Hierarchical text interaction for rating prediction

Jiahui Wen [a], Jingwei Ma [b,*], Hongkui Tu [c], Mingyang Zhong [d], Guangda Zhang [a], Wei Yin [d], Jian Fang [a]

[a] *National Innovative Institute of Defense Technology, Beijing, China*
[b] *Shandong Normal University, Jinan, Shangdong, China*
[c] *National University of Defense Technology, Changsha, China*
[d] *The University of Queensland, Australia*

ARTICLE INFO

ABSTRACT

Traditional recommender systems encounter several challenges such as data sparsity and unexplained recommendation. To address these challenges, many works propose to exploit semantic information from review data. However, these methods have two major limitations in terms of the way to model textual features and capture textual interaction. For textual modeling, they simply concatenate all the reviews of a user/item into a single review. However, feature extraction at word/phrase level can violate the meaning of the original reviews. As for textual interaction, they defer the interactions to the prediction layer, making them fail to capture complex correlations between users and items. To address those limitations, we propose a novel Hierarchical Text Interaction model (HTI) for rating prediction. In HTI, we propose to model low-level word semantics and high-level review representations hierarchically. The hierarchy allows us to exploit textual features at different granularities. To further capture complex user–item interactions, we propose to exploit semantic correlations between each user–item pair at different hierarchies. At word level, we propose an attention mechanism specialized to each user–item pair, and capture the important words for representing each review. At review level, we mutually propagate textual features between the user and item, and capture the informative reviews. The aggregated review representations are integrated into a collaborative filtering framework for rating prediction. Experiments on five real-world datasets demonstrate that HTI outperforms state-of-the-art models by a large margin. Further case studies provide a deep insight into HTI's ability to capture semantic correlations at different levels of granularities for rating prediction.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

With the prevalence of Internet, online services generate massive amounts of data on a daily basis, and users are facing the problem of information overload when finding their interested items (e.g. books, movies). To address this problem, various recommendation techniques are proposed to quickly find relevant information for the users [1]. Collaborative filtering (CF) [2,3] is one of the most widely employed techniques for recommendation. It learns latent factors of user and item based on interaction records such as ratings [4,5]. However, CF-based methods are mainly based on interaction data, and the learned latent factors do not explain why an item is recommended [6]. Therefore, those methods encounter some challenges, such as data sparseness and lack of interpretation.

To address those limitations, researchers resort to learning semantic features [7] from the reviews. In most online services, users are allowed to write reviews to express their opinions towards the consumed items. The reviews contain plenty of information that reveals user interests and item features. Prior works leverage topic modeling techniques such as Latent Dirichlet Allocation (LDA) [8] to extract high-level features based on the reviews [9,10]. Those methods demonstrate improved performance over latent factor models. However, they represent the reviews as bags of words, and fail to capture textual features and local semantics [11] expressed at the phrase and sentence level.

Recently, deep learning techniques (e.g. convolutional neural networks) have been widely applied in text-related tasks such as text classification [12]. Existing works demonstrate the effectiveness of the deep learning techniques in extracting high-level abstract representations from texts for boosting specific tasks such as recommendation. In the light of this, many works [13,14] leverage neural networks to derive high-level textual factors over the reviews, and integrate those features into latent factor models [15] for rating prediction. Some research [16,17] propose to

* Corresponding author.
E-mail addresses: wen_jiahui@outlook.com (J. Wen), majingwei0824@gmail.com (J. Ma), tuhkjet@foxmail.com (H. Tu), my.zhong@hotmail.com (M. Zhong), zhanggd_nudt@hotmail.com (G. Zhang), yinweihappy168@yahoo.com (W. Yin), fangjian_alpc@163.com (J. Fang).

capture complex user–item interaction by modeling customized textual features given the specific users/items. Although these research demonstrate better recommendation performance than topic-based models, they have several limitations [18]. First, most of the review-based models [19,20] defer the user–item interactions till the prediction layer, resulting in static latent features of users and items. Therefore, in those models, users/items are always represented with the same latent factors regardless of the target items/users. They are incapable of capturing dynamic correlations between users and items, and fail to model the complex interactions between them. Second, other review-based models [16,17] simply concatenate all reviews of each user/item into a single review. The irrelevant words may introduce noises and undermine the recommendation performance. Furthermore, their feature interrelations are modeled at word or phrase level, which can twist the meaning of the original sentences [21]. For example, it is not uncommon for an overall positive review to have a few negative words. Similarly, positive words can occasionally appear in reviews that present overall negative sentiment.

To address those limitations, we propose an HTI model that exploits textual features at different levels of hierarchies for item recommendation. In HTI, feature extractions are achieved hierarchically at both word and review levels. One advantage of the hierarchical architecture is that we can model local informativeness and global overall semantics of the reviews. The local and global information can complement each other for preserving rich textual features at different granularities. In addition, to uncover complex user–item interrelations, we propose to model semantic correlations between the users and items at different levels. Specifically, at word level, we propose an attention mechanism to summarize the words, where the attention scores are parameterized by each user–item pair. At review level, we propose an interactive network that propagates textual features between the user and item, and guide each other to select the relevant reviews for recommendation.

Compared with existing methods, we model local word informativeness and global review sentiment with a hierarchical architecture. The hierarchical review information allows us to sufficiently exploit textual features at different levels of granularities. Further, rather than representing users and items with static latent features, we dynamically identify informative textual features at both word and review levels for each specific user–item pair. We conduct experiments on HTI over five real-world datasets with reviews. The experimental results demonstrate the advantage of HTI over state-of-the-art baselines, and validate the rationale underlying the hierarchical architecture of the proposed model. In summary, the contributions of this work are listed as follows:

- We propose a review-based recommendation model (HTI for short) with hierarchical text modeling. The model preserves rich textual information at both word and review levels, allowing it to exploit textual features at different levels of granularity for accurate rating prediction.
- We propose to model texts interaction in the hierarchical architecture, enabling it to capture informative local words and global reviews, and extract interaction-based textual features for the specific task of recommendation. In this sense, we are the first to simultaneously model hierarchical and interactive textual features for recommendation.
- We conduct experiments on five real-world datasets, and demonstrate that HTI consistently outperforms the state-of-the-art baselines by a large margin. We also study the contribution of each component for the recommendation task. Finally, case studies provide a better understanding of HTI's effectiveness in modeling textual interaction at different levels.

2. Related work

Our work is related to recommendation with reviews, attention-based recommendation and hierarchical representation learning. We briefly review the recent advances in these areas.

2.1. Recommendation with reviews

Traditional collaborative filtering recommenders [22,22–24] are mainly based on user–item interactions. Therefore, they have two major limitations, namely data sparsity and cold-start problem, as users usually give few ratings given the larger number of items [25]. To tackle these limitations, many researches resort to extra knowledge such as social networks [26–28], images [29,30]. Among those auxiliary data sources, review texts have received much research attention [31]. Review texts are ubiquitous in recommender systems, and there are strong inherent correlations between the textual sentiment and user preferences. Early researches employ topic modeling such as LDA [8] to uncover latent topic factors, and combine the topical information with ratings to recommend items [32,33]. CDL [34] leverages SDAE [35] to extract deep features from texts, and uses Bayesian model [36] to achieve the combination of ratings and reviews for rating prediction. Those methods demonstrate better performance than the baselines that are mainly based on user–item rating data. However, they regard the reviews as bags of words (BOW), hence they fail to capture word-order information and local semantics [37].

To approach this limitation, several studies [38] leverage deep neural networks such as CNNs for extracting high-level text features. Neural networks [39] are proved to be powerful in text-related tasks such as text classification [40,41], as they are able to capture local semantics of different granularities when filter maps of different sizes are applied. Kim et al. [13] propose a neural network model, ConvMF, for recommendation. ConvMF utilizes CNNs to extract item latent features from item reviews, and integrate those features into a Probabilistic Matrix Factorization (PMF) framework for rating predictions. DeepCoNN [14] uses two parallel neural networks to learn user and item latent features from their respective reviews. Then the Factorization Machines (FM) [15] model is used to model the high-order interactions between the latent features of users and items. D-attn [20] models latent features of users/items with dual local and global attention. The neural networks are applied to the user/item reviews separately, and the resultant factors are used for rating prediction in a similar way as matrix factorization. NARRE [19] explores the usefulness of the reviews, and represents latent factor of each user/item as a weighted sum over the review representations. Those deep models demonstrate improved performance over the above BOW-based models [42]. However, they defer the interactions between users and items till the prediction layer, and each user/item always has the same latent features regardless of the target item/user. Therefore, they fail to model complex correlations between users and items. To deeply exploit the interrelations between latent features of users and items, recent works [16,17] consider early interactions between textual features before deriving latent features of users and items. Specifically, they first capture contextual features at each position of the texts with convolution operations. Instead of max pooling over the features, they model the mutual correlations between contextual features of users and items, and apply convolution with mean pooling operations to calculate the final latent features of users and items. Despite proven to be effective in rating predictions, those methods simply concatenate all review texts of a user into a single review, which can inevitably introduce noises and incur information loss, due to the various lengths and numbers of user/item reviews. Furthermore, in those works, semantic correlations are modeled at word/phrase level, which may violate the original meaning of the sentences [21].

2.2. Attention-based recommendation

Attention Mechanism [43] has wide applications [44] in neural networks [45,46] community. The motivation is to place higher weights to parts of information that is strongly related to the specific applications, and ignore the others that may negatively affect the model performance. In review-based area, as not all words have equal contribution, we need to pay attention to important words that have strong indications of user preferences and item properties. Seo et al. [20] employ a combination of local and global attention mechanism to identify important words. The local attention selects informative words from a local window, while the global attention filters out noisy words and helps to model overall semantics of the reviews. In [19], the authors propose to learn usefulness of the reviews with an attention mechanism. For each item/user, the attention score of each review is proportional to the relevance between the review and user/item associated with the review. Lu et al. [47] utilize a combination of PMF and an attention-based GRU network to collaboratively learn latent factors of users and items from ratings and reviews. Wang et al. [48] propose an attention mechanism to aggregate multiple predictive models for article recommendation. CARL [16] utilizes an attention mechanism to mutually model pair-based semantic relevance between textual features of users and items. DAML [17] integrates local and mutual attention of convolution network to jointly extract features from texts.

Compared with the above methods, we propose a hierarchical attention mechanism to preserve informative textual features at both word and review levels. At word level, the attentions are specialized by each user–item pair, hence we can identify useful words guided by user preferences and item characteristics. At review level, we model the semantic alignments [49] between user and item reviews through an interactive network.

2.3. Hierarchical representation learning

Hierarchical representation learning [50] is to learn latent features at a hierarchical manner, and capture user interests or item characteristics at different granularities for fine-grained recommendation. Chen et al. [51] propose a hierarchical model that consists of component-level and item-level representation learnings. At component level, an attention mechanism is proposed to select informative components for representing items. At item level, another attention module is used to capture important items for comprising user interests. Wu et al. [52] model user latent preferences for image recommendation by identifying three aspects (i.e. upload coherence, social influence and owner admiration). A hierarchical attention network is designed to capture both element and aspect level features. In [53], the authors approach group level and user level recommendation. They propose an attention mechanism to select expert users for representing the groups. Specifically, each group places high weights to a specific set of users who are confident with the target item. With the group information, the user preferences are modeled at a more general level to alleviate data sparsity [54]. DBRec [55] iteratively performs collaborative filtering and latent group discovery. These two modules can complement each other and learn better user/item representations. In addition to the user–item interaction, it interacts user/item with item/user group to capture user preferences and item properties at different levels of granularities. HRM [56] hierarchically captures both sequential behavior and users' general tastes for next item recommendation.

The proposed work is related to hierarchical representation learning, as we model textual features both at word and review levels. However, the aforementioned methods linearly fuse the latent features at each level. On the contrary, we propose to model complex user–item interrelations at different hierarchies.

Table 1
Symbols and Descriptions.

| Symbol	Description
u_i, v_j	Notations of a user and a item respectively
$U = \{u_1, \ldots, u_M\}$	A set of M users
$V = \{v_1, \ldots, v_N\}$	A set of N users
$\mathbf{u}_i, \mathbf{v}_j$	Latent factors of u_i and v_j respectively
$\mathbf{R}$	Rating matrix
$r_{ij}, \hat{r}_{ij}$	Ground truth and predictive rating score of u_i over v_j respectively
$D_{u_i} = \{d_{i,k}, \ldots, d_{i,m}\}$	A set of u_i's m reviews
$D_{v_j} = \{d_{j,1}, \ldots, d_{j,n}\}$	A set of v_j's n reviews
$d = \{w_i\}_{i=1}^p$	An example review with p words
$\{\mathbf{w}_i\}_{i=1}^p$	Embeddings of words $\{w_i\}_{i=1}^p$
$\mathbf{d}$	Review representation of an example document d
$\{\mathbf{d}_{i,k}\}_{k=1}^m$	The set of representations of u_i's reviews
$\{\mathbf{d}_{j,t}\}_{t=1}^n$	The set of representations of v_j's reviews
$\mathbf{p}_i$	Initial representation of u_i's reviews
$\mathbf{q}_j$	Initial representation of v_j's reviews
$\mathbf{s}_i$	Intermediate representation of u_i's reviews
$\mathbf{t}_j$	Intermediate representation of v_j's reviews
$\mathbf{d}_i$	Aggregated representation of u_i's reviews
$\mathbf{d}_j$	Aggregated representation of v_j's reviews
ϕ_l	lth layer neural network at the prediction module
Θ	all Model parameters
λ	Regularization parameter
$\mathcal{D}$	Training set
η	Learning rate

3. Problem formalization

In a recommendation problem, we denote $U = \{u_1, \ldots, u_M\}$ as a set of M users, and $V = \{v_1, \ldots, v_N\}$ a set of N items. The ratings of the users over the items are represented with a rating matrix $\mathbf{R} \in \mathbb{R}^{M \times N}$, where each entry r_{ij} in the matrix denotes user u_i's preference over item v_j. As we focus on explicit recommendation in this work, the ratings take explicit real values in a specific range (i.e. [1, 5]). For each user–item pair (u_i, v_j), we denote $D_{u_i} = \{d_{i,1}, \ldots, d_{i,m}\}$ as the repository of reviews written by the user, and $D_{v_j} = \{d_{j,1}, \ldots, d_{j,n}\}$ as the set of reviews written about the item, where m and n denote the number of reviews of u_i and v_j respectively.

Given a user–item pair and their respective historical reviews, the recommendation problem is to exploit the textual information in the reviews and produce the estimated rating $\hat{r}_{ij}$ of u_i over v_j. In the reminder of this paper, we denote matrices with upper case bold letters, vectors with lower case bold case letters without any specification, and scalars with non-bold letters. The major symbols used in the paper are summarized in Table 1, other intermediate variables are detailed in the corresponding sections.

4. The proposed model

4.1. Overview

Fig. 1(a) illustrates the architecture of the proposed model, which can be decomposed into two parts, namely word-level attention module and review-level interaction module. The word-level attention module is indicated in the blue rectangle in Fig. 1(a). It is applied to each review document for extracting high-level word vectors with CNNs. Then, an attention mechanism is proposed to summarize the word vectors into a review representation (e.g., $\mathbf{d}_{i,1}$). The review-level interaction module is

(a)Overall architecture of the proposed model

(b)Word-level attention module

(c)Review-level interaction module
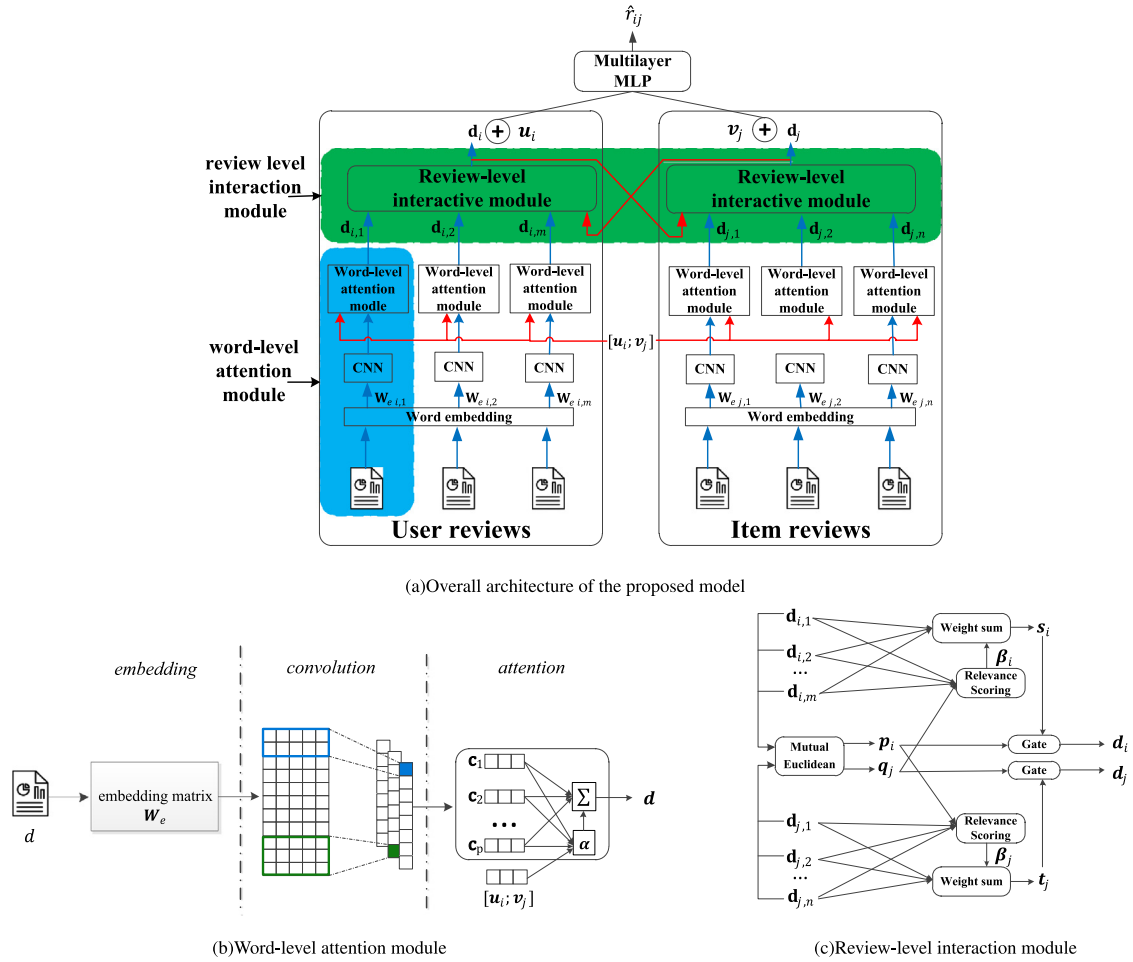
Fig. 1. (a) shows the overall architecture of the proposed model, which consists of a word-level attention module and a review-level interaction module. The two sub-modules are presented in (b) and (c), respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

illustrated with a green rectangle in Fig. 1(a). It models mutual correlations between user and item reviews, and produces an aggregated representation (i.e. $\mathbf{d}_i$, $\mathbf{d}_j$) for the user and item reviews respectively. Finally, the aggregated representations are combined with the corresponding latent factors (i.e. $\mathbf{u}_i$, $\mathbf{v}_j$) for rating prediction. The details of the two modules are depicted in Fig. 1(b) and Fig. 1(c) respectively.

In the word-level attention module, an attention mechanism is proposed to aggregate the words for each review. The attention scores are parameterized by each user–item pair, so that it is able to identify the pair-wise important words. In the review-level interaction module, an interactive network is proposed to aggregate the user and item reviews. To obtain the attention scores, we first mutually propagate textual features between user and item reviews. After that, we calculate the attention scores based on the relevances between user (item) reviews and the propagated item (user) textual features. Therefore, we can identify the most interrelated reviews for the prediction task. The combination of the two modules results in a hierarchical model, which is able to capture the most correlated textual features at different levels of granularities for rating predictions.

4.2. Word-level attention module

Previous works [14,16,17] concatenate that all the reviews of a user/item into a single review, and apply convolution and max-pooling operations for extracting review representations. On the contrary, we apply CNNs with an attention mechanism to each review, and then aggregate the resulted review representations with an interaction module. The rationale is to preserve textual information in a hierarchical manner for the prediction task.

As shown in Fig. 1(b), we first embed the words of a review into a sequence of low-dimension dense embeddings, and then apply 1-dimension CNNs to extract high word vectors. Finally, we utilize an attention mechanism to summarize those word vectors into a review representation.

Embedding. The embedding operation is to map a word to its corresponding word embedding, and it is achieved by indexing an embedding matrix with its word indices in the vocabulary. Without loss of generality, we denote a review as $d = \{w_i\}_{i=1}^p$, where p is the length of the review. The word sequence in d is then embedded into a sequence of word embeddings, $\{\mathbf{w}_i\}_{i=1}^p$, using a word embedding matrix $\mathbf{W}_e \in \mathbb{R}^{v \times dim}$, where v denotes the vocabulary size, and dim denotes the dimension of the word embeddings.

Convolution. CNNs are widely employed in natural language processing area, as they are proven to be powerful in specific tasks such as text classification [40]. In CNNs, each filter map performs convolution operation from n-gram word embeddings, and it is slid over the review to capture n-gram local features at different positions of the sequence. When filter maps of different sizes are applied, they are assumed to be capable of capturing local

semantics of different granularities. Specifically, we apply a convolution filter map $\mathbf{W}_f \in \mathbb{R}^{s \times dim}$ of size s over a window of s words centered at w_i, and produce a single convolutional feature:

$$c_{i,f} = f([\mathbf{w}_{i-(s-1)/2}; \cdots; \mathbf{w}_{i+(s-1)/2}] * \mathbf{W}_f + b_f) \tag{1}$$

where $[;]$ is the concatenation operation, f is a non-linear function (e.g. relu function [57]), b_f is a bias scalar and $*$ denotes the convolution operation. In this work, s is an odd number. In practice, we apply filter maps of various sizes to produce multiple features for each word w_i. Those features are concatenated to form a convolutional vector for w_i: $\mathbf{c}_i = [c_{i,f_1}; c_{i,f_2}; \cdots; c_{i,f_k}]$, where k is the number of filter maps, and c_{i,f_k} is the convolutional feature produced by the kth filter map.

Attention. Given the word vectors of a review, we propose a pair-specific attention mechanism to aggregate those vectors into a review representation. The basic idea is that not all words have equal contribution for comprising the review representation. In addition, users may have different preferences over the words when composing the review given different items. For example, "inexpensive" is informative for users who focus on the price, while "lightweight" needs to be paid high attention for users who care about the portability of the products. Therefore, the attention scores need to be specialized by the user–item pairs. Specifically, the review representation can be aggregated as the weighted sum over the word vectors:

$$\mathbf{d} = \sum_{i=1}^{p} \alpha_i \mathbf{c}_i \tag{2}$$

where α_i is the attention score assigned to $\mathbf{c}_i$, and it is proportional to the relevance between the user–item pair and $\mathbf{c}_i$:

$$\begin{aligned} \mathbf{m} &= tanh(\mathbf{W}^T[\mathbf{u}_i; \mathbf{v}_j] + \mathbf{b}) \\ \alpha_i &= \frac{exp(\mathbf{m}^T \mathbf{c}_i)}{\sum_{i=1}^{p} exp(\mathbf{m}^T \mathbf{c}_i)} \end{aligned} \tag{3}$$

where matrix $\mathbf{W}$ and vector $\mathbf{b}$ are model parameters, and $\mathbf{u}_i$ and $\mathbf{v}_j$ are latent factors of u_i and v_j respectively. As illustrated, we first transform the concatenated latent factors of u_i and v_j into a latent vector $\mathbf{m}$, and measure its relevance with the word vectors in the common latent space. To obtain the attention scores, the attention scores are then normalized with a softmax function.

4.3. Review-level interaction module

We denote the set of review representations of user u_i and item v_j as $\{\mathbf{d}_{i,k}\}_{k=1}^{m}$ and $\{\mathbf{d}_{j,t}\}_{t=1}^{n}$, where m and n denote the number of reviews of u_i and v_j respectively. Each of the review representation is generated by the word-level attention in parallel, as specified in Section 4.2. The essence of this module is a mutual learning mechanism, which enables textual information to propagate between each user–item pair. The propagated information provides guidelines for the user and item to select the most relevant reviews. The rationale is that not all reviews are equally important, and user/item reviews that have high relevance to item/user reviews are more informative [58]. Those reviews need to be paid high attention, as they represent semantic correlations between user preferences and item characteristics.

The dataflow of this module is illustrated in Fig. 1(c), we first calculate the initial representations (i.e. $\mathbf{p}_i$, $\mathbf{q}_j$) over the user and item reviews, respectively. Then, the initial representations are used to propagate information between users and items, and find the mutually relevant reviews for comprising the intermediate representations (i.e. $\mathbf{s}_i$, $\mathbf{t}_j$). Finally, we employ a gate network to balance the initial representations and intermediate representations, and produce the final representations (i.e. $\mathbf{d}_i$, $\mathbf{d}_j$) of user and item reviews respectively.

Initial representations. The initial representations are calculated with an attention mechanism. The attention scores are inversely proportional to the minimum Euclidean distance between each review and all the counterpart reviews. Therefore, reviews with smaller distances are weighted higher for aggregating the initial representations. To determine the textual features initially propagated between each user–item pair, we first calculate the mutual Euclidean distance between the user and item reviews.

$$\begin{aligned} e_{k,t} &= |\mathbf{d}_{i,k} - \mathbf{d}_{j,t}| \\ k &= 1, \ldots, m; \quad t = 1, \ldots, n \end{aligned} \tag{4}$$

where $e_{k,t}$ is the Euclidean distance between kth user review and tth item review. For each user/item review, we select the minimum distance to the counterpart reviews. The selection is achieved through a min-pooling operation:

$$\begin{aligned} a_k &= min(e_{k,1}, e_{k,2}, \ldots, e_{k,n}) \\ b_t &= min(e_{1,t}, e_{2,t}, \ldots, e_{m,t}) \end{aligned} \tag{5}$$

where a_k indicates the closest distance between kth user review and all the item reviews, and b_t indicates the closest distance between tth item review and all the user reviews. Therefore, a review with small a_k is assigned a large attention score for representing the initial representation of user reviews:

$$\begin{aligned} \delta_{i,k} &= \frac{exp(-a_k)}{\sum_{k=1}^{m} exp(-a_k)} \\ \mathbf{p}_i &= \sum_{k=1}^{m} \delta_{i,k} \mathbf{d}_{i,k} \end{aligned} \tag{6}$$

Similarly, the initial representation of item (*i.e.,* v_j) reviews can be obtained as follows:

$$\begin{aligned} \delta_{j,t} &= \frac{exp(-b_t)}{\sum_{t=1}^{n} exp(-b_t)} \\ \mathbf{q}_j &= \sum_{t=1}^{n} \delta_{j,t} \mathbf{d}_{j,t} \end{aligned} \tag{7}$$

where $\mathbf{p}_i$ and $\mathbf{q}_j$ are the initial representations of user (*i.e.,* u_i) and item (*i.e.,* v_j) reviews respectively. $\delta_{i,k}$ and $\delta_{j,t}$ are normalized scalars that are inversely proportional to a_k and b_t respectively. Therefore, a user review $\mathbf{d}_{i,k}$ having smaller a_k receives larger weight $\delta_{i,k}$ for comprising $\mathbf{p}_i$, and the same for each item review.

Intermediate representations. The initial representations, $\mathbf{p}_i$, $\mathbf{q}_j$, are then used to propagate textual information between user and item reviews. The propagation is to uncover mutually correlated reviews for rating prediction. The semantic correlations are captured through an attention mechanism, which assigns large attention scores to reviews that are highly relevant to the propagated information.

$$\begin{aligned} \gamma_{i,k} &= \mathbf{v}_1^T tanh(\mathbf{W}_u^T \mathbf{q}_j + \mathbf{W}_{vu}^T \mathbf{d}_{i,k} + \mathbf{b}_u) \\ \beta_{i,k} &= \frac{exp(\gamma_{i,k})}{\sum_{k=1}^{m} exp(\gamma_{i,k})} \\ \mathbf{s}_i &= \sum_{k=1}^{m} \beta_{i,k} \mathbf{d}_{i,k} \end{aligned} \tag{8}$$

where matrices $\mathbf{W}_u$, $\mathbf{W}_{vu}$ and vectors $\mathbf{v}_1$, $\mathbf{b}_u$ are model parameters. That is, we first measure the relevance between $\mathbf{q}_j$ and each of the user review $\mathbf{d}_{i,k}$, and obtain a relevance score $\gamma_{i,k}$ for each user review $\mathbf{d}_{i,k}$. After that, we use the normalized relevance scores $\{\beta_{i,k}\}_{k=1}^{m}$ to aggregate $\{\mathbf{d}_{i,k}\}_{k=1}^{m}$ into an intermediate representation $\mathbf{s}_i$. As shown in the equation, the attention over the user reviews is guided by the initial representation of item reviews. The underlying rationale is to highlight the most interrelated user reviews that can possibly meet the item characteristics.

Similarly, we use $\mathbf{p}_i$ to propagate information from user reviews to item reviews, and select the most correlated reviews to comprise the intermediate representation of item reviews:

$$\gamma_{j,t} = \mathbf{v}_2^T tanh(\mathbf{W}_v^T \mathbf{p}_i + \mathbf{W}_{uv}^T \mathbf{d}_{j,t} + \mathbf{b}_v)$$

$$\beta_{j,t} = \frac{exp(\gamma_{j,t})}{\sum_{t=1}^{n} exp(\gamma_{j,t})} \tag{9}$$

$$\mathbf{t}_j = \sum_{t=1}^{n} \beta_{j,t} \mathbf{d}_{j,t}$$

where matrices $\mathbf{W}_v$, $\mathbf{W}_{uv}$ and vectors $\mathbf{v}_2$, $\mathbf{b}_v$ are model parameters, $\gamma_{j,t}$ is the relevance score between $\mathbf{p}_i$ and $\mathbf{d}_{j,t}$ and $\beta_{j,t}$ is the softmax normalization of $\gamma_{j,t}$.

Aggregated representations. Based on the initial representations (e.g. $\mathbf{p}_i$, $\mathbf{q}_j$) and the intermediate representations (e.g. $\mathbf{s}_i$, $\mathbf{t}_j$), we produce the aggregated representations for the user and item reviews. As we do not know how each user/item balances these two representations, we employ a gate network to automatically model the combination of these two representations as follows.

$$\mathbf{g}_u = \sigma(\mathbf{W}_{g1}^T \mathbf{p}_i + \mathbf{W}_{g2}^T \mathbf{s}_i + \mathbf{b}_g)$$
$$\mathbf{d}_i = \mathbf{g}_u \odot \mathbf{p}_i + (1 - \mathbf{g}_u) \odot \mathbf{s}_i \tag{10}$$

where matrices $\mathbf{W}_{g1}$, $\mathbf{W}_{g2}$ and vector $\mathbf{b}_g$ are model parameters, $\sigma(x) = \frac{1}{1+exp(-x)}$ is the sigmoid function, and $\odot$ denotes the element-wise product. Similarly, the gate layer to model the balance between $\mathbf{q}_j$ and $\mathbf{t}_j$ is formulated as follows,

$$\mathbf{g}_v = \sigma(\mathbf{W}_{g1}^T \mathbf{q}_j + \mathbf{W}_{g2}^T \mathbf{t}_j + \mathbf{b}_g)$$
$$\mathbf{d}_j = \mathbf{g}_v \odot \mathbf{q}_j + (1 - \mathbf{g}_v) \odot \mathbf{t}_j \tag{11}$$

where the model parameters $\mathbf{W}_{g1}$, $\mathbf{W}_{g2}$, $\mathbf{b}_g$ are shared across user and item gate networks.

4.4. Prediction module

Notice that the aggregated representations of user and item reviews capture the explicit textual correlations between the users and items, while the latent factors implicitly model the rating behaviors encoded in the rating matrix $\mathbf{R}$. Therefore, we propose to seamlessly integrate the aggregated representations and latent factors with nonlinear transformations. Inspired by [19], given a user–item pair (u_i, v_j), we first linearly combine the latent factors and aggregated representations for the user and item:

$$\mathbf{x}_i = \mathbf{u}_i + \mathbf{d}_i; \quad \mathbf{y}_j = \mathbf{v}_j + \mathbf{d}_j \tag{12}$$

where $\mathbf{u}_i$ and $\mathbf{v}_j$ are latent factors of u_i and v_j. The combined user and item vectors are then input to multilayer neural networks to estimate the rating score:

$$\mathbf{h}_0 = [\mathbf{x}_i; \mathbf{y}_j; \mathbf{x}_i \odot \mathbf{y}_j]$$
$$\hat{r}_{ij} = \phi_L(...\phi_2(\phi_1(\mathbf{h}_0))...) \tag{13}$$
$$\phi_l = \sigma_l(\mathbf{W}_l^T \mathbf{h}_{l-1} + \mathbf{b}_l)$$

where ϕ_l indicates the lth layer neural network, and σ_l, $\mathbf{W}_l$, $\mathbf{b}$ are the respective activation function, weight matrix and weight vector. As we focus on explicit rating prediction, we do not apply nonlinear transformations to the last-layer neural network.

4.5. Model learning

We formulate the learning problem as a regression problem. Therefore, the objective function of the proposed model can be defined as the square loss [59,60] between the predicted and real ratings:

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{(u_i, v_j, r_{ij}) \in \mathcal{D}} (r_{ij} - \hat{r}_{ij})^2 + \lambda \|\Theta\|^2 \tag{14}$$

where $\mathcal{D}$ denotes the training set, r_{ij} is the ground truth rating and $\hat{r}_{ij}$ is the predicted rating, Θ denotes all the model parameters and λ is the regularization tradeoff. The term $\lambda \|\Theta\|^2$ is introduced to avoid overfitting. We optimize the object function with Adam optimizer [61], which is a variant of Stochastic Gradient Descent with a dynamically tuned learning rate. The model parameters are updated at every step along the gradient direction with the following protocol:

$$\Theta^t \leftarrow \Theta^{t-1} - \eta \frac{\partial \mathcal{L}}{\partial \Theta} \tag{15}$$

where η is the learning rate, and $\frac{\partial \mathcal{L}}{\partial \Theta}$ is the set of partial derivatives of the object function with respect to the model parameters. Those partial derivatives can be automatically computed with typical deep learning libraries. The algorithm of the forward pass and backpropagation of the proposed model are summarized in Algorithm 1. The presented algorithm processes a user–item record for each iteration. However, in practice, it is can be easily extended to process a batch of data per iteration. Line 4–6 correspond to word-level attention module for comprising review representations. Line 7–9 specify review-level interaction module for aggregating user and item reviews. In line 10–11, we calculate the predicted rating and back propagate the loss to optimize the model parameters.

Algorithm 1 forward pass and back-propagation learning of HTI.

Require: the training set $\mathcal{D}$; the learning rate η; the regularization parameter λ; the reviews for each user u_i and item v_j, D_{u_i} and D_{v_j}; the number of interactive gate layer, l_g

Ensure: latent factors of each user u_i and item v_j, $\mathbf{u}_i$ and $\mathbf{v}_j$; aggregated representations of user and item reviews, $\mathbf{d}_i$ and $\mathbf{d}_j$; the model parameters,Θ.

1: Initialize all model parameters Θ;
2: **while** not convergence **do**
3:      Randomly sample a tuple $(u_i, v_j, r_{ij}) \in \mathcal{D}$;
4:      Obtain the user and item embeddings;
5:      Obtain word embeddings for each review in D_{u_i} and D_{v_j};
6:      Calculate review representations $\{\mathbf{d}_{i,k}\}_{k=1}^{m}$ and $\{\mathbf{d}_{j,t}\}_{t=1}^{n}$ as specified in Section 4.2;
7:      Calculate initial representations of the user and item reviews, $\mathbf{p}_i$ and $\mathbf{q}_j$ by Eqs. (4)–(7);
8:      Calculate intermediate representations of user and item reviews, $\mathbf{s}_i$ and $\mathbf{t}_j$, as specified in Eqs. (8) and (9).
9:      Calculate aggregated representations of user and item reviews, $\mathbf{d}_i$ and $\mathbf{d}_j$, as specified in Eqs. (10) and (11).
10:      Calculate the predicted score $\hat{r}_{ij}$ as descried in Section 4.4;
11:      Calculate $\frac{\partial \mathcal{L}}{\partial \Theta}$ and update Θ by Eq. (15);
12: **end while**

4.6. Time complexity analysis

For a given user–item pair, the time complexity of a forward pass of the proposed model can be decomposed into the following parts. The time complexity of the convolution layer is $O(dim * k * l_d * (m+n))$, where dim is the embedding size, k is the number of convolutional features (i.e. number of feature maps), l_d denotes the maximum review length, and m and n denote the number of user and item reviews, respectively. In the word-level attention layer, the time complexity to calculate the attention scores is $O(k^2(m+n))$. The time complexity of review-level interaction network is $O(mnk + k^2(m+n))$, where the first term is the time complexity for computing the initial representations, while the second term is the time complexity for calculating the aggregated representations. In the prediction layer, the operations of the multi-layer neural networks can be completed in $O(\sum_{l=1}^{L} k_l k_{l+1})$, where k_l is the dimension of the lth layer. In reality, m, n, l_i, L are much smaller than dim and k, hence the time complexity of the proposed model mainly depends on the embedding size and quantity of latent features.

Table 2

Statistics of the datasets, where #doc/user and #doc/item indicate the average number of reviews per user and item respectively, and #word/doc is the average number of words per review.

Dataset	#user	#item	#rating	#doc/user	#doc/item	#word/doc	Density
Musical instruments	1,429	900	10,261	10	17	59	0.798%
Office products	4905	2420	53,228	7	11	56	0.449%
Grocery and gourmet food	14,681	8,713	151,254	10	22	89	0.118%
Video games	24,303	10,672	231,577	9	21	124	0.089%
Sports and outdoors	35,598	18,357	296,337	8	14	99	0.045%

5. Experiments

5.1. Datasets

To validate the effectiveness of the proposed model, we conduct experiments on five publicly available datasets. The five datasets are from different domains of Amazon 5-core [62]: Music Instruments, Office Products, Grocery and Gourmet Food, Video Games, Sports and Outdoors. Following the common protocol in [16], we first process all the characters into the lower cases, and filter out the stop words. After that, we select the 20,000 most frequent words as vocabulary, and remove all the words out of the vocabulary from raw reviews. Finally, following the previous protocol [19], we pad the length and the number of reviews to cover 90% users and items respectively. Following previous work [63], for a user–item pair, we remove the review that the user writes about the item from the review repositories of both of the user and item.

The statistics of the datasets are shown in Table 2. In comparison, previous works [16,17] concatenate all the reviews and represent each user/item with about a hundred words, while we organize the reviews into a review-word hierarchy. Therefore, we are able to preserve rich textual information at different levels of granularities for accurate rating prediction.

5.2. Baselines

To demonstrate the advantage of the proposed model, we compare it with the following state-of-the-art model.

- PMF [4] is a standard matrix factorization that mainly relies on user–item rating for prediction.
- NeuMF [64] integrates generalized matrix factorization and multi-layer perceptron for user–item modeling.
- CDL [34] utilizes SDAE [35] to extract textual features, and hierarchical Bayesian model to integrate ratings and reviews for rating prediction.
- ConvMF [13] leverages CNNs to learn textual features from reviews, and incorporates them into PMF for rating prediction.
- DeepCoNN [14] extracts user and item features from their respective reviews, and uses factorization machine to derive the prediction.
- D-attn [20] proposes dual attention mechanism to achieve the interpretability of latent features of user and item.
- CARL [16] applies CNNs to learn textual features from reviews, and a dynamic linear fusion mechanism for obtaining the final rating scores.
- NARRE [19] leverages two parallel CNNs and an attention mechanism to explore the usefulness of the reviews for rating prediction.
- DAML [17] employs local and mutual attention of CNNs to learn features from reviews, and integrates them with latent factor model for predicting ratings.

5.3. Setup

We tune the hyper-parameters of the proposed model in the validation set. Specifically, we initialize the word embeddings with 100-dimension Glove embeddings [65]. As for the convolution layer, we utilize two-layer CNNs for extracting word vectors. For the first layer CNN, the kernel sizes are set to 3 and 5, and the number of filter maps associated with each kernel are varied amongst [25, 150]. As for the second layer CNN, we use a kernel of size 5, and set the number of filter maps equal to the dimension of latent factors. In the prediction module, we employ two hidden layers and one output layer, and dimension of each hidden layer is halved from the previous layer. To prevent overfitting, we deploy dropout [66] layers upon all non-linear transformations with a dropout rate of 0.5. The implemented model is trained via stochastic gradient descent over shuffled mini-batches with a batch size of 128. The defined objective function is optimized using Adam [61] optimizer with an initial learning rate of 0.0001. All the experiments and training are done using a NVIDIA GeForce GTX 1070 graphics card with 8G memory. To facilitate community research, the implementation of the proposed model is made publicly available from here.[1]

In the experiment, we randomly split each dataset into training set, validation set (10%), and test set (10%). We repeat the experiments for 10 times to avoid splitting bias, and report the average results over the 10 runs. However, we notice very minimal differences among the performances of different runs. Also, to study the influence of training ratio, we vary the percentage of training set amongst [80%, 60%, 40%]. We use two standard metrics for rating predictions, namely: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as shown below.

$$
\begin{aligned}
MAE &= \frac{1}{|\mathcal{T}|} \sum_{(u_i, v_j, r_{ij}) \in \mathcal{T}} |r_{ij} - \hat{r}_{ij}| \\
RMSE &= \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u_i, v_j, r_{ij}) \in \mathcal{T}} (r_{ij} - \hat{r}_{ij})^2}
\end{aligned}
\tag{16}
$$

where $\mathcal{T}$ denotes the test set.

5.4. Performance comparison

The overall performances of the models across the datasets are presented in Tables 3–5 with different training ratio, from which we have the following conclusions. First, PMF yields the worst performance across the datasets. PMF is mainly based on user–item interactions, hence the performance is limited by the sparsity instinct of the rating matrix in recommendation problems. NeuMF is also the model that relies on the rating data for prediction. However, it outperforms PMF by a large margin on all the datasets, demonstrating the effectiveness of multi-layer nonlinear transformations for capturing complex user–item relations. The advantage of nonlinear interactions can also be validated by NeuMF's superiority to CDL across the datasets.

---

[1] https://github.com/uqjwen/HIGAN

Table 3
Performance comparison across the dataset for all models (**80% training ratio**). The best and the second-best results are highlighted by boldface and underlined respectively.

Datasets	Metrics	PMF	NeuMF	CDL	ConvMF	DeepCoNN	D-attn	NARRE	CARL	DAML	HTI
Musical instruments	MAE	1.137	0.72	0.834	0.786	0.759	0.742	0.695	0.677	<u>0.651</u>	**0.611**
	RMSE	1.352	0.904	1.080	1.026	1.003	0.956	0.922	0.878	<u>0.848</u>	**0.813**
Office products	MAE	1.265	0.73	1.062	0.728	0.711	0.716	0.681	0.647	<u>0.612</u>	**0.552**
	RMSE	1.430	0.921	1.223	0.952	0.901	0.923	0.867	0.834	<u>0.811</u>	**0.731**
Grocery and gourmet food	MAE	1.397	0.943	0.967	0.863	0.802	0.824	0.747	0.753	<u>0.735</u>	**0.672**
	RMSE	1.572	1.197	1.190	1.092	1.036	1.070	0.963	0.961	<u>0.938</u>	**0.877**
Video games	MAE	1.395	0.87	0.902	0.899	0.875	0.842	0.799	0.798	<u>0.788</u>	**0.731**
	RMSE	1.606	1.103	1.165	1.145	1.168	1.062	1.039	<u>1.029</u>	1.045	**0.966**
Sports and outdoors	MAE	1.203	0.752	0.852	0.824	0.719	0.784	0.69	0.686	<u>0.667</u>	**0.629**
	RMSE	1.376	0.979	1.089	1.013	0.885	0.997	<u>0.882</u>	0.888	0.883	**0.826**

Table 4
Performance comparison across the dataset for all models (**60% training ratio**). The best and the second-best results are highlighted by boldface and underlined respectively.

Datasets	Metrics	PMF	NeuMF	CDL	ConvMF	DeepCoNN	D-attn	NARRE	CARL	DAML	HTI
Musical instruments	MAE	1.160	0.751	0.860	0.805	0.778	0.767	0.709	0.695	<u>0.670</u>	**0.623**
	RMSE	1.377	0.928	1.103	1.035	1.028	0.973	0.928	0.898	<u>0.864</u>	**0.826**
Office products	MAE	1.281	0.743	1.100	0.751	0.735	0.737	0.681	0.660	<u>0.631</u>	**0.573**
	RMSE	1.454	0.943	1.261	0.999	0.932	0.963	0.885	0.860	<u>0.835</u>	**0.747**
Grocery and gourmet food	MAE	1.415	0.966	0.984	0.885	0.819	0.841	0.760	0.785	<u>0.755</u>	**0.691**
	RMSE	1.590	1.216	1.232	1.117	1.049	1.094	0.996	0.978	<u>0.954</u>	**0.900**
Video games	MAE	1.438	0.908	0.919	0.914	0.892	0.860	0.807	0.816	<u>0.802</u>	**0.742**
	RMSE	1.679	1.129	1.191	1.156	1.214	1.090	1.059	<u>1.054</u>	1.061	**0.986**
Sports and outdoors	MAE	1.227	0.762	0.878	0.832	0.724	0.822	0.695	0.696	<u>0.695</u>	**0.641**
	RMSE	1.400	1.005	1.124	1.036	0.908	1.039	<u>0.890</u>	0.898	0.902	**0.847**

Table 5
Performance comparison across the dataset for all models (**40% training ratio**). The best and the second-best results are highlighted by boldface and underlined respectively.

Datasets	Metrics	PMF	NeuMF	CDL	ConvMF	DeepCoNN	D-attn	NARRE	CARL	DAML	HTI
Musical instruments	MAE	1.181	0.749	0.882	0.817	0.793	0.778	0.721	0.704	<u>0.680</u>	**0.633**
	RMSE	1.434	0.952	1.157	1.075	1.047	0.999	0.962	0.919	<u>0.886</u>	**0.851**
Office products	MAE	1.317	0.767	1.107	0.758	0.740	0.742	0.696	0.675	<u>0.637</u>	**0.579**
	RMSE	1.493	0.953	1.265	0.983	0.948	0.975	0.907	0.884	<u>0.847</u>	**0.767**
Grocery and gourmet food	MAE	1.467	0.991	1.003	0.907	0.843	0.852	<u>0.778</u>	0.789	0.790	**0.701**
	RMSE	1.693	1.271	1.243	1.170	1.085	1.100	1.016	1.023	<u>0.992</u>	**0.916**
Video games	MAE	1.468	0.905	0.944	0.941	0.904	0.885	0.845	0.840	<u>0.821</u>	**0.780**
	RMSE	1.693	1.150	1.205	1.194	1.199	1.117	1.099	<u>1.085</u>	1.092	**1.005**
Sports and outdoors	MAE	1.255	0.782	0.884	0.851	0.766	0.820	0.718	0.721	<u>0.705</u>	**0.656**
	RMSE	1.427	1.011	1.166	1.052	0.930	1.044	0.930	0.934	<u>0.925</u>	**0.875**

Even though CDL exploits textual features with SDAE, it simply incorporates those features into a hierarchical Bayesian model. Therefore, it fails to capture high-order user–item correlations in the recommendation datasets.

As for review-based models, ConvMF achieves better performance than CDL. Since they both utilize shallow network for modeling user–item interactions, the performance improvement can be attributed to the CNNs, which are efficient in capturing local semantics of different levels of granularities. DeepCoNN demonstrates better performance than ConvMF, implying the advantage of factorization machine in capturing high-order nonlinear interactions of features. The performance differences between D-attn and DeepCoNN are data-dependent. Specifically, D-attn presents performance gains over DeepCoNN on *Musical Instruments* and *Video Games*. This portrays the success of the attention mechanism in selecting informative words for rating predictions. However, on *Office Products* and *Grocery and Gourmet Food*, DeepCoNN yields slightly better performance than D-attn, probably because D-attn lacks the expressiveness to capture complex high-order user–item interactions.

NARRE demonstrates improved performance over D-attn, indicating the advantage of exploiting the textual information in a hierarchical manner. D-attn concatenates all the reviews of a user/item into a single review, where the introduced noise and irrelevant information in the reviews can negatively affect the model performance. On the contrary, NARRE extracts textual features for the reviews in parallel, and proposes an attention mechanism to select the most informative reviews for prediction. CARL and DAML share similar model architecture, and they present significant improved performances over NARRE. The comparison results suggest the advantage of modeling the pair-based relevance between users and items, as it can dynamically capture complex interactions for each specific user–item pair.

The proposed HTI model achieves the best MAE/RMSE scores across the five datasets under different training ratios. Specifically, HTI achieves an average improvement of 7.48% in MAE (7.2% in RMSE) over the best baseline. Compared with existing models, NARRE is a hierarchical model that exploits both word- and review-level textual information for learning review features. However, at word level, it takes a max pooling over the word vectors to comprise a review representation. At review level, it utilizes a self-attention mechanism to aggregate the reviews into latent features. One limitation in the max pooling at word level

is information loss, namely the most salient feature can override the rest features in the same dimension of a review. In addition, the self-attention mechanism at review level defers the user–item interactions until the prediction layer, making it fail to capture complex user–item correlations. On the contrary, at word level, we propose a pair-specific attention mechanism that can capture informative words for comprising review representations. At review level, we allow textual features to be propagated between each user–item pair. Therefore, we model semantic alignments at review level for modeling complex user–item interactions. The comparison between HTI and previous models such as DAML demonstrates the effectiveness of the proposed hierarchical architecture. Those models simply concatenate all reviews of a user/item into a single review. Therefore, the irrelevant words can introduce noises and undermine the recommendation accuracy. In addition, they model aspect relevance at word/phrase level, which does not necessarily reflect the meaning of the original sentences [21]. By contrast, we propose to model textual features at both word and review levels, which allow us to capture semantic correlations at different levels of granularities for accurate prediction.

Comparing the recommendation performance of competitive models with different training rations, we can see that the proposed model is able to outperform the baselines under various data sparsity, showing its effectiveness in exploiting hierarchical and interactive textual features for alleviating data sparsity and increasing recommendation performance.

5.5. Analysis of HTI model

5.5.1. Effect of two learning modules

To study the effect of the word-level attention module and review-level interaction module, we compare HTI with its variants shown as follows,

   i HTI-wavg obtains each review representation by applying a mean-pooling layer over the word vectors.

  ii HTI-wmax obtains each review representation by applying a max-pooling layer over the word vectors.

 iii HTI-davg obtains the aggregated representations by applying a mean-pooling layer over the review representations.

 iv HTI-dmax obtains the aggregated representations by applying a max-pooling layer over the review representations.

The comparison results on the five datasets are presented in Table 7. From the table, we can see that HTI significantly outperforms its variants, demonstrating the effectiveness of modeling text interaction at different levels of granularities for rating prediction. Second, the max pooling layer yields slightly better MAE over the mean pooling layer on the datasets, as max-pooling operation is able to capture the most salient features [67]. Finally, the variants without review-level interactions consistently perform worse than the variants exclude word-level attentions, suggesting the importance of capturing semantic correlations between users and items at the review level. We believe the reasons are two-fold. First, pooling over the review representations does not allow HTI to model text interaction at the review level, making the model unable to capture pair-wise textual alignments and extracting task-specific textual features. Second, words in a review tend to have similar sentimental polarity, but this is not the case at the review level, as users may write reviews of different polarities for different items. Therefore, pooling reviews of contradictory polarities incurs more recommendation degradation then pooling over words of similar polarities.

The results indicate that the proposed model can effectively preserve textual features at different levels of hierarchies. The significant improvement over HTI-davg and HTI-dmax demonstrates the advantage of exploiting mutual correlations at review level to derive accurate recommendations.
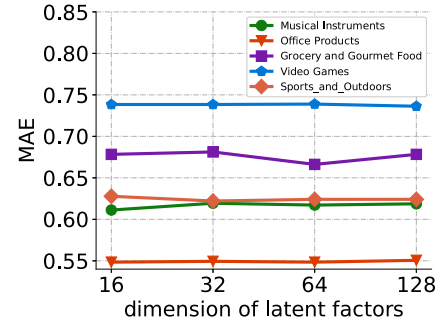

Fig. 2. Recommendation performance (MAE) with respect to different number of latent dimensions.

5.5.2. Number of latent dimension

To study the sensitivity of HTI with respect to the number of latent factors, we vary the dimension amongst [16, 32, 64, 128]. As shown in Fig. 2, the hyperparameter has little impact on the model performance, as HTI is able to provide a relatively stable MAE with respect to different dimensions. However, as we analyze in Section 4.6, the time complexity of HTI is related to the latent factor dimensions, a small latent dimension is selected to reduce computational overhead in this work.

5.5.3. Hyperparameters of CNNs

The CNNs for extracting local semantics have two important parameters, namely filter size, and the number of filter maps. We vary the filter size amongst [2, 3, 4, 5, 6] for the first and second layer CNNs. As for the number of filter maps, we vary the hyper-parameter amongst [25, 50, 100, 200, 300] for the first layer CNN and set it equal to the number of latent factors for the second. As depicted in Fig. 3, we only show the MAE with respect to different CNNs parameters, as RMSE presents the similar trend. We can see (i) the proposed model experiences an average of 2.13% in performance variants under different CNNs parameters, demonstrating its robustness for achieving consistent recommendation performance; (ii) the proposed model generally achieves the optimal results with larger filter size and more filter numbers, showing that more parameters are required for increasing the model representativeness and expressiveness; (iii) the performance on *Musical Instruments* and *Office Products* begins to degrade with more filter maps, possibly because the model gradually overfits the small datasets with complex CNNs parameters.

5.5.4. Time complexity analysis

To study the time efficiency of the proposed model, we compare its running time with some representative baselines. We record the running time across the models on the same computing environment, and set the hyperparameters according to the original works. Table 6 shows the training and prediction time for the deep models. The training time is the time for training one epoch of the data, and the prediction time is the time required to complete the prediction for the whole testing set. We can observe that PMF and NeuMF take the least time to finish the training and testing, as those two models do not involve text-related tasks. CDL and ConMF only incorporate simple text modeling on the item side, and they incur the least overhead among the text-based models. D-attn needs to calculate local attentions with kernels of various sizes, leading to the slowest model. CARL and DAML share the similar architecture, therefore, they have similar training and testing costs. NARRE is a hierarchical model, however, it does not model hierarchical text alignments. Therefore, NARRE runs faster than the proposed model in terms of training and testing.
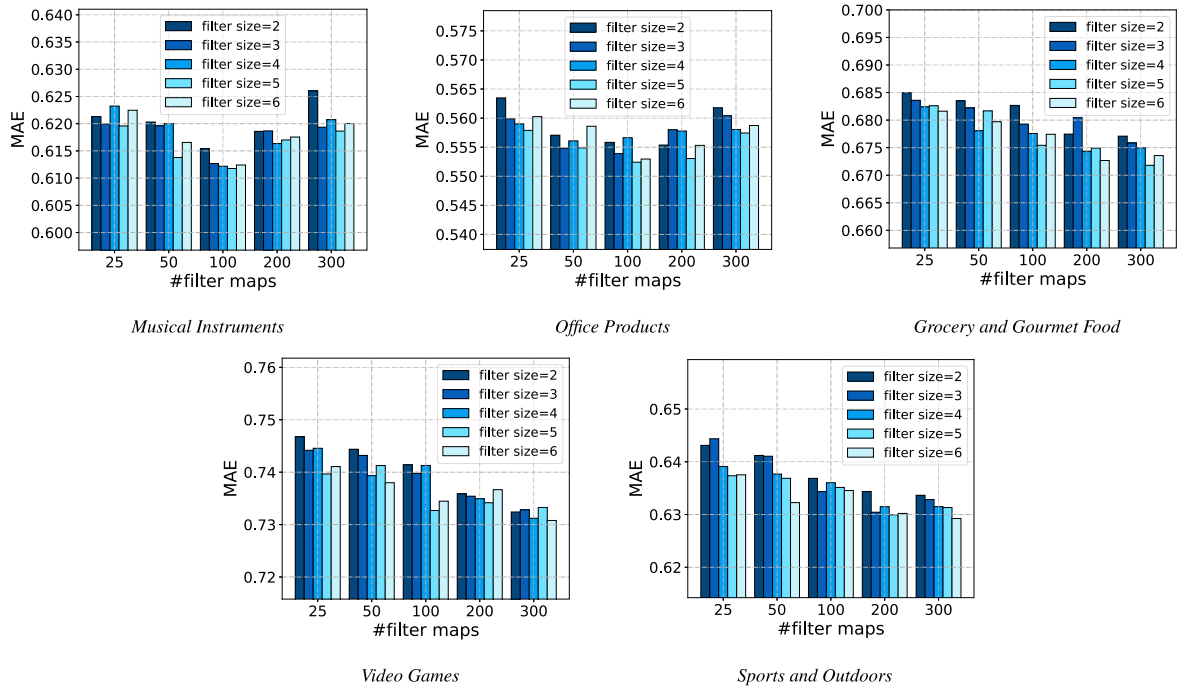
Fig. 3. MAE with respect to different combinations of filter size and number of filter maps across the datasets.

Table 6

Running time in seconds of the competitive models across the datasets. The training time is the time for training an epoch of the data.

Models	Musical instruments		Office products		Grocery and gourmet food		Video games		Sports and outdoors	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
PMF	0.616	0.031	3.370	0.142	10.098	0.406	15.399	0.768	20.448	1.022
NeuMF	0.812	0.040	4.060	0.165	11.719	0.587	18.729	0.765	26.743	1.540
CDL	2.053	0.091	10.566	0.584	31.452	1.218	53.148	2.500	70.675	2.677
ConMF	2.425	0.099	12.858	0.682	42.649	1.945	53.318	2.826	70.118	3.879
DeepCoNN	3.660	0.194	22.484	1.122	63.477	3.182	85.095	4.910	128.966	6.534
D-attn	10.118	0.590	59.501	3.128	164.936	9.124	211.647	13.608	338.507	17.912
CARL	4.961	0.218	29.619	1.122	86.245	3.340	110.168	4.910	172.168	6.644
NARRE	4.282	0.250	21.408	0.954	74.376	2.872	108.362	5.239	144.203	7.196
DAML	4.983	0.193	25.379	1.212	87.672	4.525	114.934	6.484	171.536	9.489
HTI	5.365	0.219	26.825	1.262	77.425	3.853	117.943	5.893	151.441	7.586

Table 7

Comparison results between HTI and its variants. HTI-wavg: replaces the word-level attention layer with an average pooling layer. HTI-wmax: replaces the word-level attention layer with a max pooling layer. HTI-davg: replaces the review-level interactive layer with an average pooling layer. HTI-dmax: replaces the review-level interactive layer with a max pooling layer.

Variants	Musical instruments	Office products	Grocery and gourmet food	Video games	Sports and outdoors
HTI-wavg	0.632	0.584	0.723	0.762	0.646
HTI-wmax	0.618	0.563	0.684	0.742	0.636
HTI-davg	0.669	0.635	0.788	0.782	0.690
HTI-dmax	0.654	0.619	0.760	0.793	0.669
HTI	0.611[a]	0.552[a]	0.672[a]	0.731[a]	0.629[a]

[a] a indicates the comparison results are statistically significant at the level of 0.001.

Comparing HTI with the state-of-the-art DAML, we can see the overhead differences between them are data-dependent. For example, DAML takes less time than HTI on *Musical Instruments* and *Video Games*. By contrast, HTI yields lower overhead than DAML on *Grocery and Gourmet Food* and *Sports and Outdoors*. The experiment shows that HTI is able to achieve the best recommendation accuracy without incurring noticeable overhead.
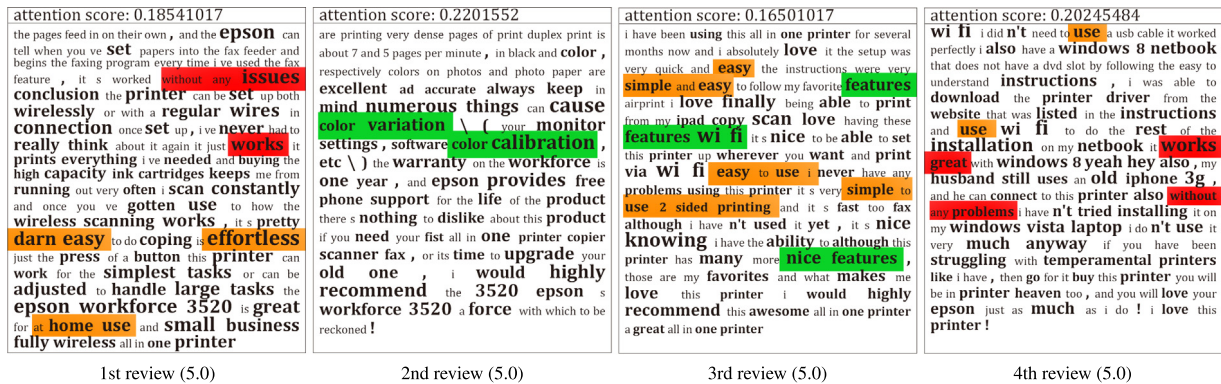
5.6. Visualization

In this subsection, to investigate whether HTI can capture text correlations at different levels of hierarchies, we visualize the attention scores both in word-level and review-level. We randomly sample two rating records having the same user, but with different rating scores. A rating record is denoted as (u_i, v_j, r_{ij}) such that the user u_i rates the item v_j with a score r_{ij}. We visualize the top-4 reviews having the largest attention scores. The attention scores are visualized by the fontsize of the words in the reviews, with large fontsize representing high attentions and small fontsize denoting low values.

Fig. 4 and Fig. 5 show the visualizations of user and item reviews of the first tuple (#4492, #1027, 5), respectively. Fig. 6 and Fig. 7 show the visualizations of the user and item reviews of the second tuple (#4492, #856, 1), respectively. The attention
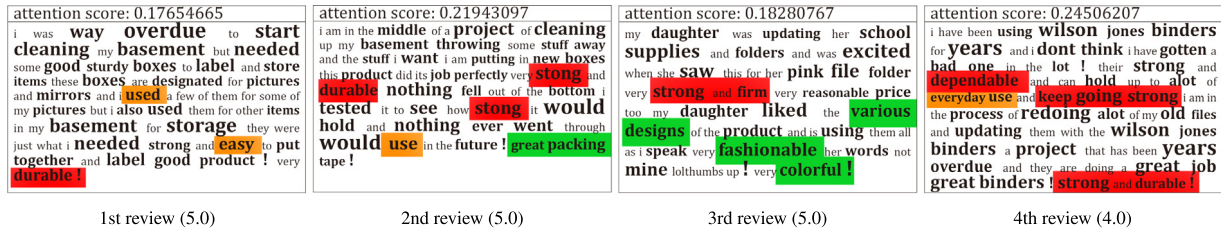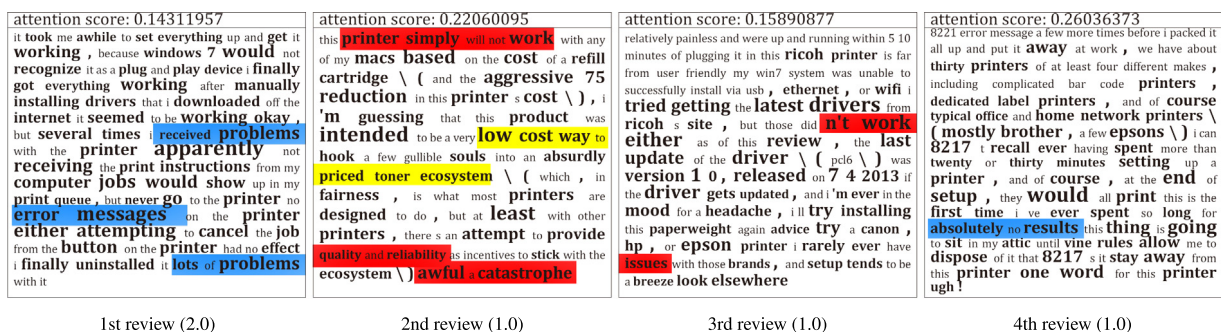
Fig. 4. Visualization of the **user reviews** given the tuple **(#4492, #1027, 5.)**, where the ground truth rating is 5 (**positive rating**). The attentions over the review are shown at the top of each review, while the attentions over the words are indicated by the font size. The ground-truth ratings of those reviews are shown in the parentheses of the sub-captions. The colors are used to present the word correlations between the user and item reviews, with each color indicating a potential aspect under which the words semantically match. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 5. Visualization of the **item reviews** given the tuple **(#4492, #1027, 5.)**, where the ground truth rating is 5 (**positive rating**). The attentions over the review are shown at the top of each review, while the attentions over the words are indicated by the font size. The ground-truth ratings of those reviews are shown in the parentheses of the sub-captions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
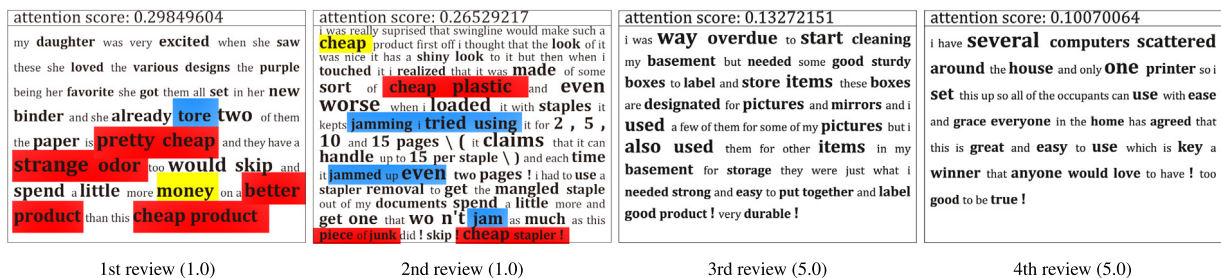
Fig. 6. Visualization of the **user reviews** given the tuple **(#4492, #856, 1.)**, where the ground truth rating is 1 (**negative rating**). The attentions over the review are shown at the top of each review, while the attentions over the words are indicated by the font size. The ground-truth ratings of those reviews are shown in the parentheses of the sub-captions. Notice that the later two reviews have opposite sentiment, however, they are placed with low weights. Therefore, we focus our discussions on the former two reviews. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
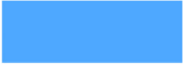

Fig. 7. Visualization of the **item reviews** given the tuple **(#4492, #856, 1.)**, where the ground truth rating is 1 (**negative rating**). The attentions over the review are shown at the top of each review, while the attentions over the words are indicated by the font size. The ground-truth ratings of those reviews are shown in the parentheses of the sub-captions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 8
The correspondence between the colors in Figs. 4–7 and the potential aspect under which the words are correlated.

color	potential aspect
🟥	quality
🟧	usability
🟩	feature
🟦	functionality
🟨	price

scores are list at the top of each review, while the ground-truth ratings (labels) are shown in the parentheses of the review captions.

For the word-level visualization, we can see that informative words are placed with high attention scores across the reviews. For example, in Fig. 4, the words "durable", "great packing", "fashionable", "dependable" are assigned with large attention scores. Those words are informative, as they have strong indications of the user's preference. In Fig. 5, the words "easy", "nice features", "work great" receive high attentions. Those words reflect different characteristics of the item.

As for the review-level visualization, given the positive tuple (#4492, #1027, 5), it is clear to see that the proposed model can place higher attention scores to positive reviews for rating prediction, as shown in Figs. 4 and 5. Similarly, given the negative tuple (#4492, #856, 1), we are able to locate the negative reviews for recommendation, as presented in Figs. 6 and 7. An exception can be found in the user reviews of the negative tuple, where the first two reviews show negative sentiment, while last two reviews present positive sentiment. However, HTI attributes high attention scores to the first two reviews and low scores to the others, demonstrating that HTI can capture informative textual features at the review level given a specific user–item pair.

It is also interesting to find the word correlations between the user and item reviews. In Figs. 4–7, we use color to highlight word correlations, with each color indicating a potential aspect under which the word are correlated. The mappings between the colors and the potential aspects are shown in Table 8. Specifically, for the positive tuple (#4492, #1027, 5), red words such as "durable" and "problem" imply the semantic interrelations between the user and item with respect to the potential quality aspect, while orange words such as "easy" and "use" depict semantic matching under the potential usability aspect. As for the negative tuple (#4492, #856, 1), we can find semantic correlations between the user and item in the functionality aspect, as indicated by the blue words such as "jamming", "tore" and "error message".

In this case study, we can see that HTI can highlight words with strong sentimental indications at the word level. At the review level, HTI can dynamically assign different attention scores to the reviews given different items. In addition, it is able to capture semantic correlations at review level for prediction. To conclude, the proposed model can exploit information textual features at different levels of hierarchies, and capture the mutual semantic correlations between specific user–item pairs to derive accurate recommendations.

6. Conclusion and future work

In this article, we propose a novel hierarchical text interaction model, named HTI, that exploits textual information from reviews for rating prediction. The essence of HTI is to model semantic correlations at different level of granularities. Two core components of HTI are word-level attention module and review-level interaction module. The former captures important words for comprising a review, while the later models text interaction and identifies the informative reviews. The experimental results on five real-world datasets show that HTI achieves significantly better recommendation accuracies than the state-of-the-art models. In addition, we validate the effectiveness of the hierarchical architecture. The validation is completed by comparing HTI with its variants that exclude the attention at word level and interaction at review level. To provide a better understanding of HTI's efficacy in capturing semantic correlations at different granularities, we visualize attentions at both word and review levels for randomly sampled user–item pairs.

In this article, we mainly focus on hierarchical text interaction for improving recommendation performance. However, in the proposed model, we linearly combine text representation with latent factors for rating prediction, ignoring personalized text usefulness among the users, as not all users equally depend on the reviews for the recommendation. In addition, the proposed HTI model only involves single data modality for the recommendation task. As part of future work, we plan to investigate the aforementioned factors for rating prediction, such as the modeling of personalized text usefulness, interaction across heterogeneous data modalities [68], etc.

CRediT authorship contribution statement

Jiahui Wen: Conceptualization, Writing - original draft. **Jingwei Ma:** Writing - review & editing, Supervision. **Hongkui Tu:** Methodology, Visualization. **Mingyang Zhong:** Figures, Tables, Visualization. **Guangda Zhang:** Programming, Data preparation. **Wei Yin:** Baseline implementation. **Jian Fang:** Formal analysis.

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.

Acknowledgment

References

[1] Q. Zhang, D. Wu, J. Lu, F. Liu, G. Zhang, A cross-domain recommender system with consistent information transfer, Decis. Support Syst. 104 (dec.) (2017) 49–63.
[2] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Adv. Artif. Intell. 2009 (2009).
[3] 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, ACM, 2008, pp. 426–434.
[4] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, in: Advances in Neural Information Processing Systems, 2008, pp. 1257–1264.
[5] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 2001, pp. 556–562.
[6] X. He, C. Tao, M.Y. Kan, C. Xiao, TriRank: Review-aware explainable recommendation by modeling aspects, in: The 24th ACM on Conference on Information and Knowledge Management, 2015.

[7] C. Cui, H. Liu, T. Lian, L. Nie, L. Zhu, Y. Yin, Distribution-oriented aesthetics assessment with semantic-aware hybrid network, IEEE Trans. Multimedia 21 (5) (2019) 1209–1220.

[8] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (Jan) (2003) 993–1022.

[9] C. Wang, D.M. Blei, Collaborative topic modeling for recommending scientific articles, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 448–456.

[10] J. McAuley, J. Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in: Proceedings of the 7th ACM Conference on Recommender Systems, ACM, 2013, pp. 165–172.

[11] L. Zhu, Z. Huang, X. Liu, X. He, J. Sun, X. Zhou, Discrete multimodal hashing with canonical views for robust mobile landmark search, IEEE Trans. Multimedia 19 (9) (2017) 2066–2079.

[12] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[13] D. Kim, C. Park, J. Oh, S. Lee, H. Yu, Convolutional matrix factorization for document context-aware recommendation, in: Proceedings of the 10th ACM Conference on Recommender Systems, ACM, 2016, pp. 233–240.

[14] 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, pp. 425–434.

[15] S. Rendle, Factorization machines, in: 2010 IEEE International Conference on Data Mining, IEEE, 2010, pp. 995–1000.

[16] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, X. Luo, A context-aware user-item representation learning for item recommendation, ACM Trans. Inf. Syst. 37 (2) (2019) 22.

[17] D. Liu, J. Li, B. Du, J. Chang, R. Gao, DAML: Dual attention mutual learning between ratings and reviews for item recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &Amp; Data Mining, ACM, 2019, pp. 344–352.

[18] Z. Cheng, X. Chang, L. Zhu, R. Catherine Kanjirathinkal, M.S. Kankanhalli, MMALFM: explainable recommendation by leveraging reviews and images, ACM Trans. Inf. Syst. 37 (2) (2019) 16:1–16:28.

[19] 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, International World Wide Web Conferences Steering Committee, 2018, pp. 1583–1592.

[20] S. Seo, J. Huang, H. Yang, Y. Liu, Interpretable convolutional neural networks with dual local and global attention for review rating prediction, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, ACM, 2017, pp. 297–305.

[21] M.T. Ribeiro, S. Singh, C. Guestrin, Why should I trust you?: Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1135–1144.

[22] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2019.

[23] 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.

[24] 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. 30 (7) (2019) 1998–2012.

[25] Y. Han, L. Zhu, Z. Cheng, J. Li, X. Liu, Discrete optimal graph clustering, IEEE Trans. Cybern. 50 (4) (2020) 1697–1710.

[26] C. Chen, M. Zhang, C. Wang, W. Ma, M. Li, Y. Liu, S. Ma, An efficient adaptive transfer neural network for social-aware recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2019, pp. 225–234.

[27] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, in: The World Wide Web Conference, ACM, 2019, pp. 417–426.

[28] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, M. Wang, A neural influence diffusion model for social recommendation, 2019, arXiv preprint arXiv:1904.10322.

[29] Z. Zhang, Z. Lai, Z. Huang, W.K. Wong, G. Xie, L. Liu, L. Shao, Scalable supervised asymmetric hashing with semantic and latent factor embedding, IEEE Trans. Image Process. 28 (10) (2019) 4803–4818.

[30] Y. Zhang, Q. Ai, X. Chen, W.B. Croft, Joint representation learning for top-n recommendation with heterogeneous information sources, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, 2017, pp. 1449–1458.

[31] X. Lu, L. Zhu, J. Li, H. Zhang, H.T. Shen, Efficient supervised discrete multi-view hashing for large-scale multimedia search, IEEE Trans. Multimed. (2019) 1, http://dx.doi.org/10.1109/TMM.2019.2947358.

[32] 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, ACM, 2014, pp. 105–112.

[33] Y. Bao, H. Fang, J. Zhang, Topicmf: Simultaneously exploiting ratings and reviews for recommendation, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.

[34] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1235–1244.

[35] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (Dec) (2010) 3371–3408.

[36] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer (8) (2009) 30–37.

[37] J. Wen, Z. Zhang, Z. Zhang, L. Fei, M. Wang, Generalized incomplete multiview clustering with flexible locality structure diffusion, IEEE Trans. Cybern. PP (99) (2020) 1–14.

[38] F. Zhang, N.J. Yuan, D. Lian, X. Xie, W.-Y. Ma, Collaborative knowledge base embedding for recommender systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 353–362.

[39] D. Shi, L. Zhu, Y. Li, J. Li, X. Nie, Robust structured graph clustering, IEEE Trans. Neural Netw. Learn. Syst. (2019) 1–13, http://dx.doi.org/10.1109/TNNLS.2019.2955209.

[40] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1746–1751.

[41] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Vol. 1: Long Papers, 2014, pp. 655–665.

[42] L. Zhu, Z. Huang, Z. Li, L. Xie, H.T. Shen, Exploring auxiliary context: Discrete semantic transfer hashing for scalable image retrieval, IEEE Trans. Neural Netw. Learn. Syst. 29 (11) (2018) 5264–5276.

[43] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2014, arXiv preprint arXiv:1409.0473.

[44] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1480–1489.

[45] Z. Li, Z. Zhang, J. Qin, Z. Zhang, L. Shao, Discriminative Fisher embedding dictionary learning algorithm for object recognition, IEEE Trans. Neural Netw. 31 (3) (2020) 786–800.

[46] Z. Zhang, L. Liu, F. Shen, H.T. Shen, L. Shao, Binary multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. 41 (7) (2019) 1774–1782.

[47] Y. Lu, R. Dong, B. Smyth, Coevolutionary recommendation model: Mutual learning between ratings and reviews, in: Proceedings of the 2018 World Wide Web Conference, International World Wide Web Conferences Steering Committee, 2018, pp. 773–782.

[48] 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.

[49] J. Li, K. Lu, Z. Huang, L. Zhu, H.T. Shen, Heterogeneous domain adaptation through progressive alignment, IEEE Trans. Neural Netw. Learn. Syst. 30 (5) (2019) 1381–1391.

[50] C. Ma, P. Kang, X. Liu, Hierarchical gating networks for sequential recommendation, 2019, arXiv preprint arXiv:1906.09217.

[51] 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.

[52] L. Wu, L. Chen, R. Hong, Y. Fu, X. Xie, M. Wang, A hierarchical attention model for social contextual image recommendation, IEEE Trans. Knowl. Data Eng. (2019).

[53] D. Cao, X. He, L. Miao, Y. An, C. Yang, R. Hong, Attentive group recommendation, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2018, pp. 645–654.

[54] W. Wang, G. Zhang, J. Lu, Member contribution-based group recommender system, Decis. Support Syst. 87 (jul) (2016) 80–93.

[55] M. Jingwei, W. Jiahui, Z. Mingyang, L. Liangchen, L. Chaojie, C. Weitong, Y. Yin, T. Honghui, X. Li, DBRec: Dual-bridging recommendation via discovering latent groups, in: Proceedings of the 2019 ACM on Conference on Information and Knowledge Management, ACM, 2019.

[56] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, X. Cheng, Learning hierarchical representation model for nextbasket recommendation, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 403–412.

[57] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 315–323.

[58] L. Zhu, X. Lu, Z. Cheng, J. Li, H. Zhang, Flexible multi-modal hashing for scalable multimedia retrieval, ACM Trans. Intell. Syst. Technol. 11 (2) (2020).

[59] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, T.-S. Chua, Attentional factorization machines: learning the weight of feature interactions via attention networks, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, AAAI Press, 2017, pp. 3119–3125.

[60] 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, ACM, 2017, pp. 345–354.

[61] D. Kingma, J. Ba, Adam: A method for stochastic optimization, Comput. Sci. (2014).

[62] 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, International World Wide Web Conferences Steering Committee, 2016, pp. 507–517.

[63] R. Catherine, W. Cohen, Transnets: Learning to transform for recommendation, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, ACM, 2017, pp. 288–296.

[64] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[65] 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.

[66] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. (2014) 1929–1958.

[67] W. Xue, T. Li, Aspect based sentiment analysis with gated convolutional networks, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Vol. 1: Long Papers, 2018, pp. 2514–2523.

[68] J. Li, K. Lu, Z. Huang, L. Zhu, H.T. Shen, Transfer independently together: A generalized framework for domain adaptation, IEEE Trans. Cybern. 49 (6) (2019) 2144–2155.