Article in Press IJIMAI journal

Attentive Flexible Translation Embedding in Top-N Sparse Sequential Recommendations

Min-Ji Seo, Myung-Ho Kim *

Department of Software Convergence, Soongsil University, 369, Sangdo-ro, Dongjak-gu, Seoul 06978 (Korea)

Received 9 September 2021 | Accepted 2 September 2022 | Early Access 19 October 2022

Abstract

Sequential recommendation aims to predict the user's next action based on personal action sequences. The major challenge in this task is how to achieve high performance recommendation under the data sparsity problem. Translation-based recommendations, which learn distance metrics to capture interactions between users and items in sequential recommendations, are a promising method to overcome this issue. However, a disadvantage of translation-based recommendations is that they capture long-term preferences of the user and complex item transitions. In this paper, we propose attentive flexible translation for recommendations (AFTRec) to tackle data sparsity problem by capturing a user's dynamic preferences and complex interactions between items in user's purchasing behaviors. In particular, we first encode semantic information of an item related to user's purchasing behaviors as the user-specific item translation vectors. We also design a transition graph and encode complex item transitions as correlation-specific item translation vectors. Finally, we adopt a flexible distance metric that considers directions with respect to the translation vectors in the same space for predicting the next item. To evaluate the performance of our method, we conducted experiments on four sparse datasets and one dense dataset with different domains. The experimental results demonstrate that our proposed AFTRec outperforms the state-of-the-art baselines in terms of normalized discounted cumulative gain and hit rate on sparse datasets.

Keywords

Deep Learning, Gated Graph Neural Network, Knowledge Graph Embedding, Recommender Systems, Self-Attention, Sequential Recommendation.

DOI: 10.9781/ijimai.2022.10.007

I. Introduction

Recommender systems (RSs) have received interest on various platforms, such as e-commerce, news portals, and social media sites. The main purpose of RSs is to suggest the most relevant recommendations to users so that they make informed purchasing choices. Traditional recommendation systems [1]–[3], such as collaborative filtering (CF), make recommendations by analyzing historical interactions or preferences based on the similarity of users or items in the past. However, following the explosive growth of e-commerce, the data sparsity problem, which refers to the difficulty in finding sufficient similar users and items due to insufficient user–item interactions, is the main challenge in RS. To address this issue, matrix factorization (MF) [4] models that map both the user and item embedding vectors and represent user–item interactions by the inner product of the user and item vectors have been proposed.

To deal with sequential user behaviors (e.g., click and purchase) in e-commerce, sequential recommendation systems [5]–[8] have been proposed in RS for data sparsity problems. Examples of such models include factorized personalized Markov chains (FPMC) [9], which combine Markov chains (MCs) [10] and MF to predict the next action of the user in sequential data. The FPMC captures both long-term user preferences and short-term sequential dynamics by modeling the interactions between user-to-item and item-to-item pairs. This underlies personalized MCs, where a user-specific transition matrix is applied to capture personalized item transitions. Achieving better performances on sparse datasets, many researchers have recently found new ways to capture interactions between user-to-item and item-to-item pairs.

Translation-based methods [11]–[14], which facilitate knowledge graph (KG) completion [15]-based approaches, such as translation-based recommendation (TransRec) [16], latent relational metric learning (LRML) [17], and collaborative metric learning (CML) [18], have achieved high performance with sparse datasets for next item recommendation. TransRec utilizes KG completion to model users as translation vectors from their previously purchased item vectors to the vector of the next items in the same translation space. To model item-to-item interactions in chronological order, TransRec adopts a translational principle, which minimizes the distance between the translation vectors. However, these translation-based recommendation methods have several drawbacks in sequential recommendations. First, they mainly adopt translating embeddings for modeling multi-relational data (TransE) [19], which is capable of 1-to-1 relations but is unable to handle 1-to-N, N-to-1, and N-to-N relations. Second, there are few studies that focus on the user's long-term and short-term preferences in translation vectors without user and item context information, such as category and user profile.

* Corresponding author.

E-mail addresses: porito2@soongsil.ac.kr (M. J. Seo), kmh@ssu.ac.kr (M. H. Kim).

Please cite this article in press as:
M. J. Seo, M. H. Kim. Attentive Flexible Translation Embedding in Top-N Sparse Sequential Recommendations, International Journal of Interactive Multimedia and Artificial Intelligence, (2022), http://dx.doi.org/10.9781/ijimai.2022.10.007

In this study, we propose an attentive flexible translation for recommendations (AFTRec) to predict the user's next item in sparse sequential recommendation datasets for the data sparsity problem. Specifically, unlike existing approaches, which primarily focus on the last consumed item, we focus on the sequential behaviors of the user and complex interactions between purchased items by users in chronological order. To facilitate KG completion in predicting the next item, we generate user-specific item translation vectors that reflect dynamic user preferences and target item translation vectors that represent the user's next item as entities. We also generated a correlation-specific item translation vector that reflects item-to-item interactions in user behavior histories as a relation vector. For KG completion to predict the next item, we propose a distance function that can flexibly handle not only 1-to-1, but also 1-to-N, N-to-1, and N-to-N relations. Our AFTRec consists of three modules: a user-specific item translation vector embedding module, correlation-based item translation vector embedding module, and attentive item translation vector embedding module. The model applies KG completion to translation vectors for moving the user's previous item vectors close to the user's next item vectors in the same translation space. First, we generated item embeddings based on user behaviors through a self-attention mechanism, which is efficient for capturing long-term item dependencies with the position information of the item. For the correlation-specific item translation vector, we initially designed the transaction graph and linked the neighbors of items based on a sliding window, which slides the item sequences in a window-by-window manner. Then, we learned item-to-item interactions in sequential user behaviors by utilizing gated graph neural networks (GGNNs) [20], which are capable of representing sophisticated item interactions with comprehensive item transitions in user behavior sequences. In the final module, we generate the attentive item translation vector that aggregates the user's sequential preferences and relationships between items and then embed translation vectors into the same space with KG embedding for translation from a previous item to the next item. Inspired by the flexible translation (FT) [21] of KG embedding, we designed our translational distance function in a new manner to model translation vectors. Therefore, unlike other existing translation-based RSs for sparse sequential datasets, AFTRec can capture not only personal item preferences but also sophisticated item interactions based on users' and users' sequential behaviors.

Our contributions can be summarized as follows:

1. We propose a novel translation-based sequential recommendation model. We adopt the KG embedding technique to encode sequential behaviors of a user and item-to-item relationships as entities and relations of a KG triple. We model various correlations between entities and relations to find the next item with our translational distance function, which releases existing translation approaches. Using this approach, AFTRec can capture pairwise relations between users and items more efficiently.
2. We embed sequential user preferences as a user-specific item translation vector as the head entity by applying a self-attention mechanism [22] in chronological order to understand long-term user preferences. For the secondary head entity of our distance function, we define the attentive item translation vector. The attentive item translation vector summarizes the item correlations related to the purchasing preference of each user through the soft-attention mechanism. Hence, we consider various perspectives on user and item information to translate the previous item into the next item.
3. We represent item-to-item interactions as correlation-specific item translation vectors as a relation of the KG triple through GGNN. Initially, we design a transaction graph by connecting adjacent items in chronological order using a sliding window method. In particular, we divide edges into incoming and outgoing edges, and thus efficiently represent item interactions with the purchase order in terms of the time position. In addition, we utilize the GGNN to analyze item interactions. Because the GGNN uses a gated recurrent unit (GRU) [23] as an updater, it helps reduce the number of parameters for analysis.
4. We conduct extensive experiments using four sparse datasets and one dense dataset from different domains to evaluate the proposed method. The experimental results demonstrate that our method outperforms other existing approaches in solving the data sparsity problem.

The remainder of this paper is organized as follows. Related studies are introduced in Section II. Next, we describe our proposed method in Section III. In Section IV, we describe the experiments conducted on publicly available datasets of several domains, evaluate our proposed method in comparison with other approaches, and analyze the experimental results. Finally, we conclude the paper in Section V.

II. Related Work

A. Traditional Recommender Systems

RSs aim to predict user preferences and suggest relevant items to the users. Traditionally, CF-based methods are used in RSs. CF recommends items in which similar users are interested based on historical data. For example, MF models the explicit feedback of a user with user and item latent factors by calculating the dot product of the two latent factors. To address item-based CF, a factored item similarity model [24] embeds each item and models the similarity between two items using the inner product of their embedding vectors. The neural attentive item similarity model [25] assigns an attentive weight to each item in the item sequences and shows good results in the calculation of the similarity between items. Bobadilla et al. [26] utilize neural CF to obtain prediction reliabilities and combine the prediction value and the reliability information in user ratings. Bobadilla et al. [27] improve fairness in RSs by combining probabilistic MF and multi-layer network. However, these methods have limitations in handling sequential patterns in interactions between users and items.

B. Sequential Recommender Systems

Sequential RSs investigate sequential behaviors of a user to recommend the next item. MC-based methods temporarily capture item transitions and predict the next item based on the last consumed item. FPMC fuses MF and MC to predict the next actions with the user's general interests and short-term item transitions.

Inspired by the success of neural networks, various neural-network-based methods have been introduced for sequential RSs. Recurrent neural network (RNN)-based recommendations [28], [29] employ variations of RNN, such as long short-term memory and GRU, which are capable of modeling sequential patterns, to predict the next action of the user. However, because RNN-based recommendations contain information regarding the final state of the model, they are limited in modeling long sequences. To address this problem, attention-based RNN methods [30], [31] have been proposed. A neural attentive recommendation machine [32] applies an attention mechanism to a stacked GRU-based encoder–decoder to model the sequential behavior and capture general preferences of the user. Recently, self-attention mechanisms have become popular, with promising performance in natural language processing (NLP) problems. Accordingly, many researchers have utilized self-attention to provide suitable recommendations in historical sequences. Self-attentive sequential recommendation (SASRec) [33] uses stacked self-attention blocks to efficiently consider long-term dependencies. A stochastic shared embeddings-personalized transformer (SSE-PT) [34] introduced personal information into self-attention by concatenating item embedding and embedding it into the self-attention embedding layer. Time interval-aware self-attention-based sequential recommendation (TiSASRec) [35] utilizes both the absolute positions of the items and the time intervals between items in a sequence. It represents the relationship between items as a time interval and shows performance improvement on a personalized sequential recommendation using two types of item positions: sequential and relative time positions.

However, these methods have several limitations. First, MC-based approaches predict the next item using only the last consumed item; thus, they do not explicitly capture the complex and long-term dependencies. Second, convolutional neural networks [36], [37] and RNN-based methods involve the risk of missing crucial information on previously consumed items and lack explanations for recommended items. Third, self-attention-based methods have insufficient ability to treat complex item-to-item and user-to-item interactions.

C. Translation-Based Embedding Model in Recommender Systems

The goal of KG completion-based RSs is to learn the relationships between users and items by minimizing the distance between the translation vectors in the same space. Fig. 1 describes TransE embedding and two KG-based approaches in RSs, namely, CML and LRML. KG-based recommendation methods initially utilize KG embeddings to predict user's item ratings or implicit next interactions between users and items in RSs. CML minimizes the distance between the user and item vectors using personalized historical implicit feedback. LRML uses a memory-based attention network to represent the latent relationships between the user and previous item vectors as latent relation vectors. Then, LRML advances the metric learning of CML, which operates via $p \approx q$ to $p + r_1 \approx q$, where p and q are the user and item vectors, respectively, and r_1 is a user-to-item relation vector. CML and LRML then apply the model's distance function to find the next item vector with the shortest distance from the user vector.

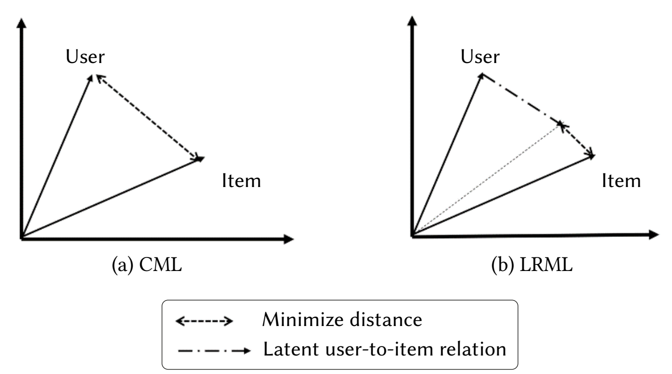

Fig. 1. Simplified illustration of (a) CML and (b) LRML.

To solve the data sparsity problem in sequential RSs, translation-based recommendations have been proposed. Translation-based recommendations embed the user and item vectors as translation vectors of a triple form (head, relation, tail) using KG completion to move the previous user's item vector to be close to the user's next item vector in sequential behavior sequences. Fig. 2 illustrates the process of providing the next item recommendation through TransRec. In Fig. 2, TransRec predicts the user's next item by modeling third-order interactions between the user's previous item, the user, and the user's next item in a translation space. The previous items of the user are also modeled to move the previous item to the next item in chronological order through TransE, as shown in Fig. 2. Fig. 3 illustrates the process of providing the next item recommendation through mixtures of heterogeneous recommenders (MoHRs) [38]. Similar to TransRec in Fig. 2, MoHR predicts the user's next item by modeling third-order interactions based on the user's item sequences and user information. Specifically, MoHR represents various sequential relationships, that is, previous item-to-next item and user-to-next item, and adopts KG embedding to predict the next item-based distance from the previous item vector in the translation space. MoHR captures three types of relationships: long-term user-to-item preferences, relationships between short-term item transitions, and exhibit relationships (e.g., also-bought/also-viewed) between the short-term item transitions by applying TransE separately for each relationship, as shown in Fig. 3. MoHR also models item vectors in the user's purchasing sequences to move the previous item vector close to the user's next purchased item vector. An attentive translation model for next item recommendation (ATM) [39] constructs a user, multiple previous items, and the next items as translation vectors to translate a user to the next item. In particular, ATM implements high-order MCs to embed a user's sequential behaviors into the relation vector. ATM then models third-order interactions (a user, the user's sequential preferences, and the next item).

Recently, translation-based recommendations have also facilitated KG completion to predict user-to-item ratings in sequential RSs [11], [14]. Translation-based factorization machines [40] combine KG completion and factorization machines to predict user–item ratings in sequential RSs. To improve the performance of translation-based recommendations, recent approaches utilize user and item context information, such as item category and user region. The adaptive hierarchical translation-based sequential recommendation [41] captures item sequence patterns based on implicit purchasing behaviors and purchased item category information by modeling sequential item interactions using KG completion.

However, these approaches require additional resources and time to consider contextual attributes. To reduce the resources of context analysis, we solved the data sparsity problem and predicted the next item using only implicit interactions between the user and items inspired by TransRec and MoHR with performance comparable to recent sequential recommendation methods such as SSE-PT and TiSASRec introduced in Section II.B.

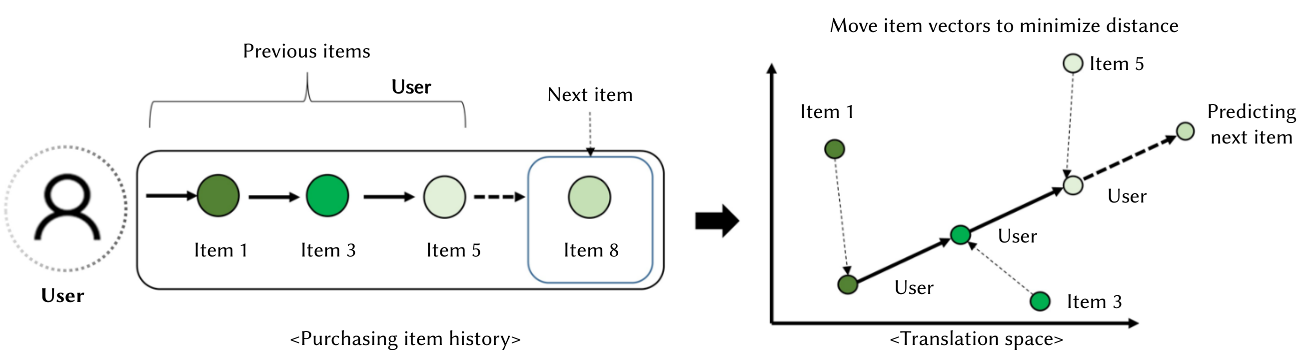

Fig. 2. Simplified illustration of TransRec based on the user's item sequences.

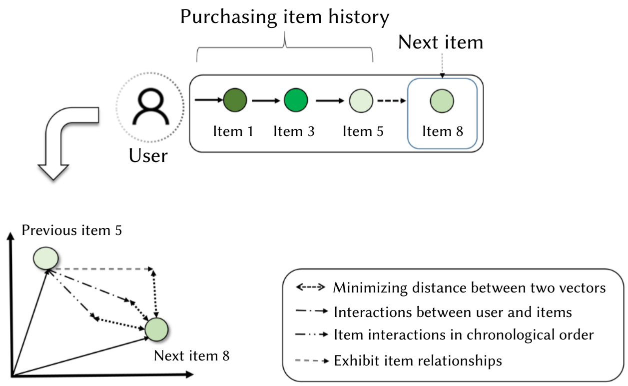

Fig. 3. Simplified illustration of MoHR. MoHR models distance functions for three types of relationships to each previous item pairs. Then, MoHR finds the item vector with the shortest distance from the last purchased item as the next item.

Many translation-based recommendation systems exhibit robust performance on sparse datasets, such as e-commerce, by adopting the translational distance for capturing third-order interactions (a user, a previously consumed item, the next item) to the next item recommendation. TransE models $h + r \approx t$, where (h, r, t) is a triple of KG, with a promising performance in 1-to-1 relationships, but it is too strict to model 1-to-n, n-to-1, and n-to-n relationships. In addition, many approaches cannot address users' long-term dependencies and thus achieve lower performance than RNN- and self-attention-based sequential recommendation systems in sequential recommendation. To address this issue, our proposed method uses a translational embedding model that handles not only 1-to-1 and other relationships but also long-term dependencies and sophisticated item interactions in sequential behaviors to recommend the most appropriate target item.

III. Methodology

In recommendation research, many translation-based approaches have been proposed that learn the relationships between users and items as translation vectors for sequential recommendation. In this section, we introduce the novel translation-based recommendation AFTRec, which applies KG embedding to improve the sequential recommendation with sparse datasets. The architecture of the proposed AFTRec is shown in Fig. 4. First, we encoded the information of a user's consumed item based on personalized sequential behaviors to the user-specific item translation vector γ_u as the head entities. In this process, the self-attention mechanism was used to capture the items' long-term dependencies in sequential behaviors (Section III.A). Next, we designed a transaction graph that included item connections in chronological purchasing order. We divided the edges into incoming and outgoing edges to learn item interactions, reflecting changes in users' purchasing preferences. Using the transaction graph, we generated a correlation-specific item translation vector γ_r as relationships between entities, which includes sophisticated interactions between items in users' item sequences through GGNN (Section III.B). Finally, we optimized the metric function to score the interactions with the target item translation vector γ_j represented as the tail entity. In this module, we additionally created comprehensive item vectors $\gamma_{u'}$ that explicitly aggregated the item's information related to user-to-item and item-to-item interactions. Inspired by FT embedding, our distance function considered the direction of translation vectors to release the strict translational principle $h + r \approx t$. Owing to the flexible metric in the proposed method, we additionally considered the relationships between consumed items and target item vectors from the two perspectives with γ_u and $\gamma_{u'}$. Owing to the three generated vectors of user behavior-based item vector, comprehensive item vector, and item correlation vector as γ_u, $\gamma_{u'}$ and γ_r, respectively, we were able to optimize the translational embedding model $(\gamma_{u'} + \gamma_r)^\top \gamma_j + (\gamma_j + \gamma_r)^\top \gamma_u$ to find the next item (Section III.C).

A. User-Specific Item Translation Vector Embedding

Let U and I represent the user and item sets, where $u \in U$ denotes a user and $i \in I$ denotes an item. For each user u, we extracted every L successive items as a user action sequence. In this module, we generated a user-specific item translation vector $\gamma_u \in R^{|L| \times d}$. By reflecting users' long-term preferences in γ_u, AFTRec considers item transitions and users' purchasing history to model relationships between items using translation-based approaches.

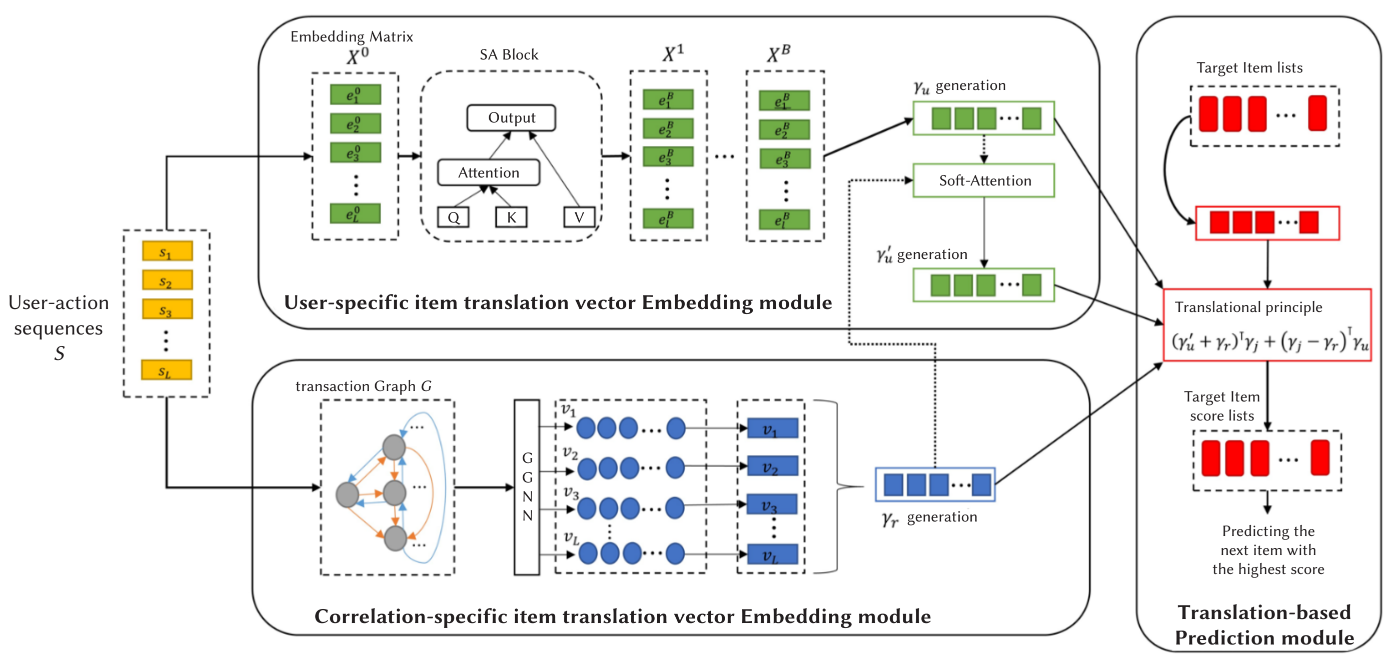

Fig. 4. Architecture of our proposed method.

Let $S_u = \{i_1, i_2, ..., i_L\}$ denote the set of all L items ordered by timestamp. In this section, a user-specific item translation vector is generated for each previous item in $\{i_1, i_2, ..., i_L\}$ as shown in Fig. 4. With the translational principle, AFTRec predicted the item for the $t + 1$ step-based translational distance with γ_u corresponding to the purchased item in step t $(0 < t < L)$ in chronological order.

In S_u, we created an item embedding matrix $M \in R^{L\times d}$, where d is the latent dimensionality. In addition, we generated a learnable position embedding matrix $P \in R^{L\times d}$ as the purchasing order information in the user sequence. We obtained the item embedding lookup matrix $E \in R^{L\times d}$ by calculating $E = M + P$. To efficiently represent item translation vectors reflecting user preferences, we utilized stacked self-attention blocks (SABs) for E. The SAB consists of a multi-head attention and a pointwise feed-forward (FF) layer. Multi-head attention runs a scaled dot-product attention mechanism several times in parallel. Because it concatenates different representations of an item's dependencies from various perspectives, it is beneficial to consider multiple relationships jointly through a separate analysis. MHA was calculated as follows:

$$MHA(Q,K,V) = Concat[hd_1, hd_2, \ldots, hd_h]W^{MHA} \tag{1}$$

$$hd_i = Attention\left(EW_i^Q, EW_i^K, EW_i^V\right) \tag{2}$$

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{3}$$

where Q, K, and V denote the sets of queries, keys, and values, respectively. In addition, W^{MHA}, W_i^Q, W_i^K, and $W_i^V \in R^{d\times d}$ are learnable parameters, and $\sqrt{d}$ is a scale factor that scales the dot products to avoid the vanishing gradient problem. We provided the item embedding lookup matrix E as input, which can be defined as a linear transformation of $Q = EW^Q$, $K = EW^K$, and $V = EW^V$. To reflect a realistic sequential behavior of a user to a user-specific item translation vector, we considered t items when generating the t-th purchased item translation vector. Therefore, we masked the queries and keys from t+1 to the last item. Then, the pointwise FF layer was calculated as follows:

$$SAB(X) = FFN\left(MHA\left(EW_i^Q, EW_i^K, EW_i^V\right)\right) \tag{4}$$

$$FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2 \tag{5}$$

where $W_1, W_2 \in R^{d\times d}$ are the learnable parameters. In addition, $b_1, b_2 \in R^{1\times d}$ are the bias parameters. A pointwise FF layer was applied to each item position separately to aggregate and normalize the attention outputs. Similar to [22], the pointwise FF layer included a residual connection and layer normalization, which are omitted in (4) for brevity. To efficiently improve the performance of capturing item transitions in long-term sequences, we stacked the SABs. The B−th $(B > 1)$ block is defined as follows:

$$X^{(B)} = SAB\left(X^{(B-1)}\right) \tag{6}$$

where $X^{(0)} = E$. In this module, we obtained the output item embedding matrix $X' \in R^{d\times d}$ using stacked SABs. We then defined X' as the head γ_u in the transition space. In contrast to TransRec and MoHR, where the translation vectors are represented only by item embeddings, our proposed method is able to represent not only long-term preferences but also item transitions.

B. Correlation-Specific Item Translation Vector Embedding

In this section, we generate a relation vector that translates the interaction between the previous and next items for personalized recommendation in the same space. Therefore, we encoded complex item-to-item interactions in users' purchasing behaviors to correlation-specific item vectors $\gamma_r \in R^{L\times d}$ based on users' item sequences, as shown in Fig. 4.

Because the basic idea of graph neural networks (GNNs) [42] is to generate node embedding by aggregating the features and topological information from the neighbors, it ensures that GNNs are capable of efficiently capturing the interactions between nodes on graph-structured data. GGNN extends GNNs to sequential data, using a GRU as an update function to propagate information. Owing to the use of a GRU, GGNN selectively aggregates information of the neighbors, and thus, it is able to reduce the computational limitations and achieve a better performance. In this study, we converted personal item sequences to graph-structured data and learned the general relationships between consumed items in the e-commerce platform through GGNN, as described below.

1. Constructing a Session Graph

For user u, given the behavior sequence S_u, we designed a transaction graph G. Let $G_u = (\mathcal{V}, \mathcal{E})$ be a directed graph, where each node denotes a purchased item at time t as $v_t \in I$, and each edge $(v_{t-1}, v_t) \in \mathcal{E}$ denotes each link for a chronologically ordered pair of items. To represent chronological item-to-item relationships, we built an adjacency matrix using a sliding window that moved a unit distance ahead. For the user sequence S_u, we moved the window in a unit time and connected the links between neighboring item nodes.

An example of the construction of an adjacency matrix is shown in Fig. 5. The adjacency matrix $A \in R^{L\times 2L}$ is represented by two adjacency matrices $A^{BF}, A^{AF} \in R^{L\times L}$, which represent connections of earlier or later purchased items as incoming or outgoing edges in the transaction graph, respectively. All edges have normalized weights with connections between earlier or later items for each item. In Fig. 5, each graph representation of a user process through the adjacency matrix is based on item sequences of each user, and the transaction graph is generated by repeating this process for all users in the data-sparse environment.

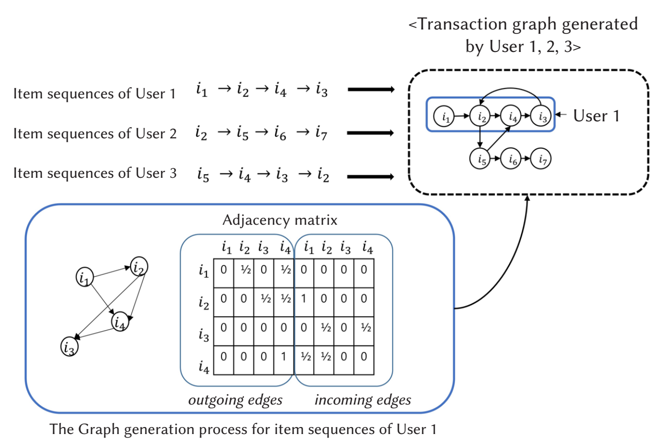

Fig. 5. An example of transaction graph generation process. Transaction graph is generated by the purchase history of users. The adjacency matrix is represented as a concatenation of two adjacency matrices, which link earlier or later purchased items, respectively.

2. Item-to-Item Interaction Learning

After constructing a transaction graph G_u of each user, we adopted a GGNN to learn item-to-item relations. Owing to the gating mechanism of a GRU, a GGNN can tackle the vanishing gradient and computational limitations by selectively gathering information from the other nodes to update the hidden state of each node. Let $h_i \in R^d$ denote the embedded node vector of the corresponding item v_i and H denote the set of all item node vectors. For the initialization step, the aggregation information a_i is defined as the concatenation of two types of adjacency matrices $A_i^{BF}, A_i^{AF} \in R^{1\times d}$ corresponding to the target node h_i:

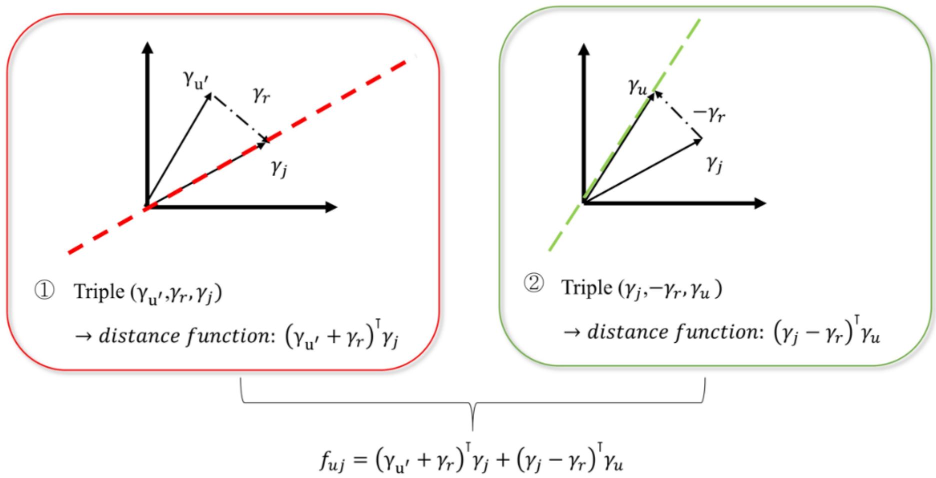

Fig. 6. Illustrations of translational principle for TransRec and AFTRec.

$$a_{i,BF}^t = A^{BF}([h_1^{t-1}, \dots, h_L^{t-1}]W_{BF}) + b_{BF} \tag{7}$$

$$a_{i,AF}^t = A^{AF}([h_1^{t-1}, \dots, h_L^{t-1}]W_{AF}) + b_{AF} \tag{8}$$

$$a_i^t = [a_{BF}^t; a_{AF}^t] \tag{9}$$

where W_{BF}, $W_{AF} \in R^{d \times d}$ are learnable parameters; b_{BF}, $b_{AF} \in R^d$ are bias parameters; $[h_1^{t-1}, \dots, h_L^{t-1}]$ is the list of item node states; and $[\cdot;\cdot]$ is the concatenation operation.

Then, the computation steps of updating h_i are defined as follows:

$$z_i^t = \sigma(W_z a_i^t + U_z h_i^t) \tag{10}$$

$$r_i^t = \sigma(W_r a_i^t + U_r h_i^t) \tag{11}$$

$$\widetilde{h_i^t} = tanh\left(W_o a_i^t + U_o(r_i^t \odot h_i^{t-1})\right) \tag{12}$$

$$h_i^t = (1 - z_i^t) \odot h_i^{t-1} + z_i^t \odot h_i^t \tag{13}$$

where W_z, W_r, $W_r \in R^{2d \times d}$, U_z, U_r, $U_o \in R^{d \times d}$ are learnable parameters. In addition, z_i^t and r_i^t are the update and reset gates, respectively. The reset gate determines the amount of past information that must be preserved or discarded. The update gate determines the amount of past information that must be passed along to the future. Moreover, σ denotes the logistic sigmoid function, and $\odot$ denotes the element-wise multiplication. This procedure was computed in a manner similar to the GRU. After this procedure, the corresponding items of all updated nodes were defined as the relations γ_r that contain high-level item-to-item interactions and short-term user interests in the transaction graph.

C. Optimization and Target Item Prediction

After obtaining the user and item translation vectors as the head and relation, respectively, we could predict the target item as the tail by optimizing the translational embedding model. In previous translation-based recommendations, (h, r, t) was modeled by the same translational principle $h + r \approx t$ in KG embedding techniques (e.g., TransE and TransR [43]). However, the translational principle $h + r \approx t$ is too strict to model the complex and diverse interactions between entities and relations (e.g., symmetric/transitive/one-to-many/many-to-one/many-to-many relations). To consider an item's diverse information related to personal preferences in a metric space, we extended the FT to generate flexible translation vectors with respect to multiple entities and relations. Originally, FT embedded multiple entities and relations by optimizing $(h + r)^\intercal t + (t - r)^\intercal h$. Given an ideal embedding $h + r \approx t$, FT applies $h + r \approx \rho t, \rho > 0$ by considering directions of vectors $h + r$ and t. To balance the constraints on the head and tail during training, FT considers both directions of vectors t and $h + r$ and h and $t - r$. Thus, it can flexibly capture more diverse and complex relationships between the head and tail.

For each triple (h, r, t), we can create an inverse triple (t, r^{-1}, h), which has also been used in [44], [45]. Thus, we can convert the translational principle $h + r \approx t$ to $t - r \approx h$. Using the FT principle, we can also apply $t - r \approx \rho h, \rho > 0$. For personal recommendations, we treat two user-specific item vectors as head entities: the user behavior-based item translation vector γ_u and the attentive item translation vector $\gamma_{u'}$. An attentive item translation vector strengthens the crucial information in the relationship between general purchased items and user preferences. Therefore, we applied the soft-attention mechanism [46] for long-term user interest and sophisticated item relations and then successfully aggregated the context pairs of user interest-to-item relations. The attentive item translation vector $\gamma_{u'}$ is defined as

$$m = tanh(ReLU(W_3(W_4\gamma_u + W_5\gamma_r) + b_3) \tag{14}$$

$$\alpha = \frac{exp\,(m)}{\sum_{l=1}^{L} exp(m_l)} \tag{15}$$

$$\gamma_{u'} = \alpha \gamma_u \tag{16}$$

where W_3, W_4, $W_5 \in R^{d \times d}$ are learnable parameters, and $b_3 \in R^d$ is the bias parameter. By considering the heads γ'_u, γ_u and the relation γ_r, we can seek the tail γ_r to predict a suitable item for the user's dynamic preferences. Fig. 6 illustrates translational embedding models of TransRec and our AFTRec. In our translational embedding model, we consider the directions of the vectors $(\gamma_{u'} + \gamma_i)$ with γ_j, and $(\gamma_j - \gamma_r)$ with γ_u.

Using balanced learning for the interactions of two triple sets $(\gamma_{u'}, \gamma_r, \gamma_j)$ and $(\gamma_{u'}, \gamma_r, \gamma_j)$ in a translation space, as shown in Fig. 6, AFTRec can flexibly capture diverse user and target item relations using different perspectives for the personal preferences of users. Finally, the model scores can be formulated as follows:

$$f_{uj} = (\gamma_{u'} + \gamma_r)^\intercal \gamma_j + (\gamma_j - \gamma_r)^\intercal \gamma_u \tag{17}$$

Based on our model's score, as shown in Fig. 6, AFTRec aims to maximize the probability of a true item under relationships in a user's behavior sequence. We adopted the binary cross-entropy loss for the optimization of the translation-based methods proposed by [47]–[49]. Given positive item set I and negative item set I', positive item $j \in I$ and negative item $j' \in I'$ are uniformly sampled. Then, we optimize the loss function as follows:

$$\mathcal{L} = \sum_{u \in U} \sum_{j \in I} -[\log \sigma(f_{uj}) + \sum_{j' \in I'} \log\left(1 - \sigma(f_{uj'})\right)] \tag{18}$$

where σ is the logistic sigmoid function used to obtain the predicted probability of a triple. In this model, we updated the parameters using an Adam optimizer [50] and regularized the parameters based on L_2 regularization to prevent overfitting. In the training process, for items purchased before the last purchased item, AFTRec modeled item vectors to predict the next item using the previous item based on our translational principle. AFTRec finally recommended an appropriate item for the user with the highest f_{uj} score with the user- and correlation-specific item translation vector for the latest purchased item.

IV. Experiments

A. Datasets

We evaluated AFTRec on five public datasets for real-world applications. All datasets had diverse domains and sizes. The statistics of all datasets are reported in Table I. For comparison with translation-based models that require standardized relationships between users and items, we used datasets from the Amazon and Steam platforms, which define specific relationship types between user-to-user and item-to-item pairs. We take five domains: "Beauty," "Toys and games (Toys)," "Clothing, shoes, and jewelry (Clothing)," and "Automotive" from Amazon review datasets in [51], and "Games" from Steam datasets generated in [33]. Amazon datasets were used as sparse datasets, whereas the Steam dataset was used as dense dataset. In this section, we demonstrate our performance for sparse datasets using Amazon datasets, and we experiment with our recommendation performance on dense datasets using the Steam dataset. All the datasets contain various user-to-item interaction data (e.g., user ratings and reviews). We followed the methods used by Kang and McAuley [33], and Wu et al. [34] to preprocess datasets to sort items in the sequential order of user sequences. First, we ordered the review behaviors as positive feedbacks by the timestamps. Second, we discarded users with fewer than five related-item interactions. Then, we transformed the users' review data to become a sequential dataset indicating the order of each user's purchase items.

For each user, we split the user's historical sequences S_u into three parts, as done by Kang and McAuley [33], and Wu et al. [34]: (1) the most recent interaction in S_u as the testing set, (2) the next interaction as the validation set, and (3) the remaining interactions as the training set.

B. Evaluation Metric

We used two common Top-K recommendations: the hit rate (HR@10) and normalized discounted cumulative gain (nDCG@10). Here, HR@10 is the rate of positive items in the top-10 recommended items, and nDCG@10 is a ranking measurement for the positions of the positive items in the top-10 recommended items. For the computational cost, we followed the previous mentioned works [33], [34]. We randomly sampled 100 negative and 1 positive item for each user and ranked them for evaluation.

C. Comparison Methods

To evaluate the performance of AFTRec, we compared it with the following eight competitive baselines:

POP: Simple baseline recommendation model that recommends the most popular items in the training set.

CML: CF-based method that applies metric learning instead of MF. It learns a metric to minimize similar user and item pairs.

FPMC: Sequential RS that combines MF and factorized first-order MC. It captures long-term user interests and item-to-item transitions by utilizing the characteristics of both methods. TransRec: Baseline translation-based method for sequential recommendations. It embeds users and items into the transition space and models three-component relationships between a user, previously visited items, and target item.

MoHR: Translation-based method that minimizes the distance between relevant item pairs in the translation space. It exhibits different relation types (e.g., also-viewed/also-bought) between user and item pairs and is integrated into the translational embedding model.

SASRec: Self-attention-based sequential recommendation model inspired by a transformer in NLP. It captures the long-term user interest in predicting the next item through multiple stacked SABs.

TiSASRec: Self-attention-based sequential recommendation model. Unlike SASRec, which considers the absolute time position of items, TiSASRec uses relative time intervals for positioning the encodings of items in stacked SABs.

D. Implementation Details

During the experiments, we implemented AFTRec using the Adam optimizer with momentum exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We set the batch size to 128 and the maximum sequence length to 50 for all datasets. In AFTRec, we set the number of SABs to two and used single-head self-attention layers to generate the user translation vector. We set the number of links in the transaction graph to three for learning the item relations. For comparison with competitive baselines, the hyperparameters were tuned through a grid search. The learning rate was {0.1, 0.001, 0.0001, 0.00001}, and the dropout rate was {0.2, 0.5}. For SASRec and TiSASRec, we set the number of SABs to two and used single-head self-attention layers. For SASRec and TiSASRec, the embedding dimensions were set to 50. For TransRec and MoHR, the embedding dimensions were set to 10. Except for POP, CML, FPMC, and TransRec, the batch size was set to 128. For SASRec and TiSASRec, the maximum sequence lengths were 50. We set all other parameters according to the respective baseline papers.

E. Recommendation Performance

Tables II and III show a performance comparison of sequential recommendations and translation-based recommendations with HR@10 and nDCG@10 on four sparse datasets and one dense dataset. On sparse datasets, AFTRec achieved the best performance for both the HR@10 and nDCG@10 metrics. These results show that AFTRec outperforms sequential recommendations using only the self-attention mechanism and translation-based sequential recommendations to resolve the data sparsity problem in the data-sparse environment such as e-commerce recommendation. Several observations of the competitive baselines are shown in Table II. For the Beauty and Toys datasets, POP, which is a traditional recommendation, achieves the worst performance in terms of nDCG and HR. TiSASRec achieved the second-best performance among the baseline methods in terms of nDCG and HR on the Beauty, Toys, and Clothing datasets. In addition, SASRec achieved the second-best performance in nDCG and HR among the baselines on the Beauty dataset.

The proposed model showed better nDCG@10 performance than the existing model for all datasets and better HR@10 performance than the existing models on sparse datasets (Table III). In particular, the proposed model showed the greatest improvement in nDCG and HR performance compared to the existing embedding-based recommendation model for the Clothing dataset. For all datasets, CML, which applies a metric function instead of MF, achieved the worst performance in terms of nDCG and HR. For sparse datasets, MoHR achieved the second-best performance in terms of nDCG and HR.

Compared with these baselines, the proposed AFTRec achieved the best performance on the four datasets. This is because our method represents the user's short-term and long-term interests as user translation vectors through self-attention to user sequences and

TABLE I. Statistics of Datasets Used in Evaluations

Dataset	# Users	# Items	# Actions	Avg of actions/user
Automotive	34,315	40,287	183,567	5.35
Beauty	52,204	57,289	394,908	7.56
Clothing	184,050	174,484	1,068,972	5.81
Toys	57,617	69,147	410,920	7.39
Steam	335,730	13,047	4,213,117	12.59

TABLE II. Comparison of Recommendation Performance on Five Public Datasets and Four Sequential Recommendations. The Best Performing Method is in Boldface. The Latent Dimension Size D for All Baselines Was Set to 50

Dataset	Metric	PopRec	FPMC	SASRec	Ti-SASRec	AFTRec
Automotive	nDCG@10	0.2084	0.1981	0.2288	0.2509	**0.4875**
	HR@10	0.3481	0.3210	0.3716	0.4032	**0.8992**
Beauty	nDCG@10	0.2277	0.2532	0.3211	0.3126	**0.4325**
	HR@10	0.4003	0.4070	0.4852	0.4734	**0.8571**
Clothing	nDCG@10	0.2166	0.2076	0.2214	0.2445	**0.4667**
	HR@10	0.3661	0.3478	0.3853	0.3974	**0.8872**
Toys	nDCG@10	0.2048	0.2651	0.3136	0.3177	**0.4730**
	HR@10	0.3601	0.4170	0.4663	0.4920	**0.8596**
Steam	nDCG@10	0.4728	0.5297	0.6211	**0.6228**	0.5716
	HR@10	0.7297	0.7830	0.8716	0.8657	**0.9036**

TABLE III. Comparison of Recommendation Performance on Five Public Datasets and Three Translation-Based Sequential Recommendations. The Best Performing Method Is in Boldface. The Latent Dimension Size D for All Baselines Was Set to 10

Dataset	Metric	CML	TransRec	MoHR	AFTRec
Automotive	nDCG@10	0.1793	0.2034	0.3478	**0.3845**
	HR@10	0.3062	0.3332	0.5382	**0.7260**
Beauty	nDCG@10	0.2532	0.2666	0.3635	**0.4004**
	HR@10	0.4070	0.4125	0.5550	**0.7416**
Clothing	nDCG@10	0.1904	0.2111	0.3015	**0.4457**
	HR@10	0.3307	0.3608	0.4919	**0.7024**
Toys	nDCG@10	0.2437	0.2890	0.4151	**0.4185**
	HR@10	0.4015	0.4474	0.6061	**0.7734**
Steam	nDCG@10	0.4699	0.5287	0.5598	**0.5835**
	HR@10	0.7481	0.7842	**0.7983**	0.7020

high-level item relations as item translation vectors. By mapping user- and item-specific vectors onto the head and the relation into the transition space, we can utilize the advantages of self-attention-based methods and translation principles. In addition, we modeled the interactions between the user and target item efficiently by optimizing the translational embedding model, which considers the directions of both user and target item vectors toward a FT. This shows that the modeling of translational relationships with users, items, and heterogeneous items is generally efficient in capturing a user's long-term interest and short-term item transitions by leveraging a translation function for the given user-to-item interactions. Neural-network-based sequential recommendations are generally superior for predicting personal recommendations on relatively large datasets with respect to interactions between users and items. In contrast, on relatively small datasets with respect to interactions between users and items, translation-based sequential recommendations can provide better recommendations by utilizing interactions between user and item translation vectors captured by transitional principle-based KG embedding techniques.

F. Limitations for AFTRec

Tables II and III show a performance comparison of sequential recommendations and translation-based recommendations with HR@10 and nDCG@10 on four sparse datasets and one dense dataset. For the Steam dataset, which is a dense dataset (Table II), TiSASRec achieved the best performance in terms of nDCG. In contrast, AFTRec achieved the best performance in terms of HT. For nDCG, self-attention-based models were advantageous for predicting the next item for dense datasets in sequential recommendations. However, AFTRec applies a self-attention mechanism to generate user-specific item translation vectors. Therefore, a user's item preferences with self-attention are advantageous for showing candidate items that include true items in terms of HR. Because the proposed model comprehensively learns the user's purchase characteristics and the comprehensive correlations between the users' purchased items, the nDCG performance is slightly lowered, but our model shows better HR performance, indicating whether the true item is exposed to the recommendation candidates. Among the sparse datasets, the proposed model showed the greatest performance improvement on the Automotive dataset, which is a representative sparse dataset, and the experimental results show that the proposed model has better recommendation performance than the existing models.

On the Steam dataset, MoHR achieved the best performance in terms of nDCG with dimensions of 10 (Table III). Considering that the user-specific item translation vector and correlation-specific item translation vector generated by the proposed model are trained by a neural network, the experimental results show that the recommendation performance of the proposed model is slightly lower

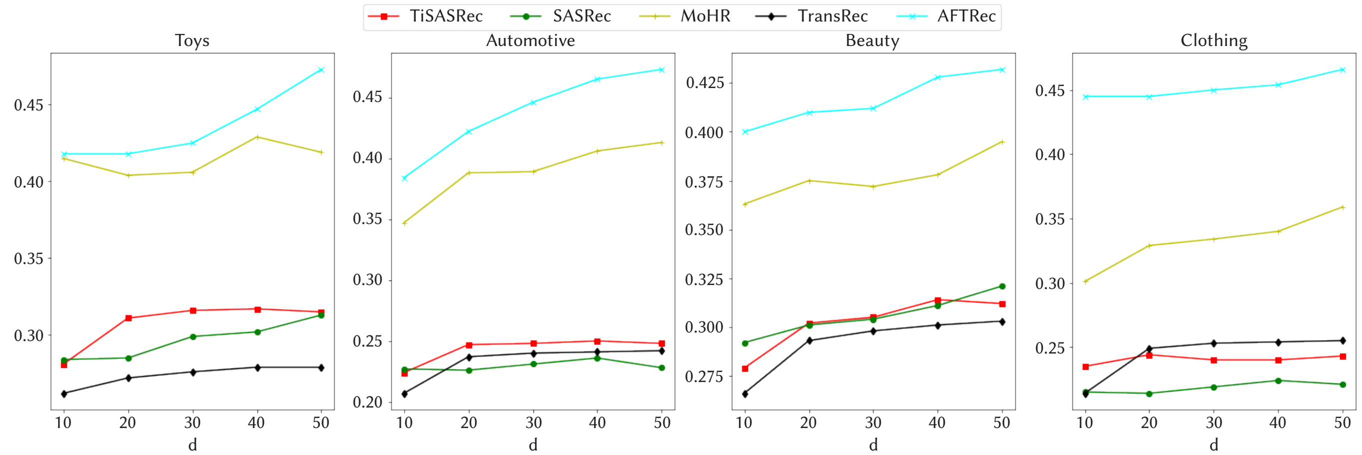

Fig. 7. Comparison of recommendation performance on four datasets (nDCG) with varying latent dimension size d of 10 to 50.

TABLE IV. Comparison of Recommendation Performance on Four Datasets (NDCG) When Varying the Number of Self-Attention Blocks (SABs) of 1 to 3

Dataset	Metric	Number of SABs		
		1	2	3
Automotive	nDCG@10	0.4371	0.4875	0.4105
Beauty	nDCG@10	0.4073	0.4325	0.4264
Clothing	nDCG@10	0.4016	0.4667	0.4090
Toys	nDCG@10	0.3819	0.4730	0.4083
Steam	nDCG@10	0.5104	0.5716	0.5367

than that of MoHR when the size of the data dimension is 10. However, because the proposed model learned various features extracted from purchased items as item translation vectors, it showed a higher recommendation performance than the existing recommendation models, which do not properly reflect the sequential purchase characteristics of the user.

G. Hyperparameter Study

We conducted an additional experiment that varied the dimension size on four sparse datasets to investigate the performance changes based on different embedding dimension sizes of d. The dimension size affects the item embedding size of self-attention and GGNN for entities and relations. We changed the dimension sizes from {10, 20, 30, 40, 50}, and the nDCG@10 results are shown in Fig. 7. For the Amazon datasets, our model outperformed the baselines. From Fig. 7, TransRec, MoHR, and AFTRec achieved better performance as the latent dimension $d \geq 30$ increased on the Automotive, Beauty, and Clothing datasets. By contrast, for the Toys dataset, the performance of the MoHR peaked when $d = 40$. A dimension size of 40 represents sufficient information for MoHR on the Toys dataset. In addition, translation-based models generally have more advanced performance than neural network-based models, such as SASRec and TiSASRec, on sparse datasets.

In Fig. 7, SASRec and TiSASRec show the following aspects. The performance of TiSASRec peaks when $d = 40$ on the Toys and Automotive datasets. For Beauty and Clothing datasets, TiSASRec achieved better performance as d increased. In addition, for the Toys and Beauty datasets, SASRec for Automotive and Clothing datasets, the performance of SASRec peaked when $d = 40$. It is indicated that a dimension size of 40 provides sufficient information for SASRec on Automotive and Clothing datasets. Thus, we find that the dimension size d affects the model's ability to represent sufficient information for user preferences.

We also changed the number of SABs to efficiently learn more complex global preferences of users (Table IV). For all datasets, AFTRec exhibited the best performance on nDCG@10 when using two SABs. The performance of AFTRec increased until the number of SABs was set to two, but AFTRec decreased performance with more than two SABs. From these results, we found that AFTRec has a more stable performance with two SABs.

V. Conclusion

In this study, we proposed AFTRec, a novel translation-based sequential recommendation method for sequential personal historical behaviors for the data sparsity problem. The process maps user preferences and sophisticated item relations to embedding vectors to model the interactions between users and items using the transitional principle. The proposed method includes three main processes. First, for the user-specific item translation vector, we utilized SABs to adaptively capture short- and long-term user preferences in user historical sequences. Second, we designed a transaction graph that links relevant items in terms of timestamps. We applied a GGNN to the transaction graph to generate the item vector, which represents complex interactions between chronologically relevant items and embeds a correlation-specific item translation vector for each item. Third, we employed an attentive user vector using a soft-attention mechanism to jointly learn user-to-item relations in diverse forms of user embedding. After considering the item translation vectors as the heads and the relation vectors, AFTRec models the interactions between the user and items in the same translation space. Because our translational embedding model considers the direction of the embedding vectors, it flexibly provides suitable recommendations for user preferences.

We conducted experiments to evaluate our method on the Automotive, Clothing, Beauty, and Toys datasets collected by the Amazon platform and the Game dataset collected by the Steam platform. The experimental results demonstrate that our method outperforms state-of-the-art baselines in terms of both nDCG and HR on a sparse dataset. Therefore, the experimental results demonstrate that our model is appropriate for predicting the next item in sparse datasets. In the future, we plan to improve the performance of our model and extend it by incorporating complex context-level user information, such as user groups, locations, and devices.

Acknowledgment

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program (IITP-2022-RS-2022-00156360) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

References

[1] C. Feng, J. Liang, P. Song, and Z. Wang, “A fusion collaborative filtering method on sparse data in recommender systems,” *Information Sciences*, vol. 421, pp. 365-379, 2020, doi: 10.1016/j.ins.2020.02.052.

[2] J. Bobadilla, S. Alonso, and A. Hernando, “Deep learning architecture for collaborative filtering recommender systems,” *Applied Sciences*, vol. 10, no. 7, pp. 2441, 2020, doi:10.3390/app10072441.

[3] A. Gazdar, and L. Hidri, “A new similarity measure for collaborative filtering based recommender systems,” *Knowledge-Based Systems,* vol. 188, 2020, doi: 10.1016/j.knosys.2019.105058.

[4] Y. Koren, R. Bell, and Chris Volinsky, “Matrix factorization techniques for recommender systems,” *Computer,* vol. 42, no. 8 pp. 30-37, 2009, doi: 10.1109/MC.2009.263.

[5] D. Jannach, M. Ludewig, and L. Lerche, “Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts,” *User Modeling and User-Adapted Interaction*, vol. 27, pp.351-392, 2017, doi: 10.1007/s11257-017-9194-1.

[6] A. Luo, P. Zhao, Y. Liu, F. Zhuang, D. Wang, J. Xu, et al., “Collaborative Self-Attention Network for Session-based Recommendation,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, Yokohama, Japan, 2020, pp.2591-2597.

[7] S. Sun, Y. Tang, Z. Dai, and F. Zhou, “Self-Attention Network for Session-Based Recommendation with Streaming Data Input,” *IEEE Access*, vol. 7, pp. 110499-110509, 2019, doi: 10.1109/ACCESS.2019.2931945.

[8] D. Hu, L. Wei, W. Zhou, X. Huai, Z, Fang, and S. Hu, “PEN4Rec: Preference Evolution Networks for Session-based Recommendation,*” arXiv preprint arXiv:2106.09306,* 2021.

[9] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, “Factorizing Personalized Markov Chains for Next-basket Recommendation,” in *Proceedings of the 19th International Conference on World Wide Web*, North Carolina, USA, 2010, pp. 811–820.

[10] D. W. Hogg, and D. Foreman-Mackey, “Data analysis recipes: Using markov chain monte carlo,” *The Astrophysical Journal Supplement Series*, vol. 236, no.1, pp.11, 2018, doi: 10.3847/1538-4365/aab76e.

[11] P. Tengkiattrakul, S. Maneeroj, and A. Takasu, “Attentive Hybrid Collaborative Filtering for Rating Conversion in Recommender Systems,” in *Proceedings of the International Conference on Energy, Water and Environment*, Venice, Italy, 2021, pp.151-165.

[12] Y. Zhang, Y. He, J. Wang, and J. Caverlee, “Adaptive Hierarchical Translation-based Sequential Recommendation,” in *Proceedings of the Web Conference 2020,* Taipei, Taiwan, 2020, pp. 2984-2990.

[13] Y. Ding, Y. Ma, W. Wong, and T. S. Chua, “Modeling Instant User Intent and Content-level Transition for Sequential Fashion Recommendation,” *IEEE Transactions on Multimedia*, preprint.

[14] A. Garcia-Duran, R. Gonzalez, D. Onoro-Rubio, M. Niepert, and H. Li, “Transrev: Modeling reviews as translations from users to items,” *Advances in Information Retrieval*, vol.12035, pp.234-248, 2020. doi: 10.1007/978-3-030-45439-5_16.

[15] Q. Wang, Z. Mao, B. Wang, and L. Guo, “Knowledge graph embedding: A survey of approaches and applications,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, 2017, pp. 2724-2743, DOI: 10.1109/TKDE.2017.2754499.

[16] M. Brandalero, M. Shafique, L. Carro, and A. C. S. Beck, “Transrec: Improving adaptability in single-ISA heterogeneous systems with transparent and reconfigurable acceleration,” in *2019 Design, Automation & Test in Europe Conference & Exhibition,* Florence, Italy, 2019, pp. 582-585.

[17] Y. Tay, L. Anh-Tuan, and S. C. Hui, “Latent relational metric learning via memory-based attention for collaborative ranking,” in *Proceedings of the 2018 World Wide Web Conference*, Lyon, France, 2018, pp. 729-739.

[18] C. K. Hsieh, L. Yang, Y. Cui, T. Y. Lin, S. Belongie, and D. Estrin, “Collaborative metric learning,” in *Proceedings of the 26th international conference on world wide web,* Geneva, Switzerland, 2017, pp. 193-201.

[19] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multirelational data,” in Advances in Neural *Information* Processing Systems, Lake Tahoe, Nevada, 2013, pp. 2787– 2795.

[20] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated graph sequence neural networks,” *arXiv preprint arXiv:1511.05493,* 2015.

[21] J. Feng, M. Huang, M. Wang, M. Zhou, Y. Hao and X. Zhu, “Knowledge graph embedding by flexible translation,” in *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*, Cape Town, South Africa, 2017.

[22] *A.* Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., “Attention is all you need,” In *Advances in neural information processing systems,* California, USA, 2017, pp. 5998-6008.

[23] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555,* 2014.

[24] S. Kabbur, X. Ning, and G. Karypis, “Fism: factored item similarity models for top-n recommender systems,” in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, Chicago, USA, 2013, pp. 659-667.

[25] X. He, Z. He, J. Song, Z. Liu, Y. G. Jiang, and T. S. Chua, “Nais: Neural attentive item similarity model for recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, 2018, pp. 2354-2366, doi: 10.1109/TKDE.2018.2831682.

[26] J. Bobadilla, A. Gutiérrez, S. Alonso and A. González-Prieto, “Neural Collaborative Filtering Classification Model to Obtain Prediction Reliabilities,” International Journal of Interactive Multimedia and Artificial Intelligence, vol. 7, no. 4, 2022, pp. 18-26, doi: 10.9781/ijimai.2021.08.010.

[27] J. Bobadilla, R. Lara-Cabrera, A. González-Prieto and F. Ortega, “DeepFair: Deep Learning for Improving Fairness in Recommender Systems,” International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 6, 2021, pp. 86-94, doi: 10.9781/ijimai.2020.11.001.

[28] B. Hidasi, and A. Karatzoglou, “Recurrent neural networks with top-k gains for session-based recommendations,” in *Proceedings of the Conference on Information and Knowledge Management*, Turin, Italy, 2018, pp. 843–852.

[29] P. M. Gabriel De Souza, D. Jannach, and A. M. Da Cunha, “Contextual hybrid session-based news recommendation with recurrent neural networks,” *IEEE Access*, vol. 7, 2019, pp.169185-169203, doi: 10.1109/ACCESS.2019.2954957.

[30] S. Sun, Y. Tang, Z. Dai, and F. Zhou, “Self-attention network for session-based recommendation with streaming data input,” *IEEE Access*, vol. 7, 2019, pp. 110499–110599, doi: 10.1109/ACCESS.2019.2931945.

[31] C. Xu, J. Feng, P. Zhao, F. Zhuang, D. Wang, Y. Liu, et al., “Long- and short-term self-attention network for sequential recommendation,” *Neurocomputing*, vol. 423, 2021, pp. 580-589, doi: 10.1016/j.neucom.2020.10.066.

[32] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, “Neural attentive session-based recommendation,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore, 2017, pp. 1419-1428.

[33] W. C. Kang, and J. McAuley, “Self-attentive sequential recommendation,” in *2018 IEEE International Conference on Data Mining*, Singapore, 2018, pp. 197-206.

[34] L. Wu, S. Li, C. J. Hsieh, and J. Sharpnack, “SSE-PT: Sequential recommendation via personalized transformer,” in *Fourteenth ACM Conference on Recommender Systems*, Brazil, 2020, pp. 328-337.

[35] J. Li, Y. Wang, and J. McAuley, “Time interval aware self-attention for sequential recommendation,” in *Proceedings of the 13th international conference on web search and data mining*, Texas, USA, 2020, pp. 322-330.

[36] J. Tang, and K. Wang, “Personalized top-n sequential recommendation

via convolutional sequence embedding," in *Proceedings of the ACM Conference on Web Search and Data Mining*, California, USA, 2018, pp. 565–573.

[37] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *Proceedings of the ACM Conference on Web Search and Data Mining*, Melbourne, Australia, 2019. pp. 582–590.

[38] W. C. Kang, M. Wan, and J. McAuley, "Recommendation Through Mixtures of Heterogeneous Item Relationships," in *Proceedings of ACM Conference on Information and Knowledge Management*, Turin, Italy, 2018, pp. 1143-1152.

[39] B. Wu, X. He, Z. Sun, L. Chen, and Y. Ye, "ATM: An attentive translation model for next-item recommendation," IEEE Transactions on Industrial Informatics, vol. 16, no. 3, 2020, pp. 1448-1459, doi: 10.1109/TII.2019.2947174.

[40] R. Pasricha, and J. Mcauley, "Translation-based factorization machines for sequential recommendation," in Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, Canada, 2018, pp.63-71.

[41] Y. Zhan, Y. He, J. Wang, and J. Caverlee, "Adoptive hierarchical translation-based sequential recommendation," in Proceedings of the Web Conference 2020, Taipei, Taiwan, 2020, pp. 2984-2990.

[42] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. *20*, no. 1, 2009, pp.61-80, doi: 10.1109/TNN.2008.2005605.

[43] C. Ma, P. Kang, and X. Liu, "Hierarchical gating networks for sequential recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, Alaska, USA, 2019, pp. 825-833.

[44] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of 29th AAAI conference on artificial intelligence*, Texas, USA, 2015, pp. 2181–2187.

[45] T. Lacroix, N. Usunier, and G. Obozinski, "Canonical tensor decomposition for knowledge base completion," in *Proceedings of the International Conference on Machine Learning*, Stockholm, Sweden, 2018, pp. 2863-2872.

[46] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[47] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Thirty-second AAAI conference on artificial intelligence*, Louisiana, USA, 2018.

[48] Z. Sun, Z. H. Deng, J. Y. Nie, and J. Tang, "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space," in *International Conference on Learning Representations*, Louisiana, USA, 2018.

[49] S. M. Kazemi, and D. Poole, "Simple embedding for link prediction in knowledge graphs," in *The Thirty-second Annual Conference on Neural Information Processing Systems*, Montreal, Canada, 2018.

[50] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[51] J. J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, Santiago, Chile, 2015, pp. 43-52.

Min-Ji Seo

She received the B.S. in Computer Science and Engineering (2015) and the M.S. and ph.D in software convergence from Soongsil University, Seoul, Korea, in 2017 and 2022, respectively. Her current research interests are data leak prevention, speech emotion recognition and deep learning.

Myung-Ho Kim

He received the B.S. in Computer Engineering from Soongsil University, Seoul, Korea (1989) and the M.S. and ph.D in Computer Engineering from POSTECH, Pohang, Korea, in 1991 and 1995, respectively. Now he is a professor in department of software convergence from Soongsil University. His current research interests are deep learning and block chain and cloud computing.