IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

Agentic RAG: Enhancing Response Accuracy

Sanika Gangaram Sawant
Master of Computer Application (MCA)
Centre for Distance and Online Education(CDOE), Mumbai University, Mumbai

Abstract: *Large Language Models is a framework of artificial intelligence which is trained on vast amounts of data to understand, analyze and generate human-like text Large Language Models LLM's. have significantly enhanced natural language understanding and generation. But due to solely relying on pre-trained data it has some drawbacks such as hallucination and sometimes ungrounded responses. Retrieval Augmented Generation overcomes these issues efficiently but due to it having a fixed retrieval pipeline it retrieves irrelevant information and lacks adaptive, optimized and self-checking paradigm. This paper proposed an Agentic RAG system which mostly overcomes the drawback of traditional RAG. Agentic RAG upgrades conventional RAG by incorporating Autonomous AI Agents. It uses intelligent query processing which helps RAG to understand and analyze user query based on intent, context, etc to improve response accuracy . Document ingestion and embeddings also increased the response accuracy by appropriately ingested useful chunks in vector databases. Retrieval And Reranking are the most crucial stages where relevant*
data is retrieved and reordered their sequence using cross encoder models to only pass significant documents to LLM. Then comes the Generation stage where
these retrieved documents, original user query and set of instructions are provided to LLM to generate accurate and reliable answers. Self-checking and verification agents review the generated response and if it finds the generated answer as low confidence it will rewrite the query and regenerate the response which enhances the response accuracy significantly. As AI applications continue to evolve.

Keywords: Agentic Rag, Retrieval Augmented Generation, Large Language Models, Artificial Intelligence, Response Accuracy, Hallucination Reduction

I. INTRODUCTION

Artificial Intelligence (AI) is a branch of computer science which is invented to create systems capable of performing tasks that typically require human intelligence, such as learning, reasoning, pattern recognition,problem-solving and decision-making.
In general, it is a technology which mimics human cognition and behaviour to solve problems from day to day life to a very difficult engineering task. Also it works as a collection of specialized tools such as:

- Machine Learning (ML): It is a system that learns from a large pool of data to make predictions or decisions without programmed explicitly for every use case.
- Deep Learning: It is a subset of machine learning which is inspired by the structure and functioning of the human brain which consists of multiple layers of neural networks to analyze and solve complex tasks.
- Generative AI: It is one of the famous concepts of Artificial intelligence that generates a content such as text, image by only giving a prompt.

AI has Extensive applications across various sectors including healthcare, education,
E-commerce, robotics, etc. Large Language Models such as GPT, Claude-based models can understand and generate human-like text.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, June 2026 Impact Factor: 8.2

A Large Language Model (LLM) is a next level of artificial intelligence that is designed to acknowledge, process, and generate human language. It has transformed the way of natural language understanding and generation. LLMs are trained on massive datasets which makes them capable of natural language understanding and generation. It is widely used for question answering, summarization, coding, etc. because it gives fast responses, human-like interaction and handling complex queries.

However,traditional LLMs which are deployed by relying solely on pre-trained data struggle with factual accuracy(Hallucinations), temporal awareness, logical reasoning, and processing constraints. Although they provide very impressive results, their responses are not always accurate or reliable because they solely generate it based on learned patterns.

Retrieval-Augmented Generation (RAG) is an artificial intelligence framework that improves the accuracy of Large Language Models (LLMs). It works by fetching facts from external, domain-specific knowledge bases before generating a response, helping the AI provide up-to-date answers and reducing incorrect information (hallucinations).

However, traditional RAG usually follows a fixed procedure i.e. retrieve documents,send them to LLM, generate answers which sometimes results in retrieval of irrelevant documents and lack of autonomous reasoning capabilities.

Because of the above mentioned drawbacks of traditional RAG, Agentic RAG comes into picture. Agentic RAG is a combination of Retrieval-Augmented Generation and AI agents. These agents extends the features of RAG and can:

- analyze user intent
- plan tasks
- decide retrieval strategy
- select tools
- evaluate retrieved information
- verify generated answers

Agentic Rag enhances Response Accuracy by Intelligent Retrieval which means agents decide which documents or information is required, Query optimization, Multi-step reasoning, Response verification and adaptive workflow.This paper focuses on analyzing the role of Agentic RAG in enhancing the reliability, factual correctness, and overall performance of LLM-based systems.

II. PROBLEM STATEMENT

The rapid evolution of Large Language Models (LLMs) has significantly improved the capabilities of artificial intelligence systems in understanding and generating human-like responses. These models are used in applications such as chatbots, information retrieval, content generation and decision-support systems. However, despite their impressive performance, traditional LLM-based systems still face several challenges related to response inaccuracy and reliability.

One of the major drawbacks of LLMs is hallucination, where models may generate responses that appear meaningful but contain incorrect data or information. Since LLMs depend on patterns learned from their training data, they may lack access to updated or domain-specific knowledge. This can result in inaccurate responses.

Retrieval-Augmented Generation (RAG) was inspired to overcome these limitations by coupling external information retrieval with LLM-based generation. By retrieving relevant information from external knowledge sources before generating a response, RAG systems can improve factual accuracy and reduce hallucination. However, traditional RAG architectures generally follow a fixed pipeline where documents are retrieved and directly provided to the language model. This RAG implementation, while promising, suffers from critical failure modes: retrieval noise (fetching irrelevant passages), semantic gaps between query embeddings and document representations, and a lack of multi-step reasoning over retrieved evidence.

To address these limitations, Agentic RAG advances this paradigm by embedding

autonomous AI agents into the retrieval-generation pipeline. These agents then perform by— including planning, tool use, and multi-agent collaboration — to dynamically manage retrieval strategies, iteratively refine contextual understanding, and adapt the workflow based on intermediate reasoning steps.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

III. PROPOSED AGENTIC RAG PIPELINE:

The proposed Agentic RAG pipeline combines Retrieval-Augmented Generation (RAG) with AI agents to overcome the limitations of traditional RAG systems. Instead of only a fixed retrieval and generation workflow like traditional RAG, the proposed pipeline introduces intelligent agents that can analyze queries, make decisions, retrieve relevant information, reason over retrieved data, and verify generated responses. This enhances response accuracy, reduces hallucination, and enables adaptive information processing. The proposed system is a multi stage pipeline .Below we describe each stage in detail.

3.1 Document Ingestion and Knowledge Base creation:

This is the very first and important stage in the pipeline where the raw data is collected and converted into structured and searchable chunks.This data is then store into a structured knowledge base like chroma db, Pinecone, Quadrant, etc. Input sources may include:

- PDFs
- websites
- research papers
- enterprise documents
- databases Document Loaders:
- LangChain Document Loaders
- LlamaIndex Readers

The data is ingested in vector databases by cleaning and processing the unwanted information to prepare it further for chunking and embedding.

3.2 Document Preprocessing and Chunking :

Preprocessing cleans the raw data(like pdf's, csv's, web pages) that is ingested to make it noise free and chunking makes that data into smaller, manageable segments. Large documents cannot directly be processed efficiently by LLMs, therefore they are divided into smaller chunks. The techniques that are used fir chunking are:
Fixed-size chunking

Splits text based on a fixed number of tokens or characters. Example:
Document1
↓
Chunk1
Chunk2
Chunk3

Semantic Chunking

Uses embedding similarity to divide documents based on meaning/semantics. Tools:

- LangChain Text Splitters
- Semantic Chunker

In above, Text Splitter is widely used for chunking.For Ex:
Splitter = RecursiveCharacterTextSplitter(chunk_size=500, chunk_overlap=200)
In this, we are dividing the text of documents in 500 chunks and overlap means after each page the first page's content is overlapped with the second page's content with 200 words so the meaning of embedding will never be lost.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

3.3 Embedding Generation Stage

Embedding Generation is the process where the unstructured data such as images, texts, audios, etc is converted into a dense numerical vector. Embedding models then capture semantic relationships between words and sentences.
Technologies that can be used for embedding generations:

OpenAI Embeddings

- text-embedding models

NVIDIA Embedding Models

- NVIDIA NeMo embedding models

Open-source models

- Sentence Transformers
- BGE Embeddings Frameworks:
- LangChain
- LlamaIndex

3.4 Vector Database Storage

Now the embeddings that are generated in earlier stages are stored in vector databases. Some of the popular vector databases are:

ChromaDB

Used for:

- lightweight applications
- local RAG systems

FAISS (Facebook AI Similarity Search)

Used for:

- large-scale similarity search
- efficient vector indexing

Pinecone / Weaviate

Used for:

- cloud-based scalable RAG applications Vector indexes use algorithms such as:

Approximate Nearest Neighbor (ANN)
This algorithm is widely used to quickly find similar vectors

3.5 Query Understanding and Query Rewriting Agent

In this stage, the agent carefully analyzes and understands the query to identify:

- intent detection
- ambiguity detection

Query Rewriting

This is very helpful to improve original query as sometimes LLM cannot understand pronouns mentioned in the query.
Example:
User:
"What is AI?" Agentic RAG:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, June 2026 Impact Factor: 8.2

"Artificial intelligence is the simulation of human intelligence" User:
"What are its uses?" Rewritten:
"What are the uses of Artificial intelligence?"

3.6 Retrieval Stage:

In the context of RAG, it is important to efficiently retrieve relevant documents from the data source. There are several key issues involved, such as the retrieval source, retrieval granularity and pre-processing of the retrieval. The retrieval agent selects relevant documents from vector databases using one of the following algorithms:

1. Similarity Search:

Similarity search (or vector search) is a data retrieval technique that matches a query using meanings or semantics of the query instead of matching with exact words. It ranks the vector embeddings by their mathematical distance. It calculates the distance metrics using cosine similarity, euclidean distance or dot product.

2. Hybrid Search:

Hybrid Search is also a data retrieval technique that integrates traditional keyword based search with semantic search. So, it will help to match a query using the exact keyword as well with the meaning of query or semantics.

3.7 Reranking Stage :

Reranking is also a data retrieval technique that improves the retrieval quality by reordering a set of retrieval documents that are retrieved earlier in the retrieval stage. It evaluates the direct relationship between user query and each document and brings the most relevant information to the top to improve the accuracy of Agentic Rag.This improves the quality of retrieved results. To achieve this reranking we can use Cross Encoder Models such as:

- BGE Reranker
- Cohere Rerank

These models then refine the retrieved documents and score them down to the final top 3 or 5 results.The highest scoring documents are passed to LLMs. Reranking improves context quality and reduces irrelevant information. Additionally, R2AG (Retrieval information into RAG) extends this stage by recursively reranking candidates during generation itself.

3.8 Context Optimization Agent :

Context optimization is a process where the information that is being fed into LLM is curated strategically ,compressed and managed well to maximize the accuracy of the generated result. This agent removes duplicate information, summarises documents, selects important passages and manages context windows.

It also handles token optimization to reduce API costs and improve response latency. It can be achieved by using key techniques such as prompt compression, utilizing API prompt caching and leveraging vector search for on-demand context.

3.9 Response Generation Stage :

This is the very crucial stage which uses the LLM to create the final answer. We can use any of the following LLM Model for this stage:

Closed-source:

- GPT-4
- Claude
- Gemini Open-source:
- Llama models
- Mistral
- Qwen

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, June 2026 Impact Factor: 8.2

LLM analyzes the prompts, interprets its meaning, and synthesises human-like text to provide an answer or accomplish a task. In this stage, LLM receives a user query that is entered by the user in the system, retrieved context that is being retrieved through all the earlier stages and a set of instructions or rules set by the developer according to the application where this agentic rag is used. With the help of these, LLM generates a grounded response.

3.10 Agent Driven Self-Verification Stage :

Here comes the major difference between traditional RAG and Agentic RAG. In this stage, the Self-Verification Agent evaluates the generated response. It will use the following
techniques for this:

Confidence Checking

The agent evaluates:

- confidence score
- source agreement
- factual consistency

Groundness Checking:

Checks if the response is truly generated by retrieved documents.

Self Reflection :

In this, the agent reviews its own answer and if it finds its confidence is low, it will rewrite the query to get a revised response.

IV. METHODOLOGY

4.1 Query Processing Phase

The Query Processing Phase is crucial for understanding, analyzing and refining the user's query before retrieval begins. In the proposed Agentic RAG framework, an intelligent agent analyzes the user's intent by understanding the query, extracts important keywords from it, and determines whether query rewriting or expansion is required. This helps improve retrieval accuracy of Agentic Rag by converting complex, ambiguous or incomplete queries into more precise search requests. LLMs such as GPT-4, Llama, or Claude can be used for query understanding and rewriting, while frameworks like LangChain and LlamaIndex
supports the implementation of agent-driven query processing workflows.

4.2 Intelligent Retrieval Phase:

Fetching of the most relevant information from the knowledge base sources, based on the processed user query is done by the Intelligent Retrieval Phase. Traditional RAG systems generally rely on a fixed retrieval mechanism, but the proposed Agentic RAG frameworkutilizes retrieval agents to dynamically select retrieval strategies, tool selection and optimize search results. The system performs semantic similarity search using vector embeddings that count numerical distances stored in vector databases and may also employ hybrid retrieval techniques that combine keyword-based(lexical) and semantic search. This successfully retrieves contextually relevant documents, improving the quality of information provided to the language model and reducing the chances of generating inaccurate responses.

Technologies\Algorithms	Purpose
ChromaDB	Vector storage and Retrieval
Cosine Similarity	Semantic Similarity Matching
BM25	Keyword-based Retrieval
Hybrid Search	Combines semantic and keyword Retrieval

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, June 2026 Impact Factor: 8.2

4.3 Document Reranking and Context Management Phase

After the retrieval stage is done, the system may fetch multiple documents that vary in relevance and quality. Therefore, the proposed Agentic RAG framework incorporates a

Document Reranking and Context Management phase to ensure that only the most relevant and accurate content is provided to the large language model. The reranking module evaluates retrieved documents and assigns relevance scores based on their relationship to the user query using cross encoder models. Documents with higher relevance scores are prioritized, while less relevant results are discarded.

Once the documents are reranked, the Context Management module further optimizes the retrieved information by removing duplicate content, selecting important passages, and compressing contexts.Technologies such as BGE Reranker, Cohere Rerank, Cross Encoder models, LangChain Compression Retriever, and LlamaIndex Postprocessors can be used to implement reranking and context optimization functionalities within the proposed system.

4.4 Response Generation Phase

The Response Generation Phase produces the final answer using the user query and the optimized context obtained from the previous stages. In the proposed Agentic RAG framework, the documents that are obtained after retrieval and reranking are provided to a Large Language Model (LLM), which uses this as an external knowledge to generate contextually relevant and factually grounded responses.Instead of traditional LLM systems that rely solely on pre-trained knowledge, the proposed approach adds retrieved information during generation, reducing hallucinations and improving response accuracy.

We have used the “meta/llma-3.3-70b-instruct” model as LLM.

The generation process can be further enhanced through agent-driven decision making, where agents dynamically select suitable models, prompting strategies, or reasoning techniques based on the complexity of the query. By combining the user's query, set of instructions with verified contextual information, the system produces more reliable and relevant responses.

4.5 Agent-Driven Verification Phase:

The Agent-Driven Verification Phase serves as the ultimate assurance level for the proposed Agentic RAG framework. After the response is generated by the Large Language Model using a user query, retrieved documents and set of instructions, a verification agent evaluates its accuracy, relevance, and confidence with the retrieved knowledge. The agent checks the groundness of the response by evaluating the response with actual retrieved documents.

Traditional RAG systems directly return the generated response to the user,whereas the proposed framework incorporates an autonomous verification paradigm that can initiate corrective actions when low-confidence or unsupported responses are detected. These actions include query rewriting, additional document retrieval, reranking, or response regeneration needed by the requirement. By incorporating an iterative feedback loop, the verification agent enhances factual correctness, improves reliability, and ensures that the final response aligns with the available evidence.

4.6 System Design and Implementation: Technology Stack:

Component	Tool/Library	Alternatives
Orchestration	LangChain	LlamaIndex, Haystack, LangGraph
Vector Database	ChromaDB	Weaviate / Pinecone
Embedding Model	nvidia/nv-embedqa-e5-v5	OpenAI text-embedding-3
Retrieval	Hybrid Search	BM25 (Elasticsearch)
Reranker	BGE-Reranker	Cohere Rerank 3.5
LLM (Generator)	llma-3.3-70b-instruct	GPT-4o / Claude 3.5

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

V. EVALUATION AND METHODOLOGY

5.1 Benchmarks and Datasets :

We evaluate the proposed pipeline of Agentic RAG on three standard benchmarks commonly used in RAG research:

- **Natural Questions (NQ)**
  - Real-world questions from most searched platforms like Google searches.
  - Measures factual question answering and accuracy.
- **HotpotQA**
  - Multi-hop reasoning questions require evidence across multiple documents.
  - Useful for testing agent reasoning capabilities and performance.

These benchmarks together assess both single-hop factual retrieval and complex multi-step reasoning, which are the two primary objectives that are widely used for assessing retrieval-augmented systems.

5.2 Evaluation Metrics :

We report the following metrics across all evaluated systems:

Hallucination Rate: It calculates the total percentage of unsupported and inaccurate information gathered or generated.

Groundedness Score: It measures whether responses are actually generated using the retrieved documents.

Response Relevance : Calculates whether the generated response aligns with the user query.

Precision : Calculates how many retrieved documents are relevant with the user query to further improve similarity search.

5.3 Baselines:

Our pipeline is compared against these baselines:

1. Standalone LLM :

Large Language Models generate responses solely on pre-trained data without using external information sources. We can use models such as GPT-4, Llama, Mistral for this. While they can generate very good responses still we can not rely for full accurate output due to their hallucination and knowledge limitations.

2. Traditional RAG:

Traditional RAG systems generate responses by retrieving relevant documents from vector database and providing them as context to LLM to generate the final response.This makes it better than standalone LLM's as this reduce hallucination still it follows fixed retrieval pipeline and lacks the advance techniques such as dynamic query optimization, reranking and self verification.

5.4 Expected Results:

It is expected that the proposed Agentic RAG framework will outperform traditional RAG systems in terms of retrieval quality, response accuracy, factual grounding and hallucination reduction.

We anticipate the following improvements of our full pipeline as:

Metric	LLM only	Traditional RAG	Proposed Agentic RAG
Hallucination Rate	High	Medium	Low
Groundness Score	Low	High	Very High
Response Relevance	Medium	High	Very high
Precision	Low	Medium	High

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

VI. DISCUSSION

6.1 Impact of Each Stage :

Each stage in the proposed pipeline plays a significant role in enhancing the quality of generated responses.The document ingestion and embedding stages helps us to create structured knowledge base that helps to enhance efficient information retrieval. Query processing enhances understanding user intent and helps retrieve more relevant information. Intelligent retrieval ensures access to contextually aligned documents, while reranking and context management further refine the retrieved information by prioritizing highly relevant content and removing redundant data by using relevance scores calculated by algorithms such as BGE ranker. The response generation stage utilizes this optimized context along with retrieved documents and set of instructions to produce grounded responses, whereas the agent-driven self-verification stage which includes query rewriting, etc. serves as a quality control mechanism by detecting hallucinations, inconsistencies. These stages contribute to improved accuracy, relevance, reduce hallucination and user trust.

6.2 Limitations

Despite its advantages, the proposed pipeline can have several challenges. Use of multiple agents in the RAG system can increase API costs which further leads to increased computational cost. With overall additional changes such as reranking, query optimization the proposed system can face high response latency. Overall generated response is dependent on retrieval quality. The proposed system can face large memory requirements and scalability challenges also.

6.3 Ethical Considerations

The deployment of Agentic RAG systems raises several ethical concerns related to fairness, transparency and privacy. This proposed pipeline relies on external knowledge sources, so the chances of biassed information is non-negotiable.Privacy concerns may also arise when processing proprietary data.

VII. FUTURE WORK

Although the proposed Agentic RAG pipeline reduces hallucination and improves response accuracy, still it has room for improvements. Future research may introduce a multi-agent Rag system which dynamically does many of the things.Also the future research may focus on the areas such as multimodal RAG, automated fact-checking agent and adaptive learning agents. Research on improved scalability and efficiency will also be needed.

VIII. CONCLUSION

As discussed earlier, Large Language Models have significantly enhanced natural language understanding and generation. But due to solely relying on pre-trained data it has some drawbacks such as hallucination and sometimes ungrounded responses. Retrieval Augmented Generation overcomes these issues efficiently but due to it having a fixed retrieval pipeline it retrieves irrelevant information and lacks adaptive, optimized and self-checking paradigm.

This paper proposed an Agentic RAG system which mostly overcomes the drawback of traditional RAG. It uses intelligent query processing which helps RAG to understand and analyze user query based on intent, context, etc to improve response accuracy . Document ingestion and embeddings also increased the response accuracy by appropriately ingested useful chunks in vector databases. Retrieval And Reranking are the most crucial stages where relevant data is retrieved and reordered their sequence using cross encoder models to only pass significant documents to LLM. Then comes the Generation stage where these retrieved documents, original user query and set of instructions are provided to LLM to generate accurate and reliable answers.

Self-checking and verification agents review the generated response and if it finds the generated answer as low confidence it will rewrite the query and regenerate the response which enhances the response accuracy significantly. As

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

AI applications continue to evolve. With the increased demand of AI applications, Agentic RAG is one of the crucial development areas for next generation decision support and knowledge intensive systems.

REFERENCES

1. https://www.geeksforgeeks.org/artificial-intelligence/what-is-agentic-rag
2. https://www.ibm.com/think/topics/agentic-rag
3. https://developer.nvidia.com/blog/traditional-rag-vs-agentic-rag-why-ai-agents-need-dynam ic-knowledge-to-get-smarter/
4. https://medium.com/@jagadeesan.ganesh/agentic-rag-with-langchain-revolutionizing-ai-wit h-dynamic-decision-making-ff1dee6df4ca
5. https://www.superannotate.com/blog/agentic-rag
6. https://medium.com/@bholaynathsingh335619/how-agentic-ai-is-different-from-rag-fb1c9e cc433b
7. https://fractal.ai/article/rag-vs-agentic-ai/