IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 22, 2020, accepted August 29, 2020, date of publication September 1, 2020, date of current version September 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020942

TextOG: A Recommendation Model for Rating Prediction Based on Heterogeneous Fusion of Review Data

ZHENYU YANG[1,2] **AND MINGGE ZHANG**[2]

[1]School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China
[2]School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China

Corresponding author: Zhenyu Yang (yang_zhenyu@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1404700.

ABSTRACT It is beneficial to use user review as a preference expression because they contain information that is not in the interaction record. However, most current research on recommendation systems only models the explicit records of users and items. It does not mine more personalized information from the review texts generated simultaneously with the interaction records. In this article, we proposed a heterogeneous fusion recommendation model for extracting fine-grained product attributes and user behavior from the review texts. The model we proposed is called TextOG. In the first half of the model, we used two blocks to learn user reviews and item reviews, one of which is dedicated to learning user behavior using reviews written by users, and the other block determines product attributes from reviews written for products. In the second half of the model, we connected the latent factors learned by users and items to perform spatial convolution on the graph. That way, implicit features can perform complex interactions in non-Euclidean spaces. We conducted experiments on a large review data set, and the results show that TextOG performs better than the baseline recommendation methods on various datasets.

INDEX TERMS Recommendation system, review, heterogeneous information, graph convolutional networks.

I. INTRODUCTION

Forecasting based on ratings has always been a research hotspot. Although many models have performed relatively well in scoring predictions, some of the recommendation systems' problems have not yet been solved, such as cold start and insufficient personalization. One reason that can be considered is that the classic models for scoring prediction (such as Matrix Factorization [1]–[3], SVD [4], [5], etc.) only use the explicit results (such as clicks, collections, purchases, etc.) generated by users on items to learn from users. The reason for scoring did not take into account the underlying reason behind the action. The rating can only reflect the user's feelings with the item, but cannot give reasons for satisfaction or dissatisfaction [6], [7].

Many studies have shown that it is not sufficient to use the user's explicit interaction records (such as clicks, favorites, purchases, etc.) to model user preferences to achieve ratings and predictions [8], [9]. Although the interaction record reflects the user's intention to a certain extent, it isn't easy to obtain the reason for the user's rating. For example, a movie that has been given a high rating cannot be determined because of the movie's style or actors. That is to say, even if we can accurately predict the score, it does not give a good explanation. Although there have been many excellent models that can predict users' ratings well, most of them only model users and products based on digital ratings provided by users. It makes these methods ignore the rich information contained in other aspects.

More and more studies begin to focus on how to improve the interpretability of recommendations [10], [11]. The important idea is to use auxiliary information to improve interpretability, such as user reviews and item descriptions. Some studies have shown that using reviews can improve the prediction accuracy of the recommendation system [12]–[15], especially for items and users with low

The associate editor coordinating the review of this manuscript and approving it for publication was Seok-Bum Ko.

ratings. It is very important for improving the interpretability of the recommendation system. The reason for considering text reviews is because reviews are explanations and reasons for the ratings provided by users. Compared to any auxiliary information, they are more credible.

In this paper, we proposed a recommendation system model TextOG based on heterogeneous data fusion. Considering that user reviews and item descriptions contain different information, we used a module for users and a module for projects to model them separately. User reviews and item reviews are provided as inputs to user block and item block, respectively. Then the output of the two modules is further studied in the graph convolutional neural network layer for heterogeneous information. The corresponding score is generated as an output through a shared layer.

We used pre-trained word vectors[1] to prepare the initial word sense features for the model, which helps us to obtain semantic word information from the text through word embedding. We implemented the serialization text layer to study the sentence sequence level of the review text, which is used to discover users and items's characteristics. It has the same representation in the user block and item block, specifically including an ON-LSTM layer and a Mean-pooling layer. Also, the output of the serialized text module allows us to obtain user representations and review representations of items after dimensionality reduction, and as input to the graph convolutional neural network layer to construct a graph structure based on users and items. We used a Graph Convolution Network that can adjust the propagation distance of node messages by controlling hyperparameters. Compared with the directly connected fully connected network layer, such a graph convolution network layer uses fewer parameters, and to a certain extent, reduces the problem of over-smoothing of graph convolution. Finally, we obtained the predicted score through the multi-layer perceptron. After extracting features of user and item reviews in a homogeneous space in parallel, we map them into a graph structure to learn more complex messaging and complex interactions. The contributions of this article are as follows:

1) We proposed a heterogeneous fusion recommendation system model, called TextOG. We first learned the features of the review text from the isomorphic space and then learned more complex interactions in the space of the graph structure. This process allows user and item data to be considered simultaneously in the task of scoring prediction.
2) A neural network (ON-LSTM) that can learn the grammar and structural information of sentences unsupervised are used to encode the reviews, making our feature processing on the text more elaborate.
3) Graph Convolutional Neural Networks are used to implement complex spatial interaction tasks in graph structures. We use the message passing ability of spatial convolution to further enrich the node features on the graph and refine user preferences' granularity.
4) We conducted many experiments on multiple categories of the Amazon review dataset, showing that our model performs better than the baseline models.

[1] https://nlp.stanford.edu/projects/glove/

II. RELATED WORK

In recent years, in the task of score-based prediction, matrix factorization (MF) is a relatively mainstream and first-welcome method. It can model users' explicit feedback (such as clicks, favorites, ratings, etc.) by taking advantage of users' potential characteristics and items. There are many studies based on the matrix decomposition method to achieve the scoring task in the recommendation system, such as [16]–[18]. The work published in 2008 [19] proposed the classic Probabilistic Matrix Factorization (PMF). This medium linear factor model uses user-related coefficients to model user preferences as a linear combination of a series of vectors. The researchers then found that in addition to explicit scoring, users can also use implicit feedback for modeling. Therefore, in addition to explicit scoring, implicit feedback information also contributes to user preference modeling, so SVD++ [20] was subsequently proposed. This method believes that in addition to the user's explicit historical score record for the item, hidden feedback information such as browsing records or favorite lists can also reflect the user's preferences to a certain extent from the side, for example, the user's favorite behavior can reflect his side of the item from the side Interested. However, when the amount of data is large and sparse, the recommendation results are not satisfactory. Another disadvantage of matrix factorization is that it does not provide interpretability for the model. When the user interacts with the item explicitly, although we can predict the user's possible score, we cannot explain what factors are at work. With the development of deep learning technology, new vitality has emerged in the task of scoring prediction. Many researches have shown that they use neural networks to learn the interaction between users and items, and they perform better in scenarios with large amounts of data. ConvMF [21] was proposed in 2016, which combines matrix factorization and neural networks. While making the inner product of the user's hidden vector and the item's hidden vector as close as possible to the original score, it also constrains the item's hidden vector, that is, the item's hidden vector. The characteristics of the documents learned by CNN are as close as possible.

There are various forms and methods to improve the interpretability of recommendations, such as knowledge graphs, social networks, images, text, etc [22], [23]. Among them, the introduction of user comments and item descriptions is one way to improve the interpretability of the model effectively. It is easy to explain, because the comment text given by the user not only contains some attributes and features of the item but more importantly, it contains the reason why the user gave the rating (such as a high rating for a movie because he Is a fan of the movie's actors). There are many

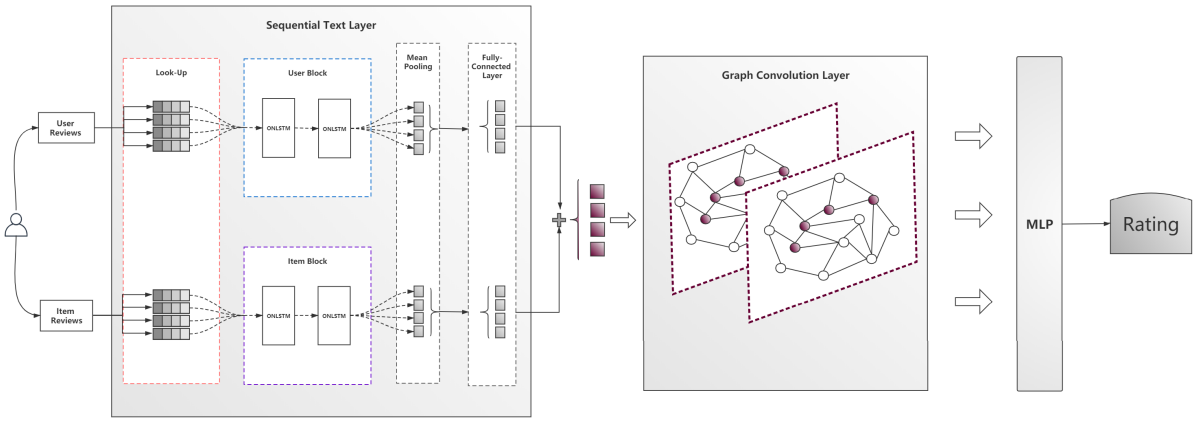


FIGURE 1. The overall structure diagram of the proposed heterogeneous recommendation model (TextOG). The whole model is composed of three main parts: a serialized review text processing module, graph convolution neural network module for learning complex interaction, and final score prediction module.

research results that verify this idea. In [24], a model named DeepCoNN for modeling user reviews and item reviews was first proposed. It uses a convolutional neural network to process the user and item review text in parallel and finally uses a factorization machine to achieve scoring prediction. CNN enables the model to extract richer text features (such as local features of sentences). The ANR model proposed in [25] aimed to model users and commodities from an aspect perspective. It abstracts different aspects into different parameters of the model, enabling the model to extract the aspects in the comment text in an implicit way. In [26], a recommendation model based on comment level (NARRE) was proposed. This model uses the attention mechanism based on the parallel structure of users and items to determine each comment's contribution, that is, to determine which reviews are more useful for modeling user preferences. Formally, these methods all obtain features of score predictions by extracting features from text reviews in parallel. The difference between the above method and ours is that we perform feature extraction on the user and item reviews in a homogeneous space in parallel, and then map it to a graph structure to learn more complex messaging and complex interactions.

III. METHODOLOGY

The recommendation model we proposed, TextOG, is described in detail in this section. TextOG uses reviews to model user behavior and product attributes. It uses the review texts to learn the potential factors of users and items so that the learned factors can estimate the score given by the user. We first use ON-LSTM to separately encode user and product review texts, which allows this layer to learn the hierarchical review syntax structure and semantic information. This part is composed of two parallel neural networks. To further study the feature representation of the user and item information in a more complex non-Euclidean space, we introduce a graph convolutional neural network layer to learn the user's information and item's information from the perspective of the graph. There is a shared layer on top. Train the network to predict the level with the smallest prediction error. We first describe the overall structure of the model and give a model diagram so that we can have an intuitive feeling for the model, and then specify the various components of the model in detail.

A. ARCHITECTURE

The architecture of our proposed model for rating prediction is shown in Fig. 1. The model first uses a module for users and a module for projects to model user descriptions and item reviews. User reviews and items descriptions are provided as inputs to user block and item block, respectively. Then the output of the two modules is further studied in the graph convolutional neural network layer for heterogeneous information, and the corresponding score is generated as an output through a shared layer.

We first define a look-up layer in the first layer of the model, the sequence presentation layer. It is because we can get the semantic information of words from the review text through word embedding. Then we conduct text sequence level learning on the review text, which is used to discover the common layer of user and project features. It has the same representation in the user block and item block, specifically including an ON-LSTM layer, Mean-pooling layer. Besides, the output of the sequence learning module allows us to obtain user representations and review representations of items after dimensionality reduction and serves as input to the graph convolutional neural network layer to construct a graph structure based on users and items. We used a graph convolution network structure that can adjust node messages' propagation distance by controlling hyperparameters. Compared with the directly connected fully connected network layer, such a graph convolution network layer uses fewer parameters, and to a certain extent, reduces the problem of over-smoothing of graph convolution. Finally, we obtain the predicted score through the multi-layer perceptron. The proposed model is shown in Fig. 1.

B. SEQUENTIAL TEXT LAYER

We use a word embedding method: $f : words \rightarrow R^n$, to map all words to an n-dimensional distributed representation vector, where *words* represents a dictionary. In the look-up layer, comments are expressed as word embedding matrices to extract their semantic information. To this end, all reviews given by user u (represented as user comments) are merged into one document $doc^u_{1:n}$, which consists of n words in total. Then, create a word vector matrix represented as $D^u_{1:n}$ for user u:

$$D^u_{1:n} = f(doc^u_1) \oplus f(doc^u_2) \cdots \oplus f(doc^u_n) \tag{1}$$

where, doc^u_i represents the $i-th$ word of user u's document. It is a look-up function that can return the corresponding word vector of words, and $\oplus$ is a concatenation operator.

We used ON-LSTM to calculate at each $D^u_{1:n}$. This layer is composed of k neurons. These neurons can be used to generate new pairs of features for review by applying them on word vectors. ON-LSTM can distinguish high-level and low-level information and specifically sort out disordered neurons, which enables the network layer to learn hierarchical grammatical information based on the meaning of words brought by word vectors. Further, we use a multi-layer ON-LSTM to some extent to learn the semantic learning of the text. It is a manifestation of the powerful representation capabilities of deep neural networks, and many studies have proven this.

The high-level information means that it must be kept in the coding area corresponding to the high-level for a longer time, and it is not easy to be filtered out by the forget gate. In contrast, low-level information means that it is easier to be updated and forgotten. For the coding task of a reviews text, such characteristics can enable the model to filter and update the useless and redundant information in the coding of the review text, while retaining the essential and backbone sentence information. We use the following formula to represent the network structure:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2}$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{3}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{4}$$
$$\hat{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{5}$$
$$\tilde{f}_t = \overrightarrow{cs}(softmax(W_{\tilde{f}} x_t + U_{\tilde{f}} h_{t-1} + b_{\tilde{f}})) \tag{6}$$
$$\tilde{i}_t = \overleftarrow{cs}(softmax(W_{\tilde{i}} x_t + U_{\tilde{i}} h_{t-1} + b_{\tilde{i}})) \tag{7}$$
$$w_t = \tilde{f}_t \otimes \tilde{i}_t \tag{8}$$
$$c_t = w_t \otimes (f_t \otimes c_{t-1} + i_t \otimes \hat{c}_t) + (\tilde{f}_t - w_t) \otimes c_{t-1} + (\tilde{i}_t - w_t) \otimes \hat{c}_t \tag{9}$$
$$h_t = o_t \otimes tanh(c_t) \tag{10}$$

Among them, f_t, i_t, and o_t respectively represent the forget gate, input gate, and output gate, and their inputs are historical information h_{t-1} and current information x_t. σ is a nonlinear activation function. $\overleftarrow{cs}$ and $\overrightarrow{cs}$ represent the left and right sequence element summing operations:

$$\overleftarrow{cs}([x_1, x_2, \cdots, x_n]) = [x_1, x_1 + x_2, \cdots, x_1 + x_2 + \cdots + x_n] \tag{11}$$
$$\overrightarrow{cs}([x_1, x_2, \cdots, x_n]) = [x_1 + x_2 + \cdots + x_n, \cdots, x_n + x_{n-1}, x_n] \tag{12}$$

Using the mean pooling operation, we can perform mean pooling operation on the feature, and take the average value as the feature corresponding to the specific user or item. This merging scheme can naturally handle various lengths of text. After the mean pooling operation, we can reduce each review encoding to a fixed-size vector:

$$p = mean\{h_1, h_2, \cdots, h_{n-t+1}\} \tag{13}$$
$$q = mean\{h_1, h_2, \cdots, h_{n-t+1}\} \tag{14}$$

Among them, p and q represent the user representation and item representation that we obtained through average pooling. According to the average pooled output, we use the output of the fully connected layer as a representation of the users and items we consider:

$$x_u = f(W_1 \times p + g) \tag{15}$$
$$x_i = f(W_2 \times q + g) \tag{16}$$

We splice the node representations of users and items to get the final user representation:

$$U = x_u \oplus x_i \tag{17}$$

We used ON-LSTM to encode user text and item text separately and embed a hierarchical structure through information hierarchy. It allows us to learn a more compact representation of sentence sequences at this layer, and get a node-level representation in the graph convolutional neural network in the next layer.

C. GRAPH CONVOLUTION LAYER

Our goal is to learn a richer representation of users and items. Still, it is not enough to rely on the encoding of text sequences because users and items have more complex relationships and more valuable information in non-Euclidean space. For example, some users have similar preferences or similar attributes. Based on such considerations, we use graph models to further model user and item information. Specifically, we use a graph convolutional neural network layer to receive the user representation and item representation output of the Sequential text layer. To take advantage of the powerful node representation and message propagation capabilities of graph convolutional neural networks, we can achieve our goal by constructing graphs with users as nodes.

We first give the symbolic representations needed in this section to describe the network structure with us clearly.

Given a graph $G = (V, E)$, V represents the set of nodes on the graph, and E represents the set of edges. n and m represent the number of nodes and the number of edges, respectively. $Z \in R^{n \times s}$ represents the feature matrix of nodes, where s

represents the number of features of each node. $L \in R^{n\times c}$ represents the label matrix, and c represents the number of labels. The label matrix is optional; that is, for Graph Convolutional Network, the input may not contain labels. We use the adjoint matrix $A \in R^{n\times n}$ to describe the structure of the graph, and accordingly, we add self-loops to the adjoint matrix,$\tilde{A} = A + I_n$. Based on the above representation, we can give the convolution operation of the graph convolution network:

$$f^{(0)} = H = z_\theta(U) \tag{18}$$

$$f^{(k+1)} = (1-\alpha)\hat{\tilde{A}}f^{(k)} + \alpha H \tag{19}$$

$$f^{(K)} = softmax((1-\alpha)\hat{\tilde{A}}f^{(K-1)} + \alpha H) \tag{20}$$

where $\hat{\tilde{A}} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ is a symmetric normalized matrix with self-loops added. H represents the prediction matrix, which is both the starting vector and the transfer set, K is the number of iteration steps, and $k \in [0, K-2]$. U is the output sample of Sequence Encoder.

Compared with the classical graph convolution method, this method does not need to add additional parameters during the propagation process. We can propagate to a longer distance on the graph through a few parameters, which makes the user between the graph and the item. The connection is closed, and you can learn more abundant interpretable implicit information.

D. RATING PREDICTION AND TRAINING

At the end of the model, we used a two-layer multi-layer perceptron (MLP)to get our score prediction results:

$$g(x) = softmax(b^{(2)} + W^{(2)}(\sigma(b^{(1)} + W^{(1)}G))) \tag{21}$$

Among them, W is a learnable connection weight matrix, b is an offset, σ is sigmoid function, and G is graph convolution layer output. Further, we use pair-wise loss to optimize our model:

$$J = \sum_{i,j\in y\cup y-} (y_{ij}log\hat{y}_{ij} + (1-y_{ij})log(1-\hat{y}_{ij})) \tag{22}$$

So far, we have wholly described each component of the TextOG model. It can be seen that the model as a whole is a combination of parallel and serial to model the interaction between users and items. It is not the same as modeling users and items separately and then making predictions. In our model, the interaction between users and items is closer, and the extracted features are more abundant.

IV. EXPERIMENTAL DESIGN

In this section, we introduced the relevant situation of the experiment. In IV-A, we introduced the basic settings of the experiment, such as the experimental environment, model parameters, etc. In IV-B, we introduced the datasets used in the experiment, in IV-C we introduced the evaluation indicators used to evaluate the model, and in IV-D we described the baseline model we chose.

A. SETTING

We used 70 percent of the data as the training set, 20 percent as the validation set, and 10 percent as the test set. All experimental environments were based on python3.6 and implemented using pytorch1.5. The model was trained and tested on NVIDIA TITAN X GPU.

B. DATASETS

In our experiment, we chose Amazon's review dataset.[2] The dataset contains a large number of product metadata and reviews, is a very large review data set, it contains 21 categories. We selected six types of data to experiment on our model. The datasets we selected are shown in TABLE 1.

TABLE 1. Comparison of the number of reviews for different datasets.

Datasets	Reviews
Movies and TV	1,697,533
Toys and Games	167,597
Grocery and Gourmet Food	151,254
Digital Music	64,706
Office Products	53,258
Musical Instruments	10,261

C. METRICS

Most recommended models use Mean Square Error (MSE) as the evaluation index of the model on the scoring task. Therefore, in order to facilitate comparison, the mean square error is also used in our experiment to evaluate the model.The smaller the value of MSE, the better the performance.

Specifically, Mean Square Error can be defined by the following formula:

$$MeanSquareError = \frac{1}{N}\sum_{n=1}^{N}(r_n - \hat{r}_n)^2 \tag{23}$$

D. BASELINES

To verify our proposed model's effect, we selected some classic models on the score prediction task as the baselines of our method. The baseline method we chose is as follows:

- Matrix Factorization (MF) [27]: This is a scoring prediction method that will be selected when the user data is very sparse. It can effectively predict the location score based on the existing ratings in the incomplete matrix. At present, MF has been highly recognized and widely used in the recommendation system.
- Probabilistic Matrix Factorization (PMF) [19]: A method of matrix factorization. This method obtains user preferences by learning implicit factors of users and items.
- Latent Dirichlet Allocation (LDA) [28]: It is a kind of topic model to learn the topic distribution from several review texts of each item as the potential characteristics of the item. LDA has essential applications in review-based recommendation systems.
- Deep Cooperative Neural Networks (DeepCoNN) [24]: This is a deep model that learns item attributes and user

[2]https://jmcauley.ucsd.edu/data/amazon/

TABLE 2. Comparison results with baseline algorithms.

Methods	Traditional Methods			Homogeneous Methods		Heterogeneous Methods
Datasets	MF	LMF	LDA	DeepCoNN	NARRE	**TextOG**
Movies and TV	1.426	1.414	1.322	1.128	1.118	**0.988**
Toys and Games	1.464	1.400	1.208	1.119	1.066	**1.098**
Grocery and Gourmet Food	1.420	1.239	1.299	1.121	1.110	**1.102**
Digital Music	1.425	1.322	1.259	0.995	1.115	**1.154**
Office Products	1.325	1.331	1.225	1.210	1.126	**1.029**
Musical Instruments	1.252	1.356	1.320	1.100	1.125	**1.095**

behavior from the review text. DeepCoNN is the first model that uses neural networks to model users and items from reviews and has a profound impact on future research on review-based recommendation systems.

- NARRE [26]: It is a typical model for modeling users and objects. It uses a review-level attention mechanism to give different weights to each review. The NARRE model has made a significant contribution to promoting the interpretability research of comment-based recommendation systems.

V. EXPERIMENTAL RESULTS

In this section, we will give experimental results to show the effectiveness of the model. Specifically, we gave the comparison results with the baseline model we selected in V-A. In V-B, we conducted three sets of ablation experiments.

A. COMPARISON RESULTS

The comparison with the baseline model is critical because it objectively shows the improvement of the model. The experimental results are shown in TABLE 2, which contains five baselines and the performance of our model in the six datasets of the Amazon review datasets.

B. ABLATION EXPERIMENTS

In the recommendation model we proposed, there are multiple parameters and conditions. Therefore, to further explain our experimental process, we will look at the results through the following sets of ablation experiments to control the conditions or parameters unchanged. It helps us find which conditions/parameters have a more significant impact on the results.

Specifically, we divided the ablation experiments into three groups. First,we compared the combination of different pooling methods with the encoder of the original review texts, so that we can find the best structure in III-B. Secondly, in V-B2, we conduct a comparative test on the hyperparameter K in III-C, which helps us control the node's receptive field. At the same time, we also compared two typical graph convolution models. Finally, we focus on the performance of homogeneous and heterogeneous models in V-B3. We separately tested the experimental results obtained without the graph convolutional layer. This will provide strong proof of the proposition of our work.

1) DIFFERENT POOLING METHODS AND SENTENCE ENCODERS PERFORMANCE

In the process of feature extraction of reviews, the choice of text encoder is significant. We chose three very typical sequence learning methods (CNN, LSTM, GRU) and the more advanced new method (ON-LSTM) selected in this article to compare the effects. The results show that ON-LSTM can bring better performance, closely related to its ability to learn the hierarchical features in sentences. We used the Movies and TV dataset as an example to compare the encoders of different reviews.

There are many researches and applications on sentence coding of CNN, LSTM, and GRU. Typically, we need to obtain more granular sentence coding information. The effect of CNN is better because it can get local feature information. Although LSTM and GRU have an excellent effect on the overall coding of sentences to a certain extent, they are not applicable in the reviews of the recommendation system. It is because user reviews often contain redundant information, and even a few keywords can express user preferences. At this time, we pay more attention to local information than the overall semantics of the sentence. ON-LSTM achieves grammatical layering, making the ability to extract critical details far beyond neural network encoders such as CNN and LSTM.

We chose to add a pooling layer further to process the sentence information output of the sequence encoder. The pooling layer can play the role of compressing features, removing redundant information, and reducing the amount of calculation in the next step. We conducted a set of comparative experiments to verify the average performance of different pooling methods.

We cross-tested the combined performance of different pooling methods and reviews encoders. The result of the comparison is shown in Fig.2.

2) THE EFFECT OF ITERATION STEPS (K) IN THE GRAPH CONVOLUTION LAYER AND THE PERFORMANCE OF OTHER SPATIAL CONVOLUTION MODELS

In III-C, we introduced the details of the graph convolutional neural network layer, where the iteration step (K) as a hyperparameter can control the receptive field of each node on the graph. In other words, the larger K is, the messages of the current node can be spread to a farther distance, and at the same time, the characteristics of these nodes can be aggregated. Conversely, the closer the propagation distance of the current node is the less aggregated features.

Also, it is necessary to compare different graph convolutional networks. We chose two typical graph convolutional networks for comparison, namely GCN (two layers) [29] and GraphSage [30]. The comparison results are shown in Fig. 3.

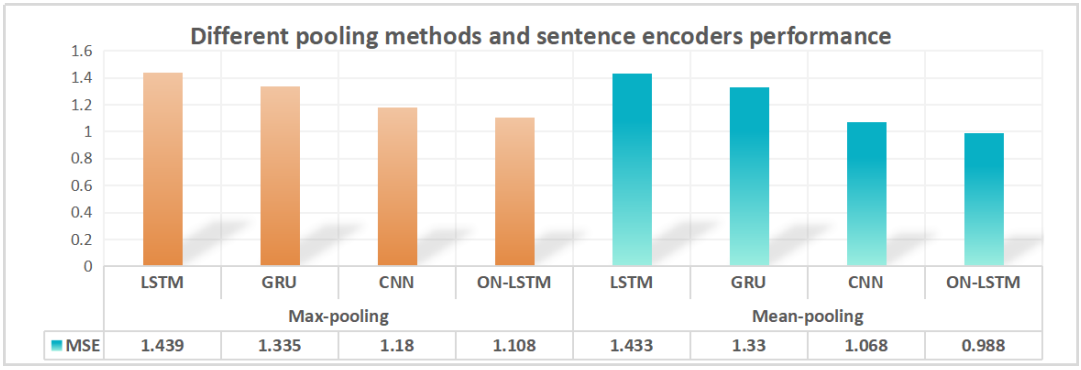


FIGURE 2. Different pooling methods and sentence encoder performance.

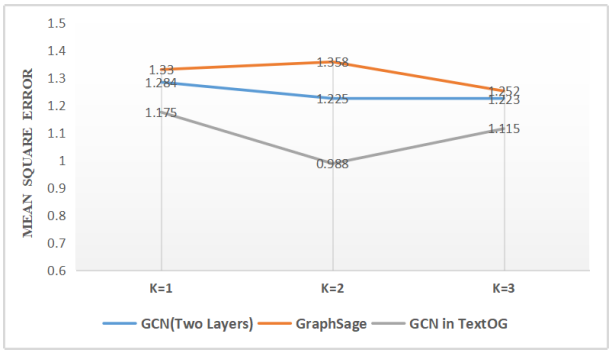


FIGURE 3. Influence of different graph convolutional networks and iterative step K.

3) DIFFERENT PERFORMANCE OF HOMOGENEOUS AND HETEROGENEOUS MODELS

Homogeneous and heterogeneous data models have always been a hot research topic. The current methods are mostly homogeneous models, that is, we treat the data of users and items equally. In our model, we combine two homogenous modules, which are processing the sequence part of the original reviews and the graph convolutional network for processing interactions, so that it has the function of a heterogeneous model. Specifically, we compared the classic homogeneous model, which is the recommended model that resembles a twin-tower structure. Regarding the part of the heterogeneous model, we have already given the experimental results in TABLE 2. Therefore, in this part we will give the experimental results of a homogeneous model. The comparative results of the experiment are shown in TABLE 3.

VI. EXPERIMENTAL DISCUSSION

A. SUMMARY

For our model, as well as tests and experiments, we gave a detailed introduction in V-B. We compared MSE internally (ablation experiment) and externally (baseline algorithms). It is worth mentioning that we can see that our model is superior to the traditional matrix factorization method and the more complex two-channel neural network model. This is because we have more considered the heterogeneous interaction of data and features, rather than simply looking at the modeling of users and items separately.

TABLE 3. Homogeneous model results of double tower structure.

Datasets	Methods		
-	GRU	CNN	ON-LSTM
Movies and TV	1.436	1.324	**1.208**
Toys and Games	1.434	1.410	**1.209**
Grocery and Gourmet Food	1.420	1.239	**1.168**
Digital Music	1.485	1.222	**1.254**
Office Products	1.325	1.255	**1.129**
Musical Instruments	1.252	1.256	**1.209**

B. MITIGATION OF OVER-SMOOTHING PROBLEM OF GRAPH CONVOLUTION

It is necessary to use graph convolutional neural networks. Its high-performance and complex interaction modeling capabilities have been widely recognized by the research community, but it has a drawback, that is, the problem of excessive smoothness between nodes. It can be seen that in V-B2, the performance of a simple two-layer GCN structure is obviously worse than GraphSage. This is because GraphSage alleviates the over-smoothing problem to a certain extent during the sampling process. We choose to solve the smoothing problem by controlling the iteration step K. The advantage of this is that we can expand the receptive field on the basis of solving the smoothing problem. This is a hyperparameter, and we can adjust it according to our experimental needs. It can be seen that the graph convolution structure we used is superior to GCN and GraphSage. Please note that this does not absolutely mean that it is superior to the other two graph convolutional networks, it is only considered in our application scenario.

C. THE IMPORTANCE OF HETEROGENEOUS INFORMATION FUSION

In our work, we used heterogeneous fusion to model user and item information. Heterogeneous fusion can help us reduce the loss of single model information. The separate graph structure will lose the order and grammatical structure information of the review text information, and the separate sequence coding structure will lose the spatial interaction characteristics in the graph structure. In addition, heterogeneous fusion can also help us learn higher-order and more complex non-European spatial interactions.

VII. CONCLUSION

Review contain a lot of information about user preferences, such as item attributes, ratings reasons, etc. Although it is ignored in many recommended models, it has received important considerations in our model. In this article, we proposed a recommendation model TextOG. It makes the model consider the text review provided by the user, and uses the characteristics of message propagation in the graph network to make the user's preference expression more abundant. We dig deeper into the review text, because we retain more granular grammatical and semantic information. We have effectively merged the heterogeneous information extracted from each module. It differs from isomorphic feature extraction only in user and project comment text.

In future work, we will further explore the fusion of heterogeneous data [31] and focus on the fusion of multiple information sources [32] and user comments. It will further advance the quality of recommendations.

REFERENCES

[1] J. Kawale, H. Bui, B. Kveton, T. T. Long, and S. Chawla, "Efficient thompson sampling for online matrix-factorization recommendation," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 1297–1305.

[2] K. Benzi, V. Kalofolias, X. Bresson, and P. Vandergheynst, "Song recommendation with non-negative matrix factorization and graph total variation," Tech. Rep., 2016.

[3] Y. Yu, C. Wang, H. Wang, and Y. Gao, "Attributes coupling based matrix factorization for item recommendation," *Int. J. Speech Technol.*, vol. 46, no. 3, pp. 521–533, Apr. 2017.

[4] L. Cui, W. Huang, Q. Yan, F. R. Yu, Z. Wen, and N. Lu, "A novel context-aware recommendation algorithm with two-level SVD in social networks," *Future Gener. Comput. Syst.*, vol. 86, pp. 1459–1470, Sep. 2018.

[5] M. Rowe, "Transferring semantic categories with vertex kernels: Recommendations with semanticsvd++," in *Proc. Int. Semantic Web Conf.*, 2014, pp. 1–8.

[6] Y. Zhang and C. Xu, "Explainable recommendation: A survey and new perspectives," Tech. Rep., 2018.

[7] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surveys*, vol. 52, no. 1, pp. 1–38, Feb. 2019.

[8] Y. F. Gao, Y. U. Wenzhe, P. F. Chao, Z. L. Zheng, and R. Zhang, "Analyzing reviews for rating prediction and item recommendation," Tech. Rep., 2015.

[9] S. M. Taheri, H. Mahyar, M. Firouzi, E. Ghalebi K., R. Grosu, and A. Movaghar, "Extracting implicit social relation for social recommendation techniques in user rating prediction," in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 1343–1351.

[10] C.-H. Tsai, "An interactive and interpretable interface for diversity in recommender systems," in *Proc. 22nd Int. Conf. Intell. User Interface Companion*, 2017, pp. 225–228.

[11] T. Rutkowski, J. Romanowski, P. Woldan, P. Staszewski, and R. Nielek, "Towards interpretability of the movie recommender based on a neuro-fuzzy approach," in *Proc. Int. Conf. Artif. Intell. Soft Comput.*, 2018, pp. 752–762.

[12] J. Tang, X. Hu, and H. Liu, "Social recommendation: A review," *Social Netw. Anal. Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.

[13] B. Jeon and H. Ahn, "A collaborative filtering system combined with Users' review mining : Application to the recommendation of smartphone apps," *J. Intell. Inf. Syst.*, vol. 21, no. 2, pp. 1–18, Jun. 2015.

[14] L. Qiu, S. Gao, W. Cheng, and J. Guo, "Aspect-based latent factor model by integrating ratings and reviews for recommender system," *Knowl.-Based Syst.*, vol. 110, pp. 233–243, Oct. 2016.

[15] Z. Cheng, X. Chang, L. Zhu, R. C. Kanjirathinkal, and M. Kankanhalli, "MMALFM: Explainable recommendation by leveraging reviews and images," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–28, Mar. 2019.

[16] G. Adomavicius and Y. Kwon, "New recommendation techniques for multicriteria rating systems," *IEEE Intell. Syst.*, vol. 22, no. 3, pp. 48–55, May 2007.

[17] Y. Shi, M. Larson, and A. Hanjalic, "Mining contextual movie similarity with matrix factorization for context-aware recommendation," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 1, pp. 1–19, Feb. 2013.

[18] J. Li, Y. Tang, and J. Chen, "Leveraging tagging and rating for recommendation: RMF meets weighted diffusion on tripartite graphs," *Phys. A, Stat. Mech. Appl.*, vol. 483, pp. 398–411, Oct. 2017.

[19] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.

[20] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Trans. Knowl. Discovery Data*, vol. 4, no. 1, pp. 1–24, Jan. 2010.

[21] 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.*, 2016, pp. 233–240.

[22] X. He, T. Chen, M.-Y. Kan, and X. Chen, "Trirank: Review-aware explainable recommendation by modeling aspects," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 1661–1670.

[23] X. Wang, X. He, F. Feng, L. Nie, and T.-S. Chua, "TEM: Tree-enhanced embedding model for explainable recommendation," in *Proc. World Wide Web Conf. World Wide Web*, 2018, pp. 1543–1552.

[24] 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*, 2017, pp. 425–434.

[25] 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.

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

[27] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[28] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

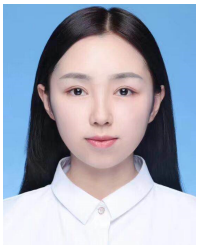
[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[30] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[31] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.

[32] S. Cheng, B. Zhang, G. Zou, M. Huang, and Z. Zhang, "Friend recommendation in social networks based on multi-source information fusion," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 5, pp. 1003–1024, May 2019.

ZHENYU YANG received the B.S. degree in computer application technology from the Qilu University of Technology (Shandong Academy of Sciences), in 2007. He is currently pursuing the M.S. degree in control engineering and theory with the China University of Mining and Technology. He is also an Associate Professor with the Qilu University of Technology (Shandong Academy of Sciences) and the Deputy Director of the Software Integration Institute. His research interests include artificial intelligence, knowledge management, and information integration.



MINGGE ZHANG received the B.Sc. degree in administration from the Tianjin University of Science and Technology, in 2018. She is currently pursuing the M.S. degree in software engineering with the Qilu University of Technology (Shandong Academy of Sciences). Her research interests include deep learning, interpretable reasoning, and recommendation systems.