Towards Personalized Cold-Start Recommendation with Prompts

XUANSHENG WU, University of Georgia, United States
HUACHI ZHOU, Hong Kong Polytechnic University, China
YUCHENG SHI, University of Georgia, United States
WENLIN YAO, Tencent AI Lab, United States
XIAO HUANG, Hong Kong Polytechnic University, China
NINGHAO LIU, University of Georgia, United States

Recommender systems play a crucial role in helping users discover information that aligns with their interests based on their past behaviors. However, developing personalized recommendation systems becomes challenging when historical records of user-item interactions are unavailable, leading to what is known as the *system cold-start* recommendation problem. This issue is particularly prominent in start-up businesses or platforms with insufficient user engagement history. Previous studies focus on user or item cold-start scenarios, where systems could make recommendations for new users or items but are still trained with historical user-item interactions in the same domain, which cannot solve our problem. To bridge the gap, our research introduces an innovative and effective approach, capitalizing on the capabilities of pre-trained language models. We transform the recommendation process into sentiment analysis of natural languages containing information of user profiles and item attributes, where the sentiment polarity is predicted with prompt learning. By harnessing the extensive knowledge housed within language models, the prediction can be made without historical user-item interaction records. A benchmark is also introduced to evaluate the proposed method under the cold-start setting, and the results demonstrate the effectiveness of our method. To the best of our knowledge, this is the first study to tackle the system cold-start recommendation problem. The benchmark and implementation of the method are available at https://github.com/JacksonWuxs/PromptRec.

Additional Key Words and Phrases: Cold-start Recommendation, Prompt Learning, Language Models.

ACM Reference Format:
Xuansheng Wu, Huachi Zhou, Yucheng Shi, Wenlin Yao, Xiao Huang, and Ninghao Liu. 2018. Towards Personalized Cold-Start Recommendation with Prompts. In . ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Recommender systems help users discover items that are tailored to their interests. An accurate recommender system is essential for online commerce as it can increase user engagement and promote sales. Traditional recommender systems such as collaborative filtering [17, 18, 43] and content-based methods [10] rely on historical user-item interactions (e.g., clicks and purchases) to learn user preferences and find matched items for users. However, this pipeline would fail in scenarios where we could *not* obtain any user-item interactions, and we call it the *system cold-start* recommendation problem, which typically happens in situations such as start-up businesses. Although cold-start recommendation scenarios have been studied in previous research [21, 28, 32], as illustrated in Figure 1, they still assume historical user-item interactions are available for training or during inference, which differs from our setting. A straightforward strategy to

Manuscript submitted to ACM

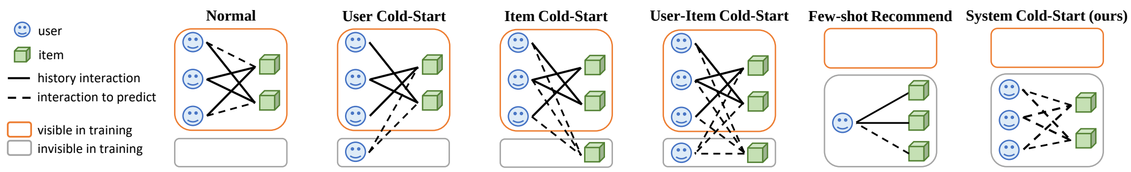

Fig. 1. Illustration of different cold-start recommendation scenarios, including user cold-start [32], item cold-start [28], user-item cold-start [21], few-shot recommendation [4, 53], and ours.

tackle the system cold-start recommendation problem is to manually design rules, such as recommending popular or seasonal items [7, 32], but the recommendation results are less personalized and could hurt user experiences.

The recent prompt learning with pre-trained language models (PLMs) in natural language processing [3, 26, 42] seems to point out a promising direction to solve the system cold-start recommendation problem. Ideally, if the relation between user interests and item properties is implicitly expressed as a natural language in public corpora, then it could be captured by PLMs to be used for recommendations. Some researchers [4, 37, 53**? ?**] began to investigate whether it is possible to conduct personalized recommendations by adopting existing PLMs. Typically, they describe the user interaction history with natural language and format the next-item-recommendation task as the language modeling task using manually designed templates. Then a PLM is asked to predict the word probability of the next item's name for recommendations. For example, one template used for movie recommendation is "*The user watched movies* A, B, C. *Now the user may want to watch the movie* [D]", where A~C are the latest movies watched by the current user and [D] is the name of a candidate movie [37, 53]. They fill [D] with each candidate and recommend the movie whose name has the highest probability. Although these studies do not train a new model using historical user-item records, they still require this data during inference, so it cannot tackle our system cold-start problem.

In this study, we propose a simple but effective approach based on prompt learning, named *PromptRec*, to tackle the challenging problem of system cold-start recommendation. PromptRec considers the user and item profile features as input, transforms them into templated sentences, and predicts recommendations through PLMs. Specifically, we first introduce a verbalizer mapping profile features to natural language descriptions, then apply a template reformatting the recommendation task as the Masked Language Modeling (MLM) task [5], and finally leverage a PLM to accomplish the MLM task and perform recommendations. We also propose a benchmark to quantitatively analyze the proposed method. We summarize our contributions as follows:

- We formalize the system cold-start recommendation problem and introduce the first benchmark to our community.
- We propose a prompt learning approach called PromptRec for system cold-start recommendation.
- We conduct experiments on different datasets and PLMs to demonstrate the effectiveness of PromptRec, and provide discussions toward the future works in this direction.

2 PRELIMINARY

This section covers the basic notations and fundamental concepts in recommender systems, and outlines a formal definition of the system cold-start recommendation problem considered in this work.

2.1 Notations

In this work, we use boldface lowercase letters (e.g., $\mathbf{c}$) to denote vectors, boldface uppercase letters (e.g., $\mathbf{R}$) to denote matrices, and calligraphic capital letters (e.g., $\mathcal{D}$) to denote sets. Specifically, each recommendation dataset $\mathcal{D} = (\mathcal{U}, \mathcal{I}, \mathbf{R})$ has a user set $\mathcal{U}$, an item set $\mathcal{I}$, and a matrix storing user-item interactions $\mathbf{R} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$, where $r_{u,i} \in \mathbf{R}$ indicates the interaction between the user u and the item i. Each user and each item has d_U and d_I profile features denoted as $\mathbf{c}_u \in \mathbb{R}^{d_U}$

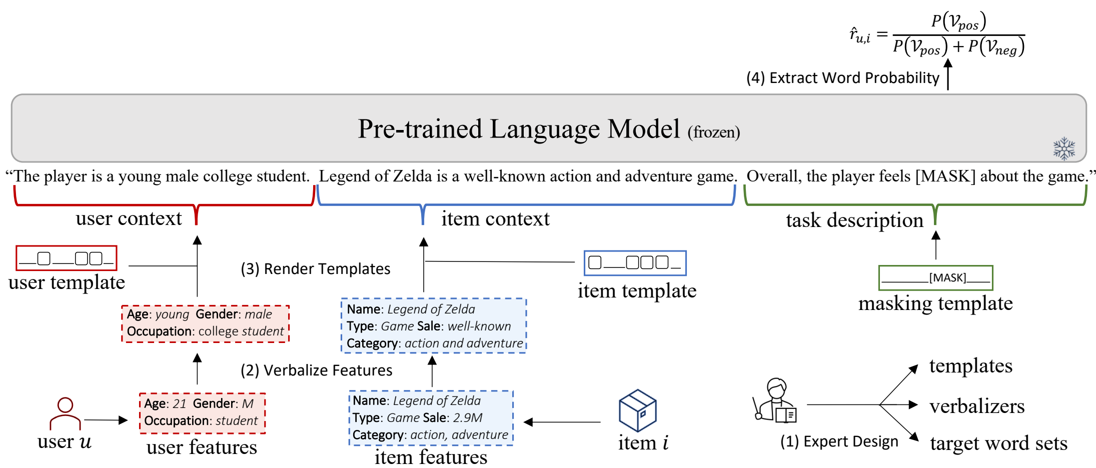

Fig. 2. PromptRec: Prompting PLMs to make personalized recommendations under the system cold-start setting.

and $\mathbf{c}_i \in \mathbb{R}^{d_I}$, respectively. The profile features are attributes that describe users or items (e.g., user's age, gender, and occupation; item's name, brand, and category).

2.2 Problem Statement

We choose the click-through rate (CTR) prediction task [34] to set up our recommendation scenario. That is, each record $r_{u,i} \in \{0, 1\}$ is a binary value, where $r_{u,i} = 1$ means user u clicked item i. The recommendation system f takes a user-item pair $(\mathbf{c}_u, \mathbf{c}_i)$ as *input* to predict the probability $\hat{r}_{u,i} \in [0, 1]$ that the user will click on the item as *output* based on model $\hat{r}_{u,i} = P(r_{u,i} = 1|\mathbf{c}_u, \mathbf{c}_i; \theta_f)$, where θ_f is the parameters of the recommendation system f. The goal of CTR prediction is to minimize the difference between the predicted probability $\hat{r}_{u,i}$ and the real user-item interaction $r_{u,i}$.

2.3 System Cold-Start Recommendation

In traditional supervised scenarios, a training set containing observed interaction triplets $\{(\mathbf{c}_u, \mathbf{c}_i, r_{u,i})\}$ between users and items is collected to optimize the model parameter θ_f. However, under the system cold-start setting, we can *not* obtain any interaction records, which is a common situation for start-ups or companies that have just launched their new business [33]. Thus, in system cold-start recommendation, we define a *target dataset* as $\mathcal{D}_{\text{tgt}} = (\mathcal{U}_{\text{tgt}}, \mathcal{I}_{\text{tgt}}, \mathbf{R}_{\text{tgt}})$ with an empty interaction matrix $\mathbf{R}_{\text{tgt}} = \emptyset$. Our goal is to recommend items in $\mathcal{I}_{\text{tgt}}$ to users in $\mathcal{U}_{\text{tgt}}$ by using their profile features $\{\mathbf{c}_u\}$ and $\{\mathbf{c}_i\}$. Using recorded interactions is not allowed, no matter in training or inference, but recommender system developers can explore available resources (e.g., account sign-up surveys) to build user and item profiles.

3 METHODOLOGY

We now introduce a prompt learning approach with pre-trained language models, called PromptRec, to deal with the challenging system cold-start recommendation problem.

The two widely used supervised learning paradigms, namely "Fully Supervised Learning" and "Pre-train and Fine-tune" [25], fail under the cold-start recommendation setting because there is no training data for tuning model parameters θ_f. To overcome this challenge, some studies [3, 36] have explored the potential of pre-trained language models (PLMs) under the few-shot setting, where the downstream task is formatted as one of the language model pre-training tasks, so-called "prompt" learning [25]. Recent prompt-based recommendation systems [4, 37, 53] align the recommendation process with the masked language modeling task, where PLMs are adopted to estimate the probability of the item's name appearing within an user-item context. For example, given a context about the user interactions as "*A user clicked hiking shoes, will also click trekking poles*", they treat the probability that "trekking poles" appears within this context to measure user preference to the candidate item "trekking poles". However, simply predicting item names is ineffective for recommendations, especially in cold-start situations. This is because the name of an item is influenced by all the words in it, and an item with a name composed of common words may have a higher probability of appearing, regardless of its relevance to the user's preferences or the context. For example, "League of Legends" naturally has a higher probability than "Legend of Zelda" since "League" is more common than "Zelda" in corpora.

To address this problem, instead of simply predicting the item names, we propose a new approach called PromptRec which: (1) provides common-sense descriptions to users and items in prompts; (2) predicts the probability of some chosen sentiment words (e.g., "good", "bad") based on the user-item context. Formally, a prompting function f_{prompt} maps the user-item pair $(\mathbf{c}_u, \mathbf{c}_i)$ into a L_{ctx}-word context $\mathcal{X}_{u,i} \in \mathcal{V}^{L_{\text{ctx}}}$ with a verbalizer g_{verb} and a template filler g_{fill}. Here, $\mathcal{V}$ is a predefined vocabulary set, g_{verb} describes profile features with natural language words, and g_{fill} generates the context combining the user and item verbalized features with a manual template $\mathcal{T}$, having a special word [MASK]. Given a PLM f_{plm} with a d-dimensional embedding space, the preference $\hat{r}_{u,i}$ from user u to item i is estimated by:

$$\hat{r}_{u,i} = \frac{P(\mathcal{V}_{\text{pos}})}{P(\mathcal{V}_{\text{pos}}) + P(\mathcal{V}_{\text{neg}})}, \tag{1}$$

where

$$P(\mathcal{V}) = \prod_{w \in \mathcal{V}} P([\text{MASK}] = w | \mathbf{E}_{u,i}; f_{\text{plm}}). \tag{2}$$

Here, $\mathbf{E}_{u,i} \in \mathbb{R}^{L_{\text{ctx}} \times d}$ is the word embedding matrix of the context $\mathcal{X}_{u,i}$; $\mathcal{V}_{\text{pos}}, \mathcal{V}_{\text{neg}} \subset \mathcal{V}$ are the predefined positive and negative sentiment vocabulary sets, respectively; $P([\text{MASK}] = w|\mathbf{E}_{u,i}; f_{\text{plm}})$ is the predicted probability that word w appears at the position [MASK] within the context $\mathcal{X}_{u,i}$ according to the model f_{plm}.

Figure 2 shows the overall framework of PromptRec. Taking game recommendation as an example, we first describe the user u's and item i's profile features with natural language as $\mathcal{X}_u = g_{\text{verb}}(\mathbf{c}_u)$ and $\mathcal{X}_i = g_{\text{verb}}(\mathbf{c}_i)$ respectively, where g_{verb} is a human-designed function verbalizing a feature vector with words. Meanwhile, we design template $\mathcal{T}$ as "*The player is a* age gender occupation. name *is categorized as a* category *video game created by* producer*. Overall, the player feels* [MASK] *about the game.*", where each underlined word is a slot. Then, the context $\mathcal{X}_{u,i} = g_{\text{fill}}(\mathcal{X}_u, \mathcal{X}_i; \mathcal{T})$[1] is generated by filler g_{fill}, which fills the slots with the corresponding verbalized features. If let $\mathcal{V}_{\text{pos}} = \{$"*good*"$\}$ and $\mathcal{V}_{\text{neg}} = \{$"*bad*"$\}$, the predicted preference $\hat{r}_{u,i}$ is computed by normalizing the probabilities of observing "good" and "bad" at position [MASK].

4 EXPERIMENT

We mainly investigate two research questions (RQs): (1) Could PromptRec make personalized recommendations in the cold-start scenario? (2) How the sizes of PLMs impact the performance of PromptRec? To answer these questions with quantitative analysis, we introduce the first benchmark evaluating cold-start recommendation systems. We wish this benchmark can facilitate future research on developing recommender systems under the cold-start setting.

[1]In this example, the description could be "*The player is a young male college student. Legend of Zelda is categorized as a action adventure video game created by Nintendo. Overall, the player feels* [MASK] *about the game.*"

Table 1. Statistics of datasets.

Dataset	#User/#Feature	#Item/#Feature	#Interaction	Density
ML-100K	943/4	1682/22	100,000	3.15%
Coupon	8312/12	6924/13	12,684	0.01%
Restaurant	138/20	939/25	1,161	0.45%

4.1 Cold-Start Recommendation Benchmark

This subsection presents the Cold-Start Recommendation Benchmark, consisting of three public datasets and a dataset partitioning strategy designed to simulate the cold-start challenge in real-world scenarios. It also considers baseline strategies, including rule-based and PLM-based methods, where all methods will be evaluated using GAUC.

4.1.1 Datasets. The constraint of cold-start recommendation is the lack of user-item interactions during training. Under this setting, recommender systems heavily rely on profile features to make personalized recommendations. We finally collect three datasets that meet the requirements: In-Vechical Coupon Recommendation (Coupon) [45], Mexico Restaurant Recommendation (Restaurant) [40], and MovieLens-100K (ML-100K) [15]. The Coupon dataset evaluates the performance of recommenders in delivering accurate shop discounts to drivers; the Restaurant dataset measures the ability of systems to predict user preferences for restaurants; the ML-100K dataset assesses the ability of models to recommend movies to users. Table 1 summarizes the dataset statistics.

4.1.2 Dataset Partition. Each dataset is split into the training, validation and testing datasets, where the training dataset consists of 250 samples, the valid dataset has 50 samples, and the rest of each dataset forms the test dataset. Notably, the training dataset is retained to enable the evaluation of recommendations under the few-shot setting, where a small number of interaction records are available for model tuning. To prevent overfitting of hyper-parameters, the validation dataset is included and kept smaller than the train dataset. All models are compared by their performances on the test dataset. We leave the evaluation under the few-shot setting for future work.

4.1.3 Data Preprocessing. In this paper, we treat recommendation as the Click-Through Rate (CTR) prediction problem. Since the initial labels of these datasets are the intensity of user preferences toward items, we transform them into binary labels $\{0, 1\}$ by introducing a threshold [30, 54], so that they can be used as benchmarks for CTR prediction. Here, the thresholds for ML-100K, Coupon, and Restaurant datasets are 4.0, 1.0, and 2.0, respectively.

4.1.4 Metrics. The CTR prediction is a binary classification task that could be evaluated by the ROC-AUC score [13]. However, AUC has limitations in personalized recommendations due to its computation involving all user-item interactions, leading to unexpected interference between different users [55]. The Group-AUC (GAUC) [16, 55] score addresses it by computing AUC for each user separately and then aggregating them by weighted average:

$$GAUC = \sum_{u \in \mathcal{U}} \frac{\#history_u \times AUC(u)}{\sum_{u \in \mathcal{U}} \#history_u}, \tag{3}$$

where $\#history_u$ is the number of records for user u, and $AUC(u)$ is the AUC over all interactions records for user u.

4.1.5 Baseline Methods. We consider two categories of baseline methods. The first category includes baselines that rely on human-designed rules. For example, *Random* strategy randomly recommend items to users. The second category includes PLM-based unsupervised methods, which use verbalized features of users and items as inputs and make recommendations by using outputs from PLMs without fine-tuning. For example, *EmbSim* [53] generates two embeddings

Table 2. Prompt learning for cold-start recommendation.

Strategy	PLM	ML-100K	Coupon	Restaurant	avg.
Baselines					
Random	-	$50.10_{\pm 0.13}$	$49.76_{\pm 1.33}$	$50.44_{\pm 2.40}$	50.10
EmbSim	BERT	$50.22_{\pm 0.01}$	$50.31_{\pm 0.12}$	$51.93_{\pm 0.87}$	50.82
PairNSP	BERT	$48.88_{\pm 0.01}$	$54.14_{\pm 0.11}$	$47.70_{\pm 2.77}$	50.24
ItemLM	BERT	$50.42_{\pm 0.01}$	$31.98_{\pm 0.16}$	$49.25_{\pm 1.76}$	43.83
Ours					
PromptRec	BERT	$52.39_{\pm 0.01}$	$\mathbf{63.77}_{\pm 0.18}$	$\mathbf{55.49}_{\pm 1.41}$	57.22
	GPT2	$54.45_{\pm 0.01}$	$55.03_{\pm 0.19}$	$51.83_{\pm 1.06}$	53.77
	T5	$\underline{56.16}_{\pm 0.01}$	$51.11_{\pm 0.21}$	$52.68_{\pm 0.88}$	53.32
	LLaMA	$\mathbf{57.03}_{\pm 1.89}$	$\underline{56.08}_{\pm 0.09}$	$\underline{54.43}_{\pm 1.45}$	55.85

- Results are converted to percentages for readability. We **bold** the best results and underline the second-best results in each dataset.

of the user and item verbalized features and predicts user-item preferences by taking the dot product of their embeddings. *PairNSP* applies the next sentence prediction task [5], where the verbalized features of the user and item are concatenated and fed into a PLM to determine whether they belong to the same context. *ItemLM* [4] concatenates user-item verbalized features, but predicts the preference by calculating the likelihood of the item's name appearing within this context.

4.2 RQ1: Prompt Learning is Effective in Cold-Start Recommendation

This subsection demonstrates that pre-trained language models empowered by prompt learning could make personalized recommendation under the cold-start setting.

4.2.1 Settings.

Pre-trained language model candidates. We consider diverse large language models across different scales, architectures, pre-training strategies, and model sizes to show the generalization of PromptRec. Based on factors including accessibility and popularity, we choose *BERT-large-uncased* [5], *GPT-2-medium* [1], *T5-large* [31], and *LLaMA-7B* [38]. The numbers of their parameters span from 355M to 7B. We use their implementations and checkpoints from Huggingface [49].

Template, verbalizer, and labeler designs. Human experts initialize the design of templates, verbalizers, and labelers for each dataset. Each template has the following parts: the user profile, the item profile, and the connection between the above profiles and the recommendation task. We consider two types of verbalizers, namely, the continuous-feature verbalizer and the discrete-feature verbalizer. The continuous-feature verbalizer breaks down the feature value range into multiple intervals, each of which is assigned with a natural language description by experts (e.g., value 72 in the age variable is verbalized with word "old"). In contrast, the discrete-feature verbalizer directly returns a description for each feature value. For example, a certain user-item interaction from the ML-100K dataset is formatted as the following sentences via going through the human designed template and verbalizer: "*The woman is a middle-aged writer living in Michigan. The Star Wars is categorized as a adventure, animated, romantic, scientific, war movie. The user is that woman, and the item is that movie. In short, the user feels* [MASK] *about the item.*" For causal language models such as GPT-Neo and T5, the texts remain mostly the same, except that the last sentence is replaced with "*In short, the user's attitude toward the item is* [MASK]". The expert-designed labeler regards "positive" as positive words, and "negative" as negative words. We leave the exploration on multiple templates and verbalizers as future work.

4.2.2 Results.

We present the results of using the PromptRec approach with different PLM backbones on our Cold-Start Recommendation Benchmark in Table 2. We also report the results of several possible zero-shot solutions on the top with BERT (see description in Section 4.1.5). To verify the recommendation is personalized enough, we compute the statistical significance between the target model performance and the Random strategy performance by using Two Independent Samples T-Test [20] with a significant level $\alpha = 0.05$. Some observations can be made as below.

Baseline methods mostly fail to make personalized recommendations under the system cold-start setting. There is no baseline that consistently makes effective personalized recommendations on all datasets under the system cold-start setting. Some possible reasons are as follows. Essentially, personalized recommendation requires the models to capture the fine-grained differences between items from the same category. However, the two pre-training tasks (text representation and NSP) of EmbSim and PairNSP can only be used to distinguish coarse-grained semantics. Thus, they can not well handle the recommendation task. On the other hand, although ItemLM relies on a fine-grained pre-training task (MLM), it suffers from the linguistic bias, which is discussed in Section 3. This observation shows that it is very challenging to conduct system cold-start recommendations.

PromptRec could be generalized to various PLM families for system cold-start recommendation. PromptRec shows a significant improvement over the Random strategy with every LLM candidate on each dataset, indicating its strong generalization ability. Also, PromptRec improves the performance of the Random strategy on the three datasets by up to 6.93%, 14.01%, and 5.05% GAUC, respectively. This result demonstrates that large language models could make personalized recommendations with their strong in-context learning ability.

4.3 RQ2: Cold-Start Recommendation Performance is Sensitive to Model Sizes

This subsection examines the potential benefits of increasing PLM scales to improve the cold-start performance.

4.3.1 Settings.

We consider smaller language models from the BERT-family, including BERT-base [5] and BERT-small [39]. We reuse the designs of the templates, verbalizers, as well ass labelers described in Section 4.2.1.

4.3.2 Results.

Table 3 shows the results on the proposed benchmark and we summarize our observations as bellows.

Table 3. Sensitivity analysis of PLM sizes.

Family	#Params	ML-100K	Coupon	Restaurant
	29.1M	$52.10_{\pm 0.01}$	$50.50_{\pm 0.16}$	$49.29_{\pm 0.81}$
BERT	110M	$53.55_{\pm 0.01}$	$62.96_{\pm 0.18}$	$52.67_{\pm 1.28}$
	336M	$52.39_{\pm 0.01}$	$63.77_{\pm 0.18}$	$55.49_{\pm 1.41}$

Increasing the scales of PLMs could generally improve the cold-start performance. We observe that the BERT family displayed a gradual improvement in performance with an increase in model sizes.This result aligns with recent studies [48] on the relation between model sizes and zero-shot performance. It provide bright implications for addressing the cold-start problem with prompt learning.

Small language models could make personalized recommendations. We found it remarkable that BERT-small achieved a GAUC score of 52.10% on the Coupon dataset, with only 29.1M parameters, significantly outperforming the baselines. Thus, adopting model distillations to obtain low-latency cold-start recommender becomes promising.

5 RELATED WORK

5.1 Cold-start Recommendation

The phrase "cold-start" describes the situation when the recommender knows nothing about its serving objects. There are several cases of cold-start recommendation.

New-community cold-start recommendation. Making personalized recommendations on items to users without any user-item interactions as training data raises the new-community cold-start problem [2, 7, 23]. Recent works studied a sub-problem called "few-shot recommendation" [6, 9, 37], which estimates user interests according to their shopping behaviors during *online inferring*. However, this work studies the "system cold-start" problem, holding the assumption that there is no any user shopping data available even during online inferring, which is common and more challenging in many real-world scenarios [27, 33, 35]. To the best of our knowledge, there has yet to be any studies formalizing this issue, proposing mathematical methods, or setting up evaluation benchmarks.

New-user/item cold-start recommendation. When a recommender system is severing online stability, new items and users will still join day by day. Recommending these new items to users is the new item cold-start problem, similarly, recommending items to new users causes the new user cold-start problem, recommending new items to new users raises the new user-item cold-start problem [21, 23, 33, 35]. Different from the new system cold-start recommendation, the new user/item cold-start recommendation can train models on historical user-item interactions. Incorporating side information to enhance the quality of representing users/items is the most traditional way [12, 44, 50]. Under the assumption that side information is not available, directly mining the historical user-item interaction is much tractable [8, 21, 41]. Lately, pre-training graph neural networks is also considered as a frontier direction [14, 51].

5.2 Prompt Learning in Recommendation

Researchers introduce prompt learning into developing recommender systems for several purposes, including interpretability [9, 22], multi-tasks learning [11, 24, 52], sequential recommendation [4, 19, 29, 37, 47, 53], and conversational recommendation [9, 46]. The intuitions behind these studies can be considered as two folds: utilizing pre-trained language models as a knowledge base to enhance recommendation [4, 24, 37, 52, 53]; using language models as a bridge to realize the interaction between humans and systems based on natural language [9, 11, 19, 22, 29, 46, 47]. These studies inspire us to push prompt learning to a more challenging scenario, namely cold-start recommendation.

6 CONCLUSIONS AND FUTURE WORK

This paper studies the system cold-start recommendation problem, including providing a formal definition and the first benchmark. We propose a prompt learning approach called PromptRec for this problem and show that PLMs can make personalized recommendations without supervised fine-tuning. We encourage future research to explore the system cold-start setting for more recommendation tasks and hope they could be deployed in real-world businesses. There are several future directions to be further explored. First, how to optimize the design of template for better releasing the potential of PLMs? Second, how to choose the pre-trained language model from various available candidates? Third, how to strike a balance between the trade-off of model size and prompt learning performance? Finally, whether bias or privacy issues exist in PLM-based recommender systems and how to tackle these problems?

REFERENCES

[1] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow*. https://doi.org/10.5281/zenodo.5297715 If you use this software, please cite it using these metadata..

[2] JesúS Bobadilla, Fernando Ortega, Antonio Hernando, and Jesús Bernal. 2012. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-based systems* 26 (2012), 225–238.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[4] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-Rec: Generative Pretrained Language Models are Open-Ended Recommender Systems. *arXiv preprint arXiv:2205.08084* (2022).

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

[6] Hao Ding, Yifei Ma, Anoop Deoras, Yuyang Wang, and Hao Wang. 2021. Zero-shot recommender systems. *arXiv preprint arXiv:2105.08318* (2021).

[7] Sun Dong-ting, He Tao, and Zhang Fu-hai. 2012. Survey of cold-start problem in collaborative filtering recommender system. *Computer and modernization* 1, 201 (2012), 59.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*. PMLR, 1126–1135.

[9] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).

[10] Ruth Garcia and Xavier Amatriain. 2010. Weighted Content Based Methods for Recommending Connections in Online Social Networks. *Recommender Systems and the Social Web* (2010), 68.

[11] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). *arXiv preprint arXiv:2203.13366* (2022).

[12] Zhabiz Gharibshah, Xingquan Zhu, Arthur Hainline, and Michael Conway. 2020. Deep learning for user interest and response prediction in online display advertising. *Data Science and Engineering* 5, 1 (2020), 12–26.

[13] James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143, 1 (1982), 29–36.

[14] Bowen Hao, Jing Zhang, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Pre-training graph neural networks for cold-start users and items representation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 265–273.

[15] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[16] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[17] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[19] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large language models are zero-shot rankers for recommender systems. *arXiv preprint arXiv:2305.08845* (2023).

[20] Betty R Kirkwood and Jonathan AC Sterne. 2010. *Essential medical statistics*. John Wiley & Sons.

[21] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1073–1082.

[22] Lei Li, Yongfeng Zhang, and Li Chen. 2022. Personalized prompt learning for explainable recommendation. *arXiv preprint arXiv:2202.07371* (2022).

[23] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. 2014. Facing the cold start problem in recommender systems. *Expert systems with applications* 41, 4 (2014), 2065–2073.

[24] Junling Liu, Chao Liu, Peilin Zhou, Qichen Ye, Dading Chong, Kang Zhou, Yueqi Xie, Yuwei Cao, Shoujin Wang, Chenyu You, et al. 2023. Llmrec: Benchmarking large language models on recommendation task. *arXiv preprint arXiv:2308.12241* (2023).

[25] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.

[26] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *arXiv preprint arXiv:2103.10385* (2021).

[27] Marwa Hussien Mohamed, Mohamed Helmy Khafagy, and Mohamed Hasan Ibrahim. 2019. Recommender systems challenges and solutions survey. In *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*. IEEE, 149–155.

[28] Wentao Ouyang, Xiuwu Zhang, Shukui Ren, Li Li, Kun Zhang, Jinmei Luo, Zhaojie Liu, and Yanlong Du. 2021. Learning graph meta embeddings for cold-start ads in click-through rate prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in*

Information Retrieval. 1157–1166.

[29] Aleksandr V Petrov and Craig Macdonald. 2023. Generative Sequential Recommendation with GPTRec. *arXiv preprint arXiv:2306.11114* (2023).

[30] Yanru Qu, Ting Bai, Weinan Zhang, Jianyun Nie, and Jian Tang. 2019. An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation. In *Proceedings of the 1st international workshop on deep learning practice for high-dimensional sparse data*. 1–9.

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[32] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl. 2002. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*. 127–134.

[33] Al Mamunur Rashid, George Karypis, and John Riedl. 2008. Learning preferences of new users in recommender systems: an information theoretic approach. *Acm Sigkdd Explorations Newsletter* 10, 2 (2008), 90–100.

[34] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*. 521–530.

[35] Rachna Sethi and Monica Mehrotra. 2021. Cold start in recommender systems—a survey from domain perspective. In *Intelligent Data Communication Technologies and Internet of Things*. Springer, 223–232.

[36] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980* (2020).

[37] Damien Sileo, Wout Vossen, and Robbe Raymaekers. 2022. Zero-Shot Recommendation as Language Modeling. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part II*. Springer, 223–230.

[38] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[39] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *arXiv preprint arXiv:1908.08962* 13 (2019).

[40] Blanca Vargas-Govea, Gabriel González-Serna, and Rafael Ponce-Medellın. 2011. Effects of relevant contextual features in the performance of a restaurant recommender system. *ACM RecSys* 11, 592 (2011), 56.

[41] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A meta-learning perspective on cold-start recommendations for items. *Advances in neural information processing systems* 30 (2017).

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[43] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 417–426.

[44] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *The world wide web conference*. 2000–2010.

[45] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. 2017. A bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research* 18, 1 (2017), 2357–2393.

[46] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards Unified Conversational Recommender Systems via Knowledge-Enhanced Prompt Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1929–1937.

[47] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. 2023. Recmind: Large language model powered agent for recommendation. *arXiv preprint arXiv:2308.14296* (2023).

[48] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).

[49] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* (2019).

[50] Hongzhi Yin, Weiqing Wang, Hao Wang, Ling Chen, and Xiaofang Zhou. 2017. Spatial-aware hierarchical collaborative deep learning for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2537–2551.

[51] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2022. Self-Supervised Learning for Recommender Systems: A Survey. *arXiv preprint arXiv:2203.15876* (2022).

[52] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001* (2023).

[53] Yuhui Zhang, Hao Ding, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. 2021. Language Models as Recommender Systems: Evaluations and Limitations. In *I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop*.

[54] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

[55] Han Zhu, Junqi Jin, Chang Tan, Fei Pan, Yifan Zeng, Han Li, and Kun Gai. 2017. Optimized cost per click in taobao display advertising. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2191–2200.