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

Weiwen Liu[1], **Yunjia Xi**[2], **Jiarui Qin**[2], **Fei Sun**[3], **Bo Chen**[1], **Weinan Zhang**[2], **Rui Zhang**[4], **Ruiming Tang**[1]

[1]Huawei Noah's Ark Lab [2]Shanghai Jiao Tong University
[3]DAMO Academy, Alibaba Group [4] ruizhang.info
{liuweiwen8,chenbo116,tangruiming}@huawei.com,
{xiyunjia,qinjr96,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 applied 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 benchmarks of 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[1].

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 in general: candidates generation (*a.k.a.,* recall or matching), ranking, and re-ranking. The system firstly generates candidates from a large pool of items. Then these candidates are scored and ranked in the ranking stage. Finally, the system conducts re-ranking on the top candidates based on certain rules or objectives to further improve the recommendation results. Specifically, the re-ranking stage takes as input the initial ranking list from the ranking stage, and outputs a reordered list by considering the listwise context (cross-item interactions). Whether a user is interested in an item is not only determined by the item itself, but also by other items placed in the same list (*i.e.*, the listwise context) [Pei *et al.*, 2019]. Thus, a key technical challenge is to model the listwise context in re-ranking.

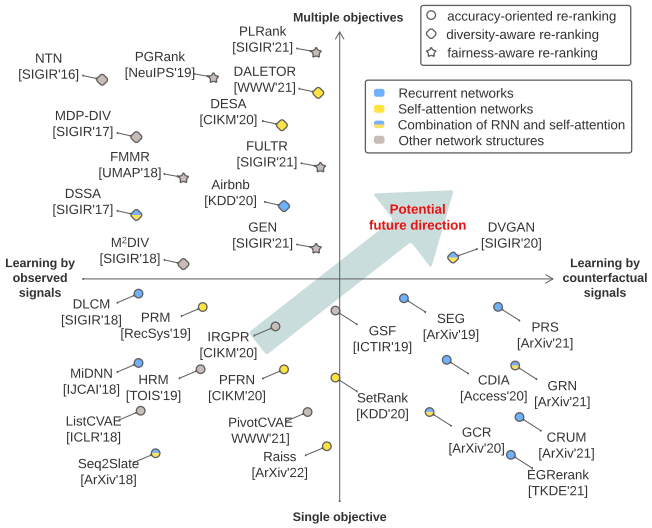

Figure 1: Four quadrants of neural re-ranking models. Shapes denote different objectives, and colors represent the major network structures for re-ranking.

Re-ranking dates back to Carbonell's work [1998], which greedily adds items to the list with maximal marginal relevance. With deep neural networks led to exciting breakthroughs in various fields [Goodfellow *et al.*, 2016; Goldberg, 2017], re-ranking methods evolved to the recent deep neural 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]https://github.com/LibRerank-Community/LibRerank

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, 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 ϕ_* that maps the input to a list of re-ranking scores as

$$\phi_* = \arg\min_{\Phi} \sum_{R,Y} \mathcal{L}(Y, \phi(R)) \,, \tag{1}$$

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.*, 2021b]. 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 method 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 to 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. A more recent work 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.

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.

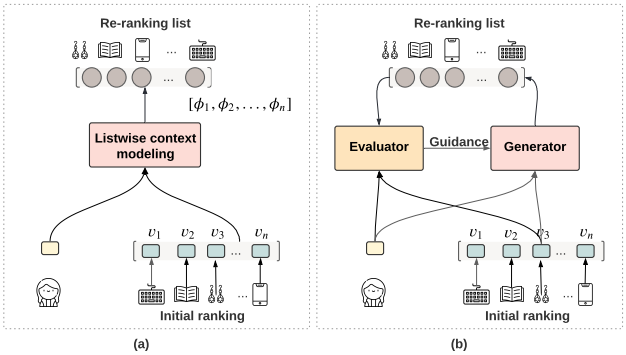

Figure 2: Re-ranking network architectures. (a) A typical neural re-ranking architecture for learning by observed signals. (b) The evaluator-generator paradigm for learning by counterfactual signals.

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) *Generalizaility*, the evaluator shall generalize well to all possible permutations. Under the guidance of the evaluator, SEG devises 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.*, 2021c; 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

Table 1: Comparison of accuracy-oriented re-ranking models. D/M/NP stands for personalized by input data/by model parameters/non-personalized, E/G represents evaluator/generator, n is the re-ranking size, h is the length of the user history, and m is the permutation size for GSF.

	Listwise context modeling	Optimization	P/NP	Complexity
DLCM [2018]	GRU	AttRank	NP	$\mathcal{O}(n)$
MiDNN[2018]	LSTM	CE	D	$\mathcal{O}(n)$
ListCVAE [2018]	CVAE	KL	NP	$\mathcal{O}(n)$
Seq2Slate [2018]	PointerNet	CE	NP	$\mathcal{O}(n^2)$
HRM [2019]	Similarity	Hinge	D	$\mathcal{O}(hn + h^2)$
PRM [2019]	Self-attention	CE	D	$\mathcal{O}(n^2)$
IRGPR [2020b]	GNN	BPR	M	$\mathcal{O}(n)$
PFRN [2020]	Self-attention	CE	D	$\mathcal{O}(n^2 + h^2)$
PivotCVAE [2021]	CVAE	KL	NP	$\mathcal{O}(n)$
Raise [2022]	Self-attention	CE	M	$\mathcal{O}(n^2)$
GSF [2019]	DNN	CE	NP	$O(\frac{mn!}{(n-m)!})$
SEG [2019]	E: BiGRU G: GRU	MSE/Q-learning	D	$\mathcal{O}(n^2)$
SetRank [2020]	Self-attention	AttRank	NP	$\mathcal{O}(n^2)$
CDIA [2020]	E: LSTM G: LSTM	Policy gradient	D	$\mathcal{O}(n^2)$
GCR [2020]	E: BiGRU+attention G: GRU	PPO-exploration	D	$\mathcal{O}(n^2)$
PRS [2021a]]	E: BiLSTM G: Beam search	—	D	$\mathcal{O}(n^2)$
GRN [2021c]	E: BiLSTM+attention G: GRU+attention+ PointerNet	Policy gradient	D	$\mathcal{O}(n^2)$
CRUM [2021]	E: BiLSTM+GNN G: MLP	LambdaLoss	D	$\mathcal{O}(n)$
EGRerank [2021]	E: LSTM G: LSTM	PPO	D	$\mathcal{O}(n^2)$

gradient with the advantage function to update the generator. To tackle the problem of large action space of $O(n!)$ 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.*, 2021c] 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.

Two exceptions without the evaluator-generator architecture are GSF [Ai *et al.*, 2019] and SetRank [Pang *et al.*, 2020], which learns permutation-invariant re-ranking models that are insensitive to permutations of the input. A group-wise 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.

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.*, 2021c].

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.*, 2012], hinge loss [Bartlett and Wegkamp, 2008]), and listwise (Attention Rank loss (AttRank) [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 $\mathcal{O}(n)$, 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 $\mathcal{O}(n^2)$ 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 $O(\frac{mn!}{(n-m)!})$. For most evaluator-generator models, the generators sequentially select the next item similar to Seq2Slate, leading to a polynomial complexity $\mathcal{O}(n^2)$. CRUM [Xi *et al.*, 2021], on the other hand, though have a $\mathcal{O}(n^2)$ training time, the time complexity for inference is $\mathcal{O}(n)$ 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 of 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 α-DCG or S-recall, and uses the policy gradient to optimize the long-term reward. M^{2}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 for 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 to leverage 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 α-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.

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 subtasks—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 interpolate 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.

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 experimen-

Table 2: Performance Comparison on Ad and PRM Public datasets. The initial ranking list (Init) is produced by LambdaMart.

	Ad				PRM Public			
	MAP@5	NDCG@5	MAP@10	NDCG@10	MAP@10	NDCG@10	MAP@20	NDCG@20
Init [2010]	0.6037	0.6840	0.6075	0.6990	0.1842	0.2178	0.1901	0.3202
MiDNN [2018]	0.6080	0.6876	0.6117	0.7021	0.3069	0.3482	0.2977	0.4265
GSF [2019]	0.6090	0.6883	0.6126	0.7028	0.3060	0.3459	0.2968	0.4241
EGRerank [2021]	0.6092	0.6890	0.6126	0.7029	0.3075	0.3502	0.2985	0.4286
DLCM [2018]	0.6126	0.6914	0.6162	0.7055	0.3082	0.3500	0.2991	0.4287
SetRank [2020]	0.6132	0.6917	0.6168	0.7060	0.3094	0.3515	0.3002	0.4297
PRM [2019]	0.6140	0.6923	0.6178	0.7066	0.3096	0.3516	0.3003	0.4301

tation 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**[2] and **PRM Public**[3]. A detailed explanation of the benchmarking experiments, and the processed datasets are also released together with the `LibRerank` library[4]. We use LambdaMART [Burges, 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 [Huzhang *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.

[2]https://tianchi.aliyun.com/dataset/dataDetail?dataId=56

[3]https://github.com/rank2rec/rerank

[4]https://github.com/LibRerank-Community/LibRerank

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 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 tradeoff 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]. Utilizing the information learned by other stages (*e.g.*, parameter transfer, gradient transfer) would be of high value for both academia and industry.

References

[Abdool *et al.*, 2020] Mustafa Abdool, Malay Haldar, Prashant Ramanathan, et al. Managing diversity in airbnb search. In *KDD*, 2020.

[Ai *et al.*, 2018] Qingyao Ai, Keping Bi, Jiafeng Guo, and W. Croft. Learning a deep listwise context model for ranking refinement. In *SIGIR*, 2018.

[Ai *et al.*, 2019] Qingyao Ai, Xuanhui Wang, Sebastian Bruch, Nadav Golbandi, Michael Bendersky, and Marc Najork. Learning groupwise multivariate scoring functions using deep neural networks. In *ICTIR*, 2019.

[Bartlett and Wegkamp, 2008] Peter L Bartlett and Marten H Wegkamp. Classification with a reject option using a hinge loss. *JMLR*, 2008.

[Bello *et al.*, 2018] Irwan Bello, Sayali Kulkarni, Sagar Jain, et al. Seq2slate: Re-ranking and slate optimization with rnns, 2018.

[Borisov *et al.*, 2016] Alexey Borisov, Ilya Markov, Maarten De Rijke, and Pavel Serdyukov. A neural click model for web search. In *WWW*, 2016.

[Burges, 2010] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 2010.

[Carbonell and Goldstein, 1998] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.

[Chen *et al.*, 2020] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. A context-aware click model for web search. In *WSDM*, 2020.

[Cybenko, 1989] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 1989.

[Feng *et al.*, 2018] Yue Feng, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. From greedy selection to exploratory decision-making: Diverse ranking with policy-value networks. In *SIGIR*, 2018.

[Feng *et al.*, 2021a] Yufei Feng, Yu Gong, Fei Sun, Junfeng Ge, and Wenwu Ou. Revisit recommender system in the permutation prospective. *arXiv preprint arXiv:2102.12057*, 2021.

[Feng *et al.*, 2021b] Yufei Feng, Yu Gong, Fei Sun, Qingwen Liu, and Wenwu Ou. Revisit recommender system in the permutation prospective, 2021.

[Feng *et al.*, 2021c] Yufei Feng, Binbin Hu, Yu Gong, Fei Sun, Qingwen Liu, and Wenwu Ou. Grn: Generative rerank network for context-wise recommendation. *arXiv preprint arXiv:2104.00860*, 2021.

[Ge *et al.*, 2020] Yingqiang Ge, Shuya Zhao, Honglu Zhou, Changhua Pei, Fei Sun, Wenwu Ou, and Yongfeng Zhang. Understanding echo chambers in e-commerce recommender systems. In *SIGIR*, 2020.

[Geyik *et al.*, 2019] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *KDD*, 2019.

[Goldberg, 2017] Yoav Goldberg. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, 2017.

[Gong *et al.*, 2020] Yu Gong, Ziwen Jiang, Yufei Feng, et al. Edgerec: recommender system on edge in mobile taobao. In *CIKM*, 2020.

[Goodfellow *et al.*, 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[Hao *et al.*, 2021] Qi Hao, Tianze Luo, and Guangda Huzhang. Re-ranking with constraints on diversified exposures for homepage recommender system. *arXiv preprint arXiv:2112.07621*, 2021.

[Hard *et al.*, 2018] Andrew Hard, Kanishka Rao, Rajiv Mathews, et al. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[Hron *et al.*, 2021] Jiri Hron, Karl Krauth, Michael Jordan, and Niki Kilbertus. On component interactions in two-stage recommender systems. *NeurIPS*, 2021.

[Huang *et al.*, 2020] Jinhong Huang, Yang Li, Shan Sun, Bufeng Zhang, and Jin Huang. Personalized flight itinerary ranking at fliggy. In *CIKM*, 2020.

[Huzhang *et al.*, 2021] Guangda Huzhang, Zhenjia Pang, Yongqing Gao, et al. Aliexpress learning-to-rank: Maximizing online model performance without going online. *TKDE*, 2021.

[Jiang *et al.*, 2017] Zhengbao Jiang, Ji-Rong Wen, Zhicheng Dou, Wayne Xin Zhao, Jian-Yun Nie, and Ming Yue. Learning to diversify search results via subtopic attention. In *SIGIR*, 2017.

[Jiang *et al.*, 2018] Ray Jiang, Sven Gowal, Yuqiu Qian, Timothy Mann, and Danilo J Rezende. Beyond greedy ranking: Slate optimization via list-cvae. In *ICLR*, 2018.

[Karako and Manggala, 2018] Chen Karako and Putra Manggala. Using image fairness representations in diversity-based re-ranking for recommendations. In *UMAP*, 2018.

[Li *et al.*, 2019] Xinyi Li, Yifan Chen, Benjamin Pettit, and Maarten De Rijke. Personalised reranking of paper recommendations using paper content and user behavior. *TOIS*, 2019.

[Liao *et al.*, 2021] Guogang Liao, Ze Wang, Xiaoxu Wu, et al. Cross dqn: Cross deep q network for ads allocation in feed. *arXiv preprint arXiv:2109.04353*, 2021.

[Lin *et al.*, 2022] Zhuoyi Lin, Sheng Zang, Rundong Wang, et al. Attention over self-attention: Intention-aware re-ranking with dynamic transformer encoders for recommendation. *arXiv preprint arXiv:2201.05333*, 2022.

[Liu *et al.*, 2019] Weiwen Liu, Jun Guo, Nasim Sonboli, Robin Burke, and Shengyu Zhang. Personalized fairness-aware re-ranking for microlending. In *RecSys*, 2019.

[Liu *et al.*, 2020a] Jiongnan Liu, Zhicheng Dou, Xiaojie Wang, Shuqi Lu, and Ji-Rong Wen. Dvgan: A minimax game for search result diversification combining explicit and implicit features. In *SIGIR*, 2020.

[Liu *et al.*, 2020b] Weiwen Liu, Qing Liu, Ruiming Tang, Junyang Chen, Xiuqiang He, and Pheng Heng. Personalized re-ranking with item relationships for e-commerce. In *CIKM*, 2020.

[Liu *et al.*, 2021] Shuchang Liu, Fei Sun, Yingqiang Ge, Changhua Pei, and Yongfeng Zhang. Variation control and evaluation for generative slate recommendations. In *WWW*, 2021.

[Oosterhuis, 2021] Harrie Oosterhuis. Computationally efficient optimization of plackett-luce ranking models for relevance and fairness. *arXiv preprint arXiv:2105.00855*, 2021.

[Pang *et al.*, 2020] Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Ji-Rong Wen. Setrank: Learning a permutation-invariant ranking model for information retrieval. In *SIGIR*, 2020.

[Pei *et al.*, 2019] Changhua Pei, Wenwu Ou, Dan Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, and Junfeng Ge. Personalized re-ranking for recommendation. In *RecSys*, 2019.

[Plackett, 1975] Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 1975.

[Qin *et al.*, 2020] Xubo Qin, Zhicheng Dou, and Ji-Rong Wen. Diversifying search results using self-attention network. In *CIKM*, 2020.

[Rendle *et al.*, 2012] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[Silver *et al.*, 2014] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *ICML*, 2014.

[Singh and Joachims, 2019] Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. *NeuIPS*, 2019.

[Song *et al.*, 2020] Junshuai Song, Zhao Li, Chang Zhou, et al. Co-displayed items aware list recommendation. *IEEE Access*, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *NIPS*, 2017.

[Wang *et al.*, 2018] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. The lambdaloss framework for ranking metric optimization. In *CIKM*, 2018.

[Wang *et al.*, 2019] Fan Wang, Xiaomin Fang, Lihang Liu, et al. Sequential evaluation and generation framework for combinatorial recommender system. *arXiv preprint arXiv:1902.00245*, 2019.

[Wei *et al.*, 2020] Jianxiong Wei, Anxiang Zeng, Yueqiu Wu, Peng Guo, Qingsong Hua, and Qingpeng Cai. Generator and critic: A deep reinforcement learning approach for slate re-ranking in e-commerce. *arXiv preprint arXiv:2005.12206*, 2020.

[Wilhelm *et al.*, 2018] Mark Wilhelm, Ajith Ramanathan, Alexander Bonomo, et al. Practical diversified recommendations on youtube with determinantal point processes. In *CIKM*, 2018.

[Xi *et al.*, 2021] Yunjia Xi, Weiwen Liu, Xinyi Dai, Ruiming Tang, Weinan Zhang, Qing Liu, Xiuqiang He, and Yong Yu. Context-aware reranking with utility maximization for recommendation. *arXiv preprint arXiv:2110.09059*, 2021.

[Xia *et al.*, 2008] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *ICML*, 2008.

[Xia *et al.*, 2016] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. Modeling document novelty with neural tensor network for search result diversification. In *SIGIR*, 2016.

[Xia *et al.*, 2017] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. Adapting markov decision process for search result diversification. In *SIGIR*, 2017.

[Xie *et al.*, 2021] Ruobing Xie, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. Hierarchical reinforcement learning for integrated recommendation. In *AAAI*, 2021.

[Yadav *et al.*, 2021] Himank Yadav, Zhengxiao Du, and Thorsten Joachims. Policy-gradient training of fair and unbiased ranking functions. In *SIGIR*, 2021.

[Yan *et al.*, 2021] Le Yan, Zhen Qin, Rama Kumar Pasumarthi, Xuanhui Wang, and Michael Bendersky. Diversification-aware learning to rank using distributed representation. In *WWW*, 2021.

[Zhang *et al.*, 2021] Ruizhe Zhang, Xiaohui Xie, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. Constructing a comparison-based click model for web search. In *WWW*, 2021.

[Zhao *et al.*, 2020] Xiangyu Zhao, Xudong Zheng, Xiwang Yang, Xiaobing Liu, and Jiliang Tang. Jointly learning to recommend and advertise. In *KDD*, 2020.

[Zhao *et al.*, 2021] Xiangyu Zhao, Changsheng Gu, Haoshenglun Zhang, et al. Dear: Deep reinforcement learning for online advertising impression in recommender systems. In *AAAI*, 2021.

[Zhu *et al.*, 2021] Ziwei Zhu, Jingu Kim, Trung Nguyen, Aish Fenton, and James Caverlee. Fairness among new items in cold start recommender systems. In *SIGIR*, 2021.

[Zhuang *et al.*, 2018] Tao Zhuang, Wenwu Ou, and Zhirong Wang. Globally optimized mutual influence aware ranking in e-commerce search. In *IJCAI*, 2018.