Shifting from Ranking to Set Selection for Retrieval Augmented Generation

Dahyun Lee[†1] **Yongrae Jo**[†1] **Haeju Park**[†1] **Moontae Lee**[1,2]
[1]LG AI Research, [2]University of Illinois Chicago
{leedhn, yongrae.jo, haeju.park, moontae.lee}@lgresearch.ai

Abstract

Retrieval in Retrieval-Augmented Generation (RAG) must ensure that retrieved passages are not only individually relevant but also collectively form a comprehensive set. Existing approaches primarily rerank top-k passages based on their individual relevance, often failing to meet the information needs of complex queries in multi-hop question answering. In this work, we propose a set-wise passage selection approach and introduce SETR, which explicitly identifies the information requirements of a query through Chain-of-Thought reasoning and selects an optimal set of passages that collectively satisfy those requirements. Experiments on multi-hop RAG benchmarks show that SETR outperforms both proprietary LLM-based rerankers and open-source baselines in terms of answer correctness and retrieval quality, providing an effective and efficient alternative to traditional rerankers in RAG systems. The code is available at https://github.com/LGAI-Research/SetR

1 Introduction

Bridging parametric knowledge with external information is vital for ensuring accurate and reliable generation in language models. Retrieval-Augmented Generation (RAG) overcomes critical limitations of Large Language Models (LLMs), particularly their inability to incorporate up-to-date or domain-specific knowledge without retraining (Mallen et al., 2023). The risk of generating hallucinations is another concern when the model capacity is not sufficiently large (Huang et al., 2024). By integrating an external retrieval system that provides contextually relevant and grounded evidence in real time, RAG improves both the accuracy and reliability of knowledge-intensive tasks.

A critical component of RAG systems is the retrieval and reranking module, as the quality of the retrieved information directly influences the accuracy and relevance of the generated answers (Shi et al., 2023; Wu et al., 2024a,b; Wadhwa et al., 2024; Hong et al., 2024; Feng et al., 2024). Numerous studies have explored optimizing the effective integration of LLMs and retrieval modules. Asai et al. (2023); Jeong et al. (2024) focus on dynamically determining the necessity of retrieval and when to stop it. Wang et al. (2023); Shao et al. (2023) involve alternating between retrieval and generation, enriching contextual references through multiple retrieval iterations. Trivedi et al. (2023); Sarthi et al. (2024) iteratively decompose and refine complex questions, addressing them through retrieval and generation. However, these multi-step approaches require significantly more resources, potentially limiting their feasibility for real-world applications. Therefore, many existing RAG systems utilize conventional reranking modules with a straightforward top-k selection strategy, originally developed and optimized for search applications.

We argue that RAG systems have distinct information demands that set them apart from traditional search engines. While traditional search engines rank individual results by relevance, RAG systems need a curated set of passages to generate accurate answers, requiring not only relevance but also diversity, completeness, and the comprehensiveness of retrieved passages. For example, when identifying a company based on a mix of business strategies, controversies and product claims, a RAG system must retrieve a diverse range of sources covering all these aspects. If the system retrieves only passages discussing product claims but not those covering controversies, business strategies, and other relevant factors, it may produce an incomplete or inaccurate response.

To address these challenges, we propose a set-wise passage selection approach that optimizes the quality of the passage set as a whole, rather than treating retrieval as an independent ranking task.

[†]These authors contributed equally to this work.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17606–17619

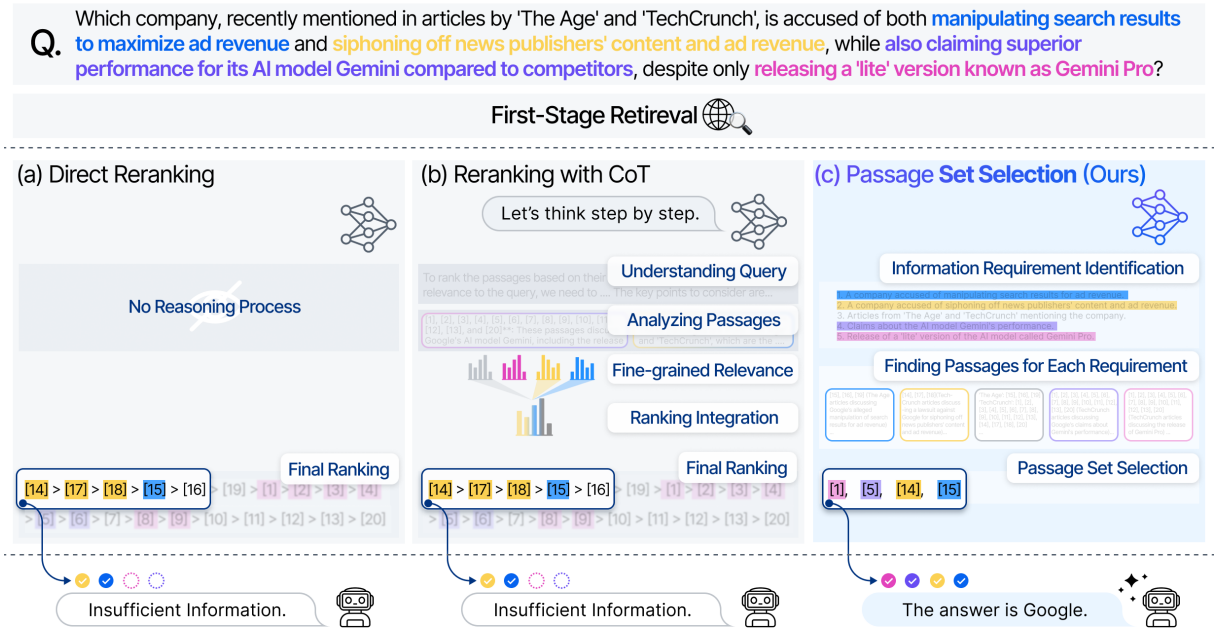


Figure 1: Overview of our SET SELECTION approach for RAG system, compared to passage reranking methods: **(a) Direct Reranking**, **(b) Reranking with CoT**, and **(c) Our SET SELECTION**. (a) lacks explicit reasoning, making it unclear whether multiple aspects are considered. (b) incorporates a reasoning process but may misrepresent or underemphasize key information when integrating relevance for final ranking. (c) SET SELECTION explicitly identifies all necessary information and selects relevant passages, ensuring more comprehensive information. This example is from the MultiHopRAG dataset (Tang and Yang, 2024).

This holistic strategy encourages comprehensive coverage of essential information while reducing redundancy within the selected set (Figure 1).

To make it practical for real-world applications, we introduce SETR, a fine-tuned LLM designed to implement our set-wise passage selection approach. The model first analyzes the user question using Chain-of-Thought (CoT) reasoning to identify its information requirements, and selects an optimal subset from the full list of retrieved passages, maximizing coverage and relevance. This enables our method to serve as an effective alternative to rerankers in RAG systems.

Experiments on multi-hop RAG benchmarks, including HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MusiQue (Trivedi et al., 2022), and MultiHopRAG (Tang and Yang, 2024), demonstrate that set-wise passage selection significantly enhances the effectiveness of RAG systems. It outperforms both proprietary LLM-based reranking and open-source rerankers, achieving higher answer correctness. Moreover, retrieval performance evaluation on MultiHopRAG (Tang and Yang, 2024) indicates improvements in precision and recall, further underscoring its strong retrieval capabilities even in isolation.

The ablation study of SETR reveals that the performance boost stems from both the set-wise passage selection approach and CoT reasoning for identifying information requirements, each making a distinct and effective contribution to retrieval quality. The analysis shows that both components enhance information coverage in the retrieved set while effectively rejecting negative candidates.

The contributions of this work are threefold:

- **Set-wise Passage Selection for RAG:** We propose an information requirement-based set-wise passage selection approach that ensures collective coverage of retrieved passages, optimizing the retrieved set as a whole rather than treating retrieval as an independent ranking task.

- **Comprehensive Evaluation of Set Retrieval and Generation:** To validate the effectiveness of our approach, we conduct extensive evaluations on both the retrieved passage sets and the final generated outputs. Our experiments on multi-hop RAG benchmarks demonstrate that our method outperforms both proprietary LLM-based rerankers and open-source alternatives, achieving better answer correctness, while also improving retrieval precision and recall.

- **Open-Source Contribution:** We release the complete and fully reproducible recipe of SETR, implementing our set-wise passage selection approach. We hope this work facilitates future research and community-driven advancements in retrieval strategies for RAG systems.

2 Related Work

2.1 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) systems combine retrieval modules with language models to enhance factual accuracy and reduce hallucinations (Lewis et al., 2021; Guu et al., 2020). Standard pipelines typically employ a first-stage retriever such as BM25 (Robertson and Zaragoza, 2009) or DPR (Karpukhin et al., 2020) followed by a reranking module that estimates relevance via pointwise, pairwise, or listwise strategies (Nogueira and Cho, 2019; Qin et al., 2023; Zhuang et al., 2023; Yoon et al., 2024).

Recently, Large Language Models (LLMs) have been applied to listwise reranking in RAG systems through prompting and distillation (Sun et al., 2024; Pradeep et al., 2023b), demonstrating strong performance. However, these models primarily focus on individual passage relevance and often overlook set-level properties such as diversity or coverage, which are crucial for generating complete and accurate answers in multi-hop or compositional question answering tasks (Tang and Yang, 2024).

2.2 Refinement and Iteration in RAG

To address the limitations of ranking-based retrieval, recent studies have explored retrieval strategies better aligned with the needs of RAG. These include adaptive retrieval based on query complexity (Jeong et al., 2024), multi-step reasoning with agent-based ranking (Niu et al., 2024), and context pruning (Chirkova et al., 2025). Iterative methods such as Self-RAG (Asai et al., 2023) and CoRAG (Wang et al., 2024) refine queries or perform multi-round retrieval to improve relevance.

While effective, these approaches often entail substantial computational overhead and are sensitive to prompt design and hyperparameter choices (Asai et al., 2023; Niu et al., 2024). In contrast, our study provides a lightweight and efficient alternative by selecting a coherent subset of passages in a single step through explicit modeling of information requirements. This set-oriented approach is compatible with iterative retrieval pipelines and can also function as a direct replacement for conventional rerankers in standard RAG systems.

3 SETR: Set-wise Passage Selection for Retrieval-Augmented Generation

In this section, we present SETR (Set-wise passage selection for Retrieval-Augmented Generation), a novel retrieval paradigm that moves beyond conventional reranking strategies. We begin by formalizing the passage selection task and motivating the need for a set-oriented perspective (§3.1). Next, we present our information requirement identification (IRI) method, which utilizes Chain-of-Thought (CoT) reasoning to guide passage selection (§3.2). Finally, we describe the architecture and training methodology of SETR, a distilled model fine-tuned for efficient set-wise passage selection (§3.3), along with the data construction details (§3.3.1) and training procedures (§3.3.2).

3.1 Task Definition

We define the passage retrieval task as the process of selecting an optimal set of passages from a pool of retrieved candidate passages to address a specific information need, such as supporting precise and coherent responses in RAG systems. Traditionally, this task has been framed as a reranking problem, where passages are scored individually and the top-k results are selected based on their relevance scores.

However, we argue that relevance-based reranking alone is insufficient for retrieval modules in RAG systems, which require more holistic retrieval strategies. To address this, we propose a set-wise retrieval approach that jointly optimizes the relevance, completeness, and conciseness of the retrieved set. This method also eliminates the need to manually select the top-k value in reranking, streamlining the process.

3.2 Information Requirement Identification via CoT Reasoning

We design a prompting strategy that enables set-wise passage selection by systematically identifying information requirements through a structured, step-by-step reasoning process. As illustrated in Figure 2, our prompt guides a zero-shot CoT reasoning process that decomposes the input question into distinct information subgoals. The key process

consists of three key steps: (1) enumerating the key information requirements necessary to answer the question; (2) identifying passages that contain relevant information for each requirement; and (3) selecting a subset of passages that collectively provide the most comprehensive and diverse coverage to effectively answer the query.

By prompting Large Language Models (LLMs) with both the question and a set of candidate passages retrieved in an earlier retrieval stage, this method enables fine-grained analysis and effective set selection. In contrast to zero-shot listwise reranking, our approach imposes no constraint on the inclusion of all candidate passages and does not enforce any ranking or ordering in the final selection.

3.3 Model Distillation

To ensure the proposed approach is practical for real-world applications, we train SETR, a distilled model fine-tuned for the set-wise passage selection task through information requirement identification. While proprietary LLMs exhibit strong performance, their cost and latency would make their use in real-time search systems impractical. Instead, we distill step-by-step reasoning ability into a specialized, lightweight model for efficiency.

3.3.1 Data Construction

For distillation, we construct a dataset based on 40K training questions[1] from Pradeep et al. (2023b), originally derived from the MS MARCO v1 passage ranking dataset (Sun et al., 2024). Each query is paired with the top-20 retrieved candidate passages. We then apply set-wise passage selection to generate teacher-labeled selections, which are subsequently distilled into our student model. To perform this labeling, we use GPT-4o with a zero-shot prompting approach. Following Pradeep et al. (2023a), we replaced all instances of [n] in passages with (n) to prevent model confusion during data synthesis and inference. We used the fix_text function from ftfy[2] to preprocess all inputs before feeding them into the model.

3.3.2 Training

We adopt Llama-3.1-8B-Instruct[4] as the base model and train it using a standard supervised fine-tuning approach. The input consists of a prompt including the user question and retrieved passages, while the output includes the CoT reasoning from the teacher model along with the selected passages.

Passage Selection Prompt of SETR

```
I will provide you with {num} passages,
each indicated by a numerical identifier
[].  Select the passages based on their
relevance to the search query: {question}.

{context}

Search Query: {question}

Please follow the steps below:
Step 1.  Please list up the information
requirements to answer the query.
Step 2.  for each requirement in Step 1,
find the passages that has the information
of the requirement.
Step 3.  Choose the passages that mostly
covers clear and diverse information to
answer the query.  Number of passages is
unlimited.  The format of final output
should be '### Final Selection: [] []',
e.g., ### Final Selection: [2] [1].
```

Figure 2: The set-wise passage selection prompt with Chain-of-Thought information requirement identification process for SETR.

For the ablation study, we present two additional model variations. We refer to the original model as SETR-CoT & IRI for comparison with these variations. Full prompt details are provided in Appendix A.4.

- **SETR-Selection only** is a model trained to generate only the final selected passages without any reasoning process.
- **SETR-CoT** is a model trained with general CoT reasoning using a standard "Let's think step-by-step prompt" prompt, but does not explicitly identifying distinct information requirements.
- **SETR-CoT & IRI** is the full model that incorporates both CoT reasoning and explicit information requirement identification, and performs passage selection accordingly.

4 Experiments

In this section, we first introduce the experimental setup (§4.1). Then, we show the results for both the generation (§4.2) and retrieval stages (§4.3), highlighting the effectiveness of the proposed SETR. More details are shown in Appendix A.

[1] https://huggingface.co/datasets/castorini/rank_zephyr_training_data

[2] https://pypi.org/project/ftfy

Retrieval	Model	# of Passages	HotpotQA EM	HotpotQA F1	2WikiMultiHopQA EM	2WikiMultiHopQA F1	MuSiQue EM	MuSiQue F1	MultiHopRAG Accuracy
BM25	RETRIEVAL ONLY								
	-	5.00	26.90	25.86	29.79	21.79	5.46	8.22	39.20
	RERANKING								
	bge-reranker-large	5.00	29.71	28.08	30.16	21.84	6.12	10.00	42.13
	RankLlama	5.00	29.48	27.82	30.30	21.91	6.04	9.26	42.09
	RankVicuna	5.00	28.69	27.31	30.46	22.42	5.99	9.03	40.53
	RankZephyr	5.00	28.96	27.76	30.29	22.34	6.78	10.03	40.10
	FirstMistral	5.00	26.71	26.10	30.15	21.97	5.29	8.42	40.29
	RankGPT (gpt-4o)	5.00	30.89	29.24	31.71	23.31	6.91	9.98	**44.36**
	SET SELECTION (OURS)								
	SetR-Selection only	2.95	31.61	30.55	**32.22**	24.20	**8.02**	**11.07**	43.62
	SetR-CoT	2.48	30.79	30.12	32.07	**24.43**	7.03	10.87	41.63
	SetR-CoT & IRI	2.63	**32.20**	**30.57**	32.17	24.22	6.62	10.57	44.13
bge-large-en-v1.5	RETRIEVAL ONLY								
	-	5.00	30.07	30.97	31.17	25.22	7.44	10.78	41.82
	RERANKING								
	bge-reranker-large	5.00	32.48	33.24	31.92	25.47	8.06	12.50	43.50
	RankLlama	5.00	31.88	32.95	32.24	25.78	7.61	11.77	43.51
	RankVicuna	5.00	32.08	32.83	32.66	26.85	7.78	11.35	42.76
	RankZephyr	5.00	31.83	32.97	32.68	26.59	8.02	11.72	41.55
	FirstMistral	5.00	30.10	31.07	31.43	25.31	6.53	10.64	42.05
	RankGPT (gpt-4o)	5.00	33.85	34.45	34.36	28.06	9.43	13.25	45.69
	SET SELECTION (OURS)								
	SetR-Selection only	3.41	**36.68**	37.84	34.84	29.40	10.38	15.28	46.20
	SetR-CoT	2.88	36.46	**38.20**	35.34	30.34	9.76	14.31	45.26
	SetR-CoT & IRI	2.91	36.62	38.11	**35.44**	**30.35**	**10.79**	**15.43**	**47.14**

Table 1: End-to-end question answering results across various ranking models. Each model applies reranking or selection over the top-20 passages retrieved using either BM25 or bge-large-en-v1.5. The **bold** and underlined indicate the best and second-best performances respectively. "# of Passages" indicates the average number of passages included in the prompt context during answer generation.

4.1 Setup

Benchmarks. For evaluation, we conduct experiments in two folds: (1) end-to-end QA, and (2) retrieval task. For comprehensive evaluation, we adopt four widely used complex multi-hop QA datasets: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and MultiHopRAG (Tang and Yang, 2024). These datasets cover diverse question types and multi-hop reasoning scenarios, offering a comprehensive evaluation of QA models in complex, real-world contexts.

Baselines. We compare SETR with state-of-the-art ranking baselines across different model categories. Specifically, we include traditional unsupervised ranking models such as BM25 (Robertson and Zaragoza, 2009), supervised dense ranking models including bge-large-en-v1.5 (Xiao et al., 2023), and bge-reranker-large (Xiao et al., 2023), as well as LLM-based ranking models such as RankLlama (Ma et al., 2024), RankVicuna (Pradeep et al., 2023a), RankZephyr (Pradeep et al., 2023b), FirstMistral (Chen et al., 2024) and RankGPT (gpt-4o[3]) (Sun et al., 2024).

Implementation Details. SETR is built upon Llama-3.1-8B-Instruct[4], trained for 5 epochs with an effective batch size of 512 and a learning rate of 5×10^{-6} using AdamW optimizer (Loshchilov and Hutter, 2019). To evaluate the effectiveness of SETR, we keep the first-stage retrieval and the generator fixed. The first-stage retrieval uses bge-large-en-v1.5 (Xiao et al., 2023), a high-performance retrieval model, while the generator is Llama-3.1-8B-Instruct[4] which we use without fine-tuning. For RAG, we adopt the standard RAG framework (Ram et al., 2023) to generate answers based on the retrieved contexts. All baselines are implemented utilizing the Rankify (Abdallah et al., 2025) toolkit[5].

4.2 End-to-end QA Evaluation

The main results of various ranking models are presented in Table 1. The results in RETRIEVAL ONLY are derived solely from first-stage retrievers, while RERANKING and SET SELECTION correspond to the results of re-ranking or selecting over the top-20 candidates retrieved by the respective first-stage retrievers. Based on the results, the key observations are as follows: (1) In terms of answer correctness, SETR significantly outperforms

[3] gpt-4o refers to gpt-4o-2024-08-06 from OpenAI

[4] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[5] https://github.com/DataScienceUIBK/Rankify

all baselines by selecting the optimal set of passages from the retrieved passages, achieving notably higher F1 and Accuracy. These results are quite impressive, considering that SETR uses 40-50% fewer passages on average compared to baselines. (2) The performance of LLM-based ranking baselines, such as RankLlama and RankZephyr, was less satisfactory. We observe that the relatively small bge-reranker-large (Xiao et al., 2023) performs comparably to, and in some cases even outperforms LLM-based baselines. This is likely due to complex questions making retrieval more challenging, introducing more noise, and requiring not only relevance but also diversity, completeness, and comprehensiveness. We posit that the key to the RAG system lies in having more useful knowledge and fewer distracting passages, enabling even a simple and smaller model to outperform LLMs when the correct RAG paradigm is applied, which we will discuss in detail in (§5.1).

4.3 Retrieval Evaluation

The retrieval evaluation is conducted using both rank-based metrics such as MRR and NDCG, as well as presence-based selection metrics including Precision and Recall, on the MultiHopRAG dataset (Tang and Yang, 2024). The results of retrieval evaluation are shown in Table 2. The results indicate that our model consistently achieves 3.8%-4.6% higher precision compared to off-the-shelf baselines, whereas it maintains competitive performance on rank-based metrics even with a small number of passages.

LLM-based rankers, paired with advanced retrievers, perform well on rank-based metrics compared to the first-stage retrieval baselines. Specifically, RankGPT (`gpt-4o`) shows notable improvements in both rank-based and presense-based metrics. However, we observe a discrepancy between rank-based metrics and end-to-end QA performance, as shown in Table 1.

This discrepancy arises from the assumption in rank-based metrics that relevance is determined on a one-to-one basis between a document and a query. For more complex question types, such as multi-hop questions, where considering relationships between multiple documents is needed for comprehensive information retrieval, these metrics may need further refinement to improve accuracy.

Model	RANK-BASED		PRESENCE-BASED	
	MRR@10	NDCG@10	Prec@5	Recall@5
RETRIEVAL ONLY				
BM25	0.4429	0.6827	0.1109	0.2413
bge-large-en-v1.5	0.4523	0.6900	0.1612	0.3232
RERANKING				
bge-reranker-large	0.6019	0.7481	0.1619	0.3276
RankLlama	<u>0.6311</u>	**0.7703**	0.1679	0.3375
RankVicuna	0.5077	0.7232	0.1372	0.2760
RankZephyr	0.5326	0.7046	0.1340	0.2685
FirstMistral	0.4521	0.6895	0.1321	0.2671
RankGPT (`gpt-4o`)	**0.6358**	<u>0.7628</u>	0.1799	<u>0.3601</u>
SET SELECTION (OURS)				
SETR-Selection only	0.5610	0.7295	<u>0.2187</u>	0.3554
SETR-CoT	0.5533	0.7281	0.2047	0.3413
SETR-CoT & IRI	0.5742	0.7255	**0.2268**	**0.3669**

Table 2: Retrieval performance on the MultiHopRAG benchmark. RANK-BASED metrics reflect passage order, while PRESENCE-BASED metrics consider only presence. Best and second-best scores are in **bold** and <u>underlined</u>, respectively.

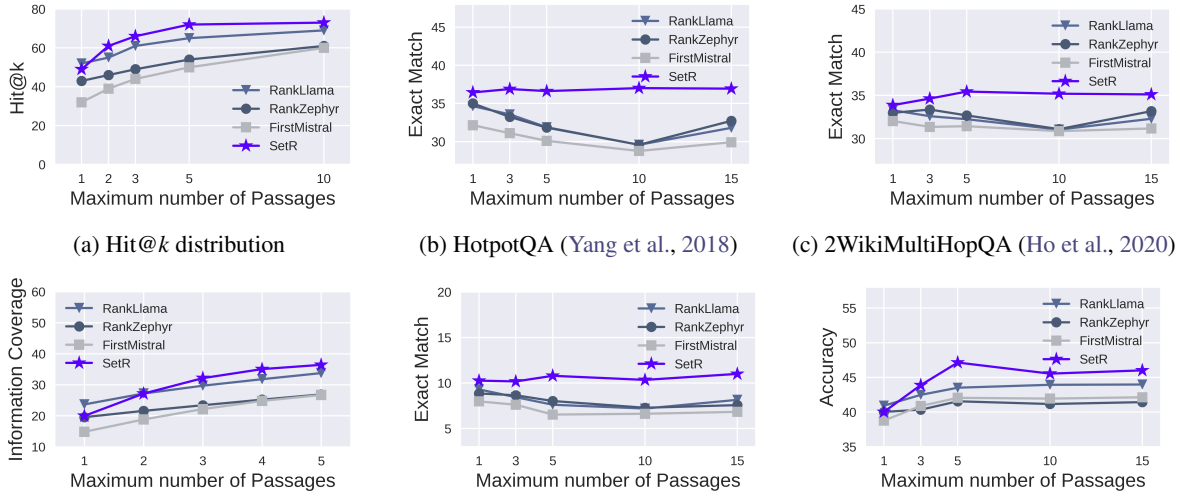
5 Analysis

In this section, we analyze the key components that contribute to the effectiveness of our set-wise passage selection approach SETR. We evaluate the quality of selected passages in terms of information coverage and robustness (§5.1), assess the impact of reasoning strategies including information requirement identification (IRI) and Chain-of-Thought (CoT) §5.2, design a controlled setup to isolate method-level effects (§5.3), and compare token efficiency between selection-based and reranking-based methods (§5.4).

5.1 Effectiveness of Set Selection: Informativeness and Robustness

We analyze the effectiveness of our selection approach in terms of two key perspectives: (1) informativeness; how comprehensively the selected passages cover the necessary information, and (2) robustness; the model's ability to discard irrelevant or redundant content.

Informativeness. Traditional ranking-based retrieval systems typically score and select passages based on their individual relevance rather than the collective information gained from multiple passages, and often fail to account for content duplication. To address this, we additionally analyze *information coverage*, measuring how *newly* retrieved gold evidence accumulates as the number of selected documents increases. This helps capture how *valid* information accumulates as additional documents are included. The MultiHopRAG dataset provides gold evidence lists, extractively collected from documents, allowing precise mea-



(a) Hit@k distribution (b) HotpotQA (Yang et al., 2018) (c) 2WikiMultiHopQA (Ho et al., 2020)

(d) Average information coverage (%) (e) MusiQue (Trivedi et al., 2022) (f) MultiHopRAG (Tang and Yang, 2024)

Figure 3: (a) Hit@k distribution, and (d) average information coverage (%) are measured based on gold evidence lists from the MultiHopRAG benchmark. (b), (c), (e), and (f) report results of the standard RAG pipeline across benchmarks, varying the maximum number of passages utilized for answer generation. Note that SETR utilizes fewer than 5 passages on average for answer generation.

surement of how much necessary information is covered by the selected passages. The formula used to calculate information coverage is detailed in Appendix A.3. We report two complementary metrics: Hit@k, which reflects whether gold evidence appears in the top-k selected passages, and *information coverage*, which measure the accumulation of distinct gold evidences across the selected evidences.

As illustrated in Figure 3a and Figure 3d, despite SETR selects passages without any ordering, it outperforms reranking methods with a notable improvement at Hit@k from 48.87% to 69.90%. Furthermore, SETR enhances information coverage from 19.33% to 36.49%, whereas other rerankers achieve only a modest average gain of 9.80% in the acquisition of new information. These findings suggest that reranking methods may struggle to retrieve documents containing previously uncovered information due to the presence of duplicate content and hard negatives. In contrast, our method captures both individual relevance and collective information coverage, which are essential for improving retrieval effectiveness and overall answer correctness.

Robustness. We further assess the robustness of our approach to retrieval noise by analyzing how answer generation quality changes as the number of passages increases. As shown in Figure 3b, 3c, 3e, and 3f, simply increasing the number of input passages often leads to performance degradation, suggesting that more is not always better. This is especially pronounced in multi-hop QA, where irrelevant or contradictory evidence can mislead the generator. In contrast, SETR consistently achieves stronger performance while utilizing significantly fewer passages. On average, it uses 2.91 passages compared to the standard top 5 used in reranking. This demonstrates the model's ability to discriminate high-utility information from distractors, thereby maintaining both efficiency and factual precision. These findings suggest that effective passage selection requires balancing recall and conciseness, and that a smaller, curated set can outperform longer, noisier contexts.

5.2 Effectiveness of Reasoning Components: The Role of CoT and IRI

To understand the role of reasoning in passage selection, we conduct an ablation study with three SETR variants: (1) without any reasoning, (2) with general CoT reasoning, and (3) with our proposed IRI-based reasoning.

As presented in Table 1 and Table 2, in terms of precision, we observe a significant performance improvement when applying IRI-based explicit requirement analysis and selection, compared to using standard CoT reasoning alone, or no reasoning process. Table 3 shows a similar trend, applying IRI-based reasoning yields stronger end-to-end RAG performance compared to other methods; more details are in §5.3. These results suggest

Method	HotpotQA		2Wiki		MuSiQue		MHRAG
	EM	F1	EM	F1	EM	F1	Accuracy
RERANKING							
Rank only♦	33.85	34.45	34.36	28.06	9.43	13.25	45.69
Rank + CoT♦	34.61	35.26	34.77	28.18	9.52	13.51	45.26
SET SELECTION							
SETR-Selection only♦	38.28	40.12	35.83	**31.14**	11.50	16.37	46.24
SETR-CoT♦	37.46	39.44	**35.85**	31.07	10.79	15.56	44.95
SETR-CoT & IRI♦	**39.16**	**40.49**	35.68	31.09	**12.33**	**16.91**	**46.40**

Table 3: Ablation study with teacher model inference to isolate method-level effects. All experiments use `gpt-4o-2024-08-06` as the reference model, annotated with ♦. For RERANKING, RankGPT4 prompt (Sun et al., 2024) is employed. Prompts for SET SELECTION are provided in Appendix A.4. **2Wiki** and **MHRAG** refer to the 2WikiMultiHopQA and MultiHopRAG benchmarks, respectively.

Model	HotpotQA		2Wiki		MuSiQue		MHRAG
	EM	F1	EM	F1	EM	F1	Accuracy
BUILT ON ZEPHYR-7B-β (Pradeep et al., 2023b)							
RankZephyr (original)	29.76	30.36	31.19	24.92	6.95	10.57	41.55
BUILT ON LLAMA-3.1-8B-INSTRUCT							
RankZephyr♣	34.69	35.04	33.87	27.83	8.61	12.79	43.90
RankZephyr + CoT♣	33.99	34.38	33.66	27.85	9.43	13.27	43.60
SETR-CoT & IRI	**36.62**	**38.11**	**35.44**	**30.35**	**10.79**	**15.43**	**47.14**

Table 4: Fair comparisons under a unified setting, with confounding factors minimized. RankZephyr♣ and RankZephyr + CoT♣ were implemented using the same LLaMA-3.1-8B-Instruct model as our method, fine-tuned on data re-annotated by `gpt-4o-2024-08-06`.

that the performance gains of our approach do not simply stem from leveraging the intrinsic thinking steps of LLMs, and highlight the critical role of our IRI step in assembling a passage set with maximum information coverage, demonstrating the best performance across all metrics and benchmarks.

5.3 A More Equitable Comparison: Method-Level Effects

While prior sections demonstrate that our approach improves end-to-end performance, such comparison can still be influenced by external factors, including differences in base models, data sources, or teacher supervision. To more rigorously assess the intrinsic effectiveness of our set-wise selection approach, we conduct a method-level evaluation that explicitly controls for these confounding factors. Specifically, we implement two strategies to minimize the impact of confounding factors: (1) upper bound; a teacher model directly performs the selection task, to isolate the contribution of the method itself, and (2) unified setting; all baselines are retrained using the same base model, training data, and teacher supervision, ensuring a fair and method-focused comparison.

Teacher model as Upper Bound. We first design an upper-bound evaluation setup where all models, whether reranking or selection-based, leverage the same powerful teacher model, GPT-4o, to generate passage selections or ranking. This setup isolates the effect of the selection formulation itself by minimizing the influence of model capacity or training-specific artifacts. Table 3, which reports the method-level effects with teacher model, indicates that our set selection method outperforms traditional reranking strategies, highlighting the individual contributions of both CoT reasoning and IRI to enhanced retrieval performance.

Specifically, the SETR method with IRI consistently achieves strong performance across all benchmarks and demonstrates competitive or superior results compared to both traditional reranking and other set selection variants. Notably, while both Rank + CoT and SETR-CoT share the same CoT prompt, the results reveal that using set-wise selection rather than integrated ranking in the final stage leads to improved retrieval outcomes. This suggests that constructing a unified ranking may lead to the omission of certain aspects of the reasoning process, potentially resulting in information loss. Furthermore, the comparison between SETR-CoT and SETR-CoT & IRI demonstrates that explicitly identifying essential information during the reasoning process helps improve selection precision and ensures broader information coverage.

Unified Setting for Direct Comparison. To further minimize potential bias, we create a unified training setting, in which all models are (1) built on the same base architecture (Llama-3.1-8B-Instruct), (2) trained on the same teacher supervision (re-annotated with `gpt-4o-2024-08-06`), and (3) evaluated using identical generation protocols. As shown in Table 4, SETR again surpasses reranking baselines in end-to-end QA accuracy across all tasks, reaffirming the advantages of selection-based approaches. This carefully controlled comparison highlights a critical insight: ranking-based reasoning, even with CoT, tends to collapse multiple reasoning chains into a single score, potentially obscuring key informational elements. In contrast, our selection strategy preserves intermediate reasoning steps, resulting in better alignment with the actual information needs of the question. These findings collectively demonstrate that the observed performance gains stem not from stronger base

Method	MultiHopRAG		HotpotQA		2WikiMultiHopQA		MuSiQue	
	Reranker Output	Generator Input	Reranker Output	Generator Input	Reranker Output	Generator Input	Reranker Output	Generator Input
DIRECT INFERENCE								
RankZephyr (Llama-3.1-8B-Inst)	80	2672	80	1426	80	1504	80	1403
SETR-Selection only	17	1441	11	461	10	422	11	499
WITH REASONING								
RankZephyr + CoT (Llama-3.1-8B-Inst)	603	2665	574	1435	560	1434	599	1385
SETR-CoT	396	1517	383	426	376	333	412	439
SETR-CoT & IRI	409	1240	317	432	276	332	340	503

Table 5: Efficiency analysis on token usage. We report the number of output tokens generated during retrieval and the number of input tokens fed into the generator, both of which serve as proxies for computational efficiency.

models or teacher supervision alone, but from our methodological reformulation of retrieval as set selection grounded in reasoning.

5.4 Efficiency Analysis

An ideal retrieval method for RAG systems should not only achieve strong performance—yielding accurate results—but also be efficient, minimizing latency, memory usage, and compute cost for real-world deployment. While direct measurement of GPU time or inference latency depends on hardware and implementation choices, token-level analysis offers a practical proxy for computational efficiency. We compare token usage across models by measuring: (1) the number of input tokens passed to the generator, and (2) the number of output tokens generated during the retrieval stage. As shown in Table 5, all SETR variants require substantially fewer input tokens than reranking-based methods. For instance, on MultiHopRAG, SETR-CoT & IRI feeds only 1,240 input tokens into the generator, compared to 2,672 in RankZephyr. Despite this sharp reduction, SETR not only maintains answer quality but often surpasses reranking models in F1 and accuracy. Interestingly, even within SETR variants, reasoning plays a role in efficiency. SETR-Selection only is the most efficient in terms of token usage, while SETR-CoT & IRI trades some marginal increase in prompt length for greater answer correctness and recall. This suggests a useful accuracy-efficiency trade-off spectrum that practitioners can tune based on resource constraints and latency budgets.

6 Discussion

Our work highlights the limitations of conventional top-k retrieval in Retrieval-Augmented Generation (RAG) systems and proposes a set-wise passage selection approach to better address the unique information needs of generative models. By incorporating information requirement identification and Chain-of-Thought reasoning for passage selection, SETR (Set-wise passage selection for Retrieval-Augmented Generation), enhances retrieval precision and improves end-to-end answer accuracy.

A promising direction for future work is enabling efficient selection from large candidate passage pools (e.g., 100 passages). Unlike traditional listwise reranking, which often relies on sequential sliding windows, our approach supports parallel, order-agnostic selection—reducing time complexity and improving scalability. Additionally, future work could explore how SETR iteratively refines queries for better retrieval and develops adaptive selection techniques to adjust the number of selected passages based on domain-specific needs.

Limitations

While our set-wise passage selection approach improves retrieval quality and efficiency in RAG systems, it has several limitations.

First, our method relies on a predefined retrieval pipeline, meaning its performance is still dependent on the initial retrieval stage. If the initial retrieved set lacks critical information, even an optimal set-wise selection cannot compensate for missing knowledge.

Second, our method optimizes retrieval for multi-hop and complex queries but has not yet been validated across diverse RAG domains such as code generation or conversational AI. While we anticipate broader applicability, further evaluation on diverse tasks is needed.

Lastly, our approach relies on the quality of the underlying language model for reasoning-based selection. While chain-of-thought reasoning improves passage selection, its effectiveness depends on the LLM's ability to accurately analyze and synthesize information.

References

Abdelrahman Abdallah, Jamshid Mozafari, Bhawna Piryani, Mohammed Ali, and Adam Jatowt. 2025. Rankify: A comprehensive python toolkit for retrieval, re-ranking, and retrieval-augmented generation. *arXiv preprint arXiv:2502.02464*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *Preprint*, arXiv:2310.11511.

Zijian Chen, Ronak Pradeep, and Jimmy Lin. 2024. An early first reproduction and improvements to single-token decoding for fast listwise reranking. *Preprint*, arXiv:2411.05508.

Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. 2025. Provence: efficient and robust context pruning for retrieval-augmented generation. *arXiv preprint arXiv:2501.16214*.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track. *Preprint*, arXiv:2102.07662.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the trec 2019 deep learning track. *Preprint*, arXiv:2003.07820.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. Don't hallucinate, abstain: Identifying llm knowledge gaps via multi-llm collaboration. *Preprint*, arXiv:2402.00367.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Giwon Hong, Jeonghwan Kim, Junmo Kang, Sung-Hyon Myaeng, and Joyce Jiyoung Whang. 2024. Why so gullible? enhancing the robustness of retrieval-augmented models against counterfactual noise. *Preprint*, arXiv:2305.01579.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *Preprint*, arXiv:2403.14403.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Tong Niu, Shafiq Joty, Ye Liu, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Judgerank: Leveraging large language models for reasoning-intensive reranking. *arXiv preprint arXiv:2411.00142*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023a. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *Preprint*, arXiv:2309.15088.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023b. Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze! *Preprint*, arXiv:2312.02724.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Preprint*, arXiv:2302.00083.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. *Preprint*, arXiv:2401.18059.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *Preprint*, arXiv:2305.15294.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. *Preprint*, arXiv:2302.00093.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2024. Is chatgpt good at search? investigating large language models as re-ranking agents. *Preprint*, arXiv:2304.09542.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *Preprint*, arXiv:2401.15391.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *Preprint*, arXiv:2212.10509.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment. *Preprint*, arXiv:2310.16944.

Hitesh Wadhwa, Rahul Seetharaman, Somyaa Aggarwal, Reshmi Ghosh, Samyadeep Basu, Soundararajan Srinivasan, Wenlong Zhao, Shreyas Chaudhari, and Ehsan Aghazadeh. 2024. From rags to rich parameters: Probing how language models utilize external knowledge over parametric information for factual queries. *Preprint*, arXiv:2406.12824.

Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, Anima Anandkumar, and Bryan Catanzaro. 2023. Shall we pretrain autoregressive language models with retrieval? a comprehensive study. *Preprint*, arXiv:2304.06762.

Ziting Wang, Haitao Yuan, Wei Dong, Gao Cong, and Feifei Li. 2024. Corag: A cost-constrained retrieval optimization system for retrieval-augmented generation. *Preprint*, arXiv:2411.00744.

Jinyang Wu, Feihu Che, Chuyuan Zhang, Jianhua Tao, Shuai Zhang, and Pengpeng Shao. 2024a. Pandora's box or aladdin's lamp: A comprehensive analysis revealing the role of rag noise in large language models. *Preprint*, arXiv:2408.13533.

Kevin Wu, Eric Wu, and James Zou. 2024b. Clasheval: Quantifying the tug-of-war between an llm's internal prior and external evidence. *Preprint*, arXiv:2404.10198.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Preprint*, arXiv:1809.09600.

Soyoung Yoon, Eunbi Choi, Jiyeon Kim, Hyeongu Yun, Yireun Kim, and Seung-won Hwang. 2024. Listt5: Listwise reranking with fusion-in-decoder improves zero-shot retrieval. *arXiv preprint arXiv:2402.15838*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. Rankt5: Fine-tuning t5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2308–2313.

A Appendix

A.1 Datasets

The experiments were conducted on the following four benchmark datasets:

- **HotpotQA** (Yang et al., 2018) is a large-scale multi-hop QA dataset with 113k question–answer pairs from Wikipedia. Each question requires reasoning over multiple documents, with sentence-level supporting facts provided. It includes diverse queries and comparison questions that test compositional reasoning and explainability, making it a strong benchmark for multi-hop retrieval systems.

- **2WikiMultiHopQA** (Ho et al., 2020) is a multi-hop QA dataset combining Wikipedia text with Wikidata triples to evaluate step-by-step reasoning. Each question includes an explicit reasoning path linking entities across documents. The dataset tests compositional reasoning and requires models to use both unstructured text and structured knowledge to answer questions.

- **MusiQue** (Trivedi et al., 2022) is a multi-hop QA benchmark designed to prevent reasoning shortcuts by requiring genuine multi-step reasoning. It includes around 25k questions composed from connected single-hop queries, each requiring 2 to 4 reasoning steps. The dataset emphasizes strong logical dependencies between steps and includes unanswerable variants to test robustness. MuSiQue is more challenging than earlier datasets and highlights significant performance gaps between humans and models.

- **MultiHopRAG** (Tang and Yang, 2024) is a recent benchmark for evaluating retrieval-augmented generation on complex multi-hop queries. Unlike the above QA datasets, MultiHopRAG is specifically built to evaluate end-to-end RAG systems. It includes 2,556 questions over a corpus of English news articles, with answers supported by evidence from 2 to 4 documents. The queries involve temporal and entity-based reasoning and require retrieving and synthesizing information across multiple sources. This makes the dataset well-suited for evaluating our set-wise passage selection approach.

A.2 Baselines

Following reranking models are considered as baselines:

- **bge-reranker-large** (Xiao et al., 2023) is a lightweight cross-encoder model from BAAI that scores query–passage pairs using full cross-attention, enabling more accurate relevance judgments than embedding-based models. Fine-tuned on large-scale data, it is commonly used to rescore top-k results in retrieval pipelines. As a strong open-source baseline, it reflects state-of-the-art conventional reranking focused on individual passage relevance.

- **RankLlama** (Ma et al., 2024) is a pointwise reranker based on LLaMA-2 7B model (Touvron et al., 2023). Given a query and a candidate passage, it outputs a score to reorder retrieved documents by their relevance. It demonstrates strong performance in both in-domain and zero-shot settings, serving as a competitive open-source baselines for passage reranking.

- **RankVicuna** (Pradeep et al., 2023a) is a 7B open-source listwise reranker built on the Vicuna 7B model (Zheng et al., 2023). It takes a query and a list of passages as input and outputs a ranked list of passage indices. Trained with GPT generated supervision, it achieves performance comparable to GPT-3.5 on benchmarks like TREC DL (Craswell et al., 2020, 2021), providing a transparent alternative to proprietary rerankers.

- **RankZephyr** (Pradeep et al., 2023b) is a zero-shot listwise reranker built on the Zephyr-7B model (Tunstall et al., 2023). Fine-tuned using GPT-4 generated ranking, it outputs ordered lists of passage indices given a query and candidate passages. It achieves performance close to GPT-4 and even surpassing it on some benchmarks. Its open-source nature and reproducibility make it a robust baseline for evaluating listwise reranking methods.

- **FirstMistral** (Chen et al., 2024) is a zero-shot listwise reranker based on Mistral 7B (Jiang et al., 2023). It reframes reranking as a single-token decoding task, enabling fast and efficient passage selection. Despite its simplicity, it achieves competitive performance

and serves as a strong open-source baseline to assess the raw ranking ability of modern instruction-tuned LLMs.

- **RankGPT4** (Sun et al., 2024) is a GPT-4-based reranker accessed via OpenAI's API, used in a zero-shot setting to rank passages given a query. It delivers state-of-the-art performance but is closed-source, non-reproducible, and costly. RankGPT4 serves as an upper-bound baseline to evaluate how well our approach performs against the strongest proprietary reranker.

A.3 Additional Experimental Details

Training. All SETR variants are fine-tuned using Llama-3.1-8B-Instruct as the base model. Training is conducted for 5 epochs using AdamW optimizer with a learning rate of 5×10^{-6} and an effective batch size of 512. We use 16×A100 GPUs and utilize Axolotl[6] framework, which integrates various efficiency-oriented training techniques. Each model is trained on 40k GPT-4o generated examples, where each input prompt includes a query and 20 retrieved passages.

Evaluation Metrics. We evaluate model performance using both retrieval and QA metrics. For retrieval, we report Mean Receprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG), Precision, and Recall, which collectively measure the quality of passage ranking and coverage of relevant information. For end-to-end QA, we adopt Exact Match (EM), F1 score, and Accuracy to quantify answer correctness and completeness.

Additionally, we evaluate *information coverage* (*IC@k*) on MultiHopRAG benchmark as it provides gold evidence lists, extractively collected from documents.

$$IC@k = \frac{|\bigcup_{i=1}^{k}\{e|e \in p_i\} \cap \{e|e \in \mathcal{E}_{\text{gold}}\}|}{|\mathcal{E}_{\text{gold}}|}$$

where p_i is the i-th top-ranked passage, e is an evidence span, $\mathcal{E}_{\text{gold}}$ is the complete set of annotated gold evidences.

For each question, we collect the complete set of gold evidence $\mathcal{E}_{\text{gold}}$ required to answer it, as annotated in the MultiHopRAG dataset. Given a top-k set of retrieved passages, we identify gold evidence spans e within the passage text using regular expression matching and remove duplicated evidences if redundant passage exist. *IC@k* is then computed as the proportion of gold evidence that appears within the top-k passages.

Generation. For answer generation, we utilize Rankify (Abdallah et al., 2025) toolkit to implement the full RAG pipeline, which consists of three components: Retrieval, Reranking/Selection, and Generation. For retrieval, we use the bge-large-en-v1.5 (Xiao et al., 2023) model to retrieve the top-20 passages per query. These candidates are then reranked or filtered via a set-wise selection mechanism. For generation, we employ Llama-3.1-8B-Instruct to generate final answers on general multi-hop QA benchmarks, including HotpotQA, 2WikiMultiHopQA, and MusiQUE. For MultiHopRAG benchmark, we follow the evaluation protocol of (Tang and Yang, 2024) and employ `gpt-4o-2024-08-06` as the generator due to its strong reasoning performance. Notably, even when provided with gold evidences, open-source models such as Llama2-70B and Mixtral-8x7B achieve relatively low accuracy (0.32 and 0.36, respectively), while GPT-4 attains a significantly higher score of 0.89, highlighting a substantial performance gap. The prompts used for each benchmark are shown in Figure 4 and Figure 5, respectively.

Prompt for General Multi-hop QA

```
{context}

Based on these texts, answer these questions:
Q: {question}
A:
```

Figure 4: Prompt template used for general multi-hop QA datasets including HotpotQA, 2WikiMultiHopQA, and MusiQue.

A.4 Prompt Templates

We provide the prompt template used in our experiment for both SETR-CoT and SETR-Selection only. Each prompt consists of a question and a set of passages, where each passage is assigned a unique numerical identifier. In SETR-CoT, the prompt concludes with a CoT reasoning, "Let's think step by step.", to encourage intermediate reasoning before producing the final selection. In contrast, SETR-Selection only removes this reasoning step and directly instructs the model to output only the final selection without any explanation. Figure 6 and 7 illustrate the two prompts, respectively.

[6] https://github.com/axolotl-ai-cloud/axolotl

Prompt for MultiHopRAG

```
Below is a question followed by some context from different sources. Please answer the question based on the context. The answer to the question is a word or entity. If the provided information is insufficient to answer the question, respond 'Insufficient Information'. Answer directly without explanation.

Question: {question}

Context:

{context}
```

Figure 5: Prompt template used for MultiHopRAG.

Prompt for SETR-CoT

```
I will provide you with {num} passages, each indicated by a numerical identifier []. Select the passages based on their relevance to the search query: {question}.

{context}

Search Query: {question}

Select the passages that mostly cover clear and diverse information to answer the query. Number of passages is unlimited.
The format of final output should be '### Final Selection: [] []', e.g., ### Final Selection: [2] [1].
Let's think step by step.
```

Figure 6: The set-wise passage selection prompt with basic Chain-of-Thought for SETR-CoT.

Prompt for SETR-Selection only

```
I will provide you with {num} passages, each indicated by a numerical identifier []. Select the passages based on their relevance to the search query: {question}.

{context}

Search Query: {question}

Select the passages that mostly cover clear and diverse information to answer the query. Number of passages is unlimited.
The format of final output should be '### Final Selection: [] []', e.g., ### Final Selection: [2] [1].
Only respond with the selection results, do not say any word or explain.
```

Figure 7: The set-wise passage selection prompt without CoT process for SETR-Selection only.