Knowledge-Based Systems 213 (2021) 106687

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Accurate and Explainable Recommendation via Hierarchical Attention Network Oriented Towards Crowd Intelligence

Chao Yang [a,*], Weixin Zhou [a], Zhiyu Wang [a], Bin Jiang [a], Dongsheng Li [b], Huawei Shen [c]

[a] College of Computer Science and Electronic Engineering, Hunan University, Lushan Road (S), Yuelu District, Changsha, China
[b] Microsoft Research Asia, No. 77 Hongcao Road, Xuhui District, Shanghai, China
[c] CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 19 June 2020
Received in revised form 11 December 2020
Accepted 13 December 2020
Available online 25 December 2020

Keywords:
Crowd intelligence
Explainable recommendation
Hierarchical attention
Review representation
Recommender system

ABSTRACT

Review-based recommendation algorithms can alleviate the data sparsity issue in collaborative filtering by combining user ratings and reviews in model learning. However, most existing methods simplify the feature extraction process from reviews by assuming that different granularities of information (e.g., word, review, and feature) are equally important, which cannot optimally leverage the most important information and thus achieves suboptimal recommendation accuracy. Besides, many existing works directly regard text features as users or items representations, which may not be enough to make precise representations due to the large amount of redundant information in reviews. To tackle the two problems mentioned above, we propose a deep learning-based method named **H**ierarchical **A**ttention **N**etwork Oriented Towards **C**rowd **I**ntelligence (HANCI). First, HANCI replaces the commonly-used topic models or CNN text processor with an RNN text processor in review feature extraction, which can fully exploit the advantages of the sequential dependencies of reviews by using the whole hidden layers of the bidirectional LSTM as outputs. Second, HANCI weighs the importance of features guided by crowd intelligence to more accurately represent each user on each item, and vice versa. Third, HANCI utilizes a hierarchical attention network based on multi-level review text analysis to extract more precise user preferences and item latent features, so that HANCI can explore the importance of words, the usefulness of reviews and the importance of features to achieve more accurate recommendation. Extensive experiments on three public datasets show that HANCI outperforms the state-of-the-art review-based recommendation algorithms in accuracy and meanwhile provides insightful explanations.

1. Introduction

As an effective solution to alleviate the information overload problem, recommender systems have recently become a popular application in both academia and industry. In many applications, the recommender systems try to predict a targeted user's ratings on unrated items, and then recommend items with high predicted ratings to minimize user efforts and thus increase user satisfaction. One of the most widely known issues in recommender systems is the data sparsity issue [1–6], i.e., users usually have ratings on a small amount of items, which makes it challenging to learn effective recommendation models. In recent years, researchers proposed review-based methods [2,4,6], which can fuse user–item rating features and review text features to alleviate the data sparsity issue. Review data can reflect the user's preference on each rated product and its specific details, and in addition can be regarded as a carrier of important influential information which will influence the behaviors of other potential users.

Existing review-based recommendation algorithms mainly adopt two types of methods to extract review features. The first type of methods adopts topic modeling [7–10] to extract features from review texts (e.g., in HFT [9] and A^{3}NCF [10]). But the bag of words approach used in topic modeling faces the challenges of oversimplifying the problem by ignoring the frequency and intensity of information and failing to properly deal with long text data. To address the limitations of the topic modeling-based methods, the second type of methods adopts CNN-based text extraction. Zheng et al. [11] propose DeepCoNN to capture better potential factors of users and items by utilizing two parallel CNN text processors to learn latent features of user behaviors and items from reviews and a shared regression layer that allows item representation and user behavior to interact. However, the limitation of this method is that it directly applies the user's review on the item when predicting a user's rating of a target item. As the user's review on the item cannot be known in advance, thus the construction of the testing data is unreasonable.

* Corresponding author.
E-mail address: yangchaoedu@hnu.edu.cn (C. Yang).

https://doi.org/10.1016/j.knosys.2020.106687
0950-7051/

In order to solve this problem, Catherine and Cohen propose a DeepCoNN's extension model TransNet [12] which additionally designs a target network based on the source network. The model is trained in the source network where those reviews published on the target item are removed in advance to maintain the data consistency. Although DeepCoNN and TransNet have improved the performance of review-based recommendation, they still have a common limitation, i.e., the CNN text processors may lose position information because it only considers the position information of local convolutions by adopting the pooling operations. More importantly, the above methods mostly assume that all reviews are equally important to the representation of user or item. This assumption is not reasonable, because different reviews express a user's different preferences for the item and different characteristics of the item. Based on this, NARRE [13] claims that the usefulness of a review should be defined as whether it can provide detailed information about an item and help users make their purchasing decisions easier. Specifically, it introduces a neural attention network to interact with the pair of target-review (target represents for user or item) and selects highly-useful reviews to improve the performance of the recommender system. However, merely considering the usefulness of reviews is not sufficient. It is common sense that the contribution of each word in a review is different and some uninformative words will affect the performance of feature extraction. Thus, it is necessary to model words' contribution distribution which reflects the varying importance of words in the reviews. The importance of a word is defined as whether it describes the preferences of a user or the attributes of an item. To capture the importance distribution of words and reviews, MPCN [14] exploits a multi-pointer co-attention network to make deep word-level interaction between user and item and captures the contribution distribution of word-level and review-level. Attention mechanism can simplify the problem and reduce time complexity, but it is somewhat weak in terms of extracting features. In addition, it may face the risk of distorting the meaning of original sentences when only the words or phrases extracted from the reviews are used to explain the recommendation results [15,16]. It is also not specific enough to regard useful reviews [13] as explanations, because it is difficult for users to get the key points from the fragments of reviews. In summary, the recommendation accuracy can be improved by considering the reviews, but the ways that existing review-based recommendation methods utilizing review information are far from optimal. The reasons are summarized as follows: (1) The review-based recommendation methods seldom adequately extract features from reviews and most of them cannot accurately define the representation of user and item. (2) Only providing word-level or review-level explanations may distort the meaning of original sentences or not be specific enough.

To address the aforementioned issues, this paper proposes a new review-based recommendation algorithm named Hierarchical Attention Network Oriented Towards Crowd Intelligence (HANCI). Inspired by DeepCoNN, TransNet and NARRE, HANCI utilizes two parallel neural networks to jointly learn the potential features of users and items. One of the networks utilizes user-written reviews to simulate user preferences, and the other utilizes reviews on items to simulate item features. Unlike these methods, we take BLH (a bidirectional LSTM using the whole hidden layers as outputs is called BLH) as the text processor to extract features, process sequence data and capture dependencies. Afterwards, we are the first to propose the importance of features guided by crowd intelligence, which replaces text features in existing work with user opinion-aware features to represent items. We treat all reviews received by an item as crowed intelligence, which collects users' experiences about the item and can guide the purchase decisions of other users. The guidance of crowd intelligence is mainly reflected in the key features, which is obtained by the feature level attention. Specifically, the crowd intelligence features are defined as the features extracted from all reviews on an item, which are more comprehensive than the features that a user focuses on. Thus, we propose the notion of the focus of features, which uses feature level attention to extract the features that a user cares about from the crowd intelligence features as the representation of the user's personalized preference on the item. Furthermore, in order to fully extract text features and obtain the optimal expression of the item, we integrate the importance of words, the usefulness of reviews and the importance of features in the same deep neural network. Finally, HANCI automatically selects important words and useful reviews when predicting the rating to explain the recommendation results. Important words make it easy to quickly locate key information in a review. Meanwhile, using useful reviews as the context of important words can avoid misinterpreting the original meaning. We evaluate our model on three public datasets: Amazon Toy_and_Games, Amazon Kindle_Store, and Yelp-2017. Experiments have proved that HANCI can outperform the state-of-the-art methods in rating prediction task and have the ability to automatically select important words and useful reviews to provide explanations at word-level and review-level.

The goal of our work is to improve the accuracy of rating prediction and provide explanations for the recommended results in review-based recommendation. To this end, we design a reasonable and effective feature extraction method to enhance the strength of feature interaction. Besides, we define more efficient representations for user and item. The main contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to propose the importance of features guided by crowd intelligence for the recommendation task. By interacting between the individual user behavior and the crowd intelligence features, the method is able to extract the crowd intelligence features that a user cares about, and then leverage the crowd intelligence as guidance to improve recommendation accuracy.
- We propose a novel hierarchical attention framework that can more accurately extract the important words at word-level, the useful reviews at review-level and the important features at feature-level. Experiments show that integrating them within a unified model can significantly enhance reviews information representation.
- Extensive experiments performed on three benchmark datasets demonstrate the effectiveness of HANCI. As a side contribution, we will publicly release our source code.[1]

2. Related work

2.1. Review-based recommendation

In recent years, the most prominent method which has been widely applied in the recommender system is collaborative filtering (CF) [17–20]. Since the main participants in CF are active users, which makes recommendation performance significantly lower when the rating matrix is very sparse. This problem is known as data sparsity in CF. Several works attempt to tackle the issue of data sparsity, for example, Parvin et al. [21] propose to integrate users' social trust statements as additional information into the recommendation model. Furthermore, some studies [22] have proved that it can significantly boost the recommendation accuracy by incorporating reviews into the collaborative filtering

[1] https://github.com/zhouweixin/hanci.

algorithm. For example, Shen et al. [23] incorporate the ratings and reviews into a probabilistic matrix factorization framework for prediction. Wang et al. [24] propose a convolutional matrix factorization framework to better model the item's reviews.

Meanwhile, many scholars have proposed various effective methods to extract features from the review texts. For example, Cheng et al. [10] design a new topic model to process review texts and achieve better representation of user preferences and item characteristics. Cheng et al. [25] apply topic models to extract aspect information automatically from reviews, which enhances the representation of users and items. Ling et al. [9] propose a novel *ratings meet reviews* algorithm, which improves prediction accuracy via applying topic modeling techniques on review texts and aligning the topics with the ratings' dimensions. Qiu et al. [26] present a novel latent factor model, which combines ratings and review texts to improve the task of rating prediction. In the light of this, we emphasize on the in-depth modeling of the review texts. Specifically, we introduce a recurrent neural network text processor to take advantage of the powerful information mining and expression capabilities of neural networks.

2.2. Deep learning for recommendation

Recently, considerable kinds of literature have grown up and have utilized deep learning in various natural language processing tasks, such as question answering [27,28], sentiment analysis [29], information retrieval [30,31] and text summarization [32]. Deep learning technology has also been widely used in review-based recommender system. For example, DeepCoNN [11] constructs two parallel neural networks and adopts a CNN text processor to process review texts. It utilizes the advantages of the convolutional neural network at extracting features and then adopts a shared regression layer to interact item representation with user behavior, which can capture better potential factors of users and items. However, it violates the consistency of data during the testing phase via using the pair of user–item review. Catherine et al. [12] extend the DeepCoNN model with an additional target network to generate the pair of user–item review features, which learns to transform the latent representations of user and item into that of their pair-wise review. Chen et al. [33] design a time-aware gated recurrent unit to model user dynamic preferences, and profile an item by its review feature extraction based on sentence-level convolutional neural network.

Meanwhile, deep learning technology has also shown their superiorities in the explanation task of recommender system. For example, Chen et al. [13] exploit the CNN text processor to process each review and introduce a novel attention mechanism to give review-level explanations. He et al. [34] explore the aspects in reviews to improve top-N recommendation and provide some words or phrases as the word-level or phrase-level explanations for recommendation results. Hou et al. [35] develop a semantic extraction network and fine-grained preferences attention module to explain the recommendation reason through intuitive visual attribute semantic (such as neckline, heel height, skirt length) highlights in a personalized manner. Chen et al. [36] propose a hierarchical sequence-to-sequence model to generate free-text natural language explanations for personalized recommendation.

Although the methods mentioned above have achieved good results, they still face the limitations of insufficient feature extraction and difficulty in accurately expressing users/items. To this end, we design a crowd intelligence-guided hierarchical attention network to fully mine the crowd intelligence features from textual reviews. Besides, we propose the importance of features module to extract the subset of crowd intelligence features that a user focuses on to accurately express users/items.

2.3. Attention mechanism for recommendation

With the introduction of self-attention [37], the attention mechanisms are now in the high tide of development and widely used, such as self-attention [38], hard-attention [39], soft-attention [40,41], multi-head attention [37] and so on. Attention is favored by scholars for its simplicity and efficiency for the following reasons. First, attention can replace the convolutional neural network (CNN) and recurrent neural network (RNN) feature extraction to reduce time complexity. For example, Vaswani et al. [37] propose a new text message processing transformer, which is based on the attention mechanism and dispenses with recurrence and convolutions entirely. Yang et al. [42] propose a representation learning network with attention and gate mechanism to learn the user embedding, item embedding and list embedding simultaneously to improve recommendation performance. Chen et al. [43] propose a hierarchical co-attentive selector to effectively extract the review-level and concept-level knowledge of users and items. Second, attention can combine with a deep network to distinguish the importance of features [40]. For example, Tay et al. [14] apply attention mechanisms to process review text and extracts word-level attention and review-level attention to improve the performance of the recommender system. Cong et al. [44] propose a hierarchical attention-based network to distinguish the importance of reviews at both word-level and review-level automatically. Yun et al. [45] propose a self-attentive integration network, which is used in the feature-level interaction layer to effectively consider interactions between multiple features. Finally, attention can be utilized to fuse multiple features [41]. For example, Chen et al. [13] utilize attention mechanism to interact with reviews and users, extracting the usefulness of reviews, and further enhancing the recommendation quality. Wu et al. [46] utilize convolution operations and attention mechanism to extract the user–item relevant features by jointly considering their corresponding reviews. Cheng et al. [2] propose a multi-modal aspect-aware topic model based on attention mechanism to interact reviews and item images.

Inspired by the success of attention mechanism, we design a hierarchical attention network based on the multi-level analysis of review text. To fully extract features, the hierarchical attention network integrates word-level, review-level and feature-level attention to capture the importance of words, the usefulness of reviews and the importance of features.

3. Preliminaries

3.1. Definition and notation

The input of our model consists of the user set U, the item set V, and the review set C. We define that each user k's id is id_k^U, each item j's id is id_j^V, and the review written by user k on item j is c_{kj}. In order to facilitate the description of the intermediate process of feature extraction, we define text feature and id feature. The text feature extracted from the review c_{kj} via the text processor is represented by F_{kj}^C. The id feature learned from user k and item j by id embedding are denoted by F_k^U and F_j^V, respectively. The Z_u^U and Z_v^V are used to represent the user u's and item v's crowd intelligence features. The operation of id embedding and extraction of crowd intelligence feature are described in Algorithm 1. Given a user u and an item v, the model can predict the rating $\hat{R}_{u,v}$ that reflects how much user u likes item v. The mathematical notations used in this paper are summarized in Table 1.

Algorithm 1 The operation of id embedding and extraction of crowd intelligence feature.

Input: The user set U; The item set V; The review set C;
Output: Each user k's and each item j's id features id_k^U and id_j^V; The user u's and item v's crowd intelligence features Z_u^U and Z_v^V;

for *each* $k \in U$ **do**
  $id_k^U = U.index(k)$
  $F_k^U = id\_embedding(id_k^U)$
end for
for *each* $j \in V$ **do**
  $id_j^V = V.index(j)$
  $F_j^V = id\_embedding(id_j^V)$
end for
for *each review* $c_{kj} \in C, k \in U, j \in V$ **do**
  $F_{kj}^C = blh\_text\_processor(c_{kj})$
end for
for *user* u*'s each review feature* $F_{uj}^C, j \in V$ **do**
  $ca_{uj} = review\_level\_attention(F_{uj}^C, id_j^V)$
  $Z_u^U = Z_u^U + F_{uj}^C * ca_{uj}$
end for
for *item* v*'s each review feature* $F_{kv}^C, k \in U$ **do**
  $ca_{kv} = review\_level\_attention(F_{kv}^C, id_k^U)$
  $Z_v^V = Z_v^V + F_{kv}^C * ca_{kv}$
end for

Table 1
Notations.

Symbols	Definitions and descriptions
id_u^U	User u's id
id_v^V	Item v's id
c_{uv}	User u's review on item v
F_u^U	Id embedding feature of id_u^U
F_v^V	Id embedding feature of id_v^V
F_{uv}^C	Text feature extracted from c_{uv}
Z_u^U	The crowd intelligence representation of user u
Z_v^V	The crowd intelligence representation of item v
q_u	User u's preference feature embedding based on id_u^U
p_v	Item v's preference feature embedding based on id_v^V
X_u	User u's latent feature
Y_v	Item v's latent feature
$R_{u,v}$	The ground truth of user u's rating on item v
$\hat{R}_{u,v}$	User u's predicted rating on item v

3.2. BLH text processor

Before introducing HANCI, we first briefly describe the BLH text feature processor. We define BLH as a bidirectional LSTM using the whole hidden layers as outputs. For example, CAQT [40] which adopts BLH and self-attention has achieved significant achievements in terms of extracting text features. In this paper, we use the same method in [40] as the text processor. Fig. 1 shows the architecture of the BLH and self-attention text processor.

As can be seen from Fig. 1, the first layer is an embedding layer. The word embedding function $f : M \rightarrow \mathbb{R}^d$ maps each word in a review to a d-dimensional vector and then converts the review text into an embedding matrix with a fixed length n (padding zeros or truncating if necessary to handle length changes). It can be any pre-trained word embedding, such as word2vec[2] [47] trained on the GoogleNews corpus, or

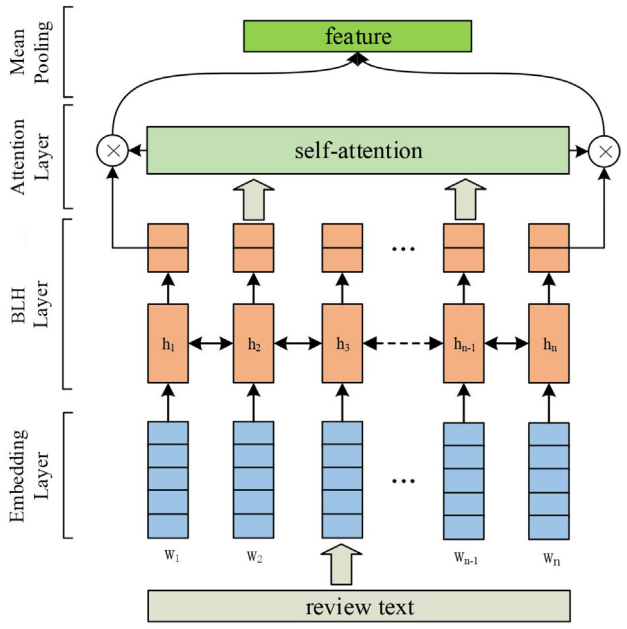

Fig. 1. The BLH text processor architecture.

GloVe[3] [48] trained on Wikipedia. Before the experiment, we conduct experiments using the word embeddings trained by GloVe and word2vec to initialize our proposed model, and the experimental results show that GloVe performs better. Therefore, we choose the latter and set the word embedding dimension as $d = 300$. The embedding matrix of a review is described as follows:

$$T = (w_1, w_2, \ldots, w_i, \ldots, w_n), \tag{1}$$

where w_i is a d-dimensional vector of the ith word of the review text, $w_i \in \mathbb{R}^d$, $T \in \mathbb{R}^{n \times d}$.

The second layer is the BLH layer. The number of hidden layer neurons is m, and the output of the bidirectional hidden layer corresponds to concatenate. The final output is as follows:

$$H = BLH(T) = (h_1, h_2, \ldots, h_i, \ldots, h_m), \tag{2}$$

where h_i is the output of the ith hidden layer of BLH, $h_i \in \mathbb{R}^{2h}$, $H \in \mathbb{R}^{m \times 2h}$.

The third layer is a self-attention layer. The attention is obtained by executing H linear transformation twice. Unlike the operations in [40], we do not utilize the softmax function, because it has the flaw of "push all pull one", leading excessive attention to one. Following the same way in [40], The self-attention module takes the whole LSTM hidden states H as input, and outputs a vector of weights a:

$$a = w_{t2} tanh(W_{t1} H^T), \tag{3}$$

where $W_{t1} \in \mathbb{R}^{d_a \times 2h}$, $w_{t2} \in \mathbb{R}^{d_a}$ and $a \in \mathbb{R}^n$. The attention weight a usually focuses on a specific part of the sentence, like a special set of related words or phrase. So it may reflect an aspect of the semantics in a review. However, there can be multiple aspects in a review that together forms the overall semantics of the whole review, especially for long sentences. Thus we need to perform multiple hops of attention weights to present the semantic of the whole review. Specifically, we extend the w_{t2} into a r-by-d_a matrix, note it as W_{t2}. The attention matrix A and the review

[2] https://code.google.com/archive/p/word2vec.

[3] https://nlp.stanford.edu/projects/glove.

embedding F are obtained by Eqs. (4) and (5):

$$A = (a_1, a_2, \ldots, a_r), \tag{4}$$

$$F = AH, \tag{5}$$

where $A \in \mathbb{R}^{r\times n}$, $F \in \mathbb{R}^{r\times 2h}$.

The fourth layer is a pooling layer. The review feature vector F^C is obtained through the mean-pooling operation.

$$F^C = mean\_pooling(F), \tag{6}$$

where $F^C \in \mathbb{R}^{2h}$.

3.3. Latent factor model

The latent factor model (LFM) is an algorithm based on the matrix factorization technique. In this paper, we take the neural latent factor model as the classifier to predict rating inspired by [13],

$$\hat{R}_{u,v} = q_u p_v^T + b_u + b_v + b_g \tag{7}$$

where $\hat{R}_{u,v}$ represents the predicted rating of the user u for the item v, q_u expresses the user u's inherent features, p_v denotes the item v's inherent features, b_u represents the user u's bias, b_v expresses the bias of the item v, and b_g is the global bias.

4. Proposed method

In this section, we will give a detailed introduction of HANCI. First, we will describe the overall architecture of the model. Then, we will introduce hierarchical attention, which includes the importance of words, the usefulness of reviews and then the importance of features guided by crowd intelligence successively. After that, we will explain the prediction layer, which is a neural latent factor model designed for rating prediction.

4.1. Overview of HANCI

The goal of HANCI is to predict a rating for a given pair of user u and item v. To achieve this task, we propose a hierarchical attention network oriented towards crowd intelligence to build the models of the user u and item v, and predict rating by the neural latent factor model. The architecture of the proposed HANCI model is shown in Fig. 2. The model consists of two parallel neural networks, one is for user modeling (Net_u) while the other is for item modeling (Net_v). The input of HANCI consists of user u's id id_u^U, item v's id id_v^V, user u's review set $(c_{u1}, c_{u2}, \ldots, c_{uj})$, the id set of items $(id_1^V, id_2^V, \ldots, id_j^V)$ that user u has commented on, item v's review set $(c_{1v}, c_{2v}, \ldots, c_{kv})$ and the id set of users $(id_1^U, id_2^U, \ldots, id_k^U)$ who has reviewed on the item v. Since Net_u and Net_v are symmetrical, the operation of Net_u is same as Net_v. Thus, we only explain the process of Net_v in detail. The output of the model is $\hat{R}_{u,v}$. In Net_v, first, the text feature $(F_{1v}^C, F_{2v}^C, \ldots, F_{kv}^C)$ are extracted from $(c_{1v}, c_{2v}, \ldots, c_{kv})$ by the BLH text processor. Second, the id feature q_u, p_v, $(F_1^U, F_2^U, \ldots, F_k^U)$ and $(F_1^V, F_2^V, \ldots, F_j^V)$ are obtained by id embedding operation. Third, the review-level attention mechanism is used to interact $(F_{1v}^C, F_{2v}^C, \ldots, F_{kv}^C)$ with $(F_1^U, F_2^U, \ldots, F_k^U)$ to obtain crowd intelligence representation Z_v^V of the item v. Fourth, the feature-level attention is performed to interact Z_v^V with q_u to get the final representation Y_v of the item v. Similarly, the Net_u can get the final representation X_u of user u. Finally, q_u, X_u, p_v and Y_v are jointly fed into the latent factor model to generate a rating $\hat{R}_{u,v}$.

4.2. Hierarchical attention network

It is known that the importance of words in each review, the contribution of reviews for each user or item, and the attention of each user or item on features are different. In light of this, we design a hierarchical attention network guided by crowd intelligence which comprehensively considers all the factors mentioned above.

4.2.1. Word-level attention

As we all know, the importance of different words varies. To strengthen important words and filter out uninformative words, we exploit the word-level attention to distinguish the importance distribution of all words. We treat each review separately with the BLH text processor combined with self-attention which is detailed presented in Section 3.2. Specifically, first, a piece of review is converted by word embedding into a matrix, which is represented as $c = (w_1, w_2, \ldots, w_n)$. Second, the review matrix c is inputted to the BLH text processor, and the output is $H = (h_1, h_2, \ldots, h_n)$. Third, It can capture semantics of review as much as possible by extracting r kinds of attention weights $A = (a_1, a_2, \ldots, a_r)$ from H. Last, we compute the r weighted sums by multiplying A and H, and the review feature F^C is obtained by mean-pooling.

4.2.2. Review-level attention

Although important words can enable users to quickly grasp key points, it is easy to misinterpret the meaning of original sentences. Therefore, it is necessary to extract useful reviews as the context of important words. What is more, extracting useful reviews usually can find more information describing the characteristics of item or the preferences of user. For this purpose, we model the review-level attention by interacting the reviews text feature $(F_{1v}^C, F_{2v}^C, \ldots, F_{kv}^C)$ and reviewers' (who wrote the reviews) ids feature $(F_1^U, F_2^U, \ldots, F_k^U)$. For simplicity, we note $(F_{1v}^C, F_{2v}^C, \ldots, F_{kv}^C)$ and $(F_1^U, F_2^U, \ldots, F_k^U)$ as F_{*v}^C and F_*^U, where $F_{*v}^C \in \mathbb{R}^{k\times 2h}$ and $F_*^U \in \mathbb{R}^{k\times m}$. Specifically, the inputs of the review-level attention module are feature F_{*v}^C and F_*^U, and the output is reviews weights ca_v.

$$ca_v = softmax(W_{v3} relu(W_{v1} {F_{*v}^C}^T + W_{v2} {F_*^U}^T + b_v)) \tag{8}$$

where, $W_{v1} \in \mathbb{R}^{t\times 2h}$, $W_{v2} \in \mathbb{R}^{t\times m}$, $W_{v3} \in \mathbb{R}^{1\times t}$, $b_v \in \mathbb{R}^t$, $ca_v \in \mathbb{R}^{t\times 1}$ and t is the size of review-level attention.

The crowd intelligence representation of item v is obtained by applying the reviews weights ca_v to the reviews feature F_{*v}^C and then performing the sum-pooling operation.

$$Z_v^V = sum\_pooling(F_{*v}^C * ca_v) \tag{9}$$

where, $Z_v^V \in \mathbb{R}^{2h}$. In the same way, the review-level attention module takes the reviews text feature $F_{u*}^C = (F_{u1}^C, F_{u2}^C, \ldots, F_{uj}^C)$ of user u and the id features of reviewers $F_*^V = (F_1^V, F_2^V, \ldots, F_j^V)$ as input, and can get the crowd intelligence representation Z_u^U of user u.

$$ca_u = softmax(W_{u3} relu(W_{u1} {F_{u*}^C}^T + W_{u2} {F_*^V}^T + b_u)) \tag{10}$$

$$Z_u^U = sum\_pooling(F_{u*}^C * ca_u) \tag{11}$$

where, $W_{u1} \in \mathbb{R}^{t\times 2h}$, $W_{u2} \in \mathbb{R}^{t\times m}$, $W_{u3} \in \mathbb{R}^{1\times t}$, $b_u \in \mathbb{R}^t$, $ca_u \in \mathbb{R}^{t\times 1}$, $Z_u^U \in \mathbb{R}^{2h}$.

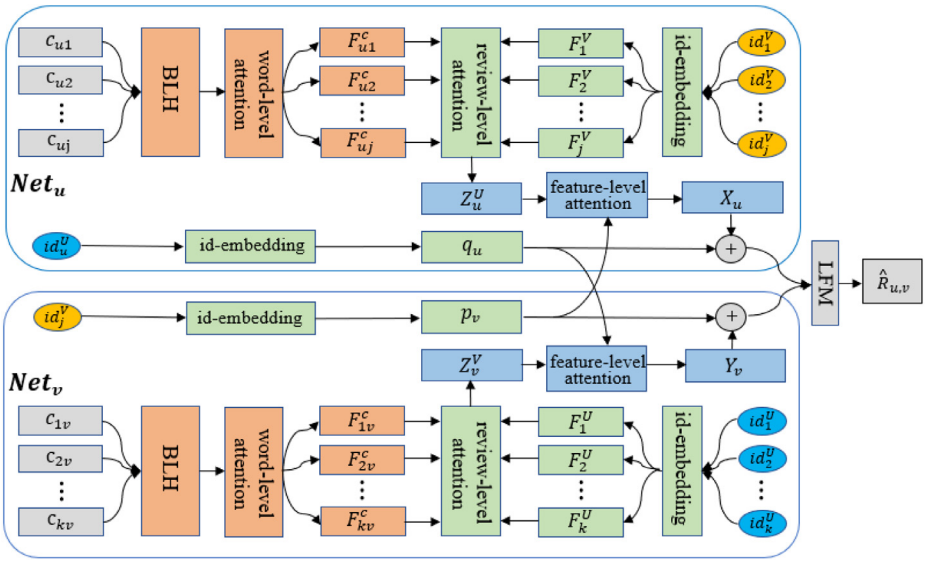

Fig. 2. The neural network architecture of HANCI.

4.2.3. Feature-level attention

The word-level and review-level attention can extract rich features from reviews, but presenting too many features can still cause selection problem. Thus, showing the features that the user cares about will be more in line with user expectations. Most existing methods regard the review features Z_u^U and Z_v^V as the representations of user u and item v, respectively. We argue that Z_u^U and Z_v^V are not the optimal representation of user u and item v. Taking Z_v^V as an example, we think that Z_v^V is the crowd intelligence features, which are extracted from all reviews of item v. Although Z_v^V contains the experience information of users $(id_1^U, id_2^U, \ldots, id_j^U)$, the crowd intelligence features are not the same as the features that user u cares about. Thus, when predicting the user u's rating on the item v, the most reasonable representation of the item v is not the crowd intelligence features Z_v^V, instead, the crowd intelligence features that the user u cares about.

In order to express the attention of user u on different features of the item v, we design a feature-level attention module via crowd intelligence features. Specifically, the feature-level attention module takes the user u's preference feature $q_u = F_u^U$ and the item v's crowd intelligence features Z_v^V as input, and first converts q_u to the common vector space of Z_v^V through two times linear transformation, and then the result is multiplied by Z_v^V. The calculation formula of the feature-level attention module is as follows:

$$Y_v = W_{v6}((W_{v5}(W_{v4}q_u^T + b_{v4}) + b_{v5}) * Z_v^V) + b_{v6}, \tag{12}$$

where $W_{v4} \in \mathbb{R}^{s\times m}$, $b_{v4} \in \mathbb{R}^s$, $W_{v5} \in \mathbb{R}^{2h\times s}$, $b_{v5} \in \mathbb{R}^{2h}$, $W_{v6} \in \mathbb{R}^{o\times 2h}$, $b_{v6} \in \mathbb{R}^o$, $Y_v \in \mathbb{R}^o$ and o is the size of user's and item's latent feature.

The final representation of the item v is Y_v, which are the features guided by the crowd intelligence and that user u cares about. It is more reasonable than the item v directly represented by Z_v^V in previous works. For the same reason, the features X_u of Net_u can be also obtained in the same way. We take X_u and Y_v as latent features of user u and item v respectively.

$$X_u = W_{u6}((W_{u5}(W_{u4}p_v^T + b_{u4}) + b_{u5}) * Z_u^U) + b_{u6}, \tag{13}$$

where $W_{u4} \in \mathbb{R}^{s\times m}$, $b_{u4} \in \mathbb{R}^s$, $W_{u5} \in \mathbb{R}^{2h\times s}$, $b_{u5} \in \mathbb{R}^{2h}$, $W_{u6} \in \mathbb{R}^{o\times 2h}$, $b_{u6} \in \mathbb{R}^o$ and $X_u \in \mathbb{R}^o$.

4.3. Prediction layer

In this paper, we apply HANCI for a recommendation task of rating prediction. To this end, we utilize a neural latent factor model of rating prediction proposed by [11]. Specifically, the latent factors of user u and item v are mapped to a shared space.

$$g = (q_u + X_u) \odot (p_v + Y_v), \tag{14}$$

$$\hat{R}_{u,u} = W_0 g + b_u + b_v + b_g, \tag{15}$$

where $\odot$ denotes the element-wise product of vector, $\hat{R}_{u,v}$ is the rating predicted by user u to item v, and b_u, b_v, b_g are user bias, item bias and global bias, respectively.

4.4. Optimization

In this paper, the rating prediction is specified as a regression task. We adopt the square loss [13,49,50] as the objective function:

$$Loss_r = \Sigma_{u,v\in U,V}(\hat{R}_{u,v} - R_{u,v})^2, \tag{16}$$

where $R_{u,v}$ is the true rating assigned by user u to item v.

To optimize the objective function, we use Adamax as the optimizer. Adamax is a variant of Adam that provides a simpler learning rate. Compared with Stochastic Gradient Descent (SGD), Adamax does not require manual adjustment of the learning rate and has a faster convergence. To alleviate overfitting, we adopt global regularization and local regularization. Global regularization is an L2 regularization for all parameters, and local regularization is an additional L2 regularization for the review-level attention module. Without local regularization, empty reviews will decrease the performance of id embedding. The joint loss function is defined in formula (17):

$$Loss = \Sigma_{u,v\in U,V}(\hat{R}_{u,v} - R_{u,v})^2 + \varepsilon\|\theta\|_2^2 + \lambda\|F\|_2^2, \tag{17}$$

where θ is the parameter set of the network, F is the parameter set of sentence-level attention network, ε, λ are the L2 regularization parameter for θ and F, respectively.

Finally, we get the sum of global regularization and local regularization to train HANCI, and experiments have verified the effectiveness.

Table 2
Statistical details of the datasets.

Dataset	# users	# items	# ratings/reviews	Sparsity
Toys_and_Games	19,412	11,924	167,597	99.93%
Kindle_Store	68,223	61,935	982,619	99.98%
Yelp_2017	199,445	119,441	3,072,129	99.99%

5. Experiments and results

5.1. Datasets, experimental setup and baselines

To validate the effectiveness of our proposed HANCI model, we conduct experiments to answer the following questions:
RQ 1: What is the overall performance of our proposed HANCI model compared with other benchmark methods?
RQ 2: Can the BLH text processor help improve the performance of HANCI?
RQ 3: Can the hierarchical attention network and crowd intelligence boost the performance of HANCI?
RQ 4: What is the explanation based on the selection of important words and useful reviews levels?

5.1.1. Datasets

We evaluate the proposed method using three publicly accessible datasets from different domains, two of them from Amazon 5-core[4] [50] and one from Yelp Challenge 2017.[5] Amazon is a well-known E-commerce platform where users can write reviews for items they have purchased. We select two datasets Toys_and_Games and Kindle_Store from the Amazon Product Review corpus to verify the effectiveness of HANCI. Yelp is an online review platform for businesses such as restaurants, bars, spas, etc. We use the dataset from the challenge of 2017 to evaluate HANCI. The statistics of the datasets are shown in Table 2.

5.1.2. Evaluation metrics

The Root Mean Square Error (RMSE) is calculated to evaluate the performance of HANCI and baselines, which is widely used for rating prediction in the recommender system. A lower RMSE value indicates better performance of a method. Given the predicted rating $\hat{R}_{u,v}$ and the ground truth $R_{u,v}$, the RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\Sigma_{u,v}(\hat{R}_{u,v} - R_{u,v})^2}, \tag{18}$$

where N represents the total number of ratings between users and items.

5.1.3. Parameter setting

In the experiment, we implement HANCI by using PyTorch.[6] The models are trained on NVIDIA GTX 1080Ti. We randomly choose 80%, 10% and 10% of samples as the training set, the validation set and the testing set for each dataset. The ratings of the testing dataset are used as the ground truth for evaluation. We initialize the hyper-parameters and select the optimal results of parameters for the baselines in the same way as those parameters from the proposed paper. In order to extract better features and improve the performance of HANCI, the optimal hyper-parameters we set via training are as follows:

- Embedded module: The dimension of word embedding is $d = 300$. The dimension of id embedding m is tested in [16, 32, 64, 128], and the optimal value is $m = 32$.
- Text processor module: The number of hidden layer neurons of BLH h is tested in [50, 100, 150, 200], and the optimal value is $h = 100$.
- The word-level attention module: The dimension of word-level attention is $da = 100$, and the number of attention is $r = 10$.
- The review-level attention module: The dimension of review-level attention is $t = 32$.
- The feature-level attention module: The dimension of feature-level attention s is tested in [50, 100, 150, 200], and the optimal value is $s = 50$. The dimension of latent feature is $o = 32$.
- Other learning parameters: The learning rate is $lr = 0.01$, and $dropout = 0.5$. The global regularization coefficient $decay$ is tuned among [0.1, 0.01, 0.001, 0.0001], and the best value is $decay = 0.001$. The local regularization coefficient λ is tested in [0.1, 0.01, 0.001, 0.0001], and the best value is $\lambda = 0.1$.

Table 3
Comparison of baseline methods. The characteristics contain TR, DL, WI, RU and FI, which respectively represent **T**extual **R**eviews, **D**eep **L**earning, **W**ord **I**mportance, **R**eview **U**sefulness and **F**eature **I**mportance. The symbol "O" indicates that this characteristic is considered.

Method	Rating	TR	DL	WI	RU	FI
PMF	O	–	–	–	–	–
NMF	O	–	–	–	–	–
SVD++	O	–	–	–	–	–
HFT	O	O	–	–	–	–
DeepCoNN	O	O	O	–	–	–
TransNet	O	O	O	–	–	–
MPCN	O	O	O	O	O	–
A^3NCF	O	O	O	–	–	–
DER	O	O	O	–	O	–
CAML	O	O	O	–	O	–
NARRE	O	O	O	–	O	–
HANCI	**O**	**O**	**O**	**O**	**O**	**O**

5.1.4. Baselines

To evaluate the performance of rating prediction, we compare HANCI with existing state-of-the-art methods, namely PMF, NMF, SVD++, HFT, DeepCoNN, TransNet, MPCN, A^3NCF, DER, CAML and NARRE. The characteristics of comparison methods are listed in Table 3.

- PMF [51], which models users and items latent factors by introducing Gaussian distribution and presents probabilistic algorithms that scale linearly with the number of observations. It performs well on very sparse and imbalanced datasets.
- NMF [52], which proves that nonnegativity is a useful constraint for matrix factorization. It only uses the rating matrix as the input.
- SVD++[53], which is an extension of the Singular Value Decomposition neighborhood model. It combines the advantages of both neighborhood and latent factor approaches, which model the item–item similarity. Further accuracy improvements are achieved by extending the models to exploit both explicit and implicit feedback by the users.
- HFT [9], which utilizes an exponential transformation function to link the stochastic topic distribution in modeling review texts and the latent factor vector in modeling ratings. By exploiting the information in both ratings and reviews, the prediction accuracy has been improved.
- DeepCoNN [11], which constructs two parallel neural networks and adopts a CNN text processor to process review texts. It trains convolutional representations of user and

[4] http://jmcauley.ucsd.edu/data/amazon.
[5] https://www.yelp.com/dataset.
[6] https://pytorch.org.

item and passes the concatenated embedding into an FM model to predict rating.

- TransNet [12], which extends the DeepCoNN model with an additional transform layer. The additional layer learns to transform the latent representations of user and item into that of their pair-wise review, then an approximate representation of the target review can be generated and used for rating prediction.
- MPCN [14], which exploits a novel pointer-based learning scheme. This enables not only noise-free but also deep word-level interaction between user and item.
- A^3NCF [10], which is a novel aspect-aware recommender model. It designs a new topic model to extract user preferences and item characteristics from review texts. They are further used to guide the representation learning of users and items and capture a user's special attention on each aspect of the target item with an attention network.
- DER [33], which designs a time-aware GRU to model user dynamic preferences and utilizes a sentence-level CNN to process item's reviews. It is not only able to improve the recommendation performance, but also can provide sentence-level explanations.
- CAML [43], which is a co-attentive multi-task learning model. It designs an encoder–selector–decoder architecture to effectively model the cross knowledge transferred for the tasks of rating prediction and explainability. Therefore, it can not only predict ratings but also generate sentence for explanations.
- NARRE [13], which is a state-of-the-art deep neural network method in recommendation. It introduces a novel attention mechanism to explore the usefulness of reviews. Specifically, it can not only predict ratings but also provide review-level explanations.

5.2. Experimental results and discussions

This paper proposes a hierarchical attention network guided by crowd intelligence to perform rating prediction. First, in order to verify the validity of the model, we conduct experiments to compare the overall performance with eleven baseline methods in Section 5.2.1. Next, as we choose the BLH text processor to extract features from review texts, we then analyze its effectiveness in Section 5.2.2. Then, we conduct module comparison experiments to confirm the validity of the hierarchical attention network and crowd intelligence. Specifically, we examine the word importance module, the review usefulness module and the feature importance module. Because the usefulness of reviews has already been proved by NARRE, we only analyze the importance of words and the importance of features in Section 5.2.3. Hereafter, we also provide the explainability and effectiveness analysis of important words and useful reviews in Section 5.2.4. Finally, we discuss the efficiency of our proposed HANCI in Section 5.2.5.

5.2.1. Overall performance comparison (RQ1)

The performance comparison of HANCI and the baselines on three datasets are illustrated in Table 4. The evaluation metric is RMSE. We take the average of 10 runs results on each dataset as the final result of HANCI. The following observations can be drawn from the results:

First, our HANCI model outperforms all baseline methods on the three real-world datasets. Specifically, the results show that the RMSEs of HANCI are 3.92%, 3.87% and 2.45% better than the baselines based on CF and 2.91%, 2.08% and 0.97% better than the baselines based on neural network. This indicates that feature extraction and interaction are different in way and intensity which will affect the performance of the experiment, as discussed in MPCN. There are three characteristics to help improve the recommendation performance of HANCI: the importance of words, the usefulness of reviews, and the importance of features guided by crowd intelligence. According to the experimental results, HANCI performs better than those baseline methods without the three characteristics (PMF, NMF, SVD++, HFT, DeepCoNN, TransNet, A^3NCF) or only with parts of characteristics (MPCN, DER, CAML, NARRE). This validates the effectiveness of the proposed HANCI solution.

Table 4
Performance comparison with baselines. Best results are highlighted in bold.

Model	Toys_and_Games	Kindle_Store	Yelp-2017
PMF	1.3076	0.9914	1.3340
NMF	1.0399	0.9023	1.2916
SVD++	0.8860	0.7928	1.1735
HFT	0.8925	0.7917	1.1699
DeepCoNN	0.8890	0.7875	1.1642
TransNet	0.9869	0.8927	1.1675
MPCN	0.9864	0.8803	1.1614
A^3NCF	0.8972	0.7794	1.1520
DER	0.9535	0.8615	1.1749
CAML	0.8812	0.7779	1.1621
NARRE	0.8769	0.7783	1.1559
HANCI	**0.8513**	**0.7621**	**1.1447**
Improvement of HANCI	2.91%	2.08%	0.97%

Table 5
The comparison of BLH text processor and CNN text processor on three datasets. The evaluation metric is RMSE. BLH stands for BLH text processor. RA stands for the review-level attention module. NARRE is a state-of-the-art method which utilizes CNN text processor and RA to process review texts.

Text processor	Toy_and_Game	Kiddle_Store	Yelp_2017
CNN+RA(NARRE)	0.8769	0.7783	1.1559
BLH+RA	0.8683	0.7712	1.1501
Improvement of BLH+RA	0.98%	0.91%	0.50%

Second, it is obvious that the methods which combine reviews with ratings (HFT, DeepCoNN, MPCN, A^3NCF, CAML, NARRE, HANCI) show better results than the collaborative filtering methods (PMF, NMF, SVD++) that only use ratings. Specifically, the results show that the RMSEs of HANCI have been improved by 3.92%, 3.87% and 2.45% on the Toy_and_Game, Kindle_Store and Yelp-2017 datasets than SVD++, respectively. Because reviews include rich information such as content, sentiment polarity, etc., it can provide supplemental information to obtain more related and high-quality features, and thus further improve the quality of the recommender system. It proves that considering the usefulness of reviews from multiple levels can improve the accuracy of rating prediction.

Third, the performance of deep learning-based text processor (DeepCoNN, CAML, NARRE, HANCI) is superior to that of the topic model-based approaches (HFT). Specifically, the results show that the RMSEs of HANCI have been improved by 4.62%, 3.74% and 2.15% on the Toy_and_Game, Kindle_Store and Yelp-2017 datasets than HFT, respectively. This demonstrates that the word bag approach applied by the topic model simplifies the problem, while it ignores the order and dependencies between words in review. [4] and [54] have also proved that neural network-based methods show better performance. This indicates that the neural network has great ability in modeling the high-order interactions among individual users, items and crowd.

5.2.2. Effectiveness of BLH text processor (RQ2)

We conduct a comparative experiment to further understand the effectiveness of BLH text processor and demonstrate its superior performance than CNN text processor (DeepCoNN, TransNet,

Table 6
The comparison of various modules of hierarchical attention network on each dataset. The evaluation metric is RMSE. BLH stands for BLH text processor module. WA, RA and FA stand for the word-level attention module, the review-level attention module, the feature-level attention module, respectively.

Module	Toy_and_Game	Kiddle_Store	Yelp_2017
BLH+RA	0.8683	0.7712	1.1501
BLH+WA+RA	0.8556	0.7644	1.1472
BLH+WA+RA+FA	0.8513	0.7621	1.1447

NARRE). The result is shown in Table 5. NARRE is currently one of the dominant models of review-based recommendation, it contains the CNN text processor and review-level attention (RA) module. We take two sub-modules as BLH text processor and review-level attention from HANCI to conduct experiments (short for BLH+RA).

We can see from Table 5 that the RMSE values of the HANCI's BLH text processor are smaller than the NARRE and decreased by 0.98%, 0.91% and 0.50% on the three datasets, respectively. This demonstrates that the BLH text processor performs better than the CNN text processor. The reason is that although CNN is good at extracting features, it may cause loss of position information because CNN only considers position information when extracting local features through convolution and adopts the pooling layer. While BLH leverages the advantages of extracting features, processing sequence information and maintaining dependencies by using the whole hidden layers as outputs. It also proves that the BLH text processor can improve the overall performance of HANCI.

5.2.3. Effectiveness of hierarchical attention network and crowd intelligence (RQ3)

The overall performance comparison shows that HANCI obtains the best accuracy, demonstrating the effectiveness of the hierarchical attention network guided by crowd intelligence. To further prove the effectiveness of each module, we conduct a comparative experiment here. The result is shown in Table 6. More specifically, we propose a hierarchical attention network to extract the importance of words, the usefulness of reviews and the importance of features. As NARRE has proved the effectiveness of the usefulness of reviews, we gradually add WA and FA modules based on the BLH+RA module to observe recommendation accuracy. Here WA and FA stand for word-level attention and feature-level attention respectively. The evaluation metric is RMSE.

It can be seen from Table 6 that the RMSE values of the BLH+WA+RA method have increased by 1.46%, 0.88% and 0.25% on the three datasets than BLH+RA method, respectively. The reason is that the importance of words in the review are different, and there are many words that are uninformative. Therefore, if considered equally will reduce the performance of feature extraction. In addition, the importance of words makes it easier to pick out useful reviews. So HANCI performs better than the approaches that do not consider the importance of words.

Existing review-based recommendation methods are all to represent an item via extracting features from all reviews received by the item directly. We argue that these are crowd intelligence features an d are not exactly the same as what the user cares about. To visualize such difference of users towards crowd intelligence features, we select some experimental results to exhibit the latent features weights distribution of different users (User 1, User 2) on the same item, and the latent features weights distribution of the same user on different items (Item 1, Item 2) in Fig. 3. As the crowd intelligence features are represented by the latent features vector Z_v^V / Z_u^U with a size of $2h$, we conduct comparison analysis of observed attention weights

Table 7
Examples of the high-weight and low-weight reviews selected by HANCI. RA and WA denote **R**eview **A**ttention and **W**ord **A**ttention, respectively. Higher weight is bolded.

Item	Rating	RA	WA	Review
Bandana	5	**0.2126**	Case 1a	These are great as party favors, the material is thin, but it is big enough to wrap around an adult's head comfortably. Great price for favors.
		0.0070	Case 1b	One side printing. Good enough for the birthday party. The moms even wanted one. They are very thin and sewn very poorly though.
Monster high	4	**0.0603**	Case 2a	We love Monster High!!! The quality,the craftsmanship, and the beauty of this product is outstanding. This is a five star doll.
		0.0022	Case 2b	The owner liked it very much. As all the other Monster High, recommended for girls not younger than 8 years old.

distribution of the first 20 dimensions of learned latent features. It can be seen from the experiment result in Fig. 3: (1) For the User 1, the 16th, 17th and 19th dimensional latent features of the target item are more important, that is, the User 1 cares more about these features. But, the User 2 is more concerned about the 17th and 20th dimensional latent features. This indicates that different users may have different attention to the latent features of the same item. (2) For the target user, the 1st, 3rd and 19th dimensional latent features of the Item 1 and Item 2 are more important. Besides, the target user is more concerned about the 10th, 14th and 20th dimensional latent features of the Item 1, and the 9th and 15th dimensional latent features of the Item 2. This demonstrates that the target user has similar attention to some features of the similar items and different attention distributions to the features of different items.

As shown in Table 6, The RMSEs of the BLH+WA+RA+FA module have been improved by 1.96%, 1.18% and 0.47% on the Toy_and_Game, Kindle_Store, and Yelp-2017 datasets, respectively, than BLH+RA module. This demonstrates that users pay different attention to the different features of an item. The importance of features can be captured by the interaction between crowd intelligence features and user preference features. Crowd intelligence is the viewpoint of many people on an item, which can be extracted from the reviews received by the item and has a strong guidance for other people to make choices. Users can get opinions about the features they care about from the features of crowd intelligence. At the same time, those key features are strengthened and the confidence is improved. Unrelated features are also easier to distinguish, avoiding users from making wrong decisions. This result proves that the effectiveness of the importance of features guided by crowd intelligence.

5.2.4. Explainability and effectiveness of important words and useful reviews (RQ4)

Attention mechanisms can provide explanations of recommendations. Therefore, we observe the attention weight distribution of selected reviews to understand the explainability of recommendations more intuitively through case study. In addition, we construct an effectiveness analysis experiment of selecting important words and useful reviews by the attention mechanism.

(1) Case Study of Explainability

To better understand the explanation for recommendation, we select four pieces of reviews on items *Bandana* and *Monster High* to exhibit and compare the weights distribution of word-level attention and review-level attention in Fig. 4 and Table 7, respectively.

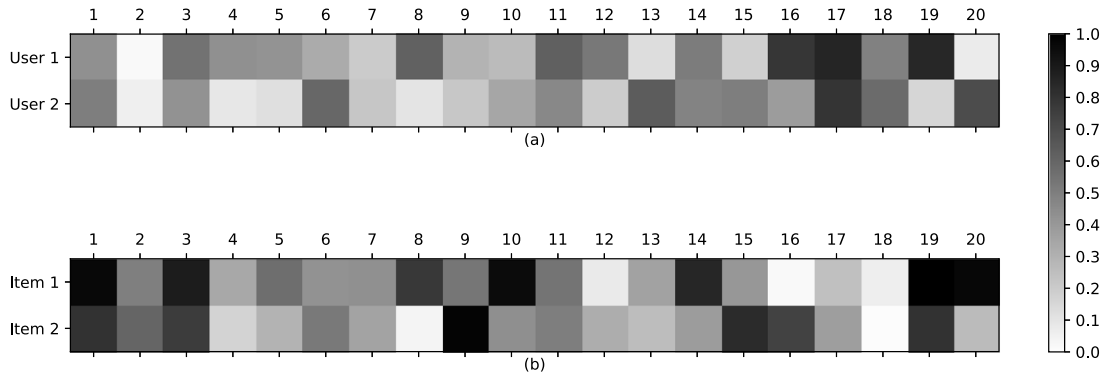

Fig. 3. Illustration of feature-level attention weights distribution. (a) The User 1 cares more about the 16, 17 and 19 dimensions of latent features of the target item. The User 2 is more concerned about the 17 and 20 dimensions of latent features of the target item. (b) The target user cares more about the 1, 3, 10, 14, 19 and 20 dimensions of latent features of the Item 1, and the 1, 3, 9, 15 and 19 dimensions latent features of the Item 2.

We can see from the results that (1) different words have different attention weights and those words describing item features or expressing user preferences hold greater weights, such as *big*, *comfortably* and *price* etc. in Case 1a. It is generally intuitive that attention focuses on item features or user preferences words around the target. It is easy to get the features of the item and the preferences of the user via these important words. So this proves the benefits of utilizing self-attention to capture more important words. By aggregating the important words from all reviews on an item, we can get a more accurate representation of the item characteristics. This is very instructive for others to make decisions. (2) By combining useful reviews with important words, the model can quickly locate the core of reviews and grasp key points promptly. In the case study, reviews 1a and 2a have greater attention weights compared with 1b and 2b, which indicates that reviews 1a and 2a are more useful reviews. In particular, reviews 1a and 2a both contain more important words or phrases. For example, review 1a contains *big enough*, *comfortably*, *great price* and review 2a contains *quality*, *beauty*, *outstanding*. Meanwhile, using useful reviews as the context of important words can avoid misinterpreting the original meaning. In contrast, reviews 1b and 2b are assigned smaller attention weights, and they contain only the author's general opinion and show little details about the item. These reviews are meaningless for users to make decisions. In this way, we provide explanations for our recommendation result by visualizing items word-level and review-level attention weights.

The case study illustrates that HANCI can pick up important words and useful reviews which not only improves the accuracy of rating prediction but also gives explanations for recommendation results. Therefore, by providing users with the highly-important words and highly-useful reviews, the explainability of the recommender system is improved.

(2) Effectiveness analysis

In this paper, we consider the importance of words, the usefulness of reviews and the importance of features. Specifically, HANCI can pick out important words and useful reviews to enhance explanatory via attention mechanism. The reviews received by an item express users' feelings and the characteristics of the item, which have a strong guiding effect on the purchase decisions of other users. Since each review is labeled as useful or not in the dataset, existing work [13] uses the Precision and Recall metrics to evaluate the performance of selecting useful reviews. Inspired by [13], we adopt the Precision and Recall metrics to analyze the performance of automatically selecting useful reviews and the enhanced effect of important words on selecting useful reviews. Specifically, the ground truth of useful reviews are the reviews labeled as helpful (named Labeled) and the items which have at least 1 labeled review are remained. For each item, the reviews are selected by five methods respectively, which are Random (The reviews are selected randomly), Latest (The latest K reviews are selected), Length (The longest K reviews are selected), HANCI^{-w} (HANCI without considering the importance of words) and HANCI (The top-K reviews with highest review-level and word-level attention values are selected). We use the Precision@K and Recall@K as the evaluation indicators [13] and calculate the Precision@K and Recall@K as follows:

$$Precision@K = \frac{\sum_{j=1}^{K} rel_j}{K}; Recall@K = \frac{\sum_{j=1}^{K} rel_j}{Re_i^{labeled}} \tag{19}$$

where $rel_j = 0/1$ indicates whether the review at rank j in the Top-K list have been labeled useful and $Re_i^{labeled}$ indicates the number of labeled reviews of item v. To evaluate different length of useful review list, we set $K = 1$ and 10. The results are shown in Table 8. As can be seen from the Table 8 (1) the results of HANCI^{-w} and HANCI achieve the best results. This demonstrates the effectiveness of our model in automatically selecting useful reviews. (2) HANCI achieves better results than HANCI^{-w}. This proves that important words help improve the selection of useful reviews.

5.2.5. Model efficiency analysis

In this paper, we develop our HANCI model based on the NARRE's [13] CNN text processor and review-level attention components, our model makes three improvements: (1) replace CNN text processor with BLH text processor (named BLH+RA) in order to extract features, handle sequence data and capture dependencies; (2) add a word-level attention module (named BLH+WA+RA) to strengthen the important words and filter out uninformative words; (3) add a feature-level attention module (named BLH+WA+RA+FA or HANCI) in order to extract the features that a user cares about from the crowd intelligence features.

To analyze the memory and computation cost required by the modules of our proposed HANCI model, we use Memory Cost, Computation Cost and Time Cost as three metrics to conduct the model efficiency analysis of our proposed HANCI model in comparison with the NARRE model which is the state-of-the-art model. Here, Memory Cost and Computation Cost respectively refer to the total amount of memory exchange and the number of floating-point operations (in FLOPs) that occurs during the model's completion of a forward propagation process for a single sample input, and Time Cost refers to the time (in seconds) that a model takes to predict the ratings of 100 samples. The tool FLOPs[7]

[7] https://github.com/JJBOY/FLOPs.

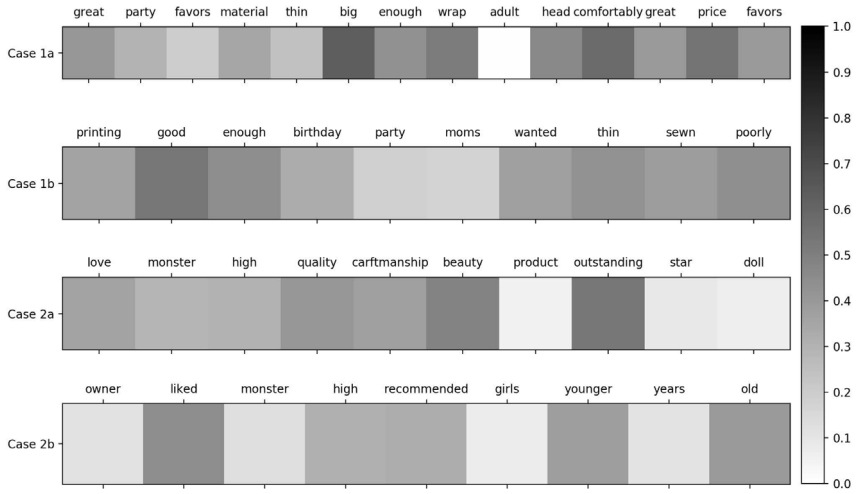

Fig. 4. Illustration of attention weights distributions for different words in reviews.

Table 8
Effectiveness analysis of automatically selected useful reviews. The results of our proposed model are highlighted in bold.

Dataset	Method	Precision@1	Recall@1	Precision@10	Recall@10
Toys_and_Games	Latest	0.1535	0.0344	0.1399	0.3856
	Random	0.3075	0.0812	0.2100	0.5765
	Length	0.2696	0.0783	0.2264	0.6413
	HANCI^{-w}	**0.3911**	**0.1399**	**0.2782**	**0.8689**
	HANCI	**0.4187**	**0.1556**	**0.3054**	**0.9012**
Kindle_Store	Latest	0.2319	0.0419	0.2433	0.4591
	Random	0.3176	0.0926	0.2673	0.5386
	Length	0.3991	0.0946	0.3129	0.6852
	HANCI^{-w}	**0.5319**	**0.1199**	**0.3654**	**0.8482**
	HANCI	**0.5922**	**0.1236**	**0.3925**	**0.8577**

Table 9
Model efficiency analysis. NARRE contains the modules of CNN text processor and review-level attention. Our proposed HANCI consists of (1) BLH stands for BLH text processor module, (2) WA stands for the word-level attention module, (3) RA stands for the review-level attention module, (4) FA stands for the feature-level attention module.

	Method	Toy	Kindle	Yelp
Memory	NARRE	47.9972M	170.5728M	55.6241M
	BLH+RA	48.9562M	171.5318M	56.5831M
	BLH+WA+RA	48.9985M	171.5740M	56.6254M
	BLH+WA+RA+FA	49.0221M	171.5977M	56.6490M
Computation (FLOPs)	NARRE	0.065962G	0.033246G	0.054827G
	BLH+RA	0.133789G	0.083382G	0.104656G
	BLH+WA+RA	0.133873G	0.083431G	0.104718G
	BLH+WA+RA+FA	0.133919G	0.083478G	0.104764G
Time (s)	NARRE	0.0045	0.0066	0.0063
	BLH+RA	0.0104	0.0118	0.0144
	BLH+WA+RA	0.0119	0.0132	0.0168
	BLH+WA+RA+FA	0.0132	0.0143	0.0180

is used to count the Memory Cost and Computation Cost. Table 9 shows the experimental results.

From the results in Table 9, we can see that (1) our proposed HANCI model costs a little bit more Memory than NARRE, which is respectively 1.02, 1.00 and 1.02 times that of NARRE in the three datasets. It is mainly because NARRE considers only the usefulness of reviews, while HANCI further considers the importance of words and the importance of features. (2) When considering only forward propagation, HANCI's Computation Cost is twice that of NARRE. Specifically, the Computation Cost of HANCI is respectively 2.03, 2.51 and 2.1 times that of NARRE in the three datasets. (3) The Memory Cost and Computation Cost of the attention module at the word level and feature level are small. (4) The execution time that HANCI takes to predict the ratings of 100 samples during testing period are all less than 20 ms in the three datasets. Besides, different models have different selectivity to the number of reviews and the length of review, which may cause unstable prediction time cost.

6. Conclusion and future work

In this work, we address the rating prediction problem from the perspective of neural Representation Learning (RL) based on rating and review data. Under the RL framework, we mainly consider two key factors to estimate a user's preference on an item: (1) how to learn effective representations for users and items and (2) how to model the interactions among users, items and reviews. To tackle these challenges, we propose a novel deep learning model named HANCI, which is implemented based on both latent feature learning and interaction mechanism design.

First, the model designs a hierarchical attention network based on the multi-level analysis of review text to explore the importance of words, the usefulness of reviews and the importance of features, and find effective expression of user personalized preferences and item latent features. Second, the complex user–items interaction mechanism is modeled in this deep learning framework via crowd intelligence guidance. To the best of our knowledge, we are the first to introduce the crowd intelligence for the rating prediction task in recommender system. By interacting with individual user behavior and crowd features, the method is able to extract the features that a user cares about, thus it can leverage the features to improve recommendation accuracy. Extensive experiments on three real-world datasets demonstrate that our proposed HANCI outperforms state-of-the-art methods in rating prediction task and automatically selects important words and useful reviews to provide word-level and review-level explanations.

Although useful reviews and important words provide a certain degree of explanation for recommendation, the performance of the model will be further improved if it can also be explained at feature-level, which is the focus of our future work.

CRediT authorship contribution statement

Chao Yang: Conceptualization, Methodology, Writing - original draft, Visualization, Project administration, Funding acquisition. **Weixin Zhou:** Software, Formal analysis, Investigation, Data curation, Writing - original draft, Validation. **Zhiyu Wang:** Formal analysis, Writing - review & editing, Visualization. **Bin Jiang:** Resources, Supervision, Writing - review & editing. **Dongsheng Li:** Writing - review & editing. **Huawei Shen:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61702176 and 62072169, and CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences under Grant CASNDST202002.

References

[1] D. Li, C. Chen, Q. Lv, J. Yan, L. Shang, S. Chu, Low-rank matrix approximation with stability, in: Proceedings of the 2016 International Conference on Machine Learning, 2016, pp. 295–303.
[2] Z. Cheng, X. Chang, L. Zhu, R. Catherine, M.S. Kankanhalli, Mmalfm: Explainable recommendation by leveraging reviews and images, ACM Trans. Inf. Syst. 37 (2019) 16:1–16:28.
[3] X.-H. Yu, Y. Chu, F. Jiang, Y. Guo, D. Gong, Svms classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features, Knowl.-Based Syst. 141 (2018) 80–91.
[4] O.S. Shalom, G. Uziel, A. Kantor, A generative model for review-based recommendations, in: Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 353–357.
[5] Q. Zhang, J. Lu, D. Wu, G. Zhang, A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities, IEEE Trans. Neural Netw. Learn. Syst. 30 (2019) 1998–2012.
[6] A. Almahairi, K. Kastner, K. Cho, A. Courville, Learning distributed representations from reviews for collaborative filtering, in: Proceedings of the 9th ACM Conference on Recommender Systems, 2015, pp. 147–154.
[7] Y. Tan, M. Zhang, Y. Liu, S. Ma, Rating-boosted latent topics: Understanding users and items with ratings and reviews, in: Proceedings of the 25th International Joint Conference on Artificial Intelligence, Vol. 16, 2016, pp. 2640–2646.
[8] J. Rashid, S.M.A. Shah, A. Irtaza, Fuzzy topic modeling approach for text mining over short text, Inf. Process. Manage. 56 (2019) 102060.
[9] G. Ling, M.R. Lyu, I. King, Ratings meet reviews, a combined approach to recommend, in: Proceedings of the 8th ACM Conference on Recommender Systems, 2014, pp. 105–112.
[10] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, M.S. Kankanhalli, A3NCF: An adaptive aspect attention model for rating prediction, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 3748–3754.
[11] 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, 2017, pp. 425–434.
[12] R. Catherine, W. Cohen, Transnets: Learning to transform for recommendation, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 288–296.
[13] C. Chen, M. Zhang, Y. Liu, S. Ma, Neural attentional rating regression with review-level explanations, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1583–1592.
[14] Y. Tay, A.T. Luu, S.C. Hui, Multi-pointer co-attention networks for recommendation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2309–2318.
[15] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, S. Ma, Explicit factor models for explainable recommendation based on phrase-level sentiment analysis, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, 2014, pp. 83–92.
[16] Z. Ren, S. Liang, P. Li, S. Wang, M. de Rijke, Social collaborative viewpoint regression with explainable recommendations, in: Proceedings of the 10th ACM International Conference on Web Search and Data Mining, 2017, pp. 485–494.
[17] 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 (2017) 29–39.
[18] C. Xu, A novel recommendation method based on social network using matrix factorization technique, Inf. Process. Manage. 54 (2018) 463–474.
[19] M.K. Najafabadi, A. Mohamed, C.W. Onn, An impact of time and item influencer in collaborative filtering recommendations using graph-based model, Inf. Process. Manage. 56 (2019) 526–540.
[20] W. Zhou, W. Han, Personalized recommendation via user preference matching, Inf. Process. Manage. 56 (2019) 955–968.
[21] H. Parvin, P. Moradi, S. Esmaeili, N.N. Qader, A scalable and robust trust-based nonnegative matrix factorization recommender using the alternating direction method, Knowl.-Based Syst. 166 (2019) 92–107.
[22] D. Rafailidis, F. Crestani, Adversarial training for review-based recommendations, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 1057–1060.
[23] R.-P. Shen, H.-R. Zhang, H. Yu, F. Min, Sentiment based matrix factorization with reliability for recommendation, Expert Syst. Appl. 135 (2019) 249–258.
[24] X. Wang, X. Yang, L. Guo, Y.P. Han, F. Liu, B. Gao, Exploiting social review-enhanced convolutional matrix factorization for social recommendation, IEEE Access 7 (2019) 82826–82837.
[25] Z. Cheng, Y. Ding, L. Zhu, M. Kankanhalli, Aspect-aware latent factor model: Rating prediction with ratings and reviews, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 639–648.
[26] L. Qiu, S. Gao, W. Cheng, J. Guo, Aspect-based latent factor model by integrating ratings and reviews for recommender system, Knowl.-Based Syst. 110 (2016) 233–243.
[27] S. Reddy, D. Chen, C.D. Manning, Coqa: A conversational question answering challenge, Trans. Assoc. Comput. Linguist. 7 (2019) 249–266.
[28] A. Rodrigo, A. Peas, A study about the future evaluation of question-answering systems, Knowl.-Based Syst. 137 (2017) 83—93.
[29] C. Yang, H. Zhang, B. Jiang, K. Li, Aspect-based sentiment analysis with alternating coattention networks, Inf. Process. Manage. 56 (2019) 463–478.
[30] S. Kadhe, B. Garcia, A. Heidarzadeh, S. el Rouayheb, A. Sprintson, Private information retrieval with side information, IEEE Trans. Inform. Theory 66 (2020) 2032–2043.
[31] M. Huang, J. Lin, Y. Peng, X. Xie, Design a batched information retrieval system based on a concept-lattice-like structure, Knowl.-Based Syst. 150 (2018) 74—84.
[32] A. Shamprasad, A.G. Krishna, B. Shashank, J. Reshma, S. Lokesh, Automatic text summarization, Int. J. Adv. Rese. Ideas Innov. Technol. 5 (2019) 287–289.
[33] X. Chen, Y. Zhang, Z. Qin, Dynamic explainable recommendation based on neural attentive models, in: Proceedings of the 2019 Conference on Association for the Advance of Artificial Intelligence, 2019.
[34] X. He, T. Chen, M.-Y. Kan, X. Chen, Trirank: Review-aware explainable recommendation by modeling aspects, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 1661–1670.

[35] M. Hou, L. Wu, E. Chen, Z. Li, V.W. Zheng, Q. Liu, Explainable fashion recommendation: A semantic attribute region guided approach, in: S. Kraus (Ed.), in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 4681–4688.
[36] H. Chen, X. Chen, S. Shi, Y. Zhang, Generate natural language explanations for recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019.
[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[38] L. Tarantino, P.N. Garner, A. Lazaridis, Self-attention for speech emotion recognition, in: Proc. Interspeech 2019, 2019, pp. 2578–2582.
[39] S. Shankar, S. Garg, S. Sarawagi, Surprisingly easy hard-attention for sequence to sequence learning, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 640–645.
[40] C. Yang, M. Jiang, B. Jiang, W. Zhou, K. Li, Co-attention network with question type for visual question answering, IEEE Access 7 (2019) 40771–40781.
[41] S. Liu, Z. Ren, J. Yuan, SibNet: Sibling convolutional encoder for video captioning, in: Proceedings of the 2018 ACM Multimedia Conference on Multimedia Conference, 2018, pp. 1425–1434.
[42] C. Yang, L. Miao, B. Jiang, D. Li, D. Cao, Gated and attentive neural collaborative filtering for user generated list recommendation, Knowl.-Based Syst. 187 (2019) 104839.
[43] Z. Chen, X. Wang, X. Xie, T. Wu, G. Bu, Y. Wang, E. Chen, Co-attentive multi-task learning for explainable recommendation, in: S. Kraus (Ed.), in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 2137–2143.
[44] D. Cong, Y. Zhao, B. Qin, Y. Han, M. Zhang, A. Liu, N. Chen, Hierarchical attention based neural network for explainable recommendation, in: Proceedings of the 2019 on International Conference on Multimedia Retrieval, 2019, pp. 373–381.
[45] S. Yun, R. Kim, M. Ko, J. Kang, SAIN: Self-attentive integration network for recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 1205–1208.
[46] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, A context-aware user-item representation learning for item recommendation, ACM Trans. Inf. Syst. 37 (2019) 22:1–22:29.
[47] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[48] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532–1543.
[49] P. Li, Z. Wang, Z. Ren, L. Bing, W. Lam, Neural rating regression with abstractive tips generation for recommendation, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 345–354.
[50] R. He, J. McAuley, Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering, in: Proceedings of the 25th International Conference on World Wide Web, 2016, pp. 507–517.
[51] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, in: Proceedings of the Advances in Neural Information Processing Systems, 2008, pp. 1257–1264.
[52] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Proceedings of the Advances in Neural Information Processing Systems, 2001, pp. 556–562.
[53] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 426–434.
[54] X. He, T.-S. Chua, Neural factorization machines for sparse predictive analytics, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 355–364.