UserBERT: Modeling Long- and Short-Term User Preferences via Self-Supervision

Tianyu Li*, Ali Cevahir*, Derek Cho, Hao Gong, DuyKhuong Nguyen, Björn Stenger

Rakuten Institute of Technology, Rakuten Group, Inc.
{derek.cho, bjorn.stenger}@rakuten.com

Abstract

E-commerce platforms generate vast amounts of customer behavior data, such as clicks and purchases, from millions of unique users every day. However, effectively using this data for behavior understanding tasks is challenging because there are usually not enough labels to learn from all users in a supervised manner. This paper extends the BERT model to e-commerce user data for pretraining representations in a self-supervised manner. By viewing user actions in sequences as analogous to words in sentences, we extend the existing BERT model to user behavior data. Further, our model adopts a unified structure to simultaneously learn from long-term and short-term user behavior, as well as user attributes. We propose methods for the tokenization of different types of user behavior sequences, the generation of input representation vectors, and a novel pretext task to enable the pretrained model to learn from its own input, eliminating the need for labeled training data. Extensive experiments demonstrate that the learned representations result in significant improvements when transferred to three different real-world tasks, particularly compared to task-specific modeling and multi-task representation learning.

Introduction

The choice of data representation, *i.e.*, how to extract meaningful features, has significant impact on the performance of machine learning applications (Bengio, Courville, and Vincent 2013). Therefore, data processing and feature engineering have been key steps in model development. To extend the applicability of the models, recent research on representation learning aims to discover the underlying explanatory factors hidden in raw data. With rapid advances in this direction, we have witnessed breakthroughs in the areas of computer vision (Doersch, Gupta, and Efros 2015; Sharif Razavian et al. 2014; Simo-Serra et al. 2015) and natural language processing (NLP) (Mikolov et al. 2013; Pennington, Socher, and Manning 2014; Lin et al. 2017).

Similarly, for building user-oriented industrial applications like next purchase prediction and recommendation, much effort has been spent on understanding business models and user behavior for creating useful features (Richardson, Dominowska, and Ragno 2007; Covington, Adams, and Sargin 2016). However, this is a time-consuming and application-specific process. It is also challenging to reuse these features, or to share the gained knowledge between different services and tasks.

To address the issue of isolated feature engineering and task-specific design, prior work has explored pretraining and transfer learning ideas. For example, multi-task learning (MTL) has shown promising results (Ni et al. 2018). However, MTL has intrinsic challenges, *e.g.*, deciding which tasks to learn jointly (Standley et al. 2019), or how to weigh tasks to achieve optimal performance (Kendall, Gal, and Cipolla 2018). More importantly, the training still hinges on large amounts of well-annotated user labels.

Inspired by the BERT model, which has been immensely useful across a host of NLP tasks (Jacob et al. 2019; Lan et al. 2020), recent work proposed pretraining user representations on unlabeled behavior data in a self-supervised manner (Wu et al. 2020; Yuan et al. 2020). However, prior work does not take inherent differences between different types of user behavior into account. Our proposal, *UserBERT*, simultaneously learns from three categories of user data, *i.e.*, long-term and short-term behavior, as well as user attributes, via a unified architecture. In this work, we consider *short-term behavior* as user actions during one browsing session, including clicks, searches, and page views. *Long-term behavior* refers to user interest over longer time frames and includes, for example, preferences for particular shops or item genres. For these two behavior types, we first present distinct strategies to discretize them into sequences of *behavioral words*. Compared to modeling single user actions sequentially, the proposed discretization leads to better generalization. The token representation of these behavioral words is computed by concatenation and averaging of the word embeddings of the attribute IDs (*e.g.*, shop, price, or product genre) of each action, and this is followed by the summation of token, position and segment embeddings. These representation vectors are then aligned with the word embeddings of user categorical attributes as the input to UserBERT. With this input, we design a novel *pretext* task, *masked multi-label classification*. The UserBERT model is pretrained via optimizing the multi-label classifications of the multiple attributes in the masked behavioral words.

Despite the parallels between user behavior and sentences, there are substantial differences and challenges in designing the learning procedure in a consistent way. Our

*These authors contributed equally.

model is able to deal with heterogeneous user behavior data, and achieve generalization via effective tokenization and the pretraining task. The UserBERT model explores integrating various types of user data in a unified architecture and learning generic representations with self-supervised signals. In our experiments, the pretrained model is fine-tuned on three different real-world tasks: user targeting, user attribute prediction, and next purchase genre prediction. The results show that UserBERT outperforms task-specific modeling and multi-task learning based pretraining.

Our contributions are summarized as follows:

- We propose UserBERT to pretrain user representations by making the analogy of actions in behavior sequences to words in sentences. We eliminates the need for collecting additional user annotation.
- UserBERT adopts a unified model architecture to enable simultaneous learning from heterogeneous data, including long-term and short-term behavior as well as demographic data.
- We design the discretization of user raw data sequences, the generation of the input representation and a novel pretext task for pretraining.
- We evaluate UserBERT in extensive experiments. Compared with task-specific models without pretraining and multi-task learning based pretraining models, the proposed model achieves higher accuracy on three real-world applications.

Related Work

Pretraining and Transfer Learning

Recent studies have demonstrated that pretraining on large, auxiliary datasets followed by fine-tuning on target tasks is an effective approach (Oquab et al. 2014; Donahue et al. 2014; Hendrycks, Lee, and Mazeika 2019; Ghadiyaram et al. 2019). Multi-task learning has been one of the commonly adopted approaches for pretraining due to its ability to improve generalization (Zhang and Yang 2017; Ruder 2017; Gong et al. 2020). It is shown that the pretrained MTL models can boost performance even when transferred to unseen tasks (Liu et al. 2015; Ni et al. 2018). Despite its success, MTL still has many challenges, such as negative transfer and the learning adjustment between different tasks (Guo et al. 2018). Also, MTL requires large amounts of well-annotated labels to produce satisfying outputs. There are two common forms of adaptation when transferring the pretrained models to a given target task, *i.e.*, *feature-based* in which the pretrained weights are frozen, and directly *fine-tuning* the pretrained model (Peters, Ruder, and Smith 2019). We fine-tune pretrained models in our experiments.

Self-Supervised Learning

Deep learning models can already compete with humans on challenging tasks like semantic segmentation in the CV area (He et al. 2015) as well as a few language understanding tasks (Liu et al. 2019). However, such success relies on adequate amounts of quality training data, which can be expensive to obtain (Kolesnikov, Zhai, and Beyer 2019). As a result, a lot of research efforts aim to reduce dependency on labeled data. Self-supervised learning (SSL), a subclass of unsupervised learning, has been drawing more attention since the recent advances in the NLP field. Instead of using supervision signals, SSL only requires unlabeled data and trains models via formulating a *pretext* learning task. There are two main types of pretext tasks: context-based (Pathak et al. 2016; Noroozi and Favaro 2016; Sermanet et al. 2018; Wu, Wang, and Wang 2019) and contrastive-based (Hjelm et al. 2019; Chen et al. 2020). BERT (Jacob et al. 2019), which our model is built upon, learns the contextual information through bi-directional transformers (Vaswani et al. 2017) in a self-supervised manner.

User Modeling

To build user-oriented machine learning applications, a key challenge is finding an expressive representation of user bevhavior data, so that models can make accurate predictions. For that reason, much effort has been spent on data preprocessing and transformations (Zhu et al. 2010). Deep learning models have successfully mitigated the dependency on human efforts due to its ability to capture underlying representations in raw data (Zhou et al. 2018; Li and Zhao 2020). However, these models need massive supervision signals for training, and they are mostly designed for specific tasks like recommendation (Pei et al. 2019; Sun et al. 2019b) and click-through rate prediction (Zhou et al. 2019).

Despite the success of these deep learning models, they fail to generate promising results for real-world industrial tasks with limited labeled data. To deal with this issue, the methodology that pretrains universal user representations on massive user data, and then fine-tunes them for downstream tasks is explored. The goal is to learn a universal and effective representation for each user that can be transferred to new tasks (Ni et al. 2018; Gong et al. 2020). However, MTL-based pretraining still requires the collection of user labels. Also, it is limited by inherent shortcomings to achieve optimal results (Kendall, Gal, and Cipolla 2018; Guo et al. 2018).

Recent work proposes learning user representations in a self-supervised way. For instance, PTUM applies Masked Behavior Prediction and Next K Behaviors Prediction to pretrain user models (Wu et al. 2020). CL4SRec uses a contrastive learning framework and proposes three data augmentation methods to construct contrastive tasks for pretraining (Xie et al. 2020). However, none of these works consider the intrinsic discrepancies of user behavior types. Also, the pretraining that sequentially models every single user actions is interfered with the randomness of user behavior, and fails to learn underlying user preferences.

The Proposed Approach

In this section, we first briefly review the BERT model, and then elaborate on how to extend it to user data including behavior sequences and demographic attributes.

The BERT Model

BERT is a language representation model that pretrains deep bidirectional representations by jointly conditioning on both

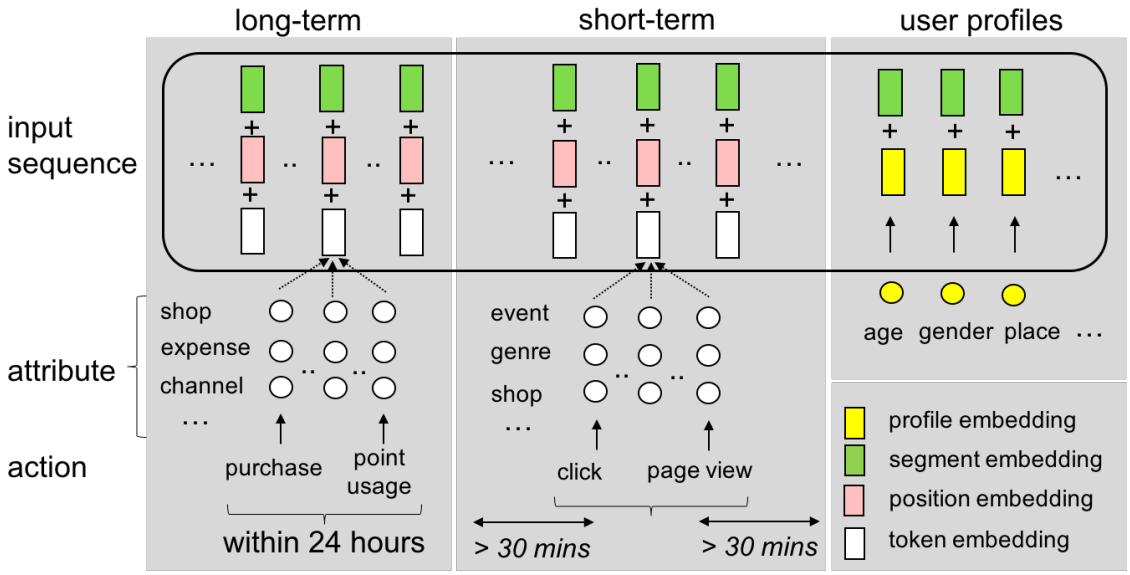

Figure 1: **Tokenization and input representation of long-term and short-term user behavior and attribute data.** *To form behavioral words, we discretize long-term behavior into 24-hour intervals, and segment short-term sequences when there is a break of longer than 30 minutes between two actions. The word embeddings of the attribute IDs of each action are first concatenated. Then, the token representation for one time interval is constructed by the mean of all action embeddings within the interval. The representation in the sequence is the sum of token embeddings and the embeddings for encoding position and segment.*

left and right contexts in all encoding layers (Jacob et al. 2019). The input to the BERT model is a sequence of tokens that can represent both a single text sentence and a pair of sentences. These discrete tokens consist of words and a set of special tokens: separation tokens (SEP), classifier tokens (CLS) and tokens for masking values (MASK). For a token in the sequence, its input representation is a sum of a word embedding, the embeddings for encoding position and segment.

The BERT model is pretrained with two tasks, masked language modeling (MLM) and next sentence prediction. In MLM, the input tokens are randomly masked and the BERT model is trained to reconstruct these masked tokens. In detail, a linear layer is learned to map the final output features of the masked tokens to a distribution over the vocabulary and the model is trained with a cross entropy loss. In next sentence prediction, the inputs are two sampled sentences with a separator token SEP between them. The model learns to predict whether the second sentence is the successor of the first. A linear layer connecting the final output representations of the CLS token is trained to minimize a cross entropy loss on binary labels. Many recent research works focus on extending the BERT model to areas beyond NLP, and successfully achieved state-of-the-art results (Sun et al. 2019a; Lu et al. 2019; Su et al. 2020; Qi et al. 2020).

UserBERT

Tokenization of User Behavior Sequences Our goal is to learn generic user representations that characterize users based on their long-term preferences as well as recent interests. We decide not to sequentially model single actions in long-term and short-term user data. While such modeling is suitable for certain tasks, it is susceptible to overfitting when learning generic user representations. Instead, we learn from a sequence of clustered user actions, in which a cluster represents a routine or a spontaneous interest. Customers often make online purchases with specific intentions, *e.g.*, shopping for a shirt, comic books, or a gift for Mother's Day. Many customers have long-standing preferences for particular stores and sales are heavily impacted by seasonality. These continuous or related actions form a 'word' in a behavior sequence. Similarly, we consider the same regarding short-term user behavior. Users commonly browse web content, moving between pages on an e-commerce site. During this time period, in order to capture the user's interest, we aim to estimate the theme or product genre rather than the specific order of individual actions.

Therefore, we first need to segment raw action data into a sequence of 'behavioral words' for each user, analogous to words in a sentence. As described by Figure 1, we adopt different approaches for long-term and short-term data. Data representing long-standing user preferences is segmented into 24-hour intervals from 4 AM of one day to 4 AM of the next day. Short-term data is segmented if there is a time interval larger than 30 minutes between two actions, similar to the processing steps in (Grbovic and Cheng 2018).

Input Representations In order to enable bidirectional representation learning, we transform the behavioral word sequence into a sequence of input embeddings. We first in-

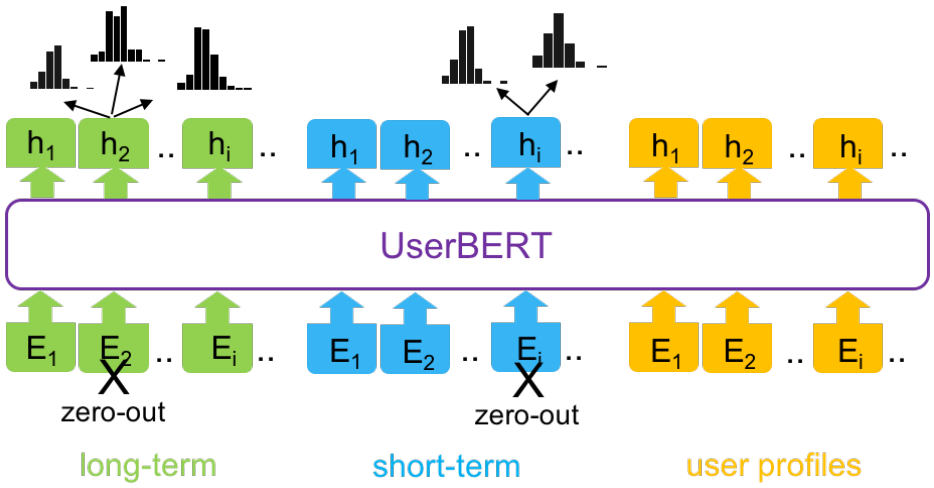

Figure 2: **Pretraining UserBERT.** *The behavioral word representation vectors E_i, in input sequences are randomly masked (zeroed-out), and the masked input is passed through UserBERT. The model is trained to reconstruct the attributes in these masked words. For each attribute, an output layer is connected to the final hidden representations h_i, at the masked positions, and is learned by minimizing the multi-label classification loss.*

troduce the concept of *action* and *attribute* in user behavior data: The action indicates what a user does, *e.g.*, making a purchase or obtaining points for using a service, while the action attributes include, for example, the type of the action, the shop name, the item genre, and price range, as shown in Figure 1. We choose different actions and their corresponding attributes in our dataset to represent long-term and short-term user behavior, and propose separate tokenization strategies for them since we expect to extract inherent user preferences from regular routines over longer time periods, and short-term interests from recent, temporary interactions. In combination with user attribute data, the learned representations are comprehensive and expressive.

To generate input representations, all attribute IDs are first mapped to fixed-length continuous vectors. These attribute vectors are concatenated for each action, obtaining an action embedding. Subsequently, the token representation for one time interval is obtained by taking the mean of all action embeddings within this interval. Finally, the input embedding vector is obtained by summing the token embeddings and the embeddings for encoding position and segment. Segment embeddings are used to differentiate the different types of user data, *i.e.*, long-term user behavior, short-term behavior, and user profile data. Long- and short-term user data share the same processing steps above, but each has their own definitions for token position. For long-term sequences we use the number of days counted from the starting point of the collected training data, for short-term data it is the number of hours. In order to incorporate non-temporal user attribute data to our modeling, we consider categorical attributes like gender as tokens in the user input sequence. Continuous-valued attributes, such as age, are segmented by heuristics and converted into categorical attributes. After mapping attributes to word embedding vectors, these are added to the segment embedding. Note that there is no position embedding for user-attribute embeddings since no order information needs to be captured for these user attributes. The input sequence for each user is formed by aligning the generated representation vectors of user behavior as well as the embeddings of user attributes, see Figure 1.

Pretraining Tasks The generated input sequences allow us to make minimal changes to the BERT architecture and follow the practice in (Jacob et al. 2019). We then pretrain our model to learn bidirectional representations. While the language modeling task seems to naturally apply to our setting, reconstructing the masked 'behavioral words' requires modification since these words contain an assembly of user actions rather than individual words used in the original BERT model. We implement *masked multi-label classification* to predict multiple attributes in the masked behavioral words. More precisely, for each target attribute in a masked token, a linear layer is connected to the final representations which maps to a distribution over the attribute vocabulary, as illustrated in Figure 2. For one masked token, the training loss is the sum of cross entropy losses of all user attribute predictions, *e.g.*, the prediction of shop IDs, genre IDs, etc. The final loss for one input sequence is the sum of the losses of all masked tokens.

For masking input tokens, we follow a similar process as BERT: 15% of tokens are selected uniformly, where 80% of the time the token is zeroed-out and remains unchanged otherwise. We distinguish between three segments of behavioral words from the three types of user data, *i.e.*, long-term, short-term and user attributes. For long and short-term segments, we apply the masking-prediction for pretraining our model, while we do not mask user attributes. To pretrain UserBERT, we first randomly sample a mini-batch of raw user sequences. Then, they are tokenized and transformed

Table 1: Actions and attributes in user behavior data

	Actions	Attributes (with vocabulary size)
long-term	purchase, point usage	action type (2), channel (742), expense range (17), shop (85,124), genre (11,438), hour (24)
short-term	click, search, page view	action type (3), shop (40,804), genre (10,386), device type (2)

to input representations, which is followed by the masking step. In the end, the masked sequences are passed through the model, and the model is trained by minimizing the prediction error for reconstructing what attributes are inside the masked tokens. For each attribute type, a linear layer is learned to map the hidden representations of masked tokens to distributions over its vocabulary for conducting the multi-label classification.

Let i be a randomly sampled index for masking, w_i and $w_{\setminus i}$ be the masked behavioral word and the input after masking to the UserBERT. Also, let n be the number of target attributes for reconstruction prediction, and $f^k(w_{\setminus i}|\theta)$ be the final output vector after softmax layer for k-th attribute in the masked w_i. The loss of the UserBERT model is:

$$L(\theta) = \mathbb{E}_{w\sim D, i\sim\{1,..,t\}} \sum_{k=1}^{n} L_{CE}(y_i^k, f^k(w_{\setminus i}|\theta)), \quad (1)$$

where w is a uniformly sampled input representation sequence from the training dataset D, t is the total number of behavioral words in the long-term and short-term data, y_i^k is the ground truth binary vector for the k-th attribute with its corresponding vocabulary size in the masked w_i and L_{CE} is the cross entropy loss for the multi-label classification. Note that long-term and short-term user behavior have different types and number of attributes in actions. With the pretrained models, we leverage them for fine-tuning on downstream tasks.

Experiments

We experimentally verify whether the proposed UserBERT model is able to yield generic user representations, and evaluate the performance when applying it to different tasks via transfer learning.

Experiment Settings

Datasets Datasets are collected from an online ecosystem of a variety of services, including an e-commerce platform, a travel booking service, a golf booking service and others. Customers can access all services via a unique customer ID, and their activities across the ecosystem are linked together.

We consider two actions as long-term user behavior. The first one is the purchase action on the e-commerce platform, and the second one is a point usage action. Points are earned whenever purchases are made, or when certain services are used. Points can be spent on any service within the ecosystem. The *channel* attribute represents from which service users obtain points or where they spend points. We collected purchase and point usage data over a three-month period. For short-term behavior, we focus on recent activities on the e-commerce website, *i.e.*, browsing and search history. The relevant actions are clicks, page views, and searches, collected over a shorter time period of seven days. More detailed information on actions and attributes in the experimental data are shown in Table 1.

The user attribute data is registered customer information such as age and gender. The unique number of users in the dataset is 22.5M, the number of daily purchase and point usage samples is approximately 5M, and the number of short-term data samples is approximately 50M. The data is preprocessed to generate user action sequences.

Target Tasks Our pretrained user model is trained in a general manner and can be adapted to a variety of user understanding tasks. We fine-tune the self-supervised pretrained model to three real-world downstream tasks that aim to improve customer experience. The datasets of the three target tasks are split 80-20 to create training and testing datasets for fine-tuning.

- The **user targeting** task is to identify potential new customers for certain services or products, and it is formulated as a binary classification problem, indicating interest or no interest in a particular service. Users who responded positively to a target service/product, *e.g.*, directly via a purchase or indirectly by clicking on a banner, form the set of positive labeled data, while negative ones are uniformly sampled from the remaining set of users with a 3:1 ratio. A new dataset is collected after the time period of the data used for pretraining.
- The **user attribute prediction** task is predicting different user attributes, *e.g.*, whether or not a customer owns a pet. It is also posed as a classification problem, where ground truth labels are obtained through questionnaires.
- The **next purchase genre prediction** task is a multi-class prediction problem with the aim to predict the next genre of items that a user will purchase on the e-commerce platform. The dataset is created from the one-month user history following the pretraining time period.

Model Baselines UserBERT is compared to direct modeling without pretraining and to multi-task learning (MTL)-based pretraining. The MTL models apply a multi-tower architecture in which each tower encodes one type of user data. For the MTL-based baselines, different types of user

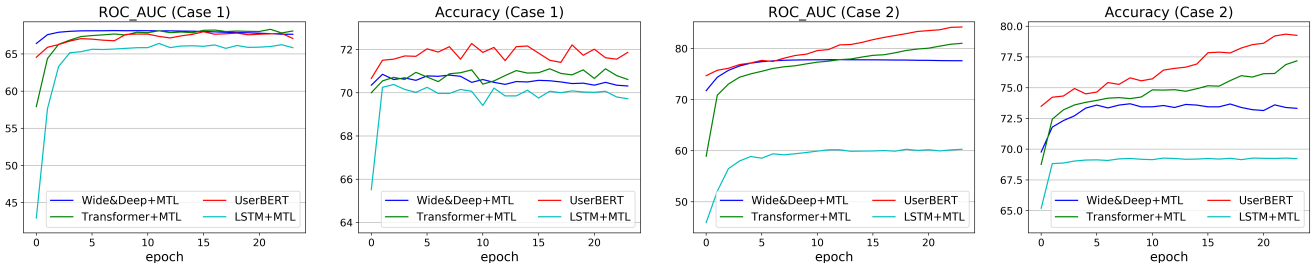

Figure 3: **Model fine-tuning performance comparison for the *user targeting* task.** *The charts show the results for task of predicting whether customers will user either of two different services (case 1 and case 2). For each case we plot the ROC AUC and accuracy metrics vs. the number of training epochs.*

data are passed through corresponding encoders, and the encoded representations are combined at the last layer before connecting to multiple training tasks. The dimension of the combined representations is set to 128 for all MTL models.

We collect user labels across the services in the ecosystem and train MTL models with 12 user attributes as pretraining tasks. These pretraining tasks classify the categories of user activities such as the usage frequency of certain services or attributes like type of occupation. We will not reveal the details of the tasks due to data governance. By learning and sharing across multiple tasks, the yielded user representations are considered to be generalized and applicable for transferring to downstream tasks.

- **Wide&Deep+MTL**: The Wide&Deep model is selected for comparison because it represents a traditionally applied approach for modeling e-commerce data. Although the model cannot directly handle user behavior data as sequences, we generate fixed-length (1130-d) embeddings by aggregating behavior data and inputting them to the deep part of the model (Cheng et al. 2016). Categorical user attribute data is mapped to word embeddings and concatenated before feeding it into the wide part of the model. The wide part is a linear model, while the dimensions of the 4 hidden layers for the deep side are 512, 256, 256 and 128, respectively.
- **LSTM+MTL**: LSTM networks are commonly used to model sequential data (Ni et al. 2018). The same discretization and input generation are applied to long-term and short-term user behavior for this model. It is a 3-tower model, in which two LSTM networks model the two types of user behavior and user attributes are modeled in the same way as the Wide&Deep model. The dimension of the hidden state in all LSTM encoders and the length limitation of both long-term and short-term data are set to 128.
- **Transformer+MTL**: The architecture is the same as the LSTM+MTL model above but with two different Transformer encoders (Vaswani et al. 2017) to model long and short-term user data separately. The length of input user behavior sequence to the encoders is limited to 128 as well. We pretrain the model via minimizing the summed cross entropy loss of the multiple training tasks.
- **UserBERT**: The proposed self-supervised learning based pretraining model, which simultaneously learns from long- and short-term actions and user attributes. Pretraining is done by reconstructing attributes in masked tokens via multi-label classifications.

Table 2: **User targeting task.** Best ROC AUC and Accuracy comparisons after fine-tuning.

	ROC AUC		Accuracy	
Model	**Case 1**	**Case 2**	**Case 1**	**Case 2**
Wide&Deep+MTL	68.14	77.81	70.61	73.67
LSTM+MTL	66.42	60.28	70.38	69.27
Transformer+MTL	**68.31**	81.21	71.11	77.19
UserBERT	67.98	**84.20**	**72.28**	**79.36**

Experimental Setup For UserBERT, we use the same notations as BERT, and set the number of Transformer blocks L to 4, the hidden size H to 128, and the number of self-attention heads A to 4. The input sequence length of both long-term and short-term data is limited to 128. For fair comparison, we pretrain all models using the Adam optimizer with a learning rate of 10^{-4} and batch size of 16. We fine-tune models using the same learning rate and a batch size of 128. Pretraining of 400K batches of the UserBERT model takes approximately 12 hours using our PyTorch implementation, running on two GeForce RTX 2080 Ti GPUs.

For fine-tuning each target task, the combined encoder representations of the MTL-based models are fed to an output layer, while the fine-tuning of UserBERT is done by connecting the hidden representations of the first token to an output layer for each task. After plugging in task-specific inputs and outputs, we fine-tune the parameters of pretrained models end-to-end.

Results

User Targeting We show the results for two different services. The sizes of the datasets are 30,204 and 31,106 samples, respectively. Compared to the size of the pretraining dataset, the use cases of this task only have few labeled samples. Classification performance per epoch in terms of accuracy and ROC AUC are shown in Figure 3. Table 2 compares

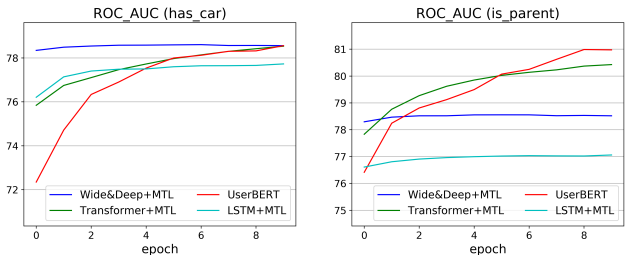

Figure 4: **Model fine-tuning comparison on *user attribute prediction* tasks.** *ROC AUC metric vs. number of training epochs for different models on two different user attribute prediction tasks.*

the best ROC AUC and accuracy results for the same two cases. The LSTM model, which sequentially models user behavior, has relatively low accuracy, indicating that the sequential order of user actions does not provide useful information for this task.

From our experience, the user targeting task focuses on patterns from relatively static user preferences. The Wide&Deep model shows competitive performance, achieving the highest ROC AUC for case 1, which is reasonable since our exploratory analysis indicates that user attributes are important features. The performance of the Transformer-based models reveal that the underlying explanatory factors for this task can be captured by attention networks. UserBERT outperforms other models in terms of accuracy by a substantial margin.

User Attribute Prediction In general, it is challenging to predict user attributes because predictive signals in the behavior data are sparse. In other words, the target user attributes may not be strongly correlated to behavior data. Therefore, this prediction task evaluates the model's ability to discover hidden explanatory factors in the raw data. We show experimental results for two use cases: one is to predict whether a user has a car, while the other one is to predict if a user is a parent. These two tasks are denoted as *has_car* and *is_parent*.

The dataset for the *has_car* task contains 448,501 samples and the one for the *is_parent* task contains 400,268. The classification results of 10-epoch fine-tuning are shown in Figure 4. Table 3 compares the best ROC AUC results in 10 epochs. From the *has_car* results, we observe that the Wide&Deep model shows good initial performance, and during training other models eventually reach similar accuracy. We believe this is due to the fact that user features such as age and location are important features for this task. It seems challenging for models to extract other discriminative patterns from either long-term or short-term user behavior. On the other hand, whether a user is a parent or not seems to present different characteristics in terms of how they behave on an e-commerce or travel booking platform. These patterns can be captured by deep learning models like UserBERT and Transformer-based models. UserBERT is able to match and eventually outperform the baseline models.

Table 3: **User attribute prediction.** Best ROC AUC comparison after fine-tuning.

Model	Has car	Is parent
Wide&Deep+MTL	**78.61**	78.52
LSTM+MTL	77.73	77.06
Transformer+MTL	78.54	80.43
UserBERT	78.56	**80.99**

Table 4: **Next purchase genre prediction.** Best mAP@10 comparison after fine-tuning.

Model	mAP@10(%)
Popular Genres	4.22
Wide&Deep+MTL	7.65
LSTM+MTL	8.13
Transformer+MTL	8.62
UserBERT	**10.90**

Next Purchase Genre Prediction The dataset contains data from 586,130 users, and we fine-tune each pretrained model for 10 epochs. The mean average precision for the top 10 purchased genres (mAP@10) comparison is shown in Table 4. The UserBERT model outperforms baseline models by a large margin. This task requires understanding of both long-term preferences as well as recent interests of customers. Prediction models should be able to identify candidate genres from user habits over a longer time range, and then identify likely ones as prediction results from recent interest trends. More specifically, a model should understand how users typically use services in the ecosystem as well as what they are currently interested in. The architecture of the baseline models learns from different types of user data separately and combines the last-layer representations for training. It fails to sufficiently capture the correlations. In contrast, UserBERT benefits from the unified structure of the user data and captures more accurate correlations, not only within certain types of user behavior, but also between different behavior types via attention networks.

Since it is common that users make purchases from only a subset of genres, we also devised an intuitive but strong baseline that predicts the most popular genres, ranked by the total number of purchases, and compared it against all pretrained models. With an mAP@10 of 4.22%, this model's accuracy is significantly lower, demonstrating the effectiveness of the pretrained models.

Ablation Studies

We perform additional experiments to better understand the effects of certain aspects of the pretraining and fine-tuning framework. More specifically, we analyze the effect of the pretraining step of UserBERT and how the number of labeled samples affects the performance of fully supervised pretraining methods.

Effect of Pretraining We directly apply UserBERT to the user targeting task without pretraining to verify whether it

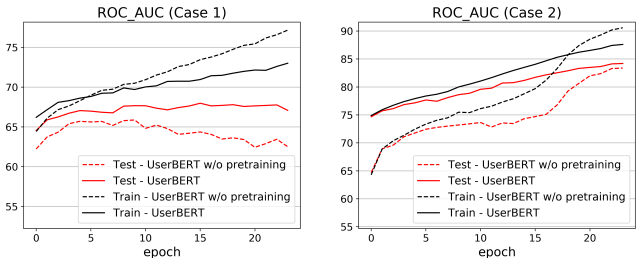

Figure 5: **Effect of pretraining on UserBERT.** *ROC AUC comparison of UserBERT with and without pretraining on the* user targeting *task.*

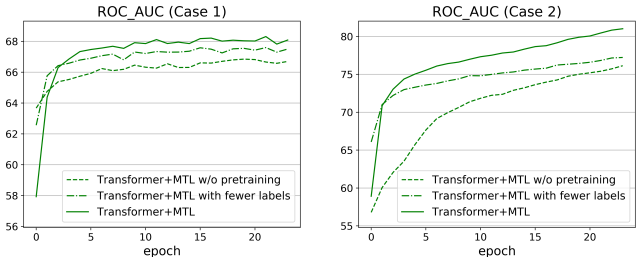

Figure 6: **Effect of pretraining on Transformer+MTL.** Comparison between Transformer-based MTL models with no pretraining, pretraining with 30% of labels, and pretraining with all labels on the *user targeting* tasks.

benefits from the pretraining step. The ROC AUC comparison between UserBERT with and without pretraining is shown in Figure 5. The pretrained models outperform the models trained from scratch significantly. This indicates that the pretraining step extracts useful information that allows fine-tuning to boost performance for downstream tasks. From the error curves during training, we also observe that models tend to overfit quickly without pretraining. The pretrained UserBERT model achieves more generic user representations and yields significant accuracy improvements when adapted to new tasks.

Effect of Additional Pretraining Labels We hypothesize that, compared to Transformer-based MTL, the learning of UserBERT is not limited by the multiple training tasks and is able to learn more expressive and generic representations from the input.

To further demonstrate the advantage of the proposed method over MTL-based pretraining, we pretrain Transformer-based MTL models with different numbers of labels before fine-tuning. We evaluate three training regimes: no pretraining, using 30% of labels and using all labels. The comparison indicates that the performance of MTL is significantly affected by the number of training samples. As shown in Figure 6, more labeled data contributes to performance gains on the user targeting task. Models without the pretraining step show the worst performance.

In contrast, the pretraining of UserBERT does not require any additional collection of supervision signals, and therefore is not impacted by either the quantity or the quality of user annotations.

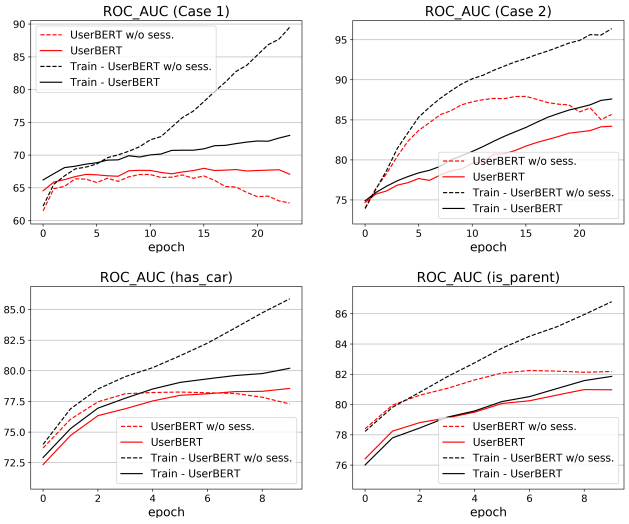

Figure 7: **Effect of discretization of user actions into 'behavioral words'.** The plot shows fine-tuning performance results with and without discretization into sessions ('sess.') on *user targeting* (above) and *user attribute prediction* (below) tasks.

Effect of the Discretization of User Behavior In this study, instead of modeling every single user action, we discretize user action sequence into *behavioral words* for better generalization on downstream tasks. Figure 7 depicts the fine-tuning performance comparisons between pretrained models with and without the discretization of raw user action sequences for the user attribute prediction and user targeting tasks. In terms of ROC AUC comparison, the pretrained model with discretization improve fine-tuning performance on 2 of the 4 cases shown in the figure. Experimental results show that the discretization of user behavior improves next purchase genre prediction on mAP@10 by 2.1%. In addition, the model without the discretization into behavioral words tends to overfit quickly as demonstrated in Figure 7.

Conclusion

This paper introduces a new method to model user behavior by adapting the BERT model, which has made significant improvements in the NLP domain. It explores and demonstrates the possibility for user-oriented machine learning tasks to alleviate the dependency on large annotated datasets. We present UserBERT for pretraining user representations in a self-supervised manner on short-term and long-term behavior as well as user profile data. We provide a novel method to tokenize raw user behavior sequences into behavioral words, which is demonstrated to reduce overfitting during pretraining. Extensive experiments show that a well-designed pretrained model with self-supervision is able to outperform fully supervised learning models when transferred to downstream applications.

References

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8): 1798–1828.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv preprint arXiv:2002.05709*.

Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; Anil, R.; Haque, Z.; Hong, L.; Jain, V.; Liu, X.; and Shah, H. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 7–10.

Covington, P.; Adams, J.; and Sargin, E. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 191–198.

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised Visual Representation Learning by Context Prediction. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, 1422–1430.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *Proceedings of the 31st International Conference on Machine Learning*, 647–655.

Ghadiyaram, D.; Feiszli, M.; Tran, D.; Yan, X.; Wang, H.; and Mahajan, D. 2019. Large-Scale Weakly-Supervised Pre-Training for Video Action Recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12038–12047.

Gong, H.; Zhao, Q.; Li, T.; Cho, D.; and Nguyen, D. 2020. Learning to Profile: User Meta-Profile Network for Few-Shot Learning. In *CIKM*, 2469–2476.

Grbovic, M.; and Cheng, H. 2018. Real-Time Personalization Using Embeddings for Search Ranking at Airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 311–320.

Guo, M.; Haque, A.; Huang, D.-A.; Yeung, S.; and Fei-Fei, L. 2018. Dynamic Task Prioritization for Multitask Learning. In *ECCV*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 1026–1034.

Hendrycks, D.; Lee, K.; and Mazeika, M. 2019. Using Pre-Training Can Improve Model Robustness and Uncertainty. In *ICML*.

Hjelm, D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2019. Learning deep representations by mutual information estimation and maximization. In *ICLR*.

Jacob, D.; Ming-Wei, C.; Kenton, L.; and Kristina, T. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7482–7491.

Kolesnikov, A.; Zhai, X.; and Beyer, L. 2019. Revisiting Self-Supervised Visual Representation Learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1920–1929.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *ICLR*.

Li, S.; and Zhao, H. 2020. A Survey on Representation Learning for User Modeling. In *IJCAI*.

Lin, Z.; Feng, M.; Santos, C. D.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A Structured Self-attentive Sentence Embedding. In *International Conference on Learning Representations*.

Liu, X.; Gao, J.; He, X.; Deng, L.; Duh, K.; and Wang, Y.-Y. 2015. Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 912–921.

Liu, X.; He, P.; Chen, W.; and Gao, J. 2019. Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4487–4496.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, 13–23.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 3111–3119.

Ni, Y.; Ou, D.; Liu, S.; Li, X.; Ou, W.; Zeng, A.; and Si, L. 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.

Noroozi, M.; and Favaro, P. 2016. Unsupervised Learning of Visual Represention by solving Jigsaw Puzzles. In *ECCV*.

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J. 2014. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1717–1724.

Pathak, D.; Krähenbühl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context Encoders: Feature Learning by Inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2536–2544.

Pei, C.; Zhang, Y.; Zhang, Y.; Sun, F.; Lin, X.; Sun, H.; Wu, J.; Jiang, P.; Ge, J.; Ou, W.; and Pei, D. 2019. Personalized Re-Ranking for Recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 3–11.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Peters, M.; Ruder, S.; and Smith, N. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP*, 7–14.

Qi, D.; Su, L.; Song, J.; Cui, E.; Bharti, T.; and Sacheti, A. 2020. ImageBERT: Cross-modal Pre-training with Large-scale Weak-supervised Image-Text Data. *CoRR*, abs/2001.07966.

Richardson, M.; Dominowska, E.; and Ragno, R. 2007. Predicting Clicks: Estimating the Click-Through Rate for New Ads. In *Proceedings of the 16th International World Wide Web Conference(WWW-2007)*.

Ruder, S. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. *ArXiv*, abs/1706.05098.

Sermanet, P.; Lynch, C.; Chebotar, Y.; Hsu, J.; Jang, E.; Schaal, S.; and Levine, S. 2018. Time-Contrastive Networks: Self-Supervised Learning from Video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1134–1141.

Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; and Carlsson, S. 2014. CNN features off-the-shelf: an astounding baseline for recognition. In *CVPR DeepVision workshop*.

Simo-Serra, E.; Trulls, E.; Ferraz, L.; Kokkinos, I.; Fua, P.; and Moreno-Noguer, F. 2015. Discriminative Learning of Deep Convolutional Feature Point Descriptors. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 118–126.

Standley, T. S.; Zamir, A. R.; Chen, D.; Guibas, L. J.; Malik, J.; and Savarese, S. 2019. Which Tasks Should Be Learned Together in Multi-task Learning? *ArXiv*, abs/1905.07553.

Su, W.; Zhu, X.; Cao, Y.; Li, B.; Lu, L.; Wei, F.; and Dai, J. 2020. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *International Conference on Learning Representations*.

Sun, C.; Myers, A.; Vondrick, C.; Murphy, K.; and Schmid, C. 2019a. VideoBERT: A Joint Model for Video and Language Representation Learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 7463–7472.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019b. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1441–1450.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

Wu, C.; Wu, F.; Qi, T.; Lian, J.; Huang, Y.; and Xie, X. 2020. PTUM: Pre-training User Model from Unlabeled User Behaviors via Self-supervision. In *Findings of the Association for Computational Linguistics: EMNLP*, 1939–1944.

Wu, J.; Wang, X.; and Wang, W. Y. 2019. Self-Supervised Dialogue Learning. In *ACL*, 3857–3867.

Xie, X.; Sun, F.; Liu, Z.; Gao, J.; Ding, B.; and Cui, B. 2020. Contrastive Pre-training for Sequential Recommendation. *ArXiv*, abs/2010.14395.

Yuan, F.; He, X.; Karatzoglou, A.; and Zhang, L. 2020. Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1469–1478.

Zhang, Y.; and Yang, Q. 2017. A Survey on Multi-Task Learning. *ArXiv*, abs/1707.08114.

Zhou, G.; Mou, N.; Fan, Y.; Pi, Q.; Bian, W.; Zhou, C.; Zhu, X.; and Gai, K. 2019. Deep Interest Evolution Network for Click-Through Rate Prediction. In *AAAI*.

Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and Gai, K. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1059–1068.

Zhu, Z. A.; Chen, W.; Minka, T.; Zhu, C.; and Chen, Z. 2010. A Novel Click Model and Its Applications to Online Advertising. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, 321–330.