Hindawi
Journal of Robotics
Volume 2022, Article ID 1256083, 9 pages
https://doi.org/10.1155/2022/1256083

Research Article

Personalized Product Recommendation Model of Automatic Question Answering Robot Based on Deep Learning

Jie Peng [1] **and Jianhui Xu** [2]

[1] *Center for Faculty Development, Sichuan Engineering Technical College, Deyang, Sichuan 618000, China*
[2] *Department of Economics and Management, Sichuan Engineering Technical College, Deyang, Sichuan 618000, China*

Correspondence should be addressed to Jie Peng; pj@scetc.edu.cn

Received 30 December 2021; Revised 16 February 2022; Accepted 8 March 2022; Published 21 March 2022

Academic Editor: Shan Zhong

The collaborative filtering algorithm widely used in recommendation systems has problems with the sparsity of scoring data and the cold start of new products. A personalized product recommendation model for automated question-answering robots using deep learning is proposed. First, a personalized attention mechanism at the word level and the comment level is proposed, and the comments and users are individually coded. Then, the bidirectional gated recurrent unit (Bi-GRU) is used to construct the score prediction matrix, and through the dynamic collaborative filtering algorithm to integrate the time characteristics of the user's interest changes. Finally, the feature codes of the users and products are input into the Bi-GRU model for learning, so as to output the recommendation list of personalized products of the automated question answering robot. Experimental results based on the JD and Tianchi datasets show that the training loss of the proposed model is lower than 45 and 23, respectively. And HR@15 and MRR@15 exceed 48 and 15, respectively, which are better than other comparison models. It can better adapt to the actual needs of automatic question-answering robots.

1. Introduction

The deep popularization of the Internet and the rapid development of communication technology have enriched online service businesses increasingly. The subsequent data scale also increased sharply, and the information overload was serious. Whether for physical goods or virtual services, the recommendation system is an important technical means to solve the problem of service information data overload [1]. Automatic question-answering technology is a new intelligent retrieval system that allows users to take natural language query as input, and the system can find and return the exact answer from the relevant documents. Especially for self-service platforms, such as automatic question-answering robots, the quality of the recommendation algorithm is directly related to the long-term benefits of its own development [2, 3]. Personalized recommendation is based on the needs of the users, mining products, or services that users are interested in from a large amount of information and according to the score prediction results for personalized display.

The recommendation algorithm is the core of personalized recommendation, which directly determines the recommendation performance of the recommendation system, and has become a hot issue in current research [4]. Reference [5] explored the consumption habits of hyper-personalized health products as unconventional luxury goods. Research shows that consumers think hyper-personalized products are worth their money, whether they want to own them or not. Therefore, potential consumption habits are one of the leading factors in the shopping process. Although the traditional recommendation algorithm has developed to a certain extent, it still has certain shortcomings. Among them, collaborative filtering is a technology commonly used in personalized recommendation systems. The basic idea is to use users with the same interests in the past to choose similar products in the future [6]. However, the algorithm relies too much on historical data, has a cold start problem, and performs poorly in the face of new or unpopular products. A typical collaborative filtering algorithm is a collaborative filtering recommendation model based on matrix factorization. Although this model has good recommendation performance, the sparsity of scoring data has always restricted the bottleneck of traditional collaborative filtering [7]. Reference [8] proposed a novel graph-based ranking-oriented recommendation algorithm based on the influence of paired preferences on recommendation diversity. It uses the users' explicit and implicit feedback to improve the resource allocation model, and matches the target users with users with similar preferences to achieve personalized recommendations. However, due to the structure of the model itself, the cold start problem still exists. In addition, because content-based recommendation only relies on the single content information of the product, the accuracy of recommending mature products is not high. In general, the performance is much lower than the collaborative filtering algorithm. It is usually used as a supplement to the collaborative filtering algorithm when the data are sparse. Reference [9] proposed a new method of personalized recommendation using rule-based semantic reasoning. It can easily and quickly generate practical solutions to personalized recommendations. It establishes a connection between the customer and the store by building a recommendation system to provide seamless information exchange. Reference [10] introduced the design and implementation of a personalized product recommendation model based on user interests. The "shopping basket analysis" function model with the Apriori algorithm as the core uses the sales data in the transaction database. It can dig out all kinds of interesting connections between the products purchased by customers and help businesses develop marketing strategies. The shelves can be reasonably arranged to guide sales and attract more customers. Although the above algorithm has achieved certain results in the field of personalized recommendation, data sparseness and poor handling of heterogeneous data still exist.

With the continuous development of computer technology, deep learning has become increasingly mature and gradually applied to personalized recommendation services. Deep learning can obtain useful information from massive amounts of data to obtain a connection between items and users. Secondly, it is also possible to pass all kinds of data through the same hidden space to ensure the consistent representation of the data [11, 12]. Therefore, deep learning can solve or alleviate the impact of these problems on the performance of the recommendation system, and improve the efficiency and accuracy of information acquisition. Reference [13] designed a personalized recommendation system using machine learning, which can recommend that students strengthen their leadership and become unique among their peers. The model was built using Python flask and Jupyter notebook, and tested using a public dataset and a private dataset. The results show that the model has good accuracy. However, the algorithm does not work well in actual application scenarios. Reference [14] proposed an intelligent humanoid robot with self-learning ability based on deep learning and big data knowledge base to communicate with people. By adopting a deep learning method based on recurrent neural network encoder, convolutional neural network encoder, and bidirectional attention flow, better human–computer interaction performance is achieved. However, it has not been able to improve the quality of personalized product recommendation services for the time being. Reference [15] proposes a personalized ranking of neural graphs. By incorporating the user–item interaction diagram into the embedded learning and directly using the user–item interaction information in the embedded learning, more complex structures can be used in interaction modeling. The experimental results prove the effectiveness of the proposed method. However, the applicability of the platform is poor, and it cannot be generally applied to the robot personalized recommendation service platform for automatic question answering.

In summary, in view of the data sparsity and cold start problems of traditional recommendation models, a personalized recommendation model based on deep learning is proposed. It is applied to the automatic question-answering robot to generate an accurate personalized product recommendation list. Most recommendation algorithms based on reviews ignore the personalized information of users or products. The proposed model uses a personalized attention mechanism to individually encode the users (commodities) and comments. It can extract deep hidden features and effectively reduce the impact of data sparsity.

2. Questions Raised

The recommendation system can accurately locate user interests and commodity characteristics. It is a bridge between the users and commodities, and realizes the perfect match between information producers and information consumers. By analyzing the behavior data generated by users' online consumption, the recommendation system can model users' interests and recommend appropriate goods under unsupervised training, which is more intelligent and personalized. The recommendation system is more and more applied in network services. After querying and browsing movies on the Douban film platform, the platform will recommend movies that users may be interested in according to users' tastes. Today's headline news recommendation system can identify users' interests and hobbies according to the news users usually browse. After users browse Taobao, Taobao's recommendation system will push "guess you like" products to users. Suppose a user wants to buy a pair of basketball shoes from Taobao, and enters keywords in the search bar, the recommendation system will recommend some products that the user may be interested in.

Recommendation algorithm directly affects the performance of the recommendation system. A good recommendation algorithm can help users quickly locate goals, save a lot of time, and improve user experience. Although the recommendation algorithm has been widely used in major websites and software, there is still much room for improvement. Traditional algorithms have shortcomings. For example, the collaborative filtering algorithm relies on historical interactive data, has a cold start problem, performs poorly in the face of new or unpopular products, and the recommendation performance decreases significantly when the interactive data of user products are very sparse. In addition, because content-based recommendation only depends on the single content information of goods, the accuracy of recommendation for mature goods is not high, and the performance is far lower than that of the collaborative filtering algorithm in general.

In order to improve the accuracy of the product recommendation scheme, the proposed model incorporates time series in the collaborative filtering process. This reduces the impact of the information process in the product recommendation process. At the same time, using the Bi-GRU for data learning can further ensure the effectiveness of the automatic question-answering robot recommendation scheme.

3. Proposed Personalized Commodity Recommendation Model

3.1. Overall Framework. The key to the current collaborative filtering product recommendation algorithm is the prediction of product scores, and there are the following three problems: First, there is the problem of sparseness of the scoring matrix. The sparsity of review data is a key factor that affects the accuracy of the final score prediction of the personalized product recommendation system. This is because the number of products on the platform is far greater than the number of comments made by a single user. The comment data present a very obvious sparseness, which in turn affects the insufficient amount of relationship between the users and commodities, and between users and users. The accuracy of the prediction results of commodity ratings has been reduced [16]. Second, the cold start problem. There are two main reasons for the cold start of the recommendation system, namely new users and new products. In the product recommendation platform, new users register every day, and new products are also on the shelves. The system cannot clarify the interest level of the new products or new users [17, 18]. Finally, there is the issue of information expiration. Over time, a user's interest preferences or the popularity of the product will change. Traditional recommendation algorithms do not consider the impact of this change.

To solve the above problems, a personalized product recommendation model based on the Bi-GRU and the dynamic collaborative filtering is proposed. Its overall structure is shown in Figure 1.

First, the text information of user reviews and product reviews is processed by the Bidirectional Encoder Representations from Transformers (BERT) model and the Bi-GRU from the transformer to extract the hidden feature vectors of the users and commodities, respectively. Then, the user's scoring behavior is introduced, and the final product scoring prediction is realized by the TimeSVD++ algorithm. The model uses the Bi-GRU to capture the hidden feature vectors of the users, and combines the image data to reduce the data sparsity and cold start problems in the process of predicting the commodity scoring matrix. Incorporating time series into the collaborative filtering process reduces the impact of the information process in the product recommendation process [19, 20].

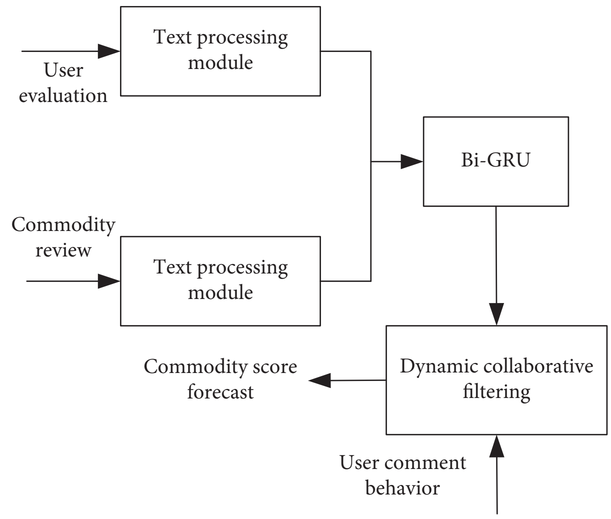

Figure 1: Recommended model architecture.

3.2. User and Product Hidden Feature Extraction

3.2.1. Personalized Comment Encoder. In the user–comment network, a comment set $R_u = \{r_1, r_2, \ldots, r_D\}$ of a user u is given, where D represents the maximum number of comments in the user–comment set. In particular, each comment r_d retains only d' words.

Using the pre-trained word vector, R_u is sent to the word vector mapping layer to obtain $\mathbf{R}_u' \in R^{d\times d'\times k}$, where k is the dimension of the word vector. In order to make the word information comprehensively consider the context information in the forward and backward directions in the comment, $\mathbf{R}_u'$ is sent to the Bi-GRU for encoding, and $\mathbf{H}_u = (h_1, h_2, \ldots, h_d) \in R^{d\times d'\times 2o}$ is obtained. o represents the output dimension of GRU. Since it is the Bi-GRU here, its output dimension is $2o$.

Since each user (commodity) has a unique identity (ID), first, use the first Multilayer Perceptron (MLP) to map the ID into a low-dimensional vector $\mathbf{u}_l \in R^n$. This vector is used to capture the personality information of the user's word level, and its expression is:

$$\mathbf{u}_l = \mathrm{ReLU}\left(\boldsymbol{\omega}_1 u_q + \mathbf{b}_1\right), \quad (1)$$

where ω_1 represents the weight of the first MLP. $\mathbf{b}_1$ is the bias term. u_q is the ID of user u.

Each user's word habits and the polarity expressed by words have individual characteristics when they post comments. In order to make word latent vectors have personalized characteristics, it is first necessary to learn a word level attention vector for a certain user u. The specific calculation is as follows:

$$\mathbf{s}_l = \mathrm{softmax}\left(\mathbf{H}_u \mathbf{P}_l \mathbf{u}_l^T\right), \quad (2)$$

where $\mathbf{u}_l^T \in R^{n\times 1}$ is the transposed vector of $\mathbf{u}_l$. $\mathbf{P}_l \in R^{2o\times n}$ is the transition matrix. $\mathbf{s}_l \in R^{d\times d'\times 1}$ is the attention score corresponding to d' words in d comments. Next, continue to use $\mathbf{s}_l$ with personalized information to adjust the words of the comment, and get the implicit expression of d comments:

$$\mathbf{R} = \mathbf{s}_l^T \otimes \mathbf{H}_u, \tag{3}$$

where transpose the last two dimensions of $s_l \in R^{d\times d'\times 1}$ to get $\mathbf{s}_l^T \in R^{d\times 1\times d'}$. $\otimes$ represents the batch matrix multiplication, such that after $\mathbf{s}_l$ and $\mathbf{H}_u$ are multiplied, the first dimension d remains unchanged, and only the second and third dimension matrix multiplications are performed. Therefore, the implicit expression of the final d comments is $\mathbf{R} \in R^{d\times 1\times 2o}$. In order to facilitate subsequent calculations, the dimension is converted to $\mathbf{R} \in R^{d\times 2o}$.

3.2.2. Personalized User (Commodity) Encoder.

Considering that not all information in $\mathbf{R}$ is conducive to constructing a user preference vector, there is a small amount of irrelevant information. Therefore, before gathering d comments, add a gating mechanism to control the flow of information. Specifically, the input of the gating mechanism is $\mathbf{R}$, and its output is a gating weight matrix $\vartheta \in R^{d\times 2o}$:

$$\vartheta = \sigma\left(\mathbf{R}\boldsymbol{\omega}_\vartheta + \mathbf{b}_\vartheta\right), \tag{4}$$

where σ the sigmoid function. $\omega_\vartheta \in R^{2o\times 2o}$ is the weight matrix. $\mathbf{b}_\vartheta$ is the bias term. Next, use ϑ to control the amount of information that each dimension in $\mathbf{R}$ can flow into the next layer:

$$\mathbf{R}^\vartheta = \mathbf{R} * \vartheta, \tag{5}$$

where $*$ is the multiplication of the corresponding elements. Multiply the corresponding elements in $\mathbf{R}$ and ϑ to obtain the adjusted expression of d comments as $\mathbf{R}^\vartheta \in R^{d\times 2o}$.

In reality, the same expression or similar comments will produce different emotional polarities for different users [21, 22]. In order to be able to gather d comments on the user's preference vector based on the user's personalized information, first, use the second MLP to map the user ID to a low-dimensional vector $\mathbf{u}_r \in R^n$ of the review level:

$$\mathbf{u}_r = \mathrm{ReLU}\left(\boldsymbol{\omega}_2 u_q + \mathbf{b}_2\right), \tag{6}$$

where ω_2 is the weight matrix of the second MLP. $\mathbf{b}_2$ is the bias term. Since different reviews have different contributions to the modeling of user preferences [23], it is necessary to learn the personalized attention vector of the review level:

$$\mathbf{s}_r = \mathrm{softmax}\left(\mathbf{R}^\vartheta \mathbf{P}_r \mathbf{u}_r^T\right), \tag{7}$$

where $\mathbf{u}_r^T \in R^{n\times 1}$ is the transposed vector of $\mathbf{u}_r$. $\mathbf{P}_r \in R^{2o\times n}$ is the transition matrix. $\mathbf{s}_r \in R^{d\times 1}$ is the attention score of each comment. Next, according to the attention score, the user preference vector $\mathbf{U} \in R^{1\times 2o}$ can be obtained by gathering each comment:

$$\mathbf{U} = \mathbf{s}_r^T \mathbf{R}^\vartheta. \tag{8}$$

The above has introduced the processing process from the user–comment set to the user preference vector $\mathbf{U}$ in the user–comment network. Similarly, in the product review network, the product feature vector $\mathbf{I} \in R^{1\times 2o}$ can also be obtained from the product review collection.

3.3. Commodity Recommendation Model Based on Dynamic Collaborative Filtering.

The dynamic collaborative filtering algorithm is based on the Singular Value Decomposition (SVD)++ algorithm and adds time series items, and hence called the TimeSVD++ algorithm. The TimeSVD++ algorithm is evolved from a simple factorization model. Assume that the user's score prediction matrix $\mathbf{y} \in R^{N\times M}$ for the product has been obtained. N and M, respectively, represent the number of users and the number of products. y_{ui} in the matrix represents the predicted value of user u's rating of product i.

The deep features of the users and products obtained by the Bi-GRU processing review text information are F_U and F_I, respectively. After coupling with the sharing layer and using a factorization machine, the predicted value $\hat{y}_{ui}$ of the user's scoring matrix for the product is obtained. The TimeSVD++ algorithm first needs to reduce the error between $\hat{y}_{ui}$ and y_{ui} to obtain the best score prediction. The specific process can be expressed as:

$$\begin{gathered}\hat{y}_{ui} = F_I F_U \\ \min Q = \sum_{y_{ui}} \left(\hat{y}_{ui} - y_{ui}\right)^2.\end{gathered} \tag{9}$$

Adding the bias term in the scoring prediction process constitutes the SVD model:

$$\hat{y}_{ui} = F_I F_U + \delta + b_u + b_i. \tag{10}$$

where δ is the average value of the predicted value of the user's product rating during the training process. b_u and b_i represent the user bias item and the product bias item, respectively, and represent the average value of the score prediction value of a user or a product.

Based on the SVD model, the SVD++ algorithm adds user interest information through implicit feedback information. In other words, as long as any user has commented on a certain product, no matter how high or low the scoring prediction value of the review content is, it means that the user is interested in the product. The degree of interest is expressed as $\gamma_j = \left\{\gamma_{j1}, \gamma_{j2}, \ldots, \gamma_{jQ}\right\}$ by a hidden factor. At this time, the user's rating prediction model for the product is revised as follows:

$$\hat{y}_{ui} = \left[F_U + \frac{\sum_{j\in N(u)} \gamma_j}{\sqrt{|N(u)|}}\right] F_I + \delta + b_u + b_i, \tag{11}$$

where $N(u)$ represents the set of all products that the user u has evaluated.

The basic idea of the TimeSVD++ algorithm is that with the passage of time, the user's preference for the product changes, i.e., F_U, b_u, and b_i in equation (11) are no longer fixed quantities, but a function of time. However, the product feature vector F_I does not change with time. It can be seen that the TimeSVD++ algorithm incorporates time series items. Therefore, it is necessary to divide multiple time periods along the time axis in the process of predicting the user's rating of the product. Predict product scores in various time periods [24, 25]. The divided time period is represented by $e(t)$. The values of F_U, b_u, and b_i are the same in the same time period, but are different in different time periods. The scoring prediction process of the TimeSVD++ algorithm considering time series items is as follows:

$$\hat{y}_{ui} = \left[F_U(t) + \frac{\sum_{j\in N(u)} \gamma_j}{\sqrt{|N(u)|}} \right] F_I + \delta + b_u(t) + b_i(t), \tag{12}$$

where $b_i(t)$ and $b_u(t)$, respectively, represent the bias of the product and the user at time T, and both consist of a static part and a dynamic part, namely:

$$\begin{aligned} b_i(t) &= b_i + b_{i,e(t)} \\ b_u(t) &= b_u + \beta_u \cdot \operatorname{sign}(t - \bar{s}_u) \cdot |t - \bar{s}_u|^{\beta}, \end{aligned} \tag{13}$$

where $b_{i,e(t)}$ represents the offset of the product in the time period $e(t)$. $\bar{s}_u$ represents the average value of all the ratings given by the user. β represents dynamic weight.

The TimeSVD++ algorithm minimizes the error between the predicted value and the true value after obtaining the predicted value of the score. The calculation is as follows:

$$\begin{aligned} \min Q = \sum_{y_{ui}} (\hat{y}_{ui} - y_{ui})^2 + \\ \lambda\left[\|F_U\|^2 + \|F_I\|^2 + \|b_u(t)\|^2 + \|b_i(t)\|^2 + \|\gamma_j\|^2 \right], \end{aligned} \tag{14}$$

where λ is the conversion factor.

4. Experiment and Analysis

In order to verify the effect of the proposed model, an experimental platform was built using the deep learning framework Tensorflow provided by Google. The framework integrates in-depth models such as GRU, which can make development simple and easy to understand. Therefore, it has become a more popular learning framework. A three-layer self-attention network will be used in the experiment, and other experimental parameter values used are shown in Table 1.

4.1. Dataset. The experiment uses two real datasets, namely the JD dataset under JD and the Tianchi dataset under Alibaba Cloud. The JD dataset is provided by the e-commerce company Jingdong, one of the largest online B2C retailers in the country. It contains 370,878,895 interactions with 28,710 products from 105,180 customers in 75 days. The Tianchi dataset is a public dataset provided by the Ali Mobile recommendation algorithm. It is a real user–commodity dataset based on Alibaba's mobile commerce platform. This dataset provides 23291027 interactions of 20,000 customers on 4,758,484 products in one month.

Table 1: Experimental training parameters.

Experimental parameters	Specific settings
Optimization function	Adam
Discarding rate	0.6
Learning rate	0.001
Embedded vector dimension	256

Before the start of the experiment, the above two datasets were preprocessed. First, products with less than 5 appearances and users with less than 10 interactions are screened out. Then, the two datasets are divided into a training set and a test set according to time. 85% of the interactions are the training set, and the rest are the test set, i.e., for the JD dataset, 64 days of data are used for training and 11 days of data for testing. For the Tianchi dataset, 26 days of data are used for training, and the remaining 4 days of data are used for testing.

At the same time, in order to control the uniqueness of independent variables, when conducting comparative experiments, consider that the most commonly used collaborative filtering in recommendation cannot recommend products that have not appeared before. Therefore, product interactions and users that have not appeared in the training set are screened out from the test set. The two datasets after being preprocessed are shown in Table 2.

4.2. Evaluation Index. When the recommendation system makes recommendations, a limited number of products will be recommended each time. If the recommendation is effective, then the products that meet the user's needs should be included in the recommendation list. Two commonly used top-k evaluation indicators are selected to evaluate the quality and effect of the recommendation list generated by the recommendation model.

(1) Hit Ratio (HR): HR is calculated as follows:

$$\mathrm{HR}@K = \frac{W@K}{\Omega}. \tag{15}$$

The meaning of the denominator is all test sets. The meaning of the numerator is the sum of the number of test sets in the top-k list recommended to each user. HR and Recall have the same functions, and both can evaluate the recall rate of the recommendation system in the recall phase. It can be seen from the definition that the larger the HR, the better the recommendation effect.

Mean Reciprocal Rank (MRR): MRR is calculated as follows:

$$\mathrm{MRR} = \frac{1}{N} \sum_{i=1}^{|N|} \frac{1}{\mathrm{rank}_i}, \tag{16}$$

where rank_i is the position of the first item in the ground truth result of the recommendation list for the ith user. Take MRR@20 as an example. If the item actually clicked by the user appears in the recommendation list and is ranked n th,

Table 2: Preprocessed dataset parameters.

Dataset	JD	Tianchi
Total users	103864	18205
Total commodity	24759	674263
Total interaction	38012329	8665937
Average number of user interactions	451.30	360.94
Training set interaction	32310478	7366046
Test set interaction	5701851	1299891

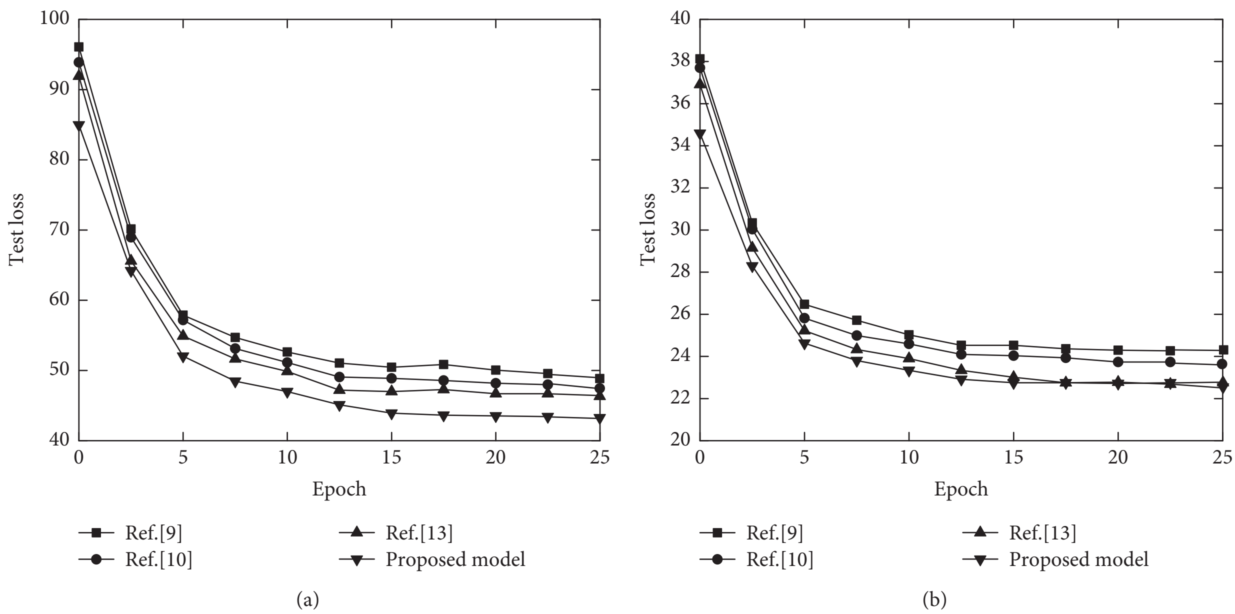

Figure 2: Test loss comparison results of different models. (a) JD dataset. (b) Tianchi dataset.

then $n \geq 1$ and if $n \leq 20$, then the MRR is equal to $1/n$. If this item does not appear in the first 20, the MRR is equal to 0. Therefore, the larger the MRR, the better. The MRR is used to evaluate the order of the items in the recommended list. Therefore, if evaluating the quality of the recommendation sequence of the recommendation model, MRR is a very important indicator.

4.3. Comparison of Loss Functions. The test loss results of the proposed model and reference [9, 10, 13] on the twodata sets are shown in Figure 2. The ordinate represents the superimposed loss of all data in each epoch.

It can be seen from Figure 2 that all four models have reached convergence. However, the test loss of the proposed model on the JD and Tianchi datasets is the smallest, which is lower than 45 and 23, respectively. The proposed model combines the Bi-GRU and the dynamic collaborative filtering algorithm to achieve personalized product recommendation. The hidden attention vector is learned in a targeted manner, which can further improve the accuracy of the recommendation. Reference [9] emphasizes rules-based semantic reasoning and reference [10] recommends products based on user interests. All lack dynamic information filtering, and hence the test loss is relatively large. Reference [13] uses machine learning to generate a product recommendation list, which has improved recommendation performance compared to reference [9, 10]. However, there is a lack of dynamic prediction of user interests, and the accuracy of the recommendations needs to be improved.

4.4. Performance Comparison with Other Algorithms. In order to demonstrate the performance of the proposed model, compare it with reference [9, 10, 13]. The comparison of the evaluation results of selecting HR@15 and MRR@15 on the two datasets is shown in Figure 3. In order to see the change trend of the index more clearly and intuitively, the ordinate is enlarged by 100 times.

On the whole, it can be seen from Figure 3 that the four models have great trends in the first 5 iterations. After 5 iterations, the indicator changes stabilized. In the end, the proposed model's indicators surpass other models, and the effect is the best. Taking HR@15 as an example, the values of the proposed models on the two datasets exceed 68 and 48, respectively. Taking MRR@15 as an example, the values of the proposed models on the two datasets exceed 25 and 15, respectively. Reference [9] proposed a rule-based semantic inference personalized recommendation model, which can quickly generate personalized recommendations and practical solutions. However, the rules are pre-set and lack dynamic updates. Therefore, the application effect of the

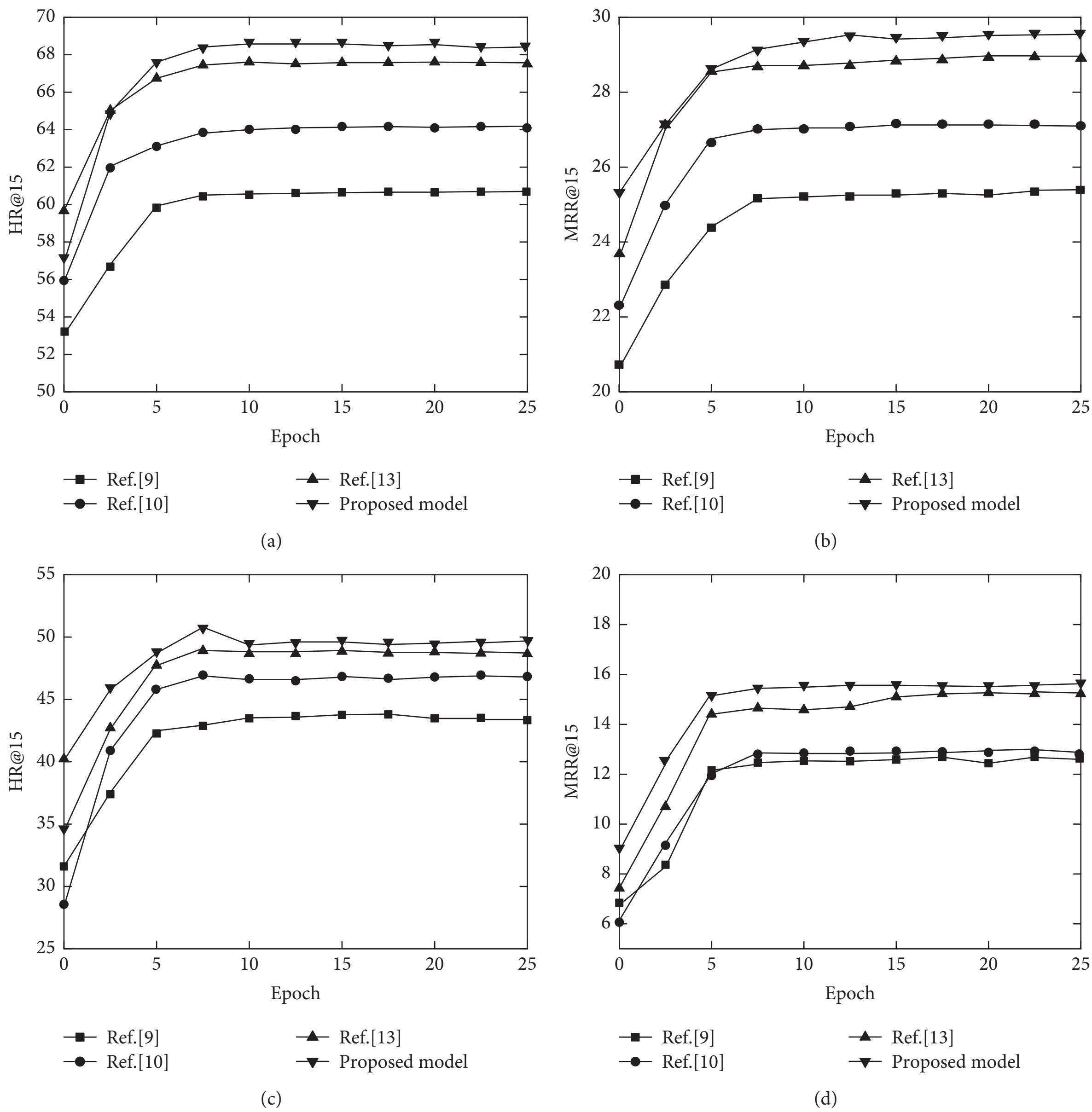

Figure 3: Comparison results of the evaluation indexes of different models. (a) HR@15 comparison results of the JD dataset. (b) MRR@15 comparison results of the JD dataset. (c) HR@15 comparison results of the Tianchi dataset. (d) MRR@15 comparison results of the Tianchi dataset.

automatic question-answering robot is not good. Taking MRR@15 as an example, which is lower than 25 and 13, respectively, reference [10] uses user interests to design personalized product recommendation models. Sales data in the transaction database is used to mine various interesting connections between the products purchased by customers. However, it lacks a powerful learning algorithm as a support, and hence the overall performance is not much different from reference [9]. Reference [13] uses machine learning to build a personalized recommendation system to generate a personalized recommendation plan. Although there are good learning algorithms for data analysis, there are still some problems in the actual application process because of the lack of dynamic user interest learning. Therefore, the maximum HR@15 should not exceed 67. The proposed model uses the Bi-GRU network to learn text features, and uses the dynamic collaborative filtering algorithm to generate product recommendations. The fused model can better adapt to the actual needs of automatic question-answering robots.

Separately, on the JD dataset, the proposed model has obvious advantages, while on the Tianchi dataset, the advantage is slightly smaller. It may be because different models of the dataset have different expressiveness because the size of the dataset and the data distribution of each latitude in the data are different, resulting in different training effects. The proposed model should also be widely used in different datasets to enhance the robustness of the model. On the other hand, the average number of interactions in the JD dataset is slightly larger than the Tianchi dataset. A larger number of interactions can capture more information. However, this can also be regarded as a research direction in the future. It is more universal and advantageous to choose and explain the number of interactions through continuous experimentation. In addition, the existing research experience shows that using data that are closer to the test set time to train the model will get better results. The experimental results also show that the effect on the JD dataset is better and more obvious. It also proves the correctness of the previous experience again.

5. Conclusion

With the rapid growth of users and product content, automatic question-answering robots using traditional recommendation algorithms have obvious shortcomings in the face of sparse historical interactive data or new product recommendations. The use of deep neural networks in the recommendation field can effectively solve the above problems. To this end, a personalized product recommendation model for automated question-answering robots using deep learning is proposed. A personalized attention mechanism is used to personalize comments and users and we input the Bi-GRU score prediction model for learning. At the same time, the dynamic collaborative filtering algorithm is used to integrate the time characteristics of user interest changes so as to get the recommendation plan of personalized products. The experimental results based on the JD and Tianchi datasets show that:

(1) The proposed model uses the attention mechanism to obtain in-depth features of products and reviews, which can reduce the error of the recommended solutions. Its training loss is lower than 45 and 23, respectively.

(2) The proposed model combines the Bi-GRU and the dynamic collaborative filtering algorithm to achieve personalized product recommendation. Comprehensive consideration of the timeliness of user interest can improve the reliability of the recommended scheme. Take MRR@15 as an example; its values on the two datasets exceed 25 and 15, respectively. The overall performance is better than other models.

Experiments have found that due to the complex model of the deep learning network, training takes a lot of time and requires high computer configuration. Therefore, next, we will consider the optimization and upgradation of the deep neural networks and the combination with the recommendation system based on efficiency to ensure that it can be more efficient in practical applications.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

References

[1] P. Nitu, J. Coelho, and P. Madiraju, "Improvising personalized travel recommendation system with recency effects," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 139–154, 2021.

[2] D. De Venuto and F. Bellotti, "A CNN based personalized product recommendation model," *International Journal of Smart Business and Technology*, vol. 8, no. 1, pp. 21–32, 2020.

[3] J. R. Bock and A. Maewal, "Adversarial learning for product recommendation," *AI*, vol. 1, no. 3, pp. 376–388, 2020.

[4] W. Deng, "Leveraging consumer behaviors for product recommendation: an approach based on heterogeneous network," *Electronic Commerce Research*, vol. 5, no. 1, pp. 1–27, 2020.

[5] M. S. Rosenbaum, G. Contreras Ramirez, J. Campbell, and P. Klaus, "The product is me: hyper-personalized consumer goods as unconventional luxury," *Journal of Business Research*, vol. 129, no. 6, pp. 446–454, 2021.

[6] B. R. Sreenivasa and C. R. Nirmala, "Hybrid location-centric e-Commerce recommendation model using dynamic behavioral traits of customer," *Iran Journal of Computer Science*, vol. 2, no. 3, pp. 179–188, 2019.

[7] B. Sheng and G. Sun, "Matrix factorization recommendation algorithm based on multiple social relationships," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6610645, 8 pages, 2021.

[8] W. Zhou and W. Han, "Personalized recommendation via user preference matching," *Information Processing & Management*, vol. 56, no. 3, pp. 955–968, 2019.

[9] X. Chen, Y. Xue, and Y. Shiue, "Rule based semantic reasoning for personalized recommendation in indoor O2O e-commerce," *International Core Journal of Engineering*, vol. 6, no. 1, pp. 309–318, 2020.

[10] J. Zhang, "Personalised product recommendation model based on user interest," *Computer Systems Science and Engineering*, vol. 34, no. 4, pp. 231–236, 2019.

[11] X. Bu, J. Zhu, and X. Qian, "Personalized product search based on user transaction history and hypergraph learning," *Multimedia Tools and Applications*, vol. 79, no. 11, pp. 1–19, 2020.

[12] B. Wu, Y. Ye, and Y. Chen, "Visual appearance or functional complementarity: which aspect affects your decision making?" *Information Sciences*, vol. 476, no. 8, pp. 19–37, 2019.

[13] G. Suganya, M. Premlatha, P. Dubey, A. R. Drolia, and S. Srihari, "Subjective areas of improvement: a personalized recommendation," *Procedia Computer Science*, vol. 172, no. 5, pp. 235–239, 2020.

[14] W. Budiharto, V. Andreas, and A. Gunawan, "Deep learning-based question answering system for intelligent humanoid robot," *Journal of Big Data*, vol. 7, no. 1, pp. 77–92, 2020.

[15] Z. Hu, J. Wang, Y. Yan, P. Zhao, J. Chen, and J. Huang, "Neural graph personalized ranking for Top-N recommendation," *Knowledge-Based Systems*, vol. 213, no. 8, pp. 106426–106437, 2020.

[16] W. Dai, Y. Wang, J. Ma, and Q. Jin, "BTR: a feature-based Bayesian task recommendation scheme for crowdsourcing system," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 3, pp. 780–789, 2020.

[17] B. Wu, X. He, Z. Sun, L. Chen, and Y. Ye, "ATM: an attentive translation model for next-item recommendation," *IEEE*

Transactions on Industrial Informatics, vol. 16, no. 3, pp. 1448–1459, 2020.
[18] S. Kang, C. Jeong, and K. Chung, "Tree-based real-time advertisement recommendation system in online broadcasting," *IEEE Access*, vol. 8, no. 3, pp. 192693–192702, 2020.
[19] T. Schreiner, A. Rese, and D. Baier, "Multichannel personalization: identifying consumer preferences for product recommendations in advertisements across different media channels," *Journal of Retailing and Consumer Services*, vol. 48, no. 5, pp. 87–99, 2019.
[20] M. Asenova and C. Chrysoulas, "Personalized micro-service recommendation system for online news," *Procedia Computer Science*, vol. 160, no. 3, pp. 610–615, 2019.
[21] B. Huynh, A. Ibrahim, Y. S. Chang, H. Tobias, and J. O'Donovan, "User perception of situated product recommendations in augmented reality," *International Journal of Semantic Computing*, vol. 13, no. 3, pp. 289–310, 2019.
[22] C. Yu, Z. Zhang, C. Lin, and Y. J. Wu, "Can data-driven precision marketing promote user ad clicks? Evidence from advertising in WeChat moments," *Industrial Marketing Management*, vol. 90, no. 5, pp. 481–492, 2020.
[23] W. Li and B. Xu, "Aspect-based fashion recommendation with attention mechanism," *IEEE Access*, vol. 8, no. 8, pp. 141814–141823, 2020.
[24] R. S. Sreepada and B. K. Patra, "Mitigating long tail effect in recommendations using few shot learning technique," *Expert Systems with Applications*, vol. 140, no. 2, pp. 112887.1–112887.17, 2020.
[25] S. Hu, A. Kumar, F. Al-Turjman, S. Gupta, S. Seth, and Shubham, "Reviewer credibility and sentiment analysis based user profile modelling for online product recommendation," *IEEE Access*, vol. 8, no. 1, pp. 26172–26189, 2020.