scientific reports

Check for updates

OPEN

Attentional factorization machine with review-based user–item interaction for recommendation

Zheng Li[1,2,3], Di Jin[1] & Ke Yuan[1]✉

In recommender systems, user reviews on items contain rich semantic information, which can express users' preferences and item features. However, existing review-based recommendation methods either use the static word vector model or cannot effectively extract long sequence features in reviews, resulting in the limited ability of user feature expression. Furthermore, the impact of different or useless feature interactions between users and items on recommendation performance is ignored. Therefore, we propose an attentional factorization machine with review-based user–item interaction for recommendation (AFMRUI), which first leverages RoBERTa to obtain the embedding feature of each user/item review, and combines bidirectional gated recurrent units with attention network to highlight more useful information in both user and item reviews. Then we adopt AFM to learn user–item feature interactions to distinguish the importance of different user–item feature interactions and further to obtain more accurate rating prediction, so as to promote recommendation. Finally, we conducted performance evaluation on five real-world datasets. Experimental results on five datasets demonstrated that the proposed AFMRUI outperformed the state-of-the-art review-based methods regarding two commonly used evaluation metrics.

With the rapid development of Internet industry and big data technology, recommender systems are playing an increasingly important role in social networks[1], academic education[2], e-commerce[3], and so on. Nowadays, recommender systems have become an indispensable part of daily life, such as online shopping[4], next point-of-interest recommendation[5], music recommendation[6], and video push[7]. According to users' historical behavioral data, recommender systems can predict users' ratings of items and perform personalized recommendation, so as to help users quickly discover items they are interested in and improve users' satisfaction. Therefore, in order to provide better personalized recommendation services, how to accurately predict users' ratings on items to boost recommendation becomes a challenge problem.

To solve the above issue, researchers have proposed a variety of item rating prediction methods, among which rating prediction method[8] based on collaborative filtering (CF) is one of the most widely used methods. Most CF methods are based on matrix factorization[9,10], learning latent features of users and items from matrix models for recommendation. Considering users ratings for items reflect their interaction behaviors and explicit features, Zhang et al.[11] obtained users and items features from user–item rating information based on deep matrix factorization. However, with the rapid growth of the number of users and items, there are more and more problems such as sparsity of the rating data. Unfortunately, the information extracted from rating data is limited, consequently restricting the recommendation performance.

Compared with rating data, review information contains rich semantics, which can not only reflect users' satisfaction with item quality and function, but also indirectly express users' preferences and item features[12]. Thus, review-based item rating prediction has attracted extensive attention from researchers, such as ConvMF[13], DeepCoNN[14], D-Attn[15], NARRE[16], and DAML[17], etc. These methods can alleviate the sparsity problem caused by rating data through review information, and thus obtain relatively accurate prediction ratings for recommendation. However, there are two major limitations as follows:

1. The expression ability of user/item features is insufficient. In above research, D-Attn[15], DAML[17], etc., leverage word vectors statically encoded such as word2vec or Glove, resulting in sparse feature representation, insufficient semantics and polysemy, which affect the ability of model to extract user and item features. Moreover,

[1]College of Computer and Information Engineering, Henan University, Kaifeng 475004, Henan, China. [2]Henan Engineering Laboratory of Spatial Information Processing, Henan University, Kaifeng 475004, Henan, China. [3]Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, Henan, China. ✉email: yuanke@henu.edu.cn

models such as ConvMF[13], DeepCoNN[14], and NARRE[16] use convolutional neural networks (CNN) to extract users and items features from reviews, which cannot effectively extract long sequence text features in reviews, and thus cannot accurately express user or item features, limiting the model performance.

2. The influence of feature interactions between users and items on the recommendation performance is ignored. For example, models, such as DeepCoNN[14], D-Attn[15], NARRE[16], DAML[17], etc., obtain prediction ratings by dot product or factorization machine after splicing of users and items features. Such feature interaction modelling methods ignore different effects of different feature interactions on recommendation results. Furthermore, useless feature interactions will introduce noise, thus reducing the recommendation performance.

To address the above issues, this paper proposed an attentional factorization machine with review-based user–item interaction for recommendation. Specifically, in order to better capture review-based user features and item features, we first obtain the embedding feature of each review through the pre-trained model RoBERTa, which alleviates the problem that static word vectors cannot adapt to polysemy; then we combine bidirectional gate recurrent unit (BiGRU) and attention network to highlight important information in reviews, and obtain user reviews embedding and item reviews embedding; furthermore, the obtained reviews embedding of user and item are concatenated together and input to attentional factorization machine (AFM) to perform more accurately rating prediction, so as to make recommendation. The main contributions of this paper can be summarized as follows:

- We build an enhanced framework for user/item feature representation, which leverages RoBERTa to obtain the embedding feature of each user/item review to alleviate the problem of polysemy, and uses BiGRU and attention network to measure the contribution of embedding feature of each review, so as to obtain better expression ability of user/item features;
- We use AFM to learn user–item feature interactions and to distinguish the importance of different feature interactions, which can further alleviate the effect of noise that may be introduced by useless feature interactions;
- We conduct comprehensive experiments on five real-world datasets, which demonstrate that our proposed AFMRUI model outperforms the state-of-the-art models.

The remainder of this paper is organized as follows. In "Related work", we provide an overview of related work. Section "The proposed approach" elaborates our proposed AFMRUI model. Next, we evaluate the effectiveness of our model and analyze the experimental results in "Experiments". Finally, "Conclusions" presents the conclusions and sketches directions for future work.

Related work

Embedding representation methods. In review-based recommendation tasks, word embedding representation methods are usually used to express user or item review embedding features. Models, such as ConvMF[13], DeepCoNN[14], D-Attn[15], NARRE[16], and DAML[17], etc., use Glove[18] and Word2Vec[19] belonging to static word vector models. However, the obtained user/item review embedding features cannot change with the contextual semantics, and the problem of polysemy will be produced. As a result, dynamic word vectors are used to solve the problem. For example, Google proposed Bidirectional Encoder Representation from Transformers (BERT)[20], a dynamic word vector pre-trained model, to achieve excellent results in 11 natural language processing tasks. In recent research, SIFN[21] and U-BERT[22] use BERT to obtain the review embedding representation, which have a large performance improvement in rating prediction compared with methods using static word vector models.

Based on BERT, an improved model RoBERTa[23] was introduced, which not only inherits the advantages of BERT, but also simplifies the next sentence prediction task in BERT. RoBERTa is retrained using new hyperparameters and a large new dataset, which allows the model to be more fully trained and has a significant improvement in performance. To this end, we adopt RoBERTa in our model to mitigate the problem of polysemy in user/item reviews by encoding the obtained word-level embedding representation of each review.

Review-based recommendation methods. With the increase of interactive information generated by users in various fields, various interactive information related to users and items, e.g., reviews, is introduced into the recommender system to improve the performance. Next, we will outline two review-based recommendation methods.

Review-based topic modeling recommendation methods. Topic modeling approaches were the first to apply reviews to recommender systems, mainly obtaining the latent topic distribution in reviews through latent dirichlet allocation (LDA) or non-negative matrix factorization, and demonstrated the usefulness of reviews. For example, Xu et al.[24] proposed a topic model-based CF model, which mainly obtained review-based features through an LDA-based extended model. Huang et al.[25] similarly obtained potential features of users in Yelp restaurant review dataset by LDA algorithm, which can help restaurant operators understand customer preferences. Since the topic model based on LDA cannot preserve the word order information, the context information in the reviews is ignored.

Aiming at the problems of LDA algorithm, Bao et al.[26] proposed a TopicMF model, which used the latent factors of users and items obtained by matrix factorization to correlate, so as to improve the accuracy of rating

prediction. Ganu et al.[27] learned preference features of each user from reviews information, and used a CF method based on latent factor model (LFM) for rating prediction. However, LFM model can only learn those linear and low-level features, which is not conducive to interactive learning among features from fusion layers.

The methods mentioned above use the bag-of-words-based topic model for review processing, which cannot preserve the word order information well, so that the local context information contained in reviews will be ignored, and only shallow semantic information can be extracted. However, the rich semantic information in user/item reviews cannot be accurately captured. While in our research, we use RoBERTa and BiGRU to model user reviews and item reviews, so as to effectively obtain user and item review embedding features with rich semantics.

Review-based deep learning recommendation methods. In recent years, CNN has been widely used in the task of review-based recommendation. For example, Kim et al.[13] first introduced CNN into recommender system and proposed ConvMF model. However, ConvMF model only uses item reviews and user ratings during training, ignoring user reviews information. For this problem, Zheng et al.[12] introduced a deep parallel network framework DeepCoNN, which alleviated the problems in ConvMF by using two parallel CNN networks to model user review documents and item review documents respectively. Considering that different words have different importance for modeling users and items, Seo et al.[15] introduced CNN with dual local and global attention to learn reviews embedding of each user and each item, so as to perform rating prediction. Chen et al.[16] introduced a neural attentional regression model with review-level explanations, which used a review-level attention mechanism to assign different weights to each review, making the recommendation interpretable. The above methods use CNN to encode reviews, but CNN-based methods fail to effectively extract features from reviews with different lengths.

To address the above problem, Tay et al.[28] learned feature representations of users and items by using pointers at the word-level and review-level based on review information, to obtain important information in reviews to improve the prediction results. Chen et al.[29] modeled dynamic preferences of users as well as item attributes through gated recurrent unit (GRU) and sentence-level CNN, and improved the interpretability of the proposed model.

According to the above analysis, review-based deep learning recommendation methods have superior performance compared with topic-based modeling recommendation methods. So in our model, we leverage BiGRU and incorporate attention network to measure the importance of each review, so as to improve user/item feature representations.

Feature interaction methods. For the feature interactions between users and items, some research uses traditional feature interaction methods, such as dot product[30], fully connected[31], factorization machines (FM)[32], etc. FM are supervised learning methods that augment linear regression models by incorporating feature interactions. For example, multi-pointer co-attention networks[28] shows that FM obtain better results than other interaction models for its good interaction ability. However, traditional methods model all feature interactions and fail to distinguish the importance of different feature interactions. Therefore, Zhang et al.[33] proposed a combination model of FM and deep neural network based on factorization machine neural network model, which generated higher-order feature combinations, and strengthened the learning ability of models to features.

However, for different samples, the weights of different feature interactions should also be different. In other words, for those unimportant feature interactions, it should reduce their weights. While for those high-importance feature interactions, it should increase their weights. To this end, Xiao et al.[34] improved FM by recognizing the importance of different feature interactions, and introduced an AFM, which can learn the importance of feature interactions through attention mechanism, so as to alleviate the problem of reduced feature representations performance caused by those useless feature interactions.

Inspired by reference[34], our AFMRUI model adopt AFM to learn the feature interactions of users and items, and obtain better feature representations by distinguishing the importance of different feature interactions, and alleviate the effect of noise that may be introduced by useless feature interactions.

The proposed approach

In this section, we first present the problem definition of our recommendation task and list key notations used in our work in Table 1, and then elaborate the model framework of AFMRUI.

Problem definition. Assume that dataset D contains M users and N items as well as plentiful reviews and the corresponding ratings. Each sample in dataset D is defined as userID-itemID-review-rating quadruplet (u, i, r, $y(x)$), meaning that user u makes a review r and gives the corresponding rating $y(x)$ on item i. For all samples in dataset D, we can obtain the review set of each user and the review set of each item by retrieving userID and itemID. In this work, we focus on predicting a user's rating on an item based on the obtained corresponding review sets of user and item. We define the review-based recommendation task as follows:

Definition (review-based recommendation task). Given a review set D_u of user u and a review set D_i of an item i, the task of review-based recommendation is to predict user u's rating $\hat{y}(x)$ on the item i and then makes recommendation.

AFMRUI framework. The architecture of the proposed AFMRUI model is shown in Fig. 1. The AFMRUI model is composed of two parallel networks with similar structures, namely, user review network RN_u and item review network RN_i. Review set D_u of a user u and review set D_i of an item i are given to RN_u and RN_i respectively as inputs, and the corresponding rating predicted on item i is produced as the output, so as to make

Notation	Interpretation
M, N	Total number of users, total number of items in a dataset
u, i	A specific user, an item
n, m	User's maximum number of reviews, item's maximum number of reviews
D_u, D_i	Review set of user u, review set of item i
RD_u	Reviews list of user u after preprocessing
RD_i	Reviews list of item i after preprocessing
r_{u_n}	The word-level embedding representation of user review d_{u_n} from RD_u
r_{i_m}	The word-level embedding representation of item review d_{i_m} from RD_i
$\mathbf{O}_u$	The embedding feature list of user reviews obtained by RoBERTa
$\mathbf{O}_i$	The embedding feature list of item reviews obtained by RoBERTa
$\mathbf{H}_u$	The whole hidden feature of user review extracted from sequence coding layer
$\mathbf{H}_i$	The whole hidden feature of item review extracted from sequence coding layer
α_u	Attention weights vector corresponding to the whole hidden feature $\mathbf{H}_u$
α_i	Attention weights vector corresponding to the whole hidden feature $\mathbf{H}_i$
R_u	Review embedding of user u obtained by attention layer
R_i	Review embedding of item i obtained by attention layer
x	The joint vector of user review embedding and item review embedding
$\hat{y}(x)$	The predicted user u's rating of item i

Table 1. Key notations used in this paper.

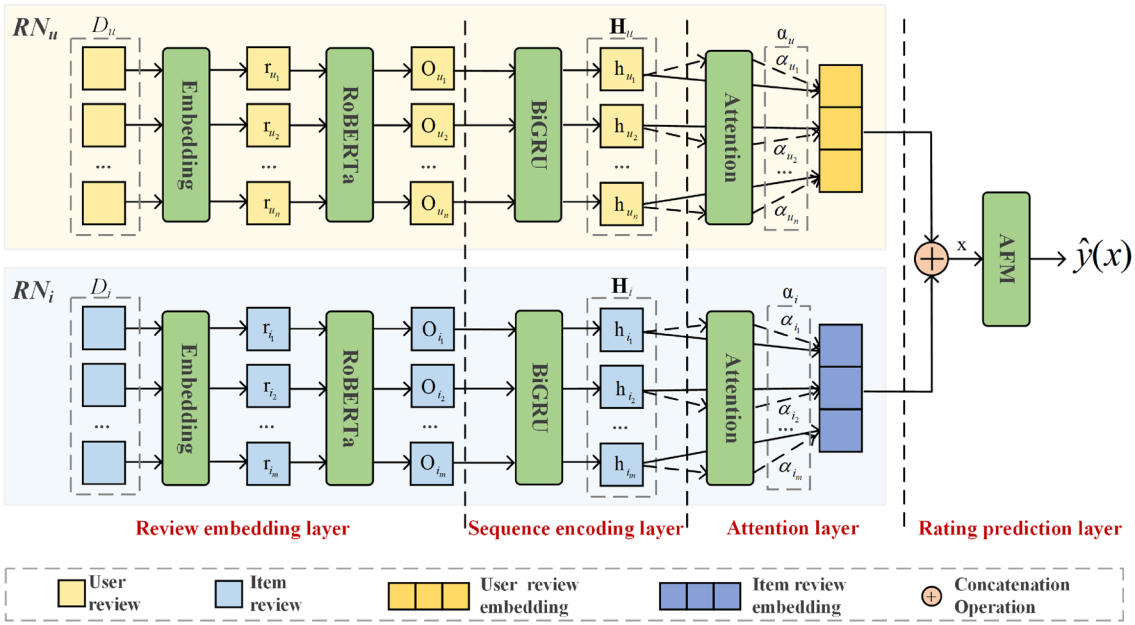

Figure 1. Illustration of AFMRUI model.

recommendation. It can be seen from Fig. 1, AFMRUI model consists of four layers. Each layer is outlined as follows:

1. Review embedding layer, which is mainly used to obtain the embedding feature of each review from the sets D_u and D_i by RoBERTa;
2. Sequence encoding layer, which mainly leverages BiGRU to encode embedding feature of each review produced by review embedding layer, and fully mines the internal dependencies among review embedding features, so as to obtain the corresponding hidden features;
3. Attention layer, which is utilized to obtain reviews embedding of a user or an item by adaptively measuring the weight of hidden feature of each review, so that the model can focus on more useful reviews and improve the feature expression ability of users and items;

4. Rating prediction layer, which first concatenates the reviews embedding of user u and item i obtained from attention layer, and further leverages AFM to learn user–item feature interactions to predict user u's rating on item i, and then makes recommendation.

Since RN_u and RN_i only differ in their inputs, so next we take RN_u network as an example to illustrate the process in detail. Note that the process described in the following subsections "Review embedding layer", "Sequence encoding layer", and "Attention layer" is also applied to RN_i network.

Review embedding layer. Review embedding layer is used to obtain embedding feature of each review from user review set D_u by RoBERTa. According to the requirements of RoBERTa, the original reviews from D_u need to be preprocessed to achieve the corresponding review embedding features.

Specifically, we first remove special characters, such as mathematical symbols, punctuation marks, in each review from D_u, and set the obtained reviews to a unified maximum length. Then, we combine each review processed into a list to get the corresponding user review list RL_u. Furthermore, we set the obtained review list of each user in the dataset to a fixed length n, where n represents users' maximum number of reviews input to RoBERTa. If the length of RL_u exceeds n, the truncation operation is performed to get the first n reviews in RL_u. Otherwise, we use zero vectors for filling operation after RoBERTa mapping to get the specified length n. Afterwards, we insert special characters <s> and </s> at the beginning and end of each review respectively after fixed length processing to obtain review list RD_u of user u, denoted as $\{d_{u_1}, d_{u_2}, \ldots, d_{u_n}\}$.

Subsequently, each review in the list RD_u needs to be expressed in the form of word-level embedding representation, which is composed of token embeddings, segment embeddings and position embeddings. Take the review "Love this album. It is such an inspiring fun album". by user A2B2J5VS139VLM on item B004L49K20 in Digital Music dataset as an example. Figure 2 shows how to obtain the word-level embedding representation of the review.

As shown in Fig. 2, the original review is preprocessed as the input of word-level embedding representation. Then we extract token embeddings, segment embeddings and position embeddings from the preprocessed review respectively, and then add them to get the word-level embedding representation of the review. For the f-th token in the preprocessed user review d_{u_i}, its word-level embedding representation is denoted as:

$$e_f = E_{token(f)} + E_{seg(f)} + E_{pos(f)} \tag{1}$$

where $E_{token(f)}$ is the token embedding corresponding to the f-th token in d_{u_i}, which is obtained by mapping the token as a 768-dimensional embedding; $E_{seg(f)}$ represents the segment embedding corresponding to the f-th token in d_{u_i}. Since each preprocessed review can be considered as a sentence, so the segment embedding of each word in d_{u_i} is the same. As shown in the "segment embeddings" in Fig. 2, the segment embedding of each token from the review in the example is E_A; $E_{pos(f)}$ is the position embedding, which represents the result of encoding the position of each word in d_{u_i}.

Based on the above processing, we can obtain r_{u_i}, the word-level embedding representation of d_{u_i} from the list RD_u, which is represented as:

$$\mathrm{r}_{u_i} = [e_0, e_1, \ldots, e_j] \tag{2}$$

By doing the same operation for each preprocessed review from RD_u, we obtain the corresponding word-level embedding representation of each review, represented as $\{\mathrm{r}_{u_1}, \mathrm{r}_{u_2}, \ldots, \mathrm{r}_{u_n}\}$, where n represents the specified maximum number of user reviews.

Considering the multi-head attention mechanism in RoBERTa can effectively capture the semantic information among tokens in a review, which can mitigate the problem of polysemy in user/item reviews. Therefore, we

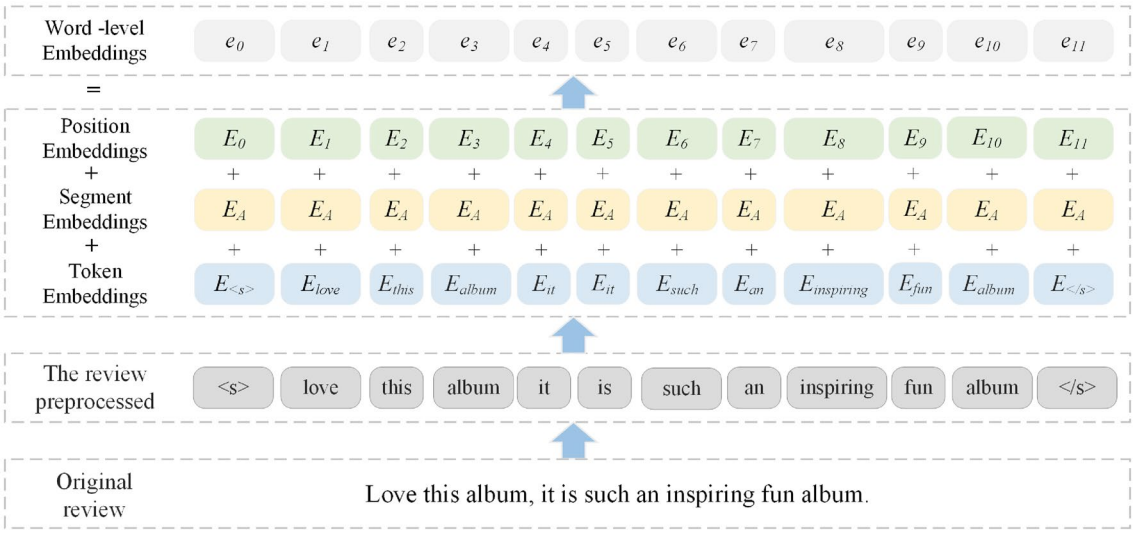

Figure 2. Illustration how to obtain the word-level embedding representation of a review.

leverage RoBERTa to semantically encode the obtained word-level embedding representation of each review. Specifically, given the word-level review embedding representation r_{u_i} as the input of RoBERTa, we can obtain the corresponding review embedding feature O_{u_i}, denoted as:

$$O_{u_i} = \text{RoBERTa}(r_{u_i}), i = 1, 2, \ldots, n \tag{3}$$

where O_{u_i} is a fixed c-dimensional semantic feature.

Then the embedding features of reviews from RD_u output by RoBERTa can be represented by a review embedding feature list $\mathbf{O}_u \in \mathbb{R}^{n \times c}$, denoted as $\{O_{u_1}, O_{u_2}, \ldots, O_{u_n}\}$.

Sequence encoding layer. Sequence encoding layer is used to obtain the corresponding hidden features of each review. In order to capture the relationships among review embedding features of user u, we use BiGRU, which has proven to be successful in practical applications[35,36], to encode embedding feature of each review from list $\mathbf{O}_u$. In this way, embedding feature of each review can be modeled from forward and backward directions, and fully mines the internal dependencies among review embedding features, so as to obtain the corresponding hidden features.

Specifically, we take the list $\{O_{u_1}, O_{u_2}, \ldots, O_{u_n}\}$ as the input of BiGRU to obtain the corresponding forward hidden feature and backward hidden feature, represented as:

$$\overrightarrow{h_{u_i}} = \overrightarrow{GRU}\left(O_{u_i}, \overrightarrow{h_{u_{i-1}}}\right) \tag{4}$$

$$\overleftarrow{h_{u_i}} = \overleftarrow{GRU}\left(O_{u_i}, \overleftarrow{h_{u_{i+1}}}\right) \tag{5}$$

where $\overrightarrow{h_{u_i}}$ represents the forward hidden feature corresponding to O_{u_i}, $\overrightarrow{GRU}$ represents forward processing from O_{u_1} to O_{u_n}, $h_{u_{i-1}}$ represents the forward hidden feature corresponding to $O_{u_{i-1}}$; correspondingly, $\overleftarrow{h_{u_i}}$ represents the backward hidden feature corresponding to O_{u_i}, $\overleftarrow{GRU}$ represents backward processing from O_{u_n} to O_{u_1}, $\overleftarrow{h_{u_{i+1}}}$ represents the backward hidden feature corresponding to $O_{u_{i+1}}$.

Then we concatenate $\overrightarrow{h_{u_i}}$ with $\overleftarrow{h_{u_i}}$ of each review to obtain the corresponding hidden feature $h_{u_i} \in \mathbb{R}^{2l}$, where l represents the hidden dimension of each GRU. h_{u_i} is denoted as:

$$h_{u_i} = [\overrightarrow{h_{u_i}}, \overleftarrow{h_{u_i}}] \tag{6}$$

Similarly, we can obtain the whole hidden feature $\mathbf{H}_u \in \mathbb{R}^{n \times 2l}$ corresponding to list $\mathbf{O}_u$ through the sequence coding layer, denoted as:

$$\mathbf{H}_u = (h_{u_1}, h_{u_2}, \ldots, h_{u_n}) \tag{7}$$

Attention layer. Considering reviews made by users on different items reflect different user preferences, we introduce attention mechanism[37,38] to adaptively measure weights to review hidden features, and aggregate those more useful informative reviews to form a user review embedding.

Specifically, the attention network takes the whole hidden feature $\mathbf{H}_u$ as input, and yields a corresponding vector of attention weights, $\alpha_u \in \mathbb{R}^{1 \times n}$, represented as:

$$\alpha_u = \text{soft}\max(w_1 \times \tanh(\mathbf{W}_1 \times \mathbf{H}_u^{\mathrm{T}})) \tag{8}$$

where $w_1 \in \mathbb{R}^{1 \times t_1}$ represents a vector of parameters, $\mathbf{W}_1 \in \mathbb{R}^{t_1 \times 2l}$ is weight matrix, t_1 represents the hidden unit number in the attention network. soft $\max(\cdot)$ is used to normalize the attention weights vector. Each dimension in α_u denotes the degree of user preference reflected by each review.

Then, we compute the weighted sums by multiplying attention weights vector α_u and whole hidden feature $\mathbf{H}_u$, to obtain user review vector $d_u \in \mathbb{R}^{1 \times 2l}$, denoted as:

$$d_u = \alpha_u \mathbf{H}_u \tag{9}$$

Next, d_u is used as the input of the fully connected layer to obtain user u's review embedding $R_u \in \mathbb{R}^k$, where k represents the latent dimension. R_u is represented as:

$$R_u = \mathbf{W}_2 \times d_u + b_1 \tag{10}$$

where $\mathbf{W}_2 \in \mathbb{R}^{k \times 2l}$ is the weight matrix of the fully connected layer, and $b_1 \in \mathbb{R}^k$ is a bias term.

Similarly, for RN_i network, we can get item i's review embedding R_i from the corresponding item review set D_i.

Rating prediction layer. In rating prediction layer, our goal is to predict user u's rating $\hat{y}(x)$ of item i based on user review embedding R_u and item review embedding R_i. In fact, the predicted user's rating of an item is actually a kind of user–item feature interactions. However, most existing approaches, such as dot product, cannot effectively learn user–item feature interactions and fail to distinguish the importance of different feature interactions. While AFM can obtain more accurate rating prediction by distinguishing the importance of different feature interactions, and alleviate the influence of noise that may be introduced by those useless feature interactions. Therefore, we adopt AFM to learn user–item feature interactions and obtain $\hat{y}(x)$.

Specifically, we concatenate $R_u \in \mathbb{R}^k$ with $R_i \in \mathbb{R}^k$ into a joint vector $x = (x_1, x_2, \ldots, x_{2k})$. Given $x \in \mathbb{R}^{2k}$ as input of AFM, it outputs the predicted rating $\hat{y}(x)$, and ensures that each user–item feature interaction in the joint vector reflects different importance. $\hat{y}(x)$ is represented as:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{|x|} w_i x_i + p^T \sum_{i=1}^{|x|} \sum_{j=i+1}^{|x|} \alpha_{ij}(v_i \otimes v_j) x_i x_j + b_u + b_i \tag{11}$$

where w_0 denotes the global bias term, w_i is the weight of the primary term, $|x|$ represents the feature number of the joint vector x. $p \in \mathbb{R}^d$ represents the weights vector for rating prediction layer. $v_i \in \mathbb{R}^d$ is an embedding vector corresponding to a certain dimension x_i. Similarly, $v_j \in \mathbb{R}^d$ is an embedding vector corresponding to a certain dimension x_j, and d is the size of embedding vector. b_u represents the user bias term, and b_i represents the item bias term. $\otimes$ represents the element-wise product of embedding vectors, α_{ij} represents the attention weight, which is calculated by:

$$\alpha_{ij} = \frac{\exp\left(\alpha'_{ij}\right)}{\sum_{i,j \in |x|, j>i} \exp\left(\alpha'_{ij}\right)} \tag{12}$$

where α'_{ij} represents the attention score of the feature interaction of x_i and $x_j\left(i, j \in |x|, j > i\right)$, which is computed by:

$$\alpha'_{ij} = h^T \mathrm{ReLU}(\mathbf{W}(v_i \otimes v_j) x_i x_j + b) \tag{13}$$

where $h \in \mathbb{R}^t$ represents the weights vector from the fully connected layer to the softmax output layer, t represents the size of hidden layer of the attention network in AFM. $\mathbf{W} \in \mathbb{R}^{t \times d}$, $b \in \mathbb{R}^t$ represent the weight matrix, the bias term, respectively.

On the basis of above operations, item recommendation can be performed according to the obtained predicted ratings.

Model learning. The squared loss function is widely used in the rating prediction task of the recommender system, so we adopt this loss function, defined as:

$$L = \sum_{z \in S} (\hat{y}(z) - y(z))^2 \tag{14}$$

where S represents the training samples, $\hat{y}(z)$ represents the predicted rating of a sample z, and $y(z)$ represents the real rating of sample z.

Experiments

In this section, we conduct experiments to evaluate the effectiveness of our proposed AFMRUI model on five real-world datasets. We first introduce the experimental setup, including datasets and preprocessing, evaluation metrics, baseline methods and experimental configuration. Afterwards, we conduct the performance comparisons and also demonstrate the corresponding ablation studies. Furthermore, we analyze the effects of different parameters on the performance of AFMRUI and discuss the impacts of different embedding representation methods and different feature interaction methods on model performance.

Experimental setup. *Datasets and preprocessing.* We evaluate the AFMRUI model on five real-world datasets with different scales and industries. Among them, four Amazon datasets, including Digital Music, Baby, Office Products and Beauty, contain real Amazon reviews from May 1996 to July 2014, and Yelp dataset for the Yelp Challenge. Each sample in each dataset includes userID, itemID, review, ratings, etc. Moreover, users in each dataset have posted at least five reviews on the corresponding items. Table 2 shows the statistics of five datasets.

To ensure the model is well trained, the samples from five datasets need to be preprocessed. According to the sample format described in "Problem definition", we mainly use the values of four fields mentioned above in samples from each dataset. Then, we use a Pandas tool to preprocess the original samples from each dataset

Datasets	Users	Items	Samples
Digital music	5541	3568	64,706
Baby	19,445	7050	160,792
Office products	4905	2420	53,258
Beauty	22,363	12,101	198,502
Yelp	1,144,046	174,013	5,000,000
Average	239,260	39,830	1,095,452

Table 2. Statistics of five datasets.

and extract four attributes, including userID, itemID, user's reviews on the item, and user's rating on the item (1–5 points). As a result, every sample is unified as a userID-itemID-review-rating quadruplet by preprocessing to facilitate the input model for training.

Evaluation metrics. We leverage mean square error (MSE) and mean absolute error (MAE) to evaluate the performance of different methods. The two metrics are utilized to measure the accuracy of rating prediction by computing the difference between predicted and actual ratings. Lower MSE and MAE values indicate higher accuracy of model prediction. The formulas for calculating MSE and MAE are:

$$\mathrm{MSE} = \frac{1}{|T|}\sum_{a\in T}(\hat{y}(a) - y(a))^2 \tag{15}$$

$$\mathrm{MAE} = \frac{1}{|T|}\sum_{a\in T}|\hat{y}(a) - y(a)| \tag{16}$$

where T represents the test samples, $|T|$ represents the number of samples in the test set, $\hat{y}(a)$ denotes the predicted rating of a test sample a, $y(a)$ is the real rating of sample a from the corresponding test dataset.

Baseline methods. To demonstrate the effectiveness of our AFMRUI model, we select a traditional recommendation model based on matrix factorization and nine models based on neural networks. The selected representative baseline methods are described as follows.

- **M**atrix **F**actorization (**MF**)[39]: This method is a regression algorithm, which only takes rating data as input, and obtains user and item features by matrix factorization.
- **Deep Co**operative **N**eural **N**etworks (**DeepCoNN**)[14]: This model utilizes two parallel convolutional layers to process review documents for users and items, respectively, and uses FM to perform rating prediction, which shows that review information can alleviate the sparsity problem of rating data.
- **D**ual **Att**ention-based **n**etwork (**D-Attn**)[15]: This model obtains review-based feature representations of users and items by combining local and global learning, and finally predicts ratings by using dot product.
- **Trans**formational Neural **Net**work**s** (**TransNets**)[40]: This model adds a transform layer to DeepCoNN, which mainly transforms the latent representations of reviews into user and item features, and uses FM to predict ratings.
- **N**eural **A**ttentional **R**egression Model with **R**eview-level **E**xplanations (**NARRE**)[16]: This model learns user and item features using CNN and attention mechanism, and uses LFM for rating prediction.
- **M**ulti-**P**ointer **C**o-attention **N**etworks (**MPCN**)[28]: This model uses a pointer network to learn user and item features from reviews and uses FM for rating prediction.
- **D**ual **A**ttention **M**utual **L**earning (**DAML**)[17]: This model utilizes local and mutual attention of CNN to jointly learn user and item features from reviews, and neural factorization machine is introduced to predict ratings.
- **N**eural **C**ollaborative **E**mbedding **M**odel (**NCEM**)[41]: This model utilizes an aspect-level attention layer to measure the correlation degree of reviews towards different aspects, and a multi-layer neural factorization machine is introduced to predict ratings.
- **C**ross-domain Recommendation Framework Via **A**spect **T**ransfer **N**etwork (**CATN**)[42]: The model learns the aspect level features of each user and item from the corresponding reviews through attention mechanism, then semantic matching is performed between such aspect level features to predict ratings.
- **M**atch **P**yramid **R**ecommender **S**ystem (**MPRS**)[43]: This model uses a CNN architecture fed by the matching matrix of corresponding reviews for a pair of user–item, and a regression layer is introduced to predict ratings.

Configuration. In our experiments, the code was written in Python 3.8, and TensorFlow 1.15.5 was utilized as a framework. All experiments were conducted on a Linux server with Intel(R) Xeon(R) Gold 6330 CPU and RTX 3090 24 GB GPU. We randomly divided each dataset used in the experiments into training set, validation set and test set according to the proportion of 8:1:1. Furthermore, we selected parameters on the validation set and performed evaluation on the test set. The settings of other parameters are described as follows:

- For MF[39] method, the latent dimensions of users and items are uniformly set to 50.
- For DeepCoNN[14], D-Attn[15], TransNets[40], NARRE[16], MPCN[28], DAML[17], NCEM[41], CATN[42] and MPRS[43] , we set the parameters for the methods based on the setting strategies in the corresponding paper. More specifically, learning rate is 0.002, dropout is set from {0.1, 0.3, 0.5, 0.7, 0.9}, and batch size is set from {32, 64, 128, 256, 512} to find the best parameters. The ID embedding dimension is set to 32 in NARRE and DAML model; in D-Attn, NARRE, DAML, NCEM and CATN models, the dimension of the attention score vector is set to 100; in DeepCoNN, TransNets, NARRE, CATN and MPRS models, CNN is used to process reviews, where the size of each convolution kernel is set to 3, and the number of convolution kernel is set to 50; the word vector model adopted is Glove and the embedding dimension is 100; in NCEM, the version of BERT is "BERT-base". Note that if FM is used in any model, the latent dimension is set to 32.
- For our proposed model AFMRUI, we carefully tested batch size from {32, 64, 128, 256, 512} and looked for the optimal value of learning rate from {0.0001, 0.0005, 0.001, 0.005} for each dataset. To prevent overfitting, we turned dropout from {0.1, 0.3, 0.5, 0.7, 0.9}. Then, batch size is set to 512, learning rate is set to 0.001,

dropout is set to 0.3, and Adam is used as the optimizer. The unified maximum length of reviews is set to 100. The version of RoBERTa is "RoBERTa-base", where we subsequently add a fully connected layer to compress the semantic feature dimension c. The hidden unit number t_1 is set to 50 in attention layer. The size d of embedding vector is set to 6 in rating prediction layer. The other parameters are determined by optimizing MSE and MAE on a validation set from each dataset.

Results and discussions. *Comparison of model performance.* In this subsection, we compare the performance of eleven methods on five datasets. Table 3 shows the results, with the best-performing ones highlighted in bold. From Table 3, we can make the following observations.

First, our proposed model, AFMRUI, outperforms other models in terms of MSE and MAE on five datasets. Notably, when compared with the best baseline method (MPRS), AFMRUI enhances performance on Digital Music dataset by approximately 3.7% for MSE and 2.1% for MAE. Similarly, high performance gains are observed on the other four datasets. These results demonstrate the superiority of our model.

Second, methods utilizing review information generally work better than those that only consider the rating data. It is clear that, DeepCoNN, D-Attn, TransNets, NARRE, MPCN, DAML, NCEM, CATN, MPRS and AFMRUI perform better than MF in terms of MSE and MAE on five datasets. The performance improvements of these methods may be due to leveraging neural networks for rating prediction by using review information, which can effectively capture user/item features from review information, and reduce the effect of data sparsity due to only using rating data. Therefore, these methods utilizing review information gain pure improvement compared with MF.

Third, our proposed AFMRUI model performs better than nine baseline models leveraging review information on five datasets. The reason is that, in our model, RoBERTa can capture global context and mitigate the problem of polysemy in user/item reviews, in which the accurately understanding of review information is guaranteed. Moreover, our model uses AFM, rather than dot product and FM, to learn different feature interactions and further to distinguish the importance of different feature interactions, which can also alleviate the effect of noise that may be introduced by useless feature interactions, so that AFMRUI achieves better performance on five datasets.

In addition, for each of these eleven methods, we also provide an order of magnitude of approximate model parameters for comparison, as shown in the second column in Table 3, where M represents millions. The comparison results from Table 3 show that ten deep learning-based methods have more parameters compared with MF, mainly due to the fact that deep learning models usually contain a multi-layer neural network, and each layer contains a large number of parameters. While NCEM and AFMRUI have much more model parameters compared with the other eight deep learning-based methods, mainly because both methods use pre-trained models to encode reviews, and pre-trained models need to learn a lot of linguistic knowledge and laws to have stronger expression and generalization ability. Compared with NCEM, AFMRUI has more model parameters, mainly because our model leverages the pre-trained model RoBERTa, which has been made improvements in model structure and optimization algorithms on the basis of BERT used in NCEM, thus requiring more parameters than NCEM.

Effectiveness of different components. In this subsection, we performed ablation experiments to analyze the effects of different components to model performance.

In order to validate the benefits brought by each component, we construct the following variants of AFMRUI based on the basic model, AFMRUI-base, which uses static word vector model Glove to represent user/item review embedding features and predicts user's rating on an item by FM.

	Params (M)	Digital music		Baby		Office products		Beauty		Yelp	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
MF[39]	0.126	1.956 ± 0.002	1.204 ± 0.009	1.755 ± 0.001	1.320 ± 0.005	1.143 ± 0.008	0.996 ± 0.009	1.950 ± 0.004	1.381 ± 0.006	1.828 ± 0.009	1.526 ± 0.005
DeepCoNN[14]	6.303	1.202 ± 0.009	0.722 ± 0.046	1.440 ± 0.005	0.873 ± 0.037	0.909 ± 0.003	0.707 ± 0.004	1.453 ± 0.015	0.922 ± 0.008	1.687 ± 0.003	1.361 ± 0.028
D_Attn[15]	9.152	1.014 ± 0.015	0.697 ± 0.007	1.325 ± 0.004	0.849 ± 0.001	0.815 ± 0.006	0.754 ± 0.005	1.419 ± 0.009	0.845 ± 0.007	1.651 ± 0.003	1.358 ± 0.009
TransNets[40]	15.373	1.055 ± 0.004	0.701 ± 0.002	1.334 ± 0.005	0.853 ± 0.003	0.824 ± 0.005	0.746 ± 0.005	1.412 ± 0.005	0.841 ± 0.007	1.623 ± 0.005	1.119 ± 0.003
NARRE[16]	11.297	0.965 ± 0.002	0.686 ± 0.005	1.312 ± 0.009	0.851 ± 0.006	0.817 ± 0.021	0.727 ± 0.004	1.396 ± 0.007	0.828 ± 0.002	1.571 ± 0.006	1.014 ± 0.007
MPCN[28]	11.879	0.970 ± 0.005	0.729 ± 0.004	1.304 ± 0.007	0.858 ± 0.005	0.779 ± 0.004	0.670 ± 0.004	1.386 ± 0.008	0.894 ± 0.001	1.608 ± 0.017	1.106 ± 0.007
DAML[17]	14.004	0.959 ± 0.021	0.705 ± 0.003	1.298 ± 0.002	0.853 ± 0.005	0.791 ± 0.007	0.689 ± 0.013	1.379 ± 0.007	0.843 ± 0.004	1.581 ± 0.009	1.052 ±0.008
NCEM[41]	110.334	0.956 ± 0.008	0.691 ± 0.012	1.290 ± 0.018	0.851 ± 0.002	0.788 ± 0.003	0.667 ± 0.002	1.370 ± 0.002	0.816 ± 0.001	1.567 ± 0.002	1.001 ± 0.001
CATN[42]	32.193	0.952 ± 0.013	0.678 ± 0.002	1.285 ± 0.005	0.847 ± 0.007	0.774 ± 0.002	0.655 ± 0.011	1.366 ± 0.003	0.806 ± 0.003	1.554 ± 0.001	0.993 ± 0.001
MPRS[43]	18.137	0.947 ± 0.002	0.678 ± 0.006	1.282 ± 0.005	0.845 ± 0.002	0.772 ± 0.010	0.653 ± 0.009	1.361 ± 0.005	0.800 ± 0.006	1.548 ± 0.004	0.981 ± 0.015
AFMRUI	**127.452**	**0.910 ± 0.002**	**0.657 ± 0.009**	**1.256 ± 0.003**	**0.821 ± 0.008**	**0.740 ± 0.007**	**0.638 ± 0.001**	**1.341 ± 0.010**	**0.786 ± 0.004**	**1.502 ± 0.003**	**0.954 ± 0.009**

Table 3. Performance comparison on five datasets (mean ± std).

- AFMRUI-Ro: This model uses RoBERTa instead of Glove to obtain user/item review embedding features on the basis of AFMRUI-base. This variant model is to verify that RoBERTa is better than Glove in extracting review embedding features.
- AFMRUI-Bi: In this model, BiGRU is added on the basis of AFMRUI-Ro to encode each user/item review embedding features output from RoBERTa. This variant model is to verify the effectiveness of BiGRU.
- AFMRUI-Att: This model adds an attention network on the basis of Review-Bi, and this variant model is to verify the effectiveness of the attention network in measuring the contribution of each review to user/item feature representation.

Table 4 shows the models with different components. We take two metrics to demonstrate the effectiveness of the models from Table 4 on five datasets. The results are shown in Table 5.

It can be seen from Table 5, the model performance of AFMRUI-Ro has been improved compared with the basic model, indicating that using RoBERTa to obtain context-related user/item review embedding features, which can alleviate the problem of polysemy and effectively enhance the feature representation. Compared with AFMRUI-Ro, AFMRUI-Bi performs better mainly because BiGRU is more suitable for dealing with sequence problems and can fully exploit the internal dependencies among reviews. While the performance of AFMRUI-Bi is worse than AFMRUI-Att, because the attention network introduced can adaptively measure the importance of each review to user/item feature representation, enabling the model to focus on more useful reviews.

In contrast, the performance of our proposed AFMRUI model is better than the other four variant models, which shows that AFM can better learn the feature interactions of users and items to obtain more accurate prediction rating, and also demonstrates that integrating these components can help to better model review features of users and items, so as to improve the model performance.

Effect of parameters. In this section, we analyzed the effects of different model parameters on the performance of AFMRUI. Here, we focused on five critical parameters, namely, the maximum number of user reviews n and item reviews m, the semantic feature dimension c, GRU hidden dimension l and the latent dimension k. Next, we analyzed the effects of five parameters on two metrics.

Effect of maximum number of reviews. The proposed AFMRUI model performs rating prediction based on user reviews and item reviews. Therefore, the maximum number of user reviews n and item reviews m directly affects the feature representations of users and items, thereby affecting the model performance. Considering that different datasets have different numbers of reviews for different users and different items, so we make statistics on the number of user reviews and item reviews from five datasets to determine the range for the maximum number of reviews, as shown in Table 6.

Take digital music dataset (the second row in Table 6) as an example, 4449 users have up to 13 reviews, accounting for 80.29% of the total number of users, and 2892 items have up to 20 reviews, accounting for 81.05% of the total number of items. According to the statistical results, considering that the noise will be introduced if the number of reviews is too large, and less effective information is extracted if the number of reviews is too small, so we set the range for maximum number of user reviews to {8, 9, 10, 11, 12, 13}, and the range for maximum number of item reviews to {15, 16, 17, 18, 19, 20}. Similarly, we set the ranges for maximum number of reviews from the other four datasets while keeping other hyper-parameters unchanged. Figure 3 shows the results on

Models	RoBERTa	BiGRU	Attention	AFM
AFMRUI-base	\	\	\	\
AFMRUI-Ro	✓	\	\	\
AFMRUI-Bi	✓	✓	\	\
AFMRUI-Att	✓	✓	✓	\
AFMRUI	✓	✓	✓	✓

Table 4. Comparison of models with different components.

	Digital music		Baby		Office products		Beauty		Yelp	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
AFMRUI-base	1.025	0.722	1.331	0.880	0.844	0.695	1.417	0.882	1.714	1.353
AFMRUI-Ro	0.968	0.681	1.309	0.856	0.796	0.672	1.392	0.822	1.605	1.121
AFMRUI-Bi	0.957	0.667	1.291	0.844	0.783	0.661	1.377	0.816	1.564	1.047
AFMRUI-Att	0.943	0.675	1.274	0.830	0.766	0.650	1.365	0.806	1.537	0.991
AFMRUI	**0.910**	**0.657**	**1.256**	**0.821**	**0.740**	**0.638**	**1.341**	**0.786**	**1.502**	**0.954**

Table 5. Effectiveness of different components on five datasets. Significant values are in [bold].

Datasets	Number of users	Percentage of total users (%)	Number of items	Percentage of total items (%)
Digital music	4449 ($n \leq 13$)	80.29	2892 ($m \leq 20$)	81.05
Baby	15,991 ($n \leq 10$)	82.23	5666 ($m \leq 28$)	80.37
Office products	4024 ($n \leq 10$)	82.04	1979 ($m \leq 11$)	81.78
Beauty	18,117 ($n \leq 10$)	81.01	9712 ($m \leq 19$)	80.26
Yelp	932,169 ($n \leq 15$)	81.48	139,767 ($m \leq 20$)	80.32

Table 6. Statistics of reviews from five datasets.

five datasets. Since the results on MAE are similar to that on MSE, so we take MSE as an example to analyze the effects of the parameters on model performance.

As shown in Fig. 3a, for digital music dataset, with the increase of n and m, MSE decreases first and then increases. This is because when the number of reviews is too large, noise may be introduced to affect the feature representations of users and items. While the number of reviews is too small to accurately express the feature representations of users and items. Therefore, we set the maximum number of user reviews n to 10 and set the maximum number of item reviews m to 20 that can get the best performance on digital music dataset. Similarly, the maximum number of user reviews and item reviews are set to $n = 10$, m = 23 on Baby dataset, respectively; for office products dataset, $n = 8$ and $m = 10$; for beauty dataset, $n = 10$ and $m = 15$; for Yelp, $n = 10$ and $m = 15$. According to the above analysis, we select such values as the corresponding maximum numbers of user reviews and item reviews on five datasets.

Effect of semantic feature dimension c. In order to investigate how sensitive AFMRUI is to the semantic feature dimension c, we fixed the dimension of the review embedding feature output by RoBERTa to 768, and further obtained the corresponding review embedding features with different semantic feature dimension c through fully connected layer compression. We demonstrated the effects of c on five datasets in Fig. 4. As shown in Fig. 4, for five datasets, with the increase of c, the model performance is gradually improved. When c is 256, the model performance reaches the best, and then begins to decline. Moreover, the computational cost is also increasing. Therefore, we set the semantic feature dimension c to 256 that can get the best performance on five datasets.

Effect of GRU hidden dimension l. To illustrate the effect of GRU hidden dimension l, we set values of l as 50, 100, 150, 200, 250, 300 while keeping other hyper-parameters unchanged. Figure 5 shows the results on five datasets. The curves show the trend of falling first and then rising on five datasets. This maybe because when GRU hidden dimension is too small, it cannot fully mine the internal dependencies among review embedding features. While when GRU hidden dimension is too large, it will make the model over-fitting. Therefore, similar to selection of the semantic dimension c, we set GRU hidden dimension to 200 that can get the best performance on five datasets.

Effect of latent dimension k. In this subsection, we investigate the impact of latent dimension k on model performance while keeping other parameters unchanged. The results are presented in Fig. 6. We observe that as k increases, MSE and MAE first decrease for digital music, baby, beauty and Yelp datasets, reach the best when k is 32, and increase thereafter. For office products dataset, MSE and MAE reach the best when k is 64. This is because a small value of k may lead to the model being unable to capture all potential information from user and item reviews, while a large value of k may cause over-fitting and increase the model complexity. Therefore, we set k to 64 on Office Products dataset and 32 on the other four datasets.

Comparison of different embedding representation methods. In this section, we discuss the impact of different embedding representation methods on the model performance. Here, we select a classical algorithm DeepCoNN[14] and the best baseline method MPRS[43] with different embedding representations. As shown in Table 7, we mainly discuss nine comparison methods.

The experimental results reported in Table 7 shows that our proposed model, AFMRUI, outperforms its variants, AFMRUI-Glove and AFMRUI-BERT-base, in terms of MSE and MAE on all five datasets. Specifically, on the Yelp dataset, AFMRUI improves performance by approximately 3.8% on MSE and 3.5% on MAE compared with AFMRUI-Glove; and the relative performance improvements are 1.5% on MSE and 1.1% on MAE compared with AFMRUI-BERT-base. The other four datasets show similarly high performance gains. These results essentially demonstrate the competitiveness of the proposed model using RoBERTa to obtain context-related user/item review embedding features, which can alleviate the problem of polysemy and effectively enhance the feature representation.

In addition, we also compared DeepCoNN[14], MPRS[43], and their variant models. The experimental results show that DeepCoNN-BERT-base and DeepCoNN-RoBERTa-base outperform DeepCoNN-Glove, MPRS-BERT-base and MPRS-RoBERTa-base outperform MPRS-Glove, mainly because the traditional word vector model cannot rely on the before-and-after review information in the review set for efficient representations of users and items. However, BERT-base and RoBERTa-base can alleviate this problem. Whereas DeepCoNN-RoBERTa-base outperforms DeepCoNN-BERT-base, MPRS-RoBERTa-base outperforms MPRS-BERT-base, mainly because RoBERTa-base not only inherits the advantages of BERT-base, but also uses new hyperparameters and a new

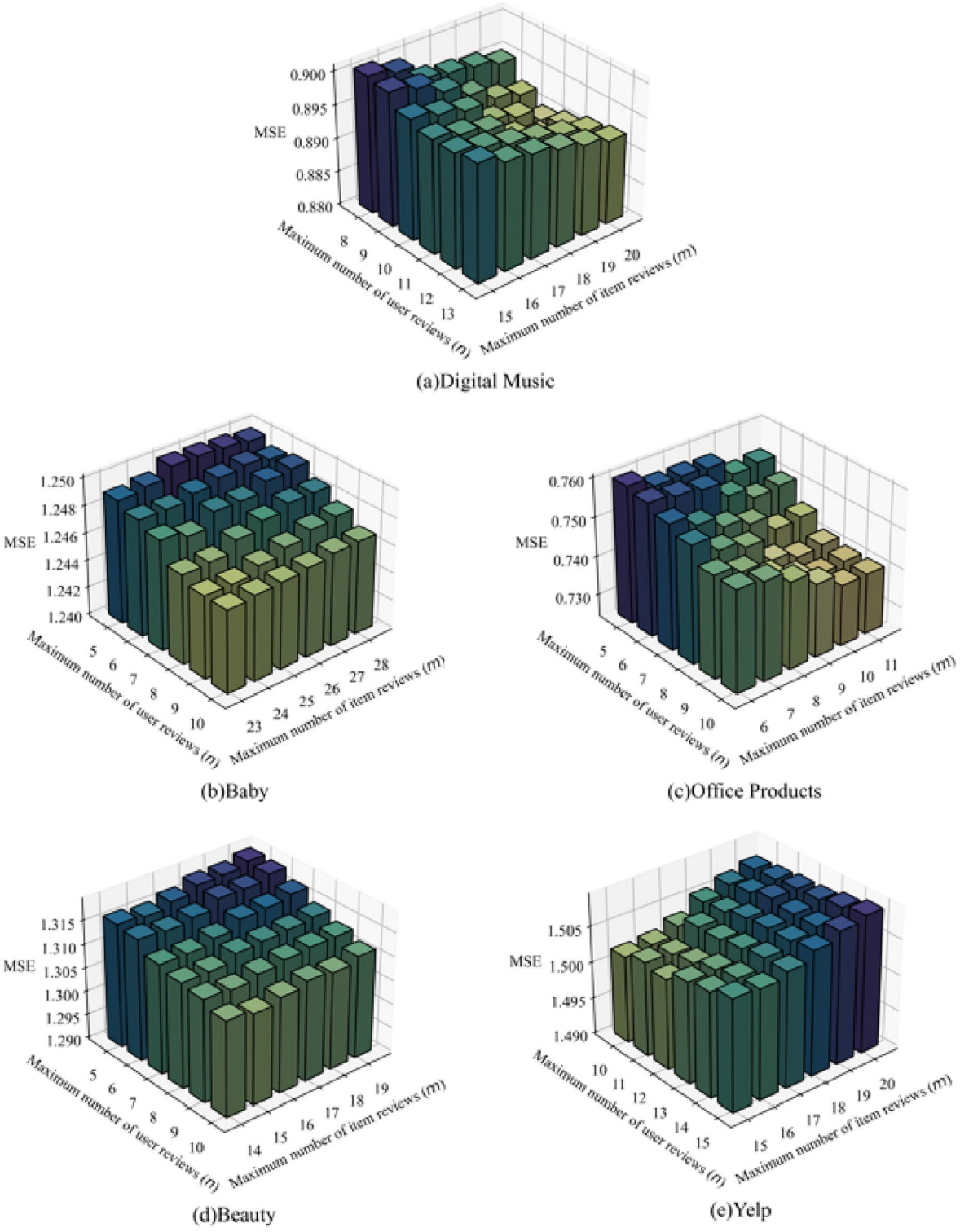

Figure 3. Effect of maximum number of user reviews and item reviews on model performance.

large dataset for retraining. Not only does it alleviate the problem of multiple meanings of words in reviews, but it also better models the global information and semantic representations of user and item reviews, resulting in more accurate predictive scores and better model performance.

Comparison of different feature interaction methods. In this section, we discuss the impact of different feature interaction methods on the model performance. We mainly discuss the following three methods.

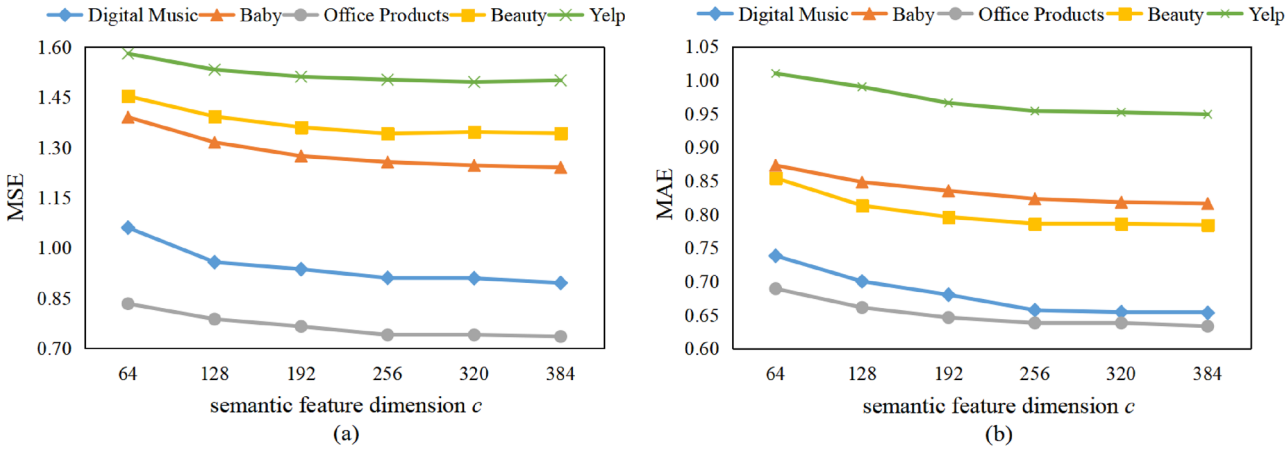

Figure 4. Effect of semantic feature dimension c on model performance.

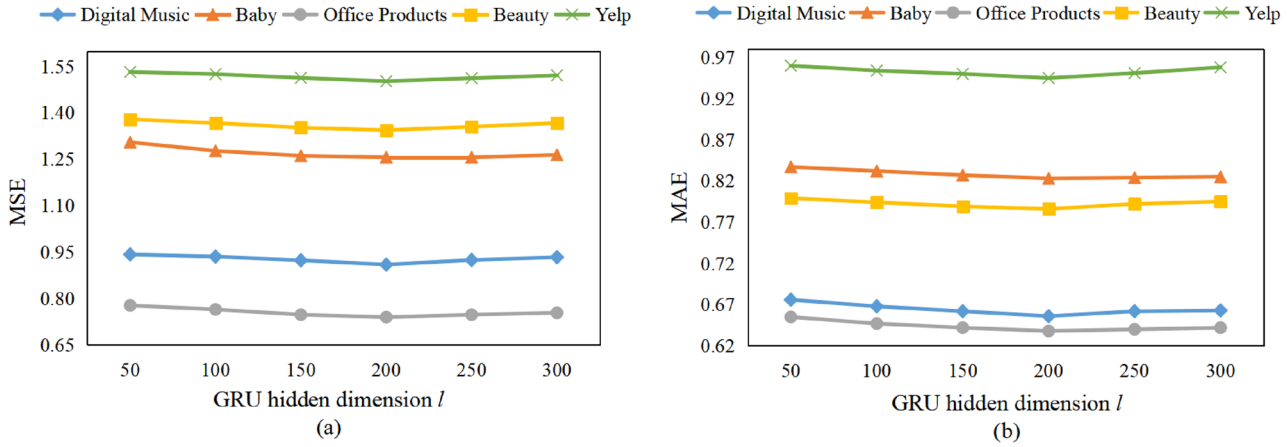

Figure 5. Effect of GRU hidden dimension l on model performance.

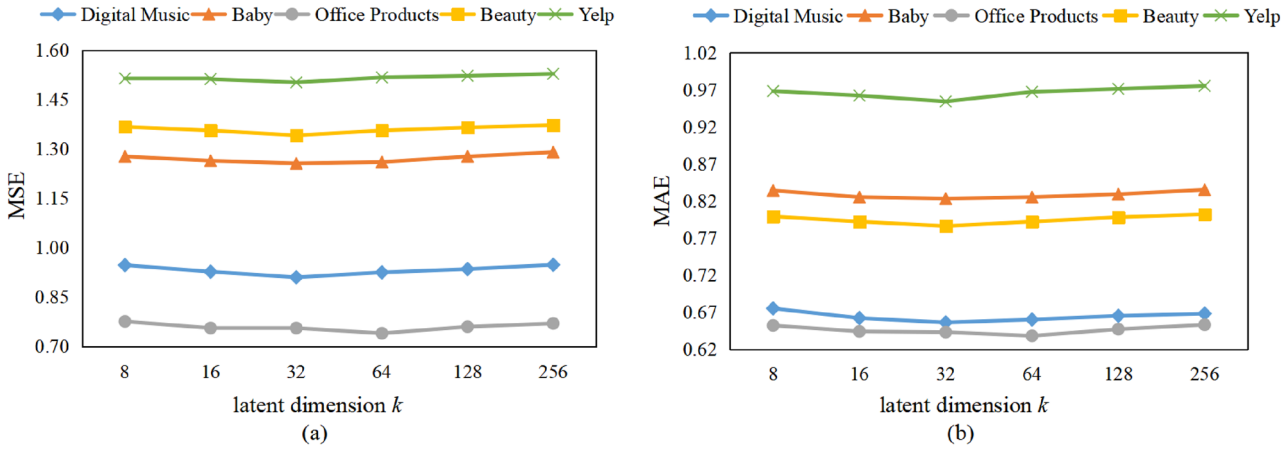

Figure 6. Effect of latent dimension k on model performance.

- AFMRUI-dp: The method conducts dot product operation on user review embedding and item review embedding to predict rating.
- AFMRUI-FM: This approach encodes a vector formed by concatenating user and item review embeddings through FM.

	Digital music		Baby		Office products		Beauty		Yelp	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DeepCoNN-Glove[14]	1.202	0.722	1.440	0.873	0.909	0.707	1.453	0.922	1.687	1.361
DeepCoNN-BERT-base	1.185	0.706	1.417	0.869	0.873	0.684	1.424	0.907	1.650	1.322
DeepCoNN-RoBERTa-base	1.172	0.698	1.403	0.856	0.856	0.675	1.406	0.897	1.633	1.309
MPRS-Glove[43]	0.947	0.678	1.282	0.845	0.772	0.653	1.361	0.800	1.548	0.981
MPRS-BERT-base	0.925	0.676	1.262	0.833	0.760	0.648	1.358	0.804	1.537	0.971
MPRS-RoBERTa-base	0.919	0.664	1.258	0.827	0.756	0.644	1.350	0.798	1.525	0.964
AFMRUI-Glove	0.934	0.675	1.280	0.845	0.767	0.658	1.365	0.809	1.540	0.979
AFMRUI-BERT-base	0.918	0.662	1.266	0.827	0.751	0.644	1.347	0.799	1.517	0.965
AFMRUI	**0.910**	**0.657**	**1.256**	**0.821**	**0.740**	**0.638**	**1.341**	**0.786**	**1.502**	**0.954**

Table 7. Effect of different embedding representation methods on model performance. Significant values are in [bold].

- AFMRUI: Our proposed method, uses AFM to learn the feature interactions of users and items to perform rating prediction.

Table 8 shows the results on five datasets. It can be seen from Table 8, AFMRUI-dp experiences the most performance decrease compared with AFMRUI-FM and AFMRUI on five datasets, whereas AFMRUI has the best performance. This is because dot product operation used by AFMRUI-dp cannot fully explore the complex internal structure of the joint vector of user review embedding and item review embedding. While the advantage of FM over dot product operation is that it can capture feature interactions between any two dimensions in the joint vector of user review embedding and item review embedding. Thus, the performance of AFMRUI-FM is better than AFMRUI-dp.

Compared with AFMRUI-FM, our AFMRUI model is more effective because AFM in our model adds attention mechanism on the basis of FM, and it can further distinguish the importance of different feature interactions, which can alleviate the effect of noise possibly introduced by useless feature interactions, so as to obtain more accurate prediction rating and then improve the model performance.

On the basis of above analysis, in order to further explore the contribution of different feature interactions in our AFMRUI model more intuitively, we use Digital Music dataset as an example to demonstrate the contributions of different feature interactions. Since our AFMRUI model achieves the best results on the Digital Music dataset when the number of latent dimensions k is 32, the dimensions of both user review embedding R_u and item review embedding R_i is set to 32, and the dimension of vector x stitched together from them is 64, i.e., $x = (R_u, R_i) = (x_1\text{-}x_{32}, x_{33} - x_{64})$. Where $x_1 - x_{32}$ is defined as user interaction object U and $x_{33} - x_{64}$ is defined as item interaction object I, so there are three types of feature interactions in vector x, as shown in Table 9. A user–item feature interaction (e.g., $x_1\, x_{33}$) can be formed by taking a random dimension from U and I. Repeatedly, we select 10 different user–item feature interactions with feature interaction type U–I. Similarly, we obtain 10 different feature interactions with the other two types, respectively. The attention scores of these feature interactions are shown in Fig. 7.

As shown in Fig. 7, the lighter the color, the lower the attention score and the less it contributes to the model performance, and vice versa. Specifically, the feature interaction type U–I, which has been adopted by models such as DeepCoNN[14] and TransNets[40], achieved good results, indicating that user–item feature interactions are beneficial for the quality of rating prediction. However, according to Fig. 7, it can be seen that the attention scores for U–I feature interactions are stable between 0.2 and 0.5, indicating that not all user–item feature interactions have positive impacts on the rating prediction. While the other types of U–U and I–Ihave more higher attention scores, mainly in the range of 0.5–0.9, indicating that although they are the same interaction objects, the feature interactions between them are more important and can have positive impacts on the model performance, resulting in more accurate prediction of user's rating of an item, and thus provide better recommendation.

In summary, it can be seen that different feature interactions have different attention scores and have different impacts on model performance. While AFM adopted in our model can distinguish the importance of different

	Digital music		Baby		Office products		Beauty		Yelp	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
AFMRUI-dp	0.968	0.681	1.301	0.844	0.785	0.661	1.380	0.816	1.564	1.019
AFMRUI-FM	0.943	0.675	1.274	0.830	0.766	0.650	1.365	0.806	1.535	0.991
AFMRUI	**0.910**	**0.657**	**1.256**	**0.821**	**0.740**	**0.638**	**1.341**	**0.786**	**1.502**	**0.954**

Table 8. Effect of different feature interaction methods on model performance. Significant values are in [bold].

Types of feature interactions	Interaction object
U–U	User–user
U–I	User–item
I–I	Item–item

Table 9. User–item feature interaction type.

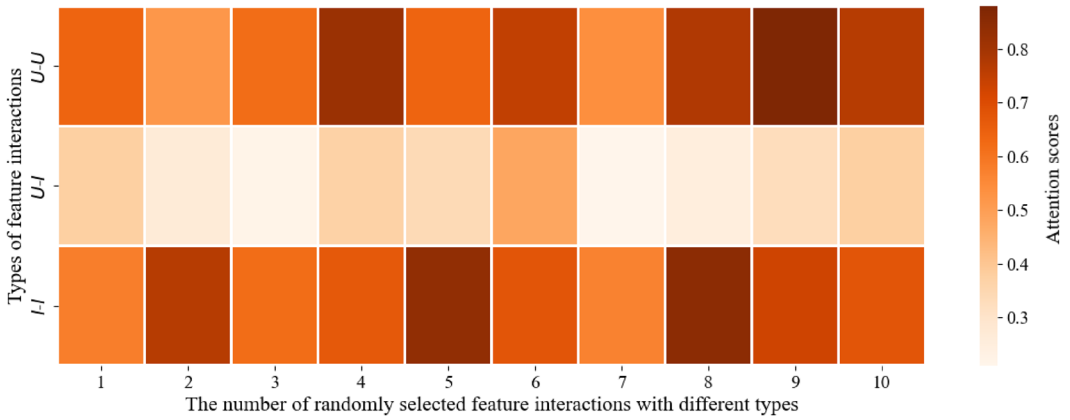

Figure 7. Attention scores of feature interactions with different types.

feature interactions through the obtained attention scores, thereby alleviating the impact of useless feature interactions on model performance.

Conclusions

In recent years, the review-based recommendation is one of hot research topics in recommender systems. In this paper, we proposed an AFMRUI model for recommendation. Specifically, AFMRUI leverages RoBERTa to mitigate the problem of polysemy in user/item reviews, and learns reviews embedding of users and items through BiGRU and attention network, so as to better model user review embedding and item review embedding. Then it utilizes AFM to learn user–item feature interactions, which can obtain more accurate prediction rating by distinguishing the importance of different feature interactions. Extensive experiments on five publicly available datasets have demonstrated that the proposed AFMRUI model outperforms the state-of the-art methods regarding two metrics.

In this paper, we just leverage review information to extract users and items features. Recently, studies have shown that user–item interaction graph[44,45] has additional useful information to promote recommendation. Therefore, in the future work, we will combine review information with user–item interaction graph to capture more accurate features of users and items, so as to provide better model performance.

Data availability

The data used to support the findings of this study are available from http://jmcauley.ucsd.edu/data/amazon/ and https://www.yelp.com/dataset.

Code availability

The source code of the proposed model is publicly available for download at Github: https://github.com/Jindidi/AFMRUI.git.

Received: 2 March 2023; Accepted: 14 August 2023
Published online: 18 August 2023

References

1. Mandal, S. & Maiti, A. Deep collaborative filtering with social promoter score-based user-item interaction: A new perspective in recommendation. *Appl. Intell.* **51**, 7855–7880. https://doi.org/10.1007/s10489-020-02162-9 (2021).
2. Wang, N. Ideological and political education recommendation system based on AHP and improved collaborative filtering algorithm. *Sci. Program* **2021**, 2648352:1-2648352:9. https://doi.org/10.1155/2021/2648352 (2021).
3. Zhu, Z., Wang, S., Wang, F. & Tu, Z. Recommendation networks of homogeneous products on an e-commerce platform: Measurement and competition effects. *Expert Syst. Appl.* **201**, 117128. https://doi.org/10.1016/j.eswa.2022.117128 (2022).
4. Baczkiewicz, A., Kizielewicz, B., Shekhovtsov, A., Watróbski, J. & Salabun, W. Methodical aspects of MCDM based e-commerce recommender system. *J. Theor. Appl. Electron. Commerce Res.* **16**, 2192–2229. https://doi.org/10.3390/jtaer16060122 (2021).

5. Li, Z., Huang, X., Liu, C. & Yang, W. Spatio-temporal unequal interval correlation-aware self-attention network for next POI recommendation. *ISPRS Int. J. Geo Inf.* **11**, 543. https://doi.org/10.3390/ijgi11110543 (2022).
6. Tahmasbi, H., Jalali, M. & Shakeri, H. Modeling user preference dynamics with coupled tensor factorization for social media recommendation. *J. Ambient Intell. Humaniz. Comput.* **12**, 9693–9712. https://doi.org/10.1007/s12652-020-02714-4 (2021).
7. Covington, P., Adams, J. & Sargin, E. Deep neural networks for Youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19* (Sen, S., Geyer, W., Freyne, J. & Castells, P. eds.), 191–198. https://doi.org/10.1145/2959100.2959190 (ACM, 2016).
8. Li, D., Wang, C., Li, L. & Zheng, Z. Collaborative filtering algorithm with social information and dynamic time windows. *Appl. Intell.* **52**, 5261–5272. https://doi.org/10.1007/s10489-021-02519-8 (2022).
9. Hu, G. *et al.* Collaborative filtering with topic and social latent factors incorporating implicit feedback. *ACM Trans. Knowl. Discov. Data* **12**, 23:1-23:30. https://doi.org/10.1145/3127873 (2018).
10. Yin, Y., Chen, L., Xu, Y. & Wan, J. Location-aware service recommendation with enhanced probabilistic matrix factorization. *IEEE Access* **6**, 62815–62825. https://doi.org/10.1109/ACCESS.2018.2877137 (2018).
11. Zhang, Z., Liu, Y., Xu, G. & Luo, G. X. Recommendation using dmf-based fine tuning method. *J. Intell. Inf. Syst.* **47**, 233–246. https://doi.org/10.1007/s10844-016-0407-6 (2016).
12. Shang, T., Li, X., Shi, X. & Wang, Q. Joint modeling dynamic preferences of users and items using reviews for sequential recommendation. In *Advances in Knowledge Discovery and Data Mining—25th Pacific-Asia Conference, PAKDD 2021, Virtual Event, May 11–14, 2021, Proceedings, Part II, vol. 12713 of Lecture Notes in Computer Science* (Karlapalem, K. et al. eds.), 524–536. https://doi.org/10.1007/978-3-030-75765-6_42 (Springer, 2021).
13. Kim, D. H., 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, September 15-19* (Sen, S., Geyer, W., Freyne, J. & Castells, P. eds.), 233–240. https://doi.org/10.1145/2959100.2959165 (ACM, 2016).
14. 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, WSDM 2017, Cambridge, United Kingdom, February 6–10* (de Rijke, M., Shokouhi, M., Tomkins, A. & Zhang, M. eds.), 425–434. https://doi.org/10.1145/3018661.3018665 (ACM, 2017).
15. 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, RecSys 2017, Como, Italy, August 27-31* (Cremonesi, P., Ricci, F., Berkovsky, S. & Tuzhilin, A. eds.), 297–305. https://doi.org/10.1145/3109859.3109890 (ACM, 2017).
16. 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 on World Wide Web, WWW 2018, Lyon, France, April 23–27* (Champin, P., Gandon, F., Lalmas, M. & Ipeirotis, P. G. eds.), 1583–1592. https://doi.org/10.1145/3178876.3186070 (ACM, 2018).
17. 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, KDD 2019, Anchorage, AK, USA, August 4–8* (Teredesai, A. et al. eds.), 344–352. https://doi.org/10.1145/3292500.3330906 (ACM, 2019).
18. Pennington, J., Socher, R. & Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (Moschitti, A., Pang, B. & Daelemans, W. eds.), 1532–1543.https://doi.org/10.3115/v1/d14-1162 (ACL, 2014).
19. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings* (Bengio, Y. & LeCun, Y. eds.) (2013).
20. Devlin, J., Chang, M., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. (2018). arXiv:1810.04805 [CoRR].
21. Zhang, K. et al. SIFN: A sentiment-aware interactive fusion network for review-based item recommendation. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1–5* (Demartini, G., Zuccon, G., Culpepper, J. S., Huang, Z. & Tong, H. eds.), 3627–3631. https://doi.org/10.1145/3459637.3482181 (ACM, 2021).
22. Qiu, Z., Wu, X., Gao, J. & Fan, W. U-bert: Pre-training user representations for improved recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence* **35**, 4320–4327 (2021).
23. Liu, Y. et al. Roberta: A robustly optimized BERT pretraining approach. (2019). arXiv:1907.11692 [CoRR].
24. Xu, J., Zheng, X. & Ding, W. Personalized recommendation based on reviews and ratings alleviating the sparsity problem of collaborative filtering. In *Ninth IEEE International Conference on e-Business Engineering, ICEBE 2012, Hangzhou, China, September 9–11*, 9–16. https://doi.org/10.1109/ICEBE.2012.12 (IEEE Computer Society, 2012).
25. Huang, J., Rogers, S. & Joo, E. Improving restaurants by extracting subtopics from Yelp reviews. iConference 2014 (Social Media Expo) (2014).
26. 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, July 27–31, 2014, Québec City, Québec, Canada* (Brodley, C. E. & Stone, P. eds.), 2–8 (AAAI Press, 2014).
27. Ganu, G., Kakodkar, Y. & Marian, A. Improving the quality of predictions using textual information in online user reviews. *Inf. Syst.* **38**, 1–15. https://doi.org/10.1016/j.is.2012.03.001 (2013).
28. 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, KDD 2018, London, UK, August 19–23* (Guo, Y. & Farooq, F. eds.), 2309–2318. https://doi.org/10.1145/3219819.3220086 (ACM, 2018).
29. Chen, X., Zhang, Y. & Qin, Z. Dynamic explainable recommendation based on neural attentive models. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27–February 1*, 53–60. https://doi.org/10.1609/aaai.v33i01.330153 (AAAI Press, 2019).
30. Chin, J. Y., Zhao, K., Joty, S. R. & Cong, G. ANR: Aspect-based neural recommender. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22–26* (Cuzzocrea, A. *et al.* eds.), 147–156. https://doi.org/10.1145/3269206.3271810 (ACM, 2018).
31. He, X. et al. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3–7* (Barrett, R., Cummings, R., Agichtein, E. & Gabrilovich, E. eds.), 173–182. https://doi.org/10.1145/3038912.3052569 (ACM, 2017).
32. Rendle, S. Factorization machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14–17 December* (Webb, G. I., Liu, B., Zhang, C., Gunopulos, D. & Wu, X., eds.), 995–1000. https://doi.org/10.1109/ICDM.2010.127 (IEEE Computer Society, 2010).
33. Zhang, W., Du, T. & Wang, J. Deep learning over multi-field categorical data—a case study on user response prediction. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings, vol. 9626 of Lecture Notes in Computer Science* (Ferro, N. et al. eds.), 45–57. https://doi.org/10.1007/978-3-319-30671-1_4 (Springer, 2016).

34. Xiao, J. et al. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25* (Sierra, C. ed.), 3119–3125. https://doi.org/10.24963/ijcai.2017/435 (ijcai.org, 2017).
35. Cao, B., Li, C., Song, Y. & Fan, X. Network intrusion detection technology based on convolutional neural network and bigru. *Comput. Intell. Neurosci.* **20**, 22 (2022).
36. Teng, F. *et al.* A gru-based method for predicting intention of aerial targets. *Comput. Intell. Neurosci* **2021**, 6082242:1-6082242:13. https://doi.org/10.1155/2021/6082242 (2021).
37. Al-Sabahi, K., Zhang, Z. & Nadher, M. A hierarchical structured self-attentive model for extractive document summarization (HSSAS). *IEEE Access* **6**, 24205–24212. https://doi.org/10.1109/ACCESS.2018.2829199 (2018).
38. Lin, Z. *et al.* A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings* (OpenReview.net, 2017).
39. Koren, Y., Bell, R. M. & Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **42**, 30–37. https://doi.org/10.1109/MC.2009.263 (2009).
40. Catherine, R. & Cohen, W. W. Transnets: Learning to transform for recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27–31* (Cremonesi, P., Ricci, F., Berkovsky, S. & Tuzhilin, A., eds.), 288–296. https://doi.org/10.1145/3109859.3109878 (ACM, 2017).
41. Feng, X. & Zeng, Y. Neural collaborative embedding from reviews for recommendation. *IEEE Access* **7**, 103263–103274. https://doi.org/10.1109/ACCESS.2019.2931357 (2019).
42. Zhao, C., Li, C., Xiao, R., Deng, H. & Sun, A. CATN: Cross-domain recommendation for cold-start users via aspect transfer network. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30* (Huang, J. X. et al., eds.), 229–238. https://doi.org/10.1145/3397271.3401169 (ACM, 2020).
43. Dezfouli, P. A. B., Momtazi, S. & Dehghan, M. Deep neural review text interaction for recommendation systems. *Appl. Soft Comput.* **100**, 106985. https://doi.org/10.1016/j.asoc.2020.106985 (2021).
44. He, X. et al. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30* (Huang, J. X. *et al.*, eds.), 639–648. https://doi.org/10.1145/3397271.3401063 (ACM, 2020).
45. Gao, Q. & Ma, P. Graph neural network and context-aware based user behavior prediction and recommendation system research. *Comput. Intell. Neurosci.* **8812370:1–8812370:14**, 2020. https://doi.org/10.1155/2020/8812370 (2020).

Acknowledgements

The work described in this paper is partially supported by the National Natural Science Foundation of China (Nos. 61402150, 61806074), Key Scientific Research Project Plan of Colleges and Universities in Henan Province (No. 23A520016), and Science and Technology Research Project in Henan Province (No. 232102211029).

Author contributions

Z.L.: writing-review and editing, supervision, funding acquisition; D.J.: methodology, software, writing-original draft, writing-review and editing; K.Y.: writing-review and editing. All authors have read and agreed to the final version of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to K.Y.