UserBERT: Pre-training User Model with Contrastive Self-supervision

Chuhan Wu
Department of Electronic Engineering,
Tsinghua University
Beijing, China
wuchuhan15@gmail.com

Fangzhao Wu*
Microsoft Research Asia
Beijing, China
wufangzhao@gmail.com

Tao Qi
Department of Electronic Engineering,
Tsinghua University
Beijing, China
taoqi.qt@gmail.com

Yongfeng Huang
Department of Electronic Engineering,
Tsinghua University
Beijing, China
yfhuang@tsinghua.edu.cn

ABSTRACT

User modeling is critical for personalization. Existing methods usually train user models from task-specific labeled data, which may be insufficient. In fact, there are usually abundant unlabeled user behavior data that encode rich universal user information, and pre-training user models on them can empower user modeling in many downstream tasks. In this paper, we propose a user model pre-training method named UserBERT to learn universal user models on unlabeled user behavior data with two contrastive self-supervision tasks. The first one is masked behavior prediction and discrimination, aiming to model the contexts of user behaviors. The second one is behavior sequence matching, aiming to capture user interest stable in different periods. Besides, we propose a medium-hard negative sampling framework to select informative negative samples for better contrastive pre-training. Extensive experiments validate the effectiveness of UserBERT in user model pre-training.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Personalization**; • **Computing methodologies** → **Transfer learning**.

KEYWORDS

User model, Pre-training, Contrastive learning

ACM Reference Format:
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 (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531810

*The corresponding author.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531810

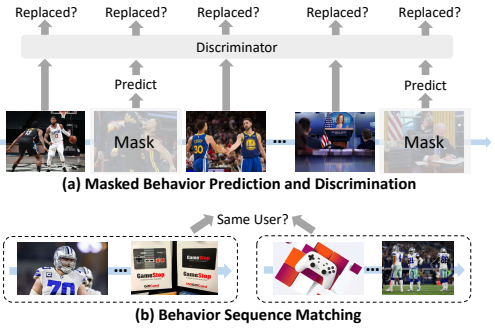


Figure 1: The two self-supervision tasks in UserBERT.

1 INTRODUCTION

User modeling is a core technique for various personalized web and smart city applications such as news [16, 24, 28] and item recommendation [12, 19]. Many existing user modeling methods learn user models based on labeled data in downstream tasks [25, 29, 31, 39, 40]. For example, Wu et al. [26] proposed a hierarchical user representation model for user demographic prediction from user search queries. An et al. [2] proposed to use an attentive multi-view learning framework for native Ad click-through rate prediction based on their search query and webpage browsing behaviors. These methods usually require a large amount of task-specific labeled data to train accurate user models [38]. However, in many scenarios labeled training data is insufficient and difficult to collect or annotate [26]. Fortunately, there are rich unlabeled user behaviors that encode rich information of user interest and characteristics [8, 18, 34]. Mining unlabeled user behavior data has the potential to generate a universal understanding of users to empower user modeling in downstream tasks [30].

Motivated by the great success of pre-trained language models in NLP, pre-training user models on unlabeled user behaviors may improve user modeling for personalization. In many language model pre-training methods like BERT [6], a masked token prediction task is used to capture the contexts of words [7, 15]. In addition, researchers found that using a discriminator to predict the replacement of tokens can effectively improve model pretraining [5]. In a

similar way, predicting the masked user behavior in a user behavior sequence meanwhile detecting behavior replacements may help model the contexts of user behaviors. As shown in Fig. 1(a), by predicting the masked user behaviors and discriminate replacements, the model can capture the relations between user behaviors and further help infer the user's interest (e.g., the Warriors team). In addition, motivated by the sentence pair prediction task in language model pre-training for enhancing sentence modeling [4, 20], matching the user behavior sequences may also be helpful for capturing user interests. As shown in Fig. 1(b), by matching the two behavior sequences from the same user in different periods, the model can better capture user interests (e.g., NFL) that are stable over time.

In this paper, we propose a method named UserBERT for pre-training user models in two contrastive self-supervision tasks to enhance user modeling. The first one is masked behavior prediction and discrimination (MBPD), which aims to model the relatedness between user behaviors. We use an encoder to predict masked behaviors, and use a discriminator to judge whether each behavior is the original one. The second one is behavior sequence matching (BSM), which aims to capture the inherent user interests that are consistent over time. The model needs to identify whether two behavior sequences in different time periods come from the same user. Negative sampling is important for contrastive learning [3, 9, 33], while random negatives [17] may not be informative and globally hardest negatives [35] may be too confusing. Thus, we propose a medium-hard negative sampling framework to select locally hardest samples for contrastive learning. We randomly construct a candidate behavior pool and a behavior sequence pool from the full candidate sets, and select the candidates in the pools with top similarities to the targets as negative samples. We synchronously update the candidate behavior pool in each iteration, but asynchronously update the candidate behavior sequence pool to reduce computational costs. Extensive experiments on two real-world datasets show that UserBERT consistently improves various user models and outperforms many user model pre-training methods.

2 METHODOLOGY

2.1 General User Model Framework

A general user model framework is shown in Fig. 2. It has a hierarchical architecture with a behavior encoder and a user encoder. The behavior encoder transforms each user behavior and its position in the behavior sequence into its embedding. It can be implemented by various models according to behavior types, such as ID embedding table [19] for ID-based behaviors, or CNN [13] and Transformer [20] for behaviors with texts. The user encoder takes the behavior embedding sequence as input to learn a user embedding. It contains a behavior context encoder to learn hidden behavior representations by capturing behavior context, and a behavior aggregator to summarize the hidden behavior representations into a unified user embedding. The behavior context encoder can be implemented by many models like CNN [26], LSTM [2] and self-attention [23] networks, and the behavior aggregator can be a max pooling, average pooling, attentive pooling [37] or last pooling module [10]. By pre-training the user model with unlabeled user behaviors via self-supervision, the model can exploit the universal user information conveyed by user behaviors to empower downstream tasks.

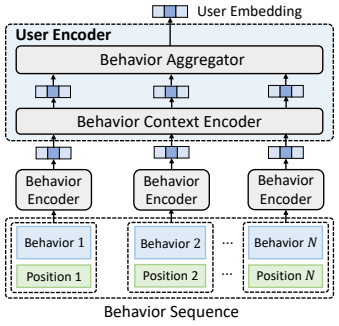


Figure 2: A general framework of the user model.

2.2 Contrastive Pre-training Tasks

Next, we introduce the two self-supervision tasks for contrastive user model pre-training, i.e., masked behavior prediction and discrimination (MBPD) and behavior sequence matching (BSM). Motivated by the success of ELECTRA [5] in language model pre-training, we propose a MBPD task to capture the relations between behaviors of the same user, as shown in Fig. 3(a). We randomly mask 15% of user behaviors (and at least one) in a user behavior sequence, and pre-train the model in a contrastive way. More specifically, we first use the behavior encoder to process the masked behavior sequence, and then use a behavior context encoder as the generator to learn hidden behavior representations for inferring the masked behaviors. For each masked behavior x (regarded as the positive candidate) we sample K negative candidate behaviors, and we use a matcher to jointly predict the matching scores of these $K+1$ candidates based on their relevance to the hidden representation at the masking position. These scores are further normalized by softmax, and the task is formulated as a $K+1$-way classification problem. We use the cross-entropy loss as follows:

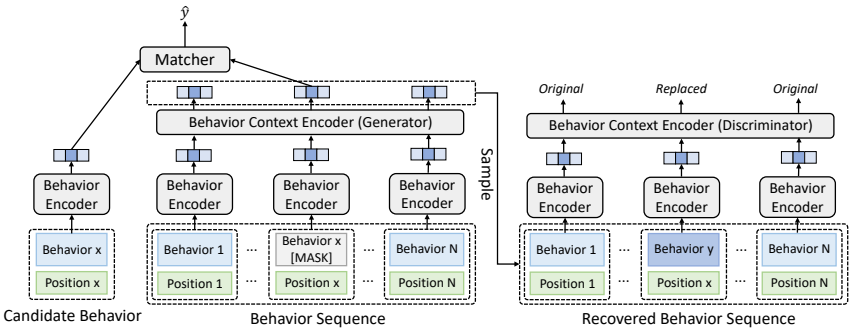
$$\mathcal{L}^{P}_{MBPD} = -\sum_{x\in\mathcal{M}}\sum_{i=1}^{K+1} y_i^x \log(\hat{y}_i^x), \tag{1}$$

where y_i^x and $\hat{y}_i^x$ are gold and predicted labels of the i-th candidate behavior given the masked behavior x, and $\mathcal{M}$ is the set of masked behaviors. We sample the candidate behaviors with the highest probability scores as the predicted masked behaviors, and they are combined with other non-masked behaviors to form a recovered behavior sequence. We use the behavior encoder to obtain behavior representations, and apply another behavior context encoder that serves as a discriminator to predict whether each behavior is the original one or has been replaced in the masked behavior prediction procedure. We use the binary crossentropy loss to train the discriminator, which is formulated as follows:

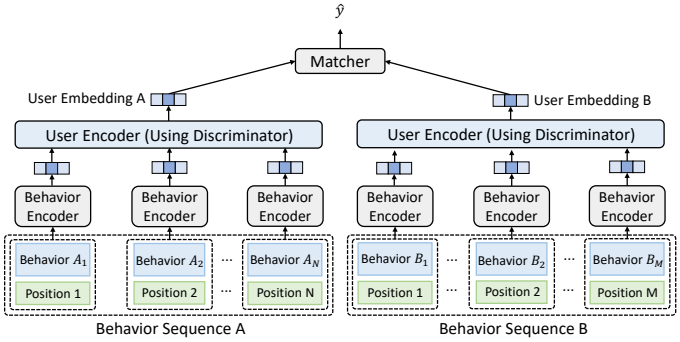
$$\mathcal{L}^{D}_{MBPD} = -\sum_{x\in\mathcal{S}} z^x \log(\hat{z}^x) + (1-z^x)[1-\log(\hat{z}^x)], \tag{2}$$

where z^x and $\hat{z}^x$ are real and predicted replacement labels of the behavior x in the behavior sequence $\mathcal{S}$. Note that the behavior encoder and generator are not tuned by this loss.

The second pre-training task is behavior sequence matching (BSM). As shown in Fig. 3(b), the goal of this task is to identify whether two behavior sequences A and B come from the same user.



(a) Masked Behavior Prediction and Discrimination (MBPD) task.



(b) Behavior Sequence Matching (BSM) task.

Figure 3: Frameworks of the two self-supervision tasks for user model pre-training.

To enforce the model to capture the stable user interests across different time periods, we ensure the behavior sequences A and B have no overlap in time. We use the user model to encode both sequences and evaluate their similarity via a matcher. Motivated by ELECTRA [5], the user encoder incorporates the discriminator behavior context encoder rather than the generator. For each pair of behavior sequences from the same user (sequence B is the positive candidate), we sample P negative candidate behavior sequences from other users that do not overlap with sequence A in time. We jointly predict the matching scores of the $P+1$ behavior sequences and normalize them via softmax. The BSM loss is as follows:

$$\mathcal{L}_{BSM} = -\sum_{A \in \mathcal{S}} \sum_{i=1}^{P+1} y_i^A \log(\hat{y}_i^A), \tag{3}$$

where y_i^A and $\hat{y}_i^A$ are the gold label and predicted score of the i-th candidate behavior sequence for A, and $\mathcal{S}$ is the set of user behavior sequences for model pre-training. We jointly pre-train the user model in both MBPD and BSM tasks. The generator is trained only by the $\mathcal{L}_{MBPD}^P$ loss, while the discriminator is optimized by the summation of the two losses, i.e., $\mathcal{L}_{MBPD}^D + \mathcal{L}_{BSM}$. The behavior encoder is optimized by the loss combination $\mathcal{L}_{MBPD}^P + \mathcal{L}_{BSM}$.

2.3 Medium-Hard Negative Sampling

Finally, we introduce our proposed medium-hard negative sampling method to select the locally hardest negative candidate behaviors and candidate behavior sequences for contrastive pre-training. Negatives uniformly selected from the full behavior set may be uninformative [11], while globally hardest negatives can also be noisy [36]. Thus, we propose to construct a smaller candidate behavior pool and select the locally hardest samples from this pool as "medium-hard" negatives. We assume that the entire behavior set $\mathcal{B}$ contains M user behaviors, and we randomly sample m ($m << M$) behaviors to construct a candidate behavior pool. This pool is synchronously updated during the model training, i.e., re-sampling user behaviors from $\mathcal{B}$ at each model training step. We denote a masked behavior at the i-th model update step as x_i and the m behaviors in the corresponding candidate behavior pool as $[b_1, ..., b_m]$. We use the behavior encoder in the user model to encode these behaviors into their hidden representations, which are respectively denoted as $\mathbf{h}_i^x$ and $\mathbf{H}^b = [\mathbf{h}_1^b, ..., \mathbf{h}_m^b]$. We evaluate the cosine similarities between $\mathbf{h}_i^x$ and each behavior embedding in $\mathbf{H}^b$, and select K behaviors with top similarity scores as negative candidates for masked behavior prediction (candidates that are identical to the target are filtered). In this way, the negatives are harder if m is larger. We can adjust the difficulties of negative samples by tuning the value of m.

To select negative candidates for the BSM task, we can similarly choose the locally hardest samples in a candidate behavior sequence pool. However, encoding behavior sequences is much more time-consuming than processing user behaviors. To reduce the computational cost, we construct a behavior sequence pool

that is asynchronously updated after every certain number of iterations. We denote the target user behavior sequence from the i-th to the j-th training steps as $[A_i, ..., A_j]$. We randomly crop u behavior sequences from the behavior sequences of other users in the full user set to form the candidate behavior sequence pool, which is denoted as $[B_1, ..., B_u]$. We ensure that these sequences do not have overlaps with $[A_i, ..., A_j]$ in time. This pool is static between the i-th and j-th training steps (the interval is $j - i + 1$ steps). We use the user model at the i-th training step to encode target and candidate behavior sequences, which are respectively denoted as $\mathbf{R}^A = [\mathbf{r}_i^A, ..., \mathbf{r}_j^A]$ and $\mathbf{R}^B = [\mathbf{r}_1^B, ..., \mathbf{r}_u^B]$. We use cosine distance to measure the similarity between each target behavior sequence embedding in $\mathbf{R}^A$ and each candidate in $\mathbf{R}^B$, and select the candidates with the highest similarity scores as negative samples. We can obtain medium-hard candidate user behavior sequences efficiently by using a moderate pool size and update interval.

3 EXPERIMENTS

3.1 Datasets and Experimental Settings

In our experiments, we conduct experiments in two tasks. The first task is news recommendation. We take the dataset (denoted as *News*) used in [22]. It contains news click behaviors of 10,000 users on Microsoft News and webpages browsing behaviors on Bing in one month. The task is to predict future news clicks of users based on the titles of their browsed webpages. The second task is Ad CTR prediction. The dataset (denoted as *Ads*) contains Ad titles and descriptions, Ad impressions, and the webpage browsing behaviors of users in one month, which is used by [27, 30]. The task is to predict whether a user clicks a candidate Ad based on Ad texts and the titles of this user's browsed webpages. We used an additional unlabeled user behavior dataset for user model pre-training, which contains the titles of browsed webpages of 500,000 users on Bing in six months. The dataset statistics are summarized in Table 1.

Table 1: Statistics of the two datasets.

News			
# Users	10,000	# News	42,255
# Impressions	360,428	# Clicked samples	503,698
Ads			
# Users	374,584	# Ads	4,159
# Impressions	400,000	# Clicked samples	364,281
# Users for pre-training	500,000	# Behaviors for pre-training	63,178,293

In our experiments, for fair comparison we followed the base user models settings in PTUM [30]. The number of negative candidates was 4. The sizes of candidate behavior and behavior sequence pools were 1,000 and 100, respectively. The behavior sequence pool update interval was 50 steps. We used Adam [14] as the optimizer. The learning rate was 1e-4. These hyperparameters were tuned according to the validation performance. We used AUC and nDCG@10 on *News*, while use AUC and AP on *Ads* as metrics. We repeated each experiment 5 times and reported the average results.

3.2 Performance Evaluation

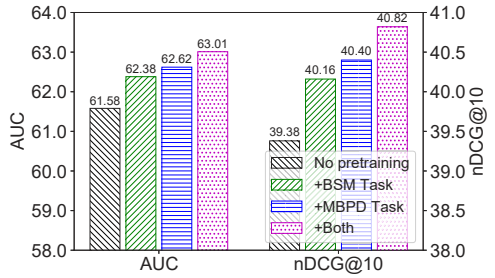
We compare the performance of our approach with several baseline methods, including (a) w/o pre-training; (b) PTUM [30] a user model pre-training method based on masked behavior prediction and next K behaviors prediction; (c) CL4SRec [34] a contrastive user model pre-training method by matching augmented user behavior sequences; SUMN [8], a user model pre-training method using a word distribution consistency regularization. On the *News* dataset we use NAML [21], LSTUR [2], and NRMS [23], which are widely used news recommendation methods [32]. on the *Ads* dataset we use GRU4Rec [10], NativeCTR [1] and BERT4Rec [19] to implement the user models. The performance under different ratios of training data on the two datasets is respectively shown in Tables 2 and 3 (here we report AUC only due to space limit). We find pre-trained user models consistently outperform the models without pre-training, and the advantage is larger when less labeled data is used for training. This is because pre-trained user models can exploit universal user information encoded by unlabeled user behaviors, which can reduce the dependency of user models on labeled data. Second, PTUM and UserBERT outperform CL4SRec and SUMN. This is because in CL4SRec the behavior sequences augmented from the same user may have many overlaps, and the model may capture overlap patterns instead of inherent user interests. In addition, the word distributions of user behaviors can

Table 2: AUC under different ratios of data on *News*.

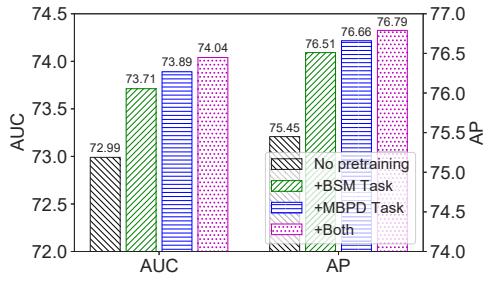
Methods	10%	25%	100%
NAML	58.83±0.27	60.20±0.23	61.02±0.20
NAML+CL4SRec	59.68±0.25	60.90±0.21	61.59±0.18
NAML+PTUM	60.22±0.23	61.53±0.20	61.94±0.17
NAML+SUMN	59.42±0.26	60.54±0.24	61.30±0.21
NAML+UserBERT	61.12±0.24	62.20±0.20	62.69±0.17
LSTUR	59.31±0.25	60.60±0.20	61.67±0.18
LSTUR+CL4SRec	60.12±0.24	61.28±0.18	62.05±0.16
LSTUR+PTUM	60.61±0.22	61.75±0.17	62.48±0.15
LSTUR+SUMN	59.82±0.27	61.02±0.22	61.96±0.19
LSTUR+UserBERT	61.50±0.22	62.53±0.19	63.12±0.16
NRMS	59.22±0.20	60.53±0.18	61.58±0.16
NRMS+CL4SRec	60.05±0.21	61.26±0.17	61.95±0.15
NRMS+PTUM	60.58±0.18	61.71±0.16	62.32±0.13
NRMS+SUMN	59.77±0.21	60.90±0.19	61.87±0.17
NRMS+UserBERT	61.43±0.17	62.49±0.14	63.01±0.13

Table 3: AUC under different ratios of data on *Ads*.

Methods	10%	25%	100%
GRU4Rec	70.96±0.11	71.51±0.09	72.20±0.06
GRU4Rec+CL4SRec	71.45±0.12	71.94±0.10	72.68±0.09
GRU4Rec+PTUM	71.94±0.13	72.37±0.12	72.79±0.10
GRU4Rec+SUMN	71.30±0.13	71.78±0.12	72.43±0.10
GRU4Rec+UserBERT	72.82±0.11	72.99±0.10	73.31±0.08
NativeCTR	71.13±0.10	71.69±0.08	72.35±0.08
NativeCTR+CL4SRec	71.81±0.09	72.20±0.08	72.77±0.07
NativeCTR+PTUM	72.19±0.09	72.58±0.07	72.91±0.06
NativeCTR+SUMN	71.65±0.11	71.89±0.10	72.53±0.09
NativeCTR+UserBERT	73.03±0.09	73.25±0.07	73.46±0.08
BERT4Rec	71.25±0.10	71.89±0.07	72.99±0.06
BERT4Rec+CL4SRec	72.03±0.08	72.65±0.06	73.39±0.06
BERT4Rec+PTUM	72.30±0.08	72.89±0.07	73.59±0.05
BERT4Rec+SUMN	71.90±0.08	72.22±0.08	72.73±0.07
BERT4Rec+UserBERT	73.22±0.08	73.49±0.08	74.04±0.05

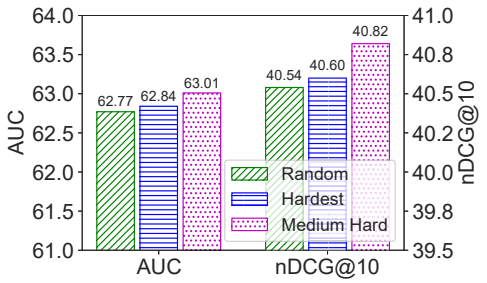


(a) *News*.

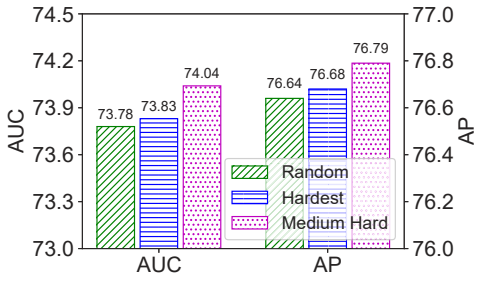


(b) *Ads*.

Figure 4: Effect of the two pre-training tasks.



(a) *News*.



(b) *Ads*.

Figure 5: Effect of medium-hard negative sampling.

be very sparse, and it is difficult for SUMN to capture word distribution consistency across different time periods. Third, UserBERT consistently outperforms PTUM and the improvement is significant ($p < 0.05$ in t-tests). This is because PTUM only predicts specific behaviors and may be disturbed by the randomness of user behaviors. UserBERT uses both MBPD and BSM tasks to capture both behavior contexts and intrinsic user interests, which may be more robust to the noisy behaviors. In addition, UserBERT is trained with more informative negatives than other methods, which yields better performance.

3.3 Influence of Pre-training Tasks

We also study the influence of the self-supervision tasks for user model pre-training. We first verify the effectiveness of each task used in UserBERT. We compare the results of NRMS on the *News* dataset and BERT4Rec on the *Ads* dataset without pre-training or pre-trained with different tasks (same base user models are used in the rest experiments). From the results shown in Fig. 4, we find that both BSM and MBPD tasks are very useful for model pre-training. This is because the MBPD task encourages the user model to capture the relatedness between behaviors and the BSM task helps to model the inherent user interests that are consistent across different time periods. In addition, combining both tasks is better than using a single one. It shows that the two tasks can provide complementary information for each other to improve user model pre-training.

3.4 Effectiveness of Medium-Hard Negatives

We then verify the effectiveness of our proposed medium-hard negative sampling framework. We compare the model performance using random, globally hardest and medium-hard negative candidates, as shown in Fig. 5. From the results, we find that using globally hardest negatives is slightly better than using random negatives on the *News* dataset. However, the performance is worse than using random ones on the *Ads* dataset. This may be because the globally hardest negatives may be too difficult for the model to distinguish and may even be misleading. Our proposed medium-hard negative sampling method outperforms random and hardest negative sampling. It shows that medium-hard negative candidates are more suitable for user model pre-training.

4 CONCLUSION

In this paper, we propose a UserBERT approach that pre-trains user models in two contrastive self-supervision tasks, i.e., a masked behavior prediction and discrimination task to capture contexts of user behaviors and a behavior sequence matching task to capture inherent user interests. In addition, we propose a medium-hard negative sampling framework to select informative negative samples to learn accurate and discriminative user models. Extensive experiments on two datasets show UserBERT can consistently improve various user models and outperform many baseline methods.

ACKNOWLEDGMENTS

This work was supported by National Key Research and Development Project of China under Grant number 2018YFB2101501, and the National Natural Science Foundation of China under Grant numbers U1936216 and U1936208.

REFERENCES

[1] Mingxiao An, Fangzhao Wu, Heyuan Wang, Tao Di, Jianqiang Huang, and Xing Xie. 2019. Neural CTR Prediction for Native Ad. In *CCL*. Springer, 600–612.

[2] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural News Recommendation with Long-and Short-term User Representations. In *ACL*. 336–345.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. PMLR, 1597–1607.

[4] Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *NAACL-HLT*.

[5] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *ICLR*.

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

[7] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NIPS*. 13063–13075.

[8] Jie Gu, Feng Wang, Qinghui Sun, Zhiquan Ye, Xiaoxiao Xu, Jingmin Chen, and Jun Zhang. 2021. Exploiting Behavioral Consistence for Universal User Representation. In *AAAI*, Vol. 35. 4063–4071.

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 9729–9738.

[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR*.

[11] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *arXiv preprint arXiv:2010.01028* (2020).

[12] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. IEEE, 197–206.

[13] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*. 1746–1751.

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

[15] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[16] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *KDD*. 1933–1942.

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

[18] Zhaopeng Qiu, Xian Wu, Jingyue Gao, and Wei Fan. 2021. U-BERT: Pre-training user representations for improved recommendation. In *AAAI*. 1–8.

[19] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*. 1441–1450.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.

[21] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Attentive Multi-View Learning. In *IJCAI*. 3863–3869.

[22] Chuhan Wu, Fangzhao Wu, Mingxiao An, Tao Qi, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with heterogeneous user behavior. In *EMNLP-IJCNLP*. 4876–4885.

[23] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Multi-Head Self-Attention. In *EMNLP-IJCNLP*. 6390–6395.

[24] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2021. User-as-Graph: User Modeling with Heterogeneous Graph Pooling for News Recommendation. In *IJCAI*. 1624–1630.

[25] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2022. Personalized news recommendation: Methods and Challenges. *TOIS* (2022).

[26] Chuhan Wu, Fangzhao Wu, Junxin Liu, Shaojian He, Yongfeng Huang, and Xing Xie. 2019. Neural Demographic Prediction using Search Query. In *WSDM*. 654–662.

[27] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2022. FedCTR: Federated Native Ad CTR Prediction with Cross Platform User Behavior Data. *TIST* (2022).

[28] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. User Modeling with Click Preference and Reading Satisfaction for News Recommendation. In *IJCAI*. 3023–3029.

[29] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering News Recommendation with Pre-trained Language Models. In *SIGIR*. ACM, 1652–1656.

[30] 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. In *EMNLP: Findings*. 1939–1944.

[31] Chuhan Wu, Fangzhao Wu, Xiting Wang, Yongfeng Huang, and Xing Xie. 2021. FairRec:Fairness-aware News Recommendation with Decomposed Adversarial Learning. In *AAAI*. 4462–4469.

[32] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *ACL*. 3597–3606.

[33] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. 2017. Joint detection and identification feature learning for person search. In *CVPR*. 3415–3424.

[34] Xu Xie, Fei Sun, Zhaoyang Liu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive Pre-training for Sequential Recommendation. *arXiv preprint arXiv:2010.14395* (2020).

[35] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[36] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. 2020. Hard negative examples are hard, but useful. In *ECCV*. Springer, 126–142.

[37] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL-HLT*. 1480–1489.

[38] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *SIGIR*. 1469–1478.

[39] Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. 2021. UNBERT: User-News Matching BERT for News Recommendation. In *IJCAI*. 3356–3362.

[40] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *AAAI*, Vol. 33. 5941–5948.