Distillation and Refinement of Reasoning in Small Language Models for Document Re-ranking

Chris Samarinas
University of Massachusetts Amherst
Amherst, MA, United States
csamarinas@cs.umass.edu

Hamed Zamani
University of Massachusetts Amherst
Amherst, MA, United States
zamani@cs.umass.edu

ABSTRACT

We present a novel approach for training small language models for reasoning-intensive document ranking that combines knowledge distillation with reinforcement learning optimization. While existing methods often rely on expensive human annotations or large black-box language models, our methodology leverages web data and a teacher LLM to automatically generate high-quality training examples with relevance explanations. By framing document ranking as a reinforcement learning problem and incentivizing explicit reasoning capabilities, we train a compact 3B parameter language model that achieves state-of-the-art performance on the BRIGHT benchmark. Our model ranks third on the leaderboard while using substantially fewer parameters than other approaches, outperforming models that are over 20 times larger. Through extensive experiments, we demonstrate that generating explanations during inference, rather than directly predicting relevance scores, enables more effective reasoning with smaller language models. The self-supervised nature of our method offers a scalable and interpretable solution for modern information retrieval systems.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

1 INTRODUCTION

Search engines and retrieval-augmented generation systems increasingly face queries that require complex reasoning and multi-step synthesis and analysis. They demand a deeper understanding of the query and the documents to identify the connections between them. For example, finding documentation for a coding error requires understanding program logic and syntax, and identifying economic case studies that share underlying theoretical principles demands sophisticated domain knowledge and analytical reasoning [29]. Traditional approaches to training ranking models for such complex tasks often rely on expensive human annotations to provide relevance judgments and explanations. In contrast, we present a framework that automatically generates its own training signal by leveraging existing question-answer pairs on the Web.

Although neural ranking models have made significant progress in recent years [7, 13, 21, 22, 27] and led to substantial performance gains on standard benchmarks such as MS MARCO [19] and the TREC Deep Learning (DL) Track [3, 4], we observe that they often struggle with reasoning-intensive queries that demand deeper understanding and explicit justification of relevance decisions. For instance, state-of-the-art dense retrievers that achieve strong performance on TREC DL show significant degradation on reasoning-intensive queries, with the best models achieving only about 18% nDCG@10 on the BRIGHT benchmark [29]–a recent

benchmark designed for reasoning-intensive ranking tasks. We argue that ranking models must engage in deliberate *reasoning* to bridge the gap between query intent and document content.

Recent work has suggested that large language models with tens of billions of parameters can effectively serve as zero-shot rerankers [23, 24, 30], demonstrating strong reasoning capabilities across diverse domains. However, deploying these models at scale remains challenging due to their computational requirements and latency constraints. While smaller models offer practical advantages, they typically lack the sophisticated reasoning abilities of their larger counterparts. Recent LLMs such as DeepSeek R1 [5] have demonstrated that encouraging models to learn explicit reasoning strategies and leveraging inference-time compute for step-by-step analysis can significantly improve performance on complex tasks. While this has been demonstrated for language modeling and generation tasks, exploring these principles in retrieval remains understudied. Our work shows that by decomposing document relevance assessment into explicit reasoning steps and optimizing for high-quality explanations, we can achieve strong performance even with relatively compact models.

In more detail, our work introduces a framework for distilling and refining reasoning capabilities in small language models for reasoning-intensive ranking. Our approach does not require any manually labeled data for training; instead, we perform a diverse data scraping approach from the Web for collecting reasoning intensive questions and a pseudo-labeling approach using a *teacher* LLM (with 70B parameters), resulting in a dataset with 20K examples. We then introduce a knowledge distillation approach that helps a compact *student* LLM (with 3B parameters) to mimic the reasoning and labeling capability of the teacher. Subsequently, we introduce a reinforcement learning approach that refines these reasoning capabilities by rewarding high-quality explanations and accurate relevance predictions.

Through this approach, we demonstrate that our efficient 3B parameter model achieves performance comparable to 70B+ parameter models on reasoning-intensive ranking tasks. Most notably, our model ranks third on the BRIGHT benchmark leaderboard and is the first effective ranking model under 8B parameters, with the only models achieving better performance being a 70B zero-shot ranker using GPT-4 for query reformulation and JudgeRank, an ensemble of three LLMs (8B, 70B, and 405B parameters). Our 3B parameter model outperforms all other baseline methods on the BRIGHT benchmark, including recent approaches like Reason-to-Rank [12] which uses an 8B parameter model, while using almost
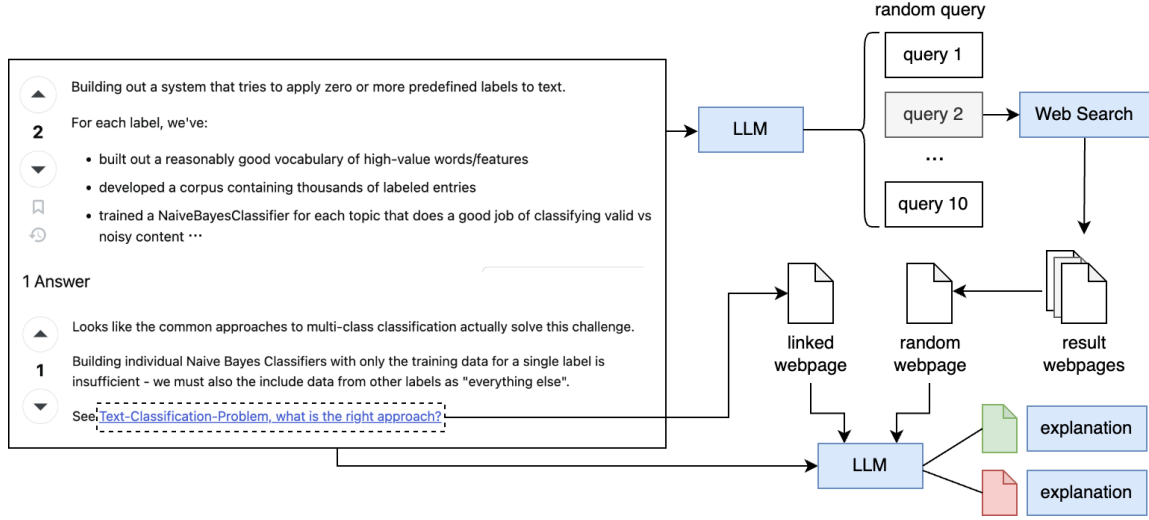
Figure 1: The automated synthetic data annotation process based on Stackexchange question-answer pairs.

three times fewer parameters and avoiding complex query rewriting or multi-step prompting strategies. We release our code and data for improved reproducibility.[1]

2 INTERANK

In this section, we present the training methodology for InteRank, a compact LLM for reasoning-intensive ranking. We leverage the reasoning capabilities of a large *teacher* LLM to train a compact *student* LLM that can both effectively re-rank documents and explain its decisions. The key insight is that by decomposing the ranking process into explicit reasoning steps and dedicating inference-time compute to step-by-step analysis, we can achieve superior performance compared to approaches that attempt to directly predict relevance scores. By training on synthetic explanations from a *teacher* LLM and optimizing for high-quality reasoning paths with reinforcement learning, we can effectively transfer reasoning capabilities to compact LLMs without requiring human-annotated data.

2.1 Model Architecture

We adopt a two-stage ranking architecture that is common in modern search systems: an efficient first-stage retrieval followed by more expressive reranking model capable of reasoning.

First-stage Retrieval: A lightweight sparse or dense retrieval model is used to retrieve potentially relevant documents from the corpus. To better understand the impact of retrieval quality on the final ranking performance, we experiment with various retrievers, including BM25 [26] and dense embedding models (see section 3). We retrieve the top 100 documents for re-ranking. While optimizing the first-stage retriever is important, in this work we focus on improving the second-stage re-ranking component.

Second-stage Re-ranking: Various learning-to-rank models, from traditional feature-based [2, 17] to transformer-based cross-encoder models [8, 21, 22], have been used for reranking. We aim at

training a reranking model for effective reasoning-intensive tasks. To do so, we train a language model that takes a query-document pair at a time and generates some reasoning to analyze and describe whether and how the provided document is relevant. These reasoning steps are then followed by a discrete relevance label as the final generation token. This relevance label is either 0 (i.e., non-relevant), 1 (partially relevant), or 2 (highly relevant). This stage is crucial for complex reasoning tasks, as it allows deeper analysis of document content in relation to the query intent. Since the scores produced by our reranker are discrete, many documents are assigned the same relevance score, and we cannot distinguish them in ranking. Therefore, we employ a hybrid scoring strategy that combines the generated discrete reranking score with the retrieval score produced in the first-stage retrieval. In fact,

$$\text{score}(q, d) = \text{retrieval score}(q, d) + \alpha \cdot \text{ reranking score}(q, d)$$

where q and d denote query and document and $\alpha \in \mathbb{R}^+$ is a hyperparameter controlling the impact of re-ranking score. α is expected to be a relatively high number $\gg 1$. ($\alpha = 100$ in our experiments).

2.2 Model Optimization

Our training process combines knowledge distillation [10] with reinforcement learning (RL). Following recent work showing the benefits of incentivizing explicit reasoning capabilities through RL [5], we structure our approach to encourage the development of effective reasoning patterns while maintaining computational efficiency. The process consists of three phases:

1. Synthetic Data Generation: High-quality training data is crucial for developing models that can handle diverse reasoning patterns. However, obtaining human annotations for reasoning-intensive ranking is expensive and time-consuming. We address this challenge through an automated data generation process that leverages existing question-answer pairs from social websites like StackExchange. Our data generation pipeline, summarized in Algorithm 1 and Figure 1, starts with a seed set of query-answer pairs C.

---

[1]https://github.com/algoprog/InteRank

In our experiments, we sampled 20K pairs round robin from 186 different communities on StackExchange. For each answer, we extract linked documents (hyperlinks) that potentially contain supporting evidence, establishing an initial set of query-document pairs. To increase diversity and to source potential negative documents, we use a *teacher* LLM to generate related queries and retrieve additional documents through web search using the Brave Search API. The teacher model is then instructed to generate an explanation and a discrete relevance label for each query-document pair, creating a distillation dataset. This approach naturally captures diverse reasoning patterns since the teacher model must explain how different types of evidence support or fail to support answers across technical domains - from code analysis to scientific explanations. The explanations demonstrate different forms of reasoning like logical deduction, causal analysis, and domain-specific technical reasoning. The next two phases are summarized in Algorithm 2.

2. Knowledge Distillation: We first transfer knowledge from a large zero-shot teacher model to a more compact student model through supervised fine-tuning. We use Llama 3.2 3B [9] as our base student model and as mentioned earlier, Llama 3.3 70B is used as our teacher model. The objective is to maximize the log likelihood of teacher-generated outputs:

$$\theta^1 = \arg\max_{\theta} \mathbb{E}_{(q,d,e,l)\sim D_{\text{synth}}} \left[\log p_\theta(e,l|q,d) \right] \quad (1)$$

where e and l denote an explanation and a discrete relevance label. θ^1 is the trained student model parameters after knowledge distillation. This phase helps the student model learn some initial reasoning patterns.

3. Reinforcement Learning: While distillation helps transfer basic reasoning patterns from the teacher, it is limited to imitating a single explanation path per example. In practice, there may be multiple valid ways to reason about document relevance. The reinforcement learning (RL) phase enables exploration of diverse reasoning strategies through sampling, with the reward model providing feedback to identify the most effective explanations. For each query-document pair, we sample $k = 8$ outputs from the model being trained (i.e., starting from the student model from Step 2) and evaluate them using a reward model. We observed that the reward values can have very high variance, and they heavily depend on query complexity and domain. For this reason, we use relative reward values after max-min normalization for each set of outputs $\hat{y}$ for a given query-document input (q, d):

$$\overline{\mathcal{R}}(q,d,\hat{y}) = \frac{\mathcal{R}(q,d,\hat{y}) - \min(\mathcal{R})}{\max(\mathcal{R}) - \min(\mathcal{R})} \quad (2)$$

where $\min(\mathcal{R})$ and $\max(\mathcal{R})$ are the minimum and maximum reward values for the given query-document input pair. High-quality output samples $\hat{y}_j$ are then selected using a threshold τ:

$$D_t = \{(q,d,\hat{y}_j,\overline{\mathcal{R}}) : \overline{\mathcal{R}}(q,d,\hat{y}_j) \geq \tau\} \quad (3)$$

The model parameters are updated using scaled rewards to further emphasize higher-reward outputs:

$$\theta^{t+1} = \arg\max_{\theta} \mathbb{E}_{(q,d,\hat{y},\overline{\mathcal{R}})\sim D_t} \left[(\overline{\mathcal{R}}(q,d,\hat{y}))^m \log p_\theta(\hat{y}|q,d) \right] \quad (4)$$

Algorithm 1 Synthetic Data Generation for Ranking

Input: Teacher LLM $\mathcal{T}$, query-answer pairs $C = \{(q_i, a_i)\}_{i=1}^N$
Output: Synthetic dataset D_{synth}
1: Initialize $D_{\text{synth}} \leftarrow \emptyset$
2: **for** each $(q, a) \in C$ **do**
3:     Extract linked documents D_{linked} from a
4:     **for** each $d \in D_{\text{linked}}$ **do**
5:         $(e, l) \leftarrow \mathcal{T}(q, d)$ ▷ Generate explanation and label
6:         $D_{\text{synth}} \leftarrow D_{\text{synth}} \cup \{(q, d, e, l)\}$
7:     **end for**
8:     $Q_{\text{gen}} \leftarrow \mathcal{T}(q, a, D_{\text{linked}})$ ▷ Generate related queries
9:     Sample random $q' \sim Q_{\text{gen}}$
10:    $D_{\text{web}} \leftarrow \text{WebSearch}(q')$ ▷ Get top-10 results
11:    Sample random $d \sim D_{\text{web}}$
12:    $(e, l) \leftarrow \mathcal{T}(q', d)$
13:    $D_{\text{synth}} \leftarrow D_{\text{synth}} \cup \{(q, d, e, l)\}$
14: **end for**
15: **return** D_{synth}

Algorithm 2 LLM alignment for ranking

Input: Student LLM $\mathcal{M}_\theta$, reward model $\mathcal{R}$, synthetic dataset D_{synth}
Output: Trained model parameters θ^{T+1}
1: $\theta^1 \leftarrow \arg\max_\theta \mathbb{E}_{(q,d,e,l)\sim D_{\text{synth}}} [\log p_\theta(e,l|q,d)]$
2: **for** $t = 1$ to T **do**
3:     **for** each (q, d, l) in training data **do**
4:         Sample $Y_{q,d} = \{\hat{y}_j\}_{j=1}^k \sim M_{\theta^t}(q,d)$ ▷ $k = 8$ samples
5:         Compute rewards $\mathcal{R}(q, d, \hat{y}_j)$ for all $\hat{y}_j$
6:         Normalize rewards: $\overline{\mathcal{R}} = \frac{\mathcal{R}-\min(\mathcal{R})}{\max(\mathcal{R})-\min(\mathcal{R})}$
7:     **end for**
8:     $D_t = \{(q, d, \hat{y}_j, \overline{\mathcal{R}}(q,d,\hat{y}_j)) : \overline{\mathcal{R}}(q,d,\hat{y}_j) \geq \tau\}$
9:     $\theta^{t+1} \leftarrow \arg\max_\theta \mathbb{E}_{(q,d,\hat{y},\overline{\mathcal{R}})\sim D_t} [\overline{\mathcal{R}}^m \log p_\theta(\hat{y}|q,d)]$ ▷ $m = 3$
10: **end for**
11: **return** θ^{T+1}

---

3 EXPERIMENTS

Evaluation Data. Our evaluation uses the BRIGHT benchmark [29], which spans diverse domains requiring complex reasoning capabilities. BRIGHT includes seven datasets from StackExchange communities (Biology, Earth Science, Economics, Psychology, Robotics, Stack Overflow, and Sustainable Living), each containing 100-200 expert-validated query-document pairs where relevance is determined by citations in accepted answers. The remaining 5 datasets focus on coding and mathematical reasoning: Pony (syntax documentation pairs), LeetCode (algorithmic problems), TheoremQA (theorem-based questions), AoPS (competition math problems), and Theorem Retrieval (problems paired with ProofWiki statements). In total, BRIGHT contains 1,384 queries with 6.37 positive documents per query on average. The queries are typically long-form questions requiring multi-step reasoning, while positive documents provide critical concepts, theories, or techniques needed to address the queries rather than direct answers.

Experimental Setup. The base LLM for InteRank is Llama 3.2 3B, while Llama 3.3 70B is our teacher model [9]. We use QLoRA [6] for parameter-efficient fine-tuning, with a 4-bit quantization of the base model and trainable rank-64 adapters. Due to resource constraints, we limit the context length to 4K tokens. Training

Table 1: Performance (nDCG@10) of different retriever and reranker combinations on the BRIGHT benchmark. Our 3B parameter model InteRank, matches or exceeds the performance of the 70B teacher LLM, with explanations being crucial for effectiveness (see "w/o expl." ablations). Adding domain-specific relevance definitions (marked with "+ instruct") further improves performance. The symbol * indicates statistical significance (paired t-test, $p < 0.05$) compared to all baseline models.

Retriever	Re-ranker	StackExchange							Coding		Theorem-based			Avg.
		Bio.	Earth.	Econ.	Psy.	Rob.	Stack.	Sus.	Leet.	Pony	AoPS	TheoQ.	TheoT.	
Sparse retrieval model														
BM25 [26]	-	19.2	27.1	14.9	12.5	13.5	16.5	15.2	24.4	7.9	6.0	13.0	6.9	14.8
BM25	InteRank (3B)	34.3	44.2	15.8	18.9	15.5	20.1	21.6	23.4	10.3	6.1	10.3	6.7	18.9
BM25	InteRank w/o expl. (3B)	15.1	20.2	12.1	10.2	11.2	13.3	12.8	19.9	6.1	4.8	10.1	5.2	11.8
BM25	**InteRank + instruct (3B)**	36.0	45.0	16.3	19.8	15.3	20.3	23.7	26.9	9.0	6.6	9.1	6.2	19.5
Dense retrieval models with < 1B parameters														
TAS-B (66M) [11]	-	2.7	10.2	6.4	5.6	7.5	8.0	4.1	24.7	14.6	8.7	7.9	1.5	8.5
BGE (0.3B) [33]	-	12.0	24.2	16.6	17.4	12.2	9.5	13.3	26.7	5.6	6.0	13.0	6.9	13.6
Inst-L (0.3B) [28]	-	15.6	21.5	16.0	21.9	11.5	11.2	13.2	20.0	1.3	8.1	20.9	9.1	14.2
GTE-L (0.4B) [15]	-	21.0	31.1	20.5	24.3	12.6	15.9	15.3	28.3	7.3	8.3	20.3	11.6	18.0
GTE-L (0.4B)	MiniLM-MARCO (33M) [25]	11.6	5.7	5.2	6.4	2.7	4.0	5.5	4.0	2.1	0.0	3.6	1.2	4.3
GTE-L (0.4B)	ModernBERT-L (0.4B) [32]	10.8	7.1	7.2	7.4	4.1	5.8	7.0	5.8	6.8	6.4	4.6	3.8	6.4
GTE-L (0.4B)	Llama3.2 (3B) [9]	18.3	22.5	11.1	17.5	7.1	11.1	12.6	18.6	7.0	4.0	18.0	15.7	13.6
GTE-L (0.4B)	Llama3.3 (70B) [9]	29.6	37.2	23.1	30.7	13.6	22.8	20.5	23.7	18.6	7.0	23.9	23.4	22.8
GTE-L (0.4B)	InteRank (3B)	**35.2**	**45.7**	24.1	27.4	16.1	21.8	**20.8**	22.0	11.7	8.7	17.4	7.5	21.5
GTE-L (0.4B)	InteRank w/o expl. (3B)	20.3	19.7	14.4	16.1	13.1	11.4	13.8	19.6	10.2	9.1	15.4	9.4	14.4
GTE-L (0.4B)	**InteRank + instruct (3B)**	37.0	46.5	24.8	28.8	15.8	22.1	23.0	25.3	10.2	9.5	15.4	7.0	22.1
Dense retrieval models with > 1B parameters														
E5 (7B) [31]	-	18.8	26.0	15.5	15.8	16.4	9.8	18.5	28.7	4.8	7.1	26.1	26.8	17.9
Inst-XL (1.5B) [28]	-	21.9	34.4	22.8	27.4	17.4	19.1	18.8	27.5	5.0	8.5	15.6	5.9	18.7
GritLM (7B) [18]	-	25.0	32.8	19.0	19.9	17.3	11.6	18.0	**29.8**	22.0	8.8	25.1	21.1	20.9
Qwen1.5 (7B) [1]	-	30.1	38.3	17.7	23.7	13.3	22.4	14.6	25.5	8.5	14.5	27.7	**32.8**	22.4
Qwen1.5 (7B)	MiniLM-MARCO (33M) [25]	9.72	6.21	6.60	6.72	3.59	5.12	6.25	5.11	6.10	5.90	4.04	3.26	5.72
Qwen1.5 (7B)	ModernBERT-L (0.4B) [32]	11.8	8.1	8.2	8.4	5.1	6.8	8.0	6.8	7.8	7.4	5.6	4.8	7.4
Qwen1.5 (7B)	Llama3.2 (3B) [9]	27.6	30.3	14.6	19.5	9.7	17.6	11.9	25.4	14.6	12.8	25.6	26.1	19.6
Qwen1.5 (7B)	InteRank (3B)	48.5	50.6	21.7	30.3	**17.6**	26.3	20.2	21.3	**26.7**	12.4	**21.7**	27.4	27.1
Qwen1.5 (7B)	InteRank w/o expl. (3B)	21.3	25.6	15.2	16.8	13.8	16.2	15.1	22.4	11.2	10.1	16.2	10.1	16.2
Qwen1.5 (7B)	**InteRank + instruct (3B)**	**51.2***	**51.4***	22.4	**31.9***	17.3	**26.6***	**22.4***	24.5	23.1	**13.5***	19.3	25.5	**27.4***

Table 2: Performance (nDCG@10) of the reranker in various training stages with GTE-large as first-stage retriever.

Domain	SFT	RL, t=1	RL, t=2	δ(t=1 vs SFT)	δ(t=2 vs t=1)
Bio.	**39.4**	35.2	30.0	-4.2	-5.2
Earth.	42.4	**45.7**	38.4	+3.3	-7.3
Econ.	23.2	**24.1**	22.7	+0.9	-1.4
Psy.	27.1	**27.4**	25.9	+0.3	-1.5
Rob.	14.1	**16.1**	13.6	+2.0	-2.5
Stack.	**21.8**	21.8	18.2	0.0	-3.6
Sus.	20.6	**20.8**	16.7	+0.2	-4.1
Leet.	19.9	22.0	**25.3**	+2.1	+3.3
Pony	7.4	11.7	**15.6**	+4.3	+3.9
AoPS	6.9	**8.7**	8.6	+1.8	-0.1
TheoQ.	12.8	17.4	**20.2**	+4.6	+2.8
TheoT.	8.8	7.5	**9.8**	-1.3	+2.3
Average	20.3	**21.5**	20.4	+1.1	-1.1

is performed on a single A100 GPU with an effective batch size of 16 (batch size 1 with 16 gradient accumulation steps) using the AdamW optimizer with learning rate 2e-4. For the sampling of outputs in the RL stage, we use temperature 1.0 for nucleus sampling, reward threshold $\tau = 0.85$, and reward scaling power $m = 3$. We perform two epochs of RL training. For the reward model, we use a pretrained Llama 3.1 8B model [16] that has demonstrated strong performance on RewardBench [14]. We found that this model has very high agreement with larger open-weight and commercial LLMs in relative comparison of explanation outputs, making it suitable for our training process.

Baselines. We compare against a diverse set of baseline models: (1) Traditional sparse retrieval using BM25 [26]; (2) Dense retrievers of varying sizes, from MSMARCO-trained models like TAS-B (66M)

[11] to recent models like BGE (0.3B) [33], Instruction-tuned models Inst-L/XL [28], GTE-large (0.4B) [15], E5 (7B) [31], GritLM (7B) [18], and Qwen1.5 (7B) [1]; (3) Cross-encoder rerankers including MiniLM fine-tuned on MSMARCO [25] and ModernBERT-large fine-tuned on our synthetic examples [32]; and (4) Zero-shot LLM rerankers using Llama 3.2 (3B) and Llama 3.3 (70B). These baselines represent the spectrum of current approaches, from lightweight traditional methods to LLMs.

3.1 Experimental Results

Our experimental results, shown in Table 1, reveal several key findings about reasoning-intensive ranking:

1. Traditional dense retrievers with small number of parameters or training data fail in reasoning-intensive domains. Smaller dense retrievers trained on MSMARCO like TAS-B (66M) perform poorly with only 8.53% average nDCG@10, highlighting their limitations beyond simple semantic matching. This is particularly evident in reasoning-intensive domains like theorem-based tasks (1.51% on TheoT) and complex StackExchange queries (2.77% on Biology). In contrast, larger dense retrievers trained on more diverse data with 100M+ training examples, show significant improvements; GTE-large (400M) achieves 18.0% and Qwen1.5 (7B) reaches 22.4% average nDCG@10, demonstrating the importance of model scale and training data for complex retrieval tasks.

2. Explanations are crucial for effective ranking. As shown in Table 1, our ablation studies reveal that removing the explanation component (rows marked "w/o expl.") causes accuracy to drop significantly from 21.5 to 14.4 nDCG@10 on average. Traditional

BERT re-ranking models that rely purely on semantic matching also show surprisingly poor performance, with ModernBERT-L achieving only 4.57% average nDCG@10. This shows that the process of generating explanations helps develop better reasoning capabilities compared to approaches that only predict relevance scores directly.

3. Distillation results in small student models with teacher performance. Our results also demonstrate that our approach successfully distills complex reasoning capabilities into a compact 3B parameter model, achieving performance comparable to models over 20 times larger (see Llama 3.3 70B in Table 1). When combined with the Qwen1.5 retriever and domain-specific relevance definitions in the ranker's prompt (rows marked with "+ instruct" in Table 1), InteRank achieves state-of-the-art performance with an average of 27.4% across all domains reaching the third spot in BRIGHT leaderboard, just below JudgeRank [20], an ensemble of 3 zero-shot LLMs (8B, 70B, and 405B parameters) and a baseline using Llama 70B with query-rewriting with GPT-4. Our 3B parameter model outperforms all other baseline methods on the BRIGHT benchmark, including recent approaches like Reason-to-Rank [12] (nDCG@5 26.2 vs 19.6) which uses an 8B parameter model.

4. RL improves reasoning for ranking. The iterative RL process shows domain-dependent effects, as detailed in Table 2. While the first iteration leads to broad improvements (+1.1% nDCG@10 on average), the second iteration reveals an interesting pattern - performance continues to improve in reasoning-intensive domains like mathematics and coding while declining in domains with simpler reasoning requirements. This suggests that additional RL iterations help refine complex reasoning capabilities but may lead to overfitting in domains where simpler strategies suffice. Table 2 presents detailed results examining the impact of different training stages. The supervised fine-tuning (SFT) stage establishes strong initial performance, particularly in domains like Biology and Earth Science. The first RL iteration shows the largest gains in theoretical domains (TheoQ), coding tasks (Pony, Leetcode), and earth science. The second iteration further improves performance specifically in reasoning-intensive tasks (Leetcode, Pony, TheoQ, TheoT) while showing decline in simpler domains, highlighting the trade-off between specialized reasoning capabilities and general performance.

4 CONCLUSIONS

This paper presents a novel approach for training compact language models to perform reasoning-intensive document ranking. Our methodology combines knowledge distillation from a large teacher model with reinforcement learning optimization to create efficient yet powerful ranking models that can explain their decisions. Through extensive experimentation we demonstrate that a 3B parameter LLM achieves performance comparable to models over 20 times larger, reaching state-of-the-art results across diverse domains. Dedicating inference-time compute to generate explanations, rather than directly predicting relevance scores, enables more effective reasoning with smaller language models.

Key findings from our work include: (1) the critical role of explanations in developing robust reasoning capabilities for ranking, as shown by significant performance drops when removing the explanation component; (2) the effectiveness of our two-stage training approach, where supervised fine-tuning establishes strong initial performance and targeted reinforcement learning helps refine complex reasoning abilities; and (3) the importance of combining efficient retrievers with reasoning-capable re-rankers, as demonstrated by the strong performance of our GTE-large and Qwen1.5 retriever combinations.

Our results suggest promising directions for future work, including exploring more sophisticated reward modeling approaches, domain adaptation techniques for specialized reasoning tasks, and developing methods to further reduce model size while maintaining reasoning capabilities. Additionally, applying similar weakly-supervised alignment techniques to optimize first-stage retrievers could potentially lead to end-to-end improvements in reasoning-intensive search. The success of our approach in creating efficient, interpretable ranking models opens new possibilities for deploying reasoning-intensive search systems at scale.

REFERENCES

[1] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and et al. 2023. Qwen Technical Report. arXiv:2309.16609 [cs.CL] https://arxiv.org/abs/2309.16609

[2] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[3] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the TREC 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[4] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Ellen M Voorhees, and Ian Soboroff. 2021. TREC deep learning track: Reusable test collections in the large data regime. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 2369–2375.

[5] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, and Bingxuan Wang et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

[6] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems* 36 (2024).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021*. Springer, 280–286.

[9] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and Ahmad Al-Dahle et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531 [stat.ML] https://arxiv.org/abs/1503.02531

[11] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *Proc. of SIGIR*.

[12] Yuelyu Ji, Zhuochun Li, Rui Meng, and Daqing He. 2024. ReasoningRank: Teaching Student Models to Rank through Reasoning-Based Knowledge Distillation. (2024). arXiv:2410.05168 [cs.CL] https://arxiv.org/abs/2410.05168

[13] Julian Killingback, Hansi Zeng, and Hamed Zamani. 2025. Hypencoder: Hypernetworks for Information Retrieval. arXiv:2502.05364 [cs.IR] https://arxiv.org/

abs/2502.05364

[14] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. RewardBench: Evaluating Reward Models for Language Modeling. arXiv:2403.13787 [cs.LG] https://arxiv.org/abs/2403.13787

[15] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281* (2023).

[16] Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024. Skywork-Reward: Bag of Tricks for Reward Modeling in LLMs. *arXiv preprint arXiv:2410.18451* (2024).

[17] Tie-Yan Liu. 2010. Learning to rank for information retrieval. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Geneva, Switzerland) *(SIGIR '10)*. Association for Computing Machinery, New York, NY, USA, 904. https://doi.org/10.1145/1835449.1835676

[18] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. *arXiv preprint arXiv:2402.09906* (2024).

[19] 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. *CoRR* abs/1611.09268 (2016). arXiv:1611.09268 http://arxiv.org/abs/1611.09268

[20] Tong Niu, Shafiq Joty, Ye Liu, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. JudgeRank: Leveraging Large Language Models for Reasoning-Intensive Reranking. (2024). arXiv:2411.00142 [cs.CL] https://arxiv.org/abs/2411.00142

[21] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[22] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424* (2019).

[23] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankVicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088* (2023).

[24] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze! *arXiv preprint arXiv:2312.02724* (2023).

[25] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

[26] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference (TREC-3)* (overview of the third text retrieval conference (trec-3) ed.). Gaithersburg, MD: NIST, 109–126. https://www.microsoft.com/en-us/research/publication/okapi-at-trec-3/

[27] Chris Samarinas, Wynne Hsu, and Mong Li Lee. 2021. Improving Evidence Retrieval for Automated Explainable Fact-Checking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, Avi Sil and Xi Victoria Lin (Eds.). Association for Computational Linguistics, Online, 84–91. https://doi.org/10.18653/v1/2021.naacl-demos.10

[28] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1102–1121. https://doi.org/10.18653/v1/2023.findings-acl.71

[29] Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, et al. 2024. BRIGHT: A Realistic and Challenging Benchmark for Reasoning-Intensive Retrieval. *arXiv preprint arXiv:2407.12883* (2024).

[30] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? Investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542* (2023).

[31] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533* (2022).

[32] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. arXiv:2412.13663 [cs.CL] https://arxiv.org/abs/2412.13663

[33] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed Resources For General Chinese Embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 641–649. https://doi.org/10.1145/3626772.3657878