M-RAG: Making RAG Faster, Stronger, and More Efficient

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) has become a widely adopted paradigm for enhancing the reliability of large language models (LLMs). However, RAG systems are sensitive to retrieval strategies that rely on text chunking to construct retrieval units, which often introduce information fragmentation, retrieval noise, and reduced efficiency. Recent work has even questioned the necessity of RAG, arguing that long-context LLMs may eliminate multi-stage retrieval pipelines by directly processing full documents. Nevertheless, expanded context capacity alone does not resolve the challenges of relevance filtering, evidence prioritization, and isolating answer-bearing information. To this end, we proposed **M-RAG**, a novel CHUNK-FREE retrieval strategy. Instead of retrieving coarse-grained textual chunks, M-RAG extracts structured, k-v decomposition meta-markers, with a lightweight, intent-aligned *retrieval key* for retrieval and a context-rich *information value* for generation. Under this setting, M-RAG enables efficient and stable query-key similarity matching without sacrificing expressive ability. Experimental results on the LongBench subtasks demonstrate that M-RAG outperforms chunk-based RAG baselines across varying token budgets, particularly under low-resource settings. Extensive analysis further reveals that M-RAG retrieves more answer-friendly evidence with high efficiency, validating the effectiveness of decoupling retrieval representation from generation and highlighting the proposed strategy as a scalable and robust alternative to existing chunk-based methods.

1 Introduction

As a promising approach for improving the factual reliability and adaptability of large language models (LLMs), Retrieval-Augmented Generation (RAG) has recently attracted increasing attention. By dynamically incorporating external, domain-specific knowledge during the generation process, RAG alleviates the inherent static-knowledge limitations of LLMs, reduces hallucination, and has demonstrated strong applicability in knowledge-intensive domains (Louis et al., 2024; Wu et al., 2025; Zhang et al., 2023).

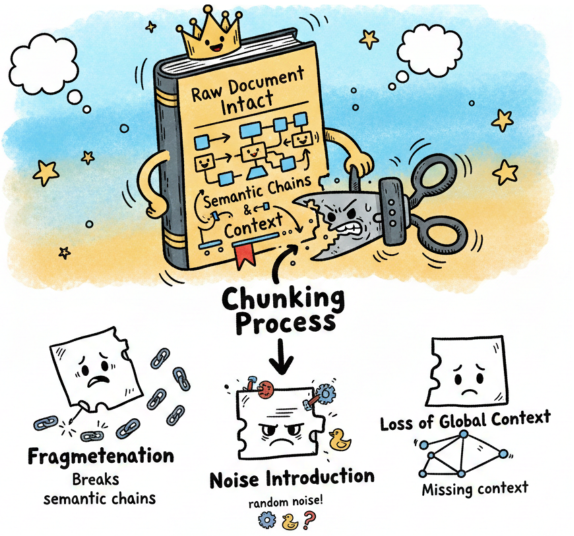


Figure 1: Insight from our concerns: information fragmentation and structural loss caused by RAG chunking. (Figure generated by Nano Banana (Team, 2025a))

Typically, a RAG system follows a multi-stage pipeline: given an input query, the system first retrieves a set of candidate documents or passages from an external knowledge base using dense or sparse retrieval models. These candidates are then segmented into fixed-length or semantically defined *chunks*, followed by optional re-ranking to filter relevant sets. Consequently, selected chunks with queries are fed into the LLMs to conduct the generation process (Lewis et al., 2020; Guu et al., 2020; Kamradt, 2024; Gao et al., 2024).

Although effective, the designation of RAG tightly couples retrieval granularity with generation context construction, making RAG systems sensitive to the choice of chunking strategy. For instance, fixed-length segmentation often fragments coherent semantic units and introduces irrelevant noise. While adaptive chunking methods such as semantic-based approaches like PIC (Wang et al., 2025c) or trained mixtures like MoC (Zhao et al., 2025) attempt to dynamically aggregate sentences, they remain fundamentally constrained by local context windows and struggle to preserve document-level causal, temporal, or hierarchical structures. More critically, chunking constitutes an inherently lossy preprocessing operation. By predefining information boundaries and granularity prior to retrieval, it limits the capacity of LLMs to reason over complete contextual dependencies, as illustrated in Figure 1. Recent studies, such as DOS RAG (Laitenberger et al., 2025), suggest that retaining the original document structure and abandoning complex chunking strategies can lead to more robust performance, which is increasingly feasible with the availability of long-context LLMs. Thus, a critical issue is naturally raised: *if model "see" the entire document, is the RAG process still necessary?*

The answer should be affirmative: long-context capability does not eliminate the need for RAG, but rather shifts its role from overcoming memory extension limitations to enabling efficient, relevance-aware knowledge utilization (Hsieh et al., 2024; Wang et al., 2024; Asai et al., 2024; Nuengsigkapian, 2025; Wang et al., 2025a). While extended context windows relax hard token constraints for LLMs, they do not inherently provide mechanisms for relevance filtering or evidence prioritization. Recent work CFIC (Qian et al., 2024) supports this view by proposing a chunk-free, in-context retrieval approach that identifies evidence text via autoregressive decoding over document representations. Although our method adopts a different problem-solving perspective, both lines of work acknowledge that RAG explicitly structures the interaction between external knowledge and generation, allowing selective access to relevant information and controlled knowledge injection. As a result, *RAG remains essential as a principled framework, but asking for more efficient and robust knowledge-retrieval mechanisms.*

To this end, we propose M-RAG, a novel CHUNK-FREE retrieval strategy within the RAG framework that fundamentally departs from text chunking. Instead, M-RAG leverages off-the-shelf LLMs (e.g., DeepSeek-V3.2 (DeepSeek-AI, 2025)) to extract structured marker entries directly from complete documents. Each marker entry, a.k.a. meta-marker, is decoupled into two complementary components: a lightweight *retrieval key* k, which serves as an efficient "anchor" for similarity matching when retrieved, and a context-rich *information value* v, which preserves the corresponding content for future generation. Unlike existing paragraph- or chunk-level retrieval, this key-value decomposition enables lightweight retrieval while providing stable, intent-aligned access to contextual information. Our main contributions are summarized as:

- We propose M-RAG, a novel and principled CHUNK-FREE retrieval strategy within the RAG framework that fundamentally departs from text chunking and rethinks knowledge injection in the long-context era.
- We design a decoupled K-V MARKER that enables lightweight and efficient retrieval without sacrificing contextual fidelity and, to our knowledge, is the first retrieval strategy to explicitly separate retrieval representations from generation content within the RAG system.
- Experiments on LongBench subtasks demonstrate strong performance, and in-depth analyses indicate that the proposed retrieval strategy offers a scalable and robust alternative to chunk-based RAG methods.

2 Methodology

In this section, we introduce M-RAG, a novel retrieval strategy that eliminates traditional text chunking in conventional RAG. As shown in Figure 2, the whole workflow consists of two stages: 1) Marker Extractor, which drives concise, semantically and retrieval-optimized meta-markers from documents, is the main contribution of this work, and 2) meta-marker retrieval and response generation, where the most query-relevant meta-markers are selected under a fixed token budget to guide response generation. We next detail the design of each module.

2.1 Marker Extractor

With simplicity and effectiveness in mind, M-RAG is designed as a model-agnostic, drop-in replacement for existing RAG systems.

Conception. Intuitively, a user query reflects an underlying information need associated with their core intent. Considering the same information need may be expressed through many semantically equivalent queries, effective retrieval must be robust to substantial surface-form variability. In this

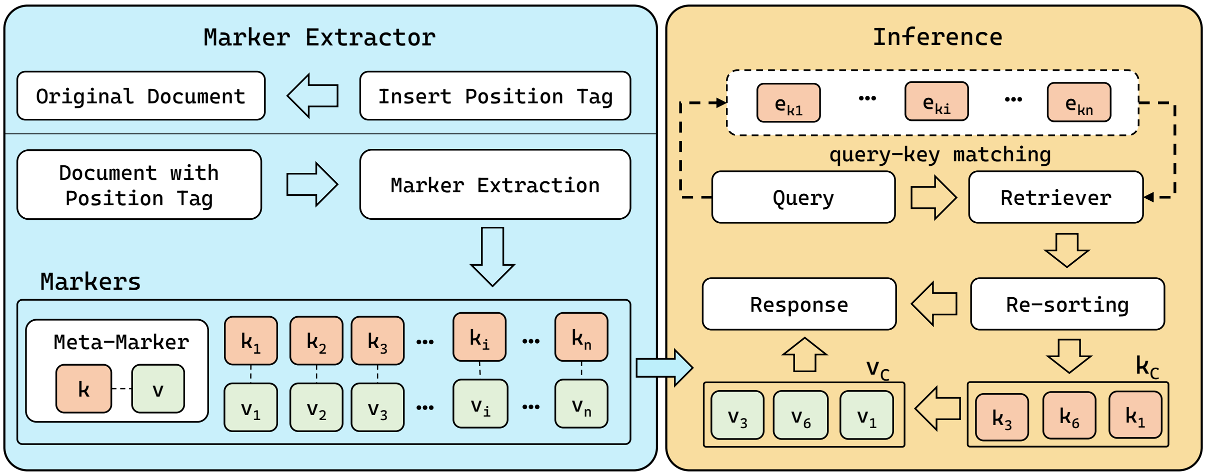


Figure 2: The overall architecture of M-RAG.

context, aligning queries to fixed or semantically aggregated text chunks is unnecessarily heavy. We therefore argue that, *rather than requiring alignment with textual chunks, effective retrieval should match user queries to lightweight semantic cues that capture the core intent of the information need.* This perspective motivates the design of M-RAG, which operates intent-aligned semantic cues as retrieval units in place of textual chunks.

Specifically, we first derive a set of meta-markers $\boldsymbol{m}_i$ to form an n-markers entry $M_{\mathcal{D}} = \{\boldsymbol{m}_1, \cdots, \boldsymbol{m}_n\}$, where each $\boldsymbol{m}_i$ is a self-contained textual string with its length l_i, and both the n and l_i are unconstrained. Herein, each meta-marker $\boldsymbol{m}_i$ is decoupled into two complementary components:

- $\boldsymbol{k}_i$, a lightweight *retrieval key*, serves as an intent-aligned semantic cue optimized for matching anticipated queries and maximizing retrieval recall.
- $\boldsymbol{v}_i$, a context-rich *information value*, preserves the substantial factual or relational content and provides grounding for generation.

Thus, a complete meta-marker $\boldsymbol{m}_i$ can be formed as $\boldsymbol{m}_i = \boldsymbol{k}_i \oplus \boldsymbol{v}_i$, where $\oplus$ is textual concatenation.

Implementation. Before extraction, a position tag will be inserted into every token-based segment (e.g., 128 tokens) in each document $\mathcal{D}$ to make use of the order information and enable precise coverage validation post-extraction, denoted as *[Paragraph N]* with N starting from 0. Subsequently, we formulate a prompt to instruct an off-the-shelf LLM to derive the marker $M_{\mathcal{D}}$ for each corpus $\mathcal{D}$. The prompt dynamically specifies an expected minimum meta-marker count, calculated as the document's total token count divided by the segment size, to guide the LLM toward sufficient granularity and coverage. A representative prompt template is shown in Table 1 (see Appendix C for details).

The output of the prompted-LLM is then parsed to extract individual meta-markers, each consisting of its *retrieval key* $\boldsymbol{k}_i$, an *information value* $\boldsymbol{v}_i$, and the associated *paragraph indices* indicating the source paragraphs in the original corpus (denoted as *[paragraph N]*). To ensure extract quality, we define coverage as the proportion of paragraph segments that appear in the *paragraph indices* of at least one meta-marker. If coverage falls below a threshold (0.95 in this work), the extraction is retried using the same prompt for up to three attempts. If all attempts fail, we select the output with the highest coverage and apply a conservative fallback strategy: each uncovered paragraph is converted into a meta-marker with $\boldsymbol{k}_i = \boldsymbol{v}_i$ set to the paragraph content, and the corresponding *paragraph indices* assigned.

Formally, the marker extractor is defined as:

$$M_{\mathcal{D}} = \mathrm{LLM}(\mathcal{D}, P), \tag{1}$$

where P refers to the prompt and $M_{\mathcal{D}} = \{(\boldsymbol{k}_i, \boldsymbol{v}_i)\}_{i=1}^{n}$ is the set of extracted meta-markers for document $\mathcal{D}$. Subsequently, each *retrieval key* $\boldsymbol{k}_i$ is encoded to a dense vector to form an "anchor" tensor $\boldsymbol{e}_K = \{\boldsymbol{e}_{k_i}, \cdots, \boldsymbol{e}_{k_n}\}$, where $\boldsymbol{e}_{k_i} = \frac{1}{|\boldsymbol{k}_i|}\mathbb{E}(\boldsymbol{k}_i) \in \mathbb{R}^{d_m}$, $\mathbb{E}(\cdot)$ refers to any embedding model, such as BAAI/bge-m3 (Chen et al., 2024a), and d_m denotes the embedding dimension. Thus, anchor $\boldsymbol{k}_i$ is prepared for similarity-based retrieval, and its corresponding context-rich information $\boldsymbol{v}_i$ is reserved for generation.

Table 1: Prompt Structure for Marker Extraction

Component	Content
Extraction Rules	1. **Fine-Grained Segmentation - STRICT PARAGRAPH LIMIT**: - Each meta-marker MUST use 1-3 paragraphs ONLY. NO EXCEPTIONS. - Prefer 1-2 paragraphs per meta-marker for best quality. - If you need to cover 10 paragraphs, create 5-10 separate meta-markers, NOT one big meta-marker. - Overlap is REQUIRED: Reuse the same paragraphs across multiple meta-markers to ensure coverage. 2. **Resolution and Binding**: - No pronouns: Replace with explicit names where ambiguity exists (e.g., 'he' → 'Keith Nichol'). - Bind attributes: For people (birth date, roles, relationships), organizations (location, founding year), works (author, year), events (dates, figures), stats (exact values). 3. **v (Information Block)** - FOCUSED AND CONCISE: - A self-contained paragraph (200-300 words) covering ONE SPECIFIC aspect or fact. - Keep it narrow and atomic - don't try to cover too much. - Include necessary context but maintain a tight focus. - Extract v first, then generate k to match it. 4. **k (Summary Query)** - GENERATE ONE DETAILED QUESTION THAT SUMMARIZES v: - Generate ONE question that BOTH summarizes the content AND serves as a retrieval query. - The question can be longer and more detailed - include key entities, names, dates, and concepts mentioned in v. - k should capture the essence of v, acting as both a summary and a search query. - The question should be specific enough to match this block's content and rich enough to stand alone. - Think: "What detailed question would both describe AND help retrieve this information?" 5. **paragraph_indices (Source Paragraph Indices)** - MAXIMUM 3 PARAGRAPHS: - List the paragraph numbers (0-based) where this meta-marker's information comes from. - ABSOLUTE MAXIMUM: 3 paragraphs per meta-marker. Arrays with 4+ elements are INVALID. - PREFERRED: 1-2 paragraphs per meta-marker for highest quality. - Paragraphs MUST be reused across multiple meta-markers to achieve full coverage.
Reference Examples	{examples}

2.2 Retrieval and Generation

When retrieval, given a user query $\boldsymbol{q}$, M-RAG first generates its embedding $\boldsymbol{e}_q = \mathbb{E}(\boldsymbol{q})$. Then, retrieval identifies candidate meta-markers via similarity matching over anchors, namely,

$$\hat{\boldsymbol{M}}_c = \sigma(\boldsymbol{e}_q, \boldsymbol{e}_K), \tag{2}$$

where σ is a similarity function, such as cosine similarity. M-RAG further follows previous works to leverage HNSW (Malkov and Yashunin, 2018) for efficient approximate nearest neighbor search.

Next, we dynamically select the top-ranked meta-markers $\boldsymbol{M}_c = \{\boldsymbol{k}_c, \boldsymbol{v}_c\}$ from $\hat{\boldsymbol{M}}_c$ until the cumulative token of $\boldsymbol{v}_c$ exceeds a predefined token budget B (computed via tiktoken). For post-filling during context construction, we apply two re-sorting strategies on *retrieval keys*: position-based ordering, which follows the original document order according to paragraph indices, and similarity-based ordering, which ranks *retrieval keys* by their semantic relevance to the user query. Due to the space limitations, the impact of two sorting strategies is reported in Appendix D.

After *retrieval key* is obtained, the generation stage follows the standard RAG pipeline. The corresponding *information values* are incorporated into generation by prompting an instruction-tuned LLM with the retrieved content, as: *Using the following retrieved information [$\boldsymbol{v}_c$] answer the query [$\boldsymbol{q}$].*

2.3 Advantage Statement

M-RAG decouples retrieval representation from generation content, enabling lightweight and robust retrieval while preserving context-rich information for generation. By retrieving lightweight, intent-aligned semantic keys instead of coarse-grained textual chunks, M-RAG avoids an inherent granularity mismatch in existing RAG systems, where fine-grained user queries are matched against large and semantically heterogeneous text chunks. In addition, our marker extraction process is guided by a zero-shot or few-shot prompting paradigm, designed as a drop-in pre-process module without task-specific samples, reducing prompt complexity and computational overhead. Together, these designs yield an efficient and robust retrieval-augmented framework, particularly well-suited for long-context settings.

Concretely, relative to chunking-based RAG, M-RAG yields several key advantages: 1) by holistically processing documents and avoiding predefined textual segmentation, the proposed strategy mitigates chunk-induced fragmentation and better preserves long-range interdependencies that are often disrupted by segmented retrieval, 2) the explicit decomposition between retrieval anchor and generation content enhances query-key alignment while reducing irrelevant context injection, thereby achieving higher retrieval effectiveness and cleaner augmented contexts for generation. Moreover, M-RAG is designed as a model-agnostic, drop-in replacement for existing RAG pipelines, enabling seamless integration without altering model architectures or retrieval infrastructures, while remaining compatible with diverse prompting and deployment settings.

3 Experiment

3.1 Setup

Benchmark. We evaluate M-RAG on the QA subtasks of LongBench (Bai et al., 2024), a multitask benchmark designed for long-context understanding. Specifically, we use NarrativeQA (Kočiský et al., 2018) and Qasper (Dasigi et al., 2021) for single-hop QA, and 2WikiMultihopQA (Ho et al., 2020) for multi-hop QA, each consisting of 200 samples. These subtasks span single- and multi-document scenarios, with average context lengths exceeding 3k tokens.

Baselines. We compare against representative chunking-based RAG methods: Fixed-Size RAG that segments documents per 128 tokens, Semantic RAG that uses similarity for dynamic chunking (Kamradt, 2024), PIC RAG that employs summary-sentence similarity for adaptive grouping (Wang et al., 2025c), and recent DOS RAG that preserves original document structure without multi-stage processing (Laitenberger et al., 2025). All models use the same BAAI/bge-m3 retriever with token budget constraints (128×1, 128×3, 128×5 tokens) for fair comparison.

Implementation Details. In M-RAG, marker extraction is conducted using DeepSeek-V3.2 (DeepSeek-AI, 2025), with a unified prompt template across all benchmarks as specified in Table 4. For retrieval, we adopt the BAAI/bge-m3 embedding model (Chen et al., 2024a). Generation is performed using the instruction-tuned Qwen3-30B-A3B-Instruct-2507 (Team, 2025b). To ensure deterministic behavior and minimize stochastic variation, both the marker extractor and the generation model are configured with the temperature set to 0. All experiments are repeated five times to account for the variability introduced. Additional details and information on source code availability are provided in Appendix A.

3.2 Main Results

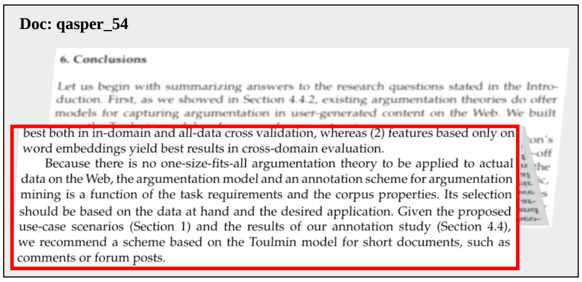
Table 2 reports the end-to-end QA performance of different RAG paradigms across LongBench subtasks under varying token budgets. Overall, M-RAG demonstrates consistently strong performance, with particularly clear advantages under low-budget settings, where chunk-based methods suffer from fragmentation and retrieval noise. On NarrativeQA at 128×1, M-RAG achieves a score of 0.0736, substantially outperforming Fixed-Size, Semantic, and PIC by 11.5%, 19.3%, and 19.1%, respectively, and ranking second only to DOS. M-RAG's advantage further expands as the budget increases, while it surpasses all baselines at 128×5. Similar trends are observed on Qasper, where M-RAG achieves the best performance at 128×1 and maintains competitive results at higher budgets, reflecting the benefit of marker-based retrieval in preserving information when only limited context can be injected. On 2WikiMultihopQA, M-RAG outperforms baselines in most configurations. Across all experiment settings, M-RAG achieves top-1 or top-2 results in 7 out of 9 settings, empirically validating the key–value decoupling design, which reduces retrieval noise while preserving expressive generation context.

3.3 Analysis

Document Coverage. To evaluate the comprehensiveness and quality of marker extraction in M-RAG, we analyze document coverage rates in Table 3. As shown, both zero-shot and few-shot prompting achieve consistently high document coverage across all benchmarks, with average coverage rates exceeding 99.8%, and fallback usage remaining below 1%. The results directly support that the marker extraction strategy almost always identifies at least one marker covering each paragraph, and the unconstrained generation of markers does not lead to systematic content omission. We also provide a random instance to further figure out the reason when fallback is active, which is illustrated in Figure 3. This case is from document Qasper_54 (Habernal and Gurevych, 2017), where fallback is triggered in the Conclusions section due to insufficient semantic content for meta-maker extraction. In addition, while few-shot prompting slightly reduces variance in some cases, the zero-shot prompting achieves comparable mean coverage across benchmarks, indicating that marker

Table 2: The performance comparison, where the best gain is highlighted in **bold** while the second best is underlined. M-RAG reports performance under zero-shot and few-shot† prompting, with both position sorting (-P) and similarity sorting (-S), respectively.

	NarrativeQA			**Qasper**			**2WikiMultihopQA**		
	128×1	128×3	128×5	128×1	128×3	128×5	128×1	128×3	128×5
Fixed-Size	.0660 ± .0000	.0952 ± .0002	.1094 ± .0001	.1316 ± .0000	.1908 ± .0000	.2425 ± .0006	.2220 ± .0000	.3239 ± .0038	.3832 ± .0023
Semantic	.0617 ± .0000	.1200 ± .0002	.1449 ± .0009	.1654 ± .0008	.2482 ± .0007	.2683 ± .0003	.1875 ± .0005	.3125 ± .0030	.3831 ± .0013
PIC	.0618 ± .0009	.1092 ± .0001	.1226 ± .0005	.1581 ± .0006	**.2530** ± **.0036**	**.2838** ± **.0021**	.2066 ± .0000	.3236 ± .0015	.3947 ± .0026
DOS	**.0843** ± **.0000**	.1191 ± .0000	.1398 ± .0001	.1643 ± .0009	.2269 ± .0009	.2635 ± .0024	.2023 ± .0000	.3255 ± .0004	**.4110** ± **.0028**
M-RAG-P	.0684 ± .0006	**.1279** ± **.0018**	.1459 ± .0022	.1436 ± .0018	.2317 ± .0013	.2456 ± .0013	.2134 ± .0016	.3229 ± .0030	.3494 ± .0057
M-RAG-S	.0657 ± .0025	.1214 ± .0014	**.1476** ± **.0031**	.1390 ± .0037	.2308 ± .0032	.2655 ± .0055	.2170 ± .0077	.3237 ± .0069	.3582 ± .0035
M-RAG†-P	.0736 ± .0029	.1247 ± .0030	.1381 ± .0018	.1806 ± .0043	.2370 ± .0037	.2693 ± .0061	.2232 ± .0039	**.3301** ± **.0003**	.3948 ± .0006
M-RAG†-S	.0644 ± .0004	.1208 ± .0029	.1357 ± .0013	**.1820** ± **.0025**	.2224 ± .0041	.2690 ± .0015	**.2321** ± **.0015**	.3290 ± .0000	.3560 ± .0006
Rank of M-RAG	2	1	1	1	3	2	1	1	2



Doc: qasper_54

6. Conclusions

Let us begin with summarizing answers to the research questions stated in the Introduction. First, as we showed in Section 4.4.2, existing argumentation theories do offer models for capturing argumentation in user-generated content on the Web. We built

best both in in-domain and all-data cross validation, whereas (2) features based only on word embeddings yield best results in cross-domain evaluation.

Because there is no one-size-fits-all argumentation theory to be applied to actual data on the Web, the argumentation model and an annotation scheme for argumentation mining is a function of the task requirements and the corpus properties. Its selection should be based on the data at hand and the desired application. Given the proposed use-case scenarios (Section 1) and the results of our annotation study (Section 4.4), we recommend a scheme based on the Toulmin model for short documents, such as comments or forum posts.

Figure 3: The sole fallback instance from document Qasper_54. Red-box highlights the retreated content.

extraction generalizes well under zero-shot settings. These results demonstrate that M-RAG can reliably extract comprehensive markers without relying on pre-defined chunking or task-specific exemplars, with fallback mechanisms activated only in rare edge cases.

Table 3: Statistics on average coverage rates under zero-shot (ZS) and few-shot (FS) prompting.

Benchmark	Prompting	Avg. Cover.	Variance	Fallback (%)
NarrativeQA	ZS	99.80	0.4040	0.7
	FS	99.87	0.1948	0.2
Qasper	ZS	99.93	0.2316	0.9
	FS	99.90	0.2776	0.1
2WikiMultihopQA	ZS	99.80	0.5379	0.0
	FS	99.84	0.3648	0.5

Retrieval Efficiency. Figure 4 isolates the time consumption of similarity matching between user queries and retrieval units, i.e., keys vs. chunks, directly reflecting the efficiency of the retrieval interface. First, M-RAG consistently achieves the lowest retrieval latency across all benchmarks, supporting our conception that matching queries against lightweight *retrieval keys* is substantially more efficient than matching against chunk-based representations. Specifically, user queries are short and intent-focused, whereas chunk-based retrieval strategies require computing similarity against long, semantically heterogeneous text spans. This granularity mismatch increases both embedding computation and the compaction of dense content. In contrast, M-RAG performs similarity matching exclusively over compact *retrieval keys*, yielding lower per-candidate computation and faster convergence during similarity search. Moreover, from a scalability perspective, the unstable retrieval latency of Fixed-size, PIC, and DOS across different benchmarks stems from their reliance on data-dependent chunking or paragraph-level retrieval units, whereas size, count, and semantic heterogeneity vary significantly with document structure and domain characteristics. Semantic chunking alleviates this issue by heuristically improving segment coherence, but its retrieval efficiency remains sensitive to domain-dependent chunking behavior. In comparison, M-RAG bounds retrieval complexity by design, performing similarity matching exclusively over compact, intent-aligned *retrieval keys*, which yields more stable and predictable retrieval latency across benchmarks. These results indicate that M-RAG efficiently reduces the cost of query-key similarity matching, validating our design choice of decoupling compact *retrieval keys* from context-rich *information values*.

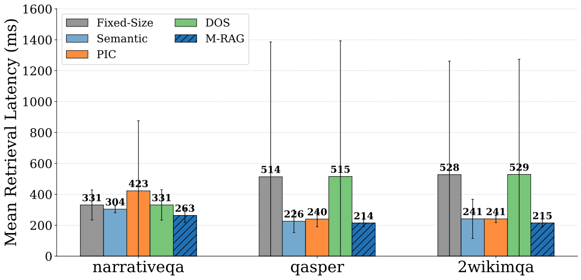


Figure 4: Retrieval time comparison.

Token Length of k, v. To evaluate whether M-RAG enforces a clear and stable separation between *retrieval key* and *information value* efficiency and contextual richness, we showcase statistics under zero-shot (ZS) and few-shot (FS) prompting in Figure 5. We have the following observations. (1) Across all benchmarks and prompting regimes, *retrieval keys* remain consistently compact (approximately 19-20 tokens), while *information values* are substantially longer (about 50-65 tokens), indicating that the proposed strategy successfully decouples retrieval representation from generation content. Towards *retrieval keys*, the standard deviation is relatively low, echoing our claim that *retrieval keys* serve as lightweight semantic cues to make similarity search efficient and robust to surface-form variation. Values consistently contain 2.5-3× more tokens than keys, which showcases their ability to take advantage of sufficient context for further generation. (2) The k-v decomposition exhibits a consistent yet domain-adaptive pattern. *Retrieval keys* are concise across all benchmarks, showcasing that, regardless of document style or task complexity, k reliably captures concise, intent-aligned semantic cues. In contrast, *information values* illustrate domain-sensitive variability. NarrativeQA favors shorter values to reflect local related-context; Qasper requires longer values, which we conjecture M-RAG needs to encapsulate more information for academic explanations; and 2wikimqa yields the longest values, which we acknowledge that this phenomenon is consistent with its multi-hop and cross-document demands. Importantly, this adaptive behavior of v does not affect the expression of k, supporting our conception that the decomposition design of k-v in M-RAG fully unleashes the advantages of RAG, which preserves retrieval efficiency while flexibly allocating context where task complexity demands it. In brief, the clear length asymmetry between *retrieval keys* and *information values* suggests the effectiveness of our core idea on k-v decomposition. Moreover, comparable statistics under ZS and FS prompting demonstrate strong cross-domain generalization of our marker-extraction strategy, supporting that M-RAG preserves expressive capacity without domain-specific adaptation.

Case Study. We conduct case studies using sampled documents from each benchmark. The Figure 6 (a) visualizes three samples using t-SNE visualizations, including user query and the corresponding markers generated by M-RAG. An obviously consistent pattern can be observed first: *retrieval keys* from localized, query-aligned neighborhoods, whereas *information values* present a broader and more diffuse region of the embedding space. This geometric separation directly supports our main conception of k-v decomposition, where *retrieval keys* act as compact semantic "anchor" for retrieval, and *information values* retain richer contextual content for the downstream generation without any influencing query-key similarity matching. Furthermore, key distributions are compact around the query across all benchmarks. Despite the inherent complexity of multi-hop reasoning in 2wikimqa, the query is still closely matched to a cluster of keys, while values are dispersed, reflecting the need to retain cross-document relations, but retrieval remains focused and stable. Few-shot prompting serves as a regularizer on geometry, inducing more compactness of keys, especially in 2wikimqa. This aligns with the above quantitative findings, demonstrating that few-shot prompting is capable of reducing variance, further improving consistency and stability of retrieval.

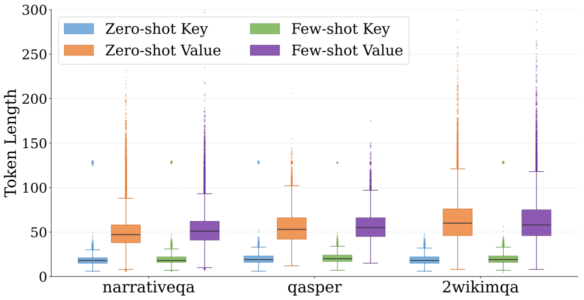
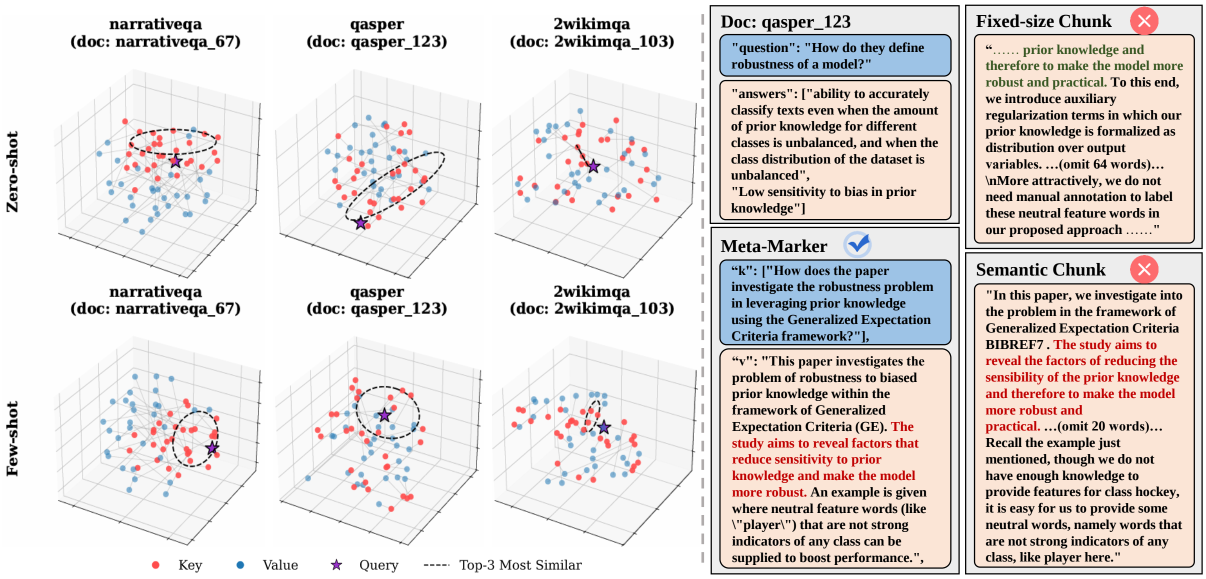


Figure 5: Token length of k, v across benchmarks, where boxes show quartiles, whiskers extend to 1.5 IQR, and outliers are marked.

A concrete case randomly selected from Qasper is provided in Figure 6 (b). The user query is a definition-seeking question regarding the term "robustness". Typically, the retrieval target is a sentence-level conceptual statement, e.g., "what robustness means in this paper", not a procedural description. While the chunks retrieved by Fixed-Size and Semantic are information-rich, they do not explicitly state the definition of robustness. Rather than being incorrect (the highlighted red text is accurate, while greens are fragmentation), these passages are too coarse for the user queries, as they mix multiple themes such as regularization, practical robustness, illustrative examples, and reference details. We therefore conjecture that these embeddings capture blended topics, resulting in insufficient semantic focus for effective retrieval and downstream generation. In contrast, our meta-marker retrieval matches the query to an intent-aligned key and returns a value that explicitly en-



(a) t-SNE visualization of meta-marker

Doc: qasper_123

"question": "How do they define robustness of a model?"

"answers": ["ability to accurately classify texts even when the amount of prior knowledge for different classes is unbalanced, and when the class distribution of the dataset is unbalanced",
"Low sensitivity to bias in prior knowledge"]

Meta-Marker ✓

“k": ["How does the paper investigate the robustness problem in leveraging prior knowledge using the Generalized Expectation Criteria framework?"],

“v": "This paper investigates the problem of robustness to biased prior knowledge within the framework of Generalized Expectation Criteria (GE). The study aims to reveal factors that reduce sensitivity to prior knowledge and make the model more robust. An example is given where neutral feature words (like \"player\") that are not strong indicators of any class can be supplied to boost performance.",

Fixed-size Chunk ✗

“…… prior knowledge and therefore to make the model more robust and practical. To this end, we introduce auxiliary regularization terms in which our prior knowledge is formalized as distribution over output variables. …(omit 64 words)… \nMore attractively, we do not need manual annotation to label these neutral feature words in our proposed approach ……"

Semantic Chunk ✗

"In this paper, we investigate into the problem in the framework of Generalized Expectation Criteria BIBREF7 . The study aims to reveal the factors of reducing the sensibility of the prior knowledge and therefore to make the model more robust and practical. …(omit 20 words)… Recall the example just mentioned, though we do not have enough knowledge to provide features for class hockey, it is easy for us to provide some neutral words, namely words that are not strong indicators of any class, like player here."

(b) A concrete case

Figure 6: Case studies.

codes the definitional statement, as the highlighted sentences in Figure 6. As a result, using the *information value* for generation is predictably more answerable with reduced irrelevant content.

4 Related Work

RAG has fundamentally enhanced LLMs by incorporating external knowledge retrieval. This effectively mitigates hallucinations and knowledge obsolescence (Lewis et al., 2020; Guu et al., 2020). Foundational works established dense retrieval and end-to-end integration for knowledge-intensive tasks, paving the way for rapid advancements. Subsequent surveys have mapped its evolution (Fan et al., 2024; Gao et al., 2024), with the latest shift toward agentic RAG. Agentic RAG employs autonomous agents for planning, reflection, tool use, and collaboration in dynamic workflows (Singh et al., 2025; Li et al., 2025).

As RAG scales, document chunking emerges as a key bottleneck. Fixed or rule-based methods disrupt contextual integrity, add noise, and fail to maintain global coherence or fine-grained relations like causality and hierarchy (Wang et al., 2025b). Metrics such as Boundary Clarity (BC) and Chunk Stickiness (CS) quantify these issues. These metrics correlate with downstream QA performance and expose limitations in traditional and semantic chunking (Zhao et al., 2025). Structure-preserving baselines outperform multi-stage pipelines (Laitenberger et al., 2025). Semantic chunking proves unstable and costly on real documents, favoring fixed methods in constrained settings (Qu et al., 2025). Recent refinements stay within segmentation, including adaptive PIC (Wang et al., 2025c), trained MoC (Zhao et al., 2025), graph-enhanced LightRAG (Guo et al., 2025), HeteRAG (Yang et al., 2025), and DOS RAG (Laitenberger et al., 2025). Finer units like propositions improve recall for long-tail queries (Chen et al., 2024b). Yet, these inherit fragmentation and often demand training or extra computation. More recently, the necessity of chunking has been questioned, both CFIC (Qian et al., 2024) and this work advocating chunk-free RAG pipelines, highlighting a promising direction for RAG systems.

5 Conclusion

This work revisits the chunk-based retrieval design under the RAG framework and proposes a novel model-agnostic, drop-in CHUNK-FREE retrieval strategy, M-RAG. This strategy extracts structured markers from original documents and creatively decouples retrieval units from generation content, enabling lightweight retrieval while providing stable, intent-aligned access to contextual information for downstream generation. Extensive qualitative and quantitative analyses show the strong performance of M-RAG, suggesting that the proposed strategy offers a scalable and robust alternative to chunk-based RAG, and provides a feasible direction for future retrieval design in RAG systems. Moreover, the explicit k-v separation highlights a valuable extension: improving retrieval accuracy by refining k alone, making targeted key refinement a high-leverage and low-cost strategy.

Limitations

Marker extraction process relies on LLMs, which may introduce hallucination issues and yield *information value* that are not fully consistent with the original documents. We partially mitigate this risk through paragraph indices, a coverage threshold, and the fallback mechanism, the impact of hallucination is not yet fully quantified. Although no such phenomenon has been identified through our observations, future work will involve manual validation of extracted markers and constructing a dedicated public benchmark for it. Moreover, due to computational resource constraints, we do not compare M-RAG with graph-based RAG methods (e.g., GraphRAG (Edge et al., 2025) or LightRAG (Guo et al., 2025)), which construct knowledge graphs and may hold advantages in multi-hop reasoning tasks. Importantly, our goal is to propose a novel, CHUNK-FREE retrieval strategy that rethinks how retrieval units are constructed more effectively. Evaluating the complementary strengths of M-RAG and graph-based methods in large-scale settings is an interesting work for the future. Additionally, we utilize a single LLM for marker extraction and do not explore the impact of alternative models (e.g., GPT or Gemini families) on extraction quality, which may introduce model-specific biases.

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*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. *Preprint*, arXiv:2308.14508.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024b. 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.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

DeepSeek-AI. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *Preprint*, arXiv:2512.02556.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

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.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. LightRAG: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761, Suzhou, China. 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*, ICML'20. JMLR.org.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational linguistics*, 43(1):125–179.

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

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Greg Kamradt. 2024. Semantic chunking.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Alex Laitenberger, Christopher D Manning, and Nelson F. Liu. 2025. Stronger baselines for retrieval-augmented generation with long-context language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32547–32557, Suzhou, China. 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. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Yangning Li, Weizhi Zhang, Yuyao Yang, Wei-Chieh Huang, Yaozu Wu, Junyu Luo, Yuanchen Bei, Henry Peng Zou, Xiao Luo, Yusheng Zhao, Chunkit Chan, Yankai Chen, Zhongfen Deng, Yinghui Li, Hai-Tao Zheng, Dongyuan Li, Renhe Jiang, Ming Zhang, Yangqiu Song, and Philip S. Yu. 2025. A survey of RAG-reasoning systems in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12120–12145, Suzhou, China. Association for Computational Linguistics.

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.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22266–22275.

Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.

Cattalyya Nuengsigkapian. 2025. Hifi-rag: Hierarchical content filtering and two-pass generation for open-domain rag. *Preprint*, arXiv:2512.22442.

Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. 2024. Grounding language model with chunking-free in-context retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1311, Bangkok, Thailand. Association for Computational Linguistics.

Renyi Qu, Ruixuan Tu, and Forrest Sheng Bao. 2025. Is semantic chunking worth the computational cost? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2155–2177, Albuquerque, New Mexico. Association for Computational Linguistics.

Chidaksh Ravuru, Sagar Srinivas Sakhinana, and Venkataramana Runkana. 2024. Agentic retrieval-augmented generation for time series analysis. *Preprint*, arXiv:2408.14484.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *Preprint*, arXiv:2501.09136.

Gemini Team. 2025a. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Qwen Team. 2025b. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Dingmin Wang, Qiuyuan Huang, Matthew Jackson, and Jianfeng Gao. 2024. Retrieve what you need: A mutual learning framework for open-domain question answering. *Transactions of the Association for Computational Linguistics*, 12:247–263.

Dingmin Wang, Ji Ma, and Shankar Kumar. 2025a. Retrieval augmented question answering: When should llms admit ignorance? *Preprint*, arXiv:2512.23836.

Yongjie Wang, Yue Yu, Kaisong Song, Jun Lin, and Zhiqi Shen. 2025b. When retrieval succeeds and fails: Rethinking retrieval-augmented generation for llms. *Preprint*, arXiv:2510.09106.

Zhitong Wang, Cheng Gao, Chaojun Xiao, Yufei Huang, Shuzheng Si, Kangyang Luo, Yuzhuo Bai, Wenhao Li, Tangjian Duan, Chuancheng Lv, Guoshan Lu, Gang Chen, Fanchao Qi, and Maosong Sun. 2025c. Document segmentation matters for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8063–8075, Vienna, Austria. Association for Computational Linguistics.

Wikipedia. 2025. Wikipedia:size of wikipedia. `https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia`. Online; accessed December 2025.

Kevin Wu, Eric Wu, Kevin Wei, Angela Zhang, Allison Casasola, Teresa Nguyen, Sith Riantawan, Patricia Shi, Daniel Ho, and James Zou. 2025. An automated framework for assessing how well llms cite relevant medical references. *Nature Communications*, 16(1):3615.

Peiru Yang, Xintian Li, Zhiyang Hu, Jiapeng Wang, Jinhua Yin, Huili Wang, Lizhi He, Shuai Yang, Shangguang Wang, Yongfeng Huang, and 1 others. 2025.

Heterag: A heterogeneous retrieval-augmented generation framework with decoupled knowledge representations. *arXiv preprint arXiv:2504.10529*.

Boyu Zhang, Hongyang Yang, Tianyu Zhou, Muhammad Ali Babar, and Xiao-Yang Liu. 2023. Enhancing financial sentiment analysis via retrieval augmented large language models. In *Proceedings of the fourth ACM international conference on AI in finance*, pages 349–356.

Jihao Zhao, Zhiyuan Ji, Zhaoxin Fan, Hanyu Wang, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2025. MoC: Mixtures of text chunking learners for retrieval-augmented generation system. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5172–5189, Vienna, Austria. Association for Computational Linguistics.

APPENDIX

A Availability

The source code for M-RAG, including the marker extraction pipeline, retrieval implementation, and benchmark scripts, is publicly available at `https://anonymous.4open.science/r/M-RAG-5904/` under the MIT License. The repository includes dependencies and instructions for reproducing experiments on LongBench. Datasets are sourced from the official LongBench repository (Bai et al., 2024). All experiments were conducted on a server with NVIDIA A800 GPUs (80GB), using DeepSeek-V3.2 API for marker generation and Qwen3-30B-A3B-Instruct-2507 for answer generation.

B AI assistants statement

In the preparation of this manuscript, we utilized AI assistants for specific tasks. We used Claude Code to assist with programming, and Nano Banana to generate Figure 1. We also used ChatGPT to polish the grammar.

C Prompt

C.1 Prompt Structure for Marker Extraction

The full prompt template for marker extraction is detailed in Table 4. It is dynamically populated with the document content (content) and expected marker count (calculated as total tokens / segment size, e.g., 128). For few-shot prompting, the Reference Examples section is filled with benchmark-specific examples (see Appendix C.2 for details).

C.2 Few-Shot Examples for Marker Extraction

For few-shot prompting in marker extraction, we use benchmark-specific examples to guide the LLM. These examples are inserted into the {examples} placeholder in the prompt template (Table 4). Below are the examples used for each benchmark, presented in Table 5.

These examples are selected to represent typical document structures and query types in each benchmark, ensuring the LLM generates markers with appropriate granularity and coverage.

C.3 Prompt for Answer Generation

To generate answers from retrieved meta-markers, we use the following zero-shot prompt template,

Table 4: Prompt Structure for Marker Extraction

Component	Content
Role Description	You are a marker extraction expert tasked with creating high-quality meta-markers from the document to replace standard text chunks in RAG systems. Your goal is to generate MANY fine-grained markers to ensure complete document coverage with high-quality retrievable information blocks.
Input Format	The document contains multiple segments, each marked with [Paragraph N] where N starts from 0.
Critical Constraint	Each marker's paragraph_indices array MUST contain AT MOST 3 paragraph numbers. Examples of VALID paragraph_indices: ✓ [0] - ONE paragraph (BEST) ✓ [0, 1] - TWO paragraphs (GOOD) ✓ [0, 1, 2] - THREE paragraphs (MAXIMUM ALLOWED) Examples of INVALID paragraph_indices: × [0, 1, 2, 3] - FOUR paragraphs (TOO MANY - REJECT) × [0, 1, 2, 3, 4, 5] - SIX paragraphs (TOO MANY - REJECT) IF YOU NEED MORE PARAGRAPHS, CREATE MULTIPLE MARKERS INSTEAD.
Key Goals	1. **Zero Information Loss**: Preserve every detail, name, date, number, relationship, and fact exactly. 2. **Fine-Grained Extraction**: Create MANY small, focused markers rather than few large ones. Each marker covers 1-3 paragraphs MAXIMUM. 3. **Retrieval Optimization**: Generate a search query (k) that would retrieve this information block (v). 4. **Complete Coverage**: Generate MANY markers (50-100+ for long documents) to cover ALL paragraphs. Paragraphs MUST be reused across multiple markers to achieve full coverage.
Extraction Rules	1. **Fine-Grained Segmentation - STRICT PARAGRAPH LIMIT**: - Each meta-marker MUST use 1-3 paragraphs ONLY. NO EXCEPTIONS. - Prefer 1-2 paragraphs per meta-marker for best quality. - If you need to cover 10 paragraphs, create 5-10 separate meta-markers, NOT one big meta-marker. - Overlap is REQUIRED: Reuse the same paragraphs across multiple meta-markers to ensure coverage. 2. **Resolution and Binding**: - No pronouns: Replace with explicit names where ambiguity exists (e.g., 'he' → 'Keith Nichol'). - Bind attributes: For people (birth date, roles, relationships), organizations (location, founding year), works (author, year), events (dates, figures), stats (exact values). 3. **v (Information Block)** - FOCUSED AND CONCISE: - A self-contained paragraph (200-300 words) covering ONE SPECIFIC aspect or fact. - Keep it narrow and atomic - don't try to cover too much. - Include necessary context but maintain a tight focus. - Extract v first, then generate k to match it. 4. **k (Summary Query)** - GENERATE ONE DETAILED QUESTION THAT SUMMARIZES v: - Generate ONE question that BOTH summarizes the content AND serves as a retrieval query. - The question can be longer and more detailed - include key entities, names, dates, and concepts mentioned in v. - k should capture the essence of v, acting as both a summary and a search query. - The question should be specific enough to match this block's content and rich enough to stand alone. - Think: "What detailed question would both describe AND help retrieve this information?" 5. **paragraph_indices (Source Paragraph Indices)** - MAXIMUM 3 PARAGRAPHS: - List the paragraph numbers (0-based) where this meta-marker's information comes from. - ABSOLUTE MAXIMUM: 3 paragraphs per meta-marker. Arrays with 4+ elements are INVALID. - PREFERRED: 1-2 paragraphs per meta-marker for highest quality. - Paragraphs MUST be reused across multiple meta-markers to achieve full coverage.
Output Format	(valid JSON): { "marker": [{ "v": "Focused paragraph with specific details on one narrow topic...", "k": ["The query that would retrieve this information"], "paragraph_indices": [0, 1] }, ...] }
Critical Requirements	1. Valid JSON only; ALL double quotes inside strings MUST be escaped with backslash. Example: "substance into n̈othingness¨ NOT "substance into "nothingness"" 2. **PARAGRAPH LIMIT - STRICTLY ENFORCED**: - Each marker's paragraph_indices array MUST contain 1, 2, or 3 elements ONLY. - Arrays with 4 or more elements are INVALID and will be REJECTED. - If you need more paragraphs, CREATE SEPARATE MARKERS. 3. **QUANTITY REQUIREMENT**: - Based on this document's length, you should generate approximately {expected_marker_count} markers. - This is the MINIMUM - generating MORE markers is encouraged for better coverage. - Do NOT be conservative with marker count. 4. **OVERLAP REQUIREMENT**: - Each paragraph should appear in MULTIPLE markers (1-3 times on average). - Create different markers focusing on different aspects of the same paragraphs. - Example: Paragraph 0 might appear in marker about person A, marker about event B, marker about location C. 5. **OTHER REQUIREMENTS**: - k must be a question (interrogative sentence) that summarizes v. - Ensure complete coverage: every paragraph must appear in at least one marker's paragraph_indices.
Input Document	INPUT DOCUMENT: {content}
Reference Examples	{examples}

Table 5: Few-Shot Examples for Each Benchmark

Benchmark	**Example**
NarrativeQA	{ "v": "Soames declared passionately that if he could be projected into the future, into the reading-room, just for one afternoon, he would sell himself body and soul to the devil for that privilege. Soames imagined seeing pages and pages in the catalogue under ’Soames, Enoch’ with endless editions, commentaries, prolegomena, and biographies. At that moment, Soames was interrupted by a sudden loud crack of the chair at the next table. A stranger, not an Englishman but one who knew London well, addressed Soames directly by name and revealed himself to be the devil, offering to make a deal with Soames.", "k": ["According to the deal Soames made, how would he find out whether his poetic talent would have been recognized in the future?"], "paragraph_indices": [69, 71] }
Qasper	{ "v": "Deep Neural Networks (DNN) have been widely employed in industry for solving various NLP tasks such as text classification, sequence labeling, and question answering. However, when engineers apply DNN models to address specific NLP tasks, they often face challenges that hinder productivity and result in less optimal solutions. Before designing the NLP toolkit NeuronBlocks, the authors conducted a survey among engineers to identify typical personas and their challenges. There are several general-purpose deep learning frameworks such as TensorFlow, PyTorch and Keras which offer huge flexibility in DNN model design. However, building models under these frameworks requires a large overhead of mastering framework details, and therefore higher level abstraction to hide framework details is favored by many engineers.", "k": ["How do the authors evidence the claim that many engineers find it a big overhead to choose from multiple frameworks, models and optimization techniques?"], "paragraph_indices": [0, 3] }
2WikiMultihopQA	{ "v": "The Bag Man (also known as Motel or The Carrier) is a 2014 neo-noir crime thriller film directed by David Grovic, based on an original screenplay by James Russo. The film stars John Cusack, Rebecca Da Costa, Crispin Glover, Dominic Purcell, Robert De Niro, and Sticky Fingaz. The Bag Man premiered on February 28, 2014, in New York and Los Angeles. Una prostituta al servizio del pubblico e in regola con le leggi dello stato (literally ’A prostitute serving the public and complying with the laws of the state’, also known as Prostitution Italian Style) is a 1970 Italian comedy-drama film written and directed by Italo Zingarelli.", "k": ["Which film came out first, Una Prostituta Al Servizio Del Pubblico E In Regola Con Le Leggi Dello Stato or The Bag Man?"], "paragraph_indices": [18, 28] }

Table 6: Prompt Structure for Answer Generation

Component	Content
User Question	User Question: {question}
Retrieved Content	Retrieved Marker: {marker_text}
Instruction	You MUST answer strictly based on the marker.
Answering Rules	Answering Rules (MUST FOLLOW): 1. Output ONLY the final answer: **one entity name, one date, one location or Yes/No.** 2. Output MUST be **JUST the answer text** with no extra words. 3. **NO full sentences.** 4. **NO explanations.** 5. **NO reasoning.** 6. **NO repeating the question.** 7. If the metadata does not contain enough information, output exactly: "Insufficient information".
Response Constraint	Your entire response must be ONLY the answer text.
Output Starter	Answer:

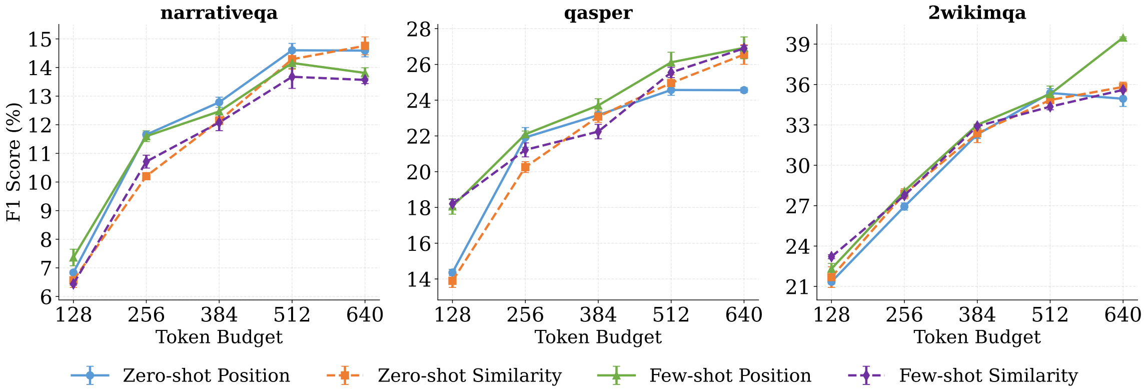


Figure 7: Performance comparison of sorting strategies and prompting regimes across benchmarks (token budgets: 128×1, 128×2, 128×3, 128×4, 128×5).

detailed in Table 6. This template enforces concise, evidence-based outputs to minimize hallucinations and facilitate evaluation. It is dynamically populated with the query {question} and retrieved content {marker_text}.

For baselines (e.g., Fixed-Size or Semantic chunking), the prompt is adapted by replacing {marker_text} with the corresponding chunk text, ensuring identical generation constraints. This adaptation maintains fairness in end-to-end QA evaluation, as the rules prioritize factual extraction over verbose reasoning.

C.4 Prompt Sensitivity Analysis

To demonstrate that M-RAG is robust to variations in the extraction prompt, we evaluate three different prompts using the DeepSeek-V3.2 model (DeepSeek-AI, 2025) on LongBench QA subtasks. We selected Qasper as the representative benchmark for this analysis due to its structured academic content, which tests the framework's sensitivity in knowledge-intensive scenarios. We vary the token budget retrieval (BAAI/bge-m3) and use Qwen3-30B-A3B-Instruct-2507 as the generator, measuring end-to-end QA performance in F1 scores under few-shot prompting with position sorting. The prompts are:

- **Prompt A**: The original prompt as detailed in Table 4.
- **Prompt B**: A variation where the Output Format and Critical Requirements sections in Table 4 are swapped in position.
- **Prompt C**: A variation where the Key Goals section in Table 4 is moved below the Critical Requirements section.

Results in Table 7 show minimal variance, confirming insensitivity to prompt phrasing, as long as core instructions (extraction count, focus on retrievable info) are preserved.

Table 7: Prompt Sensitivity on Representative LongBench QA Subtask (F1 for Qasper). Results are based on **Few-shot Position Sort** for different Token Budget retrievals.

Token Budget	Prompt A	Prompt B	Prompt C
128x1	0.1806 ± 0.0043	0.1866 ± 0.0004	0.1734 ± 0.0003
128x2	0.2210 ± 0.0018	0.2226 ± 0.0006	0.2105 ± 0.0022
128x3	0.2370 ± 0.0037	0.2370 ± 0.0004	0.2318 ± 0.0000
128x4	0.2612 ± 0.0056	0.2534 ± 0.0009	0.2571 ± 0.0010
128x5	0.2693 ± 0.0061	0.2560 ± 0.0019	0.2711 ± 0.0006

D Impact of Sorting Strategies

To examine the effects of sorting strategies (position-based vs. similarity-based) under different prompting regimes (zero-shot vs. few-shot), we conducted experiments across the three benchmarks. Figure 7 illustrates the F1 scores for each combination over five token budgets (128×1 to 128×5).

Focusing on sorting strategies, position-based sorting (PS) proves superior in structure-sensitive benchmarks. In 2WikiMultihopQA, PS achieves an average +0.9% over similarity-based sorting (SS), with few-shot PS reaching 39.48% at 640 tokens compared to 35.60% for SS. This advantage arises from preserving document order, which supports multi-hop reasoning by maintaining sequential and hierarchical dependencies. Similarly, in Qasper, PS dominates mid-to-high budgets (e.g., 26.93% for few-shot PS at 640 tokens), aligning with the benchmark's technical structure where order aids in contextual coherence.

Conversely, similarity-based sorting (SS) excels in open-ended narratives like NarrativeQA, particularly at higher budgets. For instance, zero-shot SS attains 14.76% at 640 tokens, a +1.2% gain over PS, as it prioritizes query relevance over order, enabling more flexible evidence selection in less structured content. These patterns underscore the task-dependent efficacy of sorting in M-RAG: PS enhances reasoning in ordered domains, while SS boosts relevance in narrative-heavy tasks, offering adaptability for diverse applications.