Scaling Law for Recommendation Models: Towards General-Purpose User Representations

Kyuyong Shin[12*], Hanock Kwak[1*], Su Young Kim[1], Max Nihlén Ramström[1], Jisu Jeong[12], Jung-Woo Ha[12], Kyung-Min Kim[12]

[1]NAVER
[2]NAVER AI Lab
{ky.shin, hanock.kwak2}@navercorp.com

Abstract

Recent advancement of large-scale pretrained models such as BERT, GPT-3, CLIP, and Gopher, has shown astonishing achievements across various task domains. Unlike vision recognition and language models, studies on general-purpose user representation at scale still remain underexplored. Here we explore the possibility of general-purpose user representation learning by training a universal user encoder at large scales. We demonstrate that the scaling law is present in user representation learning areas, where the training error scales as a power-law with the amount of computation. Our **C**ontrastive **L**earning **U**ser **E**ncoder (CLUE), optimizes task-agnostic objectives, and the resulting user embeddings stretch our expectation of what is possible to do in various downstream tasks. CLUE also shows great transferability to other domains and companies, as performances on an online experiment shows significant improvements in Click-Through-Rate (CTR). Furthermore, we also investigate how the model performance is influenced by the scale factors, such as training data size, model capacity, sequence length, and batch size. Finally, we discuss the broader impacts of CLUE in general.

Introduction

Recent work has demonstrated that models pretrained on enormous data at scale can perform remarkable downstream transfers in a flexible and task-agnostic manner for vision recognition (Zhai et al. 2021; Dai et al. 2021; Goyal et al. 2021), language models (Devlin et al. 2019; Brown et al. 2020; Kim et al. 2021; Rae et al. 2021), speech recognition (Baevski et al. 2020), and multimodal learning (Radford et al. 2021; Akbari et al. 2021; Wang et al. 2022). These large-scale models are called *foundation models*, which have brought seismic changes to both academia and industry by providing general-purpose utilities and promising results (Bommasani, Hudson et al. 2021). However, in contrast to the other domains mentioned above, studies on the scaling and generalization ability of general-purpose pretrained user representation learning still remain underexplored for various downstream recommendation tasks.

Here we highlight five key questions for exploring the possibilities of general-purpose user representations. 1) Can general-purpose user representations learned from multiple source data provide a promising transfer learning capability? 2) Are pretraining and downstream task performances positively correlated? 3) How various tasks can the pretrained user representations address? 4) Does scaling up pretraining models improve the generalization performance? 5) If so, which factors, such as, training data and model size, behavior sequence length, and batch size, should be scaled up?

To answer these questions, we introduce **C**ontrastive **L**earning **U**ser **E**ncoder, i.e, CLUE. CLUE demonstrates the effectiveness of the pretrained general user representations, learned from 50 billion behavior tokens of 11 million users from a search engine and e-commerce platform that share a common user pool, by adapting them to unseen downstream tasks. CLUE uses contrastive learning by constructing user representations for each service task, then treating pairs of the same user representations as positive samples while treating representations of different users as negative samples. We comprehensively evaluate the pretrained user representation of CLUE with multiple downstream tasks from industrial and benchmark datasets, including an online CTR evaluation. More specifically, we compare the performance of a simple multi-layer perceptron (MLP) employing our task-agnostic pretrained CLUE features with a task-specific model trained for each downstream task. Furthermore, we investigate the empirical scaling laws of training data size, model size, sequence length and batch size with extensive experiments, and analyze power-law scaling for training performance as a function of computing resources.

Our key findings through CLUE are summarized:

Empirical scaling law. The pretraining test error scales as a power-law with the total training cost (PF-days) unless bottlenecked by other factors, which is also observed in other domains (Brown et al. 2020). However, we observe that the performance of the models does not depend on their size alone in user representation tasks (Figure 2(a)), other factors such as batch size and sequence length also have considerable influences on the performance.

Transferability improves as the pretraining error decrease. CLUE performs transfer learning well on heterogeneous datasets even when the domains are different. The resulting test losses on the downstream tasks also show strong correlation to the pretraining test loss. These results indicate that generalization on various data distributions are strongly

*These authors contributed equally.

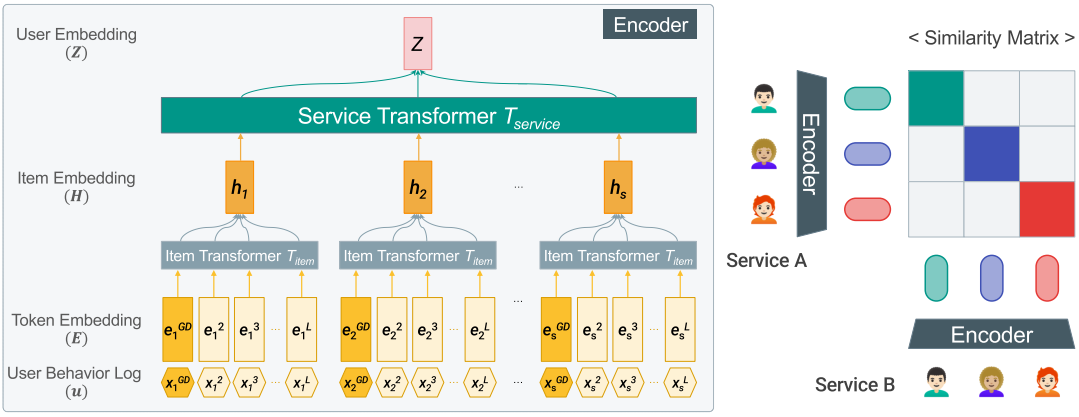


Figure 1: Overall flow of CLUE. The encoder encodes user behavior logs, a sequence of items described in natural language text (u). This description becomes token embeddings (E) and then passed through the Item Transformer followed by mean pooling (H). The sequence of item embedding vectors are then passed to the Service Transformer. The user embedding (Z) is the mean of the last hidden vectors of the Service Transformer. Finally, the user embeddings are contrastively pretrained.

dependent on the pretraining error (Figure 4-Right).

Transforming tabular data to natural language text provides common semantic representation. We transform all data into natural language texts by extracting textual information from tabular data (e.g., product descriptions from product data table). This policy alleviates the discrepancies in data format within different services; the data format of the same product varies depending on the platform, but the product name is still the same. We show that CLUE trained on data from a particular company can provide meaningful user representations for another company.

Advantages of training from multiple service logs. CLUE learns a multi-modal user embedding space from two services, and shows promising results on diverse downstream tasks. Furthermore, the results (Table 4) demonstrate that pretraining on multiple services effectively addresses the cold-start problem by learning a richer user representation.

CLUE

Performing general representation learning on training data from different services and producing a unified view is a crucial challenge in building foundation models for user modeling. In this paper, we look into a multi-modal contrastive learning framework for enhancing the representation quality of the user embeddings by considering each service as a modality. A notable example of a multi-modal contrastive learning is CLIP (Radford et al. 2021), which encodes pairs of images and texts separately into vectors and maximize their similarity scores. CLIP outperforms the best publicly available models in a wide range of computer vision tasks.

A remaining issue is aligning user semantics across multiple services. While CLIP leverages image-text paired data for semantic alignment, it is non-trivial to align the items of two heterogeneous services. To address this issue, we use the text description of each item rather than its ID as inputs. Natural language input for items enables our user representations to naturally and flexibly transfer to other services and even other company platforms where items can be described with text, thus enhancing the generality of our method. As illustrated in Figure 1, such a flexible embedding space allows us to jointly train user logs over multiple services, such as search engine and e-commerce, by maximizing the agreement between the same user with a single encoder.

Our encoder has a hierarchical structure consisting of an Item Transformer, T_{item}, and a Service Transformer, T_{service}. The Item Transformer encodes the features of each item in the user's behavior log, namely $T_{\text{item}} : \mathbb{R}^{L \times D_{\text{in}}} \to \mathbb{R}^{L \times D_{\text{out}}}$. The Service Transformer encodes a user embedding for a specific service from the sequence of item embeddings, i.e. $T_{\text{service}} : \mathbb{R}^{S \times D_{\text{out}}} \to \mathbb{R}^{S \times D_{\text{out}}}$. Here D_{in}, D_{out}, L and S denote the token embedding dimension, the output dimension, the maximum token length of an item description, and the number of items in an item sequence, respectively. For each service, we define a user behavior log as $u = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_S]$, a sequence of items that a user acts upon or selects in the service. Each item $\boldsymbol{x}_i \in \mathcal{V}^L$ is a vector of token indices where the first couple tokens represent the type of service and the following tokens are the tokenized description of the item (e.g., product description, news title, or search query). We fill the remaining spaces with zeros (see more details in Appendix A). Each token is embedded to a vector via an embedding layer $g : \mathcal{V} \to \mathbb{R}^{D_{\text{in}}}$, so each item $\boldsymbol{x}_i$ is embedded to a matrix $E_i \in \mathbb{R}^{L \times D_{\text{in}}}$. The token embedding E_i is then propagated through the model in the following manner:

$$\boldsymbol{h}_i = \texttt{MEAN}(T_{\text{item}}(E_i)), \tag{1}$$

$$H = [\boldsymbol{h}_1 | \boldsymbol{h}_2 | \cdots | \boldsymbol{h}_S], \tag{2}$$

$$\boldsymbol{z} = \texttt{MEAN}\left(T_{\text{service}}(H)\right), \tag{3}$$

where MEAN is mean pooling of row vectors in the input matrix, and $\boldsymbol{z}$ is the final user embedding. We again denote the final user embedding of user u and service A as $\boldsymbol{z}_{u,A}$. We follow the loss of CLIP (Radford et al. 2021). The loss

$l_{u,A,B}$ of each positive pair $(\boldsymbol{z}_{u,A}, \boldsymbol{z}_{u,B})$ is defined as:

$$l_{u,A,B} = -log\frac{\exp(\langle f(\boldsymbol{z}_{u,A}), f(\boldsymbol{z}_{u,B})\rangle\tau)}{\sum_v \exp(\langle f(\boldsymbol{z}_{u,A}), f(\boldsymbol{z}_{v,B})\rangle\tau)}, \quad (4)$$

where τ is a temperature parameter and $\langle\cdot,\cdot\rangle$ is the cosine similarity. We use a non-linear projection model $f(\boldsymbol{z}) = W_2\sigma(W_1\boldsymbol{z})$ to improve the representation quality of the user embeddings (Chen et al. 2020), where σ is a non-linear activation function and W is a weight matrix. We optimize the symmetric cross-entropy loss $(l_{u,A,B} + l_{u,B,A})/2$. The final user features for the downstream tasks are extracted by concatenating each service user feature, or for the case of the company-level transferability task, extracted by using only task-specific user logs.

The training details and hyperparameters of best CLUE are described in Appendix B.

Experiments

Pretraining Dataset

We construct a sufficiently large-scale dataset with more than 50B behavior tokens collected over 2 years from search engine and e-commerce platform. We exclude the users who act less than once every two months in terms of behavior log frequency. We use Byte-level BPE (BBPE) (Wang, Cho, and Gu 2020) to tokenize the textual description of each item in the user behavior logs. If a user repeated the same behavior (e.g., performed the same search query or purchased the same product multiple times), we keep only one of the entries in the behavior log, to count it as a unique behavior. As a result, the training dataset contains 11 million users and 5.3 billion user behavior logs, and 50 billion BBPE tokens collected over 2 years. The 11 million users are sampled for training, but any user can be a candidate for the downstream tasks. The statistics of the dataset for pretraining are provided in Appendix A.

Downstream Tasks

Benchmark dataset. We select two categories *"Books"* and *"Clothing Shoes and Jewelry"* from Amazon review dataset (Ni, Li, and McAuley 2019). This dataset contains product reviews and scores with product metadata like product titles and categories. We filtered the case where the review score was 4 points or higher, and only the title of the product was used as metadata. Each user's review history was listed chronologically and used as a historical log, and up to 64 were used. The most recent three reviews were used as target sets.

Industrial dataset. We build downstream tasks using data from services that are different from the service domains used in the pretraining dataset. For example, downstream tasks are built from services of an e-commerce (PCR), web-based cartoon (FWR), news (NVR), marketing messages (MMR), and an online travel agency (OTAR). Furthermore, we validate the company-level transferability of our model, defined as Inter-Company-Level Transfer (ICLT). To achieve this, we secure log data from a completely different company that has an online marketplace environment (see more details in Appendix C). We describe details of the downstream datasets in Appendix D.

Models	Inputs	Speedup	Params.	Memory
Transformer†	Task-specific logs	1	15M	1G
LightGCN	Task-specific logs	10×	0.5M	4G
CLUE	Pretrained user repr.	43×	0.5M	0.5G

† Transformer based models including BST, UserBERT, and UniSRec

Table 1: Computational cost comparison of the downstream models measured from the Books task.

Experimental Settings. The downstream tasks are composed of recommendation tasks where the models predict the next item to recommend. The datasets consist of positive and negative pairs (u, i) of users and items. We have a positive pair (u, i^+) when a user u interacted with an item i^+, while a negative pair (u, i^-) is generated through random sampling. We compute three standard metrics for evaluation: top-k Hit Ratio (HR@k), top-k Normalized Discounted Cumulative Gain (NDCG@k), and Mean Reciprocal Rank (MRR). We obtain these metrics by evaluating a pool of items consisting of a ground-truth item mixed with 100 randomly sampled negative items. To test the generalization ability of the models, we make sure there are no shared users between the training, validation, and test sets.

Downstream Models

We have two types of downstream models, **task-agnostic** and **task-specific** models. **Task-agnostic** models use the pretrained user representation as the user feature and project it with a simple MLP for each downstream task. CLUE falls into this category of models, and we compare its performance to two other task-agnostic models, ShopperBERT (Shin et al. 2021) and SimCLR-based (Chen et al. 2020) learning, to validate the effectiveness of our proposed learning strategy. **Task-specific** models—excluding ShopperBERT, SimCLR, and CLUE—use task-specific historical logs of users as the user features. In addition, we consider a **Hybrid** model that combines the task-specific logs with the pretrained user features from CLUE, for additional insight. For the item embeddings of the downstream tasks, we utilize the item's text information. We use Sentence-BERT (Reimers and Gurevych 2019) to extract item embedding vectors for the **task-specific** models, while we use the task-agnostic pretrained models (i.e., ShopperBERT, SimCLR and CLUE) for the **task-agnostic** models.

Note that using raw task-specific logs requires much more computational costs (see Table 1). The detailed description of comparison models are outlined in Appendix E.

Results

User Representations for Downstream Transfer

We present the adaptability and generality of the pretrained user representations by showing state-of-the-art performance on diverse services, which data has different distributions from our pretraining data. In addition, we show that

Downstream tasks	Metrics	DeepFM	BST	LightGCN	YTMoE	UserBERT	UniSRec	CLUE (15M)
Books	HR@1	0.0580	0.0676	0.0889	0.0947	0.0850	0.0618	0.1087
	NDCG@10	0.1857	0.1916	0.1875	0.1925	0.1826	0.1895	0.2104
	MRR	0.1552	0.1606	0.1690	0.1742	0.1644	0.1601	0.1854
Clothing	HR@1	0.0812	0.1169	0.0924	0.0846	0.1208	0.1091	0.1564
	NDCG@10	0.2082	0.2294	0.2245	0.2048	0.2544	0.2228	0.2857
	MRR	0.1777	0.2046	0.1929	0.1783	0.2228	0.1971	0.2481

Table 2: Results on the benchmark downstream tasks. Pretraining and then transferring models (i.e., UniSRec, UserBERT, and CLUE) are pretrained using the history logs of "Books" and "Clothing Shoes and Jewelry". The user feature from CLUE is extracted as same as the ICLT task. The best results are underlined.

Downstream tasks	Metrics	Task-specific					Task-agnostic			Hybrid
		DeepFM	BST	LightGCN	UserBERT	UniSRec	ShopperBERT	SimCLR	CLUE	
PCR	HR@1	0.4297	0.4832	0.4792	0.5134	0.5024	0.4991	0.5114	**0.5414**	0.5485
	NDCG@10	0.6696	0.7017	0.6988	0.7140	0.7126	0.7174	0.7231	**0.7418**	0.7466
	MRR	0.5979	0.6380	0.6344	0.6642	0.6478	0.6552	0.6626	**0.6857**	0.6912
MMR	HR@1	0.1609	0.2498	0.2485	0.2874	0.2648	0.2744	0.3032	**0.3146**	0.2987
	NDCG@10	0.4430	0.5354	0.5325	0.5627	0.5504	0.5637	0.5873	**0.6030**	0.5856
	MRR	0.3048	0.4019	0.3995	0.4514	0.4326	0.4298	0.4578	**0.4713**	0.4532
NVR	HR@1	0.5492	0.5750	0.4412	0.5790	0.5815	0.5492	0.5720	**0.5927**	0.5950
	NDCG@10	0.6992	0.7161	0.5849	0.7127	0.7108	0.7025	0.5589	**0.7253**	0.7287
	MRR	0.6615	0.6806	0.5498	0.6866	0.6903	0.6665	0.6789	**0.6924**	0.6954
OTAR	HR@1	0.1654	0.1616	0.1794	0.1628	0.1720	0.1739	0.1754	**0.1977**	0.2003
	NDCG@10	0.4097	0.4092	0.4235	0.4045	0.4161	0.4192	0.4216	**0.4445**	0.4471
	MRR	0.3285	0.3292	0.3439	0.3334	0.3429	0.3393	0.3415	**0.3653**	0.3682
FWR	HR@1	No History Logs					0.1281	0.1282	**0.1538**	No History Logs
	NDCG@10						0.7015	0.7003	**0.7220**	
	MRR						0.2464	0.2468	**0.2804**	
ICLT	HR@1	0.4109	0.4489	0.4781	0.4891	0.4915	N/A	0.4543	**0.5081**	0.5262
	NDCG@10	0.6302	0.6620	0.6832	0.7017	0.6982		0.6628	**0.7036**	0.7112
	MRR	0.5631	0.5964	0.6215	0.6334	0.6312		0.5987	**0.6440**	0.6560

Table 3: Results on the industrial downstream tasks. The best results among the Task-agnostic and all models are denoted in bold and underlined fonts, respectively.

Method	Engagement Metric			
	New	Cold	Heavy	Total
GNN (Jeong et al. 2020)	-0.7%	+9.0%	+9.8%	+2.9%
CLUE (15M)	+4.1%	+11.0%	+10.5%	+6.5%
CLUE (120M)	+4.5%	+13.4%	+10.7%	+7.3%

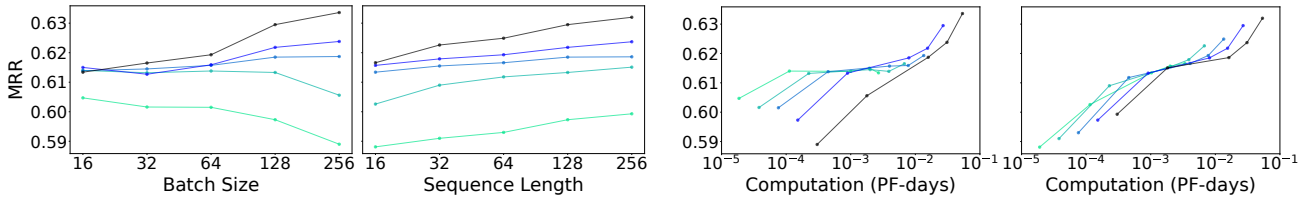
† TopPop recommends the most popular items to any user.

Table 4: Performance gain on the online PCR task, compared with a baseline method (TopPop†).

the simple transfer learning using CLUE remarkably outperforms the complex models in two benchmark datasets. This empirically demonstrates the generalization ability of the pretrained features by CLUE in wide spectrum of applications. The results are presented in Table 2, 3, and 4.

Results on the Benchmark datasets. We compare CLUE with six task-specific models, including the models train from scratch—DeepFM (Guo et al. 2017), BST (Chen, Zhao et al. 2019), LightGCN (He et al. 2020), and YTMoE (Zhao et al. 2019)—and pretraining and then finetuning models, i.e., UniSRec (Hou et al. 2022) and UserBERT (Wu et al. 2022). For all the tasks and metrics, CLUE outperforms all models by meaningful margins, particularly on MMR (over 6%). These results show effectiveness of our multi-service contrastive learning with a feature-based transfer framework compared to other seq-to-seq contrastive models with finetuning even with considerable computation efficiency (Table 1).

Results on the Industrial datasets. CLUE significantly outperforms the other methods in all tasks and metrics. For the Product Collection Recommendation and Marketing Message Recommendation tasks, CLUE outperforms best baseline models by over 3% in terms of MRR. CLUE achieves 0.3653 and 0.2804 MRR scores on Online Travel



(a) Darkness of the line corresponds to the bigger model size. (b) Darkness of the line corresponds to the bigger batch/sequence.

Figure 2: As more computing resources become available, we can choose how to allocate the resources when scaling up the training: larger models, larger batches, and longer sequence lengths. (a) The performance on the ICLT task when either the sequence length (128) or the batch size (256) is fixed. (b) To evaluate the efficiency of the training schemes, we report the performance improvement in terms of the amount of training computation (PF-days). Within the same line, each dot represents a model size ranging from 4M- to 160M-parameters in increasing order.

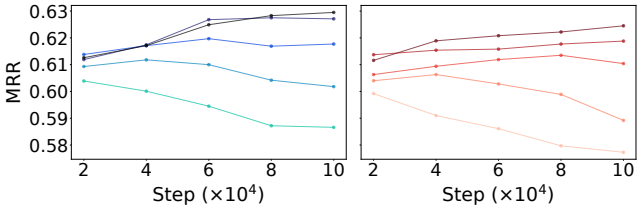


Figure 3: Performance on the ICLT task according to the number of training steps. The line in the figure becomes progressively darker (with values of 1%, 5%, 10%, 30%, and 100%), indicating the use of an increasing number of data. CLUE is trained with 160M parameters, sequence length (128), and batch size (256). (Left) With batch shuffling during training to help the model learn from the various negative examples. (Right) Without shuffling during training.

Agency Recommendation and Favorite Webtoon Recommendation, that is a 6% and 14% increase compared to the MRR score of best basline models, respectively. In the News View Recommendation tasks, CLUE shows the only marginal performance improvement compared to UniSRec.

For the Inter-Company-Level Transfer task, the results of CLUE on the MRR metric is higher than that of UserBERT by 2%. ShopperBERT cannot be evaluated on the ICLT task due to its product ID-based MLM loss, since the target company uses different product IDs from our system.

It is worth noting that the transfer learning with the user representations by CLUE results in consistent performance gains in challenging real-world downstream applications, showcasing the effectiveness of the proposed CLUE-based user representation learning. These results support the efficacy of our CLIP-style multi-service contrastive learning.

Also, we report the results of a hybrid approach combining user features from CLUE and a task-specific model with historical logs. This hybrid shows the best performances on most tasks and metrics except for MMR. These results support our CLUE can be complementary with task-specific models and improve the conventional methods by providing generalized user features.

Online Results. We conducted an online test for PCR task on our e-commerce platform for five days in November 2021. We split the users into three groups—new, cold, and heavy—based on their engagement frequency for each service. The user group 'new' corresponds to users with no recorded behavior on the service for the past month. The user group 'cold' corresponds to the bottom 10% in terms of the amount of activity while the group 'heavy' represents the top 10%. The results are presented in Table 4.

Compared with the conventional baseline TopPop, CLUE (120M) significantly increased the Click-Through-Rate (CTR) engagement by 7%. As a task-specific model, we employ a graph neural network (GNN) (Jeong et al. 2020) which is a random-walk based graph representation learning method. CLUE (120M) comfortably outperforms GNN by 5% in terms of CTR engagement. It is worth mentioning that CLUE (120M) obtains more CTR engagement than CLUE (15M). The result verifies that the universal scaling law still works in online scenarios.

Furthermore, we investigate how the models perform with different user segments. For user groups 'new' and 'cold', the user representation pretrained by CLUE significantly improves CTR compared to GNN. In particular, the CTR of GNN is lower than that of TopPop for the 'new' user group that lacks behavior logs, while CLUE consistently shows outstanding performances. Interestingly, as the amount of task-specific history logs increases from 'new' group to 'heavy' group, the CTR difference between GNN and CLUE decreases. We conjecture that CLUE might provide more generalized user representation for new and cold users, thus contributing to practical applications.

Scaling Laws and Generalization

Recently, several works empirically demonstrate the existence of a scaling law, where the training error scales as a power-law with model capacity, data size, and the amount of computation (Brown et al. 2020; Kaplan et al. 2020; Zhai et al. 2021; Bahri et al. 2021). Studies on the scaling law have significantly broadened the field of reasoning, but the findings are mostly restricted to NLP and computer vision. In this subsection, we empirically observe the power-law decrease of the pretraining error in user modeling areas w.r.t. scales. The computation (PF-days) is calculated as 6 $\times$ # of parameters $\times$ batch size $\times$ # of training steps $\times$ sequence length divided by one PF-day = 8.64×10^{19}. We train all models for 100,000 steps.

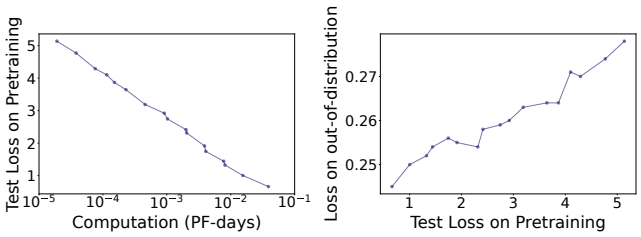


Figure 4: (Left) Log-linear plot between the test loss on the pretraining distribution and the computation (PF-days). This shows that the pretraining learning curve has a power-law scaling as a function of the total training computation (PF-days) when not bottlenecked by other factors. (Right) The generalization on downstream tasks depends on the test loss of pretraining. We observe a strong trend that a lower loss on the in-distribution data results in a lower loss on the out-of-distribution data.

Tasks	Metrics	Single	Stacking
PCR	HR@10	0.5341	0.5414
	NDCG@10	0.7377	0.7418
	MRR	0.6808	0.6857
ICLT	HR@10	0.4956	0.5081
	NDCG@10	0.6934	0.7036
	MRR	0.6333	0.6440

Table 5: Performance comparison of Single and Stacking Transformer encoder for CLUE. They both compete with the same GPU resources.

Output Dimensions	HR@10	NDCG@10	MRR
300D	0.5414	0.7418	0.6857
2160D	0.5360	0.7390	0.6822

Table 6: Decreasing the output feature dimension of CLUE does not lead to a significant difference in the performance on the downstream (PCR) tasks.

Performance on Scale. In contrast to existing work on the scaling law in other domains, we observe that the performance of the model does not depend dominantly on model capacity alone in the user representation tasks (Figure 2(a)-Left). Previous studies argued that learning high-quality representations from batch-wise contrastive loss requires a sufficient amount of negative samples (Chen et al. 2020; Mitrovic et al. 2021; Gao et al. 2021). Thus, we speculate that the scaling law when learning with a contrastive objective is more complex than that of supervisory signals due to the bottleneck induced by the batch size.

CLUE's performance grows smoothly as the sequence length of the input data increases, suggesting that user behavior models benefit from observing longer sequences of customer behavior (Figure 2(a)-Right). We can see the performance change according to the amount of training computation (Figure 2(b)). These results give us an insight into how to appropriately allocate computing resources for efficient training schemes.

We further conduct an experiment on increasing the training dataset size from 1% to 100%. Figure 3 shows the performance on the ICLT task according to the number of training steps. The result is consistent with the trend in other domains that scaling up the training dataset leads to a strict performance improvement on the downstream tasks (Brown et al. 2020; Kaplan et al. 2020; Zhai et al. 2021). Interestingly, we can observe the considerable positive effects of batch shuffling. It is also surprising that CLUE trained on 10% of the dataset—using only 1,130,000 users—with random shuffling can achieve competitive results with the LightGCN trained on the full dataset of historical logs on the ICLT task. From the results of Figure 2, 3, and 4, we can conclude that all four factors must scale up in tandem for optimal performance.

Generalization on Other Data Distributions. The in-distribution test loss as a function of computation is shown in Figure 4-Left. We empirically verify that the loss on the pretraining dataset improves smoothly with more computation. This result aligns with the reports in other domains, which has shown that increasing the amount of computation positively affects the performance of pretrained models (Brown et al. 2020; Kaplan et al. 2020; Zhai et al. 2021). Moreover, when we transfer knowledge to datasets with a different distribution than the one used during pretraining, the resulting test losses show a strong correlation with the pretraining performance. These results indicate that generalization ability to various data distributions is strongly dependent on the pretraining test loss (Figure 4-Right).

Ablation Study on Encoder Architectures

Efficacy of Stacked Transformers. We conduct an ablation study to analyze the effects of stacking Transformers, i.e, the Item Transformer and the Service Transformer, in our model. For fair comparison, we train a single Transformer encoder with 235M parameters, sequence length 2,048, batch size 256, and the same GPU resources as the best CLUE model. As presented in Table 5, the proposed stacking approach performs better than the single Transformer encoder with the same configuration. We conjecture that separating the encoding process enhances the representation quality of the encoder. Furthermore, the stacking approach allows our model to observe more user behaviors compared to using a single Transformer.

Output Dimensions. The final user feature dimensions of the non-curated best CLUE model is 2,160. The 2,160 feature dimensions for whole users results in a size of 153 GB when stored in half-precision floating-point format. If we slightly increase the loading time or the number of services, the storage becomes too large to handle. Thus, we conduct an experiment on whether the output dimension of CLUE affects its performance. We added a single MLP layer for reducing the dimension of the encoder outputs. Table 6 shows that decreasing the output dimension does not lead to any performance degradation on the downstream transfer tasks.

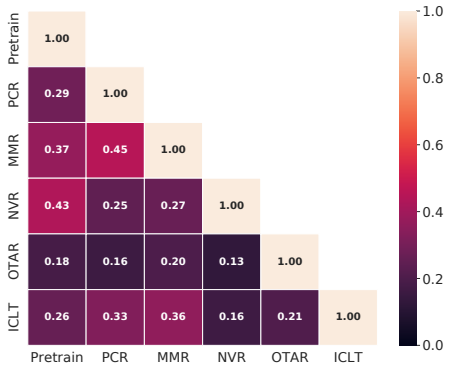


Figure 5: The Kendall rank correlation of the token distribution between the pretraining and downstream domains. We use the user behavior logs u to calculate the token distributions.

Limitations

Although we have adopted a scaling-up strategy, we believe that a one-size-fits-all approach to the user modeling area continues to be challenging, and there remains tricky downstream tasks that have not yet been addressed. The domain of the pretrain data can be a constraint in learning a complete general-purpose user representation. Thus, we investigate the similarity of source and target domain with performance on the target task. Recently, Gururangan et al. (2020) investigated whether it is helpful to transfer a pretrained model to the domain of a target task. They used the vocabulary overlap (%) to measure the similarities between the source and the target domains. Since it is difficult to measure the vocabulary overlap of the user behavior dataset between different domains, we estimate the Kendall rank correlation (Abdi 2007) using the token distribution of the user behavior logs of each domain.

Figure 5 shows the correlation across the downstream task datasets. We observe that the correlation and performance improvement do not align perfectly. According to the results of Table 3 and Figure 5, the correlation and the relative performance increase of CLUE against the task-specific models—except the pretraining models—show a trend, but not in all cases. Our work has not made much progress towards finding a criteria for a well-transferable domain. A more careful study on this subject is left for future research.

Related Work

Contrastive Learning. Contrastive learning (Hadsell, Chopra, and LeCun 2006) aims to learn high-quality representations by contrasting positive sample pairs against negative sample pairs. Chen et al. (2020) and Caron et al. (2020) demonstrated that contrastive visual representation learning could produce comparable results to the supervised method on several vision tasks. These approaches have been popularized for other domains, such as language modeling (Gao, Yao, and Chen 2021) and speech recognition (Baevski et al. 2020). Recently, Akbari et al. (2021) and Radford et al. (2021) presented a method for learning a flexible prediction space for different modalities by employing contrastive losses to train the model.

Scaling Law. Over the decades, the connection between scaling and generalization has been studied broadly from both theoretical and empirical perspectives (Mhaskar 1996; Brown et al. 2020; Kaplan et al. 2020; Bahri et al. 2021; Zhai et al. 2021; Hutter 2021). Several studies theoretically demonstrated that the training error scales as a power-law with larger model capacity and more data (Mhaskar 1996; Bahri et al. 2021; Hutter 2021). Recent research focuses on empirical analysis of scaling laws for real-world applications (Brown et al. 2020; Kaplan et al. 2020; Zhai et al. 2021). These works showed that scaling up the model and dataset size is a promising approach for achieving outstanding performances in several NLP and computer vision tasks, e.g., machine translation, generative modeling and image recognition.

General-Purpose User Representation Learning. Compared to task-specific user representation learning, the research about the general-purpose user representation learning is still in its early stage. Yuan et al. (2021) continually learn user representations on multiple tasks with a single model without network expansion and catastrophic forgetting. Wu et al. (2022) and Hou et al. (2022) contrastively pretrain user models to effectively capture the relations between user behaviors and inherent user interests. Note that these models are all restricted to fine-tuning based approaches (Yuan et al. 2021; Wu et al. 2022; Hou et al. 2022). The recent progress in general representation learning naturally attempts to build more efficient and widely adaptable user representations. Gu et al. (2021) employ a novel objective function named behavioral consistency loss to preserve the user's long-term interest and benefit from diverse behaviors. Sun et al. (2021) present interest-oriented contrastive learning, which maximizes the agreement between short- and long-term interest representations of the same users. These methods can be applicable to numerous downstream tasks without further modifications (Gu et al. 2021; Sun et al. 2021).

Following the previous studies, we focus on learning adaptable, general user representations and connecting the bridge between user representation learning and the empirical scaling law.

Conclusion and Impact

We present CLUE trained on billions of scale real-world user behavior data to learn general-purpose user representations. We benchmark the various downstream tasks with a simple MLP, and achieve promising results, including company-level transferring task. We further investigate the empirical scaling laws and the generalization ability of our method, and find that the power-law learning curve as a function of computation (PF-days) is observed in the experiments. Despite the remaining limitations, we believe that our empirical analysis can share useful insights on large-scale user representation learning.

Acknowledgments

The authors would like to thank the NAVER CLOVA ML X team for insightful comments and discussions. We would also like to thank the NAVER Smart Machine Learning (NSML) platform team (Sung et al. 2017; Kim et al. 2018) for their critical work on the software and hardware infrastructure on which all the experiments were performed.

References

Abdi, H. 2007. The Kendall rank correlation coefficient. *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, 508–510.

Akbari, H.; Yuan, L.; Qian, R.; Chuang, W.-H.; Chang, S.-F.; Cui, Y.; and Gong, B. 2021. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. In *Advances in neural information processing systems*.

Baevski, A.; Zhou, Y.; Mohamed, A.; and Auli, M. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in neural information processing systems*.

Bahri, Y.; Dyer, E.; Kaplan, J.; Lee, J.; and Sharma, U. 2021. Explaining neural scaling laws. *arXiv preprint arXiv:2102.06701*.

Bommasani, R.; Hudson, D. A.; et al. 2021. On the Opportunities and Risks of Foundation Models. *arXiv preprint arXiv:2108.07258*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; et al. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*.

Chen, Q.; Zhao, H.; et al. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, 1–4.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*. PMLR.

Dai, Z.; Liu, H.; Le, Q. V.; and Tan, M. 2021. CoAtNet: Marrying Convolution and Attention for All Data Sizes. In *Advances in neural information processing systems*.

Devlin, J.; Chang, M.-W.; Lee, K.; et al. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Gao, L.; Zhang, Y.; Han, J.; and Callan, J. 2021. Scaling deep contrastive learning batch size under memory limited setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, 316–321.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*.

Goyal, P.; Caron, M.; Lefaudeux, B.; Xu, M.; Wang, P.; Pai, V.; et al. 2021. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*.

Gu, J.; Wang, F.; Sun, Q.; Ye, Z.; Xu, X.; Chen, J.; and Zhang, J. 2021. Exploiting behavioral consistence for universal user representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4063–4071.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *IJCAI*.

Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; and Smith, N. A. 2020. Don't stop pretraining: adapt language models to domains and tasks. In *ACL*.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 1735–1742.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, 639–648.

Hou, Y.; Mu, S.; Zhao, W. X.; Li, Y.; Ding, B.; and Wen, J.-R. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. *KDD'22*.

Hutter, M. 2021. Learning Curve Theory. *arXiv preprint arXiv:2102.04074*.

Jeong, J.; Yun, J.-M.; Keam, H.; et al. 2020. div2vec: Diversity-Emphasized Node Embedding. In *ImpactRS Workshop at Recsys 2020*.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; et al. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Kim, B.; Kim, H.; Lee, S.-W.; et al. 2021. What Changes Can Large-scale Language Models Bring? Intensive Study on HyperCLOVA: Billions-scale Korean Generative Pretrained Transformers. In *EMNLP*.

Kim, H.; Kim, M.; Seo, D.; Kim, J.; Park, H.; Park, S.; et al. 2018. Nsml: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*.

Mhaskar, H. N. 1996. Neural networks for optimal approximation of smooth and analytic functions. *Neural computation*, 8(1): 164–177.

Mitrovic, J.; McWilliams, B.; Walker, J.; et al. 2021. Representation learning via invariant causal mechanisms. In *ICLR*.

Ni, J.; Li, J.; and McAuley, J. 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.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*.

Rae, J. W.; Borgeaud, S.; Cai, T.; Millican, K.; et al. 2021. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. *arXiv preprint arXiv:2112.11446*.

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*.

Shin, K.; Kwak, H.; Kim, K.-M.; Kim, M.; Park, Y.-J.; Jeong, J.; and Jung, S. 2021. One4all User Representation for Recommender Systems in E-commerce. *arXiv preprint arXiv:2106.00573*.

Sun, Q.; Gu, J.; Yang, B.; Xu, X.; et al. 2021. Interest-oriented Universal User Representation via Contrastive Learning. *arXiv preprint arXiv:2109.08865*.

Sung, N.; Kim, M.; Jo, H.; Yang, Y.; Kim, J.; Lausen, L.; Kim, Y.; et al. 2017. Nsml: A machine learning platform that enables you to focus on your models. *arXiv preprint arXiv:1712.05902*.

Wang, C.; Cho, K.; and Gu, J. 2020. Neural machine translation with byte-level subwords. In *AAAI*.

Wang, Z.; Yu, J.; Yu, A. W.; Dai, Z.; Tsvetkov, Y.; and Cao, Y. 2022. Simvlm: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations (ICLR)*.

Wu, C.; Wu, F.; Qi, T.; and Huang, Y. 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, 2087–2092. New York, NY, USA: Association for Computing Machinery. ISBN 9781450387323.

Yuan, F.; Zhang, G.; Karatzoglou, A.; et al. 2021. One person, one model, one world: Learning continual user representation without forgetting. In *SIGIR*, 696–705.

Zhai, X.; Kolesnikov, A.; Houlsby, N.; and Beyer, L. 2021. Scaling vision transformers. *arXiv preprint arXiv:2106.04560*.

Zhao, Z.; Hong, L.; Wei, L.; Chen, J.; Nath, A.; Andrews, S.; Kumthekar, A.; Sathiamoorthy, M.; Yi, X.; and Chi, E. 2019. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 43–51.