

## Research Article

# Deep Personalized Medical Recommendations Based on the Integration of Rating Features and Review Sentiment Analysis

Qihang Zhou, Lei Su , Liping Wu, and Di Jiang

Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

Correspondence should be addressed to Lei Su; s28341@hotmail.com

Received 7 January 2021; Revised 30 April 2021; Accepted 17 May 2021; Published 28 May 2021

Academic Editor: Yin Zhang

Copyright © 2021 Qihang Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To comply with the rapid development of big data in mobile services, an increasing number of websites have begun to provide users with recommendation decisions in various areas, like shopping, tourism, food, and medical treatment. However, there are still some challenges in the field of medical recommendation systems, such as the lack of personalized medical recommendations and the problem of data sparseness, which seriously restricts the effectiveness of such recommendations. In this paper, we propose a personalized medical recommendation method based on a convolutional neural network that integrates revised ratings and review text, called revised rating and review based on a convolutional neural network (RR&R-CNN). First, the review text is divided into user and doctor datasets, and BERT vectorized representations are performed on them. Moreover, the original rating features are revised by adding the sentiment analysis values of the review text. Then, the vectorized review text and the revised rating features are spliced together and input into the convolutional neural network to extract the deep nonlinear feature vectors of both users and doctors. Finally, we use a factorization machine for feature interaction. We conduct comparison experiments based on a Yelp dataset in the “Health & Medical” category. The experimental results confirm the conclusion that RR&R-CNN has a better effect compared to a traditional method.

## 1. Introduction

With the advent of the age of information intelligence, mobile computing and recommendation systems are developing rapidly. However, in the face of massive amounts of information, the personalized demands of users are constantly increasing, and thus, the requirements for recommendation results and performance are also increasing. Traditional recommendations face difficulty in satisfying users' needs. Since deep learning can learn deep nonlinear feature representations, it is of great significance to study recommendation systems combined with deep learning [1, 2]. A recommendation system covers all aspects of people's lives. Correspondingly, recommendations for medical services are gradually developing. However, some problems cannot be ignored in the field of medical recommendation systems. (1) Currently, most medical software or platforms only provide an open interactive window for

users to communicate and seek medical treatment online. Moreover, due to the excessive number of experts and the complex needs of users, a platform cannot accurately recommend suitable doctors for users and lacks personalization recommendations [3]. (2) Like those in other fields, medical recommendation systems may also face the problem of sparse rating data [4] since the doctors evaluated by each user only account for a small part of the total number of doctors. The solution to this problem is to add review text [5], as it expresses a user's emotions and explains his/her rating. However, the accuracy of user reviews and ratings is interfered with by users' sentiments. Sometimes, a user provides a poor evaluation of a doctor but gives him/her a high rating. For example, we find a negative description in a 4-star review, which affects the objectivity and accuracy of the evaluation of this doctor. Therefore, it is necessary to combine the original rating with the sentiment analysis value of the review.

To solve the problem of the inaccuracy of rating and static coding of review text in a medical recommendation system, in this paper, the main contributions are shown as follows.

RR&R-CNN uses the revised rating and review text as input to the model. We use the revised rating combining the original rating and the review sentiment analysis to provide better and more objective recommendations. This is mainly considering that, sometimes, a user's ratings and evaluations of doctors do not match, and the user's review text expresses his/her emotional tendencies. Meanwhile, we use the pretrained model—BERT [6]—to obtain the vector representation of the review text. This model can resolve the problem of polysemy to help us understand the semantics and emotions in the review text and improve the accuracy of our rating prediction. In the input layer of the convolutional neural network model, a user's review text of doctors and the revised rating are combined as input features. We conduct experiments on a Yelp dataset, and the experimental results show that RR&R-CNN effectively improves the recommendation effect.

The remainder of this paper is structured as follows. Section 2 presents the related works on recommendation systems based on deep learning, the representation of review text and sentiment analysis, and their applications in recommendation systems. Section 3 describes the components of our network model in detail. In Section 4, we introduce the design of our experiments and the analysis of the experimental results. Finally, we conclude the paper in Section 5.

## 2. Related Work

*2.1. Deep Learning-Based Recommender System.* A recommendation system can help users deal with information overload and provide them with personalized suggestions, content, and services [7]. Traditional recommendation algorithms cannot learn the deep features of users and items by using shallow models. In recent years, deep learning has made rapid progress and development in many fields. It can not only perform deep nonlinear representations on massive user and item datasets but also automatically learn features from multisource heterogeneous data, especially for unstructured data such as text and pictures to extract feature information [8]. Thus, deep learning has become a crucial research hotspot in recommendation systems.

Recommendation systems based on deep learning usually take user and item-related data such as explicit and implicit feedback data as input features, and they generate item recommendations for users from the implicit representation of users and items obtained through the deep learning model [8]. A convolutional neural network is a typical model in deep learning. It is a special feedforward neural network with convolution and pooling operations that can effectively capture global and local features and significantly improve efficiency and accuracy [9]. Convolutional neural networks are widely used in recommendation systems, mainly for learning feature representations from multisource data such as images, text, audio, and video. For review text, convolutional neural networks have been widely used because of their pow-

erful feature extraction ability. Kim et al. [10] used convolutional neural networks to obtain deep hidden vectors from the text information of commodities and used multilayer convolution to obtain the correlation between words, which improved the accuracy of rating prediction in the recommendation system. Zheng et al. [11] proposed a parallel convolutional neural network that processes both user and product text information. A convolutional neural network extracts the deep features of users and products. Then, a factorization machine is introduced as the corresponding rating estimator.

*2.2. Representation of Review Text and Its Application in Recommendation Systems.* Review text usually contains information about a user's emotional expression and product features, which is of great reference value for other users in making decisions [12]. In medical recommendations, user reviews generally include detailed and personalized evaluations of doctors, such as feelings about the medical environment and doctors' abilities and attitudes [13]. In the rating prediction task, we need to extract the necessary information from the review text and convert it into a language that can be recognized by a computer. This task involves the representation method of review text, which converts the text into a vector representation. Widely used vector representation methods include word2vec [14], Glove [15], ELMo [16], and BERT [6].

The core idea of the word2vec model is to obtain a vectorized representation of a word through its context. Although the structure of this method is easy to understand, it can still learn high-quality word embeddings. Glove is based on word2vec considering the use of statistical information in the corpus. The above two methods of expressing word vectors have achieved certain results, but both of them belong to static coding; that is, the same word has the same meaning in different contexts. This may lead to a deviation in the understanding of semantics [17]. The ELMo model, which was proposed in early 2018, dynamically generates word vectors according to context. Its network structure adopts a "two-layer and two-way" long short-term memory (LSTM) network language model, which effectively alleviates the problem of polysemy. In October of the same year, the BERT model was born, which uses dual task pretraining and fine-tuning training strategies to perform pretraining on a large number of unlabeled texts. The BERT model has made substantial progress in 11 NLP tasks [6]. The difference between BERT and ELMo is that the representation of each word in the ELMo model is based on the current entire sentence, while the BERT model considers the full text of the context when generating word vectors. In [18], a pretrained BERT model is used to replace the previous word embedding layer, which can obtain a more accurate semantic understanding and further improve performance.

*2.3. Sentiment Analysis and Its Application in Recommendation Systems.* Emotional factors play a vital role in a user's decision-making process. The feedback provided by users in the face of different types of information reflects their emotional state at the time, and this change in emotional state can directly affect their thinking [19]. Users'

TABLE 1: An example of a user’s review text.

Text for 3 stars
I’ve come here very often and in the past it was great and I loved it but it seems like things have changed a lot. The staff in the front were very nice but the nurses and doctors were rude and impatient. The doctor did not listen to what I was saying and had to repeat myself often. He seemed angry at me the whole time and made me uncomfortable. If it wasn’t for good past experiences I would’ve given 2 stars.

emotions are expressed explicitly through product ratings and implicitly through product reviews. Therefore, a recommendation system can use emotional factors to analyze a user’s behavior and then judge his/her choice in terms of different items [20]. However, Zhang et al. [21] compared user ratings with their sentimental tendencies to review text and found that the former cannot completely replace the latter. For example, as shown in Table 1, the review text expresses the negative sentiment that doctors and nurses are both rude and impatient. A user’s rating of “3 points” is not a true evaluation, but an encouraging sense, which directly affects the accuracy of the doctor’s evaluation. Therefore, it is of great significance to study the combination of rating and review sentiment analysis.

At present, quite a few researchers have proposed related methods. In [22], Abbasi-Moud et al. used semantic clustering and sentiment analysis to extract user preference information from user review text. In [23], Zhao et al. calculated the sentiment deviation of the user’s review text and integrated it into the matrix decomposition to improve the accuracy of the score prediction. Lei et al. [24] proved that users’ social emotions, including users’ emotional similarity, interpersonal emotional influence, and project reputation, play a key role in improving rating prediction. In the location recommendation, Wang et al. [25] proposed a system that could recommend green space to users. They mainly calculated the contents of various tweets into the polarity values that could reflect users’ emotions by using the emotional-relevance dictionary and understood users’ attitudes towards them through the polarity values. Zhao et al. [26] proposed a method for mining points of interest that combines geographic attributes and emotional attributes. The emotional attributes are mainly based on emotional analysis of text information. In [27], Zheng et al. proposed a new preference prediction mechanism from the perspective of users’ subjective emotions, taking into account the influence of changes in user emotions over time on user behavior in sequence recommendation. In addition, Zhang et al. [28] proposed a method of prefiltering opinions using a comprehensive measurement of a user’s emotional orientation and original rating level. This method uses word2vec and LSTM to model the sentiment analysis of the review text to predict the sentiment tendency rating of each user’s reviews of an item, and then, it performs a weighted sum with the original rating to correct the deviation between the original rating and the user’s emotional tendency. The HFT model [29] uses matrix factorization to obtain the latent factors of the scoring value. Moreover, topic distribution was used to mine the latent

topics of review texts. Then, a mapping relationship between latent factors and latent topics was established to achieve model fusion.

### 3. RR&R-CNN: Medical Recommendation Model Based on a Convolutional Neural Network’s Integration of Rating and Review Sentiment Analysis

In this paper, we divide the review text into a user dataset and a doctor dataset. Then, we use a pretrained BERT model to obtain vectorized representation of the review text. Moreover, we combine the original rating data of the user and the doctor with the sentiment analysis value of the review text to revise the rating. Finally, the vectorized representation of the review text and the revised rating is combined as the input data of the convolutional neural network. Therefore, this method can discover the high-level features of users and doctors and provide personalized doctor recommendation services for users.

*3.1. Model Structure.* RR&R-CNN consists of the following three parts, and its specific architecture is shown in Figure 1.

- (1) Input layer: the review text is divided into a user review dataset  $\{\text{User}T_1, \text{User}T_2 \cdots \text{User}T_d\}$  and a doctor review dataset  $\{\text{Doctor}T_1, \text{Doctor}T_2 \cdots \text{Doctor}T_d\}$ , and we use the pretrained BERT model to vectorize the review text; for the rating data, we use the sentiment analysis value of the review text combined with the original rating as the revised rating, and we splice it with the embedding value of the review text
- (2) Feature extraction layer: based on the feature extraction layer of the convolutional neural network, we use parallel CNN to extract the deep features of users and doctors. This layer includes the convolutional layer, pooling layer, and fully connected layer
- (3) Rating prediction layer: we interact the extracted features of the user and the doctor through the factorization machine to obtain the predicted value, and we use the loss function to calculate the error between the predicted value and the true value

*3.2. Preprocessing of the Input Layer.* We perform sentiment analysis and processing on the review text to obtain the sentiment polarity value. We use the TextBlob sentiment analysis tool, which is a tool used to perform natural language processing tasks. The range of its emotional polarity is  $[-1,1]$ , where  $-1$  denotes negative, and  $1$  denotes positive. We use a formula to combine the emotional polarity value obtained from the review text with the original rating [1–5]. The specific combined Equation (1) is as follows [30]:

$$RS_{ij} = \rho R_{ij} + (1 - \rho) S_{ij}, \quad (1)$$

where  $R_{ij}$  represents the original rating of doctor  $j$  by user  $i$ ,

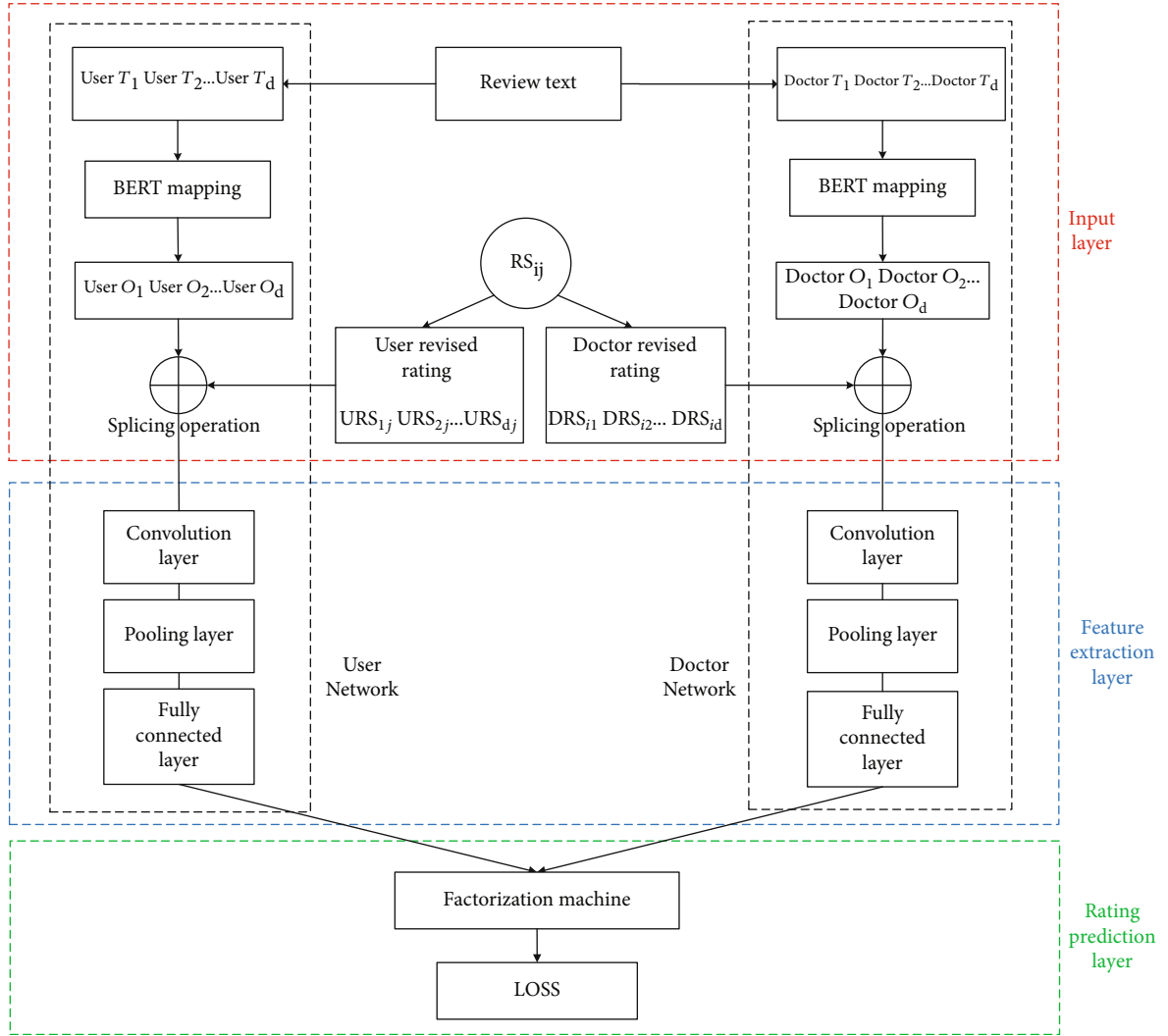


FIGURE 1: Model structure.

and the value is [1–5].  $S_{ij}$  represents the value of the sentiment polarity of the review text calculated by using TextBlob.  $RS_{ij}$  represents the final rating after combining the sentiment analysis value of the review text with the original rating.  $\rho$  is a parameter used to adjust the weight between the original rating and the sentiment polarity value.

We load the pretrained BERT model into RR&R-CNN as a part of the model to update the vector representation of the review text. For user reviews, we convert the user review text  $\{\text{User}T_1, \text{User}T_2 \dots \text{User}T_d\}$  ( $d$  as the maximum number of reviews) into a vector  $\{\text{User}O_1, \text{User}O_2 \dots \text{User}O_d\}$  after being mapped by BERT. The number of embedding dimensions is 768. For doctor reviews, we use the same method. An important reason for using BERT is that it can solve the polysemy problem that cannot be solved by word2vec.

Finally, we use the direct splicing method, that is, splicing the embedded vector corresponding to a sentence with the corresponding revised rating. Compared with the original input dimension, one-dimensional data are added, and the width of the convolution kernel is increased by a factor of

1. We splice the vectorized review texts with the revised ratings as the input of the convolutional neural network.

**3.3. Feature Extraction Layer.** The feature extraction layer mainly uses a convolutional neural network to extract deep features, including a convolutional layer, pooling layer, and fully connected layer.

**3.3.1. Convolutional Layer.** After splicing the sentence vector obtained by BERT with the revised rating feature, the convolution kernel is used to convolve it. The convolutional layer is composed of convolution kernels of different sizes. Because the weights of the convolution kernels are shared, a convolution kernel can extract only a certain type of feature, so we need to use different sizes of convolution kernels, which represent different feature extractors. After calculating a convolution kernel, a column vector is obtained, which represents the features extracted by the convolution kernel from the sentence. Each convolution kernel performs a convolution operation to obtain a feature map. The use of the convolution

operation and the activation function to obtain the corresponding feature map is expressed as Equation (2):

$$c_i = f(w * UO_{i:i+t-1} + b), \quad (2)$$

where  $c_i$  represents the operation result of the convolution kernel.  $w$  is the weight of the convolution kernel.  $b$  is the bias term of the convolution kernel.  $UO_{i:i+t-1}$  represents the vector matrix.  $*$  is the convolution operation.  $f(x)$  is the activation function. In this paper, we use the rule activation function. The representation of all feature maps  $C$  (3) obtained after the convolution operation is as follows:

$$C = \{C_1, C_2, \dots, C_{d-t+1}\}. \quad (3)$$

**3.3.2. Pooling Layer.** Since we use convolution kernels with different heights in the convolution layer, the vector dimensions obtained after convolution will be different. The pooling layer follows the convolution layer, which aims to reduce the number of features, and it decreases the amount of calculation required by reducing dimensions. In this paper, we use the max-pooling layer. The principle of the max-pooling layer is to select the largest value in the feature map. This maximum value represents the most important feature that we capture and the final feature that we want to select. The new feature map discards other unimportant features. Equation (4) is as follows:

$$C = \max(C). \quad (4)$$

After the pooling operation, we obtain a fixed-size feature representation.

**3.3.3. Fully Connected Layer.** We input the results of the pooling layer into the fully connected layer, multiply it by the weight matrix of the fully connected layer, add a bias, and adopt a random dropout strategy, that is, randomly deleting the neurons in the neural network. In each iteration, we only use part of the neuron training model to obtain the values of  $w$  and  $b$  and lose approximately half of the neurons each time, which is equivalent to training on different neural networks. Thus, the dependence of neurons is reduced to solve the problem of overfitting.

**3.4. Prediction Layer.** Although the above outputs can be regarded as the features of users and items, these two outputs may not be directly compared in different feature spaces. Therefore, a prediction layer is set at the end of the model to couple  $U_u$  and  $N_i$ , and the features of user  $u$  and item  $i$  are spliced into a single vector,  $z = (u, i)$ . We introduce the factorization machine [31] to perform feature interaction, and the user's prediction rating for doctors is obtained after multiple trainings. Given  $N$  training samples, Equation (5) is used for training:

$$P = w_0 + \sum_{i=1}^n w_i z_i + \sum_{i=1}^n \sum_{j=i+1}^n z_i z_j \sum_{f=1}^k v_{i,f} v_{j,f}, \quad (5)$$

We use the loss function of the tf.nn.L2\_loss model to train

the target. This function uses the L2 norm to calculate the error value of the tensor. Equation (6) is as follows:

$$\text{Loss}(\text{prediction}, \text{input}_y) = \frac{1}{n} \sum |\text{prediction} - \text{input}_y|^2, \quad (6)$$

where  $\text{input}_y$  represents the true value, and  $\text{prediction}$  represents the predicted value.

## 4. Experimental Results and Analysis

**4.1. Experimental Data Set and Evaluation Metrics.** This experiment uses a Yelp dataset (<https://www.yelp.com/dataset>), which has a total of 66,859 million data points. We randomly select 1,000,000 pieces of data, and then, we find, screen, and analyze the "Health & Medical" data according to the category from these 1,000,000 pieces of data. Finally, the dataset used in this experiment is selected from the medical-related classification dataset, which contains 69,813 reviews made by 57,953 users about 4,658 doctors. The density of the experimental dataset is 0.026%. In the experiment, 80% of the dataset is randomly selected as the training set, 10% is regarded as the validation set, and the other 10% is treated as the test set. The experiment was repeated three times to obtain the average value.

The root mean square error (RMSE) is used as the standard to evaluate the performance of the algorithm, as it evaluates model accuracy by calculating the difference between the predicted rating and true rating, and it is one of the most commonly used evaluation indicators in recommendation systems. Generally, the smaller the value of the RMSE is, the better the recommended effect. Equation (7) is as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^N (r_n - \hat{r}_n)^2}. \quad (7)$$

In the above formula,  $\hat{r}_n$  represents the  $n$ th predicted rating,  $r_n$  represents the  $n$ th true rating, and  $N$  represents the number of samples.

**4.2. Comparison Method.** We select two traditional methods—the latent factor model (LFM) [32] and latent Dirichlet allocation (LDA) [21]—and three deep learning methods—deep cooperative neural networks (DeepCoNN) [11], convolutional matrix factorization (ConvMF) [10], and variants of DeepCoNN (DeepCoNN+rating). First, we select the LFM model that only contains rating features but not the review text information as comparative experiments. The LFM uses matrix factorization to predict the rating of unknown commodities. Second, we select the LDA model that only contains review text information as comparative experiments. LDA is used to learn the document-topic distribution matrix from the review text as the potential feature of the item, and the optimal solution is found through gradient descent for recommendation. Third, we select the DeepCoNN model, a deep learning model, for comparative experiments. DeepCoNN convolutes the user review text and the commodity review text to predict the rating. Fourth, we select a deep learning model, ConvMF, involving ratings and reviews. This

model combines CNN and probabilistic matrix factorization (PMF) to predict rating. Fifth, we add a revised rating feature to the DeepCoNN to obtain the “DeepCoNN + rating” algorithm, which combines the original rating with the sentiment analysis rating of the review, and we use this algorithm to perform comparative experiments. Finally, we improve the fifth method by using the pre-trained BERT model instead of word2vec static encoding for the review text, which can effectively obtain semantic information, and it also revises the original rating by performing sentiment analysis on the review text. This method can effectively improve recommendation performance. The difference between RR&R-CNN and DeepCoNN is that the former uses BERT to vectorize the review text, while the latter uses word2vec; moreover, the former adds the revised rating features that are spliced with the embedded values of the review text to the input layer and inputs them into the convolutional network, whereas the latter only uses review data in the input layer.

*4.3. Experimental Environment and Parameter Settings.* The experimental environment configuration of this paper is as follows. The system version is Ubuntu 18.04, the GPU model is RTX2080Ti/11 GB, and the CUDA version is 10.2. In addition, the deep learning framework is TensorFlow 1.14. The specific experimental parameter configuration is as follows.

*4.3.1. Embedded Representation.* DeepCoNN uses word2vec pretrained vectors—Google News [33]—to embed the review text. The word vector is approximately 100 billion words trained from Google News, containing 3 million words and phrases, and the dimension of the vector is 300. The continuous word bag structure model is used for training, and the vectors are not updated during the training. If there are words in the sentence that are not in the pretrained dictionary, then, they will be used instead of random numbers. ConvMF uses the word embedding word2vec. RR&R-CNN uses the pretrained BERT model—“BERT-Base, cased\_L-12\_H-768\_A-12.” This model uses a 12-layer transformer, the parameter of multihead attention is 12, the total parameter size of the model is 110 MB, the maximum length of the selected review text is 256, and the embedding dimension of the word vector is 768. The optimizer uses Adam, and the initial learning rate is  $\mathbb{N} - 5$ .

*4.3.2. Parameter Setting of Networks.* The RR&R-CNN convolution kernel size is  $t * v$ , where  $t$  represents the number of words contained in the vertical direction of the convolution kernel, the values are 3, 4, and 5, and each size has 100 convolution kernels.  $v$  is the dimension of the embedding word vector. To prevent the model from overfitting, we add a dropout operation, in which the dropout rate is 0.2, to the network structure of the model. The number of latent factors for users and doctors in the factorization machine is 64 from {16, 32, 64, 128}. The L2 regularization parameter in the loss function is 0.6. The values of the parameters in DeepCoNN and ConvMF are consistent with those in the original paper.

*4.3.3. Traditional Model Parameters.* In the comparative experiment of the LDA model, which only uses the review text data, theme dimension  $k$  is set to 10, superparameter  $\alpha$  is 0.2, and  $\beta$  is 0.1. In the comparative experiment of the LFM that only uses rating data, the number of hidden factors is set to 20, the regularization coefficient is 0.01, and the learning rate is 0.1.

#### 4.4. Experimental Results

*4.4.1. Comparison of Experimental Results.* The experimental parameters of RR&R-CNN are as follows:  $\rho$  is the weight between the revised original rating and the sentiment polarity value. The value range of  $\rho$  is {0, 0.1, ...0.9, 1}, and 0.8 is the selected value that obtains the lowest RMSE. The number of convolution kernels is set to 100 from {100, 200, 300, 400}. The number of iterations is 6. The value of hyperparameter  $K$  in the factorization machine is set to 32 from {16, 32, 64, 128}. Table 2 shows a comparison of the experimental results.

The experimental data in Table 2 are repeated three times, and the average value is taken. At the same time, the best results are expressed in *italic*. The experimental results show that in the traditional method, the effect of the LDA model, which only considers the review text, is worse than that of the LFM, which only uses the rating. This shows that only considering the review text does not guarantee that the learned features will be helpful for rating prediction, which means it cannot be completely independent of rating features. Moreover, the performance of deep learning models is better than that of traditional methods, which demonstrates the powerful advantages of deep learning. DeepCoNN model considers the review text of the user and doctor. The ConvMF model uses only the review text of the doctor but adds the rating information of the user, and its effect is better than that of DeepCoNN, which indicates that combination of rating features and comment text is indeed better than using the comment text alone. RR&R-CNN based on the deep learning model, combined with the review text and the revised rating features, reduces prediction error by 4.2% compared with DeepCoNN, which proves that the performance of our recommendation model is better.

To further prove the effectiveness of each module in RR&R-CNN, we show the ablation study—DeepCoNN+rating and DeepCoNN-BERT. In Table 3, DeepCoNN is the baseline, DeepCoNN+rating adds the revised rating feature on the basis of the baseline, and DeepCoNN-BERT replaces word2vec with BERT based on the baseline. The experimental results show that the effect of adding the revised rating feature and using BERT to embed the review text simultaneously is better than that of using either method alone. It demonstrates the combination of the two factors is effectiveness.

*4.4.2. The Value of  $\rho$ .* As shown in Figure 2, we set the value of  $\rho$  from 0 to 1, with a step size of 0.1. When  $\rho$  is 0, it means that only the emotional polarity value of a user’s review text is considered; when  $\rho = 1$ , it means that only

TABLE 2: Comparison of the experiment results from the Yelp dataset.

Algorithm	RMSE
LDA	1.784
LFM	1.689
DeepCoNN	1.653
ConvMF	1.647
DeepCoNN+rating	1.618
RR&R-CNN	1.583

TABLE 3: The ablation study.

Algorithm	RMSE
DeepCoNN	1.653
DeepCoNN-BERT	1.643
DeepCoNN+rating	1.618
RR&R-CNN	1.583

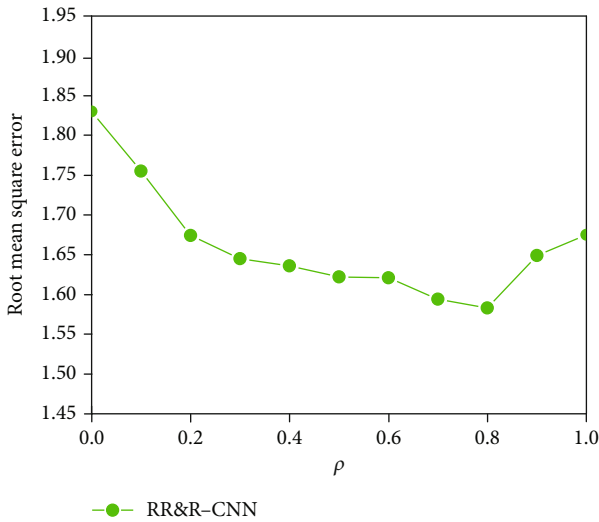


FIGURE 2: RMSE variation with different  $\rho$  values.

a user’s original rating is considered. It can be seen from the data displayed in the figure that when  $\rho$  is 0.8, the RMSE value is the smallest. When  $\rho$  is greater than 0, the corresponding RMSE value decreases to various degrees. This confirms our hypothesis that the use of a combination of the original rating and the sentiment analysis value of the review text, instead of the original rating data, can effectively alleviate the problem in which users sometimes have a low evaluation of doctors but give them high ratings.

4.4.3. *The Number of Convolution Kernels.* As shown in Figure 3, when the value of  $\rho$  is 0.8, the number of selected convolution kernels ranges from 10 to 400. When the number of convolution kernels is 100, the RMSE value is the lowest, and when the number of convolution kernels is greater than 100, the model shows an upward trend; that is, the per-

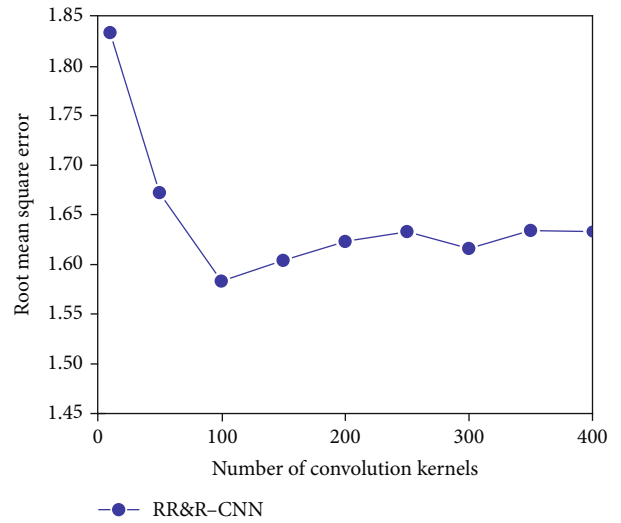


FIGURE 3: RMSE variation with different numbers of convolution kernels.

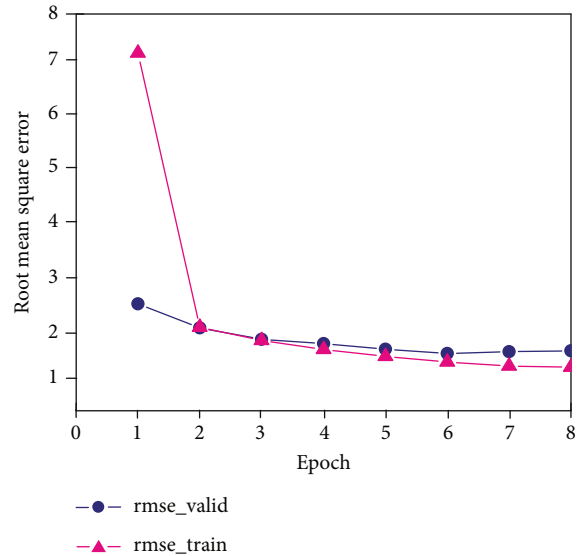


FIGURE 4: RMSE variation with different numbers of iterations.

formance becomes worse, so the number of convolution kernels is set to 100.

4.4.4. *The Number of Iterations.* As shown in Figure 4, we perform various experiments with different iterations. When the epoch is 6, the value of RMSE is the lowest, and both  $rmse_{train}$  and  $rmse_{valid}$  continuously decrease. When the number of epochs is greater than 6,  $rmse_{train}$  decreases continuously, while  $rmse_{valid}$  increases instead of decrease, indicating that the model only fits  $rmse_{train}$  very well, so the epoch value is 6.

4.4.5. *The Value of  $FM_k$ .* As shown in Figure 5, the experiment is carried out for different  $K$  values {16, 32, 64, 128}. When the fixed epoch is selected as 6, the  $k$  value is 32, and the RMSE value is the lowest.

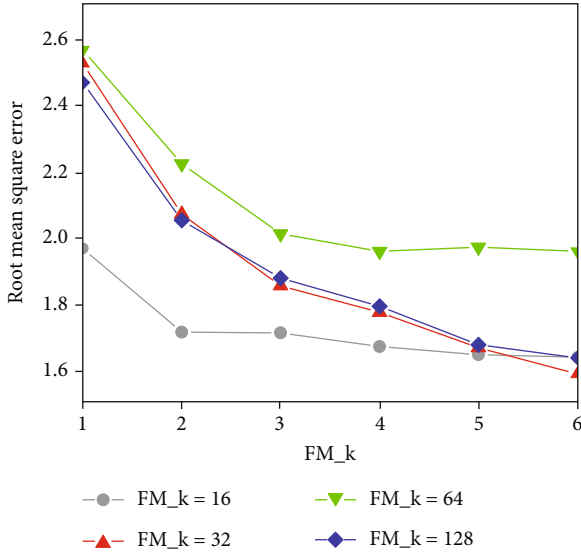


FIGURE 5: RMSE variation with different FM\_k values.

## 5. Conclusions

To solve the problem of how to better recommend doctors according to the personalized demands of users, we propose RR&R-CNN. (1) In contrast to the traditional distributed text representation, RR&R-CNN uses the pretrained BERT model to obtain the vectorized representation of the review text, which solves the problem of polysemy. (2) Regarding the rating features, considering the reasons that, sometimes, a user's evaluation does not match his/her rating, we use the revised user and doctor ratings as the final representation. The revised rating and the representation of the review text are spliced together as the input of the convolutional neural network for deep feature extraction. (3) As the model is divided into a user network and a doctor network, the factorization machine is used to perform the feature interaction between the two networks, and the loss function is used to reduce the calculation error in rating prediction.

The experimental results show that the recommendation effect is improved after adding the revised rating features and that the BERT vectorization of the review text is better than word2vec vectorization. However, a disadvantage of the BERT model is that the time and memory consumed by it are very large. At present, there are some methods for compressing and reducing the complexity of the BERT model. In future work, we will verify this. If it indeed reduces the time and memory consumed by the experiment and improves the recommendation performance, then, we will improve it within the existing method. In addition, BERT, which involves knowledge in the medical field, such as ClinicalBERT [34] and BioBERT [35], will also serve as our follow-up research work.

## Data Availability

This experiment uses the Yelp dataset (<https://www.yelp.com/dataset>).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The work is supported by the National Science Foundation of China under Grant no.61365010.

## References

- [1] R. Mu, "A survey of recommender systems based on deep learning," *IEEE Access*, vol. 6, pp. 69009–69022, 2018.
- [2] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: challenges and remedies," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 1–37, 2019.
- [3] M. S. Hossain, "Cloud-supported cyber-physical localization framework for patients monitoring," *IEEE Systems Journal*, vol. 11, no. 1, pp. 118–127, 2015.
- [4] H. Wang, N. Wang, D. Yeung et al., "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1235–1244, August 2015.
- [5] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend," in *2014 in Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14*, pp. 105–112, 2014.
- [6] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, <http://arxiv.org/abs/1810.04805>.
- [7] G. Adomavicius and A. Tuzhilin, "A toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
- [8] L. W. Huang, B. T. Jiang, S. Y. Lv, Y. B. Liu, and D. Y. Li, "Survey on deep learning based recommender systems," *Chinese Journal of Computers*, vol. 41, no. 7, pp. 1619–1647, 2018.
- [9] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system," *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–38, 2019.
- [10] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proceedings of the 10th ACM conference on recommender systems*, pp. 233–240, September 2016.
- [11] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the tenth ACM international conference on web search and data mining*, pp. 425–434, 2017.
- [12] Y. Wang, M. Wang, and W. Xu, "A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8263704, 9 pages, 2018.
- [13] Y. Qian, Y. Zhang, X. Ma, H. Yu, and L. Peng, "EARS: emotion-aware recommender system based on hybrid information fusion," *Information Fusion*, vol. 46, pp. 141–146, 2018.
- [14] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *International Conference on Learning Representations*, 2013.



- [15] J. Pennington, R. Socher, and C. D. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014.
- [16] M. E. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," in *North American Chapter of the Association for Computational Linguistics*, pp. 2227–2237, 2018.
- [17] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: a survey," *Science China Technological Sciences*, vol. 63, pp. 1872–1897, 2020.
- [18] X.-J. Feng and Y.-Z. Zeng, "Joint deep modeling of rating matrix and reviews for recommendation," *Chinese Journal of Computers*, vol. 43, no. 5, pp. 884–900, 2020.
- [19] M. Tkalcic, A. Kosir, and J. Tasic, *Affective Recommender Systems: The Role of Emotions in Recommender Systems*, CEUR-WS.org, 2011.
- [20] G. Gonzalez, J. L. De La Rosa, M. Montaner, and S. Delfin, "Embedding emotional context in recommender systems," in *2007 IEEE 23rd International Conference on Data Engineering Workshop*, pp. 845–852, Istanbul, Turkey, April 2007.
- [21] Y. Zhang, H. Zhang, M. Zhang, Y. Liu, and S. Ma, "Do users rate or review? Boost phrase-level sentiment labeling with review-level sentiment classification," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 1027–1030, July 2014.
- [22] Z. Abbasi-Moud, H. Vahdat-Nejad, and J. Sadri, "Tourism recommendation system based on semantic clustering and sentiment analysis," *Expert Systems with Applications*, vol. 167, article 114324, 2021.
- [23] G. Zhao, X. Lei, X. Qian, and T. Mei, "Exploring users' internal influence from reviews for social recommendation," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 771–781, 2018.
- [24] X. Lei, X. Qian, and G. Zhao, "Rating prediction based on social sentiment from textual reviews," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1910–1921, 2016.
- [25] J. Wang, Y. Feng, E. Naghizade, L. Rashidi, K. H. Lim, and K. Lee, "Happiness is a choice: sentiment and activity-aware location recommendation," in *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*, pp. 1401–1405, 2018.
- [26] G. Zhao, P. Lou, X. Qian, and X. Hou, "Personalized location recommendation by fusing sentimental and spatial context," *Knowledge-Based Systems*, vol. 196, p. 105849, 2020.
- [27] L. Zheng, N. Guo, W. Chen, J. Yu, and D. Jiang, "Sentiment-guided sequential recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1957–1960, July 2020.
- [28] Y. Zhang, X. Zhu, C. Xu, and S. Dong, "Hybrid recommendation approach based on deep sentiment analysis of user reviews and multi-view collaborative fusion," *Chinese Journal of Computers*, vol. 42, no. 6, pp. 1316–1333, 2019.
- [29] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172, October 2013.
- [30] Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li, "iDoctor: personalized and professionalized medical recommendations based on hybrid matrix factorization," *Future Generation Computer Systems*, vol. 66, pp. 30–35, 2017.
- [31] H. Guo, R. TANG, Y. Ye, Z. Li, and X. He, "Deep FM: a factorization-machine based neural network for CTR prediction," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, August 2017.
- [32] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
- [33] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.
- [34] K. Huang, J. Altosaar, and R. Ranganath, "Clinical BERT: modeling clinical notes and predicting hospital readmission," in *CHIL'20: ACM Conference on Health, Inference, and Learning: Workshop Track*, 2020.
- [35] J. Lee, W. Yoon, S. Kim et al., "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.