \), where \( \circ \) is the combination operator of the relationship. \( A_i \) represents the object type, and \( R_i \) represents the relationship type.

As is shown in Fig. 4, two examples of meta-paths extracted from the network schema based on Yelp website and movie systems respectively. The expression of the meta-path "User-Business-City- Business-User (UBCiBU)" in Fig. 4(a) means that multiple users have been to the business in the same city. And the expression of the meta path "Movie-Director-Movie (MDM)" in the Fig. 4(b) means that different movies were made by the same director.

III. FHREC MODEL

In our work, we will use heterogeneous information networks to represent the rich auxiliary data in the recommendation system, there will generate meta-paths based on HINs, and then we use network embedding to learn the potential vector representation for users and items. At the same time, we
extract the user features and item features from the reviews by using the Deep Cooperative neural Networks. Then the attention mechanism will be applied to fuse the obtained user features and item features respectively. Finally, using feed-forward neural networks to model relationships between higher-order features, so as to make recommendations.

As shown in Figure 1, the FHRec model is mainly divided into the following five parts:

1) Building the network schema and meta-paths;
2) Obtaining the feature vectors for users and items by network embedding based on HINs;
3) Extracting user features and item features from reviews;
4) Adopting the attention mechanism to fuse feature vectors for users and items respectively;
5) Using deep neural network to learn higher-order non-linear feature interactions, and then making rating prediction.

A. THE CONSTRUCTION OF NETWORK SCHEMA AND META-PATHS

The experimental data used in this paper are the Yelp dataset and Douban Movie dataset, so we take the construction process of the network schema and meta-paths of Yelp dataset and Douban Movie dataset as examples.

1) CONSTRUCTION OF NETWORK SCHEMA

The network schemas extracted from the heterogeneous information network based on Yelp website and Douban Movie are shown in Fig. 3. The network schemas extracted from Yelp dataset and Douban Movie dataset respectively contain five objects and five relationships between different objects.

2) CONSTRUCTION AND ANALYSIS OF META-PATHS

Random walk with different types of nodes as starting points or as ending points will generate different meta-paths based on HIN, and when the starting nodes and ending nodes are the same, different meta-paths may also be generated. Therefore, too many meta-paths will be generated based on random walks, we need to choose the most suitable meta-paths for our experiment among these meta-paths. In view of the ultimate goal of generating meta-paths we work towards is to learn the vector representations for users and items, so we will select meta-paths that start and end with user types or item types in this paper.

The meanings of meta-paths extracted from Yelp website and Douban Movie are shown in Table 1 and Table 2 respectively.

After extracting effective meta-paths from a HIN, the corresponding meta-path files need to be constructed. For example, three files need to be constructed from Yelp dataset, namely user-business relationship (UB), business-city relationship (BCi) and business-category relationship (BCa), and then we need to generate other relevant meta-path files. In

---

FIGURE 3. Network schemas extracted from the heterogeneous information networks built based on Yelp website and Movie systems.

FIGURE 4. Meta-paths extracted from the network schemas of Yelp website and Movie systems.
order to facilitate the experiment, we first need to convert the original data into numbers, and then carry out different matrix operations according to different meta-paths. For example, the matrix calculation formula of meta-path UBCiBU is:

\[ P_{UBCIBU} = M_{UB} \cdot M_{BC} \cdot M_{BCiB}^T \cdot M_{UB}^T. \]  

where \( P_{UBCIBU} \) represents the matrix corresponding to the meta-path UBCiBU, \( M_{BC} \) represents the matrix generated from the meta-path file corresponding to the meta-path BC and \( M_{BCiB}^T \) represents the transpose of \( M_{BCi} \).

### B. NETWORK EMBEDDING

Inspired by the HERec model [8], we choose to use the network embedding method to represent nodes as vectors. The process of network embedding is equivalent to a mapping function. Its purpose is to find a low-dimensional space to represent the network, and every node in the network will be converted to a low-dimensional potential vector representation [14], and then applying the vector representation to common social network applications.

Drawing on the way network embedding is done in the HERec model, we will use the DeepWalk algorithm proposed in [15] for network embedding in this paper. The algorithm applies unsupervised feature learning technology to the network analysis, so that it makes the most of the rich information of nodes sequence generated by meta-path based random walk in the network structure.

DeepWalk algorithm mainly consists of two parts: firstly, generating node sequence by meta-path based random walk, followed by the generation of node vectors using the Skip-gram model. The pseudo-code of DeepWalk algorithm is shown in Table 3.

1) GENERATING NODE SEQUENCE

The process of random walk is to start from a node, then randomly select an adjacent node, then start from this adjacent point to the next node, repeat this step, and then record the passing nodes until the preset path length is reached. By random walk, a path representing the structure information of each node can be obtained from each node.

However, it is difficult to distinguish whether each node in the node sequence generated by random walk based on heterogeneous information network belongs to user type or item type. Therefore, we need to develop a strategy that ensures each node in node sequence and its adjacent nodes belong to different types, so that the interaction between users and items can be further mined.

Given \( G = \{V, E\} \) and meta-paths \( \delta: A_1 \xrightarrow{R_1} \cdots \xrightarrow{R_{n-1}} A_i \xrightarrow{R_i} A_{i+1} \cdots \xrightarrow{R_k} A_{i+k} \), according to the given meta-paths, the node sequence is generated by using the random walk strategy defined as follows:

\[
P(n_{i+1} = x|n_i = v, \rho) = \begin{cases} \frac{1}{\left|N^+(v, x)\right|}, & (v, x) \in E, \varphi(x) = A_{i+1} \\ 0, & \text{others} \end{cases}
\]

where \( n_i \) represents the i-th node in the walk, \( v \) belongs to the type \( A_1 \), \( n_{i+1} \) is the next node of \( v \), \( N^+(A_i, x) \) represents the set of nodes belonging to type \( A_{i+1} \) in all direct adjacent nodes of \( v \).

2) GENERATING NODE VECTORS

After generating node sequences by random walk based on meta-paths, we use Skip-gram model [16] to train them to
generate node vectors.

As is shown in Figure 5, the skip-gram model includes input layer, mapping layer and output layer. We take the target node as input to the skip-gram model and then there will output the other nodes in the same node sequence, then the vector representation of the node can be obtained through maximizing the co-occurrence probability that the target node appears in the same window as other nodes in the node sequence. We represent the obtained node vector belonging to the user type as $e_u$ and belonging to the item type as $e_i$.

C. FEATURE EXTRACTING

In this part, we will apply deep learning technology to extract the features from reviews. Among the existing algorithms for text mining, DeepCoNN model proposed in [17] uses reviews to jointly model users’ behavior and items’ attributes, and can learn users’ behavior and items’ attributes from texts. Therefore, we will use this algorithm to mine the user features and item feathers from reviews.

The structure of DeepCoNN is shown in Figure 6, which mainly consists of two parallel CNN networks [18]. Finally, using a sharing layer to couple the user network and item network.

The user and the item network differ only in input data, so we will describe the user network in detail below, and the item network is the same.

1) LOOK-UP LAYER

The look-up layer is designed to use word vector technology to vectorize the review texts into a word vector matrix so as to extract the latent features from the review texts. The specific process of obtaining word vector matrix is as follows:

1) Firstly, the review texts of each user on different items are gathered together and conducting word segmentation. We set the maximum number of words in review set of each user as n, using the filling character for the review texts with less than n words, and taking the first n words according to the word frequency for the review texts with more than n words, then generating a word dictionary based on the reviews dataset;

2) Using the word2vec tool to train the word dictionary to obtain the word vectors corresponding to the words, and we set the dimension of word vector as K. The word vector matrix $V_{1:n}^u$ of the user u is defined as follow:

$$V_{1:n}^u = \phi(d_1^u) \oplus \phi(d_2^u) \oplus \phi(d_3^u) \oplus \cdots \oplus \phi(d_n^u).$$  (3)

where $d_k^u$ represents the k-th word, $\phi(d_k^u)$ returns the vector representation of the k-th word, $\phi$ is the connection operation.

2) CNN LAYER

CNN layer takes the word vector matrix of users and items as input respectively, and then extracts features from the word vector matrix.

The first layer of CNN layer is the convolution layer, which contains multiple convolution kernels, whose function is to
extract the features from the reviews. We take user u as an example, the word vector matrix $V^u_{1:n} \in \mathbb{R}^{(n \times k)}$ of user u has been obtained, then performing convolution operation, setting the number of convolution kernels as m, and adding the bias and activation function. The formula is defined as follows:

$$z_j = f(V^u_{1:n} \ast W_j) + b_j.$$  \hspace{1cm} (4)

where $W_j \in \mathbb{R}^{(c \times t)}$ is the j-th convolution kernel, t represents the width of convolution kernel, $\ast$ is the convolution operation, $b_j$ represents the bias and f represents the activation function. The formula is defined as follows:

$$ReLU(x) = \max(0, x).$$  \hspace{1cm} (5)

The object of the max-pooling layer is to find the maximum $o_j$ for each vector $z_j \in \mathbb{R}^{(N-1+t+1)}$. The formula is defined as follows:

$$a_j = \max\{z_{j1}, z_{j2}, z_{j3}, \cdots z_{j(n-t+1)}\}.$$  \hspace{1cm} (6)

The output vector of the max-pooling layer is shown in (7).

$$O = \{o_1, o_2, o_3, \cdots o_m\}.$$  \hspace{1cm} (7)

As is shown in (8), the feature vector for user can be obtained.

$$e_u = f(W \times O + g).$$  \hspace{1cm} (8)

where W and g mean the weight coefficient and bias of the full-connected layer respectively. The feature vectors for items $y_i$ can be extracted from the item reviews through the above process.

**D. FEATURE FUSING**

Random walk from a node on HIN will generate different meta-paths, different embedding vector representations of nodes can be obtained based on different meta-paths. Therefore, we need to fuse different vector representations of the same node through using a set of fusion functions. At the same time, the features obtained from the HINs belong to the features obtained from implicit feedback and the features extracted from reviews belong to the features obtained from explicit feedback. They all represent the features of a certain aspect of users and items. Therefore, it is also necessary to fuse the features extracted from the reviews with those obtained through different meta-paths.

We consider that some users have different preferences for different meta-paths in real life. For example, some users prefer movies starring their favorite stars, while others prefer movies with comedy themes. Therefore, we need to assign different weights to different meta-paths in the process of fusion. In addition, reviews are the explicit feedback of users, while features acquired through meta-paths are the implicit feedback. They have different impacts on the results of recommendations and need to be given different weights. Therefore, we use the attention mechanism to assign weights between the features obtained through meta-paths and the features extracted from reviews and between different meta-paths. Specifically, through a two-layer neural network learning attention scores for the features learned from different meta-paths and the features extracted from reviews. Taking user’s feature vector fusion as an example, the feature vector fusion process of items is the same, the calculation formula is as follows:

$$s^{(l)}_u = \sigma(W^{(l)}_1 \sigma(W^{(l)}_2 \hat{e}^{(l)}_u + b^{(l)}_1) + b^{(l)}_2).$$  \hspace{1cm} (9)

where $s^{(l)}_u$ represents the attention score of user features learned over the l-th meta-path or user features extracted from reviews, $\sigma$ represents the activation function, $W^{(l)}_1$ and $W^{(l)}_2$ represent weight coefficient, $b^{(l)}_1$ and $b^{(l)}_2$ are the bias, $\hat{e}^{(l)}_u$ represents the feature vectors of user u generated under a meta-path or extracted from reviews.

After the attention score is obtained, softmax function is used for normalization, and the formula is as follows:

$$\alpha^{(l)}_u = \frac{\exp(s^{(l)}_u)}{\sum_{i=1}^{k} \exp(s^{(l)}_u)}.$$  \hspace{1cm} (10)

where k is the number of meta-paths + 1.

After obtaining the attention weight of the features generated under the meta-paths and extracted from the reviews, we can get the final vector representation of the user u through the fusion function, the process is defined as follows:

$$e_u = \sum_{l=1}^{k} \alpha^{(l)}_u \hat{e}^{(l)}_u.$$  \hspace{1cm} (11)

where $e_u$ is the final representation of the u, k is the number of meta-paths + 1.

For a group of user-item pairs, we can obtain the final vector representation for user and item respectively through the above process, we concatenate them and form a new vector, which is denoted by $x^{(n)}$, as is shown in (12).

$$x^{(n)} = \{e_u, e_i\}.$$  \hspace{1cm} (12)

where $x^{(n)}$ represents the vector combination of the n-th user-item pair.

**E. RATING PREDICTION**

Using neural networks to learn higher-order interactions between features, which is defined as follows:

$$h_1 = \sigma(w_1 x^{(n)} + b_1),$$
$$h_2 = \sigma(w_2 h_1 + b_2),$$
$$\cdots$$
$$h_l = \sigma(w_l h_{l-1} + b_l).$$  \hspace{1cm} (13)

where $w_l$ represents the weight coefficient of the l-th hidden layer, $b_l$ is the bias and $\sigma$ represents the activation function used in the l-th hidden layer.

The output of the last hidden layer is transformed into rating prediction, as is shown in (14).

$$\hat{r}_{u,i} (x) = W h_1 + b.$$  \hspace{1cm} (14)
F. MODEL LEARNING
The objective function of model training is shown in (15):
\[
\min : Loss = \sum_{u,i \neq 0} (r_{u,i}(x) - \hat{r}_{u,i}(x))^2. \tag{15}
\]
where \( r_{u,i}(x) \) represents the actual rating, and \( \hat{r}_{u,i}(x) \) represents the predicted rating using (14). We use Adam to update the learning parameters and adjust hyperparameters through grid search method.

G. METRICS
We adopt MAE and RMSE as evaluation metrics to evaluate experimental results. The calculation process of MAE is shown in (16) and the calculation process of RMSE is shown in (17).
\[
MAE = \frac{1}{|D|} \sum_{(i,j) \in D} |r_{i,j} - \hat{r}_{u,i}|. \tag{16}
\]
\[
RMSE = \sqrt{\frac{1}{|D|} \sum_{(i,j) \in D} (r_{i,j} - \hat{r}_{u,i})^2}. \tag{17}
\]
where D is the dataset of rating records.

When the difference between the predicted rating and the actual rating is smaller, the values of MAE and RMSE will be smaller. Hence the smaller MAE and RMSE are, indicating the better the performance of the experimental model is.

In addition, we also add Recall@K, NDCG@K, HR@K and AUC as metrics to evaluate the Top-K recommendation list generated by the proposed recommendation algorithm. The higher these metrics are, indicating the more accurate the recommendation results are.

IV. EXPERIMENT AND RESULT ANALYSIS

A. EXPERIMENTAL DATA
The experimental data used are Douban movie dataset and Yelp dataset. Because the dataset used in the experiment needs to contain auxiliary information, user ratings and user reviews, we selected the Yelp dataset that meets this condition. In the commonly used recommended dataset, there are very few that meet this condition, so in addition to the Yelp dataset, we also crawled the Douban Movie dataset through a web crawler technology based on python.

Douban Movie dataset contains three attributes: actor, director and type, as well as users’ ratings and reviews on movies. The Yelp dataset comes from the Yelp website, users are able to rate the enterprises they have experienced and post their reviews on the website. The rating ranges from 1 to 5; a higher rating means that users like the company, while a lower one indicates that users’ feedback on the company is negative. The Yelp website will recommend enterprises according to users’ preferences. We divide Yelp dataset into two depending cubes with different sizes and sparsity. The three datasets are described in Table 4. We divide the three datasets into training set and test set, with the proportion of training set is 80% of the entire dataset and test set accounting for 20% of the entire dataset.

B. BASELINES
For the purpose of verifying whether the proposed model improves the recommendation performance, the following models are used for comparison, as is shown below:
1) MF [2]: a matrix factorization model using rating data for recommendation. Predicting rating by the dot product of the user’s potential factor matrix and item’s potential factor matrix which are obtained by decomposing rating matrix.
2) HERec [8]: a recommendation algorithm based on HIN, proposing a new network embedding method, and recommendation effect is optimized by the combination of embedding and MF model. Inspired by the HERec model, we choose to use the network representation learning method to learn the vector representation of nodes, so we take it as one of the baselines.
3) FHRecq: to verify whether the reviews used to the FHRec model can enhance the recommendation performance of the experimental model, we remove the part of extracting features from the reviews, and take it as one of the comparison algorithms.
4) NGCF [12]: a new recommendation framework, which used the high-order connection information between user and item to encode the interaction information, and effectively injects the collaborative signal of user and item into the embedding process in an explicit way.

C. EXPERIMENTAL RESULTS
We run our model in Tensorflow. The results of MAE and RMSE obtained performing different models on the three datasets are shown in Table 5. We calculate the improvement rates of FHRec model over the strongest baselines on three datasets in the last column of the table. The results of other evaluation indicators obtained by performing different models on Douban dataset are shown in Table 6. The following four conclusions can be drawn from the experimental results:

1) As can be seen from Table 5, NGCF model has the best performance among comparison algorithms except FHRec, while the MAE and RMSE of FHRec have improved greatly on the basis of NGCF. For example, on Douban Movie dataset, MAE and RMSE of FHRec have improved by 23.53% and 18.31% respectively compared with NGCF, thus verifying the validity of FHRec recommendation model. Meanwhile, as can be seen from Table 6, regardless of k=20 or k=100, the Recall, NDCG and HR of FHRec are the highest, and the AUC of FHRec is also higher than that of other comparison models, indicating that FHRec model performs better than other comparison models in top-N recommendation. Therefore, FHRec model has certain practicality.

2) The MAE and RMSE of FHRec, FHRecq and HeRec models based on heterogeneous information networks are smaller than those of MF model on three datasets. And all the other metrics used to evaluate top-N rec-
ommendation are higher and perform better than MF. Therefore, it is shown that using the heterogeneous information network to represent rich auxiliary data can improve the performance of the recommendation model.

3) It is obvious that FHRec model performs better than FHRecq model from the experimental results. For example, on Yelp1 dataset, MAE and RMSE of FHRecq are 0.9907 and 1.2244 respectively, MAE and RMSE of FHRec are 0.7291 and 0.9626 respectively. MAE and RMSE improved by 26.41% and 21.38% respectively. In addition, when k=20, Recall, NDCG, HR and AUC of FHRec improved by 18.31%, 19.33%, 14.45% and 7.66%, respectively, compared with FHRecq. Therefore, it is shown that the user reviews contain rich information, which can be used by the recommendation system and can help the recommendation system improve its performance.

4) We can also see that NGCF model performs better than MF. FHRecq and HeRec. For example, NGCF’s RMSE improved by 6.72%, 9.03% and 37.95% over HeRec, FHREC and MF respectively on Yelp2 dataset, indicating the importance of capturing collaborative signal in the embedding function.

D. IMPACT OF DIFFERENT META-PATHS
To research the impact of meta-paths on experimental results, we use Recall@K, NDCG@K, and HR@K as evaluation metrics and conduct experiments under different meta-paths on Yelp2 dataset. The experimental results when k is 20, 60 and 100 are shown in Fig. 7, from which we can see that the four evaluation metrics show an overall upward trend with the increase of the number of meta-paths. At the same time, it’s obvious that at the beginning, with the addition of more and more meta-paths starting and ending with the user type, the performance of the experimental model on the four metrics is relatively stable, and the fluctuation is small. Then, with the addition of "bub", the performance of the experimental model is improved rapidly, indicating that the meta-path "bub" contains rich information that can improve the performance of the experimental model. However, the performance of the experimental model will decrease when the meta-path "bcib" is added. The reason may be the meta-path "bcib" exist noisy data, which affect the experimental results. The experimental results are optimal when the meta-path "bcbab" is added, so we can determine the optimal set of meta-paths.

E. IMPACT OF DIFFERENT DATA SPARSITY
At present, one of the major problems faced by the recommendation system is the sparsity problem. In this part, we mainly discuss the performance of the experimental model on different sparsity data. We conduct experiments on three datasets and select MAE and RMSE as the experimental metrics. First, we divide the users into (0,5], (5,15], (15,30], and (30,) groups according to the numbers of their rating records in the training set, the division results are shown in Table 7, from which we can see the first group contains the largest number of users on Douban Movie dataset and Yelp1 dataset, the last group of the three datasets has the least number of users. Fig. 8 shows the improvement ratio of MAE and RMSE of FHRec model, NGCF model and FHRecq model compared with HeRec model under different data sparsity. From Fig. 8, we can find that the model proposed in this paper performs best under all sparsity. Therefore, it can be considered that FHRec model can alleviate the problem of data sparsity to a certain extent. At the same time, it can be observed that the performance improvement of the three models over the HeRec model in the first group on Douban Movie dataset and Yelp1 dataset and the third group on Yelp2 dataset is less than that of the other groups. This is because the first group on Douban Movie dataset and Yelp1 dataset and the third group on Yelp2 dataset contain more rating records than other groups, so more information is available.

F. PARAMETER TUNING
For the proposed model, the number of hidden layers is a very important parameter. Therefore, we compare the model performance under different number of hidden layers. First of all, we divide the dataset into three categories, with the proportion of training set is 80% of the entire dataset, the test set accounting for 10%, and the proportion of verification set is 10% of the entire dataset. We examine the impact of the number of hidden layers on the experimental results on the verification set. We vary the number of hidden layers in the set of {1, 2, 3, 4}, and we adjust hyperparameters through grid search method. The learning rate is tuned in the set of {0.1, 0.05, 0.01, 0.001}, the size of embedding vectors in {8, 16, 32, 64, 128}, and the dropout ratio is tuned in set of {0.1, 0.3, 0.5, 0.8}.

As is shown in Figure 9, we can see that it’s not the deeper the network is, the better the proposed model behave. When the number of hidden layers is 2, RAE and RMSE are the lowest on the three datasets, that is, the model performs best. Therefore, the number of hidden layers is set to 8 in the process of the experiment.

V. CONCLUSION
To sum up, we propose the FHRec model to make full use of auxiliary information, user ratings and user reviews, which contain large amounts of available information. We use HIN to represent rich auxiliary data, generate meta-paths starting and ending with user node types or item node types based on random walk in HIN and learn the potential vector representation of nodes through the network embedding. At the same time, the DeepCoNN algorithm is used to obtain user features vectors and item features vectors from reviews. Then the attention mechanism is used to fuse user and item features separately. Finally, these features can be used as the input vectors of the neural network to extract the deep-seated relations between features. We verify the effectiveness of FHRec model in recommendation through experiment,
but there still exist many deficiencies. For example, many user attributes and item attributes have not been utilized yet. In addition, the impact of sentiment tendency of texts on user features and item features is not taken into account when extracting user and item features from review texts. Therefore, in future research, we will further improve this FHRec model by targeting at making up these deficiencies.

REFERENCES

[1] M. Nilashi, K. Bagherifard, O. Ibrahim, et al., “Collaborative Filtering Recommender Systems,” Research Journal of Applied Sciences Engineering & Technology, vol. 5, no. 16, pp. 4168–4182, 2013.

[2] Y. Koren, R. Bell and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems,” Computer, vol. 42, no. 8, pp. 30–37,2009.

[3] S. Benda, “Factorization Machines,” 2010 IEEE International Conference on Data Mining, 2010, pp. 995–1000, DOI: 10.1109/ICDM.2010.127.

[4] C. Shi, Y. T. Li, J. W. Zhang, Y. Z. Sun, and P. S. Yu, “A survey of heterogeneous information network analysis,” IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 1, pp. 17–37, 2017.

[5] Y. Chen, R. Liu and W. Xu, “Movie recommendation in heterogeneous information networks,” 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, 2016, pp. 637–640, DOI: 10.1109/ITNEC.2016.7560438.

[6] Y. Z. Sun, J. W. Han, X. F. Yan, P. S. Yu, and T. Y. Wu, “PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks,” Proceedings of the VLDB Endowment, vol. 4, no. 11, pp. 992–1003, 2011.

[7] C. Shi, Z. Q. Zhang, Y. G. J. et al., “SemRec: a personalized semantic recommendation method based on weighted heterogeneous information networks,” World Wide Web, vol. 22, no. 1, pp. 153–184, 2019.

[8] C. Shi, B. Hu, W. X. Zhao and P. S. Yu, “Heterogeneous Information Network Embedding for Recommendation,” In IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 2, pp. 357–370, Feb. 2019, DOI: 10.1109/TKDE.2018.2833443.

[9] C. Zhou, Y. Q. Liu, X. F. Liu, Z. Y. Liu, and J. Gao, “Scalable graph embedding for asymmetric proximity,” Proceedings of Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, Feb. 2017, pp. 2942–2948.

[10] B. Hu, C. Shi, W. X. Zhao, and T. C.Yang, “Local and global information fusion for top-n recommendation in heterogeneous information network,” In Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM ’18), New York, NY, USA, Oct. 2018, pp. 1683–1686. DOI: https://doi.org/10.1145/3269206.3269278.

[11] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, “Leveraging Meta-path based Context for Top- N Recommendation with A Neural Co-Attention Model,” In Proceedings of the 24th ACM SIGKDD International Conference, New York, NY, USA, July 2018, pp. 1531–1540. DOI: https://doi.org/10.1145/3219981.3219965.

[12] W. Xiang, et al., “Neural graph collaborative filtering,” Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval, 2019.

[13] C. Shi, X. Kong, Y. Huang, P. S. Yu and B. Wu, “HetSim: A General Framework for Relevance Measure in Heterogeneous Networks,” In IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 10, pp. 2479–2492, Oct. 2014, DOI: 10.1109/TKDE.2013.2297920.

[14] Y. Yin, L. X. Ji, J. P. Zhang and Y. -L. Pei, “DHNE: Network Representation Learning Method for Dynamic Heterogeneous Networks,” in IEEE Access, vol. 7, pp. 134782–134792, 2019, DOI: 10.1109/ACCESS.2019.2942221.

[15] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, Aug. 2014, pp. 701–710. DOI: https://doi.org/10.1145/2623330.2623732.

[16] T. Mikolov, K. Chen, G. Corrado, et al., “Efficient Estimation of Word Representations in Vector Space”, 2013, arXiv: 1301.3781v3. [Online]. Available: https://arxiv.org/abs/1301.3781v3

[17] L. Zheng , V. Noroozi, P. S. Yu, “Joint Deep Modeling of Users and Items Using Reviews for Recommendation,” In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM ’17), New York, NY, USA, Feb. 2017, pp. 425-434.

[18] X. Feng and Y. Zeng, “Neural Collaborative Embedding From Reviews for Recommendation,” in IEEE Access, vol. 7, pp. 103263–103274, 2019, DOI: 10.1109/ACCESS.2019.2931357.
### TABLE 4. THE DETAILS OF DATASETS.

| Dataset (Density) | Relations (A-B) | Number of A | Number of B | Number of (A, B) |
|-------------------|-----------------|-------------|-------------|------------------|
| Douban Movie (0.35%) | User-Movie | 5647 | 647 | 12633 |
|                    | Movie-Actor | 647 | 1050 | 1281 |
|                    | Movie-Type | 647 | 28 | 1161 |
|                    | Movie-Director | 647 | 581 | 710 |
| Yelp1 (0.03%) | User-Business | 19719 | 15812 | 96477 |
|                  | Business-City | 15812 | 418 | 15812 |
|                  | Business-Category | 15812 | 1093 | 68148 |
| Yelp2 (0.22%)  | User-Business | 2100 | 12850 | 59814 |
|                | Business-City | 12850 | 376 | 12850 |
|                | Business-Category | 12850 | 1028 | 55361 |

### TABLE 5. The results of MAE and RMSE obtained performing different models on the three datasets.

| Dataset | Metrics | MF | HeRec | NGCF | FHRecq | FHRec | Improvement (%) |
|---------|---------|----|-------|------|--------|-------|-----------------|
| Douban Movie | MAE    | 1.4923 | 0.7311 | 0.5606 | 0.6804 | 0.4287 | 23.53% |
|          | RMSE   | 1.9473 | 0.9216 | 0.7415 | 0.8968 | 0.6057 | 18.31% |
| Yelp1   | MAE    | 1.7075 | 1.0675 | 0.8329 | 0.9907 | 0.7291 | 12.46% |
|          | RMSE   | 2.0871 | 1.428 | 1.0939 | 1.2244 | 0.9626 | 12.00% |
| Yelp2   | MAE    | 1.1673 | 0.7962 | 0.7324 | 0.8113 | 0.6696 | 8.57% |
|          | RMSE   | 1.5259 | 1.015 | 0.9468 | 1.0408 | 0.8886 | 6.15% |

### TABLE 6. The results of other metrics obtained by performing different models on Douban dataset.

|          | Recall@20 | Recall@100 | NDCG@20 | NDCG@100 | HR@20 | HR@100 | AUC |
|----------|-----------|------------|---------|----------|-------|--------|-----|
| MF       | 0.0309    | 0.1455     | 0.0131  | 0.0442   | 0.0424 | 0.2109 | 0.5064 |
| HeRec    | 0.0335    | 0.1681     | 0.0150  | 0.0592   | 0.06   | 0.2733 | 0.5104 |
| NGCF     | 0.0510    | 0.2285     | 0.0219  | 0.0634   | 0.0739 | 0.2891 | 0.5586 |
| FHRecq   | 0.0508    | 0.2278     | 0.0238  | 0.0648   | 0.0789 | 0.2855 | 0.5858 |
| FHRec    | 0.0601    | 0.2723     | 0.0284  | 0.0756   | 0.0903 | 0.3309 | 0.6307 |
| Dataset | Groups   | Number of users | Number of items | Number of ratings | Density(%) |
|---------|----------|-----------------|-----------------|------------------|------------|
| Douban Movie | (0,5]    | 4567            | 647             | 6297             | 0.21%      |
|         | (5,15]   | 181             | 594             | 1605             | 1.49%      |
|         | (15,30]  | 48              | 509             | 1006             | 4.12%      |
|         | (30, ]   | 22              | 529             | 1198             | 10.29%     |
| Yelp1   | (0,5]    | 15819           | 10903           | 43457            | 0.025%     |
|         | (5,15]   | 3084            | 8397            | 25051            | 0.097%     |
|         | (15,30]  | 310             | 3786            | 6285             | 0.536%     |
|         | (30, ]   | 57              | 1885            | 2388             | 2.22%      |
| Yelp2   | (0,5]    | 0               | 0               | 0                | 0          |
|         | (5,15]   | 625             | 4412            | 8472             | 0.31%      |
|         | (15,30]  | 1140            | 8098            | 23439            | 0.254%     |
|         | (30, ]   | 335             | 6732            | 15940            | 0.709%     |

**FIGURE 7.** (a), (b) and (c) respectively show Recall, HR and NDCG with the different meta-paths on Yelp2 datasets.
(a) MAE and RMSE on Douban Movie dataset.

(b) MAE and RMSE on Yelp1 dataset.

(c) MAE and RMSE on Yelp2 dataset.

FIGURE 8. (a), (b) and (c) respectively show MAE and RMSE in different groups of three datasets.

(a) Varying the number of layers on Douban Movie.

(b) Varying the number of layers on Yelp1.

(c) Varying the number of layers on Yelp2.

FIGURE 9. (a), (b) and (c) respectively show RAE and RMSE with the change of the number of hidden layers on the three datasets.