Knowledge-Based Systems 181 (2019) 104786

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

ACNN-FM: A novel recommender with attention-based convolutional neural network and factorization machines☆

Guangyao Pang [a,b,c], Xiaoming Wang [a,b,*], Fei Hao [a,b], Jiehang Xie [a,b], Xinyan Wang [a,b], Yaguang Lin [a,b], Xueyang Qin [a,b]

[a] Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an, 710062, China
[b] School of Computer Science, Shaanxi Normal University, Xi'an, 710119, China
[c] School of Data Science and Software Engineering, Wuzhou University, Wuzhou, 543002, China

ARTICLE INFO

Article history:
Received 19 November 2018
Received in revised form 16 May 2019
Accepted 20 May 2019
Available online 22 May 2019

Keywords:
Recommendation
Convolutional neural network
Attention mechanism
Factorization machines
Machine learning

ABSTRACT

With the rapid development of the Internet, the data generated from application platforms such as online shopping, e-education, and digital entertainment has exhibited dramatical growth, which has caused serious information overload to Internet users. The traditional recommendation approaches are crucial for Internet users to extract valuable information from various information. However, there exist some problems such as sparse data, cold start, and over-reliance on manual extracted feature and so on. To address the above problems, this paper proposes a novel recommender with Attention-based Convolutional Neural Network and Factorization Machines (ACNN-FM), which achieves the recommendation with comments. Firstly, from the perspective of local to overall, this paper proposes a word-level attention mechanism and a phrase-level attention mechanism to increase the ability to remember the importance and the order of historical vocabulary (phrase) in the process of text processing of convolutional neural networks. Secondly, it constructs a model to automatically extract hidden features of users and items from comments in the form of natural language. Finally, we utilize factorization machines to analyze the association between the hidden features of users and items, and implement the recommendation based on the association. Extensive experiments are conducted for demonstrating that ACNN-FM method outperforms state-of-the-art NARR method, and ACNN-FM has the highest data utilization among NARR, DeepCoNN, BCF and NMF methods, thus the recommendation performance is significantly improved in large-scale data environment.

1. Introduction

With the rapid development of technologies such as cloud computing, big data, and Internet of Things, a large number of application platforms have emerged in Internet and information industries [1] such as online shopping, e-education, digital entertainment, and so forth. In these platforms, the total amount of multi-source heterogeneous data is growing rapidly and is expected to reach 35.2ZB by 2020 in the world [2]. These massive data is valuable and can guide people to transform their behavioral decision-making model from empiricism to data-driven [3]. However, while people enjoy the convenience of massive data, they also face the problem of information overload [4]. Therefore, how to mine users' interested items from these data according to the needs and interests of users has becoming a significant research topic.

In recent years, many researchers have utilized recommendation to tackle information overload problems in some platforms such as online shopping, e-education, and digital entertainment [5]. The recommendation system mines items of interest to users (such as goods, knowledge, movies, music), and recommends the personalized results to users [6]. The commonly used recommendation methods are mainly divided into three categories: collaborative filtering-based recommendation methods [7], content-based recommendation methods [8], and hybrid recommendation methods (including machine learning algorithms) [9].

The collaborative filtering-based recommendation methods mine a user's preference from the historical behavior data of the user, divide users into different groups according to different preference, and finally recommend the user's favorite items in the same group to the other user. This type of method only focuses

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.05.029.

* Corresponding author at: School of Computer Science, Shaanxi Normal University, Xi'an, 710119, China.
E-mail addresses: pangguangyao@snnu.edu.cn (G. Pang), wangxm@snnu.edu.cn (X. Wang), fhao@snnu.edu.cn (F. Hao).

https://doi.org/10.1016/j.knosys.2019.05.029
0950-7051/

on the user's preference for the items and utilizes One-Hot encoding [10] to represent the preference matrix of users-items. In the large-scale data environment, the matrix dimension of users-items will reach tens of millions of or even hundreds of millions, which causes serious data sparseness problems [10] (The number of items evaluated by a user is only a small part of the total number of items) and results in a rapid decline in the recommended performance of collaborative filtering-based recommendation methods.

The content-based recommendation method mainly utilizes the item attributes selected by a user to find other items with similar attributes, and the performance of this type of method is excessively dependent on the accuracy of extracting valid data features for users and items. At present, the widely used manual extraction cannot adapt to the large-scale data environment, and seriously restricts the application scope of the content-based recommendation method.

In addition, new users and new items have little or no rating information (*i.e.*, so-called cold start issues [11]), which can seriously affect the performance of two recommendation approaches mentioned above. The traditional hybrid recommendation methods utilize three different combination strategies of pre-fusion, middle-fusion and post-fusion to recombine the above two recommendation methods to achieve better recommendation performance in the case of small data. Among them, the pre-fusion refers to melt multiple recommendation methods into an integrated model, which produces the recommendation results; the middle-fusion utilizes a recommendation method as main method, and then another recommendation method is incorporated into it; the post-fusion utilizes multiple recommendation methods to calculate multiple results, and then vote to select the optimal value. However, the above problems such as data sparseness, cold start, and overreliance on the artificial feature extraction still exist in the hybrid recommendation methods.

Recent years, the deep learning algorithms are applied to the recommendation field [12–14]. Their principles are to automatically extract the data features, which in a certain extent solves the problem that the feature extraction is over-reliant on the labor in the traditional recommendation method. However, there is still much space for the improvement in the accuracy of features extraction, and there are still some difficulties in the integration with the recommendation methods.

In order to solve these problems, this paper proposes a hybrid recommendation method based on the integration of the Attention mechanism-based Convolutional Neural Network and Factorization Machines (ACNN-FM). Specifically, the proposed method can automatically extract hidden features (labeled data that can express features of things like users or items [15]) of users and items from their comments, and use the rating as the supervised data to mine the association relationship between the core hidden features, and then establish the relationship between the user and the item according to the core hidden features, finally achieve recommend items in which the users are interested. Technically, our method includes two core research aspects: firstly, we propose a Convolutional Neural Network (CNN) based on attention mechanism that can improve the accuracy of features extraction in the way of improving the memory of historical information during the text analyzing process. Secondly, we employ the factorization machines to construct the relationship between users and items to accomplish the recommendation. The main contributions of this paper are summarized as follows:

- **Attention Mechanism:** We propose an attention mechanism (including word-level attention mechanism and phrase-level attention mechanism) based on CNN model in the field of Natural Language Processing (NLP). The mechanism extends CNN from two different granularities (local to global), while taking into account the impact of core word and important phrases on the target features in the comments. Thus, this mechanism solves the problem of memory losses of CNN model. It means that it can automatically extract hidden features of users and items from the comments in the form of natural language (that is, it solves the problem of content-based recommendation methods relying too much on manual extraction of features). Finally, the feature extraction accuracy of the CNN in the NLP field is significantly improved.
- **ACNN-FM Recommender:** We propose a hybrid recommendation method based on the integration of the attention mechanism-based CNN model and factorization machines. The method first automatically extracts the core hidden features of users and items, and utilizes the rating as the supervised data to mine the relationship between these core hidden features to construct the user-item association, and achieve the recommendation with higher precision. The proposed approach could overcome the defects of data sparseness, cold start, poor recommendation performance, poor interpretability, and poor applicability of the traditional recommendation model, and substantially improve the efficiency and the accuracy of recommendation.
- **Evaluation:** We conduct the experiments on four data sets to validate the effectiveness of the proposed approach. In the process of performance evaluation, this paper designs two different evaluation scenarios for the cold start problem. The experimental results show that the recommended accuracy of ACNN-FM outperforms that of NARR, DeepCoNN, BCF, and NMF.

The organization of this paper is as follows: In Section 2, we introduce the related work of the existing recommendation, the problem of this paper is formulated in Section 3. In Section 4, we present the framework and implementation details of the ACNN-FM. Further, the overall algorithm for the ACNN-FM is provided in Section 5. Section 6 verifies the validity and interpretability of ACNN-FM through extensive experiments. Section 7 summarizes the paper and presents an outlook.

2. Related works

At present, recommendation approaches include collaborative filtering-based recommendation, content-based recommendation, and hybrid recommendation, which are effective solutions to solve the problem of the information overload in the Internet and other platforms. Furthermore, in recent years, as deep learning has become a research hotspot in larger-scale data and artificial intelligence [16], a new type of hybrid recommendation method based on deep learning has emerged.

In the collaborative filtering-based recommendation methods, Ma et al. [17] proposed a recommendation method based on Local Low-Rank Matrix Approximation (LLROMA) to solve the over-fitting problem of Probabilistic Matrix Factorization (PMF). The method initially estimates the parameters with local optimum, then reduces the over-fitting phenomenon of each local model, and finally improves the recommendation effect. In the content-based recommendation methods, Chen et al. [18] proposed a recommendation method for the partial membership model in the field of image segmentation, taking into account the overlap of the images (such as fish in the water and people in the fog etc.). This method achieves the effect of recommending multiple labels for a single image. In the hybrid recommendation methods, Xu et al. [19] proposed a travel recommendation

method based on Dynamic Topic Model (DTM) and Matrix Factorization (MF), which uses DTM to obtain user and location information of topic distribution, and then analyzes the similarity between the user and the location by matrix decomposition to accomplish the recommendation and solves the problem of data sparsely to a certain extent. These methods can achieve better recommendation results in the environment of small-scale data. However, the collaborative filtering-based recommendation method is not accurate in dealing with sparse data, and the time overhead is too large when performing similarity calculation. The content-based recommendation method is overly dependent on artificial extraction in the way of features extraction, and cannot quickly extract effective features. Additionally, the hybrid recommendation method still has the difficulty of cold start, poor interpretability, and low applicability.

To tackle the problems of the above traditional recommendation methods, deep learning technologies are extracting much attention from many researchers. Importantly, deep learning can form a more dense high-level semantic abstraction by combining low-level features [20], that is, it can automatically extract features from data [21]. Therefore, the application of deep learning to the data processing of recommendation methods is a research hotspot in current recommendation systems and future research trends [9]. For example, Yin et al. [14] proposed a Spatial-aware Hierarchical Collaborative Deep Learning model (SH-CDL), which utilizes deep learning methods to extract personal preferences from individual points of interest, and then utilizes collaborative filtering to analyze the internal relationships of individual preferences to accomplish the recommendation. This method relieves the problem of cold start in a certain extent. Zheng et al. [22] adopted two parallel neural networks to learn the hidden features of users and items, and utilizes a content-based recommendation method to fuse features for recommendations. They proposed a method based on Deep Cooperative Neural Network (DeepCoNN), which solved the problem of over reliance on manual extraction of features extraction in traditional recommendation methods.

Effective utilization of the context and core content of text information can improve the accuracy of feature extraction with machine learning model [23], the emerging attention mechanism is currently used to extend the neural network [24–26], where the neural network mainly includes CNN and Recurrent Neural Networks (RNN), and its purpose is to extract the user's rich knowledge (such as users' potential preferences and community tendency) regarding various desired patterns from the high-dimensional sparse matrix which is generated by recommender systems [27,28]. Li et al. [29] proposed a Global-Local Attention (GLA). Firstly, the CNN model based on the attention mechanism is utilized to analyze which objects are included in the image, and then the RNN model based on the attention mechanism is utilized to construct the relationship between these objects and the text features, so that the GLA model can generate accurate text description for the images. Wang et al. [30] proposed a meta-attention model across multiple deep neural networks to achieve automatic recommendation of hot articles. The model first utilizes the RNN model to extract the expression characteristics of the article (such as the theme and keywords etc.), and then utilizes the attention-based DNN model to establish the relationship between these expression features and readers, which solves the problem of article recommendation manually.

According to the above related research, RNN can deal with the context of text, and has certain advantages in the field of NLP. However, its computational cost becomes very expensive as the vocabulary becomes larger. Fortunately, CNN exhibits very good performance in video and image processing, and has high extraction accuracy in high-dimensional data. If we can propose a mechanism to increase the performance on processing context of CNN in the NLP field, it will be able to improve the accuracy of recommendation system in large-scale data environment.

To sum up, there still exist three major challenges in the recommendation systems:

- How to increase the accuracy of extracting features from multi-source heterogeneous data?
- How to transform the traditional method of manual feature extraction into automatic way?
- How to incorporate deep learning with traditional recommendation methods as a strong collaborative hybrid model.

Toward this end, we first propose an attention mechanism to extend CNN. The proposed mechanism is able to accelerate the CNN to extract the hidden features of users and items from large-scale text data with high precision. Secondly, the attention-based CNN and the extended FM are combined to accomplish the precision recommendation.

3. Problem statement

The recommendation method utilizes the evaluation information as the training data in solving the problem of information overload based on various network platforms. Fig. 1 shows a scenario in which users evaluate items in the Amazon shopping platform,[1] which includes the comments and rating information. At present, the recommendation method relies solely on the rating as the training data, which is difficult in improving the recommendation accuracy. Therefore, we consider how to effectively utilize the comments to improve the recommendation accuracy. Specifically, in the evaluation scenario of the Amazon platform, we need to solve several core issues: how to extract hidden features of users and items from comments (as shown in step 2, 4 and step 1, 3, respectively)? And how to utilize the hidden features of users and items to build user-item associations for recommendation systems (as shown in step 5, 6)?

Definition 1. We use a tuple $P = (u, m, c, r)$ to represent an user's evaluation of the item, where u is an user, m is the item, c is the comment of the user u for the item m, and r is the rating of the user u to the item m.

Definition 2. The comments of a user on different items are represented by set $c^u = \{c_1^u, c_2^u, \ldots c_m^u, \ldots c_k^u\}$, and the comments of different users on an item are represented by set $c_m = \{c_m^1, c_m^2, \ldots c_m^u, \ldots c_m^N\}$. Among them, c_m^u denotes the comments of user u on item m, k is the total number of items commented by user u, and N indicates the number of users who have commented on item m.

Problem. How do we utilize comments in the form of natural language to predict user ratings for item? Based on the above definitions, the problem can be formalized as:

$$\begin{aligned} &\textbf{\textit{Input}} : c^u, c_m \\ &\textbf{\textit{Output}} : \widehat{y} \\ &\textbf{\textit{Constraints}} : \begin{cases} H_u \leftarrow c^u \\ H_m \leftarrow c_m \\ R{:=}\arg\max \mathit{Relation}(H_u, H_m) \\ \widehat{y} = f\,(R) \\ \alpha{:=}\arg\ \min \frac{1}{2}(y-\widehat{y})^2 \end{cases} \end{aligned} \tag{1}$$

[1] https://www.amazon.com/.

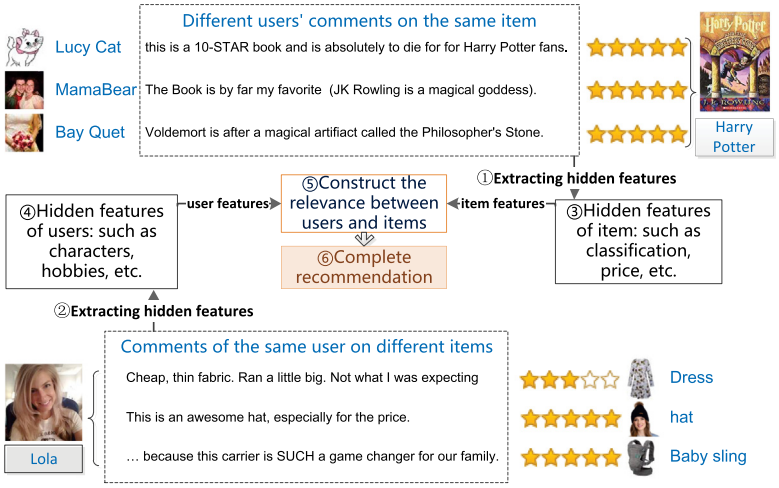

Fig. 1. Scenario of users evaluation of items on Amazon platform.

Table 1
Symbols and descriptions.

Symbols	Definitions and description
v_{ij}^{u}	The numerical value of the jth word of the ith sentence in the user u's comment
$v_{m,kq}$	The numerical value of the qth word of the kth sentence in the item m's comment
$F_{i,r}$	Expression feature vector of user ($i = 0$) or item ($i = 1$)
$F_{i,a}$	Attention feature vector of user ($i = 0$) or item ($i = 1$)
$F_{i,nr}$	New expression feature vectors incorporating attention feature vectors of user ($i = 0$) or item ($i = 1$)
b_j	The bias of jth convolutional kernel
x_j	The jth kernel in the convolutional layer
c_j	The jth feature map in the convolutional layer
g	The bias of the fully connected layer
o_j	The output of jth neuron in the convolutional layer
H_u	User hidden features
H_m	Item hidden features
∂	The learning rate
l	Length of data set
s	The number of each batch in the training set
$\widehat{y}$	User's predicted rating of the item

The goal of the addressed problem is to find a model that takes a user's comments c^u and a item's comments c_m as *Input*, and after the operation with *Constraints*, *Outputs* the user u's rating $\widehat{y}$ for the item m. Where H_u is the hidden feature of user u, H_m is the hidden feature of item, *Relation()* is a function that calculates the degree R of correlation between user u and item m, *f()* is a score prediction function, and the closer the prediction rating $\widehat{y}$ is to the actual rating y, the better the recommended performance (*i.e.* α gets the minimum).

In order to improve the accuracy of recommendation, we need to extract hidden features from comments, improve the relevance of users and items constructed by using the hidden features of users and items, and evaluate rating based on the degree of relevance. Table 1 lists used symbols and descriptions in this paper.

4. The proposed approach

For the sake of addressing the problem defined by Eq. (1), we propose an ACNN-FM approach, which extracts hidden features of users and items from comments, and completes recommendations based on the hidden features. The overall framework of ACNN-FM mainly includes the following functional modules:

- **Word embedding model:** This model utilizes n-dimensional distributed vectors to transform comments in the form of natural language into numerical value that can be recognized by CNN model;
- **Word-level attention mechanism:** Our proposed word-level attention mechanism can increase the degree of association of the comments between users and items based on the importance and order of words in sentences, and can overcome the drawbacks of lack of memory in the NLP field of CNN;
- **CNN models:** CNN can utilize comments to extract hidden features of users and items. The main operations include convolution, pooling and full connectivity;
- **Phrase-level attention mechanism:** The phrase-level attention mechanism that we proposed after the convolution layer considers the importance of the phrase in the sentence, making the features extracted by CNN more representative;
- **Factorization machines model:** It mainly utilizes the hidden features of users and items to construct the association between users and items to achieve the recommendation.

ACNN-FM consists of the following steps, and is shown in Fig. 2: *word embedding model* converts comments of users and items into numerical data (Section 4.1), *word-level attention mechanism* (Section 4.2) and *phrase-level attention mechanism* (Section 4.4) increase the memory of the numerical data, and *CNN models* extracts features of users and items from the numerical data (Section 4.3), and the *factorization machines model* uses the features for recommendation (Section 4.5).

4.1. Word embedding model

Since, deep learning can only process numerical data, we have improved the word embedding model to be able to quantify natural language. The word embedding model includes two steps: normalization and quantification. Specifically, natural language is firstly normalized by segmenting words, removing stop words and useless words, and then multi-dimensional distribution vector is used to quantize the comments.

Firstly, c_m^u denotes the comments of user u on item m, assuming $c_m^u = \{s_{m,1}^u, s_{m,2}^u, \ldots, s_{m,i}^u, \ldots, s_{m,n}^u\}$, where $s_{m,i}^u$ represents the ith sentence of the user u commenting on the item m. Then

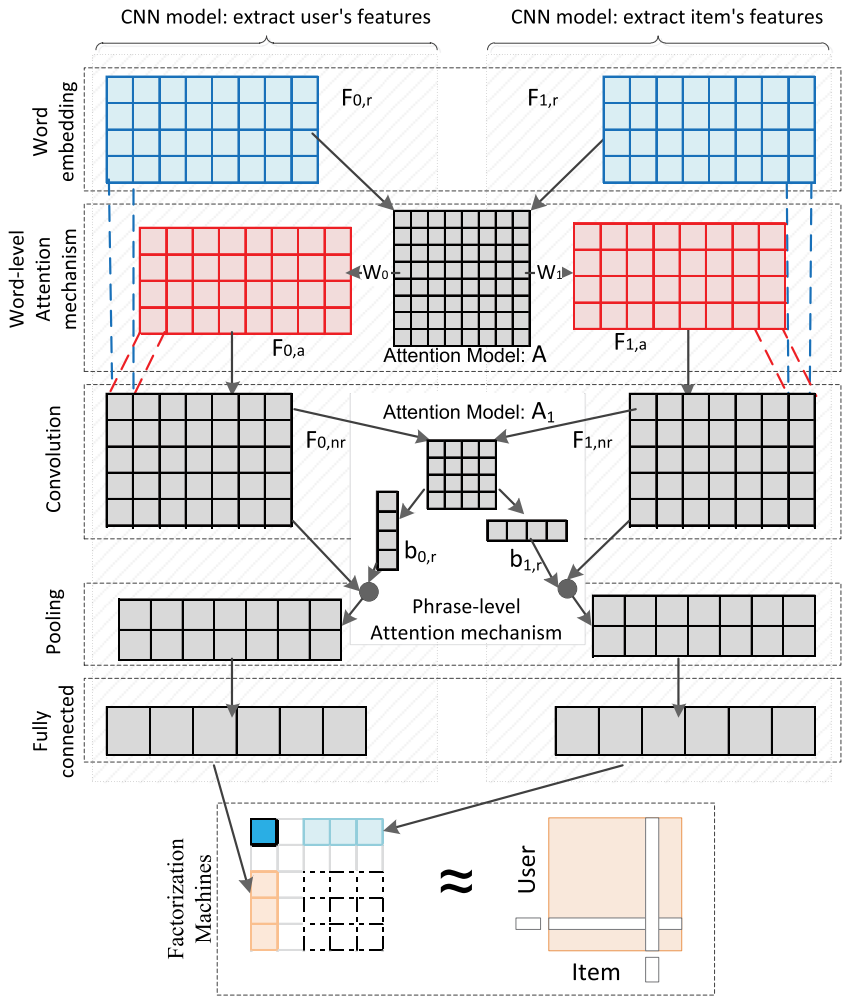

Fig. 2. The architecture of the proposed approach.

assume the sentence $s^u_{m,i} = \{w_{i1}, w_{i2}, \ldots, w_{ij}, \ldots, w_{in}\}$, where w_{ij} represents the jth word of the ith sentence, and n is the number of words for each sentence. In order to establish the correspondence between words and numerical values, we establish a mapping function $\phi\left(w_{ij}\right) : w_{ij} \to Z,\ Z \in N^+$, which represents a mapping relationship from the word w_{ij} to the numerical value Z. Then, we construct a n-dimensional distributed vectors V^u of the user u's comments which utilize numerical value to express:

$$V^u = \begin{bmatrix} \phi(w_{11}) & \phi(w_{12}) & \cdots & \phi(w_{1j}) & \cdots & \phi(w_{1n}) \\ \phi(w_{21}) & \phi(w_{22}) & \cdots & \phi(w_{2j}) & \cdots & \phi(w_{2n}) \\ \cdots & & & & & \\ \phi(w_{i1}) & \phi(w_{i2}) & \cdots & \phi(w_{ij}) & \cdots & \phi(w_{in}) \\ \cdots & & & & & \\ \phi(w_{q1}) & \phi(w_{q2}) & \cdots & \phi(w_{qj}) & \cdots & \phi(w_{qn}) \end{bmatrix} \tag{2}$$

where $V^u_{ij} \in V^u$ is a numerical result of the word embedding model after processing the word w_{ij} in the user u's comment, and q denotes the total number of sentences of all user u's comments. Similarly, we can also construct a n-dimensional distributed vectors V_m for the item m's comments, where $V_m = V^u$, and get $V_{m,kq} \in V_m$ to indicate the numerical result after the word embedding module processes the word w_{ij} in the item m's comment.

Secondly, we assume the expression feature vector of a batch of data is $F_{i,r} \in R^{d \times n}, i \in \{0, 1\}$, where $i = 0$ and $i = 1$ represent the expression feature vector of the user and item, respectively. n is the defined length of the sentence, d is the data dimension of each training batch. If the number of sentences q of the user comments is greater than the batch d, it needs to be divided into multiple batches. Otherwise, it is necessary to increase $(d - q)$ row data with zero value. That is, the user expresses the feature vector $F_{0,r}$:

$$F_{0,r} = \begin{cases} \left(V^u_{1:}\ V^u_{2:}\ \cdots\ V^u_{k:}\ \cdots\ V^u_{q}\right)^T, q \geq d \\ \left(V^u_{1:}\ V^u_{2:}\ \cdots\ V^u_{k:}\ \cdots\ V^u_{q}\ \cdots\ 0\right)^T, q < d \end{cases} \tag{3}$$

In the similar way as Eqs. (2) and (3), we can obtain the item feature expression vector $F_{1,r}$:

$$F_{1,r} = \begin{cases} \left(V_{m,1:}\ V_{m,2:}\ \cdots\ V_{m,k:}\ \cdots\ V_{m,q:}\right)^T, q \geq d \\ \left(V_{m,1:}\ V_{m,2:}\ \cdots\ V_{m,k:}\ \cdots\ V_{m,q:}\ \cdots\ 0\right)^T, q < d \end{cases} \tag{4}$$

After the above processing, we construct a word embedding model that utilizes the n-dimensional distribution vector to quantify the comments, and obtain the user expression feature vector $F_{0,r}$ and the item expression feature vector $F_{1,r}$.

4.2. Word-level attention mechanism

The CNN model has better feature extraction effects in terms of static data such as images and audio. However, in the process of natural language, the lacking memory of historical vocabularies ignores the weight of historical vocabularies, loses the location

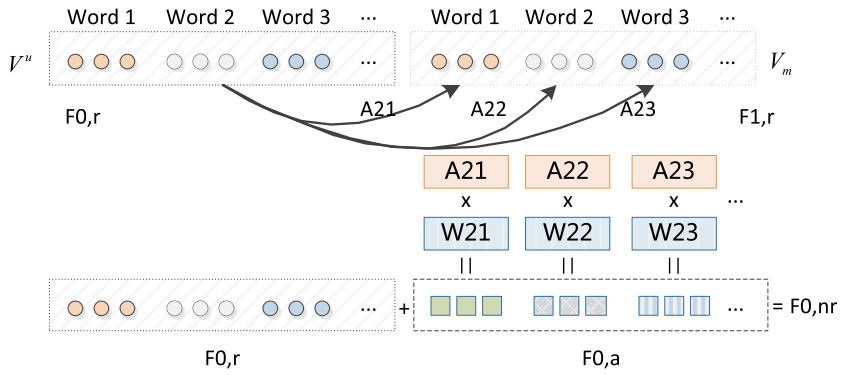

Fig. 3. Schematic diagram of attention mechanism.

information of vocabularies and results in the low accuracy of features extraction in natural processing. In order to solve the above problems, we propose a word-level attention mechanism to extend the CNN model, which mainly considers the importance of vocabulary in sentences and increases the influence degree between users and items comments in the training data to improve the memory of historical vocabularies of the CNN model. In addition, compared with the conventional in-depth learning model, the mechanism can give the contribution of each word to the target features, that is, ACNN-FM has explanatory nature.

Specifically, we propose an attention mechanism based on the defined user expression feature vectors $F_{0,r}$ and the item expression feature vector $F_{1,r}$. Assuming that the attention matrix $A \in R^{n\times n}$ represents the degree of mutual influence of words between a set of sentences in c^u and a set of sentence in c_m. Fig. 3 shows the operation details of the attention mechanism in an iterative training process. Clearly, our attention mechanism consists of the following three steps:

Step 1: This attention mechanism utilizes the Euclidean distance formula to calculate the matrix A, for any element $a_{kj} \in A$, a_{kj} is defined as follows:

$$a_{kj} = \left\| v^u_{k:} - v_{m,j:} \right\|_2^2 \tag{5}$$

where $\|\cdot\|_2^2$ is the Euclidean distance paradigm.

Step 2: We assume that the feature vector of the user attention is $F_{0,a} \in R^{d\times n}$, and the feature vector of the item attention is $F_{1,a} \in R^{d\times n}$, which indicates the relationship between users and items mined from the comments. We utilize the weight matrix in the related theory of machine learning to construct the relationship, that is, we firstly assume that the user's attention weight is $W_0 \in R^{d\times n}$ and the item's attention weight is $W_1 \in R^{d\times n}$, then randomly initialize W_0 and W_1, and finally adjust the value of W_0 and W_1 according to the error of the prediction result. The actual result then obtains the optimal value after multiple iterations. The optimal value and attention matrix A can quickly construct the optimal $F_{0,a}$ and optimal $F_{1,a}$. The detailed calculation process is formalized as follows:

$$\left(F_{i,a}\right)_{d\times n} = (W_i)_{d\times n} \cdot (A)_{n\times n}\ , i \in \{0, 1\} \tag{6}$$

Step 3: Constructing a new feature vector that combines the degree of mutual influence between users and items. It is known that the data dimension of the expression feature vector of the user $F_{0,r}$ (or the expression feature vector of the item $F_{1,r}$) and the user attention feature vector $F_{0,a}$ (or the item attention feature vector $F_{1,a}$) are both $d \times n$. Thus, we could construct a new expression feature vector of the user $F_{0,nr}$ (or a new expression feature vector of the item $F_{1,nr}$) by splicing $F_{0,r}$ and $F_{0,a}$ (or $F_{1,r}$ and $F_{1,a}$) in a higher dimension:

$$F_{i,nr} = concat\left(F_{i,r}, F_{i,a}\right), i \in \{0, 1\} \tag{7}$$

In general, the attention mechanism constructed in this paper mainly includes two core strategies. (1) Utilizing the comments to construct the interaction effect between users and items, namely the attention feature vectors $F_{0,a}$ and $F_{1,a}$. (2) Constructing new expression feature vectors $F_{0,nr}$ and $F_{1,nr}$ that combine the interaction effect of users and items.

4.3. CNN model

The comments in the form of natural language contain rich information about the relationship between users and items. We utilize the CNN model to extract the local features and core features from comments [24] to express the hidden features of users and items.

The CNN model includes a convolutional layer, a pooled layer, and a fully connected layer. Firstly, in the convolutional layer, we utilize the convolution kernel $x_j \in R^{d\times n}$ to scan $F_{0,nr}$, that is to say, we can extract rich feature vectors of x_j dimension from discrete sparse one-dimensional feature vectors. $C = [c_1, c_2, \ldots, c_j, \ldots, c_n]$, $C \in R^n$ of the local data, where we utilize the *ReLU* [24] activation function to aggregate the core feature c_j. Secondly, in the pooling layer, we perform a downsampling operation on C, select representative feature information o_j locally, and aggregate these information into a total feature set O. Finally, we recombine the total feature set O in the full join layer, to obtain H_u which can express the global users' features. To obtain the CNN model, we employ the recursive equation to calculate and update the relevant parameters [22]:

$$c_j = \mathrm{ReLU}\left(F_{0,nr} * x_j + b_j\right) \tag{8}$$

$$o_j = \max\{c_1, c_2, \ldots, c_j, \ldots, c_{n-t+1}\} \tag{9}$$

$$O = \left\{o_1, o_2, \ldots, o_j, \ldots, o_{n1}\right\} \tag{10}$$

$$H_u = f\left(W \cdot O + g\right) \tag{11}$$

where the operator $*$ indicates the convolution operation, b_j is the offset variable, and f is the activation function *ReLU*.

Generally, the CNN model can extract the hidden features H_u of a user from the user's new expression feature vector $F_{0,nr}$. Similarly, the CNN model running concurrently and having the same structure can extract the hidden feature H_m of a item from the new expression feature vector $F_{1,nr}$ of the item.

4.4. Phrase-level attention mechanism

Inspired by the LF model, the more representative the hidden features we extracted, the better the recommended accuracy of the model [31–33]. Therefore, in order to make the features extracted by the CNN model more representative, we propose a phrase-level attention mechanism. Based on the output features of the convolutional layer, the mechanism analyzes the relationship between the overall user comments on each phrase in the item (or the association of the overall item comments to each phrase in the user), and uses this relationship as a weight to influence the features of the convolution output. The reasoning process is as follows:

Firstly, we input the user convolution feature C_0 and the item convolution feature C_1 into Eq. (5) to obtain the attention matrix $A_1 \in R^{d\times t}$ $(t = n - w + 1)$. It is known that the nth line of A_1 indicates the importance of the nth word of the user comment to each word in the item comment, and the mth column indicates the importance of the mth word of the item comment to each word in the user comment. Therefore, the calculation process of the attention weight $b_{0,r}$ of the user comment to each word in the item comment and the attention weight $b_{1,r}$ of the item comment to each word in the user comment are expressed as:

$$b_{0,r} = \sum A_1[r, :] \tag{12}$$

$$b_{1,r} = \sum A_1[:, r] \tag{13}$$

where r is the number of the words for the user comments or item comments.

Finally, we make sure of $b_{0,r}$ and $b_{1,r}$ as weights to influence C_0 and C_1 to generate new features $C_0{}'$ and $C_1{}'$:

$$C_i'[:, r] = \sum_{r=t:t+w} b_{i,r} C_i[:, r]\ ,\ t = \{1, 2, \ldots, (n - w + 1)\}\ , i \in \{0, 1\} \tag{14}$$

where w is the stride of the convolution, and n-w+1 is the feature length after convolution by w. By putting features $C_0{}'$ and $C_1{}'$ back into Eq. (9), we can not only seamlessly integrate with the original CNN model, but also make the features extracted by the CNN more representative.

4.5. Factorization machines model

The factorization machines model is a general-purpose predictor extended by linear regression (LR) and Singular Value Decomposition (SVD). It can learn the relationship about different features under sparse matrix by rating, and predicts the user's preference for the new item [34].

The ACNN-FM method mainly consists of two parts: the first part is to extract hidden features H_u and H_m of users and items synchronously from the comments by using two parallel attention mechanism based CNN. The second one is that the FM model is utilized to analyze the association model $\widehat{x}$ of user-item between H_u and H_m, and predict the user's score $\widehat{y}$ on the item according to $\widehat{x}$. The specific calculation process of the FM model is as follows:

First, we concatenate H_u and H_m as a combined feature vector x. To discover more association features in the combined feature vector and to fit better training data, we extended the FM model by using the *ReLU* function to process x:

$$x = concat\ (H_u, H_m) \tag{15}$$

$$\widehat{x} = \max\{0,\ x\} \tag{16}$$

where $\widehat{x} = \left\{\widehat{x}_1,\ \widehat{x}_2, \ldots,\ \widehat{x}_i, \ldots,\ \widehat{x}_j, \ldots\right\}$.

Secondly, in order to use the nonlinear model to learn the high-order combination features in the FM model while controlling the computational complexity to an acceptable range, our FM model uses a second-order combination to analyze the features:

$$\widehat{y}\left(\widehat{x}\right) = w_0 + \sum_{i=0}^{|\widehat{x}|} w_i \widehat{x}_i + \sum_{i=1}^{|\widehat{x}|-1} \sum_{j=i+1}^{|\widehat{x}|} w_{ij} \widehat{x}_i \widehat{x}_j \tag{17}$$

where w_0 is the global bias, w_i is the weight corresponding to $\widehat{x}_i$, w_{ij} is the weight between the features $\widehat{x}_i$ and $\widehat{x}_j$. Furthermore, the FM model solves the sparse problem of real data by using the idea of matrix decomposition (that is, there are few cases where two features are not zero at the same time). That means the value in the weight matrix is equal to the product of the two hidden vectors learned, i.e. $w_{ij} = \langle v_i,\ v_j \rangle$, where v_i denotes the ith dimension vector of the weight matrix w, and $v_i = \left(v_{i,1},\ v_{i,2}, \ldots,\ v_{i,k}\right)$, v_j denotes the jth dimension vector of the weight matrix w, and $v_j = \left(v_{j,1},\ v_{j,2}, \ldots,\ v_{j,k}\right)$, k is a hyper-parameter, as follows:

$$\langle v_i, v_j \rangle = \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \tag{18}$$

Finally, for the sake of minimizing the error between the predicted value $\widehat{y}$ and the actual value y, we need to get the optimal parameters through training of the ACNN-FM. Namely, the parameter $\Theta = \left(w_0,\ w_1, \ldots,\ w_{|\widehat{x}|},\ v_{1,1},\ v_{1,2}, \ldots,\ v_{|\widehat{x}|,k}\right)$ of the model takes the value: $\arg\min_{\Theta} \sum_{i=0}^{n} l\left(\widehat{y}\left(\widehat{x}\right),\ y\right)$. Moreover, we take advantage of the ridge regression strategy to optimize the parameters of the model for our model to achieve higher accuracy in data beyond training. The formula is:

$$OPT\ (S,\ \lambda) = \arg\min_{\Theta} \left(\sum_{(\widehat{x},y)\in S} l\left(\widehat{y}\left(\widehat{x}|\Theta\right),\ y\right) + \sum_{\theta\in\Theta} \lambda_\theta \theta^2 \right) \tag{19}$$

where λ_θ is the regularization coefficient of parameter θ, and S is the training set.

We use the Adaptive Moment Estimation (Adam) [35] to solve the parameters in the minimized loss function during the optimization learning process, and use the *dropout* strategy to prevent over fitting during the learning process. After this series of training processes, we have constructed the association between users and items from the hidden feature H_u and H_m of users and items, respectively, and finally predict the user's rating of the item according to the association.

5. ACNN-FM algorithm

This section mainly focuses on the description of ACNN-FM algorithm in this paper.

The main function of ACNN-FM algorithm is to input a user comments c^u and a item comments c_m, and then predict user ratings $\widehat{y}$ for the item. The detailed implementation process is mainly divided into three stages:

(1) We utilize word embedding to quantize user comments c^u and item comments c_m in the form of natural language into expression feature vectors of users and items $F_{0,r}$ and $F_{1,r}$ (Lines 4–5). By calculating the similarity between the user comments and the item comments, we obtain a attention matrix A that expresses the influence between the user comments and the item comments (Line 6). Then we calculate the attention matrix A and the weights (W_0 and W_1) to obtain the feature vector of users' word-level attention $F_{0,a}$ and the feature vector of items' word-level attention $F_{1,a}$, which utilize the deep learning related theory

Algorithm 1 ACNN-FM Algorithm.

Require: user comments c^u, item comments c_m; Tuple $P = (u, m, c, r)$; Parameters: ∂, w_i, b_j, l, s, δ(the value of RMSE);
Ensure:
1: Initialize: $\partial = 0.0004$, $s = 100$; Random w_i, b_j;
2: **repeat**
3: **for** each $b \in [1, (l/s)]$ **do**
4: Word segmentation for c^u and c_m to get w;
5: $F_{0,r} \leftarrow V_{ij}^u = \phi(w_{ij})$, $F_{1,r} \leftarrow V_{m,ij} = \phi(w_{ij})$;
6: Calculate attention matrix A according to Eq. (5);
7: $F_{0,a}, F_{1,a} \leftarrow A \cdot W_i, i \in \{0, 1\}$;
8: $F_{0,nr}, F_{1,nr} \leftarrow concat\left(F_{i,r}, F_{i,a}\right), i \in \{0, 1\}$;
9: $c_j = \mathrm{Re}LU\left(F_{0,nr} * x_j + b_j\right)$;
10: Calculate attention matrix A_1 by substituting c_j into Eq. (5);
11: $b_{0,r} = \sum A_1[r, :]$, $b_{1,r} = \sum A_1[:, r]$;
12: $C_i'[:, r] = \sum_{r=t:t+w} b_{i,r} C_i[:, r]$, $t = 1 \cdots (n - w + 1), i \in \{0, 1\}$;
13: Replace the c_j of user with C_0' and replace the c_j of item with C_1';
14: $o_j = \max\{c_{1,}c_2, \ldots, c_j, \ldots, c_{n-t+1}\}$;
15: $O = \left\{o_1, o_2, \ldots, o_j, \ldots, o_{n1}\right\}$;
16: Calculate global features $H_u, H_m \leftarrow f\,(W \cdot O + g)$;
17: $\hat{x} = \max\{0,\ concat\ (H_u, H_m)\}$;
18: Calculate $\langle v_i, v_j \rangle$ of H_u and H_m according to Eq. (18);
19: Prediction value $\widehat{y} \leftarrow put\ \widehat{x}\ and\ w_{ij} = \langle v_i,\ v_j \rangle$ to Eq. (17);
20: Calculate value δ of RMSE based on $\widehat{y}$ and y;
21: **end for**;
22: **until** The rate of change of δ tends to be stable

to initialize and update W_0 and W_1 (Line 7). Consequently, we directly spliced the expression feature vector and the attention feature vector to form a new expression feature vector $F_{0,nr}$ and $F_{1,nr}$ with memory ability (Line 8).

(2) We utilize CNNs based on attention mechanism to convolve, pool and fully connect for $F_{0,nr}$ and $F_{1,nr}$, and extract user hidden features H_u and item hidden features H_m (Lines 9–16).

(3) We input the c_j of user and item into Eq. (5), and calculate the attention weight A_1 (Line 10); and then calculate the phrase-level attention weight $b_{0,r}$ of the entire user comment on each phrase in the item comment (the same is $b_{1,r}$) (Line 11). Finally, we use $b_{0,r}$, $b_{1,r}$ as the weight influence the c_j (Lines 12–13.

(4) We utilize the factorization machines to construct the association between users and items from the hidden features H_u and H_m of users and items respectively and achieve the user's rating prediction of the item according to the association (Lines 17–19). If the rate of change of the Root Mean Squared Error (RMSE) value tends to be smooth, the iteration terminates (Line 20).

In summary, our algorithm consists of a CNN, a Word-level attention mechanism, a Phrase-level attention mechanism, and an FM, so the time complexity of ACNN-FM algorithm is:

$$O = O_{cnn}\left(\sum_{f=1}^{D} n_{f-1} \cdot s_f^2 \cdot n_f \cdot m_l\right) + O_w\,(d \cdot n \cdot n) + O_p\left(t^2\right) + O_{fm}\,(L_h) \tag{20}$$

where O_{cnn} is the time complexity of convolutional neural network [36], f is the index of a convolutional layer, D is the depth (number of convolutional layers), n_l is the number of filters, s_l is the spatial size (length) of the filter, m_l is the spatial size of the output feature map. O_w is the time complexity of the word-level attention mechanism, it is known that d is the word embedding dimension and n is the sentence length, it can be seen that the time complexity of Eq. (5) is $O\left(n^2\right)$, and the time complexity of Eq. (6) is $O\,(d \cdot n \cdot n)$, so the final complexity is $O\,(d \cdot n \cdot n)$. O_p is the time complexity of the phrase-level attention mechanism, it is known that t is the feature length after convolution, so the time complexity of Eq. (5) is $O\left(t^2\right)$, and the time complexity of Eqs. (12)–(14) is $O\left(3t^2\right)$, and finally O_p is $O\left(t^2\right)$. O_{fm} is the time complexity of the factorization machine [34], L_h is the length of hidden features (H_u or H_m).

Table 2
The statistics of the experimental data sets.

Dataset	N_u	N_i	N_c	d(%)	$\overline{N_u}$	$\overline{N_i}$	$N_w(KB)$
Amazon Instant Video	5130	1685	37126	99.571	7	22	27450
Automotive	2928	1685	20473	99.585	7	12	13943
Patio, Lawn&Garden	1686	962	13272	99.182	8	14	14204
Musical Instruments	1429	900	10261	99.202	7	11	7272

6. Experiments

In this section we describe our experimental setup and provide an indepth discussions based on our obtained experimental results.

6.1. Datasets

The four group of data sets utilized in the experiment are based on the evaluation information of different industries in the Amazon shopping platform, which truly reflected users' preferences for the items [37,38].

Table 2 presents the statistics of the data sets, which includes different data levels, different fields, and different sparsity levels. Overall, the data sets comprehensively and objectively simulates the application scenarios of information overload. Among them, N_u represents the number of users, N_i is the number of items, N_c denotes the number of evaluations, $d = 1 - (N_c/(N_u \cdot N_i))$ refers to the sparseness of the data, $\overline{N_u}$ represents the average number of comments for a single user, $\overline{N_i}$ indicates the average number of comments for a single item, and N_w indicates the size of data sets.

In the training and testing process of the recommended model, we utilize the Holdout verification method [39], which can alleviate the over-fitting problem. The method randomly divides each data set into training set, validation set, and test set. The proportion of each part is 80%, 10%, and 10%. Among them, we utilize the training set to train the parameters of each recommendation model, utilize the validation set to evaluate the capability of the recommended model parameters after training, and finally evaluate the generalization error of the model on the test set.

6.2. Baselines and evaluation metric

In the evaluation process of the ACNN-FM, we need to select appropriate evaluation indexes and representative comparison models according to characteristic of the model.

On one hand, the recommendation accuracy is an important evaluation metric. Therefore, we utilize two common used evaluation indicators: Mean Absolute Error (MAE) and RMSE. The two indicators measure the accuracy of the recommendation results by calculating the error between the real score and the predicted

score. The smaller the values, the higher the recommendation accuracy.

$$MAE = \frac{1}{N}\sum_i |y_i - \widehat{y}_i| \tag{21}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_i (y_i - \widehat{y}_i)^2} \tag{22}$$

where N represents the total number of records in data sets, y_i indicates the true rating, and $\widehat{y}_i$ represents the predicted rating.

On the other hand, we have selected the following four comparison methods for better evaluation of our proposed approach in this paper.

(1) Non-negative Matrix Factorization (NMF) [40] is a matrix decomposition method in which all matrix elements conform to non-negative constraints. According to the principle of "local composes the whole", the method divides the complex matrix into two simplified non-negative sub-matrices, and then solves the non-negative sub matrices separately by simple iterative method.

(2) BCF [9] is an optimized collaborative filtering method. This method uses the principle that similar users have similar interests to discover the potential preferences of users for items. This method only needs to use the user's historical rating data, so it is simple and effective, and is the most successful recommendation method currently used.

(3) DeepCoNN [22] is an advanced hybrid recommendation method which combines the deep learning with the traditional recommendation model. The method firstly utilizes the information other than ratings construct the features of users and items through deep learning. Then it utilizes the content-based recommendation model to fuse features to achieve the recommendation.

(4) NARR [41] is a hybrid recommendation method based on machine learning, which increases the attention mechanism and improves the similarity between the user and the item comments to complete the rating prediction.

6.3. Parameters optimization

We determine the optimal hyper-parameters of each algorithm by repeated tests. NMF and BCF can obtain best recommended results when the number of latent factors is in the range of {10, 25, 50, 100, 150, 200}, the regularization parameters are in the range of {0.001, 0.01, 0.1, 1.0}, and the learning rate is in the range of {0.006, 0.005, 0.004, 0.003, 0.002, 0.001}.

Fig. 4 summarizes the RMSE of the ACNN-FM, NARR and DeepCoNN, when the number of convolution kernels is in the range of {10, 20, 50, 100, 150, 200, 300}, among them, the three methods get the optimal value when the number of convolution kernels is 100. Fig. 5 shows the RMSE of the ACNN-FM, NARR and the DeepCoNN when latent factors is in the range of {8,16,32,64,96,128, 256}, of which the best results are obtained when the number of latent factors is 64. In the training process of the above three methods, the learning rate is searched within the range of {0.0001, 0.0002, 0.0004, 0.0008, 0.005, 0.01, 0.02, 0.05}, and the batch size is tested in {50, 100, 150}. To prevent overfitting, we turn the dropout ratio in {0.1. 0.3, 0.5, 0.7, 0.9}. If the computational resources are insufficient, ACNN-FM method can increase the convolution layer to process the length of word embedding in the "Word-level attention mechanism" stage. Additionally, when the number of iterations is 25 (*i.e. epochs* = 25) in the training process of the ACNN-FM, NARR and the DeepCoNN, the loss function values of the model on the test set tend to be stable.

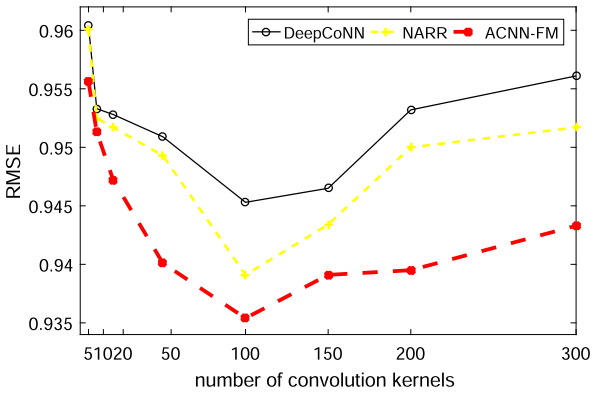

Fig. 4. RMSE for different convolution kernel quantities.

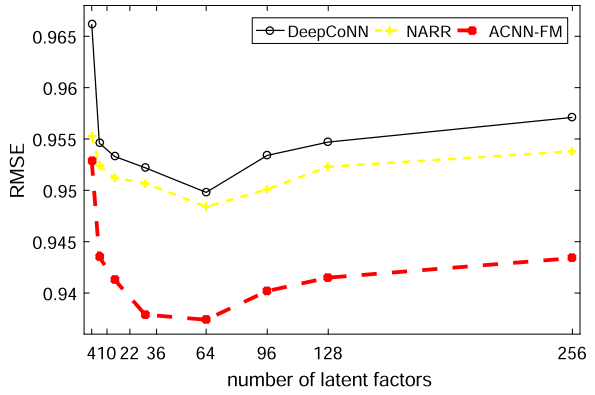

Fig. 5. RMSE for different latent factor quantities.

Table 3
The holistic data set of experimental results. * and ** denote the statistical significance for $p < 0.05$ and $p < 0.01$, respectively, compared to the best baseline.

Dataset	Indicator	NMF	BCF	DeepCoNN	NARR	ACNN-FM
Video	MAE	0.8429	0.7515	0.7039	0.6946	*0.6744***
	RMSE	1.1156	1.0536	0.9727	0.9601	**0.9564***
Automotive	MAE	0.847	0.6715	0.6242	0.6185	*0.5967***
	RMSE	1.1002	0.9922	0.8981	0.8956	**0.8925***
Patio	MAE	0.8964	0.783	0.7602	0.7527	*0.7205***
	RMSE	1.1611	1.0693	1.0163	1.0117	**0.997***
Musical	MAE	0.8518	0.6692	0.6589	0.6539	*0.6004***
	RMSE	1.0957	0.9924	0.9513	0.9509	**0.9363****

6.4. Performance evaluation

6.4.1. The effect on recommendation diversity

This paper considers the differences in the application scenarios of the recommendation system. For each data set, we extract test data sets from three different groups: the holistic data set, the cold start user group, and the long tail item group. Among them, the holistic data set is all data in the entire data set; the cold start user group is users' data of the evaluation quantity from 1 to 6; the long tail item group is the data of all the inventory items (the number of unpopular items accounts for about 80% of the total) [42]. The specific experimental results are as follows:

Table 3 illustrates the comparative experiment results of the holistic data set. The experimental results demonstrate that the RMSE of the ACNN-FM, NARR and the DeepCoNN are significantly higher than that of the NMF and the BCF. The RMSE and MAE of the ACNN-FM are 4.28% and 8.19% better than that of NARR. In

Table 4
Cold start user group of experiment results. * and ** denote the statistical significance for $p < 0.05$ and $p < 0.01$, respectively, compared to the best baseline.

Dataset	Indicator	NMF	BCF	DeepCoNN	NARR	ACNN-FM
Video	MAE	0.9164	0.7623	0.7181	0.7235	*0.681***
	RMSE	1.1926	1.0733	1.0017	1.0132	**0.9852****
Automotive	MAE	0.9645	0.7021	0.6341	0.6899	*0.5902***
	RMSE	1.2332	0.9939	0.9037	0.9395	**0.8863****
Patio	MAE	1.0226	0.832	0.8095	0.8571	*0.7701***
	RMSE	1.2917	1.1999	1.0902	1.1176	**1.0798****
Musical	MAE	0.9396	0.6899	0.6878	0.76	*0.6485***
	RMSE	1.1913	0.9653	0.9559	1.0529	**0.9453****

Table 5
The long tail item group of Experimental results. * and ** denote the statistical significance for $p < 0.05$ and $p < 0.01$, respectively, compared to the best baseline.

Dataset	Indicator	NMF	BCF	DeepCoNN	NARR	ACNN-FM
Video	MAE	1.0021	0.8662	0.7769	0.8509	*0.7588***
	RMSE	1.2692	1.1383	1.0439	1.1148	**1.0211****
Automotive	MAE	1.0059	0.6994	0.6461	0.713	*0.6026***
	RMSE	1.2712	0.9693	0.9386	0.9808	**0.9222****
Patio	MAE	1.0744	0.7922	0.7953	0.9041	*0.7295***
	RMSE	1.3387	1.0761	1.1041	1.1279	**1.0315****
Musical	MAE	0.9793	0.704	0.6608	0.7701	*0.5725***
	RMSE	1.2334	0.953	0.8572	1.0366	**0.8223****

addition, the ACNN-FM is superior to the NARR in the evaluation experiments of the high sparse data sets *Video* and *Automotive* (its sparsity is 99.571% and 99.585%). That is, the ACNN-FM has higher data utilization rate in the big data environment, and the recommended effect is obviously superior to other comparison methods.

Table 4 reports the comparative experiment results of the cold start user group. The experimental results show that with the decrease of user evaluation data, the recommendation accuracy of all methods decreases correspondingly, and the NARR recommendation accuracy decreases significantly, which is unexpectedly lower than the DeepCoNN. However, the ACNN-FM method maintains the best recommendation performance. It can be seen that compared with NARR, our attention mechanism can deal with the over fitting problem better, so that ACNN-FM has higher stability and can better alleviate the cold start problem.

Table 5 presents the comparative experiments results of the long tail item group. The long tail phenomenon is a common formulation of power law distribution, which shows that only a small number of products are sold well in the e-commerce platforms, and most products are forgotten. Unlike the cold start issue, a long tail item may have a large number of evaluations but a low sales volume. In these experiments, we sorted the products by sales volume from high to low, taking the last 70% of the data as the long tail item (The data ratio shows the advantage of algorithm performance when it is lower than 80% of the long tail distribution). Results demonstrate that the ACNN-FM achieves the best recommended performance on each data set compared to other recommended methods. Particularly, the ACNN-FM achieves the best recommendation performance and highest data utilization in the long tail phenomenon.

6.4.2. The effect on cold start for new users

The ability to recommend accurate items for a new user is one of the important performance indicators for measuring the processing ability of the cold start problem in the recommendation system. To further validate the performance of the ACNN-FM, we further conduct evaluations of cold-start issues. New users entering the system can easily construct the user's feature text by selecting tags, improving personal information, browsing popular products, and user consultation records. For this scenario, we utilize the title and descriptions of the item as the user's comments for the best favorite item, thus constructing a data set similar to the tuple *P*.

Fig. 6 shows the experimental results of an new user for the cold start problem. The experimental results show that the performance of the ACNN-FM method is optimal, and the ACNN-FM and NARR are significantly improved compared with the DeepCoNN. Therefore, the attention mechanism is greatly improved for text processing, that is to say, the ACNN-FM method based on multi-visual attention has the highest precision in extracting features from natural language text. Compared with the other comparison methods, the ACNN-FM has higher data utilization in solving data sparse problems and cold start problems for new users. It also shows that ACNN-FM can be easily extended to other application scenarios which are easy to construct user's feature text.

Table 6
String length analysis of comments.

Dataset	Data proportion	Minimum string length	Maximum string length
Video	0.8	71	863
	0.6	95	358
Automotive	0.8	88	1218
	0.6	96	442
Patio	0.8	1	2627
	0.6	1	1250
Musical	0.8	1	1256
	0.6	94	499
Average:		76	1418

6.4.3. The effect on length of comments

The main feature of the ACNN-FM is to improve the accuracy of feature extraction with CNNs by increasing the association between the comments of users and items during the training process. It can be speculated that the length of the comments has an influence on the recommended effect of this method. To further analyze the features above, we analyze the distribution of the comments number in each data set length, where we utilized logarithm to process the coordinate values. From Fig. 7, we can see that the distribution of the comments number under different length of comments is consistent with the long tail distribution.

To further analyze the distribution of comments number under different comments length, we count the comment number for each comments length, where *data ratio* refers to the reverse order of the comments number of each length of comments, and we select a certain proportion of data from high to low.

As shown in Table 6, the length of most of comments is from *76* to *1418*, and the recommended effect of evaluating different algorithms in this interval is the most representative. Therefore, we test each data set with a string length of *50* to *1500*. As shown in Fig. 8, it is obvious that the ACNN-FM method achieves the best recommended performance.

6.4.4. The effect on time cost

In practical applications, time cost is an important factor to measure the availability of recommended methods. The implementation process of the evaluation method in this paper includes the training phrase and the execution phrase. Therefore,

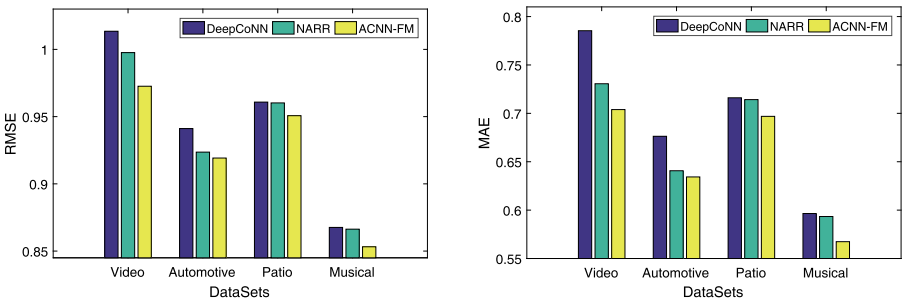

Fig. 6. Cold start evaluation for new user.

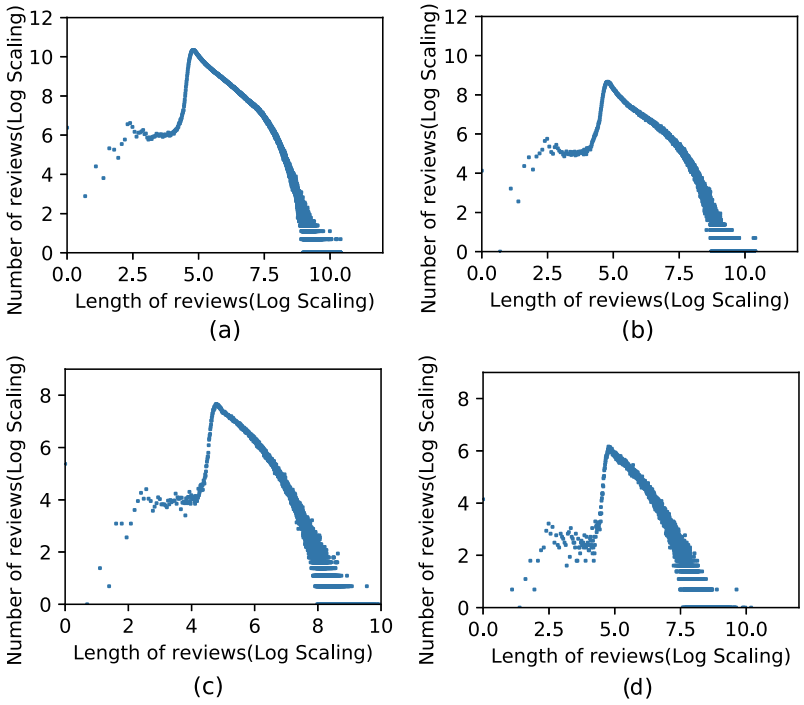

Fig. 7. The distribution of the comments number under different string length (using Log–Log Scaling): (a) Amazon Instant Video data set; (b) Automotive data set; (c) Patio, Lawn&Garden data set; (d) Musical Instruments data set.

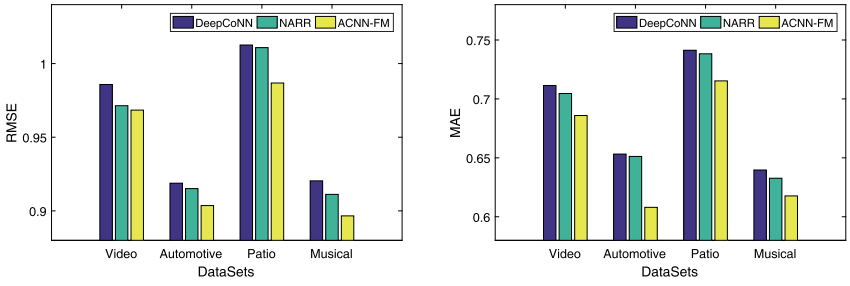

Fig. 8. The experimental result of a string length of "50–1500".

we choose the two evaluation metrics of training time and execution time. As shown in Table 7, as the amount of data increases, the execution time of the traditional recommendation method will become longer. However, the hybrid recommendation method based on machine learning has a longer training time than the traditional recommendation method, but the execution time is always kept within 0.6s. Furthermore, compared with the DeepCoNN, the NARR and ACNN-FM has a longer training time due to the increased attention mechanism, although the execution time remains basically unchanged. Consequently, we have achieved significant performance improvements in the case of smaller time cost.

7. Conclusions

It is shown that the rich comments written by a user could reveal behavior about the purchase and rating of them. Therefore, we can use the comments to solve the problems of data sparseness, cold start and feature extraction over-reliance on labor in traditional recommendation methods. Therefore, this paper

Table 7
Time cost evaluation of Experimental results.

Evaluation metrics	Dataset	Evaluation method				
		NMF	BCF	Deep-CoNN	NARR	ACNN-FM
Training time (s)	Video	1.92	1.4	1491	3299	4341
	Automotive	1.15	0.42	575	1018	1954
	Patio	0.82	0.12	757	1389	1882
	Musical	0.55	0.1	324	626	1048
Execution time (s)	Video	0.06	0.63	0.02	0.015	0.04
	Automotive	0.03	0.12	0.02	0.03	0.51
	Patio	0.02	0.11	0.02	0.32	0.07
	Musical	0.01	0.07	0.02	0.02	0.05

proposes a hybrid recommendation method based on deep learning (ACNN-FM), which utilizes two parallel CNN models based on attention mechanism to extract rich expression features of users and items from comments, and then use ratings to mine the association of these features, ultimately predicts the user's preferences for new items.

In this paper, extensive experiments demonstrate the recommended effect of ACNN-FM in various data sets has been significantly improved. In addition, in the extreme environment of data sparseness and cold start, the recommendation accuracy of ACNN-FM is superior to other recommendation methods. ACNN-FM can utilize text to build user-item associations. The approach also can be applied to a scenario in which a user has relevant text with a item, such as information retrieval and other scenarios. Therefore, our future work is to develop a model for improving the accuracy of feature extraction by extending the neural network structure.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61872228, 61802240, 61702317, 61877037, 61602289, 61862056), the Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shaanxi Province (Grant No. 2017024), the Natural Science Basic Research Plan in Shaanxi Province of China (Grant Nos. 2019JM-379, 2017JM6060), the Guangxi Natural Science Foundation, China (No. 2017GXNSFAA198148).

References

[1] A. Segatori, F. Marcelloni, W. Pedrycz, On distributed fuzzy decision trees for big data, IEEE Trans. Fuzzy Syst. 26 (1) (2018) 174–192.
[2] J.G. Reinsel, The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east, in: IDC IView:IDC Analyze the Future, 2012, pp. 1–12.
[3] C. Song, A.L. Barabási, Limits of predictability in human mobility, Science 327 (5968) (2010) 1018–1021.
[4] Y. Lin, X. Wang, F. Hao, L. Wang, L. Zhang, R. Zhao, An on-demand coverage based self-deployment algorithm for big data perception in mobile sensing networks, Future Gener. Comput. Syst. 82 (2018) 220–234.
[5] X. Qin, X. Wang, L. Wang, Y. Lin, X. Wang, An efficient probabilistic routing scheme based on game theory in opportunistic networks, Comput. Netw. 149 (2019) 144–153.
[6] F. Hao, S. Li, G. Min, et al., An efficient approach to generating location-sensitive recommendations in ad-hoc social network environments, IEEE Trans. Serv. Comput. 8 (3) (2015) 520–533.
[7] Y. Shi, M. Larson, A. Hanjalic, Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges, ACM Comput. Surv. 47 (1) (2014) 1–45.
[8] S. Wang, J. Tang, Y. Wang, H. Liu, Exploring implicit hierarchical structures for recommender systems, in: IJCAI, 2015, pp. 1813–1819.
[9] Y. L.Huang, Y.Liu, Deep learning based recommender systems, Chinese J. Comput. 40 (40) (2017) 1–30.
[10] P. Rodrìguez, M.A. Bautista, J. Gonzàlez, S. Escalera, Beyond one-hot encoding: Lower dimensional target embedding, Image Vis. Comput. 75 (2018) 21–31.
[11] J. Wei, J. He, K. Chen, Y. Zhou, Z. Tang, Collaborative filtering and deep learning based recommendation system for cold start items, Expert Syst. Appl. 69 (2016) 29–39.
[12] R. Wang, C.Y. Chow, Y. Lyu, V.C.S. Lee, S. Kwong, Y. Li, J. Zeng, Taxirec: recommending road clusters to taxi drivers using ranking-based extreme learning machines, IEEE Trans. Knowl. Data Eng. 30 (3) (2018) 585–598.
[13] F. Hao, D. Park, X. Yin, et al., A location-sensitive over-the-counter medicines recommender based on tensor decomposition, J. Supercomput. 75 (4) (2019) 1953–1970.
[14] H. Yin, W. Wang, H. Wang, et al., Spatial-aware hierarchical collaborative deep learning for POI recommendation, IEEE Trans. Knowl. Data Eng. 29 (11) (2017) 2537–2551.
[15] H.L. Liu, T. Taniguchi, Y. Tanaka, K. Takenaka, T. Bando, Visualization of driving behavior based on hidden feature extraction by using deep learning, IEEE Trans. Intell. Transp. Syst. 18 (9) (2017) 2477–2489.
[16] D. Silver, A. Huang, C.J. Maddison, et al., Mastering the game of Go with deep neural networks and tree search, Nature 529 (7587) (2016) 484.
[17] W. Ma, Y. Wu, M. Gong, C. Qin, S. Wang, Local probabilistic matrix factorization for personal recommendation, in: Computational Intelligence and Security, CIS, 2017 13th International Conference on, IEEE, 2017, pp. 97–101.
[18] C. Chao, A. Zare, H.N. Trinh, G.O. Omotara, J.T. Cobb, T.A. Lagaunne, Partial membership latent dirichlet allocation for soft image segmentation, IEEE Trans. Image Process. 26 (12) (2017) 5590–5602.
[19] Z. Xu, L. Chen, Y. Dai, G. Chen, A dynamic topic model and matrix factorization based travel recommendation method exploiting ubiquitous data, IEEE Trans. Multimed. 19 (8) (2017) 1933–1945.
[20] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.
[21] Y.X. Peng, W.W. Zhu, Y. Zhao, C.S. Xu, Q.M. Huang, H.Q. Lu, Q.H. Zheng, T.J. Huang, W. Gao, Cross-media analysis and reasoning: advances and directions, Front. Inf. Technol. Electron. Eng. 18 (1) (2017) 44–57.
[22] L. Zheng, V. Noroozi, P.S. Yu, Joint deep modeling of users and items using reviews for recommendation, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM, 2017, 425–434.
[23] V.C. Tran, N.T. Nguyen, H. Fujita, D.T. Hoang, D. Hwang, A combination of active learning and self-learning for named entity recognition on Twitter using conditional random fields, Knowl.-Based Syst. 132 (2017) 179–187.
[24] E. Li, J. Xia, P. Du, C. Lin, A. Samat, Integrating multilayer features of convolutional neural networks for remote sensing scene classification, IEEE Trans. Geosci. Remote Sens. 55 (10) (2017) 5653–5665.
[25] W. Wang, J. Shen, Deep visual attention prediction, IEEE Trans. Image Process. 27 (5) (2018) 2368–2378.
[26] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, T.-S. Chua, Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2017, pp. 335–344.
[27] X. Luo, D. Wang, M. Zhou, H. Yuan, Latent factor-based recommenders relying on extended stochastic gradient descent algorithms, IEEE Trans. Syst. Man Cybern. Syst. (2019).
[28] X. Luo, M. Zhou, Z. Wang, Y. Xia, Q. Zhu, An effective scheme for QoS estimation via alternating direction method-based matrix factorization, IEEE Trans. Serv. Comput. (2016).
[29] L. Li, S. Tang, Y. Zhang, L. Deng, Q. Tian, Gla: Global-local attention for image description, IEEE Trans. Multimed. 20 (3) (2018) 726–737.
[30] X. Wang, L. Yu, K. Ren, G. Tao, W. Zhang, Y. Yu, J. Wang, Dynamic attention deep model for article recommendation by learning human editors' demonstration, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 2051–2059.
[31] X. W. Z. e. a. Luo, J. Sun, Symmetric and non-negative latent factor models for undirected, high dimensional and sparse networks in industrial applications, IEEE Trans. Ind. Inf. 13 (6) (2017) 3098–3107.
[32] X. L. S. e. a. Luo, M.C. Zhou, An inherently non-negative latent factor model for high-dimensional and sparse matrices from industrial applications, IEEE Trans. Ind. Inf. 14 (5) (2018) 2011–2022.
[33] X. L. S. e. a. Luo, M.C. Zhou, Incorporation of efficient second-order solvers into latent factor models for accurate prediction of missing QoS data, IEEE Trans. Cybern. 48 (4) (2018) 1216–1228.
[34] S. Rendle, Factorization machines with libfm, ACM Trans. Intell. Syst. Technol. 3 (3) (2012) 1–22.
[35] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization. arXiv preprint, arXiv:1412.6980, 2014.
[36] K. S. J. He, Convolutional neural networks at constrained time cost, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5353–5360.
[37] R. He, J. Mcauley, Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering, in: International Conference on World Wide Web, 2016, pp. 507–517.

[38] J. McAuley, C. Targett, Q. Shi, A. Van Den Hengel, Image-based recommendations on styles and substitutes, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 43–52.

[39] M.F. Henrik Brink, Joseph W. Richards, Real-world machine learning: Model evaluation and optimization, https://www.developer.com/mgmt/real-world-machine-learning-model-evaluation-and-optimization.html.

[40] A. Hernando, J. Bobadilla, F. Ortega, A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model, Knowl.-Based Syst. 97 (2016) 188–202.

[41] C. Chen, M. Zhang, Y. Liu, S. Ma, Neural attentional rating regression with review-level explanations, in: Proceedings of the World Wide Web Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2018, pp. 1583–1592.

[42] E. Brynjolfsson, Y. Hu, D. Simester, Goodbye Pareto principle, hello long tail: The effect of search costs on the concentration of product sales, Manage. Sci. 57 (8) (2011) 1373–1386.