Empowering General-purpose User Representation with Full-life Cycle Behavior Modeling

Bei Yang*
bella.yb@alibaba-inc.com
Alibaba Group
Hangzhou, China

Jie Gu*
yemu.gj@alibaba-inc.com
Alibaba Group
Hangzhou, China

Ke Liu
ke.l@wustl.edu
Zhejing University
Hangzhou, China

Xiaoxiao Xu
xiaoxiao.xuxx@alibaba-inc.com
Alibaba Group
Hangzhou, China

Renjun Xu
rux@zju.edu.cn
Zhejing University
Hangzhou, China

Qinghui Sun
yuyang.sqh@alibaba-inc.com
Alibaba Group
Hangzhou, China

Hong Liu
229795716@qq.com
Alibaba Group
Hangzhou, China

ABSTRACT

User Modeling plays an essential role in industry. In this field, task-agnostic approaches, which generate general-purpose representation applicable to diverse downstream user cognition tasks, is a promising direction being more valuable and economical than task-specific representation learning. With the rapid development of Internet service platforms, user behaviors have been accumulated continuously. However, existing general-purpose user representation researches have little ability for full-life cycle modeling on extremely long behavior sequences since user registration. In this study, we propose a novel framework called full- Life cycle User Representation Model (LURM) to tackle this challenge. Specifically, LURM consists of two cascaded sub-models: (i) Bag-of-Interests (BoI) encodes user behaviors in any time period into a sparse vector with super-high dimension (*e.g.*, 10^5); (ii) Self-supervised Multi-anchor Encoder Network (SMEN) maps sequences of BoI features to multiple low-dimensional user representations. Specially, SMEN achieves almost lossless dimensionality reduction, benefiting from a novel multi-anchor module which can learn different aspects of user interests. Experiments on several benchmark datasets show that our approach outperforms state-of-the-art general-purpose representation methods.

CCS CONCEPTS

• **Computing methodologies** → *Information extraction.*

KEYWORDS

general-purpose user embedding, extremely long sequence modeling, self-supervised learning, representation learning

ACM Reference Format:
Bei Yang, Jie Gu, Ke Liu, Xiaoxiao Xu, Renjun Xu, Qinghui Sun, and Hong Liu. 2023. Empowering General-purpose User Representation with Full-life Cycle Behavior Modeling. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599331

1 INTRODUCTION

Customer first is a well-known value insisted by many businesses. Accordingly, understanding users is of great importance since it helps provide satisfactory personalized services. Researchers have reached a common view that user historical behaviors contain rich valuable information[8, 9, 13, 32]. It is intuitive to mine and capture user interests or properties from massive behavior data. A typical solution is to encode behavior sequences into low-dimensional yet informative representations. Extensive works have proved the success of such a solution, benefiting a wide range of real-world applications like user profiling, recommendation systems, search engines and online advertising.

In the literature, there are plenty of works focusing on task-specific user representation learning. That is, the representation is simultaneously learned with the specific downstream classifier, *e.g.*, CTR predictor for recommendation [14, 23, 33]. Such a task-specific paradigm would benefit the performance on the target application, but the learned representations can hardly be applied to other tasks. To avoid generalization issues and ensure performance, training a particular model for each downstream application is essential. However, it is time-consuming, expensive (requiring massive labeled data, computing and storage resources), and cannot be used in many real-world scenarios where business requirements are diverse, variable and numerous.

In contrast, general-purpose (*a.k.a.*, universal) user representation has gained growing attention recently[1, 3, 7, 11, 19, 24–28].

*Both authors contributed equally to this research.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599331

Typically, universal user representation is learned without any task-specific bias and can serve a variety of downstream tasks by treating the pre-trained model as a feature extractor. It can be extracted from user historical behavior data or other side information to express his/her interests and preferences. Note that no more rectifications (e.g., fine-tuning) are required for the representation model in downstream applications. We only need to train a simple model, like SVM or MLP, for a specific task. Unfortunately, current approaches can only process user behavior sequences with a length of tens or hundreds. According to our observations, the performance of the universal representation model is limited by the available behaviors.

An insight we want to show is that the more behaviors, the richer the information can be captured, and the better the performance on downstream (Fig. 3). Several previous works have made some attempts on this topic [5, 15, 29]. They model relatively long sequential data based on hierarchical architectures and memory networks [21, 23, 31], and verify the effectiveness of incorporating more behaviors. However, this topic is far from being solved. The existing methods suffer from the generalization issue (since they are all task-specific), as well as the lack of the capability of handling extremely long behavior sequences. With the rapid development of Internet service platforms, abundant user behavior sequences which reflect intrinsic and multi-facet user interests have been accumulated. We do believe that valuable and rich information can be mined from such massive data. The core is to efficiently encode full-life cycle behaviors of users (which may even include hundreds of thousands of behaviors) into informative general-purpose representations.

In this work, a novel framework for universal user modeling called full-Life cycle User Representation Model (LURM) is proposed. The framework has the ability to model user behaviors of arbitrary length since his/her registration, e.g., including every behavior from his/her registration on some APP to the present day. To meet the need of extremely long sequence modeling, we first introduce a model named Bag-of-Interests (BoI) to summarize items in behavior sequences similar to Bag of Visual Words. In this way, we can use a sparse vector with super-high dimension to represent user behaviors in any time period. Then, a Self-supervised Multi-anchor Encoder Network (SMEN) that maps sequences of BoI features to multiple low-dimensional user representations is proposed. SMEN consists of three modules: a multi-anchor module which can learn different aspects of user preferences, a time aggregation module which can model evolution of user behaviors, and a multi-scale aggregation module which can learn and aggregate BoI features in different scales. Considering the consistency between user behaviors in different time periods, we introduce a contrastive loss function to the self-supervised training of SMEN. With the designs above, SMEN achieves almost lossless dimensionality reduction. It is noteworthy that the proposed method allows for encoding behaviors of arbitrary length and aggregating information in different time scales. Thus, though the inspiration of this work is to capture long-term user interests from full-life cycle behaviors, LURM can also be applied to short-term interests related tasks. Extensive experiments on several benchmark datasets show the superiority of our approach against other baselines on both short-term interest-related tasks and long-term interest-related tasks.

The main contribution of our work can be summarized as follows:

- A novel framework named LURM is proposed for learning high-quality universal user representations via self-supervision. The framework is built based on Bag-of-Interests, which is capable of encoding arbitrary numbers of user behaviors. It shows great advantages in adaptively modeling long-term or relatively short-term user interests in a data-driven manner.
- More importantly, such a Bag-of-Interests formulation allows us to encode extremely long behavior sequences (even including millions of behaviors). This makes full-life cycle user modeling a reality, while without forgetting and efficiency issues as in previous RNN based or Transformer based methods.
- A sub-module named SMEN is proposed, which further incorporates interest-anchored dimension reduction and time variation modeling. Such operations ensure a compact, easy-using and informative universal user representation.
- Extensive experiments are performed on several real-world datasets. The results demonstrate the effectiveness and generalization ability of the learned user representation. Furthermore, the experiments also demonstrate that better performance can indeed be achieved by full-life cycle behavior modeling.

2 METHODOLOGY

In this work, we are committed to learning general-purpose user representation by modeling full-life cycle user behavior sequences with arbitrary length. For this purpose, we propose a framework named full-Life cycle User Representation Model (LURM), which consists of two cascaded sub-models, *i.e.*, Bag-of-Interests (BoI) and Self-supervised Multi-anchor Encoder Network (SMEN). The overall architecture of LURM is shown in Fig. 1.

2.1 Bag-of-Interests

Most of the time, there are certain patterns behind user behaviors, which are supposed to be focused on to mine user interests and preferences. To this end, we should encode behaviors (like purchase and click) by aggregating the contents of items. Bag of Words (BoW) [12] is a common method, which aggregates information of items at the word granularity, *(*e.g.), with item titles. However, it cannot deal with the data of other modalities such as images, nor can it model the dependencies between words. Inspired by Bag of Visual Words (BoVW) [10], encoding behaviors at the item granularity seems more natural and reasonable. Unfortunately, it is also inappropriate in the field of e-commerce, since usually there are billions of items and accordingly the item vocabulary would be extremely large, making it infeasible in practice.

We propose a model called Bag-of-Interests (BoI) to aggregate users' behavior data at the 'interest' granularity. Each 'interest' is a cluster of similar items and represents a certain kind of preference. The size of the 'interest' vocabulary is often selected at a level of about 10^5 for retaining enough details. As shown in Fig. 1 (a), BoI consists of an item embedding module and a large-scale clustering module. For convenience, we only focus on the text modality in this work. It should be noted that our method can be easily extended to multi-modal data.

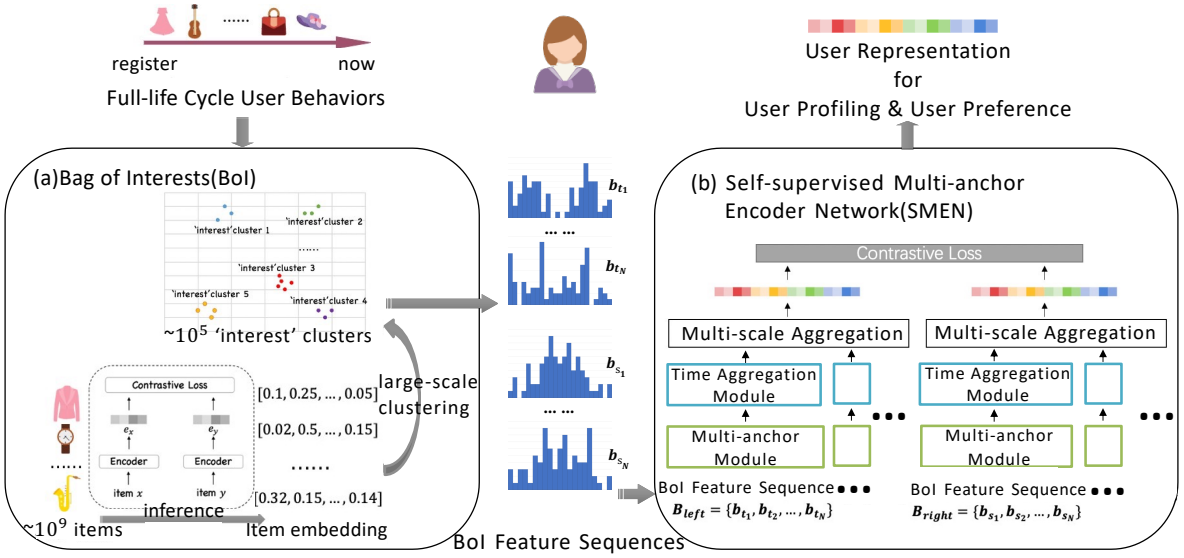


Figure 1: Illustration of our full-Life cycle User Representation Model (LURM) for user understanding. LURM consists of two sub-model: (a) Bag-of-Interests (BoI) is used to aggregate user behavior data at 'interest' granularity, and is composed of an item embedding module and a large scale clustering module. We apply the BoI model on full-life cycle behavior data to generate multi-scale BoI feature sequences, (b) Self-supervised Multi-anchor Encoder Network (SMEN) is used to learn compressed user representations from BoI features sequences, and is composed of a multi-anchor module, a time aggregation module, a multi-scale aggregation module and a contrastive learning module.

2.1.1 Item Embedding Module. An 'interest' vocabulary is supposed to be built in our BoI model, similar to BoVW. The embedding of each item is required, so that similar items with close distance in the embedding space can be clustered together to form an 'interest'. Recently, discriminative approaches based on contrastive learning in the latent space have shown great success in the field of representation learning, achieving state-of-the-art results in natural language and image processing[4, 6, 17, 20]. Inspired by these works, we design a contrastive learning task based on the relation between items drawn from a user to learn item embedding[2].

Given a set of users $U = \{u_1, u_2, ..., u_{|U|}\}$, each user $u \in U$ corresponds to a behavior sequence $S = \{x_1, x_2, ..., x_{|S|}\}$, where $x_i \in S$ denotes the i-th item. $|U|$ and $|S|$ denote the number of users and the length of u's behaviors respectively. Generally, the content of an item x can be expressed as $\{w_1, w_2, ..., w_{|x|}\}$, where w_i denotes a word from a vocabulary V, and $|x|$ denotes the number of words in the content of x. Firstly, an encoder with average operation is used to generate item embedding e:

$$e_x = encoder(w_1, w_2, ..., w_{|x|}) = proj(\frac{1}{|x|}\sum_{i=1}^{|x|} W_i), \quad (1)$$

where $W_i \in \mathbb{R}^d$ is the embedding of word w_i and will be learned during training, $proj(\cdot)$ includes two residual blocks, and a L_2 normalization layer. To construct the contrastive learning task, we sample positive pairs from behavior sequences of users randomly. Specifically, two items (x_i, y_i) are similar, i.e. a positive pair, if they are drawn from the same user behavior sequence and the time interval between the occurrence of these two items is less than β, where β is the window size controlling the interval of the two user behaviors. Without loss of generality, the sampled mini-batch with batch size n can be denoted as $\Delta = \{x^1, y^1, x^2, y^2, ..., x^n, y^n\}$, where (x^i, y^i) construct a positive pair drawn from the behavior sequence S_t^i of the i-th user in batch. Then, the contrastive prediction task is defined to identify y^i in $\Delta \setminus \{x^i\}$ for a given x^i, and all other items in $\Delta \setminus \{x^i, y^i\}$ are negatives. The loss for the positive pair (x^i, y^i) is written as

$$l(x^i, y^i) = -\log \frac{e^{g(x^i, y^i)/\tau}}{\sum_{v \in \Delta, v \neq x^i} e^{g(x^i, v)/\tau}}, \quad (2)$$

where $g(x, y) = \frac{e_x{}^T e_y}{\|e_x\| \|e_y\|} = e_x{}^T e_y$ denotes the cosine similarity between the embedding e_x and the embedding e_y , and τ is the temperature parameter. The final objective is the average loss of all positive pairs in the mini-batch, which can be written as

$$Loss = \frac{1}{2n}\sum_i (l(x^i, y^i) + l(y^i, x^i)). \quad (3)$$

2.1.2 Large-scale Clustering Module. An item embedding set $E = \{e_i\}_{i \in I}$ can be obtained, where I is the complete collection of items at the billion level. In order to retain details as many as possible, the size of the 'interest' vocabulary is set to be at $10^4 \sim 10^5$ level. In other words, all items should be clustered into D (e.g., 10^5) categories. Considering the large-scale item set, a subset $E' \subset E$ at

the million level is sampled, and an efficient clustering algorithm named HDSC [30] on $\boldsymbol{E'}$ is employed to cluster similar items into the same 'interest'.

After clustering, the cluster centers $\boldsymbol{C}$ make up an 'interest' vocabulary. Therefore, each item can be attributed to one/multiple 'interest(s)' by hard/soft cluster assignment. Take hard cluster assignment as an example, each user can obtain his/her sparsely high-dimensional BoI feature $\boldsymbol{b_t} \in \mathbb{R}^D$ in time period $\boldsymbol{t}$ according to his/her behavior sequence $\boldsymbol{S_t} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_{|S_t|}}\}$:

$$\begin{aligned}\boldsymbol{b_t} = [&log(1 + \Sigma_{i=1}^{|S_t|}\mathbb{I}_{\boldsymbol{x_i} \in c_1}), log(1 + \Sigma_{i=1}^{|S_t|}\mathbb{I}_{\boldsymbol{x_i} \in c_2}), \\ &..., log(1 + \Sigma_{i=1}^{|S_t|}\mathbb{I}_{\boldsymbol{x_i} \in c_D})]\end{aligned} \tag{4}$$

where $\boldsymbol{x_i} \in c_j$ means item i is assigned to the cluster $c_j \in \boldsymbol{C}$, $\mathbb{I}$ is an indicator function.

2.2 Self-supervised Multi-anchor Encoder Network

A BoI feature $\boldsymbol{b_t}$ for each user can be obtained, through the BoI model given behavior data in any time period $\boldsymbol{t}$. Most directly, a user representation with super-high dimension $\boldsymbol{b_T}$ can be obtained by applying the BoI model to the whole life-cycle time $\boldsymbol{T}$. However, there are two main disadvantages, *i.e.*, 1) it is not friendly to the downstream tasks since the dimension of representation is too high, and 2) it is too crude to aggregate the information in the whole life-cycle time without considering variations over time.

Therefore, we propose to get a BoI feature sequence by applying the BoI model at each time period of the whole life-cycle time. Then, a Self-supervised Multi-anchor Encoder Network (SMEN) is designed to learn compressed user representations from the sequence of BoI features. According to its design, SMEN can simultaneously generate multiple low-dimensional representations (e.g., each user representation has dimension of 10^2) with a certain degree of disentanglement, representing different aspects of user preferences.

The full-life cycle time $\boldsymbol{T}$ is divided into N parts at a fixed time interval, i.e. $\boldsymbol{T} = \{\boldsymbol{t_1}, \boldsymbol{t_2}, ..., \boldsymbol{t_N}\}$ and $\boldsymbol{t_i}$ denotes the i-th time period. In our experiments, the time interval is usually set to be monthly/seasonly/yearly granularity. In this way, a sequence of BoI features $\boldsymbol{B} = \{\boldsymbol{b_{t_1}}, \boldsymbol{b_{t_2}}, ..., \boldsymbol{b_{t_N}}\}$ can be obtained, where $\boldsymbol{b_{t_i}}$ denotes the BoI feature corresponding to the i-th time period $\boldsymbol{t_i}$. After that, SMEN is used to map $\boldsymbol{B}$ to low-dimensional user representations. As shown in Fig. 1 (b), SMEN consists of a multi-anchor module, a time aggregation module, a multi-scale aggregation module and a contrastive learning module. Details of the model will be described in the following subsections.

2.2.1 Multi-anchor Module. The data of user behavior, which is highly unstructured and complex, implies different preferences of the user. To capture diverse aspects of user preferences, a novel multi-anchor module is proposed. Due to the fact that each user has different interests, and the degree of preference for different interests is also different, the design of multi-anchor module is quite different from multi-head attention. Specifically, suppose there are M anchors, and each of them indicates a certain preference of users. Let $\boldsymbol{b}$ be a BoI feature, the module converts $\boldsymbol{b}$ to M low-dimensional representations as shown in Fig. 2. Each representation is computed as

$$\boldsymbol{r^i} = ReLU(\boldsymbol{\alpha_i}^T f(\boldsymbol{b})) = ReLU(\sum_j^D \alpha_{ij} f(b_j)) = ReLU(\sum_j^D \alpha_{ij} b_j \boldsymbol{W_j^e}), \tag{5}$$

where $f(b_j) = b_j \boldsymbol{W_j^e}$ is the 'interest' embedding function, b_j is the j-th element of $\boldsymbol{b}$, $\boldsymbol{W^e} = (\boldsymbol{W_1^e}, \boldsymbol{W_2^e}, ..., \boldsymbol{W_D^e})^T$ is the embedding matrix. And α_{ij} is the attention weight between the i-th anchor and the j-th 'interest', which measures the portion assigned to the i-th preference from the j-th behavior 'interest'. The weight α_{ij} is defined as

$$\alpha_{ij} = \frac{\exp(\boldsymbol{W_i^a k_j})}{\sum_l \exp(\boldsymbol{W_l^a k_j})}, \tag{6}$$

where $\boldsymbol{W_i^a}$ is the vector corresponding to the i-th anchor, $\boldsymbol{W^a} = (\boldsymbol{W_1^a}, \boldsymbol{W_2^a}, ..., \boldsymbol{W_M^a})^T$ is the anchor matrix, and $\boldsymbol{k_j} = \boldsymbol{W^p} ReLU(\boldsymbol{W_j^e})$ is the interest vector corresponding to the j-th 'interest'. $\boldsymbol{W^e} \in \mathbb{R}^{D \times H}$, $\boldsymbol{W^a} \in \mathbb{R}^{M \times H}$, and $\boldsymbol{W^p} \in \mathbb{R}^{H \times H}$ are learned parameters. $\boldsymbol{r^i}$ can be computed efficiently since $\boldsymbol{b}$ is a sparse vector. Due to the different anchor vectors, different attention weights can be generated for each 'interest'. Finally, a group of different aggregated representations $\boldsymbol{R} = \{\boldsymbol{r^1}, \boldsymbol{r^2}, ..., \boldsymbol{r^M}\}$ can be obtained ($\boldsymbol{M}$ indicates the total number of anchors). In this way, we can learn different aspects of user preferences. Specially, experiments prove that SMEN can achieve almost lossless dimensionality reduction mainly due to the multi-anchor module.

2.2.2 Time Aggregation Module. Through the multi-anchor module, a sequence of representation groups $\boldsymbol{\mathcal{R}} = \{\boldsymbol{R_{t_1}}, \boldsymbol{R_{t_2}}, ..., \boldsymbol{R_{t_N}}\}$ for each user is obtained, where $\boldsymbol{R_{t_i}} = \{\boldsymbol{r^1_{t_i}}, \boldsymbol{r^2_{t_i}}, ..., \boldsymbol{r^M_{t_i}}\}$ is a representation group generated by multi-anchor module corresponding to the BoI feature $\boldsymbol{b_{t_i}}$ in time period $\boldsymbol{t_i}$. In the time aggregation module, each sequence $\boldsymbol{R^i} = \{\boldsymbol{r^i_{t_1}}, \boldsymbol{r^i_{t_2}}, ..., \boldsymbol{r^i_{t_N}}\}$ is aggregated separately to yield a new representation $\tilde{\boldsymbol{r}}^{\boldsymbol{i}} \in \mathbb{R}^H$. Thus the variable-size representation sequence $\boldsymbol{\mathcal{R}}$ can be transformed into M fixed-size representations $\tilde{\boldsymbol{R}} = \{\tilde{\boldsymbol{r}}^{\boldsymbol{1}}, \tilde{\boldsymbol{r}}^{\boldsymbol{2}}, ..., \tilde{\boldsymbol{r}}^{\boldsymbol{M}}\}$.

There are various methods which can be used to aggregate variable-size sequences, such as average/max pooling and RNNs. Compared to average and max pooling, RNNs are more appropriate to capture variations over time. Among all RNN-based models, long short-term memory (LSTM) and gated recurrent units (GRU) are most commonly used. Considering that GRU has fewer parameters and is less computationally intensive, GRU is adopted to the time aggregation module in this work.

2.2.3 Multi-scale Aggregation Module. Under a time division $\boldsymbol{T} = \{\boldsymbol{t_1}, \boldsymbol{t_2}, ..., \boldsymbol{t_N}\}$, we have obtained M representations $\tilde{\boldsymbol{R}} = \{\tilde{\boldsymbol{r}}^{\boldsymbol{1}}, \tilde{\boldsymbol{r}}^{\boldsymbol{2}}, ..., \tilde{\boldsymbol{r}}^{\boldsymbol{M}}\}$ through the BoI, multi-anchor module and time aggregation module as introduced above. It's worth noting that if the time interval is too small, the input sequence of SMEN, i.e. $\boldsymbol{B} = \{\boldsymbol{b_{t_1}}, \boldsymbol{b_{t_2}}, ..., \boldsymbol{b_{t_N}}\}$ will become extremely long, which contains extensive details but causes modeling difficulties due to catastrophic forgetting. On the other hand, details may be lost if the time interval is too large.

To address this trade-off, multi-scale aggregation module is designed. The module captures diverse patterns from user behavior by aggregating several representations generated at different granularities. For example, if user behavior is aggregated at two

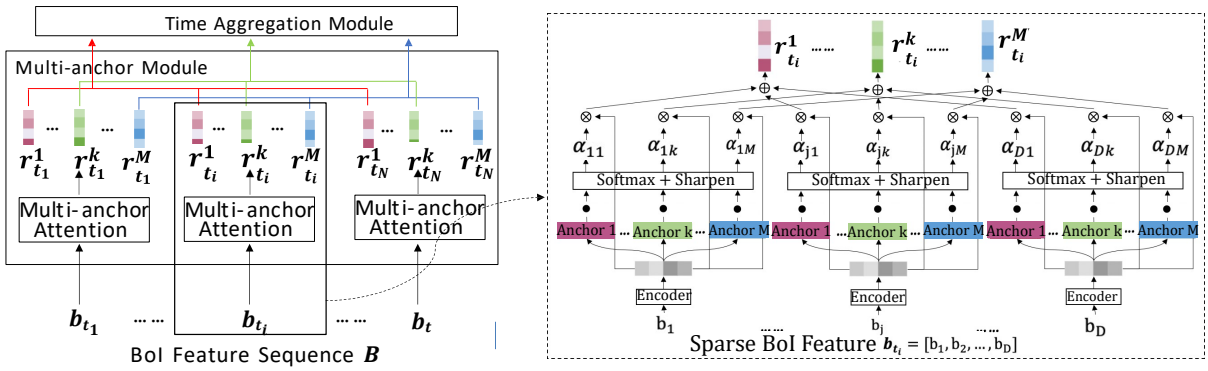


Figure 2: Illustration of the multi-anchor module in SMEN. The module outputs a group of diverse representations by assigning different portions of each 'interest' to different anchors. The • denotes the dot product operation, ⊗ denotes the scalar multiply operation, and ⊕ denotes vector sum operation.

granularities, i.e. $\boldsymbol{T} = \{t_1, t_2, ..., t_N\} = \{t'_1, t'_2, ..., t'_{N'}\}$ (e.g., if the length of time period is 5 years, $\boldsymbol{N}$ is 60 for monthly granularity, and $\boldsymbol{N'}$ is 5 for yearly granularity). Thus, two sequences of BoI feature $\boldsymbol{B} = \{\boldsymbol{b}_{t_1}, \boldsymbol{b}_{t_2}, ..., \boldsymbol{b}_{t_N}\}$ and $\boldsymbol{B'} = \{\boldsymbol{b}_{t'_1}, \boldsymbol{b}_{t'_2}, ..., \boldsymbol{b}_{t'_{N'}}\}$ at different scales correspondingly are obtained. Then, two groups of representations $\tilde{\boldsymbol{R}} = \{\tilde{\boldsymbol{r}}^1, \tilde{\boldsymbol{r}}^2, ..., \tilde{\boldsymbol{r}}^M\}$ and $\tilde{\boldsymbol{R}'} = \{\tilde{\boldsymbol{r}'}^1, \tilde{\boldsymbol{r}'}^2, ..., \tilde{\boldsymbol{r}'}^M\}$ are obtained after the multi-anchor module and time aggregation module with parameter sharing. Finally, we aggregate $\tilde{\boldsymbol{R}}$ and $\tilde{\boldsymbol{R}'}$ with self-attention as follow:

$$\hat{\boldsymbol{r}}^i = s^i([\tilde{\boldsymbol{r}}^i, \tilde{\boldsymbol{r}'}^i]) \odot \tilde{\boldsymbol{r}}^i + (1 - s^i([\tilde{\boldsymbol{r}}^i, \tilde{\boldsymbol{r}'}^i])) \odot \tilde{\boldsymbol{r}'}^i, \tag{7}$$

where $\odot$ is the Hadamard product, $[\cdot]$ is an operation concatenating vectors along the last dimension, and $s^i(\cdot)$ is the switch function which is implemented as a fully-connected layer followed by a sigmoid function. The switch function controls the fusion between BoI feature sequences of different scales. The output of this module $\hat{\boldsymbol{R}} = [\hat{\boldsymbol{r}}^1, \hat{\boldsymbol{r}}^2, ..., \hat{\boldsymbol{r}}^M]$ is served as the final user representation.

2.2.4 Contrastive Learning Module. As mentioned in 2.1.1, contrastive loss is also used to learn user representation in this network. In order to obtain a unified vector, a nonlinear projection head $h(\cdot)$ is added, as used in SimCLR [4]. The function $h(\cdot)$ is computed as

$$\begin{aligned} \boldsymbol{v} &= ReLU([\boldsymbol{W}_1^{h_1}\hat{\boldsymbol{r}}^1, \boldsymbol{W}_2^{h_1}\hat{\boldsymbol{r}}^2, ..., \boldsymbol{W}_M^{h_1}\hat{\boldsymbol{r}}^M]) \\ h(\boldsymbol{v}) &= \boldsymbol{W}^{h_2} ReLU(\boldsymbol{v} + mix(\boldsymbol{v}, \boldsymbol{W}^{h_3})). \end{aligned} \tag{8}$$

The function $mix(\boldsymbol{v}, \boldsymbol{W}^{h_3})$ is to allow interaction between different preferences, and is implemented as

$$mix(\boldsymbol{v}, \boldsymbol{W}^{h_3}) = r_2(\boldsymbol{W}^{h_3} r_1(\boldsymbol{v})), \tag{9}$$

where $r_1(\cdot)$ reshapes $\boldsymbol{v}$ to a matrix of size $M \times H$, and $r_2(\cdot)$ reshapes $\boldsymbol{W}^{h_3} r_1(\boldsymbol{v})$ to a vector. $\boldsymbol{W}_i^{h_1} \in \mathbb{R}^{H \times H}$, $\boldsymbol{W}^{h_2} \in \mathbb{R}^{MH \times MH}$, and $\boldsymbol{W}^{h_3} \in \mathbb{R}^{M \times M}$ are trainable parameters.

Two behavior sequences drawn from the same user are treated as a positive pair, and behavior sequences from other users are negatives. All sequences are continuous sub-sequences randomly sampled from the whole life time $\boldsymbol{T}$. Finally, the contrastive loss is defined the same as equation (3).

Table 1: Statistics of the datasets. $Tr(\cdot)$ indicates the truncation threshold.

Dataset	$\|U\|$	$Max(\|\boldsymbol{S}\|)$	$Tr(\|\boldsymbol{x}\|)$	$\|V\|$
Amazon	43,531,850	13,122	35	103,581
Industrial	214,317,285	1,952,546	24	178,422

3 EXPERIMENTS

In this section, we conduct several experiments to evaluate the performance. The proposed method is compared with several state-of-the-art models on two benchmarks: Amazon dataset [18] and a real-world industrial dataset. A detailed ablation study is also conducted to validate the effectiveness of each module in LURM.

3.1 Datasets & Training Pipeline

Performance comparisons are conducted on a public dataset and an industrial dataset. Table 1 shows the statistics of two datasets. The public Amazon dataset contains behavior data like product reviews and metadata like product titles and categories from Amazon. For each user, the reviewed product titles constitute a sequence of review behaviors. The industrial dataset is collected from visit/order logs on a popular e-commerce platform. The whole training process consists of two parts, i.e., training the universal representation model in a self-supervised manner (Stage I) and training an MLP for each specific downstream task (Stage II). The training datasets for stage I are collected as follows. For Amazon, the dataset is constructed by collecting users who have behaviors between 1997-01 and 2017-12. For industrial dataset, we construct a training dataset by randomly sampling hundreds of millions of users who have behaviors between 2016-06 and 2021-05. Note that we do not fine-tune the representation model on stage II, and only training simple MLPs for applications are easy, efficient and low-cost.

Table 2: Comparison in terms of AUC(%)/ACC(%) on the Amazon dataset. Several category preference identification tasks are evaluated. The best score is bold.

Method	Books of Literature	Games of Sports	Outdoor-hunting	Musical Instrument
TextCNN	73.84/72.83	61.00/69.92	66.06/72.76	68.28/71.81
HAN	79.12/77.49	66.62/70.01	71.19/73.42	70.01/72.12
TF-IDF	78.62/77.21	65.69/68.69	71.90/72.19	69.16/68.62
Doc2Vec	71.21/68.26	66.75/66.85	71.58/66.36	70.29/67.37
PTUM	78.66/77.91	66.06/68.97	70.34/73.01	69.50/72.92
SUMN	79.57/77.73	66.83/70.63	72.27/73.44	70.61/73.43
Ours-LURM	**82.55/79.13**	**68.02/72.43**	**77.59/75.19**	**74.04/74.43**

3.2 Downstream Tasks

Two kinds of downstream tasks are used to evaluate our method: category preference identification and user profiling prediction.

Category preference identification refers to the task of predicting whether users have preferences in the target category, *i.e.*, whether a user would have behaviors on the items of the target category in a future time period. For Amazon, we collect user behaviors from 1997-01 to 2017-12 to form the network inputs, and pass them through LURM to obtain full-life cycle representations (marked as 'LURM'). A user is labeled as positive if there exists at least one review log between 2018-01 and 2018-10. There are four categories being included, *i.e.*, 'Books of Literature', 'Games of Sports', 'Outdoor-hunting', and 'Musical Instrument'. On the industrial dataset, we collect behaviors of two time periods to infer user representations, aiming to show the effectiveness of full-life cycle modeling, as well as to compare to competitors under the same conditions. Specifically, one deploys behaviors from 2020-06 to 2021-05, marked as 'LURM'. The other one uses behaviors from 2021-04 to 2021-05, marked as 'LURM(2M)'. The behavior logs between 2021-07 and 2021-08 are used for labeling. We consider three categories including clothing, shoe and coffee. Due to the time and resource limitations, it is impractical to include all categories in the experiments. The selected categories are expected to be representative, covering different scales (namely the number of products in the category), industries, etc.

User profiling prediction aims to identify user properties such as gender and age. We notice that increasing the numbers of behaviors would significantly benefit the performance on user profiling tasks. To show that, we conduct experiments on the industrial dataset, where the number can reach millions. We also collect behaviors of two time periods to infer user representations. One deploys behaviors from 2016-06 to 2021-05, marked as 'LURM'. The other one uses behaviors from 2021-04 to 2021-05, marked as 'LURM(2M)'. Two specific tasks are involved: (1) user age classification (age is divided into 6 classes) which predicts the age ranges of users, and (2) baby age classification (7 classes). For both tasks, the ground-truth labels are collected from an online questionnaire.

In the experiments, we randomly select 80% of the samples for each task to train downstream models and the rest for performance validation. A detailed demonstration of the effectiveness of full-life cycle modeling is given in Section 3.6.

3.3 Competitors

We compare our LURM against a rich set of user representation methods, including TF-IDF, Doc2Vec, TextCNN, HAN, PTUM [28] and SUMN [11]. TF-IDF and Doc2Vec view the whole user behavior sequence as a document. TF-IDF generates a sparse high-dimensional vector, while Doc2Vec learns to represent the document by a dense vector. We also compare LURM with two supervised methods, namely TextCNN and HAN, which are trained particularly for each downstream task. Specifically, TextCNN adopts convolutional neural networks (CNN) on the embedding sequence of all words appeared in behaviors and uses max-pooling to get the final user representation. The user representation encoder and the classifier are trained together. HAN employs a hierarchical attention network, where two levels of attention operations are used to aggregate words and behavior embeddings respectively. Finally, we compare LURM with two newly presented self-supervised methods for user modeling, PTUM and SUMN. PTUM is designed based on BERT and proposes two self-supervision tasks for pre-training. The first one is masked behavior prediction, which can model the relatedness between historical behaviors. The second one is next K behavior prediction, which characterizes relatedness between past and future behaviors. SUMN proposes a multi-hop aggregation layer to refine user representations and uses behavioral consistency loss to guide the model to extract latent user factors.

3.4 Training Details

Item Embedding As in[11, 28], the number of words in each item $\boldsymbol{x}$ is truncated. The truncation principle is that 95% data values can be covered by the chosen threshold. As a result, we set 35 on the Amazon dataset and 24 on the Industrial dataset (see Table 1). The window size β is set to 5 days[2]., and the temperature τ is set to be 0.1.

BoI & SMEN We set the latent dimensions of the word/item embeddings and all hidden layers to be 128. The size of the 'interest' vocabulary is set to be 10^5 in order to retain details as many as possible. The number of the anchors in multi-anchor module, namely M, is set as 10, and thus the dimension of the final user representation is 1280. We will discuss the performance effects of these two hyper-parameters later. In the multi-scale aggregation module, both the inputs of month-granularity and year-granularity are deployed to capture diverse behavior patterns behind different time periods. We will also compare the performances of using inputs of different granularities in the ablation study section. Moreover, for the user

Table 3: Comparison in terms of AUC(%)/ACC(%) on Industrial dataset. Two profiling prediction tasks are evaluated. The best score is bold.

Method	Age	Baby Age
TextCNN	86.40/61.02	72.19/66.01
TF-IDF	87.73/61.75	73.08/67.26
SUMN	85.35/60.61	74.20/67.63
Ours-LURM(2M)	88.48/61.99	82.71/69.29
Ours-LURM	**96.09/78.43**	**94.59/84.91**

Table 4: Comparison in terms of AUC(%)/ACC(%) on Industrial dataset. Three category preference identification tasks are evaluated. The best score is bold.

Method	Clothing	Shoe	Coffee
TextCNN	76.64/76.13	84.33/80.24	80.49/79.46
TF-IDF	77.61/78.01	85.04/81.15	81.75/80.78
SUMN	73.86/74.16	83.47/79.63	79.89/79.42
Ours-LURM(2M)	78.37/78.57	85.15/81.18	81.94/80.78
Ours-LURM	**80.37/79.10**	**86.30/81.65**	**83.11/81.18**

representation model, the loss is optimized by Adam optimizer with a learning rate of 0.001, and a batch size of 256.

Downstream Model For downstream tasks, a simple MLP classifiers is applied after the derived user representations. The MLP contains only one hidden layer with a dimension of 64. An Adam optimizer with a learning rate of 0.001 is used. The batch size is set as 128. Note that LURM, SUMN and PTUM are not fine-tuned for downstream tasks in our experiments.

Competitors On Amazon dataset, only TF-IDF and Doc2Vec can make use of the entire behavior data to generate representations (same as our method). Meanwhile, the length of input behaviors of other competitors, namely TextCNN, HAN, SUMN and PTUM, is limited to 50 due to memory limitation. On the industrial dataset, all competitors deploy behavior data of two months as inputs, limited by their ability to handle long sequences. For supervised competitors, we use Adam with a learning rate of 0.001 as the optimizer, and the batch size is set as 256. For unsupervised ones, only downstream model needs training. The configurations are set to be the same as in the preceding paragraph.

3.5 Results

Table 2 shows the comparison of category preference identification on the public Amazon dataset. The last row shows the result of our models. It can be seen that LURM consistently outperforms other unsupervised methods, e.g., about 3.23%/1.49% average improvements than SUMN and 4.41%/2.09% average improvements than PUTM in terms of AUC and ACC respectively. Moreover, though our goal is not to beat supervised competitors, LURM still achieves convincing results in comparison with TextCNN and HAN, about 8.26% and 3.82% higher average AUC, and about 3.47% and 2.04% higher average ACC, respectively. [1]

Table 3 and Table 4 shows the results of two user profiling tasks and three category preference identification tasks on the industrial dataset, respectively. Similar observations can be concluded. LURM(2M) achieves consistently better results than any other methods with behavior sequences of the same length. One can also see that further improvements can be achieved by using more behaviors within a longer time period. For example, LURM achieves about 5.66%/10.16% average improvements than LURM(2M). Furthermore, we notice that the improvements are more significant on the user profiling tasks. This is probably because the quantity of behaviors related to user profiling is usually larger than that related to category preference prediction. Specifically, incorporating behaviors occurred even years ago can still benefit the performance of user profiling, *e.g.*, age prediction.

3.6 Ablation Studies & Discussion

Fig. 3 shows the performances of LURM on several downstream tasks of two datasets when using inputs of different lengths. It can be seen from Fig. 3(c) that the richer the behavior, the better the performance on user profiling. Compared to using only behaviors of two months, an average improvement of 9.75%/16.03% can be achieved when using behaviors of 5 years on these tasks. While on category preference prediction tasks of both datasets, there are noticeable improvements in the early stage, and then the improvements gradually decrease. The best performance is obtained when using behaviors with a length of one or two years. It may because that the information gain is limited when mining patterns from extremely long behavior sequences for these tasks.

Table 5 shows the results of LURM with different configurations on industrial dataset. The first row shows the results of using high-dimensional BoI features as user representations in downstream tasks. It can be seen that the difference between BoI and LURM is tiny, which proves that SMEN can achieve almost lossless dimensionality reduction.

We also verify the necessity of multi-scale aggregation module. The third and fourth rows of Table 5 show the results of LURM with input at the monthly/yearly granularity only, respectively. It can be seen that using multi-scale aggregation module can achieve significant improvements on user profiling prediction tasks, while the benefits are relatively marginal disappear on category preference prediction tasks. The reason is similar, refer to our explanations of the results in Fig. 3.

The last two rows of Table 5 show the results of LURM with 1 anchor, where the dimensions are set as 128 and 1280 respectively. Taking BoI as a benchmark, it can be seen that LURM achieves almost lossless dimensionality reduction compared to LURM(1anchor-128) and LURM(1anchor-1280), which verifies the effectiveness of the multi-anchor module.

The performance effects of the cluster number D are shown in Fig. 4. The experiments are conducted on the industrial dataset. In general, one can see that the performance improves if we increase

[1]BERT-like methods are not included because the number of behaviors (i.e., input tokens) are tremendous, and accordingly the resource costs are too high.

Table 5: Comparison of LURM with different configurations on industrial dataset.

Method	Age	Baby Age	Clothing	Shoe	Coffee
BoI	96.48/80.19	93.29/83.57	80.83/79.33	86.69/82.16	83.78/81.74
LURM	96.09/78.43	94.59/84.91	80.37/79.10	86.30/81.65	83.11/81.18
LURM(monthly)	95.82/77.54	93.19/82.17	80.30/79.02	86.26/81.61	83.08/81.14
LURM(yearly)	95.61/77.19	92.80/80.93	80.24/78.88	86.17/81.36	82.66/80.78
LURM(1anchor-128)	93.98/72.78	86.13/71.22	78.93/78.27	85.80/81.11	81.98/80.39
LURM(1anchor-1280)	94.35/73.71	89.21/75.35	79.38/78.54	86.07/81.32	82.34/80.59

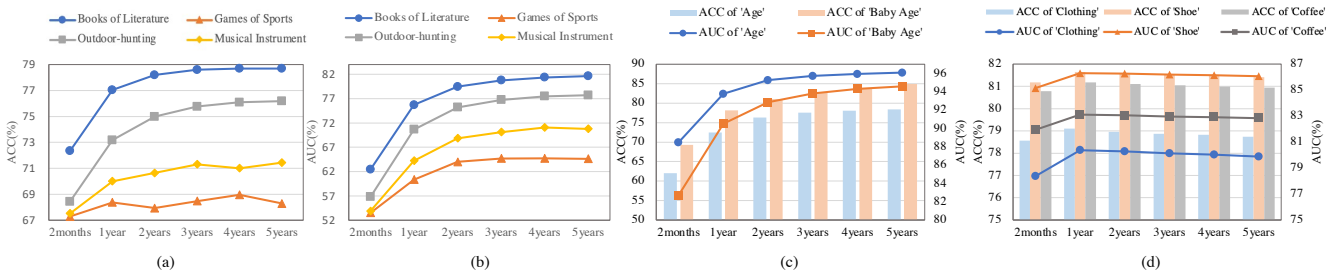


Figure 3: Comparison of LURM with inputs of different lengths on several downstream tasks. (a-b) show the performance on the Amazon Dataset, and (c-d) show the results on the Industrial Dataset

D from 1000 to 100000, while the improvements fade away when D reaches 200000.

We also explore the performance effect of the cluster assignment strategy of items. To this end, a soft assignment strategy is tested, *i.e.*, assigning a item to multiple clusters according to cosine distance. According to our observations, the performance difference of these two strategies is less than 1%. Since soft assignment would bring extra storage and calculation costs, hard assignment is adopted in this paper.

3.7 Visualization of Representation

The representations learned on the industrial dataset are intuitively illustrated in Fig. 5. We visualize the outputs of the multi-anchor module for clarity, which are mapped into the 2-dimensional space with t-SNE[16] (including representations generated by 10 anchors from 100 randomly selected users). Different colors correspond to difference anchors. Visually, this illustration demonstrates that LURM can gradually mine and characterize different aspects of user interests, showing great diversity.

3.8 Comparison to Concurrent Methods

This work has been applied widely in Alibaba. During the process, more experiments are conducted on our own dataset, including comparisons with concurrent methods [25, 26].

Though we cannot release the datasets and specific results, the conclusions and observations can be shared. Generally, LURM have similar performance with ICL[26], but LURM (long) (namely incorporating more behaviors) achieves about 2%-19% higher AUC than ICL. We also have compared our method to a CLIP-like model, which is similar to [25]. The observations are similar, due to the capacity of large model, the CLIP-like model performs slightly better than LURM, but LURM (long) shows superior performance, outperforming CLIP-like model by about 1%-3%.

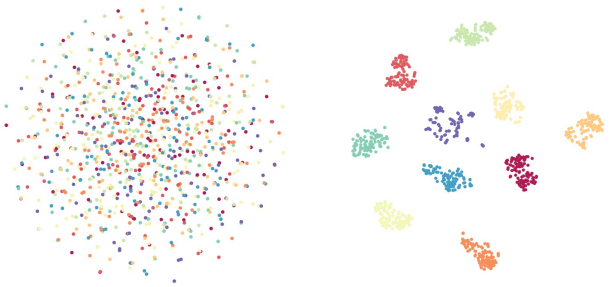
4 RELATED WORKS

4.1 Universal User Modeling

Compared with task-specific user modeling that requires more resources, universal user representations are preferred to serve different downstream tasks. In recent years, some works dedicated to learning universal user representations have been proposed [1, 7, 11, 19, 24–28]. Ni *et al.* [19] proposed a representation learning method based on multi-task learning, which enabled the network to generalize universal user representations. Extensive experiments showed the generality and transferability of the user representation. However, the effectiveness of this method may still suffer due to the selection of tasks and the need of labels. To release the burden of labeling, Andrews *et al.* [1] proposed a novel procedure to learn user embedding by using metric learning. They learned a mapping from short episodes of user behaviors to a vector space in which the distance between points captures the similarity of the corresponding users invariant features. Gu *et al.* [11] proposed a network named self-supervised user modeling network (SUMN) to encode user behavior data into universal representation. They introduced a behavior consistency loss, which guided the model to fully identify and preserve valuable user information under a self-supervised learning framework. Wu *et al.* [28] proposed pre-trained user models (PTUM), which can learn universal user models based on two self-supervision tasks for pre-training. The first one

(a) Age (b) Baby Age (c) Clothing (d) Shoe (e) Coffee

Figure 4: Performance of different cluster number D in downstream tasks. The green line denotes the AUC score, and the blue bars represent the ACC



(a) (b)

Figure 5: t-SNE visualization of user representations. The points of the same color indicate that they come from the same anchor. (a) representations after model initialization, (b) representations learning by multi-anchors after training.

was masked behavior prediction, which can model the relatedness between historical behaviors. The second one was next K behavior prediction, which can model the relatedness between past and future behaviors. With the help of self-supervised pretext tasks, these methods can obtain universal user representation that can serve different downstream tasks. Unfortunately, these methods can only process user behavior sequences with a length of thousands (e.g., the largest length of behavior sequences used in [25] is 2048, and the length is truncated to 320 in [26]), and cannot leverage the rich information brought by full-life cycle user behaviors.

4.2 Full-life Cycle User Modeling

Previous works of task-specific user modeling have shown that considering long-term historical behavior sequences for user modeling can significantly improve the performance of different tasks [3, 21–23]. Ren *et al.* [23] proposed a hierarchical periodic memory network for full-life cycle sequential modeling. They built a personalized memorization for each user, which remembers both intrinsic user tastes and multi-facet user interests with the learned while compressed memory. Pi *et al.* [21] decoupled the user modeling from the whole CTR prediction system to tackle the challenge of the storage cost and the system latency. Specifically, they proposed a user interest center module for real-time inference and a memory-based network that can be implemented incrementally. Pi *et al.* also designed a search-based interest model (SIM) with a cascaded two-stage search paradigm to capture the diverse long-term interest with target item. Unfortunately, the length of the user behavior sequence that these models can handle is still limited. Moreover, these models are all trained on specific tasks(e.g., CTR), which limits the generalization ability.

To the best of our knowledge, we are the pioneer to make full-life cycle modeling possible for learning general-purpose user representation. LURM allows for encoding even millions of historical behaviors to improve the quality of user representations.

5 CONCLUSION

In this work, a novel framework named LURM is proposed to model full-life cycle user behaviors with any length. With the ability to model full-life cycle user behaviors, our method shows promising results on different downstream tasks and datasets. Although our method has made some progress, there is still space for improvement. In the future research work, we will consider more types of tasks; input data in different modalities, such as images, video, and audio; more dedicated network architecture and so on.

ACKNOWLEDGMENTS

This work was supported in part by Major Programs of the National Social Science Foundation of China(Grant No. 22ZD147) and Alibaba Group through Alibaba Innovative Research Program.

REFERENCES

[1] Nicholas Andrews and Marcus Bishop. 2019. Learning invariant representations of social media users. *arXiv preprint arXiv:1910.04979* (2019).
[2] Oren Barkan and Noam Koenigstein. 2016. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.
[3] Yue Cao, Xiaojiang Zhou, Jiaqi Feng, Peihao Huang, Yao Xiao, Dayao Chen, and Sheng Chen. 2022. Sampling Is All You Need on Modeling Long-Term User Behaviors for CTR Prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2974–2983.
[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[5] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* (2019).

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Tao Ding, Warren K Bickel, and Shimei Pan. 2017. Multi-view unsupervised user feature embedding for social media-based substance use prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2275–2284.

[8] Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations.. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 331–338.

[9] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th international conference on world wide web*. 278–288.

[10] Li Fei-Fei and Pietro Perona. 2005. A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2. IEEE, 524–531.

[11] Jie Gu, Feng Wang, Qinghui Sun, Zhiquan Ye, Xiaoxiao Xu, Jingmin Chen, and Jun Zhang. 2020. Exploiting Behavioral Consistence for Universal User Representation. *arXiv preprint arXiv:2012.06146* (2020).

[12] Zellig S Harris. 1954. Distributional structure. *Word* 10, 2-3 (1954), 146–162.

[13] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. 1–9.

[14] Wendi Ji, Yinglong Sun, Tingwei Chen, and Xiaoling Wang. 2020. Two-stage Sequential Recommendation via Bidirectional Attentive Behavior Embedding and Long/Short-term Integration. In *2020 IEEE International Conference on Knowledge Graph (ICKG)*. IEEE, 449–457.

[15] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*. PMLR, 1378–1387.

[16] Van Der Maaten Laurens and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 2605 (2008), 2579–2605.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546* (2013).

[18] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 188–197.

[19] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. 2018. Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 596–605.

[20] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[21] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2671–2679.

[22] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based User Interest Modeling with Lifelong Sequential Behavior Data for Click-Through Rate Prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2685–2692.

[23] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, et al. 2019. Lifelong sequential modeling with personalized memorization for user response prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 565–574.

[24] Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation* (2004).

[25] K. Shin, H. Kwak, S. Y. Kim, M. N. Ramstrom, J. Jeong, J. W. Ha, and K. M. Kim. 2021. Scaling Law for Recommendation Models: Towards General-purpose User Representations. (2021).

[26] Q. Sun, J. Gu, B. Yang, X. X. Xu, R. Xu, S. Gao, H. Liu, and H. Xu. 2021. Interest-oriented Universal User Representation via Contrastive Learning. (2021).

[27] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2022. UserBERT: Pre-training User Model with Contrastive Self-supervision. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2087–2092.

[28] Chuhan Wu, Fangzhao Wu, Tao Qi, Jianxun Lian, Yongfeng Huang, and Xing Xie. 2020. PTUM: Pre-training User Model from Unlabeled User Behaviors via Self-supervision. *arXiv preprint arXiv:2010.01494* (2020).

[29] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237* (2019).

[30] Jinfeng Yi, Lijun Zhang, Jun Wang, Rong Jin, and Anil Jain. 2014. A single-pass algorithm for efficiently recovering sparse cluster centers of high-dimensional data. In *International Conference on Machine Learning*. PMLR, 658–666.

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

[32] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 729–732.

[33] Zeping Yu, Jianxun Lian, Ahmad Mahmoody, Gongshen Liu, and Xing Xie. 2019. Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation.. In *IJCAI*. 4213–4219.