electronics

MDPI

*Article*

# Attentive Review Semantics-Aware Recommendation Model for Rating Prediction

**Jihyeon Kim [1,†], Xinzhe Li [1,†], Li Jin [1], Qinglong Li [1] and Jaekyeong Kim [1,2,*]**

[1] Department of Big Data Analytics, Kyung Hee University, 26, Kyungheedae-ro, Dongdaemun-gu, Seoul 02447, Republic of Korea
[2] School of Management, Kyung Hee University, 26, Kyungheedae-ro, Dongdaemun-gu, Seoul 02447, Republic of Korea
* Correspondence: jaek@khu.ac.kr; Tel.: +82-2-961-9355
† These authors contributed equally to this study.

**Abstract:** Online reviews are crucial when building a recommendation model because they contain the specific and rich preferences of users related to different aspects of a particular item. Incorporating these online reviews into the recommendation model mitigates the data sparsity issue to some extent and contributes to better recommendation performance. Despite this success, review-based recommender systems have the limitation that they do not fully consider the relevance of the review text to the target item. Specifically, the review text should reflect the user's detailed opinion about the target item to extract detailed preference information. Meanwhile, the review content must be directly related to the target item to extract the customer's specific preferences for the item. However, previous studies have overlooked both of these aspects. Therefore, it is necessary to build a recommendation model that considers the relevance of the review content to the target item. To address this issue, this study proposes a novel recommendation model that accurately estimates users' preferences by carefully considering the relevance of the review content to the items. The proposed model effectively extracts feature representations from the text using bidirectional encoder representations from a transformer and obtains fused features by considering the importance of features through the attention mechanism. To evaluate the performance of the model, we used a publicly accessible dataset of reviews from Amazon.com and compared it to various baseline models. The experimental results demonstrated better recommendation performance of the proposed model compared to other baseline models.

**Keywords:** recommender system; deep learning; attention mechanism; natural language process; auxiliary information

check for updates

**Citation:** Kim, J.; Li, X.; Jin, L.; Li, Q.; Kim, J. Attentive Review Semantics-Aware Recommendation Model for Rating Prediction. *Electronics* **2024**, *13*, 2815. https://doi.org/10.3390/electronics13142815

Academic Editor: George Angelos Papadopoulos

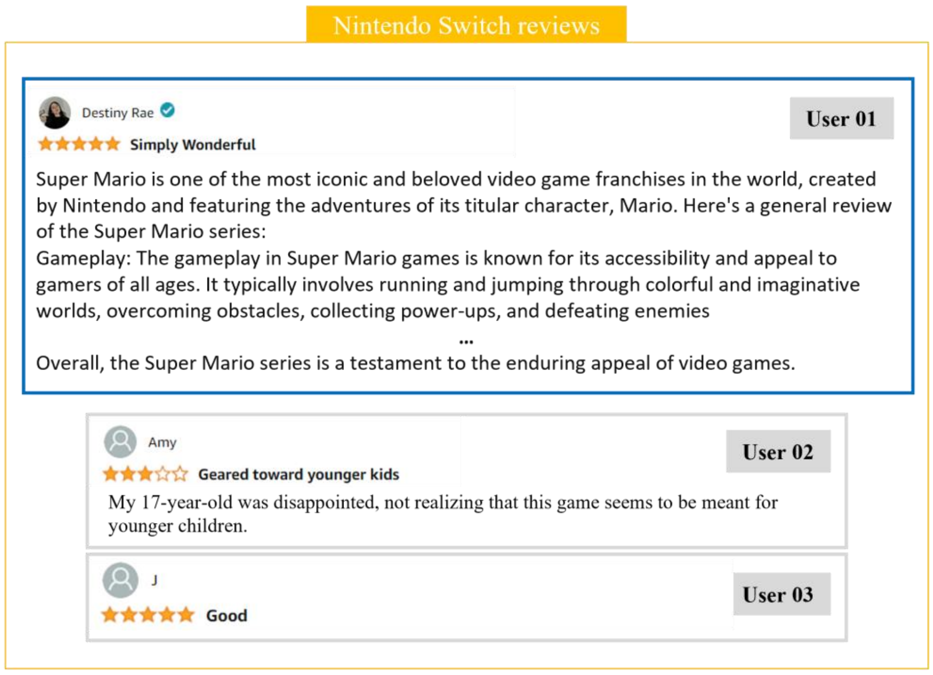
Received: 21 June 2024
Revised: 10 July 2024
Accepted: 15 July 2024
Published: 17 July 2024

## 1. Introduction

With the rapid development of information, communication technologies, and social media, the number of items and content accessible online has increased dramatically, and users are faced with the problem of information overload. This situation made it difficult for users to find items that match their needs and tastes and emphasized the need for a recommender system to solve this problem [1]. Recommender systems make it easier for users to find items that match their preferences while helping businesses maximize profits by offering items that match user preferences [2]. Collaborative filtering (CF) is the most popular recommendation model that has been widely used in previous recommender system studies and has excellent recommendation performance. The basic premise of CF-based recommender systems is that users with similar past behavioral histories (e.g., ratings, clicks, and visits) have similar preferences [3]. These CF models leverage users' past behavioral history to analyze similarities between users or items to predict their preferences for items [4].

CF-based recommender systems often rely on a user's past behavioral history as their only source of information, which limits their ability to fully understand a user's preferences and behavioral motivations. Furthermore, users often do not rate most items, which leads to the data sparsity issue [5]. Recommender system researchers have found that online reviews can be a useful source for solving the data sparsity problem. Online reviews contain detailed and rich opinions left by users about a particular item, which can help us better understand and predict various aspects of users' preferences [6]. Recent studies have explored different approaches to incorporate feature representations obtained by analyzing review text into recommender systems. Most approaches generally adopt the same paradigm of extracting feature representations from review text and integrating them with a matrix factorization (MF) model or directly into a regression model to predict a user's overall rating [7–9].

Previous studies on review-based recommender systems have made significant contributions to addressing data sparsity and improving recommendation performance, but they are limited by not fully considering the relevance of the review text to the current item. First, the review text should contain the user's detailed opinion about the target item. The review text includes user opinions on various aspects of the item, such as price, design, and brand, which can be used to estimate the user's specific preferences for a particular item [10–12]. For example, Figure 1 shows four users' perspectives on Nintendo Switch products. Here, User 1 describes in detail their satisfaction with the game content of the Nintendo Switch. On the other hand, User 2 describes their satisfaction when giving the product as a gift rather than their direct preference for the product. This shows that different users evaluate the same item from various perspectives. Conversely, User 3 did not provide specific opinions on the item. In summary, the opinions provided by Users 1 and 2 can reveal specific preferences for the target item, while the opinion provided by User 3 is relatively limited in extracting detailed preference information. Second, the content of the review should be directly related to the target item. Some reviews contain various opinions from users but may not be relevant to the target item. For example, in Figure 1, User 4 purchased a Nintendo Switch but wrote a review about a Switch case. Likewise, sometimes, users may accidentally leave reviews on other items or write something unrelated. However, it is clear that these reviews are not opinions about the target item, which limits the ability to estimate accurate user preferences. In this context, it is necessary to build a recommendation model that considers the relevance of the review content to the target item.

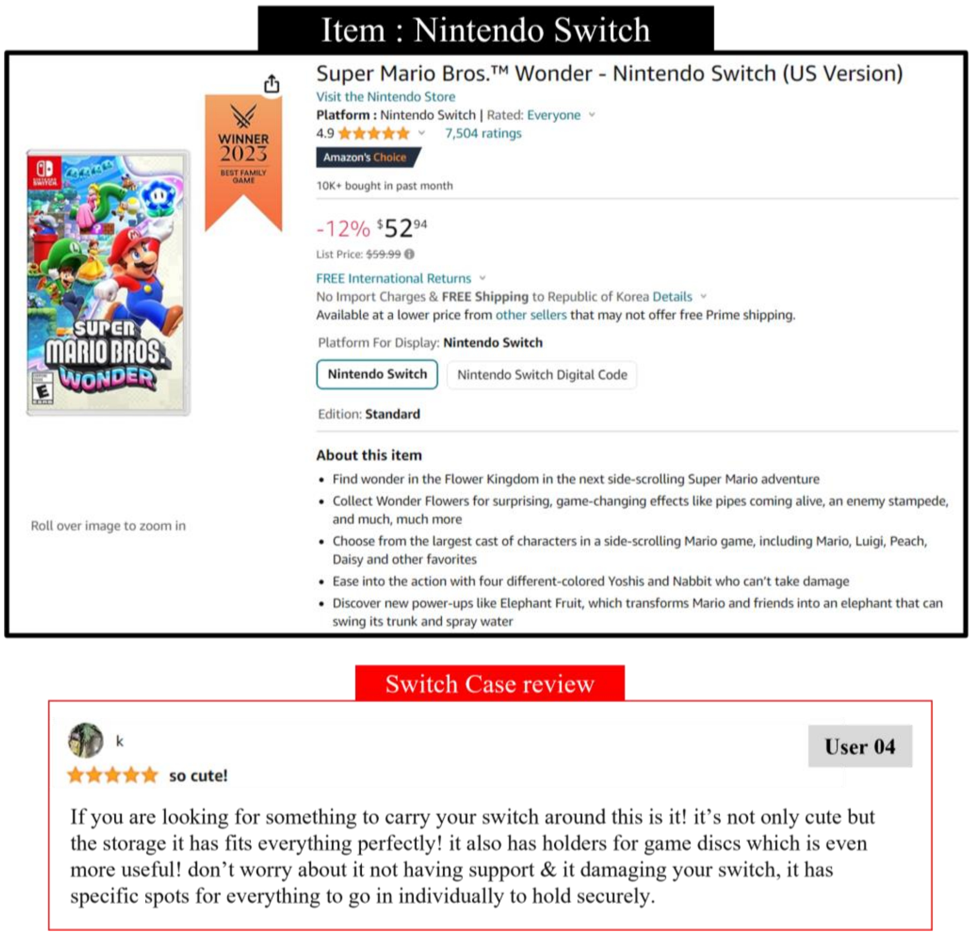

Nintendo Switch reviews

Destiny Rae
★★★★★ Simply Wonderful
User 01
Super Mario is one of the most iconic and beloved video game franchises in the world, created by Nintendo and featuring the adventures of its titular character, Mario. Here's a general review of the Super Mario series:
Gameplay: The gameplay in Super Mario games is known for its accessibility and appeal to gamers of all ages. It typically involves running and jumping through colorful and imaginative worlds, overcoming obstacles, collecting power-ups, and defeating enemies
...
Overall, the Super Mario series is a testament to the enduring appeal of video games.

Amy
★★★☆☆ Geared toward younger kids
User 02
My 17-year-old was disappointed, not realizing that this game seems to be meant for younger children.

J
★★★★★ Good
User 03

**Figure 1.** *Cont.*


Item : Nintendo Switch

Super Mario Bros.™ Wonder - Nintendo Switch (US Version)
Visit the Nintendo Store
**Platform :** Nintendo Switch | Rated: Everyone
4.9 ★★★★★ 7,504 ratings
Amazon's Choice
10K+ bought in past month

-12% $52.94
List Price: ~~$59.99~~
FREE International Returns
No Import Charges & **FREE Shipping** to Republic of Korea Details
Available at a lower price from other sellers that may not offer free Prime shipping.
Platform For Display: **Nintendo Switch**
Nintendo Switch | Nintendo Switch Digital Code
Edition: **Standard**

**About this item**
- Find wonder in the Flower Kingdom in the next side-scrolling Super Mario adventure
- Collect Wonder Flowers for surprising, game-changing effects like pipes coming alive, an enemy stampede, and much, much more
- Choose from the largest cast of characters in a side-scrolling Mario game, including Mario, Luigi, Peach, Daisy and other favorites
- Ease into the action with four different-colored Yoshis and Nabbit who can't take damage
- Discover new power-ups like Elephant Fruit, which transforms Mario and friends into an elephant that can swing its trunk and spray water

Roll over image to zoom in

Switch Case review

k User 04
★★★★★ **so cute!**
If you are looking for something to carry your switch around this is it! it's not only cute but the storage it has fits everything perfectly! it also has holders for game discs which is even more useful! don't worry about it not having support & it damaging your switch, it has specific spots for everything to go in individually to hold securely.

**Figure 1.** Examples of user reviews.

This study proposes a new recommendation model called ARSRec (Attentive review semantics-aware recommendation), which accurately estimates users' preferences by considering the relevance between review content and item information. The ARSRec uses bidirectional encoder representations from transformers (BERT) to capture detailed and rich expressions in review texts. It integrates general item information provided by sellers with item attribute representations extracted from large-scale review texts to extract comprehensive attributes of the item. By introducing a self-attention mechanism and a co-attention mechanism, it considers the complementarity between objective and subjective information. Ultimately, it predicts user preferences by integrating the relevance between the review texts left by users for specific items and the item attributes extracted in the previous step. We compared the recommendation performance of different baseline models using a publicly accessible dataset of Amazon.com reviews to evaluate the performance of the proposed model. The experimental results demonstrated the improved recommendation performance of the proposed recommendation model compared to other baseline models. The key contributions of this study are as follows:

- This study proposed a new recommendation model that captures the relevance between the review content and the target item to estimate the user's preference for the item. The proposed model combines large-scale item reviews with general item information to create a unified characteristic representation that can effectively capture the comprehensive characteristics of an item.
- The proposed model uses the self-attention mechanism to capture the correlation between item characteristics and the co-attention mechanism to capture the complementarity between these characteristics to obtain a more comprehensive and fused representation of item characteristics.

- The performance of the proposed model was fairly evaluated by comparing it with various baseline models. For this purpose, we utilized a real dataset from Amazon.com, and the experimental results showed that the proposed model outperforms existing models in making recommendations.

This paper is organized as follows. Related studies on review-based recommender systems and deep learning techniques for recommender systems are presented in Section 2. The problems in this study are addressed in Section 3. A detailed description of our proposed model is introduced in Section 4, and then the datasets and experiment setting are described in Section 5. Finally, Section 6 summarizes the results, and Section 7 presents conclusions and future works of this study.

## 2. Related Works

### *2.1. Review-Based Recommender System*

With the development of information and communication technology, the number of Internet users and items has grown exponentially, and users are facing the problem of information overload; this has highlighted the importance of recommender systems that effectively reduce the cost of information search for users and help companies maximize profits. Initially, many CF-based recommender systems were proposed that leverage similarities between users and items based on past behavioral history to predict user preferences [3]. However, there are only a limited number of items that users can purchase or rate, which creates a data sparsity problem [13]. Many researchers have proposed recommender systems that utilize review text as an effective way to solve this problem. For example, Zheng, Noroozi, and Yu [7] proposed a model that predicts ratings by embedding a set of reviews for each user and item using a convolutional neural network (CNN) technique. The study found that CNN improved performance by incorporating different local feature representations of users and items in reviews into recommendations. Cheng et al. [14] proposed a recommendation model that uses topic modeling to extract characteristics of users and items from reviews to predict ratings. As shown in these examples, various studies have been proposed to improve performance by extracting user and item attribute information from large review texts. However, while these studies can analyze users' preferences in general, they are limited in their ability to analyze detailed preferences for specific items. In response, studies proposed to predict preferences by specifically analyzing users' preferences at the individual review level. For example, Cao et al. [15] built a recommendation model that applied CNN techniques to user reviews of specific items. They could analyze more specific user preferences by leveraging large review texts while demonstrating computational efficiency. Liu et al. [16] embedded user reviews in two ways, combined the results to extract feature representations, and proposed a model that incorporated them into recommendations. The results demonstrated that embedding enrichment for individual reviews was effective in extracting feature representations, which in turn improved performance.

Although existing studies on review-based recommender systems have contributed to effectively addressing the data sparsity problem, there are still areas for improvement. Most existing studies follow the approach of extracting user and item attribute representations from review text and incorporating them into MF models or using regression analysis to predict ratings. However, these studies follow a common paradigm that assumes that item-specific characteristics are always implicit in the review text. However, some reviews contain information that is completely irrelevant to the item, or there are not enough user comments to extract accurate characterization information. Incorporating review text that is less relevant to an item into the recommendation model can limit the ability to understand a user's exact preferences and inhibit the ability to improve recommendation performance. Therefore, this study proposes a novel recommendation model that considers the relevance between review content and item information and incorporates the user's specific preference information into recommendations.

### *2.2. Deep Learning Techniques for Recommender Systems*

#### 2.2.1. Large Language Models (LLMs)

The development of social media has led to an exponential increase in textual data online as communication between online users has increased [17]. Various deep learning-based natural language processing (NLP) techniques have been developed to effectively process large amounts of textual information. Recently, various transformer-based large language models (LLMs) have been actively proposed to effectively utilize large amounts of textual data. In the field of recommender systems, these LLM techniques are also actively used for text embedding or explanation generation. Text embedding mainly aims to effectively extract feature representation vectors of users or items [18–20]. Models such as BERT and RoBERTa, which utilize the encoder part of the transformer, are mainly used for text embedding [16,21]. On the other hand, in the explanation generation field, LLM models are used mainly to explain the reasons for recommendations [22–24]. Techniques like GPT, which utilize the decoder part of the transformer, are used for this purpose.

This study aims to construct a recommendation model by effectively extracting user preference information and item feature information from user reviews and item information, respectively. Therefore, we used the BERT model, which is a representative LLM model suitable for effective text embedding [20,21,25]. Because BERT processes sentences bidirectionally, the outputs of BERT can reflect different representations for the same word depending on the context. Unlike recurrent neural network (RNN)-based models, it achieves greater computational efficiency by processing multiple sentences and words in parallel [26]. Many recommender system studies have utilized BERT techniques to extract various attribute information from review text and incorporate it into recommendations. For example, Qiu, Wu, Gao, and Fan [20] proposed a model to predict ratings by embedding user reviews as BERT and integrating them into user and item interaction vectors. Liu, Chen, and Chang [16] embedded the review texts with two techniques, BERT and robustly optimized BERT pre-training approach (RoBERTa), an applied model of BERT, and then combined them into a vector of user preference attributes to make recommendations. Both models applied the BERT technique to effectively extract attribute information from the reviews, which improved the data sparsity problem.

#### 2.2.2. Attention Mechanism

Recent studies in recommender systems have emphasized the need to extract various attribute information from reviews to address data sparsity. In particular, the self-attention mechanism, which allows for weighted output of what aspects of an item the user is paying attention to for personalized recommendations, is starting to draw attention. The self-attention mechanism primarily focuses on identifying the importance of different parts of a single sequence, thus enhancing the representation of user preferences by focusing on relevant aspects within the review texts. This method is advantageous for capturing intra-sequence relationships [27]. For example, Chen et al. [28] embedded a set of user and item reviews into a CNN to extract local features and then applied a self-attention mechanism to propose a model that utilized individual attribute weights for rating prediction. Cao, Zhang, and Wang [15] proposed a model that utilizes a self-attention mechanism and CNN to extract semantic features from review texts and reflects the weight of each feature in rating prediction. Both studies used self-attention mechanisms to achieve more personalized recommendations than previous studies.

Compared to the self-attention mechanism, the co-attention mechanism focuses on modeling the interactions between different sequences or modalities, such as review texts and item descriptions or images. This method captures inter-sequence relationships, allowing the model to leverage complementary information from multiple sources to improve recommendation accuracy. In this study, we use a co-attention mechanism to extract complementary item information from various contents. Item information can come from information provided by the seller in addition to the review text. In addition to recent reviews, attempts have been made to extract rich attribute information from various forms of data, such as item descriptions and images. Therefore, studies utilizing the co-attention mechanism are proposed to capture complementary relationships between various types of features. For example, Chin et al. [29] proposed a recommendation model that outputs aspects from the review text of each user and item and applies the co-attention mechanism to predict ratings using weights influenced by each other at the aspect level. Jang, Li, Lee, and Kim [5] proposed a model to predict ratings by fusing review text and various item attribute data with a co-attention mechanism. As shown in these studies, the co-attention mechanism calculates the interaction of different features and reflects the fused attribute information, which helps improve recommendation performance.

### *2.3. Research Gaps and Motivation*

In summary, previous studies empirically demonstrated that embedding review text using BERT techniques can effectively extract rich feature representations. We also found that by applying the self-attention mechanism to the extracted representation vectors, we can capture important characteristics and significantly improve the performance and computational efficiency of the model. In this context, this study applies the BERT technique to user review texts to effectively extract user opinions. In addition, a set of item reviews is generated by users, and item information is provided by merchants (e.g., item title and description) for a comprehensive view of item information. These two items of information emphasize important traits and obtain complementary trait representations through the self-attention mechanism and co-attention mechanism. The fused item attributes obtained from these two attention mechanisms are integrated into the recommendation model, and the relevance between reviews and item information is analyzed to accurately estimate user preferences. Previous studies did not consider the relevance between reviews and item information, assuming that all reviews contained correct user preference information and reflecting this in the recommendations. However, this study addresses this limitation by considering the relevance between product information and review content. As a result, the model is designed to minimize the influence of reviews containing irrelevant information or insufficient user preference data on the recommendations. This approach allows for more personalized and accurate recommendations compared to previous studies.

## **3. Problem Definition**

Figure 2 illustrates the overall architecture of the proposed model. The proposed model comprises five networks: user-item interaction, auxiliary information, attentive representation, user attentive preference, and prediction. The main purpose of review-based recommender systems is to extract user opinions from reviews to analyze purchase motivations and predict preferences more precisely. However, not all reviews include user comments, and some reviews contain information that is not relevant to the target item. Nevertheless, existing studies extract user characteristics from item and relevance reviews to inform recommendations [7,28]. This fragmentary approach prevents recommendations from reflecting accurate user preferences. To improve this limitation, this study proposes ARSRec, which considers the interaction between user reviews and item information to reflect users' accurate opinions in recommendations. To this end, we extract auxiliary information from a large set of item reviews written by users and item information provided by sellers. The extracted information is output as a fused item feature vector that reflects the importance of the attributes through the self-attention mechanism and co-attention mechanism. Meanwhile, user opinions about items are extracted from the individual reviews of users. We then learn the interactions between the extracted user opinions and the item feature vector to reflect the precise and granular opinions that users have about a particular item in the final rating prediction. $T$ represents a variety of information about users and items. The items in $T = (u, u_{r,i}, i, r_i, t, d, y)$ denote user identity, user review text of item, item identity, item review, title, description, and preference rating, respectively. The proposed model aims to learn the prediction model formalized as follows:

$$F(u, u_{r,i}, i, r_i, t, d; \theta) \rightarrow \hat{y}, \tag{1}$$

where $\theta$ and $\hat{y}$ are the bias and predicted preference ratings. The proposed model uses user identity, user review text of the item, item identity, item reviews, and item information (title, description) as input data. During the training process, the model is trained so that the predicted rating, $\hat{y}_{u,i}$ will get close to the actual rating $y_{u,i}$. After all training is complete, the model outputs the user's final predicted rating for the item.

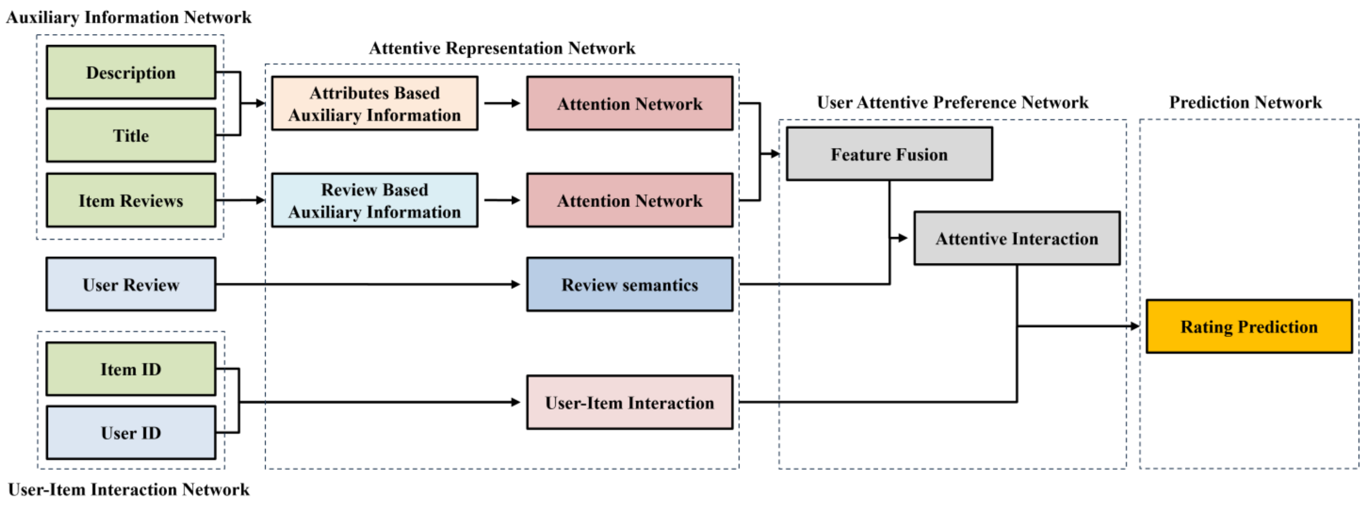

Auxiliary Information Network
Description
Title
Item Reviews
User Review
Item ID
User ID
User-Item Interaction Network
Attentive Representation Network
Attributes Based Auxiliary Information
Review Based Auxiliary Information
Attention Network
Attention Network
Review semantics
User-Item Interaction
User Attentive Preference Network
Feature Fusion
Attentive Interaction
Prediction Network
Rating Prediction

**Figure 2.** Architecture of the ARSRec model.

## 4. ARSRec Framework

In this section, we introduce a proposed model that effectively extracts user opinions from reviews that are relevant to the target item and incorporates them into recommendations. To this end, specific user opinions are effectively captured by performing interaction operations between fusion item feature vectors extracted from item reviews and item information and individual user reviews. The overall framework of the ARSRec is shown in Figure 3 and involves the following steps. First, the user-item interaction network converts unique user and item information into latent factor vectors and then analyzes complex interaction relationships through multi-layer perceptron (MLP). Second, the user attentive representation network extracts specific opinions about items from individual user reviews; this is accomplished by performing interaction operations between comprehensively fused item information and user reviews. Finally, the rating prediction network outputs the user's predicted rating for the item. A detailed description of each step is illustrated below.

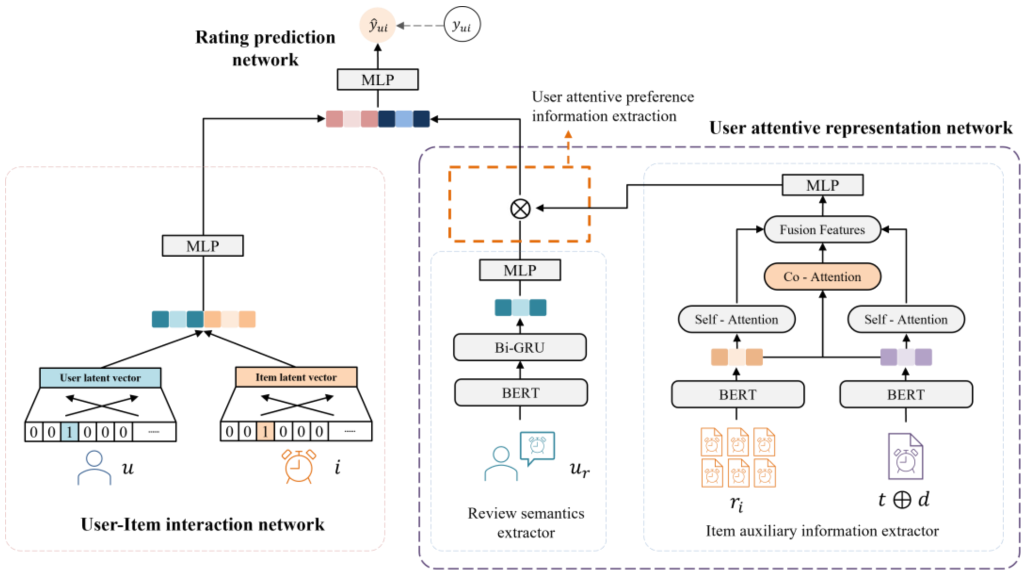

Rating prediction network
$\hat{y}_{ui}$
$y_{ui}$
MLP
User attentive preference information extraction
User attentive representation network
MLP
Fusion Features
Co - Attention
Self - Attention
Self - Attention
BERT
BERT
$r_i$
$t \oplus d$
Item auxiliary information extractor
MLP
Bi-GRU
BERT
$u_r$
Review semantics extractor
MLP
User latent vector
Item latent vector
$u$
$i$
User-Item interaction network

**Figure 3.** Framework of the ARSRec model.

### 4.1. *User-Item Interaction Network*

This network aims to extract latent factor vectors of users and items to identify interaction relationships. First, it is applied to the embedding layer to convert the sparse user and item vectors into dense vectors. The potential vectors of the user and item are computed as in Equation (2):

$$p_u = P^T v_u^U; q_i = Q^T v_i^I, \tag{2}$$

where $P \in \mathbb{R}^{m \times d}$ and $Q \in \mathbb{R}^{n \times d}$ are the user and item embedding matrices. Here, $d$ is the number of dimensions in the latent vector, and $m$ and $n$ denote the number of unique users and items, respectively. Furthermore, $v_u^U$ and $v_i^I$ are one-hot encodings of the user and item. Then, the output of $p_u$ and $q_i$ are combined, as shown in Equation (3), and used as input to an MLP to learn complex interaction relationships.

$$V_I = p_u \oplus q_u, \tag{3}$$

where $\oplus$ denotes the concatenate operation and $V_I$ is the result of the operation. This study adopts the concatenate operation to minimize the loss of user and item information.

$$\begin{gathered} V_I^1 = \sigma(W_I^1 V_I^0 + b_I^1), \\ \cdots \\ V_I^L = \sigma(W_I^L V_I^{L-1} + b_I^L), \end{gathered} \tag{4}$$

$V_I^L$ denotes the end-user-item interaction representation vector trained by the MLP in Equation (4). Here, $W^L$, $b^L$, and $\sigma^L$ are the weight, bias, and activation functions of the $L$th layer, respectively. These are rectified linear units (ReLUs), which are commonly used in deep learning techniques.

### 4.2. *User Attentive Representation Network*

#### 4.2.1. Review Semantics Extractor

To effectively extract semantics from text, we used the transformer-based pre-trained BERT model, which is the most popular in NLP, for text embedding; this allows for contextual word embeddings, and the [CLS] token, which contains important information for all tokens in a sentence, is used as the embedding vector [16]. The embedding of the BERT model is shown in Equation (5):

$$O_{BERT} = \text{BERT}(u_{r,i}), \tag{5}$$

where $O_{BERT}$ represents the value of the [CLS] token in the BERT embedding result; this is used as input to Bi-GRU, which effectively extracts textual features in both directions. The Gated Recurrent Unit (GRU) consists of a reset gate and an update gate, which effectively reflects the sequential information in the text. The Bi-GRU is a bi-directional computation technique, and the process is shown in Equation (6):

$$V_S = \left[ \overrightarrow{GRU_{(O_{BERT})}}, \overleftarrow{GRU_{(O_{BERT})}} \right], \tag{6}$$

where $V_S$ is the output vector of $O_{BERT}$ after passing through Bi-GRU; this is used as input to the MLP, and the process is shown in Equation (7):

$$\begin{gathered} V_S^1 = \sigma(W_S^1 V_S^0 + b_S^1), \\ \cdots \\ V_S^L = \sigma(W_S^L V_S^{L-1} + b_S^L), \end{gathered} \tag{7}$$

where $V_S^L$ is the semantics vector for end-user reviews. In addition, $W^L$, $b^L$, and $\sigma^L$ are the weight, bias, and activation functions of the $L$th layer, respectively. We used the ReLU, as shown in Equation (4).

### 4.2.2. Item Auxiliary Information Extractor

To comprehensively consider the auxiliary information of an item, this study utilizes a large set of user-generated item reviews and seller-provided item information. In this context, an item review refers to a set of reviews that users have left for a specific item. Regarding the item information, we used the title and description of the item, which are commonly used in the field of item value extraction studies [30,31].

First, we embed item reviews into BERT to extract subjective item information from the user's perspective, as shown in Equation (8). In this case, an item review represents a set of reviews that aggregate multiple reviews.

$$R_{BERT} = \text{BERT}(r_i), \tag{8}$$

where, $R_{BERT}$ is the subjective item information from the user's perspective, which is the output of the item review after the BERT embedding. The important attributes are then learned through the self-attention mechanism, and the process is shown in Equation (9):

$$\begin{aligned} &Q, K, V = X^Q W^Q, X^K W^K, X^V W^V, \\ &\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V, \end{aligned} \tag{9}$$

where $W^Q \in \mathbb{R}^{d_k \times d_p}$, $W^K \in \mathbb{R}^{d_k \times d_p}$, and $W^V \in \mathbb{R}^{d_v \times d_p}$ are the trainable weight matrices for $Q \in \mathbb{R}^{q \times d_k}$, $K \in \mathbb{R}^{k \times d_k}$, and $V \in \mathbb{R}^{k \times d_v}$, respectively. $\sqrt{d_k}$ is the number of dimensions of the key vector, rooted at the number of dimensions. In the self-attention mechanism, the inputs $X^Q$, $X^K$, and $X^V$, which are multiplied with the weight matrix, are all the same, and the item review vector output from Equation (8) is inputted into Equation (10):

$$R_{S-Att} = \text{Attention}(R_{BERT} W^Q, R_{BERT} W^K, R_{BERT} W^V), \tag{10}$$

where $R_{S-Att}$ is the attention score value calculated by the self-attention mechanism and reflects the important attributes of the item.

Next, we analyzed the item title and description as text data to extract objective item information provided by the seller, and the process is shown in Equation (11):

$$A_{BERT} = \text{BERT}(A), \tag{11}$$

where $A$ is an aggregation of the item title and description to minimize information loss. $A_{BERT}$ is the result of BERT embedding. It then goes through the self-attention mechanism in Equation (12) and is the same as before:

$$A_{S-Att} = \text{Attention}(A_{BERT}W^Q, A_{BERT}W^K, A_{BERT}W^V), \tag{12}$$

where $A_{S-Att}$ is the attention score value, which reflects the value of important attributes of the objective item information. This study leverages complementary and comprehensive item information by ensuring that the correlation between user-generated item reviews and seller-provided item information is learned. To perform this, we apply the co-attention mechanism, which is shown in Equation (13):

$$\begin{aligned} R_{Co-Att} &= \text{Attention}(R_{BERT}W^Q, A_{BERT}W^K, A_{BERT}W^V), \\ A_{Co-Att} &= \text{Attention}(A_{BERT}W^Q, R_{BERT}W^K, R_{BERT}W^V), \end{aligned} \tag{13}$$

where $R_{Co-Att}$ and $A_{Co-Att}$ represent the attention score from a complementary perspective. Specifically, since the important attributes of items that sellers pay attention to may be different from the important attributes of items that users pay attention to, the co-attention mechanism allows for information complementation between the two data attributes. Combine the two extracted attention scores to output a comprehensive item attribute vector as shown in Equation (14):

$$C = R_{Co-Att} \otimes A_{Co-Att}, \tag{14}$$

where $\otimes$ is the element-wise product operation, and $C$ is the multiply of the two attitude scores. We used inner product computation to account for the complex correlation between the two data attributes. The final fused item attribute vector is then subjected to Equation (15):

$$I_{Fuse} = R_{S-Att} \oplus A_{S-Att} \oplus C, \tag{15}$$

where $I_{Fuse}$ is an item attribute vector reflecting both the self-attention score and co-attention score for each of the item reviews and information. To minimize information loss, we used the concatenate operation, and the result is the input to the MLP, as follows:

$$\begin{aligned} I^1_{Fuse} &= \sigma(W^1_{Fuse} I^0_{Fuse} + b^1_{Fuse}), \\ &\cdots \\ I^L_{Fuse} &= \sigma(W^L_{Fuse} I^{L-1}_{Fuse} + b^L_{Fuse}), \end{aligned} \tag{16}$$

where $I^L_{Fuse}$ is the final fusion item feature vector, and $W^L$, $b^L$, and $\sigma^L$ are the weight, bias, and activation function ReLU of the first layer, respectively.

#### 4.2.3. User Attentive Preference Information Extraction

To compute the relevance between the semantics vector $V^L_S$ of the individual user reviews output from Equation (7) and $I^L_{Fuse}$ in Equation (16) is used:

$$V_A = V^L_S \otimes I^L_{Fuse}, \tag{17}$$

where $V_A$ is the final user attentive preference information vector output from the relevance computation between individual user reviews and the fused item attribute information.

### *4.3. Rating Prediction Network*

The purpose of this network is to predict the final rating by utilizing the user-item interaction and user attentive preference information extracted from the previous steps. To this end, we first combine $V^L_I$ and $V_A$, which are extracted from the previous user-item interaction network and the user attentive representation network. This process is shown in Equation (18):

$$V_P = V^L_I \oplus V_A, \tag{18}$$

where $V_p$ is the result of concatenating two vectors; this is fed into the final MLP layer to predict the rating, which is shown in Equation (19):

$$\begin{gathered} V_P^1 = \sigma(W_P^1 V_P^0 + b_P^1), \\ \cdots \\ \hat{y}_{u,i} = V_P^L = \sigma(W_P^L V_P^{L-1} + b_P^L), \end{gathered} \tag{19}$$

where $V_P^L$ is the user preference that the proposed model finally predicted.

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_{u,i} - \hat{y}_{u,i})^2, \tag{20}$$

In this study, we adopted mean squared error ($MSE$) as the loss function for performance measurement. $MSE$ is the sum of the squared errors divided by the number of data, where $u$ is the user, $i$ is the item, $\hat{y}_{u,i}$ is the predicted rating, $y_{u,i}$ is the actual rating, and $N$ is the number of training data. Consequently, we optimized the model by minimizing the output of the $MSE$ and effectively trained it using backpropagation. We used adaptive moment estimation (Adam) as the optimization method.

## 5. Experiments

In this study, we conducted experiments using three datasets collected from amazon.com to evaluate the performance of the ARSRec model. This study answers four research questions (RQs).

- RQ 1: Does the proposed model perform better than other baseline models?
- RQ 2: Does considering the relevance of user reviews and fusion item information really impact recommendation performance?
- RQ 3: What is the most effective computational method for capturing user opinions from user reviews?
- RQ 4: How do different hyperparameters affect the recommendation performance of the proposed model?

### *5.1. Datasets*

We used a publicly accessible dataset of reviews of musical instruments, digital music, and video games from the e-commerce platform Amazon.com to measure the performance of the proposed model. The dataset provided by Amazon.com (https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/, accessed on 13 January 2024) has been utilized in various studies because it contains review text and various item information [7,28,29]. The study utilized a variety of textual data to predict user preferences. To ensure an effective experiment process, we followed previous studies to use data from users who have made at least five purchases [32]. Meanwhile, we used item reviews with at least one helpfulness vote to effectively extract item information. Helpfulness reviews are particularly beneficial as they are known to contain rich information about the items, positively influencing users' purchase decisions [33]. Therefore, this study used reviews with helpfulness votes to effectively extract item information. The statistical information for the full dataset is shown in Table 1. The experimental data were randomly divided into training data, validation data, and test data in a 7:1:2 ratio.

**Table 1.** Statistics of the experimental datasets.

| Feature | Musical Instruments | Digital Music | Video Games |
|---|---|---|---|
| User | 40,630 | 46,440 | 63,931 |
| Item | 59,981 | 210,124 | 47,243 |
| Review & Rating | 357,804 | 505,399 | 581,465 |
| Sparsity (%) | 99.985 | 99.995 | 99.981 |

### 5.2. Evaluation Metrics

We used mean absolute error ($MAE$) and root mean squared error ($RMSE$), which are commonly used in recommender systems, as metrics to measure the performance of our model. The $MAE$ metric is calculated as in Equation (21) and is the sum of the absolute values of the errors divided by the total number of data, with equal weighting for all errors. The $RMSE$ metric is calculated as shown in Equation (22) and is the sum of the squared errors divided by the number of data and squared. Because the $RMSE$ metric takes the error squared, it is a metric that gives greater weight to values with larger errors. Both metrics measure the difference between the predicted value and the actual value as the error, with smaller values indicating higher accuracy [34].

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_{u,i} - \hat{y}_{u,i}|, \tag{21}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{u,i} - \hat{y}_{u,i})^2}. \tag{22}$$

where $u$ is the user, $i$ is the item, $\hat{y}_{u,i}$ is the predicted rating, $y_{u,i}$ is the actual rating, and $N$ is the number of predicted ratings.

### 5.3. Baseline Model

In this study, we compared the performance of the traditional recommendation model and review-based recommendation model as a baseline model to verify the reliability of the proposed model. A brief introduction to the baseline models is provided below.

- PMF [35]: This model is an MF-based recommendation model that predicts user preferences by decomposing user-item interactions into low dimensions based on Gaussian distributions. This approach works better than traditional MF models on sparse and imbalanced data.
- NCF [36]: This model effectively captures the complex interactions between users and items through linear and non-linear learning. The NeuMF structure, which combines the generalized matrix factorization (GMF) model and MLP model, was proposed. This recommendation model only uses rating information.
- DeepCoNN [7]: Deep cooperative neural networks use two parallel CNN networks to extract representation vectors from both user reviews and item reviews. The two extracted vectors are fed into the Factorization Machine for the final rating prediction.
- SAFMR [37]: This model uses CNN techniques to extract the characteristics of users and items from a set of reviews. The self-attention mechanism then considers the importance of various attributes of an item and reflects this in its recommendations.
- NARRE [28]: This model uses the review text and rating matrix as input to the model and utilizes a CNN and attention mechanism to learn the latent features of each user and item review. It uses an attention mechanism to reduce or ignore the weight of low-importance reviews, which allows it to effectively predict ratings.

### 5.4. Implementation Details

To ensure a fair comparison, all experiments were run under the same experimental environment, using the TensorFlow package (Google LLC, Mountain View, CA, USA), and performed on a 128 GB RAM system and NVIDIA V100 GPU (NVIDIA Corporation, Santa Clara, CA, USA).

All hyperparameters for the ARSRec were determined through experimentation and validation on the training dataset. The batch size was chosen from [64, 128, 256, 512, 1024] and the learning rate was chosen from [0.0001, 0.0005, 0.001, 0.005, 0.01]. To avoid overfitting the model, we optimized the dropout rate by adjusting it to [0.1, 0.2, 0.3, 0.4, 0.5]. After optimization, the batch size was set to 64, the learning rate to 0.001, and the dropout rate to 0.1. The embedding size of 64 was used, and the epoch was set to 100. We also set an early stopping if the validation loss did not decrease for five iterations to prevent overfitting of the model training. To reduce experimental error, we repeated the experiment for each model a total of five times and averaged the results. In addition, the parameters of the baseline were determined empirically to make the experiment fair. We trained the model with the training dataset, tuned the parameters on the validation dataset, and optimized the parameters by measuring performance on the test dataset.

## 6. Experimental Results and Discussion

### *6.1. Performance Comparison to Baseline Models (RQ 1)*

In this study, we used three datasets from Amazon.com to compare the performance of our ARSRec with the baseline model. The experimental results in Table 2 show that the ARSRec outperforms the baseline model for all datasets. From the results of the experiment, we can draw the following conclusions.

**Table 2.** Performance comparison to baseline models.

| **Model** | **Musical Instruments** | | **Digital Music** | | **Video Games** | |
|---|---|---|---|---|---|---|
| | **MAE** | **RMSE** | **MAE** | **RMSE** | **MAE** | **RMSE** |
| PMF | 1.1610 | 1.2200 | 1.3860 | 1.4410 | 1.1020 | 1.1650 |
| NCF | 0.7590 | 1.0190 | 0.4990 | 0.7350 | 0.9020 | 1.1650 |
| DeepCoNN | 0.7190 | 0.9820 | 0.4380 | 0.7070 | 0.8450 | 1.1190 |
| SAFMR | 0.7180 | 0.9830 | 0.4240 | 0.6990 | 0.8810 | 1.1490 |
| NARRE | 0.6804 | 0.9694 | 0.3996 | 0.6594 | 0.8420 | 1.0893 |
| ARSRec | 0.5454 | 0.8959 | 0.3070 | 0.6207 | 0.6391 | 0.9305 |

First, the PMF and NCF models utilizing only rating data show the lowest performance of the baselines. Here, NCF shows improved performance results compared to PMF because NCF uses MLPs to learn the complex non-linear interaction relationships between users and items. In contrast, PMF is limited in its ability to predict sophisticated interactions because it learns simple linear relationships. Nevertheless, both models still perform poorly; this implies that the rating data alone are insufficient to understand a user's specific purchase motivations.

Second, the review-based models DeepCoNN, SAFMR, and NARRE show improved performance compared to the rating-based models. Here, NARRE shows the most improved performance results compared to the other baselines. This is because NARRE applies embedding and attention mechanisms for the review data to precisely capture user and item attributes. Conversely, both aforementioned models combine a set of reviews to comprehensively capture item or user attribute information, which is limited for personalized attribute analysis; this shows that extracting information about a user's specific preferences for an item from reviews is an important factor for recommendations.

Finally, the ARSRec outperforms all baseline models, and the reasons are as follows. (1) PMF and NCF models use only the rating matrix to predict user preferences. However, the proposed model improves performance by extracting various opinions from reviews to understand users' purchase motivations and incorporate them into recommendations. (2) The review-based baseline models DeepCoNN, SAFMR, and NARRE assume that all review data are equally important and utilize them in their models. However, the proposed model analyzes the relevance between reviews and item information and gives higher weight to more relevant reviews in its recommendations; this creates an effective mechanism for extracting specific user opinions from reviews while at the same time discarding reviews that are not relevant to the item information.

### *6.2. Model Components Analysis (RQ 2)*

In this study, we calculated the relevance between reviews and fusion item information to extract accurate user opinions about items from user reviews. In this section, we present a novel experimental design to validate that this relevance does indeed have an impact on improving recommendation performance. To that end, we compared the performance as shown in Table 3, where 'Only User Review' refers to the case where the relevance of the review to the item information is not considered. To experiment with this, only the user review semantics vector $V_S^L$ in Equation (7) was included in the model.

**Table 3.** Performance comparison of component analysis.

| Model | Musical Instruments | | Digital Music | | Video Games | |
|---|---|---|---|---|---|---|
| | **MAE** | **RMSE** | **MAE** | **RMSE** | **MAE** | **RMSE** |
| Only User Review | 0.5781 | 0.9101 | 0.3275 | 0.6285 | 0.6669 | 0.9328 |
| ARSRec | 0.5454 | 0.8959 | 0.3070 | 0.6207 | 0.6391 | 0.9305 |

The experiment results confirmed that the ARSRec that considers the relevance of fusion item information performs well. We can infer that this is because the proposed model under-weighted reviews that were either poorly embedded with user opinions or were less relevant to the item. Therefore, the larger the performance difference between the model that does not consider item information and relevance and the proposed model, the more likely it is that the number of invalid reviews in the recommendation will be high.

### *6.3. Fusion Method Efficiency Analysis (RQ 3)*

In this section, we propose a new experimental design to validate which computational approach can most effectively extract opinions from reviews when computing the relevance between the semantics vector $V_S^L$ of user reviews in Equation (7) and the fusion item information $I_{Fuse}^L$ in Equation (16). We conducted experiments using the following four fusion methods, and the results are summarized in Table 4.

**Table 4.** Performance comparison of different fusion methods.

| Fusion Method | Musical Instruments | | Digital Music | | Video Games | |
|---|---|---|---|---|---|---|
| | **MAE** | **RMSE** | **MAE** | **RMSE** | **MAE** | **RMSE** |
| Add | 0.5910 | 0.9171 | 0.3261 | 0.6313 | 0.6505 | 0.9386 |
| Average | 0.5751 | 0.9001 | 0.3241 | 0.6242 | 0.6674 | 0.9330 |
| Concatenation | 0.5658 | 0.9238 | 0.3219 | 0.6255 | 0.6517 | 0.9327 |
| Dot Product | 0.5454 | 0.8959 | 0.3070 | 0.6207 | 0.6391 | 0.9305 |

1. Add: element-wise add, i.e., $V_S^L + I_{Fuse}^L$.
2. Average: element-wise average, i.e., $\left(V_S^L + I_{Fuse}^L\right)/2$.
3. Concatenation: concatenate and, i.e., $V_S^L \oplus I_{Fuse}^L$.
4. Dot Product: element-wise product, i.e., $V_S^L \otimes I_{Fuse}^L$.

Of the four methods, the Dot Product method performs the best. In general, the Concatenation operation performs the best because it minimizes information loss, bringing more information into the model. However, the objective of this study is not to retain a lot of information but to determine the relevance of user reviews to item information, capture relevant information as the core opinions of users, and reflect it in recommendations. Therefore, we can assume that it is more efficient to compute the interaction between two vectors with Element-wise Product than with other operations.

### *6.4. Impact of Hyperparameters (RQ 4)*

In this section, we performed a parametric analysis to explore the optimal settings for the ARSRec. Therefore, in this study, we selected batch size, dropout rate, and learning rate as hyperparameters that affect model performance and set the ranges as shown in Table 5 for further experiments.

**Table 5.** Hyperparameter setting for finding excellent performance.

| Hyper Parameters | Values |
|---|---|
| Batch Size | [64, 128, 256, 512, 1024] |
| Dropout Rate | [0.1, 0.2, 0.3, 0.4, 0.5] |
| Learning Rate | [0.0001, 0.0005, 0.001, 0.005, 0.01] |

First, the experimental results for batch size are summarized in Table 6. It was found that the smallest batch size of 64 performed the best. This is because if the batch size is too large, the model may not be able to update its parameters enough to be trained. Furthermore, while larger values may be faster to learn, they may converge to the minimum of the loss faster than smaller batch sizes, potentially leading to overfitting.

**Table 6.** Performance comparison according to different batch sizes.

| Batch Size | Musical Instruments | | Digital Music | | Video Games | |
|---|---|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| 64 | 0.5454 | 0.8959 | 0.3070 | 0.6207 | 0.6391 | 0.9305 |
| 128 | 0.5688 | 0.9170 | 0.3225 | 0.6239 | 0.6767 | 0.9346 |
| 256 | 0.5729 | 0.9162 | 0.3753 | 0.6291 | 0.6582 | 0.9654 |
| 512 | 0.6053 | 0.9032 | 0.3597 | 0.6482 | 0.6704 | 0.9458 |
| 1024 | 0.5799 | 0.9191 | 0.3469 | 0.6537 | 0.6673 | 0.9344 |

The results of the second parameter, dropout rate, are shown in Table 7. Good performance was shown when it was the smallest value of 0.1. First, this study focuses on extracting accurate user opinions through interaction with user reviews by considering item information comprehensively. In this context, it can be inferred that a high dropout rate can have a detrimental effect on performance because the higher the dropout rate, the larger the loss of information needed to optimize the model.

**Table 7.** Performance comparison according to dropout rates.

| Dropout Rate | Musical Instruments | | Digital Music | | Video Games | |
|---|---|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| 0.1 | 0.5454 | 0.8959 | 0.3070 | 0.6207 | 0.6391 | 0.9305 |
| 0.2 | 0.5637 | 0.9110 | 0.3255 | 0.6238 | 0.6542 | 0.9345 |
| 0.3 | 0.6024 | 0.9090 | 0.3244 | 0.6254 | 0.6529 | 0.9415 |
| 0.4 | 0.5869 | 0.9080 | 0.3189 | 0.6250 | 0.6483 | 0.9480 |
| 0.5 | 0.5629 | 0.9112 | 0.3128 | 0.6253 | 0.6953 | 0.9369 |

Finally, the results for the learning rate are shown in Table 8, where the optimal performance is shown at 0.001. Learning rate is an important parameter that controls the learning speed and number of errors when training a model. The higher the learning rate, the faster the model is trained, but the more likely it is to train incorrectly and not derive optimal weights. Conversely, a lower learning rate increases training time but increases the probability of reaching the optimal weight. In this study, we found optimal performance at 0.001, which is the midpoint of the range we experimented with, suggesting that choosing the right learning rate is important.

**Table 8.** Performance comparison according to learning rate.

| Learning Rate | Musical Instruments | | Digital Music | | Video Games | |
|---|---|---|---|---|---|---|
| | **MAE** | **RMSE** | **MAE** | **RMSE** | **MAE** | **RMSE** |
| 0.0001 | 0.5829 | 0.9131 | 0.3441 | 0.6269 | 0.6658 | 0.9825 |
| 0.0005 | 0.5730 | 0.8983 | 0.3218 | 0.6215 | 0.6648 | 0.9367 |
| 0.001 | 0.5454 | 0.8959 | 0.3070 | 0.6207 | 0.6391 | 0.9305 |
| 0.005 | 0.7010 | 0.9994 | 0.3632 | 0.6581 | 0.6975 | 0.9580 |
| 0.01 | 0.6947 | 1.0182 | 0.5002 | 0.7365 | 0.7585 | 1.0810 |

## 7. Conclusions and Future Work

To address the data sparsity problem, previous studies on recommender systems have extracted attribute information from review data. Because review texts contain users' diverse opinions about items, it is an effective way to extract user preference information and has been used in many recommendation studies. Previous studies have proposed recommendation models based on the assumption that all review text necessarily contains user comments. However, some reviews lack user opinions or contain information that is completely irrelevant to the item. Therefore, incorporating them into the recommendation model can hinder performance improvements. Therefore, this study proposed ARSRec, which incorporates accurate user opinions about target items from reviews into recommendations. To this end, we effectively extracted important attribute information of items through the self-attention mechanism and considered item information comprehensively by fusing various item features through the co-attention mechanism. We then computed the relevance between the integrated item information and user reviews to effectively incorporate accurate opinions into recommendations. The excellence of the ARSRec was confirmed by performance comparison experiments with the baseline. In addition, to investigate whether calculating the relevance between item information and reviews impacts recommendation performance compared to not considering relevance, we conducted further experiments on the ARSRec structure. The additional experiments demonstrated that incorporating review content with relevance consideration is more effective than without relevance consideration. Finally, to explore the most effective method for calculating relevance, we compared various computation methods. The results showed that the Dot Product method used in the ARSRec was the most effective.

The theoretical and practical implications of this study are as follows: Theoretically, this study emphasizes the importance of review texts in recommender systems and highlights the necessity of models that consider the relevance between user reviews and item information to accurately reflect user preferences. Through this approach, it addresses the data sparsity issue and confirms the potential to enhance recommendation performance by effectively extracting various item attributes. Practically, the ARSRec demonstrates its applicability across various domains using real datasets. By comprehensively analyzing user reviews and item information, this model can provide more personalized and accurate recommendations, making it valuable in industries such as e-commerce. In addition, this study explores the most effective methods for calculating the relevance between review content and item information, providing concrete guidelines for optimizing performance in actual system implementations.

Limitations of this study and future work directions are as follows. First, the fusion of item information should be discussed in more detail. In this study, we adopted the method of combining representation vectors extracted from the information to fuse user-generated item reviews and item information provided by sellers. However, there are many different fusion methods, including element-wise addition and gating. Second, the ARSRec should be validated using datasets from other domains for the generalization of this study. Although this study utilizes a public dataset from the e-commerce platform Amazon.com, it is widely applicable to many areas of recommender systems. Therefore, additional validation is required, such as in the movie, restaurant, and hotel sectors. Third, further studies should be conducted on utilizing various pre-trained text embedding techniques. In this study, we only used the BERT embedding method for text data to extract feature representation from text. However, pre-trained NLP techniques, such as RoBERTa and generative pre-trained transformers, have been recently introduced, and it is necessary to conduct further studies by adopting various methods.

**Author Contributions:** Conceptualization, J.K. (Jihyeon Kim) and Q.L.; Methodology, X.L.; Software, X.L. and L.J.; Data curation, L.J.; Writing—original draft, J.K. (Jihyeon Kim); Writing—review & editing, Q.L. and J.K. (Jaekyeong Kim); Supervision, J.K. (Jaekyeong Kim). All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the BK21 FOUR (Fostering Outstanding Universities for Research) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF).

**Data Availability Statement:** The data are available on https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/ (accessed on 13 January 2024).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Xiao, B.; Benbasat, I. E-commerce product recommendation agents: Use, characteristics, and impact. *MIS Q.* **2007**, *31*, 137–209. [CrossRef]
2. Nguyen, T.T.; Hui, P.-M.; Harper, F.M.; Terveen, L.; Konstan, J.A. Exploring the filter bubble: The effect of using recommender systems on content diversity. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Republic of Korea, 7–11 April 2014; pp. 677–686.
3. Wang, R.; Jiang, Y.; Lou, J. ADCF: Attentive representation learning and deep collaborative filtering model. *Knowl. Based Syst.* **2021**, *227*, 107194. [CrossRef]
4. Wang, J.; De Vries, A.P.; Reinders, M.J. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, 6–11 August 2006; pp. 501–508.
5. Jang, D.; Li, Q.; Lee, C.; Kim, J. Attention-based multi attribute matrix factorization for enhanced recommendation performance. *Inf. Syst.* **2024**, *121*, 102334. [CrossRef]
6. Pappas, I.O.; Kourouthanassis, P.E.; Giannakos, M.N.; Chrissikopoulos, V. Explaining online shopping behavior with fsQCA: The role of cognitive and affective perceptions. *J. Bus. Res.* **2016**, *69*, 794–803. [CrossRef]
7. Zheng, L.; Noroozi, V.; Yu, P.S. Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017; pp. 425–434.
8. McAuley, J.; Leskovec, J. Hidden factors and hidden topics: Understanding rating dimensions with review text. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 165–172.
9. Kim, D.; Park, C.; Oh, J.; Lee, S.; Yu, H. Convolutional matrix factorization for document context-aware recommendation. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 233–240.
10. Ma, Y.; Chen, G.; Wei, Q. Finding users preferences from large-scale online reviews for personalized recommendation. *Electron. Commer. Res.* **2017**, *17*, 3–29. [CrossRef]
11. Pourgholamali, F.; Kahani, M.; Bagheri, E.; Noorian, Z. Embedding unstructured side information in product recommendation. *Electron. Commer. Res. Appl.* **2017**, *25*, 70–85. [CrossRef]
12. Wang, A.; Zhang, Q.; Zhao, S.; Lu, X.; Peng, Z. A review-driven customer preference measurement model for product improvement: Sentiment-based importance—Performance analysis. *Inf. Syst. e-Bus. Manag.* **2020**, *18*, 61–88. [CrossRef]
13. Khaledian, N.; Nazari, A.; Khamforoosh, K.; Abualigah, L.; Javaheri, D. TrustDL: Use of trust-based dictionary learning to facilitate recommendation in social networks. *Expert Syst. Appl.* **2023**, *228*, 120487. [CrossRef]
14. Cheng, Z.; Ding, Y.; He, X.; Zhu, L.; Song, X.; Kankanhalli, M.S. Aˆ 3NCF: An Adaptive Aspect Attention Model for Rating Prediction. In Proceedings of the IJCAI, Stockholm, Sweden, 13–19 July 2018; pp. 3748–3754.
15. Cao, R.; Zhang, X.; Wang, H. A review semantics based model for rating prediction. *IEEE Access* **2019**, *8*, 4714–4723. [CrossRef]
16. Liu, Y.-H.; Chen, Y.-L.; Chang, P.-Y. A deep multi-embedding model for mobile application recommendation. *Decis. Support Syst.* **2023**, *173*, 114011. [CrossRef]
17. Wang, S.; Qiu, J. Utilizing a feature-aware external memory network for helpfulness prediction in e-commerce reviews. *Appl. Soft Comput.* **2023**, *148*, 110923. [CrossRef]
18. Liu, H.; Wang, Y.; Peng, Q.; Wu, F.; Gan, L.; Pan, L.; Jiao, P. Hybrid neural recommendation with joint deep representation learning of ratings and reviews. *Neurocomputing* **2020**, *374*, 77–85. [CrossRef]
19. Lee, S.; Kim, D. Deep learning based recommender system using cross convolutional filters. *Inf. Sci.* **2022**, *592*, 112–122. [CrossRef]

20. Qiu, Z.; Wu, X.; Gao, J.; Fan, W. U-BERT: Pre-training user representations for improved recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; pp. 4320–4327.
21. Kuo, R.; Li, S.-S. Applying particle swarm optimization algorithm-based collaborative filtering recommender system considering rating and review. *Appl. Soft Comput.* **2023**, *135*, 110038. [CrossRef]
22. Geng, S.; Liu, S.; Fu, Z.; Ge, Y.; Zhang, Y. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In Proceedings of the 16th ACM Conference on Recommender Systems, Seattle, WA, USA, 18–23 September 2022; pp. 299–315.
23. Liu, P.; Zhang, L.; Gulla, J.A. Pre-train, Prompt, and Recommendation: A Comprehensive Survey of Language Modeling Paradigm Adaptations in Recommender Systems. *Trans. Assoc. Comput. Linguist.* **2023**, *11*, 1553–1571. [CrossRef]
24. Li, L.; Zhang, Y.; Chen, L. Personalized prompt learning for explainable recommendation. *ACM Trans. Inf. Syst.* **2023**, *41*, 103. [CrossRef]
25. Chang, H.-S.; Sun, R.-Y.; Ricci, K.; McCallum, A. Multi-CLS BERT: An Efficient Alternative to Traditional Ensembling. *arXiv* **2022**, arXiv:2210.05043.
26. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
28. Chen, C.; Zhang, M.; Liu, Y.; Ma, S. Neural attentional rating regression with review-level explanations. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1583–1592.
29. Chin, J.Y.; Zhao, K.; Joty, S.; Cong, G. ANR: Aspect-based neural recommender. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 147–156.
30. More, A. Attribute extraction from product titles in ecommerce. *arXiv* **2016**, arXiv:1608.04670.
31. Wang, Q.; Yang, L.; Kanagal, B.; Sanghai, S.; Sivakumar, D.; Shu, B.; Yu, Z.; Elsas, J. Learning to extract attribute value from product via question answering: A multi-task approach. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 47–55.
32. Yang, S.; Li, Q.; Jang, D.; Kim, J. Deep learning mechanism and big data in hospitality and tourism: Developing personalized restaurant recommendation model to customer decision-making. *Int. J. Hosp. Manag.* **2024**, *121*, 103803. [CrossRef]
33. Park, J.; Li, X.; Li, Q.; Kim, J. Impact on recommendation performance of online review helpfulness and consistency. *Data Technol. Appl.* **2023**, *57*, 199–221. [CrossRef]
34. Isinkaye, F.O.; Folajimi, Y.O.; Ojokoh, B.A. Recommendation systems: Principles, methods and evaluation. *Egypt. Inform. J.* **2015**, *16*, 261–273. [CrossRef]
35. Mnih, A.; Salakhutdinov, R.R. Probabilistic matrix factorization. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007; Volume 20.
36. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.-S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
37. Ma, H.; Liu, Q. In-depth Recommendation Model Based on Self-Attention Factorization. *KSII Trans. Internet Inf. Syst.* **2023**, *17*, 721–739.