Computing and Informatics, Vol. 45, 2026, 320–345, doi: 10.31577/cai_2026_2_320

RECOMMENDATION MODEL INTEGRATING SEMANTIC FEATURES AND KNOWLEDGE FEATURES

Guang Zheng

College of Information and Management Science
Henan Agricultural University, Zhengzhou
450046, Henan, China
&
Henan Engineering Laboratory of Farmland Monitoring and Control, Zhengzhou
450002, Henan, China
e-mail: zhengguang@henau.edu.cn

Demeng Zhang

College of Information and Management Science
Henan Agricultural University, Zhengzhou
450046, Henan, China
e-mail: rjxyrz@henau.edu.cn

Xinming Ma, Yanna Ren

College of Information and Management Science
Henan Agricultural University, Zhengzhou
450046, Henan, China
&
Henan Engineering Laboratory of Farmland Monitoring and Control, Zhengzhou
450002, Henan, China
e-mail: {maxinming, gshili}@henau.edu.cn

Chengzhong Sun

College of Information and Management Science
Henan Agricultural University, Zhengzhou
450046, Henan, China
e-mail: xinguanyb@henau.edu.cn

Lei Xi*

College of Information and Management Science
Henan Agricultural University, Zhengzhou
450046, Henan, China
&
Henan Engineering Laboratory of Farmland Monitoring and Control, Zhengzhou
450002, Henan, China
e-mail: xil@henau.edu.cn

Abstract. The traditional collaborative filtering recommendation model usually cannot capture the correlation between user items accurately when faced with the problem of data sparsity, which leads to a poor recommendation effect. To solve these problems, this paper proposed a recommendation model integrating semantic features and knowledge features. By introducing a knowledge graph as auxiliary information, the problem of data sparseness was alleviated, and the granularity of user interest description was refined based on the rich knowledge information. Different from other recommendation models based on knowledge graphs, the model in this paper mined implicit feedback information between user projects based on semantic features and knowledge features. The Word2vec model was used to capture the semantic features of project entities, and knowledge features were learned according to the idea of preference diffusion to obtain the user preference diffusion set integrating features of different levels and forming the user preference feature sequence from the inner layer to the outer layer. The sequence was adopted as the input of the gated cycle unit, and the attention mechanism was integrated to capture the deep feature information between the user and the item, so as to enrich the user representation to improve the recommendation effect. The comparison experiment of the CTR prediction scenario and the *Top-K* recommendaton scenario, and the sparseness verification experiment were all conducted on the MovieLens dataset. The results showed that, compared with BPR-MF, DKN and RippleNet baseline models, this model had better performance in both scenarios and could effectively alleviate the impact of data sparseness.

Keywords: Recommendation mode, knowledge graph, feature fusion, gated cycle unit, attention mechanism

* Corresponding author

1 INTRODUCTION

With the explosive growth of Internet data, a large amount of redundant information is presented to users irregularly and indiscriminately, which seriously reduces the efficiency of users obtaining information. Recommendation systems are one of the most effective ways to deal with "information overload". Different from the search engines that need to actively input search terms, a recommendation system could mine users' potential interests and construct user interest preference models by analyzing the interaction information between users and projects, so as to find items that meet their interest preferences. In recent years, recommendation systems have been widely used in music [1], film [2], news [3] and other fields. Recommendation methods could be divided into collaborative filtering based recommendation methods [4, 5], content-based recommendation methods [6], and mixed recommendation methods [7, 8]. Among them, the collaborative filtering recommendation method was the most widely used. The collaborative filtering method calculated the similarity between users and projects based on the historical user interaction data and generated the recommendation results. However, collaborative filtering excessively relied on user interaction data, and the accuracy of this method was low when the problem of data sparseness existed [9]. At the same time, new users and new items also had cold start issues.

In order to alleviate the problems in traditional recommendation methods, researchers introduced users' social relationships, related attributes of items, contextual information and knowledge map as auxiliary information for the recommendation system. In order to extract features from a large amount of data related to the user preferences contained in auxiliary information, researchers usually use deep learning technology to mine them to improve recommendation performance [10].

It has become a hot research topic to introduce a knowledge graph into the recommendation field as auxiliary information. The rich semantic information in the knowledge graph helped to explore the potential connections between users and projects, build a more accurate user interest model, and improve the accuracy of recommendation results. With the help of relational links in the knowledge graph, users' potential interests can be reasonably extended, diversified information can be provided to users, and homogenization of recommendation results can be avoided. The knowledge graph can serve as a bridge between the user's historical interaction information and the candidate recommendation items, and provide interpretability for the recommendation results, thus improving user satisfaction. Deep learning technology can extract user preferences and item features hidden in the knowledge graph, which can be directly applied to the construction of a user interest model.

Based on the above background, this paper introduced the knowledge graph into the recommendation model and proposed a type of recommendation model that integrated semantic features and knowledge features. The knowledge graph was used to effectively expand user interest, and the semantic features in the knowledge graph were fused with knowledge features to obtain more accurate user interests preference.

The network structure in the knowledge graph was used to obtain the user preference diffusion set according to the idea of "preference diffusion" [11], and the gated cyclic unit, combined with the attention mechanism, was used to effectively capture the user's deep interest and enhance the effectiveness of the recommendation.

2 RELATED WORK

2.1 Knowledge Graph

A knowledge graph is essentially a semantic network, which can describe knowledge structurally and has become a hot research topic at present [12]. The knowledge graph organizes the information on the Internet in a structured way and presents it as a form of knowledge that is easy to understand, which promotes the development of information retrieval and updates the traditional data service into an intelligent knowledge service. In recent years, the application of knowledge graphs to the recommendation field has been increasing. Some researchers introduced a knowledge graph to capture users' potential interest and improve the performance of the recommendation system. A large amount of entity set information contained in the knowledge graph can be applied to a recommendation system. For example, in the knowledge graph, the triplet (*Bullets Over Broadway*, *director*, *Woody Allen*) means that the director of *Bullets Over Broadway* is *Woody Allen*. In the recommended application scenario, *Woody Allen*, the director of both *Zelig* and *Hollywood Ending*, thinks that there was a connection between the movie *Bullets Over Broadway* and *Zelig* and *Hollywood Ending*. A person who watches *Bullets Over Broadway* will probably like the director's other movies just because they like him. The recommendation model can reasonably expand users' potential interests with the help of the relationship between entities in the knowledge graph, and also provide help for the explanation of the recommendation results, thus improving the interpretability of the recommendation results.

2.2 Knowledge Graph and Recommendation System

Recommendation methods based on a knowledge graph can be divided into two categories, including path-based method and feature-learning method [13].

1. Path-based methods [14] rely on links in the knowledge graph to expand knowledge and provide auxiliary information for the recommendation algorithm. Hete-MF [15] extracted multi-element paths and calculated the similarity between paths, and introduced a non-negative matrix decomposition method to improve the feature representation of a user projects. PER [16] comprehensively considered various types of relations in heterogeneous information networks, carried out feature extraction based on meta-path, and used Bayesian personalized sorting algorithm to generate high-quality recommendation results. RKGE [17] did not need to pre-define meta paths, and could directly mine features of paths

between user interaction items in the knowledge graph. KPRN [18] constructed path sequence representation based on the semantics of entities and relationships, and based on this path sequence, carried out effective inference on user preferences. The advantages of the above methods are that they can intuitively capture the network structure of the knowledge graph, but the disadvantages are that they are highly dependent on the predefined meta-path or meta-graph and the parameter tuning operation is cumbersome. In addition, such methods usually required a lot of domain knowledge to perform predefined work and it was difficult to extract the complete path.

2. The method based on feature learning [19] used the knowledge graph embedding technology to preprocess entities in the knowledge graph to generate low-dimensional embedding vectors. This embedding vector was then used to enrich the representation of the user or project. CKE [20] used translation distance model TransR [21] and autoencoder to conduct the character representation of projects, texts and images. After comprehensively considering the above three features and user interaction matrix, the candidate projects were generated. DKN [22] carried out multi-level feature mining for news items and then integrated the attention mechanism to obtain user feature representation, so as to capture users dynamic interest preference. MKR [23] used a joint learning and training method to associate and train the knowledge graph embedded module and recommendation module through the designed cross compression unit. Then, the knowledge transfer was carried out based on the cross-compression unit to compensate for information sparsity and improve the performance of the recommendation system. RCF [24] used DistMult [25] semantic matching model to capture relational semantics. Based on attention mechanism, RCF conducted modeling of user interest preference at both relationship type level and relationship weight level, and generated recommendation results through joint learning.

As auxiliary information in the recommendation system, the knowledge graph has the following advantages: The knowledge information contained in the knowledge graph can solve the problem of data sparseness to a certain extent. On the other hand, it can provide the explanation for the recommendation reason, which helps to improve the recommendation performance and optimize the user experience.

3 THE MODEL DESIGN

3.1 Problem Definition

In the traditional recommendation model, $\boldsymbol{U} = \{u_1, u_2, \dots\}$ and $\boldsymbol{V} = \{v_1, v_2, \dots\}$ were adopted to represent the set of users and projects, respectively. The definition of user-project interaction matrix was given as: $\boldsymbol{Y} = \{y_{uv} \mid u \in \boldsymbol{U}, v \in \boldsymbol{V}\}$, as shown in Equation (1).

$$y_{uv} = \begin{cases} 1, & \text{if } interaction(u, v) \text{ is observed,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In Equation (1), $y_{uv} = 1$ represents the implicit interaction between the user and the project, such as sharing, clicking, purchasing, etc.

Knowledge graph $\boldsymbol{G}$ consisted of the triplet of "head entity, relation entity and tail entity", namely $(h, r, t) \in \boldsymbol{G}$, which contained a large amount of knowledge information.

The issues studied in this paper were defined as: given the user-project interaction matrix $\boldsymbol{Y}$ and the knowledge graph $\boldsymbol{G}$, predict the probability that user u will be interested in the uninteractive candidate project. At this point, the prediction function was defined as $\hat{y}_{uv} = F(u, v \mid \Theta, \boldsymbol{G})$, where $\hat{y}_{uv}$ represented the interaction probability between user u and candidate project v, and Θ represented the model parameter that could be learned.

3.2 Recommendation Method

This paper introduced knowledge graph into recommendation model and proposed a recommendation model (Integrating Sematic Feature and Knowledge Feature for Recommendation Model Based on Gated Recurrent Unit with Attention Mechanism, *SKRec-AttGRU*) integrating Sematic Feature and Knowledge Feature based on gated cyclic unit combining attention mechanism and integrating semantic and knowledge features. Based on the feature learning of semantic information and knowledge information contained in the knowledge graph, the gated cyclic unit fusion attention mechanism was introduced to extract the feature, so as to optimize the recommendation effect. The overall framework of the *SKRec-AttGRU* model was shown in Figure 1. The first part of Figure 1 was a semantic feature extraction module based on *CBoW* model to learn semantic features of project entities. In the middle of the figure, the propagation of user interest in the knowledge graph was obtained with the help of the idea of "preference diffusion" [11], so as to extract the knowledge features contained in the knowledge graph. After learning semantic features and knowledge features, the feature fusion strategy was used to fuse them and maintain their corresponding relationship. The gated cyclic unit fusion attention mechanism was used to mine the potential information between users and projects to generate ultimate user feature representation and output the predicted user click probability.

3.2.1 Semantic Feature Learning

In order to enrich the representation of user preferences, project descriptions containing rich semantic information were introduced to learn semantic features of the description of item entities in the knowledge graph, so as to alleviate the problem of data sparsity. As shown in Figure 1, semantic feature learning module was used

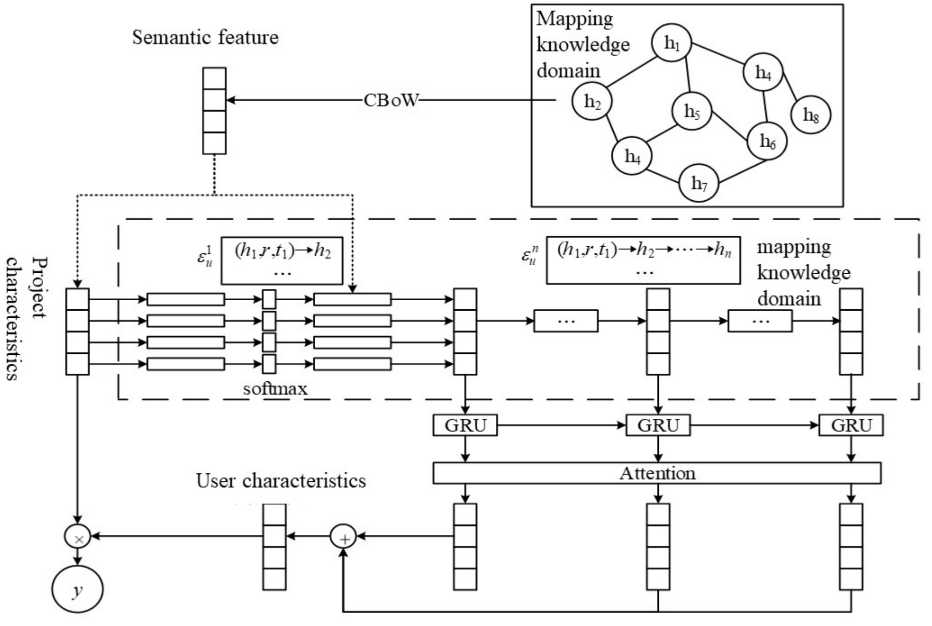


Figure 1. The overall framework of *SKRec-AttGRU*

in this paper to extract features of project entities and represent them as potential semantic features of corresponding projects. The *Word2vec* model was used to feature learn semantic information corresponding to project entities to obtain semantic feature representation. *Word2vec* was a commonly used semantic feature learning method in natural language processing tasks. It was trained by logistic regression to obtain low-dimensional embedding vector representation. The *Continuous Bag-of-Words (CBoW)* model was used in this paper. *CBoW* model trained the input text through shallow neural network. During the training process, the current word was predicted based on the context information so as to obtain the low dimensional vector representation corresponding to the current character with semantic level information.

The first layer of *CBoW* model was the input layer, which took k context words before and after the target word as input. The second layer was the projection layer, which was calculated by using the unique thermal representation of $2k$ context words and the common weight matrix. After extracting the corresponding word vectors, the weighted sum and average operation is carried out to obtain the projection layer vector representation h. The third layer was the output layer, and the probability distribution of all word vectors was obtained by softmax function. After the model iterative training process of the gradient descent method, the vector representation of the target was obtained, as shown in Formula (2), where U^T represented the weight matrix from the projection layer to the output layer.

$$y_c = \mathrm{softmax}(U^T \cdot h). \tag{2}$$

The semantic feature vectors were learned by maximizing the objective function L of the *CBoW* model. The objective function L was defined as shown in Formula (3), where *Context(w)* was the vector representation of context words.

$$L = \sum_{w \in C} \log p(w \mid \mathrm{Context}(w)). \tag{3}$$

After word segmentation and stop word filtering of the original text, the *CBoW* model was used to extract the semantic features of the project entity, so as to obtain the semantic feature representation S_i of its corresponding entity. Subsequently, knowledge feature mining would be carried out after the integration of the project entity representation, so as to obtain the deep representation of the project entity and improve the accuracy of recommendation results.

3.2.2 Knowledge Characteristic Learning

There was a lot of hidden knowledge information related to the project in the knowledge graph, and the relationship path in the knowledge graph could be effectively translated. In the recommendation model, users' potential interests could be mined with the help of the relational path in the knowledge graph. For example, users' preference for a project with behavioral interaction could be transmitted to another project entity through the relational path, which was also known as "preference propagation". Based on the idea of "preference propagation", the knowledge features corresponding to user interests were obtained through triples. The set of the tail nodes reached by user history interaction items propagated along the link was called the user interest preference diffusion set. The influence of link length in the knowledge graph on the effect of user interest transmission has been verified by researchers [11]. Similar to the phenomenon of ripple attenuation in water, with the increase of link length, the correlation between user potential preference mined based on the knowledge graph and user gradually weakens, and may bring more noise data, affecting the accuracy of user interest preference description. In the experiment, the influence of link length on the recommendation effect will be explored to ensure a relatively high correlation strength between remote entities and users when mining users' potential preferences in the knowledge graph.

Based on the idea of "preference propagation", this paper constructed the user preference diffusion set to obtain the representation of user interest at the level of knowledge characteristics. Based on the given knowledge graph $\boldsymbol{G}$ and user-item interaction matrix $\boldsymbol{Y}$, the user preference diffusion set, that is, the set of related entities from user u's historical interaction item with k jumping distance in the knowledge graph, was defined as $\boldsymbol{\varepsilon_u^k} = \left\{t \mid (h,r,t) \in \boldsymbol{G} \text{ and } h \in \boldsymbol{\varepsilon_u^{k-1}}\right\}$, $k =$

$1, 2, 3, \ldots$ When $k = 0$, $\boldsymbol{\varepsilon}_{\boldsymbol{u}}^{\mathbf{0}} = \boldsymbol{Y}_{\boldsymbol{u}} = \{y \mid y_{uv} = 1\}$ represented the collection of user history interaction items. In the knowledge graph, the user s history interaction items were taken as the starting point to spread out along the relationship link, and the user preference diffusion set was finally obtained, where M represented the link length.

In the knowledge graph, each item v had a corresponding vector representation, which was represented by $v \in \boldsymbol{R}^d$. As shown in the knowledge feature module in Figure 1, corresponding association probability was assigned to the triplet in the preference diffusion set according to user interaction items, as shown in Formula (4).

$$p_i = \mathrm{softmax}(v^T R_i h_i) = \frac{\exp(v^T R_i h_i)}{\sum_{(h,r,t)\in\boldsymbol{\varepsilon}_{\boldsymbol{u}}} \exp(v^T R_i h_i)}, \tag{4}$$

where v was the user history interaction project, $R_i \in \mathbb{R}^{d\times d}$ and $h_i \in \mathbb{R}^d$ were the vector representation of relations and entities in the knowledge graph, respectively.

3.2.3 Feature Fusion and Result Generation

In order to describe the representation of a user interest preference in a more fine-grained way, this paper proposed a recommendation method that fused semantic features and knowledge features to mine user's potential interest preference. The learned semantic features and knowledge features are integrated. The weight of the relation was designed for different features and weighted fusion was carried out. On this basis, the user's response to the preference diffusion set was calculated, as shown in Formula (5). In the formula, o_u^n was the entity response of user u to preference diffusion set $\boldsymbol{\varepsilon}_{\boldsymbol{u}}^{\boldsymbol{n}}$, which was obtained by weighted fusion of semantic features and knowledge features of this node; t_i represented the tail entity reached by entity h_i along relation r_i; p_i was the correlation degree learned in the knowledge feature learning module; α_i represented the weight of semantic feature S_i corresponding to tail entity t_i.

$$o_u^n = \sum_{(h_i,r_i,t_i)\in\boldsymbol{\varepsilon}_{\boldsymbol{u}}^{\boldsymbol{n}}} (p_i t_i + \alpha_i S_i). \tag{5}$$

After obtaining the entity response of user u under the preference diffusion set, the preference feature sequence from inner layer to outer layer was formed. Based on the feature sequence, the *Gated Recurrent Unit* (*GRU*) structure was introduced to extract deep features and the attention mechanism was incorporated to assign weight, so as to capture the more important feature representation of users in the diffusion set of user preferences and refine the granularity of user interest description. As a variant of recurrent neural network, the gate structure was used to alleviate the long-term dependency problem faced by recurrent neural network [26]. The model framework of *GRU* was shown in Figure 2.

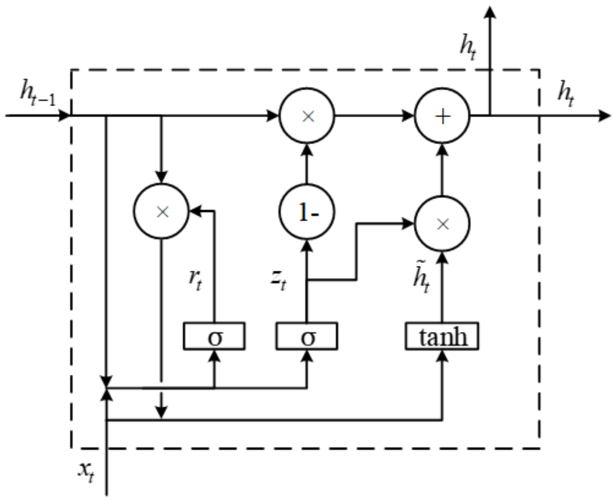


Figure 2. Model structure of *GRU*

The *GRU* used reset gate r_t control information forgetting degree. The update gate z_t was used to forget and selectively retain the state information hidden at the previous moment. The candidate hidden state $\tilde{h}_t$ was calculated based on the activation result of the update gate. On this basis, the hidden state of the current moment was obtained by calculation, and the calculation process was shown in Equations (6), (7), (8) and (9).

$$r_t = \sigma(W_r X_t + U_r h_{t-1}), \tag{6}$$

$$z_t = \sigma(W_z X_t + U_z h_{t-1}), \tag{7}$$

$$\tilde{h}_t = \tanh(W_h X_t + U_h \odot (r_t \odot h_{t-1})), \tag{8}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \tag{9}$$

In these formulas, σ represented the Sigmoid activation function. W_r and U_r represented the weight parameters of the reset gate. W_z and U_r represented the weight parameters of the update gate. The parameter $\odot$ standed for *Hadamard* product.

Considering that the user preference intensity decreased with the increase of link length in the preference diffusion set ε_u, entities farther away were less important for the representation of user interest preference. Therefore, the obtained entity responses of user u under different link lengths in the knowledge graph were input to *GRU* in the order from inner layer to outer layer. In order to extract deep features from the preference diffusion set, the attention mechanism was adopted to assign weights to feature vectors extracted from the preference diffusion set, and then generated the final vector representation of users, as shown in Equation (10).

$$u = \sum_{j=1}^{n} \theta_{jn} h_u^n. \tag{10}$$

In this formula, n represented the maximum link length of preferred diffusion set ε_u^n diffusion; h_u^n was the hidden state vector representation of user u's entity response o_u^n in *GRU* unit to preference diffusion set ε_u^n; θ_{jn} was the weight corresponding to the current j^{th} hidden state vector.

The weighted feature fusion strategy fused the acquired semantic features with knowledge features to obtain a more comprehensive representation of users' potential interests. The optimal weight was determined by minimizing the loss function based on gradient descent algorithm. The weighted fusion operation could extract effective information from different features and reduce redundant information, so as to avoid the gradient descent problem of high-dimensional features. Finally, the final vector representation of users and the vector representation of candidate projects were combined to obtain the predicted user click probability through calculation, as shown in Formula (11).

$$\hat{y}_{uv} = \sigma\left(u^T v\right). \tag{11}$$

In this formula, σ represented the *Sigmoid* function $f(x) = \frac{1}{1+e^{-x}}$.

3.2.4 Model Training

In order to obtain the predicted value closest to the real value, cross entropy was used as the loss function for model training, as shown in Equation (12).

$$L_{pre} = -\sum_{y_{uv} \in Y} \left[y_{uv} lg \hat{y}_{uv} + (1 - y_{uv}) lg (1 - \hat{y}_{uv}) \right]. \tag{12}$$

In this formula, $\hat{y}_{uv}$ represented the predicted probability distribution, and y_{uv} represented the true probability distribution. After comprehensive consideration of knowledge graph embedding loss and feature fusion for joint learning, the loss function was finally obtained, as shown in Equation (13). The function was composed of three parts: the first part was the cross entropy loss function to evaluate the difference between the predicted probability distribution and the real probability distribution. The second part represented the loss of knowledge graph embedding. The third part was the L_2 regularization to prevent the model from over-fitting.

$$\text{Loss} = L_{pre} + \lambda_1 L_{KG} + \lambda_2 \parallel \Theta \parallel_2^2. \tag{13}$$

According to the predicted click probability obtained by the above algorithm, a *Top-K* recommendation list of target users was generated.

4 EXPERIMENTS AND ANALYSIS

To verify the recommended performance of the proposed model, in a Windows server 2016 server with 16 GB memory and Intel (R) Xeon (R) CPU e5-2683 v4 @ 2.10 GHz, based on TensorFlow deep learning framework, a comparative experiment was designed in Python 3.6 environments for performance analysis. The experiment was based on *MovieLens*, a common evaluation dataset in the academic world. Parameters were adjusted based on the dataset to achieve the optimal effect of the model. Experimental comparison with the selected baseline model was conducted to verify the superiority of the proposed model. The performance of each model in datasets with different sparsity was explored, and the results proved that the constructed model could effectively alleviate the data sparsity problem faced by traditional recommendation models.

4.1 Dataset

MovieLens, an open dataset, was selected for the experiment in this paper. This dataset was a commonly used dataset in the effect evaluation experiment of recommendation model, which contained users' ratings of thousands of movies. The purpose of this experiment was to predict the probability of users' interest in non-interactive items, so implicit interaction data was more suitable for this experiment. Therefore, explicit feedback data in this dataset should be converted into implicit feedback, and all data with scores greater than or equal to the threshold should be taken as positive samples by setting the scoring threshold.

The knowledge graph data used in this paper came from the public data in *IMDb*. According to the mapping of the URL corresponding to the movie project in the *MovieLens* dataset in the Internet Movie Database, the director, screenwriter, film introduction and other information corresponding to the movie were obtained to construct the movie knowledge graph.

DataSet	MovieLens
Number of users	610
Sample size	9742
Number of scores	100 836
Rating category	0.5, 1, ..., 5
Number of cast and crew relationships	52 904
Number of movie category and relationships	24 226
Film Introduction Number	9 742

Table 1. Information of dataset

Table 1 shows the basic information of the dataset. In the experiment, the dataset is divided into training set and test set in a ratio of 4:1, and the average result of multiple experiments is taken as the final result.

The movie dataset *MovieLens* was selected in the experiment. Considering that the movie introduction was an important reference for users in the process of selecting movies, and it contained rich information to help explore the potential interests of users, the semantic feature learning was carried out based on the movie introduction corresponding to the movie project entity.

4.2 Evaluation Index

The experiment considered two recommendation scenarios: *click rate prediction* and *Top-K* recommendation. In the scenario of CTR (Click-Through Rate) prediction, *AUC* and *Accuracy* (Acc) were selected as evaluation indexes. In the *Top-K* recommendation scenario, recall rate *Recall@K* and accuracy rate *Precision@K* were selected as evaluation indexes to evaluate the quality of the generated recommendation list. In order to explain the evaluation metrics more clearly, a confusion matrix, as shown in Table 2, was constructed.

	Real Positive Example	Real Negative Example
Prediction is a positive example	TP (True Positive)	FP (False Positive)
Prediction is a negative example	FN (False Negative)	TN (True Negative)

Table 2. Confusion matrix

1. CTR prediction scenario

   The experiment analyzed the effect of the recommendation model by predicting the click probability of the project. *AUC (Area Under Curve)* index and *Acc (Accuracy)* were selected as indicators to reflect the accuracy of the model. The calculation formulas were shown in Formula (14) and Formula (15). *AUC* referred to the area under the *ROC (Receiver Operating Characteristic)* curve. *AUC* referred to the probability that positive samples were ranked before negative samples according to the prediction. The accuracy rate represented the proportion of all correctly predicted samples in the total.

$$AUC = \frac{\sum_{ins_i \in Pos} rank_{ins_i} - \frac{M \times (M+1)}{2}}{M \times N}, \tag{14}$$

$$Acc = \frac{TP + TN}{P + N}. \tag{15}$$

2. *Top-k* Recommended scenario

   The experiment analyzed the recommended performance by generating a list of k items for the user. Recall rate *Recall@K* and accuracy rate *Precision@K* were used as indicators to measure the recommended list, and the calculation formulas were shown in Equations (16) and (17). Recall rate *Recall@K* represented the proportion of items favored by users that would be recommended by the recommended model. Accuracy *Precision@K* represented the percentage of all items recommended by the model that users like.

$$Recall@K = \frac{TP}{TP + FN}, \tag{16}$$

$$Precision@K = \frac{TP}{TP + FP}. \tag{17}$$

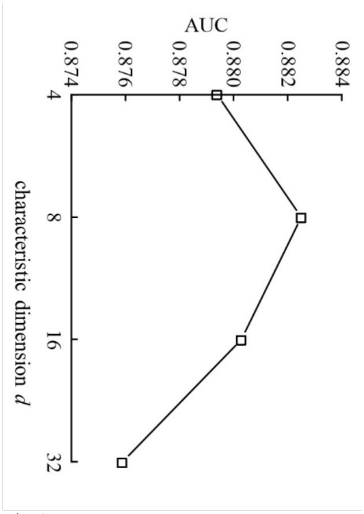
4.3 Parameter Setting and Comparative Analysis

In this section, specific parameter settings in the experiment were discussed and analyzed to obtain the best parameters for achieving the optimal effect of the model. AUC was used as an evaluation index to obtain experimental results. Where, d represented the size of feature dimension, k represented the maximum link length during preference propagation, m represented the size of preference diffusion set at each layer, λ_1 represented the weight of knowledge graph embedding loss and λ_2 represented the weight of L_2 regular term.

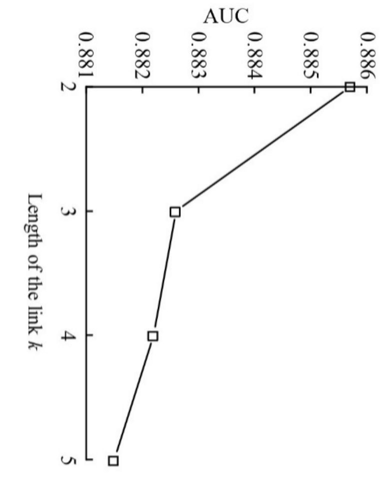
As the value of feature dimension d had a certain influence on the recommendation effect of the model, $d = 4, 8, 16, 32, 64$ was set for the experiment to determine the size of the optimal feature dimension. The results are shown in Figure 3 a). It could be observed that the recommendation effect reached the best when $d = 8$. At first, AUC increased with the increase of dimension. However, when the dimension increased to a certain extent and brought too much information, it was easy to overfit, leading to the decline of AUC index.

As the value of link length k had a certain influence on the recommendation effect of the model, $k = 2, 3, 4, 5$ was set in the experiment to find the optimal link length, and the experimental results were shown in Figure 3 b). The results showed that the recommendation effect was optimal when the link length $k = 2$. With the increase of the link length, the association between the remote entity and the user was gradually weakened, and a higher proportion of noise data would be mined. If the link length was too long, too many invalid knowledge features would be mined, and if the link length was too short, the long-distance dependencies in the knowledge graph could not be explored.

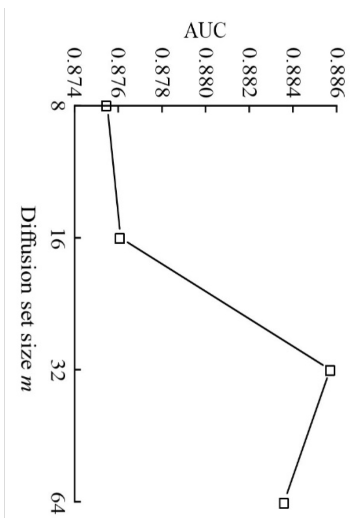
Since the size m of the preference diffusion set of each layer had a certain influence on the recommendation effect of the model, the optimal value was determined by setting $m = 8, 16, 32, 64$ in the experiment, and the results were shown in Figure 3 c). The results showed that when $m = 32$, the recommendation effect reached the best. Initially, with the increasing size of preference diffusion set, AUC increased; however, when it increased to a certain extent, it would bring a higher proportion of features with low impact on the current central entity, resulting in poor recommendation effect.



a) AUC under different characteristic dimension d



b) AUC under different Lenght k

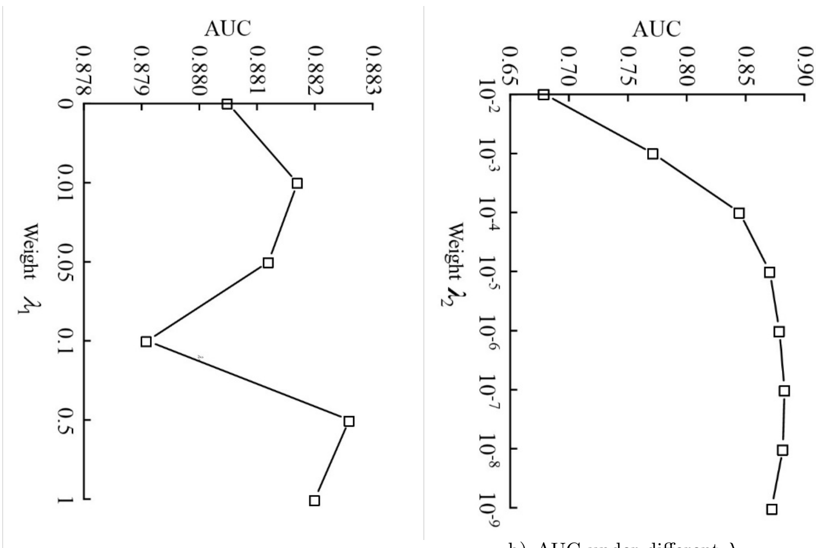


c) AUC under different diffusion set size m

Figure 3. AUC of different model parameters

In order to determine the weight of the knowledge graph embedding loss weight λ_1 and L_2 regular term weight λ_2 in the loss function, the experiment was set to adjust parameters to seek the best effect. The experimental results were shown in Figure 4. The results showed that the model had the best recommendation effect when the weight of knowledge graph embedding loss $\lambda_1 = 0.5$ and the weight of L_2 regular term $\lambda_2 = 10^{-7}$.

In summary, when the experimental parameters were set as feature dimension $d = 8$, layer numbers of diffusion preference set $h = 2$, layer size of diffusion preference set $m = 32$, knowledge graph embedding loss weight $\lambda_1 = 0.5$, and L_2 regular term weight $\lambda_2 = 10^{-7}$, the best results could be obtained.



a) AUC under different λ_1

b) AUC under different λ_2

Figure 4. AUC of different model weights

4.4 Experimental Results and Analysis

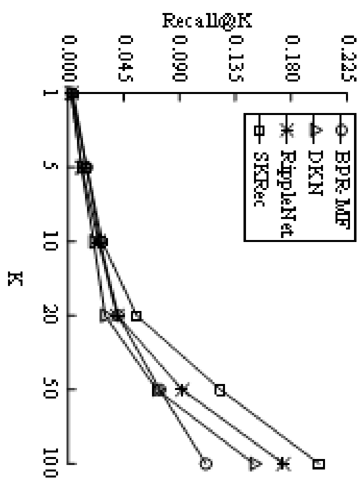
In order to explore the performance of the recommendation model combining semantic features and knowledge features proposed in this paper in CTR prediction scenarios and *Top-K* recommendation scenarios, the following recommendation models were selected as baseline models for comparative experiments.

1. *BPR-MF*: considering that users have different preferences for different projects, Bayesian personalized ranking was introduced to model users based on matrix decomposition.

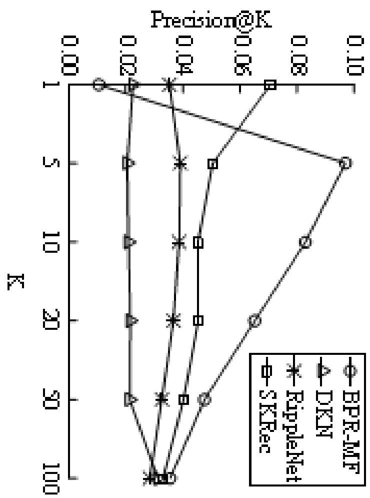
2. *DKN*: feature representation of three levels of entities was extracted based on knowledge graph, and item representation was obtained by feature fusion based on convolutional neural network to make recommendations for users.
3. *RippleNet*: based on the idea of "preference diffusion", the user preference diffusion set was obtained and the user preference model was constructed. In the CTR prediction experiment, the AUC and Acc performances of each model on the dataset were compared, and the results were shown in Table 3. In the *Top-K* recommendation scenario experiment, $K = 1, 5, 10, 20, 50, 100$ was set to compare the recall rate and accuracy rate of each model on the dataset. The results were shown in Figure 5.

	AUC	Acc
BPR-MF	0.7137	0.6396
DKN	0.7925	0.7268
RippleNet	0.8682	0.7999
SKRec	0.8857	0.8131

Table 3. Experimental results of models on CTR



a) Recall rate at different K values



b) Accuracy at different K values

Figure 5. Experimental results of models on Top-K

Based on the above experimental results, the following conclusions can be drawn by comprehensively considering the performance of each model in the two scenarios.

1. In the experiment, the overall effect of *BPR-MF* was slightly inferior to that of other baseline models based on knowledge graph, indicating that knowledge

graph could be used as auxiliary information to optimize the recommendation result, and the rich inter-entity relations in the graph could help reasonably expand user interests, thus improving the recommendation effect.

2. *SKRec-AttGRU* performed better than *RippleNet* in CTR prediction and *Top-K* recommendation scenarios. *RippleNet* did not conduct deep feature mining after acquiring preference diffusion set. However, the model proposed in this paper used gated loop unit combined with attention mechanism to conduct deep feature mining for user preference, thus optimizing user interest expression. The results showed that the gated loop structure with the integrated attention mechanism is helpful to further capture the potential interest of users.
3. *BPR-MF* performed better in the accuracy rate in the *Top-K* recommendation scenario, while recall rate and accuracy rate were a pair of contradictory indicators, so it was necessary to ensure the relative balance of the two indicators. Based on the performance of the recall rate and the CTR prediction scenario of this model, it could be seen that the actual recommendation effect of this model was average.
4. Considering the experimental results of the two scenarios comprehensively, it could be found that *SKRec-AttGRU* had the best recommendation effect, because *SKRec-AttGRU* fused semantic embedding features and knowledge embedding features based on knowledge graph, and used the gated cyclic unit structure fusion attention mechanism to mine user preference diffusion set, which made the modeling of user interest preference more fine-grained and helped to improve the recommendation effect.

Experimental results showed that the proposed recommendation model could better capture the potential information in the knowledge graph to improve the recommendation effect, which proved the superiority of the proposed model.

4.5 Ablation Experiments

In order to verify the prediction effect of semantic feature extraction module, knowledge feature extraction module and feature extraction module based on *GRU* combined with attention mechanism, a variant model was designed for ablation experiment.

1. Exacting Knowledge Feature for Recommendation Model (*KRec*): This model only considered knowledge features learned based on knowledge graph, and did not consider semantic features and did not use *GRU* for deep feature extraction.
2. Integrating Sematic Feature and Knowledge Feature for Recommendation Model (*SKRec*): In this model, semantic features and knowledge features were acquired and fused based on knowledge graph. *GRU* is not used for deep feature extraction.

3. Exacting Knowledge Feature for Recommendation Model Based on Gated Recurrent Unit (*KRec-GRU*): In this model, *GRU* was used for deep feature extraction after knowledge feature learning based on knowledge graph, without considering semantic features corresponding to items.
4. Integrating Sematic Feature and Knowledge Feature for Recommendation Model Based on Gated Recurrent Unit (*SKRec-GRU*): This model learned semantic features and knowledge features based on knowledge graph and fuses them. On this basis, *GRU* was used for deep feature extraction, and the *GRU* was not combined with attention mechanism.
5. *SKRec-AttGRU*: In this model, semantic features and knowledge features were acquired based on knowledge graph, and then *GRU* combined with attention mechanism was used to extract deep features.

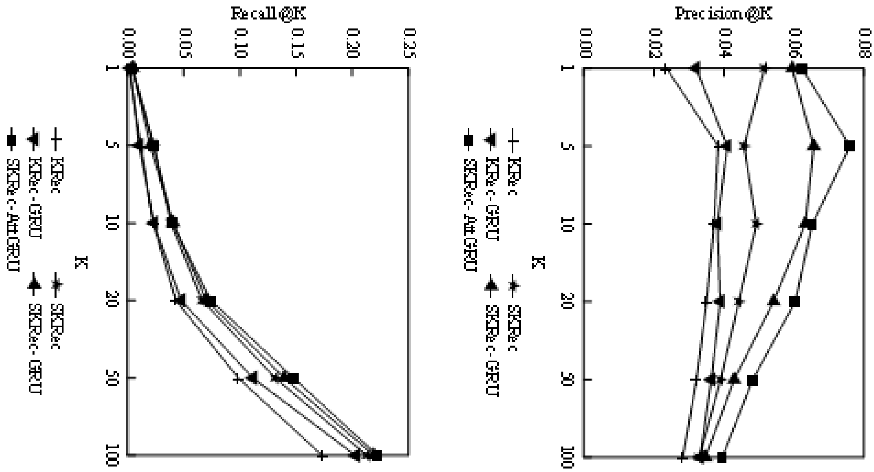
A comparative experiment was conducted in the CTR prediction scenario and *Top-K* recommendation scenario, and the results were shown in Table 4 and Figure 6.

Model	AUC	Acc
KRec	0.8656	0.7946
SKRec	0.8791	0.8086
KRec-GRU	0.8784	0.8060
SKRec-GRU	0.8829	0.8091
SKRec-AttGRU	0.8857	0.8131

Table 4. CTR prediction results under different characteristics

It could be concluded from the experimental results:

1. The recommendation effect of *SKRec* model was better than that of *KRec* model, indicating that the introduction of semantic feature module based on knowledge graph extraction could alleviate the problem of data sparsity to a certain extent, and this module played an important role in the prediction of recommendation results.
2. The recommendation effect of *KRec-GRU* model was better than that of *KRec* model, indicating that the deep feature extraction module based on *GRU* was helpful to build a more accurate user interest model and could effectively improve the recommendation effect.
3. The recommendation effect of the *SKRec-AttGRU* model was better than that of *SKRec-GRU* model without the attention mechanism, indicating that *GRU* combined with the attention mechanism could enhance the extraction effect of the key information in a user preference diffusion set.
4. The recommendation effect of the *SKRec-AttGRU* model was improved compared with the other four variant models, indicating the importance of introducing semantic features on the basis of the knowledge feature extraction based



a) Recall rate b) Accurate rate

Figure 6. Top-k recommendation results under different features

on knowledge graph and using *GRU* combined with the attention mechanism to carry out a deep feature extraction module.

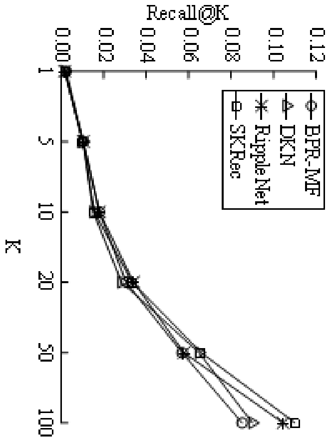
In conclusion, the *SKRec-AttGRU* model was superior to the other four variant models. After integrating semantic features and knowledge features, deep feature extraction based on *GRU* combined with attention mechanism could effectively capture users' potential interest preferences and improve the recommendation effect.

4.6 Sparsity Verification

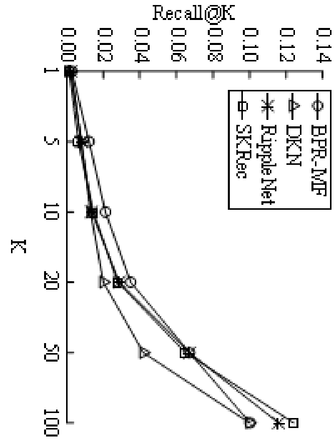
In order to verify the performance of the proposed recommendation model in alleviating the problem of data sparsity, three different data subsets were set up for sparsity verification experiments. 20 %, 50 % and 80 % user-project interaction data were randomly selected from the original dataset to form data subsets 1–3 (Set1–Set3) with different data densities for comparative experiments respectively. Set $K = 1, 5, 10, 20, 50, 100$ to compare recall rates of each model under different sparsity datasets. The experimental results were shown in Figure 7.

The following conclusions could be drawn from the experimental results in the figure:

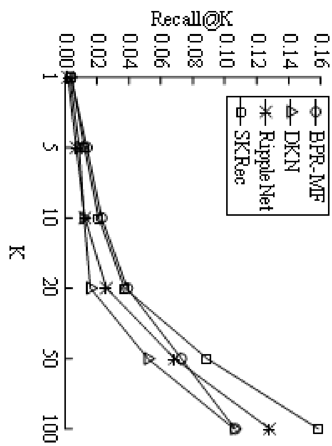
1. The recall rate of each model increased with the increase in data density in the data subset, which indicated that the sparsity of user interaction data was an important factor affecting the benefit of the recommendation model. The higher the data density, the more detailed the user interaction data acquired, the richer



a) Change in recall rate on Set1



b) Change in recall rate on Set2



c) Change in recall rate on Set3

Figure 7. AUC of datasets with different sparsity

the user interest characteristic information learned by the model, and the higher the quality of the recommendation results.

2. The recall rate of *SKRec-AttGRU* and *RippletNet*, which also used knowledge graph as auxiliary information to model user interest, was higher than that of other models in the case of extremely sparse data, indicating that the recommendation model integrating knowledge graph could better alleviate the problem of data sparsity and improve the recommendation performance.
3. *SKRec-AttGRU* performed well on the data subset *set1* with relatively sparse data, which proved that the model could maintain good performance in the case of extremely sparse user-project interaction data, and proved its superiority.

In conclusion, compared with the baseline model, the proposed model could maintain a better recommendation effect in the case of relatively sparse data, proving that the model could effectively alleviate the data sparsity problem.

5 CONCLUSIONS

In order to alleviate the problems of data sparsity and the large granularity of user data interest description in the current recommendation methods, a recommendation model combining semantic features and knowledge features was proposed in this paper. Based on the implicit feedback information between user projects, the semantic features and knowledge features contained in the connected knowledge graph were mined and fused, and the gated loop unit of the fused attention mechanism was used to mine users' deep interests, so as to provide high-quality personalized recommendation results for users. The experimental results showed that, compared with the baseline models *BPR-MF*, *DKN* and *RippleNet*, the proposed model could effectively improve the recommendation effect and alleviate the problem of data sparsity. However, this paper only considered semantic features and knowledge feature learning when constructing the user interest model. In the next step, the research will try to integrate pictures or videos and other multi-modal information to further explore users' potential interests, so as to improve the accuracy of the recommendation model.

Acknowledgements

This paper was sponsored by the Henan Province Science and Technology Development Plan Project (No. 242102521027) and the Henan Province key research and development project of China (No. 251111211300).

REFERENCES

[1] Hansen, C.—Hansen, C.—Maystre, L.—Mehrotra, R.—Brost, B.—Tomasi, F.—Lalmas, M.: Contextual and Sequential User Embeddings for Large-Scale Music Recommendation. Proceedings of the 14th ACM Conference on Recommender Systems (RecSys '20), 2020, pp. 53–62, doi: 10.1145/3383313.3412248.

[2] Zhang, J.—Wang, Y.—Yuan, Z.—Jin, Q.: Personalized Real-Time Movie Recommendation System: Practical Prototype and Evaluation. Tsinghua Science and Technology, Vol. 25, 2019, No. 2, pp. 180–191, doi: 10.26599/TST.2018.9010118.

[3] An, M.—Wu, F.—Wu, C.—Zhang, K.—Liu, Z.—Xie, X.: Neural News Recommendation with Long- and Short-Term User Representations. In: Korhonen, A., Traum, D., Màrquez, L. (Eds.): Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019). 2019, pp. 336–345, doi: 10.18653/v1/P19-1033.

[4] Xue, F.—He, X.—Wang, X.—Xu, J.—Liu, K.—Hong, R.: Deep Item-Based Collaborative Filtering for Top-N Recommendation. ACM Transactions on Information Systems (TOIS), Vol. 37, 2019, No. 3, Art. No. 33, doi: 10.1145/3314578.

[5] Lytvyn, V.—Vysotska, V.—Shatskykh, V.—Kohut, I.—Petruchenko, O.—Dzyubyk, L.—Bobrivetc, V.—Panasyuk, V.—Sachenko, S.—Komar, M.: Design of a Recommendation System Based on Collaborative Filtering and Machine Learning Considering Personal Needs of the User. Eastern-European Journal of Enterprise Technologies, Vol. 4, 2019, No. 2, pp. 6–28, doi: 10.15587/1729-4061.2019.175507.

[6] Trinh, T.—Wu, D.—Wang, R.—Huang, J. Z.: An Effective Content-Based Event Recommendation Model. Multimedia Tools and Applications, Vol. 80, 2021, No. 11, pp. 16599–16618, doi: 10.1007/s11042-020-08884-9.

[7] Gatzioura, A.—Vinagre, J.—Jorge, A. M.—Sanchez-Marrè, M.: A Hybrid Recommender System for Improving Automatic Playlist Continuation. IEEE Transactions on Knowledge and Data Engineering, Vol. 33, 2021, No. 5, pp. 1819–1830, doi: 10.1109/TKDE.2019.2952099.

[8] Barkan, O.—Koenigstein, N.—Yogev, E.—Katz, O.: CB2CF: A Neural Multiview Content-to-Collaborative Filtering Model for Completely Cold Item Recommendations. Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19), 2019, pp. 228–236, doi: 10.1145/3298689.3347038.

[9] Neysiani, B. S.—Soltani, N.—Mofidi, R.—Nadimi-Shahraki, M. H.: Improving Performance of Association Rule-Based Collaborative Filtering Recommendation Systems Using Genetic Algorithm. International Journal of Information Technology and Computer Science (IJITCS), Vol. 11, 2019, No. 2, pp. 48–55, doi: 10.5815/ijitcs.2019.02.06.

[10] Yu, S.—Wang, Y.—Yang, M.—Li, B.—Qu, Q.—Shen, J.: NAIRS: A Neural Attentive Interpretable Recommendation System. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), 2019, pp. 790–793, doi: 10.1145/3289600.3290609.

[11] Chen, P.—Wang, Q.—Tian, Y.: Exploring Entity-Level User Preference on the Knowledge Graph for Recommender System. Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI '22), 2022, doi: 10.1145/3579654.3579701.

[12] Fensel, D.—Şimşek, U.—Angele, K.—Huaman, E.—Kärle, E.—Panasiuk, O.—Toma, I.—Umbrich, J.—Wahler, A.: Introduction: What Is a Knowledge Graph? Knowledge Graphs: Methodology, Tools and Selected Use Cases, Springer, Cham, 2020, pp. 1–10, doi: 10.1007/978-3-030-37439-6_1.

[13] Cheng, S. Y.—Huang, S. H.—Yin, J.: Recommendation Model Based on Knowledge Graph and Recurrent Neural Network. Journal of Chinese Computer Systems, Vol. 41, 2020, No. 8, pp. 1670–1675 (in Chinese).

[14] Hu, B.—Shi, C.—Zhao, W. X.—Yu, P. S.: Leveraging Meta-Path Based Context for Top-N Recommendation with a Neural Co-Attention Model. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18), 2018, pp. 1531–1540, doi: 10.1145/3219819.3219965.

[15] Yu, X.—Ren, X.—Gu, Q.—Sun, Y.—Han, J.: Collaborative Filtering with Entity Similarity Regularization in Heterogeneous Information Networks. 23rd International Joint Conference on Artificial Intelligence (IJCAI'13), Workshop on Heterogeneous Information Network Analysis (HINA'13), Vol. 27, 2013, pp. 1–6.

[16] Yu, X.—Ren, X.—Sun, Y.—Gu, Q.—Sturt, B.—Khandelwal, U.—Norick, B.—Han, J.: Personalized Entity Recommendation: A Heterogeneous Information Network Approach. Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM '14), 2014, pp. 283–292, doi: 10.1145/2556195.2556259.

[17] Sun, Z.—Yang, J.—Zhang, J.—Bozzon, A.—Huang, L. K.—Xu, C.: Recurrent Knowledge Graph Embedding for Effective Recommendation. Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18), 2018, pp. 297–305, doi: 10.1145/3240323.3240361.

[18] Wang, X.—Wang, D.—Xu, C.—He, X.—Cao, Y.—Chua, T. S.: Explainable Reasoning over Knowledge Graphs for Recommendation. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, No. 1, pp. 5329–5336, doi: 10.1609/aaai.v33i01.33015329.

[19] Huang, J.—Zhao, W. X.—Dou, H.—Wen, J. R.—Chang, E. Y.: Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18), 2018, pp. 505–514, doi: 10.1145/3209978.3210017.

[20] Zhang, F.—Yuan, N. J.—Lian, D.—Xie, X.—Ma, W. Y.: Collaborative Knowledge Base Embedding for Recommender Systems. Proceedings of the 22nd ACM SIGIR International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016, pp. 353–362, doi: 10.1145/2939672.2939673.

[21] Lin, Y.—Liu, Z.—Sun, M.—Liu, Y.—Zhu, X.: Learning Entity and Relation Embeddings for Knowledge Graph Completion. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 29, 2015, No. 1, pp. 2181–2187, doi: 10.1609/aaai.v29i1.9491.

[22] Wang, H.—Zhang, F.—Xie, X.—Guo, M.: DKN: Deep Knowledge-Aware Network for News Recommendation. Proceedings of the 2018 World Wide Web Conference (WWW '18), 2018, pp. 1835–1844, doi: 10.1145/3178876.3186175.

[23] Wang, H.—Zhang, F.—Zhao, M.—Li, W.—Xie, X.—Guo, M.: Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. The World Wide Web Conference (WWW '19), 2019, pp. 2000–2010, doi: 10.1145/3308558.3313411.

[24] Xin, X.—He, X.—Zhang, Y.—Zhang, Y.—Jose, J.: Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), 2019, pp. 125–134, doi: 10.1145/3331184.3331188.

[25] Yang, B.—Yih, W.—He, X.—Gao, J.—Deng, L.: Embedding Entities and Relations for Learning and Inference in Knowledge Bases. CoRR, 2014, doi: 10.48550/arXiv.1412.6575.

[26] Niu, Y.—Meng, Y.—Niu, Q.: Text Recommendation Based on Heterogeneous

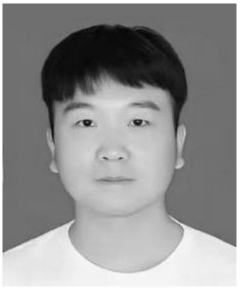
Attention Recurrent Neural Network. Computer Engineering, Vol. 46, 2020, No. 10, pp. 52–59, doi: 10.19678/j.issn.1000-3428.0055861 (in Chinese).

Guang Zheng is the Master of Internet technology and Associate Professor. He graduated from the University of Wollongong, Australia in 2003. He works at the Henan Agricultural University and his research interests include edge computing and deep learning.

Demeng Zhang is the Master of computer science and technology. She graduated from the Henan Agricultural University in 2022. Her research interests include knowledge graphs and cloud computing.

Xinming Ma received his Ph.D. from the Nanjing Agricultural University. He works in the College of Information and Management Science at the Henan Agricultural University. His research interests include crop science, agricultural system model and agricultural informatization.

Yanna Ren is the Master of agricultural informatization. She graduated from the Henan Agricultural University in 2006. Her research interests include virtual visualization and machine learning.



Chengzhong Sun is the Master of computer science and technology. He graduated from the Henan Agricultural University in 2022. His research interests include agricultural big data and visualization of data.

Lei Xi is the corresponding author. His Master's degree comes from the University of Electronic Science and Technology of China, currently he serves as Professor at the College of Information and Management Science at the Henan Agricultural University. His research interests include distributed computation, virtual simulation technology and knowledge engineering.