MAIN-RAG: Multi-Agent Filtering Retrieval-Augmented Generation

Chia-Yuan Chang[1*] **Zhimeng Jiang**[2] **Vineeth Rakesh**[2] **Menghai Pan**[2] **Chin-Chia Michael Yeh**[2]
Guanchu Wang[3] **Mingzhi Hu**[4] **Zhichao Xu**[5] **Yan Zheng**[2] **Mahashweta Das**[2] **Na Zou**[6]

[1]**Texas A&M University** [2]**Visa Research** [3]**Rice University** [4]**Worcester Polytechnic Institute** [5]**University of Utah**
[6]**University of Houston**

cychang@tamu.edu, {zhimjian,vinemoha,mengpan,miyeh,yazheng,mahadas}@visa.com,
gw22@rice.edu, mhu3@wpi.edu, zhichao.xu@utah.edu, nzou2@central.uh.edu

Abstract

Large Language Models (LLMs) are becoming essential tools for various natural language processing tasks but often suffer from generating outdated or incorrect information. Retrieval-Augmented Generation (RAG) addresses this issue by incorporating external, real-time information retrieval to ground LLM responses. However, the existing RAG systems frequently struggle with the quality of retrieval documents, as irrelevant or noisy documents degrade performance, increase computational overhead, and undermine response reliability. To tackle this problem, we propose Multi-Agent Filtering Retrieval-Augmented Generation (MAIN-RAG), a training-free RAG framework that leverages multiple LLM agents to collaboratively filter and score retrieved documents. Specifically, MAIN-RAG introduces an adaptive filtering mechanism that dynamically adjusts the relevance filtering threshold based on score distributions, effectively minimizing noise while maintaining high recall of relevant documents. The proposed approach leverages inter-agent consensus to ensure robust document selection without requiring additional training data or fine-tuning. Experimental results across four QA benchmarks demonstrate that MAIN-RAG consistently outperforms traditional RAG approaches, achieving a 2–11% improvement in answer accuracy while reducing the number of irrelevant retrieved documents. Quantitative analysis further reveals that our approach achieves superior response consistency and answer accuracy over baseline methods, offering a competitive and practical alternative to training-based solutions.

1 Introduction

Large Language Models (LLMs) have revolutionized natural language processing (NLP), achieving state-of-the-art performance across diverse tasks, such as question answering, summarization, and text generation (Vaswani, 2017; Brown, 2020). However, their reliance on pre-trained static data introduces critical limitations, including the generation of outdated or factually incorrect information—a phenomenon referred to as *hallucination* (Ji et al., 2023; Zhang et al., 2023). This issue becomes particularly pronounced in applications requiring accurate, up-to-date, and contextually relevant responses, such as healthcare, legal systems, and real-time customer support (Bommasani et al., 2021; Zellers et al., 2019; Lin et al., 2022).

Retrieval-augmented generation (RAG) has emerged as a promising solution to mitigate these challenges by integrating real-time document retrieval to ground LLM outputs in reliable external knowledge (Lewis et al., 2020; Guu et al., 2020; Karpukhin et al., 2020; Ram et al., 2023; Li et al., 2023; Wang et al., 2023). Training-based methods (Guu et al., 2020; Karpukhin et al., 2020; Wang et al., 2023) have demonstrated strong performance but require substantial computational resources and training time. In contrast, training-free approaches (Ram et al., 2023; Li et al., 2023; Jiang et al., 2023b), while simpler and more efficient as plug-and-play methods, still hinge on the quality of retrieved documents (Chen et al., 2024; Yu et al., 2024). The presence of irrelevant or noisy documents not only reduces response accuracy but also increases computational overhead and compromises system reliability. These challenges underscore the urgent need for robust mechanisms to effectively *filter* and *rank* retrieved content.

To address these challenges, we propose *Multi-Agent FIlteriNg Retrieval-Augmented Generation* (MAIN-RAG), a novel training-free framework designed to enhance the performance and reliability of RAG systems. Unlike existing methods that often rely on computationally intensive training or fine-tuning, MAIN-RAG leverages a collaborative multi-agent approach where multiple LLM agents filter and score retrieved documents. This consensus-driven strategy ensures that only the most relevant and high-quality documents are utilized for generation, significantly reducing noise without sacrificing recall.

[*] Work done as an intern at Visa Research.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2607–2622
July 27 - August 1, 2025

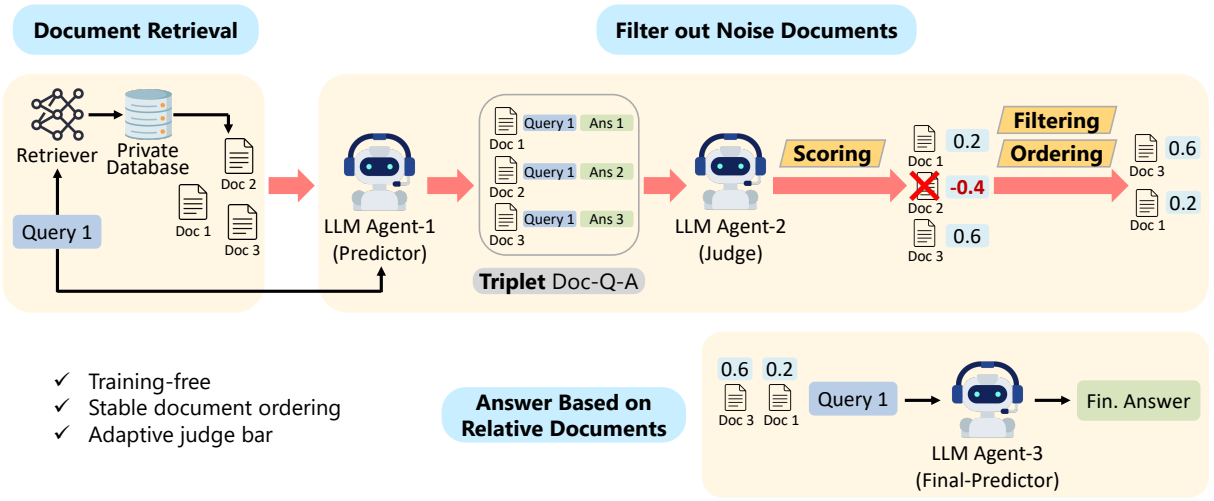

Figure 1: An overview of the proposed framework MAIN-RAG, consisting of three LLM agents to identify noisy retrieved documents for filtering (see Section 3.1). After the retrieval, `Agent-1 "Predictor"` infers answers for each query; then, `Agent-2 "Judge"` takes *Doc-Q-A Triplet* to judge if a document is supportive for LLMs to answer the query. `"Judge"` provides relevant scores for each document for filtering and ordering. Finally, `Agent-3 "Final-Predictor"` answers the query with the given document list.

MAIN-RAG introduces an adaptive filtering mechanism that dynamically adjusts the relevance threshold based on the score distribution of retrieved documents. This adaptability allows the framework to handle diverse queries effectively and ensures robust performance across diverse tasks. Furthermore, the training-free nature of MAIN-RAG eliminates the need for additional labeled data or model fine-tuning, making it a scalable and versatile solution for real-world applications.

Our contributions are as follows:

- **Training-Free Multi-Agent Filtering**: We introduce a novel training-free framework that employs multiple LLM agents to collaboratively filter and rank retrieved documents, improving retrieval precision and RAG reliability without the need for additional training.
- **Dynamic and Adaptive Filtering Mechanism**: MAIN-RAG incorporates an adaptive threshold mechanism that dynamically adjusts to query-specific score distributions, ensuring effective noise reduction while maintaining high recall of relevant documents.
- **Empirical Validation Across Multiple Benchmarks**: Our experimental results on four QA benchmarks demonstrate that MAIN-RAG outperforms baseline RAG approaches, achieving a 2-11% improvement in answer accuracy while reducing the inclusion of irrelevant documents.

By addressing the inherent challenges of noise in document retrieval and providing a training-free solution, MAIN-RAG represents a significant advancement in the field of retrieval-augmented generation. This work details the design, implementation, and evaluation of MAIN-RAG, highlighting its potential to improve response accuracy, consistency, and reliability in diverse NLP applications.

2 Preliminaries

2.1 Notations and Objectives

We consider a RAG system designed to filter noisy retrieved documents and improve response accuracy. Each query $q \in \mathcal{Q}$ retrieves a set of documents $\mathcal{D}_q = \{d_1, d_2, \ldots, d_N\}$ using a retriever model. Each document d_i is associated with a relevance score r_i, which quantifies its usefulness for the query and is determined by **`Agent-2 (Judge)`** as described in Section 3.2. Let $\boldsymbol{R} = [r_1, r_2, \ldots, r_N]$ represent the relevance scores for the retrieved documents. These scores are used to rank the documents, forming an ordered list $\mathcal{D}_q^{\text{rank}}$, where documents with higher scores are deemed more relevant. Based on these scores, an adaptive judge bar τ_q is computed for each query to filter out noisy documents (see Section 3.3). Documents with scores $r_i \geq \tau_q$ are retained, creating a filtered set $\mathcal{D}_q^{\text{filtered}} \subseteq \mathcal{D}_q^{\text{rank}}$. For $1 \leq i \leq N$, r_i represents the relevance score for document d_i. The adap-

tive judge bar τ_q dynamically adjusts based on the distribution of $\boldsymbol{R}$, ensuring robust filtering for diverse queries. For example, consider a query q that retrieves $\mathcal{D}_q = \{d_1, d_2, d_3\}$ with relevance scores $\boldsymbol{R} = [3.8, 2.5, 4.2]$. The ranked list $\mathcal{D}_q^{\text{rank}}$ becomes $\{d_3, d_1, d_2\}$. If the adaptive judge bar $\tau_q = 3.0$, the filtered set $\mathcal{D}_q^{\text{filtered}} = \{d_3, d_1\}$ retains only the most relevant documents. To this end, our work focuses on effectively identifying and filtering noisy documents, thereby enhancing the accuracy and reliability of RAG systems in a post-hoc manner.

2.2 Impact of Noisy Retrieval Documents

In RAG, irrelevant or noisy documents retrieved during the retrieval stage can mislead the LLMs during the inference stage, often resulting in incorrect answers. The presence of such noise information poses a significant challenge to the reliability of LLMs and RAG, especially when applied to tasks that require precise information, such as question answering. As observed in existing studies (Chen et al., 2024; Yu et al., 2024), LLMs exhibit vulnerabilities in noise robustness and often fail to reject irrelevant content, resulting in decreased performance. Therefore, improving noise filtering after the retrieval process is vital to enhance RAG systems' reliability and robustness.

2.3 Related Work

This section reviews RAG methodologies, focusing on training-based and training-free approaches, and discusses the challenge of noise robustness in RAG.

Training-based RAG. Training-based RAG integrates retrieval mechanisms into the training of the language model, allowing access to external information during generation. For instance, Lewis et al. (2020) combines parametric and nonparametric pretrained memory for language generation, achieving state-of-the-art results on open-domain QA tasks. Similarly, Guu et al. (2020) introduces REALM, a framework that augments language model pretraining with a latent knowledge retriever, allowing retrieval and attention to large corpora like Wikipedia. Self-RAG (Asai et al., 2024) proposes to adaptively retrieve passages and critique the generations so as to improve output quality and factuality. Albeit effective, these methods require dedicated training procedures and corresponding hardware, hindering their applicability.

Training-free RAG. Training-free RAG approaches integrate pre-trained language models with retrieval components, avoiding extensive retraining. Ram et al. (2023) perform in-context retrieval, allowing language models to dynamically access external data. Li et al. (2023) propose a framework where LLMs verify retrieved documents to ensure their relevance to queries, but this method is highly sensitive to input prompts. Similarly, Jiang et al. (2023b) introduces a strategy to actively determine when and what to retrieve during generation, but it also suffers from prompt sensitivity. While efficient, training-free RAG approaches struggle with noise robustness due to their reliance on static pre-trained data.

Challenge of noise robustness in RAG. Ensuring noise robustness is critical for the reliability of RAG systems. Chen et al. (2024) conduct a comprehensive analysis of RAG's effects on LLMs, focusing on their resilience to noise and other fundamental capabilities. Yu et al. (2024) presents a framework that strengthens LLMs' RAG performance by guiding them in context ranking and answer generation. Section 3.2, "Trade-off of Picking Top-k Contexts," underscores the significance of selecting relevant contexts to balance effectiveness and computational cost. These findings emphasize the necessity of filtering out noisy documents to uphold the accuracy and robustness of RAG systems.

3 Multi-Agent Filtering RAG (MAIN-RAG)

This section presents a comprehensive overview of our proposed MAIN-RAG framework, as depicted in Figure 1. Based on the traditional RAG workflow, MAIN-RAG focuses on reducing noisy documents after the retrieval stage. Specifically, MAIN-RAG is a training-free framework, involving three agents to identify and filter out noisy documents after retrieval. The specific roles of the three agents are defined in Section 3.1. Section 3.2 illustrates the process of supportive document judgment for filtering out misleading or irrelevant ones. Section 3.3 proposes an adaptive judge bar to adjust the judge criteria according to given retrieved documents.

3.1 Definition of LLM Agents in MAIN-RAG

The proposed framework MAIN-RAG is to identify noisy retrieved documents for filtering out, consisting of three LLM agents: **Agent-1 (Predictor)**, **Agent-2 (Judge)**, and **Agent-3 (Final-Predictor)**.

Agent-1 (Predictor). After the retrieval stage, we have several candidate documents for each query. Then, for a single query, Agent-1

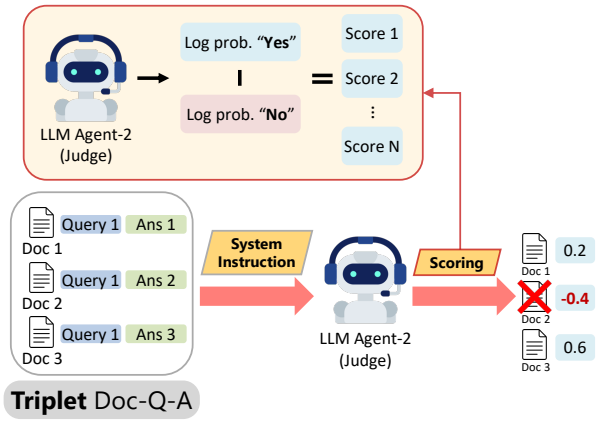

Figure 2: Quantification of document relevant score.

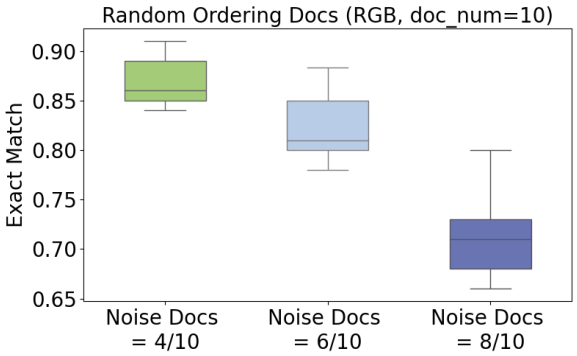

Figure 3: Impacts of document ordering on variance in RAG performance, where Noise Docs t/u means t noisy documents out of u retrieved documents.

is to infer answers to the query given each document. Then, we can form the Document-Query-Answer Triplet (**Doc-Q-A**), which is prepared for `Agent-2 (Judge)` to evaluate the relevant information among Doc-Q-A triplet, as shown in Figure 1.

`Agent-2 (Judge).` Given a Doc-Q-A triplet, `Agent-2 (Judge)` is to evaluate whether the document provides relevant information to the query and answer. `Agent-2` is prompted to answer "`Yes`" or "`No`" for each Doc-Q-A triplet, treating the relevance judgment as a True-or-False question. This simplification helps to further quantify the judgment as relevant scores of documents, which can be used for filtering and ordering. The details of `Agent-2` refer to Section 3.2 and Section 3.3.

`Agent-3 (Final-Predictor).` After `Agent-2` filters out noisy documents and orders the remaining document list by their relevant scores, `Agent-3 (Final-Predictor)` is prompted to answer the query with the document list.

3.2 Relevance Judgment Quantification

Previous research has observed that when processing long context inputs, LLMs tend to overlook information in the middle, placing greater emphasis on the beginning and end of the context (Liu et al., 2024). This suggests that in RAG, the ordering of documents may influence prediction performance. To investigate the impact of document order in RAG, we conducted an experiment on the benchmark RGB (Chen et al., 2024), where the retrieved documents were randomly shuffled and evaluated. This process was repeated ten times for each noise ratio condition. The results, illustrated in Figure 3, reveal that document order has a significant effect on performance. Notably, the maximum performances are substantially higher than the minimum ones, suggesting that certain document orders can provide stable and optimal results. This observation leads us to propose a judgment quantification to make documents sortable.

To quantify the natural language outputs "Yes" and "No," we propose computing the difference between the log probabilities of the corresponding tokens, as shown in Figure 2, where the system instruction is provided in Appendix C. In other words, we choose the log odds of the two tokens as a judgment score. By subtracting the log probabilities of the "Yes" and "No" tokens, `Agent-2` simplifies the judgment by consolidating the two factors into a single score. This relevant score then serves as the sole criterion for document filtering.

3.3 Adaptive Judge Bar τ_q

After we obtain relevant scores for each document, another challenge is how to determine the optimal judge bar for filtering out noisy documents. Here, the optimal judge bar is the score that perfectly filters out all noisy documents while retaining all relevant ones. Consider example 1 in Figure 4, where a query retrieves a higher number of noisy documents; the optimal judge bar in this case is approximately `3.7`. In example 2 in Figure 4, where more relevant documents are retrieved for a query, the optimal judge bar increases to around `4.4`. These

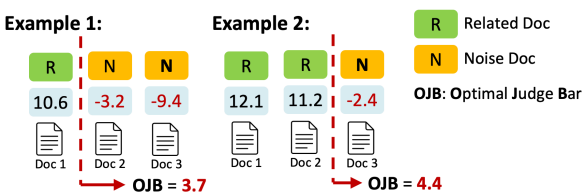

Figure 4: Examples of Optimal Judge Bar (**OJB**).

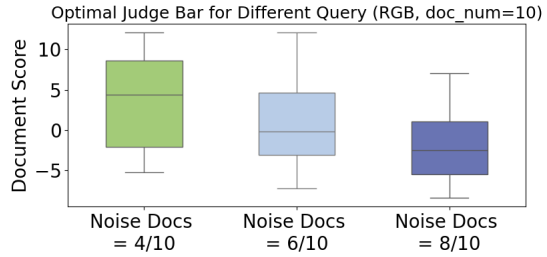

Figure 5: Optimal judge bars for different noise ratios in different queries, where Noise Docs t/u means t noisy documents out of u retrieved documents.

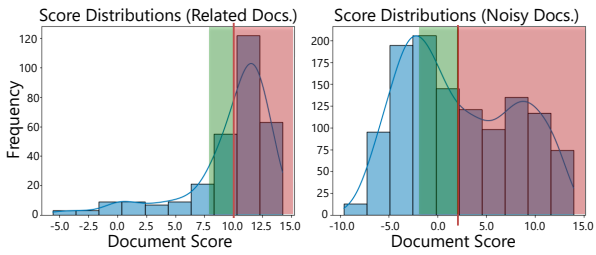

Figure 6: Score distribution of related and noisy documents with adaptive judge bar.

examples illustrate that the optimal judge bar varies with the document distribution among queries. We also observe significant variations in the optimal judge bars across different queries in RGB benchmark (Chen et al., 2024), as shown in Figure 5. This observation leads us to think about how can we adaptively determine optimal judge bars.

Analyzing the relevant score distributions for both related and noisy documents on RGB benchmark (Chen et al., 2024), we observe that the scores of related documents are skewed high with a small standard deviation, as shown in Figure 6. This indicates that the LLM (here is Mistral-7B) is more confident about these documents. In contrast, the scores of noisy documents are more uniformly distributed with a larger standard deviation, suggesting that the LLM is less confident and may misjudge them. Based on this biased LLM behavior, we propose using the average relevant score for each query as an adaptive judge bar. In Figure 6, the red line represents the average score of all documents. Documents to the right of the red line (the red area) are retained, while those to the left are filtered out. When the average score is high—indicating many relevant documents—we can filter out most low-scoring outliers, which are likely noise. Conversely, when the average score is low—indicating many noisy documents—we aim to reduce the number of documents while maintaining a high recall rate for relevant documents by still using the average score to filter out approximately half of the documents. To introduce flexibility into this framework, we adjust the adaptive judge bar τ_q by adding n times the standard deviation σ of each candidate document set, $\tau_q - n \cdot \sigma$, allowing relax τ_q when needed, as shown by the green area in Figure 6. Notably, n is the only hyperparameter in MAIN-RAG.

4 Experiments

In this section, we conduct experiments to evaluate the performance of MAIN-RAG, aiming to answer the following three research questions: **RQ1:** How does MAIN-RAG perform leveraging LLM agents as noisy document filter? **RQ2:** How to utilize adaptive judge bar τ_q for filtering and ranking? **RQ3:** How does τ_q influence performance?

4.1 Tasks and Datasets

We evaluate our MAIN-RAG model and various baselines across a range of downstream tasks, assessing the outputs for overall correctness. All evaluations are conducted in a zero-shot setting, where we provide task instructions without few-shot demonstrations (Sanh et al., 2022; Wei et al., 2021).

Closed-set Task. We evaluate MAIN-RAG on the ARC-Challenge dataset (Clark et al., 2018), a multiple-choice reasoning dataset collected from scientific exams. We use accuracy as the evaluation metric and report results on the testing set.

Open-Domain Question Answering Tasks. We evaluate MAIN-RAG on two open-domain QA datasets: TriviaQA-unfiltered (Joshi et al., 2017) and PopQA (Mallen et al., 2022), both of which require LLMs to answer arbitrary questions about factual knowledge. Since the testing set of TriviaQA-unfiltered is not publicly available, we use the validation and testing sets provided by an existing work (Asai et al., 2024), comprising 11,313 testing queries for evaluation. For PopQA, we utilize the long-tail subset, consisting of 1,399 rare entity queries with monthly Wikipedia page views of less than 100. Following prior works (Mallen et al., 2022; Schick et al., 2024), we evaluate task performance based on whether the gold answers are included in the model's generations instead of strictly requiring exact matches.

Long-form Generation Tasks. We conduct results on the long-form QA task ALCE-ASQA (Gao et al.,

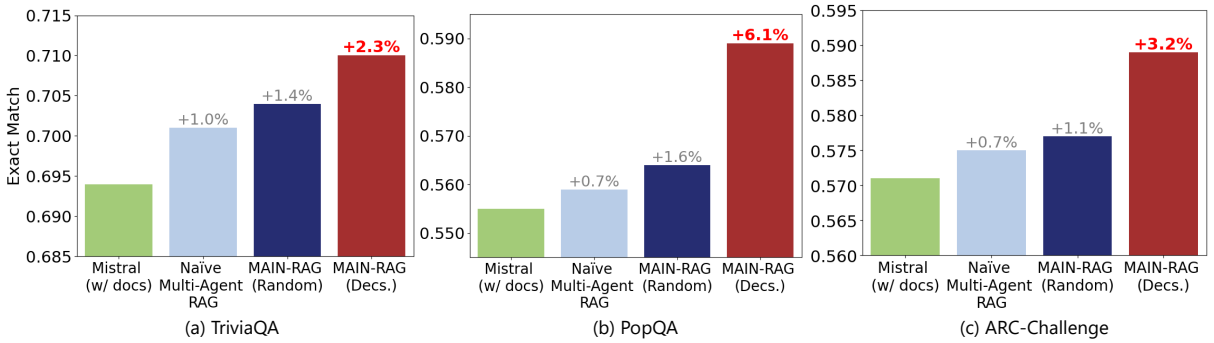

Figure 7: Performance comparison among MAIN-RAG and its variant baselines on three QA benchmarks, where all three LLM agents are pre-trained $\texttt{Mistral}_{7B}$. Comparison based on $\texttt{Llama3}_{8B}$ agents is illustrated in Appendix B.

2023; Stelmakh et al., 2022) to evaluate MAIN-RAG. We rely on the official metrics, including correctness (str-em and rouge), and fluency measured by MAUVE (mau) (Pillutla et al., 2021).

4.2 Baselines

Baselines without retrievals. We assess a range of publicly available, powerful pre-trained LLMs, including $\text{Llama2}_{7B,13B}$ (Touvron et al., 2023), Llama3_{8B} (Dubey et al., 2024), and Mistral_{7B} (Jiang et al., 2023a), as well as instruction-tuned models like $\text{Alpaca}_{7B,13B}$ (Dubois et al., 2024). We also compare our framework with a model trained and enhanced using private data, Llama2-chat_{13B}. Whenever possible, we use the official system prompts or instruction formats that were applied during the training process of these instruction-tuned models.

Baselines with retrievals. We evaluate models that incorporate retrieval, either during inference or throughout the training process. In the first category, we include three fine-tuned models. The first is Self-RAG (Asai et al., 2024), a variant of Llama2_{7B} trained to retrieve documents, generate outputs, and critically examine both retrieved passages and its own responses, expanding its vocabulary with additional reflection tokens. The second is Llama2-FT_{7B}, which is Llama2_{7B} fine-tuned on the same dataset used by Self-RAG, but without the reflection tokens or retrieved passages. We also include results from a retrieval-augmented baseline, $\text{Ret-Llama2-chat}_{13B}$, which is trained on private data collected in Self-RAG and performs inference with retrieved documents. In the second category, we consider standard RAG baselines that do not require additional training. These methods simply prepend the top retrieved documents to the query before passing them to a pre-trained LLM (e.g., $\text{Llama2}_{7B,13B}$, $\text{Alpaca}_{7B,13B}$, Llama3_{8B}, Mistral_{7B}), using the same retriever as in our system. We also consider two variants of MAIN-RAG: **Naïve Multi-agent RAG**: This MAIN-RAG variant replaces Agent-2's role with a simple natural language judgment of "Yes" or "No"; **MAIN-RAG (Random)**: In this variant, after scoring and filtering, the orders of remaining documents are randomized.

4.3 Experimental Settings

As a training-free RAG framework, the three agents in MAIN-RAG can be instantiated by different pre-trained LLMs. As default settings, we instantiate all three agents by pre-trained Mistral_{7B} (Jiang et al., 2023a) and Llama3_{8B} without further tuning. We employ the pre-trained Contriever-MS MARCO (Izacard et al., 2021) as the default retriever model, retrieving up to twenty documents from each query for MAIN-RAG to filter. We use greedy generation for all our experiments.

4.4 Quantitative Analysis (RQ1)

We evaluated the performance of our proposed MAIN-RAG framework and baselines across four well-known QA benchmarks, where MAIN-RAG (Decs.) refers to our method that orders documents in descending order after scoring and filtering, as illustrated in Figure 7, Table 1, and Appendix B. Our results demonstrate that MAIN-RAG outperforms all training-free and without retrieval baselines by margins up to 6.1% (with $\texttt{Mistral}_{7B}$) and 12.0% (with $\texttt{Llama3}_{8B}$) in all four benchmarks, as shown in Table 1. Notably, the questions in PopQA heavily rely on external knowledge to enable pre-trained LLMs to generate accurate answers. In this case, MAIN-RAG exhibits a significant advantage over the baselines, because the retriever is not fine-tuned on the target question sets and may retrieve a large number of noisy can-

Table 1: Overall experimental results on four tasks. **Bold** numbers refer to the best performance among baselines without retrieval and training-free baselines, and underline numbers refer to the second-best performance. **Gray bold** numbers refer to the best performance among proprietary models and training-based baselines. * indicates concurrent results conducted by recent works or original papers. For the metrics, *acc*, *em*, *rg*, and *mau* denote *accuracy*, *str-em*, *rouge*, and *MAUVE*, respectively.

	TriviaQA (acc)	PopQA (acc)	ARC-C (acc)	ASQA (em)	ASQA (rg)	ASQA (mau)
LMs with proprietary data						
Llama2-chat$_{13B}$*	59.3	20.0	38.4	22.4	29.6	28.6
Ret-Llama2-chat$_{13B}$*	59.8	51.8	37.9	**32.8**	34.8	43.8
Baselines with retrieval (training-based)						
Llama2-FT$_{7B}$*	57.3	48.7	65.8	31.0	**35.8**	51.2
Self-RAG$_{7B}$*	**66.4**	**54.9**	**67.3**	30.0	35.7	**74.3**
Baselines without retrieval						
Llama2$_{7B}$*	30.5	14.7	21.8	7.9	15.3	19.0
Alpaca$_{7B}$*	54.5	23.6	45.0	18.8	29.4	61.7
Llama2$_{13B}$*	38.5	14.7	29.4	7.2	12.4	16.0
Alpaca$_{13B}$*	61.3	24.4	54.9	22.9	32.0	**70.6**
Mistral$_{7B}$	54.8	26.2	55.5	11.2	18.1	27.6
Llama3$_{8B}$	68.4	29.2	58.8	19.4	30.3	54.3
Baselines with retrieval (training-free)						
Llama2$_{7B}$	68.9	50.9	51.0	16.2	23.4	33.1
Alpaca$_{7B}$*	64.1	46.7	48.0	30.9	33.3	57.9
Llama2$_{13B}$*	47.0	45.7	26.0	16.3	20.5	24.7
Alpaca$_{13B}$*	66.9	46.1	57.6	34.8	36.7	56.6
Mistral$_{7B}$	69.4	55.5	57.1	32.4	34.8	54.3
Llama3$_{8B}$	73.1	61.8	55.6	37.1	36.5	63.0
MAIN-RAG-Mistral$_{7B}$	71.0	58.9	58.9	35.7	36.2	60.0
MAIN-RAG-Llama3$_{8B}$	**74.1**	**64.0**	**61.9**	**39.2**	**42.0**	**70.6**

didate documents. Compared with training-based baselines, our training-free MAIN-RAG framework can bridge the performance gap in TriviaQA and PopQA datasets. We also found that on the metrics for rough (rg), MAIN-RAG-Mistral$_{7B}$ occasionally outperforms the two training-based baselines, Self-RAG$_{7B}$ and Llama2-FT$_{7B}$, showing the potential of improving pre-trained LLMs to outperform resource-consuming fine-tuning methods.

4.5 Ablation Studies on Adaptive Judge Bar τ_q for Filtering and Ranking (RQ2)

We assess the effectiveness of the adaptive judgment bar τ_q by comparing the default τ_q with variations adjusted by different scales of standard deviation, $\tau_q - n \cdot \sigma$. As mentioned in Section 3.3, the purpose of these adjustments is to relax the filtering threshold when the recall rate of relevant documents is low, potentially preventing the omission of critical external information required to support LLMs in question answering. Despite its flexibility, our experiments demonstrate that the default τ_q generally performs well in filtering noisy documents. As shown in Table 2, while the adjusted variants randomly achieve better performance, the default τ_q consistently ranks at least second-best across three benchmarks and two different pre-trained LLMs, indicating its practicality.

After filtering out irrelevant or noisy documents, the remaining candidate documents can be sorted in either descending or ascending order. As shown in Table 2, MAIN-RAG defaults to descending order, consistently achieving better performance compared to ascending order. This result aligns with findings from prior work, which suggests that LLMs tend to prioritize information presented at the beginning of the input (Liu et al., 2024).

4.6 Case Studies of Different Adaptive Judge Bar τ_q (RQ3)

MAIN-RAG involves adaptive judge bar τ_q to approximate optimal judge bars of each query by averaging relevant scores over retrieved documents for a query. This approach is inspired by our observation of distinct score distributions between the most relevant document set and the least relevant document set, as discussed in Section 3.3. From Figure 6, we observe that Agent-2 assigns confi-

Case Study 1

Question: In what city was Montxu Miranda born?
Adaptive Judge Bar τ_q: `9.575`
Filtered and Ordered Documents: Montxu Miranda Montxu Miranda Díez (born 27 December 1976 in Santurce) is a Spanish pole vaulter. His personal best of 5.81 metres, achieved in September 2000 in Barcelona, is still the standing Spanish national record. ... He studied at the Colegio San Calixto, then later pursued a career in Political Sciences at the Higher University of San Andrés in La Paz.
Ground Truth: "Santurtzi", "Santurce"
LLM Answer: Montxu Miranda was born in Santurce. (**correct**)

Case Study 2

Question: What is the capital of Gmina Czorsztyn?
Adaptive Judge Bar τ_q: `-8.425`
Filtered and Ordered Documents: Gmina Wolsztyn is an urban-rural gmina (administrative district) in Wolsztyn County, ... Sromowce Wyżne is a village in the administrative district of Gmina Czorsztyn, within Nowy Targ County, Lesser Poland Voivodeship, in southern Poland, close to the border with Slovakia. It lies approximately 8 km south-east of Maniowy, 25 km east of Nowy Targ, ...
Ground Truth: "Maniowy"
LLM Answer: The capital of Gmina Czorsztyn is Maniowy. (**correct**)

Case Study 3

Question: What is Arcangelo Ghisleri's occupation?
Adaptive Judge Bar τ_q: `0.4875`
Filtered and Ordered Documents: S. Michele Arcangelo, archangel in Jewish, Christian, and Islamic teachings ; Andrea di Cione Arcangelo (1308–1368), Italian painter, sculptor, and architect active in Florence ; Antonio di Arcangelo, Italian painter, active in Florence in a Renaissance style, between 1520 and 1538 ; Arcangelo Califano (1730–1750), baroque composer and cellist...
Ground Truth: "journalist", "journo", "journalists"
LLM Answer: Arcangelo Ghisleri was an Italian geographer, writer, and Socialist politician. (**wrong**)

Figure 8: Case Study: **Adaptive Judge Bar** τ_q (Dataset: **PopQA**; `LLM Agents`: **`Mistral`$_{\texttt{7B}}$**)

Table 2: Ablation studies of τ_q and document ordering. **Bold** numbers indicate the best result, and underline numbers indicate the second-best result.

	TriviaQA (acc)	PopQA (acc)	ARC-C (acc)
Mistral$_{7B}$			
MAIN-RAG (Decs.)	71.0	**58.9**	58.9
MAIN-RAG (Asc.)	70.2	53.5	57.4
MAIN-RAG ($\tau_q - 0.5 \cdot \sigma$)	**71.2**	58.6	**59.0**
MAIN-RAG ($\tau_q - 1.0 \cdot \sigma$)	70.8	58.0	58.5
MAIN-RAG ($\tau_q - 1.5 \cdot \sigma$)	70.4	58.4	57.7
Llama3$_{8B}$			
MAIN-RAG (Decs.)	74.1	**64.0**	**61.9**
MAIN-RAG (Asc.)	73.6	63.5	60.7
MAIN-RAG ($\tau_q - 0.5 \cdot \sigma$)	74.1	**64.0**	58.6
MAIN-RAG ($\tau_q - 1.0 \cdot \sigma$)	74.1	63.3	58.9
MAIN-RAG ($\tau_q - 1.5 \cdot \sigma$)	**74.3**	**64.0**	57.2

dently high relevance scores to related documents, resulting in a skewed-high score distribution. In contrast, while `Agent-2` scores noisy documents with a more uniform distribution, the lowest scores for noisy documents are significantly lower than those for related documents. This disparity allows the filtering mechanism to improve the prediction accuracy of `Agent-3`, regardless of whether τ_q is set relatively high or low. The correlation between τ_q and performance can be observed in Figure 8 and further discussed in Appendix D.

5 Conclusion and Future Work

In this work, we address the challenges of noisy document retrieval in RAG by introducing a training-free, multi-agent framework, `MAIN-RAG`. Our approach employs multiple LLM agents to collaboratively filter and rank retrieved documents, enhancing the recall of relevant information while minimizing irrelevant content. Specifically, `MAIN-RAG` utilizes an adaptive judge bar that dynamically adjusts based on the score distribution of relevant and noisy documents in different queries. Experimental results demonstrate that `MAIN-RAG` consistently outperforms training-free RAG baselines across various QA benchmarks. Regarding future directions, the `MAIN-RAG` framework unveils several potential facets that merit further exploration, such as integrating with a more fine-grained adaptive judge bar, extending the approach to other tasks beyond question answering, and incorporating human feedback or tuning-based approaches to enhance the efficacy of document filtering.

6 Limitations

We conduct experiments on four datasets using two different pre-trained LLM architectures. These experiments primarily focus on LLM inference with retrieved external documents. However, we acknowledge that LLM inference under RAG workflow contributes to carbon emissions, representing a potential limitation and environmental risk of our work. To mitigate this, we aim to reduce the need for repetitive experiments by ensuring more predictable outcomes and implementing controlled experimental settings.

Acknowledgments

The authors thank the anonymous reviewers for their helpful comments. This work is in part supported by NSF grants NSF IIS-2431515 and IIS-2525159. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

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*.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

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

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

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.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

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*.

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.

Xiaonan Li, Changtai Zhu, Linyang Li, Zhangyue Yin, Tianxiang Sun, and Xipeng Qiu. 2023. Llatrieval: Llm-verified retrieval for verifiable generation. *arXiv preprint arXiv:2311.07838*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 11:157–173.

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*.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023. Self-knowledge guided retrieval augmentation for large language models. *arXiv preprint arXiv:2310.05002*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. RankRAG: Unifying context ranking with retrieval-augmented generation in LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.