A Relevant and Diverse Retrieval-enhanced Data Augmentation Framework for Sequential Recommendation

Shuqing Bian[†]
School of Information
Renmin University of China
Beijing, China
bianshuqing@ruc.edu.cn

Wayne Xin Zhao[*]
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
batmanfly@gmail.com

Jinpeng Wang
Meituan Group
Beijing, China
wjp.pku@gmail.com

Ji-Rong Wen[◇]
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
jrwen@ruc.edu.cn

ABSTRACT

Within online platforms, it is critical to capture the semantics of sequential user behaviors for accurately predicting user interests. Recently, significant progress has been made in sequential recommendation with deep learning. However, existing neural sequential recommendation models may not perform well in practice due to the sparsity of the real-world data especially in cold-start scenarios.

To tackle this problem, we propose the model **ReDA**, which stands for Retrieval-enhanced Data Augmentation for modeling sequential user behaviors. The main idea of our approach is to leverage the related information from similar users for generating both *relevant* and *diverse* augmentation. First, we train a neural retriever to retrieve the augmentation users according to the semantic similarity between user representations, and then conduct two types of data augmentation to generate augmented user representations. Furthermore, these augmented data are incorporated in a contrastive learning framework for learning more capable representations. Extensive experiments conducted on both public and industry datasets demonstrate the superiority of our proposed method over existing state-of-the-art methods, especially when only limited training data is available.

CCS CONCEPTS

• **Information systems** → **Personalization**.

† This work was done during intership at Meituan.
∗ Corresponding author. Xin Zhao is also with Beijing Key Laboratory of Big Data Management and Analysis Methods and Beijing Academy of Artificial Intelligence.
◇ Ji-Rong Wen is also with School of Information, Renmin University of China.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557071

KEYWORDS

User Behavior Modeling, Data Augmentation

ACM Reference Format:
Shuqing Bian[†], Wayne Xin Zhao[*], Jinpeng Wang, and Ji-Rong Wen[◇]. 2022. A Relevant and Diverse Retrieval-enhanced Data Augmentation Framework for Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557071

1 INTRODUCTION

In recent years, online platforms (such as *Amazon* and *Meituan*) have supported many daily activities in our personal lives, from shopping, entertainment to education. In order to provide better service, it is critical to model the dynamic user behaviors over time. For this purpose, the task of sequential recommendation has been widely studied in the literature [27, 33].

As a major technical approach, sequential recommendation methods [6, 16] aim to capture sequential patterns or characteristics underlying users' historical behaviors. Such an approach has been further empowered with the recent advance of deep learning. A number of studies have employed recurrent neural networks (RNNs) [13], convolutional neural networks (CNNs) [33], or self-attentive Transformer [16] to learn effective representations of user preference by modeling sequential interaction sequences.

Despite the progress of existing approaches, they often suffer from the issue of data sparsity due to scarce user-item interaction [11], especially in cold-start scenarios. Without sufficient training data, the modeling capacity of a neural sequence network (*e.g.,* Transformer [35]) will be highly limited, leading to a significant performance decrease on the recommendation task. In recent years, many efforts have been devoted to tackling this challenge [4, 5]. A mainstream technique is to develop data augmentation methods based on either heuristic [24, 40] or model based methods [23, 44], and then improve the modeling ability of the recommender with specially designed objectives (*e.g.,* a typical contrastive loss). Among these works, heuristic methods directly modify the input sequences with simple editing strategies (*e.g.,* crop, mask and reorder), while

model based methods often utilize a learnable generator to produce augmented items or representations. Since these studies incorporate more augmented data for training, it can improve the model generalizability and robustness to some extent, given the sparse interaction data.

However, a fundamental issue of data augmentation remains to be solved for sequential recommendation, *i.e., the balance between relevance and diversity*. For *relevance*, it refers that the augmented data should confirm to the original data characteristics, avoiding the semantic drift problem [22]. For *diversity*, it refers that sufficient variations should be made in order to enhance the model robustness on unseen data. In practice, the two factors are often conflict, and it is difficult to make a trade-off between them. For example, heuristic methods can generate diverse augmentations with more edits, while it will hurt the relevance since the modified data is likely to deviate from original data; model based methods can control the relevance via the learnable generator, while it usually produces conservative augmentations since they are trained with original training data. In addition, existing methods mainly perform focus on item-level augmentations, either explicit items or latent embeddings. They can't fully model the entire sequential semantics when conducting the data augmentation.

To address above issues, we propose a novel Retrieval-enhanced Data Augmentation (**ReDA**) framework for producing effective representations for modeling sequential user behaviors. Given a target user, the core idea of our approach is to leverage the related information from similar users for generating both relevant and diverse augmentation. Compared with previous methods solely relying on the target user herself/himself [40], such a retrieval-augmented approach can better balance the relevance and diversity, since the retrieved users are similar to the target user but with meaningful variations. In addition, the produced augmentations are more natural than synthesized data [20, 36], because they are from real user interaction behaviors. Specifically, we first train a neural retriever to retrieve the top-k augmentation users according to the semantic similarity between user representations, and then conduct two types of data augmentation (*i.e., attentional* and *interpolative* representation fusion) to generate augmented user representations. Subsequently, these augmented data are incorporated in a contrastive learning framework, for learning more capable representations suited to the recommendation task.

To the best of our knowledge, it is the first time that a retrieval-augmented approach has been developed for sequential user modeling. To validate the effectiveness of our approach, we conduct extensive experiments on both public and industrial datasets. Experimental results on offline evaluation and online A/B test show that our approach is more effective than a number of competitive methods, especially the data sparsity scenario.

2 RELATED WORK

In this section, we summarize the related work in two aspects.

Sequential Recommendation. Sequential recommendation predicts future items in user sequences by capturing user preferences in real-world applications. As a major research direction, many methods [6, 7, 12, 30] in recommender systems have been proposed by constructing user models on user behavior data. Pioneering works [11, 27] adopt Markov chains to model the pair-wise item transition correlations. Later, recurrent neural networks (RNN) have been adapted to solve sequential recommendation, modeling sequence-level correlations among transitions. Later, hierarchical RNNs [25] enhance RNNs using personalization information. Wu *et al.* [38] apply LSTMs to explore both long-term and short-term item transition correlations. The major drawback of Markov chain and RNN models is that their receptive fields of the transition function are limited. Except for recurrent neural networks, other deep learning models are also adopted for sequential recommendation tasks and achieve excellent performance. Huang *et al.* [14] leverage the memory-augmented neural network to store and update useful information explicitly. Recently, owing to the success of self-attention models [5, 35] in NLP tasks, a series of Transformer-based sequential recommendation models have been proposed [16, 32]. SASRec [16] applies Transformer layer to learn item importance in sequences, which characterizes complex item transition correlations. Recently, inspired by BERT [5] model, BERT4Rec [32] is proposed with a bidirectional Transformer layer.

Data Augmentation. Data augmentation has been widely used in various research areas [4, 37, 43]. For the computer vision, the early method such as DIM [34], the augmented encodings of different scales of the same image are fed into the contrastive learning as positive pairs. In the follow-up methods, *e.g.,* MoCo [10] and SimCLR [4], the different augmentations of the same image are considered as positive pairs for the contrastive learning. Data augmentation has also been used in recent recommendation methods. For the collaborative filtering methods, SGL [39] advances graph-based recommender systems with self-supervised learning by employing graph structure augmentations. SSL [41] proposes a siamese network to encode the items as pre-training with embedding-level augmentations. For the contrastive learning in sequential recommendation, CL4SRec [40] proposes three augmentations for the interaction sequence and applies a similar contrastive strategy as MoCo [10] and SimCLR [4] to set these augmentations of the same sequence as the positive pair in training. A more recent work DuoRec [24] propose a model-level augmentation method, which applies two different sets of dropout masks to the sequence representation learning.

Different from these studies, our work proposes the retrieval-enhanced data augmentation for modeling sequential user behaviors. Our approach is to leverage the related information from similar users for generating both relevant and diverse augmentation, for alleviating the data sparsity issue.

3 PRELIMINARIES

In this section, we first formulate the studied task and then introduce the base model for sequential user modeling.

3.1 Task Formulation

Notations and Tasks. We consider the sequential interaction scenario between users and items in recommender systems. Generally, a user u from the user set $\mathcal{U}$ is associated with a chronologically-ordered interaction sequence s_u with items: $i_1 \rightarrow \cdots \rightarrow i_n$, where n is the number of interactions and i_n from the item set $\mathcal{I}$ is the item at the n-th interaction. Based on the above notations, given the

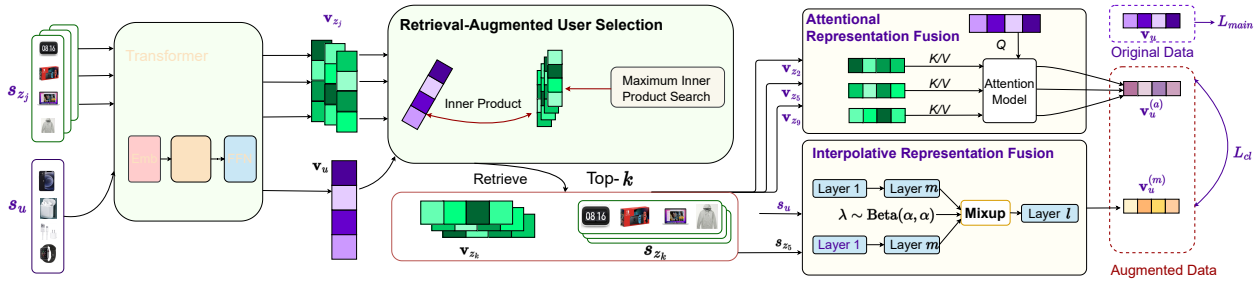


Figure 1: The overall architecture of our proposed approach.

user interaction sequence s_u, our task is to learn a d-dimensional representation for user u, denoted by $\boldsymbol{v}_u \in \mathbb{R}^d$. The learned user representation should effectively capture the sequential behavior characteristics, and can accurately predict the next interaction item at the $(n+1)$-th step.

A Retrieval-Augmented Approach. To further enhance the user representation, our core approach is to retrieval relevant *augmentation users* from user set $\mathcal{U}$ and then generate the augmentation representations based on these retrieved users. Specifically, for each user u, we adopt the retrieval module (see Section 4.1) to select augmentation users, denoted by $\mathcal{Z}_u = \{z_1, \cdots, z_k\}$, where z_k denotes the k-th augmentation user from the retrieval module. In order to discriminate between the target and augmentation users, we use the notation of z to represent an augmentation user. As a common user, each augmentation user z_j also has an interaction sequence, denoted by s_{z_j}. To generate the augmentation data conforming to user characteristics, we propose two data augmentation methods (see Section 4.2), which perform the augmentation based on either *attentional* or *interpolative* representation fusion. The derived users representations from the two augmentation methods are denoted by $\boldsymbol{v}_u^{(a)} \in \mathbb{R}^d$ and $\boldsymbol{v}_u^{(m)} \in \mathbb{R}^d$. Our aim is to produce an enhanced user representation by incorporating the augmented users representations:

$$\boldsymbol{v}_u \leftarrow f(s_u; \boldsymbol{v}_u^{(a)}, \boldsymbol{v}_u^{(m)}), \tag{1}$$

where we generate the user representation based on the interaction sequence and the augmented users representations, and $f(\cdot)$ denotes some kind of fusion function.

Since our approach follows a *retrieve-then-augment* paradigm [9], it can produce both relevant and diverse augmentation data, which is different from existing data augmentation methods in recommender systems [8, 36].

3.2 Transformer for Sequential User Behavior

For modeling sequential user behaviors, we develop the base model by following a standard Transformer architecture [35].

In the embedding mapping stage, we maintains an item embedding matrix $\mathbf{M}_I \in \mathbb{R}^{|\mathcal{I}| \times d}$. The matrices project the high-dimensional one-hot representation of an item to low-dimensional dense representations. Given a n-length item sequence, we apply a look-up operation from $\mathbf{M}_I$ to form the input embedding matrix $\mathbf{E} \in \mathbb{R}^{n \times d}$.

Based on the above embedding layer, we develop the Transformer module and produce user representation $\boldsymbol{u}$. Generally, the Transformer module is composed by multi-head self-attention (MHA) block and point-wise feed-forward network. Specifically, the multi-head self-attention is defined as:

$$\text{MHA}(\mathbf{F}^l) = [head_1, head_2, ..., head_h]\mathbf{W}^O, \tag{2}$$

$$head_i = \text{Attention}(\mathbf{F}^l\mathbf{W}_i^Q, \mathbf{F}^l\mathbf{W}_i^K, \mathbf{F}^l\mathbf{W}_i^V), \tag{3}$$

where the $\mathbf{F}^l$ is the input for the l-th layer, and the projection matrix $\mathbf{W}_i^Q \in \mathbb{R}^{d \times d/h}$, $\mathbf{W}_i^K \in \mathbb{R}^{d \times d/h}$, $\mathbf{W}_V^Q \in \mathbb{R}^{d \times d/h}$ and $\mathbf{W}^O \in \mathbb{R}^{d \times d}$ are learnable parameters.

Furthermore, we endow the non-linearity of the self-attention block by applying a point-wise feed-forward network:

$$\mathbf{F}^l = [\text{FFN}(\mathbf{F}_1^l)^\top; \cdots; \text{FFN}(\mathbf{F}_n^l)^\top], \tag{4}$$

$$\text{FFN}(\boldsymbol{x}) = (\text{ReLU}(\boldsymbol{x}\mathbf{W}_1 + \mathbf{b}_1))\mathbf{W}_2 + \mathbf{b}_2, \tag{5}$$

where $\mathbf{W}_1$, $\mathbf{b}_1$, $\mathbf{W}_2$ and $\mathbf{b}_2$ are trainable parameters. In the final layer of the Transformer layer, we utilize the output of the self-attention block at the last position as the final user representation $\boldsymbol{v}_u$. We calculate the user's preference score for the item i in the step $(t+1)$ under the context from user history as:

$$P(i_{t+1} = i | i_{1:t}) = \mathbf{e}_i \cdot \boldsymbol{v}_u, \tag{6}$$

where $\mathbf{e}_i$ is the representation of item i from item embedding matrix.

4 APPROACH

Inspired by recent progress of data augmentation [34, 40] and retrieval-augmented language model pre-training [1], we propose a novel retrieval-enhanced data augmentation framework to produce effective representations for sequential recommendation (**ReDA**). As a major technical contribution, we devise a *retrieve-then-augment* approach for effectively generating both relevant and diverse augmentation data, based on user interaction sequences.

The overview of the proposed ReDA framework is presented in Figure 1. First, we train a neural retriever to retrieve the top-k augmentation users according to the semantic similarity between user representations, and then conduct two types of data augmentation (*i.e.,* representation fusion and representation mixup) to generate augmented data $\boldsymbol{v}_u^{(a)}$ and $\boldsymbol{v}_u^{(m)}$ for user u in the contrastive learning framework. In what follows, we describe our approach in detail.

4.1 Retrieval-Augmented User Selection

In the field of recommender system, existing data augmentation methods [4, 37] mainly focus on constructing augmention data from the interaction sequence itself for a user. It is difficult to balance the two factors of relevance and diversity. As our solution, we design a retrieval-augmented selection module to retrieve similar users for enhancing the user representation.

4.1.1 User-centric Dense Retriever. In order to retrieval similar users for augmentation, we design a user-centric dense retriever that retrieves similar user embeddings for the target user.

Specifically, we first encode the interaction sequence of each user based on the Transformer architecture in Section 3.2, and derive the user embeddings $\{\boldsymbol{v}_u\}_{u\in\mathcal{U}}$. For selecting relevant users, we need to compute the relevance score between the target user and the rest users based on their user embeddings. Following REALM [9], the relevance score $r(u, z_j)$ between u and z_j is defined as:

$$r(u, z_j) = \boldsymbol{v}_u \cdot \boldsymbol{v}_{z_j}, \tag{7}$$

where $\boldsymbol{v}_u$ and $\boldsymbol{v}_{z_j}$ denote the representations of the target user u and a candidate user z_j. Here, we compute the relevance score based on the inner product of the user embeddings. With this function, we can compute the relevance score of each candidate user, and then find the top-k relevant users as the *augmentation users*.

4.1.2 Acceleration of Retrieval Efficiency. The retrieval procedure involves the selection of the top-k relevant users over a collection of candidate users. A straightforward solution is to calculate the vector inner product between each candidate user and the target user separately, and then sort the users according to the similarity scores. Since the number of candidate users can be very large, it will be time-consuming to find the top-k relevant users.

Here, we adopt the Maximum Inner Product Search (MIPS) [29] algorithm for accelerating the embedding retrieval, which has a sub-linear time complexity over the number of candidate users (*i.e.,* $|\mathcal{U}|$). Such a way leads to an approximate retrieval result with a very high accuracy, which is significantly more efficient than the exact search. After retrieval, we can obtain a set of k relevant augmentation users for user u, denoted by $\mathcal{Z}_u = \{z_1, \cdots, z_k\}$.

4.2 User Representation Augmentation

After obtaining the augmentation users $\mathcal{Z}_u = \{z_1, \cdots, z_k\}$, we next study how to leverage them to enhance the original user representation. Different from previous methods [36, 44], we don't directly modify the original interaction sequence, but instead augment user embedding at the representation level. It is easier to control such an augmentation approach, in order to avoid the semantic deviation of user characteristics as in discrete edits [40]. In what follows, we propose two kinds of representation augmentation methods, namely *attentional* and *interpolative* representation fusion.

4.2.1 Attentional Representation Fusion. The first method conducts the representation fusion at the user level, by attending to the embeddings of the augmented users. For each user u, we first aggregate the augmentation user embeddings to form an embedding matrix $\mathbf{Z}_u = [\boldsymbol{v}_{z_1}; \ldots; \boldsymbol{v}_{z_k}]$ according to the retrieval module in Section 4.1, where k denotes the number of augmentation users, and each column vector $\boldsymbol{v}_{z_k}$ encodes the embedding of the augmentation user. Then, we learn the user-specific augmentation representation using the self-attentive mechanism [19]:

$$\boldsymbol{v}_u^{(a)} = \mathbf{Z}_u \cdot \boldsymbol{\alpha}, \tag{8}$$

where $\boldsymbol{v}_u^{(a)}$ denotes the augmentation representation, and $\boldsymbol{\alpha}$ is an attention vector reflecting the importance of each augmentation user based on the relevance to the target user, calculated as:

$$\boldsymbol{\alpha} = \text{softmax}(\frac{(\boldsymbol{v}_u\mathbf{W}_1)^\top(\mathbf{Z}_u\mathbf{W}_2)}{\sqrt{d}}), \tag{9}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable parameters, endowing the model with an improved capacity to capture the user relevance. The scale factor $\sqrt{d}$ is to avoid overly large values of the inner product. In this way, we can obtain the user-specific augmented representation $\boldsymbol{v}_u^{(a)}$ to improve the prediction performance of downstream tasks.

4.2.2 Interpolative Representation Fusion. Besides the user-level fusion, we further design a fine-grained representation fusion method inspired by the *mixup* [18] strategy, which interpolates multi-modal embeddings *at the input layer*. However, the original mixup method is not directly suited for our task, since the item embeddings are shared by all the users. We propose to interpolate the hidden states *at some intermediate layer* in the Transformer encoder.

The basic procedure is described as follows. We first obtain the interaction data of the target user u and the augmentation user z_j, denoted by s_u and s_{z_j}. Then, we feed the sequences s_u and s_{z_j} into the Transformer encoder (Section 3.2), and produce layer-wise hidden states for each sequence. The total L layers of the Transformer encoder are divided into two parts at the t-th layer. At the bottom part ($1 \sim t$ layers), the encoding remains the same as in the original architecture. We perform the interpolation on the hidden states of the two sequences *at the t-th layer*. Then we forward the hidden states containing the augmented information to the upper layers ($t+1 \sim L$ layers).

Let $g_l(.;\Theta)$ denote the l-th layer in the encoder, and the hidden states of the l-th layer are computed as $\boldsymbol{h}_l = g_l(\boldsymbol{h}_{l-1};\Theta)$. We perform the forwarding operation at first t layers:

$$\boldsymbol{h}_u^l = g_l(\boldsymbol{h}_u^{l-1};\Theta), l \in [1, t], \tag{10}$$

$$\boldsymbol{h}_{z_j}^l = g_l(\boldsymbol{h}_{z_j}^{l-1};\Theta), l \in [1, t]. \tag{11}$$

Specially, the mixup operation is performed at the t-th layer:

$$\widetilde{\boldsymbol{h}}^t = \lambda\boldsymbol{h}_u^t + (1-\lambda)\boldsymbol{h}_{z_j}^t, \tag{12}$$

In our experiments, we use mixup ratio λ to perform the interpolation for each batch. Then, we continue forwarding the hidden states with interpolated information to upper layers as:

$$\widetilde{\boldsymbol{h}}^l = g_l(\widetilde{\boldsymbol{h}}^{l-1};\Theta), l \in [t+1, L]. \tag{13}$$

Via the above interpolation approach, we can derive the augmented user representation denoted by $\boldsymbol{v}_u^{(m)}$.

The rationale of interpolative representation fusion lies in that we operate on more abstractive sequence representations (instead of item embeddings as in original mixup [8]), and the model can further adjust the model parameters according to the incorporated external information (with the upper layers after the interpolation). Indeed, prior work [5] shows that the decoding from an interpolation of two hidden vectors generates a new sentence with mixed

Algorithm 1 The overall training process of our approach.

Require: Randomly initialize the parameters in the network, pre-encode augmentation users, and training epoch number T.

1: **Input:** User interaction sequence $\{s_u\}$
2: **Output:** The enhanced user representation $\boldsymbol{v}_u$
3: **for** $i = 1$ to T **do**
4: Learn the user representation $\boldsymbol{v}_u$ from Transformer networks.
5: Compute the relevance score of each candidate user using Eq. 7.
6: Retrieve the top-k relevant users users as augmentation users $\mathcal{Z}_u$.
7: Obtain the user-centric augmented representation $\boldsymbol{v}_u^{(a)}$ based on attentional representation fusion using Eq. 8 and Eq. 9.
8: Obtain the augmented representation $\boldsymbol{v}_u^{(m)}$ based on interpolative representation fusion using Eq. 12 and Eq. 13.
9: Optimize the loss in Eq. 16, jointly considering sequence prediction task and the additional contrastive learning task.
10: **end for**
11: Fine-tune the model according to downstream tasks.

meaning of two original sentences. We adopt the similar idea but interpolate the hidden states at an intermediate layer for enriching the sequential semantics. Complementary to attentional representation fusion ($\boldsymbol{v}_u^{(a)}$), the interpolative representation fusion ($\boldsymbol{v}_u^{(m)}$) can perform more fine-grained semantic fusion for data augmentation.

4.3 Learning and Discussion

This section presents the learning and discussion for our approach.

4.3.1 Contrastive Learning for Sequential Recommendation. For the sequential recommendation task, we adopt the negative log likelihood with softmax as the main loss for each user u at each time step $t+1$ as:

$$\mathcal{L}_{\text{main}} = -\log \frac{\exp(\boldsymbol{v}_u \cdot \boldsymbol{e}_{i_{t+1}})}{\exp(\boldsymbol{v}_u \cdot \boldsymbol{e}_{i_{t+1}}) + \sum_{i^- \in \mathcal{I}^-} \exp(\boldsymbol{v}_u \cdot \boldsymbol{e}_{i^-})}, \tag{14}$$

where $\boldsymbol{v}_u$, $\boldsymbol{e}_{i_{t+1}}$ and $\boldsymbol{e}_{i^-}$ denote the representations of the user, and the positive and negative items, respectively.

Furthermore, in order to enhance the augmentation consistency, we propose a contrastive learning objective for minimizing the difference between different augmentation representations of the same user and maximizing the difference with the augmentation represenations from different users. Following [34], the loss function can be defined similar to the softmax cross-entropy loss:

$$\mathcal{L}_{cl} = -\log \frac{\exp(\boldsymbol{v}_u^{(a)} \cdot \boldsymbol{v}_u^{(m)})}{\exp(\boldsymbol{v}_u^{(a)} \cdot \boldsymbol{v}_u^{(m)}) + \sum_{u^- \in \mathcal{N}} \exp(\boldsymbol{v}_u^{(a)} \cdot \boldsymbol{v}_{u^-})}, \tag{15}$$

where $\boldsymbol{v}_u^{(m)}$ and $\boldsymbol{v}_u^{(a)}$ are the augmentations representations from the attentional (Section 4.2.1) and interpolative (Section 4.2.2) fusion methods, respectively, and $\mathcal{N}$ is a negative set of augmented sequences from other users.

Finally, to enhance the performance of sequential recommendation, we adopt a joint training strategy by combining the main sequence prediction task loss and additional contrastive learning task loss as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{main}} + \beta \cdot \mathcal{L}_{\text{cl}}, \tag{16}$$

where β denotes the ratio of the contrastive learning loss in the traing process.

Table 1: Comparison of different augmentation methods.

Method	Strategy	Edit Unit	Level	Side Information
CL4SRec [40]	Heuristic	Input	Item	None
CoSeRec [20]	Hybird	Input	Item	None
MMInfoRec [23]	Model	Embedding	Item	Attributes
CASR [36]	Hybird	Input	Item	None
CauseRec [44]	Model	Input	Item	None
CCL [2]	Hybird	Input	Item	Attributes
DuoRec [24]	Heuristic	Hidden	Sequence	None
REDA	Model	Hidden	Sequence	None

Algorithm 1 presents the training algorithm for our model. The entire procedure of our approach consists of two important stages, namely retrieval-augmented user selection (line 4-6) and user representation augmentation (line 7-8). Finally, we adopt a joint training strategy by combining the contrastive learning objective and the sequence prediction task to optimize its parameters (line 9).

4.3.2 Comparison of Different Data Augmentation Techniques. Data augmentation [37, 43] is a widely used technique in machine learning, which extends the original data and generates more supervision signals. In recent years, it has been also applied to recommender systems [8, 36], in order to alleviate the data sparsity of user-item interaction.

The mainstream data augmentation techniques for recommendation can be divided into three classes, including *heuristic-based* [24, 40], *model-based* [23, 44] and *hybird* [2, 20, 36] strategies. Table 1 presents a comparison of existing methods in five aspects. Heuristic methods directly modify the original data with the edit operations such as *crop*, *mask* or *reorder*. It is easy to implement, but difficult to be effectively controlled, which is likely to cause the semantic drift issue [28, 42]. As a comparison, model based methods construct the underlying data generators, in order to augment new instances, which can be optimized according to downstream tasks. However, they still generate discrete user interactions, and involve additional parameters (from the generator) to learn. Besides, there are also some studies that combine the two kinds of methods [4, 31]. A major challenge in data augmentation is the trade-off between the relevance (confirming to original semantics) and diversity (containing sufficient variations). Different from these methods, our approach is able to generate both *relevant* (involving a dense retriever) and *diverse* (involving attentional and interpolative fusion) augmentations. The most relevant to our work is DuoRec [24]. However, this method is heuristic and inflexible in user behavior modeling.

5 EXPERIMENTS

In this section, we first set up the experiments, and then present the results and analysis.

5.1 Experimental Setup

5.1.1 Datasets. We conduct experiments on five datasets collected from four real-world platforms with various domains and sparsity levels. The statistics of these datasets after preprocessing are summarized in Table 2.

Table 2: Statistics of the datasets after preprocessing.

Dataset	Beauty	Sports	Yelp	Movie-1M	Meituan
# Users	22,363	25,598	30,431	6,040	14,866
# Items	12,101	18,357	20,033	3,953	22,108
# Avg. Actions / User	8.9	8.3	10.4	165.5	55.2
# Avg. Actions / Item	16.4	16.1	15.8	253.0	37.1
# Actions	747,827	296,337	316,354	1,000,209	821,326
Sparsity	99.93%	99.95%	99.95%	95.81%	99.83%

(1) **Meituan**[1]: this industry dataset consists of six-month (from *May 2021* to *October 2021*) transaction records in Beijing on the Meituan platform.

(2) **Amazon Beauty and Sports**: these two datasets are obtained from Amazon review datasets in [21]. In this work, we select two subcategories: "Beauty" and "Sports and Outdoors".

(3) **Yelp**[2]: this is a dataset for business recommendation. As it is very large, we only use records after *January 1st, 2019*.

(4) **MovieLens-1M**[3]: this is a popular movie recommendation dataset is used here, denoted as ML-1M.

For all datasets, we group the interaction records by users and sort them by the interaction timestamps ascendingly. Following [27], we only keep the 5-core datasets, and filter out unpopular items and inactive users with fewer than five interaction records.

5.1.2 Evaluation Settings. We adopt the *leave-one-out strategy* to evaluate the performance of each method. Concretely, for each user interaction sequence, the last item is used as the test data, the item before the last one is used as the validation data, and the remaining data is used for training. To speed up the computation of metrics, many previous works use sampled metrics and only rank the relevant items with a smaller set of random items. However, this sample operation may lead to inconsistent results with non-sampled rankings. Therefore, we evaluate each method on the whole item set without sampling and rank all the items that the user has not interacted with by their similarity scores. We employ top-K Hit Ratio (HR@K), top-K Normalized Discounted Cumulative Gain (NDCG@K) to evaluate the performance, which are widely used in related works [45]. HR focuses on the presence of the positive item, while NDCG further takes the rank position information into account. In this work, we report results on HR@$\{5, 10, 20\}$ and NDCG@$\{5, 10, 20\}$. Following previous works [3, 46], we apply the *leave-one-out strategy* for evaluation.

5.1.3 Comparison Methods. We compare our proposed approach with the following ten baseline methods:

(1) **Pop** is a non-personalized approach which recommends the same items for each user. These items are the most popular items which have the largest number of interactions in the item set;

(2) **BPR-MF** [26] is one of the representative non-sequential baselines. It utilizes matrix factorization to model users and items with the pairwise Bayesian Personalized Ranking (BPR) loss;

(3) **GRU4Rec** [13] applies GRU to model user interation sequence for session-based recommendation. We represent the items using embedding vectors rather than one-hot vectors;

(4) **SASRec** [16] is a self-attention based sequential recommendation model, which uses the multi-head attention mechanism to recommend the next item;

(5) **S^3-Rec$_{MIP}$** [50] utilizes the self-supervised learning methods to derive the intrinsic data correlation. However, it mainly focuses on how to fuse the context data and sequence data. We only compare the mask item prediction (MIP) in S^3-Rec for fairness;

(6) **CL4SRec** [40] utilizes the contrastive pre-training framework to extract meaningful user patterns and proposes three data augmentation approaches to construct pre-training tasks;

(7) **CCL** [2] designs a context-aware data augmentation approach to produce the augmented sequences and proposes a curriculum learning strategy to conduct contrastive learning;

(8) **DuoRec** [24] proposes a contrastive regularization with both the Dropout-based model-level augmentation and the supervised positive sampling to construct contrastive samples.

We reproduce most of the baseline methods with the open source library RecBole [47, 48], and implement those that are not in RecBole and our approach in PyTorch. The dimension of the embedding is set to 64. We set the number of Transformer layers is set as 6. The batch size is set to 256. We use Adam [17] optimization with its default parameter setting. Early stopping is used with a patience of 5 epochs. The weight β is mostly set as 1.0 and tuned in $\{0.05, 0.1, 0.5, 1.0, 2.0\}$. The gradient clipping restricts the norm of gradients within $[0, 0.1]$. Our code is available at this link: https://github.com/RUCAIBox/ReDA.

5.2 Experimental Results

In this section, we compare the proposed method with several baseline methods in different tasks on both public datasets and industry datasets. In Table 3, for two non-sequential recommendation baselines, the performance order is consistent across all datasets, *i.e., BPR-MF* > *Pop*. The non-personalized method Pop exhibits the worst performance on all datasets since it ignores users' unique preferences underlying their historical interactions.

In general, non-sequential recommendation methods perform worse than sequential recommendation methods, since the sequential pattern is important to consider in our task. As for sequential recommendation baseline methods, these methods utilize sequential information of users' historical interactions, which contributes to performance improvement in recommender systems. SASRec utilize the unidirectional self-attention mechanism, and achieve better performance than GRU4Rec. It indicates that self-attentive architecture is particularly suitable for modeling sequential data. Meanwhile, S^3-Rec$_{MIP}$, which only utilizes item-level self-supervision signals, suffers from a performance degradation compared to SASRec, possibly due to the weak supervision signals without using contextual information. Within three data augmentation baseline methods, these three methods have achieved better results compared with previous methods. CL4SRec uses heuristic methods to augment data and DuoRec uses dropout mask methods to augment data. For sequential data, CCL achieves the best results on all the datasets in sequential recommendation, since CCL uses context-aware data

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

Table 3: Performance comparison of different methods on sequential recommendation. Bold numbers correspond to the best performance for some metric, while underlined number correspond to the second best. Improvements over baselines are statistically significant with $p < 0.01$.

Datasets	Metric	Pop	BPR-MF	GRU4Rec	SASRec	S^3-Rec$_{MIP}$	CL4SRec	DuoRec	CCL	ReDA	Improv.
Beauty	HR@5	0.0072	0.0120	0.0239	0.0347	0.0327	0.0396	0.0408	0.0415	**0.0451**	+8.67%
	HR@10	0.0114	0.0299	0.0399	0.0630	0.0566	0.0681	0.0696	0.0704	**0.0758**	+7.65%
	HR@20	0.0195	0.0524	0.0637	0.1007	0.0905	0.1056	0.1087	0.1090	**0.1142**	+4.87%
	NDCG@5	0.0040	0.0065	0.0150	0.0185	0.0193	0.0208	0.0214	0.0225	**0.0242**	+7.77%
	NDCG@10	0.0053	0.0122	0.0201	0.0276	0.0270	0.0299	0.0310	0.0323	**0.0349**	+8.33%
	NDCG@20	0.0073	0.0179	0.0261	0.0371	0.0355	0.0394	0.0404	0.0416	**0.0440**	+6.20%
Sports	HR@5	0.0055	0.0092	0.0155	0.0185	0.0171	0.0219	0.0235	0.0247	**0.0284**	+16.38%
	HR@10	0.0090	0.0188	0.0259	0.0328	0.0298	0.0387	0.0401	0.0422	**0.0497**	+14.98%
	HR@20	0.0149	0.0337	0.0423	0.0541	0.0506	0.0595	0.0613	0.0630	**0.0687**	+9.98%
	NDCG@5	0.0040	0.0053	0.0098	0.0101	0.0106	0.0116	0.0123	0.0128	**0.0134**	+5.45%
	NDCG@10	0.0051	0.0083	0.0131	0.0148	0.0146	0.0171	0.0188	0.0204	**0.0235**	+15.54%
	NDCG@20	0.0066	0.0121	0.0172	0.0201	0.0199	0.0228	0.0240	0.0256	**0.0290**	+13.43%
Yelp	HR@5	0.0056	0.0127	0.0150	0.0157	0.0186	0.0201	0.0215	0.0227	**0.0244**	+8.06%
	HR@10	0.0095	0.0273	0.0276	0.0300	0.0313	0.0349	0.0356	0.0367	**0.0422**	+11.50%
	HR@20	0.0160	0.0500	0.0484	0.0529	0.0519	0.0598	0.0606	0.0625	**0.0687**	+10.04%
	NDCG@5	0.0036	0.0074	0.0091	0.0085	0.0116	0.0124	0.0136	0.0144	**0.0150**	+6.70%
	NDCG@10	0.0049	0.0121	0.0131	0.0131	0.0157	0.0171	0.0182	0.0193	**0.0208**	+8.92%
	NDCG@20	0.0065	0.0178	0.0183	0.0188	0.0209	0.0233	0.0245	0.0257	**0.0285**	+11.48%
ML-1M	HR@5	0.0078	0.0164	0.0993	0.1108	0.1078	0.1147	0.1168	0.1189	**0.1230**	+3.52%
	HR@10	0.0162	0.0354	0.1806	0.1902	0.1952	0.1975	0.1991	0.2021	**0.2062**	+1.98%
	HR@20	0.0402	0.0712	0.2892	0.3124	0.3114	0.3174	0.3202	0.3267	**0.3298**	+1.60%
	NDCG@5	0.0052	0.0097	0.0574	0.0648	0.0616	0.0662	0.0686	0.0695	**0.0708**	+2.16%
	NDCG@10	0.0079	0.0158	0.0835	0.0904	0.0917	0.0928	0.0956	0.0973	**0.0991**	+1.20%
	NDCG@20	0.0139	0.0248	0.1108	0.1211	0.1204	0.1230	0.1256	0.1281	**0.1294**	+1.57%
Meituan	HR@5	0.0065	0.0113	0.0232	0.0339	0.0318	0.0388	0.0392	0.0404	**0.0460**	+13.88%
	HR@10	0.0107	0.0288	0.0385	0.0621	0.0558	0.0678	0.0687	0.0695	**0.0749**	+7.96%
	HR@20	0.0186	0.0516	0.0629	0.0857	0.0806	0.0873	0.0890	0.0901	**0.0943**	+4.71%
	NDCG@5	0.0036	0.0058	0.0143	0.0177	0.0182	0.0185	0.0191	0.0202	**0.0216**	+7.54%
	NDCG@10	0.0046	0.0115	0.0192	0.0268	0.0261	0.0288	0.0302	0.0315	**0.0338**	+8.29%
	NDCG@20	0.0068	0.0172	0.0256	0.0365	0.0349	0.0382	0.0396	0.0405	**0.0428**	+6.03%

augmentation methods and considers curriculum contrastive learning to enhance the representations.

Finally, by comparing our approach ReDA with all the baselines, it is clear to see that our method performs consistently better than them by a large margin on all datasets. Different from these baselines, we adopt a retrieval-enhanced data augmentation for modeling sequential user behaviors, and utilize two types of user representation augmentation strategies at the representation level. This result shows that our approach is effective to improve the performance for sequence modeling.

5.3 Further Analysis

Next, we continue to study whether our approach works well in more detailed analysis.

5.3.1 Ablation Study of ReDA. To effectively utilize the user behavior data, our approach has made several technical contributions. Here, we examine how each of them affects the final performance. We consider the following variants of our approach for comparison:

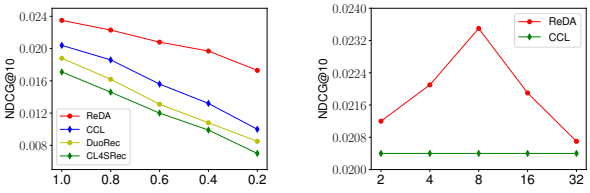
- ReDA$_{\neg \text{Att}}$: Removing the attentional representation fusion.
- ReDA$_{\neg \text{Mixup}}$: Removing the interpolative representation fusion.
- ReDA$_{\neg \text{Aug}}$: Removing the user representation augmentation.
- ReDA$_{\text{Random}}$: Retrieving users based on random selection.
- ReDA$_{\text{Jacc}}$: Retrieving users based on jaccard similarity between the item sets of interaction sequences from two users.
- ReDA$_{\text{Cos}}$: Retrieving users based on the cosine similarity between the user embeddings.

In Table 4, we report the results of these comparison methods in the sequential recommendation task on the Beauty dataset. Similar conclusions can be drawn on the other datasets or tasks. First, we can observe that removing any augmentation module would lead to the performance decrease. It indicates the data augmentation module is useful to improve the recommendation performance.

Table 4: Ablation analysis on the Beauty dataset in sequential recommendation.

Models	HR@5	NDCG@5	HR@10	NDCG@10
ReDA$_{\neg\text{Att}}$	0.0432	0.0228	0.0721	0.0328
ReDA$_{\neg\text{Mixup}}$	0.0429	0.0222	0.0713	0.0325
ReDA$_{\neg\text{Aug}}$	0.0398	0.0210	0.0695	0.0307
ReDA$_{\text{Random}}$	0.0332	0.0178	0.0618	0.0232
ReDA$_{\text{Jacc}}$	0.0407	0.0208	0.0698	0.0311
ReDA$_{\text{Cos}}$	0.0421	0.0219	0.0705	0.0319
ReDA	**0.0451**	**0.0242**	**0.0758**	**0.0349**

Among them, ReDA$_{\neg\text{Mixup}}$ performs worst, showing that the fine-grained semantic fusion is more import for our task. It is worth noting that the performance of ReDA$_{\neg\text{Aug}}$ decreases more than only removing one of data augmentation strategies, which shows that the attentional representation fusion and interpolative representation fusion are complementary to each other. Since our method is a retrieval augmentation framework, we compare the performance of different retrieval strategies on our framework. We find that both ReDA$_{\text{Jacc}}$ and ReDA$_{\text{Cos}}$ correspond to a worse performance, which indicates that our user-centric dense retriever can better capture user preferences and improve the recommendation performance.

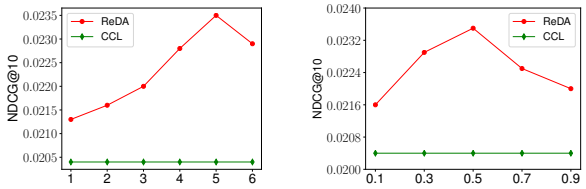


(a) The ratio of training data. (b) The number of retrieved users.

Figure 2: Performance of tuning with different sparsity levels and number of relevant users on the Sports dataset.

5.3.2 The Impact of Data Sparsity Levels. To further verify the proposed ReDA can alleviate the sparsity of interaction data, we simulate the data sparsity scenarios by using different proportions of the full dataset, *i.e.,* 20%, 40%, 60%, 80% and 100%. Figure 2(a) shows the evaluation results of the sequential recommendation task on the Sports dataset. The performance substantially drops when less training data is used. While, ReDA is consistently better than baselines in all cases, especially in an extreme sparsity level (*i.e.,* 20%). This observation implies that ReDA is able to make better use of the augmented data, which alleviates the influence of data sparsity problem for user behavior modeling to some extent.

5.3.3 The Impact of Top-k Relevant Users. To study the effect of retrieval-augmented user selection module, we vary the number of relevant users (*i.e., k* in Section 4.1.1) from 2 to 32 and report the tuning results in Figure 2(b). For simplicity, we only incorporate the best baseline CCL from Table 3 as a comparison. It is worth noting that our model achieves the overall best performance while with eight relevant users, and meanwhile a too large or too small number for k will lead to performance degradation. These findings indicate that retrieving a small number of high-quality augmented users is important to improve the recommendation performance. However, with the number of relevant users increasing, it is likely to incorporate noisy information that will impair the performance.



(a) The index t of interpolative layer. (b) The ratio λ of mixup.

Figure 3: Performance of interpolative representation fusion with different interpolative layers and mixup ratios on the Sports dataset.

5.3.4 The Impact of the Interpolative Layer Index t. To investigate the impact of different interpolative layer indices, we explore the mixup operation on different layer index t (Eq. 12) for interpolative representation fusion and the results are shown in Figure 3(a). Each layer of the Transformer captures different types of information (*e.g.,* short-term or long-term semantics). Our approach outperforms the baseline by at least +0.07 NDCG@10 score (even for $t = 1$) under all settings. It indicates that our approach is very robust with the settings of the mixup layer. Besides, when t is set to 5, our model achieves the best performance. It shows that a higher layer is more capable of learning fine-grained semantics of user by interpolative representation fusion. It demonstrates that a proper layer index t is promising for interpolative representation fusion and the proposed approach can better utilize the augmented data to enhance user representation.

5.3.5 The Impact of Mixup Ratio λ. In the interpolative representation fusion module, the ratio λ (Eq. 12) can balance the augmented user representation and original user representation for generating more effective sequence representations. To analyze the effect of λ on the recommendation performance, we vary it from 0.1 to 0.9 and report the tuning results in Figure 3(b). From this figure, we can observe λ has an important effect on the final performance of ReDA and a value of 0.5 that assigns equal weights to both representations leads to the best performance. It indicates both kinds of user representations are important to consider when learning the final user representation.

5.4 Online A/B Test

We conduct online A/B tests in real business scenario of the Meituan app to further examine the effectiveness of our approach. Since there are a large number of cold-start users in the online platform, it is especially vital that the deployed approach should be both robust and general. Specifically, we select four real business scenarios from

Table 5: Comparison of CTR metrics in online A/B test.

Models	Takeout	Group purchase	Grocery shopping	Delivery
Original	0.0456	0.0621	0.0632	0.0512
ReDA	**0.0465**	**0.0633**	**0.0652**	**0.0531**
Improvement	+2.1%	+1.9%	+3.3%	+3.8%

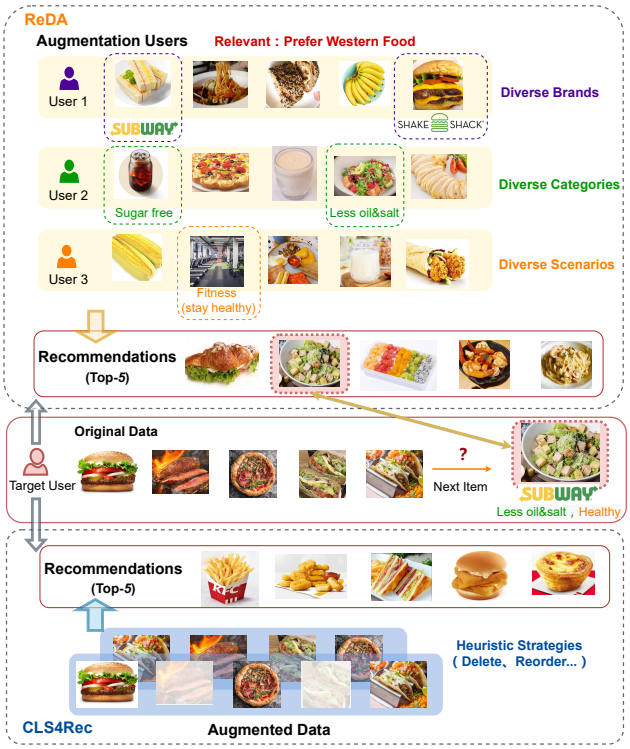


Figure 4: An illustrative example for showing the effect of our augmentation ReDA approach on the Meituan dataset. The target user generally prefers western food. We show both the recommendation list and the intermediate augmentation data for ReDA (our approach) and CLS4Rec (the baseline).

Takeout, Group purchase, Grocery shopping and Delivery, which have about 800,000 users in our sampled population.

Specifically, we split them into control group (A) and treatment group (B) with the same size randomly. For comparison, we apply our model to generate augmented representation and model user interest. In order to effectively measure the effect of our proposed method in real business scenarios, we adopt CTR (*i.e., click-through rate*) [49] as the evaluation metric. In internet marketing, CTR is a key metric that measures the number of clicks that the platform receives on their recommendations per number of impressions, and it can used to measure the online user activity for some online service. In our setting, a group with a larger CTR value means that the corresponding method can better capture user interest in the cold start scenario, attracting more attention from online users.

Here, we consider a seven-day period of the Meituan app for online A/B tests (with a small random traffic). Table 5 presents the CTR comparison between the original method and our method. Our model is better than the compared baseline, which further verifies the effectiveness of the proposed model. Our online A/B test further demonstrates the effectiveness of the proposed method.

5.5 Case Study

A key contribution of the proposed ReDA is to generate both relevant and diverse augmentations by retrieving highly similar user representations. To understand the difference between ReDA and previous augmentations approaches, we present an illustrative example to compare the recommendation results between ReDA and CLS4Rec in Figure 4, where a sample user and her/his ordered food list (*hamburger*, *pizza* and *sandwich*) are presented.

For the baseline CLS4Rec, we find that it focuses on recommending more fast food (*fries*, *hamburger*, and *sandwich*). For our ReDA approach, we present both the retrieved top three augmented users and the recommendation list (*hamburger*, *salad*, *fruit*, and *noodles*). Overall, it can be observed that the retrieved users have relevant yet diverse food tastes, covering more brands, categories and scenarios. Since previous augmentation-baed recommendation methods mainly generate data variations by modifying the original data, it is more likely to cause so called "*information cocoons*" [15]. As a comparison, ReDA directly learns the augmentations from real users' data, which can generate more relevant and diverse recommendations.

6 CONCLUSION

In this work, we presented a novel data augmentation framework, named **R**etrieval-**e**nhanced **D**ata **A**ugmentation (**ReDA**), to generate both relevant and diverse augmentation for sequential recommendation. Our contributions can be summarized in two points. First, to better balance the relevance and diversity, we adopted a neural retriever to retrieve augmentation users according to the semantic similarity between user representations. Second, we conducted two types of representation augmentation at the representation level to generate more natural augmented data for user. Extensive experiments on both public and industry datasets demonstrated the effectiveness of our proposed approach.

As future work, we will consider apply the current framework to other task scenarios for user modeling. Additionally, we will explore fusing side information such as multi-modal information into the augmented user representations for further enhancing the representation capacity.

ACKNOWLEDGEMENT

This work was partially supported by the Beijing Natural Science Foundation under Grant No. 4222027, and National Natural Science Foundation of China under Grant No. 61872369 and 61832017, Beijing Academy of Artificial Intelligence (BAAI), Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098, the Outstanding Innovative Talents Cultivation Funded Programs 2020 of Renmin Univertity of China and Meituan. Xin Zhao is the corresponding author.

REFERENCES

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML 2009*.

[2] Shuqing Bian, Wayne Xin Zhao, Kun Zhou, Jing Cai, Yancheng He, Cunxiang Yin, and Ji-Rong Wen. 2021. Contrastive Curriculum Learning for Sequential User Behavior Modeling via Data Augmentation. In *CIKM 2021*. 3737–3746.

[3] Da Cao, Xiangnan He, Liqiang Nie, Xiaochi Wei, Xia Hu, Shunxiang Wu, and Tat-Seng Chua. 2017. Cross-Platform App Recommendation by Jointly Modeling Ratings and Texts. *ACM Trans. Inf. Syst.* 35, 4 (2017).

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. [n. d.]. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML 2020*.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT 2019*. 4171–4186.

[6] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations. In *RecSys*. 152–160.

[7] Zhengxiao Du, Xiaowei Wang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Sequential Scenario-Specific Meta Learner for Online Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. ACM, 2895–2904.

[8] Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting Data with Mixup for Sentence Classification: An Empirical Study. *CoRR* (2019).

[9] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. *CoRR* abs/2002.08909 (2020).

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR 2020*.

[11] Ruining He and Julian J. McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *ICDM 2016*.

[12] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In *SIGIR 2017*. 355–364.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based Recommendations with Recurrent Neural Networks. *CoRR* abs/1511.06939 (2015).

[14] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *SIGIR 2018*. 505–514.

[15] Yijun Huang, Lan Zhou, Ziqian Zeng, Lingli Duan, and Jiayu Wang. 2020. An empirical study on the phenomenon of information narrowing in the context of personalized recommendation. In *Journal of Physics: Conference Series*, Vol. 1631. 012109.

[16] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM 2018*. 197–206.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015*.

[18] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-Action-Reflection: Towards Deep Interaction Between Conversational and Recommender Systems. In *WSDM 2020*.

[19] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-Attentive Sentence Embedding. In *ICLR 2017*.

[20] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S. Yu, Julian J. McAuley, and Caiming Xiong. 2021. Contrastive Self-supervised Sequential Recommendation with Robust Augmentation. *CoRR* abs/2108.06479 (2021).

[21] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR 2015*. 43–52.

[22] Zhengjie Miao, Yuliang Li, and Xiaolan Wang. 2021. Rotom: A Meta-Learned Data Augmentation Framework for Entity Matching, Data Cleaning, Text Classification, and Beyond. In *SIGMOD 2021*. 1303–1316.

[23] Ruihong Qiu, Zi Huang, and Hongzhi Yin. 2021. Memory Augmented Multi-Instance Contrastive Predictive Coding for Sequential Recommendation. In *ICDM 2021*. 519–528.

[24] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. In *WSDM 2022*.

[25] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *RecSys 2017*. 130–137.

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009*. 452–461.

[27] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *WWW 2010*. ACM, 811–820.

[28] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. 2018. Adversarially Robust Generalization Requires More Data. In *NeurIPS 2018*. 5019–5031.

[29] Anshumali Shrivastava and Ping Li. 2014. Asymmetric LSH (ALSH) for Sublinear Time Maximum Inner Product Search (MIPS). In *NeurIPS 2014*. 2321–2329.

[30] Ajit Paul Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *SIGKDD 2008*. 650–658.

[31] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1 (2014), 1929–1958.

[32] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *CIKM2019*. 1441–1450.

[33] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM 2018*. 565–573.

[34] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748 (2018). arXiv:1807.03748

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NeurIPS*. 5998–6008.

[36] Zhenlei Wang, Jingsen Zhang, Hongteng Xu, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Counterfactual Data-Augmented Sequential Recommendation. In *SIGIR 2021*. 347–356.

[37] Jason W. Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *EMNLP-IJCNLP 2019*.

[38] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent Recommender Networks. In *WSDM 2017*. ACM, 495–503.

[39] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In *SIGIR 2021*. 726–735.

[40] Xu Xie, Fei Sun, Bolin Ding, and Bin Cui. 2020. Contrastive Pre-training for Sequential Recommendation. *CoRR* abs/2010.14395 (2020).

[41] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix X. Yu, Ting Chen, Aditya Krishna Menon, Lichan Hong, Ed H. Chi, Steve Tjoa, Jieqi (Jay) Kang, and Evan Ettinger. 2021. Self-supervised Learning for Large-scale Item Recommendations. In *CIKM 2021*. 4321–4330.

[42] Dong Yin, Kannan Ramchandran, and Peter L. Bartlett. 2019. Rademacher Complexity for Adversarially Robust Generalization. In *ICML 2019*. 7085–7094.

[43] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. In *NeurIPS 2020*.

[44] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. CauseRec: Counterfactual User Sequence Synthesis for Sequential Recommendation. In *SIGIR 2021*. 367–377.

[45] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *IJCAI 2019*. 4320–4326.

[46] Wayne Xin Zhao, Junhua Chen, Pengfei Wang, Qi Gu, and Ji-Rong Wen. 2020. Revisiting Alternative Experimental Settings for Evaluating Top-N Item Recommendation Algorithms. In *CIKM 2020*. ACM, 2329–2332.

[47] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, Yushuo Chen, Lanling Xu, Gaowei Zhang, Zhen Tian, Changxin Tian, Shanlei Mu, Xinyan Fan, Xu Chen, and Ji-Rong Wen. 2022. RecBole 2.0: Towards a More Up-to-Date Recommendation Library. In *CIKM 2022*.

[48] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM 2021*. 4653–4664.

[49] Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *KDD 2018*. ACM, 1059–1068.

[50] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM 2020*.