International Journal of Web Information Systems

205

MSCA: a multi-scale context-aware recommender system leveraging reviews and user interactions

Zhangyu Jin, Jian Wang, Duantengchuan Li, Maodong Li and Bing Li
School of Computer Science, Wuhan University, Wuhan, China

Received 26 October 2024
Revised 7 December 2024
Accepted 11 December 2024

Abstract

Purpose – Reviews, serving as authentic user feedback, encompass rich semantic information, including descriptions of user preferences and item characteristics. Introducing reviews as auxiliary information into recommendation systems can enhance the modeling of user and item feature representations. However, existing methods insufficiently account for the semantic variations of review words across different contextual scales, often leading to suboptimal representations of review information. In addition, they fail to effectively integrate review features with interaction features, resulting in a semantic gap between the two modalities. To address these issues, this paper aims to propose a novel review-based recommendation model that incorporates a multi-scale context-aware network and a cross-attention mechanism (MSCA).

Design/methodology/approach – MSCA uses a multi-scale convolutional network to capture the semantic representations of words in reviews across different contextual scales. It then uses a self-attention layer to learn the associations among reviews, enhancing the representation of review features. Subsequently, MSCA deploys a cross-attention mechanism to emulate personalized attention across different users and items, effectively fusing interaction features with the semantic features of reviews.

Findings – Experiments on five Amazon review data sets demonstrate that the proposed model outperforms baseline methods in terms of evaluation metrics.

Originality/value – The authors propose MSCA, a novel recommendation model that more effectively extracts review features using a multi-scale context-aware network. It also uses a cross-attention mechanism to fuse semantic information with interaction features from two modalities, thereby improving the performance of recommendation systems.

Keywords Review-based recommender, User preferences, Attention mechanism, Rating prediction

Paper type Research paper

1. Introduction

Recommendation systems (Wu *et al.*, 2022, 2023; Liang *et al.*, 2024), as effective tools to alleviate information overload, are extensively employed across scenarios like e-commerce websites, social platforms and video media. Numerous existing approaches commonly rely solely on user–item interaction data to model personalized user preferences. Primarily based on collaborative filtering, these recommendation methods use techniques like matrix factorization on historical rating data to acquire representations of user preferences and characteristics of the items, yielding commendable performance in recommendation tasks. However, these methods face a data sparsity challenge, which is due to the continuously growing number of items and extremely sparse user interaction information. To tackle this issue, an increasing number of researchers are integrating diverse auxiliary information like user reviews (Liu *et al.*, 2022b; Li *et al.*, 2021, 2024a), social networks (Liu *et al.*, 2022c,

Funding: This study was funded by National Natural Science Foundation of China: No. 62032016.

International Journal of Web Information Systems
Vol. 21 No. 3, 2025
pp. 205-229

1744-0084
DOI 10.1108/IJWIS-10-2024-0311

2023a, 2023b) and knowledge graphs (Wang *et al.*, 2023a) into recommendation systems. This integration aims to enrich representations of user preferences and characteristics of the items, ultimately improving model accuracy. User reviews contain rich semantic information. They typically outline user preferences, encompassing their diverse interests in different aspects of items. Hence, in scenarios of extreme scarcity in user interaction data, further modeling of user preferences and characteristics of the items can be achieved through review data.

Figure 1 illustrates the significance of user reviews for recommendations. To predict user Alice's rating for the "Hat", the recommendation system holistically models Alice's preferences and the characteristics of the Hat, using a prediction module to output a rating. Traditional collaborative filtering methods learn interaction features from the rating matrix. Nevertheless, as a single number, a rating provides limited information and cannot convey the user's detailed feelings toward the item. Especially when two items have the same ratings, the distinctions between the two are imperceptible, thus constraining the performance of the recommendation system. In contrast, review texts are rich with semantic information, offering multifaceted portrayals of items regarding their utility, quality and size, along with expressions of the users' preferences, emotions and needs. By leveraging the

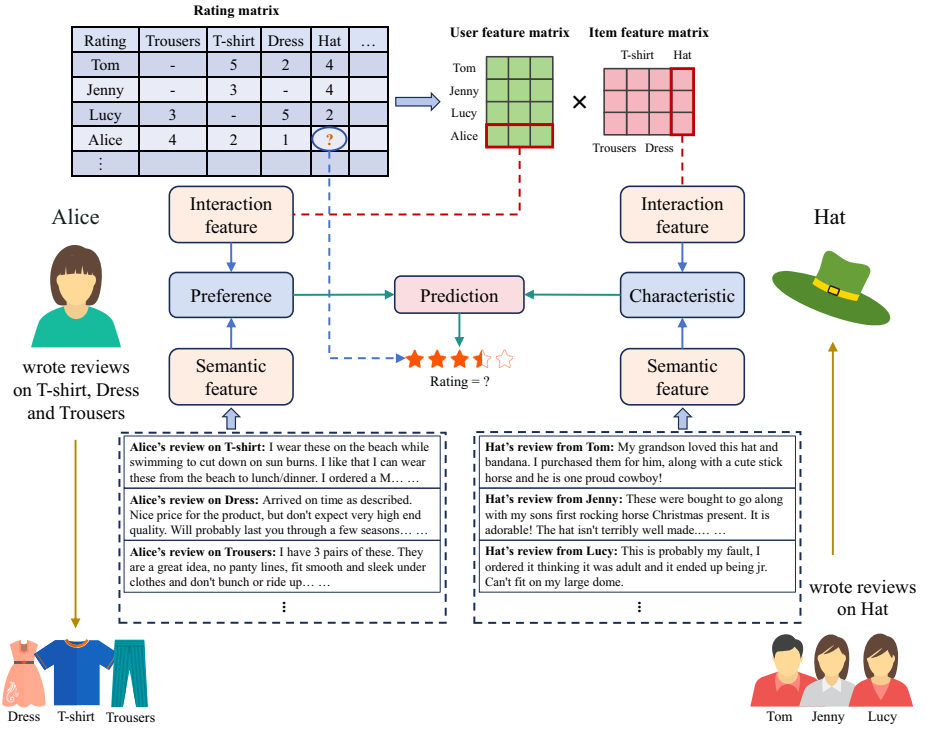


Figure 1. Example of a recommender system integrating review information and user interactions
Source(s): Authors' own work

semantic information from reviews, the recommendation system can more precisely capture user preferences and item characteristics, thereby enhancing the accuracy of recommendation systems.

In the early stages, recommendation methods relying on review information primarily used topic models to learn topic distributions and build semantic features for users and items. These methods (McAuley and Leskovec, 2013; Ling *et al.*, 2014; Bao *et al.*, 2014) successfully integrated review information into recommendation systems. However, a limitation of topic models is their inability to consider the sequential order of words and the contextual connections between them. In the past few years, deep learning techniques have been used for feature extraction from user reviews. DeepCoNN (Zheng *et al.*, 2017) uses CNN to extract semantic features from reviews, yielding promising outcomes. However, it treated all parts of reviews equally, making them vulnerable to irrelevant reviews unrelated to user preferences or item features. Subsequent studies, exemplified by Seo *et al.* (2017), Chen *et al.* (2018) and Liu *et al.* (2019, 2022a), use attention mechanisms to assign different weights to review features, which improve the extraction of valuable features and filter out less crucial features. Despite advances, they rely on pretrained word embedding models or fixed-size convolutions to extract local features, without accounting for the fact that words can have different meanings in different contexts. They also overlook the semantic relations between reviews of the same user or item. As a result, these methods struggle to learn accurate semantic representations for users and items. Moreover, these methods commonly use pooling operations to compress semantic feature sequences into feature vectors, inevitably leading to semantic loss. In addition, these methods merge interaction and review features in a straightforward manner, either through concatenation or nonlinear mappings, overlooking spatial dimension consistency in representation and lacking deep interaction across feature dimensions.

To address the mentioned limitations, we propose a recommendation model that incorporates a multi-scale context-aware network and a cross-attention mechanism (MSCA). First, we design a module comprising multiple TextCNNs (Zheng *et al.*, 2017; Kim, 2014) with varying window sizes to extract features from the review word embedding sequences, ensuring that the semantics of each word are preserved in the context of various local scales. Subsequently, we use a self-attention encoder (Vaswani *et al.*, 2017) to encode the semantic feature sequences of each user and item, aiming to capture the global contextual relationships between reviews, which enables a more accurate understanding of user preferences and item characteristics from historical reviews. Although texts and rating interactions originate from distinct modalities, their high-dimensional representations are related. To more effectively fuse the information from the two modalities, we use a cross-attention mechanism (Lu *et al.*, 2016; Chen *et al.*, 2021; Kim *et al.*, 2022). The rating interaction feature vector serves as the query, whereas the semantic feature sequence acts as both the key and value. When using the user (item) as the query, item (user) reviews are used as the key and value. This not only integrates modal information but also models the user-item interaction. As the query is a single token, not a sequence, the cross-attention mechanism directly generates the semantic vector, thus bypassing the need for pooling. Below is a summary of the main contributions of this paper:

- We propose MSCA, a novel recommendation model that uses a multi-scale convolutional network and a self-attention encoding module to extract semantic information across different contextual scales.
- We address deep interactions between review and interaction features using a cross-attention module. The user (or item) interaction features are fed as queries and the item (or user) review features are fed as keys and values into the attention network.

This simulates personalized attention to semantic features from various user (or item) perspectives, while deeply integrating both sets of features.

- Extensive experimental validation was performed on five Amazon datasets, comparing the MSCA model with several baseline methods. The MSCA model consistently outperformed across all datasets. Furthermore, we analyzed the effectiveness of the model's modules and discussed the critical parameters involved.

The structure of the subsequent parts of this paper is organized as outlined below: Section 2 discusses related work on review-based recommendation systems, encompassing methodologies leveraging topic modeling and deep learning techniques. Section 3 details MSCA's architecture. Subsequently, in Section 4, the experimental design employed to demonstrate the proposed model's effectiveness is described. Section 5 concludes the paper.

2. Related work

This paper's related works predominantly concentrate on user review-based recommendation methods, encompassing two categories: methods based on topic models and methods based on deep learning. In subsequent subsections, we will detail the research progress, strengths, and limitations of existing methods.

2.1 Topic models for review-based recommendation

To mitigate challenges like data sparsity inherent in traditional collaborative filtering recommendation algorithms reliant solely on rating information, recent studies have begun to incorporate auxiliary information encompassing user preferences and the characteristics of items, notably including user reviews. Initial methods predominantly employ topic modeling techniques for feature modeling of reviews. HFT (McAuley and Leskovec, 2013) gathers reviews from a user or an item and compiles them into the user's or the item's review document. The LDA topic model was used to conduct topic modeling for both user and item documents. By integrating with a latent-factor model, it consolidates both rating and review features for recommendations. Expanding on HFT, RMR (Ling *et al.*, 2014) uses a mixture of Gaussians to model rating data, positing that the mixture proportions align with the review topic distribution. This approach enables the model to deliver accurate recommendations even when only a few ratings are available. TopicMF (Bao *et al.*, 2014) uses non-negative matrix factorization (MF) to identify topics from individual reviews. Concurrently, it leverages MF to derive latent features from ratings, using a transformation function to bridge the latent feature spaces between reviews and ratings.

The methods described above construct representations of user preferences and item characteristics by learning latent topics related to users or items from reviews, yielding significant results. Nevertheless, topic models fail to account for the sequence of words and the contextual relationships, resulting in a loss of semantic information and a challenge in capturing the complex emotions and nuanced expressions present in reviews.

2.2 Deep learning methods for review-based recommendation

In recent years, with the continual advancement and widespread deployment of deep learning technologies across various domains, such as knowledge graph (Wang *et al.*, 2023b; Li *et al.*, 2024c, 2024b), computer vision, natural language processing, deep learning has also been increasingly used in recommendation systems (Liu *et al.*, 2024), especially for feature extraction from user reviews.

Convolutional neural networks (CNN) and attention mechanisms have been widely used in existing studies. DeepCoNN (Zheng *et al.*, 2017) uses two parallel TextCNNs to separately extract semantic features from reviews about users and items, respectively. Subsequently, TransNets (Catherine and Cohen, 2017) extends DeepCoNN by adding a target network to learn the embedding of target reviews. It minimizes the difference between the embedding of the target review and the semantic features learned by the source network. D-Attn (Seo *et al.*, 2017) incorporates a dual local and global attention mechanism before TextCNNs, individually learning representations of review word sequences under each attention type. These representations are then unified through a fully connected layer to facilitate rating prediction. Acknowledging the diverse contributions of each review, NARRE (Chen *et al.*, 2018) extends DeepCoNN by incorporating an attention pooling layer that allocates distinct weights to each review and condenses these review features into user and item representation vectors. Alternatively, ANR (Chin *et al.*, 2018) acknowledges that words exhibit varied semantic nuances across different aspects. It maps word embeddings from review texts into multiple semantic spaces and calculates attention for words within each space, thereby gleaning a variety of aspect-specific semantic features. DAML (Liu *et al.*, 2019) also uses two distinct attention mechanisms for learning review representations. Differing from D-Attn, it begins with a local attention layer for initial learning and then employs a mutual attention layer to link user and item review features. AHAG (Liu *et al.*, 2022a) uses a gated network for the dynamic fusion of review features and uses hierarchical attention mechanisms to model the long-term dependencies among review sequences.

Recently, graph neural networks (GNN) (Li *et al.*, 2023) have also been introduced to review-based recommendation tasks. RMG (Wu *et al.*, 2019) and RGCL (Shuai *et al.*, 2022) use GNN to learn feature representations of users and items. Specifically, RGCL introduces contrastive learning techniques, creating additional self-supervised signals from limited interaction behaviors. Taking into account the sequential information of words within reviews, RGNN (Liu *et al.*, 2023c) creates a review graph based on the interrelated order of word co-occurrences from user reviews. It uses a type-aware graph attention network to learn representations of the graph and produces subgraphs via personalized graph pooling. Subsequently, it combines these graph representations with user/item embeddings to construct comprehensive user/item features aimed at predicting ratings.

These review-based recommendation methods comprehensively model user preferences and characteristics of items. Nevertheless, they neglect the specific meanings of words in various contexts when learning semantic features. Furthermore, these models simply combine interaction features with semantic features, failing to deeply integrate both types of features for more effective recommendations.

3. Our approach

This study introduces the MSCA model, which leverages user reviews as side information to extract semantic features, enhancing the modeling of user preferences and item characteristics and ultimately improving rating prediction accuracy. The model comprises four components:

(1) embedding layer;

(2) multi-scale context-aware layer;

(3) cross-attention feature fusion layer; and

(4) rating prediction layer.

This section provides a detailed explanation of the model's framework, specifying the inputs, outputs, relevant notations and illustrating the details of each module. The structure of the MSCA model is illustrated in Figure 2.

3.1 Problem formulation

Task: Rating prediction. The objective is to construct representations for users and items from historical interaction data, including review texts, and to accurately predict ratings for unseen user-item pairs.

Each historical interaction is represented as a four-tuple $i_{u,v} = (u, v, r_{u,v}, s_{u,v}) \in \mathcal{I}$. u denotes the ID of user, v. indicates the ID of item, $r_{u,v}$ refers to the rating that user u assigns to item v, typically an integer between 1 and 5, $s_{u,v}$ represents the related review text and $\mathcal{I}$ encompasses the collection of all historical interactions. $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ is used to denote the set of users, with M indicating the total number of users, $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ for the set of items and N representing the total number of items. We compile the historical reviews associated with user u into a review set, denoted as $\mathcal{D}_u$. Likewise, we get the review set of item v, denoted as $\mathcal{D}_v$. The model takes $(u, v, \Delta\mathcal{D}_u, \mathcal{D}_v)$ as its input and produces $\hat{r}_{u,v}$ as its output, representing the predicted rating of user u for item v. Table 1 presents the notations defined in our method and their descriptions.

3.2 Embedding layer

Employing two embedding layers, we convert interaction and review text information into feature vectors and matrix forms, serving as inputs for the subsequent neural network layers.

3.2.1 Interaction feature embedding. Two trainable embedding matrices are used to map each user and item into an embedding vector. In this setup, $\mathbf{E}_{user} \in \mathbb{R}^{M \times d_i}$ denotes the user embedding matrix and $\mathbf{E}_{item} \in \mathbb{R}^{N \times d_i}$ is the item embedding matrix, with d_i indicating the dimensionality of the embedding layer. By inputting the current user and item IDs, u and v,

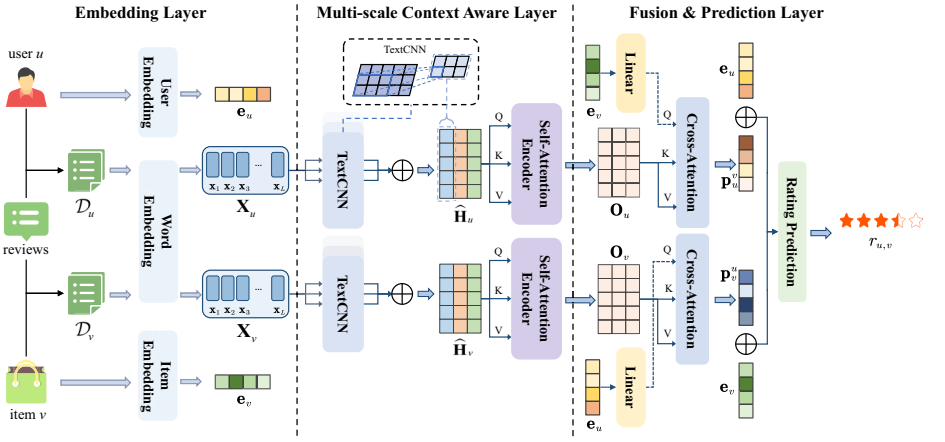


Figure 2. Architecture of MSCA
Source(s): Authors' own work

Table 1. Notations and description

Notations	Description
u, v	User u, item v
$i_{u,v},\ r_{u,v},\ s_{u,v},\ \widehat{r}_{u,v}$	Interaction between user u and item v, user u's rating for item v, user u's review on item v, predicted rating of item v by user u
$\mathcal{U}, \mathcal{V}, \mathcal{I}$	User set, item set, interaction set
M, N	Number of users, number of items
$\mathcal{D}_u, \mathcal{D}_v$	Set of reviews authored by user u, set of reviews targeting item v
$\mathbf{E}_{user}, \mathbf{E}_{item}$	User embedding matrix, item embedding matrix
d_1, d_2	Word embedding dimensionality, semantic feature sequence dimensionality
d_i, d_s	Interaction feature vector dimensionality, semantic feature vector dimensionality
$\mathbf{e}_u, \mathbf{e}_v$	User u's interaction feature vector, item v's interaction feature vector
L	Maximum length of review documents
C, K	Number of TextCNN modules, number of convolutional kernels in each TextCNN module
$\mathbf{X}_u, \mathbf{X}_v$	Review embedding matrix of user u, review embedding matrix of item v
$\mathbf{PE}$	Positional encoding
$\widehat{\mathbf{H}}_u,\ \widehat{\mathbf{H}}_v$	Review feature sequence of user u, review feature sequence of item v
$\mathbf{O}_u, \mathbf{O}_v$	Encoded semantic sequence of user u, encoded semantic sequence of item v
$\mathbf{q}_u, \mathbf{q}_v$	Query for user u in the cross-attention layer, query for item v in the cross-attention layer
$\mathbf{p}_v^u,\ \mathbf{p}_u^v$	Semantic feature vector of item v that incorporates the personalized preferences of user u, semantic feature vector of user u by taking into account the characteristics of item v
$\mathcal{L}$	Loss function

Source(s): Authors' own work

we can extract the respective user interaction feature embedding $\mathbf{e}_u \in \mathbb{R}^{1\times d_i}$ and item interaction feature embedding $\mathbf{e}_v \in \mathbb{R}^{1\times d_i}$. The process formula is as follows:

$$\mathbf{e}_u = \mathbf{E}_{user}(u), \tag{1}$$

$$\mathbf{e}_v = \mathbf{E}_{item}(v). \tag{2}$$

3.2.2 Semantic feature embedding. The word2vec model pretrained on Google News is used to map words from reviews into a low-dimensional real-valued feature space. It is extensively used across various natural language processing tasks. Given the varying lengths of review documents, a maximum review document length, denoted as the number of words L, is established to facilitate model training. When the length of collected user or item reviews falls short of L, zero-padding is performed at the end of the review document. Using user u as an example, $\mathcal{D}_u$ is inputted into the word embedding layer, resulting in the review embedding matrix $\mathbf{X}_u \in \mathbb{R}^{d_1\times L}$. As illustrated, $\mathbf{x}_n$ signifies the vector representation of the n-th word in the review document. Similarly, the review embedding matrix $\mathbf{X}_v \in \mathbb{R}^{d_1\times L}$ for an item can also be derived:

$$\mathbf{X}_u = \begin{bmatrix} \cdots & \mathbf{x}_{n-1} & \mathbf{x}_n & \mathbf{x}_{n+1} & \cdots \end{bmatrix} \in \mathbb{R}^{d_1\times L}. \tag{3}$$

3.3 Multi-scale context-aware layer

3.3.1 Multi-scale text convolutional networks. We use convolutional layers of different scales to learn semantic representations of word combinations and phrases in various contexts. This module, called multi-scale TextCNNs (MTC), consists of C parallel TextCNN modules, each using convolution kernels of different window sizes. It is denoted as $MTC = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_C\}$, where $\mathcal{F}_c = \{\mathbf{W}_k^c | k \in [1, K]\}$. $\mathbf{W}_k^c \in \mathbb{R}^{d_1 \times t_c}$ denotes the k-th convolution kernel, where t_c denotes the number of words in the convolution window, with $t_a \neq t_b$ when $a \neq b$. $\mathcal{F}_c$ signifies the collection of convolution kernels in the c-th TextCNN module.

Convolutions are applied across each $\mathcal{F}$, producing output matrices. These outputs are concatenated into a new feature sequence, which becomes the output of the MTC layer. The input matrix, either $\mathbf{X}_u$ or $\mathbf{X}_v$, is extended by adding $(t_c - 1)$ zero vectors at the end to ensure consistency in output sizes. The calculation process is as illustrated below:

$$h_i^k = \phi\left(\mathbf{W}_k^c * \mathbf{X}_{(:, i:(i+t_c-1))} + b_k\right), \tag{4}$$

$$\mathbf{h}^k = \left[h_1^k, h_2^k, \cdots, h_i^k, \cdots, h_L^k\right]^\top, \tag{5}$$

$$\mathbf{H}_c = \left[\mathbf{h}^1, \mathbf{h}^2, \cdots, \mathbf{h}^K\right], \tag{6}$$

the symbol * denotes the convolution operation, which involves the element-wise multiplication of two matrices, followed by their summation. b_k signifies the bias value and ϕ denotes the ReLU. $\mathbf{h}^k$ signifies the text feature extracted by the k-th convolution kernel. $\mathbf{H}_c$ denotes the semantic features of reviews extracted following the processing of input $\mathbf{X}$ by $\mathcal{F}_c$. Subsequently, the outputs from all TextCNN modules are concatenated to form $\widehat{\mathbf{H}}$, as depicted in the formula below:

$$\widehat{\mathbf{H}} = Concat(\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_C) = MTC(\mathbf{X}) \in \mathbb{R}^{L \times d_2}, \tag{7}$$

the *Concat* signifies the concatenation operation of matrices, where $d_2 = (C \cdot K)$.

As the review text matrices $\mathbf{X}_u$ and $\mathbf{X}_v$ for users and items come from the same semantic embedding space, a shared MTC module is used for both. Consequently, this yields the respective outputs $\widehat{\mathbf{H}}_u$ and $\widehat{\mathbf{H}}_v$.

3.3.2 Self-attention encoder. The self-attention mechanism is particularly effective for processing long text sequences, as it captures global dependencies, enhancing the model's ability to understand and represent review text. Leveraging this, two parallel self-attention modules are used to encode the user and item review feature sequences, with $\widehat{\mathbf{H}}_u$ for user u and the same process applied to item v. For simplicity, we use user u as an example in the following steps.

Positional encoding is added to preserve token positions, created using sine and cosine functions, resulting in the sequence $\mathbf{S}_u$:

$$\mathbf{PE}_{(pos, 2i)} = sin\left(pos / 10000^{2i/d_2}\right), \tag{8}$$

$$\mathbf{PE}_{(pos, 2i+1)} = cos\left(pos / 10000^{2i/d_2}\right), \tag{9}$$

$$\mathbf{S}_u = \widehat{\mathbf{H}}_u + \mathbf{PE}. \tag{10}$$

Subsequently, $\mathbf{S}_u$ is input into a multi-head self-attention layer and the specific calculation process is as follows:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}, \tag{11}$$

$$MultiHead(\mathbf{S}_u) = Concat(head_1, \ldots, head_H)\mathbf{W}^O, \tag{12}$$

$$head_j = Attention\left(\mathbf{S}_u\mathbf{W}_j^Q, \mathbf{S}_u\mathbf{W}_j^K, \mathbf{S}_u\mathbf{W}_j^V\right), \tag{13}$$

where $\mathbf{W}_j^Q \in \mathbb{R}^{d_2 \times d_k}$, $\mathbf{W}_j^K \in \mathbb{R}^{d_2 \times d_k}$, and $\mathbf{W}_j^V \in \mathbb{R}^{d_2 \times d_v}$ are the weight matrices of the j-th attention head, transforming $\mathbf{S}_u$ into query, key, and value, with $d_k = d_v = d_2/H$ and H being the number of heads. The concatenated attention heads are linearly transformed by $\mathbf{W}^O \in \mathbb{R}^{d_2 \times d_2}$ to produce the output, followed by a residual connection and layer normalization. The output then passes through a feedforward neural network, followed by another round of residual and layer normalization, producing $\mathbf{O}_u$:

$$\widehat{\mathbf{S}}_u = LayerNorm(MultiHead(\mathbf{S}_u) + \mathbf{S}_u), \tag{14}$$

$$\mathbf{O}_u = LayerNorm\left(\phi\left(\widehat{\mathbf{S}}_u\mathbf{W}_1 + \mathbf{b}_1\right)\mathbf{W}_2 + \mathbf{b}_2 + \widehat{\mathbf{S}}_u\right), \tag{15}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_2 \times d_f}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_f \times d_2}$ represent the learnable weights within the feedforward network, $\mathbf{b}_1 \in \mathbb{R}^{1 \times d_f}$ and $\mathbf{b}_2 \in \mathbb{R}^{1 \times d_2}$ signify the bias.

The user feature $\mathbf{O}_u$ is obtained from $\mathbf{X}_u$, capturing context semantics, and the corresponding item feature $\mathbf{O}_v$ is obtained from $\mathbf{X}_v$ through parallel networks.

3.4 Cross-attention feature fusion layer

The cross-attention mechanism is used for the deep integration of interaction features and semantic features. The user's interaction feature embedding $\mathbf{e}_u$ is used as the source of the query. The key and value are derived from the item's semantic feature sequence $\mathbf{O}_v$. The calculation process is as follows:

$$\mathbf{q}_u = \mathbf{e}_u\mathbf{W}_p\mathbf{W}_u^Q, \mathbf{K}_v = \mathbf{O}_v\mathbf{W}_v^K, \mathbf{V}_v = \mathbf{O}_v\mathbf{W}_v^V, \tag{16}$$

$$\mathbf{a}_v^u = softmax\left(\frac{\mathbf{q}_u\mathbf{K}_v^{\top}}{\sqrt{d_2}}\right), \tag{17}$$

$$\mathbf{o}_v^u = \sum_{i=1}^{L} \mathbf{a}_{v[i]}^u \mathbf{V}_{v[i,:]}, \tag{18}$$

where $\mathbf{W}_u^Q \in \mathbb{R}^{d_2 \times d_2}$, $\mathbf{W}_v^K \in \mathbb{R}^{d_2 \times d_2}$ and $\mathbf{W}_v^V \in \mathbb{R}^{d_2 \times d_2}$ represent the weight matrices. $\mathbf{W}_p \in \mathbb{R}^{d_i \times d_2}$ is a linear transformation matrix used for mapping dimensions. $\mathbf{a}_v^u$ represents

the attention weight that user u assigns to different tokens in $\mathbf{O}_v$. And $\mathbf{o}_v^u \in \mathbb{R}^{1\times d_2}$ represents the key features of item v from the perspective of user u.

We subsequently apply layer normalization to $\mathbf{o}_v^u$. Then, we input it into a feedforward network with residual connections and perform layer normalization once again. Finally, we project it into a fixed d_s-dimensional real-valued space using a linear layer as shown below:

$$\widehat{\mathbf{o}}_v^u = LayerNorm\left(\left(LayerNorm\left(\mathbf{o}_v^u\right)\mathbf{W}'_1 + \mathbf{b}'_1\right)\mathbf{W}'_2 + \mathbf{b}'_2\right), \tag{19}$$

$$\mathbf{p}_v^u = \widehat{\mathbf{o}}_v^u\mathbf{W}_l + b_l, \tag{20}$$

where $\mathbf{W}'_1 \in \mathbb{R}^{d_2\times d_f}$ and $\mathbf{W}'_2 \in \mathbb{R}^{d_f\times d_2}$ represent the learnable weights within the feedforward network, $\mathbf{b}'_1 \in \mathbb{R}^{1\times d_f}$ and $\mathbf{b}'_2 \in \mathbb{R}^{1\times d_2}$ signify the bias. $\mathbf{W}_l \in \mathbb{R}^{d_2\times d_s}$ and b_l are the parameters of linear layer. $\mathbf{p}_v^u \in \mathbb{R}^{1\times d_s}$ represents the semantic feature vector of item v that incorporates the personalized preferences of user u.

Similarly, we reverse the roles of user and item, using a symmetric network structure. We obtain the semantic feature vector $\mathbf{p}_u^v$ for user u by taking into account the characteristics of item v.

3.5 Rating prediction layer

A fully connected layer is used to integrate various features derived from previous modules. This includes user and item interaction feature embeddings $\mathbf{e}_u$ and $\mathbf{e}_v$, as well as the review feature vectors $\mathbf{p}_u^v$ and $\mathbf{p}_v^u$ for user u and item v. They are concatenated into $\mathbf{y}_{u,v} \in \mathbb{R}^{1\times d_y}$ and then inputted into the linear network, where $d_y = 2(d_i + d_s)$, with the process described as follows:

$$\mathbf{y}_{u,v} = Concat\left(\mathbf{p}_u^v, \mathbf{e}_u, \mathbf{p}_v^u, \mathbf{e}_v\right), \tag{21}$$

$$\widehat{r}_{u,v} = \phi\left(\mathbf{y}_{u,v}\mathbf{W}_p + b_p\right) + b_u + b_v, \tag{22}$$

where $\mathbf{W}_p \in \mathbb{R}^{d_y\times 1}$ and b_p represent the weight matrix and bias of the linear layer, b_u and b_v respectively represent the user bias and item bias and $r_{u,v}$ denotes the predicted rating of user u for item v.

3.6 Model training and loss function

This paper focuses on the rating prediction task in recommendations. The MSCA model is defined as $\widehat{r}_{u,v} = f_{MSCA}(u, v, \mathcal{D}_u, \mathcal{D}_v)$, with model training represented as $\mathcal{R}_{train} = f_{MSCA}(\mathcal{U}, \mathcal{V}, \mathcal{D}^{user}, \mathcal{D}^{item})$, where $\mathcal{R}_{train}$ represents the actual ratings from training data set. As rating prediction is a regression task, mean squared error (MSE) is used as the objective function, as shown below:

$$\mathcal{L} = \sum_{r_{u,v}\in\mathcal{R}_{train}} \left(r_{u,v} - \widehat{r}_{u,v}\right)^2 + \lambda\|\mathbf{\Theta}\|_2^2. \tag{23}$$

$r_{u,v}$ denotes the actual rating given by user u to item v. $\mathbf{\Theta}$ represents all the learnable parameters within the model. $\lambda\|\mathbf{\Theta}\|_2^2$ is the regularization term used to constrain the parameters in the model, serving to prevent overfitting in the model, where λ is the regularization coefficient.

Finally, the procedure of the MSCA model is delineated in Algorithm 1:

Algorithm 1: MSCA Procedure

Input: user set $\mathcal{U}$; item set $\mathcal{V}$; user review documents $\mathcal{D}^{user}$; item review documents $\mathcal{D}^{item}$; pertrained word2vec model and interaction records $\mathcal{I}$.
Output: predicted rating: $\hat{r}$.

Divide $\mathcal{I}$ into multiple batches according to the batch size;
for *each* $\mathcal{I}_{batch}$ *in* $\mathcal{I}$ **do**
  // The for-loop below actually conducts parallel computations for the entire batch.
  for *each* $i_{u,v}$ *in* $\mathcal{I}_{batch}$ **do**
    Fetch the current user and item and the corresponding review documents: $u, v, \mathcal{D}_u, \mathcal{D}_v$;
    Obtain the interaction features of user u and item v: $\mathbf{e}_u = \mathbf{E}_{user}(u), \mathbf{e}_v = \mathbf{E}_{item}(v)$, formulated in Eqs. (1) - (2);
    Obtain review embeddings through word2vec: $\mathbf{X}_u = \text{word2vec}(\mathcal{D}_u)$, $\mathbf{X}_v = \text{word2vec}(\mathcal{D}_v)$, formulated in Eq. (3);
    Utilize Multi-scale TextCNNs to derive semantic features: $\hat{\mathbf{H}}_u = MTC(\mathbf{X}_u), \hat{\mathbf{H}}_v = MTC(\mathbf{X}_v)$ according to Eqs. (4) - (7);
    Extract context features with Self-Attention Encoders: $\mathbf{O}_u = Encoder_{user}(\hat{\mathbf{H}}_u), \mathbf{O}_v = Encoder_{item}(\hat{\mathbf{H}}_v)$ according to Eqs. (8) - (15);
    Integrate interaction and review features using the Cross-Attention mechanism: $\mathbf{p}_u^v = Cross_{user}(\mathbf{e}_v, \mathbf{O}_u), \mathbf{p}_v^u = Cross_{item}(\mathbf{e}_u, \mathbf{O}_v)$, formulated in Eqs. (16) - (20);
    Concat all features: $\mathbf{y}_{u,v} = Concat(\mathbf{p}_u^v, \mathbf{e}_u, \mathbf{p}_v^u, \mathbf{e}_v)$ in Eq. (21);
    Predict rating: $\hat{r}_{u,v} = \phi(\mathbf{y}_{u,v}\mathbf{W}_p + b_p) + b_u + b_v$ in Eq. (22);
  end
  Objective function $\mathcal{L} = \sum_{i_{u,v} \in \mathcal{I}_{batch}} (r_{u,v} - \hat{r}_{u,v})^2 + \lambda \parallel \mathbf{\Theta} \parallel_2^2$ is according to Eq. (23);
  Take a gradient step to minimize $\mathcal{L}$;
end

4. Experiment

We carried out comprehensive experiments on five Amazon data sets from various product categories to assess the performance of MSCA and to analyze its modules. This section provides a detailed description of the experimental data sets, evaluation metrics, baseline methods, experimental setup, experimental results and analysis. To evaluate the MSCA model, we designed experiments to answer the following questions:

RQ1. How much does MSCA's performance vary compared to baseline methods on five Amazon data sets?

RQ2. What role do different modules play in influencing the performance of MSCA?

RQ3. How are MSCA's outcomes influenced by changes in hyperparameter settings?

RQ4. How do different network structures of the prediction layer affect the performance of MSCA?

4.1 Data set

We conducted experimental validation of the proposed model and baseline models on Amazon data sets (McAuley *et al*., 2015; He and McAuley, 2016), which are widely applied in the research of recommendation systems. We selected five data sets from different domains for our experiments, which are Beauty, Cell_Phones_and_Accessories (Phones), Clothing_Shoes_and_Jewelry (Clothing), Toys_and_Games (Toys) and Video_Games (Video). The experiments were conducted on five-core data sets where each user and item had a minimum of five interactions. Basic statistics of the data sets are outlined in Table 2. The sparsity of each data set is defined as the ratio of the actual number of interactions to the total possible interactions, which is determined by the product of the number of users and the number of items. From each data set, 80% was randomly chosen as the training set, with the remaining 20% divided equally into a 10% validation set and a 10% test set.

4.2 Evaluation metrics

For assessing the model's effectiveness, our experiments use MSE and mean absolute error (MAE) as evaluative metrics. They are widely used for evaluating the performance of value prediction models in regression tasks. Lower values of MSE and MAE signify better predictive capability of the model. In addition, MSE corresponds with the loss function $\mathcal{L}$. The definitions are as follows:

$$MSE = \frac{1}{|\mathcal{R}|} \sum_{r_{u,v} \in \mathcal{R}} \left(r_{u,v} - \widehat{r}_{u,v}\right)^2 \tag{24}$$

$$MAE = \frac{1}{|\mathcal{R}|} \sum_{r_{u,v} \in \mathcal{R}} |r_{u,v} - \widehat{r}_{u,v}| \tag{25}$$

4.3 Tested methods

To validate the performance of the MSCA model proposed in our paper, we compared it against the following methods:

- PMF (Mnih and Salakhutdinov, 2007): a method based on probability matrix factorization that learns latent vector representations of users/items by maximizing the posterior probability of observed ratings.

Table 2. Statistics of the selected datasets

Data set	# of users	# of items	# of Ratings and reviews	Sparsity (%)
Beauty	22,363	12,101	198,502	99.927
Phones	27,879	10,429	194,439	99.933
Clothing	39,387	23,033	278,677	99.969
Toys	19,412	11,924	167,597	99.928
Video	24,303	10,672	231,780	99.911

Source(s): Authors' own work

- SVD (Koren, 2008): decomposes the rating matrix into three matrices (user features, singular values and item features), then uses user and item features for rating prediction.
- SVD++ (Koren, 2008): extends SVD with neighborhood models and incorporates explicit interaction feedback to improve prediction accuracy.
- DeepCoNN (Zheng *et al.*, 2017): uses two TextCNNs to learn features of users (items) from respective review documents and then inputs these features into a factorization machine (FM) to predict ratings.
- NARRE (Chen *et al.*, 2018): uses two TextCNNs and an attention-based review pooling layer for feature extraction from reviews. These features are then fed into a latent factor model (LFM) to predict ratings.
- DAML (Liu *et al.*, 2019): uses a dual attention strategy that integrates both local and mutual attention to extract features from reviews. The semantic features from reviews are then fused with interaction features, and ultimately, the combined features are inputted into a neural factorization machine (NFM) for rating prediction.
- AHAG (Liu *et al.*, 2022a): uses a gated network for the dynamic fusion of review features and uses hierarchical attention mechanisms to model the long-term dependencies among review sequences, achieving dynamic interaction of features for rating prediction.
- RGNN (Liu *et al.*, 2023c): builds graphs from the words of user reviews based on words' co-occurrence relations. Using a type-aware graph attention network and a personalized graph pooling operation, it learns representations for both user and item features. Subsequently, these representations are used in an FM to predict ratings.
- MSCA: the model proposed in this paper. MSCA uses a multi-scale convolution network to learn the semantic features from review texts, encoding the features by a multi-head self-attention encoder. In addition, it uses a cross-attention module to fuse semantic features and interaction features deeply, culminating in the prediction of ratings.

4.4 Experiment settings

For the baseline methods, we conducted experiments with the parameters cited in the respective papers. For unspecified parameters, we resorted to the default settings from the corresponding open-source codes. For models requiring word embeddings, we use either word2vec or GloVe pre-trained models as dictated by the specifications within the code of each test method. For models incorporating CNN modules, we used the same parameters as in DeepCoNN, with the convolutional layer having 100 output channels and a convolution window size of 3.

In the MSCA model proposed in this study, the multi-scale convolutional layer uses three convolutional modules with window sizes [3, 4, 5], respectively, each configured with 30 convolution kernels. We use coordinate descent to optimize the model's hyperparameters. The learning rate is tuned within [0.0001, 0.001, 0.01, 0.1], the dropout rate is explored within [0.2, 0.4, 0.6, 0.8] and the dimensionalities of the interaction and review feature vectors are searched from the range [8, 16, 32, 64, 128]. All modules of MSCA are implemented in Python3 and PyTorch.

4.5 Overall comparison (RQ1)

Table 3 illustrates the performances of the MSCA model and the baseline methods introduced in this study across five data sets. The best results on each data set are highlighted in italics, and the second-best results are underlined. Experimental results demonstrate that the MSCA model consistently achieves the best MSE across data sets, except for the Toys data set where it performs comparably to AHAG. For MAE, our model achieves the best results on the Phones and Clothing data sets and results close to the best-performing method on the Beauty and Video data sets. We computed the average improvement of MSCA across all data sets compared to baseline models. The results showed that MSCA's overall performance exceeded that of all baseline approaches. In terms of MSE, compared to the classic PMF algorithm, MSCA achieved a 14.6% improvement. When compared with SVD and SVD++, it saw an improvement of over 7%. Compared to baseline methods based on review-based recommendations, our model achieved a 2.29% to 5.56% improvement in MSE. In terms of MAE, our model achieved varying degrees of improvement compared to all methods except AHAG. These demonstrate the superiority of the proposed model.

Compared to MAE, our model achieves a more significant advantage in terms of MSE. This observation may be due to the use of squared error as the loss function, which makes MSE more aligned with the model's optimization goal. The greater sensitivity of MSE to larger errors, compared to MAE, suggests that our model is particularly effective at reducing large prediction errors.

In addition, the experimental results show that methods relying solely on rating interactions for recommendations through matrix factorization, such as PMF, SVD and SVD ++, exhibit higher MSE compared to methods that use review information. This suggests a notable enhancement in recommendation system accuracy upon integrating reviews as auxiliary information. We observe that SVD and SVD++ perform better in terms of MAE, which indicates that they are effective at reducing medium-sized errors but less effective at handling large errors. DeepCoNN ranks as the least effective among all the deep learning-based methods, which aligns with our expectations. Because it relies solely on CNN modules for feature extraction from reviews without acknowledging the different importance of different review sections or interaction features. In contrast, DAML, NARRE and AHAG use attention modules to identify key features within numerous reviews and filter out noise, integrating interaction features into their models, thereby achieving significant performance improvements over DeepCoNN and securing competitive outcomes. Among them, AHAG's combination of gated networks with hierarchical attention outperforms DAML's dual attention structure. NARRE assigns varied weights to each review based on an attention operation, mitigating noise from less critical reviews and achieving stable and excellent performance. Meanwhile, RGNN begins with individual words in reviews, considers their co-occurrence and uses a graph attention network to learn review features, thereby also securing competitive results.

4.6 Ablation study (RQ2)

To understand how modules in MSCA contribute to performance improvement, we designed three variants for ablation tests, which were validated on all experimental data sets. At this stage of the experiment, MSE is used as the evaluation metric. All variants use the same settings as the best parameters of the MSCA model to make a fair comparison. We carried out experiments to examine how the multi-scale context-aware layer and the cross-attention feature fusion layer affect model performance. The variants are described as follows:

- *w/o MC*: This variant eliminates both the multi-scale context-aware and the cross-attention feature fusion layers. It uses a pooling operation to obtain semantic feature

Table 3. Result comparison for all tested methods on five data sets

Data set	Beauty		Phones		Clothing		Toys		Video		Δ(%)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE (%)	MAE (%)
PMF	1.3825	0.8846	1.5789	0.9541	1.2939	0.8617	0.9108	0.6967	1.2049	0.8287	14.60	6.41
SVD	1.2694	0.8369	1.4058	0.8886	1.1783	0.8065	0.8518	0.6606	1.1308	0.7981	7.05	0.96
SVD++	1.2604	0.8335	1.4158	0.8896	1.1910	0.8077	0.8543	0.6588	1.1303	0.7989	7.29	0.89
DeepCoNN	1.2289	0.8593	1.3697	0.9219	1.1467	0.8248	0.8514	0.6935	1.1383	0.8225	5.56	4.16
DAML	1.2032	0.8532	1.3288	0.9082	1.1387	0.8368	0.8393	0.6824	1.1455	0.8304	4.29	3.88
NARRE	1.1931	0.8314	1.3258	0.8974	1.1132	0.8122	0.8203	0.6518	1.0923	0.8075	2.29	1.12
AHAG	1.1855	0.8234	1.3930	0.9048	1.1435	0.8103	*0.7981*	*0.6250*	1.1285	*0.7848*	3.71	−0.38
RGNN	1.2153	*0.8216*	1.3314	0.9034	1.1471	0.8510	0.8122	0.6512	1.0988	0.7982	3.23	1.67
MSCA	*1.1707*	0.8267	*1.2999*	*0.8864*	*1.0822*	*0.7971*	0.7982	0.6526	*1.0688*	0.7906	–	–

Note(s): The best results are highlighted in italics, and the second-best results are displayed with an underline. Δ% represents the average percentage of improvement by MSCA over the method across all data sets
Source(s): Authors' own work

vectors from the review sequences output by the embedding layer. Then, the vectors are aligned in dimension through linear mapping before being input into the prediction layer.

- *w/o M*: This variant removes the multi-scale context-aware layer, replacing it with a linear mapping operation for aligning the feature dimensionalities of text sequences, which are then fed into the cross-attention feature fusion layer.
- *w/o C*: This variant removes the cross-attention feature fusion layer, replacing it with an average pooling operation to compress text feature sequences into semantic feature vectors, which are then input to the prediction layer.

The experimental results are displayed in Figure 3. First, all three variants underperform compared to MSCA. Furthermore, except for *w/o C* performing worse than *w/o MC* on the Clothing data set, both *w/o M* and *w/o C* outperform *w/o MC* on the other data sets. This confirms that both the multi-scale context-aware layer and the cross-attention feature fusion layer independently contribute to improved prediction accuracy. Moreover, combining these two modules resulted in higher accuracy than either module alone, validating the effectiveness and compatibility of both components. The more noticeable performance decline in *w/o C* compared to *w/o M* indicates that the cross-attention feature fusion layer significantly boosts model performance. This demonstrates its capability to effectively fuse information from two modalities, and more accurately model user preferences and item characteristics. Next, we conduct more detailed experimental analyses of these two modules.

4.6.1 Analysis of the multi-scale context-aware layer. To investigate the contribution of the two components in the multi-scale context-aware layer to model performance, we designed corresponding ablation variants and conducted experiments. For the multi-scale text convolutional networks, as described in Section 4.4, we used convolutional networks with window sizes of 3, 4 and 5 in the experiments, each with 30 convolution kernels, and concatenated their outputs. In this section, we used single-scale convolutional networks with these three window sizes as variants, setting the number of convolution kernels to 90 to maintain a constant total dimension, to verify the effect of multi-scale convolution operation. In addition, we removed the self-attention encoder as an experimental variant to verify its role in encoding semantic features at a global context scale. We evaluated these variants using MSE as the metric. The variants are described as follows:

- *3-w*: We replace multi-scale text convolutional networks with a single-scale convolutional network that has a window size of *3* and 90 convolution kernels.

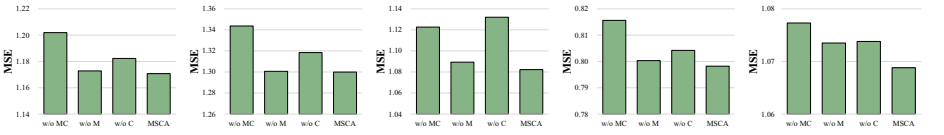


Figure 3. Performance comparison of w/o MC, w/o M, w/o C and MSCA on five data sets
Note(s): (a) Beauty, (b) Phones, (c) Clothing, (d) Toys, (e) Video
Source(s): Authors' own work

- *4-w*: We replace multi-scale text convolutional networks with a single-scale convolutional network, which has a window size of *4* and 90 convolution kernels.
- *5-w*: We replace multi-scale text convolutional networks with a single-scale convolutional network, which has a window size of *5* and 90 convolution kernels.
- *w/o self*: Removes the self-attention encoder, using the output of the multi-scale convolution directly as input to the cross-attention module.

The experimental results are shown in Figure 4. Overall, each variant exhibits varying degrees of performance degradation compared to MSCA across most data sets. Although *4-w* and *5-w* perform similarly to MSCA on the Beauty data set, the results across multiple data sets collectively demonstrate the effectiveness of both the multi-scale text convolution and the self-attention encoder components. We observed that the performance ranking of the first three variants varies across data sets, indicating that the optimal context window size for semantic feature extraction using CNNs is not consistent across different scenarios. This suggests that the multi-scale approach in MSCA, by leveraging convolutional layers of various window sizes and a self-attention mechanism, enhances the model's ability to capture diverse contextual information and adapt to different data sets effectively.

4.6.2 Analysis of the cross-attention module. The cross-attention feature fusion layer is composed of two symmetrical modules: the item semantic learning module, which uses user interaction feature embeddings as the query and item review semantic sequences as both key and value; and the user semantic learning module, which uses item interaction features as the query with user review semantic sequences as both key and value. To further explore the distinct impacts of these two symmetrical modules on model performance, as well as to delineate the differences between user and item perspectives, we devise two variants. Each variant removes one side of the cross-attention modules for experimental purposes, detailed as follows:

(1) *w/o CU*: This variant eliminates the user semantic learning module within the cross-attention feature fusion layer, using average pooling to condense the semantic sequence of user reviews into a user semantic vector.

(2) *w/o CI*: This variant removes the item semantic learning module and uses an average pooling operation to condense the semantic sequence of item reviews into an item semantic vector.

Using MSE as the evaluation metric, the experimental results are presented in Figure 5. Besides comparing with MSCA, we also conducted comparisons of the two variants against

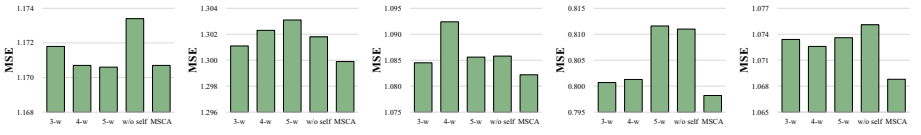


Figure 4. Performance comparison of 3-w, 4-w, 5-w, w/o self and MSCA on five data sets
Note(s): (a) Beauty, (b) Phones, (c) Clothing, (d) Toys, (e) Video
Source(s): Authors' own work

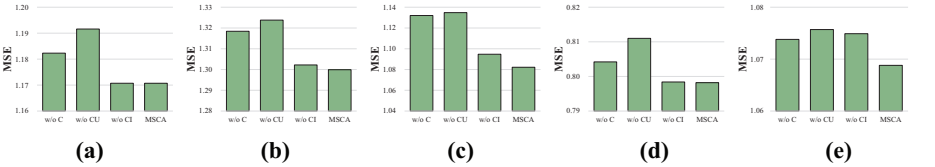


Figure 5. Performance comparison of w/o C, w/o CU, w/o CI and MSCA on five data sets
Note(s): (a) Beauty, (b) Phones, (c) Clothing, (d) Toys, (e) Video
Source(s): Authors' own work

w/o C. We noted that both variants exhibited varying levels of decreased accuracy compared to MSCA, demonstrating the positive impact of the two modules on model performance. Specifically, *w/o CU* experienced a more pronounced performance drop, which suggests that using item embeddings as attention queries facilitates the learning of critical user semantic features. Meanwhile, the performance drop of *w/o CI* is more pronounced in the Video and Clothing data sets than in others, likely due to the distinct preferences among users for products within the clothing and video game categories. For products within the other three data set sectors, the key semantic features of the items exhibit minimal variation across different users.

In addition, we discovered that the performance of *w/o CU* was significantly inferior to *w/o C* across multiple data sets. To determine whether this performance degradation is due to inherent differences between user and item entities or issues within the module itself, we conducted further experiments on the MSCA model and its two variants, *w/o CU* and *w/o CI*. Specifically, we replaced the cross-attention module with MLP and LSTM modules. The rationale for this replacement is that MLP is a fundamental and widely used structure in deep learning, whereas LSTM is a classic structure for processing sequential features. In this way, we obtained six variants, named *case-x* (where *x* ranges from 1 to 6). The detailed descriptions are as follows:

- *case-1*: Based on *w/o CU*, we replace the cross-attention module that processes item semantic sequences with an MLP module.
- *case-2*: Based on *w/o CU*, we replace the cross-attention module that processes item semantic sequences with an LSTM module.
- *case-3*: Based on *w/o CI*, we replace the cross-attention module that processes user semantic sequences with an MLP module.
- *case-4*: Based on *w/o CI*, we replace the cross-attention module that processes user semantic sequences with an LSTM module.
- *case-5*: Based on MSCA, we replace the cross-attention layer with two MLP modules, one for learning user features and the other for item features.
- *case-6*: Based on MSCA, we replace the cross-attention layer with two LSTM modules.

We conducted experiments on all data sets with these six variants, and the results are presented in Figure 6. The results indicate that for the Beauty, Toys and Video data sets, the performance of *case-1* and *case-2* is inferior to that of *w/o C*. In the other two data sets, *case-1* performs worse than *w/o C*, whereas the improvement of **c***ase-2* is negligible. In addition, it is observed that across all data sets, *case-3* outperforms *case-1* and *case-4* outperforms *case-2*. This is consistent with the previously mentioned finding that the performance of *w/o CU* is

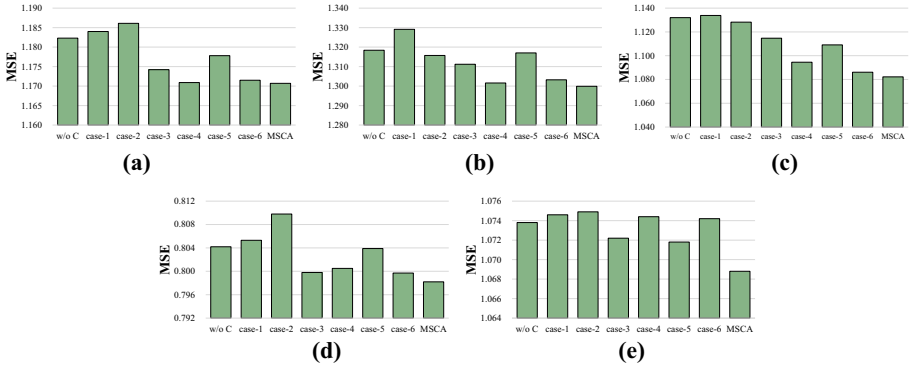


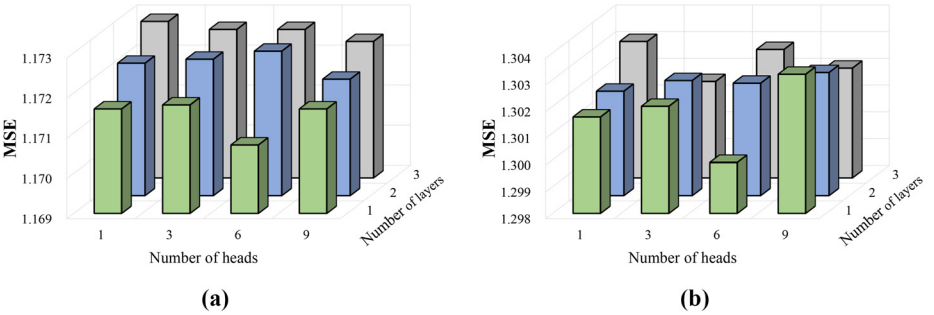
Figure 6. Performance comparison of w/o C, MSCA and six case variants across five data sets
Note(s): (a) Beauty, (b) Phones, (c) Clothing, (d) Toys, (e) Video
Source(s): Authors' own work

inferior to both *w/o C* and *w/o CI*. This suggests that the performance degradation in the *w/o CU* variant is not directly attributable to the cross-attention module, but rather to inherent differences in the semantic features of users and items. Based on the experimental results, we infer that in rating prediction tasks, user semantic features are likely the dominant influencing factor. This implies that without effectively extracting user semantic features, using the corresponding module for items fails to extract effective features. Consequently, the MSCA model, which integrates both modules, surpasses all variants, indicating that when user semantic features are effectively extracted, item semantic features serve as a valuable supplementary feature. The synergy between the two modules thus achieves optimal performance. Furthermore, the performance of *case-5* and *case-6* is inferior to that of MSCA, demonstrating that cross-attention is superior to these two classic network structures in overall effectiveness.

In conclusion, neither classic network structures nor cross-attention can mitigate the performance degradation observed in the *w/o CU* condition, indicating that this phenomenon is unrelated to the cross-attention module.

4.7 Parameter analysis of MSCA (RQ3)

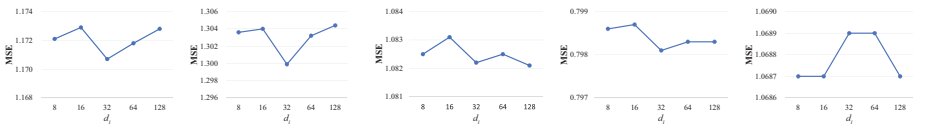
4.7.1 Self-attention encoder. We investigate the impact of the number of heads and layers in the self-attention encoder on two representative data sets, Beauty and Phones, which are frequently used in experimental validation of recommendation research. In the experiments, the number of attention heads is selected from [1, 3, 6, 9], and the number of layers is searched in [1, 2, 3]. Using MSE as the evaluation metric, the results shown in Figure 7 show an increase in MSE with more layers, with the optimal performance achieved at only one layer. This discovery implies that one layer already adeptly captures contextual features. A higher number of layers might increase the model's complexity, potentially leading to overfitting or reducing the model's generalization ability. The influence of the heads' numbers is relatively minor, and no uniform optimal value exists across different numbers of layers. With the number of layers at 1, the optimal results are obtained with six heads. Therefore, for subsequent experiments in this study, we establish the ideal configuration as one layer and six heads.



(a) **(b)**

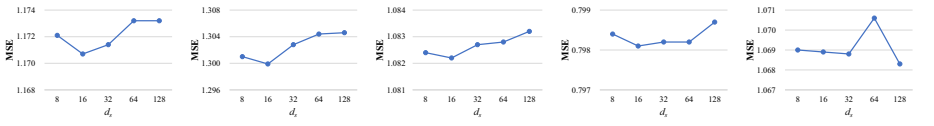
Figure 7. Impact analysis of the number of heads and layers within the self-attention encoder of MSCA
Note(s): (a) Beauty, (b) Phones
Source(s): Authors' own work

4.7.2 Dimensionalities of interaction and semantic vectors. We explore the optimal dimensionality for interaction d_i and semantic features d_s within the MSCA model across five data sets. The findings are illustrated in Figures 8 and 9, respectively. For the Beauty, Phones and Toys data sets, the lowest MSE is observed when d_i is set to 32. Although it is not optimal for the Clothing and Video data sets, the MSE is only marginally different from the best result. For example, in Video, the difference between 1.0689 (d_i = 32) and 1.0687 (d_i = 8, 16, 128) is less than 0.2‰. Across the four data sets, excluding Video, the lowest MSE is achieved when the semantic feature dimensionality, d_s, is set to 16. It is also evident that MSCA exhibits relative insensitivity to these two parameters. In light of these experimental findings, and to reduce model parameters for faster training while mitigating overfitting, we configure d_i at 32 and d_s at 16 as the optimal parameters for the MSCA model.



(a) **(b)** **(c)** **(d)** **(e)**

Figure 8. Dimensionality impact analysis on interaction feature vectors
Note(s): (a) Beauty, (b) Phones, (c) Clothing, (d) Toys, (e) Video
Source(s): Authors' own work



(a) **(b)** **(c)** **(d)** **(e)**

Figure 9. Dimensionality impact analysis on semantic feature vectors
Note(s): (a) Beauty, (b) Phones, (c) Clothing, (d) Toys, (e) Video
Source(s): Authors' own work

4.8 Influence of different prediction modules (RQ4)

The prediction layer is crucial in recommendation systems, as it needs to capture the relationships between user and item features to predict ratings effectively. In the MSCA model, the design of the prediction layer directly impacts the prediction accuracy. To evaluate the effectiveness of different structures, we conducted experiments on six common feature interaction modules across five data sets. Through these experiments, we analyzed the ability of each module to capture the relationships between user and item features and identified the most suitable prediction module for the MSCA model. Specifically, we take the users' and items' semantic feature vectors from the cross-attention layer, along with their interaction feature vectors from the embedding layer, as inputs for the prediction layer. The six structures are described as follows:

(1) Linear: concatenates all features and feeds them into a linear neural network with an output dimension of 1. The real-number output, adjusted by a bias, serves as the final predicted rating score.

(2) LFM: calculates the dot product of user and item latent vectors. The resulting dot product, combined with a bias term, provides the predicted rating score. It is commonly used in early recommendation methods based on matrix factorization.

(3) MLP: captures intrinsic relationships between features via fully connected hidden layers and nonlinear activation functions, outputting the final result through an output layer. It is a commonly used neural network structure.

(4) FM (Rendle, 2010): maps features into a low-dimensional space and forms second-order cross-terms to simulate feature interactions. The combination of bias, linear terms and second-order terms constitutes the final prediction.

(5) NFM (He and Chua, 2017): building upon FM, uses neural networks to learn complex nonlinear relations among features, thereby bolstering the modeling of feature interactions.

(6) AFM (Xiao *et al.*, 2017): An enhancement to FM, it integrates an attention mechanism, allocating distinct weights to different feature interaction terms to more effectively identify relationships among crucial features.

The experimental results are presented in Table 4, using MSE as the metric for evaluation. On every data set, LFM performed the poorest because its dot product computation does not factor in the interactions and weights among feature dimensions. Meanwhile, Linear demonstrates the lowest MSE. In addition, we observe that the more complex NFM and AFM underperform compared to the simpler FM. These findings indicate that for the MSCA model, enhancing the complexity and the number of parameters in the prediction layer does not improve performance; it may even lead to overfitting. This effect could be due to the preceding modules in MSCA already efficiently capturing interactions among features. Consequently, based on these results, we select the Linear as the prediction layer's structure of MSCA.

5. Conclusion

We propose MSCA, a rating prediction recommendation model, which mines user interests from historical reviews, jointly modeling these with interaction information to represent user preferences and item characteristics. To accomplish this objective, MSCA uses a multi-scale context-aware network to derive the semantic feature representations about users or items from reviews. A cross-attention mechanism is used, simulating differentiated preferences

Table 4. Performance comparison of various prediction modules on five datasets

	Beauty	Phones	Clothing	Toys	Video
Linear	*1.1707*	*1.2999*	*1.0822*	*0.7982*	*1.0688*
LFM	1.5163	2.0082	2.1198	1.6202	2.0420
MLP	1.3454	1.4523	1.3547	1.0207	1.2179
FM	1.1778	1.3085	1.0931	0.8042	1.0751
NFM	1.1881	1.3104	1.0992	0.8139	1.0813
AFM	1.1860	1.3127	1.0955	0.8115	1.0842

Note(s): The best results are highlighted in italics, and the second-best results are displayed with an underline
Source(s): Authors' own work

among users and integrating interaction and review features. Our extensive experiments demonstrate that MSCA outperforms all baseline methods. Compared to review-based recommendation methods, MSCA achieves improvements in MSE ranging from 2.29% to 5.56%. The ablation study shows the contribution of each module and parameter experiments identify the optimal hyperparameter configuration. These results highlight the model's effectiveness and potential for advancing recommendation systems.

However, there are some limitations to our approach. First, although we treat reviews from the same user or item as a whole for context learning, we overlook their temporal aspect. User preferences can change over time, but our model does not account for this change. This presents an important direction for future research, especially in dynamic user behavior. Second, although our model focuses on integrating interaction and semantic features, the learning module for interaction features remains relatively simple. Although this structure provides a solid baseline, adopting more advanced methods to model user-item interactions may further enhance the model's ability to capture the collaborative relationships between users and items.

For future research, we will explore integrating large language models (LLMs) into review-based recommendation systems. With the ongoing development and widespread adoption of LLMs, we aim to leverage their capabilities to process large volumes of reviews and generate higher-quality text. In addition, we will investigate replacing traditional embedding methods, such as word2vec, with embeddings derived from LLMs. Although the semantic information in reviews is rich, interaction information remains a crucial signal in recommendation systems. We will explore graph-based models and other approaches that better capture the collaborative relationships between users and items and integrate them with the semantic features from reviews.

References

Bao, Y., Fang, H. and Zhang, J. (2014), "TOPICMF: simultaneously exploiting ratings and reviews for recommendation", *Proceedings of the AAAI Conference on Artificial Intelligence*, p. 28.

Catherine, R. and Cohen, W. (2017), "TRANSNETS: learning to transform for recommendation", *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 288-296.

Chen, C.F.R., Fan, Q. and Panda, R. (2021), "CROSSVIT: cross-attention multi-scale vision transformer for image classification", *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 357-366.

Chen, C., Zhang, M., Liu, Y. and Ma, S. (2018), "Neural attentional rating regression with review-level explanations", *Proceedings of the 2018 World Wide Web Conference*, pp. 1583-1592.

Chin, J.Y., Zhao, K., Joty, S. and Cong, G. (2018), "ANR: aspect-based neural recommender", *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 147-156.

He, X. and Chua, T.S. (2017), "Neural factorization machines for sparse predictive analytics", *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 355-364.

He, R. and McAuley, J. (2016), "Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering", *Proceedings of the 25th International Conference on World Wide Web*, pp. 507-517.

Kim, Y. (2014), "Convolutional neural networks for sentence classification", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746-1751.

Kim, H.H., Yu, S., Yuan, S. and Tomasi, C. (2022), "Cross-attention transformer for video interpolation", *Proceedings of the Asian Conference on Computer Vision (ACCV) Workshops*, pp. 320-337.

Koren, Y. (2008), "Factorization meets the neighborhood: a multifaceted collaborative filtering model", *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 426-434.

Li, D., Shi, F., Wang, X., Zheng, C., Cai, Y. and Li, B. (2024b), "Multi-perspective knowledge graph completion with global and interaction features", *Information Sciences*, Vol. 666, p. 120438.

Li, Z., Zhang, Q., Zhu, F., Li, D., Zheng, C. and Zhang, Y. (2023), "Knowledge graph representation learning with simplifying hierarchical feature propagation", *Information Processing and Management*, Vol. 60 No. 4, p. 103348.

Li, D., Deng, C., Wang, X., Li, Z., Zheng, C., Wang, J. and Li, B. (2024a), "Joint inter-word and inter-sentence multi-relation modeling for summary-based recommender system", *Information Processing and Management*, Vol. 61 No. 3, p. 103631.

Li, D., Liu, H., Zhang, Z., Lin, K., Fang, S., Li, Z. and Xiong, N.N. (2021), "CARM: confidence-aware recommender model via review representation learning and historical rating behavior in the online platforms", *Neurocomputing*, Vol. 455, pp. 283-296.

Li, D., Xia, T., Wang, J., Shi, F., Zhang, Q., Li, B. and Xiong, Y. (2024c), "SDFORMER: a shallow-to-deep feature interaction for knowledge graph embedding", *Knowledge-Based Systems*, Vol. 284, p. 111253.

Liang, R., Zhang, Q., Wang, J. and Lu, J. (2024), "A hierarchical attention network for cross-domain group recommendation", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 35 No. 3, pp. 3859-3873.

Ling, G., Lyu, M.R. and King, I. (2014), "Ratings meet reviews, a combined approach to recommend", *Proceedings of the 8th ACM Conference on Recommender Systems*, pp. 105-112.

Liu, D., Li, J., Du, B., Chang, J. and Gao, R. (2019), "DAML: dual attention mutual learning between ratings and reviews for item recommendation", *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 344-352.

Liu, B., Li, D., Wang, J., Wang, Z., Li, B. and Zeng, C. (2024), "Integrating user -term intentions and long-term preferences in heterogeneous hypergraph networks for sequential recommendation", *Information Processing and Management*, Vol. 61 No. 3, p. 103680.

Liu, D., Wu, J., Li, J., Du, B., Chang, J. and Li, X. (2022a), "Adaptive hierarchical attention-enhanced gated network integrating reviews for item recommendation", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 34 No. 5, pp. 2076-2090.

Liu, Y., Yang, S., Zhang, Y., Miao, C., Nie, Z. and Zhang, J. (2023c), "Learning hierarchical review graph representations for recommendation", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, pp. 658-671.

Liu, X., Meng, S., Li, Q., Qi, L., Xu, X., Dou, W. and Zhang, X. (2023a), "SMEF: social-aware multi-dimensional edge features-based graph representation learning for recommendation", *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 1566-1575.

Liu, H., Zheng, C., Li, D., Zhang, Z., Lin, K., Shen, X., Xiong, N.N. and Wang, J. (2022c), "Multi-perspective social recommendation method with graph representation learning", *Neurocomputing*, Vol. 468, pp. 469-481.

Liu, X., Meng, S., Li, Q., Xu, X., Qi, L., Dou, W., Zhang, J. and Zhang, X. (2023b), "Disentangled hypergraph collaborative filtering for social recommendation", *2023 IEEE International Conference on Web Services (ICWS)*, pp. 475-482.

Liu, H., Zheng, C., Li, D., Shen, X., Lin, K., Wang, J., Zhang, Z., Zhang, Z. and Xiong, N.N. (2022b), "EDMF: efficient deep matrix factorization with review feature learning for industrial recommender system", *IEEE Transactions on Industrial Informatics*, Vol. 18 No. 7, pp. 4361-4371.

Lu, J., Yang, J., Batra, D. and Parikh, D. (2016), "Hierarchical question-image co-attention for visual question answering, in: Advances in neural information processing systems",

McAuley, J. and Leskovec, J. (2013), "Hidden factors and hidden topics: understanding rating dimensions with review text", *Proceedings of the 7th ACM Conference on Recommender Systems*, p. 165-172.

McAuley, J., Targett, C., Shi, Q. and van den Hengel, A. (2015), "Image-based recommendations on styles and substitutes", *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43-52.

Mnih, A. and Salakhutdinov, R.R. (2007), "Probabilistic matrix factorization", *in: Advances in Neural Information Processing Systems*,

Rendle, S. (2010), "Factorization machines", *2010 IEEE International Conference on Data Mining*, pp. 995-1000.

Seo, S., Huang, J., Yang, H. and Liu, Y. (2017), "Interpretable convolutional neural networks with dual local and global attention for review rating prediction", *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 297-305.

Shuai, J., Zhang, K., Wu, L., Sun, P., Hong, R., Wang, M. and Li, Y. (2022), "A review-aware graph contrastive learning framework for recommendation", *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1283-1293.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017), "Attention is all you need", *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000-6010.

Wang, H., Xu, Y., Yang, C., Shi, C., Li, X., Guo, N. and Liu, Z. (2023a), "Knowledge-adaptive contrastive learning for recommendation", *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pp. 535-543.

Wang, J., Zhang, Q., Shi, F., Li, D., Cai, Y., Wang, J., Li, B., Wang, X., Zhang, Z. and Zheng, C. (2023b), "Knowledge graph embedding model with attention-based high-low level features interaction convolutional network", *Information Processing and Management*, Vol. 60 No. 4, p. 103350.

Wu, B., Zhong, L., Yao, L. and Ye, Y. (2022), "EAGCN: an efficient adaptive graph convolutional network for item recommendation in social internet of things", *IEEE Internet of Things Journal*, Vol. 9 No. 17, pp. 16386-16401.

Wu, B., He, X., Wu, L., Zhang, X. and Ye, Y. (2023), "Graph-augmented co-attention model for socio-sequential recommendation", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 53 No. 7, pp. 4039-4051.

Wu, C., Wu, F., Qi, T., Ge, S., Huang, Y. and Xie, X. (2019), "Reviews meet graphs: enhancing user and item representations for recommendation with hierarchical attentive graph neural network", *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4884-4893.

Xiao, J., Ye, H., He, X., Zhang, H., Wu, F. and Chua, T.S. (2017), "Attentional factorization machines: learning the weight of feature interactions via attention networks", *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3119-3125.

Zheng, L., Noroozi, V. and Yu, P.S. (2017), "Joint deep modeling of users and items using reviews for recommendation", *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 425-434.

Corresponding author
Jian Wang can be contacted at: jianwang@whu.edu.cn