IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 June 2023, accepted 10 July 2023, date of publication 20 July 2023, date of current version 26 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3297204

RESEARCH ARTICLE

Autoregressive Decoder With Extracted Gap Sessions for Sequential/Session-Based Recommendation

JAEWON CHUNG[1], JUNG HWA LEE[2], AND BEAKCHEOL JANG[2], (Member, IEEE)
[1]Graduate School of International Studies, Yonsei University, Seoul 03722, South Korea
[2]Graduate School of Information, Yonsei University, Seoul 03722, South Korea

Corresponding author: Beakcheol Jang (bjang@yonsei.ac.kr)

This work was supported by the Yonsei University Research Fund under Grant 2023-22-0104.

ABSTRACT Learning the complex relationships between items in a sequential recommendation system (SRS) and session-based recommendation system (SBRS) is critical for obtaining higher prediction scores. In recent studies, to capture item-item information, items have been represented as the nodes of graph neural networks (GNNs) and the relevance of items with self-/soft attention layers has been calculated. GNNs have been used because standalone attention-based methods focus only on the relative significance of items within a single session, neglecting high-order item-item relationships that change through sessions. The relational summarization task is a natural language processing task that extracts the relationship between two tokens from a related corpus; however, its adaptation to SRS and SBRS is unknown. To fill this lacuna, in this study, the relationships between items from related sessions are extracted using the transformer-based abstractive summarization model PEGASUS. To improve session embedding, the proposed model, named ''gap-session transformer'' utilizes gap-session masking to learn the relationships between items within different sessions. In addition, a group of sessions are divided into multiple corpus sets based on the theme of each corpus, and the autoregressive beam-search decoder is connected to a transformer decoder for the generation of the next session while auxiliary tasks are performed to enhance the recommendation task. Extensive experiments conducted on the MovieLens1M dataset and Yoochoose dataset verify that our model significantly outperforms the state-of-the-art (SOTA) methods, and the results demonstrate the efficacy of the relational summarization task in recommendation systems.

INDEX TERMS Recommender systems, Pegasus, transformer.

I. INTRODUCTION

Sequential recommendation systems (SRSs) and session-based recommendation systems (SBRSs) are the two main types of recommendation systems. First, sequential recommendation predicts an item or group of items for a consumer's next purchase. Because such prediction requires the sequence data of an individual user, sequential recommendation enables personalized recommendations. Sequence data are a list of a user's historical behavior in chronological order without considering the start and the end of each interaction, omitting timestamp data from the inputs [39]. SBRSs have emerged as a way to recommend the next items when long-term user behavior information is unavailable [9]. Instead of determining personalized user-item relationships throughout a single sequence per user, the goal of SBRSs is to capture the item–item or item–session relationships by learning a series of sessions. A session refers to a transaction with several items purchased or rated in one event by an anonymous user.

To improve the accuracy of the recommendations, recent studies on SRSs and SBRSs have adapted the latest machine learning (ML) methodologies of transformers [34] and graph neural networks (GNNs). Recent studies based on transformers have used natural language processing (NLP) models such as bidirectional encoder representations from

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco J. Garcia-Penalvo.

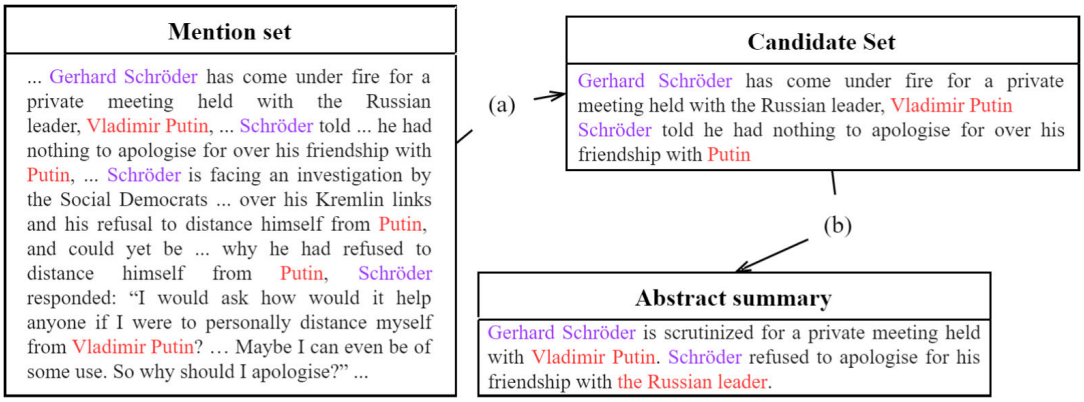

FIGURE 1. Relational summary for the relationship between the former German chancellor Gerhard Schröder and the Russian leader Vladimir Putin, referred to as (w_1) and (w_2), respectively. The "mention set" includes all the statements that contain (w_1) and (w_2). After creating mention set, the "candidate set" is generated (a) by identifying all the sentences in the mention set that coherently represent certain relations between (w_1) and (w_2). Thereafter, the "abstract summary" construction task is performed (b) to select the top θ candidates to generate a summary. It should be noted that unlike an original relational summary, an abstract summary is created, not an extractive summary.

transformers (BERT) [5] and generative pre-trained transformers (GPT) [28]. The authors applied a transformer encoder and decoder to a sequential recommendation because of the resemblance between the historic behavior of users and the text sentences; both are sequence data. To support the recent success of transformer-based NLP models on text translation and generation, SRSs such as those proposed by [17] and [32] replaced tokens from sentence datasets with items from user-item interaction sequences. In addition, [22] suggested that the estimation without time information implicitly indicates that all the adjacent elements of a user behavior list have the same time interval. This is not always true in the real world. Reference [22] further stated that if the time intervals between items differ, so do their effects on the next item recommendation, even if the temporal order is identical. Therefore, adding time-interval information to sequential recommendations is necessary for a more successful prediction.

The motivation of this paper is to develop a recommender system that can be used for both session-based recommendation and sequential recommendation with better performance. This dual purpose is to provide a more convenient experience for our recommendation system users, as previous works are built for only one type of recommendation system. This single-purpose usage refrains users from solving diverse real-world recommendation problems. To resolve this inconvenience, our suggested model can accept both session and sequential datasets as input.

As a method for time-interval-aware sequential recommendation, we introduce the concept of encoding a single-user behavior sequence into several sessions, divided by the time interval threshold. This also holds for the implementation of the latest NLP model—pre-training with extracted gap sentences for abstractive summarization sequence-to-sequence (PEGASUS) [44]. The adaptation of PEGASUS requires breaking down a sequence into multiple sessions, as the PEGASUS decoder considers a list of sentences instead of a single, absolute document. We chose PEGASUS because it has demonstrated superior performance in abstractive text summarization tasks with a substantially small number of fine-tuning datasets than in other models, including BART and the text-to-text transfer transformer (T5) [29].

Finally, we implemented relational summarization for sequential and session-based recommendations to capture the complex item–item relationships. A relational summarization task is a novel task that aims to create a natural language summary of the relation between two lexical tokens in a corpus without the help of a knowledge base [14]. The motivation for this task is to improve the user interface using a concise mind map that can depict the relationship between two entities. Although SBRS and relational summarization seem unrelated, if we replace a token with a session, creating a mind map between multiple sessions can facilitate the understanding of the complex correlations between items. This is because sessions with similar items are closer in embedding space [23]. Conversely, the complex item–item relationship lies in the session–session relationship. Figure 1 illustrates how relational summarization works. The contributions of this study are as follows:

- It proposes session encoding to capture the various time intervals between items for SRSs.
- It implements relational summarization, enabling attention-based SBRSs to capture complex item–item information.
- It introduces a corpus theme to improve loss performance.

II. RELATED WORK

Following [47] and [48], in Table 1, we compare the technical aspects of related works with those of the proposed method.

A. SESSION-BASED RECOMMENDATION SYSTEM

Earlier studies on SBRSs used Markov chains (MCs) and Markov decision processes (MDPs) to learn sequential

TABLE 1. Comparison between related works and our method on technical aspects.

models	sequential/session	awareness	decoding	mechanism
IKNN [30]	session	not applicable	not applicable	KNN
GRU4REC [12]	session	not applicable	not applicable	GRU
NARM [18]	session	not applicable	bi-linear	Attention
STAMP [19]	session	not applicable	not applicable	Attention
NISER [10]	session	sequence-aware, graph-aware	not applicable	GNN
GC-SAN [40]	session	graph-aware	not applicable	GNN+Attention
SR-GNN [35]	session	graph-aware	not applicable	GNN
SGNN-HN [24]	session	graph-aware	not applicable	GNN
GCE-GNN [37]	session	session-aware	not applicable	GNN+Attention
TAGNN [43]	session	target-aware	not applicable	GNN+Attention
CORE [15]	session	not applicable	robust distance measuring	Attention
SimGCL [45]	session	not applicable	contrastive learning	GNN
HMLET [46]	session	not applicable	not applicable	GCN
FPMC [26]	sequential	not applicable	not applicable	Matrix Factorization+Markov Chain
BPRMF [27]	sequential	not applicable	not applicable	Matrix Factorization
SASREC [17]	sequential	not applicable	not applicable	Attention
CASER [33]	sequential	not applicable	not applicable	CNN
SLRC+ [36]	sequential	not applicable	not applicable	Hawkes process
NEXTITNEXT [42]	sequential	not applicable	not applicable	CNN
BERT4REC [32]	sequential	not applicable	not applicable	BERT
TISASREC [22]	sequential	time-aware	not applicable	Attention
GST	both	time-aware	beam search	Pegasus

patterns from past interactions between the user and the item. Item-KNN [30] is an item-based recommendation generation methodology that uses cosine similarity to compute item–item similarities. The recurrent neural network (RNN)-based approach GRU4REC [12] modeled the entire session data as mini-batches that were parallel for each session using the gated recurrent unit (GRU) layers. NARM [18] was one of the first attention-based SBRSs, focusing on the user's main purpose in a given session. STAMP [19] is a short-term attention/memory priority model that captures the current user's interests based on their previous clicks and general interests from long-term memory. Reference [25] was a combination of probabilistic models and LSTM. GNN and graph convolution network (GCN) [4], [16] are emerging technologies that have gained significant attention with respect to SBRSs in recent years. SR-GNN [35] was a significant study that structured graph data based on session sequences. In addition to GNN, GC-SAN [40] used a self-attention mechanism to capture the long-term dependency between items for each session. GCE-GNN [37] learns session- and global-level item-embeddings by modeling pair-wise item transitions of GNN. The latest GCN variants such as hypergraph neural network (HGNN) [8] and hyperbolic convolution network (HGCN) [2] are used to provide higher-order information between items and sessions. Another GCN-based model, HMLET [46], comprises hybrid of linear and non-linear propagation steps. When processing each item node or user, its gating module chooses either of linear or non-linear step. Lastly, SimGCL [45] suggests the method based on contrastive learning, a learning mechanism which well extracts self-supervised signals from the input data.

B. SEQUENTIAL RECOMMENDATION SYSTEM

Similar to SBRSs, early SRS started with the MC and MDP. Item-POP was a naive frequency-based selection model. NCF [13] leveraged a multilayer perceptron (MLP) with collaborative filtering (CF), and FPMC [26] devised a factorized personalized MC using matrix factorization (MF). BPRMF [27] combined matrix factorization and the k-nearest neighborhood approach to optimize a personalized ranking system. Transformer [34] is an encoder-decoder model with an attention mechanism and has gained considerable attention in a broad range of computer science subjects. Self-attention and masked self-attention, also known as "cross attention," have been actively applied to SRSs and SBRSs. Caser [33] used a convolutional neural network (CNN) to capture the local features and global preferences. SLRC [36] trained the item-specific short-term effect and lifetime effects to understand repeated consumption by users. TiSASRec [22] modeled both the absolute positions of items and the time intervals between them using self-attention. Transformer4Rec [6] was an open-source library that allows researchers to use Transformer-based NLP techniques in recommender systems. SASRec [17] generated the next items using a transformer decoder, similar to GPT [28]. BERT [5] is a pre-trained language model that uses a bidirectional autoencoder. It uses the Cloze objective, masking a token from a sentence and using the encoder layers of autoencoder and self-attention mechanism to reconstruct the noised tokens. Because the BERT architecture does not include a decoder, it is not equipped for text generation. Nevertheless, BERT4Rec [32] proved that adjusting BERT into a sequential recommendation can benefit the embedding of sequential patterns. However, BERT's weakness in generative tasks is critical for the generation of the next item and session, as these are also generative tasks. Moreover, recommenders solely based on self-attention capture only within-session item-item relationships, neglecting item-item interactions across the sessions [24]. Hence, we propose a PEGASUS [44]-based recommender that can be used for summarizing

deep item-item information, with the help of a relational summarization task. In addition, PEGASUS is a transformer encoder-decoder model optimized for the abstractive summarization task, ensuring the generation of robust and creative sessions.

III. THE PROPOSED METHOD

In this section, we first present the definitions and notations used in this study. We then describe how session and sequential datasets are processed and modeled as a relational summarization task using PEGASUS. Subsequently, we devised our whole-session masking for training the SBRS and SRS. Finally, we introduce the transformer decoder and integrate length normalization into the autoregressive beam-search decoder to improve the relational summarization tasks.

A. NOTATIONS AND DEFINITIONS

We denote sets of item as $V = \{ v_1, v_2, v_3, \ldots, v_N \}$, where N is the number of items. Let $S = [s_1, s_2, s_3, \ldots, s_M]$ represent the vectors of sessions, where M is the number of sessions. Each session s is a vector of the interacting items v of an anonymous users; it is denoted as $s_m = [v_{1,m}, v_{2,m}, v_{3,m}, \ldots, v_{k,m}](1 \leq m \leq M, 1 \leq k \leq N)$ and $v_{k,m} \in V$. The number of anonymous users equals K, the number of unique values of s. We embedded each session $s \in S$ into the same vector space and let ${a_s}^k \in R^{d^k}$ denote the representation of session s of dimension d^k in the k-th hidden layer of the neural network. The representation of the entire session set is represented as $A^k \in R^{n \times d^k}$. The goal of the SBRS and SRS is to generate predictions for the next item $v_{k+1,m}$ for any given session s.

1) DEFINITION 1. WHOLE SESSION MASKING

Let $E = W(g, c, \beta)$ denote a session masking embedding, where g is a corpus that includes N unique tokens and L unique sequences, c is the number of themes T in the corpus, and β is the input masking probability. We refer to T as a threshold token that creates a subcorpora from the main corpus based on whether a sequence contains T or not. If the sequences contain T, they are in the same subcorpus. The value of c obtained from the experimental setup was larger than 0. Each sequence of g contains two or more tokens and is zero-padded to the maximum sequence length *max seq len*. E can be represented by the matrix size of *max seq len* $\times$ T.

2) DEFINITION 2. PEGASUS DECODER

Given the whole-session masking embedding $E = W(g, c, \beta)$, the PEGASUS decoder $P(E, p) = B(D(E, p), \gamma)$ represents the transformer-based left-to-right autoregressive decoder, where D is the transformer decoder, B is the beam-search decoder, p is a masking probability for attention maps, and γ is the beam size. As suggested by [29], P only reconstructs the masked sessions as a single output sequence.

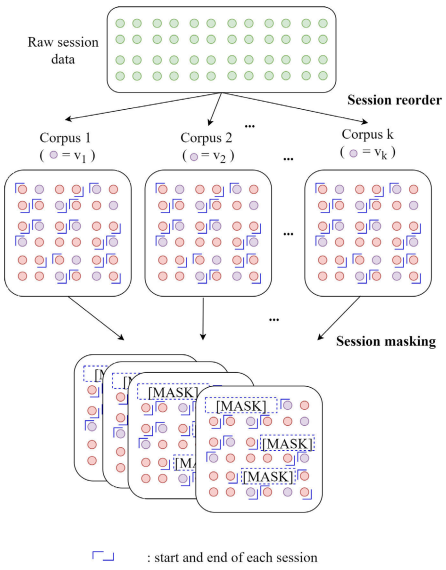

FIGURE 2. An implementation of a relational summary on session embedding. Green dots represent each item $v_1, v_2, \ldots, v_n$, purple dots represent the c most frequent items $v_1, v_2, \ldots, v_c$, and red dots represent the rest of the items $v_{k+1}, \ldots, v_n$. Here, "session reorder" refers to the generation of the mention set and "session masking" refers to the generation of the candidate set. The mention set includes all the sessions from the raw dataset that contains each of the c most frequent items. Once a mention set is created, we sampled candidate sets identifying all the sequences in the mention set that coherently represent the relationship between the most popular items and the rest of the items.

B. RELATIONAL SUMMARIZATION

A relational summarization task comprises three subtasks: mention set generation, candidate set generation, and summary construction tasks. The mention set refers to the corpus where two items co-occur, the candidate set represents a sample extracted from the corpus, and the summary refers to the extractive summary obtained from the candidate set. It should be noted that we use an abstractive summary as a final recommendation instead of an extractive summary. In contrast to extractive summarization, which simply concatenates significant sentences from the document, abstractive summarization paraphrases the document using novel sentences. Using abstractive summarization, we achieved the recommendation of next items that are similar but not the same as the items in the masked sessions. Figure 2 illustrates further details of the mention set and candidate set generation for the raw session dataset. As illustrated, to implement the relational summarization task into a session-based recommendation, we first adjusted the mention set generation task into the session reordering and corpus generation processes. Session reordering involves reordering items and session IDs, matching the IDs to the size of corpus g to avoid out-of-index errors. In corpus generation, we group g using the top c frequent tokens T, gathering sequences with T. This allows the raw data to be divided into corpora with the most popular items, each corpus with a theme related to its threshold token.

C. GAP SESSION GENERATION

After candidate set generation, we developed a whole gap-session generation network to capture deeper item-item relations.

1) SESSION ENCODING

To adequately implement PEGASUS, a single user behavior sequence should be divided into several sessions based on the time intervals between two items. Most SRSs implicitly assume that the time intervals are equal [22]. However, this assumption is incorrect. Therefore, similar to [22], but using a different scaling method with threshold α, we scale the time intervals for session s_a as represented by Equation (1):

$$s_a = \frac{|v_j - v_h|}{\alpha}, (1 \le a \le m) \tag{1}$$

$$\alpha = \frac{(\widetilde{\text{len(s)}} + \min(\text{len(s)}))}{3}, \tag{2}$$

where v_j and v_h are adjacent items at different points in time and $\widetilde{\text{len(s)}}$ is a median of the session length.

2) WHOLE-SESSION MASKING

The whole-session masking embedding function W is the encoder part of the autoencoder. First, W splits the raw data g into c corpora and maps the original corpora X to a latent space F, as expressed by Equation (3):

$$W : X \to F. \tag{3}$$

Following whole-session masking embedding, the embedding E is expressed as follows:

$$E = \{\langle start\rangle, s_1, s_2, [MASK], \ldots, [MASK], \ldots, s_{m-1}\}, \tag{4}$$

D. TRANSFORMER DECODER

Similar to the decoder part of the autoencoder, the objective of the transformer decoder function D is to map F to the output X. Equation (5) represents this:

$$D : F \to X. \tag{5}$$

As illustrated in Figure 3, the transformer decoder of the GST consists of a masked multi-head self-attention layer, multi-head self-attention layer, and position-wise feed-forward layer. To compute cross-attention coupling without using an encoder, following [17], the input of the multi-head self-attention layer is a cache memory initialized by the Xavier uniform initializer and denoted as δ. Subsequently, the output of the self-attention layer H aligns with the latent representations of the masked multi-head self-attention layer:

$$\grave{X} = D(E, p) \tag{6}$$

$$D(E, p) = \text{softmax}(\text{FFNN}(\text{concat}(\grave{v}_1, \ldots, \grave{v}_K)X)E), \tag{7}$$

$$\grave{v}_j = \text{ATT}(\grave{H}Z_{Query}, \grave{\delta}Z_{Key}, \grave{\delta}Z_{Vector}), (1 \le j \le N), \tag{8}$$

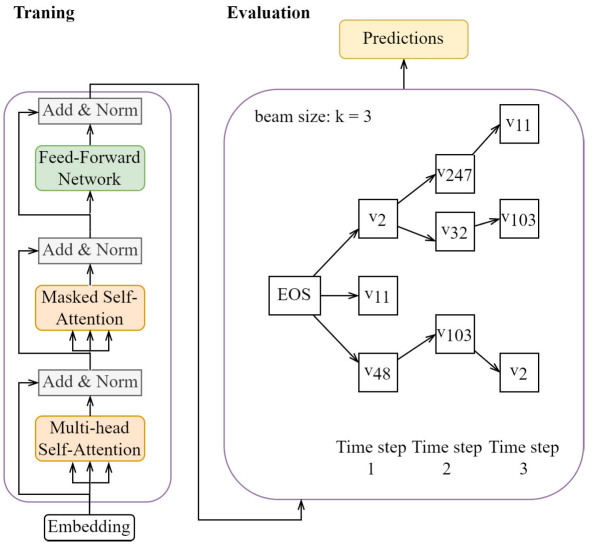

FIGURE 3. Training and evaluation of summary construction task. During the training, a transformer decoder receives embedding as an input. During the evaluation, the output of the transformer decoder is fed to the beam-search decoder to select the top 5 and 10 candidates for the generation of the summary. It should be noted that PEGASUS discovered that masked language modeling (MLM) does not improve downstream tasks for a large number of pre-training steps and, therefore, it was chosen not to include MLM in the final model. MLM was also excluded from GST.

$$\text{ATT}(Query, Key, Vector) = \text{softmax}\left(\frac{QueryKey^{\top}}{\sqrt{d/K}}\right), \tag{9}$$

where $\grave{X}$, $\grave{v}$, $\grave{H}$ respectively denote the reconstructed versions of X, v, and H. K denotes the number of dimensions and $Z \in F$.

E. IMPROVING SRS/SBRS WITH AUTOREGRESSIVE BEAM-SEARCH DECODER

In the context of NLP text generation and summarization, recent studies [1], [44] used beam search instead of greedy decoding for sequence generation at the evaluation stage. Greedy decoding selects the tokens with the best probability at the current timestamp rather than selection based on the global probability. However, greedy decoding does not allow a revision of the past selections even if the predicted sentence is wrong. To mitigate this problem, the beam search selects the tokens with the κ highest probabilities at each time step, where κ denotes the beam size.

1) CONTROLLABLE ABSTRACTIVE SUMMARIZATION

Following PEGASUS, GST adopts length normalization and constraints from [7]. This differs from pure beam search and yields much better results. In addition, we used $2 \times \kappa$ to grow "alive sequences" to differentiate between our prediction of $\kappa = 1$ and greedy decoding. Alive sequences are the sequences that have not generated an end-of-sequence token yet. If the number of alive sequences is 1, the decoding

TABLE 2. Dataset statistics.

Dataset	MovieLens-1M	Yoochoose 1/64
training sessions	99,345	23,670,982
testing sessions	113	55,898
number of items	3,417	37,484
avg. len.	165.499	3.9727

process is equal to greedy decoding, and Equation (10) explains the beam-search score, which is the product of all the probabilities:

$$\text{score}(y_1, \ldots., y_t) = \sum_{i=1}^{t} \log \mathcal{P}(y_i|X, y_1, \ldots, y_{i-1}), \quad (10)$$

where y_i denotes the output sequence at time step i ($1 \leq i \leq t$) and t is the total number of tokens in the predicted sentence.

2) MODEL OPTIMIZATION AND RECOMMENDATION GENERATION

Our pre-training objective is the negative log-likelihood of the masked labels:

$$\mathcal{L} = \frac{1}{|X|} \sum_{v_s \in X} -\log \mathcal{P}(v_s = v_s^*|\acute{X}\ ). \quad (11)$$

We produce the most likely recommendation $\grave{X}$ by maximizing the likelihood $\mathcal{L}$ through beam search:

$$\grave{X}_t = \text{argmax}_{X_i} \mathcal{L}(X_i). \quad (12)$$

Similar to [31], at each time step, we expanded each partial hypothesis in the beam with every possible session in the vocabulary.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

1) DATASETS

We used two popular datasets for the evaluation. 1) MovieLens1M (ML-1M)[1]: Created by Movie-Lens, this sequential dataset is for movie recommenders. We leave out users with less than five interactions and mask sessions with less than three item interactions as [*NO USE*] tokens. The pre-processed ML-1M contains 6040 valid users for 3416 items. 2) Yoochoose 1/64 [2]: Created by RecSys Challenge 2015, this session dataset contains a user clicks on e-commerce data within 6 months. The pre-processed Yoochoose 1/64 has 37483 items.

2) BASELINE METHODS

We compare GST with the following representative methods:

- Item-KNN [30] is an item-based algorithm with item-item similarities computation.
- GRU4REC [12] is a GRU and mini-batch.
- NARM [18] employs a bi-linear matching scheme to learn joint item-session representations.
- STAMP [19] is a short-term memory network that captures the user's current interests.
- NISER [10] applies normalized item and session graph.
- GC-SAN [40] uses self-attention with GNN.
- SR-GNN [35] formulates a session graph from session data by GNN to learn complex transitions of items.
- SGNN-HN [24] captures complex item transition relationships with highway networks.
- GCE-GNN [37] uses a global graph and session graph to capture item representation.
- TAGNN [43] recommends items with target-aware attention.
- CORE [15] is an representation-consistent encoder-decoder model.
- SimGCL [45] is a recommendation model based on contrastive learning.
- HMLET [46] is a hybrid model of non-linear and linear collaborative filtering model.
- FPMC [26] uses MC and MF for user transitions.
- BPRMF [27] optimizes personalized ranking loss based on implicit feedback.
- SASRec [17] employs transformer decoder.
- Caser [33] embeds a sequence of items as an image.
- SLRC+ [36] uses Hawkes Process into CF.
- NextItnet [42] stacks holed CNN layers.
- BERT4Rec [32] uses the Cloze objective to capture bi-directionality.
- TiSASRec [22] uses time-awareness.

3) EVALUATION METRICS

Hit Ratio is the evaluation method we use for the sequential dataset (ML-1M). We put the results of Hit Ratio@5 and Hit Ratio@10. Precision is used as an evaluation metric for the session dataset (Yoochoose 1/64). This is because a session-based recommendation system is categorized as a query suggestion and query suggestion uses Precision for an evaluation metric. We put the results of the Precision@5 and Precision@10.

4) HYPER-PARAMETERS SETTINGS

The dimension size of all proposed models is 64, the size of the intermediate layer is 256, the number of attention heads is 2, and the number of decoder layers is 2. We used the best hyper-parameters for SRS and SBRS baselines.

B. EXPERIMENT RESULTS

Our GST has variations of GST_{seq} and GST_{sess}. GST_{seq} is for sequential recommendation, and GST_{sess} is for session-based recommendation. GST_{seq} has a session encoding process because its inputs are sequence datasets. For GST_{sess}, session encoding is not necessary as the inputs are session datasets. Instead, GST_{sess} has corpus themes to group raw session data for relational summarization tasks. They both contain whole-session masking.

[1] https://grouplens.org/datasets/movielens/1m/
[2] https://www.kaggle.com/datasets/chadgostopp/recsys-challenge-2015

TABLE 3. Performances of all comparison methods on the ML-1M dataset with 21 baselines.

SBRSs baselines	HR@5	HR@10	SRSs baselines	HR@5	HR@10
IKNN	0.3684	0.5207	FPMC	0.4286	0.5876
GRU4REC	0.3008	0.4492	BPRMF	0.3051	0.4546
NARM	0.4490	0.6227	SASREC	0.48079	0.4530
STAMP	0.1528	0.2358	CASER	0.4026	0.64619
NISER	0.2065	0.2916	SLRC+	0.3187	0.5705
GC-SAN	0.1377	0.2129	NEXTITNEXT	0.182	0.2702
SR-GNN	0.3871	0.5626	BERT4REC	0.57682	0.69205
SGNN-HN	0.1869	0.281	TISASREC	0.5121	0.6674
GCE-GNN	0.1656	0.2449	GST_{seq}	**0.57699**	**0.71043**
TAGNN	0.2334	0.3255	Improvement	0.0295%	2.6559%
CORE	0.0806	0.1629			
SimGCL	0.472	0.6275			
HMLET	0.3796	0.0718			

TABLE 4. Performances of all comparison methods on the Yoochoose 1/64 dataset with 21 baselines.

SBRSs baselines	P@5	P@10	SRSs baselines	P@5	P@10
IKNN	0.1402	0.1497	FPMC	0.3397	0.2012
GRU4REC	0.3717	0.2179	BPRMF	0.094	0.0647
NARM	0.3754	0.219	SASREC	0.3832	0.2229
STAMP	0.3723	0.216	CASER	0.1430	0.0798
NISER	0.3801	0.2214	SLRC+	0.1615	0.0859
GC-SAN	0.3804	0.2214	NEXTITNEXT	0.2163	0.1493
SR-GNN	0.372	0.2175	BERT4REC	0.2953	0.1818
SGNN-HN	0.3814	0.222	TISASREC	0.1615	0.0875
GCE-GNN	0.3825	0.2226	GST_{sess}	**0.5999**	**0.2999**
TAGNN	0.0866	0.0565	Improvement	56.55 %	34.54 %
CORE	0.3805	0.2224			
SimGCL	0.1566	0.1624			
HMLET	0.0997	0.0718			

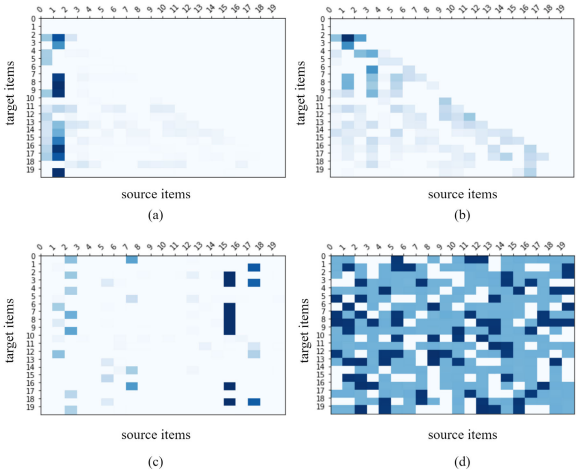

FIGURE 4. Attention heat maps (ML-1M: (a) , (b). Yoochoose 1/64: (c), (d)). (a) is GST_{seq}, (b) is GST with token encoding, (c) is GST_{sess}, and (d) is GST without relational summarization (i.e. no corpus generation).

1) OVERALL COMPARISON

The overall performance result of the experiments is reported in Table 3 and Table 4. We highlight the best results of the ML-1M and Yoochoose 1/64 dataset in boldface. Here, GST_{seq} and GST_{sess} are evaluated, compared to other SRSs and SBRSs. We believe 21 baselines support the credibility of our experimental results. The improvements are computed by the difference between the result of the best baseline and GST_{seq} and GST_{sess} divided by the former. From the analysis of Table 3 and Table 4, we can figure out the following conclusions.

- Unsurprisingly, in Table 3, the recently suggested works of sequential recommendation systems outperform session-based recommendation systems in personalized recommendations. This is understandable as SBRSs are trained to recommend for each session, rather than each user. Still, even traditional SRSs (i.e. BPRMF and FPMC) have better results than the recent SBRS models (i.e. SGNN-HN, GCE-GNN, TAGNN, and CORE). This analysis leads to the necessity of methods that encompass both recommendation systems without performance degradation. Also, standalone transformer-based SRSs (i.e. SASRec, BERT4Rec, and GST) show incredible performance compared to others. This result confirms that transformer blocks significantly benefit personalized recommendation models. Furthermore, the robustness of time awareness in TiSASRec and GST proves that converting a user sequence into time-aware sessions improves recommendation results.
- In Table 3, for the baseline models with GNN (i.e. NISER, GC-SAN, SR-GNN, SGNN-HN, GCE-GNN, and TAGNN), it is obvious that session-graph structures are generally not good at capturing user-item relationships. This is probably because the focus of these models is the relationships between items rather than those between users and items. However, because each session

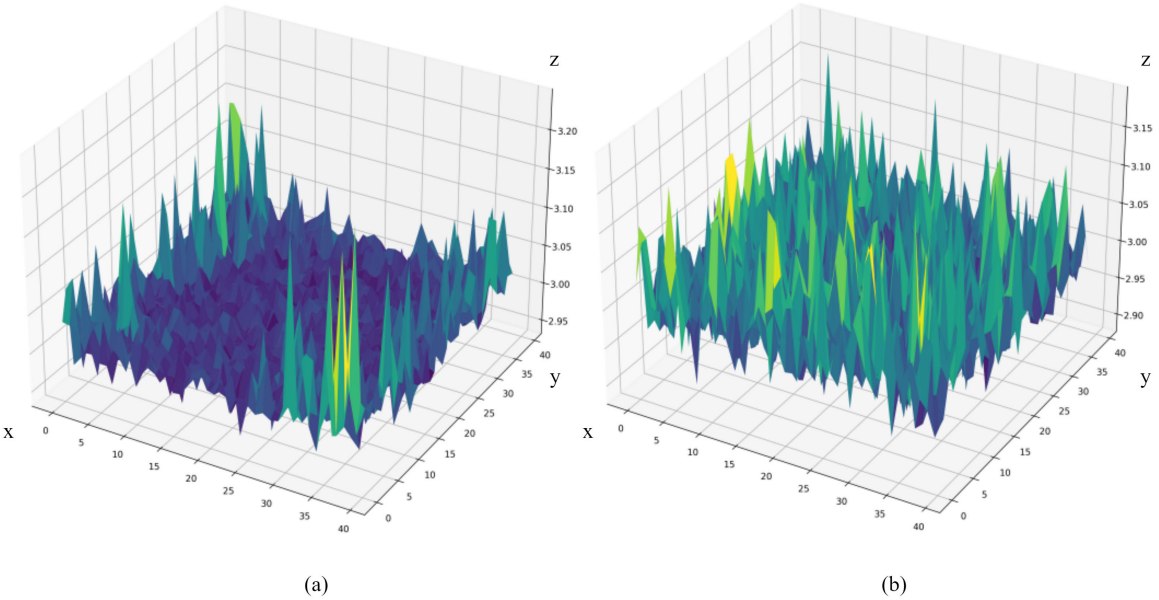

FIGURE 5. Loss landscape of GST_{sess} by corpus's theme size on Yoochoose 1/64. (a) is theme size 1 and (b) is theme size 7.

is a representation of each anonymous user, capturing global user interests can be beneficial. For example, SR-GNN considers global transitions accompanied by local interests and gets higher scores compared to the other SBRSs.

- In Table 4, GST_{sess} performs better than all SBRSs and SRSs baselines. Also, without surprise, most SBRSs achieve better scores than SRSs as SBRSs are optimized for session datasets. However, SASRec, which is an SRS, has the best result except for the proposed model. Interestingly, we find other self-attention methods such as BERT4Rec and TiSASRec do not show good performance. Even traditional models like FPMC have better precision scores than them. This demonstrates using self-attention is trivial to the success of GST_{sess}.
- Contrary to the results of Table 3, in Table 4, the GNN-based baseline models including NISER, GC-SAN, SR-GNN, and GCE-GNN show that graph structure is good for understanding the relationship between items in the session dataset. This is because, in each model, learning graph structures focuses on how the session is structured, which comprises items. Also, there could be advantages to learning hypergraphs as graphs usually have more information than one-dimensional sequential data. Nevertheless, without these advantages of GNN, our proposed model showed an impressive improvement in precision score.
- The improvement of performance in Table 3 and Table 4 differs largely. This is perhaps because of the slight difference in data preprocessing of GST_{seq} and GST_{sess}. In GST_{seq}, we sliced a sequence of user-item interactions into sessions for a better understanding of the time interval. On the other hand, in GST_{sess}, we followed the data preprocessing of [15]. For the training dataset, we combined sessions into sequences and renumbered items. For the test dataset, we converted sessions into sequences and skipped the items not in the training set. Instead of suggesting a method to predict the ability of the proposed method, our 21 baselines provide a good reason to believe that both proposed methods (GST_{seq} and GST_{sess}) are capable of achieving high-performance results as shown in Table 3 and Table 4.
- In Table 3 and Table 4, our proposed GST models show significant superiority over all the baselines. Compared to SASRec and BERT4Rec, GST has three advantages: (1) It captures different time intervals between two adjacent items. This allows our model to understand a user behavior sequence as a set of sessions and makes better recommendations. (2) Also, our model reconstructs sessions instead of tokens and then summarizes sequential or session patterns of an individual user. The resulting summary includes both global and local user interests, while others only consider the closest user-item/item-item interactions for prediction. (3) Finally, our model has a beam-search decoder that generates recommendations with higher prediction scores.

C. ABLATION STUDY

1) IMPACT OF SESSION ENCODING AND RELATIONAL SUMMARIZATION

Figure 4 shows the attention score of GST_{seq} and GST_{sess}. To study the impacts of session encoding and corpus generation, we extract attention weights from training. The x-axis of the graphs represents the 20 items GST recommended, and the y-axis is the 20 real target items. The attention scores are log probabilities after softmax. Note that (a) and (c) have

higher Hit Ratio@5 and Hit Ratio@10 scores than (b) and (d), though we only include attention maps here because of the space limitations. For (a), the weights tend to position on the left side of the recommendation, while they are broadly distributed in (b). Likewise, (c) has a more dense attention map compared to the attention weights of (d). As (a) and (c) perform better than (b) and (d), we conclude that having a denser attention map represents a better understanding of complex user-item and item-item transitions. Therefore, session encoding and relational summarization tasks significantly benefit SRS and SBRS.

2) IMPACT OF CORPUS THEME

Figure 5 illustrates the loss landscape by the size of the corpus theme. As defined in Definition 1, a theme is a key item for dividing raw datasets into multiple corpora. For example, if the theme size is 1, there is a single corpus. A loss landscape visualizes loss changes by parameters. We followed [20] to visualize the loss landscape of our models in 3D. For a fair comparison, all hyper-parameters except for the theme size are identical for (a) and (b). In the graphs, the ups and downs of the z-axis represent how loss increases and decreases, and the x- and y-axis represent contouring resolution. When the theme size is 7, the loss landscape is very complicated and sharp, implying that initialization can have a huge influence on training [20]. On the contrary, when the theme size is 1, the loss landscape is more flat and simple, meaning the model would be easier to train. Based on these observations, we conclude that corpus theme significantly affects the performance of GST and a smaller theme size is more beneficial than a larger theme size.

V. CONCLUSION

For a sequential dataset, existing standalone attention-based SBRSs lack the understanding of high-order item-item information that changes through sessions. Also, in SRSs, attention-based models neglect time intervals between each interaction. For a session dataset, most attention-based SRSs experience more challenges than their SBRS counterparts in learning item-item interactions. This result validates the triviality of the attention mechanism in the session-based recommendation. Moreover, GNN-based models show relatively higher performances than models based on attention. This is because of the advantages of learning more information on item-item interactions through many graphs. To mitigate these challenges, we designed a gap session generator with session encoding, corpus generation, and session masking. As a result, our proposed model achieved SOTA scores on all 21 baselines for both the session dataset and the sequential dataset. Extensive experiments also demonstrate session encoding and relational summarization have significant effectiveness.

REFERENCES

[1] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020, *arXiv:2004.05150*.

[2] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[3] T. Chen and R. C.-W. Wong, "Handling information loss of graph neural networks for session-based recommendation," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1–10.

[4] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[6] G. de Souza Pereira Moreira, S. Rabhi, J. M. Lee, R. Ak, and E. Oldridge, "Transformers4Rec: Bridging the gap between NLP and sequential/session-based recommendation," in *Proc. 15th ACM Conf. Recommender Syst.*, Sep. 2021, pp. 143–153.

[7] A. Fan, D. Grangier, and M. Auli, "Controllable abstractive summarization," 2017, *arXiv:1711.05217*.

[8] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3558–3565.

[9] L. Guo, H. Yin, Q. Wang, T. Chen, A. Zhou, and N. Q. V. Hung, "Streaming session-based recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1569–1577.

[10] P. Gupta, D. Garg, P. Malhotra, L. Vig, and G. Shroff, "NISER: Normalized item and session representations to handle popularity bias," 2019, *arXiv:1909.04276*.

[11] N. Guo, X. Liu, S. Li, Q. Ma, Y. Zhao, B. Han, L. Zheng, K. Gao, and X. Guo, "HCGR: Hyperbolic contrastive graph representation learning for session-based recommendation," 2021, *arXiv:2107.05366*.

[12] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015, *arXiv:1511.06939*.

[13] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.

[14] A. Handler and B. O'Connor, "Relational summarization for corpus analysis," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 1760–1769.

[15] Y. Hou, B. Hu, Z. Zhang, and W. Xin Zhao, "CORE: Simple and effective session-based recommendation within consistent representation space," 2022, *arXiv:2204.11067*.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[17] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 197–206.

[18] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 1419–1428.

[19] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "STAMP: Short-term attention/memory priority model for session-based recommendation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1831–1839.

[20] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[21] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019, *arXiv:1910.13461*.

[22] J. Li, Y. Wang, and J. McAuley, "Time interval aware self-attention for sequential recommendation," in *Proc. 13th Int. Conf. Web Search Data Mining*, Jan. 2020, pp. 322–330.

[23] L. Qu, R. Gemulla, and G. Weikum, "A weakly supervised model for sentence-level semantic orientation analysis with multiple experts," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2012, pp. 149–159.

[24] Z. Pan, F. Cai, W. Chen, H. Chen, and M. de Rijke, "Star graph neural networks for session-based recommendation," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1–12.

[25] C. Panagiotakis and H. Papadakis, "Session-based recommendation by combining probabilistic models and LSTM," in *Proc. Recommender Syst. Challenge*, 2022, pp. 39–44.

[26] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. 19th Int. Conf. World Wide web*, Apr. 2010, pp. 811–820.
[27] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, *arXiv:1205.2618*.
[28] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.
[29] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.
[30] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Apr. 2001, pp. 285–295.
[31] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.
[32] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1441–1450.
[33] J. Tang and K. Wang, "Personalized top-*N* sequential recommendation via convolutional sequence embedding," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 1–9.
[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
[35] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 1–8.
[36] C. Wang, M. Zhang, W. Ma, Y. Liu, and S. Ma, "Modeling item-specific temporal dynamics of repeat consumption for recommender systems," in *Proc. World Wide Web Conf.*, May 2019, pp. 1977–1987.
[37] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, and M. Qiu, "Global context enhanced graph neural networks for session-based recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 169–178.
[38] C. Wang, M. Zhang, W. Ma, Y. Liu, and S. Ma, "Make it a chorus: Knowledge- and time-aware item modeling for sequential recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 109–118.
[39] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," *ACM Comput. Surv.*, vol. 54, no. 7, pp. 1–38, Sep. 2022.
[40] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, "Graph contextualized self-attention network for session-based recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3940–3946.
[41] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, "Self-supervised hypergraph convolutional networks for session-based recommendation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 1–9.
[42] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 1–6.
[43] F. Yu, Y. Zhu, Q. Liu, S. Wu, L. Wang, and T. Tan, "TAGNN: Target attentive graph neural networks for session-based recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1921–1924.
[44] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 11328–11339.
[45] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary? Simple graph contrastive learning for recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 1–15.
[46] T. Kong, T. Kim, J. Jeon, J. Choi, Y.-C. Lee, N. Park, and S.-W. Kim, "Linear, or non-linear, that is the question!" in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, 2022, pp. 517–525.
[47] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: Challenges, progress and prospects," 2019, *arXiv:2001.04830*.
[48] L. Xu, J. Zeng, W. Peng, H. Wu, K. Yue, H. Ding, L. Zhang, and X. Wang, "Modeling and predicting user preferences with multiple item attributes for sequential recommendations," *Knowl.-Based Syst.*, vol. 260, Jan. 2023, Art. no. 110174.

JAEWON CHUNG was born in Seoul, Republic of Korea, in 1997. She received the B.H. degree in Japanese language and literature from Sungshin Women's University, Republic of Korea, in 2021. She is currently pursuing the M.A. degree with Yonsei University. Since 2022, she has been doing an internship with SK Chemicals on data analysis and office automation. Her research interests include recommendation systems, bigdata analysis, and time-series forecasting.

JUNG HWA LEE was born in Seoul, Republic of Korea, in 1976. He received the B.S. and M.S. degrees in computer engineering from Hongik University, Republic of Korea, in 2004. He is currently pursuing the Ph.D. degree with Yonsei University. From 2000 to 2018, he was a Researcher with the CTO Laboratory, LG Electronics. Since 2018, he has been an AI Engineer with TBrain, SK Telecom. His research interests include artificial intelligence and applications, robot, and the Internet of Things. He has multimodal-AI related patents and has presented at several conferences with his robots.

BEAKCHEOL JANG (Member, IEEE) received the B.S. degree in computer science from Yonsei University, in 2001, the M.S. degree in computer science from the Korea Advanced Institute of Science and Technology, in 2002, and the Ph.D. degree in computer science from North Carolina State University, in 2009. He is currently an Associate Professor with the Graduate School of Information, Yonsei University. His research interests include bigdata analytics, artificial intelligence, natural language processing, and computer networks. He is a member of the ACM.