URAG 2.0: An Agentic Dual Retrieval Framework for Enhanced Reasoning in RAG-Based QA Systems

Long S. T. Nguyen[1,2], Quynh T. N. Vo[1,2], Thi T. Nguyen[3], and Tho T. Quan[1,2(✉)]

[1] URA Research Group, Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam
qttho@hcmut.edu.vn

[2] Vietnam National University, Ho Chi Minh City, Vietnam

[3] Agri Sung Joint Stock Company, Ho Chi Minh City, Vietnam

Abstract. Large Language Models (LLMs) have advanced Question-Answering (QA) systems but still suffer from factual errors and limited reasoning when relying solely on implicit knowledge. Retrieval-Augmented Generation (RAG) mitigates these issues by grounding responses in external corpora, yet existing pipelines often depend on a single retrieval channel, which hampers multi-hop reasoning and underutilizes heterogeneous evidence. Graph-based extensions attempt to capture structural relations but remain costly, noisy, and ultimately constrained to one stream. To address these limitations, we propose URAG 2.0, an agentic dual-retrieval framework that extends our original URAG design. URAG 2.0 constructs two complementary indices: Frequently Asked Questions (FAQs) distilled and paraphrastically enriched from documents, and semantically chunked documents refined with context-aware rewriting. At inference, both indices are queried in parallel, and an orchestration layer fuses and ranks evidence before synthesis. Experiments across multiple QA benchmarks demonstrate that URAG 2.0 consistently outperforms advanced RAG baselines in both factual QA and multi-hop reasoning, establishing dual retrieval as a promising direction for building more accurate and explainable QA systems.

Keywords: Question-Answering Systems · Retrieval-Augmented Generation · Multi-Agent Systems · Dual Retrieval · Multi-Hop Reasoning

1 Introduction

The rapid advancement of *Large Language Models* (LLMs) has marked a turning point in natural language processing, demonstrating impressive capabilities in both understanding and generating natural text for *Question-Answering* (QA) systems [1]. Despite these achievements, LLMs still face critical limitations in

W. Buntine et al. (Eds.): SOICT 2025, CCIS 2912, pp. 469–481, 2027.
https://doi.org/10.1007/978-981-92-2584-2_37

factual recall and reasoning over external knowledge, often producing hallucinations in high-stakes domains such as education and healthcare [2,3]. To mitigate these issues, *Retrieval-Augmented Generation* (RAG) has emerged as a promising paradigm [4], coupling LLMs with retrieval modules so that models can ground responses in external knowledge bases. This integration improves factual accuracy, transparency, and source attribution, thereby broadening the applicability of LLMs in domain-specific contexts [5].

The effectiveness of RAG, however, crucially depends on its indexing and retrieval strategies [6,7]. Conventional pipelines typically employ single-stream indexing, where all documents are segmented and stored in a unified index. While this design is adequate for simple factual QA, it struggles with more complex reasoning tasks: relevant evidence may be fragmented across sources, multi-hop reasoning requires synthesizing dispersed information—common in domains such as education—and poorly segmented chunks can introduce redundancy and retrieval errors [8,9]. Although recent work on semantically coherent chunking [10] has shown improvements, indexing strategies explicitly optimized for reasoning remain underexplored. Several RAG variants attempt to address these challenges by augmenting retrieval with structured representations such as graphs [11–14]. Graph-based methods can capture entity or relational dependencies and support certain forms of multi-hop reasoning. Nevertheless, they bring their own limitations: graph construction and maintenance are computationally expensive, edges are often noisy or incomplete, and ultimately these approaches still operate within a single retrieval channel. More importantly, they do not exploit the complementary strengths of different knowledge forms. Our key observation is that concise factual knowledge and detailed contextual evidence play distinct yet reinforcing roles: *Frequently Asked Questions* (FAQs) provide precise answers for factual lookup, while raw documents preserve the richer context needed for multi-hop reasoning. Treating them uniformly in a single channel underutilizes their potential.

This gap highlights the need for *dual retrieval*: the ability to run multiple specialized retrieval channels in parallel and fuse their outputs for generation. Unlike single-stream or graph-augmented pipelines, dual retrieval provides a principled way to integrate complementary evidence sources, improving factual precision while enhancing robustness in multi-hop reasoning.

In this work, we present **URAG 2.0**, an agentic dual-retrieval framework for unified RAG that extends our original URAG design [3]. While URAG 1.0 demonstrated the benefits of retrieval-enhanced QA in domain-specific contexts, URAG 2.0 advances this line by explicitly formalizing dual retrieval and embedding it within a multi-agent architecture. Two complementary indices are constructed: a document-based index built through semantic chunking and context-aware rewriting, and a *Frequently Asked Questions* (FAQ) index distilled and enriched via paraphrastic generation. At inference, both indices are queried in parallel, and an orchestration layer fuses and ranks the retrieved evidence before handing it to a reader for synthesis. By integrating concise factual signals with

detailed contextual information, URAG 2.0 achieves higher factual precision and stronger multi-hop reasoning.

The key contributions of this work are summarized below.

- We proposed URAG 2.0, an agentic dual-retrieval framework that formalizes parallel FAQ- and document-based retrieval with coordinated fusion.
- We designed specialized indexing pipelines—semantic chunking with context-aware rewriting for documents, and FAQ generation with paraphrastic enrichment—to improve retrieval quality and reasoning readiness.
- We conducted comprehensive experiments across multiple QA benchmarks, demonstrating that URAG 2.0 consistently outperforms advanced RAG baselines in both factual QA and multi-hop reasoning.

2 Related Works

2.1 Existing Approaches to RAG and Agent Collaboration

RAG has become a cornerstone for improving the factual accuracy of QA systems by grounding LLM outputs in retrieved evidence [4,6,7]. Standard pipelines typically combine dense retrievers, lexical methods such as BM25, or hybrid strategies with generative models to produce context-aware responses [6]. While effective in many cases, these approaches often assume well-formed queries and degrade when queries are vague or underspecified. To address this, several advanced retrieval-reasoning architectures have been proposed. IRCoT [15] interleaves retrieval and reasoning for multi-step QA; LightRAG [11] and MiniRAG [12] augment indexing with graph structures and semantic-aware topologies; NodeRAG [13] integrates heterogeneous graphs to leverage structured evidence; and HippoRAG [16] employs hierarchical memory to capture long-range dependencies. These models achieve strong benchmark results but rely on high-quality inputs and complex infrastructure, making them difficult to deploy in resource-constrained domains such as educational QA. More recently, the paradigm of Agentic RAG [17] has emerged, where specialized agents collaborate across the workflow. MA-RAG [18] decomposes and sequences complex queries, while MAIN-RAG [19] focuses on filtering and evaluating retrieved passages to reduce noise. Despite their strengths, both graph-based and agentic RAG still operate on a single retrieval stream. URAG 2.0 addresses this limitation by formalizing a dual-retrieval mechanism in which FAQ-based and document-based channels operate in parallel under agentic coordination, thereby combining factual accuracy with scalable multi-hop reasoning.

2.2 Established Methods for Data Indexing

Indexing has long been a foundational component of QA systems, directly shaping retrieval efficiency and accuracy [6,7]. Traditional approaches such as the inverted index [20] provide fast and cost-effective keyword-based retrieval but

lack semantic understanding. In high-stakes contexts such as university admissions, where the same concept (e.g., "tuition fees") may be expressed in multiple ways, these methods often yield incomplete or inaccurate results. The advent of vector-based indexing and RAG [4] reshaped this landscape by mapping text into embeddings stored in vector databases such as FAISS [21], enabling semantic similarity search that better aligns retrieval with generation and reduces hallucination. Nevertheless, Naive RAG pipelines remain prone to retrieval noise, redundancy, and misalignment. Extensions such as Self-RAG [22], Corrective RAG [23], and Speculative RAG [24] enhance reliability but incur higher computational overhead and latency due to iterative feedback loops. To balance efficiency and accuracy, hybrid indexing has been explored. Our prior URAG framework [3], developed for university admission chatbots, exemplifies this approach with a two-tier structure (FAQ-first and document-level) and enrichment mechanisms URAG-F and URAG-D that expand coverage. While effective in practice, its fixed two-tier design remains limited when addressing ambiguous or multi-faceted queries that demand flexible synthesis. URAG 2.0 advances this trajectory by reframing indexing as an agent-driven process: multiple specialized agents handle semantic chunking, FAQ enrichment, embedding, and indexing within a dual-retrieval architecture. This design enables dynamic and adaptive evidence processing, providing both greater flexibility and robustness for QA systems.

3 URAG 2.0

Figure 1 illustrates the architecture of URAG 2.0, an *agentic dual-retrieval framework* that extends the original URAG design. The system comprises two phases, *Indexing* and *Inference*, coordinated by a *Coordinator Agent* within a hierarchical orchestration structure [25].

3.1 Indexing Phase

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ denote the raw document corpus. URAG 2.0 constructs two complementary indices:

FAQ Indexing. An *FAQ Agent* generates question–answer pairs from each document d_i via a transformation function (Eq. 1).

$$\mathcal{F} : d_i \mapsto \{(q_{i1}, a_{i1}), (q_{i2}, a_{i2}), \ldots\}. \tag{1}$$

Because user queries often appear in diverse surface forms, a paraphrastic enrichment function $\mathcal{P}$ expands the generated pairs to increase lexical and semantic coverage. The enriched FAQ repository is defined in Eq. 2.

$$\mathcal{I}_{FAQ} = \bigcup_{i=1}^{n} \mathcal{P}(\mathcal{F}(d_i)). \tag{2}$$

This process ensures that semantically equivalent but lexically different queries can be effectively matched against the FAQ index.

Document Indexing. A *Document Agent* segments each d_i into semantically coherent chunks using a chunking function $\mathcal{C}$ (Eq. 3). Unlike naive token-based splitting, $\mathcal{C}$ enforces a semantic coherence constraint to avoid arbitrarily breaking sentences or arguments.

$$\mathcal{C}(d_i) = \{c_{i1}, c_{i2}, \dots\}, \quad \text{s.t. } \phi(c_{ij}) \geq \tau, \tag{3}$$

where ϕ measures coherence and τ is a predefined threshold.

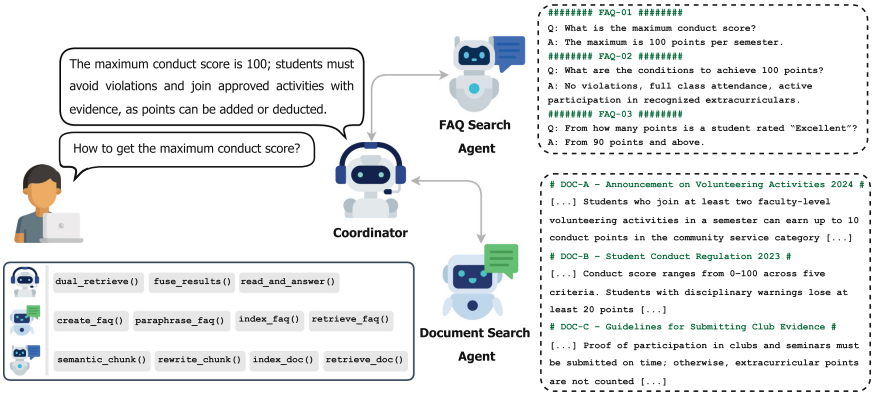

Fig. 1. The architecture of URAG 2.0, featuring dual retrieval from FAQ and document indices under agentic coordination.

Each chunk is then rewritten with global context by a rewriting function $\mathcal{W}$, yielding self-contained units that preserve referential clarity (e.g., resolving pronouns to explicit references). The resulting document index is formalized in Eq. 4.

$$\mathcal{I}_{DOC} = \bigcup_{i=1}^{n} \mathcal{W}(\mathcal{C}(d_i)). \tag{4}$$

Together, $\mathcal{I}_{FAQ}$ and $\mathcal{I}_{DOC}$ constitute a dual knowledge space: the former supports concise factual lookup with paraphrastic robustness, while the latter provides contextually enriched evidence suitable for multi-hop reasoning.

3.2 Inference Phase

Given a user query q, the *Coordinator Agent* dispatches it to both indices. Each retrieval is executed by its corresponding agent: the *FAQ Agent* for $\mathcal{I}_{FAQ}$ and the *Document Agent* for $\mathcal{I}_{DOC}$. Retrieval from the FAQ index is expressed in Eq. 5, and retrieval from the document index in Eq. 6.

$$\mathcal{R}_{FAQ}(q) = \text{Top-}k\big(\text{Retrieve}(q, \mathcal{I}_{FAQ})\big), \tag{5}$$

$$\mathcal{R}_{DOC}(q) = \text{Top-}k\big(\text{Retrieve}(q, \mathcal{I}_{DOC})\big). \tag{6}$$

The orchestration layer then fuses these results (Eq. 7).

$$\mathcal{E}(q) = \text{Fuse}\big(\mathcal{R}_{FAQ}(q), \mathcal{R}_{DOC}(q)\big), \tag{7}$$

where Fuse performs de-duplication, semantic merging, and re-ranking.

Finally, a reader model $\mathcal{L}$ generates the final answer as illustrated in Eq. 8.

$$\hat{a} = \mathcal{L}(q, \mathcal{E}(q)). \tag{8}$$

This dual-retrieval design explicitly leverages the complementary strengths of the FAQ Agent and Document Agent, under the coordination of the Coordinator Agent, leading to higher factual accuracy and stronger multi-hop reasoning than single-index retrieval.

4 Experimentations

We design our experiments to assess the effectiveness of URAG 2.0 across different QA settings.

4.1 Datasets

We evaluate on four datasets spanning two QA settings:

Single-hop QA *SQuAD* [26] is a large-scale English reading comprehension benchmark with over 100,000 crowd-sourced questions on Wikipedia articles, where answers are spans extracted directly from passages. *UIT-ViQuAD* [27] is the Vietnamese counterpart, comprising 23,000 human-generated QA pairs based on 5,109 passages from 174 Wikipedia articles.

Multi-hop QA *HotpotQA* [9] contains 113,000 Wikipedia-based QA pairs that require reasoning over multiple documents. It provides sentence-level supporting facts and includes comparison-style questions, enabling strong supervision and explainability. *VIMQA* [8] is a Vietnamese multi-hop QA dataset with 10,000 human-generated questions. It requires advanced reasoning across multiple paragraphs and provides sentence-level supporting facts, testing systems' ability to extract evidence and perform complex reasoning in a low-resource language.

For evaluation, we uniformly sample 1,000 representative questions from each dataset.

4.2 Evaluation Metrics

We evaluate URAG 2.0 using three complementary metrics:

Accuracy (LLM-as-a-Judge). Instead of relying on surface string comparison, we adopt the *LLM-as-a-Judge* protocol [28]. An external LLM is prompted

to evaluate whether the predicted answer $\hat{a}_i$ is semantically correct with respect to the gold answer a_i, returning a binary decision. Accuracy is then:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\text{LLMJudge}(\hat{a}_i, a_i) = 1\}, \tag{9}$$

where N is the number of questions and $\text{LLMJudge}(\cdot)$ denotes the judgment outcome.

F1 Score. To complement semantic evaluation, we report token-level overlap between prediction and ground truth:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{10}$$

with

$$\text{Precision} = \frac{|\hat{A} \cap A|}{|\hat{A}|}, \quad \text{Recall} = \frac{|\hat{A} \cap A|}{|A|}, \tag{11}$$

where A and $\hat{A}$ are token sets of the gold and predicted answers, respectively.

Exact Match (EM). EM provides a stricter measure by requiring exact string equivalence after normalization:

$$\text{EM} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\text{normalize}(\hat{a}_i) = \text{normalize}(a_i)\}. \tag{12}$$

This combination of LLM-judged accuracy and token-based metrics offers both semantic and surface-level perspectives on system performance.

4.3 Baselines

We evaluate URAG 2.0 by comparing it against the following baselines:

Naive RAG. A basic RAG pipeline using BM25, dense retrieval, and hybrid retrieval with score interpolation [6].

IRCoT + SOTA LLM. A multi-step QA approach that interleaves retrieval with *Chain-of-Thought* (CoT) [29] reasoning, guiding retrieval using intermediate reasoning steps and vice versa [15].

LightRAG. A framework that incorporates graph structures into text indexing and retrieval, enabling both low-level and high-level knowledge discovery for improved contextual relevance [11].

MiniRAG. A lightweight RAG system designed for resource-constrained settings, combining small language models with heterogeneous graph indexing for efficient structured retrieval [12].

NodeRAG. A graph-centric RAG framework that employs heterogeneous graph structures to better integrate structured evidence into multi-hop QA [13].

HippoRAG 2. A retrieval system inspired by hippocampal memory theory, enhancing long-term knowledge integration through deeper passage connections and online LLM reasoning [14].

4.4 Implementation Details

To ensure consistency and fairness, we adopt Google's `gemini-2.0-flash`[1] as the LLM backbone across all baselines. For dense retrieval, we use OpenAI's `text-embedding-3-large`[2], a high-performance multilingual embedding model suitable for our diverse datasets. All advanced baselines are executed with their default hyperparameters to reflect typical out-of-the-box performance.

For evaluation, we follow the LLM-as-a-Judge protocol [28], using OpenAI's `gpt-4o`[3] as the judging model. To ensure deterministic and reproducible judgments, we set `temperature=0.0` and `maxTokens=32`. The judge is blinded to system identity to avoid bias.

Evaluation Prompt. The following system prompt was used consistently for standard QA evaluation:

```
### Role
You are an expert language model evaluator. Your task is to evaluate
the model prediction given a ground truth.

### Instruction
- If the prediction is "no answer" → False
- If the prediction contains the ground truth verbatim → True
- If the prediction paraphrases the ground truth → True
- If the prediction contradicts the ground truth → False
- Evaluation is language-agnostic: ignore whether the prediction
  is in English or Vietnamese

### Note
- The prediction may contain extra explanatory text: ignore it
```

In our agent reasoning setup, we employ CoT prompting [29], where intermediate steps are explicitly generated to strengthen complex problem-solving. For the chunking function $\mathcal{C}$, we rely on the Semantic Chunking module from LangChain[4], a framework for developing applications with LLMs.

4.5 Results and Analysis

Table 1 summarizes performance across four QA benchmarks. Several clear patterns emerge.

First, Naive RAG pipelines (Dense, BM25, Hybrid) yield uniformly low scores, confirming that single-index retrieval without reasoning or structural augmentation is insufficient for complex QA. Hybrid retrieval occasionally improves

[1] https://ai.google.dev/gemini-api/docs/models/#gemini-2.0-flash.
[2] https://platform.openai.com/docs/models/text-embedding-3-large.
[3] https://platform.openai.com/docs/models/gpt-4o.
[4] https://www.langchain.com/.

Table 1. Answer correctness (↑) across four QA benchmarks. Best scores are in **bold**; second-best are underlined.

Method	SQuAD			UIT-ViQuAD			HotpotQA			VIMQA		
	Accuracy	F1	EM	Accuracy	F1	EM	Accuracy	F1	EM	Accuracy	F1	EM
Naive RAG (Dense)	30.30	11.20	5.50	18.20	1.75	0.20	11.30	6.83	2.00	41.20	**12.91**	0.90
Naive RAG (BM25)	30.10	10.84	5.20	18.70	1.56	0.56	11.80	7.23	2.20	40.00	12.35	0.25
Naive RAG (Hybrid)	30.30	10.99	5.20	18.30	1.40	0.78	11.60	7.09	**2.40**	42.40	12.79	0.50
IRCoT	37.60	**11.23**	6.40	43.50	1.78	0.23	23.40	11.70	0.50	44.60	2.33	0.60
LightRAG	39.20	11.30	7.80	47.90	1.98	0.12	22.20	12.76	0.60	43.20	4.56	**1.23**
MiniRAG	49.00	12.30	2.33	55.00	**6.89**	0.60	21.70	1.99	0.00	54.50	06.89	1.01
NodeRAG	44.10	11.90	6.93	48.10	1.27	0.78	56.70	13.70	0.60	45.30	0.00	0.15
HippoRAG 2	47.30	12.55	**9.70**	48.40	0.17	0.32	60.30	18.90	1.20	45.00	7.40	0.73
URAG 2.0 (Ours)	**57.90**	4.35	9.60	**55.90**	5.23	**0.98**	**61.40**	**19.60**	1.50	**57.80**	8.90	0.65

Table 2. Answer accuracy breakdown on HotpotQA and VIMQA across reasoning depths (2-hop, 3-hop, 4-hop, 5+ hops). Best scores are in **bold**; second-best are underlined.

Method	HotpotQA				VIMQA			
	2-hop	3-hop	4-hop	5+ hop	2-hop	3-hop	4-hop	5+ hop
Sample count	550	300	121	29	500	300	150	50
Naive RAG (Dense)	0.19	0.01	0.02	0.00	0.71	0.17	0.03	0.02
Naive RAG (BM25)	0.19	0.02	0.04	0.03	0.58	0.33	0.47	0.00
Naive RAG (Hybrid)	0.20	0.01	0.02	0.03	0.63	0.30	0.13	0.02
IRCoT	0.34	0.06	0.02	0.00	0.37	0.07	0.03	0.00
LightRAG	**0.81**	0.73	0.71	**0.41**	0.73	0.86	0.53	0.12
MiniRAG	0.36	0.05	0.03	0.00	0.76	0.46	0.18	0.02
NodeRAG	0.66	0.54	0.41	0.00	0.76	0.21	0.07	0.00
HippoRAG 2	0.75	0.52	**0.87**	0.13	0.73	**0.87**	0.23	0.15
URAG 2.0 (Ours)	0.77	**0.75**	0.74	0.34	**0.80**	0.20	**0.57**	**0.20**

EM (e.g., 2.4 on HotpotQA), but overall accuracy remains below 45%, illustrating limited robustness.

Second, reasoning-aware methods such as IRCoT outperform Naive RAG, especially on English datasets (e.g., +7.3 Accuracy on SQuAD over Naive Dense), but degrade on Vietnamese datasets, suggesting sensitivity to language resources and LLM quality. Graph-augmented systems (LightRAG, MiniRAG, NodeRAG, HippoRAG 2) deliver mixed results: LightRAG achieves strong F1 on HotpotQA (12.8), MiniRAG excels in efficiency settings (55.0 Accuracy on UIT-ViQuAD), NodeRAG attains the best single baseline Accuracy on HotpotQA (56.7), while HippoRAG 2 leads in EM (9.7 on SQuAD) and F1 (18.9 on HotpotQA).

Finally, **URAG 2.0 consistently achieves the best Accuracy across all datasets**—57.9 on SQuAD, 55.9 on UIT-ViQuAD, 61.4 on HotpotQA, and 57.8 on VIMQA—while also delivering the highest F1 on HotpotQA (19.6). Although URAG 2.0 does not always dominate token-overlap metrics such as F1 and EM, this is expected: these metrics capture surface string similarity rather than semantic correctness. Under LLM-as-a-Judge evaluation, which better reflects factual validity, URAG 2.0 shows clear and consistent superiority.

Table 2 further analyzes HotpotQA and VIMQA by reasoning depth. URAG 2.0 sustains strong accuracy up to 4-hop queries (e.g., 0.75 on 3-hop HotpotQA and 0.57 on 4-hop VIMQA), where other methods degrade sharply. While LightRAG excels at shallow 2-hop reasoning, its performance collapses beyond 3 hops. HippoRAG 2 achieves the highest 4-hop accuracy on HotpotQA (0.87), but URAG 2.0 provides more stable performance across hop counts.

Taken together, these results demonstrate that URAG 2.0 not only surpasses strong baselines in overall accuracy, but also maintains robustness and generalization across reasoning depths and language diversity. This validates the design choice of leveraging FAQ-style paraphrastic indices and document-level contextual indices under agentic orchestration.

5 Conclusion

In this work, we presented URAG 2.0, an agentic dual-retrieval framework that extends the original URAG design. By integrating FAQ-based and document-based retrieval under coordinated agent orchestration, URAG 2.0 overcomes the limitations of single-stream RAG pipelines and demonstrates consistent improvements across four QA benchmarks in both English and Vietnamese. Our experiments show that URAG 2.0 not only enhances factual accuracy but also maintains robustness in complex multi-hop reasoning, outperforming strong baselines such as IRCoT, LightRAG, and HippoRAG.

Looking forward, several avenues remain open. First, while URAG 2.0 improves retrieval accuracy and reasoning depth, efficiency challenges persist in large-scale deployments; adaptive retrieval strategies and dynamic agent orchestration may further reduce computational costs. Second, extending URAG 2.0 to multimodal domains (e.g., vision–language QA) and cross-lingual retrieval can broaden its applicability. Finally, incorporating explicit user feedback and interactive clarification mechanisms could enhance transparency, trust, and adaptability in real-world use cases. We believe these directions will contribute to advancing dual-retrieval and agentic RAG systems toward more general, reliable, and human-centered knowledge assistants.

References

1. Brown, T.B., et al.: Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, ser. NIPS 2020. Curran Associates Inc., Red Hook (2020)
2. Ji, Z., et al.: Survey of hallucination in natural language generation. ACM Comput. Surv. **55**(12) (2023). https://doi.org/10.1145/3571730
3. Nguyen, L.S.T., Quan, T.T.: URAG: implementing a unified hybrid RAG for precise answers in university admission chatbots – a case study at HCMUT. In: Buntine, W., Fjeld, M., Tran, T., Tran, M.-T., Huynh Thi Thanh, B., Miyoshi, T. (eds.) Information and Communication Technology, pp. 82–93. Springer, Singapore (2025)
4. Lewis, P., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, ser. NIPS 2020. Curran Associates Inc., Red Hook (2020)
5. Borgeaud, S., et al.: improving language models by retrieving from trillions of tokens. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, 17–23 July 2022, vol. 162, pp. 2206–2240. PMLR (2022). https://proceedings.mlr.press/v162/borgeaud22a.html
6. Fan, W., et al.: 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, ser. KDD 2024, pp. 6491–6501. Association for Computing Machinery, New York (2024). https://doi.org/10.1145/3637528.3671470
7. Arslan, M., Ghanem, H., Munawar, S., Cruz, C.: A survey on RAG with LLMs. Procedia Comput. Sci. **246**, 3781–3790 (2024). 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024). https://www.sciencedirect.com/science/article/pii/S1877050924021860
8. Le, K., Nguyen, H., Le Thanh, T., Nguyen, M.: VIMQA: a Vietnamese dataset for advanced reasoning and explainable multi-hop question answering. In: Calzolari, N., et al. (eds.) Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, pp. 6521–6529. European Language Resources Association (2022). https://aclanthology.org/2022.lrec-1.700/
9. Yang, Z., et al.: HotpotQA: a dataset for diverse, explainable multi-hop question answering. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp. 2369–2380. Association for Computational Linguistics (2018). https://aclanthology.org/D18-1259/
10. Duarte, A.V., Marques, J.D., Graça, M., Freire, M., Li, L., Oliveira, A.L.: LumberChunker: long-form narrative document segmentation. In: Al-Onaizan, Y., Bansal, M., Chen, Y.-N. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, pp. 6473–6486. Association for Computational Linguistics (2024). https://aclanthology.org/2024.findings-emnlp.377/
11. Guo, Z., Xia, L., Yu, Y., Ao, T., Huang, C.: LightRAG: simple and fast retrieval-augmented generation (2024)
12. Fan, T., Wang, J., Ren, X., Huang, C.: MiniRAG: towards extremely simple retrieval-augmented generation (2025)

13. Xu, T., et al.: NodeRAG: structuring graph-based rag with heterogeneous nodes (2025)
14. Gutiérrez, B.J., Shu, Y., Qi, W., Zhou, S., Su, Y.: From RAG to memory: non-parametric continual learning for large language models (2025)
15. Trivedi, H., Balasubramanian, N., Khot, T., Sabharwal, A.: Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In: Rogers, A., Boyd-Graber, J., Okazaki N., (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, Canada, pp. 10 014–10 037. Association for Computational Linguistics (2023). https://aclanthology.org/2023.acl-long.557/
16. Gutierrez, B.J., Shu, Y., Gu, Y., Yasunaga, M., Su, Y.: HippoRAG: neurobiologically inspired long-term memory for large language models. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024). https://openreview.net/forum?id=hkujvAPVsg
17. Singh, A., Ehtesham, A., Kumar, S., Khoei, T.T.: Agentic retrieval-augmented generation: a survey on agentic RAG, *ArXiv*, vol. abs/2501.09136 (2025). https://api.semanticscholar.org/CorpusID:275570331
18. Nguyen, T., Chin, P., Tai, Y.-W.: MA-RAG: multi-agent retrieval-augmented generation via collaborative chain-of-thought reasoning (2025)
19. Chang, C.-Y., et al.: MAIN-RAG: multi-agent filtering retrieval-augmented generation. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, pp. 2607–2622. Association for Computational Linguistics (2025). https://aclanthology.org/2025.acl-long.131/
20. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
21. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Trans. Big Data **7**(3), 535–547 (2019)
22. Asai, A., Wu, Z., Wang, Y., Sil, A., Hajishirzi, H.: Self-RAG: learning to retrieve, generate, and critique through self-reflection. In: The Twelfth International Conference on Learning Representations (2024). https://openreview.net/forum?id=hSyW5go0v8
23. Yan, S.-Q., Gu, J.-C., Zhu, Y., Ling, Z.-H.: Corrective retrieval augmented generation (2024)
24. Wang, Z., et al.: Speculative RAG: enhancing retrieval augmented generation through drafting. In: The Thirteenth International Conference on Learning Representations (2025). https://openreview.net/forum?id=xgQfWbV6Ey
25. He, J., Treude, C., Lo, D.: LLM-based multi-agent systems for software engineering: literature review, vision, and the road ahead. ACM Trans. Softw. Eng. Methodol. **34**(5) (2025). https://doi.org/10.1145/3712003
26. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: Su, J., Duh, K., Carreras, X. (eds.) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, pp. 2383–2392. Association for Computational Linguistics (2016). https://aclanthology.org/D16-1264/

27. Van Nguyen, K., Nguyen, D.-V., Gia-Tuan Nguyen, A., Luu-Thuy Nguyen, N.: A Vietnamese dataset for evaluating machine reading comprehension. In: Scott, D., Bel, N., Zong, C. (eds.) Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 2595–2605 (2020). https://aclanthology.org/2020.coling-main.233/
28. Gu, J., et al.: A survey on LLM-as-a-judge (2024)
29. Wei, J., et al.: Chain of thought prompting elicits reasoning in large language models. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022). https://openreview.net/forum?id=_VjQlMeSB_J