Information Processing and Management 61 (2024) 103631

Contents lists available at ScienceDirect

Information Processing and Management

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

Joint inter-word and inter-sentence multi-relation modeling for summary-based recommender system

Duantengchuan Li [a], Ceyu Deng [b,c,*], Xiaoguang Wang [b], Zhifei Li [d], Chao Zheng [a], Jing Wang [e], Bing Li [a,f,*]

[a] School of Computer Science, Wuhan University, Wuhan 430072, China
[b] School of Information Management, Wuhan University, Wuhan 430072, China
[c] Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China
[d] School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China
[e] School of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[f] Hubei Luojia Laboratory, Wuhan 430079, China

ARTICLE INFO

Keywords:
Summary-based recommender
Inter-word relation
Inter-sentence relation
Multi-relation modeling
Transformer

ABSTRACT

Review is an essential piece of information that influences users' decisions, but excessively long reviews not only degrade the user experience but also affect the accuracy of the recommender system. Therefore, Joint Inter-Word and Inter-Sentence Multi-Relation Modeling for the Summary-based Recommender System (MRSR) is proposed in this paper. In MRSR, the concise summary information serves as representation data, and a multi-relation modeling module is designed to construct user and item characteristics from two levels. Specifically, the inter-word relation modeling module, which consists of the Transformer and the pooling layer, is used to learn the long dependencies of summaries and extract word-level features by calculating the relative weights between words within sentences. The inter-sentence relation modeling module is used to enrich the sentence-level features of users and items, where an attention mechanism is employed to perceive the relative weights between different summary sentences. Finally, the fusion layer based on multi-head attention and the prediction layer based on attentional factorization machine are implemented to conduct the shallow and deep interactions between user and item features, based on which MRSR completes the final rating prediction. Extensive experimental results on five publicly available datasets demonstrate that MRSR achieves a 5.94% improvement in RMSE metrics compared to state-of-the-art methods. Furthermore, the accuracy of most existing models is improved by 1%~2% while the inference time is reduced by 10% by utilizing summaries as representation data. It proves the efficiency and effectiveness of our proposed approach, which has promising application prospects.

1. Introduction

As a crucial method of overcoming information overload and facilitating precise search, recommender systems (Wu et al., 2022) have been widely applied in various fields such as e-commerce (Liu, Li, et al., 2022; Wang et al., 2022; Wu, He, Wu, et al., 2023), social media (Zhang & Hara, 2023; Zhang & Yamasaki, 2021), travel and tourism (Zhu et al., 2022), leisure and entertainment (Huang, 2021), and industrial applications (Hu et al., 2020; Yin et al., 2023). The recommender system proposed

* Corresponding author.
E-mail addresses: dtclee1222@whu.edu.cn (D. Li), ceyudeng@gmail.com (C. Deng).

https://doi.org/10.1016/j.ipm.2023.103631
Received 6 August 2023; Received in revised form 24 November 2023; Accepted 26 December 2023
Available online 9 January 2024

in this paper utilizes the summary of users' comments (referred to as "summary") to construct user preferences and relevant item attributes. It provides personalized recommendation services in industrial applications through the correlation rating between the two features.

Over the past decade, scholars have proposed many classic algorithms, including collaborative filtering (Chen et al., 2023) and matrix factorization (Koren et al., 2009). Both methods construct a user–item matrix based on user ratings or behaviors toward certain items, representing the user's preference information. However, with the explosive growth of data, issues such as data sparsity, cold start, and poor interpretability have become more prominent, significantly limiting the generalization and effectiveness of the models (Su & Khoshgoftaar, 2009). To address the issues above, many researchers have proposed recommender systems that incorporate additional information to enhance the models' representation ability. Examples of such systems include review-based recommendation (Pan et al., 2022), social relationship-based recommendation (Tang et al., 2022), and knowledge graph-based recommender systems (Chen et al., 2022). Relevant experiments have demonstrated that these recommendation systems, which incorporate additional information, achieve better recommendation performance than traditional methods. However, additional information can pose challenges, such as incomplete or inaccurate data and redundant or misleading review content that could negatively impact the recommendation quality. Moreover, such recommendation systems may require more complex models to handle such information, which can increase model complexity and computational requirements.

Nevertheless, one of the current trends in research is to utilize user-generated data, such as user reviews and user tags, as inputs for recommendation algorithms (Kanwal et al., 2021). As a typical information-enhanced recommendation system, the advantage of review-based recommendations is primarily from user-generated specific reviews about the target items, which can utilize the rich information embedded in the comment context to construct user/item features (Wu, He, Wang, et al., 2023). These reviews contain the reasons for user ratings and provide additional descriptions of items, reflecting user sentiments and preferences and objectively portraying the attributes of items, including appearance, functionality, utility, etc. When the rating data of particular users is sparse, analyzing their written comments can effectively supplement their preference features and alleviate data sparsity and cold start problems (Chen et al., 2015). In addition, the model can summarize the reasons for recommendations from comments, which helps the model better understand user behavior and intention and improves the interpretability of the recommendation system. Although existing review-based recommendation systems have demonstrated good recommendation performance, they often need complex model parameter training and long inference time due to the large number and length of comments. For instance, 32% of comments in Amazon's Automobile dataset are longer than 100 words, with an average length of 106 words. However, user preferences and item attributes often only relate to a few keywords or sentences. Consequently, long reviews may introduce a large amount of noise, resulting in unclear extraction of user/item features by the model. In addition, existing review-based models still face challenges such as inadequate representation of user preferences and item attributes and insufficient interaction between user and item features. The specific problems are as follows:

(1) Comment texts can be too lengthy, leading to much noise and unclear user preferences and item attributes. Additionally, existing models lack the ability to learn the dependency relationships within long comment texts, which limits their accuracy in prediction.
(2) Although existing models can distinguish the importance of different sentences in a comment, they neglect the importance of word-level information in modeling user and item features.
(3) Existing models typically use linear connection methods to fuse user and item features, resulting in ignoring the correlation and interaction between the two features.

To tackle the challenges mentioned above, we propose Joint Inter-Word and Inter-Sentence Multi-Relation Modeling for the Summary-based Recommender System (MRSR), which removes noise data from the reviews and extracts concise and clear summary information to learn user and item feature representations. Examples in Fig. 1 show the summaries and the reviews of the same items. According to the rating, user Tom's evaluation of the clothes is positive, and both the review and summary contain Tom's preference for the clothes, such as material and season (bold part). However, the review also includes useless information (underlined part). Similarly, user Mike's evaluation of the clothes is negative, and the review and summary contain the description for style, quality, and price. And there is also some noise in his review. Therefore, it can be seen that the user's summary and review contain the same amount of information that represents user preference and item attributes. Compared to reviews, summaries are more concise and have less noise, making them more suitable for recommendation in today's big data environment. To model the Summary information at different granularities, MRSR employs two multi-relationship modeling modules, the inter-word and inter-sentence modules, to learn the user and item features contained in the Summary information. The former utilizes Transformer (Vaswani et al., 2017) and a pooling layer to perceive the relative relationships between words in a single summary and learn representative features. At the same time, the latter employs a self-attention mechanism to differentiate the importance of different summaries in predicting the rating and constructing sentence-level feature representations. The multi-headed attention (MHA) fusion layer is then used to learn the correlation of user and item features, performing feature fusion with weighted importance. Finally, the attentional factorization machine (AFM) (Xiao et al., 2017) is utilized to realize deep interaction and computation between high-order nonlinear features of user–item fusion features for rating prediction.

1.1. Research objectives and contributions

In order to mitigate the adverse effects of extraneous information in reviews on recommendations, our focus is to utilize concise and explicit summaries as the representation data for users and items. Based on this, a multi-relational modeling module and

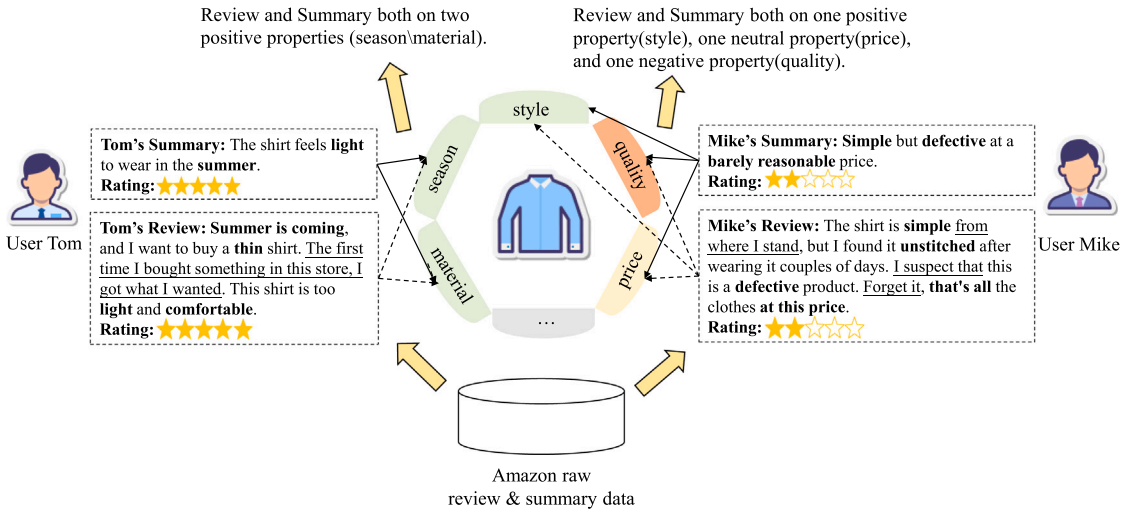

Fig. 1. Summary vs. Review. Both summaries and reviews contain equal information on the user's tendencies and item's attributes, but it is clear that the summaries are more concise and have less noise than the reviews.

an attention-based fusion layer are designed to effectively improve the representation of user preferences and item attributes and enhance the correlation between user–item features. To sum up, the research objectives of this study can be summarized as follows: (1) a summary-based recommender system is designed to represent user preferences and item attributes more clearly and straightforwardly; (2) both inter-word and inter-sentence relations in the summary are utilized to enrich the representation of user and item; (3) both shallow and deep interactions between user and item features are achieved through a multi-head attention feature fusion layer and an attention-based factorization machine.

Our contributions could be summarized as follows:

(1) This article proposes using summaries rather than reviews as the data source for feature extraction in MRSR. Furthermore, the MRSR effectively mitigates the issue of long-text dependence by Transformer, which models global context features.
(2) The design of the inter-word and inter-sentence multi-relation modeling module aims to extract fine-grained word-level and coarse-grained sentence-level text features from the summary, thereby enhancing the expression and learning ability of user preferences and item attributes.
(3) MRSR employs the MHA fusion module and AFM prediction module to facilitate both shallow and deep interactions between user and item features. The MHA fusion module is employed to understand the correlation (shallow interaction) between the information expressed by user–item features. On the other hand, the AFM prediction module predicts ratings (deep interaction) by combining high-order attention-based computations with linear low-order computations, thereby enhancing the precision of prediction.
(4) The experiments with MRSR were conducted on five public datasets from Amazon, demonstrating that MRSR outperforms state-of-the-art methods. The proposed method improves the recommender system's performance and accuracy, enhancing its learning and prediction efficiency.

The remaining sections of this paper are outlined as follows: In Section 2, we provide an overview of related works in the recommendation field, encompassing both deep learning-based and review-based recommender systems. Section 3 introduces the architecture of MRSR. Moving on to Section 4, we outline the design of four experiments conducted to validate the superiority of our proposed method. Section 5 offers a comprehensive summary of the research and discusses its theoretical and practical significance. Finally, Section 6 concludes the article with a recap and provides insights into potential avenues for future research.

2. Related work

2.1. Deep learning for recommendation

In recent years, convolutional neural networks (CNNs) have been used in various fields (Li et al., 2024; Li, Zhao, et al., 2023; Wang, Zhang, et al., 2023; Wang, Zhao, et al., 2023; Wu et al., 2021). Researchers have introduced CNNs into traditional recommendation algorithms to represent complex relationships between a large number of users and items, leading to improved predictive performance of the models. For instance, approaches such as Wide & Deep (Cheng et al., 2016) and Deep & Cross (Wang et al., 2017) model historical interaction information to enhance the model's ability to memorize past data and generalize to inactive users or rare items. DeepFM (Guo et al., 2017) and FNN (Zhang et al., 2016), on the other hand, leverage factorization machines (FM) (Rendle, 2010) and deep networks to respectively model low-dimensional and high-dimensional features, enhancing

the learning speed and expressive power of FM. These techniques have shown great promise in improving recommender systems by effectively capturing intricate user–item relationships and leveraging historical interaction data for more accurate predictions.

With the advancement of deep learning, various recommender systems have been designed in recent years. In sequential recommendation, notable examples include MLP4Rec (Li et al., 2022b) and Bert4Rec (Sun et al., 2019), which employ dynamic modeling of item interaction sequences and user behavior sequences, respectively, to facilitate sequential recommendations. Dialog-based recommender systems aim to capture user preferences through multi-turn interactions. UCCR (Li, Xie, et al., 2022), centered around the user, emphasizes the similarity between historical and current dialogs to model user preferences based on past and present information. Additionally, UniMIND (Deng et al., 2023) is a multi-objective dialog system that leverages multiple user preferences and dependencies to model user features. In social recommendation, SMIN (Long et al., 2021) extracts user and item relationship structures, aggregating dependencies from different user and item types. This approach enables the construction of a self-supervised social recommendation. These recent advancements in deep learning have led to the development of diverse and effective recommendations.

The above deep learning-based models have achieved good recommendation results, proving the feasibility of combining recommender systems with deep learning. However, they still need to fundamentally solve critical problems such as cold start and poor interpretability of recommender systems. Therefore, researchers have introduced reviews into recommender systems, effectively alleviating these issues.

2.2. Review for recommendation

With the great success of deep learning in natural language processing tasks, many researchers have introduced user reviews into recommender systems. One approach is to extract topic features from the review text using topic models, and then integrate the features into Collaborative Filtering or Matrix Factorization models for prediction (Bao et al., 2014; Ling et al., 2014; McAuley & Leskovec, 2013). Though these methods provide a higher degree of accuracy, they often rely on tons of manual pre-processing processes and cannot extract relevant features other than the score and interactive information, resulting in a lack of partial information.

Therefore, researchers have also explored using extracted context features from reviews to predict user and item preferences in recommendations. Approaches like DeepCoNN (Zheng et al., 2017) and ConvMF (Kim et al., 2017) utilize parallel CNN networks to extract user preferences and item attributes from comments, effectively addressing the issue of data sparsity. Introducing attention mechanisms further enhances the model's ability to analyze and process text, leading scholars in the recommendation domain to apply attention mechanisms to comment-based recommender systems. For example, NARRE (Chen et al., 2018) and TARMF (Lu et al., 2018) extract comment features and then use attention mechanisms to learn the importance of reviews in user–item interactions. MPCN (Tay et al., 2018) used similarity matrices to calculate attention weights, selecting the most critical textual features for user–item interactions.

In recent years, some researchers used user's comments for prediction and focused on enhancing the interpretability of recommender systems through additional approaches. Counterfactual Review-based Recommendation (Xiong et al., 2021) aimed to extract user preferences from reviews and use counterfactual learning to seek simple and effective explanations for the model's decisions. NRCMA (Luo et al., 2021) associated the features of user and item encoders to select informative words and comments for feature representation. CFAA (Liu, Zheng, et al., 2022) tackled the Non-overlapped Recommendation (RNCDR) problem by using review to establish user and item features to eliminate differences between different domains.

3. Our approach

The paper proposes MRSR, as depicted in Fig. 2, which includes a summary data preprocessing module, an inter-word and inter-sentence multi-relation modeling module, and a fusion and prediction module. Firstly, to reduce the amount of noise in the data and simplify the model training, this study uses existing summary information in the dataset as the data source. All summaries from a user are used to represent the user's preferences, and all summaries about a particular item are used to describe the item's attributes. Secondly, to address the problem of insufficient expression ability of user and item features, an inter-word and inter-sentence multi-relation modeling module is designed to enrich the information contained in the features. Specifically, the summary is processed by an embedding layer and a positional encoding layer. A Transformer and a pooling layer are applied to achieve fine-grained inter-word relation modeling, and a self-attention mechanism is employed to achieve coarse-grained inter-sentence relation modeling. Finally, to deal with the issue of insufficient interaction between user and item features, MRSR employs the MHA-based fusion module to achieve shallow interaction fusion between user and item features, and then the AFM prediction module performs attentional high-order non-linear deep interaction calculation on the user–item feature vector to get the predicted rating.

3.1. Summary data process

Compared to reviews, the conciseness and explicit expression of information in summaries can simplify the complexity of learning user and item features in MRSR, enabling it to express user preferences and item attributes more efficiently and accurately. Therefore, in this work, we use summaries from the Amazon datasets as data sources for extracting user and item features. Specifically, the user feature $\boldsymbol{f}_u$ characterizes the user's interest preferences and emotional polarity reflected in all summaries for user u, while the item

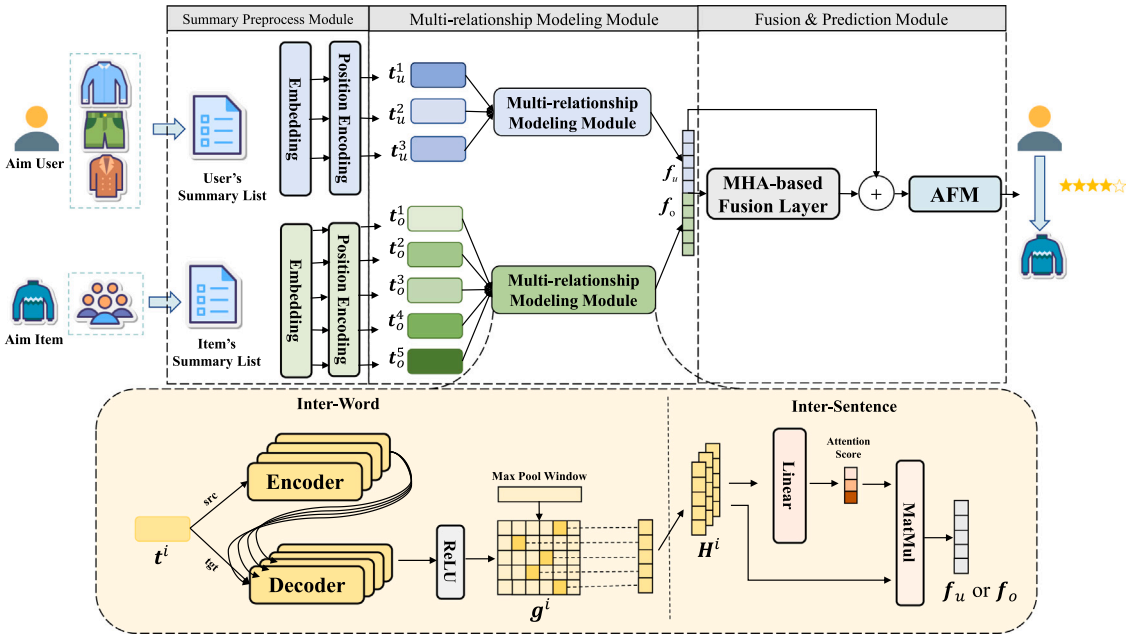

Fig. 2. The architecture of MRSR.

feature, represented by $\boldsymbol{f}_o$, characterizes the item attributes reflected in all user summaries for the item o. To model user inclination and item attributes, it is necessary to preprocess the summaries and convert them into a low-dimensional dense word vector matrix.

First, the total number of summaries s_i for user u and item o within a certain period are represented by n and m, respectively. The corresponding sets of comment summaries for user and item are denoted as $S_u = \{s_0, s_1, \ldots, s_n\}$ and $S_o = \{s_0, s_1, \ldots, s_m\}$.

Next, the average length L of all summaries in S is computed, and the lengths are normalized (longer summaries are truncated, and shorter summaries are zero-padded). Pretrained BERT (Devlin et al., 2019) corpus (Bert Dictionary) and tokenizer are used to tokenize the summary texts, and the word-level data is encoded and converted into a word-level encoded sequence S'.

Subsequently, all word-level encoded sequences S' are mapped to a dense matrix $\boldsymbol{C}$ of dimension d through an embedding layer. Assuming there are N encoded sequences, then $C \in \mathbb{R}^{N\times L\times d}$ The initialization word vector matrix $C^i \in \mathbb{R}^{L\times d}$ for the ith (i = 1, 2, 3 ... N) summary s^i is expressed as:

$$\boldsymbol{C}^i = \begin{bmatrix} | & | & | & \\ \ldots & \boldsymbol{c}^{i,p-1} & \boldsymbol{c}^{i,p} & \boldsymbol{c}^{i,p+1} & \ldots \\ | & | & | & \end{bmatrix} \tag{1}$$

where $\boldsymbol{c}^{i,p} \in \mathbb{R}^d$ represents the word vector corresponding to the pth (where $p = 1, 2, 3, \ldots, L$) word in the ith summary, with a corresponding learnable weight $\boldsymbol{w}^{i,p}$ that is updated during training. The weighted average calculation of the initial word vectors $\boldsymbol{c}^{i,p}$ yields the relevant vector representation $\boldsymbol{m}^{i,p}$. The vector representations of the pth word in the ith summary of user u and item o can be respectively represented as:

$$\boldsymbol{m}_u^{i,p} = \frac{\boldsymbol{w}_u^{i,p}\boldsymbol{c}_u^{i,p}}{\sum_{j=1}^{d}\boldsymbol{w}_u^{i,p}} \tag{2}$$

$$\boldsymbol{m}_o^{i,p} = \frac{\boldsymbol{w}_o^{i,p}\boldsymbol{c}_o^{i,p}}{\sum_{j=1}^{d}\boldsymbol{w}_o^{i,p}} \tag{3}$$

The low-dimensional dense word vector matrices obtained by embedding layer for S_u and S_o are represented as $\boldsymbol{M}_u = \{\boldsymbol{m}_u^0, \boldsymbol{m}_u^1, \ldots, \boldsymbol{m}_u^N\}$ and $\boldsymbol{M}_o = \{\boldsymbol{m}_o^0, \boldsymbol{m}_o^1, \ldots, \boldsymbol{m}_o^N\}$.

Finally, position encoding is applied on the word vector matrix $\boldsymbol{M}$ to differentiate the information of words at different positions. By using alternating sine and cosine functions, the hidden interaction features of the $2k$-th(even position) and $2k+1$-th(odd position) elements corresponding to the pth word in the summary s_i are calculated. Then, the position encoding results are added to the initial word vectors $\boldsymbol{m}^i$ to obtain the final vector representation $\boldsymbol{t}^i \in \mathbb{R}^{L\times d}$. The relevant formulas are as follows:

$$\mathrm{PE}(p, 2k) = \sin\left(\frac{p}{10000^{\frac{2k}{d}}}\right) \tag{4}$$

$$\mathrm{PE}(p, 2k+1) = \cos\left(\frac{p}{10000^{\frac{2k+1}{d}}}\right) \tag{5}$$

$$t^i_{p,k} = m^i_{p,k} + \mathrm{PE}(p,k) \tag{6}$$

where $t^i_{p,k}$ represents the final representation of the pth word's kth dimensional feature element in the comment summary s_i.

This module processes all the summaries of the target user and other users' summaries on the target item, converting them into two low-dimensional dense vector representations, which serve as inputs to the inter-word and inter-sentence multi-relation modeling module.

3.2. Inter-word and inter-sentence multi-relation modeling module

The inter-word and inter-sentence multi-relation modeling module aims to construct feature representations of summary information from two different perspectives: word-level and sentence-level. The function of word-level is realized by the Inter-Word Module, which is designed to analyze the relationships between words in the same sentence. The module extracts the most salient word features within a sentence as its representation. Subsequently, the Inter-Sentence Module utilizes a linear attention layer to assign attention weights to the feature vectors of each sentence. These weighted features are then mapped to a one-dimensional space, serving as the behavior features for users or the attribute features for items.

3.2.1. Inter-word relation modeling

To build user and item features from the perspective of word relations, MRSR utilizes Transformer and pooling layers to analyze the correlation and representational information between words to model user preference and item attributes. Taking user features as an example, the vector representation of the preprocessed ith summary text s^i_u is denoted as t^i_u. The masked version of t^i_u is then computed, where the attention weights of the [PAD] part of the original summary are set to 0, and these parts are not calculated by Transformer.

The masked t^i_u is then used as input to the Encoder and Decoder of the Transformer. The Encoder performs nonlinear transformations based on multi-head attention on the text vector t^i_u to get a hidden vector representation $e^i_u \in \mathbb{R}^{L\times d}$. Then, the Decoder uses the original text vector t^i_u and e^i_u to learn text features at different time steps, and the output feature vector is obtained through a ReLU nonlinear activation layer to get the final contextual text feature $g^i_u \in \mathbb{R}^{L\times d}$. The Transformer can be seen as a function with two parameters: the original text and the target text. The formula is as follows:

$$g^i_u = \mathrm{ReLU}\left(\mathrm{Transformer}_{\Theta}(t^i_u, t^i_u)\right) \tag{7}$$

where Θ represents the learnable parameters of Transformer.

Finally, the most significant word feature in the contextual feature vector is extracted through the maximum pooling layer as a single user propensity feature representation $h^{i,j}_u \in \mathbb{R}^d$. A user has N summary texts, so the word-level feature module captures N feature vectors from fine-grained text information, denoted as $H_u \in \mathbb{R}^{N\times d}$. The formula is as follows:

$$h^i_u = \max\left(g^{i,1}_u, g^{i,2}_u \dots g^{i,d}_u\right) \tag{8}$$

$$H_u = \left\{h^1_u, h^2_u \dots h^3_u\right\} \tag{9}$$

Similarly, the summaries corresponding to each item are subjected to word-level relationship modeling to obtain word-level feature vector representations of the related summaries. These are denoted as $H_i \in \mathbb{R}^{N\times d}$.

3.2.2. Inter-sentence relation modeling

A user or an item typically has more than one summary that contains descriptions of different preferences and attributes. To differentiate the importance of these summaries in predicting user–item pairs, the model needs to identify user preferences for different items and the correlation of different users' evaluations for the same item, thus allowing for a better understanding of expressing preferences and characteristics. Therefore, we employ a self-attention mechanism to calculate the relative weights among multiple user preference features and item attribute features. Taking user features as an example, a fully connected network is used to learn the interaction information between behavior features and generate a latent interaction feature vector. This latent interaction feature vector describes the different importance of different summary text features for the result. The latent interaction feature vector is non-linearly mapped onto a d-dimensional space by the Softmax function to obtain attention weights, which are calculated as inner products with the user features generated by the inter-word relation modeling to obtain the final user feature vector $f_u \in \mathbb{R}^d$. The computation process can be represented as follows:

$$f_u = \mathrm{Softmax}\left(W_H H_u + b_H\right) \times H_u \tag{10}$$

where W_H, $b_H \in \mathbb{R}^N$ are weight and bias values of the fully connected layer, respectively. Similarly, the multiple item attribute feature vectors H_i are modeled through inter-sentence relationships, and the attention mechanism is used to analyze the correlations among multiple item attribute features, resulting in the final item feature vector f_o.

The pseudo code of this subsection is shown in algorithm 1.

Algorithm 1: Multi-Relation Modeling Module

Input: Token set of User's summaries $S'_u \in \mathbb{R}^{N\times L}$; Token set of Item's summaries $S'_i \in \mathbb{R}^{N\times L}$.
Output: User's feature f_u; Item's feature f_o.

// extracting user's feature
for *each* s_u *in* S'_u **do**
  // Word Embedding:mapping s_u into a dense matrix
  Do Word embedding and Position Encoding to s_u and get $\boldsymbol{t}_u^i$
  // Inter-Word Modeling Module:extracting feature between the words
  Extracting context features with Transformers: $\boldsymbol{g}_u^i = \mathrm{ReLU}\left(\mathrm{Transformer}_\Theta(t_u^i, t_u^i)\right)$
  Extracting most salient features of each words: $\boldsymbol{h}_u^i = \mathrm{MaxPool}\left(\boldsymbol{g}_u^{i,1}, \boldsymbol{g}_u^{i,2} \ldots \boldsymbol{g}_u^{i,d}\right)$
// Inter-Sentence Modeling Module:extracting feature between the sentences
Let $\boldsymbol{H}_u = \left\{\boldsymbol{h}_u^1, \boldsymbol{h}_u^2 \ldots \boldsymbol{h}_u^3\right\}$
Do Linear Attention to the sentence sets and obtain User's feature $f_u \in \mathbb{R}^d$: $f_u = \mathrm{LinearAttention}(\boldsymbol{H}_u)$
// extracting item's feature
for *each* s_o *in* S'_o **do**
  Do Word embedding and Position Encoding to s_o and get $\boldsymbol{t}_o^i$
  Extracting context features with Transformers: $\boldsymbol{g}_o^i = \mathrm{ReLU}\left(\mathrm{Transformer}_\Theta(t_o^i, t_o^i)\right)$
  Extracting most salient features of each words: $\boldsymbol{h}_o^i = \mathrm{MaxPool}\left(\boldsymbol{g}_o^{i,1}, \boldsymbol{g}_o^{i,2} \ldots \boldsymbol{g}_o^{i,d}\right)$
Let $\boldsymbol{H}_o = \left\{\boldsymbol{h}_o^1, \boldsymbol{h}_o^2 \ldots \boldsymbol{h}_o^3\right\}$
Do Linear Attention to the sentence sets and obtain User's feature $f_o \in \mathbb{R}^d$: $f_o = \mathrm{LinearAttention}(\boldsymbol{H}_o)$
Return f_u, f_o

3.3. Fusion and prediction

To enhance the interaction between user and item features, MRSR uses a fusion module based on MHA to shallowly integrate the user feature vector $\boldsymbol{f}_u$ and item feature vector $\boldsymbol{f}_o$ into a fused feature representing the user–item interaction $\boldsymbol{f} \in \mathbb{R}^{2d}$. Then, a prediction module based on AFM is used to calculate high-order non-linear deep interactions on the fused feature, and the results of different orders are added together to obtain the predicted rating score $\hat{r}$.

3.3.1. Fusion module based on multi-head attention

The approach used by MRSR for fusing user and item features differs from baseline methods. Instead of the traditional fusion method, an MHA-based fusion module analyzes the correlations between different feature components and achieves shallow interaction fusion between user and item features.

First, the user feature $\boldsymbol{f}_u$ and item feature $\boldsymbol{f}_o$ are concatenated to obtain the connection vector $\boldsymbol{f}_c \in \mathbb{R}^{2d}$. This vector is used as the input to the MHA fusion module, where the calculation of multi-head attention is as follows:

$$\mathrm{Multihead}\,(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Concat\left(head_1, head_2 \ldots, head_k\right) \tag{11}$$

$$head_i = \mathrm{Softmax}\left(\frac{\boldsymbol{Q}_i \boldsymbol{K}_i^T}{\sqrt{d_\kappa}}\right) \times \boldsymbol{V}_i \tag{12}$$

The variables $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$ are the inputs to the multi-head attention mechanism, where $\boldsymbol{Q}_i$, $\boldsymbol{K}_i$, $\boldsymbol{V}_i \in \mathbb{R}^{d_k}$ are sub-vectors of $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$ vectors obtained by mapping through a linear layer in a lower-dimensional space, and $head_i$ denotes the calculation result for each head. Here, d_k represents the dimension of the κ-th space. These vectors are all assigned to the joint vector $\boldsymbol{f}_c$, and multi-head self-attention is performed. To improve the performance of the MRSR fusion module, the computation result of the multi-head attention is subject to residual and normalization processing, followed by a fully connected layer to obtain the final fused feature $\boldsymbol{f}$. The calculation is expressed as:

$$\boldsymbol{x} = \mathrm{Multihead}\left(\boldsymbol{f}_c, \boldsymbol{f}_c, \boldsymbol{f}_c\right) + \boldsymbol{f}_c \tag{13}$$

$$\boldsymbol{f} = \left(\frac{\boldsymbol{x} - \mathrm{E}(\boldsymbol{x})}{\mathrm{Var}\,(\boldsymbol{x})} + \boldsymbol{x}\right) W_{\boldsymbol{x}} + \boldsymbol{b}_{\boldsymbol{x}} \tag{14}$$

where $\boldsymbol{x} \in \mathbb{R}^{2d}$ is the results of MHA, $W_{\boldsymbol{x}} \in \mathbb{R}^{2d\times 2d}$ and $\boldsymbol{b}_{\boldsymbol{x}} \in \mathbb{R}^N$ are learnable parameters of the fully connected layer. $\mathrm{E}(\boldsymbol{x})$ and $\mathrm{Var}\,(\boldsymbol{x})$ represent the mean and variance of $\boldsymbol{x}$, respectively.

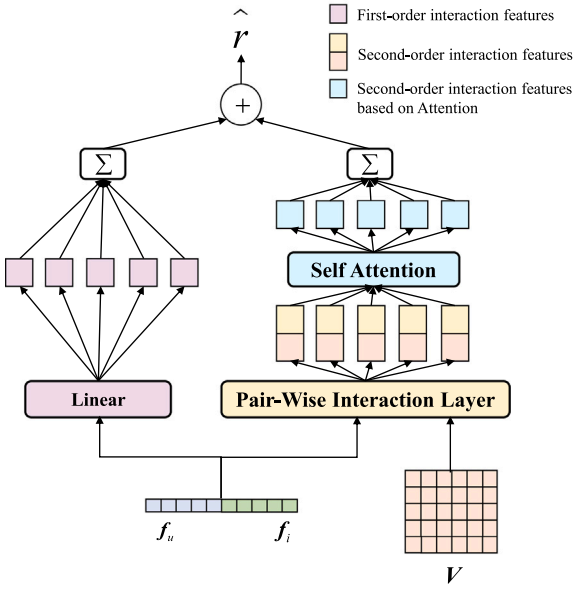

Fig. 3. The architecture of AFM-based prediction Module.

3.3.2. *Prediction module based on AFM*

To fully utilize the information from different levels of the fused feature vector $\boldsymbol{f}$ during prediction, an AFM-based prediction module is used to calculate the first and attentional second orders of $\boldsymbol{f}$, achieving deep interaction computation and ultimately obtaining the prediction result. The structure of the module is illustrated in Fig. 3.

Firstly, the prediction module based on AFM takes the fusion feature vector, denoted as $\boldsymbol{f}$, as input to a fully connected layer to obtain the AFM's first-order linear feature. To address the problem of insufficient model learning due to data sparsity, an auxiliary matrix $\boldsymbol{V} \in \mathbb{R}^{2d \times k}$ is introduced to assign a dimension of k to each associated feature element $\boldsymbol{f}_j$. Then, the auxiliary matrix $\boldsymbol{V}$ is combined with the interaction feature $\boldsymbol{f}$, and both are simultaneously fed into the Pair-wise Interaction layer to obtain the AFM's second-order interaction feature. Finally, the attention mechanism is applied to optimize the second-order feature by learning the importance of each part of the feature for the prediction result.

During the rating prediction stage, this module sums up the results of the high-order non-linear deep interactions, i.e., the first-order linear features and the second-order interaction features based on the attention mechanism. Then, it linearly adds the sum to obtain the predicted rating value, denoted as $\hat{r}$. The complete formula is as follows:

$$\hat{r} = \varpi_0 + \sum_{j=1}^{2d} \varpi_j \boldsymbol{f}_j + z^T \sum_{p=1}^{2d} \sum_{q=p+1}^{2d} a_{pq} \left(\boldsymbol{v}_q \odot \boldsymbol{v}_p \right) \boldsymbol{f}_q \boldsymbol{f}_p \tag{15}$$

where $\varpi, z^T \in \mathbb{R}^{2d}$ represent the weights of the first-order and second-order combined features respectively, $\boldsymbol{v}q \in \mathbb{R}^k$ is the auxiliary vector corresponding to the qth feature value, and a_{qp} represents the weight of the qth and pth feature values.

Finally, the overall framework pseudocode of MRSR is represented as Algorithm 2.

4. Experiment

4.1. *Dataset*

We performed extensive experiments on MRSR employing several datasets spanning diverse domains. Five datasets were procured from Amazon[1], namely Wine, Office_and_School_Supplies (OSS), Automobile, Musical_Instrument (MI), and Amazon_Instant_Video (AIV), furnishing summary data. These five datasets serve to evaluate the overarching performance of the models delineated in the presented paper. Then, we utilized the library named "sumy" to generate summaries from reviews and created an additional three datasets: Automobile_5 (Sumy), Musica_Instrument_5 (Sumy), and Yelp2020_5 (Sumy) (from Yelp[2]), using which we conducted ablation experiments(in section 4.6). The ratings of all the datasets fall within the range of [1, 5]. Due to the presence of a long-tail effect in the datasets, we filtered active users based on the dataset size: (1) Unfiltered active users: Wine, Office_and_School_Supplies;

[1] http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/

[2] https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset?select=yelp_academic_dataset_review.json

Algorithm 2: MRSR algorithm

Data: Review Records List:[...,{user:u, item:o, summary:s, rating:r},...]; Pretrained BERT Vocabulary.
Input: all learnable parameters: Ω; the amount of Transformer latent factors: d; the amount of attention heads: κ; the amount of AFM latent factors: k; batch size b and learning rate lr; evaluation function RMSE(r, r̂).
Output: Predicted Rating: $\hat{r}$

Initialize BERT Vocabulary and do data pre-processing;
Split the randomized data into Training Set $\mathcal{S}$ and Test Set $\mathcal{T}$;
loop
  // The actual batch operation is realized by gradient accumulation
  for *each* S_u, S_o *and* r *in* $\mathcal{S}_b$ **do**
    Utilize algorithm 1 to obtain user's and item's feature: $f_u, f_o = \text{Multi}-\text{Relation}(S_u, S_o)$
    Concat f_u and f_o and do fusion based on Multi-head Attention
    Do prediction based on AFM: $\hat{r} = \varpi_0 + \sum_{j=1}^{2d} \varpi_j \boldsymbol{f}_j + z^T \sum_{p=1}^{2d} \sum_{q=p+1}^{2d} a_{pq} \left(\boldsymbol{v}_q \odot \boldsymbol{v}_p\right) \boldsymbol{f}_q \boldsymbol{f}_p$
    // $\hat{r}(\Omega, d, \kappa, k)$ represents the ratings predicted under the parameter (Ω, d, κ, k)
    $\Omega' \leftarrow \text{RMSE}(\text{r}, \hat{\text{r}}(\Omega, \text{d}, \kappa, \text{k}))$
  for *each* S_u, S_o *and* r *in* $\mathcal{T}$ **do**
    Do Evaluation on Test set: $\text{RMSE}(\text{r}, \hat{\text{r}}(\Omega', \text{d}, \kappa, \text{k}))$
end loop;

Table 1
Statistical details of the datasets for recommender system.

Dataset	#users	#items	#ratings&reviews	#words/review	#words/summary	#reviews/user
Wine	1920	1228	2396	71	6	1.2
Office and School Supplies	10 641	3229	10 853	76	5	1.0
Automobile_5	2928	1853	20 437	108	6	7.0
Music_Instrument_5	1429	900	10 261	115	6	7.2
Amazon_Instant_Video_10	690	903	8685	183	7	12.6
Yelp2020_5	18 144	25 975	154 170	137	–	8.5
Automobile_5(Sumy)	2928	1853	20 437	108	21	7.0
Music_Instrument_5(Sumy)	1429	900	10 261	115	19	7.2
Yelp2020_5(Sumy)	18 144	25 975	154 170	137	23	8.5

(2) Users with a minimum of 5 reviews considered active: Automobile_5, Musical_Instrument_5, Yelp2020_5; (3) Users with a minimum of 10 reviews considered active: Amazon_Instant_Video_10. Specific details for each dataset are provided below: (see Table 1).

4.2. Tested models:

Twelve models are included in our experiment as follows:

(1) PMF (Salakhutdinov & Mnih, 2007): a classic matrix factorization model based on rating prediction.
(2) SVD (Koren, 2008): maps user and item information to a joint latent space and utilizes these latent features to predict ratings.
(3) SVD++ (Koren, 2008): a model that combines factorization and neighborhood models, using explicit interaction feedback to enhance model prediction.
(4) ConvMF (Kim et al., 2017): combines convolutional neural network (CNN) and probabilistic matrix factorization (PMF) to capture context information and handle Gaussian noise differently.
(5) DeepCoNN (Zheng et al., 2017): learns latent feature vectors of users and items through two parallel CNNs trained on user and item reviews, and uses FM to predict ratings.
(6) NARRE (Chen et al., 2018): utilizes two parallel CNN networks to learn latent features of reviews, and furthermore consider the usefulness of reviews for recommendation quality with attention mechanisms before predicting ratings.
(7) TARMF (Lu et al., 2018): jointly learns user and item information from ratings and customer reviews by optimizing matrix factorization and attention-based GRU networks.
(8) MPCN (Tay et al., 2018): By using a pointer mechanism to directly match reviews and leveraging word-level attention interaction to learn a fixed-dimensional representation, a multi-pointer learning scheme is further introduced to extract multiple multi-level interactions between user and item reviews.
(9) CARL (Wu et al., 2019): learns user–item pair-related features using CNNs and obtains the final rating through a dynamic linear fusion mechanism.

Table 2
Details of parameter settings in comparison models.

Method	Parameter1	Values	Parameter2	Values
PMF	Latent factors	20, 30, 40	–	–
SVD	Latent factors	20, 30, 40	–	–
SVD++	Latent factors	20, 30, 40	–	–
ConvMF	Shared weights per window size	50, 100, 200	Latent factors	50, 100
DeepCoNN	Latent factors	20, 50, 80	kernel size	50, 100, 150
NARRE	Latent factors	16, 32, 64	–	–
TARMF	Latent factors	64, 128, 256	Attention dimension	64, 128, 256
MPCN	#pointers	2, 4, 5	Latent factors	50, 100
CARL	Latent factors	15, 25, 50	#conv. filters	50, 100
DAML	Latent factors	8, 16, 64	Kernel size	3, 5
MRSR-FM	Latent factors	32, 64, 128, 256	#attention heads	1, 2, 4, 8
MRSR	Latent factors	32, 64, 128, 256	#attention heads	1, 2, 4, 8

(10) DAML (Liu et al., 2019): composed of two parallel CNNs that learn user behavior and item attributes from reviews, and utilize neural factorization machines to fuse user and item features and predict final rating scores.
(11) MRSR-FM: The Multi-Relationship Modeling Module is proposed in this paper for feature extraction. To validate the effectiveness of the AFM-based prediction module, we substitute it with Factorization Machines (FM) for comparison.
(12) MRSR: The model introduced in this paper. MRSR employs the Multi-Relationship Modeling Module to extract user and item features. Besides, it utilizes the MHA-based fusion module and the AFM-based prediction module to facilitate both shallow and deep interactions between the two, ultimately leading to accurate rating predictions.

4.3. Evaluation metrics

To evaluate the effectiveness of the model, our study uses RMSE (Root Mean Square Error) as the evaluation metric. RMSE measures the difference between the predicted and actual values by taking the square root of the average of the squared differences between the predicted and actual values. The formula is as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(r_{u,i} - \hat{r}_{u,i})^2} \tag{16}$$

where $\hat{r}_{u,i}$ represents the score predicted by the model, $r_{u,i}$ is the true score, and N represents the total score of the test set.

4.4. Experiment settings

Text Length: After random sorting, 80% of the data is used as the training set and 20% as the test set, and the data is pre-processed as follows: (1) As there is a significant difference in the length of reviews or summaries, padding all sequences to the same length may add excessive irrelevant information and affect the model's performance. Therefore, we set the maximum sequence length as the average length of all texts, and pad sequences shorter than this length with [PAD] token while discarding longer sequences. (2) We converted the texts into word-level sequences using the BERT-uncased vocabulary as a reference.

Common Parameters Details: The Adam optimizer is used for fast gradient descent at the beginning of training. After training stops, the parameters and training data are saved, and the stochastic gradient descent (SGD) algorithm is used to further search for the optimal result. Besides, we set the learning rate to {0.001, 0.002, 0.003}, weight decay to 0.001, and dropout probability to 0.5. Instead of setting the number of training epochs, we set a stopping threshold of 1×10^{-4}. When the difference between the training losses of two consecutive epochs is less than the threshold, the model can be considered to have completed training and automatically stops. We executed each model over three times with the optimal parameters and assessed the model performance based on the minimum RMSR score.

Hyperparameter Setting: We tune the parameters for all the baselines following the strategies outlined in their papers and keep the parameters not explicitly mentioned the same as the original papers. The specific details of the parameter settings are presented in Table 2.

For MRSR, in more detail, the default setting of 6 layers is used for both the Encoder and Decoder in the Transformer. The dimension of the implicit factor matrix in AFM is set to 20, and the feature dimension is optimized within the range of {32, 64, 128, 256}, while the number of attention heads is optimized within the scope of {1, 2, 4, 8}. Further experiments showed that the optimal RMSE score was achieved on the test set when the feature dimension was 256 and the number of attentional heads was 4.

4.5. Overall comparison

To validate the effectiveness of MRSR, the experiments in this section compare MRSR and all baseline models on five publicly available datasets from Amazon, and the results are shown in Table 3. Underlined parts indicate the RMSE scores achieved by the

Table 3
Comparative results of RMSE on five datasets.

Method	Wine	OSSupplies	Automobile	MI	AIViedo	Avg.
PMF	2.6090	2.1806	0.8164	0.7034	1.4072	1.5433
SVD	0.6230	0.9902	0.6451	0.6319	0.9118	0.7604
SVD++	0.6006	0.9268	0.6019	0.6295	0.9134	0.7416
ConvMF	0.4365	0.8729	0.5217	0.5121	0.8066	0.6300
DeepCoNN	0.7096	0.8477	0.5434	0.5545	0.7045	0.6719
NARRE	0.7057	0.9889	0.5417	0.5374	0.6980	0.6945
TARMF	0.4525	0.7871	0.5220	0.5136	0.7674	0.6085
MPCN	0.4954	0.7842	0.5759	0.5733	0.6954	0.6248
CARL	0.8276	0.9294	0.5517	0.5703	0.7165	0.7191
DAML	0.5923	0.9037	0.5232	0.4985	0.7127	0.6461
MRSR-FM	0.5653	0.8700	0.5426	0.5310	0.7059	0.6432
MRSR	**0.4231**	**0.7671**	**0.5068**	**0.4776**	**0.6724**	**0.5724**
Improve	2.86%	1.15%	3.07%	4.19%	2.18%	5.94%

best baseline approach, and **bold parts** indicate the improved RMSE scores achieved on datasets and the improvement rates based on the best baseline approach.

Among these baseline methods, the first three methods only use user rating data for prediction. The remaining seven methods are review-based recommender systems, and our method is a summary-based recommender system. After the above comparative experiment, we validated the improvement of our method in different areas and obtained better behavior than state-of-the-art models. The findings are as follows:

The first three methods yielded unsatisfactory results, with PMF performing the worst. This suggests that using only Bayesian probability is insufficient in accurately representing user and item features. SVD improved upon PMF by incorporating the average historical rating and introducing user and item biases to represent their independence. SVD++ further enhanced SVD by incorporating explicit interaction information, leading to additional improvements.

Generally, models based on reviews performed better. ConvMF and TARMF combine natural language processing models with matrix factorization, allowing them to extract user and item information from comments and utilize statistical information from the data to make predictions. Experimental results show that these two methods have higher prediction accuracy than most review-based recommender systems. MPCN employs a pointer mechanism to match user and item reviews, learning distinctive importance features for both. It utilizes a word-level co-attention layer to capture comment features, resulting in high prediction accuracy. In contrast, NARRE and DeepCoNN leverage CNN to extract contextual features from reviews, exhibiting poor performance on the Wine and OOS datasets. This suggests that these two methods are less effective in addressing the long-tail effect. CARL and DAML models have many similarities, but the latter achieved the highest prediction accuracy among the baselines, demonstrating that utilizing NeuralFM for low-order and high-order feature interactions with neural network can effectively improve the accuracy.

Although intuitively, reviews are richer in corpus information than summaries, summaries contain the same amount of information about user tendencies and item characteristics as reviews. Instead, excessively long reviews introduce noisy information that affects the modeling of users items. Therefore, our modeling approach based on comment summaries is more advantageous regarding conciseness and amount of noise. The proposed MRSR achieved the best performance on all datasets, with an average improvement of 5.94% over the best baseline methods. The results demonstrate that incorporating summaries to the inter-word and inter-sentence multi-relation modeling module is able to capture user preferences and item attributes more effectively, while the deep interaction of features through fusion and prediction modules significantly improves prediction accuracy. Summaries contain more concise information than the comments, enhancing the model's modeling ability. The multi-relation modeling module can effectively extract user and item features from two levels. Additionally, the fusion and prediction layer based on MHA and AFM can fully integrate features, distinguish the degree of correlation between feature information, and effectively select information for prediction while reducing the interference of noisy information.

4.6. Summary vs review

The main innovation of this paper is the recommender system based on summaries. To prove the conciseness of summaries and their effectiveness in improving model prediction performance, we replaced the representation data sources of all baseline methods in Section 4.2 with summaries. We compared the RMSE and training time of these methods when using comments and summaries (taking Automobile as an example), and the experimental results are shown in Table 3 and Table 4, respectively (see Table 5).

The experimental results demonstrate that summaries can significantly enhance model performance, offering more precise information to the model and greatly reducing the training time. Specifically, most of the review-based recommender systems achieve a 1% to 2% increase in RMSE and a 10% reduction in training time when using summaries. Although some models, such as TARMF, show a decrease in prediction performance of 1.26%, their training time decreases by 17.29%. Overall, this result is acceptable, indicating that our proposed method is practical and efficient. Moreover, MRSR is designed for short texts like summaries, which uses Transformer to extract word-level features. However, using long text similar to comments not only reduces its ability to characterize inter-word relations but also introduces a lot of noise. Therefore, the prediction results using comments are unsatisfactory. According

Table 4
Comparison of RMSE using reviews and summaries for different models on Automobile dataset.

Method	Review	Summary	Improve
ConvMF	0.5217	0.5168	0.94%
DeepCoNN	0.5434	0.5360	1.36%
NARRE	0.5417	0.5324	1.72%
TARMF	0.5220	0.5286	−1.26%
MPCN	0.5759	0.5652	1.86%
CARL	0.5517	0.5429	1.60%
DAML	0.5232	0.5144	1.68%
MRSR	0.5703	0.5068	11.81%

Table 5
Comparison of training time using reviews and summaries for different models on Automobile dataset.

Method	Review (s)	Summary (s)	Improve
ConvMF	270.35	238.73	11.70%
DeepCoNN	283.17	245.64	13.25%
NARRE	335.18	298.95	10.81%
TARMF	203.58	168.39	17.29%
MPCN	608.67	550.08	9.63%
CARL	160.42	144.47	9.94%
DAML	283.94	247.39	12.87%
MRSR	2731.91	2446.59	10.44%

Table 6
Results of the ablation experiment on RMSE.

Method	Automobile	Music_Instrument	Avg	Improvement
IS	0.8215	0.8203	0.8209	\
+IW	0.7953	0.7701	0.7953	3.12%
+Fusion	0.6703	0.6798	0.6751	15.12%
+AFM	0.5068	0.4776	0.4922	27.09%

to the experiments, MRSR achieved an 11.81% improvement in prediction accuracy when using comment summaries instead of comments for prediction while reducing training time by 10.44%. The experiment demonstrates that the conciseness of comment summaries can decrease the complexity of Transformer in capturing contextual information, and the explicit user/item information contained in summaries enables the modeling of the inter-word and inter-sentence relationships to better represent user preferences and item attributes. It is worth mentioning that the Automobile dataset has limited comment data, so the improvement in training time is not notable. However, using summaries instead of comments in real-world industrial applications with massive data can remarkably improve recommendation efficiency while maintaining prediction accuracy.

4.7. Ablation experiments

4.7.1. Module-incremental ablation

To validate the effectiveness of each module in improving the prediction results, we conducted a module-incremental ablation experiment. We started with only using the Inter-Sentence module and Hadamard product interaction for prediction (referred to as IS). Then, we gradually added the Inter-Word module (+IW), MHA-based fusion module (+Fusion), and AFM-based prediction module (+AFM). The experimental results are presented in Table 6.

The experimental results demonstrate that the addition of each module leads to a significant improvement in prediction accuracy compared to the previous stage, indicating the effectiveness of the modules proposed in Section 3 for enhancing model accuracy. Specifically, adding the inter-word relationship module on top of the inter-sentence relationship modeling enables the model to learn feature relationships at the word level, resulting in an average improvement of 3.12% on the two datasets. The results demonstrate that modeling at different levels, namely, at the inter-word and inter-sentence levels, can better explore the fine-grained features in summary and increase the richness of the representation information. Building on top of the previous modules, adding the MHA-based fusion module led to a further increase in prediction accuracy by 15.12%, proving that conducting shallow-level interaction fusion between user and item features to establish their correlation is crucial for improving the predictive performance of the model. Finally, replacing the Hadamard product prediction with the AFM-based prediction module enhanced the model's accuracy by 27.09%, indicating the imperative of using AFM for low-order and high-order deep-level interaction calculations in fusing features.

Table 7
Comparison of RMSE values for review, summary, and generative summary.

Dataset	Review	Summary	Generative summary	GS Improve vs. review	GS Improve vs. summary
Automobile	0.5703	0.5068	0.5120	10.2%	−1.0%
Musical_Instrument	0.4921	0.4776	0.4803	2.4%	−0.6%
Yelp2020	0.9133	–	0.8961	1.9%	–

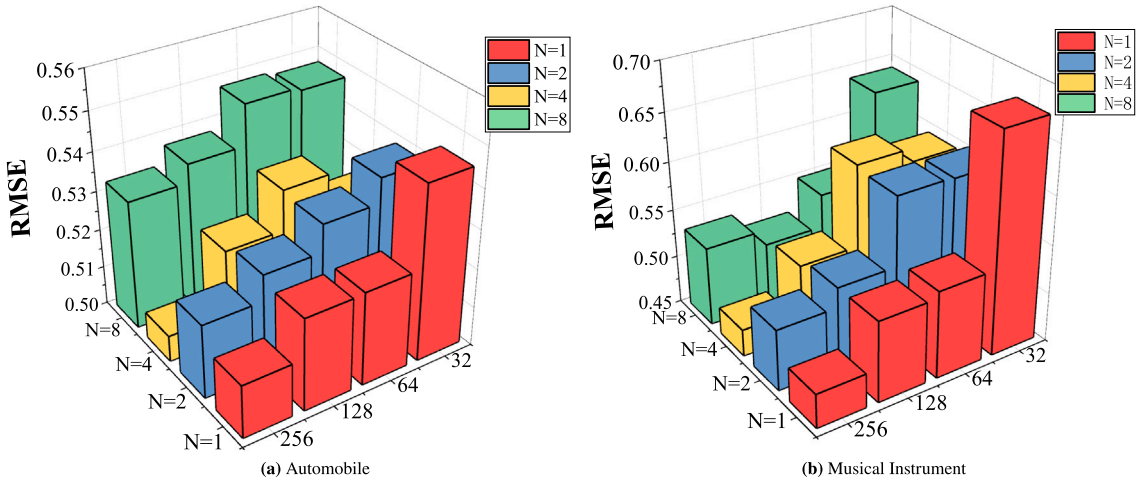

Fig. 4. Parameters analysis of the number of attention heads N and the feature dimension D on (a) Automobile, (b) Musical_Instrument.

4.7.2. Generative summary ablation

In order to investigate the impact of Generative Summary(GS) on model performance, we utilized "sumy" to extract GS from the reviews of Automobile, Musical_Instrument, and Yelp2020 and conducted a comparative analysis on MRSR with the original review and provided summaries for ablation. The experimental results are presented in Table 7.

The experiments indicate that GS positively affects the model's predictive capability compared to the original review, but it slightly lags behind the provided summary. We believe this could be due to GS extracting concise critical information from reviews. However, the clarity and brevity of the information are not as pronounced as those provided in the summaries. The experiments also show that the model's predictive performance is influenced slightly by the quality of the generative summary. Therefore, we believe that in situations where datasets lack summaries while requiring more concise textual data than reviews, particularly when the model's predictive results are not excessively sensitive, generative summaries can serve as a viable compromise.

4.8. Hyperparameter experiments

To figure out the appropriate hyperparameters, this paper conducted hyperparameter experiments by adjusting the size D of MRSR feature dimensions and the number N of attention heads separately (using the Automobile and Musical_Instrument dataset experiment results as an example).

The size D of MRSR feature dimensions represents the dimension of user and item feature vectors, which controls the size of the entire representation space. To ensure both accuracy and efficiency, D is set to [32, 64, 128, 256]. Meanwhile, the number of attention heads N, which is set to [1, 2, 4, 8], represents the number of low-dimensional spaces in which features are mapped when calculating attention. When D is determined, the number of features in each low-dimensional space can be calculated as F ($F = D \div N$). The different parameters of RMSE scores of D and N are drawn in Fig. 4. When $N = 4$ and $D = 256$, MRSR performs best on both datasets. In addition, it is found that the feature dimension has a significant impact on accuracy. As the dimension increases from [32,64,128,256], the model's prediction ability generally shows an upward trend, proving that the higher the feature dimension, the stronger the model's ability to represent complex information. However, correspondingly, the training time and memory usage also increase, which cannot guarantee the efficiency of the model. In addition, the experiment also found that the higher the dimension, the less prone the model is to overfitting. Small dimensions such as 32 and 64 cannot match the complexity of models such as Transformer, resulting in gradient explosion.

The essence of multi-head attention is the parallel processing of multiple self-attentions. As shown in Fig. 4, the prediction results are not significantly affected by the number of attention heads N. Therefore, we conducted a new experiment to explore the impact of the feature dimension F of single-head attention on prediction accuracy, displayed in Fig. 5. The results show that when

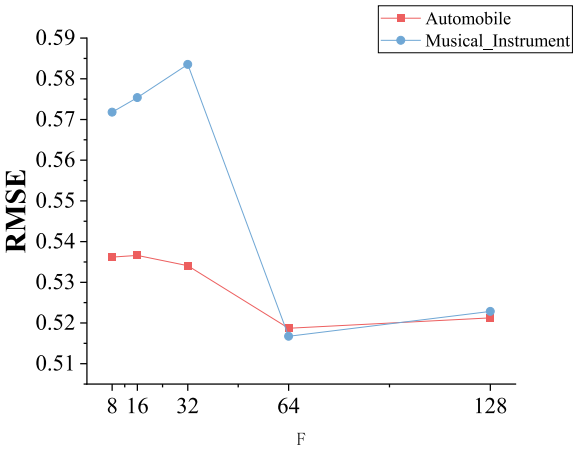

Fig. 5. The effect of the single-headed attention dimension F on Automobile and Musical_Instrument.

$F = 64$, the model achieves the peak performance, indicating that a single-head attention feature dimension of 64 can ensure that each low-dimensional attention space can sufficiently represent feature information, while also avoiding complex calculations and negative impacts on results caused by high number of dimensions.

Accordingly, the optimal model parameters are as follows: the dimension of the MRSR feature is 256. To ensure that the dimension of each attention space is 64, the number of attention heads in the multi-head attention mechanism is set to 4. This parameter setting ensures both the efficiency and accuracy of the model prediction.

5. Further discussions and implications

5.1. Results analysis

Our approach aims to capture the most salient features in summaries that reflect user preferences and item attributes through a multi-relational modeling framework. This is intended to mitigate the potential negative impact of irrelevant or noisy details present in comments, thereby reducing their influence on recommendations. In the comprehensive comparison, MRSR demonstrates a better ability to represent user preferences and item attributes, leading to more precise recommendations. It outperforms baseline methods across all datasets in terms of RMSE metrics. Another pivotal experiment involves the contrast between Summary-based and Review-based recommendations. This clearly demonstrates that our proposed Summary-based recommendation approach not only enhances the recommendation accuracy of most baseline methods but also significantly amplifies the model's recommendation efficiency, thereby reducing training time. Based on the results of incremental module-based ablation experiments, both intra-sentence relationship modeling and inter-sentence relationship modeling, as well as the fusion of user and item features and the prediction based on AFM, all demonstrated a positive impact on the outcomes. The generative summary ablation experiment indicates that generative summaries outperform reviews but fall short of standard summaries. Therefore, generative summaries can serve as an alternative method in scenarios where summary data is missing and there is no sensitivity to training time and accuracy. Finally, by fine-tuning the hyperparameters, MRSR achieved its optimal recommendation performance.

Through comparison with the baseline model, the strengths of our work are primarily manifested in the following aspects: (1) Introducing summaries as data sources and leveraging their conciseness and clarity to reduce the complexity of model representation. This improvement is evident in both accuracy and inference time. (2) By employing a multi-relational modeling approach, we extracted features of different granularity from the summaries, which enriched the feature representation of user preferences and item attributes.(3) Our approach also emphasized the correlation between user and item features. By conducting both shallow and deep-level fusion of these features, we further enhanced the predictive accuracy of the model. During our research, we have also recognized certain limitations of our approach: (1) We employed a large Transformer model for representation, which, compared to baseline models, entails more extensive training parameters and higher computational costs; (2) In some datasets, Summary data is lacking, necessitating the introduction of additional NLP models to summarize and extract information from Reviews.

5.2. Theoretical and practical implications

From a theoretical perspective, the innovative essence of MRSR lies in its introduction of summary-based recommendations. Initially, previous scholars incorporated reviews into recommendation systems to capture comprehensive user preferences and item attributes, effectively mitigating data sparsity. However, in today's environment, various websites and applications often contain massive datasets. Consequently, we want to use summaries to represent users and items more concisely and explicitly. Furthermore, MRSR recognizes the information embedded in summaries and establishes a multi-relational modeling framework to characterize

users and items from two distinct dimensions: the relationships between words and the relationships between sentences. Finally, our focus extends to the potential impact of feature interactions between users and items. As a result, MRSR incorporates various feature fusion patterns. Shallow feature interactions aim to learn the association between users and items, while deep interactions involve the fusion of computation results from different orders of features.

From a practical standpoint, MRSR leverages textual data (belonging to explicit user behavior) to learn user information and achieve rating prediction for items. This prediction helps determine whether an item should be recommended to a user. However, it currently lacks the application of implicit user behaviors such as browsing and dwell time. Thus, our study mainly focused on how to employ succinct user comment summaries in the context of review-based rating recommendations to unearth user preferences and item attributes while establishing a meaningful connection between the two features. Furthermore, more precise modeling of user tendencies and item characteristics enhances accuracy in rating predictions and recommendations while leveraging shorter computation times and reducing computational costs.

6. Conclusion

This paper proposes a Joint Inter-Word And Inter-Sentence Multi-Relation Summary-based Recommendation Model. We discovered the effectiveness of the simplicity of summary and its simplified model representation learning capability as a source of data for user and item representation. We also designed to use inter-word and inter-sentence multi-relation modeling: The Inter-Word module uses Transformer and pooling layers for word-level relation modeling, constructing the most essential word-level features in each summary; the Inter-Sentence module uses self-attention layers for sentence-level relation modeling, learning the importance weight of different summary sentences for predicting ratings. Finally, the fusion and prediction module based on MHA and AFM is used to achieve shallow fusion and deep interaction between user features and item features, and perform high-order non-linear calculations based on attention mechanism. Experimental results show that MRSR has significantly improved prediction accuracy compared to existing state-of-the-art models on five publicly available datasets from Amazon.

In future work, we will upgrade the encoding of initial data by introducing BERT pre-training models or domain knowledge models and explore more effective ways of generating summaries to create summaries datasets with higher quality. This may improve the model's ability to extract fine-grained features, better comprehend word expressions and associations within comment summaries, and ultimately improve recommendation performance for summary-based recommender systems.

CRediT authorship contribution statement

Duantengchuan Li: Conceptualization, Methodology, Software, Supervision, Writing – original draft, Writing – review & editing. **Ceyu Deng:** Software, Writing – original draft, Writing – review & editing. **Xiaoguang Wang:** Methodology, Supervision, Writing – review & editing. **Chao Zheng:** Data curation, Software, Writing – review & editing. **Jing Wang:** Conceptualization, Supervision, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62032016, 62207011), Key Research and Development Program of Hubei Province (No. 2021BAA031), Major Project of the National Social Science Fund, China (No. 21&ZD334), and the Open Research Fund of Intellectual Computing Laboratory for Cultural Heritage of Wuhan University, China (No. 2023ICLCH008, 2023ICLCH009).

References

Bao, Y., Fang, H., & Zhang, J. (2014). TopicMF: Simultaneously exploiting ratings and reviews for recommendation. In *Proceedings of the 28th conference on artificial intelligence, vol. 28, no. 1* (pp. 2–8).

Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: The state of the art. *User Modeling and User-Adapted Interaction*, *25*(2), 99–154.

Chen, H., Li, X., Zhou, K., Hu, X., Yeh, C. C. M., Zheng, Y., & Yang, H. (2022). TinyKG: Memory-efficient training framework for knowledge graph neural recommender systems. In *Proceedings of the 16th ACM conference on recommender systems* (pp. 257–267).

Chen, J., Xin, X., Liang, X., He, X., & Liu, J. (2023). GDSRec: Graph-based decentralized collaborative filtering for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *35*(5), 4813–4824.

Chen, C., Zhang, M., Liu, Y., & Ma, S. (2018). Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 world wide web conference* (pp. 1583–1592).

Cheng, H. T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., & Shah, H. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems* (pp. 7–10).

Deng, Y., Zhang, W., Xu, W., Lei, W., Chua, T.-S., & Lam, W. (2023). A unified multi-task learning framework for multi-goal conversational recommender systems. *ACM Transactions on Information Systems*, *41*(3).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, Volume 1 (Long and Short Papers)* (pp. 4171–4186).

Guo, H., TANG, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: A factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th international joint conference on artificial intelligence* (pp. 1725–1731).

Hu, H. X., Tang, B., Zhang, Y., & Wang, W. (2020). Vehicular Ad Hoc network representation learning for recommendations in internet of things. *IEEE Transactions on Industrial Informatics*, *16*(4), 2583–2591.

Huang, F. (2021). Personalized marketing recommendation system of new media short video based on deep neural network data fusion. *Journal of Sensors*, *2021*, Article 3638071.

Kanwal, S., Nawaz, S., Malik, M. K., & Nawaz, Z. (2021). A review of text-based recommendation systems. *IEEE Access*, *9*, 31638–31661.

Kim, D., Park, C., Oh, J., & Yu, H. (2017). Deep hybrid recommender systems via exploiting document context and statistics of items. *Information Sciences*, 72–87.

Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 426–434).

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Li, D., Xia, T., Wang, J., Shi, F., Zhang, Q., Li, B., & Xiong, Y. (2024). SDFormer: A shallow-to-deep feature interaction for knowledge graph embedding. *Knowledge-Based Systems*, *284*, Article 111253.

Li, S., Xie, R., Zhu, Y., Ao, X., Zhuang, F., & He, Q. (2022). User-centric conversational recommendation with multi-aspect user modeling. (pp. 223–233).

Li, Z., Zhang, Q., Zhu, F., Li, D., Zheng, C., & Zhang, Y. (2023). Knowledge graph representation learning with simplifying hierarchical feature propagation. *Information Processing & Management*, *60*(4), Article 103348.

Li, M., Zhao, X., Lyu, C., Zhao, M., Wu, R., & Guo, R. (2022). MLP4rec: A pure MLP architecture for sequential recommendations. In *Proceedings of the thirty-first international joint conference on artificial intelligence* (pp. 2138–2144).

Ling, G., Lyu, M. R., & King, I. (2014). Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM conference on recommender systems* (pp. 105–112).

Liu, D., Li, J., Du, B., Chang, J., & Gao, R. (2019). DAML: Dual attention mutual learning between ratings and reviews for item recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & Data mining* (pp. 344–352).

Liu, D., Li, J., Wu, J., Du, B., Chang, J., & Li, X. (2022). Interest evolution-driven gated neighborhood aggregation representation for dynamic recommendation in e-commerce. *Information Processing & Management*, *59*(4), Article 102982.

Liu, W., Zheng, X., Hu, M., & Chen, C. (2022). Collaborative filtering with attribution alignment for review-based non-overlapped cross domain recommendation. In *Proceedings of the ACM web conference 2022* (pp. 1181–1190).

Long, X., Huang, C., Xu, Y., Xu, H., Dai, P., Xia, L., & Bo, L. (2021). Social recommendation with self-supervised metagraph informax network. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 1160–1169).

Lu, Y., Dong, R., & Smyth, B. (2018). Coevolutionary recommendation model: Mutual learning between ratings and reviews. In *Proceedings of the 2018 world wide web conference* (pp. 773–782).

Luo, S., Lu, X., Wu, J., & Yuan, J. (2021). Review-aware neural recommendation with cross-modality mutual attention. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 3293–3297).

McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on recommender systems* (pp. 165–172).

Pan, S., Li, D., Gu, H., Lu, T., Luo, X., & Gu, N. (2022). Accurate and explainable recommendation via review rationalization. In *Proceedings of the ACM web conference 2022* (pp. 3092–3101).

Rendle, S. (2010). Factorization machines. In *Proceedings of the 2010 IEEE international conference on data mining* (pp. 995–1000).

Salakhutdinov, R., & Mnih, A. (2007). Probabilistic matrix factorization. In *Proceedings of the 20th international conference on neural information processing systems* (pp. 1257–1264).

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, *2009*, Article 421425.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). BERT4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1441–1450).

Tang, X., Liu, Y., He, X., Wang, S., & Shah, N. (2022). Friend story ranking with edge-contextual local graph convolutions. In *Proceedings of the fifteenth ACM international conference on web search and data mining* (pp. 1007–1015).

Tay, Y., Luu, A. T., & Hui, S. C. (2018). Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & Data mining* (pp. 2309–2318).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 6000–6010).

Wang, R., Fu, B., Fu, G., & Wang, M. (2017). Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17* (pp. 1–7).

Wang, J., Zhang, Q., Shi, F., Li, D., Cai, Y., Wang, J., Li, B., Wang, X., Zhang, Z., & Zheng, C. (2023). Knowledge graph embedding model with attention-based high-low level features interaction convolutional network. *Information Processing & Management*, *60*(4), Article 103350.

Wang, S., Zhang, P., Wang, H., Yu, H., & Zhang, F. (2022). Detecting shilling groups in online recommender systems based on graph convolutional network. *Information Processing & Management*, *59*(5), Article 103031.

Wang, S., Zhao, A., Lai, C., Zhang, Q., Li, D., Gao, Y., Dong, L., & Wang, X. (2023). GCANet: Geometry cues-aware facial expression recognition based on graph convolutional networks. *Journal of King Saud University - Computer and Information Sciences*, *35*(7), Article 101605.

Wu, B., He, X., Chen, Y., Nie, L., Zheng, K., & Ye, Y. (2022). Modeling product's visual and functional characteristics for recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, *34*(3), 1330–1343.

Wu, L., He, X., Wang, X., Zhang, K., & Wang, M. (2023). A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *35*(5), 4425–4445.

Wu, B., He, X., Wu, L., Zhang, X., & Ye, Y. (2023). Graph-augmented co-attention model for socio-sequential recommendation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *53*(7), 4039–4051.

Wu, F., Li, D., Lin, K., & Zhang, H. (2021). Efficient nodes representation learning with residual feature propagation. In *Advances in knowledge discovery and data mining* (pp. 156–167).

Wu, L., Quan, C., Li, C., Wang, Q., Zheng, B., & Luo, X. (2019). A context-aware user-item representation learning for item recommendation. *Transactions on Information Systems*, 1–29.

Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., & Chua, T. S. (2017). Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *Proceedings of the 26th international joint conference on artificial intelligence* (pp. 3119–3125).

Xiong, K., Ye, W., Chen, X., Zhang, Y., Zhao, W. X., Hu, B., Zhang, Z., & Zhou, J. (2021). Counterfactual review-based recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 2231–2240).

Yin, Y., Li, Y., Gao, H., Liang, T., & Pan, Q. (2023). FGC: GCN-based federated learning approach for trust industrial service recommendation. *IEEE Transactions on Industrial Informatics*, *19*(3), 3240–3250.

Zhang, W., Du, T., & Wang, J. (2016). Deep learning over multi-field categorical data. In N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio, C. Hauff, & G. Silvello (Eds.), *Proceedings of the 38th european conference on IR research* (pp. 45–57).

Zhang, Y., & Hara, T. (2023). Explainable integration of social media background in a dynamic neural recommender 17 (3).

Zhang, Y., & Yamasaki, T. (2021). Style-aware image recommendation for social media marketing. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 3106–3114). Association for Computing Machinery.

Zheng, L., Noroozi, V., & Yu, P. S. (2017). Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the 10th ACM international conference on web search and data mining* (pp. 425–434).

Zhu, F., Chen, Z., Zhang, F., Lou, J., Wen, H., Liu, S., Rao, Q., Yuan, T., Ni, S., Hu, J., Sun, F., & Lu, Q. (2022). SASNet: Stage-aware sequential matching for online travel recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 3725–3734).