electronics

MDPI

Article

ITS-Rec: A Sequential Recommendation Model Using Item Textual Information

Dongsoo Jang [1], Seok-Kee Lee [2] and Qinglong Li [2,*]

[1] Department of Big Data Analytics, Kyung Hee University, Seoul 02447, Republic of Korea
[2] Division of Computer Engineering, Hansung University, Seoul 02876, Republic of Korea
* Correspondence: leecy@hansung.ac.kr; Tel.: +82-2-760-4133

Abstract: As the e-commerce industry rapidly expands, the number of users and items continues to grow, making it increasingly difficult to capture users' purchasing patterns. Sequential recommendation models have emerged to address this issue by predicting the next item that a user is likely to purchase based on their historical behavior. However, most previous studies have focused primarily on modeling item sequences using item IDs without leveraging rich item-level information. To address this limitation, we propose a sequential recommendation model called ITS-Rec that incorporates various types of textual item information, including item titles, descriptions, and online reviews. By integrating these components into item representations, the model captures both detailed item characteristics and signals related to purchasing motivation. ITS-Rec is built on a self-attention-based architecture that enables the model to effectively learn both the long- and short-term user preferences. Experiments were conducted using real-world Amazon.com data, and the proposed model was compared to several state-of-the-art sequential recommendation models. The results demonstrate that ITS-Rec significantly outperforms the baseline models in terms of Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). Further analysis showed that online reviews contributed the most to performance gains among textual components. This study highlights the value of incorporating textual features into sequential recommendations and provides practical insights into enhancing recommendation performance through richer item representations.

Keywords: sequential recommendation; textual information; online reviews; self-attention mechanism; deep learning; item representation; e-commerce

Academic Editor: Cecilio Angulo

Received: 30 March 2025
Revised: 20 April 2025
Accepted: 23 April 2025
Published: 25 April 2025

Citation: Jang, D.; Lee, S.-K.; Li, Q. ITS-Rec: A Sequential Recommendation Model Using Item Textual Information. *Electronics* **2025**, *14*, 1748. https://doi.org/10.3390/electronics14091748

1. Introduction

The e-commerce industry has experienced rapid growth with the development of information and communication technology along with a continuous increase in the number of users and items [1,2]. The proportion of e-commerce transactions in the global retail market exceeds 20% [3]. Users can easily access item-related information through online platforms. However, a vast amount of information often leads to information overload. As a result, the user's purchase decision-making process becomes increasingly complex, which can lead to inefficiencies such as higher marketing costs [4,5]. Recommender systems have been actively studied as a solution to these challenges, as they provide users with personalized item suggestions based on their past behavior. Simultaneously, they support sellers by enhancing their sales performance and customer retention [6].

Collaborative filtering (CF) has traditionally been one of the most widely adopted approaches in recommender systems [4]. CF infers user preferences by analyzing the

similarity between users or items. Although effective in many cases, CF-based methods suffer from scalability and sparsity issues as the number of users and items increase. To overcome these challenges, matrix factorization (MF) was introduced, which learns latent features from user–item interaction matrices [7]. MF improves both scalability and accuracy compared to traditional CF but still models user–item interactions linearly. To capture more complex relationships, neural collaborative filtering (NCF) was proposed, which combines embedding layers with multilayer perceptrons to learn nonlinear interactions [8]. Further advancements such as neural graph collaborative filtering (NGCF) leverage graph structures to model higher-order connections among users and items [9]. These approaches have improved recommendation performance by enriching the representation of preferences and interactions.

However, these general recommendation models generally assume that user preferences are static and inferred from the entire history of interactions without considering temporal ordering. This study has several limitations. First, they treat all past interactions as equally important and overlook the potential relevance of recent purchases. Second, user preferences can change over time and are influenced by evolving interests and trends [10]. Third, the absence of order awareness in these models makes it difficult to provide timely and context-sensitive recommendations. To address these issues, sequential recommender systems have been developed that focus on learning temporal patterns in purchase behavior to predict the next likely item [11].

Sequential recommender systems treat a user's interaction history as a sequence and model the dependencies between items sequentially. This perspective aligns with natural language processing (NLP), in which sequences of words are learned to predict context or generate language. Deep learning techniques developed for NLP have been successfully adapted to sequential recommendations. For example, Hidasi et al. [12] introduced a session-based recommendation model using gated recurrent units (GRUs). Kang et al. [13] proposed a model based on the transformer architecture with self-attention, which effectively captures the relative importance of items in a sequence. Sun et al. [14] further extended this method to Bert4Rec by incorporating bidirectional encoding and masked learning to capture complex patterns. These studies highlight that the recommendation performance can be significantly improved by learning sequential patterns using techniques inspired by NLP.

Despite this progress, many sequential recommendation models rely solely on item IDs as inputs. This limits their ability to capture deeper semantic features or contextual information associated with the items. In reality, user decision-making is influenced not only by purchase history, but also by the content of items, including textual elements that describe product characteristics or convey user sentiment. Item-related texts (e.g., titles, descriptions, reviews) offer valuable information about an item's features, usage, and perceived value [15–17]. Prior research has shown that user-generated content plays an essential role in purchasing behavior; however, the use of such information in sequential recommendations remains limited [18,19].

The integration of textual information into sequential recommendations offers several advantages. First, item-related texts provide a rich semantic context that helps to disambiguate similar items and captures subtle differences that item IDs alone cannot represent. This is particularly effective in alleviating sparsity issues, particularly in cold-start or long-tail scenarios. Moreover, text embeddings can complement sequential patterns by introducing content-aware signals into the recommendation process [15–17]. While item sequences reflect behavioral dynamics over time, textual features offer insights into item properties and user sentiments, thereby enabling a more accurate understanding of user intent.

To address this gap, we propose an item textual information-based sequential recommendation (ITS-Rec) model that integrates textual information to enhance item representation. The model aims to capture user preferences more accurately by combining traditional sequence modeling with semantic item features. ITS-Rec consists of two main stages. First, item representations were generated using Bidirectional Encoder Representations from Transformers (BERT) to extract semantic information from product titles, descriptions, and reviews. Subsequently, a self-attention-based sequential learning module extended from SASRec [13] models user behavior over time. This combination enables ITS-Rec to reflect both the long-term item semantics and short-term user preferences. BERT provides rich contextual embeddings of the item content, whereas self-attention captures the temporal importance of previous interactions. Unlike previous models that rely only on item IDs, ITS-Rec leverages textual features to provide a more informative and context-aware representation. Experiments on the Amazon.com dataset demonstrate that ITS-Rec outperforms strong baselines, and ablation studies confirm the effectiveness of incorporating different types of textual information. The main contributions of this study are as follows:

- We propose ITS-Rec, which comprehensively integrates item-related textual data (e.g., titles, descriptions, and reviews) to model detailed item characteristics and user purchase motivation.
- We investigated the effect of each type of textual information within an attention-based sequential recommendation model, offering insights into the most effective data sources.
- Through experiments on real-world datasets, we demonstrated that ITS-Rec achieves superior performance compared with existing sequential recommendation models.

The remainder of this paper is structured as follows: Section 2 reviews the related work, Section 3 presents the proposed model, Section 4 details the experimental setup and results, and Section 5 concludes the study and discusses future work.

2. Related Works

2.1. General Recommender System

Early studies on recommender systems primarily utilized CF methodologies, which measure the similarity between users or items based on user purchase history to provide recommendations [20,21]. Sarwar et al. [22] first proposed an item-based neighborhood recommendation model that estimated user preferences by calculating item similarities based on user–item interactions. However, neighbor-based recommendation methodologies face limitations, as computational complexity increases rapidly when the number of users or items increases [4].

MF has been widely adopted because of its high performance and scalability. It utilizes latent factors derived from user–item interaction matrices [23,24]. Koren et al. [25] proposed an MF-based recommendation model using singular value decomposition, estimating user preferences through the inner product of user and item vectors. Bao et al. [26] enhanced the recommendation performance by incorporating topic information extracted from online reviews into an MF-based model. Similarly, Liu et al. [27] proposed a model that simultaneously captures horizontal and hierarchical patterns using heterogeneous information. Owing to its performance and scalability, MF has been the basis of many recommendation studies. However, MF has limitations in modeling complex user–item interactions, owing to its reliance on linear inner products.

To overcome such limitations, recent studies have applied deep learning techniques. He et al. [8] proposed the NCF model, which simultaneously captures both linear and nonlinear relationships between users and items using embedding vectors and a multilayer perceptron (MLP). This model demonstrated superior performance compared with

traditional MF methods, showing that nonlinear modeling is effective in improving recommendation accuracy.

For example, Covington et al. [28] introduced a deep learning-based YouTube recommendation model that jointly uses user behavior data and item features. Sedhain et al. [29] proposed a model that combines autoencoders with CF, enabling the estimation of specific user preferences by enhancing the interaction matrix.

Graph neural networks (GNNs) have recently attracted significant attention in the field of recommender systems owing to their ability to capture high-order connectivity between users and items [30]. By modeling user–item interactions as a graph structure, GNN-based models effectively encode collaborative signals and generate more expressive representations, thereby enabling more accurate preference estimation [31]. A representative model in this line of research, NGCF, was proposed by Wang et al. [9] NGCF integrates the GNN architecture into CF by iteratively propagating embeddings over the user–item graph, demonstrating improved performance through extensive empirical evaluation. Subsequently, LightGCN was introduced as a simplified yet powerful GNN-based recommendation model [32]. Unlike traditional GCNs, LightGCNs remove nonlinear activation functions and feature transformation layers, and rely solely on neighborhood aggregation to preserve pure collaborative signals. This simplification not only reduces the computational overhead but also improves the recommendation accuracy in both dense and sparse scenarios [32].

Traditional recommendation models such as MF, NGCF, and LightGCN have contributed to performance improvements by leveraging various modeling techniques. These approaches are effective for identifying relevant items for users during the purchase decision-making process [31]. However, one of their limitations is that they do not consider the temporal order of item interactions. Consequently, they may fail to capture dynamic user preferences or provide timely recommendations based on recent behavior [10]. Moreover, in the absence of sufficient user interaction history, general recommender systems often struggle to make meaningful predictions. To address these challenges, sequential recommendation models have been proposed that explicitly model the order of user–item interactions. These models have demonstrated promising results in capturing short-term preferences and adapting to evolving user needs, thereby overcoming the static nature of traditional recommendation frameworks [33].

2.2. Sequential Recommender System

Sequential recommender systems provide recommendations by considering the order of item purchases based on a user's past behavior. The objective is to predict the next item that the user is likely to be interested in by analyzing behavioral patterns.

General recommender systems assume that all past interactions are equally important for predicting user preferences. These models do not consider the order in which items are purchased, which makes it difficult to generate timely and context-aware recommendations. By contrast, sequential recommender systems can offer more relevant and timely suggestions by incorporating purchase orders. In other words, they provide recommendations by modeling the sequential nature of user purchase behavior and aim to predict the next item of interest.

Rendle et al. [34] proposed the Factorizing Personalized Markov Chains (FPMC) model, which learns user purchase sequences using a Markov chain framework. This model recommends items with a high probability of purchase by considering prior purchase transitions. Building on this idea, more recent studies have converted user–item interaction histories into sequence-structured data and applied advanced sequence modeling techniques, particularly inspired by NLP, to predict the next item in a user's sequence.

For example, Hidasi et al. [12] introduced GRU4Rec, a session-based recommendation model that employs a GRU to learn sequential dependencies within user click histories. This model was the first to successfully apply RNNs to the recommendation domain, enabling the prediction of the next likely item based on session-level behavior. Kang et al. [13] introduced SASRec, a self-attention-based sequential recommender system that dynamically weighs the importance of previous items in the sequence. This model demonstrated strong performance by effectively capturing long-range dependencies in user behavior. To address the limitations of unidirectional sequence modeling, Sun et al. [14] proposed BERT4Rec, which adopts bidirectional transformer encoding to capture complex item transition patterns more comprehensively.

Sequential recommender systems leverage the temporal order of user interactions and have become an important research direction [11]. They have demonstrated clear advantages in capturing evolving user preferences and overcoming the limitations of general recommendation models that ignore the order of interactions. In particular, sequence models based on RNNs and transformers have been widely adopted owing to the sequential nature of user behavior data. However, most existing models rely solely on item IDs to represent user histories, which can be problematic in large-scale environments, where the item space is extremely sparse. As the number of items on e-commerce platforms continues to grow, modeling user preferences based only on IDs has become increasingly insufficient. Although sequential dependencies can still be captured, the lack of semantic information limits the ability of models to generalize to unseen or infrequent items.

To address this limitation, this study proposes a sequential recommendation framework that integrates textual information relevant to user decision-making. Specifically, each item in the user sequence is represented by its ID and its associated textual features, including titles, descriptions, and user reviews. Textual semantics are extracted using BERT, whereas user behavior over time is modeled using a self-attention mechanism. This combination allows the model to reflect both the content of items and temporal dynamics of user preferences. By incorporating these rich semantic features, the proposed model aims to capture dynamic user interests more accurately and enhance the recommendation quality, particularly in sparse or cold-start scenarios.

3. Problem Definition

The recommendation model proposed in this study predicts the next purchased item based on the previous item purchase sequence of a user. The problem is formally defined as follows: Let $U = \left\{u_1, u_2, \cdots, u_{|U|}\right\}$ and $V = \left\{v_1, v_2, \cdots, v_{|V|}\right\}$ denote the sets of users and items, respectively, where $|U|$ and $|V|$ represent the total number of users and items. The item purchase sequence for user u is represented as $S_u = \left\{v_1^u, v_2^u, \cdots, v_{n_u}^u\right\}$, where v_t^u is the item interacted with by user u at time t, and n_u is the total number of items purchased by user u. Textual information embedding is denoted by $D = \left\{d_1, d_2, \cdots, d_{|V|}\right\}$, where d_i is a vector that integrates the item's title, description, and online reviews.

The proposed ITS-Rec model comprises two main stages, as illustrated in Figure 1. In the first stage, the title, description, and online reviews of each item, which may influence a user's purchase decision, were converted into vector representations using BERT, a pretrained NLP model. These three vectors were then integrated to construct the textual information embedding for each item. Textual embedding captures rich semantic features that reflect item properties, usage contexts, and sentiments, providing a more informative representation than item IDs alone.

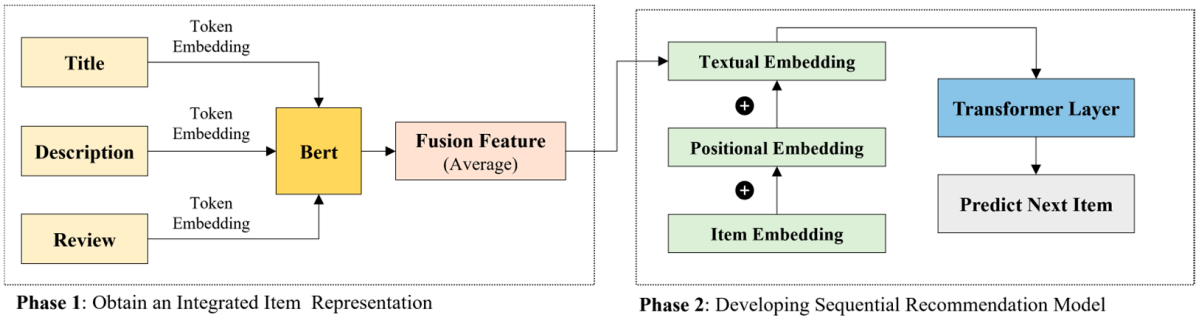

Figure 1. Overview architecture of proposed ITS-Rec model.

In the second stage, a self-attention mechanism was used to model user behavior over time, allowing the model to identify past items that are most relevant for predicting the next interaction. By combining BERT-based semantic encoding with self-attention-based sequence modeling, ITS-Rec effectively captures both long-term and short-term user intentions. The main distinction between the proposed model and previous sequential recommendation models is the use of integrated textual information, item ID, and positional embedding. This richer representation enables more effective modeling of items in sequential contexts. In summary, the proposed two-stage ITS-Rec model aims to capture user purchase patterns accurately and predict the next item to be purchased based on a sequence of previous interactions. A detailed explanation of the model architecture is provided in Section 4.

4. ITS-Rec Framework

The ITS-Rec framework is illustrated in Figure 2. First, item ID, positional information, and item textual information embeddings were input into the model. Using these embeddings, item sequence vectors were constructed for each user, and the model learned the item patterns in sequence using a self-attention mechanism. Based on this learning process, the proposed model predicts the next item to be purchased. The ITS-Rec model consists of three main components: an embedding layer, self-attention layer, and prediction layer.

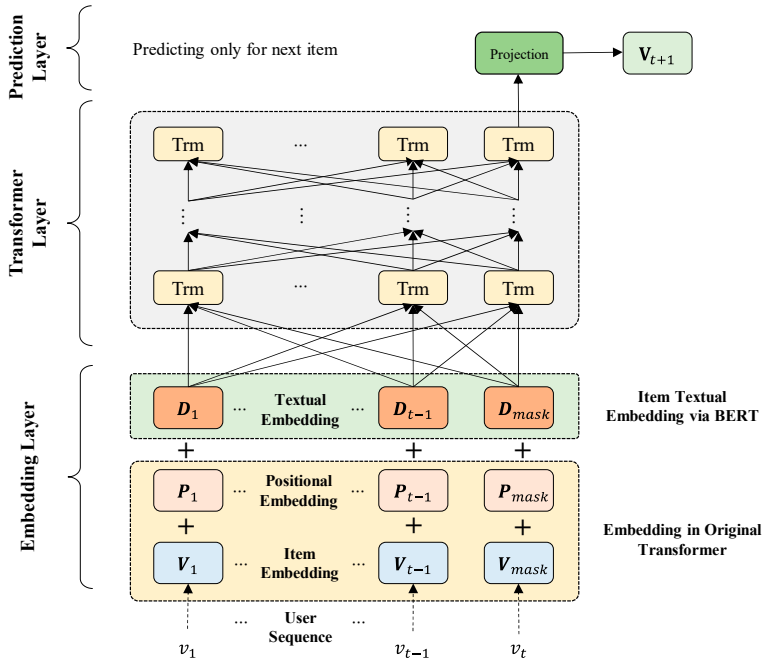

Figure 2. Framework of ITS-Rec model.

4.1. Embedding Layer

The proposed model predicts the next purchased item based on the item sequence with which the user has previously interacted. In the first step, the model generates item ID, positional, and textual information embeddings. Item ID embedding represents the unique identity of each item, whereas positional embedding captures the position of each item within the user's purchase sequence. To construct a rich item representation from textual content, embeddings were created for item titles, descriptions, and online reviews. These vectors are then integrated to form a unified textual information embedding that represents an item in detail. Consequently, the embedding layer generates three types of embeddings for each item in the sequence.

First, item ID embedding $e_v \in R^{|V| \times d}$ was generated by applying an embedding function to each item ID, where $|V|$ denotes the total number of items and d is the embedding dimension. Second, positional embedding $p_i \in R^{N \times d}$ was used to encode the position of each item in the sequence. This is necessary because the attention-based model lacks a recursive structure; N denotes the maximum sequence length considered in the model.

Next, the item titles, descriptions, and online reviews were vectorized to generate the textual representation of each item. To extract meaningful information, we utilized BERT, a pre-trained language model, to obtain token-level embeddings for each type of textual content. The resulting vectors for item titles, descriptions, and online reviews are denoted as $i_v \in R^{|V| \times m}$, $d_v \in R^{|V| \times m}$, and $r_{v,i} \in R^{|I| \times m}$, respectively, where m is the text embedding dimension and $|I|$ is the total number of reviews.

Because the number of online reviews per item varies, we computed the average of all review embeddings associated with each item to align the dimensionality with other textual features. This averaging process is expressed in Equation (1).

$$r_v = average([r_{v_1}; r_{v_2}; \cdots; r_{v_n}]), \tag{1}$$

where r is the average vector of online reviews generated through the interaction between the item and user, and v_n is the number of online reviews included for each item.

The final input embedding for each item at time t is obtained by combining the item ID embedding, positional embedding, and textual information embedding, as shown in Equation (2).

$$h_v^t = e_v + p_t + i_v + d_v + r_v. \tag{2}$$

The complete item sequence for each user is then represented as shown in Equation (3). If a user's sequence exceeds the maximum length N, only the most recent N items are retained. If the sequence is shorter than N, it is padded with zero vectors from the past.

$$\hat{E} = \begin{bmatrix} h_{s_1}^1, \\ h_{s_2}^2, \\ h_{s_3}^3, \\ \vdots \\ h_{s_N}^N \end{bmatrix}. \tag{3}$$

Each row of $\hat{E}$ is composed of the user's purchase item by time order, and it represents an input of the model to predict the user's next purchase item.

4.2. Self-Attention Layer

The attention mechanism is defined by Equation (4). Self-attention is a method in which the same vector is used as the query, key, and value [35].

$$c_j = f\left(w * k_j + b_j\right) \tag{4}$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d denotes the dimension used for scaling. The attention operation computes a weighted sum of values based on the relevance between the query and key. When applied to sequential recommendations, it determines the importance of each item in the sequence. Therefore, item sequence embeddings are passed through the self-attention operation.

$$\begin{gathered} W^Q, W^K, W^V \in R^{d \times d}, \\ SA(\hat{E}) = Attention(\hat{E}W^Q, \hat{E}W^K, \hat{E}W^V), \\ S = SA(\hat{E}), \end{gathered} \tag{5}$$

where W^Q, W^K, and W^V are the weight matrices for the query, key, and value transformations, respectively. The resulting S is an item sequence vector derived through self-attention.

This self-attention-based sequence vector effectively integrates the item information by reflecting the relative importance of each item. However, because the operation is linear, it may have limitations in capturing complex patterns within the user's item sequence. To address this, a pointwise feed-forward network (FFN) was applied to introduce nonlinearity, as shown in Equation (6).

$$F_i = FFN(S_i) = ReLU(S_i W^1 + b^1)W^2 + b^2, \tag{6}$$

The FFN performs two linear transformations with a nonlinear activation function in between. Here, W^1 and W^2 are weight matrices, b^1 and b^2 are bias terms, and F_i is the resulting item sequence representation that integrates contextual information across all positions.

To ensure stable training and prevent overfitting, the proposed model incorporates three regularization techniques: dropout, layer normalization, and residual connections [13]. First, dropout was applied to input $\hat{E}$, randomly excluding elements in the network with a given probability p to reduce overfitting during training [36]. This enhances generalization and mitigates performance degradation. Second, layer normalization is used to stabilize the training by normalizing each input vector independently of the other samples in the batch [37]. Unlike batch normalization, which uses statistics across batches, layer normalization computes the mean and variance from individual instances, allowing for more robust training in sequence models. Third, residual connections are employed to preserve the information from the earlier layers and prevent it from vanishing during training in deep networks [38]. Because items at later positions in the sequence often play a crucial role in next-item prediction, residual connections ensure that earlier learned representations are retained throughout the learning process. In summary, the proposed model performs item sequence learning by applying self-attention and FFN, supported by dropout, layer normalization, and residual connections, to enhance the training stability and model performance.

4.3. Prediction Layer

Based on F_t, which incorporates the item sequence embedding up to time t, the proposed model predicts the next item to be purchased at time $t+1$. To achieve this, the

model applies a matrix decomposition approach, as shown in Equation (7). The model calculates the relevance between F_t and each item embedding to identify the most relevant item at the next time step.

$$score_{i,t} = F_t N_i^T, \tag{7}$$

where N_i denotes the embedding matrix of item i, and $score_{i,t}$ indicates the level of relation between item i and the item sequence. It may be understood that an item with a high *score* relates highly with the item sequence F_t. Accordingly, the model can provide recommendations based on ranking the scores of the items. In the learning process of the deep learning model, the loss function utilizes binary cross-entropy, which is shown in Equation (8), and the optimization function operates the adaptive moment estimation (Adam) [39,40]. In addition, the proposed model selects a random negative item from each epoch and uses it for learning.

$$l = argmin\left(-\sum_{S^u \in S}\sum_{t \in [1,2\cdots n]}\left[\log(\sigma(r_{o_t})) + \sum_{j \notin S^u}\log(1-\sigma(r_{o_t}))\right]\right) \tag{8}$$

5. Experiments

5.1. Datasets

This study used data collected from Amazon.com to verify the performance of the proposed model [41]. The Amazon dataset was categorized by item type and included user purchase histories from May 1996 to July 2018. It contains various types of item-related textual information, such as item titles, descriptions, and online reviews. This study utilized the Movies and TV subset, a representative dataset frequently used in prior research, to examine the effect of textual information on sequential recommendation models [13,14,42]. The following data preprocessing steps were performed based on the strategy used in a previous study [14]. First, for textual information, spaces and non-English characters were removed. The remaining text was converted to lowercase and word-level tokenization was applied [43]. Second, the interactions between users and items are converted into implicit feedback [33]. Third, the users' purchase records were sorted in chronological order, and item sequences were generated for each user. To effectively evaluate the recommendation performance, only users with more than four purchase records were included, and data containing missing values were removed [14]. The basic statistics of the preprocessed data are summarized in Table 1.

Table 1. Statistics of dataset.

Feature	Statistics
Number of users	381,441
Number of items	126,300
Number of total interactions	3,462,063
Average sequence length per user	9.08
Max sequence length per user	3801
Average sequence length per item	27.41
Max sequence length per item	8896
Density (%)	0.007

5.2. Evaluation Metric

This study used a leave-one-out evaluation method to measure the performance of the sequential recommendation model [13,14]. Specifically, the most recently purchased item in each user's purchase history was used as the test data. Items purchased immediately before the test item were used as validation data, and the remaining purchase records were used as training data. A common evaluation strategy widely adopted in sequential recommendation studies was employed to ensure a fair and consistent performance comparison.

To reduce the computational cost, 100 negative items that each user had not purchased were randomly sampled. These items were ranked along with the actual purchased item (used as test data) according to the purchase probability predicted by the model. Therefore, model performance was evaluated by ranking 101 items, and the following evaluation metrics were used.

This study assessed recommendation performance using the Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) [12–14]. HR@K determines whether the actual purchased item appears among the top K items with the highest predicted probability of purchase. HR@K is equivalent to recall @K and is proportional in nature to precision @K [14]. HR@K is presented in Equation (9).

$$HR@K = \frac{1}{|U|} \sum_{u \in U} (R_{g,u} \leq R_K), \tag{9}$$

Here, U denotes the total number of users, $R_{g,u}$ is the ranking position of the ground-truth item purchased by user, and R_K is the top K ranking threshold used as a measurement criterion.

NDCG is a ranking-aware evaluation metric that considers the position of the actual purchased item in the recommendation list and assigns higher weights to items that are ranked higher. NDCG@K is presented in Equation (10). In this study, K values of 1, 5, 10, and 20 were used to obtain a range of experimental results.

$$NDCG@K = \frac{1}{|U|} \sum_{u \in U} \frac{(R_{g,u} \leq R_K)}{log_2(R_{g,u} + 1)}. \tag{10}$$

5.3. Baseline Model

This study aimed to address the limitations of previous sequential recommendation models that relied solely on item IDs. The proposed ITS-Rec model incorporates textual information containing detailed item characteristics and features that influence user purchase decisions. To verify the effectiveness of the proposed model, its recommendation performance was compared with representative sequential recommendation models based on recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer architectures. The baseline models used in this study are summarized as follows:

- POP: Basic recommendation model based on item popularity, such as sales volume or number of clicks. In this study, bestselling items were used to generate a recommendation list.
- BPR-MF [44]: A model that utilizes implicit feedback and provides recommendations by optimizing MF through Bayesian inference. Learning was conducted using a pairwise ranking loss function during the optimization process.
- GRU4Rec [12]: A recommendation model that learns user item sequences using an RNN. It employs a ranking-based loss function to train a model using sequential data.

- Caser [33]: A model that captures higher-order Markov chains by applying CNN operations to item sequence embeddings. Item embeddings were learned both horizontally and vertically using CNN layers.
- SASRec [13]: A high-performance sequential recommendation model based on transformer architecture. It uses self-attention to identify the importance of items in a sequence to predict the next item to be purchased.

5.4. Parameter Settings

The baseline models were implemented using the parameter settings presented in their original studies, whereas the proposed ITS-Rec model was fine-tuned through a parameter adjustment process to optimize performance.

For BPR-MF, the item embedding size was set to 50 with a dropout ratio of 0.5, and the Adam optimizer was used. GRU4Rec was configured with a learning rate of 0.01, embedding size of 50, L2 regularization of 0.01, and Adam optimizer. Caser used a learning rate of 0.001, dropout ratio of 0.5, L2 regularization of 0.01, and Adam optimizer. The model was trained to predict the next three items based on the previous five items. SASRec employed two self-attention layers and two FFN layers with the Adam optimizer, a learning rate of 0.001, and a dropout ratio of 0.5.

The proposed ITS-Rec model was designed with reference to the SASRec structure. It consisted of two self-attention layers and two FFN layers, and learnable positional embeddings were applied to enhance sequential understanding [13]. To process textual information, a pre-trained BERT-based model from the Hugging Face Transformers library was used. The BERT encoder was used in a frozen state, without fine-tuning, to reduce the training complexity and prevent overfitting on limited textual data. A bert-based uncased tokenizer was employed for tokenization. The maximum input length for each textual field (i.e., title, description, and review) was set to 100 tokens. Texts shorter than this limit were padded, and longer ones were truncated from the end.

All experiments were implemented using the PyTorch framework and conducted on a system equipped with an Intel Core i9-11900F CPU, 128 GB RAM (Intel Corporation, Santa Clara, CA, USA), and an NVIDIA GeForce RTX 3080 Ti GPU (Nvidia Corporation, Santa Clara, CA, USA). Under these settings, each training epoch consisted of 373 mini-batches and required approximately 16 min and 19 s (979 s). Given a batch size of 1024, this corresponds to processing 381,952 training samples per epoch, with an average throughput of approximately 390 samples per second. Although the integration of BERT increases the computational load compared with lightweight ID-based models, the overall training cost remains acceptable for practical applications.

6. Experiments

6.1. Performance Comparison to Baseline Models

Table 2 shows the performance comparison results between the ITS-Rec and baseline models.

Table 2. Performance comparison with baseline models.

Metric	POP	BPR-MF	GRU4 Rec	Caser	SASRec	ITS-Rec
HR@1	0.0000	0.3503	0.1474	0.2400	0.3196	**0.3551**
HR@5	0.0124	0.5341	0.2548	0.5564	0.6290	**0.6754**
HR@10	0.0244	0.6504	0.3359	0.6904	0.7543	**0.8177**
HR@20	0.0403	0.7576	0.4549	0.8047	0.8546	**0.8821**

Table 2. *Cont.*

Metric	POP	BPR-MF	GRU4 Rec	Caser	SASRec	ITS-Rec
NDCG@5	0.0069	0.3789	0.2019	0.4053	0.4826	**0.5250**
NDCG@10	0.0107	0.4235	0.2280	0.4488	0.5233	**0.5621**
NDCG@20	0.0148	0.4472	0.2578	0.4778	0.5487	**0.5857**

Note: Bold values indicate best performance for each metric.

First, POP, which recommends popular items, showed the lowest performance among all models. As a basic recommendation method, POP does not require complex operations and can quickly provide item suggestions to all users, even in the absence of user–item interactions. However, it provides the same items to all users based on popularity and cannot reflect individual user preferences. As a result, it performs poorly compared to sequential recommendation models that use personalized purchase history.

Second, RNN-based GRU4Rec can capture long-term dependencies in item sequences. However, its learning is limited to sparse data owing to the long-term dependency problem. In the Amazon.com dataset, the interaction matrix is sparse because the number of purchase records is relatively small. Consequently, GRU4Rec showed a lower performance than the other baselines, excluding POP. This result emphasizes the need to choose learning strategies suited for sparse e-commerce environments.

Third, BPR-MF uses Bayesian inference to optimize MF and predict the next likely item. This method has demonstrated good scalability and is widely used in general recommender systems. However, because it models item sequences linearly, it struggles to capture complex relationships, resulting in lower performance compared to Caser and SASRec in most metrics, except for HR@1.

Fourth, Caser captures higher-order Markov chains through CNN-based operations on item sequences. It effectively predicts the next item using limited historical data, making it suitable for sparse environments. In this experiment, Caser outperformed BPR-MF and GRU4Rec, which are limited by linearity and bottlenecks. However, Caser does not fully utilize long-term item dependencies, which limits its overall performance.

Fifth, SASRec uses a self-attention mechanism to weigh the importance of each item in a sequence, considering both recent and earlier purchases. This allows for the effective modeling of sequential patterns and enables SASRec to achieve the best performance among the baseline models. However, it relies only on item IDs to represent sequences, which limits its ability to capture detailed item features or to reflect the underlying context of user behavior.

Sixth, the proposed ITS-Rec achieved the best performance among all the models. The reasons for this are as follows: (1) Unlike POP and BPR-MF, ITS-Rec can learn complex sequential patterns using deep learning, thereby enabling more sophisticated personalization. (2) Compared with GRU4Rec and Caser, which are limited in capturing overall historical information, ITS-Rec leverages self-attention to integrate both long- and short-term patterns. (3) Unlike SASRec, which uses only item IDs, ITS-Rec incorporates rich textual information to represent item features and user behavior contexts. These findings demonstrate that using detailed textual features, including item descriptions and online reviews, helps the model better understand user preferences. Representing items with informative textual embeddings enables the model to capture user purchase patterns more accurately and improve recommendation performance.

6.2. Impact of Textual Information Types on Recommendation Performance

The proposed ITS-Rec model integrates item representations extracted from various textual sources and learns sequential patterns based on the detailed features obtained from these representations. Item titles, descriptions, and online reviews capture different aspects of an item, and their impact on recommendation performance may vary. To examine the effects of each type of textual information, an ablation study was conducted by systematically removing one component at a time. The results, shown in Table 3, present the performance for each feature combination in terms of HR@10 and NDCG@10.

Table 3. Effect of each feature combination on recommendation performance.

Feature Component	HR@10	NDCG@10
W/O Online Review (Item Title + Item Description)	0.7677	0.5441
W/O Item Description (Item Title + Review Text)	0.7778	0.5552
W/O Item Title (Item Description + Review Text)	0.7879	0.5596
All Features (Proposed Model)	0.7894	0.5621

First, when online reviews were excluded, recommendation performance decreased the most, indicating that they were the most influential feature among the textual components. Online reviews often contain user experiences, opinions, and purchase motivations, which are highly relevant to user preferences and decision-making. Therefore, incorporating online reviews helps the model better understand user intent and item characteristics, leading to improved performance. Second, item description contributed more to performance than item title. Although the title may capture the basic or high-level features of the item, it lacks the detailed context provided in the description. Because item descriptions typically include specific attributes and functionalities, they play a greater role in enhancing recommendation accuracy. In summary, online reviews were found to have the most significant impact on recommendation performance, followed by item descriptions and titles.

6.3. Impact of Review Selection Strategies on Performance

To generate item representations, the proposed ITS-Rec model calculates the average embedding of all online reviews for each item. Although this approach captures rich textual signals, it may also introduce noise from irrelevant or low-quality reviews, potentially degrading item embedding quality. To address this, we conducted additional experiments to assess the impact of various review selection strategies on recommendation performance with the aim of striking a balance between representational quality and computational efficiency.

Specifically, we compared the full review set with three filtered subsets: (1) the top-20 helpful reviews based on user votes, (2) the most recent 20 reviews, and (3) 20 randomly selected reviews. The results, summarized in Table 4, show that using all available reviews yielded the highest performance, as it provided the most comprehensive item-level information. However, among the selection strategies, helpful reviews achieved the best performance, confirming that user-evaluated quality serves as an effective proxy for noise reduction. Recent reviews have also outperformed random selections, suggesting that recency is a meaningful signal in dynamic environments where user preferences and product characteristics are constantly evolving.

Table 4. Effect of review selection strategies on recommendation performance.

Selection Method	HR@10	NDCG@10
Random reviews	0.7766	0.5482
Recent reviews	0.7786	0.5493
Helpful reviews	0.7798	0.5550
All reviews	0.7894	0.5621

The results suggest that, although it is preferable to utilize all reviews, filtering based on helpfulness or recency can be an effective alternative when resources are limited. They also demonstrated that our model is robust to variations in review selection and benefits from quality-aware filtering mechanisms.

6.4. Hyperparameter Analysis

The sequential recommendation model predicts the next item to be purchased based on a user's past purchase history. The number of past purchases used for prediction significantly influences the performance of the sequential recommendations. For example, RNN-based sequential recommendation models may exhibit degraded performance with long sequences owing to bottleneck problems. In contrast, the self-attention mechanism can effectively address this issue, and its effectiveness has been validated in previous studies [13,14]. The proposed ITS-Rec model adopts a self-attention-based approach. To evaluate the impact of sequence length, an experiment was conducted to compare the recommendation performance across different sequence lengths. Table 5 presents the results.

Table 5. Effect of maximum sequence lengths on recommendation performance.

Sequence Length	HR@10	NDCG@10
25	0.7654	0.5430
50	0.7747	0.5507
75	0.7791	0.5541
100	0.7894	0.5621

The experiment showed that shorter sequences resulted in lower performance, whereas longer sequences improved performance. This is because previously purchased items provide meaningful information to learn about user purchase patterns. However, overly long sequences may include outdated or irrelevant items that introduce noise. The proposed model alleviates this issue by applying a self-attention mechanism that assigns varying degrees of importance to each item in a sequence. This allows the model to appropriately reflect relevant information from long sequences, thereby enhancing performance.

The model also employs multi-head attention and an FFN to capture complex patterns in item sequences. These components enable learning through both linear and nonlinear operations, allowing the model to estimate purchase behavior more effectively. However, an excessively deep architecture can lead to overfitting, whereas insufficient depth may hinder learning. Therefore, it is necessary to optimize the number of multi-head attention and FFN layers. To identify the optimal configuration, an experiment was conducted to compare the recommendation performance across different layer depths.

The results shown in Table 6 indicate that using one or two layers yields the best performance. Because the proposed model simultaneously processes diverse item features, excessive computation can easily lead to overfitting. When three or more layers were

applied, the performance gradually declined, indicating that deeper architectures may negatively impact model generalization.

Table 6. Effect of number of multi-head attention and feed-forward network layers on recommendation performance.

The Number of Layers	HR@10	NDCG@10
1	0.7878	0.5665
2	0.7894	0.5621
3	0.7838	0.5535
4	0.7810	0.5446

7. Conclusions and Future Works

The scale of the e-commerce industry and the number of users and items continue to grow, making it increasingly difficult to estimate user purchase patterns. Item textual information can assist in capturing these patterns and enhancing the learning process in recommendation systems. However, many sequential recommendation studies have focused primarily on developing models that learn item sequences using NLP methodologies, often without incorporating rich item-related content. To address this limitation, this study proposes ITS-Rec, a model that utilizes various types of textual information to reflect detailed item characteristics and user purchasing motivations. By integrating this information into item representations, the proposed model achieved superior performance compared to various baseline models. The experimental results demonstrate the effectiveness of incorporating textual information into sequential recommendation tasks. This study provides meaningful insights into how such information can be used to improve recommendation accuracy in a rapidly growing e-commerce environment.

Several directions for future research are suggested based on the limitations of this study. First, this study was conducted using only the Amazon Movies and TV dataset, which may have led to domain-specific bias. Because product categories differ in review content, user behavior, and metadata, results from a single domain may not generalize well. Future work should evaluate the model using additional datasets to assess its robustness across different domains. Second, future studies should explore the integration of the proposed approach into more advanced sequential recommendation models. The ITS-Rec model is based on SASRec, which uses a self-attention mechanism. Recently, more sophisticated models, such as Bert4Rec and GNN-based recommender systems, have been introduced. Investigating whether item textual information can also enhance the performance of these state-of-the-art models would be a valuable next step. Third, it is necessary to consider additional types of item-related information beyond the text, such as images, prices, and vendor details. While this study focused on textual information to construct item representations, other modalities, such as visual data, can provide further complementary information. Many multimodal recommendation studies have demonstrated the benefits of combining diverse data types. Therefore, future studies could enhance the ITS-Rec model by integrating multiple modalities to represent items more comprehensively.

Author Contributions: Conceptualization, D.J. and Q.L.; Methodology, D.J. and Q.L.; Software, D.J. and Q.L.; Data curation, D.J. and Q.L.; Writing—Original Draft, D.J. and Q.L.; Writing—Review and Editing, S.-K.L. and Q.L.; Supervision, S.-K.L. and Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by Hansung University.

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 15 November 2023).

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

References

1. Jang, D.; Li, Q.; Lee, C.; Kim, J. Attention-based multi attribute matrix factorization for enhanced recommendation performance. *Inf. Syst.* **2024**, *121*, 102334. [CrossRef]
2. 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]
3. Dogan, O.; Kem, F.C.; Oztaysi, B. Fuzzy association rule mining approach to identify e-commerce product association considering sales amount. *Complex Intell. Syst.* **2022**, *8*, 1551–1560. [CrossRef]
4. Su, X.; Khoshgoftaar, T.M. A survey of collaborative filtering techniques. *Adv. Artif. Intell.* **2009**, *2009*, 421425. [CrossRef]
5. Haucap, J.; Heimeshoff, U. Google, Facebook, Amazon, eBay: Is the Internet driving competition or market monopolization? *Int. Econ. Econ. Policy* **2014**, *11*, 49–61. [CrossRef]
6. Grbovic, M.; Radosavljevic, V.; Djuric, N.; Bhamidipati, N.; Savla, J.; Bhagwan, V.; Sharp, D. E-commerce in your inbox: Product recommendations at scale. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 1809–1818.
7. Mnih, A.; Salakhutdinov, R.R. Probabilistic matrix factorization. *Adv. Neural Inf. Process. Syst.* **2007**, *20*.
8. 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.
9. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.-S. Neural graph collaborative filtering. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 165–174.
10. Wang, S.; Cao, L.; Wang, Y.; Sheng, Q.Z.; Orgun, M.A.; Lian, D. A survey on session-based recommender systems. *ACM Comput. Surv.* **2021**, *54*, 1–38. [CrossRef]
11. Quadrana, M.; Cremonesi, P.; Jannach, D. Sequence-aware recommender systems. *ACM Comput. Surv.* **2018**, *51*, 1–36. [CrossRef]
12. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. *arXiv* **2015**, arXiv:1511.06939.
13. Kang, W.-C.; McAuley, J. Self-attentive sequential recommendation. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 197–206.
14. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1441–1450.
15. Gao, R.; Li, J.; Li, X.; Song, C.; Zhou, Y. A personalized point-of-interest recommendation model via fusion of geo-social information. *Neurocomputing* **2018**, *273*, 159–170. [CrossRef]
16. Mazumdar, P.; Patra, B.K.; Babu, K.S.; Lock, R. Hidden location prediction using check-in patterns in location-based social networks. *Knowl. Inf. Syst.* **2018**, *57*, 571–601. [CrossRef]
17. Chen, H.; Qian, F.; Chen, J.; Zhao, S.; Zhang, Y. Attribute-based neural collaborative filtering. *Expert Syst. Appl.* **2021**, *185*, 115539. [CrossRef]
18. Yengikand, A.K.; Meghdadi, M.; Ahmadian, S. DHSIRS: A novel deep hybrid side information-based recommender system. *Multimed. Tools Appl.* **2023**, *82*, 34513–34539. [CrossRef]
19. Xie, Y.; Zhou, P.; Kim, S. Decoupled side information fusion for sequential recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1611–1621.
20. Kim, J.; Li, X.; Jin, L.; Li, Q.; Kim, J. Attentive Review Semantics-Aware Recommendation Model for Rating Prediction. *Electronics* **2024**, *13*, 2815. [CrossRef]
21. Li, Q.; Kim, J. A deep learning-based course recommender system for sustainable development in education. *Appl. Sci.* **2021**, *11*, 8993. [CrossRef]
22. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001; pp. 285–295.
23. Jamali, M.; Ester, M. A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; pp. 135–142.
24. Takács, G.; Pilászy, I.; Németh, B.; Tikk, D. Scalable collaborative filtering approaches for large recommender systems. *J. Mach. Learn. Res.* **2009**, *10*, 623–656.

25. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
26. 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.
27. Liu, T.; Wang, Z.; Tang, J.; Yang, S.; Huang, G.Y.; Liu, Z. Recommender systems with heterogeneous side information. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3027–3033.
28. Covington, P.; Adams, J.; Sargin, E. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 191–198.
29. Sedhain, S.; Menon, A.K.; Sanner, S.; Xie, L. Autorec: Autoencoders meet collaborative filtering. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 111–112.
30. Guo, Q.; Zhuang, F.; Qin, C.; Zhu, H.; Xie, X.; Xiong, H.; He, Q. A survey on knowledge graph-based recommender systems. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 3549–3568. [CrossRef]
31. Wu, S.; Sun, F.; Zhang, W.; Xie, X.; Cui, B. Graph neural networks in recommender systems: A survey. *ACM Comput. Surv.* **2022**, *55*, 1–37. [CrossRef]
32. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 639–648.
33. Tang, J.; Wang, K. Personalized top-n sequential recommendation via convolutional sequence embedding. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, 5–9 February 2018; pp. 565–573.
34. Rendle, S.; Freudenthaler, C.; Schmidt-Thieme, L. Factorizing personalized markov chains for next-basket recommendation. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 811–820.
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
36. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
37. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
39. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
40. Ruby, U.; Yendapalli, V. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends. Comput. Sci. Eng.* **2020**, *9*.
41. He, R.; McAuley, J. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Proceedings of the 25th International Conference on World Wide Web, Montreal, QC, Canada, 11–15 April 2016; pp. 507–517.
42. Islek, I.; Oguducu, S.G. A hierarchical recommendation system for E-commerce using online user reviews. *Electron. Commer. Res. Appl.* **2022**, *52*, 101131. [CrossRef]
43. Li, X.; Li, Q.; Kim, J. A Review Helpfulness Modeling Mechanism for Online E-commerce: Multi-Channel CNN End-to-End Approach. *Appl. Artif. Intell.* **2023**, *37*, 2166226. [CrossRef]
44. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.