frontiers in Big Data

REVIEW
published: 18 March 2021
doi: 10.3389/fdata.2021.602071

Knowledge Transfer via Pre-training for Recommendation: A Review and Prospect

Zheni Zeng[1,2†], Chaojun Xiao[1,2†], Yuan Yao[1,2], Ruobing Xie[3], Zhiyuan Liu[1,2], Fen Lin[3], Leyu Lin[3] and Maosong Sun[1,2]*

[1]Department of Computer Science and Technology Institute for Artificial Intelligence, Tsinghua University, Beijing, China, [2]Beijing National Research Center for Information Science and Technology, Beijing, China, [3]WeChat Search Application Department, Search Product Center, Shenzhen, China

Recommender systems aim to provide item recommendations for users and are usually faced with data sparsity problems (e.g., cold start) in real-world scenarios. Recently pre-trained models have shown their effectiveness in knowledge transfer between domains and tasks, which can potentially alleviate the data sparsity problem in recommender systems. In this survey, we first provide a review of recommender systems with pre-training. In addition, we show the benefits of pre-training to recommender systems through experiments. Finally, we discuss several promising directions for future research of recommender systems with pre-training. The source code of our experiments will be available to facilitate future research.

Keywords: recommender system, pre-trained model, knowledge transfer, cross-domain transfer, cold start

OPEN ACCESS

Edited by:
Fuzheng Zhang, Independent researcher, Beijing, China

Reviewed by:
Chaochao Chen, Independent researcher, Hangzhou, China
Lidan Shou, Zhejiang University, China

***Correspondence:**
Zhiyuan Liu
liuzy@tsinghua.edu.cn

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Data Mining and Management, a section of the journal Frontiers in Big Data

Received: *02 September 2020*
Accepted: *18 January 2021*
Published: *18 March 2021*

Citation:
Zeng Z, Xiao C, Yao Y, Xie R, Liu Z, Lin F, Lin L and Sun M (2021) Knowledge Transfer via Pre-training for Recommendation: A Review and Prospect. Front. Big Data 4:602071. doi: 10.3389/fdata.2021.602071

1 INTRODUCTION

With the rapid development of the Internet, users are faced with information overload, where the large quantity of online items makes it hard for users to make decisions effectively. Recommender systems aim to provide recommendations by capturing user preferences for items (e.g., movies, books, songs, news, and websites) from explicit item ratings given by users or implicit user-item interactions (e.g., browsing and purchasing histories). The application of recommender systems has enabled personalized services in many scenarios, such as e-commerce and website browsing.

Recommender systems are usually faced with data sparsity in real-world scenarios. Recommender systems can suffer when providing recommendations for new items or users due to lack of information, which is known as the cold start problem (Gope and Jain, 2017). Without sufficient data, the model parameters cannot be well estimated and users' preference cannot be well modeled. It has been shown that the data sparsity problem in recommender systems can be alleviated by transferring knowledge from other domains or tasks (Cantador et al., 2015) and integrating heterogeneous external knowledge (Guo et al., 2020).

In the field of natural language processing, pre-trained models have achieved great success recently on a broad variety of tasks by knowledge transfer (Qiu et al., 2020). Models are usually first pre-trained on large-scale unsupervised data to learn universal language representations and then fine-tuned on downstream tasks to achieve knowledge transfer. The models can be pre-trained to learn either shallow context-free word embeddings (Mikolov et al., 2013) or deep context-aware language representations (Devlin et al., 2019). The resulting language representations have proven to be useful not only for different tasks [e.g., natural language inference and question answering (Devlin

et al., 2019)] but also for different scenarios, such as few-shot learning (Brown et al., 2020) and domain adaptation (Rietzler et al., 2020).

In the context of recommender systems, we can group works that utilize pre-training mechanisms to improve the precision of recommendation into two categories: feature-based models and fine-tuning models. The feature-based models generally use pre-trained models to obtain features from side-information (e.g., the content of items and knowledge graphs) for users and items (Guo et al., 2020). The fine-tuning models leverage the user-item interaction records to pre-train a deep transferable neural model, which is subsequently fine-tuned to downstream recommendation tasks (Chen et al., 2019c). Generally, the benefits of pre-training to recommender systems can be summarized as being twofold: 1) pre-training tasks can better exploit user-item interaction data to capture user interests, and 2) pre-training can help integrate knowledge from different tasks and sources into universal user/item representations, which can be further adapted to various scenarios in recommender systems, such as cold starts and cross-domain transfer.

The contributions of this survey can be summarized the following. 1) *Systematic Review*: We provide a systematic review of the pre-training methods for recommender systems with a clear taxonomy. 2) *Empirical Results*: We present empirical results to show the benefits of pre-training to recommender systems. We conduct experiments on the task of movie recommendation where different types of knowledge are integrated by pre-training for better recommendations, especially in the cold start and cross-domain transfer scenarios. 3) *Future Directions*: Based on the review and experiments, we discuss several promising directions for future research, including how to better leverage pre-training approaches to improve recommender systems and how recommender systems can motivate better pre-training models.

The rest of the survey is organized as follows: In **Sections 2** and **3**, we provide a review of existing methods of recommender systems with pre-training; in **Section 4**, we conduct experiments to empirically show the benefits of pre-training to recommender systems; and in **Section 5**, we discuss promising directions for future research.

2 FEATURE-BASED MODELS

Feature-based models leverage side-information (e.g., contents of items, knowledge graphs, and social networks) using pre-trained models to directly enrich the representations of users or items. Different from collaborative filtering (CF) methods that learn the representations from user-item interaction records (Hu et al., 2008; Su and Khoshgoftaar, 2009; Wang et al., 2020a), feature-based models focus on extracting widely applicable features from external information sources with pre-trained models and then integrate these features into the recommendation framework. By combining rich side-information and user-item interaction data, feature-based models can potentially solve some challenges, such as the data sparsity problem.

The general idea can be illustrated as follows. Given the external information resource, the pre-trained models are applied to obtain the external feature vectors, $\hat{u}_i$ and $\hat{v}_j$ for user u_i and item v_j respectively. Denote $\tilde{u}_i$ and $\tilde{v}_j$ as the features learned from the user-item interaction records for user u_i and item v_j respectively. Then the final representations u_i and v_j for each user and item are obtained by aggregating external feature vectors and the vectors from the user-item interaction data:

$$u_i = g_u(\tilde{u}_i, \hat{u}_i), \quad v_i = g_v(\tilde{v}_i, \hat{v}_i), \tag{1}$$

where $g_u(\cdot)$ and $g_i(\cdot)$ are aggregate functions. The preference score for user u_i and item v_j is calculated by

$$s(u_i, v_j) = f(u_i, v_j), \tag{2}$$

where $f(\cdot)$ is a recommendation function, which can be factorization machines (Rendle, 2010) and deep neural networks (He et al., 2016; Fan et al., 2018) and the like.

According to the type of external information resources, feature-based pre-trained models can be roughly categorized into content-based recommendation, knowledge graph-based recommendation, and social recommendation models. Different types of meta-information require different pre-trained models. We will introduce how the pre-training mechanism is used in these three kinds of recommender systems in detail.

2.1 Content-Based Recommendation

Content-based recommendation assumes that users prefer items similar to those being historically interacted with. Therefore, it is important for content-based recommender systems to encode the content of items into expressive low-dimensional representations. The pre-trained models have proven to be powerful in extracting generally applicable representations from text, images, audio, etc. Hence, many works learn features from the content of items to serve recommendation models.

Liang et al. (2015) pre-train a multilayer neural network to extract audio features for music recommendation via a semantic tag prediction task. In terms of dealing with textual data for recommendation, such as reviews (Zheng et al., 2017), tweets (Gong and Zhang, 2016), and news (Cao et al., 2017), pre-trained word embeddings or pre-trained sentence encoders become indispensable. Some works simply use the average of word embeddings to represent the whole documents (Brochier, 2019; Cenikj and Gievska, 2020). Other works focus on the design of task-specific frameworks, where the input word embeddings will be fed into a complex document encoder to generate the document representation (Song et al., 2016; Tan et al., 2016; Xu et al., 2016; Nguyen et al., 2017; Zhang et al., 2017). Similarly, the pre-training mechanism is widely used in image feature extraction for recommender systems (Chu and Tsai, 2017; He and McAuley, 2016a, He and McAuley, 2016b).

2.2 Knowledge Graph-Based Recommendation

Knowledge graph-based recommendation introduces knowledge graphs (KGs) as side-information to better characterize users and items. A KG is a structured graph

containing fruitful facts and connections between users, items, and other related entities. Amounts of side-information, such as the user profiles, the attributes of items, and relations between cross-domain items, can be integrated into KGs. Hence, KGs can help recommender systems to capture essential knowledge and provide explanations for the recommendation results.

Various KGs have been used in different works. For instance, some works construct knowledge graphs with items and their related attributes (Zhang et al., 2016; Wang et al., 2018b; Huang et al., 2018). Some other works add users to build user-item graphs, which contain information, including the item attributes, user behaviors (e.g., purchase, click), and user profiles (Wang et al., 2018a; Cao et al., 2019; Dadoun et al., 2019). With the informative heterogeneous user-item graphs, the potential relations between users and items can be modeled directly.

In order to exploit KGs, one line of KG-based methods seeks to encode the KG into low-dimensional pre-trained embeddings with the knowledge graph embedding (KGE) methods (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015; Ji et al., 2015; Trouillon et al., 2016; Shi and Weninger, 2017; Balazevic et al., 2019). Then as stated in **Eq. 1**, the knowledge graph embeddings are aggregated with user/item features obtained from interaction data. Experimental results show that KGs are powerful information resources and can improve the performance of recommendation significantly (Qin et al., 2020).

2.3 Social Recommendation

Social recommendation is a type of recommendation method that utilize online social relations as an additional input (Tang et al., 2013). Different from KGs, which integrate various information about users and items, social graphs focus on modeling the social relation between users. Homophily theory indicates that the preference of a user is similar to or influenced by their socially connected friends (McPherson et al., 2001). Similar to the KG-based recommendation, many social recommender systems seek to integrate the pre-trained social network embeddings, which indicates the degree that a user is influenced by his/her friends (Guo et al., 2018; Wen et al., 2018; Zhang et al., 2018; Sathish et al., 2019; Chen et al., 2019a, Chen et al., 2019b).

2.4 Summary

Feature-based models preprocess side-information with various pre-trained models to obtain the embedding of users or items, which are then integrated into the recommender systems. By utilizing side-information, feature-based approaches are able to construct expressive representations for users and items and can achieve significant improvement for recommendation. In addition to have side-information, exploiting large-scale interaction data is also crucial to recommender systems. Therefore, some recent efforts have been made to pre-train models with user-item interaction records, which are introduced in the following section.

3 FINE-TUNING MODELS

The fine-tuning models for recommendation first pre-train the parameters with large-scale interaction data. The models are then transferred to downstream tasks by simply fine-tuning the pre-trained parameters. The fine-tuning paradigm has shown the effectiveness in other areas, such as natural language processing (Devlin et al., 2019; Joshi et al., 2020). According to the model architecture, we can categorize existing works in recommender systems into two classes: shallow neural networks (Hu et al., 2018; Ni et al., 2018) and deep residual neural networks. Existing deep residual neural networks for recommendation can be further divided into BERT-based models (Chen et al., 2019c; Sun et al., 2019; Yang et al., 2019) and parameter-efficient pre-trained convolutional neural networks (Yuan et al., 2020).

3.1 Shallow Neural Networks

Early works attempt to achieve knowledge transfer with shallow neural networks as base models, such as shallow multilayer perceptrons (MLP), and recurrent neural networks. Hu et al. (2018) attempt to improve recommendation via cross-domain knowledge sharing. They conduct a baseline experiment where an MLP together with user and items embeddings is pre-trained on the source domain. The user embeddings are then transferred to the target domain as warm-up. Experimental results show that this simple method cannot achieve obvious improvement in recommendation performance. The results demonstrate that the model architecture and pre-training tasks need to be carefully designed to achieve effective knowledge transfer in fine-tuning models. Therefore, many efforts have been devoted to investigating efficient pre-training tasks and transferrable model architectures.

Ni et al. (2018) propose the DUPN model which is able to learn universal user representations across multiple recommendation tasks. DUPN takes interaction sequence as inputs and then applies LSTM and an attention layer to obtain the user representations. DUPN is pre-trained by multiple tasks objectives, including click-through rate prediction, shop preference prediction, and price preference prediction. Experimental results show that DUPN can achieve not only improvement on the tasks used for pre-training but also faster convergence and promising results in related new tasks. Though the user representations learned with DUPN are powerful, DUPN requires many extra information sources like user profiles to facilitate different pre-training tasks.

3.2 BERT-Based Models

In order to capture the dynamic user preference, many researchers attempt to exploit the user chronological interaction sequence, which is called the session-based recommendation. Similar to natural language processing (NLP) that targets on word sequence, session-based recommendation investigates the item sequence and aims to take sequential information into account. Inspired by the rapid progress in pre-trained language models in NLP (Devlin et al., 2019; Liu et al., 2019; Joshi et al., 2020), many efforts have been devoted to capturing information from the user behavior sequence with pre-trained models, especially BERT-based models. In this section, we introduce the pre-trained BERT-based models for recommendation, including the widely

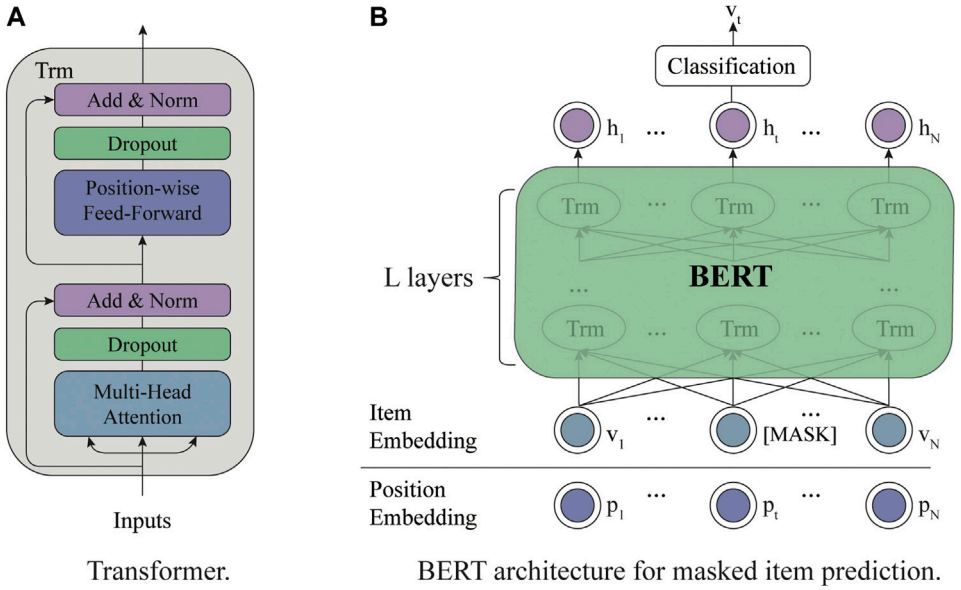

FIGURE 1 | (A) is the framework of Transformer. **(B)** is the overall architecture of BERT for masked item prediction task.

used masked item prediction task, the architecture of BERT, and advanced BERT-based models for recommendation.

3.2.1 Masked Item Prediction

Similar to the masked language modeling task in NLP, the masked item prediction task (MIP) is proposed and widely applied in many recommender systems. In the MIP task, given the interaction sequence, some of the items are randomly masked. The models are required to reconstruct the masked item. Formally, we denote the interaction sequence in chronological order for user u as $s_u = \{v_1, v_2, \ldots, v_N\}$, where v_i is the item that u interacted with at time step i. For pre-training stage, some input items in the interaction sequence are randomly masked with special token [MASK]. Then the models are asked to predict the masked items:

$$\begin{aligned} \textit{Inputs}: \quad & \{v_1, v_2, v_3, v_4, v_5\} \rightarrow \{v_1, v_2, [\text{MASK}]_1, v_4, [\text{MASK}]_2\}, \\ \textit{Labels}: \quad & [\text{MASK}]_1 = v_3, \qquad [\text{MASK}]_2 = v_5. \end{aligned}$$

The objective of this task is the negative likelihood of the masked targets. Unlike the left-to-right Next Item Prediction (NIP) task that is used in many session-based recommender systems (Yu et al., 2016; Wu et al., 2017; Hidasi and Karatzoglou, 2018), MIP enables the models to learn representations of user behavior sequences from the whole context. Moreover, the MIP task can overcome the limitation that a rigidly ordered sequence is not always practical in user behaviors (Sun et al., 2019). Therefore, models pre-trained with the MIP task can achieve promising results.

3.2.2 BERT for Recommender System

Inspired by the success of BERT (Devlin et al., 2019) in text understanding, many researchers adopt BERT for recommendation. In this section, we will introduce the architecture of BERT and how to utilize BERT for recommender systems. For convenience, we denote the BERT model pre-trained with the MIP task as BERT4RS. As shown in **Figure 1**, BERT is based on a multilayer bidirectional Transformer (Vaswani et al., 2017). The Transformer contains two sublayers: multihead attention sublayer and point-wise feed-forward network.

- *Multihead Attention*: an attention mechanism has been used successfully in various sequence modeling tasks, which enables models to focus on important information. The attention function takes queries $Q \in \mathbb{R}^{l_Q \times d_k}$, keys $K \in \mathbb{R}^{l_K \times d_k}$, and values $V \in \mathbb{R}^{l_K \times d_v}$ as inputs and computes the outputs as follows, where d_k and d_v are the dimensions and l_Q and l_K are the length of sequences.

$$\textit{Attention}(Q, K, V) = \textit{soft}\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{3}$$

Self-attention is a special attention function aiming to learn the representation for a single sequence, in which the input sequence performs as the queries, keys, and values. Instead of performing a single attention function, Transformer employs the multihead self-attention function, which allows the model to jointly attend to information from different vector subspaces. Specifically, this mechanism first linearly projects the input sequence into h subspaces and then produces the output representation with h attention functions.

$$MultiHead(H) = Concat(head_1, head_2, \ldots, head_h)W^O, \quad (4)$$

$$head_i = Attention(HW_i^Q, HW_i^K, HW_i^V), \quad (5)$$

where $H \in \mathbb{R}^{l \times d_o}$ is the input sequence, and $W_i^Q \in \mathbb{R}^{d_o \times d_k}$, $W_i^K \in \mathbb{R}^{d_o \times d_k}$, $W_i^V \in \mathbb{R}^{d_o \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_o}$ are learnable parameter matrices.

- *Point-wise Feed-Forward Network*: the multihead attention function enables the model to integrate information from different positions with linear combinations. Then the point-wise feed-forward network endows the model nonlinearity. In this sublayer, a fully connected feed-forward network is applied to each position separately and identically.

$$FFN(x) = \mathrm{ReLU}(xW_1 + b_1)W_2 + b_2. \quad (6)$$

The sublayer consists of two linear transformations and a ReLU activation. It should be noted that though the transformations are the same across different positions, the parameters are different for different layers.

Following each sublayer, a residual connection (He et al., 2016) and a layer normalization operation (Ba et al., 2016) are employed for stabilizing and accelerating the network training. As shown in **Figure 1B**, after L layers of Transformer, the final hidden states of masked tokens are fed into a feedforward network to get the output distribution over target items.

BERT4RS is effective in modeling user preference from the historical behaviors. Sun et al. (2019) train a two-layer BERT with the MIP task, which achieves the state-of-the-art performance on the session-based next item recommendation task. They observe that both the BERT architecture and the MIP task can significantly improve the performance, and stacking multiple Transformer layers can further boost the performance on large-scale datasets, which provides fundamental support for the following models.

Chen et al. (2019c) proposed to fine-tune BERT4RS with a content-based click through prediction task. Specifically, the user representation u is produced with pre-trained BERT from the historical behavior sequence, and the item representation v is extracted from its content. Then the preference score is calculated with **Eq. 2**, where $f(\cdot)$ is an MLP layer.

Some downstream recommendation tasks, such as next basket recommendation (Rendle et al., 2010; Yu et al., 2016) and list-wise recommendation (Shi et al., 2010; Zhao et al., 2017), require the model to capture the relations between item sequences. Therefore, in addition to MIP, researchers propose some sequence-level pre-training tasks to pre-train the model. Yang et al. (2019) adopts the BERT4RS for next basket recommendation task, which is pre-trained with MIP and next basket prediction (NBP) tasks. In real-world scenarios, a user usually buys or browses a series of items (a basket) at a time. Given two baskets, NBP requires the model to predict whether the two baskets are adjacent in the purchase records:

$$\begin{aligned} Inputs: \quad & \{[\mathrm{CLS}], v_1^1, v_2^1, v_3^1, [\mathrm{SEP}], v_1^2, v_2^2, v_3^2, [\mathrm{SEP}]\} \\ \rightarrow \quad & \{[\mathrm{CLS}], v_1^1, [\mathrm{MASK}]_1, v_3^1, [\mathrm{SEP}], v_1^2, v_2^2, [\mathrm{MASK}]_2, [\mathrm{SEP}]\} \\ MIP\ Lables: \quad & [\mathrm{MASK}]_1 = v_2^1, \quad [\mathrm{MASK}]_2 = v_3^2 \\ NBP\ Lables: \quad & IsNext/NotNext, \end{aligned}$$

where v_i^1 and v_i^2 are items from different baskets and [CLS] and [SEP] are special tokens. The final hidden state of [CLS] is used to predict the NBP label.

3.3 Parameter-Efficient Pre-trained Model

The pre-training mechanism can enable models to capture user preference from behavior history via self-supervised learning. Experimental results show that it can achieve significant improvement for recommendation. However, fine-tuning models separately for different tasks is computationally expensive and memory intensive (Mudrakarta et al., 2018; Stickland and Murray, 2019), especially for resource-limited devices.

To address this issue, Yuan et al. (2020) proposed the Peterrec, which utilizes a grafting neural network in fine-tuning, termed as model patch (MP). By inserting MPs in the pre-trained models, the fine-tuning networks can keep all pre-trained parameters unchanged. As shown in **Figure 2**, Peterrec is a stack of dilated convolutional (DC) layers (Yu and Koltun, 2016), and there is a residual connection between every two DC layers. Similar to other pre-trained models, Peterrec employs the MIP task to pre-train. During the fine-tuning stage, the MP blocks, which are simple residual two-layer convolutional networks, are inserted around the original DC layers. The pre-trained parameters are shared across different tasks. Only parameters of the MP blocks will be fine-tuned. To accelerate fine-tuning and minimize the number of parameters, the MP blocks are designed in a bottleneck architecture. Specifically, the first convolutional layer projects the k dimensional channels into d ($d \ll k$) dimensional vectors, and the second layer projects it back to its original dimension. Thus, the number of inserted parameters can be less than 10% parameters of original pre-trained models. Empirical results show that the pre-trained Peterrec is useful for various downstream tasks, including user profile prediction and top-k recommendation. Moreover, it can achieve promising results even when the user is cold in the new target domain, which proves the effectiveness of pre-trained models in knowledge transfer for recommendation.

3.4 Summary

The pre-training mechanism works well in many recommendation tasks. Early works explore pre-training with shallow neural networks. Inspired by the success of pre-trained language models in NLP, the deep residual models pre-trained with MIP task are widely used in many different recommender systems. Although it has been proven effective to employ pre-trained models for recommendation, there are many open challenges to be addressed, which will be discussed in **Section 5**.

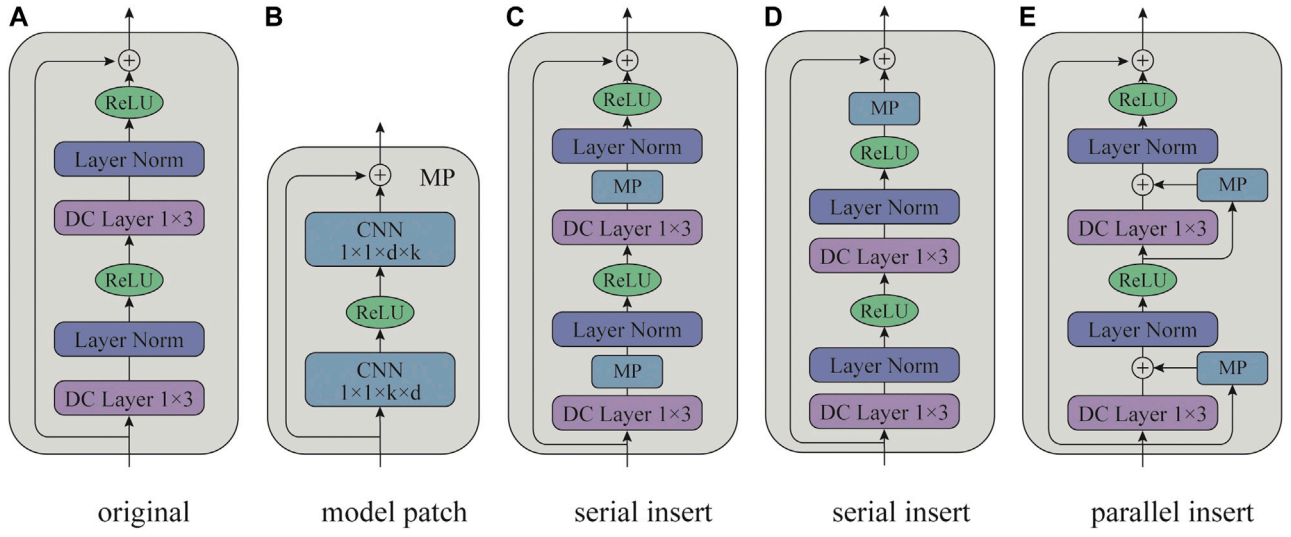

FIGURE 2 | The model architecture of Peterrec (Yuan et al., 2020). **(A)** is the original residual block of the pre-trained model. **(B)** is the model patch (MP) block. **(C–E)** are different approaches to insert MP blocks for fine-tuning. ⊕ indicates element-wise addition operation. 1×3 and $1 \times 1 \times d \times k$ refer to the kernel size of convolutional layers.

4 EXPERIMENT OF RECOMMENDER SYSTEM WITH PRE-TRAINING

In this section, we conduct experiments to verify the benefits of pre-training to recommender systems. We take next-item and rating prediction recommender systems as examples and investigate the potential of pre-trained recommender systems in cold start problem and cross-domain knowledge transfer.

4.1 Dataset

We evaluate the models on MovieLens[1], which is a representative and popular real-world benchmark in recommendation field. We choose the well-established lightweight version MovieLens 1 m (ML-1m). Each user-item interaction contains a timestamp and an integer rating score from 1 to 5. We follow Tang and Wang (2018) for data preprocessing. Users and items with too few ratings (< 5) are filtered out. We leave the last 5 items of each sequence to form the validation set (2 items for each user) and the test set (3 items for each user). To investigate the effectiveness of pre-trained recommender systems in cross-domain context, we divide all the movies in ML-1m into two domains according to movie genres and obtain interaction sequences of the same users on two domains. The scale of the target domain is smaller than the source domain (ML-1m-src). To better evaluate the models under the cold start scenario, we further truncate the interaction sequences in the target domain such that the sequence length of each user is less than 10, which results in the ML-1m-tgt set.

Apart from user-item interactions, we also use the meta information from IMDB[2] (IMDB 5000 Movie Dataset) to provide side-information for the items. For each movie, we query the director and three main actors. The statistics of the processed datasets are summarized in **Table 1**.

TABLE 1 | Statistics of datasets. #User, #Item, #ItMeta, and #InA indicate the number of users, items, items with meta information, and interactions, respectively. AvgLen refers to the average length of input training sequences.

Dataset	#User	#Item	#ItMeta	#InA	AvgLen
ML-1m	6,040	3,706	1,987	999k	165.7
ML-1m-src	6,040	2,656	1,381	622k	97.9
ML-1m-tgt	6,040	1,012	512	81k	8.85

4.2 Task Settings and Baselines

We regard user ratings as interaction sequence based on timestamps and evaluate the models on two representative recommendation tasks, including NIP and rating prediction. For both tasks, models are first pre-trained on ML-1m-src and then fine-tuned and evaluated on ML-1m-tgt.

Next Item Prediction: NIP aims to predict the next item that is likely to interest the user given the historical interaction sequence. For each sequence, we generate negative samples in which items are not seen in training data. Since movie rating behavior is not strictly sequential, we treat the three items in each test sequence as equivalent. We adopt Normalized Discounted Cumulative Gain (NDCG@K) and Recall@K as evaluation metrics. Models are provided with IMDB 5000 as side-information for items.

For baseline methods, we choose two representative models. Caser (Tang and Wang, 2018) is a shallow recommendation model which combines vertical and horizontal CNNs to extract user and item representations. BERT4Rec (Sun et al., 2019)

[1]https://grouplens.org/datasets/movielens/

[2]https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset

utilizes BERT as architecture backbone for session-based recommendation. To investigate the potential of pre-training in deep recommendation models, we increase the number of layers in BERT4Rec.

Rating Prediction. We also evaluate the pre-trained recommender systems on the rating prediction task, where models are required to predict the rating of the items based on users' historical feedback and profile. Following previous works in rating prediction (Zheng et al., 2016; Xing et al., 2019), we formalize the rating prediction task as a regression task and adopt Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as evaluation metrics. In addition to BERT4Rec (Sun et al., 2019), we choose DACF (Kuchaiev and Ginsburg, 2017) which utilizes deep autoencoder for rating prediction as our baseline models.

4.3 Model Design Choices

We investigate the effect of different design choices on pre-trained recommendation systems, including knowledge transfer approach and pre-training tasks. We also explore the potential of pre-trained recommendation systems in integrating external side-information.

Knowledge Transfer Approach. Our main purpose is to verify the effectiveness of pre-training on recommender systems. Therefore, we first investigate pre-training session-based recommendation models on ML-1m-src before fine-tuning on ML-1m-tgt. We hope that the knowledge learned from pre-training can be transferred between domains. During fine-tuning, we investigate two approaches to knowledge transfer: 1) embedding transfer where we can transfer and fine-tune only the input embedding layer (i.e., user embeddings and external knowledge embeddings) and leave other layers randomly initialized and learned from scratch in the target domain and 2) model transfer where we also explore transferring and fine-tuning the whole model on the target domain. In this way, user-item interaction knowledge is also expected to be transferred[3].

Pre-training Tasks: the design of pre-training tasks is crucial for pre-trained models to capture general knowledge from large-scale data. We compare two widely used pre-training tasks in recommender systems: 1) NIP recurrently predicts the next item in a left-to-right fashion, and 2) Masked Item Prediction (MIP), which randomly masks the items and predicts the masked item according to bidirectional context.

External Knowledge: external knowledge is shown to be effective in handling data sparsity problem in recommender systems. We investigate whether external knowledge can be combined with general knowledge learned from pre-training to achieve better results. Specifically, we constrain the item features by concatenating the external knowledge embeddings (i.e., director embeddings and actor embeddings) with the item embeddings to obtain external knowledge enhanced item embeddings.

[3]Note that in both knowledge transfer approaches, item embeddings are randomly initialized on the target domain, since items are not shared between domains.

4.4 Implementation Details

We find the optimal settings of hyperparameters via grid search on the validation set.

For the Caser model, we use implementation provided by the authors[4]. Batch size is searched from $\{128, 256, 512, 1024\}$ and tuned as 512. Hidden dimension size is searched from $\{30, 50, 100, 150\}$ and tuned as 100. We employ Adam to optimize the model, and we set the learning rate as $1e-3$, weight decay as $1e-6$. For masked item prediction, we set the mask probability as 0.2. Following Devlin et al. (2019), we do not always replace the masked item with [MASK] token. If an item is chosen, we replace it with 1) [MASK] 80% of the time, 2) a random item 10% of the time, and 3) the unchanged item 10% of the time.

For BERT4Rec (Sun et al., 2019) model, we choose a PyTorch implementation[5]. We keep most of the original hyperparameter and initialization strategy settings. Batch size is searched from $\{128, 256, 512\}$ and tuned as 256. Hidden dimension size is used to search $\{64, 128, 256\}$ and is tuned to 128. We employ Adam to optimize the model, and we set the learning rate as $1e-3$. We add user embedding and set the user embedding size as 32. We change the layer of BERT blocks to 6. For masked item prediction, we set the mask probability as 0.15 and use the same masking strategy as in Caser.

When incorporating external knowledge, the size of director and main actor embedding is 25 for Caser and 16 for BERT4Rec. The number of negative samples is set to 100. In evaluation, negative samples size is 100 for both models.

For rating prediction task, the rating scores are regarded as classification tags, and we simply concatenate rating embedding for each user-item interaction and set the embedding size to 128. The output layer of BERT4Rec is modified from classification to regression. The model is required to predict the rating scores for rating-masked items during training and the last three items of each user sequence during testing. For collaborative filtering model, we choose a deep learning implementation DACF[6] and keep the default settings.

4.5 Experiment Results

The experiment results on ML-1m-tgt are reported in **Table 2** and **Table 3**, from which we have the following observations[7]:

1) Pre-training boosts the recommendation performance for both models. However, the effects of knowledge transfer approaches are correlated with model capacity. Specifically, for the shallow Caser model, embedding transfer (i.e., only transfer the embedding layer) tends to achieve larger improvements with model transfer (i.e., transfer the whole model). In contrast, model transfer achieves much better performance for deep BERT4Rec model. We hypothesize that general knowledge about user-item interactions can be

[4]https://github.com/graytowne/caser_pytorch

[5]https://github.com/jaywonchung/BERT4Rec-VAE-Pytorch

[6]https://github.com/artem-oppermann/Deep-Autoencoders-For-Collaborative-Filtering

[7]The observations are supported by the statistical significance test on the experiment results with $p < 0.05$.

TABLE 2 | Performance (%) comparison of different settings on NIP of ML-1m-tgt.

Model	Transfer	Task		With meta	Test result					
		NIP	MIP		NDCG@3	NDCG@5	NDCG@10	Recall@3	Recall@5	Recall@10
Caser	None				14.46	16.82	19.83	13.20	17.30	23.91
				✓	14.34	16.76	19.73	13.10	17.27	23.82
	Embedding	✓			14.70	16.92	19.89	13.39	17.24	23.79
				✓	14.41	16.76	19.62	13.35	17.43	23.72
			✓		15.08	17.25	20.02	13.74	17.49	23.59
				✓	15.40	17.62	20.39	14.01	17.85	23.93
	Model	✓			15.23	17.41	20.36	13.57	17.34	23.82
				✓	12.00	14.62	17.88	11.44	15.97	23.14
			✓		15.03	17.19	19.94	13.73	17.47	23.54
				✓	11.88	14.69	17.93	11.47	16.33	23.43
BERT4Rec	None				42.34	51.56	61.18	40.94	56.79	77.87
				✓	43.92	53.03	62.41	42.31	58.03	78.58
	Embedding	✓			42.95	52.09	61.61	41.46	57.25	78.10
				✓	43.95	53.11	62.66	42.38	58.17	79.08
			✓		43.82	53.22	62.74	42.53	58.76	79.61
				✓	44.89	54.35	63.84	43.33	59.65	80.41
	Model	✓			43.75	52.84	62.20	42.28	57.95	78.48
				✓	44.58	53.59	63.03	42.93	58.49	79.15
			✓		44.58	54.13	63.71	43.31	59.78	80.75
				✓	45.12	54.48	63.97	43.72	59.88	80.64

TABLE 3 | Performance (%) comparison of different settings on rating prediction of ML-1m-tgt.

Model	RMSE	MAE
DACF	0.987	0.867
DACF transfer	0.976	0.859
BERT4Rec	1.371	0.960
BERT4Rec transfer	1.261	0.903

better captured by high-capacity models with pre-training, leading to better performance in transferring whole models.

2) For pre-training tasks, masked item prediction achieves better performance than NIP for BERT-based models, which is consistent with the results reported by Sun et al. (2019). One possible reason is that movie rating behavior does not strictly follow chronological order (i.e., the chronological order of two items are likely to be swapped). Therefore, by integrating bidirectional information, masked item prediction can learn better representations during pre-training. However, the superiority of masked item prediction does not seem obvious on Caser. It is probably because masked item prediction creates gap between pre-training and fine-tuning (e.g., prediction during fine-tuning is based on unidirectional context), which cannot be easily overcome by shallow models.
3) Incorporating external knowledge improves the performance of pre-trained BERT4Rec model. We note that for pre-trained Caser, the effect of external knowledge is not significant and sometimes even negative. We speculate that the limited capacity of simple models hinders the integration of external knowledge. Besides, simple concatenation cannot well integrate external knowledge with general knowledge learned from pre-training either. More advanced methods can be developed to inject external knowledge into pre-trained recommendation models.
4) Pre-training also improves the performance of the recommender system in rating prediction task. The two models both achieve lower error scores when first pre-trained on ML-1m-src. However, since the rating value of an item barely depends on sequential information, DACF outperforms BERT4Rec, which is designed for sequential recommendation. Most existing fine-tuning models can be only applied with sequential data.

In conclusion, recommender systems could benefit from pre-training, which effectively transfer knowledge between domains and tasks, and have potential in solving problems, including cold start. It is worth exploring design models to utilize diverse data for various downstream tasks.

5 OPEN CHALLENGES AND FUTURE DIRECTIONS

Although the pre-trained models have shown their power in recommender systems, the challenges still exist. In this section, we suggest five open challenges and future directions

of pre-trained models in recommendation, where (1), (2), and (3) are discussion about how to better utilize pre-trained models in various recommendation scenarios and (4) and (5) mainly focus on how to improve the pre-training mechanism to better serve recommender systems.

1) *Cold Start*: collaborative filtering recommender systems rely heavily on user historical behavior data and can suffer from the cold start problem. To alleviate the problem, some approaches (Ma et al., 2011; Manotumruksa et al., 2016; Yu et al., 2018) use side-information, such as user profile and item attributes, to enrich user/item representations. Besides, there are some efforts utilizing more efficient learning mechanism to alleviate the heavy data reliance, such as few-shot learning (Lee et al., 2019; Li et al., 2019).

Pre-trained language models can significantly improve few-shot performance (Brown et al., 2020) in NLP tasks. Similarly, in terms of recommendation, pre-trained models can be applied for cold start problem by learning transferable representations of the shared information between large-scale general domain and sparse target domain. For example, if a user is cold in the target domain, it is useful to transfer his/her representation pre-trained in the general domain; if an item is cold, its representation can be estimated by leveraging the pre-trained representations of external information. Peterrec (Yuan et al., 2020) is a good exploration which achieves user cold start with pre-trained models.

2) *Knowledge Enhanced Pre-training*: knowledge graphs can provide rich domain knowledge, world knowledge, and commonsense knowledge for recommendation. Therefore, by incorporating KGs into recommendation, user preference and relations between items can be captured more accurately. Many KG-based approaches are proposed recently and have achieved promising results (Zhang et al., 2016; Wang et al., 2018b; Tang et al., 2019; Qin et al., 2020). However, few works consider directly injecting external structured knowledge into pre-trained models for recommendation.

In fact, many knowledge-enhanced pre-trained language models (Zhang et al., 2019; Liu et al., 2020; Wang et al., 2020b) have shown that fusing the structured knowledge into pre-trained models can significantly boost the performance of original models. Knowledge information can help models better characterize users and items and thus can improve the performance of recommendation.

3) *Social Relation Enhanced Pre-training*: social relations provide a possible perspective for personalized recommendation. Users who are connected are more likely to share similar preferences. Pre-trained models are proficient in capturing user interest from their historical interaction records. Therefore, the social relations between users can be viewed as meta-relations between user-item interaction sequences; i.e., the interaction sequences of closely connected users are encouraged to share similar representations. Based on this, sequence-level pre-training tasks can be proposed to help models to generate more expressive user/item representations.

Another possible direction is to employ social relation enhanced pre-trained models to solve user cold start problem. Social relations can provide clues for the user interest. However, it is still challenging to make full use of the rich information contained in the neighboring users in social relation graphs during pre-training.

4) *Pre-training Tasks*: currently, all the deep fine-tuning approaches rely on the MIP task to pre-train the model. And these works focus on extracting user interest from their historical sequential records. However, limited by the computing ability and memory of GPUs, only the most recent interaction records that represent recent user preference can be utilized by recommendation models. Besides, MIP can only utilize sequential data, while rich heterogeneous information is usually available in many real-world scenarios. Therefore, designing new self-supervised pre-training tasks is important to make full use of the large-scale heterogeneous data for recommendation.
5) *Model Architecture and Model Compression*: the pre-trained models are effective in various recommendation tasks. However, their high computation complexity makes it hard to deploy them in real-world scenarios. To address the problem, it would be helpful to perform model compression (McCarley, 2019; Gordon et al., 2020) or improve the model architecture. Besides, fine-tuning separately for each down-stream task is quite time consuming and memory intensive. The model patch (Yuan et al., 2020) is a good attempt to reduce memory cost. However, it is still an urgent need to achieve fast and effective knowledge transfer from pre-trained models to multiple down-stream tasks.

6 CONCLUSION

In this article, we investigate pre-trained models for recommendation and summarize the efforts devoted to this direction. We conduct a comprehensive overview of two types of pre-trained models for recommendation, including feature-based models and fine-tuning models. Then we conduct experiments to show the benefits of pre-training for recommender systems. Finally, open challenges and future direction are discussed, hoping to promote the progress of this domain.

AUTHOR CONTRIBUTIONS

ZZ, CX, YY, and RX contributed to the structure, design, and writing of the manuscript. CX collected the related studies. ZZ conducted the experiments. ZL, FL, LL, and MS provided valuable suggestions and helped in revising the manuscript.

FUNDING

This work was supported by the National Key Research and Development Program of China (no. 2018YFB1004503). Yao is also supported by 2020 Tencent Rhino-Bird Elite Training Program.

REFERENCES

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. Preprint: arXiv: 1607.06450.

Balazevic, I., Allen, C., and Hospedales, T. (2019). "Tucker: tensor factorization for knowledge graph completion," in Proceedings of EMNLP-IJCNLP, Hong Kong, China, November 2029.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). "Translating embeddings for modeling multi-relational data," in Proceedings of the NIPS, Red Hook, NY, December 5–8, 2013, 2787–2795.

Brochier, R. (2019). "Representation learning for recommender systems with application to the scientific literature," in Proceedings of the WWW, San Francisco, CA, May 2019, 12–16.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). Language models are few-shot learners. Preprint: arXiv:2005.14165.

Cantador, I., Fernández-Tobías, I., Berkovsky, S., and Cremonesi, P. (2015). "Cross-domain recommender systems," in *Recommender systems handbook* (New York, NY: Springer), 919–959.

Cao, S., Yang, N., and Liu, Z. (2017). "Online news recommender based on stacked auto-encoder," in Proceedings of the ICIS, Wuhan, China, 24–26 May, 2017 (New York, NY: IEEE), 721–726.

Cao, Y., Wang, X., He, X., Hu, Z., and Chua, T.-S. (2019). "Unifying knowledge graph learning and recommendation: towards a better understanding of user preferences," in Proceedings of the WWW, San Francisco, CA, May 2019, 151–161.

Cenikj, G., and Gievska, S. (2020). "Boosting recommender systems with advanced embedding models," in Proceedings of the Web Conference, Taipei, Taiwan, April 2020, 385–389.

Chen, J., Chen, W., Huang, J., Fang, J., Li, Z., Liu, A., et al. (2019a). "Co-purchaser recommendation based on network embedding," in Proceedings of the WISE, Hong Kong, China, November 26–30, 2019 (New York, NY: Springer), 197–211. doi:10.1007/978-3-030-34223-4_13

Chen, J., Wu, Y., Fan, L., Lin, X., Zheng, H., Yu, S., et al. (2019b). N2vscdnnr: a local recommender system based on node2vec and rich information network. *IEEE Trans. Comput. Soc. Syst.* 6, 456–466. doi:10.1109/tcss.2019.2906181

Chen, X., Liu, D., Lei, C., Li, R., Zha, Z.-J., and Xiong, Z. (2019c). "Bert4sessrec: content-based video relevance prediction with bidirectional encoder representations from transformer," in Proceedings of the MM, Nice, France, October 2019, 2597–2601.

Chu, W.-T., and Tsai, Y.-L. (2017). A hybrid recommendation system considering visual information for predicting favorite restaurants, *World Wide Web* 20, 1313–1331. doi:10.1007/s11280-017-0437-1

Dadoun, A., Troncy, R., Ratier, O., and Petitti, R. (2019). "Location embeddings for next trip recommendation," in Proceedings of the WWW, San Francisco, CA, May 2019, 896–903.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). "Bert: pre-training of deep bidirectional transformers for language understanding," in Proceedings of the NAACL, Minneapolis, MN, USA, June 2–7, 2019, 4171–4186.

Fan, W., Li, Q., and Cheng, M. (2018). "Deep modeling of social relations for recommendation," in Proceedings of the AAAI, Minneapolis, MN, USA, June 2–7, 2019. (New Orleans, Louisiana, USA: AAAI press), 8075–8076.

Gong, Y., and Zhang, Q. (2016). "Hashtag recommendation using attention-based convolutional neural network," in Proceedings of the IJCAI. 2782–2788.

Gope, J., and Jain, S. K. (2017). "A survey on solving cold start problem in recommender systems," in Proceedings of the ICCCA, Greater Noida, May 5–6, 2017, 133–138.

Gordon, M. A., Duh, K., and Andrews, N. (2020). Compressing bert: studying the effects of weight pruning on transfer learning. Preprint: arXiv:2002.08307.

Guo, L., Wen, Y.-F., and Wang, X.-H. (2018). Exploiting pre-trained network embeddings for recommendations in social networks. *J. Comput. Sci. Technol.* 33, 682–696. doi:10.1007/s11390-018-1849-9

Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., et al. (2020). A survey on knowledge graph-based recommender systems. Preprint: arXiv:2003.00911.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in Proceedings of the CVPR, Las Vegas, NV, June 27–30, 2016, 770–778.

He, R., and McAuley, J. (2016a). "Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering," in Proceedings of the WWW, Montréal Québec, Canada, April 2016, 507–517.

He, R., and McAuley, J. (2016b). "Vbpr: visual bayesian personalized ranking from implicit feedback," in Proceedings of the AAAI, Minneapolis, MN, USA, June 2-7, 2019, 144–150.

Hidasi, B., and Karatzoglou, A. (2018). "Recurrent neural networks with top-k gains for session-based recommendations," in Proceedings of the CIKM, Minneapolis, MN, USA, June 2-7, 2019, 843–852.

Hu, G., Zhang, Y., and Yang, Q. (2018). "Conet: collaborative cross networks for cross-domain recommendation," in Proceedings of the CIKM, Torino, Italy, October 2018, 667–676.

Hu, Y., Koren, Y., and Volinsky, C. (2008). "Collaborative filtering for implicit feedback datasets," in Proceedings of ICDM, Pisa, Italy, December 15–19, 2008, 263–272.

Huang, J., Zhao, W. X., Dou, H., Wen, J.-R., and Chang, E. Y. (2018). "Improving sequential recommendation with knowledge-enhanced memory networks," in Proceedings of the SIGIR, Ann Arbor, MI, July 2018, 505–514.

Ji, G., He, S., Xu, L., Liu, K., and Zhao, J. (2015). "Knowledge graph embedding via dynamic mapping matrix," in Proceedings of ACL-IJCNLP, Beijing, China, December 5-8, 2013 687–696.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). Spanbert: improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics* 8, 64–77. doi:10.1162/tacl_a_00300

Kuchaiev, O., and Ginsburg, B. (2017). Training deep autoencoders for collaborative filtering. preprint: arXiv:1708.01715.

Lee, H., Im, J., Jang, S., Cho, H., and Chung, S. (2019). "Melu: meta-learned user preference estimator for cold-start recommendation," in Proceedings of the SIGKDD. Anchorage, AK, August 2019, 1073–1082.

Li, J., Jing, M., Lu, K., Zhu, L., Yang, Y., and Huang, Z. (2019). From zero-shot learning to cold-start recommendation, *Aaai* 33, 4189–4196. doi:10.1609/aaai.v33i01.33014189

Liang, D., Zhan, M., and Ellis, D. P. (2015). "Content-aware collaborative music recommendation using pre-trained neural networks," in Proceedings of the ISMIR, Malaga, Spain, October 26–30, 295–301.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). "Learning entity and relation embeddings for knowledge graph completion," in Proceedings of the AAAI, Austin, Texas, USA, January 25–30 (Austin, Texas, USA: AAAI Press), 2181–2187.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., et al. (2020). K-bert: enabling language representation with knowledge graph, *Aaai* 34, 2901–2908. doi:10.1609/aaai.v34i03.5681

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al. (2019). Roberta: a robustly optimized bert pre-training approach. Preprint: arXiv:1907.11692.

Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. (2011). "Recommender systems with social regularization," in Proceedings of the WSDM, Hong Kong, China, February 2011, 287–296.

Manotumruksa, J., Macdonald, C., and Ounis, I. (2016). "Regularising factorised models for venue recommendation using friends and their comments," in Proceedings of the CIKM, Indianapolis, IN, October 2016, 1981–1984.

McCarley, J. S. (2019). Pruning a bert-based question answering model. Preprint: arXiv:1910.06360.

McPherson, M., Smith-Lovin, L., and Cook, J. M. (2001). Birds of a feather: homophily in social networks. *Annu. Rev. Soc.* 27, 415–444. doi:10.1146/annurev.soc.27.1.415

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). "Distributed representations of words and phrases and their compositionality," in Proceedings of the NIPS, Red Hook, NY, December 5-8, 2013, 3111–3119.

Mudrakarta, P. K., Sandler, M., Zhmoginov, A., and Howard, A. (2018). "K for the price of 1: parameter-efficient multi-task and transfer learning," in Proceedings of the ICLR, New Orleans, LA, USA, May 6-9. doi:10.18653/v1/p18-1176

Nguyen, H. T. H., Wistuba, M., Grabocka, J., Drumond, L. R., and Schmidt-Thieme, L. (2017). "Personalized deep learning for tag recommendation," in Proceedings of the PAKDD, Jeju, South Korea, May 23–26, (New York, NY: Springer), 186–197. doi:10.1007/978-3-319-57454-7_15

Ni, Y., Ou, D., Liu, S., Li, X., Ou, W., Zeng, A., et al. (2018). "Perceive your users in depth: learning universal user representations from multiple e-commerce

tasks," in Proceedings of the SIGKDD, London, United Kingdom, August 2018, 596–605.

Qin, C., Zhu, H., Zhuang, F., Guo, Q., Zhang, Q., Zhang, L., et al. (2020). A survey on knowledge graph-based recommender systems. *Sci. Sin. Informationis* 50, 937–956. doi:10.1360/SSI-2019-0274

Qiu, X., TianXiang, S., Yige, X., Yunfan, S., Ning, D., and Xuanjing, H. (2020). Pretrained models for natural language processing: a survey. *Sci. China Technol. Sci.* 63, 1872–1897. doi:10.1007/s11431-020-1647-3

Rendle, S. (2010). "Factorization machines," in Proceedings of the ICDM, Sydney, NSW, December 13–17, 2010 (New York, NY: IEEE), 995–1000.

Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2010). "Factorizing personalized Markov chains for next-basket recommendation," in Proceedings of the WWW, Raleigh North Carolina USA, April 2010, 811–820.

Rietzler, A., Stabinger, S., Opitz, P., and Engl, S. (2020). "Adapt or get left behind: domain adaptation through bert language model finetuning for aspect-target sentiment classification," in Proceedings of the LREC, Marseille, 11–16 May 2020, 4933–4941.

Sathish, V., Mehrotra, T., Dhinwa, S., and Das, B. (2019). Graph embedding based hybrid social recommendation system. Preprint: arXiv:1908.09454.

Shi, B., and Weninger, T. (2017). "Proje: embedding projection for knowledge graph completion," in Proceedings of AAAI, San Francisco, California, USA, February 4–9, 1236–1242.

Shi, Y., Larson, M., and Hanjalic, A. (2010). "List-wise learning to rank with matrix factorization for collaborative filtering," in Proceedings of the RecSys, Barcelona, Spain, September 2010, 269–272.

Song, Y., Elkahky, A. M., and He, X. (2016). "Multi-rate deep learning for temporal recommendation," in Proceedings of the SIGIR, Pisa, Italy, July 2016, 909–912.

Stickland, A. C., and Murray, I. (2019). "Bert and pals: projected attention layers for efficient adaptation in multi-task learning," in Proceedings of the ICML, Long Beach, California, USA, June 9–15, 5986–5995.

Su, X., and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Adv. Artif. Intell.* 2009, 421425:1–421425:19. doi:10.1155/2009/421425

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., et al. (2019). "Bert4rec: sequential recommendation with bidirectional encoder representations from transformer," in Proceedings of the CIKM, Beijing, China, November, 2019, 1441–1450.

Tan, J., Wan, X., and Xiao, J. (2016). "A neural network approach to quote recommendation in writings," in Proceedings of the CIKM, Indianapolis, IN, USA, October 24–28, 65–74.

Tang, J., Hu, X., and Liu, H. (2013). Social recommendation: a review. *Soc. Netw. Anal. Min.* 3, 1113–1133. doi:10.1007/s13278-013-0141-9

Tang, J., and Wang, K. (2018). "Personalized top-n sequential recommendation via convolutional sequence embedding," in Proceedings of the WSDM, Marina Del Rey, CA, USA, February 5–9, 565–573.

Tang, X., Wang, T., Yang, H., and Song, H. (2019). "Akupm: attention-enhanced knowledge-aware user preference model for recommendation," in Proceedings of the SIGKDD, Anchorage, AK, USA, August 4–8, 1891–1899.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). "Complex embeddings for simple link prediction," in Proceedings of ICML, New York, NY, June 19–June 24, 2016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). "Attention is all you need," in Proceedings of the NIPS, Long Beach, CA, USA, December 4–9, 5998–6008.

Wang, H., Zhang, F., Hou, M., Xie, X., Guo, M., and Liu, Q. (2018a). "Shine: signed heterogeneous information network embedding for sentiment link prediction," in Proceedings of the WSDM, Marina Del Rey, CA, February 2018, 592–600.

Wang, H., Zhang, F., Xie, X., and Guo, M. (2018b). "Dkn: deep knowledge-aware network for news recommendation," in Proceedings of the WWW, Lyon, France, April, 2018, 1835–1844.

Wang, C., Zhu, H., Zhu, C., Qin, C., and Xiong, H. (2020a). Setrank: a setwise bayesian approach for collaborative ranking from implicit feedback. *Aaai* 34, 6127–6136. doi:10.1609/aaai.v34i04.6077

Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X., Cao, C., et al. (2020b). K-adapter: infusing knowledge into pre-trained models with adapters. Preprint: arXiv:2002.01808.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). "Knowledge graph embedding by translating on hyperplanes," in Proceedings of the AAAI, Quebec City, Quebec, Canada, July 27–31, 1112–1119.

Wen, Y., Guo, L., Chen, Z., and Ma, J. (2018). "Network embedding based recommendation method in social networks," in Proceedings of the WWW, Lyon, France, April 23–27, 11–12.

Wu, C.-Y., Ahmed, A., Beutel, A., Smola, A. J., and Jing, H. (2017). "Recurrent recommender networks," in Proceedings of the WSDM, Cambridge, United Kingdom, February, 6–10, 495–503.

Xing, S., Liu, F. a., Wang, Q., Zhao, X., and Li, T. (2019). A hierarchical attention model for rating prediction by leveraging user and product reviews. *Neurocomputing* 332, 417–427. doi:10.1016/j.neucom.2018.12.027

Xu, Z., Chen, C., Lukasiewicz, T., Miao, Y., and Meng, X. (2016). "Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling," in Proceedings of the CIKM, Indianapolis, IN, USA, October 24–28, 1921–1924.

Yang, J., Xu, J., Tong, J., Gao, S., Guo, J., and Wen, J. (2019). Pretraining of context-aware item representation for next basket recommendation. Preprint: arXiv: 1904.12604.

Yu, F., and Koltun, V. (2016). "Multi-scale context aggregation by dilated convolutions," in Proceedings of ICLR San Juan, Puerto Rico, May 2–4, 2016.

Yu, F., Liu, Q., Wu, S., Wang, L., and Tan, T. (2016). "A dynamic recurrent model for next basket recommendation," in Proceedings of the SIGIR, Pisa, Italy, July 2016, 729–732.

Yu, J., Gao, M., Li, J., Yin, H., and Liu, H. (2018). "Adaptive implicit friends identification over heterogeneous network for social recommendation," in Proceedings of the CIKM, Torino, Italy, October 2018, 357–366.

Yuan, F., He, X., Karatzoglou, A., and Zhang, L. (2020). "Parameter-efficient transfer from sequential behaviors for user modeling and recommendation," in Proceedings of the SIGIR, Virtual Event, China, July 25-30Virtual Event, China, July 25–30, 1469–1478.

Zhang, F., Yuan, N. J., Lian, D., Xie, X., and Ma, W.-Y. (2016). "Collaborative knowledge base embedding for recommender systems," in Proceedings of the SIGKDD, San Francisco, CA, August 2016, 353–362.

Zhang, M., Hu, B., Shi, C., Wu, B., and Wang, B. (2018). "Matrix factorization meets social network embedding for rating prediction," in Proceedings of the APWeb-WAIM, Macau, China, July 23–25 (New York, NY: Springer), 121–129. doi:10.1007/978-3-319-96890-2_10

Zhang, Y., Ai, Q., Chen, X., and Croft, W. B. (2017). "Joint representation learning for top-n recommendation with heterogeneous information sources," in Proceedings of the CIKM, Singapore, November 2017, 1449–1458.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. (2019). "Ernie: enhanced language representation with informative entities," in Proceedings of the ACL, Florence, Italy, July 28–August 2, 1441–1451.

Zhao, X., Zhang, L., Xia, L., Ding, Z., Yin, D., and Tang, J. (2017). Deep reinforcement learning for list-wise recommendations. Preprint: arXiv:1801.00209.

Zheng, L., Noroozi, V., and Yu, P. S. (2017). "Joint deep modeling of users and items using reviews for recommendation," in Proceedings of the WSDM, Cambridge, United Kingdom, February 2017, 425–434.

Zheng, Y., Tang, B., Ding, W., and Zhou, H. (2016). A neural autoregressive approach to collaborative filtering. Preprint: arXiv:1605.09477.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.