electronics

MDPI

Article

Incorporating Implicit and Explicit Feature Fusion into Hybrid Recommendation for Improved Rating Prediction

Qinglong Li [1], Euiju Jeong [2], Seok-Kee Lee [1] and Jiaen Li [3,*]

[1] Division of Computer Engineering, Hansung University, Seoul 02876, Republic of Korea
[2] Department of Big Data Analytics, Kyung Hee University, Seoul 02447, Republic of Korea
[3] Division of International Trade, Kwangwoon University, Seoul 01897, Republic of Korea
* Correspondence: jiaenlee@kw.ac.kr; Tel.: +82-2-940-5569

Abstract: Online review texts serve as a valuable source of auxiliary information for addressing the data sparsity problem in recommender systems. These reviews often reflect user preferences across multiple item attributes and can be effectively incorporated into recommendation models to enhance both the accuracy and interpretability of recommendations. Review-based recommendation approaches can be broadly classified into implicit and explicit methods. Implicit methods leverage deep learning techniques to extract latent semantic representations from review texts but generally lack interpretability due to limited transparency in the training process. In contrast, explicit methods rely on hand-crafted features derived from domain knowledge, which offer high explanatory capability but typically capture only shallow information. Integrating the complementary strengths of these two approaches presents a promising direction for improving recommendation performance. However, previous research exploring this integration remains limited. In this study, we propose a novel recommendation model that jointly considers implicit and explicit representations derived from review texts. To this end, we incorporate a self-attention mechanism to emphasize important features from each representation type and utilize Bidirectional Encoder Representations from Transformers (BERT) to capture rich contextual information embedded in the reviews. We evaluate the performance of the proposed model through extensive experiments using three real-world datasets. The experimental results demonstrate that our model outperforms several baseline models, confirming its effectiveness in generating accurate and explainable recommendations.

Keywords: recommender system; online review; co-attention mechanism; BERT; explainable recommendation

Academic Editor: Xianzhi Wang

Received: 10 May 2025
Revised: 4 June 2025
Accepted: 9 June 2025
Published: 11 June 2025

Citation: Li, Q.; Jeong, E.; Lee, S.-K.; Li, J. Incorporating Implicit and Explicit Feature Fusion into Hybrid Recommendation for Improved Rating Prediction. *Electronics* **2025**, *14*, 2384. https://doi.org/10.3390/electronics14122384

1. Introduction

With the rapid advancement of information and communication technologies and the broad use of the internet, the e-commerce market has continued to expand [1,2]. As new products are introduced yearly, users can conveniently browse and purchase various items online [3]. However, this increased availability of options has led to the problem of information overload, where users must select suitable items from a large volume of information [4]. In response to this challenge, businesses have recognized the growing importance of recommender systems that provide personalized item suggestions based on individual user preferences. Recommender systems typically analyze users' historical behavioral data, such as purchase history, click patterns, and viewing records, to estimate preferences and recommend items that are more likely to be accepted or purchased [5]. This reduces users' information search costs, enhances their overall shopping experience, and

enables businesses to improve customer satisfaction and secure a sustainable competitive advantage [6]. Accordingly, recommender systems have become a core component of e-commerce platforms, significantly improving user experience and business performance.

Collaborative filtering (CF) is one of the most widely used recommendation models in e-commerce, which provides suggestions based on users' past behavior records (e.g., ratings, clicks, and visits) [6,7]. However, since these approaches rely solely on historical behavioral data, they often fail to capture the underlying motivations behind user preferences, which can limit recommendation accuracy [8]. In particular, data sparsity refers to the lack of sufficient historical user–item interactions and is a major factor that negatively affects recommendation performance [9]. To address this limitation, researchers have proposed models that incorporate auxiliary information, such as online review texts [6,10]. These reviews contain valuable insights into user preferences related to various attributes of an item. Integrating such information into recommendation models makes it possible to clarify the rationale behind specific recommendations and enhance prediction accuracy [3,11]. This approach is beneficial for users with limited behavioral history, often referred to as cold-start users [12]. Consequently, many studies have focused on analyzing review texts to extract preference information and incorporate it into recommendation frameworks [6,9,13].

Review-based recommendation approaches can generally be divided into implicit and explicit methods, depending on how features are extracted from the review texts [14]. Implicit methods aim to capture latent feature representations without explicitly interpreting the semantic content of the reviews [15]. For instance, convolutional neural networks (CNN) are commonly employed to encode review texts into dense representations that serve as embeddings for users and items [7]. Although such deep learning-based methods have demonstrated strong performance in preference prediction, their limited transparency in the training process often reduces interpretability.

In contrast, explicit methods are grounded in domain knowledge and extract predefined features from reviews using analytical techniques such as topic modeling and sentiment analysis. For example, topic distributions derived from review texts using Latent Dirichlet Allocation (LDA) can be integrated into recommendation models to help explain why a specific item was suggested to a user [16]. While explicit methods provide a high level of explanatory capability, they may compress complex textual information into simplified numerical forms, which can lead to the loss of valuable semantic details in the original text. Considering these characteristics, implicit and explicit approaches each offer distinct advantages [14]. By leveraging the complementary strengths of both, it is possible to build more accurate and explainable recommendation models that fully utilize the information embedded in user-generated reviews.

Owing to the success of deep learning in computer vision and natural language processing, increasing attention has been given to the effective fusion of heterogeneous features [17]. Early studies combined different types of features using element-wise product or simple concatenation. However, these approaches are limited in their ability to fully capture and represent the complex interactions between features [18]. To address this limitation, recent studies have proposed advanced feature fusion methods that incorporate attention mechanisms to better model the complementarity between features [19]. By jointly learning the interactions among diverse features, attention-based fusion techniques can capture how one feature influences another and generate richer and more expressive representations [20]. In practice, the fused features produced by these techniques have been shown to significantly improve model performance [19]. In this context, incorporating such fused features into a recommendation framework enables a more effective model by integrating the complementary strengths of implicit and explicit methods.

However, research considering the complementarities between explicit and implicit methods in recommender systems remains relatively limited. Previous studies have primarily evaluated these two approaches independently [7,21] or combined them using simple fusion strategies such as concatenation or weighted averaging [22,23]. Although attention-based fusion has demonstrated promise in related domains, its structured application within hybrid recommendation models, particularly for jointly modeling intra-feature relevance and inter-feature interactions, has not been sufficiently explored.

To address the limitations of previous studies, we propose a novel recommendation model called HNNER (Hybrid Neural Network for Explainable Recommendation), which integrates LDA-derived explicit features with BERT-based implicit representations through a hierarchical attention architecture. In our approach, self-attention mechanisms are employed to capture contextual dependencies within each feature type, while co-attention mechanisms are used to model their mutual interactions. This design enables more dynamic and interpretable fusion compared to prior methods, facilitating a richer integration of semantic and topic-level information for improved recommendation performance. To evaluate the recommendation performance of the proposed HNNER, we conducted experiments using three product category datasets from Amazon. The results demonstrate that HNNER outperforms various baseline models. The key contributions of this study are as follows:

- We propose HNNER, a novel recommendation model designed to evaluate the complementary effects of explicit and implicit methods by fully leveraging review texts. The model comprehensively captures the complementarity between the two approaches to enhance the recommendation.
- We introduce a self-attention mechanism and a co-attention mechanism to fully exploit intra-method and inter-method feature information. The self-attention mechanism emphasizes the importance of each feature within explicit and implicit methods, while the co-attention mechanism captures the complementarity between the two.
- We validate the superiority of the proposed HNNER by comparing it with baseline models using real-world Amazon datasets across three product categories. The experimental results confirm that HNNER achieves better performance than existing models.

This paper is organized as follows. Section 2 reviews related work. Section 3 defines the research problem addressed in this study. Section 4 presents the proposed HNNER model. Section 5 describes the dataset and experimental design. Section 6 reports and discusses the experimental results. Finally, Section 7 concludes the paper and outlines directions for future research.

2. Related Work

2.1. Review-Based Recommendation

CF is one of the most widely used approaches in recommender system research, generating recommendations based on a user's past behavioral records such as ratings, clicks, and visits [4]. However, these models rely solely on user behavior data and do not capture the underlying motivations behind purchasing decisions [15]. Another critical issue is data sparsity, as many users have limited interaction history [4]. To address this challenge, numerous studies have proposed incorporating review texts into recommendation models [5]. Review texts contain rich information about user preferences related to various item attributes [3]. Accordingly, researchers have applied text mining techniques to extract feature representations embedded in review texts and integrate them into recommendation models to enhance prediction accuracy [5,15].

In review-based recommendations, feature extraction methods can largely be categorized into explicit and implicit approaches [14]. Explicit methods use techniques such as

topic modeling to extract interpretable features based on domain knowledge. For instance, McAuley and Leskovec [16] proposed a model that uses LDA to extract topics from review texts and combines them with matrix factorization (MF), significantly improving recommendation performance. Similarly, Bao et al. [24] presented a hybrid model that integrates latent factor vectors of users and items, derived via MF, with topic distribution parameters obtained through LDA.

With the advancement of deep learning, implicit methods such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been widely adopted to extract latent features from review texts [4]. These methods automatically learn semantic representations that are difficult to extract manually, using neural architectures without explicitly analyzing the text content. For example, Zheng, Noroozi and Yu [7] proposed a model that applies CNNs to sets of user and item reviews to learn their respective representations, achieving strong recommendation performance. Chen, Zhang, Liu, and Ma [21] introduced a review-level attention pooling mechanism to focus on informative features, while Liu, Wang, Peng, Wu, Gan, Pan, and Jiao [5] proposed a model that combines ratings and review texts to learn deep user and item representations, further enhancing performance using an attention mechanism to identify important reviews.

Overall, explicit methods offer the advantage of extracting interpretable, domain-informed features that describe user preferences for item attributes [17]. However, they require manual preprocessing and often fail to capture the contextual semantics present in reviews, resulting in relatively shallow representations [14,15]. In contrast, implicit methods can automatically learn deeper and more complex semantic features through deep learning architectures. Despite their strong predictive power, they often lack transparency, making the recommendation process harder to interpret [17].

In summary, explicit and implicit methods each have distinct strengths. When combined, they can complement each other to improve both the accuracy and interpretability of recommendations. Nevertheless, to the best of our knowledge, few studies have effectively integrated these two approaches. In this study, we propose a novel recommendation model that incorporates fused features derived from explicit and implicit methods to leverage their complementary strengths.

2.2. Deep Learning Techniques for Recommendation

2.2.1. BERT

The remarkable success of deep learning has led to its widespread application in natural language processing (NLP) tasks [18]. In particular, the way words are represented has a significant impact on NLP performance, and this aspect has been studied extensively [25]. Recently, pre-trained language models such as ELMo, BERT, and GPT-3 have been widely adopted. Among these, BERT, developed by Google in 2018, has demonstrated outstanding performance across a wide range of NLP tasks.

The core strength of BERT lies in its ability to represent each word in a sentence within a bidirectional context [26]. This allows the model to consider both semantic and syntactic relationships between words, resulting in a more nuanced and context-aware representation than conventional static word embeddings. One of BERT's distinguishing features is its ability to generate different vector representations for the same word depending on the surrounding context, thereby capturing the diversity and complexity of natural language more effectively [27].

In the context of recommendation systems, several studies have leveraged BERT for feature extraction from textual data. For example, Ray et al. [28] proposed a hotel recommender system that uses an ensemble of BERT and random forest models to classify the sentiment of review texts. They also employed Word2Vec and TF-IDF to categorize reviews

by aspect. Kaviani and Rahmani [27] developed a recommendation model that applies BERT-generated embeddings to the K-means clustering algorithm to recommend hashtags for tweets. Yang et al. [29] proposed a research literature and researcher recommender system using a semi-supervised learning framework that combines pre-trained BERT with LDA. Their model was compared against other embedding methods, and the experimental results showed that the BERT-based approach achieved superior performance.

Recently, more advanced techniques such as BERTopic and large language model-based representations (e.g., GPT-3 or RoBERTa) have been introduced, offering contextually rich and dynamically adaptive topic modeling capabilities [9]. While these methods exhibit strong semantic abstraction, they often involve substantial computational overhead and lack the interpretability required in recommendation scenarios where transparency and feature attribution are critical [30]. In contrast, our framework employs BERT to extract implicit semantic representations while maintaining compatibility with explicit topic signals from LDA. This hybrid configuration not only ensures computational tractability but also enhances the interpretability of recommendations, making it more suitable for real-world deployment where explainability and efficiency are both essential [17].

These findings suggest that pre-trained BERT models are effective for extracting rich feature representations from review texts, which can be leveraged to enhance the quality of recommendations. Based on this motivation, we propose a novel recommendation framework that utilizes BERT to identify diverse semantic features embedded in review texts and integrates them into a recommendation model to improve predictive performance.

2.2.2. Feature Fusion Techniques

With the success of deep learning in computer vision and NLP, there has been increasing interest in research focused on effective feature fusion techniques [17]. Early studies typically employed element-wise operations or simple concatenation to combine features. However, such methods are limited in their ability to capture the complex interactions and dependencies among features [20]. To address this limitation, recent studies have introduced co-attention mechanisms as an advanced feature fusion strategy to better understand and integrate interdependent features [19].

Co-attention mechanisms are designed to simultaneously learn the interactions between multiple features, enabling the model to evaluate the contribution of each feature and extract more informative representations [19]. Several studies have demonstrated the effectiveness of co-attention in various domains. For example, Lu et al. [31] proposed a co-attention mechanism that jointly processes image and question data for visual question answering. Li, Liu, Hu, and Jiang [20] developed a hashtag recommendation model that combines topic modeling and LSTM, applying a co-attention mechanism to capture the interaction between semantic features. Ren, Diao, Guo, and Hong [17] proposed a multi-modal recommendation model that integrates textual and visual features to predict review helpfulness, leveraging a co-attention mechanism to extract fused representations by modeling the interdependence between modalities. Furthermore, Liu, Liu, Cao, and Du [19] designed a novel co-attention mechanism by modifying the self-attention structure in the Silver Transformer model. Their approach effectively integrates mutual attention across different vector types to generate fused encodings. Experimental results confirmed the superior performance of their model in classification tasks, showing that it successfully captures relevant parts of both the input text and associated labels. They also highlighted the utility of self-attention mechanisms, which are widely used to identify salient features within a single modality and contribute to effective feature fusion [6].

While deep learning-based representations can extract rich and complex features, capturing dependencies among these features remains a key challenge. Self-attention

mechanisms offer a means to address this challenge by evaluating the relative importance of each feature, filtering out irrelevant information, and modeling intra-feature relationships in a flexible and data-driven manner [17].

Recent approaches, such as BERTopic or transformer-based topic fusion methods typically rely on either static feature concatenation or simplified embedding integration. While these approaches demonstrate promising semantic abstraction, they often overlook the dynamic interdependence between feature types. Furthermore, large language model-based fusion is frequently treated as a black-box process, lacking explicit control over feature-level interactions and offering limited interpretability [2,17].

In contrast, our proposed framework enhances recommendation accuracy by complementarily leveraging the respective strengths of explicit and implicit methods. We apply a self-attention mechanism to both feature types to emphasize their most informative aspects and reduce representational noise. Additionally, we introduce a co-attention mechanism to explicitly model their mutual complementarity and generate a unified fused representation. By integrating these fused features into the recommendation framework, we can validate the architectural efficacy of dual attention for robust and explainable recommendations.

3. Problem Definition

The overall architecture of the proposed HNNER model is illustrated in Figure 1. The model comprises three main components: the User–Item Interaction (UII) network, the Feature Extraction (FE) network, and the Preference Prediction (PP) network. Research involving explicit and implicit methods in review-based recommendation has been critical to understanding diverse user preferences and behaviors. However, previous studies have typically adopted the two methods separately using general approaches. In this study, we propose HNNER, a recommendation model that integrates fused features by capturing the complementary strengths of explicit and implicit methods. The model incorporates a self-attention mechanism to highlight the most informative aspects of each feature and a co-attention mechanism to model the dependencies between features, thereby generating enriched fused representations. These attentive vectors and fused features are subsequently passed to the Preference Prediction network to estimate user ratings.

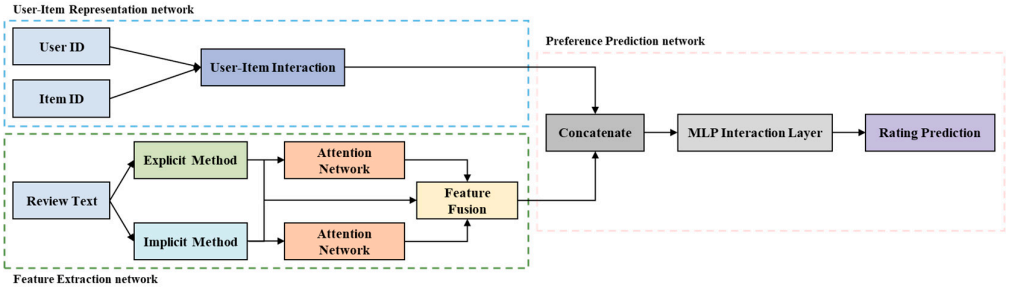

Figure 1. Architecture of the HNNER model.

Let D denote the set of interactions between users and items, where $D = (u, r, i, y)$ is a tuple consisting of a user u, an item i, the user's review text r, and the associated preference rating y. The objective of the proposed model is to learn a function C that predicts the preference rating $\hat{y}_{u,i}$ for a given user–item pair. The prediction function C can be defined as

$$C(u, r, i, y; \theta) \rightarrow \hat{y}, \tag{1}$$

where θ represents the model parameters, and $\hat{y}_{u,i}$ is the predicted rating. During training, the model learns to minimize the difference between the predicted rating $\hat{y}_{u,i}$ and the actual

rating $y_{u,i}$ by leveraging user u, item i, and review text r. After training, the model outputs a predicted preference rating for unseen items.

4. HNNER Framework

In this study, we propose HNNER, a recommendation model that leverages the complementary strengths of explicit and implicit methods. Specifically, the model applies self-attention and co-attention mechanisms to effectively capture the importance of user preference features and the interdependencies between them. As illustrated in Figure 2, the architecture of HNNER consists of three networks, which are described in detail below.

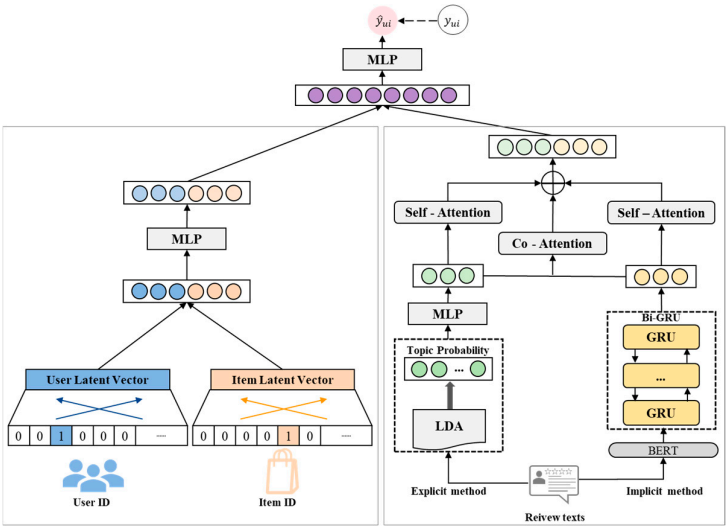

Figure 2. Framework of the HNNER model.

4.1. User–Item Interaction Network

The objective of the UII network is to learn the complex interactions between users and items. First, user u and item i are embedded to obtain their latent representations, p_u and q_i, respectively. These representations are computed as shown in Equation (2):

$$\begin{aligned} p_u &= P^T v_u^U, \\ q_i &= Q^T v_i^I, \end{aligned} \tag{2}$$

where $P \in \mathbb{R}^{m \times d}$ is the user embedding matrix, $Q \in \mathbb{R}^{n \times d}$ is the item embedding matrix; m and n denote the numbers of unique users and items, respectively; and d is the number of dimensions of the latent vector. v_u^U and v_i^I represent the one-hot encodings of user u and item i.

Next, the user and item latent vectors are combined using a concatenation operation, as shown in Equation (3):

$$V^U = [p_u \oplus q_i], \tag{3}$$

where $\oplus$ denotes the concatenation operator. The output vector V^U is then passed through a multi-layer perceptron (MLP) to capture high-order interactions via nonlinear transformations. This process is defined in Equation (4):

$$\begin{aligned} V_1^U &= a_1\left(W_1^T V_0^U + b_1\right), \\ &\vdots \\ V_S^U &= a_S\left(W_S^T V_{S-1}^U + b_S\right), \end{aligned} \tag{4}$$

where W_S^T, b_S, and a_S represent the weight matrix, bias vector, and activation function (ReLU) at the $s-$ 1st layer, respectively. As a result, the final output of this network is

the high-level vector representation V_S^U which encodes the interaction between user u and item i.

4.2. Feature Extraction Network

The objective of the FE network is to extract fused feature representations embedded in the review text by leveraging both explicit and implicit methods. Let $R_{u,i} = \{r_1, r_2, \ldots r_l\}$ denote the review written by user u for the item i, where l is the length of the review and r_n represents the n-th word in the review. The details of the feature extraction process are described below.

4.2.1. Explicit Method

The explicit method extracts features from the review text $R_{u,i}$ using the LDA technique, which is a representative approach for explicitly modeling semantic content. Specifically, the LDA technique models the distribution of topics within a document and the distribution of words within each topic. First, for each document, the proportion of topics is initialized using a Dirichlet distribution, which, in turn, determines the word distribution for each topic. Each word in the document is then assigned to a topic based on this distribution. This process is iterated to optimize both the topic distribution across the document and the word distribution within each topic. Accordingly, we apply LDA to the review text $R_{u,i}$ and extract topic probability values, as represented in Equation (5):

$$O^{Explicit} = LDA(R_{u,i}), \tag{5}$$

where $O^{Explicit}$ denotes the topic probability vector derived from the review text. This vector is then passed through a MLP, as defined in Equation (6):

$$\begin{aligned} O_1^{Explicit} &= a_1\left(W_1^T O_0^{Explicit} + b_1\right), \\ &\vdots \\ O_L^{Explicit} &= a_L\left(W_L^T O_{L-1}^{Explicit} + b_L\right), \end{aligned} \tag{6}$$

where W_L^T, b_L, and a_L refer to the weight matrix, bias vector, and ReLU activation function at the $L-1$st layer, respectively. The resulting $O_L^{Explicit}$ represents the feature vector extracted by the explicit method after MLP processing.

The self-attention mechanism computes the contextualized representation of the explicit feature vector $O_L^{Explicit}$ by modeling pairwise interactions between feature components. As shown in Equation (7), the attention output is computed as

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{7}$$

where $Q,\ K\ V \in \mathbb{R}^{q \times d_k}$ represent the query, key, and value matrices, with l denoting the number of features and d their embedding dimension. These are derived via learned linear projections from d, their embedding dimension. These are derived via learned linear projections from $O_L^{Explicit}$, as shown in Equation (8):

$$Q = O_L^{Explicit} W^Q, K = O_L^{Explicit} W^K, V = O_L^{Explicit} W^V, \tag{8}$$

where $W^Q,\ W^K,\ W^V \in R^{d \times d}$ are trainable weight matrices. This formulation allows the model to compute dynamic attention weights $\alpha = \text{Softmax}\left(QK^T/\sqrt{d}\right)$, which reflect the relative importance of each feature. The resulting vector E^q encodes the attended representation of the explicit features for downstream fusion and prediction.

4.2.2. Implicit Method

The implicit method extracts features from the review text $R_{u,i}$ using a pre-trained BERT model, which has demonstrated strong performance in natural language processing tasks. We use the [CLS] token as the text embedding vector, since the [CLS] representation in BERT captures the overall semantic meaning of the input sentence. This is because the [CLS] token is connected to a layer that is fully connected to all other tokens [32]. The output of the [CLS] token is a 768-dimensional vector, which is denoted as shown in Equation (9):

$$O^{BERT} = BERT(R_{u,i}), \tag{9}$$

where O^{BERT} represents the embedding vector corresponding to the [CLS] token in the review text $R_{u,i}$.

To incorporate contextual information in both forward and backward directions, we apply a bidirectional gated recurrent unit (Bi-GRU) to the output of BERT. The GRU is a type of recurrent neural network that uses a reset gate and an update gate to effectively model sequential dependencies. Bi-GRU extends this mechanism by processing the input in both directions to enhance context representation. The computation procedure is described in Equation (10):

$$O^{Implicit} = \left[\overrightarrow{GRU_{O^{BERT}}}, \overleftarrow{GRU_{O^{BERT}}} \right], \tag{10}$$

where $O^{Implicit}$ denotes the feature representation extracted by the implicit method. This output is then passed through a self-attention mechanism to identify the most informative features. The attention operation is defined in Equation (11):

$$I^a = Attention\left(O^{Implicit}W^Q, O^{Implicit}W^K, O^{Implicit}W^V\right), \tag{11}$$

where I^a is the attentive representation vector computed by the self-attention mechanism, which reflects the relative importance of the features extracted by the implicit method. Consequently, the outputs of this process are $O^{Implicit}$, the latent feature representation, and I^a, the attention-weighted representation vector that emphasizes significant components in the implicit method.

4.2.3. Feature Fusion

Feature fusion complementarily combines explicit and implicit methods to extract enriched representations. The goal of this study is to leverage the strengths of both approaches to improve recommendation accuracy. To this end, we employ a co-attention mechanism that simultaneously learns the interactions among features, identifies their relative influence, and captures deeper semantic representations. The co-attention mechanism is implemented as a variant of the multi-head attention module used in transformer networks. The multi-head attention mechanism enables the model to jointly attend to information from different representation subspaces. As shown in Equation (12), it computes h independent attention heads, each defined as

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right), \tag{12}$$

where $W_i^Q \in \mathbb{R}^{d \times d_q}$, $W_i^K \in \mathbb{R}^{d \times d_k}$, and $W_i^V \in \mathbb{R}^{d \times d_v}$ are the projection matrices for the i-th head. The outputs of all heads are concatenated and linearly transformed using the output projection matrix $W^O \in \mathbb{R}^{h \cdot d_h \times d}$, where $d_h = d/h$ denotes the dimension per head. By assigning distinct projections to each head, the model captures diverse and complementary aspects of the input through parallel attention mechanisms.

In this context, the explicit and implicit feature vectors obtained from previous steps, $O_L^{Explicit}$ and $O^{Implicit}$, are input into the multi-head attention mechanism, as shown in Equation (13):

$$\begin{aligned} E_{co-att} &= MultiHead\left(O_L^{Explicit}, O^{Implicit}, O^{Implicit}\right), \\ I_{co-att} &= MultiHead\left(O^{Implicit}, O_L^{Explicit}, O_L^{Explicit}\right), \end{aligned} \tag{13}$$

where E_{co-att} is the review representation generated by attending explicit features to implicit features, and I_{co-att} is the reverse.

Subsequently, we apply residual connections and two independent feed-forward networks (FFNs), preceded by layer normalization (LN), to generate the fused encodings E_{fuse} and I_{co-att}. This process is described in Equation (14):

$$\begin{aligned} E_{fuse} &= LN\left(FFN(E_{co-att}) + O_L^{Explicit}\right), \\ I_{fuse} &= LN\left(FFN(I_{co-att}) + I_L^{Implicit}\right), \end{aligned} \tag{14}$$

After obtaining E_{fuse} and I_{co-att}, the two vectors are merged via element-wise product, as defined in Equation (15):

$$M = E_{co-att} \odot I_{co-att}, \tag{15}$$

where $\odot$ denotes the element-wise product operation. The attentive representation vectors E^a and I^a, obtained from the explicit and implicit methods, respectively, are then combined with M to construct the final fused feature representation. This operation is defined in Equation (16):

$$F_F = [E^a \oplus I^a \oplus M], \tag{16}$$

where F_F denotes the fused feature vector obtained by complementarily integrating explicit and implicit features. As a result, the output of this module is the final fused representation F_F, which serves as a comprehensive semantic embedding used in downstream preference prediction.

4.3. Preference Prediction Network

The PP network estimates the final user rating based on both the user–item interaction representation and the fused review-based features. To accomplish this, we first combined V_s^U, obtained from the UII network, with F_F extracted from the FE network. This computation is represented in Equation (17):

$$V^c = \left[V_S^U \oplus F_F\right], \tag{17}$$

where V^c denotes the concatenated vector of the two representations. This combined vector is then passed through a MLP to predict the rating, as defined in Equation (18):

$$\begin{aligned} V_I^c &= a_1\left(W_1 V_0^c + b_1\right), \\ &\vdots \\ V_m^c &= a_m\left(W_m V_{m-1}^c + b_m\right), \end{aligned} \tag{18}$$

where W_m^T, b_m, and a_m refer to the weight matrix, bias vector, and ReLU activation function at the m-th layer, respectively. V_m^a is the output of the MLP, which is subsequently input into the final prediction layer. The predicted rating is obtained through a regression layer, as described in Equation (19):

$$\hat{y}_{u,i} = f\left(W_u^i V_m^c\right), \tag{19}$$

where W_u^i is the weight matrix of the output layer, and $\hat{y}_{u,i}$ denotes the predicted user preference score. Since rating prediction is a regression task, we follow prior research and train our model using mean squared error (MSE) as the loss function. The loss is calculated as shown in Equation (20):

$$L = \sum_{u,i \in \mho} (\hat{y}_{u,i} - y_{u,i})^2, \tag{20}$$

where $\mho$ denotes the set of user–item pairs used for training, $\hat{y}_{u,i}$ is the predicted rating, and $y_{u,i}$ is the actual rating.

To optimize the model parameters and minimize the loss function during training, we adopt the Adaptive Moment Estimation (Adam) optimizer, which is based on the stochastic gradient descent (SGD) method. Adam dynamically adjusts the learning rate for each parameter and ensures that large gradients do not result in overly large parameter updates, thereby maintaining training stability.

In addition, given that neural networks are prone to overfitting, we implement several regularization strategies. First, dropout is applied, and the dropout rate is fine-tuned for each dataset. Second, if the validation loss does not decrease, the learning rate is reduced by 10% to enable more refined gradient updates. Finally, early stopping is employed to prevent overfitting when the validation loss does not improve for five consecutive epochs.

5. Experiments

To evaluate the recommendation performance of the proposed HNNER model, we conducted a series of experiments using datasets from three product categories on Amazon.com. This section aims to address the following four research questions (RQs):

- RQ 1: Does the proposed HNNER model outperform baseline recommendation models?
- RQ 2: Does the training time and training loss of the HNNER model indicate computational efficiency?
- RQ 3: What is the most efficient fusion method for recommendation performance?
- RQ 4: How do different hyperparameter settings influence the performance of the proposed HNNER model?

5.1. Datasets

To evaluate the recommendation performance of the proposed HNNER model, we used publicly available datasets from the Amazon e-commerce platform, specifically the Musical Instruments, Digital Music, and Video Games categories [33]. Amazon is the world's largest e-commerce platform and is widely used in recommender system research due to its extensive data, including purchase histories and user-generated review texts [6]. The review data spans from May 1996 to October 2018, covering over two decades of user activity and product interactions. The datasets vary in scale and content complexity: Video Games contains relatively longer and more descriptive reviews, whereas Digital Music tends to include shorter user feedback. Dataset sparsity is summarized in Table 1, and all three datasets exhibit high sparsity levels exceeding 99%, a common challenge in recommender systems that motivates the need for leveraging auxiliary textual information. The dataset was preprocessed through the following steps. First, review texts were tokenized and converted to lowercase. Second, stop words, extra spaces, non-English characters, special characters, and numeric values were removed. Third, words with a frequency of three or fewer were filtered out to reduce noise. Fourth, we excluded users who had purchased five or fewer items to ensure sufficient interaction data per user. The final datasets were randomly split into training (70%), validation (10%), and test (20%) sets. A detailed statistical summary is presented in Table 1, including the number of users, items, reviews, and sparsity ratios.

Table 1. Summary statistics of the Amazon.com review datasets.

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

5.2. Metrics

We used Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as evaluation metrics to measure the predictive performance of the proposed model. These metrics are widely adopted in recommender system research [6]. MAE is calculated by dividing the sum of the absolute differences between the actual and predicted ratings by the total number of rating instances. It treats all prediction errors equally, regardless of their magnitude. RMSE is computed by taking the square root of the average of the squared differences between the actual and predicted ratings. Compared to MAE, RMSE penalizes larger errors more heavily, making it sensitive to outliers. The formulas for MAE and RMSE are defined in Equations (21) and (22), respectively:

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

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

where N is the total number of ratings, $\hat{y}_{u,i}$ is the predicted rating, and $y_{u,i}$ is the actual rating. Both metrics evaluate the accuracy of the predicted ratings by quantifying the discrepancy between actual and predicted values. Lower MAE and RMSE values indicate higher prediction accuracy.

5.3. Baseline Model

To reliably validatede the proposed HNNER model, we compared its recommendation performance with those of selected baseline models that are widely used in existing recommendation studies.

PMF [34]: This model capture models user–item interactions by assuming Gaussian-distributed latent factors for users and items. This probabilistic approach is well-suited for dealing with sparsity and imbalance in rating data.

HFT [16]: This model utilizes topic distributions to learn latent factors between users and items in review text. Specifically, we use LDA techniques to extract hidden topics within the review text and combine these with hidden factors obtained through rating matrix decomposition.

DeepCoNN [7]: This model uses two CNN processors for each user review and item review to extract representations of users and items. It then uses a factorization machine to predict ratings.

NARRE [21]: This model incorporates review texts and rating data, using CNNs to extract features and an attention mechanism to identify informative reviews. By assigning lower weights to irrelevant content, the model improves the quality of user–item latent representations for rating prediction.

SAFMR [35]: This model integrates CNN-based feature extraction with a self-attention mechanism to model user and item representations from review texts. The self-attention

mechanism captures the significance of different textual features, contributing to more accurate rating predictions.

HNNER-E: This model is a variant of the proposed HNNER model that considers only the explicit method. Specifically, we used topic probability values extracted through the LDA technique and applied a self-attention mechanism to consider the importance of features.

HNNER-I: This model is a variant of the proposed HNNER model that considers only the implicit method. Specifically, we used a feature representation extracted using the BERT and Bi-GRU techniques and applied a self-attention mechanism to consider the importance of features.

To sum up the different characteristics of these models, Table 2 presents the data type and the key methods employed in these models.

Table 2. The summary of models in the comparison experiments.

Model	Rating Matrix	Review Text	Explicit Method	Implicit Method
PMF	√	\	\	\
HFT	√	\	√	\
DeepCoNN	√	√	\	√
NARRE	√	√	\	√
SAFMR	√	√	\	√
HNNER-E	√	√	√	\
HNNER-I	√	√	\	√
HNNER	√	√	√	√

Note. "√" indicates that the corresponding attribute is used in the model, whereas "\" indicates that it is not included.

5.4. Implementation

For a fair performance comparison, all models were evaluated under an identical experimental setup, consisting of 128 GB of RAM and an NVIDIA V100 GPU (NVIDIA Corporation, Santa Clara, CA, USA). Each result was averaged over five runs to ensure robustness. Hyperparameters for HNNER were tuned based on performance on the validation set. In our implementation, we utilized a pretrained BERT-base model to extract contextual embeddings from review texts. Specifically, we used the [CLS] token representation as the sentence-level embedding for each review. The maximum input length was set to 512 tokens [9].

In our hyperparameter experiments, we fixed the embedding size to 64. The batch size was optimized among [64, 128, 256, 512, 1024] based on validation MAE. The learning rate was selected from among [0.001, 0.005, 0.0001, 0.0005, 0.00001, 0.00005] by monitoring convergence stability and final validation loss. The number of multi-heads was set to [2, 4, 6, 8, 10, 12], and the best-performing value was selected by comparing validation accuracy and computational cost. The dropout rate was optimized among [0.1, 0.2, 0.3, 0.4, 0.5], with the optimal value chosen based on generalization performance across epochs. Finally, the optimal hyperparameter values were batch sizes of 128 for the Musical Instruments and Video Games datasets and 256 for the Digital Music set, dropout rates of 0.1 for the Musical Instruments and Digital Music datasets and 0.2 for the Video Games set, and a learning rate of 0.001 and two multi-heads for all the datasets.

In addition, to set the number of topics in the LDA method, we set the maximum number of topics to 15 and selected the value that achieved the highest coherence score, which reflects the semantic consistency of discovered topics. Therefore, the optimal number of topics was set to 6, 7, and 12 for the Musical Instruments, Digital Music, and Video Games datasets, respectively.

To ensure a fair comparison, the hyperparameters of the baseline models were determined empirically. Each model was trained on the training dataset, hyperparameters were tuned on the validation set, and performance was evaluated on the test set. The tuned hyperparameters included the latent vector dimension, learning rate, dropout rate, and mini-batch size. For CNN-based models (i.e., DeepCoNN, NARRE, and SAFMR), CNN-specific parameters such as the number of kernels, window size, and number of channels were set according to the configurations reported in the original papers.

6. Experimental Results and Discussion

6.1. Performance Comparison with Baseline Models (RQ 1)

We evaluated the performance of the proposed HNNER model through comparisons with several baseline models on three Amazon product datasets. As shown in Table 3, the proposed model consistently demonstrated superior performance across all evaluation cases. To verify the statistical significance of the observed improvements, we conducted paired *t*-tests between the proposed model and each of the baseline models. The results confirmed that the performance gains achieved by HNNER are statistically significant ($p < 0.05$) across all datasets. Based on these results, we conducted a detailed analysis from the following three perspectives.

Table 3. Comparison of HNNER and baseline model performance.

Model	Musical Instruments		Digital Music		Video Games	
	MAE	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
PMF	0.721 ± 0.005	0.983 ± 0.001	0.434 ± 0.000	0.709 ± 0.000	0.848 ± 0.004	1.119 ± 0.007
HFT	0.676 ± 0.002	0.969 ± 0.007	0.394 ± 0.006	0.660 ± 0.004	0.847 ± 0.001	1.090 ± 0.000
DeepCoNN	0.734 ± 0.008	0.986 ± 0.009	0.423 ± 0.008	0.686 ± 0.005	0.882 ± 0.004	1.153 ± 0.002
NARRE	0.748 ± 0.008	0.948 ± 0.008	0.382 ± 0.004	0.635 ± 0.007	0.745 ± 0.005	1.015 ± 0.008
SAFMR	0.677 ± 0.008	0.890 ± 0.005	0.352 ± 0.006	0.608 ± 0.008	0.665 ± 0.001	0.951 ± 0.003
HNNER-E	0.570 ± 0.004	0.854 ± 0.009	0.325 ± 0.009	0.604 ± 0.004	0.651 ± 0.007	0.927 ± 0.003
HNNER-I	0.721 ± 0.005	0.983 ± 0.001	0.434 ± 0.000	0.709 ± 0.000	0.848 ± 0.004	1.119 ± 0.007
HNNER	0.676 ± 0.002	0.969 ± 0.007	0.394 ± 0.006	0.660 ± 0.004	0.847 ± 0.001	1.090 ± 0.000

First, the review-based models, including HFT, DeepCoNN, NARRE, and SAFMR, achieved better recommendation performance than PMF. The former group incorporates review text to predict user preferences, while the latter relies solely on rating data. Using ratings as the only information source is limited in its ability to comprehensively capture the underlying motivations of user purchasing behavior. In contrast, review texts contain rich semantic content that enhances the representation of user and item characteristics, thereby improving predictive performance.

Second, among the review-based models, those employing implicit methods (DeepCoNN, NARRE, and SAFMR) outperformed the explicit method (HFT). This is primarily because deep learning-based models can capture complex semantic structures and nonlinear relationships in review texts. Moreover, the application of regularization techniques such as dropout contributes to mitigating overfitting, further enhancing recommendation accuracy.

Finally, the proposed HNNER model outperformed all baseline models. Specifically, it achieved RMSE improvements ranging from 29.75% to 57.92% and MAE improvements ranging from 47.92% to 76.46% compared to PMF. These improvements suggest that incorporating review texts allows the model to capture nuanced aspects of user rating behavior that traditional rating-only models overlook. In comparison with the HFT model, HNNER achieved 17.11% to 23.37% improvement in RMSE and 27.24% to 35.40% in

MAE. Furthermore, compared to DeepCoNN, NARRE, and SAFMR, the proposed model demonstrated improvements of 11.80% to 17.24% in RMSE and 19.71% to 24.18% in MAE on average.

These results demonstrate that a recommendation model integrating both implicit and explicit methods can effectively capture the complementary information embedded in user-generated content. To validate the contribution of each component, we conducted ablation experiments by evaluating the effects of implicit and explicit methods separately while maintaining all other components of the HNNER model. The results revealed that the version of the model utilizing only implicit methods outperformed the one using only explicit methods. This confirms that deep learning-based models are more effective at capturing the deeper semantic features present in review texts.

6.2. Comparative Analysis of Training Efficiency (RQ2)

HNNER combines explicit and implicit representations through two feature learning modules, resulting in a relatively more complex model architecture. To evaluate the training efficiency of HNNER, we compared the per-epoch training time and GPU memory usage of HNNER with those of baseline models using the full training datasets. Among the baselines, we focused particularly on DeepCoNN and SAFMR, which demonstrated the strongest performance among deep learning-based methods. The results are summarized in Tables 4 and 5.

Table 4. Training time comparison to baseline models.

Model	Musical Instruments	Digital Music	Video Games
DeepCoNN	25 s	58 s	63 s
SAFMR	36 s	39 s	54 s
HNNER	19 s	25 s	28 s

Table 5. Model parameters and GPU memory comparison to baseline models.

Type	Model	Musical Instruments	Digital Music	Video Games
Parameter	DeepCoNN		330,207	
	SAFMR		11,681	
	HNNER		17,614,353	
GPU Memory	DeepCoNN	4170.93 MB	2067.46 MB	2475.30 MB
	SAFMR	2265.31 MB	2987.10 MB	5172.56 MB
	HNNER	2203.96 MB	3669.43 MB	3728.39 MB

Despite its larger parameter size, HNNER achieves superior training efficiency by leveraging a design that processes individual review texts without aggregation. This approach substantially reduces input redundancy and minimizes unnecessary computation. Furthermore, the model incorporates shared attention layers across feature modalities, contributing to more compact and computationally efficient learning without compromising representational depth.

As shown in Table 4, HNNER outperforms DeepCoNN and SAFMR with significantly reduced training times across all datasets, recording up to 56.9% and 47.2% faster training on the Digital Music and Video Games datasets, respectively. Moreover, Table 5 reveals that HNNER consumes the least GPU memory among all models, indicating efficient memory utilization despite its parameter scale. This contrast—between high parameter count

and low memory consumption—suggests that the model's architectural design effectively decouples representation richness from computational burden.

These empirical findings are corroborated by Figures 3 and 4, which visualize the convergence patterns and training dynamics. HNNER achieves rapid convergence and consistently yields the lowest training loss, MAE, and RMSE values across epochs, further confirming its ability to extract semantically rich and discriminative features from review texts with high computational efficiency.

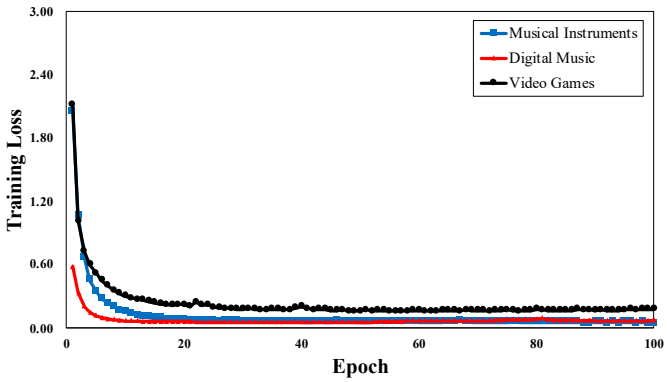

Figure 3. Training loss according to the number of epochs.

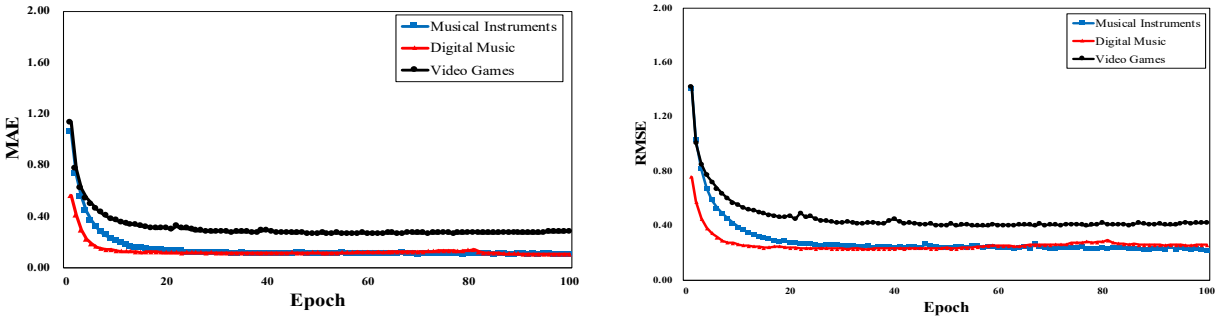

Figure 4. (**a**) MAE according to the number of epochs; (**b**) RMSE according to the number of epochs.

6.3. Comparative Analysis of Fusion Strategies (RQ3)

In this section, we empirically evaluate multiple fusion strategies to determine which method most effectively captures the complementary strengths of explicit and implicit features. We compare the following approaches: Add, Average, Concatenation, and Element-wise Product, as well as ablation variants of the proposed attention-based fusion, specifically excluding either the self-attention or co-attention component.

The results in Table 6 indicate that the co-attention fusion strategy achieves the best performance across all three datasets. This superiority is attributed to its ability to explicitly model the mutual dependencies between explicit and implicit features, thereby generating more discriminative fused representations.

Interestingly, the removal of either self-attention or co-attention leads to performance degradation, confirming that both mechanisms contribute complementary benefits. The exclusion of self-attention, which captures internal relationships within each modality, results in higher error values across two of the datasets. Likewise, removing co-attention, which facilitates cross-modal interaction, reduces the model's ability to integrate semantic alignment between features. These findings validate the architectural design choice of incorporating both attention mechanisms.

Table 6. Comparison of recommendation performance across fusion strategies.

Fusion Strategy	Musical Instruments		Digital Music		Video Games	
	MAE	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
Add	0.631	0.891	0.399	0.618	0.718	0.950
Average	0.667	0.899	0.373	0.613	0.701	0.943
Concatenation	0.587	0.867	0.352	0.606	0.692	0.937
Element-wise product	0.665	0.901	0.357	0.622	0.723	0.958
W/o self-attention	0.591	0.884	0.342	0.620	0.665	0.941
W/o co-attention	0.587	0.867	0.353	0.607	0.692	0.938
Co-attention fusion	0.570	0.8537	0.3249	0.604	0.651	0.926

In contrast, simpler fusion strategies, such as Add, Average, and Element-wise Product, perform consistently worse. These methods treat feature dimensions independently and fail to capture the rich inter-feature dependencies necessary for nuanced recommendation decisions. Based on these findings, we retain the full attention-based fusion scheme, comprising both self- and co-attention modules, in the proposed HNNER model.

6.4. Effect of Hyperparameter Settings (RQ4)

To validate the effectiveness of HNNER, we conducted experiments focusing on the impact of key hyperparameter settings that influence model performance. Specifically, we performed four additional experiments, each varying one of the following hyperparameters: the batch size, dropout rate, learning rate, and the number of attention heads (multi-head).

First, as shown in Table 7, the best performance was achieved with a batch size of 128 for the Musical Instruments and Video Games datasets, and 256 for Digital Music. These results highlight the importance of tuning batch size, as excessively large batches may hinder effective parameter updates, while moderately smaller batches tend to yield better generalization.

Table 7. Effect of batch size on recommendation performance.

Batch Size	Musical Instruments		Digital Music		Video Games	
	MAE	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
64	0.591 ± 0.006	0.858 ± 0.003	0.339 ± 0.007	0.610 ± 0.003	0.662 ± 0.004	0.929 ± 0.007
128	0.570 ± 0.002	0.854 ± 0.003	0.382 ± 0.007	0.609 ± 0.004	0.651 ± 0.002	0.927 ± 0.004
256	0.623 ± 0.008	0.860 ± 0.001	0.325 ± 0.007	0.604 ± 0.004	0.694 ± 0.002	0.928 ± 0.003
512	0.631 ± 0.003	0.857 ± 0.007	0.386 ± 0.006	0.612 ± 0.005	0.685 ± 0.001	0.936 ± 0.002
1024	0.609 ± 0.001	0.894 ± 0.005	0.342 ± 0.007	0.617 ± 0.008	0.731 ± 0.003	0.946 ± 0.001

Second, we examined the effect of the dropout rate on the performance of HNNER. As presented in Table 8, a dropout rate of 0.1 yielded the best performance on the Musical Instruments and Digital Music datasets. In contrast, a rate of 0.2 resulted in slightly superior outcomes on the Video Games dataset. These findings suggest that lower dropout rates contribute to better generalization in the context of our architecture.

In contrast, performance degradation was observed as the dropout rate increased beyond 0.3. For instance, in the Musical Instruments dataset, the MAE increased from 0.570 at a 0.1 dropout rate to 0.782 at a 0.4 dropout rate, reflecting a relative error increase of over 37%. Similar patterns were observed in the other datasets across both MAE and RMSE metrics. These results suggest that although dropout helps mitigate overfitting, excessively high dropout levels may impair the model's ability to retain salient feature representations,

ultimately leading to underfitting. This highlights the importance of precisely calibrating the dropout rate to strike a balance between regularization and representational capacity [4,36].

Table 8. Effect of dropout rate on recommendation performance.

Dropout Rate	Musical Instruments		Digital Music		Video Games	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
0.1	0.570 ± 0.002	0.854 ± 0.001	0.325 ± 0.003	0.604 ± 0.002	0.653 ± 0.008	0.929 ± 0.007
0.2	0.623 ± 0.008	0.864 ± 0.002	0.334 ± 0.007	0.610 ± 0.003	0.651 ± 0.008	0.927 ± 0.004
0.3	0.617 ± 0.002	0.879 ± 0.006	0.381 ± 0.007	0.620 ± 0.006	0.674 ± 0.008	0.950 ± 0.000
0.4	0.782 ± 0.005	1.030 ± 0.005	0.503 ± 0.003	0.739 ± 0.006	0.799 ± 0.002	1.040 ± 0.002
0.5	0.722 ± 0.006	0.995 ± 0.003	0.503 ± 0.003	0.739 ± 0.004	0.931 ± 0.008	1.128 ± 0.000

Third, we assessed the impact of learning rate settings on model performance. As shown in Table 9, a learning rate of 0.001 consistently achieved the best results across all datasets. This demonstrates the critical importance of selecting an appropriate learning rate to balance convergence speed and generalization. Improper learning rates may lead to underfitting or overfitting.

Table 9. Effect of learning rate on recommendation performance.

Learning Rate	Musical Instruments		Digital Music		Video Games	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
0.001	0.570 ± 0.001	0.854 ± 0.008	0.325 ± 0.004	0.604 ± 0.008	0.651 ± 0.004	0.927 ± 0.004
0.005	0.714 ± 0.007	0.978 ± 0.004	0.417 ± 0.008	0.654 ± 0.008	0.899 ± 0.003	1.160 ± 0.002
0.0001	0.666 ± 0.008	0.894 ± 0.006	0.819 ± 0.009	0.911 ± 0.001	0.658 ± 0.001	0.940 ± 0.008
0.0005	0.608 ± 0.007	0.861 ± 0.007	0.393 ± 0.009	0.616 ± 0.005	0.677 ± 0.007	0.938 ± 0.005
0.00001	0.982 ± 0.008	1.109 ± 0.003	0.921 ± 0.006	1.010 ± 0.001	0.718 ± 0.004	0.938 ± 0.007
0.00005	0.664 ± 0.008	0.868 ± 0.006	0.448 ± 0.005	0.629 ± 0.006	0.674 ± 0.006	0.938 ± 0.000

Finally, to evaluate the effect of the multi-head attention configuration, we varied the number of attention heads and analyzed its influence on performance. As shown in Table 10, the best results were consistently observed with two attention heads. While multiple attention heads can improve feature diversity, they may also introduce excessive transformation steps, increasing the risk of cumulative errors due to the discretization of attention distributions. This can negatively affect the model's ability to represent semantic relationships. Therefore, selecting an appropriate number of attention heads is crucial for maintaining model stability and effectiveness.

Table 10. Effect of multi-head attention head sizes on recommendation performance.

Multi-Head Attention Heads	Musical Instruments		Digital Music		Video Games	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
2	0.570 ± 0.003	0.854 ± 0.005	0.325 ± 0.008	0.604 ± 0.001	0.651 ± 0.004	0.927 ± 0.007
4	0.579 ± 0.007	0.858 ± 0.009	0.330 ± 0.006	0.608 ± 0.005	0.679 ± 0.007	0.930 ± 0.001
6	0.593 ± 0.007	0.860 ± 0.006	0.358 ± 0.008	0.608 ± 0.006	0.663 ± 0.004	0.928 ± 0.001
8	0.615 ± 0.003	0.868 ± 0.004	0.334 ± 0.006	0.608 ± 0.008	0.713 ± 0.004	0.930 ± 0.000
10	0.590 ± 0.003	0.903 ± 0.004	0.335 ± 0.008	0.608 ± 0.004	0.702 ± 0.004	0.928 ± 0.006
12	0.587 ± 0.009	0.863 ± 0.003	0.333 ± 0.007	0.607 ± 0.008	0.670 ± 0.006	0.930 ± 0.007

6.5. Case Study

Previous studies primarily used qualitative methods to assess the explanatory capability of recommendation models [37]. To more intuitively evaluate the explanatory power of the proposed model along these lines, we present generated topic descriptions of a few randomly selected items from our test dataset. Table 11 shows such an example. From this, we can draw observations on three aspects of the proposed model. The first column shows the image of the target item, followed by the images of two previously purchased items in the second column. The third column presents the user's review of the target item, and the fourth highlights the relevant topic description. In the Review Text column, bold italic text indicates the parts of the user review that align with the extracted topic description.

Table 11. Examples of the topic explanations, where each row represents the target item of a user.

No.	Target Item	Historical Records	Review Text	Topic Explanation
1				As a self-proclaimed cool dad with a lifelong love for video games, I was excited to dive into the Mario Wonder Nintendo Switch Game. . . . The game pack of Mario Wonder Switch is nothing short of magical.	Mario, Nintendo, Game Pack
2				This awesome white mouse is exactly what I wanted. . . . The honeycomb pattern not only lightens the weights, but provides an interesting texture and grip.	White, Honeycomb. Design, Mouse
3				Kid loves this toy trumpet and plays with it throughout the day. Good quality product and also satisfied with the fast and accurate delivery.	Design, Baby, Delivery, Trumpet
4				Wow. This is the best sports game I've ever played. Graphics are top notch and so are the controls. I've played Madden and have been an NFL fan for years. I have recently fallen in love with soccer. This game represents the sport well.	Sports, Game
5				This is my favorite controller of all the controllers I bought. The most beautiful blue I've ever seen. The color is so pretty. Good to catch.	Blue, Color Controller
6				This is only the third Red Dragon Gaming mouse I've had, but considering the last one I had lasted me about 6 years or so, these products will have quite a long life span to them. The mouse feels very comfortable to use and is great for playing games.	Brand, Redragon, Mouse

The proposed model can provide meaningful explanations because it is capable of extracting topics that represent user preferences from user reviews through Topic Modeling. For example, in Case 2, "Color" and "Style" were highlighted as key themes, whereas in the other cases, "Design" and "Delivery" emerged as key themes. These results are based on the items that users had purchased.

In Case 4, although there are many aspects between the target item and the user's past purchases, the core similarity of "Sports Game" was captured successfully in the topic description. This demonstrates that our model can accurately identify user preferences from review text.

Cases 2 and 6 illustrated that the proposed model emphasized different features (color and brand) of similar items (e.g., mouse) for each user. This reveals that the topic descriptions are personalized and demonstrates the effectiveness of the attention mechanism. This mechanism enables the model to more closely reflect individual preferences.

This case analysis reveals that the proposed model 1) can leverage review text to provide effective information regarding user preferences, which can be used to generate more accurate topic descriptions, and 2) works effectively by integrating explicit and implicit methods.

7. Conclusions and Future Studies

Numerous studies have addressed the data sparsity problem in recommender systems by extracting rich features from user reviews and incorporating them into model architectures. Review texts contain valuable user preference information related to item attributes, which not only clarifies the rationale behind recommendations but also enhances recommendation accuracy. Review-based recommender systems are typically classified into implicit and explicit methods based on how features are extracted from textual content. Both methods are effective and offer unique advantages. However, relatively few studies have explored the complementary use of both approaches within a unified framework.

To address this gap, we proposed a novel recommendation model, HNNER, which explicitly considers the complementarity between implicit and explicit methods. The model captures the importance of individual feature representations and the interdependence between them using self-attention and co-attention mechanisms. Experimental comparisons with baseline models confirmed the superior performance of the proposed approach. These findings emphasize the value of combining explicit and implicit representations for improving user preference prediction. Accordingly, this work offers a new direction for advancing research in recommender systems and demonstrates the practical effectiveness of the HNNER model.

Despite the promising results, this study has several limitations that suggest directions for future research. First, while the proposed model achieves interpretability and efficiency by leveraging LDA and BERT, it does not incorporate more recent advances in semantic modeling, such as BERTopic, RoBERTa, or GPT-based encoders. Future work should perform systematic comparisons with these models to better position the architectural choices within the current landscape of hybrid recommender systems. Second, the model operates solely on review texts and ratings, without integrating structured metadata (e.g., brand and price) or behavioral signals (e.g., clicks and browsing patterns). This modality limitation may restrict its generalizability across domains and reduce effectiveness in real-world applications. Exploring multi-modal fusion strategies could enhance contextual awareness and performance in heterogeneous environments. Third, the model's performance in cold-start scenarios remains untested. For users or items with limited historical data, reliance on review-based features may be insufficient. Incorporating auxiliary features or pre-trained user and item profiles could help extend applicability to sparse or zero-shot settings. Fourth, practical considerations such as memory usage and scalability were not evaluated in this study. While training time comparisons were included, future work should report GPU memory consumption, computational complexity (e.g., FLOPs), and inference latency to assess real-world deployment feasibility. Additionally, ethical implications such as bias

propagation from BERT embeddings and privacy risks from review parsing deserve further attention through techniques like debiased representations and differential privacy.

Author Contributions: Conceptualization, J.L. and Q.L.; methodology, E.J. and J.L.; software, E.J. and Q.L.; data curation, E.J. and Q.L.; writing—original draft, E.J. and J.L.; writing—review and editing, Q.L., S.-K.L. and J.L.; supervision, S.-K.L. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by Hansung University for Qinglong Li and Seok-Kee Lee. Also, the present research has been conducted by the Research Grant of Kwangwoon University in 2025 for Jiaen Li.

Data Availability Statement: The original data presented in the study are openly available in the Amazon product data repository at https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/ (accessed on 12 January 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jang, D.; Lee, S.-K.; Li, Q. ITS-Rec: A Sequential Recommendation Model Using Item Textual Information. *Electronics* **2025**, *14*, 1748. [CrossRef]
2. Kim, J.; Li, X.; Jin, L.; Li, Q.; Kim, J. Attentive Review Semantics-Aware Recommendation Model for Rating Prediction. *Electronics* **2024**, *13*, 2815. [CrossRef]
3. 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]
4. Zhu, Z.; Yan, M.; Deng, X.; Gao, M. Rating prediction of recommended item based on review deep learning and rating probability matrix factorization. *Electron. Commer. Res. Appl.* **2022**, *54*, 101160. [CrossRef]
5. 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]
6. Jang, D.; Li, Q.; Lee, C.; Kim, J. Attention-based multi attribute matrix factorization for enhanced recommendation performance. *Inf. Syst.* **2024**, *121*, 102334. [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, MA, USA, 6–10 February 2017; pp. 425–434.
8. Yang, S.; Li, Q.; Lim, H.; Kim, J. An attentive aspect-based recommendation model with deep neural network. *IEEE Access* **2024**, *12*, 5781–5791. [CrossRef]
9. Lim, H.; Li, Q.; Yang, S.; Kim, J. A BERT-Based Multi-Embedding Fusion Method Using Review Text for Recommendation. *Expert Syst.* **2025**, *42*, e70041. [CrossRef]
10. Li, Q.; Jang, D.; Kim, D.; Kim, J. Restaurant recommendation model using textual information to estimate consumer preference: Evidence from an online restaurant platform. *J. Hosp. Tour. Technol.* **2023**, *14*, 857–877. [CrossRef]
11. Li, Q.; Li, X.; Lee, B.; Kim, J. A hybrid CNN-based review helpfulness filtering model for improving e-commerce recommendation Service. *Appl. Sci.* **2021**, *11*, 8613. [CrossRef]
12. Duan, R.; Jiang, C.; Jain, H.K. Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem. *Decis. Support Syst.* **2022**, *156*, 113748. [CrossRef]
13. Kim, D.; Li, Q.; Jang, D.; Kim, J. AXCF: Aspect-based collaborative filtering for explainable recommendations. *Expert Syst.* **2024**, *41*, e13594. [CrossRef]
14. Li, L.; Dong, R.; Chen, L. Context-aware co-attention neural network for service recommendations. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW), Macao, China, 8–12 April 2019; pp. 201–208.
15. Cao, R.; Zhang, X.; Wang, H. A review semantics based model for rating prediction. *IEEE Access* **2019**, *8*, 4714–4723. [CrossRef]
16. 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.
17. Ren, G.; Diao, L.; Guo, F.; Hong, T. A co-attention based multi-modal fusion network for review helpfulness prediction. *Inf. Process. Manag.* **2024**, *61*, 103573. [CrossRef]
18. Ma, Y.; Xiang, Z.; Du, Q.; Fan, W. Effects of user-provided photos on hotel review helpfulness: An analytical approach with deep leaning. *Int. J. Hosp. Manag.* **2018**, *71*, 120–131. [CrossRef]
19. Liu, M.; Liu, L.; Cao, J.; Du, Q. Co-attention network with label embedding for text classification. *Neurocomputing* **2022**, *471*, 61–69. [CrossRef]

20. Li, Y.; Liu, T.; Hu, J.; Jiang, J. Topical co-attention networks for hashtag recommendation on microblogs. *Neurocomputing* **2019**, *331*, 356–365. [CrossRef]
21. 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.
22. Abdi, A.; Shamsuddin, S.M.; Hasan, S.; Piran, J. Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion. *Inf. Process. Manag.* **2019**, *56*, 1245–1259. [CrossRef]
23. Liu, J.; Wang, X.; Tan, Y.; Huang, L.; Wang, Y. An Attention-Based Multi-Representational Fusion Method for Social-Media-Based Text Classification. *Information* **2022**, *13*, 171. [CrossRef]
24. Bao, Y.; Fang, H.; Zhang, J. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014.
25. Kaliyar, R.K. A multi-layer bidirectional transformer encoder for pre-trained word embedding: A survey of bert. In Proceedings of the 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Qufu, China, 11–12 December 2020; pp. 336–340.
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. Kaviani, M.; Rahmani, H. Emhash: Hashtag recommendation using neural network based on bert embedding. In Proceedings of the 2020 6th International Conference on Web Research (ICWR), Bucharest, Romania, 23–24 July 2020; pp. 113–118.
28. Ray, B.; Garain, A.; Sarkar, R. An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews. *Appl. Soft Comput.* **2021**, *98*, 106935. [CrossRef]
29. Yang, N.; Jo, J.; Jeon, M.; Kim, W.; Kang, J. Semantic and explainable research-related recommendation system based on semi-supervised methodology using BERT and LDA models. *Expert Syst. Appl.* **2022**, *190*, 116209. [CrossRef]
30. Saleh, H.; Alhothali, A.; Moria, K. Detection of hate speech using BERT and hate speech word embedding with deep model. *Appl. Artif. Intell.* **2023**, *37*, 2166719. [CrossRef]
31. Lu, J.; Yang, J.; Batra, D.; Parikh, D. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2016; Volume 29.
32. 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]
33. Ni, J.; Li, J.; McAuley, J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 188–197.
34. Mnih, A.; Salakhutdinov, R.R. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2007; Volume 20.
35. Ma, H.; Liu, Q. In-depth Recommendation Model Based on Self-Attention Factorization. *KSII Trans. Internet Inf. Syst.* **2023**, *17*, 721–739.
36. Liu, D.; Li, J.; Du, B.; Chang, J.; Gao, R.; Wu, Y. A hybrid neural network approach to combine textual information and rating information for item recommendation. *Knowl. Inf. Syst.* **2021**, *63*, 621–646. [CrossRef]
37. Liu, P.; Zhang, L.; Gulla, J.A. Dynamic attention-based explainable recommendation with textual and visual fusion. *Inf. Process. Manag.* **2020**, *57*, 102099. [CrossRef]