ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Personalized Pushing Approach of Teaching Resources in Higher Vocational Colleges Based on User Browsing Information Mining

Aiqiao Wang[1,*]

[1] School of Marxism, Henan Polytechnic School, Zhengzhou, Henan, 450000, China

SUMMARY: *User browsing information contains user's interest preference, which is of great value for realizing personalized push of teaching resources. In this paper, we developed a user intent classification model grounded in IndRNN-Attention, extracted user intent from the user's historical browsing records, and subsequently designed a network for explicitly modeling user intent based on knowledge graph (KG-UIN), and assessed the practical effect of collaborative filtering recommendation algorithms based on KG-UIN on the task of delivering teaching resources for higher vocational colleges and universities. The study reveals that the intent classification accuracy of the IndRNN-Attention model across the training set, validation set, and test set attains 96.97%, 80.44%, and 83.32%, respectively, all of which surpass those of the LSTM and the IndRNN model, which shows that the IndRNN-Attention model introduced in this paper yields a pronounced effect. In addition, the recommendation system based on KG-UIN achieves better results in user satisfaction, accuracy and surprise of recommended resources, which can be applied to the personalized push task of teaching resources in higher vocational institutions and universities.*

KEYWORDS: *IndRNN; attention mechanism; user browsing information mining; user intent; knowledge graph, teaching resources suggestion*

1 Introduction

As the Internet becomes increasingly widespread, online education is developing rapidly, online teaching resources are growing at an exponential rate, and the broad reach of the Internet equips every learner with the capacity to retrieve teaching resource information at any time and place [1, 2]. Learners are growing ever more attentive to the collective sharing of digital educational resources, and the appetite for high-quality educational resources continues to rise. Currently, education informatization has become a research hotspot, which integrates contemporary information technology with the foundational characteristics of digitalization, multimedia, networking and intelligence [3]. Digital conversion renders complex and variable educational information into quantifiable numbers or data, multimedia expands the modes of information exchange throughout the instructional process, networking renders educational resources accessible and shareable, and intelligence empowers the system to personalize the teaching behavior [4-7]. However, the increasingly convenient access to educational information has also brought about a succession of challenges, namely, given the extraordinarily swift accumulation of educational data, learners find it increasingly hard to locate content of personal relevance within a vast pool of information, and the study of personalized teaching has arisen as a result [8, 9].

*hnzy6035@163.com
https://doi.org/10.65102/is2026444
Published: 30 April, 2026

At present, there are many personalized teaching recommendation methods that simply draw on ordinary commodity recommendation methods, directly targeting learners in online education platforms to consumers in e-commerce platforms, and targeting teaching resources to commodities [10]. However, the recommendation effect of this approach is often unsatisfactory, mainly because resource recommendation in the education domain is different from other common resource recommendation. Some existing methods only find the teaching resources that users have studied historically, and then use their own feature information to make recommendations, without taking into account the structural information of teaching resources, which makes the final recommendation results not accurate enough [11, 12].

Over the past several years, data mining technology has advanced at a remarkable pace, and all aspects of people's lives can be applied to data mining technology. Such as the MICAPS system in the field of weather analysis and forecasting of the weather cloud map statistics, analysis and prediction [13, 14]. Leveraging data mining technology within the educational domain to retrieve critical information from databases can furnish people with valuable information, and at the same time provide powerful information support and work guidance for educational researchers [15, 16]. Leveraging data mining technology within the educational domain to retrieve critical information from databases can furnish people with valuable information, and at the same time provide powerful information support and work guidance for educational researchers

In this paper, a user intent classification model is proposed by combining IndRNN with the attention mechanism to realize the classification of user intent through the mining of user browsing information. On this basis, the collaborative filtering recommendation model based on intent network KG-UIN is designed by explicitly modeling the obtained user intent based on knowledge graph. The model captures the synergistic signals embedded between users at a finer-grained intent level by introducing user intents between users and item nodes, and incorporates the relationships between nodes into the node representations. With the aim of verifying the efficacy of the model, practical experiments are carried out on the model separately and the recommendation effect of the recommender system is evaluated in terms of the dimensions of user satisfaction, TopN and surprise.

2 User intent classification model based on user browsing information mining

In this chapter, IndRNN-Attention model is constructed, which achieves the classification of user's intention by mining the user's historical browsing data, so as to establish the groundwork for the personalized push of teaching resources in higher vocational colleges and universities.

2.1 User Intent Classification Based on IndRNN-Attention

The structure of the IndRNN-Attention based user intent classification model introduced in this paper is illustrated in Fig. 1, which comprises an input layer, a coding layer, an attention layer and a classification layer.

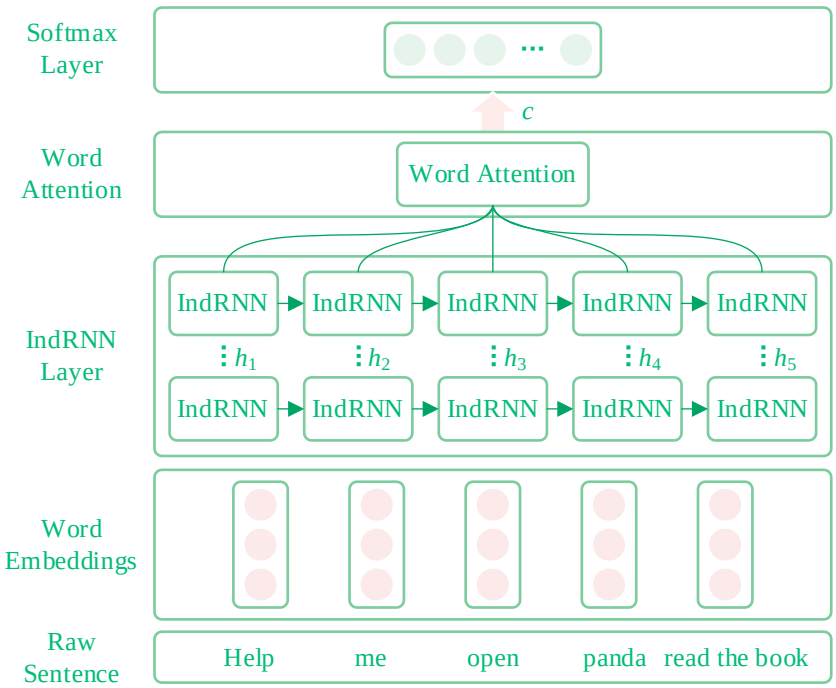


Figure 1: IndRNN-Attention model structure

2.1.1 Input layer

What is fed into the model is a matrix of word vectors corresponding to a sentence entered by the user, and the word vectors are obtained from a pre-trained word2vec model. For any sentence in the dataset, the word vector matrix of the sentence is obtained by partitioning it into words and then filling it with reference to the maximum length of the sentence in the dataset, and converting each word in the sentence into the corresponding word vector representation.

2.1.2 IndRNN layer

RNNs are widely used across domains such as action recognition, scene labeling and natural language processing. The basic RNN unit architecture is depicted in Figure 2.

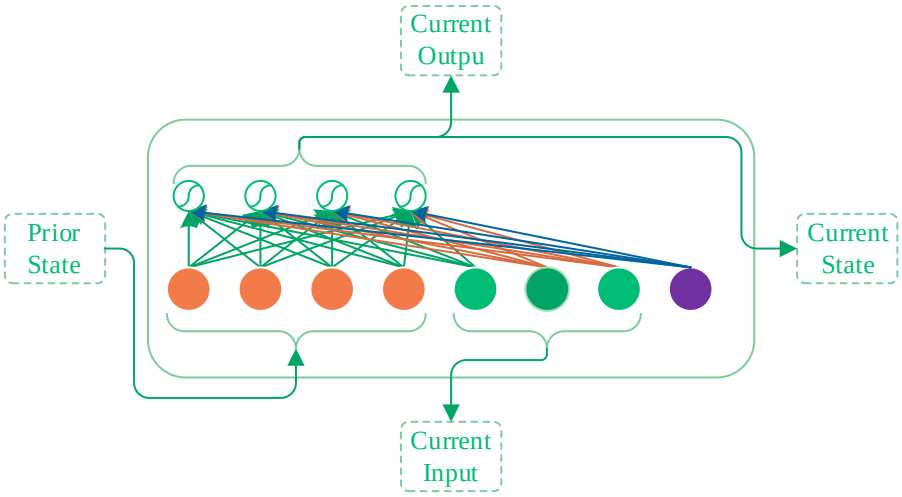


Figure 2: Basic RNN unit structure

In RNN, each neuron takes in both the current time-step input and the output from the

preceding time-step, and the state at moment t is refreshed as:

$$h_t = \sigma\left(Wx_t + Uh_{t-1} + b\right) \tag{1}$$

where $x_t \in \mathbb{R}^M$ and $h_{i-1} \in \mathbb{R}^N$ are the inputs at the moment t and the hidden state at the moment $t-1$, respectively, $W \in \mathbb{R}^{N\times M}$ and $U \in \mathbb{R}^{N\times N}$ are the weight matrices of the neuron's current moment's inputs and recurrent inputs, respectively, $b \in \mathbb{R}^N$ denotes the bias term, σ represents the neuron's activation function, N refers to the number of neurons per RNN layer, and M indicates the dimension of the input layer.

RNNs are prone to "gradient vanishing" and "gradient explosion" problems during training, which makes the RNN's capacity for capturing long-term dependencies comparatively weak. To address these problems, researchers put forward the Long Short-Term Memory Network (LSTM) and the Gated Recurrent Unit (GRU). Recurrent Unit (GRU). Although these 2 network structures improve these gradient problems to some extent, building and optimizing a deep recurrent neural network is in practice considerably more challenging because LSTM and GRU use tanh function and sigmoid function as activation function, which leads to gradient decay from layer to layer.

The above problem can be effectively solved by using IndRNN.The structure of IndRNN unit is shown in Fig. 3.

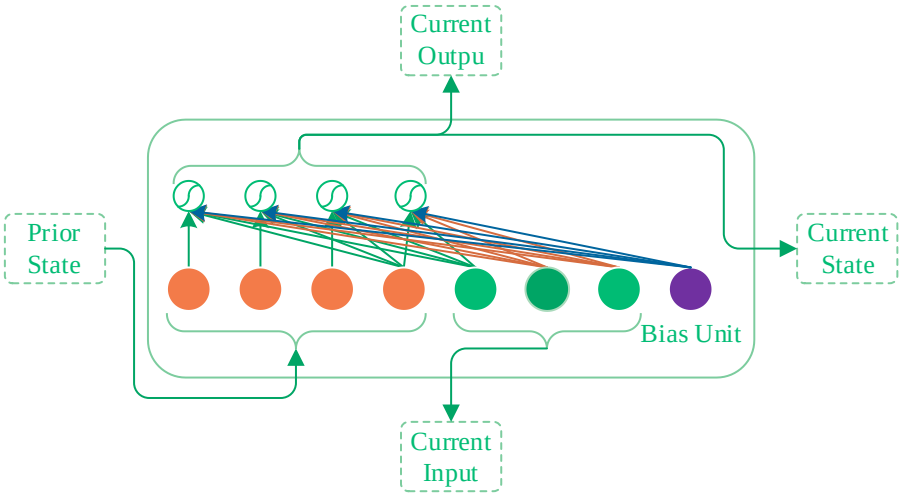


Figure 3: IndRNN unit structure

The IndRNN state update is shown in equation (2):

$$h_t = \sigma\left(Wx_t + U \odot h_{t-1} + b\right) \tag{2}$$

where $x_t \in \mathbb{R}^M$ and $h_{t-1} \in \mathbb{R}^N$ are the inputs at moment t and the hidden state at moment $t-1$, respectively, and $W \in \mathbb{R}^{N\times M}$ denotes the weight from the input layer to the hidden layer, $U \in \mathbb{R}^N$ denotes the weight from the hidden layer of the previous moment to the current hidden layer, and $\odot$ denotes the Hadamard product.Each neuron in each layer of the IndRNN is independent from each other, and the connection between the neurons can be realized by stacking the 2-layer or multi-layer IndRNN units. For the n neuron, the hidden state of the moment t can be computed:

$$h_{n,t} = \sigma\left(w_n x_t + u_n h_{n,t-1} + b_n\right) \tag{3}$$

where w_n and u_n denote the input weights and hidden layer weights of the nth row of neurons, respectively.

Within IndRNN, each neuron exclusively receives inputs from the current time-step and its own hidden state from the preceding time-step, with each neuron autonomously handling a single type of spatio-temporal pattern. While traditional RNNs are often viewed as multilayer perceptual machines that share parameters through time, IndRNNs show a new perspective of aggregating spatial patterns independently as time steps extend. By stacking 2 or more layers of neurons, each neuron in the next layer autonomously handles the outputs of all neurons in the preceding layer, which lowers the complexity of building deep network structures and strengthens the capacity to model longer sequences. Meanwhile, by leveraging non-saturated activation functions such as ReLU, the IndRNN exhibits greater robustness following training.

2.1.3 Attention mechanism layer

A common approach to obtain the semantic representation of text is to take the hidden state output vector of the last moment of the encoder as the final encoded vector, the advantage is that it can cover the textual semantic information effectively to a certain extent, the disadvantage is that it is difficult for this approach to encode all the information of the input text in a fixed-length vector. If the output vectors of each moment are directly summed or averaged, then it can be considered that each input word or phrase contributes equally to the result of calculating the semantic representation of the input text, and this approach reduces the degree of contribution of those words or phrases that are important for expressing the meaning of the text.

Therefore, we introduce a word-level attention mechanism to capture information regarding words that are semantically significant to the sentence. Given a sequence $S = \left(w_1, w_2, \cdots, w_T\right)$, T denotes the sequence length. The hidden state h_{it} of the ith word in the sequence S at the moment t can be computed by Eq. (3), and the word-level attention mechanism can be realized in 3 steps:

(1) Obtain the hidden representation h_{it} of u_{it} using a multilayer perceptual machine:

$$\mathbf{u}_{it} = \tanh\left(\mathbf{W}_{\mathrm{w}}\mathbf{h}_{it} + \mathbf{b}_{\mathrm{w}}\right) \tag{4}$$

(2) Calculate the correspondence between u_{it} and the word-level context u_w as a word importance measure, and compute the normalized weight α_{it} by softmax function:

$$\alpha_{it} = \frac{\exp\left(u_{it}^T u_w\right)}{\sum_t \exp\left(u_{it}^T u_w\right)} \tag{5}$$

The word context vector u_{w} undergoes random initialization and is simultaneously updated throughout the training process.

(3) Derive the sentence vector C:

$$C = \sum_t \alpha_{it} h_{it} \tag{6}$$

2.1.4 Classification layer

The sentence encoding result C is passed into a fully connected layer and the output of the fully connected layer serves as the final feature vector of the sentence, which is subsequently channeled into a softmax layer to yield the probability of each category. The calculation method is:

$$\hat{y} = \text{softmax}(WC + b) \tag{7}$$

where $\hat{y}$ is the predicted probability vector, W is the learned weight parameter, and b represents the bias term.

2.2 Experimentation and Analysis

In this section, the proposed IndRNN-Attention based model is applied experimentally to validate the effectiveness of the model.

2.2.1 Experimental data

The data used in this experiment is the browsing behavior data collected by the lab where Claypool is located, in which there are 2815 browsing records and 14 kinds of browsing behaviors collected, including page browsing time, horizontal scrolling time, vertical scrolling time, mouse movement time, number of mouse clicks, number of times of pulling the scroll bar, the number of times and time of PgUp/PgDn button clicks, the number of Up/Dn button clicks and time, etc. Test volunteers randomly viewed some of the web pages and labeled the pages with subjective interest ratings.

2.2.2 Data pre-processing

(1) Normalization

Considering that the data of various browsing behaviors differ greatly, if the original features are not normalized, it may lead to problems such as low accuracy or non-convergence of the objective function throughout the model optimization process, so they are normalized to the $[0,1]$ interval, which is calculated as shown in Eq. (8):

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{8}$$

In the above equation $x_{\max}$ and $x_{\min}$ respectively represent the upper and lower boundary values observed within the dataset.

(2) Unique heat coding

Since the rating scale, i.e., the user's interest level category is divided into 5 categories: dislike (1), not too much (2), general (3), like (4), and very much (5), in the process of training the model, with a view to streamlining the computation of the loss function, each learner's interest rating scale 1~5 are coded with unique heat: 10000, 01000, 00100, 00010, and 00001, respectively.

2.2.3 Parameter setting

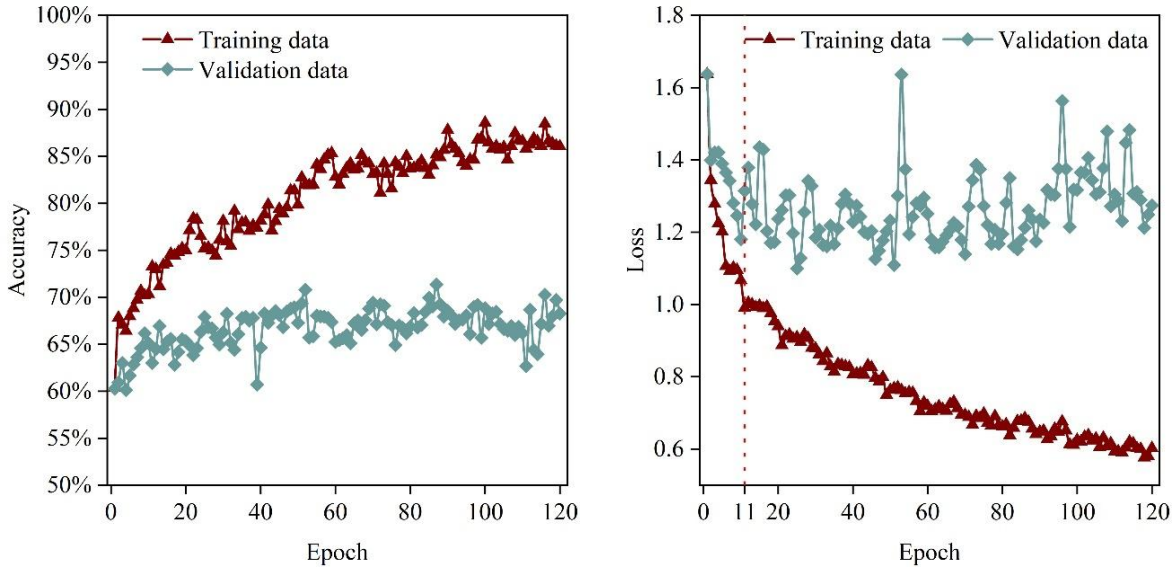
The whole experimental process was carried out in PyCharm programming environment, and the experimental data was partitioned at random into a training subset, a validation subset, and a test subset following an 8:1:1 proportion. The training batch size bath_size is configured as

15 and the total number of iterations is designated as 120.

2.2.4 Analysis of experimental results

(1) Examination of the convergence trajectory of the model across the training set and the validation set

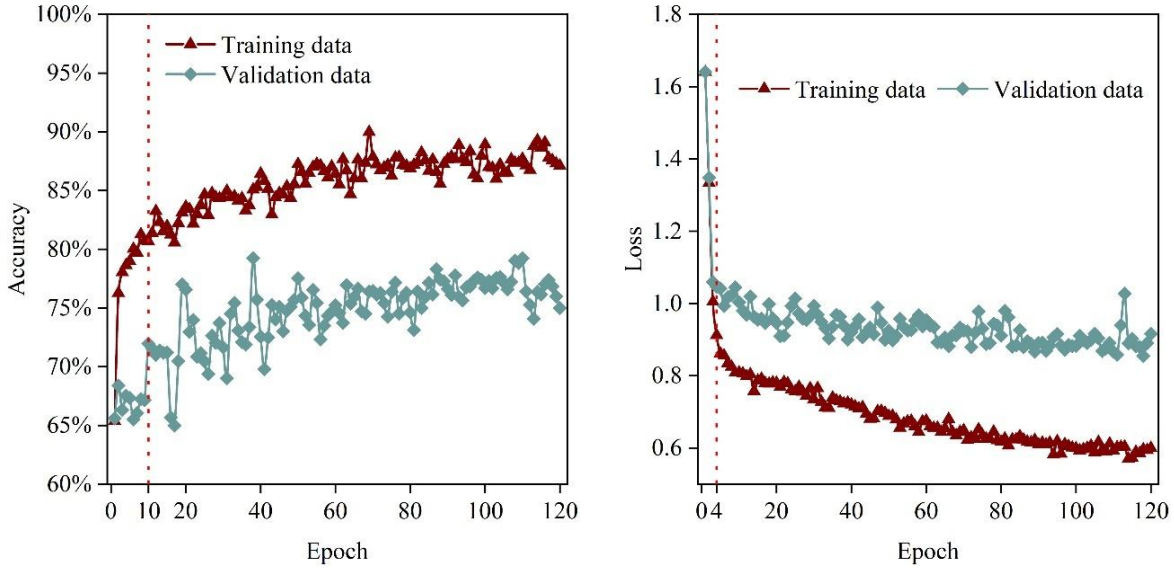
The fluctuation in accuracy and loss throughout the 120-iteration training of the LSTM model is presented in Fig. 4, where panels (a) and (b) illustrate the variation in accuracy and loss values, respectively, and similarly for those that follow. It is observable that the accuracy of the LSTM model across both the training and validation sets demonstrates a sustained upward trajectory, with the accuracy on the training set consistently exceeding that on the validation set. The training set accuracy climbs from approximately 60% at the outset to roughly 85%, whereas the validation set accuracy advances from around 60% to approximately 70%. Furthermore, the cross-entropy loss of the LSTM model on the training set falls considerably below that recorded on the validation set beginning from the 11th iteration onward.



(a) Accuracy (b) Loss

Figure 4: Convergence trajectory of the LSTM model during training

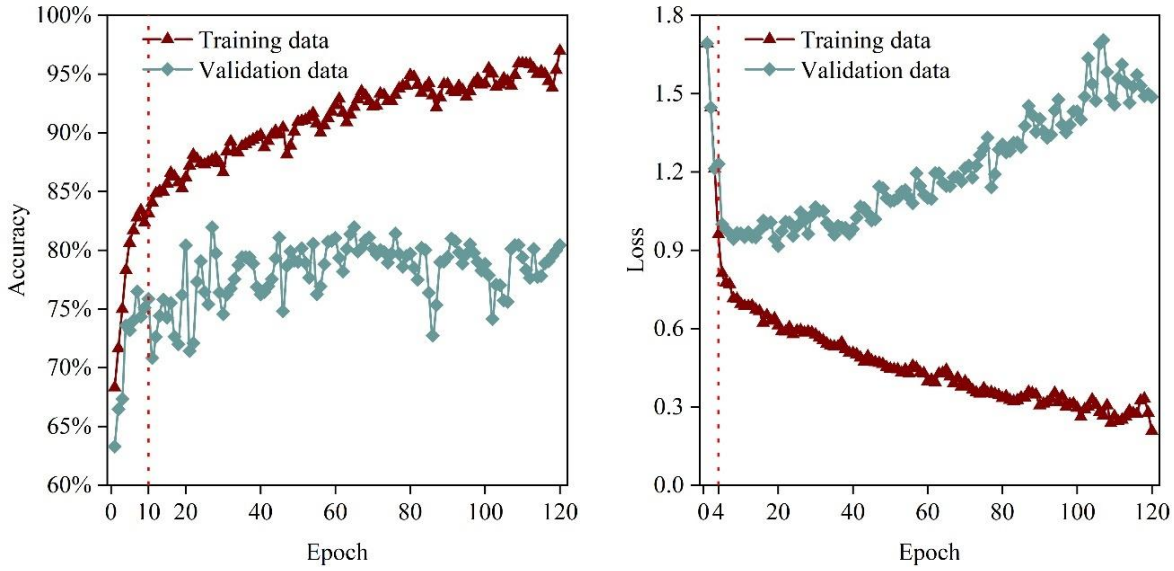
The iterative training trajectory of the IndRNN model is illustrated in Figure 5. It can be seen that during these 120 iterations, the accuracy of the model IndRNN continues to grow in the training subset and validation subset, with the accuracy in the training subset marginally surpassing that of the validation subset, occasional overlaps notwithstanding, and the accuracy in the validation subset escalates from roughly 65% at the beginning to approximately 75%, whereas the accuracy in the training subset exceeds 80% beyond the 10th iteration. The loss of the model shows a gradual decrease in both the training and validation sets, while the cross-entropy loss of the validation subset exhibits minor fluctuations and remains higher than the cross-entropy loss of the training subset after approximately 4 iterations.



(a) Accuracy (b) Loss

Figure 5: Training iteration process of the IndRNN model

The IndRNN-Attention model underwent a total of 120 training iterations, as depicted in Figure 6. It is evident that the accuracy of the IndRNN-Attention model across both the training and validation sets follows a consistently ascending trend, with the training set accuracy persistently outpacing that of the validation set. Specifically, the training set accuracy rises from an initial level of approximately 68% to beyond 95%, whereas the validation set accuracy climbs from around 63% at the outset to roughly 80%. Concurrently, the loss recorded on the training set undergoes a marked reduction, while the cross-entropy loss on the validation set oscillates and rises amid an overall downward tendency. Following approximately 4 iterations, the cross-entropy loss on the training set drops below that of the validation set, and the disparity between the two continues to widen as the number of iterations increases.



(a) Accuracy (b) Loss

Figure 6: Training iteration process of the IndRNN-Attention model

A comparison of the iterative process of the model's performance on the training set is presented in Figure 7. It is apparent that the accuracy of the IndRNN-Attention model introduced in this chapter substantially surpasses that of both IndRNN and LSTM throughout the training procedure, with the LSTM model recording notably inferior accuracy relative to IndRNN. Meanwhile, both the IndRNN and LSTM models attain accuracy levels exceeding 85% in the course of training, whereas the IndRNN-Attention model achieves an even more remarkable accuracy of over 95%.

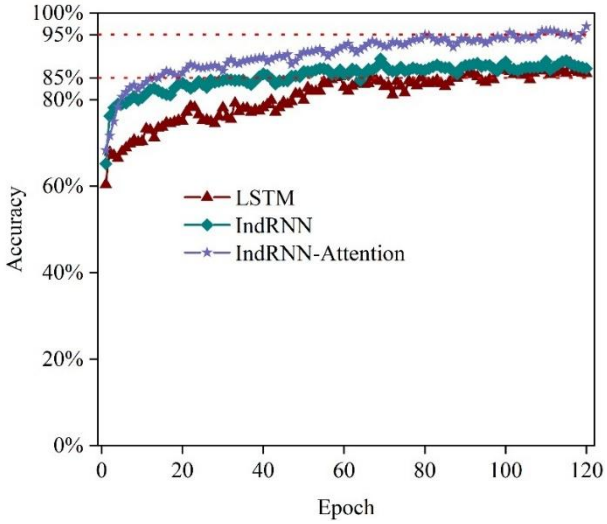


Figure 7: Comparison of the accuracy iteration process across the training set

A comparison of the loss convergence trajectories of the three models on the training set is presented in Figure 8. It is observable that the cross-entropy loss of the IndRNN-Attention model introduced in this chapter is considerably lower than that of IndRNN and LSTM during the training process, which can be as low as about 0.21, while there is not much difference between the cross-entropy loss of the IndRNN and LSTM models after the cross-entropy loss basically overlaps after about 100 iterations, and can be as low as about 0.60.

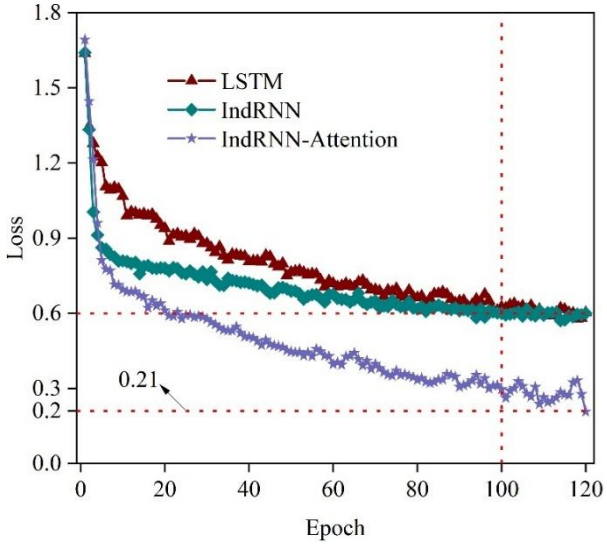


Figure 8: Comparison of loss iterations of the three models on the training set

(2) Classification comparison of three models

Next, the classification accuracies of LSTM, IndRNN and IndRNN-Attention are compared from three aspects: training set, validation set and test set.

The results of the user intent classification experiment are shown in Figure 9. It can be seen that after 120 iterations of training, LSTM, IndRNN and IndRNN-Attention all achieve more than 86% accuracy on the training set, where the IndRNN-Attention model has the highest intention classification accuracy of 96.97%, followed by IndRNN accuracy of 87.11% and Lastly, LSTM accuracy reached 86.06%. On the validation set, the accuracy of user intent classification reaches more than 68% for both, and the IndRNN-Attention model is found to be more accurate than the IndRNN model, while and both are more accurate than LSTM. And on the test set, the classification accuracy of IndRNN-Attention reaches up to about 83.32%, which is higher than that of IndRNN and LSTM, followed by LSTM, all of which are lower than that of IndRNN.This shows that the IndRNN-Attention model proposed in this chapter has a significant effect on the classification of browsing behavior data.

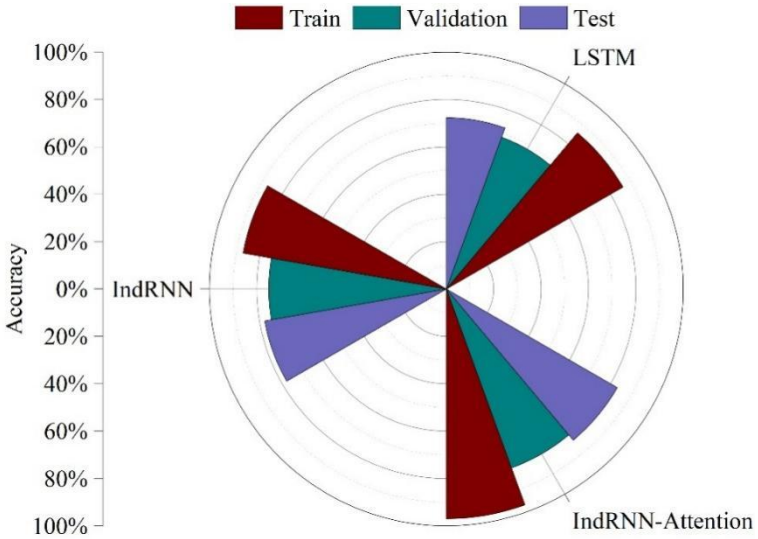


Figure 9: Experimental results of user intention classification

3 Knowledge graph-based recommendation algorithms for explicitly modeling user intent

On the basis of intent classification by mining user browsing information, this chapter proposes a network for explicitly modeling user intent based on knowledge graph (KG-UIN), which models the combination of relationships in the knowledge graph as potential user intent to achieve personalized push of instructional resources in higher vocational institutions based on knowledge graph and user intent.

3.1 Relevant symbols and mission objectives

3.1.1 Related symbols

Interaction dataset: notation U and I designate the collection of users and items, respectively. $O^{+} = \{(u,i) \mid u \in U, i \in I\}$ encompasses the affirmative samples within the training set, and each dichotomy (u,i) signifies that user u has previously engaged with item i.

Project Knowledge Graph: notation V and R epresent the set of entities and the collection of inter-entity relations within the project knowledge graph, respectively, and the

entity set of the project knowledge graph encompasses a range of entities pertaining to the project and its associated attributes. $G=\{(h,r,t)\,|\,h,t\in V,r\in R\}$ constitutes an assemblage of knowledge triples, each of which (h,r,t) indicates that head node h is linked to tail node t via relation r.

3.1.2 Mission objectives

Given interaction data O^{+} and an item knowledge graph G, learn a function that is used to generate the probability value that a given user will interact with any item.

3.2 KG-UIN architecture

In this chapter, the KG-UIN algorithm is obtained by improving the Bilinear Graph Attention Network Recommendation Algorithm for Fusion of Knowledge Graphs (KG-BGAT).The overall structure of KG-UIN is shown in Fig. 10, which contains three main components: a user intent modeling module, an information dissemination module based on relational paths, and a prediction layer for multiscale fusion.

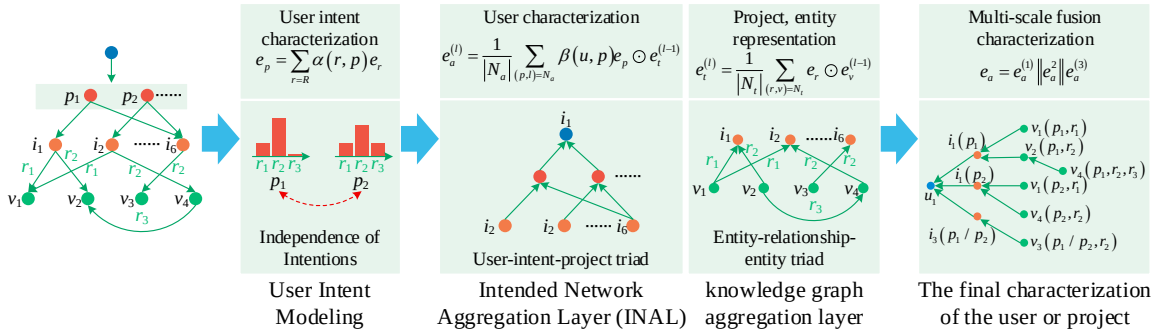


Figure 10: Overall framework of KG-UIN

3.2.1 User Intent Modeling

KG-UIN decomposes the unified "interaction" relationship used in KG-BGAT to connect users and projects into $|P|$ distinct intents. where P is the set of intentions shared by all users, and thus every binary (u,i) in the interaction dataset can be decomposed into $\{(u,p,i)\,|\,p\in P\}$.Eventually, a heterogeneous graph can be built on the user-item interaction data accordingly. This structure is designated as an Intent Network (IN) within the framework of this chapter, and is formally expressed as: $\mathrm{IN}=\left\{(u,p,i)\,|\,(u,i)\in O^{+},p\in P\right\}$.

(1) Representation Learning of Intentions

Unlike KGCN, KTUP and CKAN, which propose one-to-one correspondence of each intention with each relation r in the project knowledge graph G, KG-UIN considers interactions and combinations between relations instead of isolating the relations one-by-one in the knowledge graph. Therefore, in this chapter, each user intent p is modeled as a combination of all relations r within the project knowledge graph G, and an attention mechanism is employed to derive the embedding representations of user intents:

$$e_p=\sum_{r\in R}\alpha(r,p)e_r \tag{9}$$

wherein e_r denotes the embedding representation of relation r .and $\alpha(r,p)$ serves as an attention coefficient, normalized via Softmax to quantify the significance of a given relation within a specific intent, the computation of which proceeds as follows:

$$\alpha(r,p)=\frac{\exp\left(w_{rp}\right)}{\sum_{r'\in R}\exp\left(w_{r'p}\right)} \tag{10}$$

where w_{rp} serves as an optimizable weight parameter governing a specific relation r and a particular intention p , which can be understood as the importance or contribution of a particular relation to a particular intention.

The final computed vector representation of each user intention e_p is a linear combination of all e_r . In the user intent modeling module, the collection of user intents is universally accessible across all users, and different users establish connections through the same intents to mine the unified user behavior pattern based on the idea of collaborative filtering.

(2) Intent independence modeling

Different intents should reflect different preferences of users, and intents with unique information can provide a more accurate and refined perspective for portraying users' behavioral patterns. Therefore, this paper adds the module of independence modeling of intention to KG-UIN. This module guides the learning of representation of intents by applying statistical metrics to keep different intents away from each other. This chapter performs independence modeling of intentions grounded in mutual information theory.

The concept of mutual information characterizes the extent of statistical dependence between two random variables. The mutual information between any two different intentions is kept at a minimum through loss function optimization so as to reinforce the independence among the intentions. The loss function is as follows:

$$L_{IN}=\sum_{p\in P}-\log\frac{\exp\left(s\left(e_p,e_p\right)/\tau\right)}{\sum_{p'\in P}\exp\left(s\left(e_p,e_{p'}\right)/\tau\right)} \tag{11}$$

where $s(\cdot)$ is a function that measures the correlation between two intention representations, which is set as a cosine similarity function in this paper. τ is a temperature hyperparameter in Softmax, which is set to a value less than 1 in this model to guide the representations between the intentions to be non-uniformly distributed so that the embedded representations of the intentions are far away from each other.

3.2.2 Relational Path Modeling

In this chapter, a path-aware aggregation mechanism for relationships is designed in KG-UIN to integrate the structured knowledge built based on relationships between nodes into the node representation as well.

(1) Information Aggregation on Intent Network IN

For a particular user u in the intent network IN, the history of intent-based interactions between user u along with item i is captured by $N_u=\{(p,i)\,|\,(u,p,i)\in \mathrm{IN}\}$,a set that also inscribes the primary connectivity of node u .The embedding of user u .is generated by consolidating intent-aware information drawn from historical items:

$$e_u^{(1)} = f_{\mathrm{IN}}\left(\left\{e_u^{(0)}, e_p, e_i^{(0)}\right\} | (p,i) \in N_u\right) \tag{12}$$

wherein $e_u^{(1)} \in \mathbb{R}^d$ constitutes an aggregated vector encoding the first-order neighboring nodes and relations associated with user u. $f_{\mathrm{IN}}(\bullet)$ is an information aggregator in the intentional network, and in KG-UIN, this aggregation function is computed as follows:

$$e_u^{(1)} = \frac{1}{\left|N_u\right|} \sum_{(p,i)\in N_n} \beta(u,p) e_p \odot e_i^{(0)} \tag{13}$$

where $e_i^{(0)}$ denotes the embedding representation of item i. $\odot$ is the element product. The variability of how different users treat different intents is described using $\beta(u,p)$, which is an attention score that portrays the importance of different intents to the current user and is computed as follows:

$$\beta(u,p) = \frac{\exp\left(e_p^{\top} e_u^{(0)}\right)}{\sum_{p' \in P} \exp\left(e_{p'}^{\top} e_u^{(0)}\right)} \tag{14}$$

where $e_u^{(0)} \in \mathbb{R}^d$ is the original embedding of user u, e_p is the user intent embedding generated by the user intent modeling module, and $e_p^{\top}$ transforms this column vector into a row vector.

(2) Information Aggregation on the Knowledge Graph KG

For a particular item i, $N_i = \{(r,v) | (i,r,v) \in G\}$ is used to represent the characteristic features and primary connectivity structure of item i, and subsequently consolidates the relation-aware information from adjacent entities into the resulting representation:

$$e_i^{(1)} = f_{KG}\left(\left\{\left(e_i^{(0)}, e_r, e_v^{(0)}\right) | (r,v) \in N_i\right\}\right) \tag{15}$$

Herein $e_i^{(1)} \in \mathbb{R}^d$ encompasses the information gathered from the immediate neighbors of item i, and $f_{KG}(\bullet)$ constitutes an aggregation mechanism responsible for extracting and consolidating information from each item-relationship-entity triad (i,r,v). The aggregator $f_{KG}(\bullet)$ of this model incorporates the relationships into the modeling scope and $e_i^{(1)}$ is calculated as follows:

$$e_i^{(1)} = \frac{1}{\left|N_i\right|} \sum_{(r,v)\in N_i} e_r \odot e_v^{(0)} \tag{16}$$

where $e_v^{(0)}$ is the embedding vector of the entity v and $\odot$ is the element product. Similarly, the embedding $e_v^{(1)}$ of each entity $v \in V$ beyond the item node in G can be derived accordingly.

3.2.3 Higher-order propagation

KG-UIN collects information from higher-order neighbors by stacking multiple information aggregation layers. The recursive aggregation formula is as follows:

$$e_u^{(l)} = f_{\mathrm{IG}}\left(\left\{\left(e_u^{(l-1)}, e_p, e_i^{(l-1)}\right) | (p,i) \in N_u\right\}\right) \tag{17}$$

$$e_i^{(l)} = f_{\mathrm{KG}}\left(\left\{\left(e_i^{(l-1)}, e_r, e_v^{(l-1)}\right) | (r,v) \in N_i\right\}\right) \tag{18}$$

where $e_u^{(l-1)}, e_i^{(l-1)}, e_v^{(l-1)}$ denote the encoded vectors of the user, the item, and the entity, respectively, each of which retains information propagated from its $(l-1)$ hop neighborhood encompassing both the node itself and the corresponding relational path.

3.2.4 Model predictions

After the higher-order propagation of the l layers, the model can obtain the encoded vectors of user u and item i across multiple layers, and ultimately, the multi-level representations of u and i are recursively consolidated into the definitive vector embeddings of the user and the item:

$$e_u^* = e_u^{(0)} \left\| e_u^{(1)} \right\| \cdots \left\| e_u^l, e_i^* = e_i^{(0)} \right\| e_i^{(1)} || \cdots || \ e_i^l \tag{19}$$

As a result, the sequence of relationships in the user's intent and path are encoded in the final representation. Afterwards, the user embedding vector and the item embedding vector are subjected to inner product computation to yield a score that estimates the probability of interaction between the user and the item:

$$\hat{y}_{ui} = e_u^{*\top} e_i^* \tag{20}$$

3.2.5 Loss function

KG-UIN also uses the Bayesian personalized ranking objective BPR as an optimization criterion of the following form:

$$L_{CF} = \sum_{(u,i,j)\in \mathrm{O}} -\ln \sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right) \tag{21}$$

where $\mathrm{O} = \left\{(u,i,j)\mathrm{I}(u,i) \in \mathrm{O}^+, (u,j) \in \mathrm{O}^-\right\}$ constitutes a training collection comprising the affirmative training samples, namely the documented history of interaction records, alongside the negative training samples, namely the unobserved historical interaction records consisting of the training dataset. The $\sigma(\cdot)$ is the softplus nonlinear activation function.

The final loss function of the model is as follows:

$$L_{KG-UIN} = L_{CF} + \lambda_1 L_{IN} + \lambda_2 \left\| \Theta \right\|_2^2 \tag{22}$$

where $\Theta = \left\{e_u^{(0)}, e_v^{(0)}, e_r, e_p, W \mid u \in U, v \in V, p \in P\right\}$ is the model parameter, where the set V contains the items to be recommended as well as the entities related to the item attributes. λ_1

and λ_2 are the two hyperparameters controlling the loss of independence modeling of intentions and the regularization term, respectively.

3.3 Experiments on the application of the KG-UIN model

3.3.1 Experimental data sets

With the aim of validating the efficacy of the KG-UIN model in the teaching resource recommendation task, this paper constructs the teaching resource dataset CoLR and its corresponding knowledge graph based on the characteristics of the teaching resource recommendation task as well as natural language processing and other technologies. In addition to the self-constructed CoLR dataset, the openly accessible Yelp2018 and Amazon-Book datasets are additionally incorporated into the experimental evaluation.

(1) The Yelp2018 dataset is employed for commercial venue recommendations and encompasses location details, ratings, review contents, and user information of registered merchants across the U.S. in 2018, comprising upwards of 5 million user review records spanning all 50 U.S. states.

(2) The Amazon-Book dataset is utilized for book product recommendations and aggregates metadata for over 82,000 books on the Amazon shopping platform, encompassing attributes such as title, ISBN code, author, publisher, publication date, user reviews, and category. The data size is about 85,000 rows with 13 fields per row, which includes information such as book title, ISBN code, author, publisher, user reviews and category.

(3) The CoLR dataset is designated for teaching resource recommendation. CoLR is a pedagogical resource dataset developed in this paper, which gathers information pertaining to course content, instructor details and laboratory support from various higher education teaching platforms, with a data size of approximately 24,000 entries, primarily covering the domains of computer science and electronic information. Each dataset consists of two parts: user item interaction graph and knowledge graph.

3.3.2 Experimental setup

(1) Experimental environment

This experiment is based on the GPU version of the PyTorch framework, the code is written and completed under VSCode, the Anaconda open source manager is used for version and dependency control of Python and PyTorch, and the NVIDIATITAN Xp graphics card is used for model training.

The hardware environment for this experiment is: Intel Xeon E52650 v4 @ 2.20GHz processor, 256G RAM, and NVIDIA TITAN Xp graphics card.

The software environment for this experiment is: 64-bit operating system Ubuntu 24.04.3, LTSCUDA 22.04, Python 3.13.0.

(2) Baseline method

With a view to assessing the performance of the proposed method on the recommendation task, it is compared with the following baseline methods:

BPR: a conventional collaborative filtering driven approach for prioritizing item candidates through pairwise ranking loss.

LightGCN: a GCN-based collaborative filtering suggestion method that streamlines the convolution operation during message passing between users and items.

CKE: An embedding-based knowledge-aware recommendation approach that leverages TransR to encode semantic information about items and integrates it into a denoising autoencoder to achieve knowledge-aware item representations.

RippleNet: a path-based knowledge-aware recommendation approach that enhances user representation by disseminating user preferences across a knowledge graph and constructing paths centered on that user.

KGAT: a GNN-based knowledge-aware recommendation approach that devises an attention messaging scheme for a knowledge-aware collaborative graph for embedding fusion and differentiates the relevance of neighboring nodes during propagation.

KG-BGAT: A recommendation method that fuses knowledge graph and bilinear graph attention network.

3.3.3 Evaluation of indicators

For the dataset described above, 80% of the interaction records for each user were drawn at random to form the training set, 10% of the interactions were allocated as the validation set to fine-tune the hyperparameters, and the remaining data were reserved as the test set. Each documented user-item interaction is treated as a positive sample, while sampled connections with items that have not been interacted with are designated as negative samples.

The experimental part uses Recall@K and NDCG@K as two metrics to measure the efficacy of the recommender system, with K defaulting to 30. Where Recall@K refers to the fraction of genuinely relevant items appearing among the top K entries in the recommender list, serving primarily as an indicator of the retrieval capability of the recommender system. And NDCG@K captures the relative ranking quality of real items as reflected by the top K entries of the recommendation list, which can measure the accuracy of the recommendation system.

With respect to each user in the test set, the items actually engaged with are treated as positive samples, and the extent of overlap between the outputs of the evaluation recommendation algorithms within the space of positive items is adopted as the criterion of accuracy, with the final experimental outcome representing the mean correctness of the outputs across all users in the test set.

3.3.4 Experimental results and analysis

(1) Model recommendation performance experiments.

The accuracy outcomes of the recommendation tasks are presented in Table 1. It is observed that the Recall@30 index values of KG-UIN across the three datasets of Yelp2018, Amazonbook and CoLR stand at 0.0771, 0.1368 and 0.0897 respectively, with the corresponding NDCG@30 index values being 0.0471, 0.0728 and 0.0529 respectively. Across the enumerated distinct datasets and several mainstream baseline methods, KG-UIN demonstrates superior performance in the majority of cases, which substantiates the generalizability and adaptability of the proposed KG-UIN model.

Compared with traditional recommendation models, most knowledge-aware recommendation-driven models are capable of attaining enhanced performance, which validates the viability of incorporating knowledge graph information to address the data sparsity problem in collaborative filtering.

In addition, KG-UIN also has better recommendation performance compared to other knowledge-aware models. This is because the KG-UIN model designs a new information aggregation mechanism for graph convolutional neural networks, and simultaneously, by treating the combination of relationships in the knowledge graph as potential user intentions, it achieves superior modeling capability and interpretability, which can effectively alleviate the problem of single recommendation results brought by long-tailed distribution.

Table 1: Test results of the baseline model

Model	Yelp2018		Amazon-Book		CoLR	
	Recall@30	NDCG@30	Recall@30	NDCG@30	Recall@30	NDCG@30
BPR	0.0564	0.0371	0.1158	0.0633	0.0637	0.0324
LightGCN	0.0671	0.0443	0.1251	0.0667	0.0705	0.0344
CKE	0.0659	0.0448	0.1317	0.0654	0.0763	0.0346
RippleNet	0.0565	0.0446	0.1097	0.0662	0.0742	0.0328
KGAT	0.0672	0.0446	0.1318	**0.0737**	0.0813	0.0413
KG-BGAT	0.0664	0.0462	0.1295	0.0728	0.0792	0.0431
KG-UIN	**0.0771**	**0.0471**	**0.1368**	0.0728	**0.0897**	**0.0529**

(2) Experiments on the effectiveness of mitigating long-tailed distribution and data noise

This subsection focuses on evaluating the performance of KG-UIN in mitigating the skewed data distribution and data noise by configuring the data to examine the efficacy of the proposed model.

1) With the aim of substantiating the enhancement brought by KG-UIN to recommendation tasks involving knowledge graphs with unevenly distributed items, the items in the CoLR dataset and the Yelp2018 dataset are partitioned into four equally sized data groups, and the density of inter-item interactions in each group escalates progressively with increasing group rank, meaning that the density of long-tailed items in the first group is the greatest, and the connectivity among the data is the most sparse. Different algorithms are then employed to generate recommendation outcomes for the four data groups. The experimental findings pertaining to long-tailed data processing are illustrated in Fig. 11, where panels (a) and (b) correspond to the CoLR dataset and the Yelp2018 dataset, respectively.

With regard to the recommendation prediction for different densities of interaction data, the results are represented by calculating the relative accuracy of Recall@30 and regularizing each group to between [0, 1]. Figure 11 shows that when the group size is small, i.e., when the long tail of data in the knowledge graph is prominent, KG-UIN possesses a better performance compared to other models, which demonstrates that the proposed method markedly enhances the recommendation outcomes for long-tail items and has an advantage in mitigating recommendation popularity bias. In addition, whereas the performance advantage of KG-UIN model over other models is gradually lost when the group size increases, because information aggregation based on intent networks tends to integrate noisy user and item related information into the representation of the target node, which instead exacerbates the popularity bias.

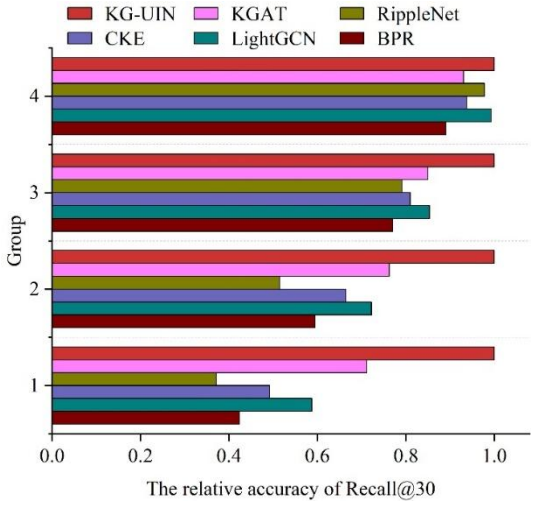


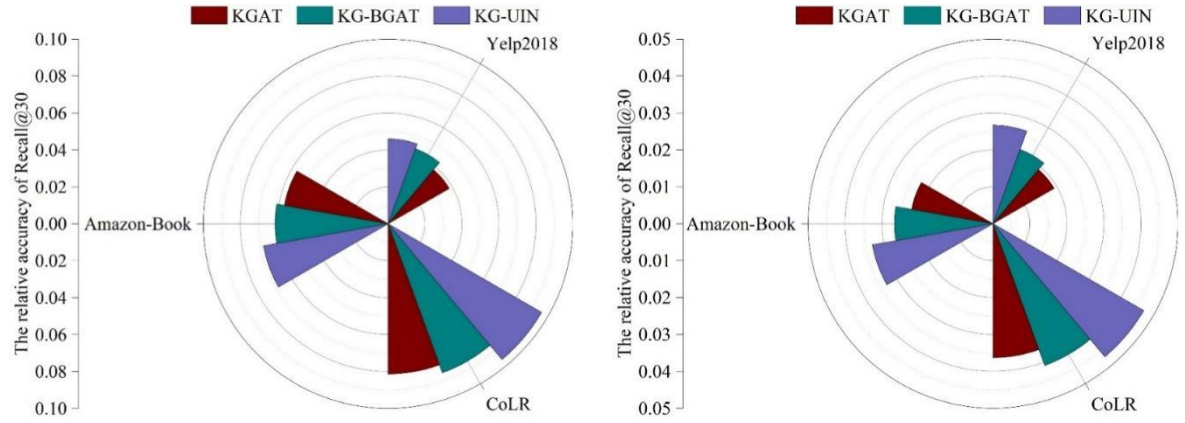
Figure 11: Experimental results of processing long-tail data

2) With the aim of validating the utility of KG-UIN for recommender systems operating on knowledge graphs containing noisy data, experiments are carried out on three datasets, Yelp2018, Amazon Book and CoLR. The outcomes of processing noisy data are presented in Table 2, achieved by introducing noisy triples and arbitrary connections into the knowledge graph so as to replicate knowledge graph noise attributable to irrelevant entities. Furthermore, to replicate knowledge graph noise stemming from long-tailed entities, 20% of the long-tailed entities along with their connected items are designated for evaluating the recommendation model in this experiment, and the ultimate results reflecting the accuracy of the two metrics, Recall@30 and NDGC@30, against the noise of long-tailed knowledge entities are depicted in Figure 12.

An integrated examination of Table 2 and Figure 12 reveals that the proposed KG-UIN model continues to deliver superior performance when benchmarked against other knowledge-aware recommendation models. Specifically, KG-UIN excels at suppressing knowledge graph noise, while simultaneously yielding the most favorable evaluation outcomes on items with sparse knowledge entities. This corroborates the soundness of explicitly modeling user intent and relationship paths based on knowledge graphs. In addition, by comparatively learning different view representations of users and items, the model demonstrates the capacity to efficiently identify the most informative patterns, thereby rendering the modeling of user intent less susceptible to dependence on specific nodes or edges. In summary, KG-UIN provides a method that can effectively mitigate noisy interaction information.

Table 2: Experimental results of processing noise data

Model	Yelp2018		Amazon-Book		CoLR	
	Recall@30	NDCG@30	Recall@30	NDCG@30	Recall@30	NDCG@30
KGAT	0.0656	0.0406	0.1329	0.0628	0.0771	0.0363
KG-BGAT	0.0685	0.0454	0.1338	0.0703	0.0813	0.0321
KG-UIN	**0.0823**	**0.0562**	**0.1351**	**0.0779**	**0.0897**	**0.0422**



(a) The accuracy of Recall@20 (b) The accuracy of NDGC@20

Figure 12: Experimental results against long-tail knowledge entity noise

3.4 Experiments on the effectiveness of KG-UIN based recommender system

In this section, the proposed KG-UIN-based collaborative filtering recommendation algorithm is systematically applied to analyze the accuracy of similar resources in the recommendation process, and user satisfaction, TopN, and surprise methods are selected to evaluate the

recommendation effect of this recommendation system.

3.4.1 Similar Resource Accuracy

The main test subject of the experiment is physics, and 12 resources are randomly selected to detect the similar resource accuracy rate situation, mainly by manually searching for resources, checking the recommended resources section in the resource page, and counting the ratio of similar resources to the total recommended resources to derive the accuracy rate of the similar resources, and the similar resource accuracy rate statistics are shown in Table 3.

In general, the accuracy rate of randomly selected resources is high, and the average value of the accuracy rate is 83.33%, but from the table, we can see that the accuracy rate of similar resources of the two resources numbered 9 and 10 is 55% and 65%, respectively, and the accuracy rate of the similar resources of these two resources is on the low side, and after analyzing, it is concluded that the reasons for the low accuracy rate may include:

(1) Resource quality problems. The content of the resources is not standardized, the normal resources should be in accordance with a certain knowledge point to develop the introduction of the description, but some resources have less content and do not focus on a specific knowledge point, resulting in the extraction of keywords when the deviation occurs, so the screening of resources should be increased to reduce the incidence of such a situation.

(2) Because of the large number of resources, it is not possible to screen some resources individually, but only to set a value that meets most of the resources, so a small part of the resources are not sensitive to the set value, resulting in unsatisfactory results.

Table 3: Accuracy statistics of similar resources

Number	Resource id	Accuracy /%
1	92034	100
2	85748	85
3	89012	80
4	61434	100
5	89025	80
6	86947	95
7	72638	80
8	61542	85
9	61251	55
10	88036	65
11	89154	85
12	61632	90
Average accuracy		83.33

3.4.2 Tests of the effectiveness of recommendations

(1) User satisfaction

User satisfaction is a direct reflection of the intuitive feelings of users after use, and is one of the most important indicators in the evaluation index of the recommender system. This experiment on user satisfaction data collection using questionnaire survey method, recommending the use of the system object is a higher vocational college of two classes of 74 students. In total, 74 questionnaires were administered, of which 60 remained valid upon elimination of the invalid responses.

The first part of the questionnaire is the overall evaluation of the recommendation effect, which is divided into three levels: satisfied, basically satisfied and dissatisfied. Satisfaction

means that the recommended resources meet the current learning needs of students, can quickly and accurately recommend resources to students, and the system has a good recommendation effect. Basic satisfaction means that the resources recommended by the system basically align with the academic requirements of students, yet further refinement remains necessary. Unsatisfactory means that the resources recommended by Purple Copper fail to address the current needs of students entirely or a substantial proportion of them fall short of the current needs of students, indicating that the recommendation system has not achieved its intended purpose.

The results of the questionnaire show that the number of people who are satisfied, basically satisfied, and dissatisfied account for 50.00%, 48.33%, and 1.67% respectively, and the overall satisfaction level is high. Upon further investigation, it was found that the reason for basic satisfaction was the presence of a few resources in the recommendation list that did not match the cognitive level of that student or the resources that the student thought they were most interested in and had the highest demand were not at the top of the list.

The second part of the questionnaire was to rate the resources appearing in the recommended list individually, out of a total of 10 points, and the students scored them according to their satisfaction with the resource. A higher score means that the resource is more in line with current student needs and vice versa. The average resource scoring is shown in Figure 13.

In order to match the overall aesthetics of the page and the overall level of recommendation, the recommendation list displays 10 recommended resources at a time. The order of the resources indicates the system's prediction of how much the user is likely to like the resource, and the resources at the top of the list are the ones that the system predicts to be the most compatible with the current user. It can be seen that the average scoring of the resources is trending downwards, which is in line with the system's predicted trend described above. The resources at the top of the list have higher scores, between 7.8 and 8.5, but the resources at the back of the list have lower scores, with the lowest scores between 7.5 and 7.8, and the overall scores above 7.5, with the accuracy rate needing to be improved. The result reflects the overall problem of recommendation, whether it is the user model, resource model or the design of the recommendation strategy may affect the satisfaction score of the resources in the recommended list. Therefore, if you want to continue to improve on the current score, you need to consider all three aspects.

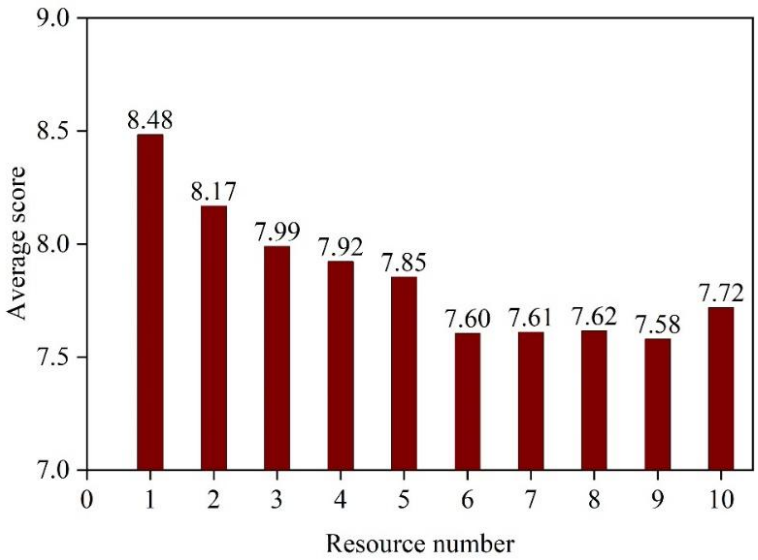


Figure 13: Average scoring of each resource

(2) Prediction accuracy

The experiments were carried out in three groups, each of which was unfolded in the previous group of experiments. Each group of experiments respectively made statistics on the number of resource entries recommended according to the history record, the number of entries recommended according to similar users, and the final accuracy rate in the first 10 multiples of the resources, with the purpose of calculating whether the source of the resources in the recommended list originates from policy-driven recommendation outcomes and the precision rate of the recommended results. The statistics of the three groups of experiments are presented in Table 4.

The findings of the experiments demonstrate that the accuracy rate in the three groups of experiments decreases gradually with the increase of the number of list entries, and this result is in line with the expectations of the experiments. After calculation, it is found that the accuracy rate of the first 10 resources in the list is above 0.95, that is, more than 95% of the first 10 resources are consistent with the user's requirements, and the precision level is sufficiently high to satisfy the user's current needs, so the overall recommendation results are better.

Table 4: Statistics of prediction accuracy

Group 1						
The number of list resources K	10	20	30	40	50	60
Historical records	5	8	15	20	27	32
Similar users	8	12	15	18	19	20
Accuracy rate	1	0.95	0.95	0.88	0.85	0.82
Group 2						
The number of list resources K	10	20	30	40	50	60
Historical records	8	12	15	24	28	35
Similar users	3	6	8	9	10	10
Accuracy rate	0.95	0.86	0.72	0.78	0.72	0.70
Group 3						
The number of list resources K	10	20	30	40	50	60
Historical records	10	14	18	25	32	38
Similar users	2	7	8	10	10	10
Accuracy rate	1	0.92	0.83	0.81	0.80	0.77

After further analysis, it is found that the number of resource articles recommended according to the history of browsing information escalates in tandem with the growing number of list articles. For instance, the number of resource articles recommended on the basis of browsing history among the first ten resources across the three experimental groups stand at 5, 8, and 10 respectively, while the corresponding figures for the first twenty resources are 8, 12, and 14 respectively, exhibiting a consistent upward trajectory.

With the increase in the volume of users engaging with the system, the accumulated records of resource searches and resource views expand progressively. The recommendation system transitions from an initial state of limited user understanding to a subsequent phase of detailed behavioral analysis, progressively capturing the user's preferences, whereupon the derived insights into user interests begin to inform the recommendation outcomes. This shows that the user's historical browsing data serves as both the foundational starting point and the ultimate reference for the computation of recommended resources by the recommender system, such that as the volume of historical browsing data grows, a greater proportion of recommendations derived from browsing history emerges.

(3) Surprise

Recommender system surprise index refers to the recommended resources are not related to the user's history of browsing resources, but will give the user a certain surprise, may be helpful to the user.

From the recommended list to find out the resources that have no relationship with the user's own browsing resources but meet the needs of the current user for statistical calculations, only statistics can be displayed on the list of the first 10 resources, the surprise degree of statistical results as shown in Table 5.

The surprise degree of the latter three groups is significantly lower than the surprise degree of the first two groups, this is because at the beginning of the user's browsing record is small so most of them are recommended according to the similarity between the users, so the degree of surprise is higher, with the increase of the user's record, the degree of surprise gradually decreases, and then in the process of the use of the system in the back of the degree of surprise of the approximate statistics, are in the vicinity of 40%.

Table 5: Surprise degree statistics

The resources searched	Resource surprise degree
1	80%
2	65%
3	40%
4	30%
5	40%

4 Conclusion

In this paper, we construct a user intent classification model based on IndRNN-Attention by mining user browsing information, and then design a collaborative filtering recommendation algorithm based on Knowledge Graph Explicitly Modeled User Intent Network (KG-UIN), and actually measure its recommendation effect.

The cross-entropy loss of the IndRNN-Attention model can be as low as about 0.21, which is significantly lower than that of the IndRNN and LSTM models of 0.60. Meanwhile, the accuracy of the IndRNN-Attention model in intent classification on the training subset, validation subset, and test subset surpasses 80%, representing a substantial improvement over that of the IndRNN and LSTM models, which shows that The IndRNN-Attention model proposed in this paper is able to achieve accurate classification of user's intention by mining user's browsing information, which lays the foundation for personalized push of teaching resources.

Except for the NDCG@30 metrics in the Amazon-Book dataset, KG-UIN achieves optimal results for both Recall@30 and NDCG@30 metrics values in the three datasets of Yelp2018, Amazon-Book, and CoLR, which substantiates the generalizability and adaptability of the proposed KG-UIN model. Meanwhile, comparing with other baseline models, KG-UIN can effectively mitigate the data long-tail distribution and data noise. Applying the recommendation algorithm based on KG-UIN to the recommender system, the user satisfaction and prediction accuracy are high, which meets the students' needs, and at the same time, the novelty level of the recommender system fulfills the operational demands of the resource recommendation system. Therefore, the pedagogical recommendation approach of this study is well-suited for the personalized delivery of teaching resources in higher vocational colleges. After practical use, the recommendation system can effectively recommend the content that aligns with users' interests to users and provide a good platform for their learning.

References

[1] Tang, J. (2025). Online English teaching resource recommendation method design based on LightGCN-CSCM. Systems and Soft Computing, 200294.

[2] Qiao, X. (2021). Integration model for multimedia education resource based on internet of things. International Journal of Continuing Engineering Education and Life Long Learning, 31(1), 17-35.

[3] Kuchai, O., Skyba, K., Demchenko, A., Savchenko, N., Necheporuk, Y., & Rezvan, O. (2022). The importance of multimedia education in the informatization of society. IJCSNS, 797.

[4] Vovk, O., Ishchuk, V., Kucheriavyi, O., Buzenovskyi, V., & Nevoisa, A. (2025). Impact of higher education digitalization on students' capacity for educational self-analysis in electronic environments. Revista Conrado, 21(107), e4965-e4965.

[5] Bilynska, K., Markova, O., Chornobryva, N., Kuznietsov, Y., & Mingli, W. (2024). The power of digitalization in education: improving learning with interactive multimedia content. Amazonia Investiga, 13(76), 188-201.

[6] Ma, H. (2021, August). Design and application of teaching resources sharing platform for physical education major based on Internet. In Journal of Physics: Conference Series (Vol. 1992, No. 2, p. 022197). IOP Publishing.

[7] Montenegro-Rueda, M., Fernández-Cerero, J., Fernández-Batanero, J. M., & López-Meneses, E. (2023). Impact of the implementation of ChatGPT in education: A systematic review. Computers, 12(8), 153.

[8] Barrot, J. S., Llenares, I. I., & Del Rosario, L. S. (2021). Students' online learning challenges during the pandemic and how they cope with them: The case of the Philippines. Education and information technologies, 26(6), 7321-7338.

[9] Romero, M. Á. M., Silva, G. J. S., Aguila, O. E. P., Valderas, P. J. V., Huallpa, J. J., Sosa, L. E. F., & Arias-Gonzáles, J. L. (2023). Personalized education path for students; a conceptual basis for a digitalized education environment. Journal of Namibian Studies: History Politics Culture, 33(1), 394-415.

[10] Tatineni, S. (2020). Recommendation Systems for Personalized Learning: A Data-Driven Approach in Education. Journal of Computer Engineering and Technology (JCET), 4(2), 18-31.

[11] Da Silva, F. L., Slodkowski, B. K., Da Silva, K. K. A., & Cazella, S. C. (2023). A systematic literature review on educational recommender systems for teaching and learning: research trends, limitations and opportunities. Education and information technologies, 28(3), 3289-3328.

[12] Elwarraki, O., Aammou, S., Jdidou, Y., & Lahiassi, J. (2023). Toward the future of personalized learning: Emerging trends and challenges in recommendation systems. ICERI2023 Proceedings, 7226-7235.

[13] Hong, H., Tsangaratos, P., Ilia, I., Liu, J., Zhu, A. X., & Chen, W. (2018). Application of fuzzy weight of evidence and data mining techniques in construction of flood susceptibility map of Poyang County, China. Science of the total environment, 625, 575-588.

[14] Zhuo, S. U. N., Song, G. A. O., Jiangbo, L. I., Jiangang, Z. E. N. G., & Xiaoliang, Y. A. N. G. (2022). Design and application of disastrous weather case retrieval and training system based on MICAPS4. Torrential Rain and Disasters, 39(2), 207-212.

[15] Salih, N. Z., & Khalaf, W. (2021). Prediction of student's performance through educational data mining techniques. Indonesian Journal of Electrical Engineering and Computer Science, 22(3), 1708-1715.

[16] Pal, A. K., & Pal, S. (2013). Analysis and mining of educational data for predicting the performance of students. International Journal of Electronics Communication and Computer Engineering, 4(5), 1560-1565.