

# TaoSR1: The Thinking Model for E-commerce Relevance Search

Chenhe Dong\*  
dongchenhe.dch@alibaba-inc.com  
Taobao & Tmall Group of Alibaba  
Hangzhou, China

Jianhui Yang  
yangjh23@mails.tsinghua.edu.cn  
Tsinghua University  
Beijing, China

Dan Ou  
oudan.od@taobao.com  
Taobao & Tmall Group of Alibaba  
Hangzhou, China

Shaowei Yao\*†  
yaoshaowei@taobao.com  
Taobao & Tmall Group of Alibaba  
Hangzhou, China

Yiming Jin  
enxian@alibaba-inc.com  
Taobao & Tmall Group of Alibaba  
Hangzhou, China

Haihong Tang  
piaoxue@taobao.com  
Taobao & Tmall Group of Alibaba  
Hangzhou, China

Pengkun Jiao  
pkjiao23@m.fudan.edu.cn  
Fudan University  
Shanghai, China

Zerui Huang  
Xiaojiang Zhou  
huangzerui.hzr@taobao.com  
zxjbupt@163.com  
Taobao & Tmall Group of Alibaba  
Hangzhou, China

Bo Zheng  
bozheng@alibaba-inc.com  
Taobao & Tmall Group of Alibaba  
Beijing, China

## Abstract

Query-product relevance prediction serves as a foundational technology in e-commerce search engines, enabling users to discover desired products and ensuring optimal user experience. Previous approaches have primarily relied on BERT-based models which excel at textual and basic semantic matching but demonstrate poor performance in understanding and reasoning capabilities for more complex queries. Consequently, numerous recent studies have explored the application of Large Language Models (LLMs) in search systems. However, most still adopt discriminative paradigms or ultimately distill knowledge to BERT models for deployment. In this paper, we propose an optimization framework based on large language models and directly deploy these models in online systems. Nevertheless, practical deployment presents several challenges, including online deployment, error accumulation in Chain-of-Thought (CoT) leading to performance degradation, and discriminative hallucination. To address these challenges, we propose an LLM-based optimization framework called **Taobao Search Relevance Model v1 (TaoSR1)** comprising three stages: (1) Supervised Fine-Tuning (SFT) with CoT to endow models with reasoning capabilities; (2) Offline multiple sampling based on a pass@N strategy, combined with Direct Preference Optimization (DPO), to enhance model generation quality; and (3) Difficulty-based dynamic sampling integrated with Group Relative Policy Optimization (GRPO) to further mitigate model's discriminative hallucination problems. Finally, by incorporating

post-CoT processing and a relevance tier partitioning method based on cumulative probability, our model achieves more feasible and efficient online deployment. Experimental results demonstrate that our proposed model significantly outperforms baseline methods on challenging offline evaluation datasets, while achieving substantial improvements in online side-by-side human evaluations. Our proposed framework introduces a novel optimization paradigm for incorporating CoT reasoning into relevance classification tasks; we contend that this methodology provide valuable insights into the application of LLMs for other classification tasks.

## CCS Concepts

• **Information systems** → **Language models**; • **Computing methodologies** → **Natural language processing**.

## Keywords

E-commerce Relevance Search, Large Language Models, Natural Language Process, Web Search

## ACM Reference Format:

Chenhe Dong, Shaowei Yao, Pengkun Jiao, Jianhui Yang, Yiming Jin, Zerui Huang, Xiaojiang Zhou, Dan Ou, Haihong Tang, and Bo Zheng. 2018. TaoSR1: The Thinking Model for E-commerce Relevance Search. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## 1 Introduction

Large-scale e-commerce platforms such as Taobao and Amazon serve hundreds of millions of users daily, tasked with retrieving desired products from a massive item corpus. Query-item relevance prediction ensures the relevance between returned products and search queries. It is a critical determinant of user experience and a significant factor in long-term user value. Displaying irrelevant products not only negatively impacts user experience but also reduces merchant operational efficiency and satisfaction. Therefore,

\*Equal contribution.

†Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

Conference acronym 'XX, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/2018/06

<https://doi.org/XXXXXXX.XXXXXXX>

query-item relevance prediction technology constitutes the cornerstone of e-commerce search engines.

Previous approaches primarily relied on BERT-based models, leveraging an encoder-only architecture and bidirectional attention mechanisms to achieve strong text matching capabilities, typically satisfying 80-90% of user search needs. However, the remaining over 10% of long-tail complex queries present a significant challenge, often resulting in inconsistent user experiences. These long-tail queries demand stronger semantic understanding and reasoning capabilities from models. The low-rank problem inherent in BERT's bidirectional attention limits parameter scaling, thereby constraining the upper bound of BERT model capabilities. Recent research has increasingly explored the application of Large Language Model (LLM) technologies in search systems, including works such as RankLLaMA[2], LREF[1] and so on[3, 4]. However, most approaches still employ discriminative optimization paradigms or ultimately distill knowledge to BERT models for deployment. As a result, they fall short of effectively addressing queries that require complex reasoning online.

In this paper, we propose a novel framework to optimize and directly deploy a generative LLM for online relevance prediction. However, practical implementation presents significant challenges across three dimensions: (1) Deployment Latency: While Chain-of-Thought (CoT) reasoning enhances model reasoning capabilities and significantly improves the upper bound for solving complex problems, the proportional increase in output tokens correspondingly increases response latency. For relevance models, a single request requires computing relevance between a query and hundreds of candidate items, making real-time online generation of hundreds of CoT sequences computationally prohibitive. (2) Error Accumulation in CoT: Since CoT generates longer intermediate processes, a single hallucination or reasoning error at any intermediate step can propagate and affect final results, leading to incorrect classification outcomes. (3) Discriminative Hallucination: Even with a correct reasoning chain, the model may still produce an incorrect final answer.

To address these challenges, we propose an LLM-based optimization framework comprising three stages: (1) Supervised Fine-Tuning (SFT) with CoT to endow models with reasoning capabilities, improving the upper bound of model discrimination ability; (2) Offline pass@N sampling with Direct Preference Optimization (DPO) to enhance correct answering capability, reducing result errors caused by process errors; (3) Difficulty-based Dynamic Sampling with Group Relative Policy Optimization (GRPO) to further alleviate model discriminative hallucination. Experimental results demonstrate that our trained thinking model significantly outperforms baseline models on a curated offline dataset of challenging queries, exhibiting stronger reasoning and comprehension capabilities. Furthermore, it achieves substantial improvements in online side-by-side (SBS) human evaluations.

## 2 Related Works

### 2.1 Relevance Search

Relevance modeling is one of the most fundamental tasks in information retrieval, aiming to model the semantic relevance between queries and items. The field of relevance modeling has experienced extensive historical development. Early approaches focused on feature-based modeling, employing algorithms such as BM25[10] and TF-IDF[11], which relied on manually defined features to address relevance modeling challenges. While these methods proved effective in early information retrieval systems, they lacked cross-scenario generalizability and demanded substantial manual effort. Subsequently, deep learning-based representation modeling methods[12, 31] emerged, which utilized deep learning models to encode queries and items as embeddings and further modeled their relevance relationships. However, these approaches lacked the capacity to model broad knowledge and thus possessed limited potential for solving complex cases, with performance remaining suboptimal. The introduction of BERT[13] marked a paradigm shift, making fine-tuning of pre-trained language models with Transformer[14] architecture a mainstream approach for relevance modeling[15, 32, 33]. These methods demonstrated the capability to model complex semantic relationships, while their unified task paradigm and end-to-end training architecture significantly reduced the manual effort required for feature engineering. In recent years, decoder-only large language models have exhibited exceptional natural language understanding and generation capabilities, virtually unifying upstream and downstream task paradigms across natural language processing, leading to the emergence of relevance modeling research based on large language models. Notable works including LREF[16], ProRBP[17], and research by Mehrdad et al.[3] have employed training methodologies such as supervised fine-tuning (SFT) and Direct Preference Optimization (DPO)[18] to endow large language models with query-item relevance modeling capabilities. However, by adhering to discriminative optimization paradigms or distilling knowledge to smaller BERT-like models for deployment, most prior work fails to effectively address queries that require complex reasoning. In this paper, we address this gap by proposing a framework to optimize a large language model for relevance and deploy it directly in a production online setting.

### 2.2 Reasoning Large Language Models

While large language models have demonstrated exceptional capabilities across numerous downstream tasks, conventional large language models still struggle to excel in domains requiring strong reasoning abilities, such as mathematics and programming. To address this limitation, early approaches employed prompt engineering[20] and supervised fine-tuning[19] to guide models in generating step-by-step reasoning chains, thereby improving their reasoning capabilities. However, these methods lacked cross-domain generalizability and failed to fully leverage the rich world knowledge embedded in large language models. The emergence of o1[22] and DeepSeek-R1[21] has sparked a surge in using reinforcement learning to enhance the reasoning capabilities of large language models. These reasoning models, including o1 and DeepSeek-R1, employ reinforcement learning algorithms such as PPO[23] or GRPO[24] to reward answers sampled from the models themselves, thereby activating

the models' inherent rich world knowledge and improving their performance on challenging reasoning problems in mathematics and programming. Subsequently, additional reinforcement learning algorithms[25][26] and reasoning models[27][28] have been introduced, establishing the use of reinforcement learning algorithms on large language models to enhance reasoning capabilities as a widely adopted technical paradigm. However, current research predominantly focuses on the two domains of mathematics and programming, with limited exploration of reinforcement learning applications in vertical industries and diverse task paradigms. Our work provides an exploratory investigation into the application of reinforcement learning in the e-commerce search domain.

### 2.3 Reinforce Learning for Classification

E-commerce search relevance tasks can be formulated as binary or multi-classification problems, where the input consists of query and item features, and the output is the model-predicted relevance label. While numerous studies have explored reinforcement learning for training large language models, most current research focuses primarily on complex generation tasks in mathematics and programming domains, with relatively limited exploration of reinforcement learning applications in generative classification tasks. A particularly relevant work is GenCLS++[29], which systematically investigates the performance variations of large language models on generative text classification tasks under different generation and training paradigms. Interestingly, unlike their performance in mathematics and programming domains, in generative classification tasks, models often achieve better performance when they either do not output reasoning paths or place the reasoning process after the output label. Furthermore, conducting reinforcement learning after supervised fine-tuning can yield additional performance improvements. Related research includes the work by Li et al.[30], who explored the application of reinforcement learning in multi-modal large language models (MLLMs) for generative classification. Addressing the phenomenon that models perform better without explicit reasoning in generative classification tasks, they further investigated Adaptive-Thinking and achieved promising results. However, current research still lacks deep investigation into the underlying mechanisms behind the performance improvements brought by reinforcement learning. Our work systematically explores the performance gains of SFT, DPO[18], and GRPO[24] on generative classification tasks under different training and data configurations, while analyzing the underlying causes of these improvements. This provides an effective reference for applying reinforcement learning in generative classification tasks.

## 3 Methods

The task of e-commerce search relevance is formulated as a multi-class classification problem, in which the input comprises features of the query and the candidate product, and the output is a predicted relevance label. In the Taobao e-commerce search setting, we adopt a four-tier relevance class scheme: 4-Excellent, 3-Related, 2-Mismatch and 1-Irrelevant.

In this section, we detail our multi-stage optimization framework, which comprises four stages: SFT with CoT, Pass@N-based DPO, GRPO with Difficulty-based Sampling, and Online Deployment.

We first perform SFT to endow the model with basic relevance-judgment capability and introduce a method to extract continuous relevance scores for downstream tiering. To embed nuanced, domain-specific decision logic and prevent the model from degenerating into a purely discriminative model, we integrate CoT during SFT. CoT samples are synthesized via a Retrieval-Augmented Generation (RAG) pipeline to effectively incorporate complex business rules into training. Subsequently, motivated by the model's pass@N accuracy metric, we construct large-scale chosen/rejected pairs for DPO from two disjoint subsets: instances with pass@N accuracy  $> 0$  and instances with pass@N accuracy  $= 0$ . The goal is twofold: steer the model to correctly solve instances it is potentially capable of solving, and inject knowledge from a stronger model for the hard cases it consistently fails. Finally, informed by analysis of data distribution, we incorporate GRPO with online difficulty-based sampling, which yields further improvements even with small-batch data. Case studies indicate that small-scale online reinforcement learning mitigates discriminative hallucination and improves the quality of the model's reasoning process.

### 3.1 SFT

**3.1.1 SFT.** Conventionally, training data for query-product relevance prediction consists of  $\langle \text{query}, \text{item}, \text{label} \rangle$  triplets. A discriminative model is fine-tuned on a pre-trained BERT backbone in a point-wise manner, formulated as either a classification or regression task. The corresponding objective functions are typically formulated as follows: For regression to a single score  $\hat{y}$ :

$$\hat{y} = \mathbf{W}_{\text{mse}}^T \pi(\mathbf{x}) \quad (1)$$

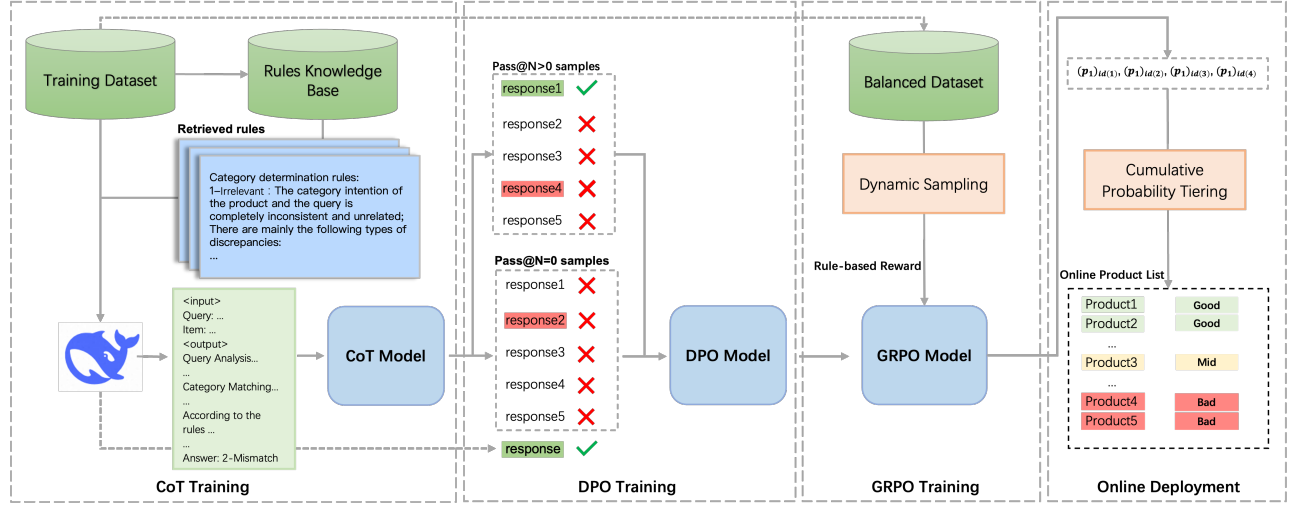
$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [(y - \hat{y})^2] \quad (2)$$

For classification to a probability vector  $\mathbf{p}$

$$\mathbf{p} = \text{Softmax}(\mathbf{W}_{\text{ce}} \pi(\mathbf{x})) \quad (3)$$

$$\mathcal{L}_{\text{CE}} = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} \left[ - \sum_{c=1}^C y_c \log(\mathbf{p}) \right] \quad (4)$$

where  $C$  is the total number of categories,  $\pi(\mathbf{x})$  denotes the model's output representation (e.g., the [CLS] token embedding or a pooled output of all tokens), and  $\mathbf{W}_{\text{mse}}$  or  $\mathbf{W}_{\text{ce}}$  represents a learnable projection matrix. For online serving, the resulting scalar  $\hat{y}$  or probability vector  $\mathbf{p}$  is used in downstream applications. More recently, this paradigm has been extended to Large Language Models (LLMs), which are now frequently adopted as the base model for SFT. In this context,  $\pi(\mathbf{x})$  is typically derived from the hidden state of the final End-of-Sequence (EOS) token. Following large-scale data SFT, both BERT-based and LLM-based approaches yield effective relevance discrimination models, serving as strong baselines. However, training an LLM solely on such discriminative samples effectively collapses an LLM into a pure discriminator, undermining its inherent generative capabilities and misaligning with its original pretraining objective. This prevents the model from fully leveraging the knowledge acquired during pretraining. Consequently, we adopt a generative training paradigm in which the model directly generates the label text (e.g., "4-Excellent") (Eq.5). For downstream applications, we extract the probability of the first generated token as a continuous-valued score (Eq.6). Formally, this is expressed as:



**Figure 1: Our proposed TaoSR1 framework comprising three stages: (1) SFT with CoT to endow models with reasoning capabilities; (2) Offline multiple sampling based on a pass@N strategy, combined with DPO, to enhance model generation quality; and (3) Difficulty-based dynamic sampling integrated with GRPO to further mitigate model’s discriminative hallucination problems.**

$$\mathcal{L}_{lm}(\pi) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} \left[ - \sum_{t=1}^{|\mathbf{y}|-1} \log \pi(y_{t+1} | \mathbf{x}_1, \dots, y_t) \right] \quad (5)$$

$$\mathbf{P}(y_1 = \mathbf{c} | \mathbf{x}) = (\mathbf{p}_1)_{id(\mathbf{c})} = \frac{\exp(\pi(\mathbf{x})_{id(\mathbf{c})})}{\sum_{j=1}^4 \exp(\pi(\mathbf{x})_{id(\mathbf{c}_j)})} \quad (6)$$

where  $(\mathbf{p}_1)_{id(\mathbf{c})} \in \mathbb{R}^4$ ,  $id(\mathbf{c})$  denotes the vocabulary index for the special tokens, here  $\mathbf{c}$  includes ["1", "2", "3", "4"];  $\mathbf{x}_t$  denotes the  $t$ -th token in a text sequence  $\mathbf{x}$ ;  $y_1$  denotes the extracted first generated token, and its probability is represented by  $\mathbf{p}_1$ ;  $(\mathbf{p}_1)_{id(\mathbf{c})}$  can be converted into a scalar by Eq.11 or directly used by downstream task. We will discuss it in detail in Sec. 3.4.

**3.1.2 RAG-based CoT Generation.** Nevertheless, e-commerce relevance annotations are governed by intricate, domain-specific business rules. A single label is insufficient to convey this underlying logic to the model. Simply switching to a generative objective still only leads the model to fit the empirical label distribution of the training data. To address this, we explicitly incorporate a CoT process during SFT to instill domain-specific reasoning capabilities into the model. Our structured CoT framework consists of five sequential steps: (1) Query Understanding, (2) Product Comprehension, (3) Category Matching, (4) Attribute Matching, and (5) Relevance Class Determination. Using the powerful Deepseek-R1 [9], we synthesized CoT samples based on our existing labeled data. Moreover, business rules in e-commerce relevance are complicated and often customized. For example, retrieving "Mate50 Pro" for the query "Mate50" is considered 3-Related, whereas retrieving "Mate50" for the query "Mate50 Pro" is rated 2-Mismatch. We therefore aim to enable the LLM to more faithfully internalize the business rules in our relevance scenarios.

During CoT sample synthesis, we incorporate business annotation rules into the prompt and require the model to explicitly cite the applied rules in the final step. However, the full documentation of these rules is extensive. Incorporating it entirely would create impractically long contexts and increase the model’s comprehension burden. To circumvent this limitation, we establish a structured knowledge base by decomposing the complete rule set into a collection of fine-grained, “atomic” rules, which are indexed by relevant metadata (e.g., category, brand, style) to facilitate efficient retrieval. During human annotation, we additionally label the atomic factors pertinent to each sample. When constructing CoT samples, we use the atomic factors for a given instance to dynamically retrieve the corresponding rules from the knowledge base and incorporate them into the prompt. This entire procedure constitutes a Retrieval-Augmented Generation (RAG) pipeline. More formally, for each training sample  $\langle \text{query}, \text{item}, \text{label}, \text{reason}_i \rangle$ , we use its atomic reasons to dynamically retrieve the corresponding rules from the RAG repository. These retrieved rules are then injected into the prompt, and Deepseek-R1 is prompted with  $\langle \text{prompt}, \text{RAG-rule}, \text{query}, \text{item}, \text{label} \rangle$  to synthesize the CoT training data. This pipeline yields the final training data in the form  $\langle \text{query}, \text{item}, \text{CoT}, \text{label} \rangle$ .

**3.1.3 SFT with CoT.** We experimented with two training paradigms: (1) a “think-then-respond” approach, where the model generates the CoT before the label ( $\langle \text{CoT}, \text{label} \rangle$ ), and (2) a “respond-then-think” approach with an output format of  $\langle \text{label}, \text{CoT} \rangle$ . Counterintuitively, the “think-then-respond” strategy, despite its theoretical advantages, consistently failed to outperform the purely discriminative baseline. Our analysis suggests this is due to error accumulation: while CoT generation elevates the model’s performance ceiling, showing encouraging results on challenging and long-tail cases (e.g., “alternatives to Miu Miu”, where text-only discriminative models often mishandle it by retrieving Miu Miu itself, failing to capture

**Table 1: Pass@N sampling Results**

Models	Accuracy
LLM base	75.01
TaoSR1(CoT) Pass@1	67.38
TaoSR1(CoT) Pass@2	74.26
TaoSR1(CoT) Pass@3	77.68
TaoSR1(CoT) Pass@4	80.18
TaoSR1(CoT) Pass@5	81.73

the semantics of “alternatives”), the longer generation process is more susceptible to error accumulation. Minor hallucinations or reasoning deviations in the intermediate steps accumulate and ultimately corrupt the final prediction. In contrast, the “respond-then-think” setting mitigates this issue by predicting the label first, leading to significantly improved metrics. Nevertheless, a performance gap relative to the discriminative baseline still remained. This leads us to hypothesize that CoT-based SFT, while effective at instilling rule-based logic, cannot generalize well across all scenarios. We posit that subsequent reinforcement learning is essential to unlock model potential.

### 3.2 Pass@N-based DPO

Our offline experimental analysis further corroborates the aforementioned hypothesis. Specifically, we find that the model’s pass@N accuracy, derived from multi-pass offline sampling, significantly exceeds that of single-pass decoding. Furthermore, as N increases, this metric eventually surpasses the performance of the pure discriminative baseline. Table 1 provides details of these pass@N results. These results suggest substantial room for improvement through reinforcement learning. We begin by optimizing the model using DPO, leveraging the results of our offline pass@N sampling on training set to construct a preference dataset. This process comprises two stages for constructing the preference dataset:

- **Self-Correction for Solvable Cases:** For instances where the model produces at least one correct response (i.e.,  $\text{pass@N} > 0$ ), we construct preference pairs. A correct response from the sampled candidates is designated as the “chosen” ( $y^+$ ), and a randomly selected incorrect response is designated as the “rejected” ( $y^-$ ). This process yields a preference dataset  $\mathcal{D}_{\text{pass}}$ , derived from a subset of our original training set.
- **Oracle-Guided Correction for Hard Cases:** A key limitation of self-correction is its inability to address cases the model consistently fails to solve. These hard cases with  $\text{pass@N} = 0$ , representing the model’s consistent failures, are the most critical targets for optimization. Therefore, for this cohort, we sample from DeepSeek-R1 and designate its correct outputs as the “chosen” responses. These oracle-generated responses are then paired with our model’s own incorrect “rejected” responses to form a supplementary preference dataset  $\mathcal{D}'_{\text{pass}}$ . Through this method, approximately 50% of the original  $\text{pass@N} = 0$  instances were curated into this new set.

The model is subsequently trained by minimizing the DPO loss over the combined preference datasets  $\mathcal{D}_{\text{pass}} \cup \mathcal{D}'_{\text{pass}}$ .

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(\mathbf{x}, y_w, y_l) \sim \mathcal{D}_{\text{pass}} \cup \mathcal{D}'_{\text{pass}}} \left[ \log \sigma \left( \beta \log \frac{\pi_\theta(y^+ | \mathbf{x})}{\pi_{\text{ref}}(y^+ | \mathbf{x})} - \beta \log \frac{\pi_\theta(y^- | \mathbf{x})}{\pi_{\text{ref}}(y^- | \mathbf{x})} \right) \right] \quad (7)$$

### 3.3 GRPO with Difficulty-based Sampling

The self-correction component of our DPO strategy is conceptually analogous to GRPO, as both encourage models to explore correct reasoning trajectories through multiple sampling of the model’s own outputs. For the chosen responses in our  $\text{pass@N} = 0$  set, the label-conditioned synthesis process often induced discriminative hallucination, where the generated rationale contradicts the final results. Relative to DPO, GRPO allows for online sampling with larger N, facilitates more diverse exploration to discover more logically coherent reasoning paths. This makes it a natural subsequent step for further optimization. Inspired by DAPO [5], we propose a difficulty-aware dynamic sampling strategy. A key departure from DAPO is our handling of homogeneous batches: when a sampled set of responses is all correct or all incorrect, we discard the sample entirely rather than resampling. This is because for query-item pairs with definitive relevance relationships, if multiple sampling still fails to produce correct answers, it indicates that the model may lack requisite knowledge. In such cases, the probability of obtaining a correct answer through further sampling becomes extremely low, which would instead slow down overall training efficiency. For groups in which all responses are correct, the within-group relative advantage is zero, leading to vanishing gradients that also degrade training efficiency. Moreover, during online RL we aim to focus learning on difficult instances; therefore, we compute advantages and backpropagate gradients only for groups whose empirical accuracy lies within the range  $(0, \gamma)$ . The training objective is defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{GRPO}}(\theta) = & \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \mathcal{D}} \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{o_i} \min \left( r_{i,t}(\theta) \hat{A}_t, \text{clip} \left( r_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon \right) \hat{A}_t \right) \right. \\ & \left. - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right] \\ \text{s.t. } & 0 < |\{o_i \mid \text{is\_equivalent}(y, o_i)\}| < \gamma \end{aligned} \quad (9)$$

where

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} | \mathbf{x}, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} | \mathbf{x}, o_{i,<t})}, \quad \hat{A}_t = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^G)}{\text{std}(\{R_i\}_{i=1}^G)} \quad (10)$$

,  $R$  is the verifiable reward, indicating whether the final predicted relevance class is correct. The interval configuration for difficulty-based sampling directly determine which data the model emphasizes. Our experiments with various  $\gamma$  settings, as reported in Table 5, showed substantial variation in performance. Further analysis revealed a crucial insight: we identified a strong inverse correlation between the model’s final performance and the coefficient of variation (CV) of the label distribution in the GRPO training data. This indicates that a more balanced label distribution yields superior performance on the relevance classification task. This insight motivated a data-centric optimization strategy. Specifically, we curated a balanced subset from the training data by downsampling all majority classes to match the size of the smallest. We then applied a minimal-rejection sampling policy, discarding only homogeneous batches (i.e.,  $\gamma$  lies in  $[0.01, 0.99]$ ). The final experimental results demonstrate that this approach, despite using

significantly less data, outperforms GRPO trained on the full, imbalanced dataset. Furthermore, a qualitative analysis of sampled outputs reveals that the incidence of errors attributable to “discriminative hallucination” in its failure cases is 30% lower compared to that of the DPO model.

### 3.4 Cumulative Probability Tiering

In the Taobao production search system, the model’s relevance predictions are operationalized through a three-tier stratification system: Good, Mid, and Bad, which correspond to our four-point annotation scale (4-Excellent and 3-Related correspond to “Good”, 2-Mismatch corresponds to “Mid”, 1-Irrelevant corresponds to “Bad”). Products are ultimately ranked in descending order of Good, Mid, and Bad tiers. In online applications, we usually do not directly use the raw classification output, but instead employ a threshold-based mechanism to assign items to these tiers. This allows us to precisely control the precision-recall trade-off based on specific business objectives. For instance, a high-precision setting prioritizes showing only highly relevant items, while a high-recall setting aims to capture a wider set of potentially relevant products. Conventionally, models trained with MSE loss typically introduce two hyperparameters as score anchor points for each class: usually 1-Irrelevant is fixed to 0, 4-Excellent is fixed to 1, and the remaining two class’s anchor  $\alpha_1, \alpha_2$  requires manual adjustment. Two subsequent thresholds  $\beta_1, \beta_2$  are then applied to produce the three tiers. For models trained with cross-entropy loss, four hyperparameters are similarly introduced. By computing a weighted average of the probabilities of the first generated class tokens  $((p_1)_{id(c)})$ , where  $c = [“1”, “2”, “3”, “4”]$ , we obtain a continuous score via Eq.11, followed by two thresholds  $\beta_1, \beta_2$  are used to achieve the three-tier division (See formula 12).

$$score = 0 * (p_1)_{id(1)} + \alpha_1 * (p_1)_{id(2)} + \alpha_2 * (p_1)_{id(3)} + 1 * (p_1)_{id(4)} \quad (11)$$

$$rel\_tier = \begin{cases} \text{good} & \text{if } score \geq \beta_2 \\ \text{mid} & \text{if } \beta_1 \leq score < \beta_2 \\ \text{bad} & \text{otherwise} \end{cases} \quad (12)$$

Both conventional methods, therefore, share a significant drawback: they necessitate multiple hyperparameters (at least four), which introduces considerable operational complexity in both training and deployment. Furthermore, the manual tuning of these hyperparameters via grid search is computationally expensive and often yields suboptimal configurations. To overcome these limitations, we propose **Cumulative Probability Tiering** (CumPT), a novel method that extends the standard practice from binary classification—using the positive class probability as the scoring threshold—to the multi-tier setting. CumPT elegantly unifies the process, requiring only a single hyperparameter for online multi-tier assignment, thereby eliminating the complex calibration and thresholding steps. Under the CumPT framework, model training remains unchanged, utilizing the standard cross-entropy objective (Eq.5) to predict probabilities for each of the four relevance classes. This mechanism operates by sequentially accumulating probabilities in descending order of the class tokens (from 4 down to 1) and comparing the sum against a single threshold  $\beta_{cum}$ . For instance, if the probability of 4-Excellent  $(p_1)_{id(4)}$  or cumulative probability  $(p_1)_{id(4)} + (p_1)_{id(3)}$  exceeds  $\beta_{cum}$ , the item is assigned to the “Good” tier. If not, we evaluate the cumulative probability  $(p_1)_{id(4)} + (p_1)_{id(3)} + (p_1)_{id(2)}$ ; if this sum exceeds  $\beta_{cum}$ , the item is assigned to the “Mid” tier. If this condition also fails, the item is assigned to the “Bad” tier. This procedure is formally outlined in Algorithm 1.

## 4 Experiments

### 4.1 Experiment Setup

**4.1.1 Dataset and Metrics.** Our test set is collected from Taobao’s online search logs, sampled from the scoring candidate space of the relevance

---

#### Algorithm 1 Cumulative Probability Tiering

---

**Require:** Cumulative probabilities  $(p_1)_c$  and threshold  $\beta_{cum}$

**Ensure:** Relevance tier  $rel\_tier$ ;

```

1: if  $(p_1)_{id(4)} \geq \beta_{cum}$  then
2:    $rel\_tier \leftarrow \text{Good}$ 
3: else if  $(p_1)_{id(4)} + (p_1)_{id(3)} + (p_1)_{id(2)} \geq \beta_{cum}$  then
4:    $rel\_tier \leftarrow \text{Good}$ 
5: else if  $p_4 + p_3 + p_2 \geq \beta_{cum}$  then
6:    $rel\_tier \leftarrow \text{Mid}$ 
7: else
8:    $rel\_tier \leftarrow \text{Bad}$ 
9: end if
10: return  $rel\_tier$ 

```

---

model. It comprises about 70,000 manually annotated query-item pairs. To rigorously evaluate the model’s reasoning capabilities, the query distribution is intentionally focused on four challenging categories: negation, affordable alternatives, question-answering (QA), and knowledge-based queries. The overall test set label distribution is shown in Table 3. It can be observed that for the relevance model’s scoring candidates, after filtering through the preceding pipeline with relevance factors, the proportion of completely irrelevant (1-Irrelevant) samples has become quite small. However, the more challenging partially irrelevant (2-Mismatch) samples still account for a relatively large proportion. Overall, the dataset maintains a relatively balanced distribution between relevant (i.e., 4-Excellent and 3-Related combined) and non-relevant (2-Mismatch and 1-Irrelevant combined) instances.

For offline metrics, we evaluate our model using macro-F1 and compare F1 metrics for each tier. For online evaluation, we conduct live A/B experiments and rely on human assessment to compute the following key metrics:

- **GSB (Good/Same/Bad):** This is a side-by-side comparative framework where human assessors compare results returned by two methods for the same query in A/B experiments. GSB measuring the relative superiority of the test bucket compared to the baseline bucket. GSB+x% can be understood as x% of results in the test bucket being better than the base bucket.
- **Query Goodrate:** This is a page-level relevance metric. The entire search results page for a query is classified into “Good”, “Mid”, or “Bad” tiers based on the proportion of relevant items displayed. The metric itself is the proportion of queries whose pages are rated as “Good” or “Mid”.
- **Item Goodrate:** This is an item-level metric that measures the absolute proportion of highly relevant items (i.e., 4-Excellent, 3-Related) per request. The item goodrate for this evaluation is the average of item goodrates across all requests.

It is important to note the distinction between these metrics: Query Goodrate and Item Goodrate measure absolute performance levels, for which we report the absolute lift. In contrast, GSB captures the relative advantage of our proposed model, for which we report the relative improvement.

**4.1.2 Implementation Details.** During the Supervised Fine-Tuning (SFT) phase, we used a batch size of 1024 and a learning rate of  $1e-6$ , managed by a cosine scheduler with a warmup ratio of 0.05. We employed the AdamW [6] optimizer with  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ . The model was trained for 2 epochs with a maximum input length of 4096 tokens.

For Direct Preference Optimization (DPO), we maintained a batch size of 1024 and a learning rate of  $1e-6$ , again using a cosine scheduler with a 0.05 warmup ratio and the AdamW optimizer. The training proceeded for 500 steps with the maximum input length kept at 4096 tokens. To stabilize training, we incorporated an auxiliary SFT loss term applied specifically to the “chosen” responses, with a weighting coefficient of 0.5. The pass@N

**Table 2: Offline Evaluation Results**

Models	Class-1 F1	Class-2 F1	Class-3 F1	Class-4 F1	Macro F1	Accuracy
BERT	65.74	69.63	33.87	76.06	61.33	69.36
Qwen3-0.6B	42.02	68.13	23.50	78.14	52.95	70.29
Qwen3-30B-A3B	65.09	68.80	32.47	81.68	62.01	74.42
LLM-base	65.19	68.86	32.91	81.90	62.22	75.04
TaoSR1(CoT)	43.30	67.54	19.68	75.62	51.54	68.22
TaoSR1(CoT&DPO)	62.90	71.20	37.96	82.25	63.58	75.54
TaoSR1(CoT&DPO&GRPO)	65.51	69.12	34.82	81.80	62.81	74.93
TaoSR1(CoT) <sub>postCoT</sub>	57.63	72.64	27.91	81.88	60.01	75.12
TaoSR1(CoT&DPO) <sub>postCoT</sub>	65.74	71.95	39.43	83.00	65.03	76.49
TaoSR1(CoT&DPO&GRPO) <sub>postCoT</sub>	<b>67.34</b>	<b>73.15</b>	<b>44.94</b>	<b>83.06</b>	<b>67.12</b>	<b>76.86</b>

**Table 3: Dataset Statistics**

Label	Count	Ratio
L4	38,739	50%
L3	3,863	5%
L2	27,913	36%
L1	6,865	9%

**Table 4: Pass@5 Statistics**

Pass@5	0	1	2	3	4	5
Ratio	13.7%	4.9%	4.6%	5.3%	7.4%	64.1%

**Table 5: Impact of Difficulty Ratio**

Difficulty Ratio	L1	L2	L3	L4	L-cv	macro F1
[0.2-0.8]	3.6%	29.1%	6.4%	60.9%	1.06	63.55
[0.2-0.6]	4.6%	29.9%	8.2%	57.3%	0.97	63.87
[0.1-0.5]	6.0%	31.8%	10.5%	51.7%	0.84	64.25
[0.01-0.9]	25.0%	25.0%	25.0%	25.0%	0	67.12
w balance label						

**Table 6: Ablation of DPO**

Models	Macro F1
TaoSR1(CoT) <sub>postCoT</sub>	60.01
TaoSR1(CoT&GRPO) <sub>postCoT</sub>	66.84
TaoSR1(CoT&DPO) <sub>postCoT</sub>	65.02
TaoSR1(CoT&DPO&GRPO) <sub>postCoT</sub>	<b>67.12</b>

samples are generated with hyperparameters top-k as 20, top-p as 0.95, and temperature as 0.6. A detailed statistics of the pass@N data distribution is provided in Table 4.

In the GRPO reinforcement learning phase, we configured a rollout batch size of 64 with a learning rate of  $1e-6$ . For each sample, we generated 16 responses using a sampling temperature of 0.99 and top-k of 100. The thresholds for our online difficulty sampling were set to a range of [0.01, 0.9]. For stabilization, we clipped the importance sampling ratio to a range of

$1 \pm 0.2$  and the advantage values to  $\pm 2$ . Our training pipeline is implemented based on the open-source framework ROLL[7].

**4.1.3 Baselines.** We employ multiple models as our baselines, including BERT, Qwen3-0.6B, and Qwen3-30BA3B[8]. Our model is iteratively developed based on Tbstar. Tbstar is a closed-source e-commerce large language model foundation trained from scratch by Taobao, utilizing over 10T tokens of general corpus and rich domain-specific e-commerce corpus. Based on a Mixture of Experts (MoE) architecture, it has a total parameter count of 42B with 3.5B activated parameters.

- BERT: A 24-layer BERT model foundation trained from scratch based on general corpus and Taobao’s internal e-commerce corpus, subsequently fine-tuned on the full training set using the formula 4.
- Qwen3-0.6B: The 0.6B parameter open-source dense model from the Qwen family. It was supervised fine-tuned (SFT) on our complete training dataset using the formula 5.
- Qwen3-30BA3B: An open-source Mixture-of-Experts (MoE) model from the Qwen family, featuring 30 billion total parameters and 3 billion active parameters per forward pass. This model was also supervised fine-tuned (SFT) on our full training set same as Qwen3-0.6B.
- LLM base: This serves as our strongest baseline and represents the starting point for our further optimizations. It is the Tbstar-42B model after being supervised fine-tuned (SFT) on our full training dataset using the formula 5.
- TaoSR1: This represents our full, multi-stage optimization pipeline built upon the LLM Base model, incorporating the CoT, DPO, and GRPO phases as described in Sec. 3. We specifically report results for TaoSR1<sub>postCoT</sub>, which denotes our final model employing the “respond-then-think” architecture for efficient deployment.

## 4.2 Offline Evaluation

Table 2 presents a comprehensive comparison between our model and various baselines. We focus on the macro-F1 metric rather than accuracy, as it provides a balanced measure of classification performance that is insensitive to class imbalance. Our results reveal three key observations:

**Domain-specific pre-training is crucial.** The 24-layer BERT pre-trained on our in-house e-commerce corpus significantly outperforms the Qwen3-0.6B model. Similarly, LLM base shows a substantial performance gain over the comparably sized Qwen-30BA3B.

**Directly incorporating CoT in SFT degrades classification performance due to error propagation.** This is evident in the TaoSR1(CoT) model, which shows a marked drop in F1-scores across all classes compared to the LLM Base. In contrast, our TaoSR1(CoT)<sub>postCoT</sub> model, which adopts the “respond-then-think” architecture, circumvents this issue by predicting

**Table 7: Performance Comparison of Different Methods with Threshold Tuning**

Methods	threshold	Macro F1	Online Macro F1	Good F1	Good Precision	Good Recall
Formula-based	$\beta_2 = 0.3$	67.12	41.42	73.66	58.73	98.77
	$\beta_2 = 0.4$	67.12	64.25	85.29	79.81	91.59
	$\beta_2 = 0.5$	67.12	64.83	85.34	80.43	90.89
	$\beta_2 = 0.6$	67.12	65.26	85.36	80.87	90.37
	$\beta_2 = 0.7$	67.12	56.83	83.35	81.05	85.78
CumPT	$\beta_{\text{cum}} = 0.3$	67.12	67.05	85.37	80.50	90.88
	$\beta_{\text{cum}} = 0.4$	67.12	67.13	85.39	80.75	90.59
	$\beta_{\text{cum}} = 0.5$	67.12	67.14	85.42	81.04	90.30
	$\beta_{\text{cum}} = 0.6$	67.12	67.15	85.40	81.26	89.99
	$\beta_{\text{cum}} = 0.7$	67.12	67.17	85.39	81.51	89.66

**Table 8: Side-by-Side Human Evaluations**

Query Type	Case	GSB	Query Goodrate	Item Goodrate
Q&A	What medicine can make hair black?	+16.62%	+6.53pt	+5.66pt
Alternative	Miumiu alternative	+34.43%	+13.11pt	+10.69pt
Negative	Short sleeves that don't stick to hair	+10.92%	+3.80pt	+3.74pt
Knowledge	Paint that is not afraid of car pressure	+18.45%	+6.85pt	+4.44pt

the label before generating the CoT. This leads to a significant performance recovery over the standard CoT approach.

**Reinforcement learning unlocks the full potential of CoT for downstream classification task.** TaoSR1(CoT&DPO) achieves substantial performance improvements compared to TaoSR1(CoT) and also surpasses LLM base. Similarly, TaoSR1(CoT&DPO)<sub>postCoT</sub> also demonstrates superiority compared to TaoSR1(CoT)<sub>postCoT</sub> and LLM base. Further optimization with GRPO yields additional performance gains, indicating that larger sampling spaces are key to reinforcement learning’s ability to enhance model capabilities.

Ultimately, our model achieves optimal performance under the post-CoT configuration through reinforcement learning optimization, with a 4.9 pt point improvement in macro-F1 compared to LLM base. TaoSR1<sub>postCoT</sub> not only resolves the error accumulation problem but also makes online deployment of the model feasible.

## 4.3 Ablation Study

**4.3.1 Ablation of Post-CoT.** We contrast the ‘think-then-respond’ paradigm with the ‘respond-then-think’ paradigm (distinguished by the postCoT subscript) in Table 2. We evaluate the performance of each paradigm both at the SFT stage and subsequent DPO stage. At the SFT-only stage, the post-CoT approach demonstrates a significant advantage over the standard approach. This provides strong evidence that the post-CoT effectively mitigates the issue of error accumulation in CoT, positioning it as a potentially superior approach for classification tasks. Furthermore, the application of DPO consistently yields notable performance gains under both configurations. This confirms that even within the post-CoT framework, there remains substantial headroom for improvement through reinforcement learning techniques.

**4.3.2 Why DPO before GRPO.** A natural question arises regarding the necessity of a distinct DPO stage, given that its core mechanism for pass@N > 0 samples—constructing preference pairs from offline multi-pass sampling—is conceptually analogous to the online sampling in GRPO. The key reason for the DPO stage lies in its handling of “hard cases” where our model consistently fails (i.e., pass@N = 0). For this samples, self-sampling is insufficient. Instead of relying on our model’s flawed outputs, we leverage

a more powerful, general-purpose “oracle” model to generate the “chosen” responses. These are then paired with our model’s own incorrect “rejected” responses. Through this approach, DPO training not only optimizes the model’s own reasoning paths but also introduces incremental information from external models. This beneficial information cannot be obtained solely through sampling the model itself. The value of this intermediate DPO stage is empirically validated by our experiments (see Table 6). These results demonstrate that initiating the GRPO phase from a DPO-refined checkpoint yields superior offline performance compared to applying GRPO directly to the SFT-only baseline.

**4.3.3 Ablation of CumPT.** This section provides a detailed analysis of the advantages of our proposed CumPT method. Macro F1 is computed directly on the model’s raw outputs, representing its intrinsic classification capability. Online Macro F1 is computed after the tiering mentioned in Sec. 3.4 (either the traditional formula-based approach via Eq.11, Eq.12 or CumPT). Additionally, we report the Precision, Recall, and F1-score specifically for the “Good” tier.

To illustrate the sensitivity of the traditional method, we fixed a subset of its many hyperparameters ( $\alpha_1 = 0.33$ ,  $\alpha_2 = 0.66$ ,  $\beta_1 = 0.15$ ) and varied only a single parameter  $\beta_2$  for comparison. The results in Table 7 reveal two critical findings: (1) CumPT achieves robust and stable performance with a single hyperparameter. Notably, the resulting Online Macro F1 score closely tracks the original Macro F1, indicating minimal performance degradation during the tiering process. (2) The traditional method exhibits high sensitivity to its hyperparameter settings, leading to significant variance in performance. This highlights a major practical challenge that identifying the optimal combination of its multiple hyperparameters requires extensive and costly experimentation.

In summary, CumPT not only simplifies the deployment process by reducing the number of required hyperparameters but also delivers more consistent and predictable performance, effectively bridging the gap between a model’s offline potential and its online efficacy.

## 4.4 Online Evaluation

To validate the effectiveness of the model in real online applications, we conducted a side-by-side evaluation based on manual annotation. We selected 2,000 queries, requested both experimental buckets for each query, and compared the top-10 results across three metrics. The specific results are shown in Table 8. We provide a fine-grained analysis of these results by segmenting the queries into four distinct categories. The most substantial gains were observed for queries seeking affordable alternatives. This is because such queries pose a classic challenge for traditional term-matching models, which often fail to distinguish between a target brand and its alternatives. For instance, a query for a “Miumiu alternative” would often cause a term-based model to incorrectly retrieve authentic Miumiu products with high relevance scores due to strong lexical overlap. Furthermore, significant improvements were also recorded across the other query categories. These results collectively demonstrate that the reasoning and general knowledge capabilities of our LLM-based model can effectively address long-standing challenges in search relevance—particularly for queries that require a deep, semantic understanding of user intent rather than simple lexical matching.

## 5 Conclusion

In this paper, we introduce a comprehensive framework for optimizing large language models for search relevance. Our methodology begins by endowing the model with structured reasoning capabilities through SFT with CoT. We leverage a Retrieval-Augmented Generation (RAG) pipeline to synthesize the CoT samples, effectively integrating complex business logic into the training process. Crucially, we explore both a conventional “think-then-respond” structure and a novel “respond-then-think” paradigm. The latter allows the model to internalize sophisticated reasoning patterns while remaining compatible with the low-latency demands of real-time online deployment. Furthermore, we employ a multi-stage reinforcement learning pipeline, incorporating DPO and GRPO, to significantly enhance the model’s robustness on challenging instances. This complete framework demonstrates significant and consistent improvements in both offline and online evaluations. The proposed framework offers a new paradigm for developing and deploying LLM-based relevance models. Moreover, our findings provide valuable insights into the application of LLMs for classification tasks, particularly on how to balance advanced reasoning capabilities with practical system constraints.

## References

- [1] T. Tang, Z. Tian, Z. Zhu, C. Wang, H. Hu, G. Tang, L. Liu, and S. Xu, “Lref: A novel llm-based relevance framework for e-commerce search,” in *Companion Proceedings of the ACM on Web Conference 2025*, ser. WWW ’25. New York, NY, USA: Association for Computing Machinery, 2025, p. 468–475. [Online]. Available: <https://doi.org/10.1145/3701716.3715246>
- [2] X. Ma, L. Wang, N. Yang, F. Wei, and J. Lin, “Fine-tuning llama for multi-stage text retrieval,” in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 2421–2425. [Online]. Available: <https://doi.org/10.1145/3626772.3657951>
- [3] N. Mehrdad, H. Mohapatra, M. Bagdouri, P. Chandran, A. Magnani, X. Cai, A. Puthenpuhussery, S. Yadav, T. Lee, C. Zhai *et al.*, “Large language models for relevance judgment in product search,” *arXiv preprint arXiv:2406.00247*, 2024.
- [4] Z. Chen, H. Wu, K. Wu, W. Chen, M. Zhong, J. Xu, Z. Liu, and W. Zhang, “Towards boosting LLMs-driven relevance modeling with progressive retrieved behavior-augmented prompting,” in *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert, K. Darwish, and A. Agarwal, Eds. Abu Dhabi, UAE: Association for Computational Linguistics, Jan. 2025, pp. 784–793. [Online]. Available: <https://aclanthology.org/2025.coling-industry.65/>
- [5] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, W. Dai, T. Fan, G. Liu, L. Liu, X. Liu, H. Lin, Z. Lin, B. Ma, G. Sheng, Y. Tong, C. Zhang, M. Zhang, W. Zhang, H. Zhu, J. Zhu, J. Chen, J. Chen, C. Wang, H. Yu, Y. Song, X. Wei, H. Zhou, J. Liu, W.-Y. Ma, Y.-Q. Zhang, L. Yan, M. Qiao, Y. Wu, and M. Wang, “Dapo: An open-source llm reinforcement learning system at scale,” 2025. [Online]. Available: <https://arxiv.org/abs/2503.14476>
- [6] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=Bkg6RiCqY7>
- [7] W. Wang, S. Xiong, G. Chen, W. Gao, S. Guo, Y. He, J. Huang, J. Liu, Z. Li, X. Li, Z. Liu, H. Zhao, D. An, L. Cao, Q. Cao, W. Deng, F. Du, Y. Gu, J. Li, X. Li, M. Liu, Y. Luo, Z. Liu, Y. Wang, P. Wang, T. Wu, Y. Wu, Y. Zhao, S. Zhao, J. Yang, S. Yang, Y. Tan, H. Yi, Y. Xu, Y. Yuan, X. Zhang, L. Qu, W. Su, W. Wang, J. Wang, and B. Zheng, “Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library,” 2025. [Online]. Available: <https://arxiv.org/abs/2506.06122>
- [8] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu, “Qwen3 technical report,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.09388>
- [9] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, S. Ye, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Zhao, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Xu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, and Z. Zhang, “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning,” 2025. [Online]. Available: <https://arxiv.org/abs/2501.12948>
- [10] K. M. Svore and C. J. Burges, “A machine learning approach for improved bm25 retrieval,” in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 1811–1814.
- [11] A. Aizawa, “An information-theoretic perspective of tf-idf measures,” *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003.
- [12] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, “Learning semantic representations using convolutional neural networks for web search,” in *Proceedings of the 23rd international conference on world wide web*, 2014, pp. 373–374.
- [13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.
- [14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [15] L. Bo, L. Pang, G. Wang, J. Xu, X. He, and J.-R. Wen, “Modeling relevance ranking under the pre-training and fine-tuning paradigm,” *arXiv preprint arXiv:2108.05652*, 2021.
- [16] T. Tang, Z. Tian, Z. Zhu, C. Wang, H. Hu, G. Tang, L. Liu, and S. Xu, “Lref: A novel llm-based relevance framework for e-commerce search,” in *Companion Proceedings of the ACM on Web Conference 2025*, 2025, pp. 468–475.
- [17] Z. Chen, H. Wu, K. Wu, W. Chen, M. Zhong, J. Xu, Z. Liu, and W. Zhang, “Towards boosting llms-driven relevance modeling with progressive retrieved behavior-augmented prompting,” *arXiv preprint arXiv:2408.09439*, 2024.
- [18] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, “Direct preference optimization: Your language model is secretly a reward model,” *Advances in neural information processing systems*, vol. 36, pp. 53 728–53 741, 2023.
- [19] C. Snell, J. Lee, K. Xu, and A. Kumar, “Scaling llm test-time compute optimally can be more effective than scaling model parameters,” *arXiv preprint arXiv:2408.03314*, 2024.
- [20] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, “Large language models are zero-shot reasoners,” *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.
- [21] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning,” *arXiv preprint arXiv:2501.12948*, 2025.

