ParetoRAG: Leveraging Sentence-Context Attention for Robust and Efficient Retrieval-Augmented Generation

Ruobing Yao[1,2,3], Yifei Zhang[*], Shuang Song[1,2,3], Yuhan Liu[4], Neng Gao[1,3†], Chenyang Tu[1,3]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China,
[2]School of Cybersecurity, University of Chinese Academy of Sciences, Beijing, China,
[3]State Key Laboratory of Cyberspace Security Defense, Beijing, China,
[4]China Mobile Group, Hunan Company Limited, Changsha, China,

Abstract

While Retrieval-Augmented Generation systems enhance Large Language Models by incorporating external knowledge, they still face persistent challenges in retrieval inefficiency and the inability of LLMs to filter out irrelevant information. We present **ParetoRAG**, an unsupervised framework that optimizes RAG systems through sentence-level refinement guided by the **Pareto principle**. By decomposing paragraphs into sentences and dynamically reweighting core content while preserving contextual coherence, ParetoRAG achieves dual improvements in retrieval precision and generation quality without requiring additional training or API resources, while using only 40% of the tokens compared to traditional RAG approaches. This framework has been empirically validated across various datasets, LLMs, and retrievers. Furthermore, we show that ParetoRAG's architectural improvements are orthogonally compatible with adaptive noise-robust models, enabling retrieval-augmented optimization and robust training to enhance generation quality mutually. This highlights complementary architectural refinements and noise mitigation, offering insights for integrating retrieval augmentation with robustness enhancement.

1 Introduction

With the development of Large Language Models (LLMs), their general capabilities have become increasingly powerful (Achiam and Adler, 2023; Dubey et al., 2024). However, even the most advanced LLMs still face challenges with factual errors (Min et al., 2023; Huang and Chen, 2024). One major limitation lies in their static parametric memory, which prevents them from adapting to dynamically evolving knowledge demands or covering unknown domains beyond their training data

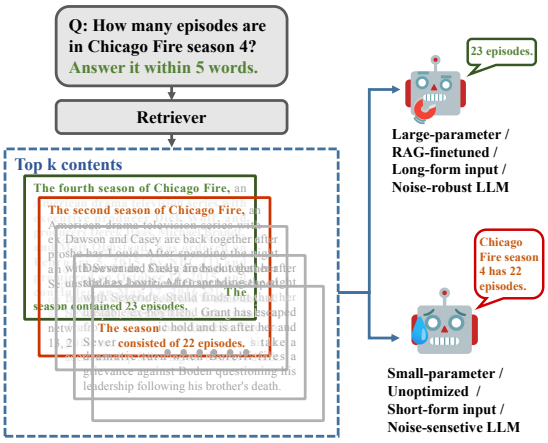


Figure 1: The examples show that much noise impedes the LLM from acquiring accurate knowledge from the retrieved content and could potentially misdirect its reasoning. Finding the correct answer relies on the ability of LLM to identify a small portion of key information.

(Kasai et al., 2023). LLMs are prone to generating plausible hallucinations but lack factual accuracy (Huang et al., 2024). These challenges significantly hinder the performance of LLMs in knowledge-intensive tasks (Ram et al., 2023). To address these limitations, Retrieval-Augmented Generation (RAG) (Lewis et al., 2020a; Xiong et al., 2020; Izacard et al., 2021) integrates relevant passages from external databases into the input context, effectively enhancing the reliability and performance of models in open-domain question answering and dynamic knowledge retrieval tasks.

However, the effectiveness of RAG highly depends on the quality of the retrieved information (Fan et al., 2024). Additionally, interference from redundant information and increased input length are critical factors that significantly impact model performance. In the retrieval stage, the relevance scores of core sentences can be overshadowed by redundant content at the same passage level, reducing the prominence of key information in the retrieved content. In the generation stage, retriev-

[*]Corresponding author: ifzh@foxmail.com
[†]Corresponding author: gaoneng@iie.ac.cn

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 4137–4151
November 4-9, 2025

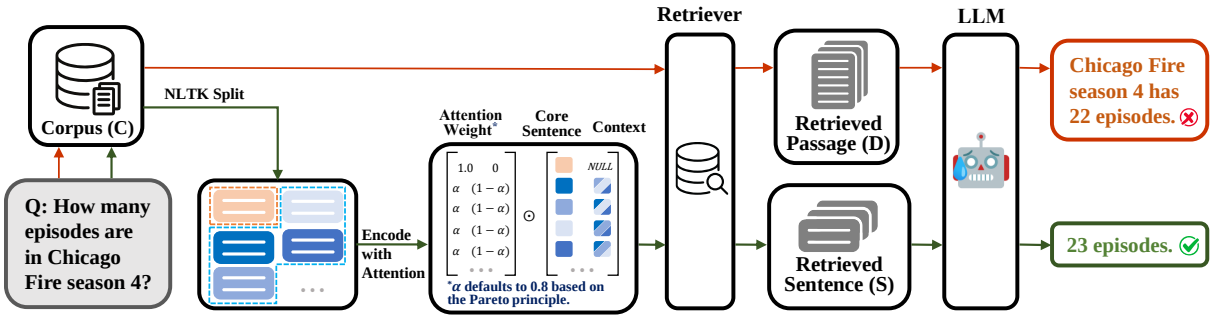


Figure 2: Comparison of the traditional RAG (red path) and ParetoRAG(green path). The traditional method retrieves and directly uses entire passages, often introduces redundant information, leading to inaccurate answers. In contrast, our method utilizes a preprocessed sentence-level corpus, assigning higher weights to key sentences while appropriately preserving and weighting contextual information to avoid loss of coherence. Inspired by the Pareto principle (the 80/20 rule), this design emphasizes critical information while maintaining necessary semantic consistency. The selected sentences are fed into the LLM, resulting in more accurate answers.

ing excessive content to provide rich context can result in overly lengthy inputs, which may cause the model to lose focus and diminish its ability to concentrate on critical information (Jin et al., 2024; Shi et al., 2023). Figure 1 shows that core sentences account for only a small portion of the top k retrieved content. Excessive irrelevant or redundant information hinders the ability of the model to extract accurate knowledge and increases the risk of generating hallucinations (Zhang et al., 2024; Liu et al., 2024a). In addition, in the zero-shot Chain-of-Thought (Wei et al., 2022) prompting setup, the ability of LLM to follow instructions shows a significant decline as the input size increases. The model tends to directly generate answers before completing reasoning steps, and this tendency becomes more pronounced as inputs grow longer (Levy et al., 2024).

Retrieval-augmented language models (RALMs) (Zhang et al., 2024; Lin et al., 2024), Long-context LLMs (Dubey et al., 2024; Team et al., 2024), and Adaptive Noise-Robust Model (Yoran et al., 2024; Fang et al., 2024) can be considered as solutions. These models enhance the ability to process long-form text and improve the robustness of noisy information, enabling them to focus more effectively on key information and reduce the impact of redundant content. However, these approaches require additional training resources and high computational costs for further training and model fine-tuning. Another possible solution is to reduce the granularity of retrieval from the document level to the sentence level (Lee et al., 2021; Chen et al., 2024). However, this approach may inadvertently lose some important contextual information (e.g., in the example in Figure 1, "The season" refers to Chicago Fire season 4 in one context and season 2 in another), which is crucial for accurately answering the given query (Choi et al., 2021). Therefore, we propose a method that does not require additional training resources while effectively preserving contextual information and reducing document redundancy.

In this work, we present **ParetoRAG**, an unsupervised framework built upon the RAG system. Our approach leverages a preprocessed sentence-level corpus, assigning higher weights to key sentences while carefully preserving and weighting contextual information to maintain coherence. Drawing inspiration from the **Pareto principle** (the 80/20 rule), ParetoRAG prioritizes critical information while ensuring semantic consistency, effectively enhancing both the retrieval and generation stages of the RAG pipeline. Notably, ParetoRAG requires neither additional training resources nor extra API calls. The overall ParetoRAG framework is illustrated in Figure 2.

We validate ParetoRAG across three datasets over three retrievers. Our main contributions are as follows:

- A plug-and-play method named ParetoRAG is proposed to achieve the decomposition from paragraph level to sentence level, effectively retaining contextual information during retrieval without additional training.
- ParetoRAG achieves notable improvements in accuracy and fluency while reducing token consumption to approximately 40% of the original cost. Furthermore, it demonstrates

strong generalization, as this conclusion is consistently validated across various datasets, LLMs, and retrievers.

- We investigate the methodological compatibility between ParetoRAG's improvements and the adaptive noise-robust model. The findings suggest that retrieval-augmented architectures and robust training strategies can be orthogonally beneficial, providing architectural-level enhancements that complement rather than interfere with existing noise mitigation approaches.

2 Related Work

Retrieval-Augmented Generation with Noisy Context RAG (Guu et al., 2020; Lewis et al., 2020a) is considered a useful method to address hallucinations, which improves the input questions of generative LLM with retrieved documents. It usually provides an extra knowledge source from a specific corpus, i.e., Wikipedia, which greatly improves the performance of LLM in a variety of tasks, especially in the knowledge-intensive ones (Ram et al., 2023). However, due to the limitation of retrieval capabilities, retrieval-augmented systems inevitably introduce irrelevant or partially relevant knowledge to the models (Yin et al., 2023). In recent years, the impact of noisy information on the performance of RAG systems has received increasing attention (Zhu et al., 2019; Yu et al., 2024). Some studies (Jia and Liang, 2017; Creswell et al., 2022) have shown that the introduction of irrelevant noise significantly degrades model performance. Further analyses (Chen et al., 2025) indicate that as the proportion of noise in the retrieval context increases, the performance of large language models (LLMs) deteriorates significantly. In addition, research (Fang et al., 2024) has explored the effects of different types of noise on RAG systems and found that counterfactual retrieval noise has the most detrimental impact on retrieval systems.

Advanced RAG Many advanced approaches have been developed from the original RAG in recent years (Kim et al., 2023; Zhang et al., 2024; Liu et al., 2024b; Patil et al., 2024). Considering that retrieval is sometimes unnecessary for some queries, responses without retrieval are even more accurate in many situations. SelfRAG (Asai et al., 2023) is proposed to selectively retrieve knowledge and introduce a critical model to decide whether to retrieve it. SAIL (Luo et al., 2023) is tuned on instructions to insert the retrieved documents before the instructions. RECOMP (Xu et al., 2024) is designed to refine the retrieved passages by either abstractively or extractively summarizing them with additional models.

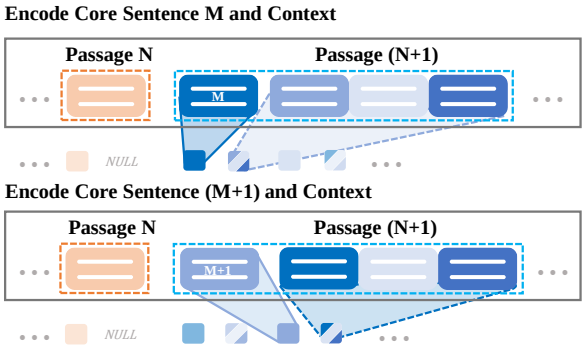


Figure 3: The example of ParetoRAG encodes core sentence M and core sentence (M+1). The content within the same dashed box is split from the same passage. The context of a core sentence consists of sentences from the same passage, excluding the core sentence itself.

Compared with recent studies (Hwang et al., 2024; Chen et al., 2024) that are the most relevant to our work, a primary difference should be highlighted. Dense X reduces information redundancy by degrading documents into proposition sentences. The authors trained a fine-tuned text generation model to decompose paragraphs into propositions to supplement contextual information. In contrast, our approach utilizes the Sentence-Context Weighted Attention mechanism to supplement contextual information without requiring any fine-tuning.

3 Method

In this section, we introduce a novel framework, ParetoRAG, for improving the accuracy of retrieval results by leveraging sentence-level weighting inspired by the Pareto Principle integrated into the RAG system. Notably, ParetoRAG requires neither additional training resources nor extra API calls.

3.1 Encoding Step

The encoding step involves extracting the core sentence and its corresponding context from the passages and encoding them into dense vector representations. This process ensures the inclusion of both the key information from the core sentence and the supplementary information from its surrounding context, enabling semantically rich representation for subsequent retrieval.

Passage Segmentation. The passages in the retrieval corpus $\mathcal{C}$ are segmented into sentences using NLTK. The retrieval corpus is represented as a collection of passages:

$$\mathcal{C} = \{P_1, P_2, \ldots, P_m\},$$

where m is the total number of passages. Each passage $P_j \in \mathcal{C}$ is represented as a sequence of sentences:

$$P_j = \{s_1^j, s_2^j, \ldots, s_{n_j}^j\},$$

where s_i^j represents the i-th sentence in the j-th passage, and n_j is the total number of sentences in P_j.

Core Sentence and Context Extraction. For each passage $P_j \in \mathcal{C}$, every sentence s_i^j is considered a core sentence, and its corresponding context is defined as the concatenation of all other sentences in the same passage, excluding s_i^j itself. Formally, for a passage $P_j = \{s_1^j, s_2^j, \ldots, s_{n_j}^j\}$, the context for the core sentence s_i^j is defined as:

$$\text{Context}(s_i^j) = \begin{cases} \{s_1^j, \ldots, s_{i-1}^j, s_{i+1}^j, \ldots, s_{n_j}^j\}, & \text{if } n_j > 1, \\ \text{NULL}, & \text{if } n_j = 1. \end{cases}$$

Here, $\text{Context}(s_i^j)$ captures the surrounding information in the passage P_j without including the core sentence s_i^j itself. This ensures that each core sentence s_i^j can be analyzed independently while still being informed by its contextual sentences. If a passage consists of only one sentence ($n_j = 1$), the context is defined as NULL.

Encode Core Sentence, Context and Query. For each core sentence s_i^j and its corresponding context $\text{Context}(s_i^j)$, a configurable encoder $\text{Enc}_\theta(\cdot)$ is applied to obtain their vector representations. Here, θ represents the model selection parameter, which determines the specific encoder to be used (e.g., Contriever, ANCE, or DPR).

The core sentence s_i^j, its context $\text{Context}(s_i^j)$, and the query are encoded into d-dimensional vector representations using the same encoder $\text{Enc}_\theta(\cdot)$. The encoding process is as follows:

$$\begin{aligned} \mathbf{h}_{\text{core}}^i &= \text{Enc}_\theta(s_i^j), \\ \mathbf{h}_{\text{context}}^i &= \text{Enc}_\theta(\text{Context}(s_i^j)), \\ \mathbf{q} &= \text{Enc}_\theta(\text{Query}), \end{aligned}$$

where $\mathbf{h}_{\text{core}}^i$, $\mathbf{h}_{\text{context}}^i$, and $\mathbf{q}$ are all d-dimensional vectors. These representations are used for similarity computation and ranking.

Figure 3 shows the example of ParetoRAG encodes core sentence M and core sentence (M+1).

3.2 Retrieval Step

The retriever step takes the encoded core sentence, context, and query vectors to compute their similarity and rank the results for retrieval. This process consists of the following key substeps:

Sentence-Context Weight Adjustment To balance the contributions of the core sentence and its context, an attention mechanism assigns weights based on a hyperparameter α. The weighted representation is computed as:

$$\mathbf{h}_{\text{weighted}}^i = \begin{cases} \mathbf{h}_{\text{core}}^i, & \text{if Context}(s_i^j) = \emptyset\}, \\ \alpha \cdot \mathbf{h}_{\text{core}}^i + (1-\alpha) \cdot \mathbf{h}_{\text{context}}^i, & \text{otherwise.} \end{cases}$$

This mechanism ensures that both the key information from the core sentence and the supplementary information from its context are considered during similarity computation.

Similarity Computation Following previous studies (Lewis et al., 2020a; Zou et al., 2024), the similarity between the weighted sentence representation $\mathbf{h}_{\text{weighted}}^i$ and the query vector $\mathbf{q}$ is computed using dot similarity by default.

$$\text{Sim}(s_i^j, \mathbf{q}) = \mathbf{h}_{\text{weighted}}^i \cdot \mathbf{q}.$$

This step quantifies how closely each sentence-context pair matches the semantic meaning of the query.

Top-k Sentence Ranking The top-k sentences are ranked based on their similarity scores in descending order:

$$\text{Top-}k = \arg\text{top}_k \left(\text{Sim}(s_i^j, \mathbf{q}) \right),$$

where $\arg\text{top}_k$ returns the indices of the k sentences with the highest similarity scores. These top-k sentences are selected as the retrieval results, providing the most relevant information based on the query.

3.3 Generation Step

After the ranking step, the top-k ranked sentences, denoted as Top-k, are passed to the language model M to generate the final answer a for the given query $\mathbf{q}$. The generation process integrates the query and the retrieved sentences to produce a response that is both accurate and contextually relevant. The generation step can be formalized as:

$$a = \text{Generate}(\mathbf{q}, \text{Top-}k; M),$$

Model	NQ(acc)				Hotpot(acc)				MS(mauve)				MS(rouge)		
	# tok	Contriever	ANCE	DPR	# tok	Contriever	ANCE	DPR	# tok	Contriever	ANCE	DPR	Contriever	ANCE	DPR
Without RAG															
Vicuna-7B	/	23.2			/	16.1			/	88.3			40.5		
Vicuna-13B		28.2				20.2				82.1			40.7		
Llama2-7B-chat		20.9				16.0				85.6			36.2		
Llama2-13B-chat		29.9				18.4				90.1			39.6		
Naive RAG															
Vicuna-7B	1277	33.2	36.1	41.9	1202	25.0	22.3	23.9	810	84.1	84.9	87.9	35.8	35.6	37.5
Vicuna-13B		37.4	41.0	45.6		22.6	20.2	22.7		86.8	87.7	87.0	37.8	36.9	36.9
Llama2-7B-chat		33.2	37.9	40.8		23.6	23.4	23.3		85.0	88.6	89.2	33.6	34.4	35.4
Llama2-13B-chat		38.3	39.6	42.7		27.1	25.3	26.8		77.5	87.0	88.8	33.2	33.5	34.7
Recomp (Xu et al., 2024)															
Vicuna-7B	26	35.9	39.3	43.4	41	29.0	25.5	27.9	26	79.2	83.8	85.3	40.4	42.3	41.2
Vicuna-13B		36.8	41.5	42.5		28.9	25.4	25.6		85.4	85.7	87.2	40.5	41.6	40.3
Llama2-7B-chat		24.3	31.5	33.9		26.7	22.4	25.2		25.7	41.2	38.5	36.7	39.2	37.6
Llama2-13B-chat		31.5	36.2	39.5		30.1	25.2	28.9		40.4	45.9	41.6	37.0	39.0	37.7
Dense X (Chen et al., 2024)															
Vicuna-7B	466	34.1	38.5	41.5	427	15.4	14.4	14.4	396	88.1	87.4	91.5	42.6	45.7	44.3
Vicuna-13B		34.3	38.1	41.9		16.0	17.1	17.6		84.3	88.8	86.7	42.2	44.1	43.8
Llama2-7B-chat		31.7	35.2	37.4		14.1	13.6	13.6		92.0	89.9	92.6	42.2	44.1	43.5
Llama2-13B-chat		34.5	43.1	43.3		15.0	14.6	15.5		70.4	73.9	78.6	40.0	43.1	43.0
ParetoRAG (Ours)															
Vicuna-7B	457 ↓ 64%	36.6	43.4	46.7	478 ↓ 60%	25.4	25.3	24.7	326 ↓ 60%	90.9	90.7	91.4	42.5	44.4	43.4
Vicuna-13B		39.1	44.2	48.2		26.7	25.9	26.0		87.7	92.6	88.6	41.6	43.6	42.6
Llama2-7B-chat		34.0	41.7	42.3		24.6	25.0	24.0		92.0	89.9	92.6	40.4	43.3	41.9
Llama2-13B-chat		36.1	41.8	47.4		25.9	26.1	25.2		92.2	90.1	92.0	39.0	42.2	40.8

Table 1: Overall experiment results of three retrievers on three tasks, based on top 10 recall contents. Dense X top 20 is used to ensure consistent input token numbers. The lower the relative improvement, the deeper the red background. In contrast, the deeper the blue background, the higher the relative improvement compared to Naive RAG. The underlined numbers indicate the best-performing results on the current dataset.

where Generate$(\cdot)$ represents the generation function that combines the query $\mathbf{q}$, the retrieved top-k sentences Top-k, and the language model M to produce the output a.

4 Experiment Setups

In this section, we describe the experimental setup for evaluating ParetoRAG across various scenarios. The specific model parameters can be found in Appendix E. The selection and meaning of the evaluation metrics can be found in Appendix F.

4.1 Datasets.

We experiment on three different open-domain QA datasets as the retrieval source: Natural Questions (NQ) (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), and MS-MARCO (Nguyen et al., 2016), where each dataset has a knowledge database. The knowledge databases of NQ and HotpotQA are collected from Wikipedia. The knowledge database of MS-MARCO is collected from web documents using the MicroSoft Bing search engine. These datasets encompass different tasks, such as open-domain question answering, multi-hop reasoning. Each dataset also contains a set of questions. We randomly sampled 1,000 data pairs for testing. Table 3 shows statistics of text unit counts before and after ParetoRAG encoding.

4.2 Dense Retrieval Models

We compare the performance of the three following unsupervised, semi-supervised, or supervised dense retriever models. Following previous studies (Lewis et al., 2020b), by default, we use the dot product between the embedding vectors of a question and a text in the knowledge database to calculate their similarity score.

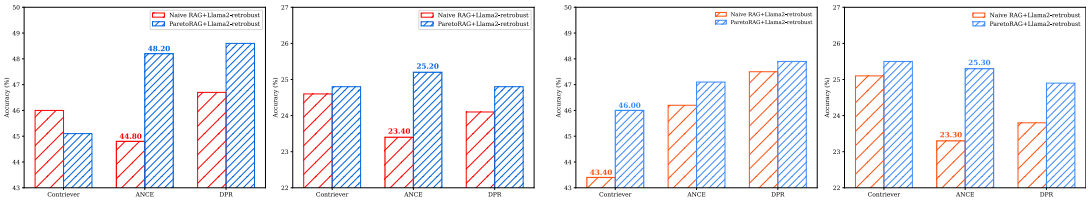
Contriever (Izacard et al., 2021) is an unsupervised retriever implemented using a BERT-base encoder. Contriever is contrastively trained on segment pairs constructed from unlabeled documents in Wikipedia and web crawl data.

ANCE (Xiong et al., 2020) is a dual-encoder BERT-base model designed for dense retrieval tasks. It is trained using weakly supervised signals from query-document pair labels, typically sourced from datasets such as MS-MARCO.

DPR (Karpukhin et al., 2020) is a dual-encoder BERT-base model fine-tuned on passage retrieval tasks directly using the question-passage pair labels from NQ, TQA (Joshi et al., 2017), SQuAD (Rajpurkar et al., 2016) and WebQ (Berant et al., 2013).

4.3 Baselines

For these three baselines, we evaluated publicly available instruction-tuned models, such as Vicuna-7B and Vicuna-13B (Zheng et al., 2023), as well



(a) NQ (Top 10) (b) HotpotQA (Top 10) (c) NQ (400 words) (d) HotpotQA (400 words)

Figure 4: Comparison of ParetoRAG and Naive RAG on the adaptive noise-robust LLM (`llama-2-13b-peft-nq-retrobust` and `llama-2-13b-peft-hotpotqa-retrobust` (Yoran et al., 2024)): (a)(b) show performance under the same recall size (Top 10), while (c)(d) illustrate performance under the same input word count (400).

as models trained and reinforced with private data, including Llama2-7B-Chat and Llama2-13B-Chat (Touvron et al., 2023).

Baselines without retrievals. We evaluate the performance of various LLMs without employing RAG technology across multiple datasets.

Baselines with Naive RAG. We employ the most basic RAG technique without incorporating complex retrieval optimization methods or advanced generation mechanisms, relying solely on the fundamental retrieval-generation workflow.

Baselines with SOTA methods. We implement three advanced approaches: (1) Recomp (Xu et al., 2024) using abstractive summarization (excluding extractive variants) to synthesize retrieved passages with dedicated models. (2) LLM trained with adversarial noise to improve robustness (Yoran et al., 2024). (3) Dense X (Chen et al., 2024) reduces information redundancy by degrading documents into proposition sentence.

5 Experimental Results and Analysis

In this section, we show the overall experimental results with in-depth analyses of our framework. We also provide a cost analysis in the Appendix I.

5.1 Main Results

Table 1 presents the results of three retrievers on three datasets, based on top-10 recall contents. Figure 4 illustrates the performance of ParetoRAG on `llama2-13b-retrobust`. From these results, we can conclude the following findings:

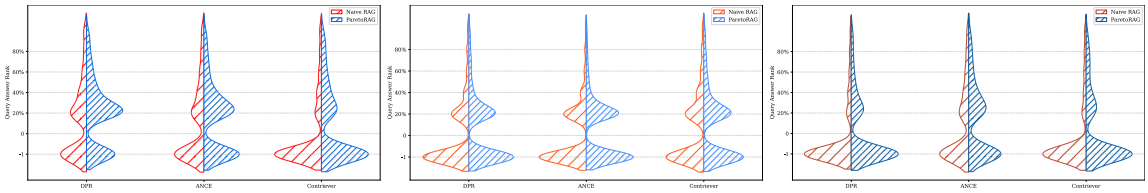
ParetoRAG, while consuming only about 40% of the original token cost, still delivers notable improvements in accuracy and fluency. Specifically, as shown in Table 1, in NQ, the accuracy of Vicuna-7B + ANCE increases from 36.1% to 43.4% (+7.3%), with the token count reduced to 26% of the original. Similarly, the accuracy of Llama2-13B-Chat + DPR increases from 42.7% to 47.4% (+4.7%), with the token count reduced to approximately 30% of the original. In HotpotQA, the accuracy of Vicuna-13B + ANCE improves from 20.2% to 25.9% (+5.7%), with the token count reduced to approximately 40% of the original.

In addition, in MS-Marco, ParetoRAG achieves notable improvements in both mauve (fluency) and rouge (correctness) metrics. For example, the fluency score of Llama2-13B-Chat + Contriever increases from 77.5 to 92.2 (+14.7%), while the token count is reduced to approximately 32% of the original. Similarly, the fluency score of Vicuna-7B + DPR improves from 87.9 to 91.5 (+3.6%). In terms of correctness (rouge), the rouge score of Vicuna-13B + ANCE increases from 46.8 to 55.2 (+8.4%), while Llama2-13B-Chat + ANCE improves from 46.1 to 55.1 (+9%). These results further highlight ParetoRAG's capability to deliver consistent and measurable improvements in both fluency and accuracy, even with significantly reduced token consumption.

ParetoRAG demonstrates strong generalizations. We analyze its effectiveness from three perspectives:

Effectiveness across multiple datasets: ParetoRAG consistently improves performance across a diverse range of datasets, including NQ, HotpotQA and MS-Marco. In contrast, despite having fewer tokens, Recomp does not have a specialized abstract model for the MS-MARCO task, resulting in a significant drop in performance on MAUVE (fluency) and a smaller improvement on ROUGE-L (correctness) compared to ParetoRAG.

Compatibility with different types of retrievers: ParetoRAG proves effective with various dense retriever types, including Contriever, ANCE, and DPR. This shows that the method is not tied to a specific retriever and adapt well to different re-



(a) NQ distribution (b) HotpotQA distribution (c) MS-MARCO distribution

Figure 5: Correct answer rank distributions across different datasets under the the same input word count (400).

trieval methods. Specific analysis of the impact on retrievers can be found in 5.2.2.

Applicability across multiple LLMs: ParetoRAG achieves improvements when applied to large language models, such as Vicuna-7B, Vicuna-13B, Llama2-7B-Chat, and Llama2-13B-Chat. Notably, we also test the method on models trained with anti-noise techniques. As shown in Figure 4, the results still show improvements. More detailed analysis can be found in 5.2.4.

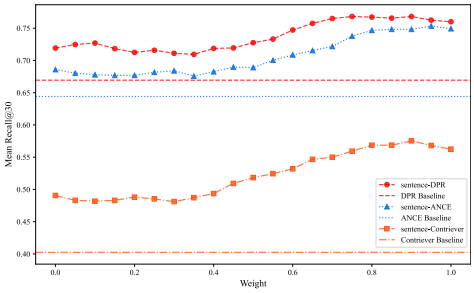


Figure 6: Impact of Core Sentence Weight on Recall across NQ Dataset.

5.2 Ablation Study

We study the impact of core sentence weight, retriever types, and top k size on ParetoRAG. The variation of core sentence weight on HotpotQA and MS-MARCO can be found in Appendix G, while the impact of model parameters on ParetoRAG is detailed in Appendix H.

5.2.1 Impact of core sentence weight

From Figure 6, it can be observed that when the weight of core sentences is adjusted to approximately 0.80, the Mean Recall@30 for ANCE, DPR and Contriever methods reaches optimal performance. This phenomenon reflects the impact of weight adjustment on the balance between contextual information and core sentences, which can be analyzed as follows:

Performance Improvement at Optimal Weight (Around 0.80): When the core sentence weight is set to approximately 0.80, the model effectively integrates contextual information with the content of core sentences. This balance enables the model to preserve semantic integrity while more accurately capturing key information relevant to the retrieval task, thereby achieving optimal recall performance.

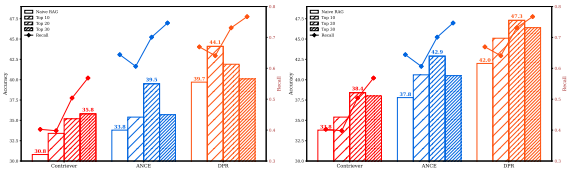
Performance Decline with Increased Weight (Beyond 0.80): As the core sentence weight increases further toward 1.0, contextual information in the text is progressively diminished or even neglected, causing the model to rely more heavily on core sentences for retrieval. However, excessively weakening contextual information leads to a loss of semantic completeness, which adversely affects the accuracy of retrieval results. Consequently, performance declines beyond the 0.80 threshold.

High Weight Still Outperforms the Baseline (At 1.0): Even when the core sentence weight reaches 1.0, resulting in the complete disregard of contextual information, the model's performance remains superior to the baseline of paragraph-level retrieval. This indicates that paragraph-level information often contains significant redundancy, while core sentences play a pivotal role in enhancing retrieval performance. By adjusting the weighting, ParetoRAG effectively reduces the spatial burden of paragraph content while incorporating more core sentences, thereby improving retrieval precision and optimizing efficiency simultaneously.

5.2.2 Impact of retriever

Figure 5 compares the ranking distribution of correct answers across different datasets when using ParetoRAG and Naive RAG. The y-axis shows the percentage position of the correct answer within the ranked retrieval results, and the x-axis shows the density distribution of correct answer positions in the retrieval results. 20% indicates that the correct answer appears in the top 20% of the retrieval results. Higher percentages correspond to lower positions in the ranking, and values near -1 represent cases where the correct answer is not retrieved.

After being optimized by ParetoRAG, the three



(a) Vicuna-7B (b) Vicuna-13B

Figure 7: Comparison of accuracy and recall rates of different retrievers under various top k conditions.

retrievers, DPR, ANCE, and Contriever, exhibit the following trends: First, the correct answer rankings for all retrievers form a peak around 20%, indicating that ParetoRAG effectively pushes correct answers to higher positions in the retrieval results.

Second, the density near -1 is significantly reduced, demonstrating that ParetoRAG decreases the cases where correct answers are not retrieved, thus improving the retrieval comprehensiveness. Lastly, the distribution curves of ParetoRAG (blue lines) are smoother compared to Naive RAG (red lines), particularly in the mid-to-high ranking regions (e.g., 40% to 80%). This indicates that ParetoRAG stabilizes the performance of retrievers and reduces erroneous distributions.

5.2.3 Impact of wider top k size

Since the input size of ParetoRAG at the top 30 is similar to that of Naive RAG at the top 10 (more details can be seen in Appendix C), we set the Top 10 performance of Naive RAG as the baseline. We then evaluate the performance of ParetoRAG in the top 10, top 20, and top 30 settings to investigate the impact of different top k configurations on model performance. As shown in Figure 7, our key observations are as follows:

Although Naive RAG can achieve high document coverage, they often include a large amount of irrelevant information, which can interfere with the accuracy of LLM when answering questions. In contrast, with the fine-grained retrieval approach of ParetoRAG, although the recall rate is relatively lower under the same top k settings (e.g., Top 10), the accuracy of the language model's responses is significantly improved. By more precisely selecting sentences relevant to the question, ParetoRAG effectively reduces the interference of irrelevant content, allowing the language model to focus more on processing key information and lowering the inference complexity. Ultimately, this fine-grained retrieval strategy helps the model find the correct answer more efficiently, improving the overall quality and efficiency of the responses.

5.2.4 Complementary effect of ParetoRAG

In this section, we evaluate the impact of ParetoRAG on robustly trained models, which are fine-tuned for the NQ and HotpotQA datasets respectively. These models are trained to enhance robustness against irrelevant context. As shown in Figure 4, in NQ, under the Top 10 retrieval setting, the accuracy improved from 44.80% to 48.20% (+3.4%) when using ANCE. In HotpotQA, with input word length limited to 400 words, the accuracy increases from 23.3% to 25.3% (+2.0%). These results demonstrate that ParetoRAG can enhance performance in addition to robustly trained models.

While robust training improves the model's resilience to noisy contexts, it may still struggle with redundant or dense information in tasks involving long texts or multi-hop reasoning. ParetoRAG mitigates this limitation by reducing redundancy and increasing information density through sentence-level representations, allowing the model to focus more on relevant content, thereby serving as a valuable complement to robustly trained models.

The complementary effect between ParetoRAG and robust training LLM indicates that combining these two approaches can further optimize retrieval and generation quality. Future work could explore integrating ParetoRAG with other training techniques to further enhance its performance across broader scenarios.

6 Conclusion

In this work, we propose ParetoRAG, an unsupervised framework that enhances RAG systems through sentence-level optimization guided by the Pareto principle. By decomposing paragraphs into sentences and dynamically re-weighting critical content while preserving contextual coherence, ParetoRAG achieves dual improvements in retrieval precision and generation quality without requiring additional training or API resources. Extensive experiments demonstrate its effectiveness: the framework reduces token consumption by 70% while improving the accuracy and fluency of the answers in diverse datasets, LLMs, and retrievers. Our analysis further reveals synergistic effects when integrating ParetoRAG with robustly trained language models, suggesting enhanced generalization capabilities. This study not only validates the viability of resource-efficient sentence-level refinement for RAG systems but also opens avenues for exploring hybrid methodologies that com-

bine retrieval-augmented mechanisms with adaptive training strategies.

7 Limitation

While ParetoRAG demonstrates promising results in improving retrieval-augmented generation, it is important to acknowledge several potential limitations that could be addressed in future work. First, the sentence-level decomposition and re-weighting approach may weaken the complex cross-sentence logic or narrative connections within paragraphs, especially in tasks requiring multi-step reasoning or long-range semantic coherence (such as story generation or scientific argumentation). The local focus on key information might lead to a loose overall structure, which could impact the quality of the generated content. Second, when dealing with longer documents, ParetoRAG faces challenges related to segmenting the text and optimizing it at the sentence level. Breaking down long texts into sentences for individual optimization might not effectively preserve the global structure and logical flow of the document. Lastly, while ParetoRAG has been tested on open-domain QA datasets, it has not yet been applied to more specialized domains, such as law or medicine, which could be explored in future work.

References

OpenAI Josh Achiam and etl Adler. 2023. GPT-4 Technical Report.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *The Twelfth International Conference on Learning Representations*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2025. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, volume 38 of *AAAI'24/IAAI'24/EAAI'24*, pages 17754–17762. AAAI Press.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. Dense X Retrieval: What Retrieval Granularity Should We Use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, Miami, Florida, USA. Association for Computational Linguistics.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making Sentences Stand-Alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-Inference: Exploiting Large Language Models for Interpretable Logical Reasoning. In *The Eleventh International Conference on Learning Representations*.

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*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Yang, Xiaojun Chen, and Ruifeng Xu. 2024. Enhancing Noise Robustness of Retrieval-Augmented Language Models with Adaptive Adversarial Training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10028–10039, Bangkok, Thailand. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML'20*, pages 3929–3938. JMLR.org.

Chao-Wei Huang and Yun-Nung Chen. 2024. FactAlign: Long-form Factuality Alignment of Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16363–16375, Miami, Florida, USA. Association for Computational Linguistics.

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 Trans. Inf. Syst.*

Taeho Hwang, Soyeong Jeong, Sukmin Cho, SeungYoon Han, and Jong C. Park. 2024. DSLR: Document Refinement with Sentence-Level Re-ranking and Reconstruction to Enhance Retrieval-Augmented Generation. *Preprint*, arXiv:2407.03627.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised Dense Information Retrieval with Contrastive Learning. *Trans. Mach. Learn. Res.*

Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O. Arik. 2024. Long-context llms meet rag: Overcoming challenges for long inputs in rag. *Preprint*, arXiv:2410.05983.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2023. RealTime QA: What's the Answer Right Now? In *Thirty-Seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2023. SuRe: Summarizing Retrievals using Answer Candidates for Open-domain QA of LLMs. In *The Twelfth International Conference on Learning Representations*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Jinhyuk Lee, Alexander Wettig, and Danqi Chen. 2021. Phrase Retrieval Learns Passage Retrieval, Too. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3661–3672, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353, Bangkok, Thailand. Association for Computational Linguistics.

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. 2020a. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 9459–9474, Red Hook, NY, USA. Curran Associates Inc.

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. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 9459–9474, Red Hook, NY, USA. Curran Associates Inc.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2024. RA-DIT: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yanming Liu, Xinyue Peng, Xuhong Zhang, Weihao Liu, Jianwei Yin, Jiannan Cao, and Tianyu Du. 2024b. RA-ISF: Learning to Answer and Understand from Retrieval Augmentation via Iterative Self-Feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4730–4749, Bangkok, Thailand. Association for Computational Linguistics.

Hongyin Luo, Tianhua Zhang, Yung-Sung Chuang, Yuan Gong, Yoon Kim, Xixin Wu, Helen Meng, and James Glass. 2023. Search Augmented Instruction

Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3717–3729, Singapore. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2024. Gorilla: Large Language Model Connected with Massive APIs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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. In *Advances in Neural Information Processing Systems*, volume 34, pages 4816–4828. Curran Associates, Inc.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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.

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. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, D. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, A. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, M. Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *ArXiv*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *Preprint*, arXiv:2007.00808.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Xunjian Yin, Baizhou Huang, and Xiaojun Wan. 2023. ALCUNA: Large Language Models Meet New Knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1397–1414, Singapore. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024. Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14672–14685, Miami, Florida, USA. Association for Computational Linguistics.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. RAFT: Adapting language model to domain specific RAG. In *First Conference on Language Modeling*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. FreeLB: Enhanced Adversarial Training for Natural Language Understanding. In *International Conference on Learning Representations*.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2024. PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models. *Preprint*, arXiv:2402.07867.

A System Prompt

The following is the system prompt used to let a LLM generate an answer without any information:

> You are a helpful assistant. Answer the question as concisely as possible, using only the specific phrase, entity, or number that directly answers the question. Within five words.
> **Query:** [question]
> **Short Answer:**

The following is the system prompt used in RAG to let a LLM generate a NQ answer based on the given context:

> You are a knowledgeable assistant tasked with answering questions based on the Natural Questions dataset. Each question is accompanied by contexts extracted from Wikipedia. Answer the question by providing only the specific phrase, entity, or number that directly answers the question. Within five words.
> **Contexts:** [context]
> **Query:** [question]
> **Short Answer:**

The following is the system prompt used in RAG to let a LLM generate a MS answer based on the given context:

> You are a knowledgeable assistant tasked with answering questions based on the MS-marco dataset. Answer the question given the information in those contexts. Answer the question in a single, brief sentence.
> **Contexts:** [context]
> **Query:** [question]
> **Answer:**

The following is the system prompt used in RAG to let a LLM generate a HotpotQA based on the given context:

> You are a knowledgeable assistant tasked with answering questions based on the HotPotQA dataset. Each question is accompanied by contexts extracted from Wikipedia. Answer the question as concisely as possible, using only the specific phrase, entity, or number that directly answers the question. Within five words.
> **Contexts:** [context]
> **Query:** [question]
> **Short Answer:**

B Analysis of Information Utility in Retrieved Documents

In this section, we conduct a detailed analysis to quantify the useful/relevant information ratio in documents retrieved by our approach compared to baseline methods. This analysis helps demonstrate the effectiveness of ParetoRAG in preserving relevant information while reducing redundancy.

We leverage HotpotQA's sentence-level annotations to measure the percentage of sentences containing supporting facts (i.e., information directly useful for answering questions) within the retrieved contexts. For fair comparison, we controll the total token count to approximately 450 tokens for both NaiveRAG and ParetoRAG across different retriever models.

Retriever	NaiveRAG	ParetoRAG
ANCE	20.79%	26.08%
Contriever	21.74%	28.82%
DPR	21.87%	27.65%

Table 2: Useful information ratio comparison across different retrievers

Table 2 presents the results of this analysis, showing the percentage of useful information (sentences containing supporting facts) in the retrieved contexts.

These results quantitatively demonstrate that ParetoRAG more effectively identifies and preserves relevant information while reducing redundancy, regardless of the underlying retriever used. Across all three retriever models tested, ParetoRAG consistently achieves a higher ratio of useful information in the retrieved context, with improvements ranging from 5.29 to 7.08 percentage points over NaiveRAG.

This analysis provides additional evidence for ParetoRAG's ability to optimize the information density of retrieved contexts, which contributes to the performance improvements observed in our main experimental results.

C Calculation of Document Retrieval Ratio for Input Size Consistency

In the original corpus, the average token count per paragraph for the top 30 retrieved paragraphs is 80.8575 tokens. After applying the Sentence-RAG, the average token count per paragraph for the top 30 retrieved paragraphs decreases to 23.85 tokens. Therefore, to maintain consistency in the total token count of input paragraphs, Sentence-RAG would theoretically need to retrieve the top $80.8575/23.85 \approx 34$ paragraphs to match the token scale of the top 10 paragraphs retrieved by naive-RAG. While strict calculations suggest retrieving approximately 34 paragraphs, selecting 30 paragraphs strikes a balance between maintaining the validity of experimental results and ensuring clarity and simplicity in presentation.

D Statistics of Datasets

	NQ	HotpotQA	MS-MARCO
Passages	2,681,468	5,233,329	8,841,823
Ours	9,320,506	12,425,366	30,137,968

Table 3: Statistics of text unit counts before and after ParetoRAG encoding.

Table 3 shows statistics of text unit counts before and after ParetoRAG encoding.

E Statistics of Models

All model weights are derived from Hugging Face, which were used without additional training. In the following, we list the specific hugging face model names corresponding to the weights used in the experiment:

E.1 Model Weights

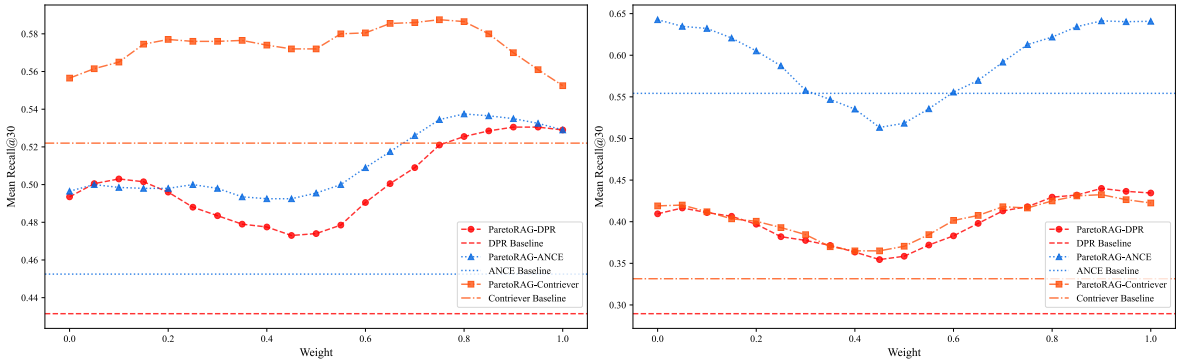
- **DPR**:
  - `facebook/dpr-question_encoder-multiset-base`
  - `facebook/dpr-ctx_encoder-multiset-base`
- **Contriever**:
  - `facebook/contriever`
- **ANCE**:
  - `sentence-transformers/msmarco-roberta-base-ance-firstp`
- **RECOMP**:
  - `fangyuan/nq_abstractive_compressor`
  - `fangyuan/hotpotqa_abstractive_compressor`
- **Llama2**:
  - `meta-llama/Llama-2-7b-chat-hf`
  - `meta-llama/Llama-2-13b-chat-hf`
- **Viccuna**:
  - `lmsys/vicuna-7b-v1.3`
  - `lmsys/vicuna-13b-v1.3`
- **RetRobust**:
  - `Ori/llama-2-13b-peft-nq-retrobust`
  - `Ori/llama-2-13b-peft-hotpotqa-retrobust`

E.2 Model Hyperparameter

The model's configuration is as follows: `max_output_tokens` is set to 150, limiting the maximum number of tokens in the generated output; `temperature` is set to 0.1, which controls the randomness of the generation process, ensuring more deterministic and focused outputs; `seed` is fixed at 100 to ensure reproducibility of the results across different runs; and `per_gpu_batch_size` is set to 16, specifying the number of samples processed per GPU in each batch during training or inference.

F Evaluation Metrics

Following the experimental setup in (Asai et al., 2023), we use MAUVE(Pillutla et al., 2021) and ROUGE-L (Lin, 2004) as evaluation metrics for long-form generation. For short-form generation, we use accuracy (ACC). For each question, if the standard answer is contained within the generated answer and the length of the generated answer is less than or equal to 15, it is counted as 1. Here's a brief explanation of the evaluation metrics:



(a) HotpotQA (b) MS-MARCO

Figure 8: Impact of Core Sentence Weight on Recall across HotpotQA and MS-Marco Dataset.

- Accuracy: Measures the percentage of correct predictions made by the model. It's a basic metric that indicates how well the model is performing on a classification or question-answering task.

- ROUGE: Evaluates text summarization or generation by comparing the overlap between generated text and reference text. It focuses on recall, ensuring the generated text captures key information from the reference. Common variants include ROUGE-N (n-gram overlap) and ROUGE-L (longest common subsequence).

- MAUVE: Assesses text generation quality by comparing the distribution of generated text to reference text in an embedding space. It uses divergence measures to evaluate semantic and structural alignment, making it particularly useful for open-ended tasks like story or dialogue generation.

G Impact of core sentence weight

As shown in the Figure 8, HotpotQA and MS-MARCO generally follow the trends analyzed in Section 5.2.1, where increasing the core sentence weight typically improves recall performance. However, there are noticeable differences in the details of recall variations between these two datasets. Specifically, in the MS-MARCO dataset, the recall rates of ParetoRAG-DPR and ParetoRAG-ANCE decrease significantly in the core sentence weight range of 0.3 to 0.6. This phenomenon can be attributed to the following key factors:

G.1 Differences in Retrieval Model Training Approaches

- **Asynchronous Global Index Updates versus Local Contrastive Learning**: ANCE relies on *asynchronous global index updates*, whereas DPR and Contriever adopt *local contrastive learning* with positive and negative samples. This distinction makes ANCE more susceptible to contextual noise when the core sentence and contextual sentence weights are close (0.3–0.6), leading to less precise retrieval and subsequently lower recall performance.

- In contrast, DPR and Contriever primarily depend on *local contrastive learning* during training. Since they do not suffer from the lag introduced by global index updates, their recall rate decline in the 0.3–0.6 weight range is relatively less pronounced.

G.2 Differences in Task Types and Information Requirements

- **MS-MARCO (Single-hop QA)**: In this dataset, queries typically require matching a specific *core sentence* in the text to retrieve the correct answer, while paragraph-level information may contain substantial redundant content. Consequently, when the core sentence weight falls within the 0.3–0.6 range, paragraph-level information introduces interference in the retrieval process, leading to a decline in recall performance.

- **HotpotQA (Multi-hop QA)**: In contrast, HotpotQA involves multi-hop reasoning, where queries require integrating information from

Table 4: Computational Resource Comparison

Method	Processing Time	GPU Memory Usage	Total Processing Time	Estimated Cost*
Dense X	51.39s per batch	37.57 GB peak	50.03 hours	$150.09
ParetoRAG	46.53s per batch	35.08 GB peak	4.02 hours	$12.06
NaiveRAG	49.00s per batch	42.87 GB peak	1.23 hours	$3.69

*Cost estimation based on cloud GPU pricing of $3.00 per hour for A100 GPU.

multiple paragraphs to derive the final answer. As a result, even when the core sentence weight is relatively low, the model can still leverage other paragraphs to improve retrieval performance. Therefore, unlike MS-MARCO, HotpotQA does not exhibit a sharp decline in recall within the 0.3–0.6 weight range.

H Impact of Model Size on Accuracy with Varying Top K in ParetoRAG

As show in Figure 7, for smaller models (such as Vicuna-7b), their ability to process a large number of documents is weaker, leading to a faster decline in accuracy as the Top K increases. However, this decline is not due to a lack of retrieval quality by ParetoRAG, but rather because smaller models are unable to fully utilize the richer information provided. On the other hand, for larger models (such as Vicuna-13b), their greater parameter size and reasoning capability enable them to handle more information within a larger scope. As a result, even when the Top K is increased to a certain extent (e.g., Top 20 or Top 30), they still maintain high accuracy. Notably, larger Top K settings (e.g., Top 20) outperform Top 10 and the baseline, demonstrating that ParetoRAG can provide richer information retrieval, offering more effective context for language models.

I Cost Analysis

We conduct a comprehensive analysis of the computational resources required for each method in our study. All experiments were performed on an NVIDIA A100 80GB GPU.

Method-Specific Configurations. ParetoRAG and NaiveRAG are both configured with a batch size of 512 documents. This larger batch size is possible due to their efficient encoding mechanisms that process documents in a single forward pass.

Dense X is limited to a batch size of 256 documents due to its more memory-intensive processing requirements. The primary performance bottleneck for Dense X is the document atomization process, which requires the `propositionizer-wiki-flan-t5-large` model. This model breaks down documents into atomic propositions, a computationally expensive operation that significantly increases processing time compared to the other methods.

The memory usage patterns reflect these differences in processing approaches. While all methods are configured to use approximately 35-43GB of GPU memory at peak, Dense X exhibits a distinctive pattern of high memory usage during generation followed by significant reduction post-generation, indicating its intensive but ephemeral computational requirements during the atomization process.

These configuration differences explain the substantial variation in total processing time across methods, with Dense X requiring significantly more time to process the complete dataset despite similar per-batch processing durations.