Multi-Source Knowledge Pruning for Retrieval-Augmented Generation: A Benchmark and Empirical Study

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) is increasingly recognized as an effective approach to mitigating the hallucination of large language models (LLMs) through the integration of external knowledge. While numerous efforts, most studies focus on a single type of external knowledge source. In contrast, most real-world applications involve diverse knowledge from various sources, a scenario that has been relatively underexplored. The main dilemma is the lack of a suitable dataset incorporating multiple knowledge sources and pre-exploration of the associated issues. To address these challenges, we standardize a benchmark dataset that combines structured and unstructured knowledge across diverse and complementary domains. Building upon the dataset, we identify the limitations of existing methods under such conditions. Therefore, we develop **PruningRAG**, a plug-and-play RAG framework that uses multi-granularity pruning strategies to more effectively incorporate relevant context and mitigate the negative impact of misleading information. Extensive experimental results demonstrate superior performance of PruningRAG and our insightful findings are also reported. Our dataset and code are publicly available[1].

1 Introduction

In recent years, the excellent performance of large language models (LLMs) (Brown, 2020; Jiang et al., 2023; Luo et al., 2023) in various tasks has attracted widespread attention from researchers. Nevertheless, since LLMs rely solely on internal knowledge acquired during training, they are often susceptible to hallucination (Zhou et al., 2020; Ji et al., 2023a,b; Mallen et al., 2022). To address this dilemma, retrieval-augmented generation (RAG) (Lewis et al., 2020; Guu et al., 2020; Cheng et al., 2024) integrates external knowledge, to bridge the gap between static, often limited internal knowledge of LLMs and vast real-world information, thereby reducing hallucinations.

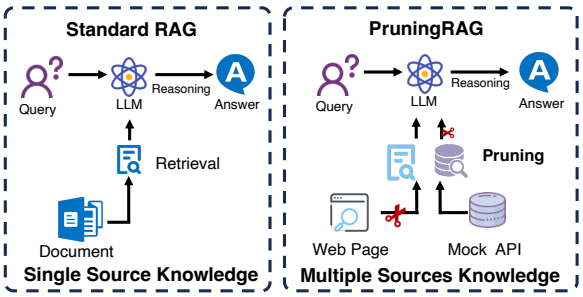

Figure 1: Comparison of standard RAG and PruningRAG. Standard RAG typically focuses on utilizing a single knowledge source, whereas PruningRAG optimizes the use of multiple external knowledge sources through multi-granularity pruning.

Numerous studies on RAG have been proposed to effectively integrate external knowledge source with the internal knowledge of LLMs (Gao et al., 2022; Chan et al., 2024; Su et al., 2024). Through a review of current research on RAG, we found that most studies primarily focus on the utilization of a single knowledge source. However, practical applications often require access to multiple knowledge sources, which can vary significantly in format, timeliness, and domain. Despite this need, research on RAG with multiple external knowledge sources remains limited, primarily due to the lack of suitable benchmark datasets and insufficient preliminary exploration of the current field.

Fortunately, we found that the KDD Cup 2024 CRAG competition dataset (Yang et al., 2024) comprises two distinct types of external knowledge sources: unstructured web pages of variable quality with limited timeliness but broad coverage, and APIs, which offer structured accurate information with strong real-time performance. However, the dataset still encounters some challenges in its suitability for broad research applications. For in-

[1]https://anonymous.4open.science/r/PruningRAG-BBAC

stance, the HTML-formatted web page data it contains present significant challenges for LLM processing, with no unified standards currently available for cleaning and parsing these data. Furthermore, how to effectively prune multi-source external knowledge and reduce misleading information has been less explored. In this paper, we standardize the dataset and establish a new benchmark, providing a solid foundation for future research in the field. To standardize this dataset, we undertake significant efforts. For example, we clean the web page knowledge by removing excessive HTML tags and converting it into an LLM-friendly Markdown format, enhancing data quality, ensuring compatibility with existing RAG frameworks, and enabling fair evaluation.

Building upon this dataset, we introduce PruningRAG, a new framework for RAG that performs multi-granularity pruning of diverse knowledge sources. Coarse-grained pruning effectively removes misleading information from inappropriate sources, thereby mitigating hallucinations. Meanwhile, adaptive fine-grained pruning further refines the knowledge retained from coarse-grained pruning, ensuring higher relevance while reducing extraneous information and thereby improving overall accuracy. After obtaining pruned knowledge, we use strategies such as in-context learning (ICL) to enhance the performance of reasoning. In addition, our framework is plug-and-play, facilitating further exploration and application.

Based on our dataset and framework, we conduct extensive experiments and report key insights. For coarse-grained pruning, the fine-tuned LLM proves adept at dynamically selecting relevant knowledge sources, maximizing utility while reducing misleading context. For fine-grained pruning, tailored strategies effectively handle various knowledge formats, further enhancing reliability. Notably, the relevance of the examples provided by in-context learning to the query also significantly influences the reasoning ability of RAG.

Main contributions of this paper are as follows: (1) We standardize a benchmark dataset that integrates structured and unstructured external knowledge across diverse domains. (2) We develop PruningRAG, a plug-and-play framework featuring multi-granularity pruning to optimize the integration of relevant context while mitigating misleading information. (3) We conduct extensive experiments and report our results and key insights to support future research in the RAG community.

2 Related Work

2.1 Retrieval-Augmented Generation

RAG (Lewis et al., 2020) has emerged as a strong approach to mitigating hallucinations in LLMs by incorporating external knowledge. Early methods utilized a straightforward "retrieve-then-generate" pipeline, whereas more advanced frameworks now integrate query refinement (Ma et al., 2023; Chan et al., 2024), iterative retrieval (Shao et al., 2023; Press et al., 2022) and modular architectures (Gao et al., 2023). Dynamic retrieval frameworks, such as Self-RAG (Asai et al., 2024) and DRAGIN (Su et al., 2024), progressively refine retrieved information, while GraphRAG (Edge et al., 2024) leverages graph-based indexes for structured knowledge.

However, these advancements generally overlook the complexities of managing multiple, diverse knowledge sources. Although some methods incorporate multiple sources of knowledge (Sarmah et al., 2024; Wang et al., 2024; Zhao et al., 2024), they often lack diversity in data formats, fields, or timeliness. To bridge this gap, we propose PruningRAG, which uses multi-granularity pruning to reduce misleading information, thereby consolidating multi-source knowledge.

2.2 Existing Benchmarks for RAG

As RAG frameworks evolve, new benchmarks have emerged to measure and guide their capabilities. For instance, RGB (Chen et al., 2023) evaluates robustness, integration, and counterfactual reasoning; CRUD-RAG (Chen et al., 2023) follows a structured Create-Read-Update-Delete workflow; RAG-Bench (Friel et al., 2024) focuses on explainability with detailed metrics; and RAGEval (Zhu et al., 2024) automates dataset generation for rigorous testing. These benchmarks offer a comprehensive framework for assessing RAG performance.

However, most existing benchmarks focus on single-source knowledge integration and do not evaluate the utilization of multi-source knowledge. Although the CRAG benchmark (Yang et al., 2024) incorporates both web pages and API sources, its dataset lacks standardized HTML parsing and hinders LLMs from effectively utilizing JSON-formatted API information. To address these limitations, we standardize the dataset and introduce a new benchmark to tackle multi-source heterogeneous knowledge integration, reduce hallucinations, and enhance reasoning capabilities.

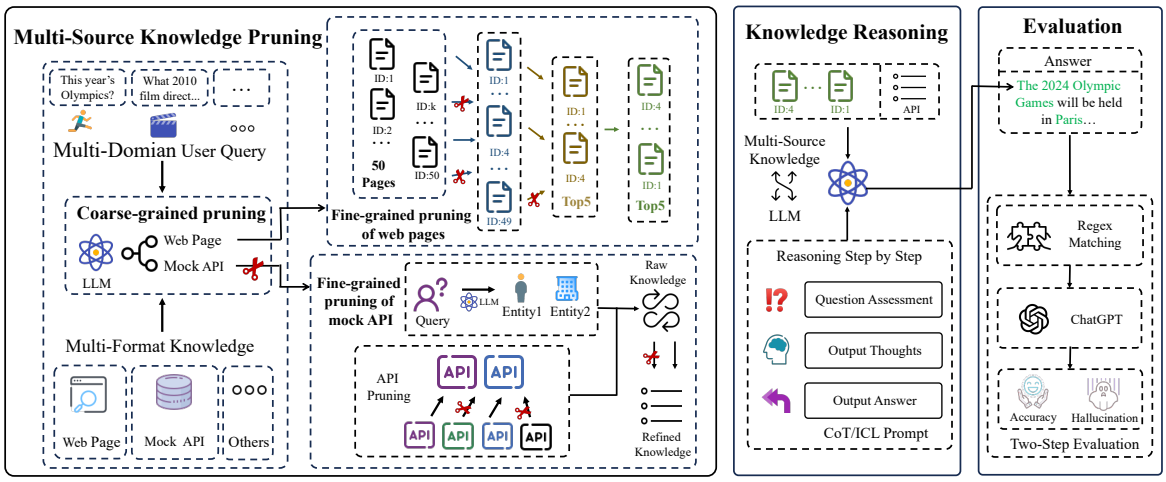

Figure 2: An illustration of PruningRAG, including multi-source knowledge pruning, knowledge reasoning and evaluation. Knowledge pruning filter out irrelevant knowledge sources and improve context relevance. The pruned knowledge is combined with the query to reason and the answer is evaluated based on accuracy and hallucination.

3 Preliminaries

In this section, we formally define the retrieval-augmented generation (RAG) involving multi-sources external knowledge and introduce how we propose a standardized dataset.

3.1 Problem Formulation

We consider a query q and a collection of knowledge sources $\mathcal{K} = \{K_1, K_2, \ldots, K_p\}$ with corresponding set of document $\mathcal{D} = \bigcup_{i=1}^{p} \mathcal{D}_{K_i}$. Each K_i is the i-th knowledge source, and $\mathcal{D}_{K_i} \subseteq K_i$. The goal is to generate an answer A by selecting and merging documents most relevant to q and then using LLM to produce the final response. Formally, let $\mathcal{D}_q$ be the subset of $\mathcal{D}$ retrieved for q. The final answer is given by:

$$\text{Answer} = \text{LLM}\big(\mathcal{D}_q \oplus q\big), \tag{1}$$

where $\oplus$ indicates the concatenation of $\mathcal{D}_q$ and q.

3.2 A Multi-Source Knowledge RAG Dataset

In recent years, numerous datasets have been introduced, each presenting its own challenges for RAG and contributing to its ongoing development (Kwiatkowski et al., 2019; Yang et al., 2018; Joshi et al., 2017; Stelmakh et al., 2022; Tang and Yang, 2024). However, most of these datasets rely on only a single external knowledge source, despite the fact that real-world applications often involve multiple sources. To address this gap, we leverage the dataset from the KDD Cup 2024 CRAG competition, which uniquely provides both unstructured web content (in HTML) and structured API data.

Although the original CRAG dataset offers diverse knowledge types, it also presents several practical hurdles—such as noisy HTML tags that complicate knowledgek extraction, the absence of a uniform parsing standard, and JSON-formatted API results that are not directly conducive to LLM-based reasoning. To overcome these limitations, we systematically refined and standardized the dataset. First, we clean the HTML knowledge and convert it into a Markdown format, eliminating extraneous tags and ensuring consistency. Second, we employed rule-based processing to simplify API access by aligning entities with query terms and transforming JSON outputs into natural language. These enhancements remove the initial shortcomings of the CRAG dataset, thereby creating a robust multi-source benchmark that more accurately supports RAG research and development.

4 Methodology

In this section, we first present an overview of the entire framework, followed by the details of each component. The illustration of the PruningRAG framework is shown in Figure 2.

4.1 Overview of the PruningRAG Framework

PruningRAG consists of three components: multi-source knowledge pruning, knowledge reasoning and performance evaluation. First, coarse-grained pruning screen out irrelevant or low-quality knowledge sources, thus narrowing the overall search space. Subsequently, fine-grained pruning refines

the remaining sources. This two-tier pruning ensures that only relevant context remains.

Once pruned, the selected contexts are combined with the user query in a knowledge-enhanced prompt. This prompt leverages Chain-of-Thought (CoT) reasoning and In-Context Learning (ICL) to guide the LLM toward producing fact-based and minimally hallucinatory outputs. Finally, the framework employs a rigorous evaluation process, including metrics for accuracy, hallucination rate, missing rate, and an overall score. These metrics use both string-matching and GPT-based assessments to gauge how well the system retrieves essential information and avoids misleading content, ultimately ensuring that PruningRAG delivers reliable and contextually precise responses.

4.2 Multi-Source Knowledge Pruning

In this section, we explain the specific strategies for pruning multi-source knowledge, including coarse-grained pruning to filter knowledge sources and fine-grained pruning to obtain key context.

4.2.1 Coarse-Grained Knowledge Pruning

In scenarios involving multiple sources of external knowledge, relevant information may reside in $\mathcal{K} = \{K_1, K_2, \ldots, K_p\}$, within internal knowledge of LLM K_0, or be completely unavailable. This makes pruning of irrelevant sources essential to prevent conflicts and hallucinations. We define a subset-selection function:

$$\mathcal{K}_q = \Theta\big(\{K_i\}_{i=0}^{p}, q\big), \tag{2}$$

where Θ identifies the knowledge sources most appropriate for query q. An LLM is used to determine $\mathcal{K}_q$. However, initial experiments[2] showed that prompting the LLM with only q was insufficient for accurate source selection.

To address this, we built a specialized dataset from the training set, labeling each query with the subset of sources that provided correct answers in practice. For queries that none of the available sources could answer accurately, we included sources exhibiting the highest overall accuracy on similar queries. We then fine-tuned the LLM on these (query, source-subset) pairs to discard low-relevance sources while retaining those critical for generating a correct answer.

[2]The experimental results are detailed in Appendix E.

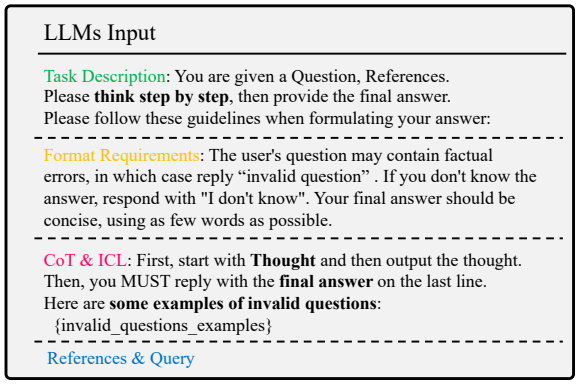

Figure 3: Prompt design template incorporating CoT and ICL for enhanced reasoning.

4.2.2 Fine-Grained Knowledge Pruning

When integrating diverse external knowledge sources, fine-grained pruning is crucial to extract relevant information. Given a corpus $\mathcal{D}$ comprising multiple documents (e.g., fifty web pages), we first perform a sparse retrieval step using BM25 (Cheng et al., 2021), formally defined as:

$$\mathcal{D}_q^{\text{sparse}} = \text{BM25}(\mathcal{D}, q), \tag{3}$$

where $\mathcal{D}_q^{\text{sparse}} \subseteq \mathcal{D}$ denotes the subset of documents identified as relevant to the query q by BM25. Subsequently, we refine the context through dense passage retrieval (DPR) (Karpukhin et al., 2020), which selects text chunks with high relevance to q:

$$\mathcal{D}_q^{\text{dense}} = \text{DPR}\big(\mathcal{D}_q^{\text{sparse}}, q\big), \tag{4}$$

where $\mathcal{D}_q^{\text{dense}}$ is the final context from web pages. In scenarios where only a limited number of documents (e.g., five web pages) are available, we bypass the sparse retrieval step and directly apply dense retrieval for more precise chunk selection.

For knowledge from APIs, fine-grained pruning enhances context quality by filtering redundant APIs and irrelevant parts of the retrieved information. To achieve this, named entity recognition (NER) is employed to extract key entities from the query, guiding the API to focus its responses on key information. Furthermore, queries are directed to specific APIs based on their characteristics, allowing for the exclusion of irrelevant APIs and reducing unnecessary data retrieval. The structured API output is then transformed into natural language using rule-based post-processing, ensuring that the refined information is seamlessly integrated into LLM's response generation.

	Naive RAG		HyDE		Iter-RETGEN		RRR		Self-Ask		Self-RAG	
Knowledge Source	Acc.	Hall.	Acc.	Hall.	Acc.	Hall.	Acc.	Hall.	Acc.	Hall.	Acc.	Hall.
5 Web Pages	24.80	22.69	24.65	18.24	28.59	18.45	15.61	12.77	24.46	29.98	14.29	8.97
50 Web Pages	32.82	29.54	30.48	17.06	31.50	25.16	19.76	15.31	30.78	26.26	14.51	8.75
API	32.38	11.67	32.75	11.46	32.38	11.67	31.66	12.48	25.23	17.80	14.36	8.82
5 Web Pages + API	39.97	22.32	40.41	21.23	43.69	18.52	35.08	15.32	29.83	32.89	14.00	9.19
+ Pruning	**44.56**	**21.23**	**43.03**	**18.31**	**43.83**	**17.21**	**36.03**	**13.42**	**32.68**	**28.37**	**14.44**	**8.75**
↑ Gain	11.48%	4.88%	6.48%	13.75%	0.32%	7.71%	2.71%	14.16%	9.55%	13.74%	3.14%	4.79%
50 Web Pages + API	39.53	22.76	40.40	21.22	**43.69**	18.53	34.64	15.75	33.62	35.30	12.54	10.72
+ Pruning	**43.15**	**18.01**	**41.76**	**15.97**	41.46	**15.89**	**35.77**	**14.58**	**33.84**	**32.45**	**13.03**	**9.34**
↑ Gain	9.16%	20.87%	3.37%	24.74%	-5.10%	14.24%	2.26%	7.43%	0.65%	8.07%	3.91%	12.87%

Table 1: Performance of RAG frameworks with and without PruningRAG on different knowledge sources.

4.3 Knowledge-Enhanced Reasoning

As shown in Figure 3, we design a prompt that integrates CoT and ICL to optimize the use of pruned knowledge for reasoning. The prompt begins by clearly stating the task and instructing the model to answer based on the context. If uncertain, the model is directed to output "I don't know" to prevent hallucinations. To enhance reasoning capabilities, we include a few-shot example section, where examples are chosen from domains different from domain of the query to reduce overfitting to domain-specific patterns. The pruned knowledge and the query are then presented with a CoT instruction, prompting the model to reason step by step. Finally, the prompt asks the model to output both its reasoning and a final, well-considered answer, ensuring clarity and accuracy.

4.4 Performance Evaluation

We use four key metrics to evaluate the performance of the RAG framework: accuracy (Acc.), hallucination (Hall.), missing (Miss.), and an overall score, which is defined as the difference between accuracy and hallucination. This score reflects the framework's ability to extract key knowledge while avoiding misleading information. The evaluation process combines string matching and GPT-based assessments. First, if the predicted answer exactly matches the ground truth, it is recorded as accurate; if the response is "I don't know," it is categorized as missing information. For non-exact matches, GPT-3.5 Turbo (Ouyang et al., 2022) semantically compares the prediction with the ground truth, marking it as accurate if aligned or as hallucination if not.

5 Benchmark Evaluation of RAG

In this section, we introduce the evaluation of PruningRAG and various baselines in different knowledge sources using the standardized dataset, including experimental setup and analysis of the results.

5.1 Experimental Setup

Implementation. In our experiments, for coarse-grained pruning, we use a fine-tuned Llama-3.1-8B-Instruct (Dubey et al., 2024) to filter out inappropriate knowledge sources. For the fine-grained stage, we deploy the BGE-M3 (Chen et al., 2024). If not specified otherwise, we use Llama-3.1-8B-Instruct as the generator. Detailed hyperparameter configurations are provided in Appendix A.2.

Baselines. We apply PruningRAG to the following RAG frameworks: naive RAG, and five state-of-the-art train-free frameworks, including HyDE (Gao et al., 2022), Iter-RETGEN (Shao et al., 2023), RRR (Ma et al., 2023), and Self-Ask (Press et al., 2022), as well as Self-RAG (Asai et al., 2024), which requires fine-tuning. All methods share the same dataset post-processing and evaluation protocols to ensure the robustness of the pruning strategy's performance gains across different approaches.

5.2 Experimental Results

Table 1 compares six RAG frameworks across three knowledge sources: 5 web pages, 50 web pages (50 web pages contain 5 web pages) and API and examines the impact of applying our PruningRAG on performance, measured in terms of accuracy (Acc.) and hallucination rate (Hall.). From the results we have the following findings: First, for the multi-knowledge source scenario, PruningRAG improves the performance of almost all RAG frameworks, improving accuracy and reducing hallucinations. Second, compared with improving accuracy, the PruningRAG framework has a more obvious effect on reducing hallucinations, which indicates that in

Naive RAG	Llama-3.1-70B-Inst.		Llama-3.1-8B-Inst.		Llama-3.2-3B-Inst.		Llama-3.2-1B-Inst.	
Knowledge Source	Acc.	Hall.	Acc.	Hall.	Acc.	Hall.	Acc.	Hall.
5 Web Pages	30.49	15.31	24.80	22.69	29.68	23.63	12.83	16.33
50 Web Pages	33.99	21.01	32.82	29.54	31.50	25.74	13.05	16.11
API	34.28	5.83	32.38	11.67	28.74	11.23	4.23	3.79
5 Web Pages + API	47.84	18.16	39.97	22.32	36.90	25.02	**16.19**	16.92
+ Pruning	**48.14**	**16.49**	**44.56**	**21.23**	**37.41**	**20.49**	14.29	**15.17**
↑ Gain	6.27%	9.20%	11.48%	4.88%	1.38%	18.11%	-11.73%	10.34%
50 Web Pages + API	47.99	19.54	39.53	22.76	38.41	24.00	**15.17**	19.62
+ Pruning	**52.58**	**17.87**	**43.15**	**18.01**	**38.88**	**21.73**	14.08	**15.10**
↑ Gain	9.56%	8.55%	9.16%	20.87%	1.22%	9.46%	-7.18%	20.01%

Table 2: Performance comparison of LLMs with varying parameter scales with and without PruningRAG.

the case of insufficient knowledge, the pruned information induces LLM to make incorrect answers. We observe that PruningRAG slightly reduces accuracy in Iter-RETGEN, likely due to the removal of some effective information after multiple retrieval rounds. However, it still reduces hallucinations.

Table 2 presents a comparison of the performance of LLMs with varying parameter scales, evaluated within the naive RAG framework. It also discusses the performance enhancements achieved through PruningRAG. The experimental results demonstrate a general trend wherein model performance improves progressively with increasing model size, highlighting the ability of larger models to better leverage external knowledge sources due to their enhanced expressive power. LLMs of varying sizes show improved accuracy and reduced hallucinations with PruningRAG, with the only exception being an accuracy drop in Llama-3.2-1B-Instruct, highlighting the robustness and effectiveness of our method. PruningRAG may have removed knowledge that, though redundant for larger models, remained essential for the smaller 1B model. However, due to PruningRAG's effective removal of detrimental information, the 1B model can also significantly reduce hallucinations.

6 Extensive Empirical Studies

In this section, we leverage PruningRAG to conduct further experimental exploration on our dataset and present key insights from three perspectives: coarse-grained pruning, fine-grained pruning, and knowledge-enhanced reasoning.

6.1 Impact of Coarse-Grained Pruning

Table 3 presents an evaluation of four knowledge utilization strategies. One approach relies exclusively on either the LLM's internal knowledge or external knowledge. Another combines the LLM's internal knowledge with one or more external sources to generate responses collaboratively. A further strategy prioritizes internal knowledge, consulting external sources only when the internal context is insufficient to produce an answer. Finally, our proposed method incorporates a knowledge source pruning mechanism to optimize the selection and integration of relevant knowledge.

The experimental results indicate that directly relying on multiple knowledge sources simultaneously often introduces conflicting information, resulting in performance degradation compared to

Experiment Setting	**Acc.**	**Score**
LLM	17.94	-0.36
Web Pages	24.80	2.11
API	32.38	20.71
Web Pages+API	39.97	17.65
LLM+Web Pages	17.94	7.80
LLM+API	40.55	22.25
LLM+Web Pages+API	**45.73**	14.37
LLM→ Web Pages	25.30	-5.84
LLM→ API	35.01	11.31
LLM→ Web Pages+API	38.22	6.64
Knowledge Source Pruning	44.56	**23.33**

Table 3: Comparison of performance of different strategies for leveraging knowledge sources.

Setting	**Acc.**	**Hall.**	**Latency(s)**
w/o Sparse Retrieval	**32.93**	30.18	**3.29**
w. Sparse Retrieval	32.82	**29.54**	33.54

Table 4: Comparison of effectiveness and efficiency with and without broad retrieval.

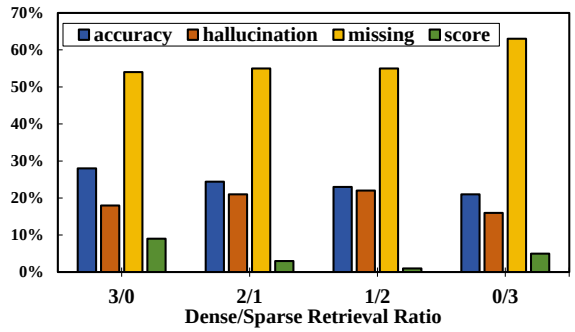

Figure 4: Performance comparison of sparse, dense and hybrid retrieval.

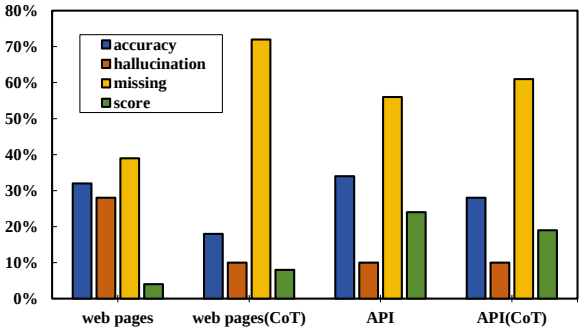

Figure 5: Impact of CoT across knowledge sources.

using a single source. Additionally, prioritizing the internal knowledge of LLM before retrieval tends to generate hallucinations due to the inherent inaccuracies in the model's internal knowledge. In contrast, our knowledge source pruning strategy dynamically prunes knowledge sources based on the characteristics of each query, enabling the effective utilization of each knowledge source.

6.2 Impact of Fine-Grained Pruning

Table 4 compares the performance of the PruningRAG with and without the initial broad retrieval step in the fine-grained pruning process. The results highlight that incorporating the sparse retrieval stage significantly improves system efficiency by reducing latency, particularly in cases involving large volumes of external knowledge. Serving as an initial filter, sparse retrieval narrows the search scope, allowing the subsequent dense retrieval to operate with higher precision and speed. This multi-stage fine-grained pruning approach reduces latency while ensuring context relevance.

Figure 4 demonstrates that dense retrieval outperforms sparse retrieval. Specifically, dense retrieval is more effective at capturing semantic relationships compared to sparse retrieval. When dense and sparse retrieval methods are combined, the accuracy improves relative to sparse retrieval alone. However, this hybrid approach also leads to an increase in hallucinations. This suggests that while the hybrid retrieval retains important information, it struggles to effectively prune misleading context (Cheng et al., 2022; Gu et al., 2018).

Category	N	Acc.	Hall.	Miss.	Score
Overall	0	13.20	**10.50**	76.29	2.70
	1	16.05	12.62	71.33	3.43
	2	16.12	12.98	70.90	3.14
	3	15.17	12.69	72.14	2.48
	1*	16.12	11.89	71.99	4.23
	2*	**18.02**	11.23	70.75	**6.78**
	3*	16.41	11.60	72.00	4.81
False Premise	0	**25.00**	**5.77**	69.23	**19.23**
	1	16.03	14.10	69.87	1.93
	2	16.57	13.46	69.87	3.11
	3	17.31	12.82	69.87	4.49
	1*	20.51	12.18	67.31	8.33
	2*	19.87	11.54	68.59	6.33
	3*	23.08	9.62	67.30	13.46

Table 5: Impact of few-shot learning on LLM reasoning. N^* indicates that the N examples provided for in-context learning are cross-domain examples.

6.3 Analysis of Knowledge Reasoning

In this section, we analyze the impact of our strategies for enhancing LLM reasoning over pruned knowledge, including CoT reasoning and ICL.

Role of CoT reasoning. Figure 5 illustrates the varying impact of incorporating chain-of-thought (CoT) reasoning (Wei et al., 2022; Trivedi et al., 2022) on performance, depending on the type of external knowledge sources. When internal LLM knowledge is combined with unstructured network data, which is often noisy and sparsely populated with relevant information, CoT's step-by-step reasoning helps filter out irrelevant details and reduce hallucinations, thereby improving response accuracy. In contrast, when an API is used as an external knowledge source, CoT's multi-step process can lead to overly cautious responses. While this cautious approach reduces hallucinations, it may significantly compromise accuracy, even when the API provides reliable information.

Impact of ICL with cross-domain examples. Table 5 illustrates the impact of incorporating false premise examples on performance of PruningRAG. False premise questions, which include intentional

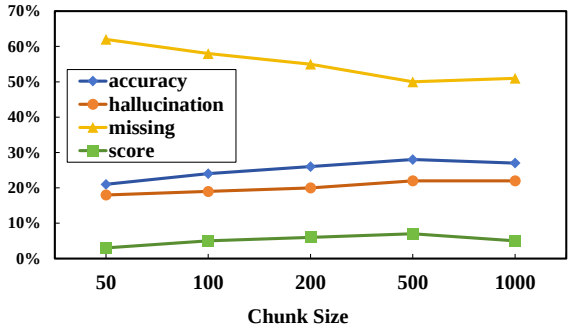

Figure 6: Performance of PruningRAG under different values of chunk size.

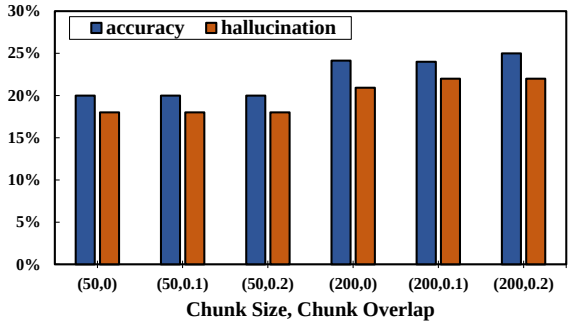

Figure 7: Impact of chunk size and chunk overlap on PruningRAG performance.

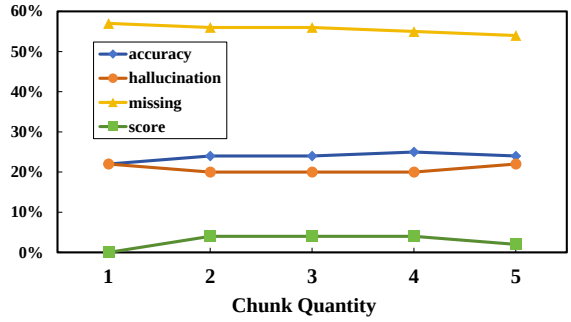

Figure 8: Impact of the number of retrieved chunks on PruningRAG performance.

inaccuracies requiring LLM to respond with "invalid question," were used to assess the ability to identify flawed queries. To aid in this, the model was provided with sample invalid questions and explanations, in two conditions: one with domain-aligned examples and another with cross-domain examples. Our findings reveal that few-shot examples enhance the general performance of the RAG by improving task comprehension and reasoning capabilities (Dong et al., 2022). However, accuracy on false premise questions declines compared to the zero-shot setting, with domain-specific examples performing worse than cross-domain examples. This discrepancy may stem from overfitting to domain-specific patterns, while cross-domain examples introduce greater variability, mitigating overfitting and enhancing reasoning ability.

6.4 Hyperparameter Sensitivity Analysis

In this section, we analyze the impact of hyperparameters such as chunk size, overlap, and the number of retrieved chunks on RAG performance.

Chunk Size. Figure 6 illustrates how chunk size affects PruningRAG. Increasing chunk size from 50 to 500 enhances accuracy by providing richer context, but slightly raises hallucination rates as LLM must filter knowledge from more noise. Once chunk size reaches 1000, accuracy drops because excessive content dilutes relevance and hinders the identification of key details. Thus, a moderate chunk size achieves the best balance between context richness and focus.

Chunk Overlap. As shown in Figure 7, overlap notably affects performance, particularly with larger chunk sizes. For small chunks (e.g., size 50), overlap offers minimal gains given limited context. Conversely, for larger chunks (e.g., size 200), overlap enhances continuity and accuracy but slightly raises hallucination due to redundancy.

Chunk Quantities. As shown in Figure 8, increasing the number of retrieved chunks initially boosts accuracy but eventually plateaus and then declines, while hallucination rates gradually rise. In contrast to too large chunks, which dilute focus and reduce accuracy, providing too many retrieved chunks primarily increases hallucination by introducing excessive context.

7 Conclusion

This paper presents a standardized multi-source knowledge dataset and introduces the PruningRAG framework, which leverages multi-granular pruning to optimize the utilization of diverse knowledge sources. Through our framework, we uncover valuable insights, including the impact of knowledge source pruning and the effectiveness of adaptive fine-grained pruning. Furthermore, we have made our dataset, the PruningRAG framework, code, and experimental results publicly available. We hope that our work will inspire further research into advanced knowledge pruning to better tackle the complexities of multi-source knowledge, contributing to the progress of the RAG community.

Limitations

In this work, we standardized a benchmark dataset containing multiple knowledge sources and proposed a novel plug-and-play framework to improve the current RAG approach. However, because the external knowledge in our dataset is not annotated, we were unable to evaluate retrieval quality directly and instead focused on end-to-end performance. In addition, due to limited computing resources, we did not train the 70B model to test the performance of Self-RAG. As part of future work, we hope to perform more detailed annotation work on the dataset to conduct a more comprehensive evaluation, and also try to explore the performance of PruningRAG with more knowledge sources.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yi-Ting Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *ArXiv*, abs/2404.00610.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 2318–2335.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Benchmarking large language models in retrieval-augmented generation. In *AAAI Conference on Artificial Intelligence*.

Mingyue Cheng, Qi Liu, Wenyu Zhang, Zhiding Liu, Hongke Zhao, and Enhong Chen. 2024. A general tail item representation enhancement framework for sequential recommendation. *Frontiers of Computer Science*, 18(6):186333.

Mingyue Cheng, Zhiding Liu, Qi Liu, Shenyang Ge, and Enhong Chen. 2022. Towards automatic discovering of deep hybrid network architecture for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 1923–1932.

Mingyue Cheng, Fajie Yuan, Qi Liu, Xin Xin, and Enhong Chen. 2021. Learning transferable user representations with sequential behaviors via contrastive pre-training. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 51–60. IEEE.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. 2024. Ragbench: Explainable benchmark for retrieval-augmented generation systems. *ArXiv*, abs/2407.11005.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

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.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023a. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023b. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Junzhe Jiang, Shang Qu, Mingyue Cheng, Qi Liu, Zhiding Liu, Hao Zhang, Rujiao Zhang, Kai Zhang, Rui Li, Jiatong Li, et al. 2023. Reformulating sequential recommendation: Learning dynamic user interest with content-enriched language modeling. *arXiv preprint arXiv:2309.10435*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

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*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yucong Luo, Mingyue Cheng, Hao Zhang, Junyu Lu, Qi Liu, and Enhong Chen. 2023. Unlocking the potential of large language models for explainable recommendations. *arXiv preprint arXiv:2312.15661*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *arXiv preprint arXiv:2305.14283*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. 2024. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 608–616.

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. *arXiv preprint arXiv:2305.15294*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. *arXiv preprint arXiv:2204.06092*.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. *arXiv preprint arXiv:2403.10081*.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Hongru Wang, Wenyu Huang, Yang Deng, Rui Wang, Zezhong Wang, Yufei Wang, Fei Mi, Jeff Z Pan, and Kam-Fai Wong. 2024. Unims-rag: A unified multi-source retrieval-augmented generation for personalized dialogue systems. *arXiv preprint arXiv:2401.13256*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, et al. 2024. Crag–comprehensive rag benchmark. *arXiv preprint arXiv:2406.04744*.

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. *arXiv preprint arXiv:1809.09600*.

Qingfei Zhao, Ruobing Wang, Xin Wang, Daren Zha, and Nan Mu. 2024. Towards multi-source retrieval-augmented generation via synergizing reasoning and preference-driven retrieval. *arXiv preprint arXiv:2411.00689*.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. 2020. Detecting hallucinated content in conditional neural sequence generation. *arXiv preprint arXiv:2011.02593*.

Kunlun Zhu, Yifan Luo, Dingling Xu, Ruobing Wang, Shi Yu, Shuo Wang, Yukun Yan, Zhenghao Liu, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Rageval: Scenario specific rag evaluation dataset generation framework. *Preprint*, arXiv:2408.01262.

A Reproducibility

A.1 Dataset Processing

In our experiments, we used the official training set provided by the KDD Cup 2024 CRAG competition dataset to construct fine-tuning data for

coarse-grained pruning, and used the validation set to obtain our experimental results. To enhance the usability of the web-based knowledge within the dataset, we converted HTML-formatted web pages into markdown format using the Jina framework. This conversion was essential to improve the compatibility of the data with Large Language Models (LLMs), enabling more effective inference and retrieval of relevant information.

This processing step was crucial for ensuring that the external knowledge sources were optimally formatted for existing retrieval-augmented generation (RAG) framework. The parsed markdown dataset, which is publicly available, supports further research and underscores the practical improvements brought by our approach in handling complex question-answer (QA) scenarios.

A.2 Experimental Setup

To ensure the reproducibility and consistency of our experiments, we establish a base configuration for our PruningRAG, detailed in Table 6. For the coarse-grained pruning, we use a fine-tuned Llama 3.1 8B to filter out inappropriate knowledge sources . For the fine-grained stage we deployed the BGE M3 embedding model. The chunk size for retrieval is set to 200 tokens with no overlap, and the TopK retrieved chunks per query is set to 3. For reasoning, we use Llama 3.1 8B as the backbone model. The generation parameters include a maximum of 500 new tokens per output. We set the temperature to 0, ensuring deterministic outputs, and use a TopP value of 1.0.

Hyperparameter	Value
Chunk Size	200 tokens
Chunk Overlap	0 (no overlap)
Embedding Model	BGE-M3
Temperature	0 (deterministic)
TopP	1.0 (all tokens considered)
LLM Backbone	Llama-3.1-8B-Instruct

Table 6: Base hyperparameter configuration.

A.3 Configuration details for RAG methods

In this section, we provide the detailed configuration of the baseline RAG methods used in our experiments. All algorithm parameters are set to the optimal values reported in their respective original papers to ensure a fair comparison and optimal performance.

Method	Parameter	Value
Self-RAG	beam_width	2
	max_depth	7
	w_rel	1.0
	w_sup	1.0
	w_use	0.5
	threshold	0.2
Iter-RETGEN	max_iteration	3
Self Ask	max_iteration	5

Table 7: Configuration details for RAG methods.

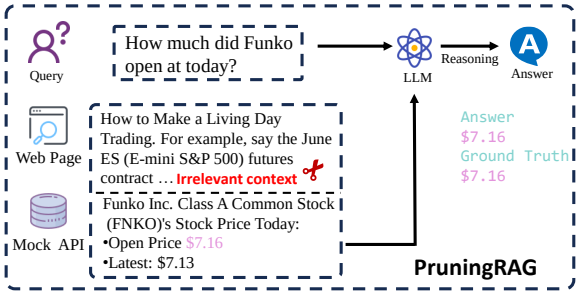

Figure 9: Case Study of PruningRAG.

B Computing Infrastructure

All the experiments are conducted on 2 × Nvidia GeForce RTX 4090 GPUs (24GB memory each). Other configuration includes 2 × Intel Xeon Gold 6426Y CPUs, 503GB DDR4 RAM, and 1 × 893.8GB SATA SSD, which is sufficient for all the baselines. However, due to the limited computational resources, we are unable to locally deploy Llama-3.1-70B-Instruct. Therefore, for experiments involving this model, we utilize an API for execution. On average, each inference task takes approximately one hour to complete. Additionally, training the Llama-3.1-8B-Instruct model on the Self-RAG dataset requires around 50 hours.

C Dataset Details

Our dataset comprises 4,409 QA pairs, with queries covering a wide range of domains (e.g., finance, sports) and temporal categories (e.g., real-time, static), across eight distinct question types (e.g., simple, conditional, multi-hop), split into training, validation, and test sets, with 1,371 QA pairs reserved for testing and the remainder allocated to training and validation. This design facilitates a comprehensive evaluation of RAG systems, setting it apart from specialized datasets , which predominantly focus on multi-hop questions. Each QA pair in our dataset is paired with either five or fifty

unstructured web pages, along with a API providing structured access to knowledge from a knowledge graph containing 2.6 million entities. The knowledge from web pages is generally static and broad in scope, making it well-suited for answering static queries in open domains. In contrast, the knowledge accessed via the API is more real-time and domain-specific, which is particularly effective for addressing time-sensitive queries in areas like finance. Additionally, some queries may not align well with either external knowledge source, in which case the model must rely on its internal knowledge base. Our dataset incorporates multiple external knowledge sources, a feature that distinguishes it from many existing datasets, which typically rely on a single knowledge source, with answers directly extracted from that source. The external knowledge in our dataset, does not always guarantee the presence of relevant information to answer the queries. A further challenge arises when inappropriate knowledge sources are selected, as this can introduce misleading information, exacerbating hallucination issues.

D Case Study

Figure 9 illustrates a case study of the PruningRAG framework applied to answer the query: "How much did Funko open at today?" The system processes two external knowledge sources: a web page and a API. The web page contains irrelevant context, such as information about trading strategies and futures contracts, which is pruned during the knowledge refinement stage. The API provides structured and accurate information, including the open price of Funko Inc.'s stock at $7.16 and the latest price at $7.13. After pruning irrelevant knowledge, the refined information is passed to the LLM reasoning component, which generates the answer. In this example, the answer "$7.16" matches the ground truth, demonstrating the effectiveness of PruningRAG in filtering irrelevant context and focusing on relevant knowledge to improve response accuracy.

E More Experimental Results

Effectiveness of knowledge source pruning with fine-tuned LLM. As shown in Table 8, the overall performance of the knowledge source pruning approach with a fine-tuned large model is evaluated and compared to the performance achieved without fine-tuning. The results demonstrate that when pruning is performed using a large model without fine-tuning, the performance actually worsens, highlighting the importance of fine-tuning in enhancing the model's effectiveness. It appears that the knowledge source pruning process benefits from the model's ability to adapt to the specific task or domain through fine-tuning, as it allows for more accurate and relevant information retention. In contrast, the unrefined model struggles to effectively discard irrelevant knowledge, leading to a reduction in accuracy and an increase in hallucinations. Only when fine-tuned models are used for knowledge source pruning can we achieve significant improvements in both accuracy and hallucination reduction, showcasing the value of task-specific adaptation in our approach.

	Naive RAG		HyDE	
	Acc.	Hall.	Acc.	Hall.
5 web pages + API				
w/o Pruning	39.97	22.32	40.41	21.23
Pruning w/o fine-tuning	38.37	23.85	40.04	21.15
Pruning w/ fine-tuning	**44.56**	**21.23**	**43.03**	**18.31**
5 web pages + API				
w/o Pruning	39.53	22.76	40.40	21.22
Pruning w/o fine-tuning	39.75	28.08	39.23	23.34
Pruning w/ fine-tuning	**43.15**	**18.01**	**41.76**	**15.97**

Table 8: Performance of PruningRAG frameworks with and without fine-tuning.