IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 October 2022, accepted 30 October 2022, date of publication 1 November 2022, date of current version 4 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3218882

RESEARCH ARTICLE

RTN-GNNR: Fusing Review Text Features and Node Features for Graph Neural Network Recommendation

BOHUAI XIAO[1,2], XIAOLAN XIE[1,2], (Member, IEEE), CHENGYONG YANG[3], AND YUHAN WANG[1,2]

[1]School of Information Science and Engineering, Guilin University of Technology, Guilin 541004, China
[2]Guangxi Key Laboratory of Embedded Technology and Intelligent System, Guilin University of Technology, Guilin 541004, China
[3]Network and Information Center, Guilin University of Technology, Guilin 541004, China

Corresponding author: Xiaolan Xie (xxl@glut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61762031, in part by the Science and Technology Major Project of Guangxi Province under Grant AA19046004, in part by the Natural Science Foundation of Guangxi under Grant 2021JJA170130, in part by the Innovation Project of Guangxi Graduate Education under Grant YCSW2022326, and in part by the Research Project of Guangxi Philosophy and Social Science Planning under Grant 21FGL040.

ABSTRACT Recent recommendation systems have achieved good results by applying Graph Neural Network (GNN) to user-item interaction graphs. However, these recommendation systems can only handle structured interaction data and cannot handle unstructured review text data well. Based on the user-item interaction graph, combining review text can effectively solve the problem of data sparsity and improve recommendation quality. Most of the current recommendation methods combining review texts stitch the data from different modalities, leading to insufficient interactions and degrading the recommendations' performance. A model called RTN-GNNR to fuse **R**eview **T**ext feature and **N**ode feature for **G**raph **N**eural **N**etwork **R**ecommendation is proposed to solve these problems and get better item recommendations. RTN-GNNR consists of four modules. The review text feature extraction module proposes a Bi-directional Gated Recurrent Unit (Bi-GRU) text analysis method that combines Bidirectional Encoder Representation from Transformers (BERT) and attention mechanism to enable the model to focus on more valuable reviews. The node feature extraction module proposes a GNN combined with the attention mechanism for the interactive node extraction method, which enables the model to have better higher-order feature extraction capability. The feature fusion module proposes the method of tandem Factorization Machine (FM) and Multilayer Perceptron (MLP) to realize interactive learning among multi-source features. The prediction module inner-products the fused higher-order features to achieve recommendation effect. We conducted experiments on five publicly available datasets from Amazon, showing that RTN-GNNR outperforms state-of-the-art personalized recommendation methods in both RMSE and MSE, especially in the sparser two datasets. The effectiveness of each module of the model is also demonstrated by a comparison of the ablation experiments.

INDEX TERMS Recommendation algorithm, graph neural network, review-based recommendation, multimodal data fusion, attentional mechanism, data sparsity.

I. INTRODUCTION

The digital lifestyle is becoming popular, it is difficult for people to select items that meet their needs from the vast array of items available. Recommendation Systems (RS) [1] have thus emerged as an essential tool to help users find the best product or service among various choices.

Most data in RS inherently contain graph structures, and most objects are connected either explicitly or implicitly. This

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

inherent data characteristic makes it necessary to consider complex inter-object relationships when making recommendations. Therefore, with the research and development of Graph Neural Network (GNN), more and more researchers are using GNN for RS to extract node information about the associations between users and items. Berg et al. [2] used GNN to fill in the missing rating information in the interaction graph. Ying et al. [3] combine wandering and GNN to generate node embeddings that contain graph structure and feature information of the nodes. Zhuang et al. [4] proposed to use of multi-connected graph convolutional encoders to learn node representations. Wang et al. [5] introduced the idea of residuals into GNN to multiple aggregate layers of neighbor representations into the final node representation. Although GNN-based RS excels in feature extraction and can effectively extract potential higher-order features, the recommendation effect is always poor because the user-item interaction graph node information is usually very sparse.

As user engagement continues to increase, a large amount of User-generated Content (UGC) is generated. As the UGC that best reflects users' emotions, review text is rich in a large amount of multidimensional information. Therefore, combining review texts to alleviate data sparsity has become one of the hot issues in recent research. Fan et al. [6] model the user-item interaction graph with textual attributes as a knowledge graph and learn the embeddings in the interaction graph for the recommendation. Breitfuss et al. [7] extract user sentiments from chat logs and combine them with knowledge graphs to provide users with movie recommendations. Jian et al. [8] designed a multimodal collaboration graph model that aggregates visual and collaboration signals embedded in users and items to personalize matching pairs of users and items. The review text is fused with the rating information of user and item interactions. Potential higher-order features are extracted using graph ideas, improving recommendations' quality while effectively alleviating data sparsity. However, review text and rating information are multimodal data, and fusing different modalities by simple stitching will lead to insufficient mutual interaction between features and degrade the performance of recommendations.

In summary, this paper proposed a GNN-based item recommendation model (RTN-GNNR) with the following main contributions:

- A new methodological framework is proposed. The method uses a GNN combined with an attention mechanism to extract the main interaction graph node features, and fuses the review text features to achieve predictive recommendations based on the fused features.
- A new text analysis model is proposed. First, semantic information is extracted using Bidirectional Encoder Representation from Transformers (BERT). Then, the information of the review text is captured by Bi-directional Gated Recurrent Unit (Bi-GRU) to enable the model to analyze the review text features better. Finally, the attention mechanism gives each review a different attention weight, which enables the model to extract more valuable reviews.
- A feature fusion method for multimodal data is proposed. The node representation and the review text representation are learned interactively through Factorization Machines (FM) and Multilayer Perceptron (MLP) to achieve feature-level fusion and further improve the accuracy of recommendations.
- We experimented with the model on five publicly available Amazon datasets. The proposed method proved better than the existing item recommendation methods with improved results.

The rest of this paper is organized as follows. Section 2 describes the work related to the recommendation algorithm. Section 3 describes the specific description of each module of RTN-GNNR. Section 4 conducts experiments and detailed analysis. Finally, we conclude the paper in Section 5.

II. RELATED WORK

In this section, work related to analyzing recommendation algorithms is presented, mainly traditional recommendation algorithms, GNN-based recommendation algorithms, and review-based recommendation algorithms.

A. TRADITIONAL RECOMMENDATION ALGORITHMS

Traditional recommendation algorithms can be broadly classified into three categories: collaborative filtering recommendation algorithms, content-based recommendation algorithms, and hybrid recommendation algorithms [9].

The core idea of the collaborative filtering recommendation algorithm is to use the preferences of a group of people with similar interests to recommend information of interest to users [10]. Collaborative filtering only requires users' historical rating data, and data acquisition is relative while achieving better recommendation results, so it is currently widely used. However, although collaborative filtering-based recommendation algorithms are simple and effective, they often face data sparsity and cold-start for newly added users or items because the user rating data for items are minimal compared to the total number of interactions.

Content-based recommendation algorithms are proposed to alleviate the data sparsity and cold-start problems of collaborative filtering [11]. The core idea of a content-based recommendation algorithm is to obtain the items that users have interacted with through implicit feedback or explicit, and then use the content information of the interaction to capture the user's preference settings and make recommendations by the similarity between the preference settings and the items that have not been interacted with. Content-based recommendation algorithms can use the content information of items to alleviate the data sparsity problem effectively, and can alleviate the item cold-start problem by simply performing item content feature extraction. However, content-based recommendation algorithms have difficulty extracting features and cannot learn potential higher-order features.

Considering the shortcomings of a single recommendation method, some researchers have tried to combine different algorithms to achieve good recommendation results. Wu et al. [12] combined collaborative filtering and content-based recommendation algorithms to search a list of items based on ratings and content to achieve the effect of recommending books. Pazzani et al. [13] considered the results of multiple recommendation algorithms as a kind of voting combined these results. Zhang et al. [14] combined collaborative filtering with a recommendation algorithm based on grid technology to improve the effectiveness of recommendations.

In summary, most traditional recommendation algorithms use simple user-item interaction data for a recommendation, which faces a severe data sparsity problem. Although content-based and hybrid recommendations can somewhat alleviate the problem, these methods usually require time-consuming and labor-intensive manual operations, and the ability to extract features cannot be improved. Therefore, we perform feature extraction on multimodal data by GNN and deep learning techniques to save the workforce and improve the quality of extracted features simultaneously.

B. GNN-BASED RECOMMENDATION ALGORITHMS

GNN, as a neural network inspired by graph embedding, can model the data of graph structure and learn the relationship between nodes. In life, the field of recommendation has been studied intensely in processing Euclidean data, but much data in real life exists in the form of non-Euclidean data in graphs or grids. Therefore, with the continuous research and development of GNN, it has been widely used in RS. Wu et al. [15] applied gated graph neural networks to learn item representations from session graphs to obtain more accurate session embeddings. Xu et al. [16] utilized both GNN and self-attention mechanism to learn local dependencies and long-range dependencies, respectively, for session-based recommendations. Qiu et al. [17] used multiple weighted graph attention networks to learn item embeddings and obtain user preferences embeddings. He et al. [18] simplified the design of graph convolutional networks by removing feature transformations and nonlinear activations to achieve better results than graph convolutional networks. Duan et al. [19] used GNN to handle heterogeneous attributes and designed a component to explore the relationship between potential neighbor nodes. Sang et al. [20] used a self-attentive graph neural network and a soft attention mechanism to capture the dependencies between items.

However, user-item interaction data is usually very sparse. To address the data sparsity problem, many works have started incorporating attribute features of items into interaction features to achieve multimodal data fusion. Fan et al. [21] model users and items with attributes and their interactions as a knowledge graph to learn embeddings of users and items for recommendation. Fan et al. [22] proposed combining social graphs between users and user-item interaction graphs for social recommendation. Zhao et al. [23] proposed a framework to simultaneously capture the heterogeneous relationship between explicit user preferences and edge information.

In summary, the superiority of GNN in feature extraction has made applying GNN in recommendation an inevitable trend of development. Since the user-item interaction data is very sparse, some works have fused the attribute features of items into the features of interaction graphs. However, existing works directly learn the interaction graph features and item attribute features jointly, which has the disadvantage of insufficient interaction because graph data features and item attribute features are multimodal data with different homogeneity. Therefore, we design a GNN-based model, which can extract the main interaction graph features by GNN and combine the item attribute features, and fuse the item attribute features and the node features of the interaction graph for multi-source features, so that the features of different classes interact adequately and thus improve the performance of multimodal features.

C. REVIEW-BASED RECOMMENDATION ALGORITHMS

Topic modeling is the most common approach in recommendation algorithms that combine review text. Weber et al. [24] were the first to propose using textual information to alleviate the sparsity problem of rating data but did not perform deep mining of review text. Rogers et al. [25] proposed a method to input user review text into a probabilistic topic model to obtain potential features of users and products. Bao et al. [26] extracted text features using a non-negative matrix decomposition model for the review text, used a matrix decomposition model for the rating matrix for feature extraction, and fused them by mapping the hidden vector to the topic distribution parameters. Although the above recommendation algorithms combining review texts have good performance, they still have limitations. On the one hand, they ignore the contextual associations in the text and cannot accurately extract the potential features of users and items; on the other hand, they do not fully explore the depth features expressed in the review text.

In recent years, deep mining of review text using deep learning techniques has become the concern of many scholars with the development of deep learning. Kim et al. [27] proposed introducing a Convolutional Neural Network (CNN) into recommendation systems and capturing the contextual features of review text using CNN. Chen et al. [28] proposed using review text and rating information as data sources and introduced an attention mechanism, which led to a substantial improvement in the recommendation performance. Lu et al. [29] proposed a coevolutionary recommendation model which co-learns user and item information from ratings and customer reviews by optimizing matrix factorization and an attention-based GRU network.

In summary, we choose review text as the attribute features of items, and fuse review text extraction features based on deep learning techniques with graph neural network extraction features can improve the model's ability to

TABLE 1. Symbols' definition.

Symbols	Description
W	Word sequences of review text
S	Low-dimensional vector of review text
H	Review text overall hidden features
C	Review text overall features
U_{text}	Final features of user review text
I_{text}	Final features of item review text
$e_u^{(0)}$	Initial embedding vector of user u
$e_i^{(0)}$	Initial embedding vector of item i
$e_u^{(k)}$	Higher-order features of the user on the ith layer
U_{node}	Final node characteristics of the users
I_{node}	Final node characteristics of the items
F^U	Initial features of users in the fusion module
F^I	Initial features of items in the fusion module
U_{lowf}	Low-order features of users
I_{lowf}	Low-order features of items
U_{highf}	High-order features of users
I_{highf}	High-order features of items
$\hat{r}_{ui}$	Predicted matching score
r_{ui}	Actual matching core

extract higher-order features while alleviating data sparsity and cold-start, and significantly improving recommendation performance.

III. PROPOSED FRAMEWORK

In this section, the proposed model RTN-GNNR is described in detail, and the overall framework of the model is shown in Figure 1. The model includes the following modules:

- Review text feature extraction module, which is used to extract hidden features of reviews text, including user review text and item review text;
- Node feature extraction module, which is used to extract features of nodes on the user-item interaction graph;
- Feature fusion module, which is used to fuse features of review text and interaction graph nodes;
- Prediction module, which uses the fused features to predict the recommendation results.

Due to the many symbols involved, we have organized the main symbols, as shown in Table 1.

A. REVIEW TEXT FEATURE EXTRACTION MODULE

Review text includes the user's review text and the item's review text. Since the feature extraction methods are the same for both, only introduce user's review text feature extraction is introduced.

1) WORD EMBEDDING LAYER

The word embedding layer is used to transform the sequence of the input text into a low-dimensional vector output. The Bidirectional Encoder Representation from Transformers (BERT) is used because BERT is composed of multiple transformer overlays, which can solve the problem of multiple meanings of a word; also, BERT can selectively utilize information from all layers, allowing the multi-layer properties of words to be exploited [30].

First, the user review text is represented as $W = [w_1, w_2, \ldots, w_l]$, where l is the length of the review text. Then, the review text is input into the BERT, and the low-dimensional vector is obtained by an encoder, which denotes the vector as $S = [s_1, s_2, \ldots, s_l]$. In particular, since the number of review texts differs for each user, a fixed-length strategy is used to select only a fixed number of review texts. In this case, the operation of truncation is done for review texts that exceed the fixed length, and the operation of zero-vector complementation is used for review texts that do not reach the fixed length.

2) VECTOR ENCODING LAYER

The vector encoding layer extracts the embedded low-dimensional vectors for hidden features. When the encoding layer encodes the word vectors, it needs to include contextual information. The typical encoder only keeps the data content of the current moment and ignores the data content of the last moments, which can significantly increase the prediction error. To overcome this problem, Bi-directional Gated Recurrent Unit (Bi-GRU) [31] is used to encode the word vectors. Bi-GRU consists of two one-way GRU models with opposite directions. Finally, the vectors of the two directions are stitched together to fuse the positive and negative features to obtain a complete feature of the text. Such a model structure can not only obtain the following information of the text, but also capture the above information of the text which can link the output state of the current moment with the previous and following states, and is more conducive to the extraction of deep-level features of the text.

First, Bi-GRU is used for forward and backward encoding of low-dimensional vectors of words, as shown in equations (1) and (2). The forward encoding performs feature extraction in the order from vector to S_1 vector S_l, and the backward encoding performs feature extraction in the order from vector S_l to vector S_1.

$$h_{f_i} = GRU_{forward}(s_i), \quad i \in [1, 2, \ldots, l] \tag{1}$$

$$h_{b_i} = GRU_{backward}(s_i), \quad i \in [1, 2, \ldots, l] \tag{2}$$

Then, the forward features h_{f_i} and backward features h_b are concatenated to obtain the overall hidden features h of Bi-GRU for each review, as shown in equation (3).

$$H_i = h_{f_i} \oplus h_{b_i}, \quad i \in [1, 2, \ldots, l] \tag{3}$$

Finally, the overall hidden features of each review are integrated and output, denoted as $H = [h_1, h_2, \ldots, h_l]$.

3) ATTENTION LAYER

The attention layer assigns different attention weights to each review [32], which is used to determine the importance of each review and enables the model to extract useful review texts better. Calculate the attention distribution α of the

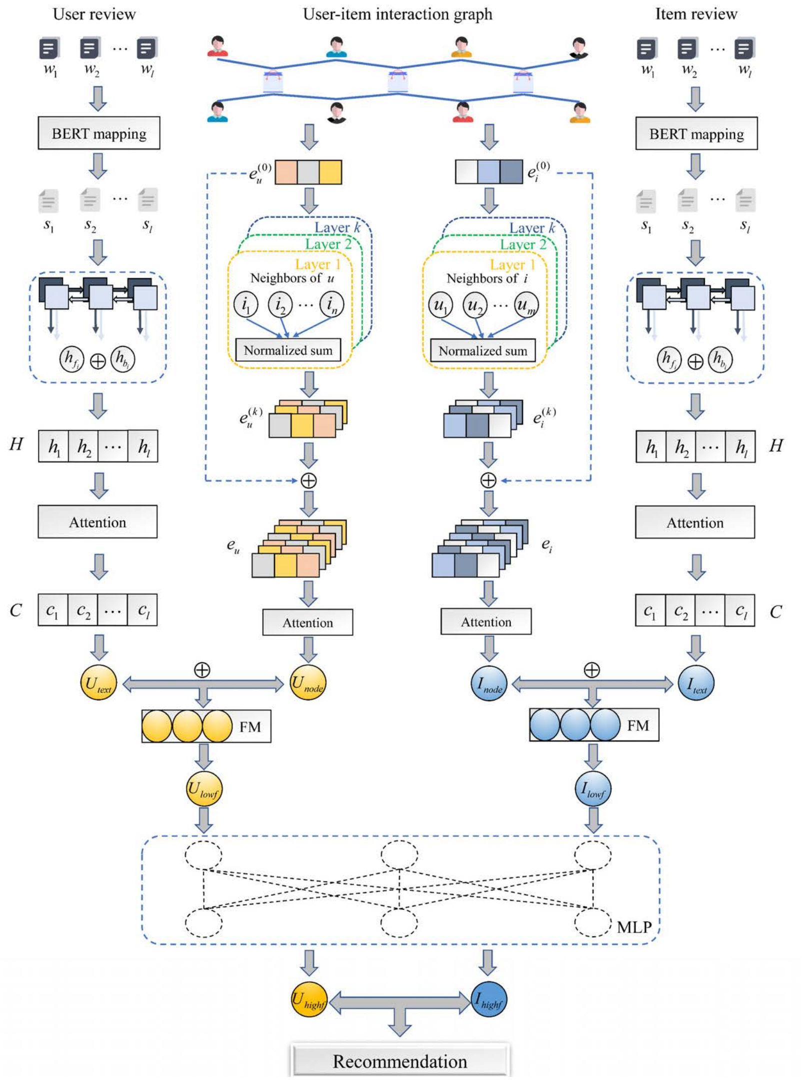


FIGURE 1. The overall framework of the proposed model.

importance of each user review text, as shown in equation (4).

$$A = softmax(\omega_q \times \tanh(\omega_k \times H^{\mathrm{T}})) \tag{4}$$

where ω_q is the weight matrix of *query* in the attention mechanism, ω_k is the weight matrix of *key* in the attention mechanism, softmax function is the normalization operation, and H is the hidden feature of Bi-GRU output.

The hidden features provided by the vector encoding layer are weighted and summed according to the attention distribution, and represent the overall features of the user's review text as $C = [c_1, c_2, \ldots, c_l]$, calculated as shown in equation (5).

$$C = \alpha H \tag{5}$$

4) FEATURE MAPPING LAYER

The feature mapping layer takes the features C output from the attention layer as input and maps the final features U_{text} of the user review text, as shown in equation (6).

$$U_{text} = \omega_0 \times C + b_0 \tag{6}$$

where ω_0 is the weight parameter of the feature mapping layer and b_0 is the bias term of the feature mapping layer.

Similarly, the same method is used to obtain the final features of the item review text I_{text}.

B. NODE FEATURE EXTRACTION MODULE

To better extract the hidden higher-order features and obtain the correlations between interaction information, GNN combined with the attention mechanism is used to extract the node features of the user-item interaction graph. Similarly, since node features of the user and node features of the item in the interaction graph are extracted similarly, only the node feature extraction method for the user is introduced.

1) ID FEATURE EMBEDDING LAYER

The ID feature embedding layer embeds each user and item into a dense vector by their respective IDs. If there are M users and N items, the initial embedding vector of users is denoted as $e_u^{(0)} = [e_{u_1}^{(0)}, e_{u_2,\ldots,}^{(0)} e_{u_m}^{(0)}]$, and the initial embedding vector of items is denoted as $e_i^{(0)} = [e_{i_1}^{(0)}, e_{i_2,\ldots,}^{(0)} e_{i_m}^{(0)}]$.

2) FORWARD PROPAGATION LAYER

The forward propagation layer computes higher-order features for each user and item. The neighbor nodes are aggregated by GCN [33] and the forward propagation computation is performed. First, the initial ID embeddings of the item nodes in all neighboring nodes of user u is aggregated. Thus, the first layer embedding expression of user u in the GCN is obtained, as shown in equation (7).

$$e_u^{(1)} = \sum\nolimits_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_i^{(0)} \tag{7}$$

where $e_u^{(1)}$ denotes the first-order feature of user u on the first layer GCN, N_u denotes the set of neighboring nodes of user u, and N_I denotes the set of neighboring nodes of item i.

After modeling the first-order relationship features between users and items using GCN, the same propagation approach is taken to obtain higher-order features, as shown in equation (8).

$$e_u^{(k)} = \sum\nolimits_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_i^{(k-1)} \tag{8}$$

where k and $k-1$ are the number of network layers of the GCN.

Finally, the user features obtained from each layer are stitched to obtain the final expression $e_u = e_u^{(0)} \oplus e_u^{(1)} \oplus \ldots \oplus e_u^{(k)}$.

3) ATTENTION LAYER

The attention layer assigns different weights to each layer embedding, thus determining the importance of each layer embedding. The attention distribution for each layer of embedding is calculated, as shown in equation (9).

$$\beta = softmax(\omega_{\hat{q}} \times \tanh(\omega_{\hat{k}} \times e_u^{\mathrm{T}})) \tag{9}$$

where, to distinguish from the attention mechanism of the review text, the weight matrix of *query* is denoted as $\omega_{\hat{q}}$, and the weight matrix of *key* is denoted as $\omega_{\hat{k}}$.

The attention distribution is used to weigh and sum the embedding vectors of each layer to obtain the final node feature U_{node} of the user in the interaction graph, as shown in equation (10).

$$U_{node} = \beta e_u \tag{10}$$

Similarly, the same method is used to obtain the final features I_{node} of the item review text.

C. FEATURE FUSION MODULE

Since review text and user-item interaction information belong to different modalities, the extracted user and item information belong to multimodal data. If the features of these data are simply stitched together, the model's prediction performance will be significantly reduced. Therefore, the feature fusion module is designed to fuse the feature vectors from multiple sources using the vector corresponding dimensional summation.

First, the review text features and interaction graph node features are stitched using vector corresponding dimension summing to obtain the initial features of the fusion module, as shown in equations (11) and (12).

$$F^U = U_{text} \oplus U_{node} \tag{11}$$

$$I^U = I_{text} \oplus I_{node} \tag{12}$$

Then, the low-order features U_{lowf} is extracted using the Factorization Machines (FM) [34], as shown in equation (13). The FM transforms the operation of F_i^U into $v_{if} F_i^U$ and computes the interaction between any two-dimensional features, enhancing the model's expressiveness.

$$\begin{aligned} U_{lowf} &= \omega_1 + \sum\nolimits_{i=1}^{|F^U|} \omega_i F_i^U + \frac{1}{2}\left(\left(\sum\nolimits_{i=1}^{|F^U|} v_{if} F_i^U\right)^2 \right. \\ &\qquad \left. - \sum\nolimits_{i=1}^{|F^U|} \left(v_{if} F_i^U\right)^2\right) \end{aligned} \tag{13}$$

where ω_1 is the global deviation, F_i^U is the ith variable of F^U, and ω_i is the weight of F_i^U.

Finally, the low-order features U_{lowf} are input into Multilayer Perceptron (MLP) [35], and the fusion interaction is performed by the MLP to obtain the high-order features U_{highf}, as shown in equation (14).

$$\begin{aligned} U_{highf}^{(1)} &= \omega_2^{(1)} U_{lowf} + b_1^{(1)} \\ U_{highf}^{(2)} &= \sigma(\omega_2^{(2)} U_{lowf} + b_1^{(2)}) \\ &\vdots \\ U_{highf}^{(k)} &= \sigma(\omega_2^{(k)} U_{lowf} + b_1^{(k)}) \end{aligned} \tag{14}$$

where $U_{highf}^{(k)}$ is the output vector of the kth layer, $\omega_2^{(k)}$ is the weight vector of the kth layer, $b_1^{(k)}$ is the bias term of the kth layer, and $\sigma(\cdot)$ is the activation function.

Similarly, the same method is used to obtain the higher-order features I_{highf} after the fusion of item review texts.

D. PREDICTION MODULE

The prediction module inner-produces the fused review text and node features to predict their matching scores $\hat{r}_{ui}$, as shown in equation (15).

$$\hat{r}_{ui} = U_{highf}^{\mathrm{T}} \otimes I_{highf} \tag{15}$$

E. OBJECTIVE FUNCTION

Predicting the interaction between users and items is essentially a linear regression problem, so the most common

TABLE 2. Datasets details.

Dataset	Number of users	Number of items	Number of interactions	Data sparsity
Auto	15280	8157	226477	99.82%
Baby	19445	7050	160792	99.88%
SO	33816	17142	533041	99.91%
VG	19412	11924	167597	99.93%
TG	24303	10672	231780	99.84%

squared loss function is used as the objective function to train and optimize the model [36], as shown in equation (16).

$$\text{Loss} = \sum\nolimits_{u,i\in R} (r_{ui} - \hat{r}_{ui})^2 \tag{16}$$

where r_{ui} is the actual rating of item i by user u.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, experimental validation is performed on the Amazon public item dataset, which consists of parameter optimization experiments, performance analysis experiments, and ablation experiments to confirm the effectiveness of RTN-GNNR from multiple aspects.

A. DATASETS

The Amazon dataset, one of the most widely used datasets for item recommendation systems, has a large dataset to support our experiments. Therefore, five datasets with review text in the Amazon dataset are chosen as the datasets for the experiments, namely Automotive (Auto), Baby, Sports & Outdoors (SO), Toys_and_Games (TG), and Video_Games (VG). The number of users, number of items, number of interactions, and data sparsity are shown in Table 2. To ensure feasibility and fairness, each dataset is randomly divided into a training set, a test set, and a validation set in the ratio of 7:2:1, debug the optimal parameters on the validation set, and complete the performance evaluation of the model on the test set.

As seen in Table 2, although the data of each sample is not small, these datasets are sufficient to learn and validate the proposed model because the data volume is large enough. In addition, the sparsity of each dataset is above 99%, which illustrates the significance of our introduction of review text to alleviate sparsity.

B. EXPERIMENTAL SETUP

1) EVALUATION INDICATORS

Since the recommended rating prediction is essentially a regression problem, the Root Mean Square Error (RMSE) and the Mean Square Error (MSE), the most common evaluation metrics, are used for regression problems, as shown in formulas (17) and (18), respectively.

$$\text{RMSE} = \sqrt{\frac{1}{|D|} \sum\nolimits_{u,i\in D} (r_{ui} - \hat{r}_{ui})^2} \tag{17}$$

$$\text{MSE} = \frac{1}{|D|} \sum\nolimits_{u,i\in D} (r_{ui} - \hat{r}_{ui}) \tag{18}$$

where, D is the number of interactions, r_{ui} is the actual rating of item i by user u, and $\hat{r}_{ui}$ is the predicted rating of item i by user u. Smaller values of RMSE and MSE indicate lower prediction error and higher prediction accuracy of the model.

2) BASELINES

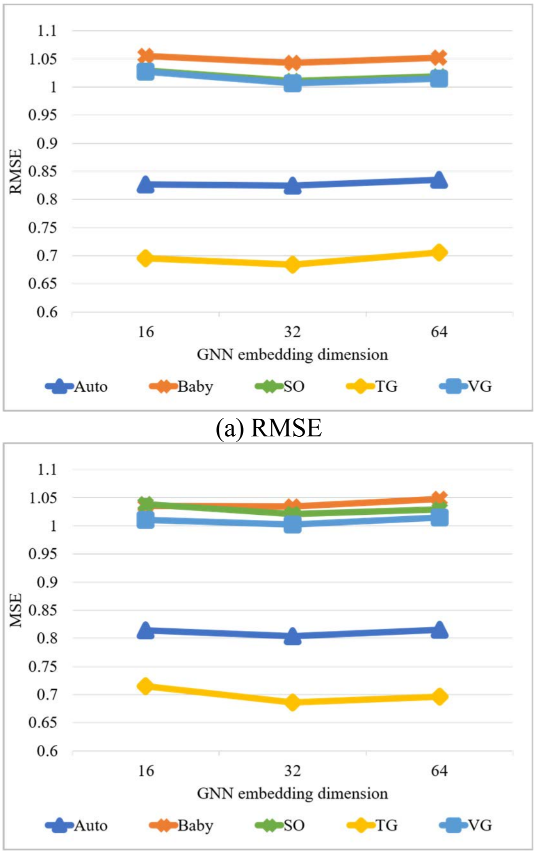
The baselines are divided into three categories: recommendation methods combining review texts (DeepCoNN, ANR, and NRPA), recommendation methods based on GNN only (LightGCN, HANRec, and HA-GNN), and GNN-based recommendation methods fusing attribute features (MEIRec, GraphRec, and IntentGC).

- DeepCoNN [37]: Potential features are extracted from user and item review texts by two parallel CNNs, and rating prediction is performed using FM.
- ANR [38]: The representation of the review text is extracted in an end-to-end manner, and the idea of joint attention is introduced to estimate the importance of users and items.
- NRPA [39]: A neural recommendation method with personalized attention learns personalized representations of users and items from reviews and uses a user-item encoder to learn representations of users and items from reviews.
- LightGCN [18]: Use a GCN to model user-items' higher-order connectivity and simplify the GCN's redundant parts.
- HANRec [19]: A GNN is used to handle heterogeneous attributes, and a component is designed to explore the relationships between potential neighbor nodes.
- HA-GNNN [20]: Dependencies between items are captured using a self-attentive GNN, higher-order relationships in the graph are learned using a soft-attention mechanism, and the embeddings of items are updated using a fully connected layer.
- MEIRec [21]: Fusing user and query attributes with interactive graphs to learn embeddings using heterogeneous graph networks.
- GraphRec [22]: Combining social graph between users and user-item interaction graph for social recommendations.
- IntentGC [23]: Proposing a framework to simultaneously capture the heterogeneous relationship between explicit user preferences and edge information.

3) PARAMETER SETTING

To better improve the recommendation effect of the model, the essential parameters of the model are debugged.

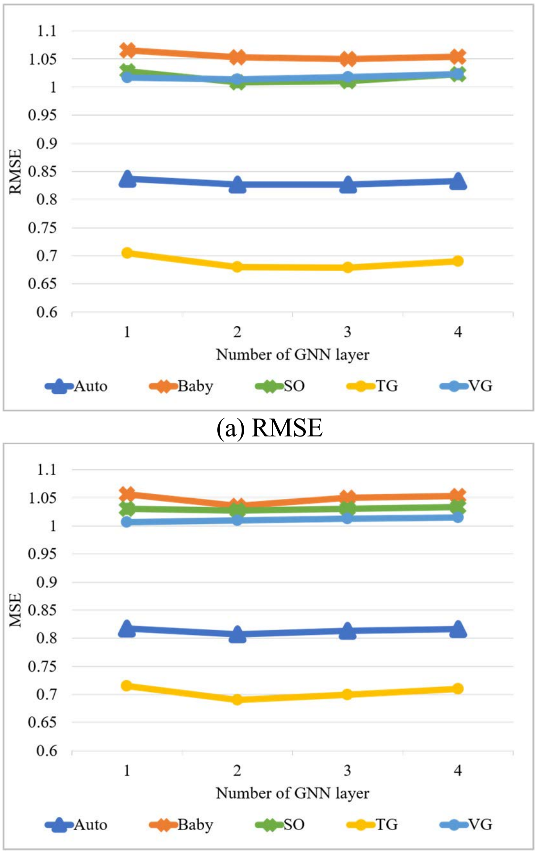
The appropriate GNN embedding dimension is selected in {16,32,64}, and the results are shown in Figure 2. The best result is achieved when the embedding dimension of the GNN is 32. However, the model performance does not change significantly but tends to increase when the embedding dimension is larger, which may be due to the overfitting of the model caused by too large embedding dimension. Therefore, the GNN embedding dimension is set to 32.



(a) RMSE

(b) MSE

FIGURE 2. The effect of the GNN embedding dimension on the model.



(a) RMSE

(b) MSE

FIGURE 3. The effect of the GNN layer on the model.

The appropriate number of GNN layers is selected in {1,2,3,4}, and the results are shown in Figure 3. The number of GNN layers achieves the best results at layer 2. At the same time, deeper GNN does not improve the model's performance much, which may be due to the smoothing problem of the model as the representation between nodes is too similar after multi-layer neighborhood aggregation. Therefore, the number of GNN layers is set to 2.

The appropriate word embedding dimension for the review text is selected in {50,100,200,300}, and the results are shown in Figure 4. As the word embedding dimension increases, the model performance does not achieve significant improvement, which may be because the smaller word embedding dimension is already sufficient to capture the implicit information contained in the reviews. Therefore, to speed up the model's training, the word embedding dimension is set to 50.

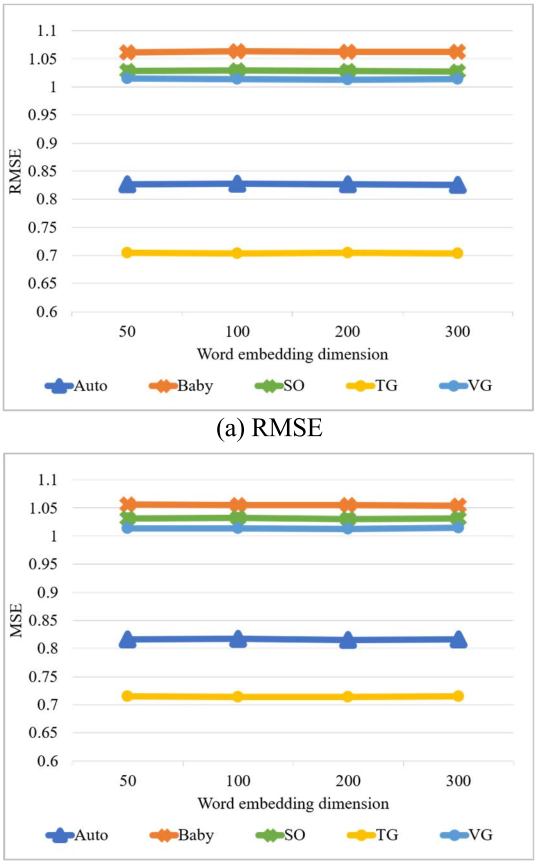
The appropriate number of MLP hidden layers is selected in {1,2,3,4,5}, and the results are shown in Figure 5. The experimental results change significantly with the increase of the number of MLP hidden layers due to the extremely high sparsity of the data itself, which enables the feature vectors of users and goods to be fully interacted with when performing MLP fusion. However, when the number of MLP hidden layer are 3 or even higher, the model performance degrades, which may be due to the repeated interactions that make the model overfitting. Therefore, we set the number of MLP hidden layers to 3.

C. EXPERIMENTAL RESULTS AND COMPARISON

Experiments were conducted with optimal parameters and the optimal parameters for each model were set by corresponding literature and compared the RMSE and MSE of each model under the optimal parameters. The results are shown in Table 3. Among them, the bolded data are the best results in the same group of comparison experiments, the italicized data are the second-best results in the same group of comparison experiments, and the value Improved is the growth ratio of the best compared with the second-best effect. As seen in Table 3, the RTN-GNNR model proposed in this paper has the best overall performance, which is in line with expectations.

To more visually analyze the effectiveness of review information on improving rating prediction and the effectiveness of the RTN-GNNR model proposed in this paper, histograms for each model on five data sets are shown in Figure 6.

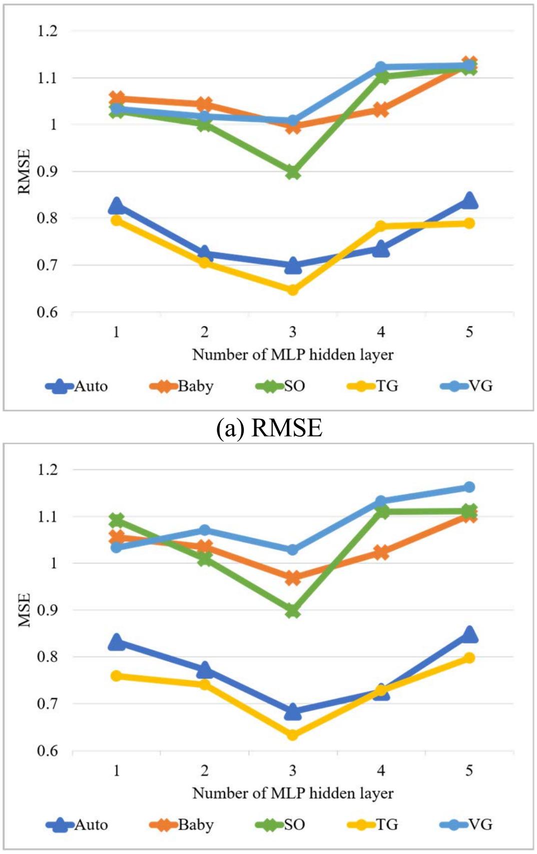
First, recommendation methods based on GNN only (LightGCN, HANRec, and HA-GNN) all achieve acceptable results, even occasionally higher than those combining review texts, which demonstrates the excellent performance of GNN. Specifically, LightGCN simplifies the embedding



(a) RMSE

(b) MSE

FIGURE 4. The effect of the Word embedding dimension on the model.



(a) RMSE

(b) MSE

FIGURE 5. The effect of the MLP hidden layer on the model.

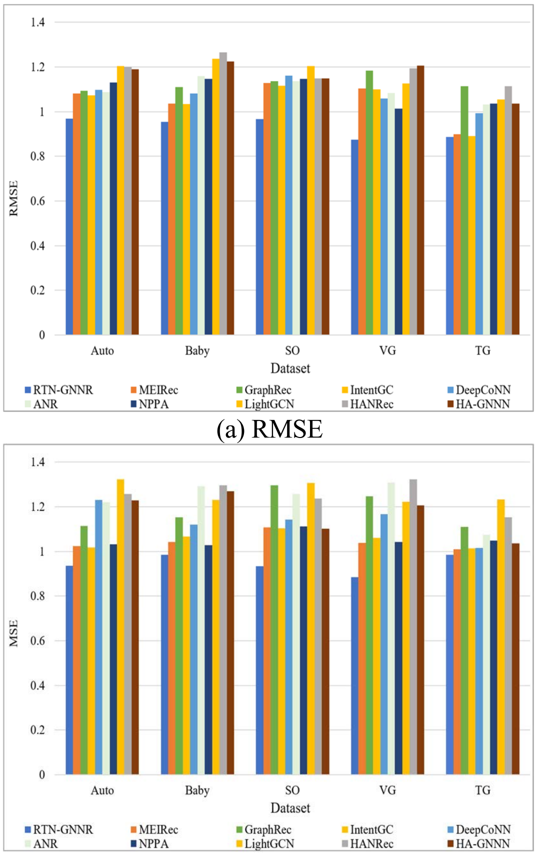
process by removing nonlinear activations and feature transformations but does not consider the importance of each node embedding. HANRec generates each entity embedding for heterogeneous graphs through an attention mechanism, but heterogeneous graphs are essentially multimodal data, and direct embedding leads to inadequate feature fusion. HA-GNN utilizes an attention mechanism to learn hidden features and a fully connected layer to learn the representation of multimodal features, which achieves the best experimental results among the recommendation methods based on GNN only.

Second, for recommendation methods combined review text (DeepCoNN, ANR, and NPRA) achieve better experimental results than the recommendation methods based on GNN only in most cases, which demonstrates that combining review text can tap more potential features to personalize the understanding of each user's preferences than using only rating interaction data, which further illustrates the significance and effectiveness of the proposed model.

Third, GNN-based recommendation methods fusing attribute features (MEIRec, GraphRec, and IntentGC) achieve better experimental results than recommendation methods based on GNN only and recommendation methods that combine review text in most cases. Specifically, MEIRec improves prediction performance by fusing static features and interaction relations. However, its fusion layer is implemented only by MLP without adding additional auxiliary structures, which leads to slow roving and incomplete fusion of features from multiple sources. IntentGC captures the heterogeneous relationship between explicit user preferences and item edge information with good effectiveness and efficiency, achieving the best experimental results among most of the baselines.

Overall, the recommendation methods based on GNN only do not dominate in the dataset with high sparsity and are even lower than the other two types of baselines. The recommendation methods combining review text achieved good results, especially on the two datasets with high sparsity, SO and VG. The GNN-based recommendation methods fusing attribute features achieve the best experimental results in the vast majority of baselines, thanks to their fusion of interaction features with attribute features to reduce the effect of data sparsity.

Finally, the proposed RTN-GNNR works better than the other baselines in each dataset. In particular, the improvement of RTN-GNNR is bigger than several other datasets in SO and VG, two datasets with high sparsity, which proves that RTN-GNNR plays a role in mitigating data sparsity. Meanwhile, by fusing review text features and interaction graph node features, RTN-GNNR can extract



(a) RMSE

(b) MSE

FIGURE 6. Effect comparison histogram.

deeper hidden features in the interaction information and obtain more acceptable user preferences to achieve better recommendation results.

In addition to the two metrics, RMSE and MSE, the running time of each model was tested, as shown in Table 4.

First, the recommendation methods based on GNN only take the most time to run overall, probably because the sparse data makes the invalid sampling of GNNs take more time. In contrast, LightGCN speeds up slightly but not much due to discarding the structure of feature transformation and nonlinear activation. Second, the recommendation methods combined review text have the shortest overall running time, which may be because the structure of the three baselines is designed to help extract features of the review text quickly, thus speeding up. Third, the GNN-based recommendation methods fusing attribute features have an overall running time between the first two types of baselines, among which, IntentGC can process interaction data and item-side information data. Finally, the proposed RTN-GNNR is slower in terms of time spent compared to all the baselines. On the one hand, although the GNN structure is simplified by borrowing ideas from LightGCN, loading the BERT pre-training model and incorporating the attention mechanism leads to slow extraction of node features; on the other hand, the inclusion of a structure for extracting review text and fusing features increases the spatial complexity thus slowing down the runtime.

D. ANALYSIS OF ABLATION EXPERIMENTS

To validate the effectiveness and advantages of the RTN-GNNR, ablation experiments were conducted for the critical parts of the model - the review text feature extraction module, the node feature extraction module, and the feature fusion module, and selected the SO and VG datasets with high sparsity to show the experimental results, the results are shown in Table 5, where RTN-GNNR-gnn is the model with only the node feature extraction module, RTN-GNNR-review is the model with only the review text feature extraction module, and RTN-GNNR-concat is the model with the feature fusion module removed and only simple splicing features.

To show the effect more intuitively, a visual graph of the ablation experiment is shown in Figure 7. First, the model using only node feature extraction outperforms most models in the baseline because RTN-GNNR-gnn uses the attention mechanism on top of GNN to obtain the node feature representation of the final interaction graph, which further optimizes the performance of GNN. Meanwhile, the model using only the review text feature extraction module achieves good recommendation results, because RTN-GNNR-review adopts the model of BERT and Bi-GRU to increase the text feature extraction effect and introduces the attention mechanism to treat each review differently, which improves the prediction accuracy. Finally, after removing the feature fusion module, the model becomes much larger in both RMSE and MSE values, which because RTN-GNNR-concat directly splicing two features of different orders leads to insufficient interaction proves the effectiveness of the proposed feature fusion method.

To further verify the effectiveness and advantages of each module of the RTN-GNNR, separate ablation experiments are conducted on the review text feature extraction module, the node feature extraction module, and the feature fusion module. The experiment results on SO and VG datasets with high sparsity are selected for demonstration.

First, the review text feature extraction module of RTN-GNNR is compared with Word2vec_CNN and BERT_GRU. Word2vec_CNN is the same as models DeepCoNN and NRPA using pre-trained static word vector technique Word2vec combined with a CNN for feature extraction of review text, and BERT_GRU is word vector using the BERT, encoding layer using a Bi-GRU for feature extraction of review text. The results are shown in Table 6. Word2vec_CNN has the worst result, which indicates that Word2vec cannot solve the word multiple meanings, and the CNN is not suitable for dealing with sequence problems, which is one of the reasons for the significant prediction error of previous recommendation models based on the review text. BERT_GRU has a slightly larger error than RTN-GNNR, which proves that the attention mechanism benefits the model's accuracy. RTN-GNNR word embedding layer

TABLE 3. Comparison of experimental results of RTN-GNNR and each model.

	Auto		Baby		SO		VG		TG	
	RMSE	MSE	RMSE	MSE	RMSE	MSE	RMSE	MSE	RMSE	MSE
DeepCoNN	1.097	1.230	1.080	1.119	1.160	1.143	1.058	1.166	0.992	1.016
ANR	1.088	1.221	1.158	1.292	1.137	1.257	1.083	1.309	1.031	1.075
NRPA	1.130	1.031	1.147	*1.027*	1.147	1.112	*1.013*	*1.043*	1.036	1.048
LightGCN	1.203	1.323	1.237	1.230	1.204	1.307	1.125	1.223	1.055	1.232
HANRec	1.199	1.256	1.266	1.296	1.149	1.236	1.193	1.322	1.114	1.152
HA-GNNN	1.190	1.228	1.224	1.269	1.148	*1.101*	1.205	1.205	1.035	1.043
MEIRec	1.081	1.023	1.036	1.042	1.128	1.107	1.103	1.037	0.899	1.010
GraphRec	1.093	1.113	1.109	1.153	1.136	1.296	1.183	1.247	1.114	1.109
IntentGC	*1.073*	*1.017*	*1.033*	1.067	*1.115*	1.103	1.099	1.061	*0.891*	*1.014*
RTN-GNNR	**0.968**	**0.936**	**0.953**	**0.984**	**0.967**	**0.933**	**0.874**	**0.885**	**0.885**	**0.984**
Improved	9.79%	7.96%	7.74%	4.19%	13.27%	15.26%	13.72%	15.15%	6.73%	2.96%

TABLE 4. Comparison of time results of RTN-GNNR and each model.

	Auto	Baby	SO	VG	TG
DeepCoNN	23.97h	23.07h	23.78h	23.05h	23.60h
ANR	23.34h	23.84h	23.46h	23.53h	23.22h
NRPA	23.11h	23.67h	23.43h	23.74h	23.99h
LightGCN	25.02h	25.62h	25.89h	25.26h	25.21h
HANRec	25.92h	25.87h	25.91h	25.98h	25.32h
HA-GNNN	25.73h	25.77h	25.93h	25.34h.	25.73h
MEIRec	24.66h	24.12h	24.79h	24.89h	24.99h
GraphRec	24.23h	24.34h	24.17h	24.33h	24.16h
IntentGC	21.25h	21.36h	22.37h	22.58h	21.37h
RTN-GNNR	24.39h	24.11h	24.76h	24.41h	24.63h

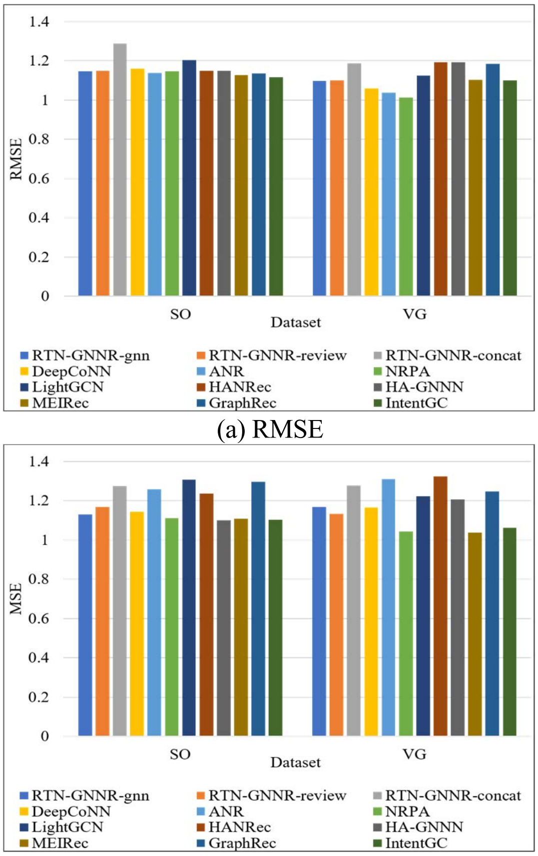
TABLE 5. Ablation analysis.

	SO		VG	
	RMSE	MSE	RMSE	MSE
DeepCoNN	1.160	1.143	1.058	1.166
ANR	1.137	1.257	1.083	1.309
NRPA	1.147	1.112	1.013	1.043
LightGCN	1.204	1.307	1.125	1.223
HANRec	1.149	1.236	1.193	1.322
HA-GNNN	1.148	1.101	1.205	1.205
MEIRec	1.128	1.107	1.103	1.037
GraphRec	1.136	1.296	1.183	1.247
IntentGC	1.115	1.103	1.099	1.061
RTN-GNNR-gnn	1.145	1.131	1.098	1.167
RTN-GNNR-review	1.148	1.167	1.099	1.132
RTN-GNNR-concat	1.288	1.274	1.188	1.278

TABLE 6. Review text feature extraction module ablation analysis.

	SO		VG	
	RSME	MSE	RMSE	MSE
Word2vec_CNN	1.095	1.037	0.996	1.003
BERT_GRU	0.991	0.945	0.895	0.897
RTN-GNNR	0.967	0.933	0.874	0.885

BERT is used, the encoding layer uses a Bi-GRU to extract text features, and the attention mechanism is introduced to treat each review differently, which improves the accuracy of the model prediction.



(a) RMSE

(b) MSE

FIGURE 7. Histogram of ablation analysis.

Second, the node feature extraction module of RTN-GNNR is compared with Case_MEIRec, Case_GraphRec, and Case_IntentGC. Case_MEIRec uses MEIRec's heterogeneous GNN for node feature extraction, Case_GraphRec uses GraphRec's heterogeneous GNN for node feature extraction, and Case_IntentGC uses Case_IntentGC's graph convolutional network for node feature extraction. The results are shown in Table 7 Case_MEIRec has the worst results, which may be because the meta-paths in the designed heterogeneous graphs do not achieve good higher-order feature extraction capabilities. Case_GraphRec has better results

TABLE 7. Node feature extraction module ablation analysis.

	SO		VG	
	RSME	MSE	RMSE	MSE
Case_MEIRec	0.989	0.996	0.946	0.926
Case_GraphRec	0.991	0.980	0.977	0.962
Case_IntentGC	0.973	0.949	0.886	0.893
RTN-GNNR	0.967	0.933	0.874	0.885

TABLE 8. fusion module ablation analysis.

	SO		VG	
	RSME	MSE	RMSE	MSE
Case_FM	1.002	0.950	0.906	0.901
Case_Fully	1.036	0.996	0.937	0.981
RTN-GNNR	0.967	0.933	0.874	0.885

than Case_MEIRec, which may be because the attention mechanism is added to the designed heterogeneous graphs, making the difference between features of different orders more obvious. Case_IntentGC is slightly more effective than RTN-GNNR error because the designed convolutional graph is a double convolutional structure, which can handle the heterogeneity between users and items well and simplify the embedding of users and items. RTN-GNNR simplifies the structure of GCN and uses the attention mechanism, which can better extract the different order features of users and items with an excellent ability to extract features.

Finally, the fusion module of RTN-GNNR is compared with Case_FM and Case_Fully. Case_FM directly inputs homologous and non-homologous features into FM for fusion, and Case_Fully simplifies the model into a fully connected layer without considering low-order and high-order processing and directly splices all features into a vector. The results are shown in Table 8. Case_Fully has the worst results, due to the direct regression without considering the interaction effects of low-order and high-order features. Case_FM has only a slightly larger error compared to RTN-GNNR, due to considering the second-order interaction between users and items, and the second-order interaction is very helpful for multimodal data fusion. RTN-GNNR models based on the second-order interaction between users and users and goods and goods, which can exclude the interference of other second-order interactions and is more suitable for rating prediction.

V. CONCLUSION

In this paper, we propose a GNN recommendation model, called RTN-GNNR, that fuses review text features and node features to make more accurate item recommendations. Specifically, we extract review text features through Bi-GRU combined with attention mechanism, extract node features of user-item interaction graph through GNN combined with attention mechanism, and finally combine FM and MLP to fuse them deeply to achieve a high-performance item recommendation effect. Experiments show that RTN-GNNR can obtain better results than most current methods on five publicly available item datasets from Amazon.

In future research work, we will extend our work in two directions: first, the complexity of the model leads to the low speed of recommendations, especially for machines with insufficient arithmetic power, so we intend to simplify the structure of the model to speed up the recommendations without affecting the results. Second, since the interaction data is too large to be simply randomly sampled, we intend to design a sampling strategy that improves the performance of the recommendation while speeding it up.

REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl. Syst.*, vol. 46, pp. 109–132, Jul. 2013.

[2] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*.

[3] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983.

[4] J. Zhang, X. Shi, S. Zhao, and I. King, "STAR-GCN: Stacked and reconstructed graph convolutional networks for recommender systems," 2019, *arXiv:1905.13129*.

[5] B. Wang, X. Lyu, J. Qu, H. Sun, Z. Pan, and Z. Tang, "GNDD: A graph neural network-based method for drug-disease association prediction," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2019, pp. 1253–1255.

[6] Z. Yang and M. Zhang, "TextOG: A recommendation model for rating prediction based on heterogeneous fusion of review data," *IEEE Access*, vol. 8, pp. 159566–159573, 2020.

[7] A. Breitfuss, K. Errou, A. Kurteva, and A. Fensel, "Representing emotions with knowledge graphs for movie recommendations," *Future Gener. Comput. Syst.*, vol. 125, pp. 715–725, Dec. 2021.

[8] M. Jian, J. Guo, G. Shi, L. Wu, and Z. Wang, "Multimodal collaborative graph for image recommendation," *Appl. Intell.*, vol. 2022, pp. 1–14, Apr. 2022.

[9] Y. Ding and W. Jiang, "Research and analysis of recommendation algorithm based on convolutional neural network," *J. Phys., Conf. Ser.*, vol. 2132, no. 1, 2021, Art. no. 012011.

[10] H. Papadakis, A. Papagrigoriou, C. Panagiotakis, E. Kosmas, and P. Fragopoulou, "Collaborative filtering recommender systems taxonomy," *Knowl. Inf. Syst.*, vol. 64, no. 1, pp. 35–74, Jan. 2022.

[11] Y. Peng, "A survey on modern recommendation system based on big data," 2022, *arXiv:2206.02631*.

[12] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, "A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 25, 2022, doi: 10.1109/TKDE.2022.3145690.

[13] M. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites," *Mach. Learn.*, vol. 27, no. 3, pp. 313–331, 1997.

[14] Y.-C. Zhang, M. Blattner, and Y.-K. Yu, "Heat conduction process on community networks as a recommendation model," *Phys. Rev. Lett.*, vol. 99, no. 15, Oct. 2007, Art. no. 154301.

[15] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 346–353.

[16] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, "Graph contextualized self-attention network for session-based recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3940–3946.

[17] R. Qiu, J. Li, Z. Huang, and H. Yin, "Rethinking the item order in session-based recommendation with graph neural networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 579–588.

[18] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 639–648.

[19] Z. Duan, H. Xu, Y. Huang, J. Feng, and Y. Wang, "Multivariate time series forecasting with transfer entropy graph," 2020, *arXiv:2005.01185*.

[20] S. Sang, N. Liu, W. Li, Z. Zhang, Q. Qin, and W. Yuan, "High-order attentive graph neural network for session-based recommendation," *Applied Intelligence*, vol. 2022, pp. 1–15, Mar. 2022.

[21] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, and Y. Li, "Metapath-guided heterogeneous graph neural network for intent recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2478–2486.

[22] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," 2019, *arXiv:1902.07243*.

[23] J. Zhao, Z. Zhou, Z. Guan, W. Zhao, W. Ning, G. Qiu, and X. He, "IntentGC: A scalable graph convolution framework fusing heterogeneous information for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2347–2357.

[24] N. Jakob, M. C. Müller, I. Gurevych, and S. H. Weber, "Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations," in *Proc. 1st Int. CIKM Workshop Topic-Sentiment Anal. Mass Opinion*, 2009, pp. 57–64.

[25] J. Huang, S. Rogers, and E. Joo, "Improving restaurants by extracting subtopics from yelp reviews," in *Proc. iConf., Social Media Expo*, Berlin, Germany, 2014, pp. 1–5.

[26] Y. Bao, H. Fang, and J. Zhang, "TopicMF: Simultaneously exploiting ratings and reviews for recommendation," in *Proc. AAAI*, Jun. 2014, pp. 2–8.

[27] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. 10th ACM Conf. Recommender Syst.*, Boston, MA, USA, Sep. 2016, pp. 233–240.

[28] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proc. World Wide Web Conf.*, 2018, pp. 1583–1592.

[29] Y. Lu, R. Dong, and B. Smyth, "Coevolutionary recommendation model: Mutual learning between ratings and reviews," in *Proc. World Wide Web Conf.*, 2018, pp. 773–782.

[30] G. Penha and C. Hauff, "What does BERT know about books, movies and music? Probing BERT for conversational recommendation," in *Proc. 14th ACM Conf. Recommender Syst.*, Sep. 2020, pp. 388–397.

[31] Y. Zhang, P. Zhao, Y. Guan, L. Chen, K. Bian, L. Song, B. Cui, and X. Li, "Preference-aware mask for session-based recommendation with bidirectional transformer," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 3412–3416.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, p. 30.

[33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[34] S. Rendle, "Factorization machines with libFM," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 1, p. 22, 2012.

[35] M. Morshedizadeh, M. Kordestani, R. Carriveau, D. S.-K. Ting, and M. Saif, "Power production prediction of wind turbines using a fusion of MLP and ANFIS networks," *IET Renew. Power Gener.*, vol. 12, no. 9, pp. 1025–1033, Jul. 2018.

[36] S. Guo, C. Chen, J. Wang, Y. Liu, K. Xu, Z. Yu, D. Zhang, and D. M. Chiu, "ROD-revenue: Seeking strategies analysis and revenue prediction in ride-on-demand service using multi-source urban data," *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2202–2220, Sep. 2020.

[37] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 1–10.

[38] J. Y. Chin, K. Zhao, S. Joty, and G. Cong, "ANR: Aspect-based neural recommender," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 147–156.

[39] H. Liu, F. Wu, W. Wang, X. Wang, P. Jiao, C. Wu, and X. Xie, "NRPA: Neural recommendation with personalized attention," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, Jul. 2019, pp. 1233–1236.

BOHUAI XIAO received the B.E. degree from the Department of Computer Engineering, Fuzhou University, Fuzhou, China, in 2020. He is currently pursuing the M.S. degree with the Guilin University of Technology, Guilin, China. His research interests include deep learning, graph neural networks, and recommendation systems.

XIAOLAN XIE (Member, IEEE) received the Ph.D. degree in mechanical manufacturing and its automation from Xidian University, Xi'an, China, in 2009.

She is currently the Dean of the Department of Information Science and Engineering, Guilin University of Technology. She has published more than 100 scientific papers, including more than 50 SCI/EI indexed papers. She is hosting some research projects funded from the National Natural Science Foundation of China. Her main research interests include cloud computing, big data, and manufacturing information.

CHENGYONG YANG received the M.S. degree in software engineering from the University of Electronic Science and Technology, Chengdu, China.

He is currently the Deputy Director of the Network and Information Center, Guilin University of Technology. His research interests include cloud computing, the Internet of Things, big data processing, and recommendation algorithm research.

YUHAN WANG received the B.E. degree from the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China, in 2019. He is currently pursuing the M.S. degree with the Guilin University of Technology, Guilin, China. His research interests include deep learning, cloud computing, and recommendation systems.