# Reinforcement Learning for Effective Few-Shot Ranking

Shiva Soleimany
University of Toronto
Toronto, Canada

Sajad Ebrahimi
University of Guelph
Guelph, Canada

Shirin Seyedsalehi
Toronto Metropolitan University
Toronto, Canada

Fattane Zarrinkalam
University of Guelph
Guelph, Canada

Ebrahim Bagheri
University of Toronto
Toronto, Canada

## Abstract

Neural rankers have achieved strong retrieval effectiveness but require large amounts of labeled data, limiting their applicability in *few-shot settings*. In this paper, we address the sample inefficiency of neural ranking methods by introducing a Reinforcement Learning (RL)-based re-ranking model that achieves high effectiveness with minimal training data. Built on a Deep Q-learning Network (DQN) framework, our approach is designed for *few-shot* settings, maximizing sample efficiency to ensure robust generalization from limited interactions. Extensive experiments show that our model significantly outperforms data-intensive methods and existing few-shot baselines, demonstrating RL's potential to enhance IR capabilities in *few-shot* scenarios.

## CCS Concepts

• **Information systems** → **Novelty in information retrieval**.

## Keywords

Ranking, Reinforcement Learning, Few-shot Learning, Markov Decision Processes (MDPs), Deep Q-learning Networks (DQN)

**ACM Reference Format:**
Shiva Soleimany, Sajad Ebrahimi, Shirin Seyedsalehi, Fattane Zarrinkalam, and Ebrahim Bagheri. 2025. Reinforcement Learning for Effective Few-Shot Ranking. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3726302.3730243

## 1 Introduction

Neural rankers have greatly enhanced IR effectiveness [38, 42], However, these models typically require vast amounts of labeled training data to perform well, limiting their applicability in *few-shot* settings, where only a small number of labeled examples are available due to time, cost, or data constraints [3, 5, 25, 30]. A natural solution in *few-shot settings* is lexical retrievers like BM25 [26], which rank documents based on term frequency statistics without requiring training. However, these models rely on surface-level term matching and so fail to capture deep semantic relationships [1, 2, 6, 9]. Neural rankers overcome this limitation by leveraging large-scale training data [13, 17, 37], but their reliance on extensive labeled data makes them impractical in few-shot settings.

**Background Approaches.** Given the limitations of both lexical models and neural ranking methods in few-shot settings, there is a growing need for ranking approaches that can effectively operate with minimal labeled data. One promising direction is to explore methods that can learn from limited feedback, rather than relying on large labeled datasets. Sun et al. [34] proposed MetaAdaptRank, which employs synthesizing contrastive weak supervision and using meta-learning to filter noisy signals. Unlike MetaAdaptRank, which generates synthetic data, Sinhababu et al. [32] proposed a method that leverages prompting by retrieving similar queries from a training set and using them as pairwise ranking examples during inference. This augmentation allows LLMs to make more informed ranking decisions, improving both in-domain and out-of-domain retrieval without requiring model fine-tuning. On the other hand, $P^3$ Ranker [10] bridges the gap between pre-trained language models (PLMs) and ranking tasks by using prompt-based learning to align ranking with PLM training and pre-finetuning to inject ranking-specific knowledge. Unlike the two aforementioned methods, the $P^3$ Ranker focuses on structured PLM adaptation, making it suitable for *few-shot ranking with minimal labeled data*. While $P^3$ Ranker demonstrates strong performance in few-shot settings, its effectiveness still depends on pre-finetuning, which may not always be feasible when intermediate tasks are unavailable or when labeled data is highly limited.

On the other hand, Reinforcement Learning (RL) [36] provides a suitable framework by enabling models to learn optimal ranking behaviors through interactions and rewards rather than extensive labeled data. Contrary to common belief, RL can be effective in certain *few-shot scenarios* [12, 28, 29]. By framing ranking as a sequential decision-making task [40], RL allows models to iteratively refine rankings based on feedback signals, making it particularly adaptable in *few-shot scenarios*.

Reinforcement learning (RL) [35] has gained traction in several information retrieval (IR) tasks, particularly in modeling document ranking as a sequential decision-making process through Markov Decision Processes (MDPs). In this framework, at each time step, an agent selects a document based on the current observation (e.g., ranking position and remaining unranked documents), with rewards often defined in terms of ranking metrics like NDCG (Normalized Discounted Cumulative Gain). Various IR tasks, such as session search, have been formulated as MDPs to model user interactions over multiple queries, optimizing document ranking


*SIGIR '25, Padua, Italy*

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3730243

across sessions [7, 44]. Similarly, RL-based ranking has been applied to search result diversification [7, 41] and multi-page search [43], where the RL agent learns to balance relevance and diversity across search results. Specific approaches such as MDPRank [11, 43, 46] and REINFORCE-based document ranking [40] optimize ranking policies using policy gradient methods. For instance, in [41], search result diversification is modeled as an MDP, where each ranking position represents a decision point, and the agent selects documents sequentially. However, policy gradient methods tend to be sample-inefficient, requiring extensive interactions with the environment due to noisy gradient estimates and high variance in training [20]. This inefficiency poses challenges for few-shot settings.

The CUOLR model [45] extends the MDP-based ranking framework by making the ranking task click model-agnostic, enabling generalization across different user feedback models. To achieve this, CUOLR incorporates the Soft Actor-Critic (SAC) algorithm, a reinforcement learning approach originally designed for continuous action spaces. However, SAC's performance and sample efficiency degrade in discrete action spaces due to its design for continuous domains. Additionally, actor-critic algorithms like SAC rely on an on-policy critic, whereas value-based methods like DQN typically achieve better performance in discrete-action environments [33].

**Rationale and Proposed Approach.** To address sample inefficiency, which limits methods like MDPRank in few-shot settings, we propose a ranking strategy based on Deep Q-learning Networks (DQN), a sample-efficient value-based RL approach [18, 19]. In this framework, we approximate the Q-function with a neural network to learn the expected reward of ranking decisions. The key features of our approach that make it well-suited for few-shot settings include: **(1)** *Experience replay*, which stores and reuses past interactions, breaking correlations between consecutive ranking decisions and enhancing learning diversity—critical when training data is limited. **(2)** *Temporal credit assignment* [22], which evaluates long-term rewards, allowing the model to learn cumulative effects over time rather than focusing solely on immediate rewards. This is particularly valuable in ranking, where a document's position may have delayed effects on overall ranking quality.

**Key Contributions.** We address the challenge of sample inefficiency in RL-based ranking for few-shot settings, proposing a re-ranking model specifically designed to perform effectively with limited training data. Our approach leverages DQN to maximize data efficiency and improve generalization in few-shot settings. Our approach enables the model to learn robust ranking policies from a minimal training dataset, achieving competitive ranking effectiveness even in data-constrained scenarios. We provide empirical evidence that our model can achieve ranking performance surpassing lexical ranking methods that do not require training data and far superior performance to neural rankers that by their nature require significantly larger training datasets. We further show that our approach surpasses earlier RL-based rankers, such as MDPRank, in learning from limited training data.

## 2 Proposed Approach

Let us assume that for a few-shot (FS) settings, there exists a query pool $Q_{FS}$ consisting of a limited set of queries $Q_{FS} = \{q_1, q_2, \ldots, q_n\}$. Further let each query $q_i \in Q_{FS}$ be associated with a set of relevant documents $D_{q_i}$ with $k$ documents $D_{q_i} = \{d_1, d_2, \ldots, d_k\}$. The objective of our task is to train a re-ranking model *FSRank* with this small-sized training dataset.

**Description of the RL Model.** We formulate document re-ranking as a Markov Decision Process (MDP). [23], MDPs are stochastic models well-suited for sequential decision-making. In this formulation, each step involves selecting a document for the next position in the list. This enables the integration of contextual information, such as the current time step and remaining documents, into the state representation, leading to more informed ranking decisions. The MDP in our work is defined as a quadruple $\langle S, A, T, R \rangle$, representing states, actions, a transition function, and rewards as follows:

**States.** The states $S$ represent the environment. For ranking, the agent must be aware of the current ranking position and the set of candidate documents $C$. At time step $t$, the state $s_t$ is defined as the pair $[t, C_t]$, where $C_t$ denotes the unsorted set of candidate documents that remain to be ranked.

**Actions.** The actions $A$ refer to the set of discrete actions available to the agent. The feasible actions are determined by the current state $s_t$ and are represented as $A(s_t)$. At each time step $t$, the agent takes action $a_t \in A(s_t)$, which involves selecting a document $c_i \in C_t$ for the next ranking position $t + 1$.

**Transition function.** The transition function $T(s, a)$ returns the next state $s_{t+1} \in S$ resulting from taking action $a_t$ in state $s_t$. In a deterministic environment, the outcome of this function is unique, meaning that for each state-action pair, there is a specific next state. In a given state, $s_t$, after taking action $a_t$, the next state is constructed by updating the candidate set and also incrementing the time step. The candidate set $C_t$ is updated by removing the chosen document $c_i$ from the candidate set, and the time step is incremented by one, forming the next state $s_{t+1}$ according to Equation 1:

$$s_{t+1} = T(s_t, a_t) = [t + 1, C_{t+1}] \quad \text{where} \quad C_{t+1} = C_t \setminus \{c_i\} \tag{1}$$

**Reward.** The reward $\mathcal{R}(S, A)$ provides immediate feedback, also known as reinforcement. It represents the reward the agent receives for executing action $a_t \in A(s_t)$. In the context of ranking, the action $a_t$ corresponds to the selection of a document $c_i$ and $\mathcal{R}(s_t, a_t)$ is correlated to the quality of $c_i$. The function $R(s, a)$ is designed to prioritize positioning the most relevant documents at the top. Thus, it can depend on the relevancy of the document $c_i$ selected by action $a_t$, denoted as $\Psi(c_i)$, and its position. To promote the early selection of highly relevant documents, we apply a time-based penalty. The reward function is formulated according to Equation 2:

$$\mathcal{R}(s_t, a_t) = \frac{\Psi(c_i)}{\log_2(t + 1)} \quad where \quad a_t : select \quad c_i \in C_t \tag{2}$$

As shown in Equation 2, the logarithmic denominator of the current time step $t$ ensures that selecting relevant documents earlier yields a higher reward, encouraging the agent to place the most relevant documents at the top of the ranked list.

In this context, the model consists of two components: **(1)** a language model which serves as the feature extractor and whose weights are not updated during the training. This language model takes a concatenated query and document pair as input and generates a vector representation. The current time-step $t$ is then appended to the beginning of this vector representation to build

a feature vector, $x$, which acts as the feature for the RL agent: $x = t + LM(q \oplus d)$.

**(2)** The agent consists of two components: a) an experience replay buffer, $B$, which stores and randomizes past experiences, and b) a neural network $\mathcal{N}$.

**Action Value Function.** The action value function, i.e., $Q(s, a)$, estimates the expected future rewards of taking action $a_t$ in state $s_t$. It combines immediate rewards and discounted future rewards to provide a measure of the value of actions. In an MDP, the future reward is worth less than the current reward, and therefore, a discount factor $\gamma \in (0, 1)$ is applied to future rewards. This discount factor, along with the time step-related penalty in the reward function, encourages the RL agent to try to select the most relevant documents sooner in order to maximize its total reward. When trying to estimate $Q$, traditional RL methods like Q-learning struggle with high-dimensional state spaces due to their inflexibility in scaling state-action pairs. To overcome this scalability limit, we use DQN [18], a popular reinforcement learning algorithm that employs a neural network as a non-linear function approximator to estimate the action-value function. The neural network in our RL agent, $\mathcal{N}$ is parameterized by $\phi$. The input to this network, $x_{a_t}$, is the feature vector of action $a_t$. At a given time step $t$, we calculate the value of action $a_t$ according to Equation 3:

$$Q(s, a_t; \phi_t) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_0 = s_t, a_0 = a_t\right] \tag{3}$$

where $\gamma \in [0, 1)$ is the discount factor, which determines the importance of future rewards, and $r_t$ is the reward received at time step $t$. For each action, the expected value is defined as the sum of the immediate reward and the expected future reward.

**The Learning Process.** Our proposed learning process consists of two phases, explained below and formally described in Algorithm 1.

**The Experience Collection Phase**: The first phase accumulates experiences in the experience replay buffer $B$. For each query in $Q_{FS}$, we uniformly sample a document from $D_{q_i}$ by taking action $a_t$ and add the observation tuple $(s_t, a_t, r_t, s_{t+1})$, to $B$. This can be seen in lines 3-7 of Algorithm 1. Experience replay enhances data efficiency by allowing each experience tuple to contribute to multiple weight updates [19, 39]. Additionally, experience replay helps prevent catastrophic forgetting, where new experiences overwrite prior knowledge, a critical issue in data-scarce domains where forgetting learned interactions can degrade performance [27, 31].

**The Training Phase**: Once the replay buffer is filled, our RL process randomly samples from the replay buffer and updates the network based on these samples as shown on Line 14 of Algorithm 1.

Randomly sampling experiences breaks the correlation between consecutive experiences, leading to reduced variance, improved stability, and better learning performance [15, 19, 39].

For each experience tuple $(s_t, a_t, r_t, s_{t+1})$, the feature vector of action $a_t$, $x_{a_t}$, is constructed using the language model $LM$. Then $x_{a_t}$ is processed through the network $\mathcal{N}$ to output the current Q-value, $\hat{Q}(a_t)$, which needs to approximate the target value, $\Omega_{a_t}$. The current Q-value is defined in Equation 4 as follows:

$$\hat{Q}(s_t, a_t; \phi) = \mathcal{N}(x_{a_t}; \phi_t) \tag{4}$$

On this basis, $Q'$ is calculated for all the possible actions in $s_{t+1}$. The maximum value of $Q'$ is denoted as $U_i$ and represents the maximum reward that can be expected by taking action $a_t$ and transitioning to state $s_{t+1}$. The target Q-value, $\Omega_{a_t}$, is calculated as the sum of immediate reward, $r_t$ and the discounted $U_i$. This is shown in Lines 15-19 of Algorithm 1 and Equation 5, as follows:

$$\Omega_{a_t} = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \phi_t) \tag{5}$$

As the RL model is trained, it is expected $\hat{Q}$ to move towards $\Omega$ to indicate how much reward can be expected if an action $a_t$ is taken in time step $t$. In order to find the optimal values of $\phi*$ for the networks, we adopt the mean squared error (MSE) between $\hat{Q}$ and $\Omega$, shown in Equation 6, as the training loss function. This corresponds to Line 20 in Algorithm 1.

$$L(\phi) = \mathbb{E}\left[\left(\Omega_{a_t} - \hat{Q}(s_t, a_t; \phi_t)\right)^2\right] \tag{6}$$

Finally, the weights of the network are updated using gradient descent and learning rate $\eta$, as shown on Line 21 of Algorithm 1.

**Algorithm 1** Sample Collection and RL Model Training

```
1:  Q ← Initialize(φ_0)
2:  B ← ∅,   max_size = N        ▷ Phase 1: Filling the Replay Buffer
3:  for q_i ∈ Q_FS do
4:      for timestep t: 0 ≤ t < len(D_{q_i}) do
5:          a_t ~ Uniform(D_{q_i})
6:          Execute action a_t in s_t, then observe (r_t, s_{t+1})
7:          B ← B ∪ {(s_t, a_t, r_t, s_{t+1})}
8:          if |B| = max_size then
9:              break
10:         end if
11:     end for
12: end for
        ▷ Phase 2: Sampling from Replay Buffer and RL Model Training
13: for {(s_i, a_i, r_i, s_{i+1})} ~ Uniform(B) do
14:     Q(s_i, a_i; φ_i) = N(x_{a_i}; φ_i)
15:     for a' ∈ C_{i+1} do
16:         Q' = N(x_{a'}; φ_i)
17:     end for
18:     U_i = max_{Q'}
19:     Ω_{a_i} = r_i + γU_i
20:     L(φ) = (Ω_{a_i} − Q(s_i, a_i))^2
21:     φ_{i+1} ← φ_i − η∇L(φ_i)
22: end for
```

## 3 Experiments

**Research Questions (RQs).** We explore three research questions as follows: (**RQ1**) we assess whether our proposed model is *generalizable* on different language models and whether it shows *stable performance* when the number of training samples change; (**RQ2**) we investigate whether the performance of our model is competitive with existing state of the art neural rankers, a state-of-the art few shot ranker, and the unsupervised lexical BM25 approach; and, (**RQ3**) we explore whether our RL-based approach is able to show better performance compared to strong RL-based rankers.

**Dataset.** We conduct experiments on the MS MARCO v1 dataset [21], which contains 8.8 million passages. For training, we randomly

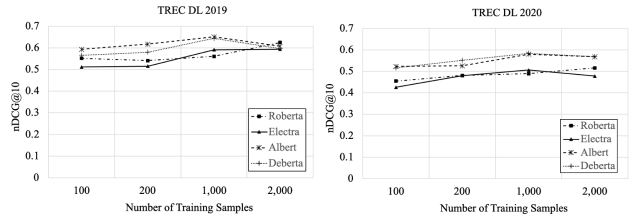


**Figure 1: Generalizability on different LLMs & train set sizes.**

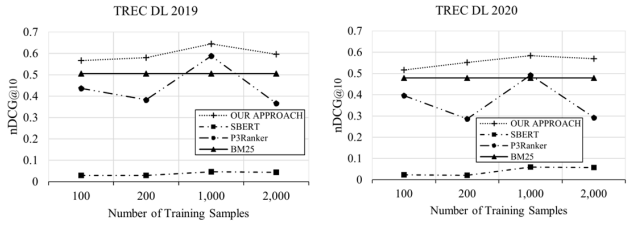


**Figure 2: Comparison *w.* neural, few-shot & lexical baselines.**

sample 2,000 queries (*a small set to replicate a few shot scenario*) from the 501k queries with relevance judgments. For evaluation, we use the TREC Deep Learning Track (DL-2019, DL-2020), which features more challenging queries and richer relevance labels.

**Implementation Details.**[1]. We trained our model on a 9-layer FFN, with the learning rate of 0.001, a discount factor of 0.99, a batch size of 1, a replay buffer size of 10, 000, and 100,000 episodes.

**Findings.** In **(RQ1)**, we investigate the generalizability and stability of our proposed approach. For the sake of *generalizability*, we report the performance of our proposed approach when applied on different language models, namely RoBERTa [16], ELECTRA [4], DeBERTa [8], and ALBERT [14]. These models are used in their original pre-trained format without any further fine-tuning for the ranking task. As shown in Figure 1, our proposed approach shows similar performance on both TREC DL 2019 and TREC 2020 regardless of the language model that is used for its training. Furthermore, in order to assess the *stability*, we train our proposed model on all four language models using four different train set sizes, including 100, 200, 1000, and 2000 training samples. The results can again be seen in Figure 1. As seen in the figure, model performances are enhanced as the size of the training set increases from 100 samples to 2,000 samples by approximately 10%. The increase in performance is smooth for all models on both datasets. We also note that regardless of the test set and the language model, all models perform quite strongly, even when trained on 100 samples and exhibit stable performance as train set size increases.

In the second research question (**RQ2**), we compare our approach against a state-of-the-art SBERT neural ranking baseline using a cross-encoder architecture [24], as well as the state-of-the-art few-shot ranker [10], and the lexical-based BM25 baseline, which requires no training. Based on findings from RQ1, we report results only for the DeBERTa model due to space constraints. Figure 2 compares our model with SBERT, $P^3Rank$, and BM25. BM25 remains

[1]Our code and data is available on GitHub: https://github.com/ShivaSoleimany/rl_few_shot_ranker

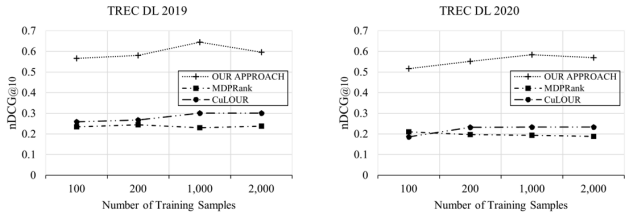


**Figure 3: Benchmarking with state-of-the-art RL baselines.**

unaffected by training size, achieving nDCG@10 scores of 0.505 and 0.479 on TREC DL 2019 and DL 2020, respectively. The key finding in RQ2 is that SBERT fails to learn effectively from limited samples, maintaining nDCG@10 below 0.1 across all training sizes, even with 2,000 training samples. Our model consistently outperforms $P^3Rank$ ( *the state-of-the-art few-shot ranker*) and scales effectively with increasing training samples, unlike SBERT and $P^3Rank$. It also achieves consistently higher performance than the BM25 baseline.

In **RQ3**, we compare our approach against two state-of-the-art RL-based ranking models: MDPRank [40] and CUOLR [45]. This research question examines (1) whether the efficiency of our RL-based method in learning from limited samples extends to other RL baselines, and (2) whether our approach is more sample-efficient due to its architectural design. Figure 3 compares our approach with MDPRank and CUOLR on both test sets, leading to three key observations. *(i)* Both MDPRank and CUOLR outperform neural rankers like SBERT in low-resource settings, consistently achieving nDCG@10 above 0.2, whereas SBERT remains below 0.05 under similar conditions. This highlights the effectiveness of RL-based methods for few-shot learning. *(ii)* While more effective than neural rankers, MDPRank employs a policy gradient algorithm, which is sample inefficient due to noisy gradient estimates and high variance during training [20]. As a result, it performs worse than our approach, which is more sample-efficient. *(iii)* MDPRank plateaus in performance as training data increases, whereas our model continues improving with more training samples. *(iv)* Although CUOLR outperforms neural rankers, it relies on a soft actor-critic algorithm originally designed for continuous action spaces, making it inefficient for discrete action spaces [33]. Additionally, actor-critic methods depend on an on-policy critic, limiting their effectiveness compared to DQN-based models in discrete settings [33]. Consequently, CUOLR exhibits lower performance than our approach, which is significantly more efficient in practice.

## 4 Concluding Remarks

We propose a reinforcement learning (RL)-based re-ranking model to address data inefficiency in neural rankers for *few-shot scenarios*. Built on a Deep Q-learning Network (DQN), our approach enhances sample efficiency through experience replay and optimized action selection via Q-value estimation. Extensive experiments show our model significantly outperforms both data-intensive, RL-based and strong few-shot ranking baselines, achieving high effectiveness in NDCG while learning meaningful ranking policies from limited data.

## References


[1] Negar Arabzadeh, Radin Hamidi Rad, Maryam Khodabakhsh, and Ebrahim Bagheri. 2023. Noisy Perturbations for Estimating Query Difficulty in Dense Retrievers. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 3722–3727. doi:10.1145/3583780.3615270

[2] Ebrahim Bagheri, Faezeh Ensan, and Feras N. Al-Obeidat. 2018. Neural word and entity embeddings for ad hoc retrieval. *Inf. Process. Manag.* 54, 4 (2018), 657–673. doi:10.1016/J.IPM.2018.04.007

[3] Amin Bigdeli, Negar Arabzadeh, and Ebrahim Bagheri. 2024. Learning to Jointly Transform and Rank Difficult Queries. In *Advances in Information Retrieval*. Springer Nature Switzerland, Cham, 40–48. doi:10.1007/978-3-031-56066-8_5

[4] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=r1xMH1BtvB

[5] Sajad Ebrahimi, Sara Salamat, Negar Arabzadeh, Mahdi Bashari, and Ebrahim Bagheri. 2025. exHarmony: Authorship and Citations for Benchmarking the Reviewer Assignment Problem. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part III*. Springer-Verlag, 1–16. doi:10.1007/978-3-031-88714-7_1

[6] Faezeh Ensan and Ebrahim Bagheri. 2017. Document Retrieval Model Through Semantic Linking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, Maarten de Rijke, Milad Shokouhi, Andrew Tomkins, and Min Zhang (Eds.). ACM, 181–190. doi:10.1145/3018661.3018692

[7] Yue Feng, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. 2018. From Greedy Selection to Exploratory Decision-Making: Diverse Ranking with Policy-Value Networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 125–134. doi:10.1145/3209978.3209979

[8] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=sE7-XhLxHA

[9] Seyed Mohammad Hosseini, Negar Arabzadeh, Morteza Zihayat, and Ebrahim Bagheri. 2024. Enhanced Retrieval Effectiveness through Selective Query Generation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24)*. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3627673.3679912

[10] Xiaomeng Hu, Shi Yu, Chenyan Xiong, Zhenghao Liu, Zhiyuan Liu, and Ge Yu. 2022. P3 Ranker: Mitigating the Gaps between Pre-training and Ranking Fine-tuning with Prompt-based Learning and Pre-finetuning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. ACM, 1956–1962. doi:10.1145/3477495.3531786

[11] Yujing Hu, Qing Da, Anxiang Zeng, Yang Yu, and Yinghui Xu. 2018. Reinforcement Learning to Rank in E-Commerce Search Engine: Formalization, Analysis, and Application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 368–377. doi:10.1145/3219819.3219846

[12] Scott Jeen, Tom Bewley, and Jonathan Cullen. 2024. Zero-Shot Reinforcement Learning from Low Quality Data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=79eWvkLjib

[13] Maryam Khodabakhsh and Ebrahim Bagheri. 2023. Learning to rank and predict: Multi-task learning for ad hoc retrieval and query performance prediction. *Information Sciences* 639 (2023), 119015. doi:10.1016/j.ins.2023.119015

[14] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=H1eA7AEtvS

[15] Long Ji Lin. 1992. Self-Improving Reactive Agents Based On Reinforcement Learning, Planning and Teaching. *Mach. Learn.* 8 (1992), 293–321. doi:10.1007/BF00992699

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[17] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 1291–1299. doi:10.1145/3038912.3052579

[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *CoRR* abs/1312.5602 (2013). arXiv:1312.5602 http://arxiv.org/abs/1312.5602

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nat.* 518, 7540 (2015), 529–533. doi:10.1038/NATURE14236

[20] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. 2017. Bridging the Gap Between Value and Policy Based Reinforcement Learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 2775–2785. https://proceedings.neurips.cc/paper/2017/hash/facf9f743b083008a894eee7baa16469-Abstract.html

[21] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016 (CEUR Workshop Proceedings, Vol. 1773)*, Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne (Eds.). CEUR-WS.org. https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf

[22] Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, and Laura Toni. 2024. A Survey of Temporal Credit Assignment in Deep Reinforcement Learning. *Trans. Mach. Learn. Res.* 2024 (2024). https://openreview.net/forum?id=bNtr6SLgZf

[23] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley. doi:10.1002/9780470316887

[24] Nils Reimers and Iryna Gurevych. 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 4512–4525. doi:10.18653/V1/2020.EMNLP-MAIN.365

[25] Navid Rekabsaz and Markus Schedl. 2020. Do Neural Ranking Models Intensify Gender Bias?. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 2065–2068. doi:10.1145/3397271.3401280

[26] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (2009), 333–389. doi:10.1561/1500000019

[27] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Gregory Wayne. 2019. Experience Replay for Continual Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 348–358. https://proceedings.neurips.cc/paper/2019/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html

[28] Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron C. Courville, and Philip Bachman. 2020. Data-Efficient Reinforcement Learning with Momentum Predictive Representations. *CoRR* abs/2007.05929 (2020). arXiv:2007.05929 https://arxiv.org/abs/2007.05929

[29] Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, R. Devon Hjelm, Philip Bachman, and Aaron C. Courville. 2021. Pretraining Representations for Data-Efficient Reinforcement Learning. *CoRR* abs/2106.04799 (2021). arXiv:2106.04799 https://arxiv.org/abs/2106.04799

[30] Shirin Seyedsalehi, Sara Salamat, Negar Arabzadeh, Sajad Ebrahimi, Morteza Zihayat, and Ebrahim Bagheri. 2025. Gender disentangled representation learning in neural rankers. *Machine Learning* 114, 5 (2025), 121. doi:10.1007/s10994-024-06664-2

[31] Guangyuan Shi, Jiaxin Chen, Wenlong Zhang, Li-Ming Zhan, and Xiao-Ming Wu. 2021. Overcoming Catastrophic Forgetting in Incremental Few-Shot Learning by Finding Flat Minima. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 6747–6761. https://proceedings.neurips.cc/paper/2021/hash/357cfba15668cc2e1e73111e09d54383-Abstract.html

[32] Nilanjan Sinhababu, Andrew Parry, Debasis Ganguly, Debasis Samanta, and Pabitra Mitra. 2024. Few-shot Prompting for Pairwise Ranking: An Effective Non-Parametric Retrieval Model. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, 12363–12377. https://aclanthology.org/2024.findings-

emnlp.720

[33] Denis Steckelmacher, Hélène Plisnier, Diederik M. Roijers, and Ann Nowé. 2019. Sample-Efficient Model-Free Reinforcement Learning with Off-Policy Critics. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part III* (Würzburg, Germany). Springer-Verlag, Berlin, Heidelberg, 19–34. doi:10.1007/978-3-030-46133-1_2

[34] Si Sun, Yingzhuo Qian, Zhenghao Liu, Chenyan Xiong, Kaitao Zhang, Jie Bao, Zhiyuan Liu, and Paul Bennett. 2021. Few-Shot Text Ranking with Meta Adapted Synthetic Weak Supervision. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 5030–5043. doi:10.18653/V1/2021.ACL-LONG.390

[35] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). The MIT Press. http://incompleteideas.net/book/the-book-2nd.html

[36] Richard S Sutton, Andrew G Barto, et al. 1999. Reinforcement learning. *Journal of Cognitive Neuroscience* 11, 1 (1999), 126–134.

[37] Michael J. Taylor, Hugo Zaragoza, Nick Craswell, Stephen Robertson, and Chris Burges. 2006. Optimisation methods for ranking functions with multiple parameters. In *Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006*, Philip S. Yu, Vassilis J. Tsotras, Edward A. Fox, and Bing Liu (Eds.). ACM, 585–593. doi:10.1145/1183614.1183698

[38] Duc-Thuan Vo, Fattane Zarrinkalam, Ba Pham, Negar Arabzadeh, Sara Salamat, and Ebrahim Bagheri. 2023. Neural Ad-Hoc Retrieval Meets Open Information Extraction. In *Advances in Information Retrieval*, Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo (Eds.). Springer Nature Switzerland, Cham, 655–663. doi:10.1007/978-3-031-28238-6_57

[39] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Rémi Munos, Koray Kavukcuoglu, and Nando de Freitas. 2017. Sample Efficient Actor-Critic with Experience Replay. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=HyM25Mqel

[40] Zheng Wei, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2017. Reinforcement Learning to Rank with Markov Decision Process. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 945–948. doi:10.1145/3077136.3080685

[41] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. 2017. Adapting Markov Decision Process for Search Result Diversification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 535–544. doi:10.1145/3077136.3080775

[42] Andrew Yates, Rodrigo Frassetto Nogueira, and Jimmy Lin. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 2666–2668. doi:10.1145/3404835.3462812

[43] Wei Zeng, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2018. Multi Page Search with Reinforcement Learning to Rank. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2018, Tianjin, China, September 14-17, 2018*, Dawei Song, Tie-Yan Liu, Le Sun, Peter Bruza, Massimo Melucci, Fabrizio Sebastiani, and Grace Hui Yang (Eds.). ACM, 175–178. doi:10.1145/3234944.3234977

[44] Sicong Zhang, Jiyun Luo, and Hui Yang. 2014. A POMDP model for content-free document re-ranking. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (Eds.). ACM, 1139–1142. doi:10.1145/2600428.2609529

[45] Zeyu Zhang, Yi Su, Hui Yuan, Yiran Wu, Rishab Balasubramanian, Qingyun Wu, Huazheng Wang, and Mengdi Wang. 2023. Unified Off-Policy Learning to Rank: a Reinforcement Learning Perspective. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/3f1b6e97a5eb3b10e6b0c99b022988eb-Abstract-Conference.html

[46] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. 2019. Reinforcement Learning to Optimize Long-term User Engagement in Recommender Systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 2810–2818. doi:10.1145/3292500.3330668