applied sciences

Article

Multipointer Coattention Recommendation with Gated Neural Fusion between ID Embedding and Reviews

Jianjie Shao [1,2], Jiwei Qin [1,2,*], Wei Zeng [1,2] and Jiong Zheng [1]

[1] School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; jianjie_shao@stu.xju.edu.cn (J.S.); zeng_w@stu.xju.edu.cn (W.Z.); zhengjiong@xju.edu.cn (J.Z.)
[2] Key Laboratory of Signal Detection and Processing, Xinjiang Uygur Autonomous Region, Xinjiang University, Urumqi 830046, China
* Correspondence: jwqin@xju.edu.cn

Abstract: Recently, the interaction information from reviews has been modeled to acquire representations between users and items and improve the sparsity problem in recommendation systems. Reviews are more responsive to information about users' preferences for the different aspects and attributes of items. However, how to better construct the representation of users (items) still needs further research. Inspired by the interaction information from reviews, auxiliary ID embedding information is used to further enrich the word-level representation in the proposed model named MPCAR. In this paper, first, a multipointer learning scheme is adopted to extract the most informative reviews from user and item reviews and represent users (items) in a word-by-word manner. Then, users and items are embedded to extract the ID embedding that can reveal the identity of users (items). Finally, the review features and ID embedding are input to the gated neural network for effective fusion to obtain richer representations of users and items. We randomly select ten subcategory datasets from the Amazon dataset to evaluate our algorithm. The experimental results show that our algorithm can achieve the best results compared to other recommendation approaches.

Keywords: recommendation system; multipointer learning scheme; ID embedding; gated neural fusion layer

check for updates

Citation: Shao, J.; Qin, J.; Zeng, W.; Zheng, J. Multipointer Coattention Recommendation with Gated Neural Fusion between ID Embedding and Reviews. *Appl. Sci.* **2022**, *12*, 594. https://doi.org/10.3390/app12020594

Academic Editor: Andrea Prati

Received: 12 November 2021
Accepted: 29 December 2021
Published: 8 January 2022

1. Introduction

With the rapid development of the Internet, the problem of information overload has become increasingly serious [1–3]. As an information filtering system, the recommendation system utilizes rating prediction methods to predict users' rating of items and generates a ranking list of items according to each user's preferences for personalized recommendations. Matrix factorization (MF) [4] is widely adopted for rating prediction in the recommendation field. MF represents user preferences and item attributes as vectors of potential factors in the joint potential space by factorizing the user–item interaction matrix. However, in the absence of user historical behavior, MF has a data sparsity problem and cannot make effective recommendations. To solve the data sparsity problem, researchers have begun to use review information to improve the rating prediction quality [5–7].

Recently, some efforts have been made in reviews using deep learning models. For example, Zheng et al. [5] adopted a CNN (convolutional neural network) to extract potential features from user reviews and item reviews to simulate users and items. Chen et al. [6] used an attention mechanism to evaluate the importance of each review and selected the highly useful reviews. By adopting a coattention network at the review level and word level, the MPCN model proposed by Tay et al. [7] can dynamically select important reviews (words) for the target user according to the target item and dynamically select important reviews (words) for the target item according to the target user. Although these studies have achieved excellent results, there are still some problems. Zheng et al. [5] concatenate all reviews into a document as input without considering that the quality of reviews written by different users is different. NARRE, proposed by Chen et al. [6], only judges the usefulness within the user and item reviews but does not consider the interaction between users and items. Tay et al. [7] utilized only review information to represent user preferences and item attributes and neglected to utilize additional auxiliary information to enhance the representations of users (items).

Some previous works adopted the attention mechanism to obtain useful information from users (item) reviews without considering the interactions between users and items. However, if the target item is not relevant to the item described by the selected review, the information of the selected review has no referential value for rating prediction. Therefore, modeling the interactions of user reviews and item reviews can better identify useful review information. In addition, in review-based neural recommender models, the introduction of ID embedding can reveal the potential relationships between entities and enhance user (item) representations. However, some existing approaches only string review features and ID embedding together and cannot effectively merge them.

Inspired by the above, a multipointer coattention recommendation model (MPCAR) with gated fusion between ID embedding and reviews aiming at learning the comprehensive representations of users and items is proposed. The main contributions of this paper are as follows:

- The proposed MPCAR model first uses a review gating mechanism to extract important reviews from the input sequence (user reviews and item reviews). Then, it uses review-level coattention and a multipointer learning scheme to extract the most informative reviews and models these reviews at the word level to capture richer interactions;
- ID embedding that reveals the identity of users and items is introduced. In addition, a gated neural fusion layer is designed to effectively integrate ID embedding and review features to generate the final comprehensive representation of users (items). Finally, the final representations of users and items are fed into a factorization machine (FM) to predict users' ratings of the target items;
- The MPCAR model was evaluated on real datasets from Amazon in ten different domains. The experimental results of the model outperformed those of existing popular methods.

2. Related Work

The traditional recommendation algorithm of collaborative filtering (CF) [8] has been extensively researched in academia and industry. Traditional recommendation algorithms using collaborative filtering learn the latent features of users and items based on a rating matrix. The algorithm is often formulated as a rating prediction problem [9–11], with the goal of minimizing the overall error between the actual rating and the corresponding predicted rating. MF [4] has been widely used as a simple and effective collaborative filtering recommendation algorithm. In order to further improve the performance of MF in rating prediction, researchers have proposed many methods. For example, the TimeSVD++ [12] algorithm proposed by Wei J et al. further optimized the LFM (latent factor model) model by considering user implicit feedback and changes in user preferences for items over time, which led to a better improvement in recommendation accuracy. The PMF [13] model proposed by Mnih et al. solves large-scale and sparse data by adding probability distribution to MF. However, MF relies excessively on user behavioral data for items, which leads to problems such as data sparsity, cold start, and inadequate feature extraction [14–18] that greatly affect the recommendation accuracy.

With the increase of user-to-platform interaction information, various data related to users and items is used to alleviate the above problems. Currently, introducing review information into recommendation systems to improve recommendation performance is one of the popular recommendation methods [19–26]. Early review-based works used topic modeling techniques combined with reviews to generate potential factors for users and items. For example, Huang J et al. constructed an LDA model [27] to mine subtopics from Yelp review datasets to construct hidden factor features and predict ratings. However, LDA can only mine word-level topic distribution, and cannot accurately represent the distribution of compound topics. Yang et al. proposed TopicMF [28], which obtains the potential topic of each review through NMF (non-negative matrix factorization) [29] and establishes a mapping relationship with the hidden factors of users (items). Finally, user preferences and item features are reflected through the topic distribution.

In recent years, with the in-depth study of neural networks, researchers have found that embedding words from deep learning into representation models can simply and effectively represent different data in the same vector space using deep neural networks, which can construct embedding representation information of both users and items [30]. Therefore, increasingly more review-based recommendation methods apply deep learning neural networks to mine review information to improve the performance of recommendation systems. For example, the ConvMF model [31] proposed by Kim et al. uses a CNN to process the text information of items, learns the hidden features of items, and integrates the features into a rating matrix decomposed by the PMF model to improve the rating prediction accuracy. However, ConvMF only considered using the text information of items and ignored the text information of users. For the problem of ConvMF, Zheng et al. proposed DeepCoNN model [5], which combined all reviews of users or items and then used a CNN to learn representations from reviews. The D-ATTN model [32] proposed by Seo et al. introduced a word-level attention mechanism based on DeepCoNN, which not only confirmed that different words in reviews have different importance for user and item modeling, but also introduced two attention mechanisms, local attention and global attention, to find words that have richer semantic information and assign higher weights to these words. The DRMF model [33] proposed by Wu et al. added a layer of a bidirectional GRU network structure behind two CNN networks to improve the review feature extraction quality. ConvMF, DeepCoNN, D-Attn, and DRMF all combine all reviews of users or items into one long document and then perform modeling without considering the different importance of different reviews. To address the above issues, the NARRE [6] model proposed by Chen et al. uses two parallel convolutional structures in DeepCoNN to add an importance evaluation of each review based on the modeling of users and items. However, the NARRE model cannot capture word-level interactions between user reviews and item reviews. Therefore, the MPCN model proposed by Yi Tay et al. [7] uses a pointer network to learn the representations of users and target items at the word level and review level.

The recommendation methods described above all refine the review information to better learn the representations of users and items. Therefore, how to adopt advanced techniques to better extract user and item representations from reviews has become a hot topic in recommendation research. In addition, most existing approaches fuse review features and ID embeddings using simple concatenation operations and cannot effectively fuse them to obtain deep interaction information. In response to the abovementioned problems, this paper adopts coattention and a multipointer scheme to obtain user (item) representation in a word-by-word manner and introduces ID embedding to enhance the representations of users and items learned from reviews. To obtain more information about user preferences and item features, our model applies a gated neural network to efficiently fuse ID embeddings and review features to generate a final combined representation of users and items. Finally, the MPCAR model feeds back the comprehensive representation of the gated neural network output to the FM (factorization machine) [34] for rating prediction.

3. Model Architecture

In this section, we will introduce our proposed multipointer coattention recommendation with gated neural fusion between ID embedding and reviews (MPCAR), as shown in Figure 1. MPCAR is a neural network model consisting of a review feature learning module, a user (item) embedding module, a gated fusion layer, and a prediction layer (FM). The key notations used in our proposed method are shown in Table 1.

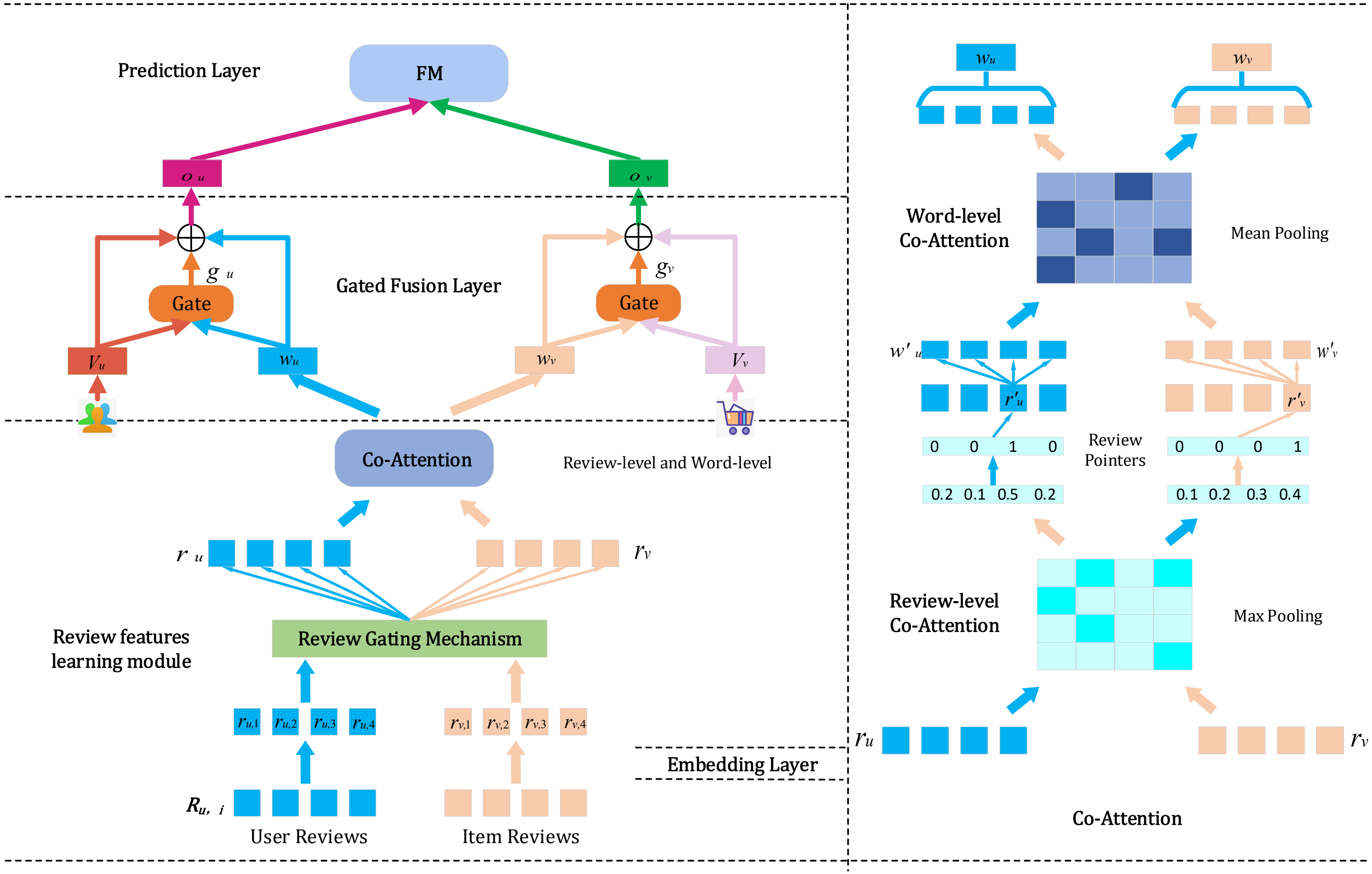


Figure 1. Neural network architecture of MPCAR.

Table 1. Notations and definitions.

Notations	Definitions
$R_{u,i}$	User reviews and item reviews
$r_{u,i}$	The i-th review of user u
$r_{v,j}$	The j-th review of item v
r_u and r_v	Informative reviews from review libraries of user u and item v
φ_{uv}	TThe affinity matrix between r_u and r_v
r'_u and r'_v	The most informative reviews from user u and item v
p_u and p_v	One-hot vector (pointers)
θ_{ij}	The affinity matrix between r'_u and r'_v
w'_u and w'_v	The word-level representations of user u and user v
w_u and w_v	Combinations of multiple word-level representations of user u and user v
V_u and V_v	The ID embedding of user u and user v
h_u and h_v	The interaction information between ID embedding and review features
o_u and o_v	The final representations of user u and item v

3.1. Review Feature Learning Module

In order to extract deeper interaction information between users and items, in the review feature learning module, we use a multipointer coattention network to process review information. In this module, we take the users' list of reviews and the items' list of reviews as two input sequences.

A. Embedding Layer. In the embedding layer, we calculate the initial review embedding according to the embedding method in the MPCN model. Given review $R_{u,i}$ consisting of a series of ℓ_w words, these words are represented as one-hot encoding vectors. Then, all words are passed into an embedding matrix. With the embedding matrix, we retrieve a d-dimensional vector for each word and finally construct user review embeddings $r_{u,1}$, $r_{u,2}$, … , r_{u,ℓ_d} and item review embeddings $r_{v,1}$, $r_{v,2}$, … , r_{v,ℓ_d} according to a series of words.

B. Review Gating Mechanism. Regardless of whether reviews written by users or reviews written for items are used, the information contained in these reviews is different, and not every review contains useful information. Here, we use a review gating mechanism to filter out useful review information.

$$r'_i = \sigma(W_g r_i) + b_g \odot tanh(W_u r_i + b_u) \tag{1}$$

where $\odot$ is the Hadamard product and σ is the sigmoid activation function. r_i is the i-th review in sequence r. W_g, $W_u \in \mathbb{R}^{d\times d}$ and b_g, $b_u \in \mathbb{R}^n$ are the parameters of this layer.

C. Review-level Coattention. We use a review gating mechanism to select informative reviews r_u and r_v from review libraries of user u and item v, respectively, as this layer of the input list. We first calculate the affinity matrix between them using Formula (2) and then obtain the row and column maximum values of the matrix using the max pooling function of Formula (3). We transform the maximum value into a one-hot vector (pointers p_u and p_v) using the Gumbel–Softmax and apply these pointers to the original review list of the users (items) to obtain the p_uth review of user u and the p_vth review of item v, respectively. The description is as follows:

$$\varphi_{uv} = F(r_u)^T W_r F(r_v) \tag{2}$$

$$r'_u = (Gumbel(^{max}_{col}(\ \varphi)))^T r_u \text{ and } r'_v = (Gumbel(^{max}_{row}(\varphi)))^T r_v \tag{3}$$

$$y_i = \frac{exp(\frac{\log(\pi_i)+g_i}{\tau})}{\sum_{j=1}^{k} exp(\frac{\log(\pi_j)+g_i}{\tau})} \tag{4}$$

$$y_i = \begin{cases} 1, i = argmax_j(y_j) \\ 0,\ otherwise \end{cases} \tag{5}$$

where $W_r \in \mathbb{R}^{d \times d}$ and $\varphi \in \mathbb{R}^{\ell r \times \ell r}$. $F(\cdot)$ is a feed-forward neural network with L layers. The reason for adopting the maximum pooling function is that max pooling will select the most influential review among all the reviews of its partners. Gumbel represents the Gumbel–Softmax [35], which returns a one-hot vector, as shown in Formulas (4) and (5). Because of the nondifferentiability of argmax, it is challenging to use discrete variables in neural networks. However, the Gumbel–Softmax replaces the argmax function with a differentiable softmax function that can support the use of discrete vectors in end-to-end neural networks.

D. Word-level Coattention. At the review level coattention layer, each review is compressed into a single embedding resulting in word information smoothing. To prevent word information smoothing, we extract the most informative reviews r'_u and r'_v using review pointers and then use word-level coattention to model these reviews to obtain deep-level word-level interaction information. Formula (6) computes an affinity matrix between r'_u and r'_v:

$$\theta_{ij} = F(r'_u)^T W_w F(r'_v) \tag{6}$$

where $W_w \in \mathbb{R}^{d \times d}$, $\theta \in \mathbb{R}^{\ell w \times \ell w}$, and $F(\cdot)$ is the standard L-layer feed-forward neural network.

We perform an average pooling operation on the affinity matrix and then use the softmax function to obtain the corresponding vector. The vector is weighted to reviews r'_u and r'_v to obtain the coattention representations w'_u and w'_v, respectively. The description is as follows:

$$w'_u = (S(avg_{col}(\theta)))^T r'_u \text{ and } w'_v = (S(avg_{row}(\theta)))^T r'_v \tag{7}$$

Here, $S(\cdot)$ is the standard softmax function. The reason why the mean pooling function is used here is that the max pooling function may be biased toward the same words, which is not suitable for the calculation of word-level coattention.

Since users may consider multiple reviews, we need to select and aggregate multiple pointers. We ran review-level coattention n_p times, and each time a unique pointer pointing to the relevant review was generated. We then using the word-level coattention mechanism to model each pair of reviews word-by-word. The final output is the combination vector:

$$w_u = \left\{w'_{u,1}; \cdots w'_{u,n_p}\right\} \text{ and } w_v = \left\{w'_{v,1}; \cdots w'_{v,n_p}\right\} \tag{8}$$

where the number of pointers n_p is a user-defined hyperparameter.

3.2. Gated Fusion Layer

Although the mutual learning of different reviews at the review level and the word level can provide abundant interactive information to express user preferences and item features, there are also some user (item) reviews with few reviews or no review. Therefore, when review information is sparse, it is insufficient to use review information as a representation of users and items. To solve the above problem, we embed users and items to extract the user ID and item ID and use the ID embedding to enhance the user and item representation.

Given ID embedding V_u of user u, it is known that w_u comes from the review information of user u; therefore, we integrate w_u and V_u to enhance the comprehensive performance of user u. At present, some existing approaches use simple addition or concatenation operations to fuse nonhomologous features, which may not be able to effectively integrate ID embedding and review features. To extract richer information from ID embedding and review features, a gated neural fusion layer is designed to integrate w_u and V_u. The calculation formula of the gating layer is as follows:

$$h_u = 1 - \sigma\left(\left(W_g w_u + b_g\right) \odot V_u\right) \tag{9}$$

where σ is the sigmoid activation function, $\odot$ denotes element multiplication, W_g denotes the parameter matrix, and b_g denotes the bias.

Since review features contain rich semantic information, ID embedding contains the intrinsic features of users, and h_u contains the interaction information between the two, we perform a concatenation operation on w_u, V_u, and h_u to obtain the final representation of user u. The description is as follows:

$$o_u = h_u \oplus w_u \oplus V_u \tag{10}$$

where $\oplus$ is the join operator. For item v, we can use the same method to obtain the final representation o_v of item v.

3.3. Prediction Layer

We use a factorization machine (FM) to predict the score. First, we stitch the user feature vector o_u and the item feature vector o_v to obtain the vector F:

$$F = o_u \oplus o_v = [f_1, f_2, \ldots, f_k] \tag{11}$$

where f_i denotes the value of the i-th dimension in vector F. Then, we pass vector F into the factorization machine (FM) to obtain the final prediction score $\hat{r}_{u,v}$:

$$\hat{r}_{u,v} = w_0 + \sum_{i=1}^{|\hat{F}|} w_i f_i + \sum_{i=1}^{|\hat{F}|} \sum_{j=i+1}^{|\hat{F}|} \langle v_i, v_j \rangle f_i f_j \tag{12}$$

The prediction score $\hat{r}_{u,v}$ can fully consider the influence of the second-order combination of different dimensions in vector F. Its final output is a scalar that can represent the intensity of the user–item interaction, where w_0 is the global bias term; w_i is the weight of the primary term; $\langle v_i, v_j \rangle$ denotes the vector inner product, which is used to capture the weight of the second-order term interaction; and v_i denotes the factorization machine hidden vector corresponding to the i-th dimension in vector F.

Since rating prediction is the research task of this paper, it is essentially a regression problem. For the regression problem, we used the minimized standard mean square error loss function (MSE) [36,37] to train the network in an end-to-end manner. The loss function formula is as follows:

$$loss = \frac{1}{2\Gamma} \sum_{u,v \in \Gamma} \left(\hat{r}_{u,v} - r_{u,v}\right)^2 \tag{13}$$

where Γ represents the number of samples in the training sets, and $r_{u,v}$ represents the actual rating of the item v by the user u.

4. Experimental Evaluation

In the experimental section, we focus on the following three research questions to develop our experiment:

RQ1. Is our proposed MPCAR model effective compared to the current popular review-based recommendation algorithm?

RQ2. How do the different parameters in our MPCAR model affect the model?

RQ3. Can the gated fusion layer effectively use nonhomologous hidden factor information (ID embedding data and review data)?

4.1. Datasets and Evaluation Metric

In this section, we conduct extensive experiments on 10 subcategory datasets from Amazon dataset to evaluate the performance of our proposed approach.

- Regarding the dataset, we choose the Amazon 5-core version [38,39] of the Amazon review public dataset, which contains a total of 24 subcategory datasets. In the experiment, we used 10 subcategory datasets to verify our MPCAR model. These 10 subdatasets all contain real user reviews from Amazon between May 1996 and July 2014. All datasets contain users, items, and user reviews and ratings of items. Each user in the dataset has posted at least five or more reviews on the platform. The details of the dataset are shown in Table 2. We randomly divided the interactive data into training set (80%), validation set (10%), and test set (10%).
- In this paper, we use mean squared error (MSE) and mean absolute error (MAE) as the evaluation metrics for model performance. These two metrics are derived by calculating the difference between the true rating and the predicted rating to measure the accuracy of rating prediction. Smaller values of MSE and MAE indicate that the predicted value is closer to the true value and the accuracy of the model prediction result is higher. Given the predicted rating $\hat{r}_{u,v}$ and the true rating $r_{u,v}$ of user u for item v, MSE and MAE are defined by the following formulas:

$$MSE = \frac{1}{N} \sum\nolimits_{u,v} \left(\hat{r}_{u,v} - r_{u,v}\right)^2 \tag{14}$$

$$MAE = \frac{1}{N} \sum\nolimits_{u,v} \left|\hat{r}_{u,v} - r_{u,v}\right| \tag{15}$$

where N represents the number of samples in the testing sets.

Table 2. The statistics of the ten Amazon datasets.

Datasets	Number of Users	Number of Items	Number of Reviews
Instant Video	5130	1685	37,126
Instruments	1429	900	10,261
Beauty	22,363	12,101	198,475
Cellphone	27,879	10,429	194,340
Gourmet Food	14,681	8713	151,232
Health	38,609	18,534	346,307
Office Products	4905	2420	53,237
Baby	19,445	7050	160,732
Digital Music	5541	3568	64,705
Pet Supplies	19,856	8510	157,683

4.2. Compared Models

To explore the performance of our model, MPCAR, we compared MPCAR with a classic algorithm and four recently popular deep learning recommendation algorithms. As shown in Table 3, the classic algorithm MF does not use ID embedding and review information, it only uses rating data as input. DeepCoNN and D-ATT adopt a document-based approach to concatenate multiple reviews of users (item) to obtain a long document as input information, and then extract global interests (features) of users (item). NARRE, MPCN, and the proposed MPCAR model each review individually and then aggregate features of reviews into user (item) features. Review-based approach captures information about a user's preference for a particular item. In addition, among these five deep learning models, NARRE and MPCAR use ID embedding information.

- Matrix factorization (MF) [4] is a commonly used benchmark. It uses the inner product to represent user and item scores.
- The deep collaborative neural network (DeepCoNN) [5] combines a user review set and an item review set to model users and items through a CNN. It trains the convolutional representation of the user and the item and passes the cascaded embedding into the FM model.

- Dual attention CNN model (D-ATT) [32], which uses reviews to make recommendations is the latest model based on a CNN. The model is characterized by using two forms of attention (local and global). End-user (item) representations are learned by combining representations learned from local and global attention. The representation of users and items is predicted by scoring using the dot product.
- NARRE [6] uses two parallel neural networks, both of which include a convolutional layer and an attention layer, to capture the usefulness of reviews, to model users, and items.
- MPCN [7] uses a pointer network to learn the characteristics of users and target items from words and reviews and transfers the final representation of users and items to the FM model.

Table 3. Comparison methods.

Methods	ID Embedding	Document	Review	Deep Learning
MF	×	×	×	×
DeepCoNN	×	√	×	√
D-ATT	×	√	×	√
NARRE	√	×	√	√
MPCN	×	×	√	√
MPCAR	√	×	√	√

4.3. Experimental Setting

We implemented our proposed MPCAR model using the PyTorch framework. All models are trained using Adam [40]. We set the training period for all models to 20 epochs. In the interaction-only model, the dimensionality of the user hidden vector and the item hidden vector is set to 50 uniformly. For the CNN-based models (DeepCONN, D-ATT, and NARRE), we set the number of convolutional kernels to 50 and the size of each convolutional kernel to 3. The word vectors are pretrained with the GloVe model. We added dropout layers after the CNN and fully connected layers for each model and set a dropout rate of 0.2. We adopted a fixed L2 regularization of 10^{-6} to regularize these models. For our proposed model, we tested it on four batch sizes (32, 64, 128, 256) and four learning rates (0.0001, 0.0005, 0.001, 0.005) to find the best parameters. The number of hidden factors is set to 10. The number of pointers p is adjusted among $\{1, 3, 5, 8, 10\}$. On most datasets, 2–3 pointers allow the model to achieve the best performance. In addition, we will use the Appendix A to describe Dropout, L2 regularization, the number of hidden factors and the number of pointers in detail.

4.4. Performance Evaluation (RQ1)

The prediction results of the MPCAR model proposed in this paper and other comparison methods for 10 Amazon subdatasets are shown in Table 4.

Table 4 shows that our proposed MPCAR is the highest performing model on the 10 benchmark datasets. This establishes the validity of our proposed model. The following conclusions can be obtained by analyzing the experimental results.

First, models that consider review information (D-ATT, DeepCoNN, NARRE, MPCN, and MPCAR) perform better than traditional collaborative filtering models (MF) that consider only rating data. Review information can solve the effect of data sparsity caused by using only rating data and improve the expression quality of hidden factors. Thus, the method using review text information (DeepCoNN D-ATT, NARRE, MPCN, and MPCAR) has better recommendation quality.

Table 4. Performance comparison (MSE and MAE) on 10 amazon subdatasets. Δ_{MN} is the relative improvement of MPCAR over MPCN (%).

		Baseline Approaches					Our Approach	Improvement (%)
Metrics	**Datasets**	**MF [4]**	**D-ATT [32]**	**DeepCoNN [5]**	**NARRE [6]**	**MPCN [7]**	**MPCAR**	Δ_{MN}
MSE	Instant Video	2.769	1.004	1.285	1.096	0.997	0.906	10
	Instruments	6.720	0.964	1.483	0.951	0.923	0.858	7.6
	Beauty	1.950	1.409	1.453	1.396	1.387	1.270	9.2
	Cellphone	1.972	1.452	1.524	1.429	1.413	1.225	15.3
	Gourmet Food	1.537	1.143	1.199	1.106	1.125	1.054	6.7
	Health	1.882	1.269	1.299	1.246	1.238	1.077	14.9
	Office Products	1.143	0.805	0.909	0.817	0.779	0.682	14.2
	Baby	1.755	1.325	1.440	1.318	1.304	1.213	7.5
	Digital Music	1.956	1.000	1.202	0.965	0.970	0.857	13.1
	Pet Supplies	1.736	1.337	1.447	1.316	1.328	1.258	5.6
MAE	Instant Video	1.467	0.770	0.839	0.768	0.781	0.715	9.2
	Instruments	2.38	0.689	0.751	0.718	0.697	0.670	4
	Beauty	1.381	0.837	0.922	0.828	0.894	0.813	10
	Cellphone	1.494	0.871	0.893	0.874	0.867	0.797	8.7
	Gourmet Food	1.206	0.731	0.718	0.731	0.704	0.693	1.6
	Health	1.27	0.725	0.739	0.727	0.712	0.703	1.3
	Office Products	0.996	0.754	0.707	0.720	0.670	0.615	8.9
	Baby	1.32	0.845	0.873	0.851	0.858	0.803	6.8
	Digital Music	1.204	0.697	0.722	0.686	0.729	0.660	10.4
	Pet Supplies	1.375	0.823	0.850	0.826	0.822	0.809	1.6

Second, among the recommendation models that also consider review information, review-based recommendation models (NARRE, MPCN, and MPCAR) have better recommendation performance than document-based recommendation models (DeepCoNN and D-ATT). This is because the former models use an approach that considers the usefulness of individual reviews. These methods capture information about users' preferences for a particular product by modeling each review. Therefore, considering the usefulness of individual reviews can further improve the performance of the model and achieve better recommendation results.

Third, in the review-based recommendation model, our MPCAR model has better results than the MPCN and NARRE models. This fully proves that it is meaningful to consider the deep word-level interaction between users and items and the introduction of additional auxiliary information (ID embedding) to enhance the representation of users and items.

4.5. Parameter Sensitivity Analysis (Q2)

Performance of rating prediction varies with model parameters. In this section, we adopt the control variables approach to explore the effect of three hyperparameters (number of hidden factors, dropout rate, and number of pointers) on rating prediction. Based on the previous work, we determine the range of values for the different parameters.

In the process of exploring the effect of the number of hidden factors on the performance of each model, we found that the MSE values of MPCAR were much smaller than those of the comparison models. This also leads to the fact that we cannot clearly observe the variation of each model performance with the number of hidden factors in the plotted graphs. Therefore, we only show the effect of different numbers of hidden factors on the performance of the proposed MPCAR model, as shown in Figure 2. For the deep learning-based models (D-ATT, DeepCoNN, NARRE, MPCN, and MPCAR), we explored the effect of different dropout rates on the performance of each model, as shown in Figure 3.

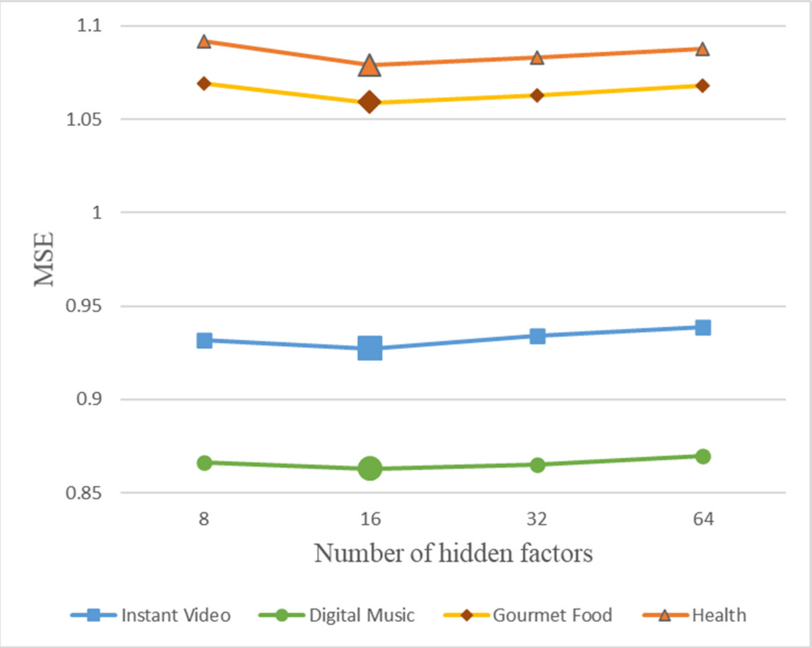
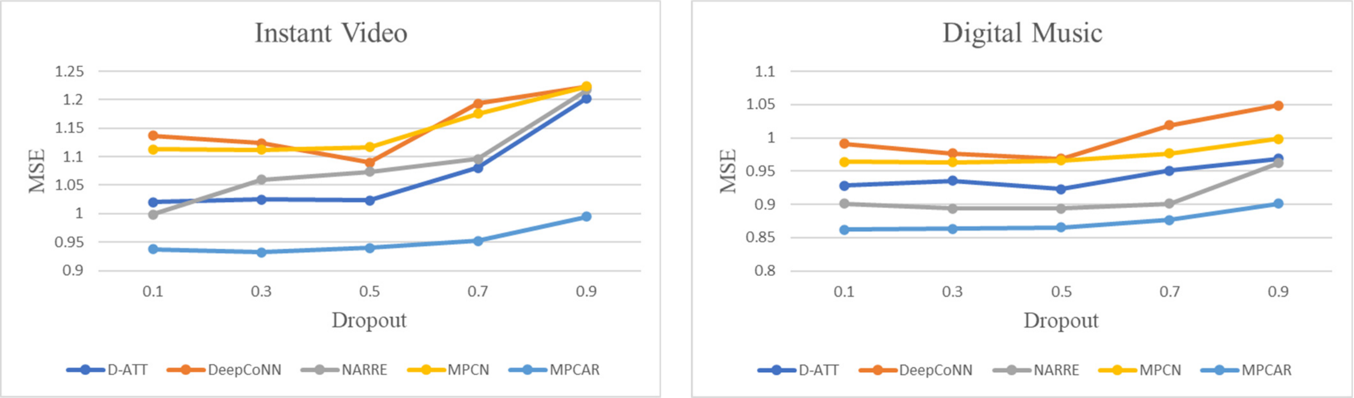


Figure 2. Effect of the number of hidden factors on model performance.



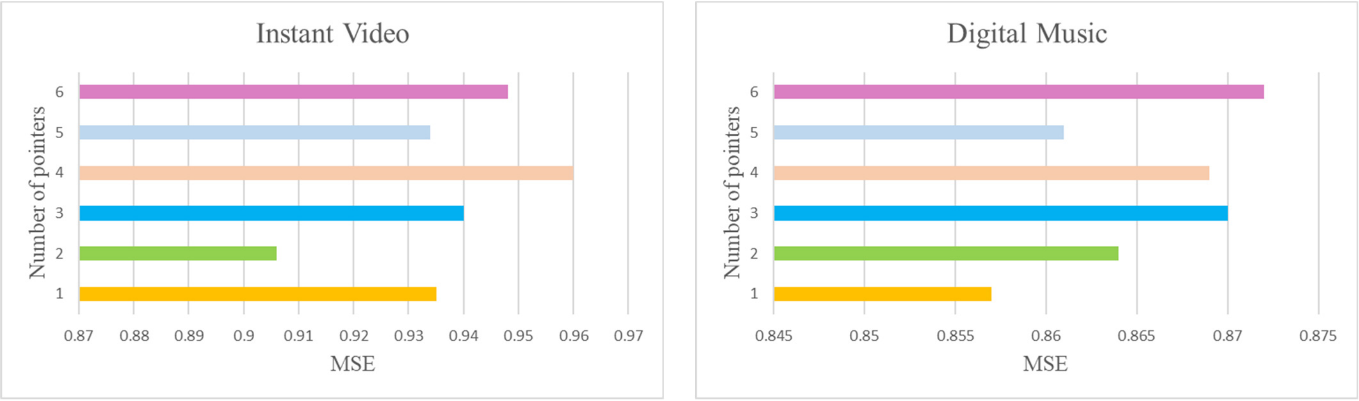
(**a**) (**b**)

Figure 3. (**a**) Effect of the dropout rate on model performance (Instant Video); (**b**) Effect of the dropout rate on model performance (Digital Music).

The length of hidden vectors of users and items is the so-called number of hidden factors in the recommender system. The optimal result obtained by the model depends on whether the appropriate number of hidden factors is selected. By observing Figure 2, we can find that, as the number of hidden factors increases, MSE first decreases and then increases, reaching the lowest when the number of hidden factors is 16. In addition, with the increase in the number of hidden factors, the proposed MPCAR model does not appear to be over-fitting (the slow growth rate of the MSE value indicates that the model is not overfitting). This fully demonstrates the effectiveness of using deep learning to build recommendation models. Deep learning enables the models to obtain better generalization performance because the models do not rely too much on some local features.

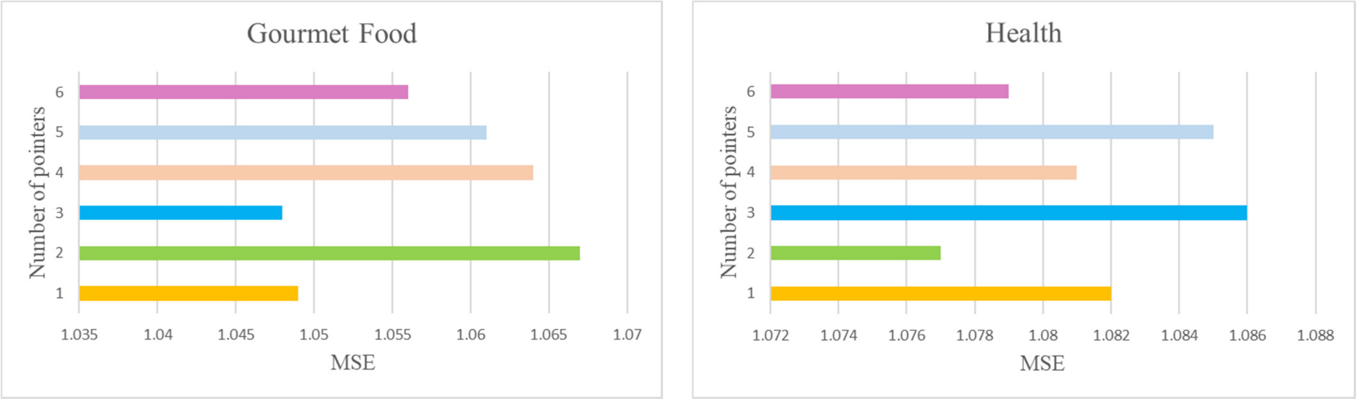
To prevent the model from overfitting, we use the dropout method to address the phenomenon of overfitting. During forward propagation, the dropout method randomly causes a certain neuron to stop working with a certain probability p. The dropout rate is one of the model hyperparameters, and different settings will also have different effects on model performance. Figure 3 shows the effect of different dropout rates on the performance of different models. After comparing the performance of the five models on the two datasets, we find that the results of these models on digital music are more stable than those on instant video. After analysis, we found that the sample size of the Instant Video dataset is very small, containing only 37,126 reviews. Therefore, these models are unable to learn stable parameters in the Instant Video dataset, which leads to overfitting more easily.

Considering that multiple review pairs may provide richer information, we adopt a multipointer approach to aggregate these selected review pairs. To verify the effectiveness of the multipointer scheme, we select four subdatasets (Instant Video, Digital Music, Gourmet Food, and Health) to observe the change of the MSE value with different pointer numbers. Figure 4 shows the effect of the different numbers of pointers on the model performance on smaller datasets. The optimal number of pointers for the MPCAR model on the two datasets (Instant Video and Digital Music) are two and one, respectively. Figure 5 shows the effect of different numbers of pointers on model performance on larger datasets. The optimal number of pointers of the MPCAR model on the two datasets (Gourmet Food and Health) are three and two, respectively. Through analysis, we can infer that the optimal number of pointers for the model does not vary with different datasets and is not affected by the size of the dataset. Besides, the number of optimal pointers is always in the range of 1–3 for different datasets.



(**a**) (**b**)

Figure 4. (**a**) Effect of number of pointers on the performance of MPCAR model (Instant Video); (**b**) Effect of number of pointers on the performance of MPCAR model (Digital Music).



(**a**) (**b**)

Figure 5. (**a**) Effect of number of pointers on the performance of MPCAR model (Gourmet Food); (**b**) Effect of number of pointers on the performance of MPCAR model (Health).

4.6. Gated Neural Network Ablation Analysis (Q3)

To clearly explore whether the gated neural network in the MPCAR model has an impact on the final performance of the model, we modified the MPCAR model to obtain a variant model MPCAR-RG. First, we delete the gated neural network, then combine review features and ID embedding in a tandem manner to obtain the representation of users and items, and finally use a factorization machine to predict the score. We randomly selected four datasets, Beauty, Cell Phone, Baby, and Pet Supplies, and conducted comparison experiments with our model on these four datasets. The comparison of the performances of the MPCAR-RG variant model and MPCAR are shown in Figure 6.

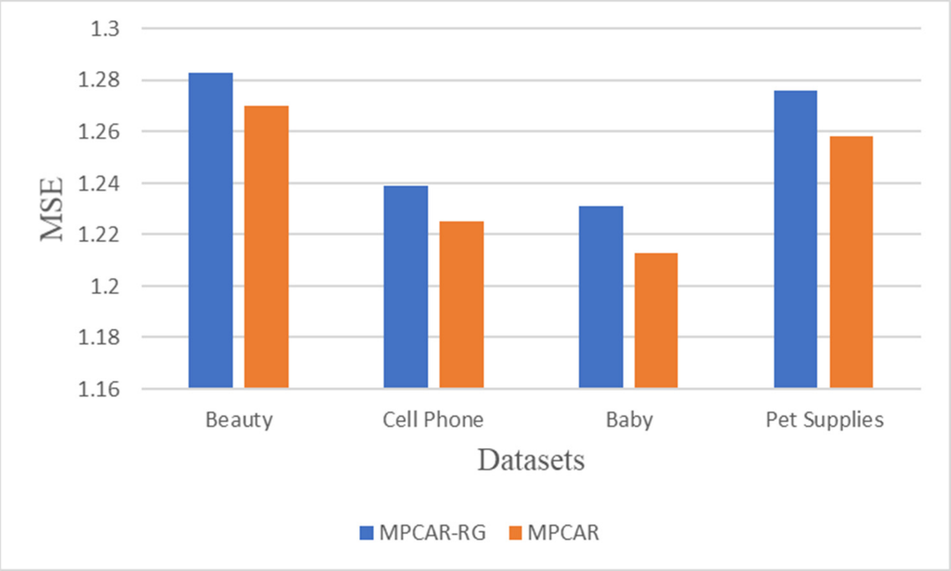


Figure 6. MPCAR-RG and MPCAR comparison results.

Figure 6 shows that the MPCAR-RG variant model will reduce the performance of the model on these four datasets. This also proves the effectiveness of our model using a gated neural network to fuse nonhomologous information. Therefore, the MPCAR model can learn user preference vectors and item feature vectors in a comprehensive and reasonable manner and finally enables the model to achieve relatively high performance.

5. Conclusions

To improve the performance of rating prediction in recommender systems, this paper proposes a multipointer coattention recommendation model (MPCAR) with gated fusion between reviews and ID embeddings. In response to the problem that some current recommendation methods do not make full use of review information and auxiliary information, and that this affects the recommendation accuracy, this paper draws the following conclusions.

In this paper, we use a multipointer learning scheme to extract important reviews from user and item reviews and then represent users (items) in a word-by-word manner to deeply mine the interactive information between users and items. The results of the comparison experiments show that citing review information in the recommender system has great research significance. In addition, considering the usefulness of individual reviews and representing users (items) in a word-by-word manner can help the model learn user and item features more accurately.

To enrich the representations of users and items, we embed users and items to obtain ID embedding and introduce a gated neural network fusion layer to effectively integrate review features and ID embedding. We input the final representations of users and items into FM for rating prediction and conduct extensive experiments to evaluate our model on

10 real Amazon subdatasets. The experimental results show that our method outperforms some existing state-of-the-art methods.

Author Contributions: Conceptualization, J.S.; methodology, J.S.; software, J.S.; validation, W.Z.; formal analysis, J.Q.; writing—original draft preparation, J.S.; writing—review and editing, J.Q.; supervision, J.Z.; funding acquisition, J.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science Fund for Outstanding Youth of Xinjiang Uygur Autonomous Region under grant no. 2021D01E14, the National Science Foundation of China under grant no. 61867006, the Major Science and Technology Project of Xinjiang Uygur Autonomous Region under grant no. 2020A03001, the Innovation Project of Sichuan Regional under grant no. 2020YFQ2018, and the Key Laboratory Open Project of Science and Technology Department of Xinjiang Uygur Autonomous Region named "Research on video information intelligent processing technology for Xinjiang regional security".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We evaluate our algorithm on ten Amazon subdatasets: Instant Video, Instruments, Beauty, Cellphone, Gourmet Food, Health, Office Products, Baby, and Digital Music. http://jmcauley.ucsd.edu/data/amazon accessed on 17 October 2021.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

To prevent the model from overfitting, we have taken the following three measures:

- Dropout can be used as a trick choice for training neural networks. In each training batch, ignoring a certain number of neurons with a certain probability can significantly reduce overfitting. In our approach, we set the dropout parameter to 0.2 to prevent the model from overfitting.
- We use L2 regularization to reduce the complexity of the MPCAR model. L2 regularization reduces the complexity of the neural network by reducing the size of the parameter value and prevents the model from overfitting to a certain extent.
- The optimal result of the model depends on whether the appropriate number of hidden factors is selected. Too many hidden factors may cause the model to overfit. According to previous work, we determine the range of the number of hidden factors as (8, 16, 32, 64), and find the optimal number of hidden factors in this range.

Since a user may consider multiple reviews when providing reviews, we adopt a multipointer learning scheme to obtain richer review information. To determine the optimal number of pointers for the MPCAR model on different datasets, we adjust the number of pointers between {1, 3, 5, 8, 10}. On most datasets, 2–3 pointers can make the model achieve the best performance.

References

1. Sharma, L.; Gera, A. A survey of recommendation system: Research challenges. *Int. J. Eng. Trends Technol.* **2013**, *4*, 1989–1992.
2. Shah, L.; Gaudani, H.; Balani, P. Survey on recommendation system. *Int. J. Comput. Appl.* **2016**, *137*, 43–49. [CrossRef]
3. Kanwal, S.; Nawaz, S.; Malik, M.K.; Nawaz, Z. A Review of Text-Based Recommendation Systems. *IEEE Access* **2021**, *9*, 31638–31661. [CrossRef]
4. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
5. Zheng, L.; Noroozi, V.; Yu, P.S. 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, Cambridge, UK, 6–10 February 2017; pp. 425–434.
6. Chen, C.; Zhang, M.; Liu, Y.; Ma, S. Neural attentional rating regression with review-level explanations. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1583–1592.
7. Tay, Y.; Luu, A.T.; Hui, S.C. Multi-pointer co-attention networks for recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2309–2318.

8. Ma, H.; King, I.; Lyu, M.R. Effective missing data prediction for collaborative filtering. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, 23–27 July 2007; pp. 39–46.
9. Zhang, Z.; Guo, X. Optimized collaborative filtering recommendation algorithm based on item rating prediction. *Appl. Res. Comput.* **2008**, *9*, 2658–2660.
10. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001; pp. 285–295.
11. Herlocker, J.L.; Konstan, J.A.; Riedl, J. Explaining collaborative filtering recommendations. In Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, Philadelphia, PA, USA, 2–6 December 2000; pp. 241–250.
12. Wei, J.; He, J.; Chen, K.; Zhou, Y.; Tang, Z. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst. Appl.* **2017**, *69*, 29–39. [CrossRef]
13. Mnih, A.; Salakhutdinov, R. Probabilistic matrix factorization. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2008; pp. 1257–1264.
14. Junliang, L.; Xiaoguang, L. Technical progress of personalized recommendation system. *Comput. Sci.* **2020**, *47*, 47–55.
15. Cao, L. Coupling learning of complex interactions. *Inf. Process. Manag.* **2015**, *51*, 167–186. [CrossRef]
16. Cao, L. Non-iid recommender systems: A review and framework of recommendation paradigm shifting. *Engineering* **2016**, *2*, 212–224. [CrossRef]
17. He, X.; Chua, T.-S. Neural factorization machines for sparse predictive analytics. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 355–364.
18. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.-S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
19. Catherine, R.; Cohen, W. Transnets: Learning to transform for recommendation. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; pp. 288–296.
20. Wu, L.; Quan, C.; Li, C.; Wang, Q.; Zheng, B.; Luo, X. A context-aware user-item representation learning for item recommendation. *ACM Trans. Inf. Syst.* **2019**, *37*, 1–29. [CrossRef]
21. Chin, J.Y.; Zhao, K.; Joty, S.; Cong, G. ANR: Aspect-based neural recommender. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 147–156.
22. Liu, D.; Li, J.; Du, B.; Chang, J.; Gao, R. Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 344–352.
23. Hyun, D.; Park, C.; Yang, M.-C.; Song, I.; Lee, J.-T.; Yu, H. Review sentiment-guided scalable deep recommender system. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 965–968.
24. Liu, H.; Wu, F.; Wang, W.; Wang, X.; Jiao, P.; Wu, C.; Xie, X. NRPA: Neural recommendation with personalized attention. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 1233–1236.
25. Wu, C.; Wu, F.; Liu, J.; Huang, Y. Hierarchical user and item representation with three-tier attention for recommendation. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Long and Short Papers. Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1, pp. 1818–1826.
26. Dong, X.; Ni, J.; Cheng, W.; Chen, Z.; Zong, B.; Song, D.; Liu, Y.; Chen, H.; De Melo, G. Asymmetrical hierarchical networks with attentive interactions for interpretable review-based recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 22 February–1 March 2020; pp. 7667–7674.
27. Huang, J.; Rogers, S.; Joo, E. Improving restaurants by extracting subtopics from yelp reviews. In Proceedings of the iConference 2014 (Social Media Expo), Berlin, Germany, 4–7 April 2014; iSchools: Grandville, MI, USA, 2014; pp. 1–5.
28. Bao, Y.; Fang, H.; Zhang, J. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014.
29. Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791. [CrossRef] [PubMed]
30. Du, Y.; Wang, L.; Peng, Z.; Guo, W. based hierarchical attention cooperative neural networks for recommendation. *Neurocomputing* **2021**, *447*, 38–47. [CrossRef]
31. Kim, D.; Park, C.; Oh, J.; Lee, S.; Yu, H. Convolutional matrix factorization for document context-aware recommendation. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 233–240.
32. Seo, S.; Huang, J.; Yang, H.; Liu, Y. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; pp. 297–305.
33. Wu, H.; Zhang, Z.; Yue, K.; Zhang, B.; He, J.; Sun, L. Dual-regularized matrix factorization with deep neural networks for recommender systems. *Knowl. Based Syst.* **2018**, *145*, 46–58. [CrossRef]

34. Rendle, S. Factorization machines. In Proceedings of the 2010 IEEE International Conference on Data Mining, Washington, DC, USA, 13–17 December 2010; pp. 995–1000.
35. Jang, E.; Gu, S.; Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv* **2016**, arXiv:1611.01144.
36. Ling, G.; Lyu, M.R.; King, I. Ratings meet reviews, a combined approach to recommend. In Proceedings of the 8th ACM Conference on Recommender Systems, Silicon Valley, CA, USA, 6–10 October 2014; pp. 105–112.
37. McAuley, J.; Leskovec, J. Hidden factors and hidden topics: Understanding rating dimensions with review text. In Proceedings of the 7th ACM conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 165–172.
38. He, R.; McAuley, J. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Proceedings of the 25th International Conference on World Wide Web, Montréal, QC, Canada, 11–15 April 2016; pp. 507–517.
39. McAuley, J.; Targett, C.; Shi, Q.; Van Den Hengel, A. Image-based recommendations on styles and substitutes. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 43–52.
40. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.