Search Results for:

i want to know the latest agentic RAG based approaches

Refined Queries

- Latest advancements in agentbased RAG models, September-06-2024
- RAGpowered agents for task automation and decision making, September-06-2024
- Applications of agentbased RAG models in specific domain , September-06-2024

Search Results

Result 1

Introducing Agent-based RAG. An implementation with LangGraph, Azure… | by Valentina Alto | Aug, 2024 | Medium Open in app Sign up Sign in Write Sign up Sign in Member-only story Introducing Agent-based RAG An implementation with LangGraph, Azure AI Search and Azure OpenAI GPT-4o Valentina Alto · Follow 13 min read · Aug 16, 2024 -- 3 Share Among the various architectural patterns in the field of Generative AI, Retrieval Augmented Generation (RAG) was the first and probably still most polular to be around. RAG is a technique that allows the generative models to access external knowledge sources, such as documents, databases, or web pages, and use them as additional inputs for generating responses. By doing so, RAG can improve the quality, diversity, and reliability of the generated content, as well as provide transparency and verifiability for the users. Over the last months, many variations of RAG have been developed (GraphRAG , Adaptive RAG , Corrective RAG …), with the goal of improving some weaknesses of the “traditional” RAG pipeline. In this article, we are going to see one of these variations: Agentic RAG. Before diving into the topic, let’s refresh how the two main ingredients of this solution — RAG and Agents — are defined. What is RAG? Retrieval Augmented Generation (RAG) is a powerful technique in LLM-powered applications scenarios that addresses the following problem: “what if I want to ask my LLM something that is not part of the training set where the LLM was… Follow Written by Valentina Alto 5.3K Followers Data&AI Specialist at @Microsoft | MSc in Data Science | AI, Machine Learning and Running enthusiast Follow Help Status About Careers Press Blog Privacy Terms Text to speech Teams Unlocking the Power of Agentic RAG — Engineering Business Outcomes | Infogain Unlocking the Power of Agentic RAG — Engineering Business Outcomes | Infogain Posted on : June 14, 2024 Share on Linkedin Share on Twitter Share on Facebook Industry : Corporate Service : Analytics, Data Science and AI Type : Blog We are all familiar with Retrieval Augmented Generation (RAG) by now. RAG is a framework designed to enhance text quality by integrating relevant information retrieved from an external knowledge base into the generated content. By combining retrieval mechanisms with

generative capabilities, RAG produces more accurate, contextually appropriate, and informative text, significantly improving the overall results. Recently, Agentic RAG has emerged as a new and powerful AI technique. In this blog, we'll examine the problems with traditional RAG, then dive into the next advancement in the field of large language models (LLM)—agentic RAG—and explore its features and benefits. A typical RAG pipeline involves: Data Indexing User Query Retrieval & Generation Problems with Traditional RAG: Summarization issues: Summarizing large documents can be tricky. The traditional RAG framework retrieves the top K chunks and may miss crucial information if the document is extensive. Document comparison challenges: Comparing documents effectively is still a hurdle. The RAG framework tends to pull random top K chunks from each document, often leading to an incomplete comparison. Structured data analysis needs: Handling structured data queries, such as determining an employee's next leave based on their region, proves to be a challenge. Accurate retrieval and analysis of specific data points aren't spot-on. Dealing with multi-part questions: Tackling multi-part questions remains a limitation. For instance, identifying common leave patterns across all regions in a large organization is difficult when constrained to K chunks, which limits comprehensive analysis. Now, to overcome these limitations, Agentic RAG comes to the rescue . Agentic RAG = Agent-based RAG implementation Beyond Traditional RAG: Adding Agentic Layers Agentic RAG revolutionizes the way questions are answered by introducing an agent-based framework . Agentic RAG = Agent-based RAG implementation Beyond Traditional RAG: Adding Agentic Layers Agentic RAG revolutionizes the way questions are answered by introducing an agent-based framework. Unlike traditional methods that rely solely on large language models (LLMs), Agentic RAG employs intelligent agents to tackle complex questions that require: Intricate planning Multi-step reasoning Utilization of external tools These agents act like expert researchers, adeptly navigating multiple documents, comparing information, generating summaries, and delivering comprehensive and accurate answers. It's like having a team of specialists working collaboratively to meet your information needs. Whether you need to compare perspectives across documents or synthesize information from various sources, Agentic RAG agents are equipped to handle the task with precision and efficiency. Why Agentic RAG? An AI agent is essential for: Reasoning: Determining which actions to take and their sequence. Task Management: Using agents instead of LLMs directly for tasks requiring planning, multi-step reasoning, tool usage, and learning over time. In the context of RAG: Agents enhance reasoning before selecting RAG pipelines. Improve retrieval or re-ranking processes within a pipeline. Optimize synthesis before responding. This approach automates complex workflows and decision-making for non-trivial RAG use cases. Agentic RAG Benefits: Orchestrated question answering: Breaks down the process into manageable steps, assigns appropriate agents, and ensures seamless coordination for optimal results. Goal-driven: Agents understand and pursue specific goals, allowing for more complex and meaningful interactions. Planning and reasoning: Agents can determine the best strategies for information retrieval, analysis, and synthesis to answer complex questions effectively. Tool use and adaptability: Agents leverage external tools and resources, such as search engines, databases, and specialized APIs, to enhance their information-gathering and processing capabilities. Context-aware: Systems

consider the current situation, past interactions, and user preferences to make informed decisions and take appropriate actions. Learning over time: Agents are designed to learn and improve, expanding their knowledge base and enhancing their ability to tackle complex questions. Flexibility and customization: The framework provides exceptional flexibility, allowing customization to suit specific requirements and domains. Improved accuracy and efficiency: By leveraging the strengths of LLMs and agent-based systems, agentic RAG achieves superior accuracy and efficiency in question answering . Improved accuracy and efficiency: By leveraging the strengths of LLMs and agent-based systems, agentic RAG achieves superior accuracy and efficiency in question answering. RAG Agents can be categorized based on their functions, including routing, one-shot query planning, tool use, Reason + Act (ReAct), and dynamic planning & execution . These functions vary in complexity, cost, and latency and range from simple, low-cost, low-latency tasks to complex, high-cost, high-latency operations. For example: Routing Agents (aka Routers): The routing agent relies on an LLM to select the appropriate downstream RAG pipeline. This process is known as agentic reasoning, since the LLM analyzes the input query to determine the best-fit RAG pipeline. It represents the most straightforward form of this type of reasoning. One scenario may involve choosing between summarization and question-answering RAG pipelines. The agent evaluates the input query to decide whether to route it to the summary query engine or the vector query engine. Query Planning Agent : It simplifies a complex query by dividing it into smaller, parallelizable sub-queries. Each sub-query can be executed across various RAG pipelines that are linked to different data sources. The individual responses from these pipelines are combined to form the final response. In essence, the process involves breaking down the query into manageable parts, executing them across suitable RAG pipelines, and finally merging the results into a coherent response. Several other flows are there based on the functionalities and use cases. Agentic RAG represents a significant advancement in the field of large language models. By incorporating custom agents that can interact with multiple systems, automate reasoning, and dynamically select the best tools for the task at hand, Agentic RAG addresses the shortcomings of traditional RAG. This makes it a more effective solution for handling complex queries and a wider range of use cases. For more information, visit our website and check out Infogain’s analytics, data, and AI services. About the Author Rishabh Kesarwani Rishabh transforms data into actionable insights and innovative products as a data scientist. He specializes in Generative AI applications. In his current role at Infogain, Rishabh works in the agentic space of Generative AI, striving to build innovative solutions within this cutting-edge field LlamaIndex: Building a Smarter RAG-Based Chatbot - PyImageSearch Skip to primary navigation Skip to main content Skip to primary sidebar Skip to footer Chatbot Generative AI Generative Models Large Language Models LlamaIndex Retrieval Augmented Generation Tutorial by Puneet Mangla on September 2, 2024 Click here to download the source code to this post Home » Blog » LlamaIndex: Building a Smarter RAG-Based Chatbot Table of Contents LlamaIndex: Building a Smarter RAG-Based Chatbot Understanding Retrieval Augmented Generation Limitations of Standalone Large Language Models in Production What Is Retrieval Augmented Generation (RAG)? Different Stages of a RAG System Loading Indexing and Storing Querying Evaluating Introduction to LlamaIndex Core Components

of LlamaIndex Data Ingestion Data Indexing Query Engines Evaluating and Benchmarking Observability and Evaluation Building a RAG-Based Chatbot with LlamaIndex for AWS Downloading the Dataset Building the Vector Store Index Loading and Running an LLM Locally Retrieving Context and Querying the LLM Summary Citation Information LlamaIndex: Building a Smarter RAG-Based Chatbot In this tutorial, you will learn how to build a RAG-based chatbot using LlamaIndex. In the dynamic world of artificial intelligence (AI), Retrieval Augmented Generation (RAG) is making waves by enhancing the generative capabilities of Large Language Models (LLMs). At its core, RAG combines the strengths of retrieval- and generation-based models, allowing LLMs to fetch relevant information from a vast database and generate coherent, contextually appropriate responses. One of the latest advancements in this field is the LlamaIndex, a powerful tool designed to boost the capabilities of RAG-based applications further. LlamaIndex provides a very seamless way of combining retrieval- and generation-based models to build RAG-based applications. In this blog post, we'll explore the intricacies of LlamaIndex and guide you through the process of building a smarter, more efficient RAG-based chatbot. To learn how to build a RAG-based chatbot using LlamaIndex, just keep reading

---

Result 2

. Decision models are executed by DMN-compliant engines, ensuring consistent, automated, and efficient decision-making processes. Decision Automation Agents integrate with Business Processes using BPMN standards to enable the seamless automation of end-to-end business processes, further enhancing operational efficiency. Composite AI Decision Automation Agents leverage Composite AI, a powerful approach to decision automation, to harness the synergy of various AI models and techniques. Composite AI enhances the accuracy, reliability, and adaptability of automated decision-making processes. By combining machine learning algorithms, expert systems, risk assessment models, and multi-modal data processing, Decision AIgents make more informed decisions while mitigating risks. Intelligent Automation Decision Automation Agents integrate with Robotic Process Automation (RPA) systems to enable intelligent automation tools by infusing automated processes with data-driven decision-making, contextual adaptability, and compliance assurance. Decision AIgents analyze data, prioritize tasks, and determine the best course of action based on context, predictive models, and prescriptive analytics. When combined with RPA, these decisions guide bots in real time, enabling more efficient and flexible process execution. Decision Controls Decision Automation Agents operate within defined decision controls and constraints to ensure trust and confidence in decision automation systems. These control mechanisms establish boundaries and safeguards to adhere to regulatory compliance, ethical guidelines, and data privacy. Human oversight and explainability are emphasized for critical decisions, and predefined thresholds and fail-safe mechanisms prevent unintended outcomes. Continuous monitoring, feedback loops, and scenario analysis using simulation contribute to ongoing compliance and risk mitigation. Decision Orchestration Decision Orchestration Center streamlines the entire decision-making process, enhancing decision flow, monitoring, governance,

and optimization. Real-time monitoring, alerts, and audit trails provide transparency and compliance, while performance metrics enable organizations to assess decision quality. Decision orchestration leverages data-driven insights, machine learning, and scenario modeling for continuous optimization, allowing organizations to refine decision models and improve outcomes over time. OPTIMIZE BUSINESS OUTCOMES Significant impact on efficiency, accuracy, and business performance Decision automation offers a wide range of benefits, including increased efficiency, reduced errors, cost savings, and improved decision quality. These advantages enable organizations to operate more effectively, respond to challenges more rapidly, and achieve better business outcomes Intelligent Automation Powered By Decision AIgents PRODUCT: Decision AIgent Intelligent Automation with Decision AIgents Decision Automation agents (AIgents) automate decision-making based on decision models, prescriptive analytics, and intelligent automation combines AI/ML methods without human intervention. Decision automation integrates with business applications, workflow management systems, and Robotic Process Automation (RPA tools) to execute actions. GET A DEMO Contact Us POWERFUL FEATURES Automations Empowered by Decision Intelligence Decision Intelligence and Decision Automation Agents (AIgents) empower business teams with data-driven decision-making and operational efficiency. Decision Intelligence leverages composite intelligent automation (IA) models for prescriptive insights. Decision automation streamlines operational decisions with business process automation. Multi-modal interface Decision Automation Agents (AIgents) engage with users and the environment through various communication and input modes, including voice recognition, text-based communication, touch and gestures, computer vision, and audio output. Seamless interactions using different modalities adapt to users' needs, enhancing the overall user experience and making AI interactions more versatile and accessible. Natural Language Processing Decision Automation Agents (AIgents) utilize Natural Language Processing (NLP) capabilities provided by Large Language Models (LLMs) to comprehend, communicate, and process textual information. NLP enables AIgents to analyze and interpret text, recognize sentiment, extract named entities, perform language translation, engage in conversations through chatbots and virtual assistants, retrieve information from large text datasets, classify text into categories, answer questions, transcribe spoken words, and generate human-like text. Data Fabric The Data Intelligence platform and Data Fabric framework provide centralized access to data from diverse sources, facilitating seamless integration and preprocessing of data for artificial intelligence (AI) model training and decision-making. The Data Fabric platform ensures your unstructured data is monitored for quality, security, and governance, scales to accommodate the growing data requirements of AI workloads, enhances data discovery and access using data cataloging and virtualization, and enables real-time data processing. Standard Decision Models Decision Models are created using the Decision Model and Notation (DMN) standard to bolster decision automation by offering a standardized visual framework for representing and executing business decisions. Decision models are executed by DMN-compliant engines, ensuring consistent, automated, and efficient decision-making processes . Automation allows human employees to focus on strategic and creative tasks rather than repetitive, routine decisions. The majority of

employees don’t feel engaged in their work as they often feel that it is too repetitive and not fulfilling. Intelligent automation frees employees from these tedious tasks by automating them, allowing people to focus on more value-adding activities. This can lead to increased innovation and value-added activities. Decision automation optimizes resource allocation by making data-driven decisions on resource allocation, workforce scheduling, and process management. This ensures resources are used efficiently and effectively. GET A DEMO LET'S CHAT FAQ Need clarification? What is Intelligent Automation? Intelligent automation integrates AI and automation to revolutionize business processes. This transformative technology empowers systems to learn, adapt, and autonomously make informed decisions. Embracing intelligent automation unlocks operational efficiency, accuracy, and decision-making speed. It drives productivity and fosters innovation, making it essential in navigating the complexities of the modern business landscape. Incorporate intelligent automation to propel your organization towards unparalleled success. How is Decision Automation/ Intelligent Automation Related to Robotic Process Automation (RPA)? Decision Automation and Robotic Process Automation (RPA) are related concepts in the field of automation, but they serve different purposes and operate at different levels of an organization's processes. Decision automation focuses on automating complex decision-making processes, while RPA specializes in automating repetitive, rule-based tasks. Decision Automation can be integrated into various systems and processes to provide decision support or automate decision-making steps within larger workflows. RPA bots are often integrated into existing applications and systems to perform tasks like a human operator would, interacting with user interfaces. Decision Automation can handle complex decision-making scenarios by analyzing data, considering multiple factors, and applying sophisticated AI and machine learning algorithms to enhance decision quality. While RPA can handle data, its primary focus is on data entry, retrieval, and transfer between systems. Decision Automation and RPA can complement each other within an organization's automation strategy, with Decision Automation guiding processes as part of Intelligent Automation and RPA executing tasks efficiently. How Can We Monitor and Govern Automated Decisions? Automated decisions are rigorously managed and governed by the Decision Orchestration Center to ensure transparency, accountability, trust, and compliance. The Orchestration Center logs every automated decision, maintaining a comprehensive record that includes inputs, logic, and outcomes . The key challenge in the development of MM-LLMs lies in the seamless integration of LLMs with models of other modalities, ensuring coherent operation that aligns with human intentions and understanding. A recent study by researchers from Tencent AI Lab, Kyoto University, and Shenyang Institute of Automation delves deep into this challenge. The study offers a foundational overview of MM-LLMs, from their architectural design to their training pipelines, highlighting the crucial components that facilitate their functionality: 1. Modality Encoder: Transforms input from various modalities into a comprehensible format for the LLM. 2. LLM Backbone: Provides the core language processing and generation capabilities. 3. Modality Generator: Essential for models focusing on multimodal comprehension and generation, translating LLM outputs into diverse modalities. 4. Input Projector: Integrates and aligns encoded multimodal inputs with the LLM, ensuring effective communication. 5. Output Projector:

Transforms LLM output into formats suitable for multimodal expression. The study's comprehensive evaluation of 26 MM-LLMs offers valuable insights into the current state of the field, showcasing the unique features and capabilities of each model. By examining the performance of MM-LLMs against industry benchmarks and in real-world scenarios, the research highlights successful training strategies that enhance MM-LLMs' effectiveness. This exploration into MM-LLMs not only sheds light on the complexities of creating such advanced systems but also underscores their vast potential in revolutionizing AI's role in understanding and generating multimodal content. As MM-LLMs continue to evolve, they promise to unlock new possibilities in AI applications, making interactions with technology more seamless and intuitive. 👉 Explore the full paper on arXiv for an in-depth understanding of MM-LLMs and the future of multimodal AI. #AI #MachineLearning #MultiModalAI #LLMs #ArtificialIntelligence #Innovation 3 1 Comment Like Comment Share Copy LinkedIn Facebook Twitter To view or add a comment, sign in Deepak S 💡Generative AI Alchemist | Crafting the Future 🖥️with LLMs (Llama & Beyond) | LLM | Llama | Langchain | Llamaindex| Amazon Bedrock 6mo Report this post GPT-4 vs . Incremental Data Feeding: Developing processes for incrementally feeding data into the RAG system is crucial. Handling Inaccuracies: Putting processes in place to handle reports of inaccuracies and to correct or delete those information sources in the RAG system is necessary. Automate manual tasks and workflows with our AI-driven workflow builder, designed by Nanonets for you and your teams. Get Started Request a Demo How to get started with creating your own RAG Workflow: Implementing a RAG workflow requires a blend of technical knowledge, the right tools, and continuous learning and optimization to ensure its effectiveness and efficiency in meeting your objectives. For those looking to implement RAG workflows themselves, we have curated a list of comprehensive hands-on guides that walk you through the implementation processes in detail - Nanonets tutorial on building RAG workflows using Llamaindex. Medium tutorial on building a chatbot with GPT and LLMs. Nanonets blog on how to build your own Zendesk Answer Bot with LLMs? Introducting Llamaindex Data Agents . Scalable RAG applications on GCP with Serverless architecture . AWS tutorial on deploying tool-using LLM agents using AWS SageMaker. Streamlit tutorial on building a chatbot with custom data sources . LangChain Agents: Simply Explained! Building a LangChain Custom Medical Agent with Memory . Each of the tutorials comes with a unique approach or platform to achieve the desired implementation on the specified topics. If you are looking to delve into building your own RAG workflows, we recommend checking out all of the articles listed above to get a holistic sense required to get started with your journey. Implement RAG Workflows using ML Platforms While the allure of constructing a Retrieval Augmented Generation (RAG) workflow from the ground up offers a certain sense of accomplishment and customization, it's undeniably a complex endeavor. Recognizing the intricacies and challenges, several businesses have stepped forward, offering specialized platforms and services to simplify this process. Leveraging these platforms can not only save valuable time and resources but also ensure that the implementation is based on industry best practices and is optimized for performance. For organizations or individuals who may not have the bandwidth or expertise to

build a RAG system from scratch, these ML platforms present a viable solution

---

Result 3

Introducing Agent-based RAG. An implementation with LangGraph, Azure… | by Valentina Alto | Aug, 2024 | Medium Open in app Sign up Sign in Write Sign up Sign in Member-only story Introducing Agent-based RAG An implementation with LangGraph, Azure AI Search and Azure OpenAI GPT-4o Valentina Alto · Follow 13 min read · Aug 16, 2024 -- 3 Share Among the various architectural patterns in the field of Generative AI, Retrieval Augmented Generation (RAG) was the first and probably still most polular to be around. RAG is a technique that allows the generative models to access external knowledge sources, such as documents, databases, or web pages, and use them as additional inputs for generating responses. By doing so, RAG can improve the quality, diversity, and reliability of the generated content, as well as provide transparency and verifiability for the users. Over the last months, many variations of RAG have been developed (GraphRAG , Adaptive RAG , Corrective RAG …), with the goal of improving some weaknesses of the “traditional” RAG pipeline. In this article, we are going to see one of these variations: Agentic RAG. Before diving into the topic, let’s refresh how the two main ingredients of this solution — RAG and Agents — are defined. What is RAG? Retrieval Augmented Generation (RAG) is a powerful technique in LLM-powered applications scenarios that addresses the following problem: “what if I want to ask my LLM something that is not part of the training set where the LLM was… Follow Written by Valentina Alto 5.3K Followers Data&AI Specialist at @Microsoft | MSc in Data Science | AI, Machine Learning and Running enthusiast Follow Help Status About Careers Press Blog Privacy Terms Text to speech Teams Unlocking the Power of Agentic RAG — Engineering Business Outcomes | Infogain Unlocking the Power of Agentic RAG — Engineering Business Outcomes | Infogain Posted on : June 14, 2024 Share on Linkedin Share on Twitter Share on Facebook Industry : Corporate Service : Analytics, Data Science and AI Type : Blog We are all familiar with Retrieval Augmented Generation (RAG) by now. RAG is a framework designed to enhance text quality by integrating relevant information retrieved from an external knowledge base into the generated content. By combining retrieval mechanisms with generative capabilities, RAG produces more accurate, contextually appropriate, and informative text, significantly improving the overall results. Recently, Agentic RAG has emerged as a new and powerful AI technique. In this blog, we’ll examine the problems with traditional RAG, then dive into the next advancement in the field of large language models (LLM)—agentic RAG—and explore its features and benefits. A typical RAG pipeline involves: Data Indexing User Query Retrieval & Generation Problems with Traditional RAG: Summarization issues: Summarizing large documents can be tricky. The traditional RAG framework retrieves the top K chunks and may miss crucial information if the document is extensive. Document comparison challenges: Comparing documents effectively is still a hurdle. The RAG framework tends to pull random top K chunks from each document, often leading to an incomplete comparison. Structured data analysis needs: Handling structured data queries, such as determining an employee's next

leave based on their region, proves to be a challenge. Accurate retrieval and analysis of specific data points aren't spot-on. Dealing with multi-part questions: Tackling multi-part questions remains a limitation. For instance, identifying common leave patterns across all regions in a large organization is difficult when constrained to K chunks, which limits comprehensive analysis. Now, to overcome these limitations, Agentic RAG comes to the rescue . Agentic RAG = Agent-based RAG implementation Beyond Traditional RAG: Adding Agentic Layers Agentic RAG revolutionizes the way questions are answered by introducing an agent-based framework . Agentic RAG = Agent-based RAG implementation Beyond Traditional RAG: Adding Agentic Layers Agentic RAG revolutionizes the way questions are answered by introducing an agent-based framework. Unlike traditional methods that rely solely on large language models (LLMs), Agentic RAG employs intelligent agents to tackle complex questions that require: Intricate planning Multi-step reasoning Utilization of external tools These agents act like expert researchers, adeptly navigating multiple documents, comparing information, generating summaries, and delivering comprehensive and accurate answers. It’s like having a team of specialists working collaboratively to meet your information needs. Whether you need to compare perspectives across documents or synthesize information from various sources, Agentic RAG agents are equipped to handle the task with precision and efficiency. Why Agentic RAG? An AI agent is essential for: Reasoning: Determining which actions to take and their sequence. Task Management: Using agents instead of LLMs directly for tasks requiring planning, multi-step reasoning, tool usage, and learning over time. In the context of RAG: Agents enhance reasoning before selecting RAG pipelines. Improve retrieval or re-ranking processes within a pipeline. Optimize synthesis before responding. This approach automates complex workflows and decision-making for non-trivial RAG use cases. Agentic RAG Benefits: Orchestrated question answering: Breaks down the process into manageable steps, assigns appropriate agents, and ensures seamless coordination for optimal results. Goal-driven: Agents understand and pursue specific goals, allowing for more complex and meaningful interactions. Planning and reasoning: Agents can determine the best strategies for information retrieval, analysis, and synthesis to answer complex questions effectively. Tool use and adaptability: Agents leverage external tools and resources, such as search engines, databases, and specialized APIs, to enhance their information-gathering and processing capabilities. Context-aware: Systems consider the current situation, past interactions, and user preferences to make informed decisions and take appropriate actions. Learning over time: Agents are designed to learn and improve, expanding their knowledge base and enhancing their ability to tackle complex questions. Flexibility and customization: The framework provides exceptional flexibility, allowing customization to suit specific requirements and domains. Improved accuracy and efficiency: By leveraging the strengths of LLMs and agent-based systems, agentic RAG achieves superior accuracy and efficiency in question answering . In contrast to conventional methods relying solely on large language models (LLMs), agentic RAG employs these agents to tackle complex questions that demand intricate planning, multi-step reasoning, and the utilization of external tools. These agents function as proficient researchers, skillfully navigating through multiple documents, analyzing information, crafting summaries, and furnishing comprehensive and precise

answers. The implementation of agentic RAG is highly scalable; additional documents can be seamlessly integrated, each managed by a sub-agent. Picture it as having a team of expert researchers at your disposal, each possessing unique skills and capabilities, collaborating to meet your information requirements. Whether you seek to compare perspectives across various documents, explore the nuances of a particular document, or synthesize information from diverse summaries, agentic RAG agents are adeptly equipped to handle the task with accuracy and efficiency. Various usage patterns of agentic RAG Agents operating within a RAG framework demonstrate diverse usage patterns, each finely tuned to specific tasks and goals. These patterns underscore the adaptability and flexibility of agents when engaging with RAG systems. Below are the primary patterns of agent usage within the RAG context: 1. Utilization of existing RAG pipelines as tools: Agents can employ established RAG pipelines to execute particular tasks or generate outputs efficiently. By tapping into these pipelines, agents streamline their operations and capitalize on the framework's inherent capabilities. 2. Autonomous operation as standalone RAG tools: Agents possess the capability to operate independently as RAG tools within the framework. This autonomy enables agents to generate responses directly from input queries, without dependence on external tools or pipelines. 3. Dynamic tool retrieval based on query context: Agents can dynamically retrieve relevant tools from the RAG system, such as a vector index, based on the contextual cues provided by the query. This adaptive tool retrieval empowers agents to tailor their actions according to the specific needs of each query. 4. Query planning across available tools: Agents excel in query planning tasks by analyzing input queries and selecting appropriate tools from a predefined set within the RAG system. This capacity enables agents to optimize tool selection based on query requirements and desired outcomes. 5. Selection of tools from the candidate pool: In scenarios where the RAG system offers a diverse array of tools, agents assist in selecting the most suitable option from the pool of candidate tools retrieved based on the query. This selection process ensures alignment between the chosen tool and the query context and objectives . Improved accuracy and efficiency: By leveraging the strengths of LLMs and agent-based systems, agentic RAG achieves superior accuracy and efficiency in question answering. RAG Agents can be categorized based on their functions, including routing, one-shot query planning, tool use, Reason + Act (ReAct), and dynamic planning & execution . These functions vary in complexity, cost, and latency and range from simple, low-cost, low-latency tasks to complex, high-cost, high-latency operations. For example: Routing Agents (aka Routers): The routing agent relies on an LLM to select the appropriate downstream RAG pipeline. This process is known as agentic reasoning, since the LLM analyzes the input query to determine the best-fit RAG pipeline. It represents the most straightforward form of this type of reasoning. One scenario may involve choosing between summarization and question-answering RAG pipelines. The agent evaluates the input query to decide whether to route it to the summary query engine or the vector query engine. Query Planning Agent : It simplifies a complex query by dividing it into smaller, parallelizable sub-queries. Each sub-query can be executed across various RAG pipelines that are linked to different data sources. The individual responses from these pipelines are combined to form the final response. In essence, the process involves breaking down the query into manageable parts, executing them

across suitable RAG pipelines, and finally merging the results into a coherent response. Several other flows are there based on the functionalities and use cases. Agentic RAG represents a significant advancement in the field of large language models. By incorporating custom agents that can interact with multiple systems, automate reasoning, and dynamically select the best tools for the task at hand, Agentic RAG addresses the shortcomings of traditional RAG. This makes it a more effective solution for handling complex queries and a wider range of use cases. For more information, visit our website and check out Infogain's analytics, data, and AI services. About the Author Rishabh Kesarwani Rishabh transforms data into actionable insights and innovative products as a data scientist. He specializes in Generative AI applications. In his current role at Infogain, Rishabh works in the agentic space of Generative AI, striving to build innovative solutions within this cutting-edge field

---

URLs

Latest advancements in agentbased RAG models, September-06-2024

- https://www.infogain.com/blog/unlocking-the-power-of-agentic-rag/
- https://valentinaalto.medium.com/introducing-agent-based-rag-9b7141ae1cd7
- https://pyimagesearch.com/2024/09/02/llamaindex-building-a-smarter-rag-based-chatbot/
- https://www.harrisonclarke.com/blog/challenges-and-future-directions-in-rag-research-embracing-data-ai
- https://arxiv.org/pdf/2407.19994
- https://www.glean.com/blog/rag-revolutionizing-ai-2024
- https://www.researchgate.net/publication/382655364_A_Study_on_the_Implementation_Method_of_an_Agent-Based_Advanced_RAG_System_Using_Graph
- https://www.securityindustry.org/2024/07/16/understanding-the-evolution-from-classic-chatbots-to-rag-chatbots-to-ai-powered-assistants/
- https://www.superannotate.com/blog/rag-explained
- https://sdtimes.com/ai/rag-is-the-next-exciting-advancement-for-llms/

RAGpowered agents for task automation and decision making, September-06-2024

- https://www.infoq.com/articles/efficient-devsecops-workflows/
- https://nanonets.com/blog/retrieval-augmented-generation-workflows/
- https://kanerika.com/blogs/retrieval-augmented-generation/
- https://circuitry.ai/intelligent-automation-decision-agent
- https://medium.com/@foadmk/enhancing-data-retrieval-with-vector-databases-and-gpt-3-5-reranking-c58ec6061bde
- https://www.nvidia.com/en-us/ai-data-science/products/riva/

- https://fabrity.com/blog/curbing-chatgpt-hallucinations-with-retrieval-augmented-generation-rag/
- https://www.linkedin.com/posts/raphaelmansuy_apple-openelm-an-efficient-llm-family-activity-7188882854523666432-dKOO
- https://liu.diva-portal.org/smash/get/diva2:1846505/FULLTEXT01.pdf
- https://inrule.com/the-benefits-of-decision-automation/

Applications of agentbased RAG models in specific domain , September-06-2024

- https://hyperight.com/7-practical-applications-of-rag-models-and-their-impact-on-society/
- https://www.infogain.com/blog/unlocking-the-power-of-agentic-rag/
- https://valentinaalto.medium.com/introducing-agent-based-rag-9b7141ae1cd7
- https://pyimagesearch.com/2024/09/02/llamaindex-building-a-smarter-rag-based-chatbot/
- https://www.superannotate.com/blog/rag-explained
- https://www.researchgate.net/publication/381461839_Retrieval_Augmented_Generation_RAG_based_Restaurant_Chatbot_wi
- https://www.elastic.co/search-labs/blog/domain-specific-generative-ai-pre-training-fine-tuning-rag
- https://arxiv.org/html/2405.13622v1
- https://www.linkedin.com/pulse/agentic-rag-what-its-types-applications-tarun-gujral-dkqqc
- https://opendatascience.com/7-rag-tools-to-make-the-most-out-of-your-llms/