Grenze International Journal of Engineering and Technology, June Issue

A Taxonomy and Survey of Multi-Agent Retrieval-Augmented Generation: Architectures, Evaluation, and Open Challenges

Renuka L[1], Tejaswi K[2], Shree Kantesh K H[3] and Divyashree B K[4]
[1-4]Department of Computer Application, JSS Science and Technology University, Mysore, India
Email: rkl@jssstuniv.in, tejaswi@jssstuniv.in, skh@jssstuniv.in, divyashreebk13@gmail.com

***Abstract*— Retrieval-Augmented Generation (RAG) was an addition created to support large language model (LLM) outputs in external, verifiable knowledge—without needing to spend so much money to reduce hallucination. model retraining. However, single-agent RAG pipelines fall short every time when tasks require multi-hop. reasoning, noisy retrieved content or heterogeneous retrieved content or domain specific disambiguation. Multi-Agent RAG (MA-RAG) tackles these limitations by breaking down the retrieval-generation process into distinct tasks. Pipeline across an ensemble of specialized agents: Planners, retrievers, filters, reasoners, Each of these provides a specific cognitive function: synthesisers, and verifiers. This paper offers a In-depth survey of the MA-RAG landscape. The field of development is traced back to its beginnings and Take the systems from reflective to full collaborative architecture, and add a taxonomy of seven categories. Analyse typical examples of each system. Communication protocols, memory designs, The following reasoning strategies and verification mechanisms are explored in detail: benchmark coverage, and evaluation metrics. We conclude by stating seven open challenges and a list of high-priority Directions that include adaptive topologies, scalable retrieval, standardised evaluation, and trustworthy. deployment.**

***Keywords*— Retrieval-Augmented Generation · Multi-Agent Systems · Large Language Models· Agent Collaboration · Hallucination Mitigation · Agentic AI.**

I. INTRODUCTION

LWMs include the world knowledge into the parameters they have learned, Makes knowledge expensive to check, update and grow over time [14]. RAG was launched at just the right moment. This is addressed by providing dynamic access to external knowledge to supplement parametric knowledge. at inference time, corpora. The two ground-breaking frameworks REALM [9] and RAG [21] were laid down. canonical design: a retriever is responsible for extracting relevant passages from an indexed corpus, and a generator is responsible sequence modeling, memorization, or reasoning—while retaining the ability to learn and recognize vector domains. It controls its output with those passages and retains the capacity to learn and recognize vector domains as a form of learning, passage retrieval, or reasoning. sequence generation.

Although single-agent RAG pipelines show some promise in the early stages, there are many drawbacks in terms of realistic. settings. Multi-hop queries, noisy and contradictory retrieved documents, heterogeneous data.

Grenze ID: 01.GIJET.12.2.34_25

Disambiguation is frequently required using a single generalised modality, and domain-specific disambiguation requires regularly goes beyond such a single modality. agent can handle reliably [8,29]. MA-RAG responds to these limitations by decomposing the pipeline into specialised, cooperating roles:

- Planner agents break complex queries into manageable sub-tasks;
- Retrieval agents gather evidence from vector stores, knowledge graphs, or live web APIs;
- Filtering agents remove irrelevant or misleading context;
- Reasoning agents construct multi-hop inference chains across evidence fragments;
- Synthesis agents reconcile heterogeneous evidence into a coherent answer;
- Verification agents check factual consistency before the final response is issued.

This distribution of cognitive labour has allowed MA-RAG systems to outperform single-agent baselines across accuracy, noise robustness, and—in some configurations—latency, as consistently shown in the 2023–2026 literature [1,4,29,42]. Readers seeking a broader treatment of LLM-based autonomous agents are directed to Wang et al. [38].

A. Contributions

This survey makes four contributions to the field:

- A taxonomy of MA-RAG systems organised into seven principled categories, reflecting the primary architectural strategy of each system class.
- A structured comparative analysis of representative systems published between 2020 and 2026, covering architecture, communication design, retrieval strategies, reasoning mechanisms, and empirical performance.
- A synthesis of three dominant agent-communication paradigms and a four-dimensional evaluation framework covering factual accuracy, retrieval quality, reasoning depth, and computational efficiency.
- An articulation of seven open challenges and a corresponding set of high-priority research directions.

B. Survey Methodology, Scope, and Organisation

Scope and sources. We surveyed peer-reviewed and pre-print literature published between 2020 and 2026, drawn from major venues (NeurIPS, ICML, ICLR, ACL, EMNLP, KDD) and curated pre-print repositories. Papers were identified by querying combinations of 'retrieval-augmented generation' with 'multi-agent', 'agentic', 'collaborative', 'filtering', 'verification', and 'multimodal', followed by backward and forward citation tracing.

Inclusion criteria. We retained systems that (i) employ two or more cooperating agents or explicit role specialisation within a retrieval-generation pipeline, (ii) report empirical evaluation on at least one public benchmark, and (iii) introduce a distinguishable architectural or coordination contribution. Foundational single-agent and reflective RAG systems are included to provide historical context.

Organisation. Section 2 reviews the background. Section 3 presents the taxonomy. Section 4 analyses architectures and collaboration mechanisms. Section 5 surveys benchmarks and metrics. Section 6 discusses open challenges. Section 7 identifies future directions. Section 8 concludes.

II. Background: From Retrieval-Augmented to Reflective Generation

A. Foundational and Reflective RAG

REALM [9] introduced end-to-end training of a neural retriever within the language model pre-training objective, using salient-span masking to encourage knowledge retrieval rather than reliance on parametric memorisation. An important practical property of REALM is that its knowledge corpus can be replaced at inference time without retraining. RAG [21] extended this design to open-ended generation, pairing Dense Passage Retrieval (DPR) [18] with a BART generator over a 21-million-passage index. It added RAG-Sequence and RAG-Token marginalisation strategies and This has been shown to be true on knowledge-intensive tasks, such as in the case of dense retrieval and sparse BM25 [34] retrieval. tasks.

Later research investigated closer connections between retrieval and generation. Fusion-in-Decoder (FiD) [12] processes each retrieved passage independently by a shared encoder and combines the results of each into one. passage representations for decoder pass, allowing for evidence aggregation across-passages Multiple (dozens) documents at once. This was taken one step further by Atlas [13] who trained the retriever and language model jointly together under an instruction-tuned objective, resulting in impressive few-shot performance. It outperforms previous methods on tasks that require knowledge, but have much fewer labeled examples.

A second layer was added and the pipeline was rendered reflective. ReAct [45] interleaved language-based reasoning Compared to chain-of-thought, traces with concrete environment-facing actions were significantly less prone to hallucination.Traces with concrete environment-facing actions were significantly less prone to

hallucination compared to chain-of-thought. thought prompting [40] alone. SELF-RAG [2] integrated self-reflection directly into generation by: special tokens that determine when retrieval is appropriate and check the relevance of retrieved passages to facts, Quality of input information, support and overall output quality. Corrective actions when an evaluation fails.
CRAG [43] added a light-weight evaluator that can trigger corrective actions if an evaluation fails. actions based on confidence-thresholded actions—knowledge refinement, web search or hybrid synthesis— Quality assessments of retrievals.
One of the most impactful findings is the interleaved retrieval [37] presented by IRCoT.Interleaved retrieval [37] is a particularly influential finding, formalised by IRCoT. In addition to that, the chain-of-thought reasoning should follow and not precede it. Every intermediate step is directed to the next retrieval query, which greatly enhances multi-hop performance.

B. Enabling Orchestration Frameworks

Contemporary MA-RAG implementations rest on a mature ecosystem of orchestration frameworks that differ in their primary abstractions and communication models, which in turn shapes the agent topologies they most naturally support.

TABLE I. ORCHESTRATION FRAMEWORKS COMMONLY USED TO BUILD MA-RAG SYSTEMS

Framework	Primary Abstraction	Communication Model	Key MA-RAG Feature
LangGraph [46]	Directed acyclic graph	Shared mutable graph state	Stateful multi-step agent workflows
AutoGen [41]	ConversableAgent	Peer-to-peer message passing	Interactive retrieval and group chats
LangChain [46]	Chain / tool	Sequential pipeline	Tool integration and vector-store access
LightRAG	Knowledge graph	Graph traversal + LLM	Dual-level (local/global) graph retrieval
FAISS / Qdrant [16]	Vector index	Similarity-search API	Scalable dense-retrieval backend

Beyond framework infrastructure, advances in communicative and role-playing agents have contributed reusable coordination primitives. CAMEL [22] demonstrated that assigning distinct roles to LLM agents and mediating their interactions through a structured dialogue protocol enables autonomous problem-solving across extended task horizons. MetaGPT [11] encodes professional workflows as agent role specifications, showing that structured multi-role coordination substantially reduces hallucinated outputs in complex tasks. A comprehensive survey of LLM-based autonomous agents is provided by Wang et al. [38].

III. A TAXONOMY OF MULTI-AGENT RAG SYSTEMS

We propose a seven-category taxonomy based on the primary architectural strategy of each system class (Table 2). The categories are not mutually exclusive—a given system may combine several strategies—and the primary assignment reflects each system's defining innovation. Looking across categories, several recurring design patterns emerge.
Nearly all contemporary MA-RAG systems perform explicit post-retrieval processing rather than relying on retrieval quality alone; adopt graph- or DAG-based orchestration for agent coordination; employ inter-agent consensus or verification before final generation; and demonstrate that smaller models operating within a multi-agent framework can match or exceed larger monolithic models—evidence that architectural decomposition is a more efficient path to improved capability than simply scaling parameters.

TABLE II. A SEVEN-CATEGORY TAXONOMY OF MA-RAG SYSTEMS WITH REPRESENTATIVE EXAMPLES

Category	Defining Characteristic	Representative Systems
Planner-based	A dedicated planner decomposes queries into atomic sub-tasks for iterative multi-hop retrieval	MA-RAG [29], ReAct [45], MetaGPT [11], ChatDev [33]
Filtering-based	Specialised filter agents evaluate and prune retrieved documents before generation	MAIN-RAG [4], MASS-RAG [42], CRAG [43]
Synthesis-based	A synthesis agent reconciles heterogeneous evidence from parallel retrieval branches	MASS-RAG [42], HM-RAG [24], FiD [12]
Hierarchical	A tiered structure (decompose → execute → decide) manages multi-source retrieval	HM-RAG [24], ER Framework [1]
Multimodal	Agents handle and bridge text, image, graph, and web modalities	HM-RAG [24]
Domain-specific	Agent roles tailored to domain logic (entity resolution, medical QA, legal analysis)	ER Framework [1]
RL / adaptive	Agents adjust strategies via feedback, confidence scores, or RL objectives	SELF-RAG [2], CRAG [43], WebGPT [28]

IV. Architectures and Collaboration Mechanisms

A. Common Agent Roles

In spite of the variety of architecture observed throughout the MA-RAG systems examined, there is a clear understanding of the functional aspects of the systems. structure [29,38,42]. A planner (or decomposition) agent analyzes incoming query and breaks it down into sub-problems; retrieval agents by picking up external evidence; filtering agents eliminate irrelevant or noisy information. content; reasoning agents build chains of inferences on multiple hops; synthesis agents combine intermediate transitions to a logical conclusion; and verifiers check for the accuracy of the facts before the response is committed. Adding specialised agent information to the balance sheet in a consistent fashion has been a recurring empirical result of this literature. roles has a greater impact on performance than any single model will have.

B. Communication and Coordination

There are three prominent paradigms in the literature. In a shared graph-state model (as in the example of) by LangGraph), agents can exchange information using a shared mutable structure that holds queries, retrieved answers, and other information. documents, traces of reasoning, and confidence scores, which is at the same time used as a working memory, An audit trail of the pipeline's execution, and a communication infrastructure. In the message-passing model, agents exchange structured messages with retrieval scores, intermediate reasoning output, A natural application is parallel retrieval branches, in which independent agents operate concurrently and are suitable for and task assignments [22,41]. In a sequential pipeline, each agent's result is fed directly into the next stage of processing. These are easier to implement, but may have latency as the number of stages in the pipeline increases.

Consensus voting is based on self-consistency [39], which demonstrates that if multiple independent samples are obtained, they will all tend to agree with each other. The robustness is significantly increased by reasoning traces and taking the most frequently correct answer. Longer-term memory coordination is based on generative agent designs [31] which preserve episodic memory per agent stores updated following interactions! Role-playing architectures like CAMEL [22] further Show that the dialogue between agents can be organized so as to enable multi-turn, coherent collaborative reasoning. over extended tasks. Coordination strategies also include centralised planner-directed orchestration, confidence-threshold routing, and adaptive agent invocation—where specialised agents are activated only when query complexity warrants. Memory across the system typically operates at four levels: short-term working memory, shared session memory, retrieval memory backed by an indexed knowledge store, and the long-term parametric memory of the underlying LLM.

C. Retrieval and Post-Retrieval Processing

MA-RAG systems use a varied set of retrieval strategies depending on the task. Dense vector retrieval [18] targets semantic similarity, sparse BM25 [34] handles keyword-sensitive queries, and hybrid combinations of both cover a wider range of retrieval demands. Graph-based retrieval [32] supports multi-hop structural reasoning over knowledge graphs, while web-based retrieval [28] provides access to real-time information. FAISS-based approximate nearest-neighbour search [16] serves as the scalable backend for dense retrieval, although it introduces approximation errors at very large scale.

There are several adaptive retrieval strategies that have increased coverage and diversity of evidence. FLARE [15] introduces new retrieval episodes conditionally when the model is unsure about the token generation, rather than just at the end of a query. Based on the reformulation of queries, a new query is formulated by HyDE [7] as a The query representation is the hypothetical relevant passage, which is enhanced by the dense retrieval approach. performance. IRCoT [37] leverages each intermediate step in the chain of thoughts to inform future retrievals. for multi-hop questions. There are other methods also, such as multi-query expansion, complexity-aware routing, and so on. Low confidence assessment-based and corrective re-retrieval [43].

Optimisation that takes place after extraction has become a hallmark of contemporary MA-RAG. Before evidence Then it is subjected to the relevance assessment, confidence scoring, and The techniques of similarity thresholding, knowledge refinement, and context-aware reranking [4,43]. These operations Together, minimise noise transmission and enhance the scientific basis of downstream generation.

D. Reasoning, Verification, and Hallucination Mitigation

Chain-of-thought prompting [40] is another powerful combination of reasoning strategies employed by MA-RAG systems.Another powerful combination of reasoning strategies used by MA-RAG systems is the chain-of-thought prompting [40]. and justification [36] are key components of computational problem-solving.Some of

the fundamental features of computational problem solving are multi-hop evidence integration, planning-based decomposition, tool-assisted reasoning [35] and iterative justification [36]. Consensus-based decision making [39] and self-refinement [25] are examples. Distributing chain-of-thought reasoning across agents allows planners to formulate strategies, and retrievers to gather evidence, while filters to select and eliminate information. refine contextual information, reasoners to construct inference chains, and synthesisers to integrate them [29]— To gain more reasoning depth and maintain modularity.

Hallucination in LLMs has been widely documented in various generation tasks [14] and MA-RAG Tackles it at all stages of the pipeline. Retrieval agents eliminate extraneous information; Filtering agents knowledge bases are used for consistency checking of logical consistency; and verification agents - agents who make final factual checks prior to issuing a response.

Toolformer-style tool invocation [35] is a method to reason without the need to fine-tune. Self-Refine [25] further enhances reliability by having the model self-judge and adjust its results in a self-refining fashion, and Inter-agent critique can be used to provide one agent with feedback on the reasoning of another agent before synthesis occurs [2,45].

V. Evaluation Benchmarks and Metrics

A. Primary Benchmarks

MA-RAG systems are tested in terms of general knowledge, fact-verification, domain-specific, multimodal benchmarks (Table 3).

Single-hop and multi-hop tasks are relevant, as coverage is important, The key insight here is that the major benefit of multi-agent designs is not merely because of their fact information, but because of their ability to reason about their compositional structure. recall.

TABLE III. CHOSEN REPRESENTATIVE BENCHMARKS FOR MA-RAG SYSTEMS EVALUATION BRACKETED CITATIONS: CITATIONS CONTAINING THE "BRACKETS" SYMBOL DATASET PAPERS

Benchmark	Domain	Task Type	Reasoning
Natural Questions [20]	General	Open-domain QA	Single-hop
HotpotQA [44]	General	Multi-hop QA	2-hop
TriviaQA [17]	General	Open-domain QA	Single-hop
2WikiMultiHopQA [10]	General	Complex QA	Multi-hop
FEVER [36]	Fact verification	Claim verification	Entailment
PopQA [26]	Long-tail entities	Entity-focused QA	Single-hop
ARC-Challenge [5]	Science	Multiple choice	Reasoning
ScienceQA / CrisisMMD	Multi-subject / disaster	Multimodal QA	Multimodal
MedMCQA / PubMedQA	Medical	Domain QA	Domain-specific

B. A Four-Dimensional Evaluation Framework

Factual accuracy. The metric for Exact Match (EM), overall accuracy, FactScore (precision over verifiable atomic) For long-form output assessment, it uses ROUGE/BLEU and for claims it uses the same.

Retrieval quality.

Recall@k, the Mean Reciprocal Rank (MRR), noise robustness, measured by Recall@k, There are a number of measurements, including performance with injected documents with irrelevant information and Evidence Coverage Rate (ECR).

Reasoning depth. The difference in performance between two different machines (e.g., HotpotQA–NQ). Retrieval precision/recall over retrieved documents.

Computational efficiency. Average retrieval and inference latency, API calls per query, token usage, and the runtime-overhead multiplier compared to one-agent model.

The RAGAS framework [6] operationalises three of these dimensions—context relevance, answer faithfulness, and answer relevance—as automated metrics that do not require human annotations, making it a practical complement to benchmark-based evaluation. ARES [6] offers similar automated coverage using lightweight fine-tuned judges.

C. Comparative Performance

Table 4 consolidates representative results reported in the primary sources. Direct comparison across systems is complicated by differences in retrieval corpora and backbone model versions (see Section 6); the figures should therefore be read as indicative rather than strictly commensurable across rows.

TABLE IV. REPORTED PERFORMANCE OF REPRESENTATIVE SYSTEMS. DASHES DENOTE UNREPORTED VALUES; * DENOTES THE SCIENCEQA BENCHMARK

System	NQ EM	HotpotQA EM	TriviaQA	ARC-C	Key Strength
RAG [21]	44.5	—	68.0	—	Foundational dense retrieval
SELF-RAG-13B [2]	—	—	69.3	73.1	Adaptive retrieval + self-reflection
MAIN-RAG (Llama3-8B) [4]	—	—	74.1	61.9	Training-free noise filtering
MASS-RAG (Llama3-8B) [42]	—	—	76.7	78.7	Multi-perspective synthesis
MA-RAG (GPT-4o-mini) [29]	59.5	52.1	—	—	Collaborative multi-hop CoT
HM-RAG [24]	—	—	—	93.7*	Multimodal hierarchical synthesis
Hybrid RAG [3]	92 (Top-10)	—	—	—	~4× latency reduction

VI. OPEN CHALLENGES

Despite rapid progress, seven interconnected challenges constrain the production deployment of MA-RAG systems.

Computational overhead and latency. Multi-agent pipelines invoke multiple LLM calls per query and per document. MASS-RAG, for example, reports a four- to seven-fold increase in runtime compared to single-agent baselines [42]. No principled methodology currently exists for navigating the thoroughness–latency trade-off at query time rather than only at design time, which represents a genuine gap between research prototypes and production deployment.

Coordination and communication costs. As the number of agents grows, inter-agent messaging, state serialisation, and dependency management all scale accordingly [22]. There is no analytical framework analogous to distributed-computing complexity theory that would allow coordination overhead to be predicted or bounded prior to deployment.

Scalability to large corpora. Most published systems are evaluated on corpora of 10^5 to 10^6 documents; performance on billion-scale repositories remains largely unexplored. FAISS-based dense retrieval [16] introduces approximation errors at scale, and sequential per-agent filtering becomes prohibitive in high-throughput settings. Hierarchical indexing approaches such as HNSW [27] partially address this but have not yet been fully integrated into multi-agent pipelines.

Prompt sensitivity. Agent behaviour is highly sensitive to prompt wording and role descriptions [1]. While frameworks such as DSPy [19] treat prompt optimisation as a compilation problem for single-agent settings, systematic multi-agent extensions remain nascent, and no established methodology exists for robustness testing across networks of interacting agents.

Multimodal integration. Cross-modal alignment across text, images, graphs, and web content lacks a theoretical foundation comparable to cosine similarity in single-modality retrieval. BLIP-2-based [23] visual-to-text conversion can yield over-condensed or ambiguous descriptions [24], and no general cross-modal retrieval metric has been universally adopted by the community.

Evaluation standardisation. Divergent retrieval corpora, backbone versions, and evaluation splits make direct comparison difficult across published systems [8]. The automated metrics proposed by RAGAS [6] are promising but not yet universally adopted, and existing benchmarks [20,44,10] were not designed with multi-agent pipelines in mind.

Trustworthiness, safety, and privacy. The transparency of agents' decision processes is constrained by LLMs. The ability to be heard in high-stakes environments [14]. You'll end up making multiple calls to LLM's to route sensitive documents. Adversarial prompt injection (API) is a major privacy threat, and can be done using a poisoned retrieval corpus. A threat to security which is not explicitly protected by most present systems.

VII. FUTURE RESEARCH DIRECTIONS

We organise twelve high-priority directions under four themes.

A. Adaptive Agent Topologies

Dynamic topology adaptation. Systems must adapt the number of agents as well as the type of agents and roles in a system. Complexity of queries and real-time feedback of communication structure. Neural architecture Automated topology optimisation is drawn from the concept of search, as a good design suggestion.

Resource-efficient allocation. High capacity models are mostly used for planning and synthesis; Distilled agents are smaller, and they can be used effectively for filtering and extracting. Principled methods There is still a lack of solutions for assigning LLM capacity according to the needs of each role, based on the computation needs of the role.

Workflow learning. Task demonstrations could be used to learn optimal agent workflows by adapting The objectives of multi-agent coordination tasks are rewarded using reinforcement learning from human feedback [30]. accuracy and efficiency of computation.

B. Advanced Retrieval and Knowledge Integration

Neurosymbolic hybrid retrieval. Using neural encoders and symbolic representations. With (description logics, first order constraints) retrieval would be possible which would take account of the structure of the domain. ontologies, especially useful for applications in legal or scientific or medical MA-RAGs [32].

Web-scale scalability. Hierarchical retrieval architectures coarse subcorpus selection followed by fine selection of sentences. fine-grained agent-level retrieval—combined with HNSW [27] and ScaNN indexing are needed to support billion-scale settings. Active retrieval strategies like FLARE [15] provide a good template For scalable, cost-effective retrieval with flexibility and dynamism.

Temporal knowledge management. that do not require a complete rebuild of the entire table.Incremental index updates that carry out updates to the index while preserving the existing structure. Full re-indexing, along with the ability to detect conflicts when incoming evidence conflicts with already indexed evidence. For successful deployment in the real world, content, along with other factors, is a key component [13].

C. Evaluation and Benchmarking

Unified evaluation framework. Three-dimensional PM2.5 was used to assess the Global Change Plan, and the evaluation results were included in a standardised evaluation framework that was extended from RAGAS [6]. Within the context of factuality, retrieval quality, accuracy, multi-hop performance, efficiency and scalability—applied to Common "held-out" benchmarks [20,44,10] with the same corpora and backbone versions—would make System comparisons for the first time commensurable.

Reasoning-chain interpretability. Evaluation of the chain of evidence, per-step evidence relevance, and the logical validity of multi-hop inference chains would enhance understanding of the behaviour of the system and help This is based on the principles of debugging and chain of thought evaluation [40,37].

D. Trustworthiness, Safety, and Deployment

Adversarial robustness. Formal threat models of prompt injection via retrieved documents, role hijacking, and shared-state poisoning, along with retrieval-level adversarial detection and agent-output sanitisation [14].

Privacy-preserving MA-RAG. Federated training with multiple private corpora, or This would facilitate deployment of differentially private filtering decisions, encrypted similarity search [16]. on sensitive data without revealing the contents of the documents to the outside models.

Long-context multi-session reasoning. Hierarchical compression (where lower agents make) In addition, episodic long-term memory [31] between sessions was used to summarize their learning for higher level synthesis. would facilitate long-range knowledge-intensive reasoning. Cross-lingual and cross-modal generalisation. Multilingual MA-RAG requires language-agnostic embeddings, cross-lingual knowledge-graph alignment, and translation-aware consensus mechanisms. What if these capabilities could be expanded to audio, video and tabular data, and bring the field to truly universal multimodal systems [24].

VIII. CONCLUSION

This survey has provided a structured narrative on Multi-Agent Retrieval-Augmented Generation, mapping the development in the field from the fundamental RAG [9,21] and tightly coupled retrieval-generation systems.the evolution of the field from basic RAG [9,21] and tightly coupled retrieval-generation systems. From the point of view of its architectures it is possible to distinguish between collaborative systems like MA-RAG [29], MASS-RAG [42] or HM-RAG [24] and other non-collaborative systems. and Althaf et al. 's [1] entity-resolution framework. From this comes three general conclusions: review

First, breaking up the RAG pipeline by role of specialised agents is empirically better than The collaborative MA-RAG can achieve far more compact systems scaling model parameters. More recently, larger monolithic models were developed for various benchmarks [20,44,5].

Second, the selection of a good oven is very crucial. There are three communication paradigms used: shared graph state, message-passing [22,41] or sequential pipeline. significant consequences for latency, scalability, and fault tolerance; must be considered as a primary Architectural design decision and not an implementation detail.

Third, post-retrieval processing, turns the use of retrieval quality plus out has been shown to be more effective than retrieval quality alone: agent-level filtering, self-refinement [25], and These gains in performance are primarily due to multi-perspective synthesis.

There remain significant challenges to production scalability, adversarial robustness, and privacy preserving. Standardisation of operation, and evaluation [6,8]. The twelve directions mentioned in Section 7 form A near-term research agenda with the potential to produce within a few years reliable, MA-RAG systems. Reasoning about knowledge and privacy in arbitrary modalities and domains. Multi Agent RAG is not just a single-agent RAG with some additions to it; it is a qualitative improvement on the single-agent RAG. A shift where the basic element of intelligent behaviour is a society of cooperating agents instead of One of the more important lines of applied artificial intelligence today—a single model.

REFERENCES

[1] Althaf, A. M., Mohammed, M. A., Milanova, M., Talburt, J., & Cakmak, M. C. (2025). Multi-Agent RAG Framework for Entity Resolution: Advancing Beyond Single-LLM Approaches with Specialized Agent Coordination. Computers, 14(12), 525. https://doi.org/10.3390/computers14120525

[2] Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2023). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. arXiv. https://arxiv.org/abs/2310.11511

[3] Baban, H., Abhishek, S., Pidaparthi, Gulati, S., & Nema, A. (2025). Optimizing retrieval-augmented generation with multi-agent hybrid retrieval. In Proceedings of the Generative AI for Personalization and Recommendation Workshop (GenAIRecP 2025). https://doi.org/10.1145/3690624

[4] Chang, C.-Y., Jiang, Z., Rakesh, V., Pan, M., Yeh, C.-C. M., Wang, G., Hu, M., Xu, Z., Zheng, Y., Das, M., & Zou, N. (2025). MAIN-RAG: Multi-agent filtering retrieval-augmented generation. In W. Che, J. Nabende, E. Shutova, & M. T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2607–2622). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.acl-long.131

[5] 5. Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., & Tafjord, Ø. (2018). Think you have solved question answering? Try ARC, the AI2 reasoning challenge. arXiv. https://arxiv.org/abs/1803.05457

[6] Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2025). Ragas: Automated evaluation of retrieval augmented generation. arXiv. https://arxiv.org/abs/2309.15217

[7] Gao, L., Ma, X., Lin, J., & Callan, J. (2022). Precise zero-shot dense retrieval without relevance labels. arXiv. https://arxiv.org/abs/2212.10496

[8] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. arXiv. https://arxiv.org/abs/2312.10997.

[9] Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M.-W. (2020). REALM: Retrieval-augmented language model pre-training. arXiv. https://arxiv.org/abs/2002.08909

[10] Ho, X., Nguyen, A.-K. D., Sugawara, S., & Aizawa, A. (2020). Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Proceedings of the 28th International Conference on Computational Linguistics (pp. 6609–6625). International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.580

[11] Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Zhang, C., Wang, J., Wang, Z., Yau, S. K. S., Lin, Z., Zhou, L., Ran, C., Xiao, L., Wu, C., & Schmidhuber, J. (2024). MetaGPT: Meta programming for a multi-agent collaborative framework. arXiv. https://arxiv.org/abs/2308.00352

[12] Izacard, G., & Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. arXiv. https://arxiv.org/abs/2007.01282

[13] Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., & Grave, E. (2022). Atlas: Few-shot learning with retrieval augmented language models. arXiv. https://arxiv.org/abs/2208.03299

[14] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12), Article 248, 1–38. https://doi.org/10.1145/3571730

[15] Jiang, Z., Xu, F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., & Neubig, G. (2023). Active retrieval augmented generation. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (pp. 7969–7992). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.495

[16] Johnson, J., Douze, M., & Jégou, H. (2021). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535–547. https://doi.org/10.1109/TBDATA.2019.2921572

[17] Joshi, M., Choi, E., Weld, D., & Zettlemoyer, L. (2017). TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1601–1611). Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1147

[18] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-T. (2020). Dense passage retrieval for open-domain question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 6769–6781). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.550

[19] Khattab, O., Singhvi, A., Maheshwari, P., Zhang, Z., Santhanam, K., Vardhamanan, S., Haq, S., Sharma, A., Joshi, T. T., Moazam, H., Miller, H., Zaharia, M., & Potts, C. (2023). DSPy: Compiling declarative language model calls into self-improving pipelines. arXiv. https://arxiv.org/abs/2310.03714

[20] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., & Petrov, S. (2019). Natural questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7, 452–466. https://doi.org/10.1162/tacl_a_00276

[21] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2021). Retrieval-augmented generation for knowledge-intensive NLP tasks. arXiv. https://arxiv.org/abs/2005.11401

[22] Li, G., Hammoud, H. A. A. K., Itani, H., Khizbullin, D., & Ghanem, B. (2023). CAMEL: Communicative agents for "mind" exploration of large language model society. arXiv. https://arxiv.org/abs/2303.17760

[23] Li, J., Li, D., Savarese, S., & Hoi, S. (2023). BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv. https://arxiv.org/abs/2301.12597

[24] Liu, P., Liu, X., Yao, R., Liu, J., Meng, S., Wang, D., & Ma, J. (2025). HM-RAG: Hierarchical multi-agent multimodal retrieval augmented generation. arXiv. https://arxiv.org/abs/2504.12330

[25] Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K., Welleck, S., Yazdanbakhsh, A., & Clark, P. (2023). Self-refine: Iterative refinement with self-feedback. arXiv. https://arxiv.org/abs/2303.17651

[26] Mallen, A., Asai, A., Zhong, V., Das, R., Khashabi, D., & Hajishirzi, H. (2023). When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 9802–9822). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.546

[27] Malkov, Y. A., & Yashunin, D. A. (2020). 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. https://doi.org/10.1109/TPAMI.2018.2889473

[28] Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., & Schulman, J. (2022). WebGPT: Browser-assisted question-answering with human feedback. arXiv. https://arxiv.org/abs/2112.09332

[29] Nguyen, T., Chin, P., & Tai, Y.-W. (2025). MA-RAG: Multi-agent retrieval-augmented generation via collaborative chain-of-thought reasoning. arXiv. https://arxiv.org/abs/2505.20096

[30] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. In Advances in Neural Information Processing Systems (NeurIPS 2022). https://mlanthology.org/neurips/2022/ouyang2022neurips-training/

[31] Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. arXiv. https://arxiv.org/abs/2304.03442

[32] Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C., Zhang, Y., & Tang, S. (2024). Graph retrieval-augmented generation: A survey. arXiv. https://arxiv.org/abs/2408.08921

[33] Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C., Zhang, Y., & Tang, S. (2024). Graph retrieval-augmented generation: A survey. arXiv. https://arxiv.org/abs/2408.08921

[34] Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends® in Information Retrieval, 3(4), 333–389. https://doi.org/10.1561/1500000019

[35] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. arXiv. https://arxiv.org/abs/2302.04761

[36] Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: A large-scale dataset for fact extraction and verification. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (pp. 809–819). Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1074

[37] Trivedi, H., Balasubramanian, N., Khot, T., & Sabharwal, A. (2023). Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. arXiv. https://arxiv.org/abs/2212.10509

[38] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., & Wen, J. (2024). A survey on large language model based autonomous agents. Frontiers of Computer Science, 18(6). https://doi.org/10.1007/s11704-024-40231-1

[39] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., & Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language models. arXiv. https://arxiv.org/abs/2203.11171

[40] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models. arXiv. https://arxiv.org/abs/2201.11903

[41] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., & Wang, C. (2023). AutoGen: Enabling next-gen LLM applications via multi-agent conversation. arXiv. https://arxiv.org/abs/2308.08155

[42] Xiao, X., Huang, H., Liu, R., & Xie, J. (2026). MASS-RAG: Multi-agent synthesis retrieval-augmented generation. arXiv. https://arxiv.org/abs/2604.18509
[43] Yan, S.-Q., Gu, J.-C., Zhu, Y., & Ling, Z.-H. (2024). Corrective retrieval augmented generation. arXiv. https://arxiv.org/abs/2401.15884
[44] Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., & Manning, C. D. (2018). HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 2369–2380). Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1259
[45] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. arXiv. https://arxiv.org/abs/2210.03629
[46] LangChain. (2025). LangGraph: Build resilient language agents as graphs. Retrieved June 5, 2026, from https://langchain-ai.github.io/langgraph/
[47] HKUDS. (2024). LightRAG: Simple and fast retrieval-augmented generation. GitHub. Retrieved June 5, 2026, from https://github.com/HKUDS/LightRAG