Neural Re-ranking in Multi-stage Recommender Systems: A Review

Weiwen Liu\textsuperscript{1}, Yunjia Xi\textsuperscript{2}, Jiarui Qin\textsuperscript{2}, Fei Sun\textsuperscript{3}, Bo Chen\textsuperscript{1}, Weinan Zhang\textsuperscript{2}, Rui Zhang\textsuperscript{1}\textsuperscript{*} and Ruiming Tang\textsuperscript{1}\textsuperscript{*}

\textsuperscript{1}Huawei Noah’s Ark Lab
\textsuperscript{2}Shanghai Jiao Tong University
\textsuperscript{3}Institute of Computing Technology, Chinese Academy of Sciences
ruizhang.info

{liuweiwen8,chenbo116,tangruiming}@huawei.com,
{xijunia,qjr1996,wnzhang}@sjtu.edu.com, rayteam@yeah.net

Abstract

As the final stage of the multi-stage recommender system (MRS), re-ranking directly affects users’ experience and satisfaction by rearranging the input ranking lists, and thereby plays a critical role in MRS. With the advances in deep learning, neural re-ranking has become a trending topic and been widely adopted in industrial applications. This review aims at integrating re-ranking algorithms into a broader picture, and paving ways for more comprehensive solutions for future research. For this purpose, we first present a taxonomy of current methods on neural re-ranking. Then we give a description of these methods along with the historic development according to their objectives. The network structure, personalization, and complexity are also discussed and compared. Next, we provide a benchmark for the major neural re-ranking models and quantitatively analyze their re-ranking performance. Finally, the review concludes with a discussion on future prospects of this field. A list of papers discussed in this review, the benchmark datasets, our re-ranking library LibRerank, and detailed parameter settings are publicly available at https://github.com/LibRerank-Community/LibRerank.

1 Introduction

Multi-stage Recommender Systems (MRS) are widely adopted by many of today’s largest online platforms, including Google [Bello et al., 2018], YouTube [Wilhelm et al., 2018], LinkedIn [Geyik et al., 2019], and Taobao [Pei et al., 2019]. MRS is a natural solution to the computational limits in practical recommendation applications, where the numbers of users and items grow into billions. The recommendation task is split into multiple steps in MRS—each step narrows down the relevant items with a slower but more accurate model [Hron et al., 2021], to guarantee low response latency. A common structure for MRS consists of three stages

\textsuperscript{*}Corresponding authors.
architectures. Neural re-ranking models get rid of the handcrafted features and benefit from the automatic learning of the listwise context, due to the universal approximation property of neural networks [Cybenko, 1989]. Therefore, applying neural networks for re-ranking is the main focus in both academia and industry in recent years. This paper provides a first review on neural re-ranking for recommendation.

1.1 A Taxonomy

We differentiate neural re-ranking models by objectives (single accuracy objective or multiple objectives) and the supervision signals (observed signals or counterfactual signals). The resulting four quadrants are outlined in Fig. 1.

Considering the objectives, most studies focus on the single accuracy objective, as accurately predicting users’ interests is the foundation of recommender systems. While beyond accuracy, several other objectives are also desired from a re-ranking model like diversity or fairness [Kaminskas and Bridge, 2016], thus leading to recent work on how to optimize multiple objectives in re-ranking and better manage the tradeoff between them.

Another important factor that separates different re-ranking models is the supervision signals for relevance. Most work is directly trained by the displayed initial ranking lists and the corresponding observed labels. Some other work, however, points out that the relevance of each item depends on the listwise context, and different permutations of the input list yield different relevance labels [Feng et al., 2021a]. The supervision signals are therefore provided by an extra evaluator on the unobserved counterfactual permutations that have not been actually displayed to the user, to model the listwise context under different permutations.

From Fig. 1, we observe the following development characteristics of existing neural re-ranking work: (i) Most studies seek to purely enhance accuracy with a single accuracy objective, while diversity/fairness-aware methods with multi-objectives have been relatively less explored. (ii) Self-attention [Vaswani et al., 2017] or a combination of RNN [Hochreiter and Schmidhuber, 1997] and the attention have become popular network structures in re-ranking. (iii) Few works discuss the influence of counterfactual permutations on relevance in multi-objective learning (the first quadrant of Fig. 1), which could be a potential research direction. We will elaborate on the details of each method according to our proposed taxonomy in the following sections.

2 Neural Re-ranking for Recommendation

Neural re-ranking usually aims to construct a multivariate scoring function, whose input is a whole list of items from the initial ranking, to model the listwise context/cross-item interactions [Pang et al., 2020; Ai et al., 2019]. This is in contrast to ranking models where the ranking functions are mostly univariate that take one item at a time, and the correlations between items are only modeled at loss level using pairwise or listwise loss functions [Xia et al., 2008].

For a specific user, given the initial list \( R \) of \( n \) items, and the corresponding supervision signals \( Y \in \mathbb{R}^n \), a neural re-ranking problem is to find the optimal ranking function \( \phi_x \) that maps the input to a list of re-ranking scores as

\[
\phi_x = \arg \min_{\phi} \sum_{R,Y} \mathcal{L}(Y, \phi(R)),
\]

where \( \mathcal{L}(\cdot) \) is the loss function. The major goal of the re-ranking is to optimize accuracy, which is usually measured by ranking metrics like NDCG or MAP. While beyond accuracy, encouraging diversity or fairness of the re-ranking is also one of the critical goals.

This general formulation of Eq.(1) provide another perspective to describe our proposed taxonomy in Fig. 1. The design of the loss function, either is purely accuracy-oriented or a combination of multiple objectives, differentiates re-ranking models into the single objective and the multi-objective ones. On the other hand, whether the supervision signal \( Y \) comes from the data log or an evaluator, separates re-ranking models into learning by observed signals or by counterfactual signals. Fig. 2 shows typical network architectures for re-ranking. Learning by observed signals usually follows a direct architecture, outputting re-ranking scores with listwise context modeling. Whereas learning by counterfactual signals generally adopts a generator-evaluator paradigm—the generator generates re-ranking lists under the guidance of an evaluator, where both the generator and the evaluator attend to the listwise contexts. Later we will introduce different neural re-ranking models according to the four quadrants in detail.

3 Single Objective: Accuracy-oriented

Recommendation accuracy of the re-ranking model is the fundamental goal for MRS, and the evaluation of a re-ranking model is usually the overall listwise utility like NDCG or MAP of the re-ranking list.

According to the supervision signal, we further divide existing re-ranking models into two groups: learning by observed signals and learning by counterfactual signals. Learning by observed signals directly uses the initial ranking list \( R \) and the corresponding label \( Y \), which is actually displayed to the user and obtained feedback, to train the model. On the contrary, learning by counterfactual signals presumes the
item’s relevance varies under different permutations—even with the same items, users respond distinctly to different permutations of these items. Therefore, they introduce an additional evaluator to provide signals for counterfactual permutations that have not been actually displayed to the user. Listwise context is thereby estimated on the counterfactual permutations. Below we describe various attempts with their advantages and disadvantages for methods of learning by observed signals and learning by counterfactual signals.

3.1 Learning by Observed Signals

Learning by observed signals is simple and straightforward. A typical architecture of the re-ranking model for learning by observed signals can be outlined as in Fig. 2(a), which firstly embeds user and item features into low-dimensional dense vectors, and then extracts cross-item interactions by the listwise context modeling to generate the re-ranking scores. The observed labels are actual feedback from users, and thus are less noisy and easier to train. Moreover, the initial list provides strong signals for items’ relevance estimated by previous ranking models. Several existing studies have shown the effectiveness of directly learning by observed signals [Ai et al., 2018; Pei et al., 2019; Pang et al., 2020; Feng et al., 2021a]. By the network structure adopted to model the listwise context, we further classify the existing methods into recurrent listwise modeling with recurrent neural networks (RNN), attentive listwise modeling with self-attention, and others like multi-layer perceptrons (MLP), graph neural networks (GNN), etc., where different network structures are also plotted in Fig. 1 by different colors.

Recurrent listwise modeling. As one of the earliest neural re-ranking methods for improving accuracy, DLCM [Ai et al., 2018] uses gated recurrent units (GRU) to sequentially encode the top-ranked items with their feature vectors. The recurrent unit combines the information for the current item with previous items, which naturally captures the sequential dependencies among items and the positional effect of the initial list. MiDNN [Zhuang et al., 2018] also applies recurrent networks, the long-short term memory (LSTM), with a global feature extension mechanism to capture cross-item influences. It formulates the re-ranking as a sequence generation problem, and sequentially selects the next items with beam search to conform to the users’ browsing habit. Seq2Slate [Bello et al., 2018] extends MiDNN by adopting a more flexible pointer network to solve the re-ranking problem. The pointer network produces the next item with an attention mechanism, attending to the items in the initial list.

Attentive listwise modeling. Lately, inspired by the success of the self-attention architecture used in natural language processing [Vaswani et al., 2017], several re-ranking models that apply the multi-head self-attention are proposed. Compared to RNN, the self-attention mechanism directly models the interactions between any pair of candidate items without degradation over the encoding distance. PRM [Pei et al., 2019] is a generally straightforward adaptation of the self-attention structure, which is a stack of multiple blocks of self-attention layers and feed-forward networks with position embeddings of the initial list. A pretrained personalized embedding is used to extract user-specific mutual influences between candidate items. PFRN [Huang et al., 2020] employs multiple self-attention structures for flight itinerary re-ranking. Instead of a simple concatenation of a pre-trained user representation with the candidate item representation as in PRM, PFRN exploits users’ multiple behaviors, like long-term booking behaviors, real-time clicking behaviors, by individual multi-head self-attentions. The final prediction is generated by capturing the interactions between candidate items and users’ multiple behaviors. Raiss [Lin et al., 2022] attempts to improve personalization in re-ranking by maintaining individual attention weights in modeling cross-item interactions for each user. A more recent work PEAR [Li et al., 2022] proposes to model cross-item interactions between both the initial list and the users’ historical clicked items by a designed cross-attention structure. The feature-level interactions and an auxiliary classification task are also incorporated to improve the performance for re-ranking.

Other network structures. To avoid potential position or contextual bias, List-CVAE [Jiang et al., 2018] further explores conditional variational auto-encoders (CVAE) and directly learns the joint distribution of items conditioned on user responses. Liu et al. [2021] find that the generative model of List-CVAE is usually trapped in a few items and fails to cover item variation in the re-ranking list. They propose a pivot selection phase (PivotCVAE) to improve the variation of the list. HRM [Li et al., 2019] finds that introducing user behaviors and computing the similarity between candidate items and interested items in history improves the quality of re-ranking. Liu et al. [2020b] further investigate the complementary and substitutable relationships among candidate items and propose a graph-based model, IRGPR.

Above we have provided a brief review of the development for methods of learning by observed signals. Most of the work formulates the re-ranking as a sequential modeling problem to extract the cross-item interactions on initial ranking lists. We also witness a trend of evolving from recurrent to self-attentive structures. However, despite the various network structures, the above models are only trained with the only permutation that is displayed to the users, with other $n! - 1$ permutations unexplored, limiting the potential of selecting the optimal permutation for re-ranking. In the next section, we will focus on learning by counterfactual signals that estimates listwise contexts on different permutations.

3.2 Learning by Counterfactual Signals

To provide signals on counterfactual lists, methods that learn by counterfactual signals usually follow an evaluator-generator paradigm (EG)—with a generator to generate feasible permutations and an evaluator to evaluate the listwise utility of each permutation, as shown in Fig. 2(b).

Wang et al. [2019] first adopt the evaluator-generator paradigm and propose the SEG model. They point out two desired properties for an evaluator: (i) Order-sensitivity, the evaluator needs to be sensitive to the order of the input lists; and (ii) Generalizability, the evaluator shall generalize well to all possible permutations. Under the guidance of the evaluator, SEG devises both a supervised learning approach
and a reinforcement learning approach to train the generator. The supervised learning approach directly learns the estimated utility provided by the evaluator, while the reinforcement learning approach further pursues long-term reward in each step by the temporal difference (TD) error [Silver et al., 2014].

A series of follow-up studies [Wei et al., 2020; Feng et al., 2021a; Feng et al., 2021b; Xi et al., 2021] explores various network structures like RNN or self-attention for evaluators and the generators. Specific structures are listed in Table 1. Though the structures for the generator diverge, we observe that a common choice for the evaluator is the RNN-based structure, due to its satisfying performance in modeling users’ sequential behaviors [Borisov et al., 2016].

In addition, these studies also focus on improving the training procedure of the generator. CDIA [Song et al., 2020] applies the actor-critic reinforcement learning and uses policy gradient with the advantage function to update the generator. To tackle the problem of large action space of \(O(nl)\) for the generator, Wei et al. [2020] adapt the proximal policy optimization (PPO) algorithm and introduce PPO-exploration to train the generator in the proposed GCR model. PRS [Feng et al., 2021a] exploits the beam search to generate feasible permutations and directly uses the evaluator to select the optimal list. GRN [Feng et al., 2021b] also employs the policy gradient for optimization, whereas CRUM [Xi et al., 2021] utilizes LambdaLoss to train the generator for utility optimization. Huzhang et al. [2021] notice that the evaluator is trained by the offline labeled data and may not generalize well to unseen distribution, and introduce EGRerank with a discriminator to provide a self-confidence score for the evaluation.

A few exceptions without the evaluator-generator architecture are GSF [Ai et al., 2019], SetRank [Pang et al., 2020], and MIR [Xi et al., 2022], which learns permutation-invariant re-ranking models that are insensitive to permutations of the input. A groupwise scoring function (GSF) [Ai et al., 2019] is devised with DNN on all the size-\(m\) permutations of items in initial lists \((m \leq n)\). Pang et al. [2020] apply a variant of self-attention structure without positional encoding and dropout (SetRank) to preserve the permutation invariant property. MIR [Xi et al., 2022] introduces user behavior history to the permutation-invariant re-ranking, and considers various interactions between the candidate items and items in the users’ behavior history.

Though potentially effective in selecting the optimal re-ranking list by modeling counterfactual permutations, the training procedure is often more complex and the performance depends greatly on the quality of the evaluator.

### 3.3 Qualitative Model Comparison

Next, we give a thorough comparison of the above-mentioned models in terms of network structure, optimization, personalization, and computational complexity, as shown in Table 1.

**Network structure.** We notice that using self-attention, or a combination of RNN and the attention mechanism in re-ranking has been especially popular in recent years. For the design of the evaluator, bi-directional RNN (e.g., BiLSTM, BiGRU) is proved to be more effective in many studies [Wang et al., 2019; Feng et al., 2021a; Xi et al., 2021], where BiRNN is capable of capturing the two-way evolution of user’s interests during browsing [Feng et al., 2021b].

**Optimization.** For those models that learn by observed signals (the upper table), the loss function can be broadly grouped into pointwise (cross-entropy loss (CE)), pairwise (BPR loss [Rendle et al., 2009], hinge loss [Bartlett and Wegkamp, 2008]), and listwise (Attention Rank loss (AtRank) [Ai et al., 2018], KL loss). The pointwise CE loss is the most adopted loss due to its simplicity and effectiveness. Methods that learn by counterfactual signals often follow the evaluator-generator paradigm, where the training of the generator is guided by the evaluator. Since the ranking operation is discrete and non-differentiable, these models often rely on the policy gradient [Silver et al., 2014] or LambdaLoss [Wang et al., 2018] to optimize the model.

**Personalization.** Re-ranking results should be user-specific and cater to individual users’ preferences and intents. Moreover, the cross-item interactions of item pairs vary from user to user. Thus personalization is an essential requirement for re-ranking. We observe a trend of emphasizing personalized models over non-personalized ones in recent years. There are mainly two ways to provide personalized re-ranking results: (i) personalization by input data and (ii) personalization by model parameters. The former way simply takes user features as input, e.g., the user profiles or the user historical behaviors, and extracts personal preferences by specific network architectures like self-attention in PFRN [Huang et al., 2020]. The network parameters are shared across users. While the latter maintains an individual set of parameters for each user as in Raiss [Lin et al., 2022] or IRGPR [Liu et al., 2020b].

**Complexity.** Learning by observed signals directly predicts
the re-ranking scores for \( n \) items and are mainly of the linear time complexity \( O(n) \), except for the ones that apply the attentive listwise modeling. The runtime for the self-attention is quadratic in \( n \) as it computes the interactions between any pair of items, but the calculation can be made parallel to accelerate the process [Vaswani et al., 2017]. Seq2Slate [Bello et al., 2018] also yields a \( O(n^2) \) complexity—for each one of the \( n \) steps, Seq2Slate examines the current remaining items and selects the best one from them.

As for learning by counterfactual signals, GSF takes \( m \) permutations of \( n \) items as input so that the complexity is \( O \left( \frac{m n!}{n} \right) \). For most evaluator-generator models, the generators sequentially select the next item similar to Seq2Slate, leading to a polynomial complexity \( O(n^2) \). CRUM [Xi et al., 2021], on the other hand, though have a \( O(n^2) \) training time, the time complexity for inference is \( O(n) \) by directly predicting the re-ranking scores for all the \( n \) items with an MLP structure.

4 Multiple Objectives

Accuracy is no doubt the most important objective for recommender systems. Apart from accuracy, other objectives like diversity or fairness are also crucial measurements in MRS. Purely optimizing accuracy, if applied carelessly, can yield similar or near-duplicate results and further result in the echo chamber effects [Ge et al., 2020]. Many studies aim to simultaneously optimize accuracy and other objectives (diversity/fairness). How to delicately manage the tradeoff between multiple objectives becomes a key problem, as sometimes diversity and fairness can contradict accuracy [Liu et al., 2019].

We introduce diversity-aware and fairness-aware re-ranking in Section 4.1 and 4.2, respectively.

4.1 Diversity-aware Re-ranking

Diversity usually measures the dissimilarity of the re-ranking list for each user. In contrast to non-learning re-ranking methods like MMR [Carbonell and Goldstein, 1998], neural diversity-aware models usually conduct an end-to-end learning scheme, with no need for handcrafting relevance and diversity features. Below we give a brief review of neural diversity-aware re-ranking by broadly classifying existing studies into implicit approaches and explicit approaches. The implicit approaches measure diversity by inter-item similarity and do not require subtopics (e.g., category of items) to evaluate diversity, while explicit approaches aim at promoting the coverage of items over specified subtopics.

For implicit approaches, NTN [Xia et al., 2016] proposes a neural tensor network to learn the dissimilarity between any pairs of items. The re-ranking list is sequentially generated by a linear combination of the relevance and the dissimilarity of candidate items. MDP-DIV [Xia et al., 2017] directly optimizes general diversity measures like \( \alpha \)-DCG or \( S \)-recall, and uses the policy gradient to optimize the long-term reward. MF-DIV [Feng et al., 2018] enhances MDP-DIV by introducing LSTM and the lookahead Monte Carlo Tree Search (MCTS) to the ranking policy. Yan et al. [2021] derive a smooth approximation of diversity metrics in the proposed DALETOR model and apply a self-attention structure to model the listwise context.

While for explicit approaches, Jiang et al. [2017] notice the advantage of using the attention mechanism to determine the importance of the under-covered subtopics, and propose DSSA, where the relevance and the diversity are jointly estimated with a subtopic attention. As a follow-up to DSSA, DVGAN [Liu et al., 2020a] formulates the problem of generating diverse re-ranking lists as a minimax game. It adapts DSSA as a generator, and involves a discriminator to determine how relevant and diverse the given list is. DESA [Qin et al., 2020] explores leveraging item dependencies in terms of both relevance and diversity, which is composed of an encoder and a decoder with the self-attention to extract item and subtopic correlations. Abdool et al. [2020] investigate the potential of deploying a diversity-aware re-ranking to Airbnb search. They design a metric for measuring the distance between two lists and use an LSTM structure to generate the re-ranking list.

The balance between accuracy and fairness is managed either by learning a trade-off parameter [Xia et al., 2016], or directly optimizing a specific metric that combines accuracy and fairness like \( \alpha \)-NDCG [Yan et al., 2021].

4.2 Fairness-aware Re-ranking

Fairness, with a growing influence on IR community, has been made a critical objective for re-ranking. In this review, we focus on the item fairness, since it is the main focus of existing re-ranking literature. Item fairness ensures each item or item group receives a fair proportion of exposure (e.g., proportional to its merits or utility). Neural re-ranking, however, has been a relatively under-explored domain. FMMR [Karako and Manggala, 2018] first constructs fairness representation for each demographic group using CNN and adopts MMR to trade-off relevance and fairness. Singh and Joachims [2019] aim to optimize a general utility metric while satisfying the fairness of exposure constraints by the Plackett-Luce model [Plackett, 1975] in PGRank. A follow-up study, PLRank [Oosterhuis, 2021], improves the policy gradient in PGRank by deriving an unbiased estimate of the gradient. FULTR [Yadav et al., 2021] further explores a counterfactual estimate for both utility and fairness constraints for the Plackett-Luce model. Zhu et al. [2021] empirically show the prevalence of unfairness in cold-start recommendation, and propose an auto-encoder re-ranking model, GEN, to alleviate the fairness issue for cold-start items.

5 Emerging Applications

Neural re-ranking has also been seen in many emerging and interesting industrial applications.

5.1 Integrated Re-ranking

Integrated re-ranking (a.k.a., mixed re-ranking) is a rapidly emerging domain driven by practical problems, where the MRS is required to display a mix of items from different sources/channels with heterogeneous features e.g., integrated feeds of articles, videos, and news [Xie et al., 2021]. The input is extended from a single list to multiple lists. DHANR [Hao et al., 2021] proposes a hierarchical self-attention structure to consider cross-channel interactions. Xie et al. [2021]
decompose the integrated re-ranking problem into two sub-tasks—source selection and item ranking, and use hierarchical reinforcement learning (HRL) to solve the problem. DEAR [Zhao et al., 2021; Zhao et al., 2020] learns to inter-polate ads and organic items by the designed deep Q-networks. Liao et al. [2021] also adopts a reinforcement learning solution with a cross-channel attention unit.

5.2 Edge Re-ranking

In a framework of cloud-to-edge, Gong et al. [2020] find that real-time computing on edge helps capture user preferences more delicately and improve the performance of recommendations. Therefore, they propose EdgeRec, which generates initial ranking lists on cloud, and conducts re-ranking with instant feedback on mobile devices. Edge re-ranking opens up interesting research topics especially for on-device personalized models or federated learning [Hard et al., 2018].

6 Experiments

For understanding and analyzing the performance of re-ranking algorithms, we provide a re-ranking library—LibRerank, which automates the re-ranking experimentation and integrates a major collection of re-ranking algorithms. It is designed to support researchers by simplified access to popular re-ranking algorithms, thereby making experimental results more reproducible.

We conduct benchmarking experiments on two public recommendation datasets, Ad and PRM Public. A detailed explanation of the benchmarking experiments, and the processed datasets are also released together with the LibRerank library. We use LambdaMART [Burges et al., 2010] to produce the initial ranking lists. Baselines include: MiDNN [Zhuang et al., 2018], GSF [Ai et al., 2019], DLCM [Ai et al., 2018], PRM [Pei et al., 2019], SetRank [Pang et al., 2020], and EGRerank [Hu et al., 2021]. We anticipate adding support for more re-ranking algorithms, including diversity- or fairness-aware ones in the near future.

Principles. For fair comparisons, our implementation follows several principles: (i) To cover every detail in each algorithm, we use the open-sourced implementation if applicable. Otherwise, the algorithms are implemented according to the original paper. (ii) We conduct careful parameter tuning for every algorithm and report the best results.

6.1 Quantitative Evaluation

For the quantitative evaluation, we focus on the popular ranking metrics MAP@k and NDCG@k, with k = 5, 10 for Ad, k = 10, 20 for PRM Public due to the different re-ranking sizes. The results are reported in Table 2, from which we have the following observations.

(i) Effectiveness of Re-ranking. The first row in Table 2 shows the performance of the initial ranking, generated in the ranking stage by LambdaMART. The results of all the re-ranking algorithms are appealing and outperform the initial ranker by a large margin. This confirms the necessity of the re-ranking stage in MRS by integrating the listwise context.

(ii) Listwise Context Modeling. Considering re-ranking algorithms with different listwise context modeling structure, algorithms with self-attention architecture like SetRank and PRM, achieves better results. It is because the self-attention structure effectively encodes the cross-item interactions between any pairs of items.

(iii) Robustness of EG framework. EGRerank adopts an evaluator-generator (EG) paradigm, but its performance is less impressive. Possible reasons may be that the performance of the generator largely depends on the quality of the evaluator, which is relatively hard to measure and select.

7 Summary and Future Prospects

Over the past several years, neural re-ranking has continued to become an inspiring domain, motivated by both scientific challenges and industrial demands. A considerable amount of studies have been conducted, and many of them have already found use in industrial applications. Major advances in this domain are summarized in Fig. 1. A review of the re-ranking algorithms with corresponding objectives can be found in Section 3 and 4. Our benchmarking results manifest the superiority of the neural re-ranking models. Despite the great progress in recent years, we still note that there are some significant challenges and open issues in this domain.

Sparse Feedback. The re-ranking problem is challenging due to the sparse supervised signal, where only the feedback for the displayed lists can be observed—feedback for the other n! − 1 permutations is unavailable. Evaluators or click models [Borisov et al., 2016; Chen et al., 2020; Zhang et al., 2021] can be potentially used to generate feedback, but current click models are just trained to fit the offline

<table>
<thead>
<tr>
<th></th>
<th>Ad</th>
<th>PRM Public</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP@5</td>
<td>NDCG@5</td>
</tr>
<tr>
<td>Init [2010]</td>
<td>0.6037</td>
<td>0.6840</td>
</tr>
<tr>
<td>MiDNN [2018]</td>
<td>0.6080</td>
<td>0.6876</td>
</tr>
<tr>
<td>GSF [2019]</td>
<td>0.6090</td>
<td>0.6883</td>
</tr>
<tr>
<td>EGRerank [2021]</td>
<td>0.6092</td>
<td>0.6890</td>
</tr>
<tr>
<td>DLCM [2018]</td>
<td>0.6126</td>
<td>0.6914</td>
</tr>
<tr>
<td>SetRank [2020]</td>
<td>0.6132</td>
<td>0.6917</td>
</tr>
<tr>
<td>PRM [2019]</td>
<td>0.6140</td>
<td>0.6923</td>
</tr>
<tr>
<td>Init [2010]</td>
<td>0.6037</td>
<td>0.6840</td>
</tr>
<tr>
<td>GSF [2019]</td>
<td>0.6090</td>
<td>0.6883</td>
</tr>
<tr>
<td>EGRerank [2021]</td>
<td>0.6092</td>
<td>0.6890</td>
</tr>
<tr>
<td>DLCM [2018]</td>
<td>0.6126</td>
<td>0.6914</td>
</tr>
<tr>
<td>SetRank [2020]</td>
<td>0.6132</td>
<td>0.6917</td>
</tr>
<tr>
<td>PRM [2019]</td>
<td>0.6140</td>
<td>0.6923</td>
</tr>
</tbody>
</table>

Table 2: Performance Comparison on Ad and PRM Public datasets. The initial ranking list (Init) is produced by LambdaMart.
click data by performance metrics like log-likelihood, without particular designs on how evaluators should be trained to address the data sparsity problem and help improve the training of the re-ranking models.

**Personalization for Diversity/Fairness.** Personalization is the core of MRS, but recent literature mostly focuses on personalization in accuracy-oriented re-ranking, leaving personalization in diversity and fairness unexplored. Different users have various demands for diversity and fairness. It is of great potential to involve personalized diversity or fairness.

**Tradeoff between Multiple Objectives.** Different recommendation scenarios have different degrees of demand for diversity or fairness. Existing studies mainly manage the trade-off by heuristics or parameter tuning. It could be a promising topic to automatically balance multiple objectives without human intervention.

**Joint Training of MRS.** Re-ranking models are trained separately, decoupling from other stages in MRS. But the ranking quality of other stages affects the performance of re-ranking [Bello et al., 2018; Qin et al., 2022]. Utilizing the information learned by other stages (e.g., using parameter transfer, gradient transfer) would be of high value for both academia and industry.

Acknowledgments

The SJTU team is supported by “New Generation of AI 2030” Major Project (2018AAA0100900), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and National Natural Science Foundation of China (62076161, 62177033). The work is also sponsored by Huawei Innovation Research Program. The author Jiawei Qin is supported by Wu Wen-Jun Honorary Doctoral Scholarship, AI Institute, SJTU. We thank MindSpore [MindSpore, 2020], a new deep learning computing framework, for the partial support of this work.

References


