

Research Article

Beyond Static Retrieval: A Reinforcement Learning Framework for Dynamic and Adaptive RAG

Gopichand Agnihotram¹ , Joydeep Sarkar² , Magesh Kasthuri^{2,*} 

¹Chief Technology Office, Wipro Limited, Bengaluru, India

²Technology Services, Wipro Limited, Bengaluru, India

Abstract

Retrieval-Augmented Generation (RAG) is a widely adopted technique that enhances large language models (LLMs) by grounding their outputs in external knowledge sources. This approach reduces hallucinations, increases factual accuracy, and adapts well to rapidly evolving domains. Despite these strengths, traditional RAG implementations rely on static, heuristic-based retrieval strategies that operate independently of feedback or contextual learning. In today's fast-changing information landscape, it's crucial for language models to go beyond static retrieval when grounding their responses. That's where a RL framework comes into play for RAG. Rather than sticking to fixed, rule-based selection methods, RL allows the retrieval component to learn and adapt over time—much like how a person refines their search strategies with experience and feedback. By framing the process of document selection as a Markov Decision Process (MDP), the system can make context-aware choices that consider both immediate and future gains. This white paper explores how Retrieval-Augmented Generation can be significantly enhanced by integrating Markov Decision Processes (MDPs) and Reinforcement Learning (RL). We present a conceptual framework that models retrieval as a sequential decision-making problem. By treating document selection as an MDP and employing RL algorithms to optimize retrieval strategies, we introduce adaptivity, context sensitivity, and long-term reasoning into the RAG pipeline, leading to demonstrably more accurate and relevant generated content. The paper also outlines applications, implementation strategies, and future research directions that combine symbolic and neural methods for improved decision-making and document relevance.

Keywords

Retrieval-Augmented Generation (RAG), Reinforcement Learning (RL), Markov Decision Processes (MDP), Adaptive Retrieval, Agentic RAG, RL Policy Optimization

1. Introduction

As language models become more powerful and pervasive, their integration with external knowledge repositories is essential for applications that demand factual correctness, explainability, and domain awareness. Retrieval-Augmented Generation (RAG) addresses this by incorporating a retriever

component that sources relevant documents from external databases, which are then fed into a generator to craft a final response.

While effective, most existing RAG implementations treat retrieval as a one-shot operation based on fixed heu-

*Corresponding author: magesh.kasthuri@wipro.com (Magesh Kasthuri)

Received: 12 September 2025; **Accepted:** 23 September 2025; **Published:** 18 October 2025



Copyright: © The Author(s), 2025. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

ristics, such as top-k document selection using cosine similarity in embedding space. These approaches do not consider the broader impact of document selection on the quality of the generated output. Nor do they adapt based on feedback from user interactions, task success, or contextual evolution.

While recent advancements like active retrieval [1, 2] and multi-step reasoning [3] have started to address these limitations by allowing for iterative information gathering, they often rely on heuristics or are not optimized end-to-end with respect to the final generation quality.

To address these limitations, we propose a paradigm shift: framing retrieval as a dynamic decision-making process using the formalism of Markov Decision Processes (MDPs). This enables the system to learn optimal retrieval policies over time. We then use Reinforcement Learning (RL) to train these policies using reward signals derived from downstream outcomes. Together, MDP and RL empower RAG systems to become adaptive, context-aware, and capable of learning retrieval behaviors that evolve with experience.

2. Background: RAG, MDP, and RL

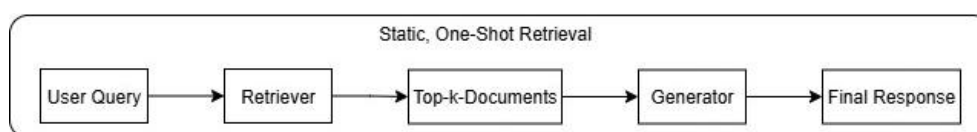


Figure 1. The Pipeline of a Traditional Static RAG System.

2.2. Markov Decision Processes (MDPs)

An MDP is a mathematical framework for modeling environments in which outcomes are partly random and partly under the control of a decision-making agent. It allows the agent to make a sequence of decisions that maximize cumulative reward.

Core elements:

- 1) States (S): Represent the environment at a specific point in time.
- 2) Actions (A): Choices the agent can make.
- 3) Transitions (T): Rules or probabilities for moving between states.
- 4) Rewards (R): Numerical feedback indicating the value of a state-action pair.
- 5) Policy (π): A strategy mapping states to actions.

In the context of RAG, the MDP formulation provides a formal structure to guide the retrieval agent's decisions at each step of the retrieval process.

2.1. Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation is a hybrid approach where a retriever fetches relevant external content, and a generator (typically a large language model) synthesizes an answer using the retrieved documents, as depicted in Figure 1. The retriever can be based on traditional lexical matching (e.g., BM25) or neural embedding-based methods (e.g., dense vector retrieval).

Key components:

- 1) Query Encoder: Converts a user query into a format suitable for retrieval.
- 2) Retriever: Searches a document corpus for relevant content.
- 3) Generator: Combines the query and retrieved passages to generate a response.
- 4) Fusion Strategy: Methods like Fusion-in-Decoder or Fusion-in-Context merge multiple documents for use during generation.

This framework offers grounding, improved factuality, and adaptability, but its performance depends heavily on the quality of the retrieved content.

2.3. Reinforcement Learning (RL)

Reinforcement Learning is a machine learning paradigm in which an agent learns to take actions in an environment to maximize cumulative rewards. It is particularly powerful when rewards are delayed or sparse, as is often the case in information retrieval and generation.

RL operates through exploration and exploitation:

- 1) Exploration: Trying new actions to discover their effects.
- 2) Exploitation: Choosing known good actions to maximize rewards.

By applying RL to the RAG context, the retriever can learn which document selection strategies lead to better generation outcomes, such as accurate answers, coherent narratives, or user satisfaction.

3. Modeling RAG Retrieval as an MDP

To leverage MDPs in RAG, we redefine the retrieval process as a sequential decision-making problem. Unlike tradi-

tional RAG pipelines, where document retrieval is treated as a single, isolated step, the MDP-based formulation considers retrieval as a series of dependent decisions that unfold over time.

3.1. States

Each state captures the current retrieval context, including:

- 1) The user query or its reformulated version
- 2) Documents retrieved so far
- 3) Accumulated metadata or similarity scores
- 4) Historical user interaction patterns (optional)

3.2. Actions

Actions correspond to decisions the retriever agent can make:

- 1) Selecting the next document to add to context
- 2) Reformulating or expanding the query [4]
- 3) Terminating the retrieval phase and initiating generation

3.3. Transitions

When an action is taken, the state transitions to reflect the new context. For example, adding a document updates the context vector. Query reformulation changes the embedding

used for future document scoring.

3.4. Rewards

Reward signals are critical for guiding policy learning. Potential reward mechanisms include:

- 1) Document relevance based on known labels or user feedback
- 2) Final response quality (e.g., BLEU, ROUGE, factual correctness)
- 3) Diversity and coverage of information retrieved
- 4) User-level metrics such as satisfaction, clicks, or task completion

Framing retrieval as an MDP allows the system to make retrieval decisions that account for long-term outcomes, not just immediate document similarity.

4. Applying Reinforcement Learning to RAG Retrieval

Once the retrieval process is modeled as an MDP, we can use RL to learn an optimal retrieval policy, as illustrated in the MDP-RAG framework in Figure 2. The retriever acts as an RL agent, improving its behavior based on feedback from past interactions.

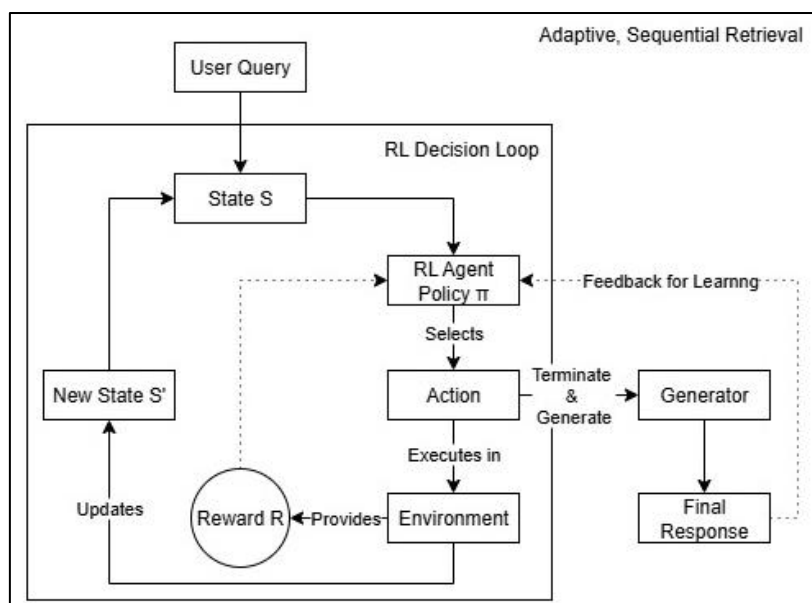


Figure 2. The Adaptive Loop of the MDP-RAG Framework.

Figure 2 illustrates the process that starts with the User Query initializing the State. RL Decision Loop is the core cycle. The State is fed to the RL Agent, which chooses an Action. The Action is performed on the Environment, which results in a New State (feeding back into the loop) and a

Reward (a dashed feedback line to the Agent). The loop continues until the agent chooses the "Terminate & Generate" action. This action breaks the cycle and sends the final set of retrieved documents to the generator. Final Feedback from the Final Response is used to improve the RL Agent over time.

4.1. Q-Learning

Q-learning learns the value of taking a certain action in a given state. In RAG, Q-values can help estimate which documents, when added to the context, are likely to lead to high-quality generations [5].

4.2. Policy Gradient Methods

These methods directly learn a policy that selects actions based on state representations. This is well-suited to neural models, as policies can be represented with differentiable architectures and trained end-to-end [6].

4.3. Actor-Critic Architectures

Actor-Critic methods combine value-based and policy-based approaches. The actor proposes actions, while the critic evaluates them. This hybrid structure provides both

stability and flexibility in training.

4.4. Imitation Learning

When rewards are difficult to compute or sparse, the system can learn from expert trajectories or human annotations. This allows for supervised pretraining of policies, which can then be fine-tuned using RL.

4.5. Multi-Agent RL

A very relevant and recent development is the framing of the RAG pipeline as a multi-agent RL (MARL) system where the query rewriter, document retriever, and answer generator are individual agents learning to cooperate [7].

RL enhances retrieval by allowing the system to prioritize actions that improve overall performance over time, rather than relying on static rules.

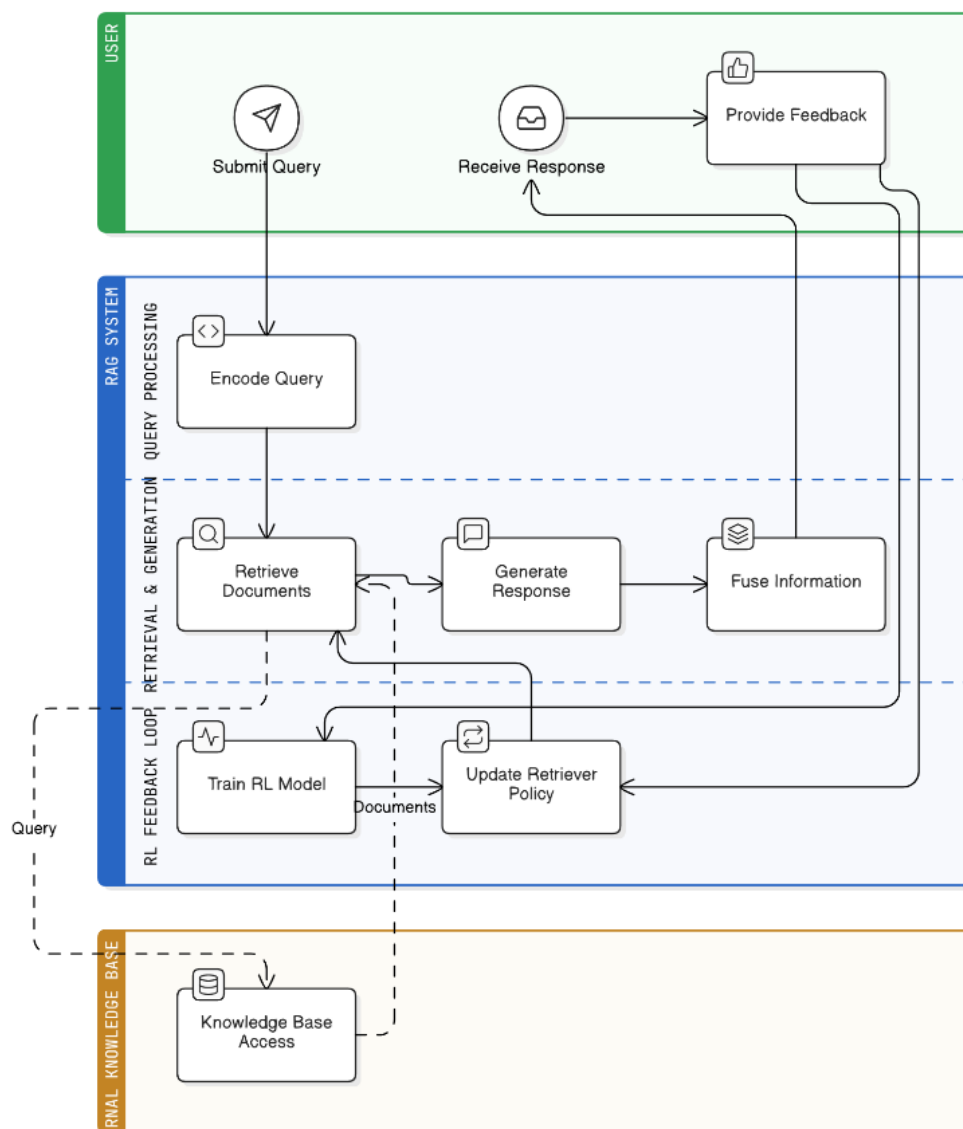


Figure 3. Business process in Reinforcement learning.

In the above Business process diagram, when we look at the intersection of business processes and reinforcement learning frameworks, especially within the realm of dynamic and adaptive Retrieval-Augmented Generation (RAG), we're really talking about how organizations can use advanced AI to continually improve and evolve the way they manage, retrieve, and generate business-critical information.

At its core, a business process is simply a structured set of activities or tasks that produce a specific outcome—whether it's processing an invoice, onboarding a new customer, or handling inventory. Traditionally, these processes are set up with fixed rules and thresholds, which means there's a limit to how much they can adapt to changing circumstances. This is where reinforcement learning (RL) comes in, breathing new life into those rigid frameworks.

4.5.1. Understanding Reinforcement Learning in the Business Context

Reinforcement learning is a subset of machine learning where an agent learns to make good decisions by interacting with an environment [8] — it receives feedback in the form of rewards or penalties and gradually improves its actions over time. RL isn't just theoretical; it's already been applied to things like trading algorithms, supply chain optimizations, and dynamic pricing strategies.

In a business context, using RL means that the system is constantly learning from each decision it makes, tweaking its future actions to maximize positive outcomes [9]. Think of it as a dedicated employee who never stops learning on the job, fine-tuning their approach with every experience.

4.5.2. The Role of Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation is a paradigm for large language models where the system retrieves relevant documents, facts, or snippets from a database and uses them to ground its generated responses [10]. This dramatically improves both the accuracy and relevance of AI-generated content, making it especially vital for dynamic business environments where information flows and requirements are always shifting.

Dynamic and adaptive RAG systems, empowered by RL, can tailor what information is retrieved and how it's integrated, based on the specific needs of a business process at any given time [11]. In other words, the system isn't just passively pulling data—it's actively learning which data sources and retrieval strategies are most effective for different business scenarios.

4.5.3. How Business Processes Integrate with RL-based RAG Systems

Let's imagine an organization that wants to streamline its customer support workflow. With a traditional rule-based

process, every customer complaint is handled the same way, regardless of context or history [12]. But with an RL-driven, RAG-powered system, the AI can dynamically choose which knowledge base articles, troubleshooting steps, or escalation paths are most likely to resolve each unique issue—because it's learning what works best over time.

Here's how the typical business process might look:

- 1) Initial Data Collection: The system gathers information about the specific business scenario—let's say, an incoming customer complaint, including their history, the product involved, and previous resolutions.
- 2) Dynamic Retrieval: Using RAG, relevant documents and knowledge sources are retrieved. RL kicks in by guiding the retrieval strategy, based on which sources have historically led to the best outcomes.
- 3) Adaptive Generation: Next, the system synthesizes a response, using both retrieved content and its generative capabilities. Again, RL shapes the generation process, learning which types of responses satisfy customers and resolve issues most effectively.
- 4) Feedback and Learning: The outcome (was the complaint resolved quickly? Was the customer satisfied?) is fed back into the system as a reward signal. The RL framework adapts its policies, improving future performance.

This cycle repeats, with each interaction helping the system get smarter and more tailored to the business's evolving needs.

4.5.4. Benefits of RL-Driven Dynamic and Adaptive RAG in Business Processes

The integration of RL with dynamic and adaptive RAG brings several transformative benefits to business processes:

- 1) Continuous Improvement: Processes don't stagnate—they get better over time, automatically adapting to changing circumstances and feedback.
- 2) Hyper-Personalization: Customer interactions, internal workflows, and decision-making can be finely tuned to individual needs and preferences, leveraging the full power of context-aware retrieval and generation.
- 3) Increased Efficiency: By learning which actions, retrievals, and responses drive the best outcomes, organizations can streamline operations, reduce bottlenecks, and save both time and resources.
- 4) Resilience and Adaptability: RL-based systems are naturally robust to change. If a process needs to pivot—say, due to regulatory changes, market shifts, or internal restructuring—the system can learn and adapt on the fly.
- 5) Data-Driven Decision Making: RL frameworks thrive on data, using real-world interactions to inform and refine business processes continuously.

4.5.5. Challenges

Of course, integrating RL-driven RAG into business pro-

cesses isn't all smooth sailing. There are a few important hurdles to keep in mind:

- 1) **Complexity of Implementation:** Setting up RL frameworks requires careful design, robust infrastructure, and thoughtful reward structures. Businesses need skilled teams to manage deployment and ongoing maintenance [13].
- 2) **Quality of Data:** RL systems are only as good as the data they learn from. Incomplete, biased, or outdated data can lead to suboptimal policies and outcomes.
- 3) **Transparency and Explainability:** Dynamic and adaptive systems can sometimes make decisions that are hard to interpret. Businesses must work to ensure that AI-driven processes remain understandable and audit-friendly.
- 4) **Alignment with Business Goals:** The reward functions in RL need to truly reflect what the business values—whether it's customer satisfaction, cost reduction, or speed of resolution. Poorly designed rewards can lead to unintended behaviors.
- 5) **Ethical and Regulatory Considerations:** As with all AI systems, there are privacy, fairness, and compliance issues that must be actively managed.

4.5.6. Best Practices for Humanizing RL-Based RAG Business Processes

To make RL-driven dynamic and adaptive RAG frameworks work for real people in real organizations, a few best practices help keep things on track:

- 1) **Start Small and Iterate:** Begin with a pilot project focused on a well-defined process. Use early feedback to adjust both the RL framework and the retrieval-generation strategies before scaling up.
- 2) **Engage Stakeholders:** Involve employees, customers, and leadership in the design, rollout, and ongoing evaluation of RL-driven processes. Their insights are invaluable for aligning technology with business realities.
- 3) **Maintain Transparency:** Establish clear reporting and monitoring of how RL and RAG systems are making decisions. This builds trust and helps catch problems early.
- 4) **Focus on Human-AI Collaboration:** Use RL and RAG to augment—not replace—human expertise. The most effective processes blend AI's adaptability with the judgement and empathy of people.
- 5) **Regularly Review and Refine:** Business environments change, and so should reward functions, retrieval sources, and generative policies. Schedule periodic reviews to keep the system aligned with evolving goals.

4.5.7. The Future of Dynamic and Adaptive RAG in Business

It's clear that RL-powered, dynamic RAG frameworks have a huge potential to reshape business processes for the

better. As organizations get more comfortable with these tools, we'll likely see processes that are not only more efficient but also more creative, responsive, and satisfying—for both employees and customers.

Imagine an HR onboarding system that learns to personalize training modules for each new hire, or a compliance workflow that adapts in real-time to new regulations without a major overhaul [14]. The possibilities are vast, and the key is to approach RL-driven RAG not as a replacement for human expertise, but as a powerful partner in the constant quest for better business outcomes.

In summary, RL frameworks coupled with dynamic, adaptive RAG are ushering in a new era of business process management: one where learning is continuous, adaptation is automatic, and workflows are increasingly tuned to the needs of those who use them. By keeping the human element at the forefront and embracing the strengths of both people and machines, businesses can unlock new levels of efficiency, agility, and insight [15]. The future isn't just automated—it's adaptive, collaborative, and profoundly human.

5. Benefits of MDP and RL in RAG

Adopting an MDP-RL framework within RAG unlocks several important capabilities:

- 1) **Adaptive Retrieval:** Retrieval strategies evolve over time, learning from past successes and failures.
- 2) **Context Sensitivity:** Decisions are informed by accumulated history and current state.
- 3) **Multi-Hop Reasoning:** The system can plan multi-step retrievals that build on prior steps.
- 4) **Improved Precision and Recall:** Rewards guide the retriever to maximize useful information and minimize noise.
- 5) **Exploratory Behavior:** RL enables retrieval of novel or unexpected content that might otherwise be overlooked.

These benefits lead to more robust and intelligent RAG systems that can operate effectively across diverse domains and tasks.

6. Applications and Use Cases

6.1. Legal and Compliance AI

Retrieving statutes, prior rulings, and legal arguments across multiple steps requires planning and precision. MDP-RAG agents can prioritize legally relevant content while avoiding redundancy.

6.2. Enterprise Search Systems

Adaptive retrieval policies trained on internal user logs and preferences improve the relevance and utility of search results within organizations.

6.3. Scientific Research and Literature Review

Exploring citation networks, identifying supporting evidence, and discovering contrasting viewpoints benefit from sequential, multi-hop document retrieval.

6.4. Customer Support and Virtual Agents

Agents that retrieve from manuals, troubleshooting guides, and prior tickets must adapt to diverse queries. RL-trained retrieval enables more helpful and dynamic interactions.

7. Challenges and Mitigations

Despite its promise, the MDP-RAG approach presents challenges:

- 1) High-Dimensional State Space: Representing retrieval history and document context can be computationally intensive.

Mitigation: Use attention-based summarization or dense embedding compression.

- 2) Sparse or Delayed Rewards: Final output quality may be difficult to measure in real time.

Mitigation: Use proxy rewards like document relevance scores or simulated user models.

- 3) Cold Start Problem: Learning effective policies from scratch is slow.

Mitigation: Begin with supervised learning or imitation learning, followed by fine-tuned RL.

- 4) Training Stability: RL training can be unstable due to noisy feedback or large action spaces.

Mitigation: Apply robust algorithms (e.g., PPO) and reward shaping techniques.

8. Implementation Considerations

Developing an MDP-RAG system [16] requires close coordination of components across machine learning, information retrieval, and systems engineering. Several best practices and architectural elements can improve success:

- 1) Data Logging and Feedback Loops: Instrument user interactions to generate signals such as click-through rates, time-on-page, or explicit ratings, which can serve as reinforcement signals.
- 2) Simulator Environments: Where real-world feedback is sparse, simulate user behavior using models trained on historical interactions. These environments allow for scalable offline RL.
- 3) Policy Modularization: Separate modules for query reformulation, document selection, and stopping criteria help maintain interpretability and control.
- 4) Scalability Concerns: MDP-RAG systems must operate within strict latency budgets. Efficient candidate pruning, approximate nearest neighbor search, and batching

are key.

- 5) Ethical Guardrails: Reinforcement learning should not inadvertently optimize for deceptive, biased, or privacy-violating behaviors. Regular audits and human feedback help prevent these issues.

9. Future Directions

As RAG systems continue to evolve, the integration of MDP and RL provides a foundation for more advanced research and applications:

- 1) Memory-Augmented MDPs: Incorporate persistent memory across retrieval sessions, enabling agents to recall past interactions and improve continuity.
- 2) Multi-Agent Collaboration: Assign specialized roles to agents—one focused on document discovery, another on summarization, and a third on synthesis—each learning its own policy.
- 3) Causal and Counterfactual Reasoning: Integrate models that reason about the causal effect of retrieved content on output quality, and learn from hypothetical outcomes.
- 4) Cross-Domain Generalization: Develop RL policies that generalize across knowledge bases and domains by incorporating meta-learning or domain adaptation strategies.

10. Conclusion

Incorporating Markov Decision Processes and Reinforcement Learning into Retrieval-Augmented Generation transforms static, one-shot retrieval into a dynamic, intelligent, and adaptive process. By treating document selection as a sequential decision-making task, RAG systems become capable of learning from past experiences, adjusting strategies to fit specific queries, and delivering more relevant and accurate responses.

This union of symbolic reasoning, sequential planning, and neural generation provides a powerful foundation for the next generation of AI agents. Whether deployed in legal research, enterprise systems, customer support, or scientific exploration, MDP-RAG systems promise higher quality, greater adaptability, and more trustworthy outcomes.

Author Contributions

Gopichand Agnihotram: Writing - original draft, Conceptualization, Data curation

Joydeep Sarkar: Writing - original draft, Investigation, Methodology

Magesh Kasthuri: Visualization, Writing - review & editing

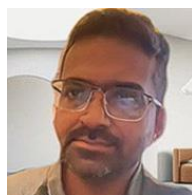
Conflicts of Interest

The authors declare no conflict of interest.

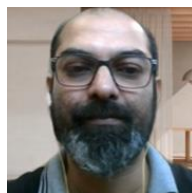
References

- [1] Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J.,... & Neubig, G. (2023, December). Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 7969-7992).
- [2] Raieli, S., & Iuculano, G. (2025). *Building AI Agents with LLMs, RAG, and Knowledge Graphs: A practical guide to autonomous and modern AI agents*. Packt Publishing Ltd.
- [3] Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., & Spanò, S. (2021). Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, 11(11), 4948.
- [4] Chen, Y., Yan, L., Sun, W., Ma, X., Zhang, Y., Wang, S.,... & Mao, J. (2025). Improving retrieval-augmented generation through multi-agent reinforcement learning. *arXiv preprint arXiv: 2501.15228*.
- [5] Bacciu, A., Cuconasu, F., Siciliano, F., Silvestri, F., Tonellotto, N., & Trappolini, G. (2023). RRAML: reinforced retrieval augmented machine learning. *arXiv preprint arXiv: 2307.12798*.
- [6] Zhang, K., Yang, Z., & Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, 321-384.
- [7] Ren, J., Xu, Y., Wang, X., Li, W., Ma, W., & Liu, Y. (2025). Effective and Transparent RAG: Adaptive-Reward Reinforcement Learning for Decision Traceability. *arXiv preprint arXiv: 2505.13258*.
- [8] Jeong, S., Baek, J., Cho, S., Hwang, S. J., & Park, J. C. (2024). Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv: 2403.14403*.
- [9] Li, Y., Luo, Q., Li, X., Li, B., Cheng, Q., Wang, B.,... & Qiu, X. (2025). R3-RAG: Learning Step-by-Step Reasoning and Retrieval for LLMs via Reinforcement Learning. *arXiv preprint arXiv: 2505.23794*.
- [10] Sharma, P., & Mohammad, A. K. (2024, December). Dynamic Retriever Selection in RAG Systems: An RL Approach to User-Centric NLP. In *2024 International Conference on Electrical and Computer Engineering Researches (ICECER)* (pp. 1-6). IEEE.
- [11] Kulkarni, M., Tangarajan, P., Kim, K., & Trivedi, A. (2024). Reinforcement learning for optimizing rag for domain chatbots. *arXiv preprint arXiv: 2401.06800*.
- [12] Singh, A. (2024). Agentic RAG Systems for Improving Adaptability and Performance in AI-Driven Information Retrieval. Available at SSRN 5188363.
- [13] Kamra, V., Gupta, L., Arora, D., & Yadav, A. K. (2025, June). Evaluating Reinforcement Learning Based Models for Test Time Enhancement in RAG. In *2025 Second International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS)* (pp. 254-259). IEEE.
- [14] Pichappan, P. (2025). Adaptive and Enhanced Retrieval Augmented Generation (RAG) Systems: A Summarised Survey. *Journal of Digital Information Management*, 23(3).
- [15] Su, W., Ai, Q., Zhan, J., Dong, Q., & Liu, Y. (2025, July). Dynamic and parametric retrieval-augmented generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 4118-4121).
- [16] Wang, Q., Ding, R., Zeng, Y., Chen, Z., Chen, L., Wang, S.,... & Zhao, F. (2025). VRAG-RL: Empower Vision-Perception-Based RAG for Visually Rich Information Understanding via Iterative Reasoning with Reinforcement Learning. *arXiv preprint arXiv: 2505.22019*.

Biography



Gopichand Agnihotram is a Distinguished Member of Technical Staff (Senior Principal Member) and Director of AI/ML in Wipro Limited. He is associated in AIML and Integrated cloud research, thought leadership, innovative solutions to various clients in Banking, Financial Services, Retail, Insurance and Manufacturing. He is leading the Generative AI Special Interest Group (SIG) in Wipro.



Joydeep Sarkar is Distinguished Member of Technical Staff (Principal Member) & Senior Architect in Wipro Limited. He has worked in numerous client activities in solutioning, proposal, pre-sales, consulting and delivery roles. He is leading the Agentic AI Special Interest Group (SIG) in Wipro.



Magesh Kasthuri is Distinguished Member of Technical Staff (Master) & Chief Architect in Wipro Limited. He has worked with various clients in Banking, Financial Services, Insurance, Capital Markets and Securities, Retail and Media sectors. He is part of Distinguished Member of Technical Staff (DMTS) and leading Cloud Special Interest Group (SIG).

Research Field

Gopichand Agnihotram: Artificial Intelligence, Generative AI, Agentic AI, Fake voice detection

Joydeep Sarkar: Artificial Intelligence, Generative AI, Agentic AI, Fake voice detection

Magesh Kasthuri: Artificial Intelligence, Generative AI, Agentic AI, AI for Data, Cloud technology