2023 IEEE International Conference on Data Mining (ICDM) | 979-8-3503-0788-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDM58522.2023.00082

Collaborative Word-based Pre-trained Item Representation for Transferable Recommendation

Shenghao Yang[1,2,3,4], Chenyang Wang[1,2,3,4], Yankai Liu[4,6], Kangping Xu[1], Weizhi Ma[5], Yiqun Liu[1,2,3],
Min Zhang[1,2,3,4,✉], Haitao Zeng[4,6], Junlan Feng[6] and Chao Deng[6]
[1]*Department of Computer Science and Technology, Tsinghua University*
[2]*Quan Cheng Laboratory* [3]*Zhongguancun Laboratory*
[4]*THU-CMCC Joint Institute* [5]*AIR, Tsinghua University* [6]*China Mobile Research Institute*
{ysh21,xukp20}mails.tsinghua.edu.cn, thuwangcy@gmail.com,
{liuyankai,zenghaitao,fengjunlan,dengchao}@chinamobile.com, {mawz,yiqunliu,z-m}@tsinghua.edu.cn

***Abstract*—Item representation learning (IRL) plays an essential role in recommender systems, especially for sequential recommendation. Traditional sequential recommendation models usually utilize ID embeddings to represent items, which are not shared across different domains and lack the *transferable* ability. Recent studies use pre-trained language models (PLM) for item text embeddings (text-based IRL) that are universally applicable across domains. However, the existing text-based IRL is unaware of the important collaborative filtering (CF) information. In this paper, we propose CoWPiRec, an approach of Collaborative Word-based Pre-trained item representation for Recommendation. To effectively incorporate CF information into text-based IRL, we convert the item-level interaction data to a word graph containing word-level collaborations. Subsequently, we design a novel pre-training task to align the word-level semantic- and CF-related item representation. Extensive experimental results on multiple public datasets demonstrate that compared to state-of-the-art transferable sequential recommenders, CoWPiRec achieves significantly better performances in both fine-tuning and zero-shot settings for cross-scenario recommendation and effectively alleviates the cold-start issue. The code is available at: https://github.com/ysh-1998/CoWPiRec.**

***Index Terms*—Recommender System, Item Representation Learning, Transfer Learning**

I. Introduction

Item representation learning (IRL) is a crucial technology in recommender systems since items interacted by users largely reflect their preferences. IRL is especially important for sequential recommendation, where user representations are typically obtained by aggregating the representations of interacted items [1], [2]. Specifically, sequential recommender comprises two main components: the IRL module used to obtain item representations, and the sequence representation learning (SRL) module used to aggregate the representations of the chronologically-ordered items. Recent neural sequential recommendation models typically use an ID-based IRL module to map item IDs to hidden vectors and an SRL module with advanced neural networks, e.g., transformer layers [3]. Then the two modules are trained simultaneously with optimization objective of the next-item prediction task [1], [3]. Although promising results have been achieved, these methods heavily rely on rich ID-based interactions. When new scenarios arise, the models need to be trained from scratch since the ID embeddings are not shared across scenarios and may suffer the cold-start issue. Therefore, sequential recommendation models with ID-based IRL lack the *transferable* ability.

Recently, many content-based sequential recommendation models have been proposed to alleviate the above issue. Especially, considering the generalization of the text and the cross-scenario shared vocabulary, many works use the representation of item text instead of the ID embedding, i.e., text-based IRL. Due to the remarkable performance of pre-trained language model (PLM) [4] in neural language processing, existing works typically use PLM as the text-based IRL module. Specifically, these works obtain text-based item representations offline with PLM and feed the item representations into the SRL module. Then the SRL module is pre-trained on mixed-domain data to learn cross-domain general sequential representation patterns and the learned knowledge is transferred to a new domain, resulting in transferable sequential recommender [5], [6].

However, Although text-based item representations have effective semantic representation capabilities, they do not contain collaborative filtering (CF) information. In fact, some words that are not similar in semantics might be closely related in the context of recommendation. For example, “health” and “cycling” are two words that are not very close in terms of semantic representation space. While in the recommendation scenario, a user interested in healthy food may also prefer to buy some cycling equipment for exercise. To alleviate this issue, we argue that it is desired to incorporate CF-related signals into the text-based IRL. While most existing approaches focus on pre-training the SRL module and the PLM is frozen in training and unaware of important CF signals.

In this paper, we propose a **Co**llaborative **W**ord-based **P**re-trained **i**tem representation for **Rec**ommendation, **CoWPiRec**. Specifically, we extract word-level CF signals, i.e. co-click words, from user interaction history and construct a word graph to integrate these co-click relationships. Subsequently,

✉ Corresponding author.

This work is supported by the Natural Science Foundation of China (Grant No. U21B2026, 62002191) and Quan Cheng Laboratory (Grant No. QCLZD202301).

979-8-3503-0788-7/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDM58522.2023.00082

we design a novel word-level pre-training task to incorporate CF signals into PLM. The word graph serves as a CF-related knowledge source to instruct the pre-training procedure.

The merits of our proposed item representation learning approach are threefold. Firstly, since CoWPiRec is pre-trained independent of the SRL module, it is convenient to be integrated into different sequence aggregation networks as the text-based IRL module. Secondly, the item representation generated by CoWPiRec provides both effective semantic matching and CF-related signals, it can be used to perform recommendation tasks without any training stage when transferring to a new domain, i.e., zero-shot recommendation. Thirdly, CoWPiRec further achieves outperforming recommendation results with in-domain training utilizing the CF-related knowledge learned in pre-training.

We evaluate the effectiveness of CoWPiRec in the cross-scenario setting. We first use datasets from multiple domains to construct the word graph and pre-train CoWPiRec. Then, considering the efficiency in a new scenario, we utilize CoWPiRec as a feature extractor to offline generate item representations. The item representations can be used to perform downstream recommendations. The experiment results on the public datasets demonstrate that CoWPiRec outperforms state-of-the-art approaches in the zero-shot recommendation and further improves in-domain training effectiveness.

The main contributions of our work are summarized as follows:

- We propose a pre-trained item representation learning approach that aligns semantic and collaborative information for the recommendation.
- We design a novel pre-training task to incorporate word-level CF signals from the co-click word graph into the text-based IRL.
- Comparative experimental results on multiple public datasets demonstrate that our proposed approach achieves significantly better performances and effectively alleviates the cold-start issue.

II. Related Work

A. Sequential Recommendation

Sequential recommendation is a widely researched topic in the recommendation system community, with the objective of predicting the next item of a user's interaction history [2], [3]. Early studies are based on Markov chain assumptions to estimate the transition relationships between items [7], [8]. In recent years, with the development of deep learning, neural sequential recommendation models based on deep neural networks have emerged. These models usually comprise item representation learning (IRL) and sequence representation learning (SRL) modules to model the representation of item and user sequences. The SRL module utilizes various network structures, including Recurrent Neural Networks (RNN) [1], [9], [10], Convolutional Neural Networks (CNN) [11], Transformer [3], [12]–[15], and Graph Neural Networks (GNN) [16]–[18], to modeling the user sequence representation by aggregating the item representations. The item representations are obtained with the IRL module. Most IRL modules utilize item ID embedding to map item ID to a hidden vector [1], [3]. Limited by unshareable item IDs, these approaches with the ID-based IRL module lack transferable ability across scenarios. Different from relying on explicit item IDs, we represent items based on item text to enhance the transferable ability of sequential recommender.

B. Recommendation with Pre-trained Language Model

Inspired by the rapid development of the pre-trained language model (PLM), many recent works use PLM as the IRL module of the recommendation model [5], [6], [19]–[25]. With semantically enhanced item representations, these approaches achieve significant performance improvement in the recommendation and effectively alleviate the cold-start issue. These works can be divided into two main lines. One line is to perform joint training of PLM and the SRL module to adapt to the recommendation tasks. PLM-NR [20] utilizes PLM and an attention network to obtain item text representations. Then perform joint training on the SRL module and the last two layers of the PLM in the news recommendation. Due to the high computation complexity of PLM, another line is to generate item text representations offline with PLM. IDA-SR [24] utilizes PLM to obtain the item representations as input to the SRL module. Subsequently, three pre-training tasks are used to bridge the gap between text semantics and sequential user behaviors. Works of this line only train the SRL module and PLM is unaware of task-specific information, which leads to a suboptimal performance. Considering performance and efficiency tradeoffs, our approach trains PLMs in the pre-train stage to learn CF-related knowledge. When transferring to a new domain, we use the tuned PLMs to generate item representations offline, thus improving efficiency.

C. Transferable Recommendation Systems

Improving the transferable ability of recommender systems is a rapidly growing research area. It aimed at leveraging knowledge learned from multiple domains to enhance the performance of the recommendation model in new domains [26], [27]. Early studies typically assume the presence of commonalities across various domains, such as users with similar preferences [28]–[31] and common items [32], [33], to enable mapping between the source and target domains. Recent works have attempted to achieve transferable sequential recommender by learning cross-domain universal representations [5], [6], [22], [23], [34]. ZESRec [5] utilizes the universal item text representations obtained by PLM and performs the next item prediction task on the SRL module. The trained SRL module could transfer to a new domain with the item text representations as input. UniSRec [6] further adapt item text representations with an MoE module and enables the SRL module to learn a universal sequence pattern with the sequence-item and sequence-sequence contrastive pre-training tasks.

Most existing works focus on pre-training a transferable SRL module and the PLM is frozen. The item representation obtained by PLM can only provide semantics information and lacks CF-related signals, which limits the overall performance. To address this issue, we propose to incorporate recommendation signals into PLM via CF-related tasks. MoRec [25] is a recently proposed work with an idea close to ours. It performs a joint training of the PLM and the SRL module with a next-item-prediction task. However, since PLM is typically pre-trained with the word-level task, e.g., masked language modeling [4], the supervision signals of item-level recommendation tasks don't match PLM well. To align with the modeling strategy of PLM, we incorporate word-level CF signals into PLM through a word-level pre-training task.

III. Methodology

In this section, we present our proposed transferable item representation learning approach, CoWPiRec. Utilizing the word-level CF knowledge learned from the co-click word graph, CoWPiRec generates item representation with both semantic and CF-related information based on item text. When transferring to a new domain, the enhanced item representation could directly perform recommendations without training procedure and contribute to the in-domain training.

A. Framework Overview

The overall framework of our proposed text-based IRL approach is shown in Figure 1. Text-based IRL approach utilizes item text representation generated by PLM to replace the ID-based item representation of traditional sequential recommendation models. It has achieved promising transferable recommendation performance combined with a pre-training scheme on the SRL module [5], [6]. We argue that these transferable recommenders are suboptimal since the text-based IRL modules are unaware of CF-related information and it is desired to incorporate CF-related signals into PLM.

Considering PLM is typically trained with the word-level task, the item-level next-item-prediction task is not applicable to integrate CF signals into PLM. Therefore, we first extract word pairs with co-click relationships from interaction data and construct a word graph that contains these relationships. The co-click relationships between these words can be seen as word-level CF signals. Then we incorporate the word-level CF signals from the word graph into PLM through a word-level pre-training task. We will explain each key component of our proposed approach in the following sections.

B. Word Graph Construction

In this section, we present the process of extracting co-click words and constructing the word graph. A sub-graph of our constructed word graph is shown in Figure 1 (a). The co-click relationship is a common concept in recommender systems while previous works mostly focus on item-level co-click relationships. To align with the modeling format of PLM and incorporate the recommendation signal more effectively, we extract the word-level co-click relationships from the item text.

In different recommendation scenarios, although items have different presentation formats, they usually have basic textual descriptions. Due to the universality of language, different domains share a common vocabulary, making the text bridge different recommendation scenarios. Additionally, item texts often contain some word-level user preferences. If a user clicks several items containing words like "health" or "fitness", it indicates that this user may be focused on a healthy lifestyle. Therefore, the user may be also interested in nutritionally balanced food or some fitness equipment. These items may contain words such as "balance", "exercise" and "cycling".

We construct a word graph to organize the co-click relationships based on user interaction. For each word, a candidate set of words is generated based on co-click relationships and then filtered to retain only the top N words as neighboring nodes.

Specifically, given a user's interaction sequence $s = \{i_1, i_2, i_t, ..., i_n\}$, where i_t represents the t-th item in the sequence. A co-click word pair is defined as two words from each item text respectively, denoted as w_i and w_j. We count the occurrences of all co-click word pairs, denoted as $(w_i, w_j, c_{ij}, c_{ji})$, where $c_{ij} = c_{ji}$. Since each word contains a large number of co-click words, we follow [35] and filter the candidate co-click words using the tf-idf algorithm. The tf-idf value of a pair of co-click words is calculated by Equation (1):

$$tf_{i,j} = \frac{c_{i,j}}{\sum_{k=1}^{V} c_{i,k}}, \quad idf_j = \lg \frac{V}{|\{c_{k,j} \mid \forall k, c_{k,j} > 0\}|}, \qquad (1)$$
$$tf\text{-}idf_{i,j} = tf_{i,j} \times idf_j,$$

where V is the vocabulary size, and the denominator of idf is the number of words that have the co-click relationship with w_j. The higher the tf-idf value, the more times w_j and w_i are co-clicked and the less w_j is co-clicked with other words. For each w_i, only the top N words with the highest tf-idf values will be selected as its neighbor nodes in the word graph.

By constructing edges between co-click word pairs, we obtain a word graph fusion of word-level CF signals. We construct the word graph based on the user interaction data of multiple domains to improve the generality ability of extracted word-level CF signals. The word pairs with edges in the word graph may relate to different domains, e.g. "health" and "balance" in the "Food" domain, "cycling", "indoor" and "exercise" in the "Home" domain, as shown in Figure 1 (a).

C. Word Graph-based Pre-training Task

With the remarkable semantic representation ability of PLM, text-based IRL based on PLM provides an effective semantic matching ability. While PLM cannot capture CF-related information and this limits the representation ability of text-based IRL. To incorporate the recommendation signal into PLM, an intuitive idea is to train PLM and SRL simultaneously via the next item prediction task, thus introducing task-specific information into PLM. However, since PLM's modeling method on large-scale corpora is word-level, the

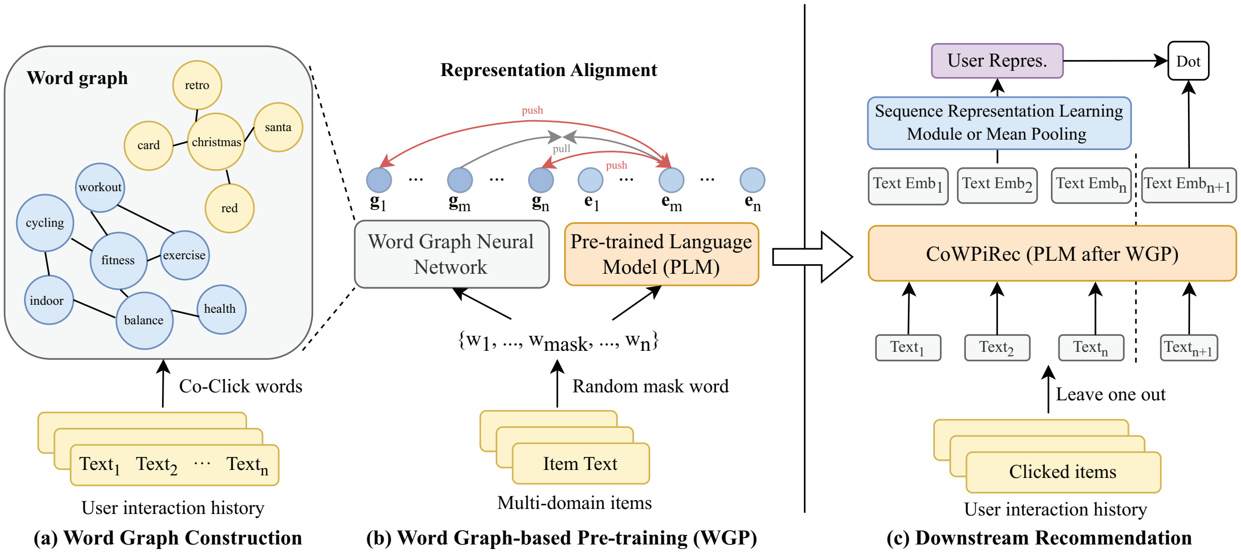

Fig. 1. The overall framework of our proposed collaborative word-based pre-trained item representation for recommendation (CoWPiRec). (a) The word-level collaborative filtering (CF) signals are from the co-click relationships of word pairs in the word graph. (b) A word graph-based pre-training (WGP) is performed to align the semantic- and the CF-related representation of the PLM and word graph with contrastive learning. w_i denotes the word in item text, $\mathbf{g}_i$ and $\mathbf{e}_i$ is the word representation after word graph modeling and semantic modeling (c) When transferring CoWPiRec to a new domain, the item representations generated offline utilizing CoWPiRec are fed into a sequence representation learning (SRL) module or a simple mean-pooling to perform downstream recommendations in the fine-tuning and zero-shot settings, respectively.

above item-level supervision signal cannot be well integrated into PLM.

Considering many works have demonstrated that aligning with PLM's modeling format in downstream tasks can better inspire its learned knowledge [36], we propose a word-level pre-training task to incorporate the word-level CF information contained in the word graph into the PLM, as shown in Figure 1 (b). Specifically, we use item text as the input of the PLM and add special symbols [CLS] and [SEP] before and after the input in accordance with the input format of the PLM. We randomly mask words in the item text using the [MASK] special symbol. For an item text input $i = \{cls, w_1, ..., w_m, ..., w_n, sep\}$, where w_m is the masked word, the initialize word embedding of each word is obtained with PLM's word embedding, i.e., $\{\mathbf{v}_{cls}, \mathbf{v}_1, ..., \mathbf{v}_m, ..., \mathbf{v}_n, \mathbf{v}_{sep}\}$, where $\mathbf{v}_i \in \mathbb{R}^d$ and d is the dimension of word embedding. Then two different modeling procedures are performed for input word embedding, namely *semantic modeling* and *word graph modeling*.

1) Semantic Modeling: In this modeling procedure, the word embedding of each word in item text $\mathbf{v}_i$ is firstly concatenated, i.e., $\mathbf{x} = [\mathbf{v}_{cls}; \mathbf{v}_1; ...; \mathbf{v}_m; ...; \mathbf{v}_n; \mathbf{v}_{sep}] \in \mathbb{R}^{n\times d}$, where n is the input length and we omit the special token at the head and tail for convenience. $[;]$ is the concatenation operation. Then $\mathbf{x}$ is fed into the L-layer Transformer encoder of the PLM. Each Transformer encoder consists of a multi-head self-attention layer and a position-wise feed-forward layer. A residual connection and layer normalization are performed in the above two parts. We set $\mathbf{x}^0 \in \mathbb{R}^{n\times d}$ as the input, and the output after $l+1$-layer Transformer encoder is obtained by Equation (2).

$$\begin{aligned}\mathbf{x}^{l+1} &= Trm(\mathbf{x}^l) = LN(\mathbf{s}^l + FFN(\mathbf{s}^l)),\\ \mathbf{s}^l &= LN(\mathbf{x}^l + MHAttn(\mathbf{x}^l)),\end{aligned} \tag{2}$$

where $Trm(\cdot)$ is the Transformer encoder layer, $LN(\cdot)$ is the layer normalization function, $FFN(\cdot)$ is the position-wise feed-forward layer, $MHAttn(\cdot)$ is the multi-head self-attention layer, $\mathbf{s}^l \in \mathbb{R}^{n\times d}$ is the output of the multi-head self-attention layer. The output of the last layer is $\mathbf{x}^L = [\mathbf{e}_{cls}; \mathbf{e}_1; ...; \mathbf{e}_m; ...; \mathbf{e}_n; \mathbf{e}_{sep}] \in \mathbb{R}^{n\times d}$.

With the self-attention mechanism of the Transformer Encoder, $\mathbf{e}_i \in \mathbb{R}^d$ integrates the contextual information of other words in the item text, which demonstrates effective semantic representation ability in many tasks. While in the recommendation system, semantic similarity and recommendation relevance are not related, so the PLM is expected to capture additional recommendation signals to improve the recommendation performance.

2) Word Graph Modeling: In this part, the representation of each word in input $\mathbf{x}$ is obtained by aggregating the embedding of its neighboring nodes through a graph neural network (GNN). Specifically, we follow [35] and use the GraphSAGE algorithm [37] to learn a function for aggregating neighbor node representations.

$$\mathbf{h}_i^t = \sigma\left(\mathbf{W}_g\left(\mathbf{h}_i^{t-1} \oplus \text{AGG}\left(\{\mathbf{h}_j^{t-1}, \forall w_j \in \mathcal{N}_{w_i}^*\}\right)\right)\right), \tag{3}$$

where $\mathbf{h}_i^t \in \mathbb{R}^d$ is the representation of central word w_i in the t-th layer of GNN, which is aggregated with the representation of itself $\mathbf{h}_i^{t-1}$ and its neighbors $\mathbf{h}_j^{t-1}$ in the $t-1$ layer. The initial representation of each word is the initialized word embedding, i.e., $\mathbf{h}_i^0 = \mathbf{v}_i$. σ is a non-linear activation function. $W_g \in \mathbb{R}^{d\times 2d}$ is the weight of a linear layer. $\mathcal{N}_{w_i}^*$ is the sampled neighbors. $\oplus$ is a concatenate operator. AGG is an aggregating function based on the attention mechanism. It aggregates the representation of neighbors with Equation 4.

$$\mathbf{q}_g^t = \sigma\left(\sum_{w_j\in\mathcal{N}_{w_i}^*} Q_g \mathbf{h}_j^{t-1}\right), \quad \mathbf{k}_j^t = \sigma\left(K_g \mathbf{h}_j^{t-1}\right)$$
$$a_j^t = \frac{\exp\left(\mathbf{q}_g^{tT}\boldsymbol{k}_j^t\right)}{\sum_{w_k\in\mathcal{N}_{w_i}^*}\exp\left(\mathbf{q}_g^{t^T}\boldsymbol{k}_k^t\right)}, \quad \mathbf{h}_{\mathcal{N}_{w_i}^*}^t = \sum_{w_j\in\mathcal{N}_{w_i}^*} a_j^t \mathbf{h}_j^{t-1}, \tag{4}$$

where $Q_g, K_g \in \mathbb{R}^{d\times d}$ is the weight of the projection layer and a_j^t is the attention weight of each neighbor. The output after T layers of GNN is

$$\mathbf{g}_i = \mathbf{h}_i^T = \sigma(W_g(\mathbf{h}_i^{T-1} \oplus \mathbf{h}_{\mathcal{N}_{w_i}^*}^{T-1})). \tag{5}$$

The central word representation $\mathbf{g}_i \in \mathbb{R}^d$ aggregated with co-click words is fused with the word-level CF signal of the word graph.

3) Representation Alignment: In order to incorporate the word-level CF signals extracted from the word graph into the representation space of PLM, we adopt a widely used contrastive learning method to align the semantic representation of PLM $\mathbf{e}_i \in \mathbb{R}^d$ with the CF-related representation of word graph $\mathbf{g}_i \in \mathbb{R}^d$. Specifically, for a masked word w_m in the input, we obtain its representations of PLM and word graph, i.e., $\mathbf{e}_m \in \mathbb{R}^d$ and $\mathbf{g}_m \in \mathbb{R}^d$. We treat them as a positive pair and treat $\mathbf{g}_i$ of other words in the same input $(i \neq m)$ as negatives. We aim to pull $\mathbf{e}_m$ and $\mathbf{g}_m$ closer and push $\mathbf{e}_m$ away from other $\mathbf{g}_i$ by minimizing the following contrastive learning loss:

$$\mathcal{L} = -\frac{1}{M}\sum_{m=1}^{M}\log\frac{\exp\left(\mathbf{e}_m\cdot\mathbf{g}_m/\tau\right)}{\sum_{i=0}^{n}\exp\left(\mathbf{e}_m\cdot\mathbf{g}_i/\tau\right)}, i \neq m, \tag{6}$$

where M is the number of masked words of the input item text.

It is worth noting that, during the training process, there is a parameter sharing between the word embedding of the PLM and the node embedding of the word graph. As a result, the output of a word in the PLM gradually approaches its aggregated representation of neighbor nodes in the word graph. This process results in the PLM's output containing both semantic information and word-level CF information. We refer to this recommendation-orient trained PLM as CoWPiRec.

D. Downstream Recommendation

Through constructing word graphs and pre-training on multiple domains, we obtain a text-based IRL module, CoWPiRec, that captures word-level CF signals. When transferring to a new domain, we consider two settings to evaluate the effectiveness of CoWPiRec: *fine-tuning setting* and *zero-shot setting*. The downstream recommendation pipeline is shown in Figure 1 (c).

1) Fine-tuning Setting: In this setting, we train a sequential recommendation model using all training data in the new domain. Following the standard pipeline, given a user's click sequence $s = \{i_1, i_2, ..., i_n\}$, for each $i_t = \{w_1, w_2, ..., w_n\}$, it is fed into CoWPiRec after adding special symbols [CLS] and [SEP]. The item representation is obtained by Equation (7).

$$\mathbf{i}_t = CoWPiRec([cls; w_1; w_2; ...; w_n; sep]), \tag{7}$$

where $CoWPiRec(\cdot)$ takes the representation of the [cls] position as the item representations $\mathbf{i}_t \in \mathbb{R}^d$. Then we follow [6] and used an MoE module consisting of multiple whitening networks to adapt the item representations and reduce the dimension, resulting $\widetilde{\mathbf{i}}_t \in \mathbb{R}^{d_V}$.

We adopt a widely used transformer network to aggregate the item representations. Specifically, we sum the item representations and the absolute position embedding $\mathbf{p}_t \in \mathbb{R}^{d_V}$ as the input.

$$\mathbf{f}_t^0 = \widetilde{\mathbf{i}}_t + \mathbf{p}_t. \tag{8}$$

Then $\mathbf{F}^0 = [\mathbf{f}_1^0; ...; \mathbf{f}_n^0] \in \mathbb{R}^{n\times d_V}$ is fed into L transformer layers, the output after $l+1$ layers is:

$$\mathbf{F}^{l+1} = FFN(MHAttn(F^l)). \tag{9}$$

We take the t-th position hidden state of the last layer, i.e., $\mathbf{f}_n^L \in \mathbb{R}^{d_V}$ as the user representation $\mathbf{u} \in \mathbb{R}^{d_V}$.

Note that since CoWPiRec already has the ability to capture recommendation signals, we don't need to update the parameters of CoWPiRec during training. Therefore we offline obtain all item representations, which significantly improves efficiency. For user representation $\mathbf{u}$, we calculate the score of candidate next item i_{t+1} using the dot product:

$$score_{(i_{t+1}|s)} = Softmax(\mathbf{u}\cdot\widetilde{\mathbf{i}}_{t+1}). \tag{10}$$

We use the cross-entropy loss for the next item prediction task during training. In the inference stage, we rank the items based on the dot product score.

2) Zero-shot Setting: In contrast to the cold-start problem, the objective of zero-shot recommendation is to determine whether a model has basic recommendation capabilities without any in-domain training. It can not be achieved with traditional ID-based recommendation models. Since the item representations generated by CoWPiRec have a remarkable semantic matching ability and could capture recommendation signals. Therefore, we directly use the nearest neighbor search with the dot product to perform the recommendation. Specifically, given all item representations in a user sequence $\{\mathbf{i}_1, \mathbf{i}_2, ..., \mathbf{i}_n\}$ obtained by CoWPiRec with Equation (7). We use mean-pooling to aggregate the item representations to obtain the user representation $\mathbf{u}$.

$$\mathbf{u} = \frac{1}{n}\sum_{t=1}^{n}\mathbf{i}_t. \tag{11}$$

Then the score of the candidate item i_{t+1} is calculated with Equation (10) and we directly predict the next item according to the scores.

TABLE I
COMPARISON OF DIFFERENT SEQUENTIAL RECOMMENDERS.

Methods	Used information		Pre-training on		Transferable
	ID	Text	Item	Sequence	
SASRec	✓	✗	✗	✗	✗
BERT4Rec	✓	✗	✗	✓	✗
S3Rec	✓	✓	✗	✓	✗
ZESRec	✗	✓	✗	✓	✓
UniSRec	✗	✓	✗	✓	✓
MoRec	✗	✓	✓	✓	✓
CoWPiRec	✗	✓	✓	✗	✓

E. Discussion

In this section, we present the differences between our proposed CoWPiRec compared with other sequential recommendation models. The comparison focuses on the two components of sequential recommendation models, i.e., the IRL and SRL modules, and the model's transferable ability. The comparison results are shown in Table I.

ID-based IRL approaches such as SASRec [3] and BERT4Rec [14] obtain item representations with explicit item IDs. SASRec utilizes transformer layers to aggregate item ID representations and BERT4Rec performs a mask item prediction task to pre-train the bidirectional transformer layer. Since item IDs are not shared across scenarios, these approaches need to be trained from scratch when applied to new scenarios and lack transferable ability. CoWPiRec does not rely on the item ID to perform recommendations and adopt a text-based IRL module. With the shared vocabulary across scenarios, CoWPiRec achieves transferable recommendations.

Text-based IRL approaches such as S^{3}Rec [38] incorporate item text representation as an auxiliary feature and perform self-supervised tasks to integrate the representation of sequence, item, and feature. Since S^{3}Rec also utilizes the item id embedding, the pre-train task can only be performed in-domain. Different from S^{3}Rec, ZESRec [5] and UniSRec [6] purely use item text representations and perform a cross-domain pre-training on the SRL module. The pre-trained SRL module can learn general sequence modeling patterns and contribute to the cross-scenario recommendations. Instead of focusing only on pre-training the SRL module, MoRec [25] train the text-based IRL and SRL module jointly with the next-item-prediction task. We don't pre-train the SRL module in our proposed approach and perform a word graph-based pertraining task to obtain a transferable text-based IRL module, i.e., CoWPiRec.

IV. EXPERIMENTS

In this section, we first introduce how to evaluate the transferable ability of CoWPiRec in cross-scenario settings and then present experimental results and analysis.

TABLE II
STATISTICS OF THE DATASETS AFTER PREPROCESSING. "AVG. n" DENOTES THE AVERAGE LENGTH OF ITEM SEQUENCES. "AVG. c" DENOTES THE AVERAGE NUMBER OF TOKENS IN THE ITEM TEXT.

Datasets	#Users	#Items	#Inters.	Avg. n	Avg. c
Pre-trained	1,361,408	446,975	14,029,229	13.51	139.34
- Food	115,349	39,670	1,027,413	8.91	153.40
- CDs	94,010	64,439	1,118,563	12.64	80.43
- Kindle	138,436	98,111	2,204,596	15.93	141.70
- Movies	281,700	59.203	3,226,731	11.45	97.54
- Home	731,913	185,552	6,451,926	8.82	168.89
Scientific	8,442	4,385	59,427	7.04	182.87
Pantry	13,101	4,898	126,962	9.69	83.17
Instruments	24,962	9,964	208,926	8.37	165.18
Arts	45,486	21,019	395,150	8.69	155.57
Office	87,436	25,986	684,837	7.84	193.22
Online Retail	16,520	3,469	519,906	26.90	27.80

A. Experiment Setup

1) Datasets: We use mixed-domain user interaction data to pre-train CoWPiRec, and then use multiple downstream datasets to evaluate the transferable performance of CoWPiRec. The statistics of the dataset used are shown in Table II.

- **Pre-trained datasets**: We select the datasets from five domains in the Amazon dataset [39] to construct the word graph and pre-train CoWPiRec, i.e., "*Grocery and Gourmet Food*", "*Home and Kitchen*", "*CDs and Vinyl*", "*Kindle Store*" and "*Movies and TV*".
- **Downstream datasets**: In the downstream recommendation task, we select another five datasets in the Amazon dataset as cross-domain datasets, namely "*Industrial and Scientific*", "*Prime Pantry*", "*Musical Instruments*", "*Arts, Crafts and Sewing*", and "*Office Products*". We also select a cross-platform dataset, namely *Online Retail*[1], a UK online shopping dataset containing transaction records between 01/12/2010 and 09/12/2011.

For all datasets, we remove users and items with fewer than five interactions and arrange the items interacted by users in chronological order following [6]. For item text, we use title, categories, and brand in the Amazon dataset, and item description in the *Online Retail* dataset.

2) Baselines: In this paper, we compare CoWPiRec with several baseline methods, including:

- **SASRec** [3] uses the self-attention mechanism to aggregate ID-based item representations in the user sequence.
- **BERT4Rec** [14] models user sequence representations based on cloze objective task.
- **SASRec$_\mathbf{T}$** simply replaces the item ID embedding of SASRec with the item text embedding generated by PLM and maintains the same SRL module.
- **S^{3}Rec** [38] pre-trains SRL modules with four self-supervised tasks on in-domain data to integrate representations at different levels of features, items, and sequences.

[1] https://www.kaggle.com/carrie1/ecommerce-data

TABLE III

Downstream recommendation performance of different models in the fine-tuning setting. The best and the second-best performances are denoted in bold and underlined fonts, respectively. "H@K" is short for "HR@K" and "N@K" is short for "NDCG@K", respectively. The subscript 'T' denotes that item text is used in the IRL module of the model. The superscripts * and ** indicate $p \leq 0.05$ and $p \leq 0.01$ for the paired t-test of CoWPiRec vs. the best baseline.

Setting		Baselines							Ours	
Dataset	Metric	SASRec	BERT4Rec	S^3Rec_T	$SASRec_T$	$ZESRec_T$	$UniSRec_T$	$MoRec_T$	$CoWPiRec_T$	Improv.
Scientific	H@10	0.1063	0.0488	0.0897	0.1163	0.1066	0.1124	0.1174	**0.1264****	+7.67%
	H@50	0.2034	0.1185	0.1913	0.2259	0.2095	0.2284	0.2300	**0.2388****	+3.83%
	N@10	0.0552	0.0243	0.0496	0.0631	0.0582	0.0595	0.0635	**0.0664****	+4.57%
	N@50	0.0763	0.0393	0.0716	0.0870	0.0808	0.0847	0.0880	**0.0909****	+3.30%
Pantry	H@10	0.0493	0.0267	0.0393	0.0603	0.0629	0.0646	0.0639	**0.0679****	+5.11%
	H@50	0.1333	0.0932	0.1275	0.1676	0.1658	0.1747	0.1682	**0.1783***	+2.06%
	N@10	0.0219	0.0136	0.0177	0.0295	0.0308	0.0309	0.0310	**0.0320****	+3.23%
	N@50	0.0399	0.0277	0.0366	0.0528	0.0531	0.0546	0.0535	**0.0559***	+2.38%
Instruments	H@10	0.1126	0.0788	0.0996	0.1175	0.109	0.1087	0.1229	**0.1270****	+3.34%
	H@50	0.2087	0.1485	0.1886	0.2224	0.2044	0.2079	0.2278	**0.2344****	+2.90%
	N@10	0.0618	0.0579	0.0623	0.0690	0.0649	0.0622	0.0717	**0.0735****	+2.51%
	N@50	0.0826	0.0728	0.0815	0.0917	0.0855	0.0837	0.0944	**0.0967****	+2.44%
Arts	H@10	0.1074	0.0647	0.0952	0.1078	0.1010	0.1099	0.1101	**0.1164****	+5.72%
	H@50	0.1986	0.1316	0.1815	0.2050	0.1934	0.2118	0.2127	**0.2231****	+4.89%
	N@10	0.0571	0.0403	0.0567	0.0613	0.0568	0.0602	0.0637	**0.0650****	+2.04%
	N@50	0.0769	0.0548	0.0754	0.0825	0.0769	0.0823	0.0860	**0.0882****	+2.56%
Office	H@10	0.1064	0.0794	0.1085	0.1043	0.0955	0.1046	0.1096	**0.1141****	+4.11%
	H@50	0.1641	0.1232	0.1683	0.1709	0.1625	0.1751	0.1794	**0.1867****	+4.07%
	N@10	**0.0710**	0.0573	0.0666	0.0640	0.0567	0.0627	0.0673	0.0703	-
	N@50	0.0835	0.0668	0.0797	0.0785	0.0714	0.0780	0.0825	**0.0861****	+3.11%
Online Retail	H@10	0.1460	0.1343	0.1433	0.1366	0.1320	0.1444	0.1465	**0.1515****	+3.41%
	H@50	0.3872	0.3582	0.3762	0.3479	0.3378	0.3653	0.3728	**0.3928****	+1.45%
	N@10	0.0671	0.0645	0.0639	0.0666	0.0628	0.0675	0.0712	**0.0723****	+1.54%
	N@50	0.1201	0.1133	0.1146	0.1129	0.1077	0.1158	0.1204	**0.1247****	+3.57%

- **ZESRec** [5] obtains item representations using PLM firstly. Then pre-trains the SRL module on data from multiple domains and transfers it to new domains.
- **UniSRec** [6] also obtains item representations using PLM and uses an MoE module to adaptively adjust the representations in different domains. Then the MoE and SRL modules are pre-trained on multi-domain datasets with sequence-item and sequence-sequence contrastive learning tasks.
- **MoRec** [25] performs a joint training on PLM and SRL module with next-item-prediction task. With the item-level supervision signals, the tuned PLM could better adapt to the recommendation task.

Among all the above methods, SASRec and BERT4Rec are ID-based IRL methods. $SASRec_T$, ZESRec, UniSRec, MoRec, and our proposed CoWPiRec belong to the text-based IRL methods. Different from most baselines, CoWPiRec only pre-trains the IRL module by constructing a word graph containing word-level CF signals and performing a word graph-based pre-training task on datasets from multiple domains. Note that we don't compare CoWPiRec with the cross-domain recommendation models since it has been proven that these approaches usually underperform one of our baselines, i.e., UniSRec [6].

3) Evaluation Metric: We use two widely used evaluation metrics, HR@K and nDCG@K, to evaluate the performance of all models in the next item prediction task on downstream datasets. K is set to 10 and 50. Following previous work [3], we use the leave-one-out method to construct the dataset. Specifically, given a user interaction sequence, the last item is used for testing, the second to last item is used for validation, and the rest of the items are used for training. When predicting the next item, we sort all items in the dataset based on the dot-product score. The reported evaluation metrics are the average values of all test users.

4) Implementation Details: We implement CoWPiRec using RecBole [40] and transformers [41] library. For baseline methods, most are implemented by RecBole and we run MoRec with official code[2]. During the pre-training stage of CoWPiRec, we construct the word graph by retaining the top 30 co-click words based on their tf-idf scores. Item text is tokenized using the BERT tokenizer and we set the maximum length of all item texts to 128. Following the BERT masking strategy, we randomly select 15% of words in the input sequence and replace them with the [MASK] token in 80% of cases, a random token in 10% of cases, and leaving them unchanged in 10% of cases. In the word graph modeling step, the number of GNN layers T in the GraphSAGE algorithm is set to 1. We use an official checkpoint of BERT in the hugging-face hub, i.e., *bert-base-uncased*[3] to initialize CoWPiRec's

[2] https://github.com/westlake-repl/IDvs.MoRec

[3] https://huggingface.co/bert-base-uncased

parameters. We pre-train CoWPiRec with a batch size of 100 and a learning rate of 5e-5 and use the AdamW optimizer with a linear warm-up rate of 0.1 to update model parameters. CoWPiRec is trained for 30 epochs on one Nvidia RTX 3090.

In the fine-tuning setting of the CoWPiRec, we followed [6] and set the number of whitening networks of the MoE module to 8. The number of transformer layers and the head of the multi-head self-attention layer in the SRL module are both set to 2. For all methods in the downstream recommendation, we use the Adam optimizer and carefully search for hyperparameters, with a batch size of 2048 and early stopping with the patience of 10, using nDCG@10 as the indicator. We tune the learning rate in $\{0.0003, 0.001, 0.003, 0.01\}$ and the embedding dimension in $\{64, 128, 300\}$.

B. Overall Performance

1) Fine-tuning Setting: We compare the performance of CoWPiRec with multiple baseline models on five cross-domain datasets and a cross-platform dataset, and the experimental results are shown in Table III.

From the results, several observations could be concluded. Firstly, Among several baseline methods with ID-based IRL, SASRec achieves better performance when interactions are sufficient while performing poorly on datasets with relatively fewer interactions, e.g., Scientific. It indicates that the sequential recommender with ID-based IRL heavily relies on ID-based interactions. Secondly, The methods with the text-based IRL module effectively improve the performance, especially in datasets that the ID-based model does not specialize in. Thirdly, with effective joint training on the PLM and the SRL module, MoRec achieves overall better results than other baselines. It indicates the significance to enable PLM aware task-specific signals. While limited by the unsuitable item-level task, the overall performance of MoRec is suboptimal compared to our proposed CoWPiRec.

Compared to all baseline models, it is clear that CoWPiRec achieves the best performance in almost all cases. That demonstrates the effectiveness of incorporating word-level CF signals into the text-based IRL module. It is worth noting that CoWPiRec trains the MoE module and SRL module from scratch in fine-tuning stage, unlike UniSRec which pre-trains these two modules with mix-domain datasets. It indicates that the superior result of our model mainly comes from the pre-trained text-based IRL module's ability to capture CF-related information.

2) Zero-shot Setting: For transferable sequential recommenders, the zero-shot performance after transferring to a new domain intuitively reflects the knowledge learned in pre-training. Following the zero-shot recommendation setting in [5], we directly use the pre-trained checkpoint of transferable sequential recommenders to perform recommendations without any training stage. Note that in this setting, the model can access all interactions of the user except the last item in the user sequence, but no next-item prediction task training is performed to update the model's parameters. The experiment results are shown in Table IV. From the results, we can conclude several observations. Firstly, S^3Rec performs poorly in the zero-shot setting. We speculate the reason is that the modeling procedure of S^3Rec's SRL module is different in pre-training and downstream, i.e., bidirectional and unidirectional. Secondly, ZESRec, UniSRec, and MoRec perform better than S^3Rec, which demonstrates that the pre-training stage contributes to the zero-shot recommendation performance. Thirdly, CoWPiRec gives clearly better results than other baselines in most cases. Note that CoWPiRec is not pre-trained with a recommendation-related task, e.g. next-item-prediction task, It indicates the effectiveness of word graph-based pre-training. We believe that the significant improvement of CoWPiRec benefits from the word-level CF knowledge learned from the word graph.

TABLE IV
ZERO-SHOT RECOMMENDATION PERFORMANCE OF DIFFERENT MODELS ON THE DOWNSTREAM DATASETS. THE BEST AND THE SECOND-BEST PERFORMANCES ARE DENOTED IN BOLD AND UNDERLINED FONTS, RESPECTIVELY. S^3REC IS PRE-TRAINED WITH THE SAME DATASETS AS DOWNSTREAM AND OTHER MODELS ARE PRE-TRAINED WITH AMAZON PRE-TRAINED DATA. THE SUPERSCRIPTS * AND ** INDICATE $p \leq 0.05$ AND $p \leq 0.01$ FOR THE PAIRED T-TEST OF COWPIREC VS. THE BEST BASELINE.

Dataset	Metric	ZESRec	S^3Rec	UniSRec	MoRec	CoWPiRec
Scientific	H@10	0.0519	0.0025	<u>0.0553</u>	0.0481	**0.0614****
	H@50	0.1063	0.0158	<u>0.1149</u>	0.0943	**0.1228****
	N@10	<u>0.0284</u>	0.0011	0.0281	0.0222	**0.0287***
	N@50	0.0403	0.0039	<u>0.0411</u>	0.0324	**0.0422****
Instruments	H@10	<u>0.0356</u>	0.0079	0.0299	<u>0.0356</u>	**0.0429****
	H@50	0.0738	0.0213	**0.0846**	0.0649	<u>0.0830</u>
	N@10	<u>0.0187</u>	0.0045	0.0148	0.0178	**0.0198****
	N@50	<u>0.0271</u>	0.0072	0.0265	0.0241	**0.0286****
Online Retail	H@10	<u>0.0375</u>	0.0065	0.0369	0.0331	**0.0440****
	H@50	0.0780	0.0421	<u>0.0814</u>	0.0792	**0.1011****
	N@10	<u>0.0180</u>	0.0028	0.0177	0.0153	**0.0191****
	N@50	0.0268	0.0102	<u>0.0273</u>	0.0253	**0.0316****

C. Cold Start Performance

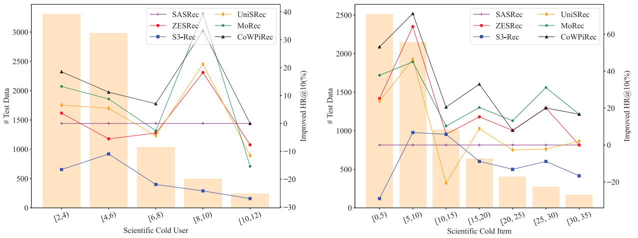

Fig. 2. Performance comparison in cold user and cold item experiment on "Scientific" dataset. The bar graph represents the number of users or items in test data for each group. The line chart represents the improvement ratios for HR@10 compared with SASRec.

One goal of the transferable sequential recommender is to alleviate the cold start issue in new domains. We evaluate CoWPiRec's performance compared to baseline models on the cold start setting from two perspectives: cold users and cold items. Specifically, for cold user experiments, we group the

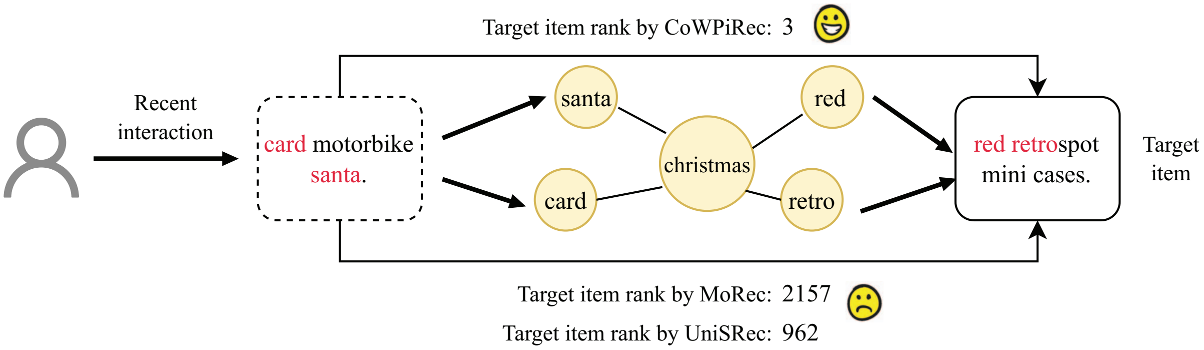
Target item rank by CoWPiRec: 3
Recent interaction
card motorbike santa.
santa
card
christmas
red
retro
red retrospot mini cases.
Target item
Target item rank by MoRec: 2157
Target item rank by UniSRec: 962

Fig. 3. The interaction history of a user in the "Online Retail" dataset, a sub-graph of our constructed word graph, and the rank results of the target item of models in the zero-shot setting. The word "card" and "santa" have co-click relationships with "retro" and "red" in the word graph. CoWPiRec utilizes the word-level CF signal learned from the word graph and captures "red" and "retro" in the target item. Therefore, CoWPiRec ranks the target item at a high position and achieves a clearly better performance than other models.

users in the test set based on the number of their interactions in the training set. For cold item experiments, we split the test set based on the target item's popularity in the training set. We present the relative improvement of CoWPiRec and several baselines over SASRec in terms of HR@10, as shown in Figure 2.

From the result, several observations can be concluded. Firstly, CoWPiRec achieves the most improvement over SASRec in most user groups while other baseline models underperform SASRec in some groups. Secondly, in the cold item experiment, CoWPiRec significantly improves the performance in most item groups, especially in the items group that are less interacted with by users, i.e., group [0,5) and [5,10). The experiment result demonstrates that CoWPiRec can effectively alleviate the cold-start issue in cross-scenario recommendations utilizing the item representations capturing the word-level CF signals.

D. Case Study

From the experimental results in section IV-B, we can see that CoWPiRec achieves significantly better performance than other methods in most cases. Since we did not perform cross-domain pre-training for the SRL module, or even don't leverage it (i.e., zero-shot setting). We believe that the performance improvement of CoWPiRec mainly comes from the ability learned in the pre-training stage to capture word-level CF signals. We will show a case to illustrate how CoWPiRec leverages the knowledge learned from the word graph-based pre-training to improve the performance of downstream recommendation tasks.

In the case shown in Figure 3, CoWPiRec ranks the ground-truth next item at the 3rd position without any in-domain user interaction data training (i.e., zero-shot setting). It is significantly better than two strong baselines, i.e., MoRec and UniSRec. We believe CoWPiRec achieves significantly better ranking performance by capturing the word-level user preferences, i.e., the words "santa" and "card" in the recent interaction and the words "red" and "retro" in the target item. We can find co-click relationships with similar word-level preferences in the word graph. It indicates that CoWPiRec learns these word-level CF signals from word graph-based pre-training and applies the learned knowledge to the recommendation task in downstream datasets.

V. Conclusion

In this paper, we proposed a transferable item representation learning framework, named CoWPiRec. Different from previous transferable sequential recommenders that typically utilize the text-based IRL module as an offline feature extractor and learn a universal SRL module, we focus on incorporating recommendation knowledge into the text-based IRL module allowing it to capture CF signals. Considering the item-level CF signal is not suitable for the widely used text-based IRL module, i.e., PLM. We first construct a word graph fused with CF signals by collecting co-click word pairs and then integrating these signals into the PLM via a word-level pre-training task. With the ability to capture word-level recommendation information, CoWPiRec can even perform recommendations with a simple SRL module without trainable parameters, i.e., mean pooling. Furthermore, combining CoWPiRec with the SRL module and performing downstream training can achieve significantly better performance compared with state-of-the-art transferable sequential recommenders. Note that the SRL module used in the experiment is not tailored for CoWPiRec and just follows a previous architecture. It leaves us a future work of exploring a sophisticated SRL to improve the performance of CoWPiRec.

References

[1] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[2] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: challenges, progress and prospects," *arXiv preprint arXiv:2001.04830*, 2019.

[3] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018, pp. 197–206.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] H. Ding, Y. Ma, A. Deoras, Y. Wang, and H. Wang, "Zero-shot recommender systems," *arXiv preprint arXiv:2105.08318*, 2021.

[6] Y. Hou, S. Mu, W. X. Zhao, Y. Li, B. Ding, and J.-R. Wen, "Towards universal sequence representation learning for recommender systems," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 585–593.

[7] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 811–820.

[8] B. Hidasi and D. Tikk, "General factorization framework for context-aware recommendations," *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 342–371, 2016.

[9] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1419–1428.

[10] S. Jang, H. Lee, H. Cho, and S. Chung, "Cities: Contextual inference of tail-item embeddings for sequential recommendation," in *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2020, pp. 202–211.

[11] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the eleventh ACM international conference on web search and data mining*, 2018, pp. 565–573.

[12] Z. Liu, M. Cheng, Z. Li, Q. Liu, and E. Chen, "One person, one model—learning compound router for sequential recommendation," in *2022 IEEE International Conference on Data Mining (ICDM)*, 2022, pp. 289–298.

[13] Z. He, H. Zhao, Z. Lin, Z. Wang, A. Kale, and J. McAuley, "Locker: Locally constrained self-attentive sequential recommendation," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3088–3092.

[14] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.

[15] Y. Hou, B. Hu, Z. Zhang, and W. X. Zhao, "Core: simple and effective session-based recommendation within consistent representation space," in *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 2022, pp. 1796–1801.

[16] J. Chang, C. Gao, Y. Zheng, Y. Hui, Y. Niu, Y. Song, D. Jin, and Y. Li, "Sequential recommendation with graph neural networks," in *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 2021, pp. 378–387.

[17] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 346–353.

[18] X. Xiao, H. Dai, Q. Dong, S. Niu, Y. Liu, and P. Liu, "Social4rec: Distilling user preference from social graph for video recommendation in tencent," *arXiv preprint arXiv:2302.09971*, 2023.

[19] Q. Zhang, J. Li, Q. Jia, C. Wang, J. Zhu, Z. Wang, and X. He, "Unbert: User-news matching bert for news recommendation." in *IJCAI*, 2021, pp. 3356–3362.

[20] C. Wu, F. Wu, T. Qi, and Y. Huang, "Empowering news recommendation with pre-trained language models," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 1652–1656.

[21] Y. Yu, F. Wu, C. Wu, J. Yi, and Q. Liu, "Tiny-newsrec: Effective and efficient plm-based news recommendation," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 5478–5489.

[22] Y. Hou, Z. He, J. McAuley, and W. X. Zhao, "Learning vector-quantized item representation for transferable sequential recommenders," *arXiv preprint arXiv:2210.12316*, 2022.

[23] J. Wang, F. Yuan, M. Cheng, J. M. Jose, C. Yu, B. Kong, Z. Wang, B. Hu, and Z. Li, "Transrec: Learning transferable recommendation from mixture-of-modality feedback," *arXiv preprint arXiv:2206.06190*, 2022.

[24] S. Mu, Y. Hou, W. X. Zhao, Y. Li, and B. Ding, "Id-agnostic user behavior pre-training for sequential recommendation," in *Information Retrieval: 28th China Conference, CCIR 2022, Chongqing, China, September 16–18, 2022, Revised Selected Papers*. Springer, 2023, pp. 16–27.

[25] Z. Yuan, F. Yuan, Y. Song, Y. Li, J. Fu, F. Yang, Y. Pan, and Y. Ni, "Where to go next for recommender systems? id-vs. modality-based recommender models revisited," *arXiv preprint arXiv:2303.13835*, 2023.

[26] F. Zhu, Y. Wang, C. Chen, J. Zhou, L. Li, and G. Liu, "Cross-domain recommendation: challenges, progress, and prospects," *arXiv preprint arXiv:2103.01696*, 2021.

[27] K. Xu, Z. Wang, W. Zheng, Y. Ma, C. Wang, N. Jiang, and C. Cao, "A centralized-distributed transfer model for cross-domain recommendation based on multi-source heterogeneous transfer learning," in *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2022, pp. 1269–1274.

[28] G. Hu, Y. Zhang, and Q. Yang, "Conet: Collaborative cross networks for cross-domain recommendation," in *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018, pp. 667–676.

[29] C. Wu, F. Wu, T. Qi, J. Lian, Y. Huang, and X. Xie, "Ptum: Pre-training user model from unlabeled user behaviors via self-supervision," *arXiv preprint arXiv:2010.01494*, 2020.

[30] C. Xiao, R. Xie, Y. Yao, Z. Liu, M. Sun, X. Zhang, and L. Lin, "Uprec: User-aware pre-training for recommender systems," *arXiv preprint arXiv:2102.10989*, 2021.

[31] F. Yuan, G. Zhang, A. Karatzoglou, J. Jose, B. Kong, and Y. Li, "One person, one model, one world: Learning continual user representation without forgetting," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 696–705.

[32] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 650–658.

[33] F. Zhu, C. Chen, Y. Wang, G. Liu, and X. Zheng, "Dtcdr: A framework for dual-target cross-domain recommendation," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1533–1542.

[34] K. Shin, H. Kwak, S. Y. Kim, M. N. Ramstrom, J. Jeong, J.-W. Ha, and K.-M. Kim, "Scaling law for recommendation models: Towards general-purpose user representations," *arXiv preprint arXiv:2111.11294*, 2021.

[35] S. Shi, W. Ma, Z. Wang, M. Zhang, K. Fang, J. Xu, Y. Liu, and S. Ma, "Wg4rec: Modeling textual content with word graph for news recommendation," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1651–1660.

[36] P. Liu, L. Zhang, and J. A. Gulla, "Pre-train, prompt and recommendation: A comprehensive survey of language modelling paradigm adaptations in recommender systems," *arXiv preprint arXiv:2302.03735*, 2023.

[37] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[38] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 1893–1902.

[39] J. Ni, J. Li, and J. McAuley, "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)*, 2019, pp. 188–197.

[40] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian *et al.*, "Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4653–4664.

[41] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.