IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 8, 2019, accepted July 18, 2019, date of publication July 31, 2019, date of current version August 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931357

Neural Collaborative Embedding From Reviews for Recommendation

XINGJIE FENG AND YUNZE ZENG
College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China

Corresponding author: Yunze Zeng (540117253@qq.com)

This work was supported in part by the National Natural Science Foundation of China and China Civil Aviation Administration Joint Fund Project under Grant U1233113 and Grant U1633110, and in part by the National Natural Science Youth Foundation of China under Grant 61301245 and Grant 61201414.

ABSTRACT This paper mainly studies the personalized rating prediction task based on review texts for the recommendation. Recently, most of the related researches use convolutional neural networks to capture local context information, but it loses word frequency and global context information. In addition, they simply equate the user (item) embedding to review embedding, which brings some irrelevant information of the review text into user preference or item property. Moreover, they only consider the low-order interactions, which remain the fine-grained user-item interactions to be explored. To solve these problems, we propose a novel method neural collaborative embedding model (NCEM). We first adopt pre-trained BERT model, which has been proven to improve most of the downstream NLP tasks, to simultaneously capture the global context and word frequency information. In addition, a self-attention mechanism is introduced to learn the contribution of each review. Next, we develop a neural form of standard factorization machine, which can model first-order and second-order user-item interactions by stacking multiple layers. The extensive experiments on four public datasets showed that NCEM consistently outperforms the state-of-the-art recommendation approaches. Furthermore, the recommendation interpretability can be improved by showing users the high score reviews accompanied recommended item.

INDEX TERMS Recommender system, collaborative filtering, deep learning, factorization machine.

I. INTRODUCTION

With the growing popularity of the Internet and smart mobile devices, people's online time is rising. In order to improve office efficiency and consumption experience, the company provides a variety of products and services to meet the different needs of users, but it is also more difficult for users to quickly make satisfactory choices from a large amount of information. Due to it can help different users to find out the products they are interested in through their historical behavior, the recommender system has become an extremely important part of online activities, such as online shopping, reading articles, and watching movies. To provide a personalized recommendation service, how to accurately predict the user's rating of the item is a key issue that the recommender system needs to solve.

In the field of recommender systems, Collaborative Filtering (CF) is the most outstanding type of model. It mainly models user preferences and item characteristics based on historical data (rating, click, expenditure) [1]–[4]. Among the various CF methods, the most successful is Matrix Factorization (MF) [1], [5]. It maps users and items into a shared latent space for representing them with latent embeddings. Then, the user's rating of the item is defined as the inner product of their corresponding embeddings. However, the MF-based algorithm only utilizes the rating data as the input, and the further improvement of the rating prediction accuracy is greatly limited by problems of data sparsity (a user's rated item only accounts for a very small part of the total number of items) and cold start (new users or items are often not rated). As more and more data on the Internet can be acquired, multi-source heterogeneous data including images, texts, and tags contain rich user behavior and personalized demand information. The hybrid recommendation method combining these side information has been paid more and more attention for it can alleviate the problem of data sparsity and cold start.

Among the various side information, the most popular one is the review text data. The main reasons are as follows: First, the review text intuitively describes the reason why the item obtains such rating, which can help the model understand the

The associate editor coordinating the review of this manuscript and approving it for publication was Longzhi Yang.

item property. Second, different users have different evaluation points and narratives for the same item, so the review text can also make the model more effective in learning user preferences. Third, due to the development of e-commerce companies (Amazon), video sharing sites (Youtube) and community sites (Yelp), a large number of review texts can be easily collected. As a consequence, performing rating prediction based on review text in collaborative filtering has become a research hotspot.

The earliest personalized rating prediction work based on review text was proposed by McAuley and Leskovec in 2013 [6]. From 2013 to 2015, many methods were inspired by their work [7]–[9]. However, most of the above methods use the topic model LDA (Latent Dirichlet allocation) to process the review text, ignoring the context information which is seriously degrading the embedding quality. In order to solve this problem, most of the work in recent years has used convolutional neural networks to process review texts [10]–[13]. Although the accuracy of the rating prediction in these methods is getting higher and higher, they suffer from at least one of the following three limitations: First (**limitation 1**), for CNN, when the maximum value of the feature map generated by a convolution kernel occurs multiple times, max-pooling will lose the word frequency information. In addition, the context information captured by CNN is only limited to the information within the width of the word window (kernel size), and the global context information is extremely lost. Second (**limitation 2**), it is very inappropriate to directly equate the embedding of the review text to that of users and items. Intuitively, it is relatively difficult to fully represent users (items) from a limited length of review text. On the other hand, direct equivalence may bring some irrelevant information in review text for some dimension of their embeddings. Third (**limitation 3**), the final rating prediction stage still takes the simple linear transformation like inner product operation [10], [13] or factorization machines [11], [12], there is no guarantee that the users' (items') complex internal structure in the data will be fully learned [14].

To solve the above three limitations, we propose a novel neural collaborative embedding model (named NCEM) for personalized rating prediction tasks. Overall, the model consists of three modules (see Figure 3). In module 1, the embedding of each review is obtained from the pre-trained BERT model [15], which can fully retain the word frequency information and also consider the forward and backward global context information. It solves the **limitation 1**. In addition, to build the better document representation of the user (item) reviews, we distinguish the contribution of each review to the user (item) modeling by an attention mechanism. In module 2, we adjust the embedding of the review text into a more sophisticated representation of the user (item) in a new fashion, which alleviates the **limitation 2**. In Module 3, we first extend the standard factorization machine [16] into a neural netwrok form. And then feed the embeddings of users and items into a nonlinear multiple neural factorization machine layers to perform rating prediction, so that they (user and item) can interact in a nonlinear deeper space to calculate more accurate ratings, which solves the **limitation 3**.

Our major contributions and advantages of NCEM can be summarized as follows:

1. This paper solves the three limitations of previous works.
2. To the best of our knowledge, in the task of rating prediction based on review texts, we are the first work incorporates the pre-trained BERT model and verifies its effectiveness also apply to recommender system.
3. Thanks to the attention mechanism, we can improve the interpretablility of the recommendation by providing the recommended item accompanying high attention score reviews.
4. Extensive experiments on four public datasets show that NCEM can not only use the review text to alleviate the impact of data sparsity, but also consistently outperform the state-of-the-art recommendation approaches.

The remainder of the paper is arranged as follows:

we present related works in Section 2. In Section 3, we describe the preliminaries of our work. In Section 4, we combine BERT with attention mechanism to process the review and propose a neural FM to predict the ratings. Section 5 detail the experimental settings and analyze the results. Finally, Section 6 concludes the paper.

II. RELATED WORK

In order to alleviate the impact of data sparsity on rating prediction, combination with review texts has become a hot area of research in recent years. According to the technology of processing review texts, recent methods can be divided into two categories: topic-based methods and deep learning methods.

A. TOPIC-BASED METHODS

Literature [6] proposes an outstanding model HFT by combining the latent factors of the rating data with the topic distribution of the review text. Subsequently, similar studies have emerged. The literature [8] proposes a probabilistic model based on collaborative filtering and topic models for rating prediction, but this method does not use rating data when modeling review texts. In order to further apply the rating data, the RMR model proposed in [17] combines the topic model with a mixed Gaussian model based on ratings to further improve the recommendation performance. However, the literature [7] believes that LDA can only mine the topic distribution of word level, which can not accurately express the distribution of composite topics, so the TopicMF model is proposed. TopicMF obtains the latent topic of each review through non-negative matrix factorization, and maps the latent factors of users and items. Finally, the topic distribution reflects user preferences and item characteristics. Reference [9] pointed out that the model, which uses the topic model to extract the latent factor of reviews in combination with the rating matrix, is classified as the topic matrix factorization model. The literature [18] fuses the topic

of the review with the rating matrix factorization model by manually creating partial keywords. In [19], a hybrid model based on implicit probability and random walk is proposed. The probability model is responsible for mining potential preferences of users and the hidden features of items. Random walks can construct global potential associations to predict users' preference for unrated items.

The above related works are to mine the latent feature distribution of the user or item through the topic model, and then combine matrix factorization to perform the rating prediction. However, they have at least the following limitations: First, the topic model based on the assumption of 'Bag of Words', which cannot capture word order information. In fact, local context is extremely important information in sentiment analysis [20]. Second, the final rating prediction still uses the inner product operation, which is limited to linear transformation, and it is not as effective as stacking several nonlinear fully connected layers [14].

Recently, the deep learning model CNN has been proved to be more advantageous than the topic model [11], [13], [21]. It can effectively retain the local context information, and can also combine different attention mechanisms to improve the extraction quality of text information. Moreover, deep neural networks are capable of nonlinear transformations, theoretically infinitely approximating any continuous function [22]. Therefore, in recent works, CNN is mainly used to process review texts.

B. DEEP LEARNING METHODS

To capture the local context information in reviews, CNN has been mainly used in recent years to replace the topic model. At the same time, with the goal of improving the modeling ability, most of the work is to replace the the inner product of matrix factorization with the MLP (multi layer perceptron) to introduce the nonlinear transformation. In 2016, Kim *et al*. proposed ConvMF (Convolution Matrix Factorization) to generate deeper latent embedding from article description texts using convolutional neural networks [10]. This method takes into account the word order information of words, thus producing more accurate hidden space of items. However, ConvMF only considers the textual information of items and ignores the user's textual information. To overcome above limitations, the works in future are to divide review text into two sets: user review set and item review set. The CNN is then used to learn the latent factors of users and items from both sets, respectively. A typical model that inspired a lot of work is DeepCoNN (Deep Cooperative Neural Network) [11]. DeepCoNN consists of two parallel CNN networks, one CNN is responsible for learning user embeddings from user reviews, and the other CNN is responsible for learning item embeddings from item reviews. Finally, a shared layer is used to connect the two networks, and Factorization Machines (FM) is introduced to capture the interaction between the user and item. Then in August of the same year, Catherine *et al*. extended DeepCoNN to further improved the prediction accuracy by adding a layer to reconstruct the embedding of the target user-item pair reviews [12]. Particularly, these two works illustrated that utilizing FM instead of inner product may be a good idea.

Since the attention mechanism can find out the most informative part of the data, it is an inevitable trend to use the attention mechanism to process the review text for recommender systems. Proposed by Seo *et al*., D-Attn (CNNs with dual attention model) scores each word in the review text by combining local and global attention to catch the most relevant word for the rating, which simultaneously improves the accuracy of the prediction and the interpretability of the recommendation result [24]. However, they still adopted the inner product operation in prediction stage. Later, Wang *et al*. pointed out that the previous CNN-based methods may ignore word frequency information. To solve this problem, Wang *et al*. proposed WCN [13] that can combine the topic model with CNN. The topic model can capture the word frequency information to make up for the shortage of CNN. In 2018, Chen *et al*. proposed the NARRE (Neural Attentional Rating Regression with Review-level Explanations) model [21]. NARRE scores each review through a attention mechanism and combines attention scores with user latent factors to improve the quality of embedding. A similar work to NARRE is MPCN [25] proposed by Tay *et al*., who point out that a user's historical review is not always related to the target item. Therefore, they use a new type of dual attention mechanism to identify more relevant reviews.

Overall, the most above works are to use CNN to process the review text to capture the local context information, failing to consider the global context information. As such, those existing modeling paradigms will eventually hit a dead-end.

At present, any other famous technique for capturing global context in text data is recurrent neural networks (RNN), which have been adopted in review-based recommender systems. Wu *et al*. combined CNN with a bidirectional GRU network to propose DRMF [23], which can supplement the global context information lost by CNN. More straightforwardly, there are several models such as TARMF [26], GRU-MTL [27], BoWLF [28], directly use RNN to encode the word embedding matrix of reviews. However, they directly equate the representation of the review text with the embeddings of the user and item, which will bring the irrelevant information and it is unfavorable for the rating prediction. Moreover, perhaps because of the limitations of the times, they are deeply affected by matrix factorization. The final prediction rating of the method still takes the linear transformation such as inner product, and remains the further improvement to be explored.

To sum up, although there are shortcomings in recent works, they still have something useful experience. First, using FM instead of inner product may be a good idea, because FM is capable of capturing interaction of first and second order. Second, differentiating the contribution of each review, the quality of the user (item) embedding will be improved. Considering the advantages and disadvantages of related works, we propose NCEM whose structure is

completely different from previous models and without any CNN and RNN modules.

III. PRELIMINARIES

In this section, we formally define the problem and notations. Intuitions of this work is described in detail. Finally, a brief description of the BERT mode is showed.

A. PROBLEM FORMULATION

Given a training set D consists of N samples, each samples (u, i, r_{ui}, w_{ui}) denotes a review w_{ui} written by user u for item i with rating r_{ui}. The task for this work is to build a model that can predict a rating $\hat{r}_{ui}$ depending on the user review set R_u (the set of all reviews written by user u) and the item review set R_i (the set of all reviews written by item i), meanwhile minimize the error between $\hat{r}_{ui}$ and r_{ui}.

B. INTUITIONS

As mentioned above, although using CNN to process the review text for personalized rating prediction has achieved good results, there are three limitations in the past work. Therefore, this section details the intuitions for this work around these limitations.

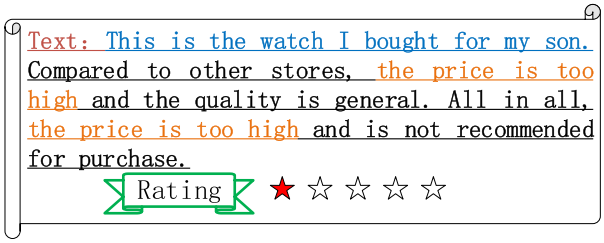


FIGURE 1. An example of review text. Generally, when users purchase an item, they write a review describe why they make the purchase decision and rate the item.

First, context and word frequency information are key factors in accurately representing the user preferences and item properties in review text. Suppose we consider the example in Figure 1. The polarity of the word "high" in the review is dependent on the aspect it describes(context information). For example, "high price" reflects negative sentiment while "high quality" expresses positive sentiment. In related works [10]–[13], the CNN is used to scan the text in a sliding window to capture local context information. However, when the max feature value appears more than once in the feature map, frequency information is obviously lost by max-pooling. In fact, frequency information is also significant in review text. The "higher price" in Fig 1 appears twice is more convincing than once. But only one "high price" is retained after max-pooling. Another important thing is that CNN is unable to learn global context information. For example, CNN can not understand this user purchased a watch for her son, because the word "purchase" and "son" are far apart in the review. In general, if the context information and count information are not considered, the review will not be accurately analyzed, which is very unfavorable for the rating prediction. Inspired by this intuitive example, we directly replace the CNN of past works with the BERT model, which can get the representation of the review text considering the global context information and word frequency information.

Second, it is not appropriate to directly equate the embedding of the review text to user (item). For user u, item i and review w_{ui}, we define their embedding respectively as $p_u \in \Re^\kappa$, $q_i \in \Re^\kappa$ and $d_{ui} \in \Re^\kappa$, where κ denotes the dimension of embedding (the number of latent factors). In previous works [11]–[13], they perform direct equivalence between user, item and review embeddings, e.g. $d_{ui} \approx p_u$ or $d_{ui} \approx q_i$. However, this way has the following two problems: First, although it can encode all the information into the embedding of user or item, it is unable to ensure all information in the review contributes to the rating, as shown in the blue font portion in Figure 1. Second, because the length of the review is limited, it cannot fully reveal the characteristics of users (items) [29], which is needed to adopt a new way to mine the characteristics. Therefore, we devise a new projection to adjust the representation of the review, so that it can better reflect the characteristics of the user (item), which can alleviate the above two problems to some extent.

Third, the literature [14] pointed out that the rating prediction can also be regarded as a task of mining interaction between user and item based on history data. Most previous works utilize the inner product or FM to predict rating [10]–[13]. It is only a linear transformation that the complex structure of user and item in the latent space is not fully constructed. In addition, neural networks have been proven to be capable of approximating any continuous function [22], and more recently deep neural networks (DNNs) have been widely studied in several realms, ranging from computer vision, speech recognition, and text processing [30]–[33]. Inspired by this, we extend the standard factorization machine into a neural network form to learn the fine-grained user-item interactions.

C. PRE-TRAINED BERT MODEL

BERT is a method of pre-training language representations, meaning that it is a general-purpose "language understanding" model trained on a large text corpus (like Wikipedia), and then we can use it for downstream NLP tasks that we care about (like question answering, text classification). The overall architecture of BERT is showed in Figure 2.

Previous models (ConvMF [10], DeepCoNN [11], WCN [13]) are all based on the Context-free models such as word2vec or GloVe generate a single "word embedding" representation for each word in the vocabulary, so bank would have the same representation in bank deposit and river bank. BERT instead generate a representation of each word that is based on the other words in the sentence.

Thanks to its deeply bidirectional system, according to the report [15], BERT has been proven to improve most of the downstream NLP tasks, but not include our review based rating prediction task. Therefore, this work attempts to

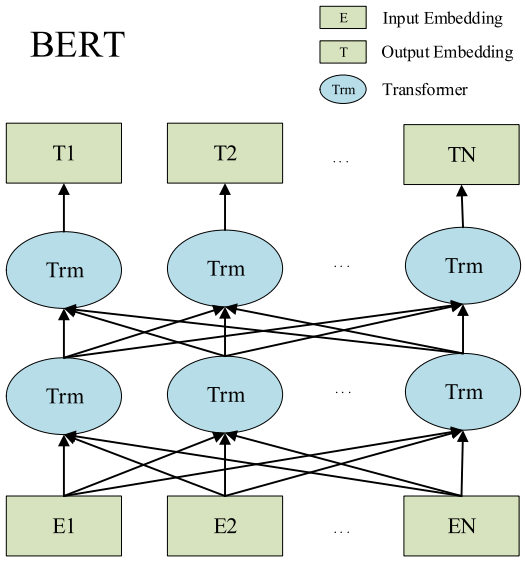


FIGURE 2. The architecture of BERT, which is the state of the art pre-trained NLP model.

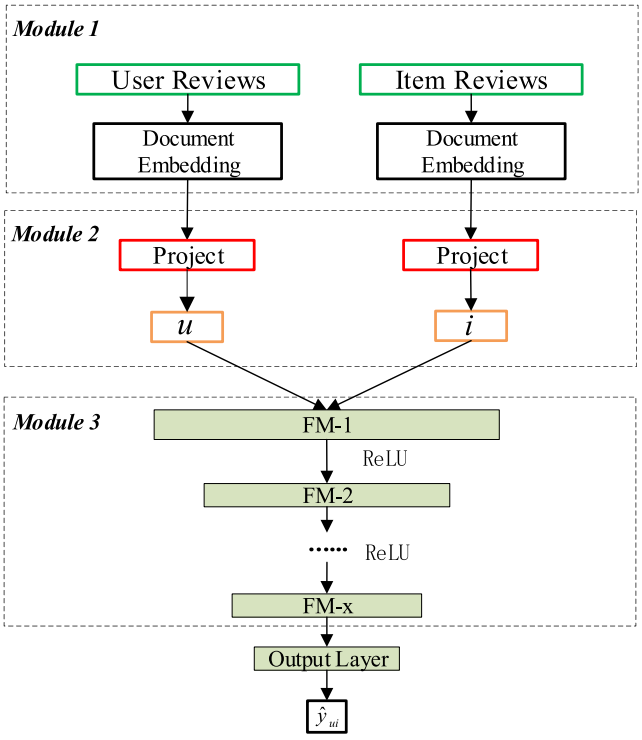


FIGURE 3. The architecture of NCEM.

combine BERT model with collaborative filtering recommender system, and further explore the power of reviews to alleviate the impact of data sparsity.

IV. THE PROPOSED MODEL

The detailed structure of the model NCEM in this paper is shown in Figure 3, which is divided into three modules. Module 1 and Module 2 contain two parallel networks: the user network is responsible for learning the embedding $p_u \in \Re^\kappa$ of the user u from the user review set R_u, while the item network models the item embedding $q_i \in \Re^\kappa$ of item i in a same fashion. Module 3 feeds p_u and q_i into the deep neural network (neural factorization machine) to predict the rating $\hat{y}_{ui}$.

A. MODULE 1

Given a review set of user u, i.e., a list of the user's historical reviews $\{R_{u1}, R_{u2} \dots R_{uc}\}$, where c represents the maximum number of reviews for each user's review set. The $\{R_{u1}, R_{u2} \dots R_{uc}\}$ is sent to the BERT, where the reviews are processed one by one to obtain a embedding list $d_u = \{d_{u1}, d_{u2} \dots d_{uc}\} \in \Re^{c\times k}$(if the number of historical reviews is less than c, a number of zero vectors are added, so that the length of the list is c).

Intuitively, not each review can reflect the user's preference, so this module utilizes the attention mechanism [34] to measure the contribution of each review, and get the attention vector $a \in \Re^{1\times c}$:

$$a = softmax(w_1 \times tanh(w_2 \times d_u^T)), \tag{1}$$

where $w_1 \in \Re^{1\times t}$,$w_2 \in \Re^{t\times k}$, and t are hyperparameters that can be set to any dimension. $softmax()$ is used to normalize attention weights. Next, according to the attention vector a, each of the reviews is weighted and summed, and then obtain the output $text_u \in \Re^{1\times\kappa}$ which is the user preference vector embodied in the user review set:

$$text_u = ad_u, \tag{2}$$

Similarly, the embedding $text_i \in \Re^{1\times\kappa}$ of item i can be learned from the item network.

B. MODULE 2

Intuitively, $text_u$ is a text embedding composed of a user's all historical reviews, which contains some irrelevant information that cannot reflect the user's preference. Therefore, here we follows the idea of [35]: projecting the user's preference embedding from the text embedding $text_u$. Specifically, the occurrence probability of user u based on $text_u$ is :

$$p(u|text_u) = \frac{\exp(p_u text_u^T)}{\sum_{u'=1}^{|U|} \exp(p_{u'} text_u^T)}, \tag{3}$$

where $|U|$ is the total number of users. For user network, the loss function is to minimize the log-probability below:

$$\ell_1 = -\sum_{u\in U} \log p(u|text_u), \tag{4}$$

Similarly, in the item network, the feature embedding $q_i \in \Re^{1\times\kappa}$ of the item i can be calculated, and its loss function is denoted as ℓ_2.

C. MODULE 3

In module 2, we have obtained embeddings p_u and q_i of user u and item i. Most of the previous researches are based on matrix factorization, which directly equates the predicted rating $\hat{y}_{ui}$ to $p_u^T q_i$. However, this simple strategy does not fully explore the complex internal structure of the data. In DeepCoNN [11]and TransNet [12], they all concatenate p_u and q_i into a single vector $Z \in \Re^{2k}$, and then feed it into factorization machine(FM), which can regress the predictive rating $\hat{y}_{ui}$ from $Z = (z_1, z_2, \ldots z_{2k})$. Compared to the inner product, the advantage of FM is that it can capture the interaction between any two dimensions in Z. The formula of standard FM is as follows:

$$\hat{y}_{ui} = b + WZ + \sum_{i=1}^{|\hat{z}|} \sum_{j=i+1}^{|\hat{z}|} \langle v_i, v_j \rangle z_i z_j, \tag{5}$$

where $v_i \in \Re^{k'}$ is the parameter vector corresponding to $z_i \in \Re^1$, $W \in \Re^{2k}$.

However, standard FM is also a kind of linear transformation. At present, deep neural networks have been shown to be able to approximate any continuous function indefinitely. Inspired by this idea, we extend standard FM into a neural network form by stacking multiple FM to learn simultaneously. For the first order term, we directly modify its parameter shape as $W \in \Re^{k' \times 2k}$. The first order term $FM\_1(z) \in \Re^{k'}$ in neural FM is defined as:

$$FM\_1(z) = WZ, \tag{6}$$

For the second order term in standard FM, we first need to transform it into an equation as:

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle z_i z_j = sum\left(\frac{1}{2} \times \left[(\sum_{i=1}^{n} v_i z_i \otimes \sum_{i=1}^{n} v_i z_i) - \sum_{i=1}^{n} (v_i z_i \otimes v_i z_i) \right]\right), \tag{7}$$

where $\otimes$ indicates element wise product of vectors, $sum()$ denotes to sum all elements in a vector to get a real number.

As we have seen, before the $sum()$ is performed, all the information of the second-order interaction term is already included. Thus, we simply cancel the $sum()$ and we can get the neural second order term:

$$FM\_2(z) = \frac{1}{2} \times \left[(\sum_{i=1}^{n} v_i z_i \otimes \sum_{i=1}^{n} v_i z_i) - \sum_{i=1}^{n} (v_i z_i \otimes v_i z_i) \right], \tag{8}$$

where $FM\_2(z) \in \Re^{k'}$. Next, by concatenating the neural first and second order term, we can get the neural FM as below:

$$FM(z) = FM\_1(z) \oplus FM\_2(z) \tag{9}$$

where $\oplus$ denotes concatenation of two vectors. The purpose of module 3 is to stack multiple neural FM to build a deep neural network , its precise formulation is:

$$\begin{aligned} z &= p_u \oplus q_i, \\ f_1 &= a_1(FM_1(z)), \\ &\cdots\cdots\cdots \\ f_x &= a_x(FM_x(f_{x-1})), \\ \hat{y}_{ui} &= h f_x + b_u + b_i + \mu, \end{aligned} \tag{10}$$

where a_x denotes the activation function ReLU in the x layer, $f_x \in \Re^{2k'}$ is the output of the x layer. $h \in \Re^{1 \times 2k'}$,$b_u \in \Re^1$, $b_i \in \Re^1$, $\mu \in \Re^1$ denote the weight matrix, user bias, item bias, and global bias for the output layer, respectively. Note that each FM layer has the same architecture.

D. LEARNING

For module 3, the objective function that is widely used in recommender system realm for rating prediction [36]–[38]:

$$\ell_3 = \sum_{u,i \in D} (\hat{y}_{ui} - y_{ui})^2, \tag{11}$$

Since both the user network and the item network have their own objective functions, the final optimization goal of NCEM is the sum of the above three objective functions:

$$\ell = \lambda_1 \ell_1 + \lambda_2 \ell_2 + \ell_3, \tag{12}$$

where ℓ_1 and ℓ_2 can be regarded as a kind of regular term and we set the weight of of $\lambda_i \in [0, 1]$.

In order to minimize the objective function, we use Adam(Adaptive Moment Estimation) [39] as the optimizer. Its main advantage is that it can adjust the appropriate learning rate in the training process, which can eases the pain of the manual selection for a proper learning rate and leads to faster convergence than the vanilla SGD.

V. EXPERIMENTS

In this section, we conduct experiments with the aim of answering the following research questions:

RQ1 Can NCEM proposed in this paper be better than other collaborative filtering methods in rating prediction tasks?
RQ2 Can NCEM capture simultaneously global context and frequency information to improve the accuracy of rating prediction?
RQ3 In module 2, whether projection of user and item embeddings can alleviate the limitation 2?
RQ4 Does deep neural FM help improve the quality of user (item) embedding?
RQ5 How NCEM can help improve the recommendation's interpretablility?

A. EXPERIMENTAL SETTINGS

1) DATASETS AND EVALUATION METRIC

The experiments in this paper were conducted using the public dataset Toys_and_Games, Instant_Video, and

Digital_Music, which are the subset of the Amazon Product Reviews.[1] These datasets have different themes and sizes, with Toys_and_Games being the largest dataset (with a total of more than 160,000 reviews) and Instant_Video being the smallest one (a total of 37,000 reviews). Another dataset is from Yelp Challenge 2017,[2] which is an online review platform for businesses such as restaurants, bars, spas, etc.

In these datasets, for each sample, there are four features used in this paper: user ID, item ID, user's rating on the item (1∼5 points), and user's review text on the item.

TABLE 1. The statistics of four datasets.

	Users	Items	Samples	Sparsity
Toys_and_Games	19,412	11,924	167,597	99.9%
Digital_Music	5,541	3,568	64,706	99.6%
Instant_Video	5,130	1,685	37,126	99.5%
Yelp-2017	199,445	119,441	3,072,129	99.9%
Average	57,382	34,154	835,389	99.7%

As seen from Table 1, although the number of users and items in each dataset is huge, the sparsity of each dataset is about 99%, which seriously affects the performance of the traditional method based solely on rating data.

In the experiments, our evaluation metric is the mean square error (MSE), which is widely used in the works [6], [11], [12]:

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(\hat{y}_{ui} - y_{ui})^2, \tag{13}$$

where N is the number of samples. MSE is sensitive for outliers due to it is the quadratic difference between the predicted values with the ground-truth.

2) BASELINES

To answer RQ1, verify weather NCEM's rating prediction performance is better than other methods. We choose HFT [6], TopicMF [7], ConvMF [10], DeepCoNN [11], WCN [13] as baselines. The specific differences between the various methods are shown in Table 2, where ''Nonlinear Interaction'' means whether the predicted rating is obtained by the neural FM. The ''\'' of ''Nonlinear Interaction'' denotes predicted ratings calculated by inner product operation or standard FM.

These methods can be roughly divided into two categories: the first is the method of processing the review text using the topic model like HFT and TopicMF; the second is the kind of processing text by the deep learning model CNN like ConvMF, DeepCoNN, WCN.

[1] http://jmcauley.ucsd.edu/data/amazon
[2] https://www.yelp.com/dataset_challenge

TABLE 2. Comparison of the baselines.

	HFT	TopicMF	ConvMF	DeepCoNN	WCN	NCEM
Deep Learning	\	\	√	√	√	√
Word Frequency	√	√	\	\	√	√
Local Context	\	\	√	√	√	√
Global Context	\	\	\	\	\	√
Nonlinear Interaction	\	\	\	\	\	√

- **HFT**: This is the first model based on the review text to predict the rating, which provides inspiration for many later works.
- **TopicMF**: By expanding HFT, TopicMF combines user review and item review sets based on non-negative matrix factorization.
- **ConvMF**: ConvMF combines CNN and probability matrix factorization, which is a model that can use ratings and item review set as input.
- **DeepCoNN**: This is a model that utilizes only review data. It divides the review set into user sets and item sets as input and achieves good results.
- **WCN**: Based on DeepCoNN, WCN adding the LDA latent topic of the review as input to alleviate the problem that CNN may lose word frequency information.

3) EXPERIMENTS DETAILS

We use Tensorflow to implement the proposed model and accelerate the training precess by GPU (GTX 1080Ti). We randomly divided the experimental dataset into training set (80%), validation set (10%) and test set (10%). The hyper-parameters were selected on the validation set, and finally the test set was used for performance evaluation.

For the topic model based approaches HFT and TopicMF, the number of latent factors is equal to that of topics. For the deep learning models ConvMF, DeepCoNN, WCN, and NCEM, the number of latent factors is the dimension of the embedding. According to the report in [10], [11], [13], in order to ensure the quality of review embedding, their convolution kernels are all set to 100 and the number of convolution layers is 1. Due to the large number of parameters of the deep learning model, we carefully tested the batch size from [128, 256, 512, 1024] and looked for the optimal value of the learning rate from [0.0001, 0.0005, 0.001, 0.005]. It is worth mentioning that the more latent factors, the more likely it is to cause over-fitting and affect the performance of the model. For NCEM, the version of the pre-trained BERT is ''uncased_L-12_H-768_A-12'', the number of layers for neural FM is set to 3, and its parameter k' is set to 6, which we empirically found to be a reasonable setting. The maximum

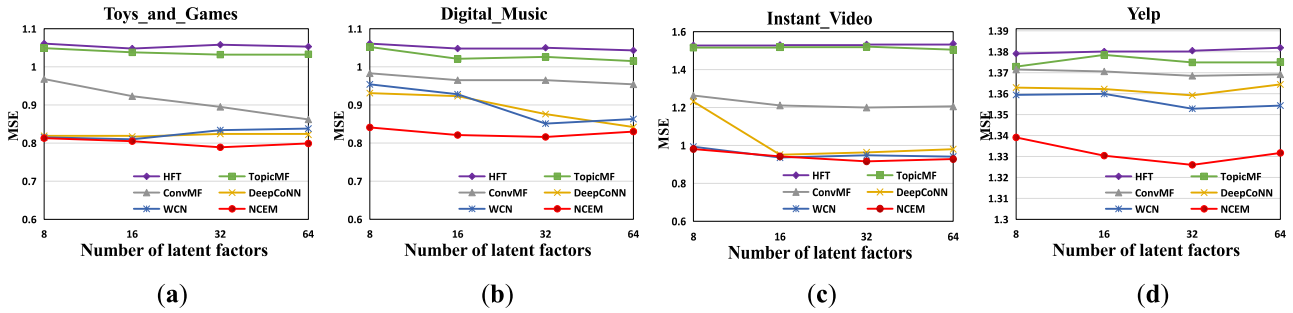


FIGURE 4. Impact of the number of latent factors. On the whole of four datasets, NCEM performs better than other models.

TABLE 3. Results of different algorithms on four datasets. ∗ indicates the best result of the baseline. The results of NCEM and the percentage of improvement has been marked in bold.

	Toys_and _Games	Digital_Music	Instant_Video	Yelp
HFT	1.048	1.043	1.527	1.379
TopicMF	1.032	1.015	1.516	1.372
ConvMF	0.862	0.954	1.199	1.368
DeepCoNN	0.817	0.842*	0.951	1.359
WCN	0.810*	0.863	0.936*	1.352*
NCEM (Improve)	**0.789 (2.59%)**	**0.816 (3.08%)**	**0.912 (2.56%)**	**1.326 (1.92%)**

number of reviews for each user and item is set to 10 and 20, respectively.

B. PERFORMANCE EVALUATION

The performance of NCEM and other baselines presented in this paper are shown in Table 3 and Figure 4. We analyze the experimental results and have the following conclusions:

First, the deep learning models (ConvMF, DeepCoNN, WCN, and NCEM) perform better than traditional models (HFT, TopicMF). We believe that there are three reasons: First, the traditional way of processing text data is mostly based on the topic model LDA which has been proved that it will ignore context information and is not the best text processing technology; Second, the limitation of the traditional model is that it only learns linear features, while the deep learning approach can model users and items in a nonlinear way. Third, some deep learning techniques such as dropout [40] and batch normalization [41] can effectively suppress over-fitting and further explore the potential of the model.

Second, ConvMF and DeepCoNN only use CNN to extract features from the text where the word frequency information will be lost by max-pooling and the improvement of prediction accuracy also be limited. In order to solve this problem, WCN combines the extraction features of CNN with the topic factors of LDA to make up for the shortcomings of CNN and achieve better results than ConvMF and DeepCoNN. However, NCEM adopts a structure that is completely different from WCN to deal with this problem. In module 1, the pre-trained BERT model of NCEM can capture global context and word frequency information at the same time, in which the accurate understanding of the review information is guaranteed. In addition, the other methods are to combine all the reviews of the user(item) review set into a document. They treat each comment equally, and fail to identify reviews that are useless for modeling. On the contrary, NCEM measures the contribution of each review through the attention mechanism, which effectively guarantees the quality of the representations of users and items (see Section C for detailed analysis).

Third, as shown in Table 3, the average prediction error of NCEM is decreased by 2.56% to 3.08% compared with DeepCoNN and WCN. And most importantly, compared to DeepCoNN and WCN, which directly equate the review embedding to the user (item) embedding, NCEM can get better performance by alleviating the impact of some irrelevant information in review (see Section D for detailed analysis).

Fourth, as shown in Figure 4, since NCEM's final rating prediction is to use neural FM instead of inner product and standard FM, its generalization is better than other methods. The performance of NCEM is more stable, and the prediction error does not change significantly with the number of latent factors, which can simplify the tuning process (see section E for detailed analysis).

C. IMPACT OF MODULE 1 (RQ2)

Module 1 processes the review text based on the pre-trained BERT model, which can learn global context information and word frequency information at the same time. It makes up for the shortcomings of CNN. The BERT model and the attention mechanism are the two major components of Module 1. In order to study their influence on the performance of the model, we conducted a ablation study on its three variants.

- CNN: The variant uses the pre-trained word vector (GloVe) to express each word. It does not distinguish the contribution of each review, and directly concatenates all the reviews in the user review set (item review set) into a single document for processing.
- CNN+ Attention: Based on the variant CNN, each review is processed one by one, taking the attention

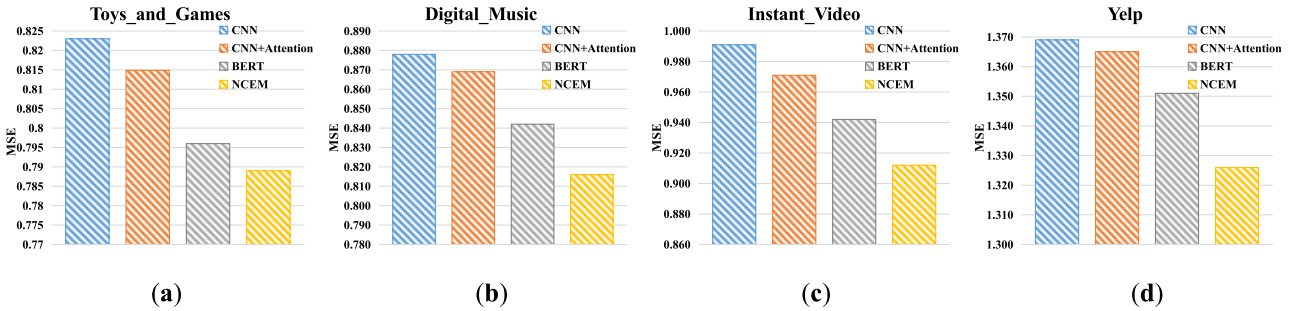


FIGURE 5. The MSE of different variants. To study the impact of module 1, we devise three variants experimented in different datasets to verify the rationality of our model.

TABLE 4. Visualization of reviews with high attention score in instant_video.

	Attention Score	Review
Item 1	0.2219	This season has some of the greatest episodes in Sunny history!
	0.0682	It was a disappointment. I gave it a few episodes to get into it, but it didn't get any better.
Item2	0.1753	This show is plain dumb and not funny. It is sad how much TV comedy has deteriorated.
	0.0281	Love having the Instant View feature on Amazon.com
User 1	0.2123	There was lots of action, I love it.
	0.0317	Got these for my son's birthday. He says they are great! They are funny so give them a try! OK!
User 2	0.2316	Love to see where men build and keep striving for better. It's nice to live in a day when things naturally progress.
	0.0998	I believe it's a very positive show, she can learn both languages, English and Spanish and also, about friendship!

mechanism of module 1 to capture the contribution of each review.

- BERT: The only difference between NCEM is that it canceled the attention mechanism.
- NCEM:Simultaneously use of BERT model and attention mechanism.

The performance of various variants is shown in Figure 5. Overall, the performance of the model using the pre-trained BERT (BERT and NCEM) is better than that of using the pre-trained word vector (CNN and CNN+ Attetnion). We think the reasons are as follows. First, the GloVe word pre-trained word vector belongs to a context-free word vector, which cannot distinguish polysemes, so the poor understanding of the review limits the power of their downstream module. Second, the convolutional neural network can only capture local context information within the word window size, and lose global context information, which is solved in the BERT model. Therfore, this experiment proves that the pre-trained BERT model is not only effective in other common NLP tasks, but also apply to review-based rating prediction tasks in recommender system.

In addition, under the same conditions (using CNN or BERT), using the attention mechanism to distinguish the contributions of each review can improve the representation quality of users(item) to a certain extent. As we can see, in Table 4, the high-weight and low-weight reviews are selected by the attention scores, which denotes the contribution of a review for modeling user preferences and item characteristic. The high score means that the item review can precisely reveal item characteristic, while the user review is able to accurately reflect user preferences. As a consequence, considering the contribution of each view can obtain a more accurate predicted rating. As for the discussion of recommendation interpretability, see Section F for details.

D. IMPACT OF MODULE 2 (RQ3)

Module 2 only adjust the embedding of item characteristics if λ_1 is 0. When λ_2 is 0, module 2 only adjust the embedding of user preferences. In particular, $\lambda_1 = \lambda_2 = 0$ indicates that module 2 used the same strategy as previous works (ConvMF, DeepCoNN, WCN): the review embedding was directly equivalent to the user and item embeddings.

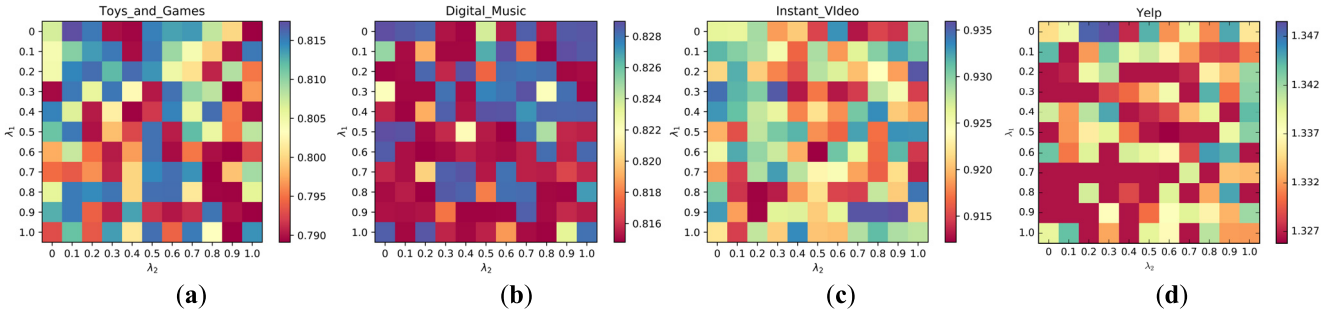


FIGURE 6. The MSE of NCEM vary with the parameter and in four datasets.

TABLE 5. Effect of the number of layers in module 3.

Datasets	Latent Factors	Inner Product	Standard FM	Layer-1	Layer-2	Layer-3
Toys_and_Games	8	0.835	0.825	0.831	0.826	**0.813**
	16	0.830	0.819	0.821	0.811	**0.805**
	32	0.818	0.807	0.812	0.795	**0.789**
	64	0.826	0.816	0.813	**0.794**	0.799
Digital_Music	8	0.869	0.865	0.862	0.852	**0.841**
	16	0.855	0.838	0.836	0.826	**0.821**
	32	0.848	0.836	0.839	0.827	**0.816**
	64	0.847	0.842	0.841	0.831	**0.822**
Instant_Video	8	0.971	0.965	0.962	**0.943**	0.951
	16	0.969	0.955	0.963	0.948	**0.942**
	32	0.952	0.943	0.946	0.925	**0.912**
	64	0.963	0.941	0.950	**0.924**	0.928
Yelp	8	1.359	1.350	1.348	**1.339**	1.342
	16	1.361	1.347	1.351	1.345	**1.330**
	32	1.357	1.348	1.342	1.335	**1.326**
	64	1.368	1.354	1.356	1.343	**1.331**

To explore the effect of λ_1 and λ_2 on different datasets, we performed different combinations from 0 to 1. The experimental results are shown in Figure 6. Form the whole view of Figure 6, the result seems unstable and uncontrollable. However, in fact, the difference between the best and worst MSE is limited within 0.02, which is desirable and stable.

Observing the distribution of the optimal results (red squares) in the graph, we can see that best results are not achieved on the four datasets when $\lambda_1 = \lambda_2 = 0$. It indicates that NCEM projects the embeddings of users and items from review to be more reasonable than direct equivalence, which can reduce the influence of irrelevant information in the reviews on user (item) modeling to a certain extent. In addition, $\lambda_1 < \lambda_2$, NCEM can achieve better results in the Toys_and_Games, indicating that the rating is mainly determined by item properties. On the contrary, the rating in Digital_Music and Yelp mostly depends on user preferences where $\lambda_1 > \lambda_2$. However, differing from the above datasets, the distribution of Instant_Video is relatively uniform showing that user preferences and item properties are equally important for ratings.

Overall, on different datasets, the effects of λ_1 and λ_2 are completely different, which is very difficult to select the appropriate value of λ_1 and λ_2. However, taking λ_1 as 0.5 and then traversing the value of λ_2 may be a advisable solution. First, in Figure 6, λ_1 and λ_2 have a total of 121 combinations. Now λ_1 is fixed at 0.5, and only 11 combinations need to be traversed to find the best parameters, which greatly simplifies the fine tuning process. Besides, $\lambda_1 = 0.5$ is a compromise strategy. It does not take the maximum or minimum value extremely, and can comprehensively consider user preferences.

E. IMPACT OF MODULE 3 (RQ4)

At present, a lot of works calculate the predicted rating by inner product or standard FM. But we use deep neural FM to mine the deeper interaction between users and items. In order to explore the effectiveness of module 3, we retained the inner product strategy and standard FM, experimented with neural FM from 1 to 3 hidden layers, respectively. The results are shown in Table 5.

In particular, the Layer-1 performs the worst, but it is still better than the inner product. This indicates that neural FM is more efficient than inner product, and it is feasible to introduce nonlinear transformation through activation function for collaborative filtering. In addition, the standard FM is comparable to the performance of Layer-1, which is in line with our intuitive understanding: the neural FM is a neural network

TABLE 6. The prediction and recall of useful review list on four datasets.

	Toys_and_Games				Digital_Music			
	Latest	Random	Length	NCEM	Latest	Random	Length	NCEM
Precision@1	0.1398	0.3316	0.2445	**0.4027**	0.2862	0.4974	0.3918	**0.6476**
Recall@1	0.0391	0.0862	0.0624	**0.1422**	0.0416	0.0968	0.0694	**0.1573**
Precision@10	0.1650	0.2161	0.2383	**0.2579**	0.2235	0.2953	0.3492	**0.3597**
Recall@10	0.4673	0.5936	0.6372	**0.7106**	0.3822	0.4735	0.5340	**0.7948**
	Instant_Video				Yelp			
	Latest	Random	Length	NCEM	Latest	Random	Length	NCEM
Precision@1	0.2528	0.4219	0.3921	**0.5227**	0.1098	0.3165	0.2230	**0.3723**
Recall@1	0.0418	0.1029	0.0897	**0.1168**	0.0346	0.0842	0.0594	**0.1319**
Precision@10	0.2415	0.2761	0.3174	**0.3709**	0.1574	0.2237	0.2419	**0.2671**
Recall@10	0.4728	0.5861	0.6657	**0.8623**	0.4268	0.6295	0.6627	**0.7324**

form of standard FM, which contains all the information of the standard one.

Generally, under the same number of latent factors, the model achieves better results with the increase of hidden layers. For the rating is generated by the interaction of users and items, especially in non-linear spaces, it requires deeper network structures for modeling. To further verify this, we removed all of the ReLU units where the results were very bad (worse than the standard FM), so we do not show them.

F. RECOMMENDATION INTERPRETABILITY (RQ5)

As we can see in Table 4, the high-weight reviews contain more representative information of items, which is not only useful for item modeling, but also useful for users' reference to help them make informed purchase decisions. Therefore, by providing users with the highly-useful reviews, the interpretability of recommender system is improved.

In fact, this kind of interpretability approach has been adopted in some e-commerce sites. However, the reviews they provide are selected by three simple strategies and can not satisfy users. By the first strategy (named Latest), the review list of an item is generated by selecting the latest N reviews. In the second strategy (named Random), the reviews in a list are selected randomly. The third one (named Length), the longest N are selected.

To verify whether our NCEM can pick up the useful reviews, we conducted a prediction and recall test on four datasets, which contain some reviews that have been rated useful by other users. We assumed that the rated review are ground truth to analyse the performance of attention mechanism in NCEM. We only keep the items having at least one rated review. The evaluation of the performance is according to:

$$Precision@N = \frac{\sum_j^N rel_j}{N} \tag{14}$$

$$Recall@N = \frac{\sum_j^N rel_j}{Re_j^{rated}} \tag{15}$$

where $rel_j = 1/0$ denotes whether the No.j review in the Top-N list have been rated useful. Re_i^{reted} indicated the number of rated reviews in item i. To further study the effect of length of the review list, we set N to 1 and 10.

In Table 6, we can see that NCEM can more precisely find out the useful reviews than other three methods. On the other hand, NCEM has the ability to automatically select useful reviews without manual rated label, which can helpfully apply to the recommender systems that have rare rated reviews to provide interpretable recommendation.

VI. CONCLUSION

In this work, we combine the pre-trained BERT model and the neural FM to perform the rating prediction task. The NCEM proposed in this paper can effectively overcome the shortcomings of the recent CNN based method, which can capture word frequency and global context information to further explore the power of review data. In addition, recent works have proved that FM is better than inner product. This paper extend the standard FM to neural FM, and further improves the modeling capability and generalization performance by constructing deep networks and introducing non-linear transformations. Further, we provide the high attention score reviews to improve the recommendation interpretability. In the future work, we will try to introduce interactions in the learning process between user network and item network, and get the dynamic latent factors of users and items.

REFERENCES

[1] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model.," in *Proc. SIGKDD*, Las Vegas, NV, USA, 2008, pp. 426–434.

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

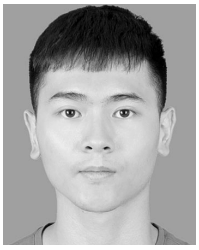
[3] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. NIPS*, Vancouver, BC, Canada, 2001, pp. 556–562.

[4] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. NIPS*, Vancouver, BC, Canada, 2008, pp. 1257–1264.

[5] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proc. SIGIR*, Pisa, Italy, 2016, pp. 549–558.

[6] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. RecSys*, Hong Kong, China, 2013, pp. 165–172.
[7] Y. Bao, H. Fang, and J. Zhang, "TopicMF: Simultaneously exploiting ratings and reviews for recommendation," in *Proc. AAAI*. Montreal, QC, Canada, 2014, pp. 2–8.
[8] Q. Diao, M. Qiu, C. Y. Wu, A. J. Smola, J. Jiang, and C. Wang, "Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS)," in *Proc. SIGKDD*, New York, NY, USA, 2014, pp. 193–202.
[9] G.-N. Hu, X.-Y. Dai, Y. Song, S.-J. Huang, and J.-J. Chen, "A synthetic approach for recommendation: Combining ratings, social relations, and reviews," in *Proc. IJCAI*, Buenos Aires, Argentina, 2015, pp. 1756–1762.
[10] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. RecSys*, Boston, MA, USA, 2016, pp. 233–240.
[11] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. WSDM*, Cambridge, U.K., 2017, pp. 425–434.
[12] R. Catherine and W. Cohen, "TransNets: Learning to transform for recommendation," in *Proc. RecSys*, Como, Italy, 2017, pp. 288–296.
[13] Q. Wang, S. Li, and G. Chen, "Word-driven and context-aware review modeling for recommendation," in *Proc. CIKM*, Turin, Italy, 2018, pp. 1859–1862.
[14] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *Proc. WWW*, Perth, WA, Australia, 2017, pp. 173–182.
[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2019, *arXiv:1810.04805*. [Online]. Available: https://arxiv.org/abs/1810.04805
[16] S. Rendle, "Factorization machines," in *Proc. ICDM*, Sydney, NSW, Australia, Dec. 2010, pp. 995–1000.
[17] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend," in *Proc. RecSys*, Foster, SV, USA, 2014, pp. 105–112.
[18] X. Li, G. Xu, E. Chen, and L. Li, "Learning user preferences across multiple aspects for merchant recommendation," in *Proc. ICDM*, Atlantic City, NJ, USA, Nov. 2015, pp. 865–870.
[19] S. Feng, J. Cao, J. Wang, and S. Qian, "Recommendations based on comprehensively exploiting the latent factors hidden in items' ratings and content," *Trans. Knowl. Discovery Data*, vol. 11, no. 3, Apr. 2017, Art. no. 35.
[20] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proc. ACL*, Stroudsburg, PA, USA, 2012, pp. 90–94.
[21] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proc. WWW*, Lyon, France, 2018, pp. 1583–1592
[22] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.
[23] H. Wu, Z. Zhang, K. Yue, B. Zhang, J. He, and L. Sun, "Dual-regularized matrix factorization with deep neural networks for recommender systems," *Knowl.-Based Syst.*, vol. 145, pp. 46–58, Apr. 2018.
[24] S. Seo, J. Huang, H. Yang, and Y. Liu, "Interpretable convolutional neural networks with dual local and global attention for review rating prediction," in *Proc. RecSys*, Como, Italy, 2017, pp. 297–305.
[25] Y. Tay, A. T. Luu, and S. C. Hui, "Multi-pointer co-attention networks for recommendation," in *Proc. SIGKDD*, London, U.K., 2018, pp. 2309–2318.
[26] Y. Lu, R. Don, and B. Smyth, "Coevolutionary recommendation model: Mutual learning between ratings and reviews," in *Proc. WWW*, Lyon, France, 2018, pp. 773–782
[27] T. Bansal, D. Belange, and A. McCallum, "*Ask the GRU*: Multi-task learning for deep text recommendations," in *Proc. RecSys*, New York, NY, USA, 2016, pp. 107–114.
[28] A. Almahairi, K. Kastner, K. Cho, and A. Courville, "Learning distributed representations from reviews for collaborative filtering," in *Proc. RecSys*, New York, NY, USA, 2015, pp. 147–154.
[29] W. Zhang and J. Wang, "Prior-based dual additive latent Dirichlet allocation for user-item connected documents," in *Proc. IJCAI*, Buenos Aires, Argentina, 2015, pp. 1405–1411.
[30] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. ICML*, Helsinki, Finland, 2008, pp. 160–167.
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[32] R. Hong, Z. Hu, L. Liu, M. Wang, S. Yan, and Q. Tian, "Understanding Blooming human groups in social networks," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1980–1988, Nov. 2015.
[33] H. Zhang, Y. Yang, H. Luan, S. Yang, and T.-S. Chua, "Start from scratch: Towards automatically identifying, modeling, and naming visual attributes," in *Proc. ACM MM*, Orlando, FL, USA, 2014, pp. 187–196.
[34] K. Al-Sabahi, Z. Zuping, and M. Nadher, "A hierarchical structured self-attentive model for extractive document summarization (HSSAS)," *IEEE Access*, vol. 6, pp. 24205–24212, 2018.
[35] W. Zhang, Q. Yuan, J. Han, and J. Wang, "Collaborative multi-level embedding learning from reviews for rating prediction," in *Proc. IJCAI*, New York, NY, USA, 2016, pp. 2986–2992.
[36] X. He and T. S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. SIGIR*, Tokyo, Japan, 2017, pp. 355–364.
[37] P. Li, Z. Wang, Z. Ren, L. Bin, and W. Lam, "Neural rating regression with abstractive tips generation for recommendation," in *Proc. SIGIR*, Tokyo, Japan, 2017, pp. 345–354.
[38] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," 2017, *arXiv:1708.04617*. [Online]. Available: https://arxiv.org/abs/1708.04617
[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[41] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167

XINGJIE FENG received the B.S. degree in mathematics from Nankai University, in 1991, the M.S. degree in computer science from the Nanjing University of Aeronautics and Astronautics, in 2000, and the Ph.D. degree in computer science from Nankai University, in 2004. He is currently a Professor with the School of Computer Science and Technology, Civil Aviation University of China. His current research interests include recommendation systems, database and data warehouse, and intelligent information processing theory and technology.



YUNZE ZENG received the B.S. degree in computer science from the Civil Aviation University of China, in 2017, where he is currently pursuing the M.S. degree in computer science. His current research interests include recommender systems and deep leaning techniques.