sciendo

JAISCR, 2024, Vol. 14, No. 2, pp. 171 – 189
10.2478/jaiscr-2024-0010

DIAREC: DYNAMIC INTENTION-AWARE RECOMMENDATION WITH ATTENTION-BASED CONTEXT-AWARE ITEM ATTRIBUTES MODELING

Hadise Vaghari[1], Mehdi Hosseinzadeh Aghdam[2,*], Hojjat Emami[2]

[1]*Department of Computer Engineering, Qeshm branch, Islamic Azad University, Qeshm, Iran*

[2]*Department of Computer Engineering, University of Bonab, Bonab, Iran*

[*]*E-mail: mhaghdam@ubonab.ac.ir*

Submitted: 26th November 2023; Accepted: 12th February 2024

Abstract

Recommender systems (RSs) often focus on learning users' long-term preferences, while the sequential pattern of behavior is ignored. On the other hand, sequential RSs try to predict the next action by exploring relations between items in a user's last interactions but do not consider the general preference. Recently, the performance of RSs has increased by unifying these two types of paradigms. However, existing methods still have two limitations. First, the user's behavior uncertainty impedes precise learning of preferences. Second, being unable to understand the semantics of items makes the effect of the same item considered in the same way. These limitations jointly prevent RS from learning multifaceted preferences to capture the actual intentions of users. Existing methods have not properly addressed these problems since they ignore context-aware interactions between the user and item in terms of the links between the user and item attributes and sequential user actions over time. To address these challenges, this paper proposes a novel model, called the Dynamic Intention-Aware Recommendation with attention-based context-aware item attributes modeling (DIARec), which is capable of determining users' preferences based on their goal intention, taking into account the influence of various item features on user decision-making in their current context. Specifically, to model users' dynamic intentions, we introduce a dynamic intent-aware module to represent the hierarchical relations between items and their attributes in a given session. Experiments on benchmark datasets indicate that the proposed model DIARec outperforms other state-of-the-art methods.

Keywords: Unified recommender system, User intention, Context awareness, Attention mechanism, Collaborative projection, Item attribute relation

1 Introduction

Due to the explosion of information, people often face a large number of choices. Recommender systems (RSs) can assist users in the decision-making process to find the most relevant items of their interest [1, 2].

There are two types of paradigms based on modeling various aspects of users' behaviors: traditional RSs and sequential RSs [3, 4]. Traditional RSs aim to know what items users usually liked [5]. Collaborative filtering (CF) is a practical and efficient approach compared to other tradi-



Figure 1. Two challenges: behavior uncertainty and semantics of similar items, hindering the capture of users' actual intentions.

tional methods due to the recommendation of appropriate items without understanding their meanings [7]. Matrix factorization (MF) is one of the most successfully used techniques in CF model-based, which learns latent vectors of items and users to model the users' general preferences [36]. On the other hand, sequential RSs aim to uncover sequential patterns from past interactions [6]. Markov chain (MC) is one of the classic sequential recommendation methods, assuming that the next visited item depends on the previously seen items [7]. Recurrent Neural Networks (RNNs) can better capture item relationships within sequences of user interactions through a recurrent structure. However, due to the relatively high computational cost, the model is constrained to model short sequences only [8, 9]. Thus, both classes of paradigms mentioned above have strengths and shortcomings [4]. Although traditional RSs have been widely adopted to discover users' general preferences, they ignore the sequential information of users' behaviors. On the other hand, while sequential RSs capture relations between items by exploring patterns in the sequence, they lack the modeling of interactions between the user and the item, thus failing to extract general user interests. Therefore, traditional RSs are effective in extracting user long-term preferences, whereas sequential RSs excel at predicting the next user action given his/her last actions. Combining the advantages of these paradigms in building RSs provides a better recommendation.

Recently, some combined recommendation models with both traditional and sequential RSs have emerged. FPMC is a combination of MF and MC, using an individual transfer matrix for each user [11]. FPMC extracts both the general and sequential behavior of the user and then combines them linearly. HRM is another model that captures both sequential behavior and the general preference of the user to model complex interactions among these factors in the recommendation process, applying different types of aggregation operations, particularly nonlinear operations [4]. Especially, a new sequence model called Transformer has been successfully applied to machine translation tasks [10]. Inspired by the transformer approach, some models in SRSs have been suggested with different architectures such as SASRec [12], BERT4Rec [13], and MrTransformer [14]. The key to sequential RSs transformer-based models is a multi-head self-attention layer followed by a feed-forward layer. To maintain a constant number of parameters in the attention layer, the common approach is to scale the size of each head in the attention model by a factor of 1/(number of attention heads). However, in the traditional multi-head self-attention structure, each head independently performs its computations without interacting with or considering other heads. Meanwhile, most of them model the entire interaction sequences of users without distinguishing the long-term sequences from the short-term sequences.

Hence, the pattern of the user's decision is not fully exploited by the current methods, as they mostly fail to learn multifaceted preferences to capture the dynamic changes in user context, which

may shift over time for the following two reasons. First, the user's behavior pattern uncertainty impedes precise learning of preferences. The first row in Figure 1 illustrates a session created by user Bob, who has an intrinsic interest in comedy movies. Nevertheless, Bob's interests may vary depending on the specific context he finds himself in. In this session, Bob and his girlfriend have watched a romance movie (e.g., Titanic), which is outside his usual preferences. These random or spontaneous choices made by users introduce noise to their behavior, making it challenging for recommender systems to accurately learn their preferences and tastes.

Second, being unable to understand the semantics of items makes the effect of the same item considered in the same way. It's important to note that the meanings of items are not static; they dynamically change depending on the context. The second row in Figure 1 illustrates the semantic variability of pumpkins for the user Alice. As shown, Alice has purchased two sets of items, including {pumpkin, flour, sugar, egg} and {pumpkin, hanging ghost, lollipop, plastic blood bag}. In the first session, the pumpkin reflects baking sweets, where attributes like freshness and ripeness take center stage, whereas in the second session, it represents a Halloween party where attributes such as size and carving potential become paramount. Obviously, the semantic differentiation of items significantly reduces the complexity of learning user preferences, and the model becomes capable of understanding the primary intent of the user's choices. It is still unclear in prior studies how to combine multiple attribute relationships with consumed products for inferring the preferences of users in context items.

To address the aforementioned limitations, this paper introduces a novel recommendation framework called Dynamic Intention-Aware Recommendation with attention-based context-aware item attributes modeling (DIARec). DIARec leverages latent context factors and item attributes to capture the user's multi-faceted preferences. The framework comprises two main components: the Intention Learning Module and the Preference Module. In the Intention Learning Module, we begin by embedding items, users, multiple attribute relationships, and attribute weights into a sequence of low-dimensional vectors. To infer multifaceted preferences during the generation of a user's dynamic intention, we utilize both the item sequence and the attribute sequence. To facilitate this, we introduce the Intention-Aware Unit (IAU) module, designed to selectively focus on essential item attributes and fully exploit the varying informativeness of different items. Within the IAU module, a novel Multi-Head Self-Attention model is introduced, incorporating collaborative projection for interaction among different attention heads. This enhances the expressive power and representation of each attention head, ultimately improving overall model performance. In the Preference Learning Module, we integrate the IAU to identify the most relevant attributes of items concerning the user's primary intention. A time-aware attention mechanism is introduced to model the user's evolving preferences, and a Gated Recurrent Unit (GRU) is employed to capture sequential selection patterns among short-term interactive items, enabling a more comprehensive understanding of the user's interests. Finally, both long-term and short-term user preferences are integrated into a unified structure to learn the final user representation. The main contributions of this paper are as follows:

- This paper introduces a novel hierarchical attention-based model to learn the users' preferences and purposes.
- The proposed model utilizes a hierarchical attention network called IAU, which focuses on both items and their associated attributes. This unit examines the types of attributes that join interacted items to the current intention of the user and distinguishes which attributes have a greater impact within the user's context.
- In order to generate high-level user preference proposed a novel collaborative multi-head self-attention network that combines information from all the different attention heads.
- Experiments on two real-world datasets show the proposed model outperforms the other models.

The remainder of the paper is organized as follows: the related work is discussed in Section 2. Section 3 describes the details of the proposed model. Experiments are presented in Section 4. Finally, the conclusion is summarized in Section 5.

2 Background and Related Work

RSs are divided into two paradigms based on the modeling of users' preferences: traditional RS and sequential-aware RS. Both paradigms have their weaknesses and strengths.

2.1 Traditional Recommendation

The primary goal of traditional RSs is to model the general taste of the user by exploiting his/her historical interactions. Early studies on this type of paradigm of recommender systems often use Collaborative Filtering (CF) to model user preferences. Matrix factorization techniques [5, 15] are widely adopted approaches to CF model-based methods that, in general, aim to mine the user's general interest by converting the user-item rating matrix into two low-rank latent matrices. These latent vectors represent user preferences and item features in a common low-dimensional space, respectively, and are used to estimate the unknown ratings. Basically, MF-based methods deal with two different types of data: explicit and implicit feedback.

In explicit feedback-oriented methods, the recommendation problem is formulated as a rating prediction, and the user's ratings directly reflect his/her interest [16]. Nevertheless, explicit ratings of users are not often available, and they mainly interact with products via implicit feedback (e.g., mouse clicks). Although this general RS models the general factors of users' behaviors, its performance is limited without considering the sequential patterns.

2.2 Sequential Recommendation

The sequential pattern-based models capture the sequential behaviors of users in selecting items from historical interactions and provide recommendations by identifying the users' sequential patterns [7]. Although sequential pattern-based models generate recommendations based on changes in users' preferences, they fail when faced with large numbers of items or sparse data. Markov chain (MC) is a traditional sequential recommendation method to capture sequential signals from historical interactions [17]. However, Markov-based methods cannot capture all sequence information, have high computational complexity, and exhibit low prediction accuracy. On the other hand, recurrent neural networks (RNNs) have become effective approaches for modeling item sequences and generating sequential predictions. In [8], the authors suggested a session-based recommendation method called GRU4REC, which used RNNs with gated recurrent units (GRU) to learn the dynamics of user preference based on all previous items. Also, they extended a session-based RS with parallel RNN architectures to model sessions based on items' features that users interact with [18]. The authors proposed a data augmentation technique to enhance the performance of RNN-based methods by considering temporal shifts in the data [19]. In [20], the authors suggested a Time-LSTM framework for session-based recommendations that uses time gates to model the duration between sequential items. The authors proposed Caser, which used CNN to extract the user's preferences by embedding a sequence of recent items into an "image" for top-N sequential recommendations [9]. In [21], the authors suggested a hybrid approach that considers the short-term interest of users and collaborative neighborhood information to identify important items in interactions for session-based recommendations. Nevertheless, sequential RSs can only model sequential dependencies and neglect general preferences.

2.3 Unified Recommendation

Recently, there has been some research to build a unified system that combines both traditional and sequential RSs. In [22], the authors proposed a hierarchical RNN for session-based recommendation to personalize the current session by learning the short-term and long-term preferences of users. In [23], the authors suggested a Neural Attentive Recommendation Machine (NARM) model for session-based RS that combines vanilla attention with a GRU-based model. They apply the last hidden units of GRU as the representation of the sequential user behavior and then merge all hidden units to capture the primary purpose of the session. Finally, the two vectors are encoded to generate the user-to-item scores. The authors proposed a sequential hierarchical attention network for the next item recommendation, which uses two vanilla attention layers to learn users' long-term and short-term preferences [24]. In [25], the authors presented an attention mechanism that simultaneously extracts users' general preferences and recent in-

terests, relying on users' interest drift. The authors utilized a co-attention mechanism to extract dynamic interactions between the users' representations of long-term and short-term preferences [26]. In [12], the authors suggested a self-attention-based sequential recommendation method, which uses a self-attention mechanism to process long-range dependencies.

The authors proposed BERT4Rec to model user behavior sequences using deep bidirectional self-attention [13]. Most current unified recommendations ignore the purpose of users' selection behavior, resulting in reduced performance. In this paper, we introduce an Intention-Aware Unit (IAU) cell designed for use with both short and long-term interactive item sets. It is employed to extract the user's intention by considering the influence of various attribute information types on user preferences. We propose a novel multi-head attention network, a key component of our model, which effectively combines information from various attention heads, enhancing the model's ability to capture complex user preferences. The proposed model calculates attention weights for historical sessions to embed short-term preferences with temporal signals.

3 Methodology

In this section, we will first define and formulate the task of recommendation and then provide a detailed introduction to DIARec.

3.1 Overview

The framework of DIARec is illustrated in Figure 2. The method comprises two primary modules: the Intention Learning Module and the Preference Learning Module. The intention learning module aims to understand the user's primary intention in long-term sessions. It takes a sequence of user-item interactions, attributes relations, and weights of attributes in each session as input, converting them into a low-dimensional vector. The IAU component, equipped with a hierarchical attention network, captures the session's purpose from various semantic aspects. A Multi-head-self-attention network is utilized to discern the user's primary purpose in each session. The multi-head self-attention mechanism incorporates a collaborative structure to comprehensively capture various facets of user interests using information from all different attention heads, facilitating the identification of the main intention. The second module focuses on learning users' current preferences. Recognizing that users may exhibit diverse interests when selecting items, we introduce an IAU module to distinguish the informativeness of choosing the same item for different users in the last session. The time-aware attention mechanism is employed to capture the dynamic preferences of users in the current session. Additionally, GRU is utilized to retain the sequential selection pattern in the user's current preference.

Table 1. Notations.

Notation	Description
U, V	User and item set
A	Set of attributes relations
w	Set of attribute weight classes
S^u	Set of sessions of user u
V_a^u	Set of interaction items by user u with attribute type a
h	Number of heads
$long$	User's s long-term interest
$short$	User's s Short-term interest
I	Session representation
L	Number of long-term sessions
$e_j^v\ e_j^u\ e_j^a\ e_j^w$	Embedding Vectors including item, user,attribute, and attribute weight

3.2 Problem formulation and notations

This subsection formulates the task of recommending the next item as follows: Let $U = \{u_1, u_2, \ldots, u_{|u|}\}$ represents the set of users and $V = \{v_1, v_2, \ldots, v_{|v|}\}$ denote the set of the items, where $|U| = m$ and $|V| = n$ refers to the number of users and items, respectively. $A = \{a_1, a_2, \ldots, a_{|A|}\}$ denotes a set of attributes for all items, and $|A|$ is the number of all attributes. For each user $u \in U$, the historical interaction records of user u is denoted by $S^u = \{s_1^u, s_2^u, \ldots. s_T^u\}$, which T is the whole time steps. $s_t^u \in S^u (t \in [1, T])$ denotes the session at time step t for user u. Afterward, we associate each item v with its corresponding contextual relation $a \in A$. V_a^u represents the items that user u has consumed with an attribute relation type of a. We denote the attribute weight by w and divide w into five weight classes (w_1,...,w_5), where each class is calculated by computing the percentage per attribute

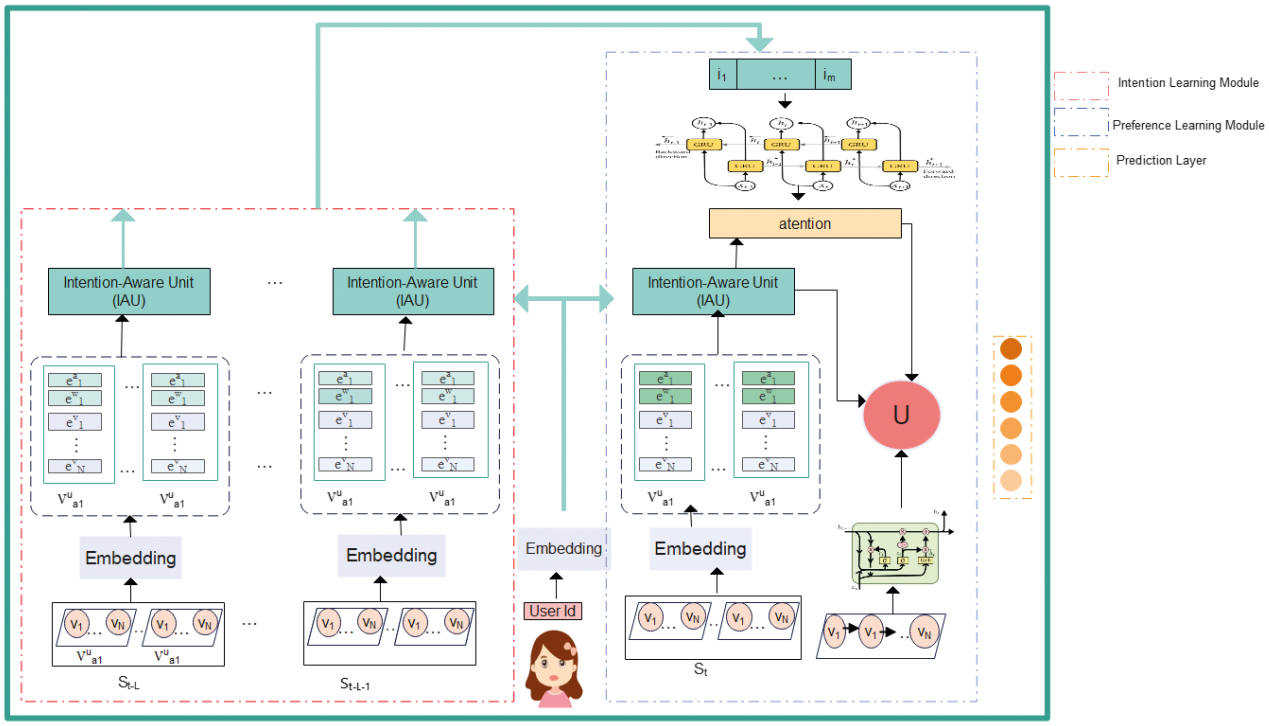


Figure 2. The architecture of the DIARec model consists of two primary modules purpose module and a preference learning module.

and mapped as follows: $\{0 \leq w_1 < 20, 20 \leq w_2 < 40, 40 \leq w_3 < 60, 60 \leq w_4 < 80, 80 \leq w_5 \leq 100\}$. The user's general or long-term preference at time step t is calculated based on the last L sessions $(s_{t-L}, \ldots, s_{t-2}, s_{t-1})$. Also, the user's short-term preference plays an important role in choosing the next item. In this paper, s_t^u shows the current session of user u at time step t to compute short-term preference. After learning short-term and long-term preferences of the user, the next item is predicted by computing the recommendation scores of all items. Main notation is summarized in Table 1.

3.3 Intention Learning Module

The intention learning module comprises two main components: (i) an embedding layer, and (ii) the intention-aware unit (IAU). Typically, the user's decision-making process involves two sequential steps: determining the user's primary intention and identifying their preferences. Generally, a user initiates their shopping journey with a specific intention and explores various products until they find one that aligns with their interests. In this module, our aim is to convert a session, consisting of a sequence of user-item interactions, item attribute relations, and attribute weights, into a sequence of low-dimensional vectors.

Subsequently, these vectors are fed into the IAU component, which aims to capture the hierarchical profiling of users at multiple levels of taste. To illustrate, consider two users, A and B, dining at the same restaurant on the same day. User A, a mother with young children, may choose the restaurant due to its child-friendly atmosphere and good service, despite disliking the food. On the other hand, user B, an older woman, may choose the same restaurant for its good food and service, despite disliking the child-friendly atmosphere. Despite their differing preferences, both users end up making the same choice. This discrepancy can be attributed to users assigning different weights to various attributes during their decision-making process. Building on these examples, our goal is to discern the diverse aspects of item attributes that signify the purchasing intent for each user. The IAU is employed to differentiate the importance of various attributes in elucidating the user's motivation for purchase.

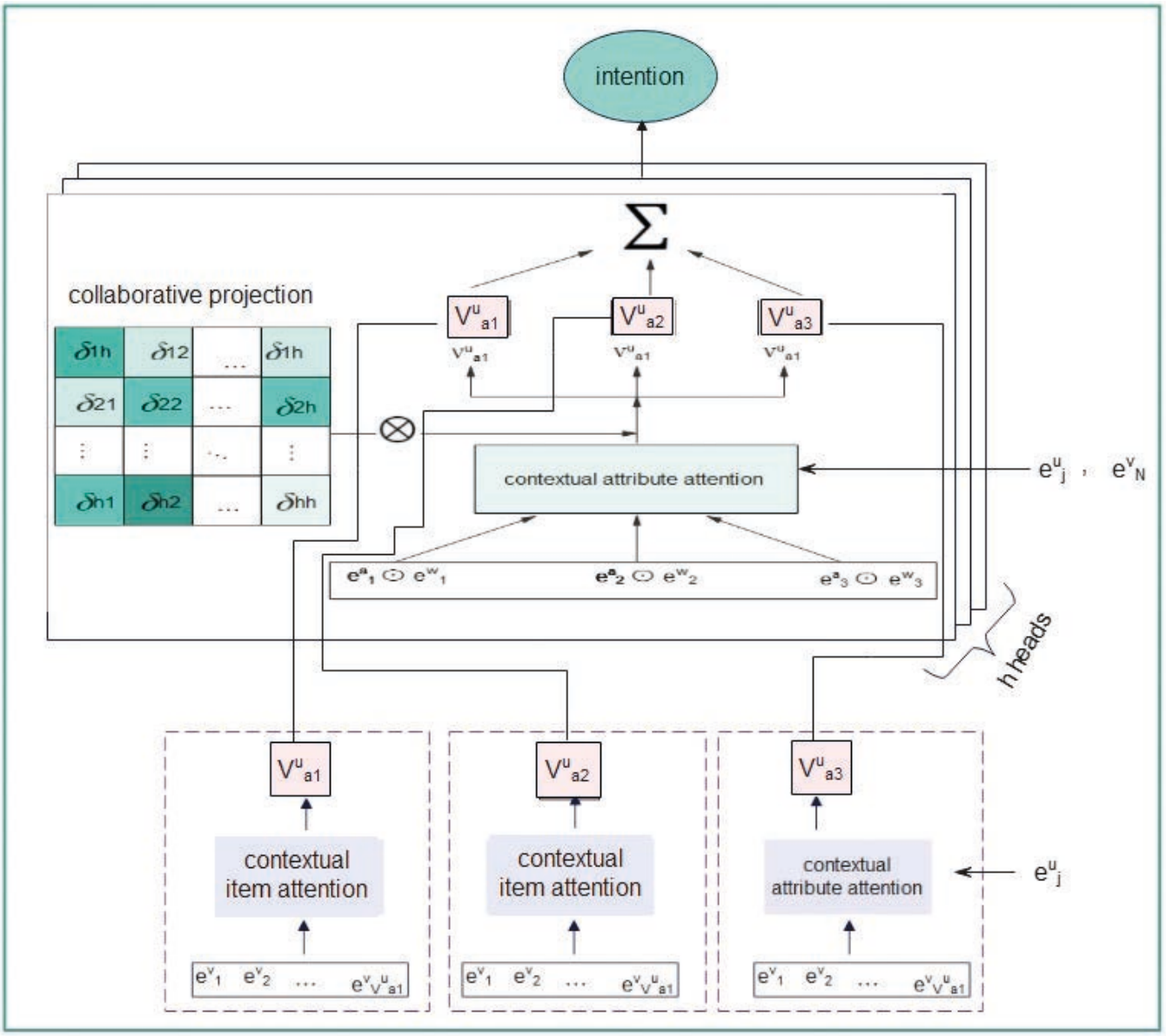


Figure 3. The details of the Intention-Aware Unit (IAU)

3.3.1 Embedding layer:

First, we embed the sparse user and item IDs (i.e., one-hot representation), contextual items, and weighted attributes into four low-dimensional dense vectors. e^v denotes the embedding vector for item i, u_j indicates the embedding vector for user j, and the embedding vectors for contextual items and weighted attributes are denoted by e^a and e^w, respectively. The items' embedding matrix is represented by $E = [e_1, e_2, \ldots, e_v] \in \mathbb{R}^{|V| \times |D|}$, where $| V |$ is the number of items, and D is the embedding dimension. The user embedding matrix is denoted by $U \in \mathbb{R}^{|D| \times |U|}$, and $W \in \mathbb{R}^{|D| \times |W|}$ and $A \in \mathbb{R}^{|D| \times |a|}$ are the weighted and attribute embedding matrices. The vectors u_j, e_j^u, e_j^w, and e_j^a represent the user embedding vector of user j, as well as the weighted and attribute vectors.

3.3.2 Intention-Aware Unit (IAU)

This module attends to the important attributes of items differently in each session based on the user's intention. In the real world, users assign different importance to the attributes of items based on their context. Therefore, identifying the contribution of attributes of each item for each user plays a significant role in personalized recommendation. For example, suppose the user likes horror and comedy movies; the latent representation of the user should indicate the user's preference for movies of both genres. At the time of prediction, comedy and horror movies get higher scores, but a specific genre (e.g., horror) may not have the same importance to the user in all movies. Therefore, the proposed model should be able to distinguish between the items of each genre and provide interpretable recommendations for each item. However, most existing approaches ignore the variety of item attributes and consequently overlook the user's purpose. To learn the informativeness of items, this paper proposes the IAU component shown in Figure 3. IAU comprises two attention cells: (1) the Contextual Item-Attention Cell and (2) the Contextual Attribute-Attention Cell, for identifying the most significant attributes of items relevant to the user's primary intention within long-term sessions.

1. Contextual Item-Attention Cell

This cell models various content-related or attribute-based relationships for context items separately. It's a straightforward fact that multiple-item attributes can explicitly reveal user interests. Therefore, this study utilizes item attributes as a factor to provide contextual information for creating a multi-faceted view of user interests in the proposed intention-aware module within the context-aware recommendation system.

First, we divide user-interacted items into various sets V_a^u according to the types of attribute relationships type a. Note that the same item may have many attribute relationships; hence, it may occur in various V_a^u. Since, at a given moment, an item can have various impacts on different users, we employ an attention network to learn weights that are guided by user embeddings in an Item Relevant-Attention Layer.

We map each set of attribute items in each session (V_a^u) as key-value, and the user embedding vector serves as the query. We compute weighted scores for each item as follows:

$$\theta_j = ReLu(W_1 e_j^v + W_2 e_j^u + b_1) \tag{1}$$

$$\beta_j = \frac{exp(\theta_j)}{\sum(exp(\theta_j)} \tag{2}$$

Where W_1, W_2, and b_1 are model parameters. Then, we normalize the scores using the softmax function and calculate each item's weight in the related V_a^u, which indicates the importance of each item. Finally, the weighted sum of all the item latent vectors is generated for each V_a^u using the normalized scores. Their corresponding attention scores are as follows:

$$V_a^u = \sum_{j=1....N} \beta_j e_j^v \tag{3}$$

2. Contextual Attribute-Attention Cell

Inspired by the method of [10], the suggested model utilizes a multi-head attention network to recognize various facets of users' interests. In order to uncover the high-level preferences of users, this study suggests a novel collaborative multi-head attention cell to identify the main users' purposes.

Since each user has various levels of interest in attribute relationships, to extract items' semantics, the multi-head self-attention is applied to utilize h separate self-attention models to jointly attend to

the information from several presentation subspaces at different positions. Then, we obtain h representations of user purposes in each session $[I^1, I^2, \ldots, I^h]$.

However, in the conventional Multi-Head Self-Attention model, each head independently performs its computations and operates separately without coordination with other heads. In other words, each attention head focuses on only a portion of the data and works independently of what other attention units are doing. This limitation can hinder the model's ability to understand complex relationships within the data and prevent the optimal utilization of information generated by other heads. To address these challenges, this study introduces a combined Multi-Head model that breaks free from isolation and allows information to be shared among the heads more effectively. This change enhances the model's capability to grasp intricate data relationships and make better use of the information generated by other heads. The final item can in some way have a substantial effect on the user's next choice and reflect some useful information. So, this paper applies the concatenation of $[e^u; e^v_N]$ as a query and the element-wise product of attribute relation embedding and attribute weight embedding as a key value. At last, the weighting scores of each attribute relation are computed as follows:

$$\begin{aligned} a_j^{*(1)} &= \mathrm{ReLu}(W_1^k[e^c \times e^w] + W_2^k[e^u; e_N^v] + b_1), \\ a_j^{*(2)} &= \mathrm{ReLu}(W_1^k[e^c \times e^w] + W_2^k[e^u; e_N^v] + b_1), \\ &\vdots \\ a_j^{*(h)} &= \mathrm{ReLu}(W_1^k[e^c \times e^w] + W_2^k[e^u; e_N^v] + b_1), \end{aligned} \tag{4}$$

In our approach, we introduce a collaborative projection that is applied to the attention-heads prior to the softmax function. This projection transforms the information and attention weights in a way that enables to movement of information between different attention heads.

$$\begin{pmatrix} a^{(1)} \\ a^{(2)} \\ \vdots \\ a^{(h)} \end{pmatrix} = \begin{pmatrix} \delta_{11} & \delta_{12} & \ldots & \delta_{1h} \\ \delta_{21} & \delta_{22} & \ldots & \delta_{2h} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{h1} & \delta_{h2} & \ldots & \delta_{hh} \end{pmatrix} \begin{pmatrix} a_j^{*(1)} \\ a_j^{*(2)} \\ \vdots \\ a_j^{*(h)} \end{pmatrix}, \tag{5}$$

Where $\delta(ij)$ denotes the attention mixture weight among the i-th attention head and the j-th attention head. $a^{(1)}, a^{(2)}, \ldots a^{(h)}$ These represent the attention matrices associated with attention heads. Then, we normalize the scores using the softmax function and calculate each item's weight in related V_a^u, which indicates the importance of each item.

$$\begin{aligned} \alpha^{(1)} &= \frac{exp(a_j)}{\sum(exp(a_j)}, \\ \alpha^{(2)} &= \frac{exp(a_j)}{\sum(exp(a_j)}, \ldots, \\ \alpha^{(h)} &= \frac{exp(a_j)}{\sum(exp(a_j)}, \end{aligned} \tag{6}$$

At last, the weighting scores of each attribute relation are computed as follows:

$$\begin{aligned} I^{(1)} = \sum_{j=1....|a|} \alpha^{(1)} V_{a_j}^u,\ I^{(2)} = \sum_{j=1....|a|} \alpha^{(2)} V_{a_j}^u, \ldots, \\ I^{(h)} = \sum_{j=1....|a|} \alpha^{(h)} V_{a_j}^u, \end{aligned} \tag{7}$$

We then combine all facets-level preferences with a trainable transformation matrix Wo, and compute the multi-head representation of long-term preferences as follows:

$$I = W^O Concat(I^1, I^2, \ldots, I^h) \tag{8}$$

3.4 Preference Learning Module

This module learns the user's current preference. As apparent from Figure **??**, IAU is also utilized in the preference learning module to learn a representation of the user's short-term preference. People may have different interests when choosing the same items, and various items may be more or less informative for user interest modeling. Therefore, IAU is used here as well to model the different informativeness of the same products for various users. Moreover, the user's interest may be hidden inside long-term sessions. The proposed model introduces a novel time-aware attention mechanism to compute attention weights for long-term sessions, embedding short-term preference with the time signal. Instead of using weights for all historical sessions, a bi-directional GRU [34, 35] is employed to

encode the temporal sequence of sessions and provide a fixed long-term preference representation per user.

$$\overrightarrow{h}_t = GRU(I_t), t \in [|S|-L, |S|-1] \quad (9)$$

$$\overleftarrow{h}_t = GRU(I_t), t \in [|S|-L, |S|-1] \quad (10)$$

$$h_j = [\overrightarrow{h}_t, \overleftarrow{h}_t] \quad (11)$$

In this way, we obtain h_j as an encoded temporal sequence of sessions representing the long-term preference for each user. We use IAU as a query in the current session and the fixed long-term preference representation (h_j) as key-values in this attention function.

$$m_j = ReLu(W^1 h_j + W^2 I) \quad (12)$$

Where W^1, and W^2 are model parameters. The weights of sessions are computed by normalizing the scores with a softmax function.

$$\eta_j = \frac{exp(m_j)}{\sum exp(m_j)} \quad (13)$$

At last, we calculate the weighted sum of the corresponding attention scores and obtain the short-term preference for each user.

$$I(long * short) = \sum_{j=1....L} \eta_j h_j \quad (14)$$

The selection of the next item may be influenced by a portion of the current session. Hence, GRU is used for its ability to extract the sequential pattern of the user in the current session. The update functions of GRU are as follows:

$$Z_t = sigmoid(W_z s_t + U_z h_{t-1} + b_z) \quad (15)$$

$$R_t = sigmoid(W_z s_t + U_z h_{t-1} + b_z) \quad (16)$$

$$h_t = tanh(W_h s_t + U_h(r_t \star ht - 1 + b) \quad (17)$$

$$h_t = (1 - Z_t) \star h_{t-1} + Z_t \star h_t) \quad (18)$$

Where R_t and Z_t are the reset and update gates, respectively. These gates extract the user's latest intent. The final hidden state $h_t \in R_d$ contains the items' transition relevance in the current session.

Finally, we concatenate preferences and get the final user representation $U = [I; I(\text{long*short}); h_t]$. U is then fed to a dense layer for feature abstraction, as follows:

$$U_j = ReLu(W^1 U + b^1) \quad (19)$$

Where U_j is U dense vector and W^1, b^1 denote the model parameters.

3.5 Prediction Layer

After computing the final representation of the user (U_j), the recommendation score is calculated by applying a dense layer softmax to get the output vector as:

$$\hat{y} = softmax(E) \quad (20)$$

Finally, we represent the loss function as the cross-entropy among the ground truth V_N and prediction $\hat{y}$, where y represents the one-hot coded vector of the ground truth item, and $\hat{y}$ is a predicted probability distribution.

$$Loss = -\sum_{i=1}^{|V|} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (21)$$

4 Experiments

This section performs experiments to answer the following four research questions:

Q1: How is the performance of our proposed method in the next item recommendation task compared to other methods?

Q2: Are the critical modules in DIARec (i.e., intention learning module and preference learning module) useful for improving recommendation performance?

Q3: How does the IAU component affect the recommendation? Does it help boost the performance of the recommendation?

Q4: What is the effect of the k different values on the performance of the top-K recommendation in the proposed architecture?

Q5: How is the efficiency of DIARec compared to other models?

4.1 Experimental setting

Experiments are performed on two datasets from different domains to evaluate the performance compared to other methods.

4.1.1 Datasets

Movielens: MovieLens[1] The MovieLens dataset, obtained from the MovieLens website, consists of movie rating data collected by the GroupLens research group. We specifically use the ML 100k version, which includes 100,000 ratings from 943 users on 1682 movies. To ensure an adequate amount of data for modeling, we filter out users and items with fewer than five interactions. For session-based analysis, we split all historical interactions of each user into sessions using a 30-minute inactivity window. To manage session length, we set a maximum limit of 50 interactions per session. Additionally, we filter out users with fewer than L+2 sessions to ensure there is sufficient cross-session data for effective modeling of returning users.

Yelp: [2] The Yelp dataset, sourced from the Yelp challenge dataset, is utilized as another benchmark for recommendation tasks. For a focused analysis on the restaurant business, we specifically retain data related to the food category. Similar to the preprocessing steps applied to the MovieLens dataset, we filter out users and items with fewer than five interactions. The historical interactions of each user are organized into sessions using a 1-day inactivity window. To manage session length, a maximum limit of 50 interactions per session is set. Additionally, we filter out users with fewer than L+2 sessions to ensure an adequate amount of cross-session data for modeling. The statistics for both datasets are shown in Table2.

Train/validation/test splits: For each user, the dataset is split into training and test sets. The first 80% of a user's sessions are designated as the training set, and the remaining 20% are used as the test set. A sliding window approach is employed with a window size of L to split each user's sessions into subsequences of fixed length L+1. For instance, given a user's sessions sequence $s = \{\{v_1, v_2\}, \{v_3, v_4\}, \{v_5, v_6\}, \{v_7, v_8\}, \{v_9, v_{10}\}\}$, three instances can be extracted with a window size of 2. In each instance, the first L sessions represent long-term preferences, the last session represents short-term preferences, and the last item is treated as the ground truth. Instances 1 and 2, representing both long and short-term preferences, are derived from the first 80% of the training set. The remaining instances for testing are formulated accordingly as follows:
train set=$\{\{v_1, v_2\}, \{v_3, v_4\}, \{v_5, v_6\}, \{v_7, v_8\}\}$
test set=$\{\{v_9, v_{10}\}\}$
#instance1:$Long_p = \{\{v_1, v_2\}, \{v_3, v_4\}\}$
$short_p = \{\{v_5\}\}$ $groundtruth = v_6$.
#instance2:$Long_p = \{\{v_3, v_4\}, \{v_5, v_6\}\}$
$short_p = \{\{v_7\}\}$ $groundtruth = v_8$.
#instance3:$Long_p = \{\{v_5, v_6\}, \{v_7, v_8\}\}$
$short_p = \{\{v_9\}\}$ $groundtruth = v_{10}$.

4.1.2 Baselines

In order to demonstrate the superiority of our proposed method DIARec, we compare it with the following representative baselines:

- **POP**: A simple algorithm that ranks items according to their popularity which popularity is the number of occurrences of item interactions.
- **BPR-MF** [27]: This model is a classic baseline for binary implicit feedback.
- **Item-KNN** [28]: An item-KNN model is based on the cosine similarity to make recommendations to the user.
- **GRU4** [18]This is a baseline which applies RNNs to be able to model the whole session.
- **BERT4G** [13] It utilizes a bidirectional self-attention mechanism for the next-item recommendations.
- **SASRec** [12] Utilizes a self-attention mechanism to extract users' long-term sequential behavior.
- **FISSA** [29] FISSA employs the SASRec model to obtain user interest representations. It uses a transformer-based layer to incorporate item similarity for modeling the global user interest representation.

[1] http://grouplens.org/datasets/movielens/
[2] https://www.yelp.com/dataset

Table 2. Statistics of our datasets.

Datasets	Users	Items	Action	Attributes
Movielens	943	1682	92,287	Drama, Action, Comedy, Romance, Thriller, Crime Adventure
Yelp	30,431	20,033	316,354	Restaurants Delivery, Business Accepts Credit Cards, Outdoor Seating, Restaurants Reservations, Restaurants TakeOut Restaurants Good For Groups, Good For Kids

Table 3. The performance comparison of DIARec with other methods on Movielens and Yelp

Dataset	Metric	POP	BPR-MF	Item-KNN	GRU4	SASRec	SSE-PT	FISSA	BERT4Rec	S3-rec	DIARec
Yelp	Recall	0.1870	0.1970	0.2278	0.72148	0.7384	0.7557	0.7608	0.7621	0.7982	0.8251
	NDCG	0.0677	0.0691	0.0777	0.3632	0.3789	0.38039	0.39308	0.4048	0.4123	0.4488
Movielens	Recall	0.2396	0.4655	0.4835	0.6348	0.6423	0.6475	0.6488	0.6506	0.6755	0.6998
	NDCG	0.0996	0.1992	0.2032	0.2718	0.2801	0.2885	0.3009	0.3133	0.3309	0.3789

– **SSE-PT** [30] SASRec is an impersonal model because it lacks the user's personalized embeddings.SSE-PT enhances SASRec by suggesting embeddings shared stochastic.

– **S3Rec** [31] Utilizes self-supervised learning to derive intrinsic data correlation to learn data representations.

4.2 Evaluation Metrics

In this article, we utilize Recall and Normalized Discounted Cumulative Gain (NDCG), two widely adopted ranking metrics in top-k recommendation, to assess the recommendation lists generated by all models [32, 33]. Higher values indicate better model performance.

Recall@K: measures the ratio of ground-truth items in the top-K positions of the recommendations list for users. The recall@K value is calculated as:

$$Recall@K = \frac{n_{hit}}{N} \tag{22}$$

Where n_{hit} is the number of testing instances in the top-K positions of the recommendation list, and N is the total number of testing instances.

NDCG@K: evaluates the ranking quality of recommendation results by considering the position of correct items. The $NDCG@K$ is calculated as:

$$NDCG@K = \frac{1}{G}\sum_{i}\left(\frac{2^{r_i-1}}{\log(i+1)}\right) \tag{23}$$

Where G refers to the ground-truth set of test instances, and r_i denotes the rank of item i in the testing set among the higher positioned items in the recommendation list. Higher values for both metrics indicate better model performance.

4.2.1 Parameter Settings and Implementation Detail

We implement our proposed RS with Keras and conduct all experiments using the Google Colab platform. The parameters used are updated with an Adam optimizer. For both datasets, we set the dimensionality of the embedding vectors to $d = 16$ and the window size to $L = 5$. The number of heads in the multi-head self-attention is set to 3. All weighting parameters in the model are initialized using a normal distribution $N(0.01)$. The learning rate and batch size are set to 0.01 and 32, respectively. We utilize dropout with a dropout ratio of 0.5 to prevent overfitting, and the number of epochs is set to 10.

4.3 Comparison with baseline methods (RQ1)

This subsection compares DIARec with other methods on Movielens 100k and Yelp datasets, and the results are summarized in Table 3.

1. As is apparent from Table 3, where the higher performance in each row is denoted in bold, the proposed approach outperforms both the Movielens and Yelp datasets, significantly excelling in NDCG@K and Recall@K. Specifically, compared to the best state-of-the-art method in terms

Table 4. Impact DIARec Modules on Yelp

Yelp		
Methods	Recall@20	NDCG@20
Pur-DIARec	0.7789	0.3745
Pre-DIARec	0.7543	0.3438
DIARec	0.8251	0.4488

Table 5. Impact DIARec Modules on Movielens

Movielens		
Methods	Recall@20	NDCG@20
Pur-DIARec	0.6243	0.3274
Pre-DIARec	0.6021	0.3045
DIARec	0.6998	0.3789

Table 6. Impact of attention mechanisms in AIU Module on Movielens and Yelp

	Movielens		Yelp	
Methods	Recall	NDCG	Recall	NDCG
DIARec multi-attention	0.6803	0.3489	0.8099	0.4023
DIARec collaborative-multi-attention	0.6998	0.3789	0.8251	0.4488

Table 7. The results of Recall@k, NDCG@k with different K.

		@5		@10		@20	
Datasets	Methods	Recall	NDCG	Recall	NDCG	Recall	NDCG
Movielens	POP	0.0965	0.0602	0.1431	0.0755	0.2396	0.0996
	BPR-MF	0.1866	0.1203	0.3138	0.1610	0.4655	0.1992
	Item-KNN	0.1845	0.1176	0.3276	0.1642	0.4835	0.2032
	SASRec	0.2948	0.2035	0.4746	0.2655	0.6548	0.3088
	SSE-PT	0.3017	0.2106	0.4688	0.2599	0.6723	0.2993
	FISSA	0.2585	0.1624	0.4492	0.2230	0.6409	0.2718
	BERT4Rec	0.3003	0.2028	0.4662	0.2562	0.6506	0.3033
	S3-rec	0.3012	0.1881	0.4761	0.2448	0.6755	0.2985
	DIARec	**0.4676**	**0.2832**	**0.5688**	**0.3254**	**0.6998**	**0.3789**
Yelp	POP	0.0474	0.0288	0.0960	0.0443	0.1897	0.0677
	BPR-MF	0.0500	0.0291	0.0983	0.0447	0.1970	0.0691
	Item-KNN	0.248	0.203	0.476	0.265	0.634	0.303
	SASRec	0.3884	0.2663	0.5609	0.3224	0.7684	0.3789
	SSE-PT	0.3884	0.2663	0.5609	0.3224	0.7684	0.3789
	FISSA	0.4520	0.3140	0.6153	0.3631	0.7757	0.4089
	BERT4Rec	0.4561	0.3163	0.6138	0.3673	0.7621	0.4048
	S3-rec	0.4224	0.2880	0.5926	0.3430	0.7771	0.3860
	DIARec	**0.4689**	**0.427**	**0.6378**	**0.4312**	**0.8251**	**0.4488**

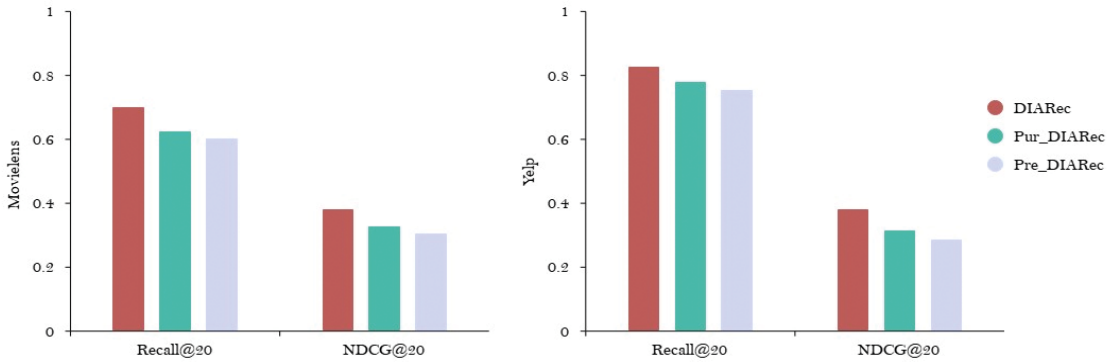


Figure 4. Impact DIARec Modules on Yelp and Movielens.

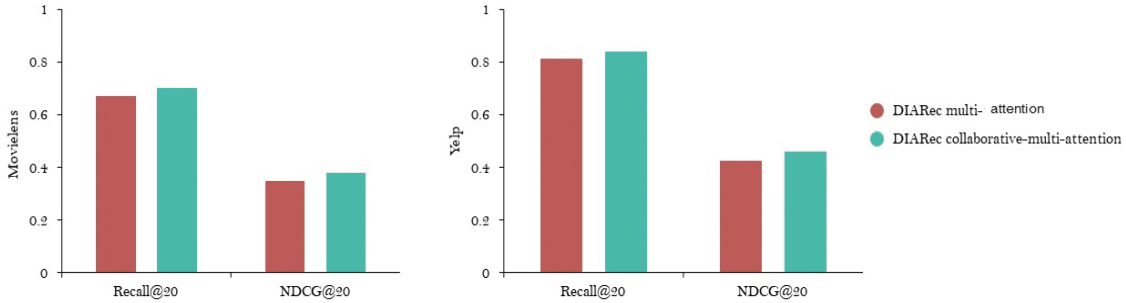


Figure 5. Impact of attention mechanisms in AIU Module on Movielens and Yelp.

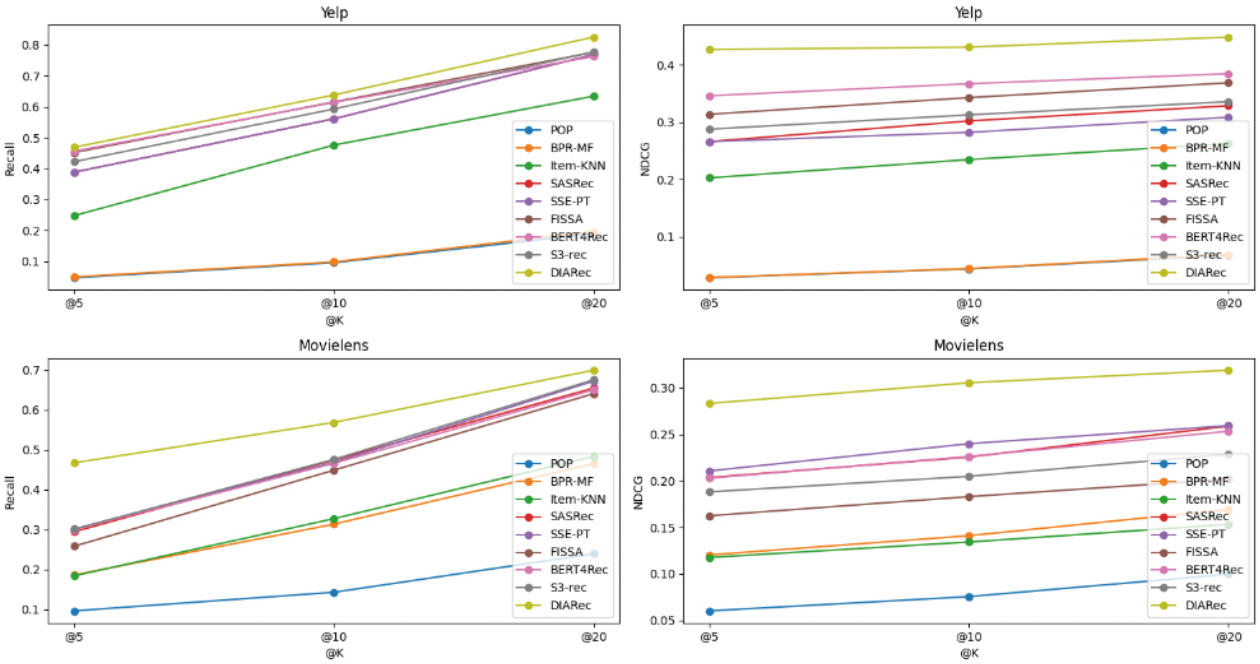


Figure 6. Performance comparison DIARec with different K on Movielens and Yelp

of both evaluation measures, DIARec has shown a 4% and 3% improvement in terms of Recall on Movielens and Yelp datasets, respectively. The most likely reason is that the proposed method effectively shares information among the heads, enhancing its capability to grasp complex data relationships and make better use of the information generated by other heads.

2. Generally, all unified RSs (DIARec, SASRec, BERT4G, SSE-PT, FISSA, and S3Rec) outperform traditional and sequential approaches (i.e., POP, Item-KNN, BPR-MF, and GRU4Rec), respectively.

3. Overall, all attention-based approaches outperform general- and deep learning-based recommenders, respectively. This is likely due to the attention mechanism's ability to recognize the most important items in user interactions.

4. Among all attention-based approaches, DIARec outperforms others. This indicates the effect of multi-head attention in identifying different aspects of users' interests. Moreover, although SSE-PT, BERT4, and FISSA use attention mechanisms, they cannot model the attributes of different items. Also, unlike all the mentioned methods, DIARec can recommend momentous items according to user purposes and interests.

4.4 Impact of DIARec Modules (RQ2)

In this section, to evaluate the performance of the DIARec modules, we introduce the following two methods: (1) Pur_DIARec, which only utilizes the user purpose module, and (2) Pre_DIARec, which only considers the user's preference. Table 4 and Table 5, and Figure 4 illustrates the results of experiments using the NDCG@20 and Recall@20 metrics on two benchmark datasets. According to Table 4 and Table 5, we can make the following observations:

1. The Pur_DIARec method effectively improves the performance of our approach. This is probably due to capturing attribute information in a long-term interacted item set through a multi-head attention mechanism and emphasizing the importance of items based on user interests by the AIU cell.

2. The Pre_DIARec method also contributes to improving the performance of our approach DIARec compared to other state-of-the-art methods. This may be attributed to the use of GRU in the short-term to capture the user's sequential pattern in session sequences, aiding in better modeling user representations.

3. Since DIARec's performance is better than either of the two modules, Pre_DIARec and Pur_DIARec, it can be concluded that the fusion of these two modules is effective in learning user representations and predicting the next items.

4.5 Impact of IAU component(RQ3)

In this section, we verify the efficiency of the IAU module with the presence and absence of a collaborative multi-head attention mechanism. As evident from Figure 5 and in Table 6, we observe that applying a collaborative multi-head self-attention mechanism in our model outperforms the multi-head self-attention. The reason behind these results may be the enhanced ability of the Collaborative Multi-Head Self-Attention model to capture various contextual elements and nuances from different latent subspaces. This is achieved by allowing information to be shared among the attention heads, ultimately improving the model's capacity to comprehend intricate data relationships and make better use of information generated by other heads.

As a result, our approach can effectively highlight important attributes of items according to the user's preferences, thereby aiding in learning a more accurate user representation. Based on the obtained results, we can confidently validate the efficiency of the IAU component within the DIARec approach.

4.6 Impact of under different K in model performance (RQ4)

In the real world, users often browse only the primary few pages of a website and tend not to pay attention to the rest of the pages. A strong RS can accurately predict the top-k items; therefore, we explore the performance of our approach in terms of Recall@K and NDCG@K when K is set to 5, 10, and 20. The results are shown in Table 7. To display experimental results more intuitively, we drew the corresponding line chart in Figure 6.

Table 8. Computational overheads for testing time and each epoch at training time.

Datasets	Movielens		Yelp	
Methods	Training time (s)	Testing time (s)	Training time (s)	Testing time (s)
POP	2.16	1.12	6.26	3.87
BPR-MF	9.11	1.75	13.10	5.22
Item-KNN	7.41	1.27	10.31	4.71
GRU4Rec	16.19	2.24	54.18	7.23
SASRec	21.18	3.34	60.13	9.81
BERT4Rec	23.68	3.55	68.73	10.30
FISSA	24.19	3.40	65.49	10.49
SSE-PT	23.72	3.37	63.05	10.68
S3Rec	25.54	3.74	71.13	11.21
DIARec	27.81	4.01	75.56	11.51

In general, for all values of K, our proposed model performs best on both datasets. According to the results displayed by Recall and NDCG on both datasets, increasing the value of K will increase the Recall in all methods, and the magnitude is great and relatively stable in Recall and NDCG, respectively. The reason is that the value of Recall will increment by 1 for each hit, and the value of NDCG will increment by a number between 0 and 1.

4.7 Efficiency comparison (RQ5)

In this section, we analyze computing requisites in terms of time complexity compared to other state-of-the-art methods during the training and testing processes. Due to the difficulty of providing exact computing complexity in theory for deep learning models, we report the computational time (in seconds) for all compared algorithms on both datasets in Table 8. The evaluation duration includes the time required to obtain Recall@K and NDCG@K. Observations from the results in Table 8 are as follows:

Among non-deep learning approaches, POP has the least running time on both datasets because it ranks items according to their frequency of occurrence. The execution time of Item-KNN was higher than POP, possibly due to the complexity of the model, as it needs to compute the similarity of each item with all other items, making these computations time-consuming. BPR-MF required more time compared to Item-KNN, particularly during the training phase, owing to the increased model complexity resulting from matrix decomposition calculations. GRU4Rec requires more computing time than traditional approaches in both datasets. This increased time demand can be attributed to the internal gates within the model architecture. Generally, all transformer-based techniques need more computing time than traditional and deep learning-based approaches in both datasets. Among all transformer-based approaches, DIARec has more computing time on both datasets. A plausible explanation is that DIARec has more complexity due to its hierarchical structure. In summary, models like DIARec, BERT4Rec, and S3Rec demonstrate improved performance while achieving a balance between model accuracy and running time. Nevertheless, in practical scenarios, these models can undergo offline training, allowing the recommendation system to utilize a pre-trained model for predictions. The consumed testing time for all methods varied from around 1 to 4 seconds on the Movielens dataset and from 3 to 11 seconds on the Yelp dataset. Therefore, DIARec is suitable for practical applications, given that its testing time is comparable with that of state-of-the-art models, and has the dual competitiveness of effectiveness and efficiency with other state-of-the-art methods.

5 Conclusions

This paper introduces a novel context-aware recommendation approach that unifies traditional and sequential RSs. In our proposed model, DIARec, we utilize an Intention-Aware Unit (IAU) to identify essential aspects of item attributes based on users' intentions. We present a novel collaborative

multi-head self-attention network that combines information from all the different attention heads effectively to enhance the model's ability to capture complex user preferences. The results of experiments on two datasets demonstrate that our proposed approach, DIARec, outperforms other models.

The model introduced in this article warrants further investigation. We currently utilize latent context factors and item attributes as the primary data sources for learning user preference representation. However, in real-world recommendation processes, additional valuable information is often available, such as users' trust relationships in the social network, explicit user-item ratings, diverse item attributes, and explicit contextual information. The inclusion of rich side information typically leads to improved performance.

DIARec also has specific real-world implications. For instance, in an online shopping platform, DIARec can efficiently and accurately analyze items that users may purchase based on their historical interactions. This analysis takes into account both the items themselves and their associated attributes within the context of the user, distinguishing which attributes wield a greater impact in the user's specific context. Additionally, DIARec can suggest similar items to users by extracting the item's semantic information, thereby enhancing their shopping experience and satisfaction.

In our future work, we intend to incorporate the time gap in user interactions within sequential recommenders to enhance the accuracy of recommendations. Additionally, we plan to improve the user intent learning process by incorporating convolutional neural networks (CNN).

References

[1] Justyna Pawłowska, Klara Rydzewska, Adam Wierzbicki Using Cognitive Models to Understand and Counteract the Effect of Self-Induced Bias on Recommendation Algorithms, Journal of Artificial Intelligence and Soft Computing Research, Volume 13, Issue 2, pages 73-94, 2023.

[2] Piotr Woldan, Piotr Duda, Andrzej Cader, Ivan Laktionov, A New Approach to Image-Based Recommender Systems with the Application of Heatmaps Maps, Journal of Artificial Intelligence and Soft Computing Research, Volume 13, Issue 2, pages 63–72, 2023.

[3] Li X., Sun A., Zhao M., Yu J., Zhu K., Jin D., Yu R., Multi-Intention Oriented Contrastive Learning for Sequential Recommendation, In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pages 411-419, February 2023.

[4] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. Learning hierarchical representation model for nextbasket recommendation. In Proceedings of the 38th International ACM SIGIR, Conference on Research and Development in Information Retrieval, pages 403–412, 2015.

[5] Mehdi Hosseinzadeh Aghdam, A novel constrained non-negative matrix factorization method based on users and items pairwise relationship for recommender systems, Expert Systems with Applications, 195:116593, 2022.

[6] Jinjin Zhang, Xiaodong Mu, Peng Zhao, Kai Kang, and Chenhui Ma. Improving current interest with item and review sequential patterns for sequential recommendation, Engineering Applications of Artificial Intelligence, 104:104348, 2021.

[7] Mehdi Hosseinzadeh Aghdam, Context-aware recommender systems using hierarchical hidden markov model, Physica A: Statistical Mechanics and Its Applications, 518:89–98, 2019.

[8] Balazs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk, Session-based recommendations with recurrent neural networks, arXiv preprint arXiv:1511.06939, 2015.

[9] Jiaxi Tang and KeWang, Personalized top-n sequential recommendation via convolutional sequence embedding, In Proceedings Of The Eleventh ACM International Conference On Web Search And Data Mining, pages 565–573, 2018.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, Attention is all you need, Advances in neural information processing systems, 30, 2017.

[11] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme, Factorizing personalized markov chains for next-basket recommendation, In Proceedings of the 19th international conference on World wide web, pages 811–820, 2010.

[12] Wang-Cheng Kang and Julian McAuley, Self-attentive sequential recommendation, In 2018 IEEE international conference on data mining (ICDM), pages 197–206. IEEE, 2018.

[13] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang, Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the 28th ACM international conference on information and knowledge management, pages 1441–1450, 2019.

[14] Ma M., Ren P., Chen Z., Ren Z., Liang H., Ma J., De Rijke M., Improving transformer-based sequential recommenders through preference editing. ACM Transactions on Information Systems, 41(3), pages 1-24, 2023.

[15] Francisco Pajuelo-Holguera, Juan A Gomez-Pulido, and Fernando Ortega, Recommender systems for sensorbased ambient control in academic facilities. Engineering Applications of Artificial Intelligence, 96:103993, 2020.

[16] Yehuda Koren, Robert Bell, and Chris Volinsky, Matrix factorization techniques for recommender systems. Computer, 42(8):30–37, 2009.

[17] Ruining He and Julian McAuley, Fusing similarity models with markov chains for sparse sequential recommendation. In 2016 IEEE 16th international conference on data mining (ICDM), pages 191–200. IEEE, 2016.

[18] Balazs Hidasi and Alexandros Karatzoglou, Recurrent neural networks with top-k gains for session-based recommendations. In Proceedings of the 27th ACM international conference on information and knowledge management, pages 843–852, 2018.

[19] Yong Kiam Tan, Xinxing Xu, and Yong Liu, Improved recurrent neural networks for session-based recommendations. In Proceedings of the 1st workshop on deep learning for recommender systems, pages 17–22, 2016.

[20] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. What to do next: Modeling user behaviors by time-lstm. In IJCAI, volume 17, pages 3602–3608, 2017.

[21] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. A collaborative sessionbased recommendation approach with parallel memory modules. In Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pages 345–354, 2019.

[22] Massimo Quadrana, Alexandros Karatzoglou, Balazs Hidasi, and Paolo Cremonesi. Personalizing session-based recommendations with hierarchical recurrent neural networks. In proceedings of the Eleventh ACM Conference on Recommender Systems, pages 130–137, 2017.

[23] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma, Neural attentive session-based recommendation, In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pages 1419–1428, 2017.

[24] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu, Sequential recommender system based on hierarchical attention network, In IJCAI International Joint Conference on Artificial Intelligence, 2018.

[25] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang, Stamp: short-term attention/memory priority model for session-based recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1831–1839, 2018.

[26] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke, A dynamic co-attention network for sessionbased recommendation. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pages 1461–1470, 2019.

[27] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618, 2012.

[28] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web, pages 285–295, 2001.

[29] Jing Lin, Weike Pan, and Zhong Ming, FISSA: Fusing item similarity models with self-attention networks for sequential recommendation. In The 14th ACM Conference on Recommender Systems, pages 130–139, 2020.

[30] L. Wu, S. Li, C. Hsieh, J. Sharpnack, SSE-PT: Sequential recommendation via personalized transformer, in: RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020, ACM, pages 328–337, 2020.

[31] Kun Zhou, Hui Wang, WayneXin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen, S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization, In The 29th Conference on Information and Knowledge Management, pages 1893–1902, 2020.

[32] Jung-Hsien Chiang, Chung-Yao Ma, Chi-Shiang Wang, and Pei-Yi Hao. An adaptive, context-aware,

and stacked attention network-based recommendation system to capture users temporal preference. IEEE Transactions on Knowledge and Data Engineering, 2022.

[33] Zhang Y., Yang B., Liu H., Li D., A time-aware self-attention based neural network model for sequential recommendation, Applied Soft Computing, 133, 109894, 2023.

[34] El Zini J., Rizk Y., Awad M., An optimized parallel implementation of non-iteratively trained recurrent neural networks, Journal of Artificial Intelligence and Soft Computing Research, 11(1), pages 33-50, 2021.

[35] Shewalkar A., Nyavanandi D., Ludwig S. A., Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. Journal of Artificial Intelligence and Soft Computing Research, 9(4), pages 235-245, 2019.

[36] Wang Y., Gao M., Ran X., Ma J., Zhang L. Y. An improved matrix factorization with local differential privacy based on piecewise mechanism for recommendation systems, Expert Systems with Applications, 216, 119457, 2023.



Hadise Vaghari is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Qeshm Branch, Islamic Azad University, Qeshm, Iran. Her research interests include deep learning and recommender systems.
https://orcid.org/0000-0003-4176-3576



Mehdi Hosseinzadeh Aghdam is an associate professor at the University of Bonab. He received Ph.D. in artificial intelligence from Iran University of Science and Technology in 2016. Before that, he received his Master's degree from the University of Isfahan. His research interests include machine learning, information retrieval and recommender systems.
https://orcid.org/0000-0002-3922-9991



Hojjat Emami Ph.D in Computer Engineering- Artificial. He currently is an Associate Professor at the University of Bonab, Department of Computer Engineering. His research interests include machine learning and social network analysis.
https://orcid.org/0000-0002-5280-4620