iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IX **Month of publication:** September 2026

DOI: https://doi.org/10.22214/ijraset.2026.84791

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IX Sep 2026- Available at www.ijraset.com

Transforming Intelligent Systems through Retrieval-Augmented Generation: Opportunities and Emerging Challenges

Mallikarjunarao Sunke[1], Srikanth Gudi[2], Sriharsha Gudi
[1]*Individual Researcher, 4821 San Mateo LN, NE, Albuquerque, NM, 87109*
[2] *Individual Researcher, Senior Software Engineer, 1251 Parkington Ave, Sunnyvale, CA, 94087.*
[3]*Individual Researcher, Senior Data Engineer, 11723 NE 117th CT, Kirkland, WA, USA 98034.*

Abstract: ***Thanks to the development of basic models and the high quality of the data, the emergence of AI-generated content has accelerated. Despite its incredible success, there are still challenges that are yet to be addressed in AI content generation, such as the processing of long-trail information, maintaining up-to-date knowledge, addressing high inference and training expenses, and addressing data leakage. The shift to address those challenges has been called Retrieval Augmented Generation (RAG). RAG has brought the process of gathering information, which improves the process of data generation by recovering relevant information from available data sources, resulting in robustness and accuracy. RAG has become the foundation of Natural Language Processing (NLP) to effectively fill the gap between factual accuracy of knowledge and fluency of "Large Language Models (LLMs)". This study traces the root of RAG from its beginnings as a framework for knowledge-based tasks to its current state as an agentic, modular and complex runtime of knowledge. This study is an in-depth analysis of the evolution of Naïve RAG to Modular and Advanced RAG models, and the introduction of new innovations, such as self-reflection, dense vector recovery, and the use of different models. They are then examined to provide detailed feedback on how to make RAG truly dynamic and usable as a verifier when applying them to organizations.***

Keywords: ***Natural Language Processing, Retrieval Augmented Generation, Large Language Models, AI models, AI-generated content.***

I. INTRODUCTION

The "Large Language Models (LLMs)" (Husain et al, 2019), are a brilliant advancement in NLP, Understanding and Generation which falls under the "Artificial Intelligence (AI)" segment. The architectures such as Meta LLaMA, Google's PaLM and OpenAI's GPT, which are heavily text-driven, with billions of parameters and can create text that appears human-like in a multitude of applications, are demonstrated to be possible. Despite their impressive development, one of the most common problems with LLM is difficulty maintaining factual consistency, acquiring information and revising it appropriately with dynamic information. The major problems with static corpora are: ageing, misinformation, propagation, hallucination and the issue of not accepting real time updates. The "Retrieval-Augmented Generation (RAG)" has been emerged with these challenges to improve LLMs by retrieving information into the process of generation. In this regard, RAG has been employed with LLM for creating a hybrid system that utilizes LLM with additional new information and data accuracy (Husain et al, 2019). Their parameters do not only produce texts from the parameters, but also from other knowledge sources outside of the parameters, such as search engines, databases or knowledge graphs (Husain et al, 2019). The retrieval mechanism helps to create content that is very relevant, based on concurrent and verifiable data—that's a very positive effect on the factual accuracy of the content. The birth of AI-based language systems has come in a radical change from the key challenges of the parameter storage based retrieval systems, with the help of retrieval-oriented augmentation. There are a number of aspects to Innovation in the RAGs.

The first is that LLM are soon to be obsolete due to their training on static corpora, and thus their lack of access to new information during training (Berant et al, 2023). Rags can also be used to add new knowledge as it progresses, as required in some fields where knowledge is constantly changing, such as the medical field, legal, finance and others. Secondly, another problem that crops up is that models can only be trained so many times and if there's a lot of data to train with, there will be a lot of computing power and power consumed. Models are continuous solutions that are sustainable and scalable; can be used to build up their knowledge base without re-training.

©IJRASET: All Rights are Reserved

Thirdly, there are topics of high-stakes output verifiability and output interpretability problems. In the case of text generation, external sources will be referenced explicitly and the user can easily check on the credibility of the generated text and trace its source through RAG-based tools (Bang et al, 2025).

RAG consists of two major components - Generation and Retrieval. In the retrieval phase, the knowledge snippets or the relevant documents could be retrieved using the query provided by the user, retrieval models like 'dense passage retrieval (DPR)', neural search models or BM25 (Wang et al, 2024). These are then passed on to the intermediate/prompt symbols which are then used to control the next part of the text generation process (Bai et al, 2022). Many implementations of RAG exist that have different methods for using data they retrieve in the prompt; some models include texts retrieved in the prompt, some change frameworks based on facts retrieved (Karpukhin et al, 2018). For the current architectures the re-ranking systems and reinforcement learning and/or attention based fusion can be implemented to ensure the relevance of information, and optimize the retrieval of the QOD. One direction of research that is important for RAG is to increase its efficiency of data retrieval, and keeping high fidelity (Saha et al, 2023) is crucial. The traditional retrieval mechanisms do not offer a mechanism for capturing the semantic relations, e.g., keyword search (Saha et al, 2023).

It is observed that the problem can be tackled using a neural approach, where documents can be matched with queries with increased accuracy, by employing dense vector. Hybrid strategies, distillation of knowledge and contrastive learning are other methods which enhance the capability of models in synthesizing and retrievinging high quality data. Other issues are latency and computational complexity if it is necessary to retrieve dynamic data efficiently, which needs to be indexed, hardware accelerated and cached. Important implications for domain related applications and personalisation, and factual consistency. In the past, AI assisted humans to learn from their interactions with AI, adjusting to the user's context and displaying the AI's answer to the user. In legal and biomedical research, LLM can be helpful for fetching good quality information which is not present in their training sets. With this adaptability, RAG emerges as a major innovation to enhance the applicability of LLMs around wide range of academic and professional areas (Saha et al, 2018).

Some of the challenges to RAG are: Retrieval noise – some of the documents retrieved might contain conflicting and inaccurate information, which can affect the quality of the generated text.Even though RAG offers several benefits (NVIDIA, 2025), there are several problems with it as well, including Retrieval noise, where the retrieved documents may include contradictory and inaccurate information, which can affect the quality of the generated text. Moreover, RAG models must be able to deal with the compromise and latency. Important research is robustness and security in retrieval mechanisms as hackers may be able to access external sources that could generate outputs. The filtering system, retrieval method, and countermeasures need to be further developed to cope with these challenges (Ma et al., 2023).

This study provides a comprehensive concept of RAGs, its phases of evolution, its applications and comparison of the different models. This study's results are aligned with previous studies and deployments, and can be a platform for practitioners, researchers and policymakers in this ever-changing field of AI systems.

II. LITERATURE REVIEW

To do this, several robust text-generating models have been developed, and applied in a range of different contexts, including more recent technology, known as "Large Language Models (LLMs)" which are now being developed. Understand how to prevent getting caught up in bias and misinformation and how to be original and responsible in creating content with AI. Neha et al (2025) provided much detailed knowledge of capabilities, ethical issues and evolution of "AI Text Generators (AITGs)". Recently they also introduced a new paradigm—RAG which will improve the accuracy and relevance of generated text, by retrieving dynamic information, based on the context. RAG does not necessarily have to be based on static data, can be based on real-life data. They also discuss the knowledge about difference between AI Generated content and Human generated content and ethics.

Using RAG it can be robust and correct at a high level in the information it can retrieve from the data source(s) that it has. Han et al (2025) delved into the significance and impact of RAG in LLM. These LLMs can also help to automate tasks such as summarization or data extraction, and trend identification. However, LLM is not stateful, as heavily relying on pre-trained data and it can produce text, likewise be able to understand the subtlety of text. But, with good generation capabilities of LLMs, RAG can overcome these challenges. The crucial components of RAG model are retrieval, generation and augmentation. They also shared their thoughts on whether or not it would be possible to automate this with this LLM, and further studies that should be conducted.

To assess the capabilities of the four RAG chatbots, Parekh et al (2025) then used a new data set to compare the multiple LLM models and the responses of LLM to that of the four RAG chatbots. It also evaluates and compares the performance and answers of Claude and GPT-4o models as well as Llama and DeepSeek models on a number of corpora, involving complex content and

structure of the questions. An open-source database (Chroma) will be used to house ALL RAGs. Collected Documents and Query: The document(s) and query the answer to the question posed to the LLM is based on. They monitored the activity of each of the chatbots, how each bot responded to the users and then used the traditional means to evaluate each bot's response quality.

Currently, RAG is being employed to feed into real-time information to boost LLM's capabilities and supply it with more relevant and timely information. In terms of workflow, however, Traditional RAG systems are less flexible when performing multiple steps, and have a negative impact with respect to reasoning. The limitations are alleviated when Autonomous AI is added to RAG. In this article Singh et al (2025) introduces Agentic RAG systems, explores the evolution of Agentic RAG systems, and provides a taxonomy for Agentic RAG systems, categorizing it into three types: Control structure, Cardinality of the agents, Representation of knowledge and Autonomy. They additionally explored practical illustrations of the application of AI in the finance, education, health and document processing industries (Khapra, 2022). The Artificial Intelligence has been changed the world of technology after a couple of decades. AI applications can revolutionize businesses with the support of various other technologies such as RAG systems, Convolutional Neural Networks (CNNs) and LLMs. The LLM has been improved to process natural language, and even form and communicate natural language by interacting with humans. The generative abilities are used to improve the accuracy and relevance of the content produced by RAG systems, by leveraging external knowledge. CNN applications in computer vision image processing, however, could be much more complex as Khapra et al (2024) and Shukla et al (2024) point out respectively. In above implementations Ramdurai(2025) does the analysis of Implementation and Problems faced in above implementation.

III. EVOLUTION STAGES OF RAG

The new frontier of times is indeed Retrieval-Augmented Generation (RAG) and it has grown up to an agent and modularity. The new systems are not only consecutive (fixed pipeline), but they also attempt to overcome a smart flexible system architecture that rewards an LLM agent which can get a score for a sequence of given modules of the toolkit. This agent can make choices on how to retrieve (multiple strategies), make several steps of reasoning, plan and even repeat steps of reasoning if it is wrong (iterative loop). The pattern allows RAG to be flexible with the reasoning process of more complex and multi-faceted questions which require verification and summarization and requires different levels of reasoning and decision making. Therefore, RAG is developed to closely follow the trajectory of the AI mentioned by Fezari and Al-Dahoud (2025) from simple AI to integrated AI tools and devices which are meant to “drink in” the world of knowledge.

There are four different stages of journey of RAG:

A. Naïve RAG

If there is a system provided it will be converted to a vector embedding. A second vector database (Dogan, 2024) then is used to perform similarity search on the chunks of the document and assemble the chunks based on a user prompt. A new RAG has emerged – Naïve RAG. A very widely used and user friendly basis line in pipelining. Firstly, from a knowledge base a certain number of contexts/chunks is retrieved by simple semantic similarity search based on the query of a user. In this case, this context is added into the first query, irrespective of the context. Finally they are presented with this enhanced question, with the last documents, providing them with the final answer. It can be useful in certain situations, but there is a potential issue: If this person gets information that they have no intention of taking any action on, it is not going to help eliminate the number of search results, or if they get a lot of information that is contradictory/duplicated, it may not be a very enjoyable experience. However, there are a number of issues with the naive RAG, the most prominent being the ‘lost in the middle' issue in which it fails to search for information in longer contexts (Kalotra, 2025).

B. Vanilla RAG

But, as shown in the figure, without AI planning, AI self-reflecting or AI agents, Vanilla RAG was the most practical and easy-to-use of the various RAGs. It is the most accessible & easiest to make type of ALL!, due to its ease of access and simplicity of creation. The one which looks something like Google or search engine + ChatGPT, but doesn't have any other agenda but to get them to achieve. I can only assume it's reliable on that aspect as well – timely, relevant data and answers to my questions about the data on the model. Here they don't have to write the complicated query optimisation, complicated loops etc to pull/push it out, it is pulled here without having to call any agents. The difference between them, and human employees (RAG models) is the intern does only what he/she is asked to do while the RAG models are human employees and they do what they are asked to do. It's not only a very fast framework, it's also has a low latency as well. It's in addition cost-efficient to more intricate models. Also, it's very easy to use in very simple and simple queries.

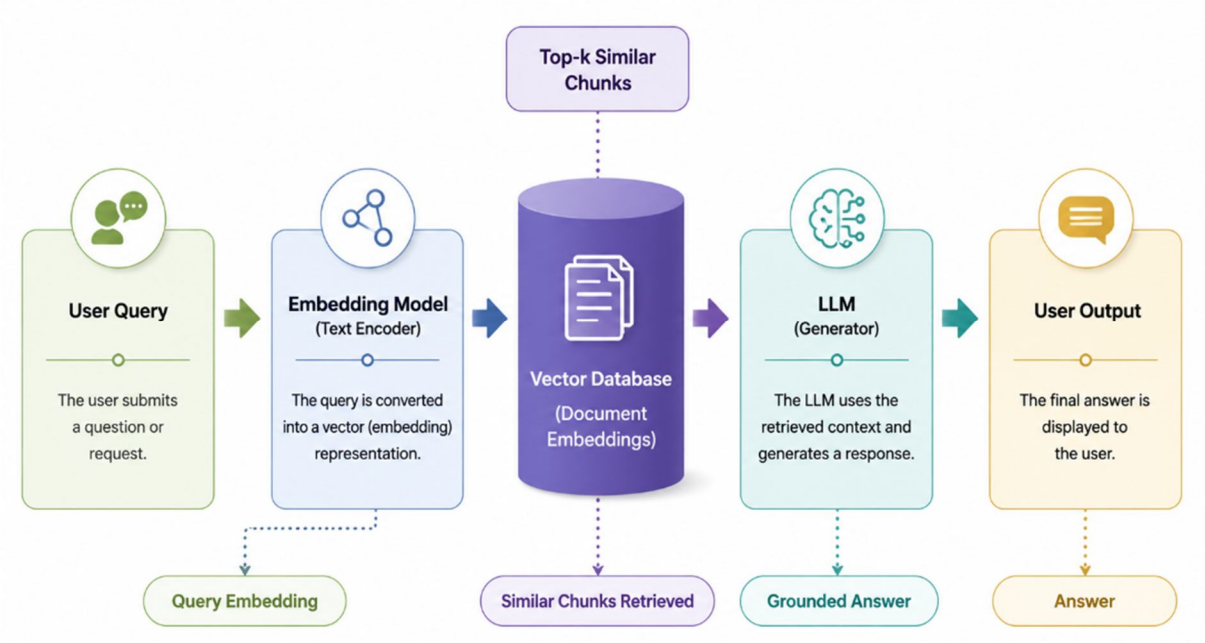


Figure 1 – A Flowchart of Vanila RAG Framework
Source – Fezari and Al-Dahoud (2025)

C. Advanced RAG

That is, it's simply a RAG model with the appropriate improvements added to it throughout the pipeline, to overcome some of the naive RAG adoption pitfalls. This can be extended to other tasks, such as retrieval, because it is based on query expansion, query rewriting, use of hybrid search (Sparse + Dense) and the benefit of smart re-ranking can be used for very relevant aspect.

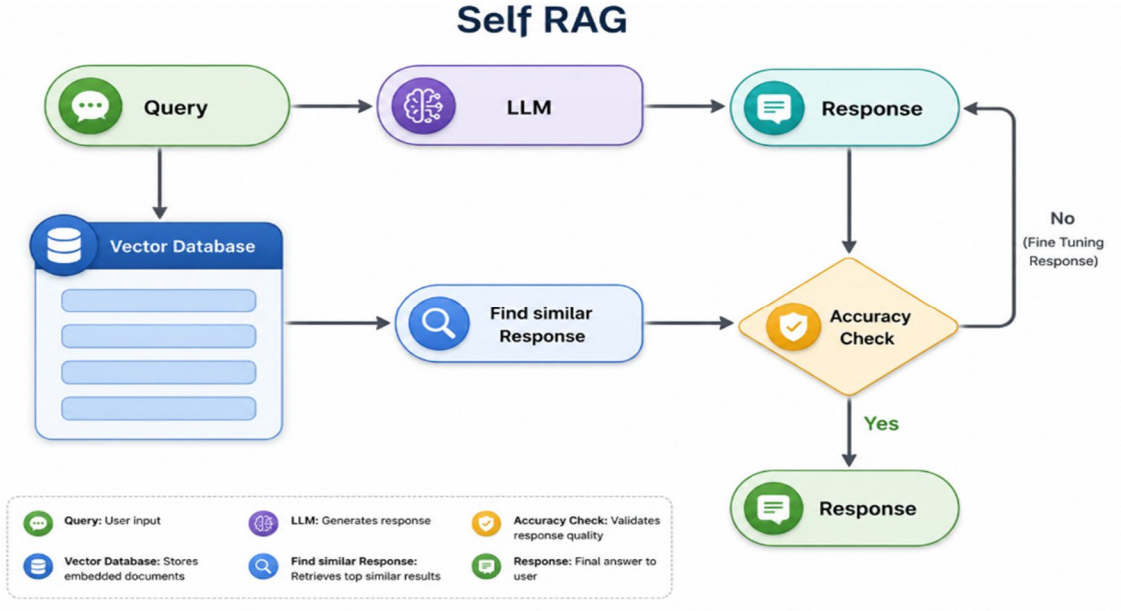


Figure 2. A flowchart of Advanced RAG
Source – Fezari and Al-Dahoud (2025)

The old paradigms of approach are mentioned but it is suggested that pre- and post-retrieval optimizations (Advanced RAG) be aimed at. Pre-retrieval: In the case of the relation, embedding of documents, query are used both documents and queries are similar (Naminas 2025). To make sure that only high quality data is retrieved and processed by LLMs, chunks are reranked by relevance.

D. Agentic/Modular RAG

Currently RAG is on the Agentic and Modular stage. These are modern systems which are then repeated and self-correcting (Karlsson, 2025.) is carried out. They can use the outcome of their models as corrective RAG (CRAG) and/or self-RAG to help them to decide if further search is required or if a back up is required.

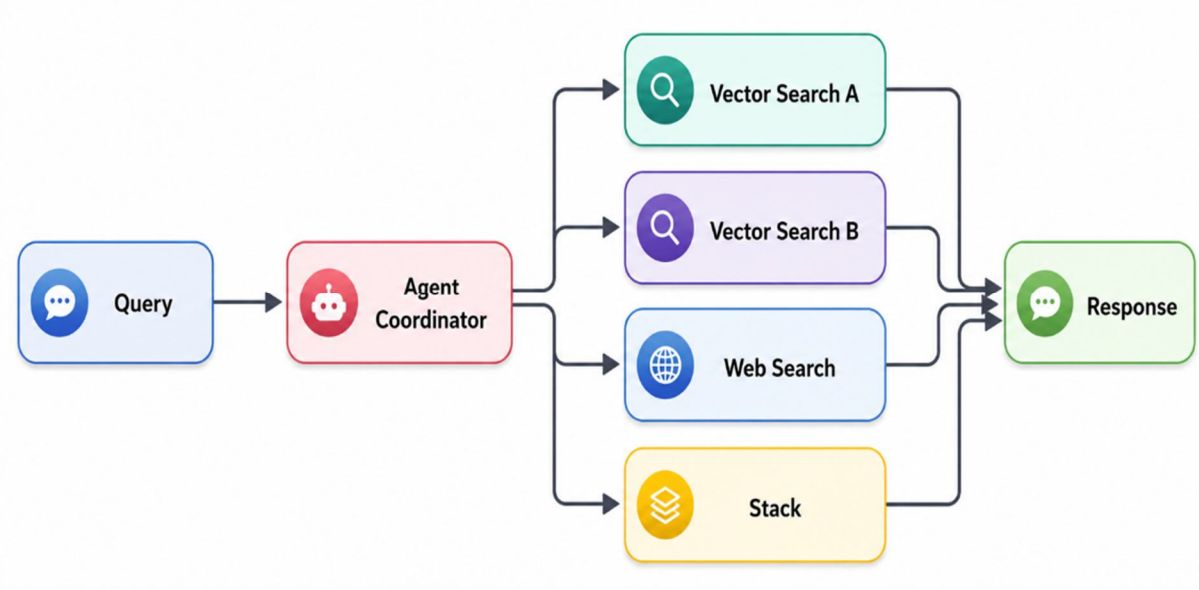


Figure 3 – A Flow Diagram of Agentic RAG
Source - Fezari and Al-Dahoud (2025)

Agentic RAG flowchart: is a flowchart which shows the process of Agentic RAG. Both Agentic RAG and Modular RAG are out-of-the-box explorations of new spaces for RAG, looking towards a consistent and static pipeline towards an objective-based paradigm with compositional knowledge. The RAG is composed of 4 different special architectures, namely the retrievers, query planners, response generators and re-rankers, all of which will be the controller/agent architecture. Agentic (or Modular) agents can be built atop of LLM agents to reason and decide, synthesize, retrieval and refine in a loop to provide the correct answer. There may be many steps in a RAG system, and a degree of intelligence and autonomy can be useful even if it is merely to retrieve data from a document—it could also be the starting point to create smarter and more autonomous AI assistants of the future that can perform actions and make decisions.

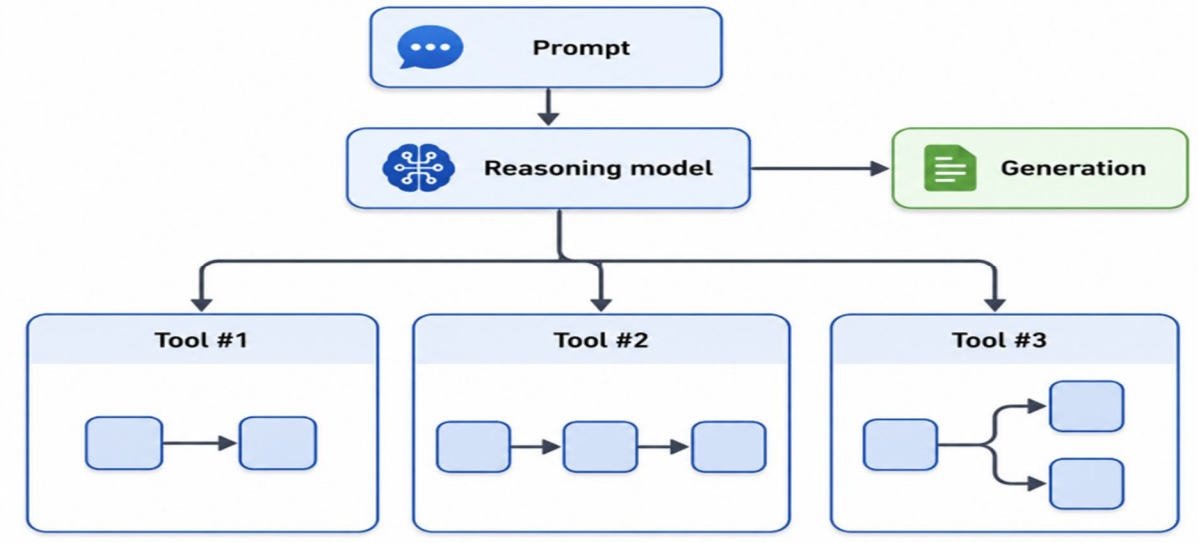


Figure 4 – A flowchart of Modular RAG Architecture
Source - Fezari and Al-Dahoud (2025)

IV. RAG APPLICATIONS

Retrieval models have helped LLM to achieve a remarkable leap in various applications of AI models. RAG can provide the models with the accurate, timely, and relevant information that will enhance their reliability, accuracy and transparency across various tasks, including writing, content moderation, and customer support. In this chapter some aspects of RAG applications which will help in improving the performance in real world – will be discussed.

A. Q&A Systems

Q&A system is an open domain (Madaan et al, 2023) based system. The traditional systems are mainly based on two structures: one is parametric knowledge that is explicitly encoded and the other is a language model (Huang and Chang, 2023). RAG provides a semi-automatic approach by storing the retrieved documents when an inference is made, and then generating grounded responses (Saad-Falcon, 2024) at that time.

1) Open-domain answering

This model should also be able to answer questions which are based on common knowledge about different topics. Firstly, the classical Dense Passage Retrieval (DPR) method is used to retrieve relevant dense passages in the text from Wikipedia or any other dense passages in the text, which will be used later for answering the question. The RAG-based solution has demonstrated impressive results in providing relevant and accurate answers, as well as tripling the number of documents retrieved from the RAG (Zhang 2023). Open domain systems' efficiency is determined by the quality of process related to retrieval (Jiang et al, 2025). A hybrid retrieval is used when there are documents that have been retrieved that are relevant to the query, to improve the recall.

2) Domain-centric Q&A

The niche areas such as finance, law or medicine would make RAG easily adaptable. Expert queries and trained on general corpora are not possible with the traditional models as the depth of the models is not sufficient. This is assisted by RAG which gathers data from legal documents, documents and scientific studies. In biomedical domain, for instance, the RAG models can be used to retrieve articles from PubMed, and collect the answers from the retrieved articles. Medical information stays on authoritative sources so as to decrease misinformation.

B. Chatbots

The need for precise details and understanding of the context is essential in other use cases, such as chatbots and conversational AI systems, and RAG has made a significant impact in these cases. RAG-based agents have no parametric memory or any type of script, instead they obtain facts from external resources when required to enhance the answer (Ye et al, 2020). An example of a virtual assistant enhanced by RAG would be when the user provides information, the virtual assistant would respond in an intelligent way to that information, in real time.Real-time (Zeng, 2023): RAG can be integrated with virtual assistants to give them intelligence, and intelligence enough to give the user information-based answers to his/her question. For example, AI legal assistant can search and match the appropriate contracts or case laws that match the customer's requirements. RAG can also be utilized to offer FAQs, policies, and former solutions to help customers via customer support chatbots. RAG is different to the traditional approach, which may lead to hallucinated/generic responses.The information from the most up-to-date databases is not obtained using the traditional approach as described by Jiang et al (2023), instead the RAG is used.

C. Students will create their own text, and summarizes text

The tasks that need to be summarized would benefit from being augmented with data collected from multiple sources (Gao et al, 2023). RAG can summarize multiple documents, create coherent summary and retrieve multiple relevant documents. Additionally, it is used to retrieve the final conclusion of pertinent research and to analyze and summarize legal documents as well as contracts, case law and regulations (Yang et al, 2023).

D. To create designs and design software programs

RAG can be very useful in software development and coding, as it can be used to collect relevant documents, code snippets, and discussions from platforms like Stack Overflow, which can then be used to guide AI-generated solutions. RAG models can be employed in programming to provide pertinent code snippets and records, mainly dependent on queries (Mallen et al, 2023). It can complete the code when code is partially entered, suggest to users the potential bugs based on previous bug reports (Xia et al, 2019).

If you use the RAG-based coder to retrieve API, and suggest the necessary libraries to use, some of the tasks can be fulfilled (Bajaj et al, 2016). Examples of this are that the system can use a multitude of solutions, download them from other repositories or from websites such as Stack Overflow, Python repositories or from a source provided by GitHub (Singh et al, 2025).

E. Biomedical

The application of RAG in this area is highly beneficial particularly in medical and drug discovery and decision making. Assists in making clinical decisions, collecting medical indications, medical records and scientific studies. Retrieval involves going through the medical journals and official health records, so that the information retrieved is credible in terms of the latest treatment of Type 2 type diabetes. Furthermore, RAG can be used in biomedical research; this is very valuable for the pharmaceutical industry for developing new drugs and repurposing already available drugs. In the case of a disease the RAG can be used to look for a relevant search query and then provide information to be added to the workflow.

F. Fact Checking

LLM systems tend to give wrong and biased information (Seo et al, 2024). By engaging in fact-checking over trusted sources, RAG is able to identify misinformation and fake news. It has been utilized by fact-checking firms to retrieve confirmed sources and to compare against over claims that are made on the web.

G. Analyzing Compliance

RAG is critical for the legal sector in various applications such as case law retrieval and analysis of contracts and compliance auditing. RAG is an approach used by Lawyers to look up precedent for similar cases, and to summarizes arguments. Moreover, enterprises can track the changes in regulation, and they can obtain the latest version of the regulation, analyzed by RAG.

In general, RAG can be used across various areas, such as QA, conversations, law and biomedical. Models can now get external knowledge, enhance adaptability, accuracy and transparency.

V. COMPARING NAÏVE, ADVANCED AND MODULAR RAG

There was a variation in the complexity of the models, adaptability and quality of the models. What could be the relationship between generator and retriever; what the relationship between these and learning process; how could this other knowledge be used in the generation and how could they be linked/untethered to the learning process. In this portion, three crucial types of RAG - Advanced RAG, Naive RAG and Modular RAG, are discussed. In Naïve RAG, the retriever is a static one, which is embedded in the RAG, and is based on an LLM (like OpenAI, Mistral or Google) that is not trained with any relevant data, followed by a sparse retrieval model (like BM25 or TF-IDF) (Gao et al, 2023). The first step of the pipeline is a user query and then it is fed to the pre-indexed corpus for relevant information (shown in Figure 5).

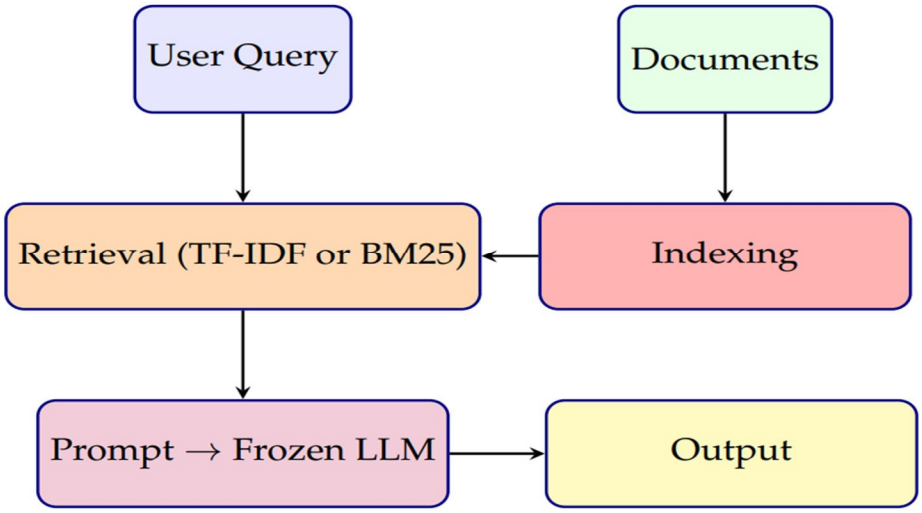


Figure 5 – A Simplified Architecture of Naïve RAG
Source – Neha et al (2025)

All the top-K passages returned are added to the question and top-K passages to create the prompt, and sent together to the frozen LLM. The model was used to generate a response to the system without giving the system any feedback (no optimization) which resulted in a sub-optimal and static system yielding, an improved retriever. It's a simple answer, but there are a number of issues. Often offers redundant/repetitive information, sometimes has difficulty building material between pieces of information in documents, verification responses based on information retrieved, but does not link information between documents. If you don't have advanced RAG, there are a number of considerations that need to be taken into account for naïve RAG. These are tackled in the pre-retrieval, post-retrieval and retrieval pipelines of advanced RAG (Gao et al, 2023) (Figure 6). During its operation, it does this: It generates a user's query, and indexes the documents (corpus). The pre-retrieval, query is changed to improve the quality of retrieval, with the help of rewriting, query routing and query expansion techniques. The retrieval stage involves a semantically encoded query that is matched to chunks of documents which are indexed with finely-tuned models and measures of similarity to the tasks. Further improvements on relevance using more advanced techniques: Prompt adaptive conditioning, Hybrid retrieval (dense, thin).

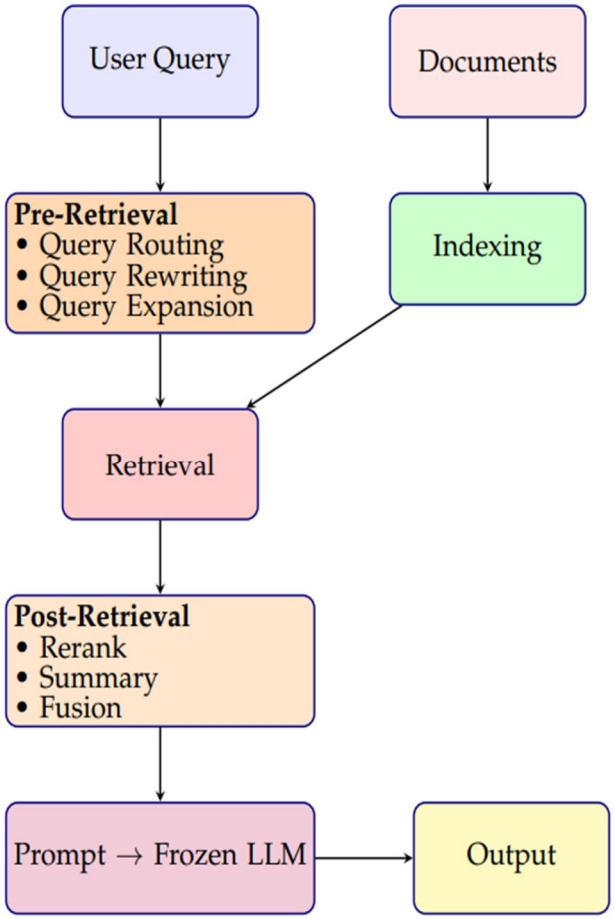


Figure 6 – An architecture of Advanced RAG
Source – Neha et al (2025)

The retrieval process is used to retrieve the chunks and then send the chunks to the post retrieval process which will re-rank, filter and compress the chunks. The following approaches will remove noisy/redundant output: summarizing/re-ranking will remove the output that is not very relevant, and only the most relevant context will be left to the frozen LLM to generate output from. If they don't have it, the naive models would be utterly devoid of any semantic organization, factuality and reliability of the architecture. The architecture is broken down into modules, which are then used to extend the possibilities of breaking down the architecture (Modular RAG (Gao et al, 2024). When the retrieval isn't trained with the generator, then it would be modular RAG. The dense dual encoder (that can be trained using a contrastive loss function) is called retriever. Maximum estimation of the probability can be used for optimizing per document generator (on top k documents). It is flexible, modular and has impacted the retrieval adaptation.

The various modules are connected to each other and are able to be reconfigured and/or selectively configured to the various tasks (as illustrated in Fig. 7) in the Modular RAG Architecture. These modules are: (1) Search to add retrieval functionality to other sources like knowledge graphs or Web results; (2) Memory to keep track of the previous context for consistency in multi-turn/dialogue applications; and (3) Routing to route queries to the right operations, like entity extraction or summarization. Several of them reformulate the query, remove noise from the results and then fuse the results in the rerank fusion, adapting the results to the context.

Modular RAG's extensibility and control as well as a set of novel approaches to domain knowledge and adaptation, enable scientific, clinical and enterprise level research. When using designs and loops for retrieval (as special systems) can be helpful, when: (with the following) write one example of when using designs and loops to retrieve are helpful. All three RAG modules have their pros and cons but none of them are suitable to be adapted to the clinical context. Naïve RAG is a simple and efficient approach, but is not robust to noise in retrieval and is not suitable for high stakes. Reranking and hybrid will be used to balance latency and factual bases to improve the performance in diagnostics and summarization in Advanced RAG. However, in some special circumstances, it is hard to deploy it, due to complexity.

The flexibility and transparency of integrating knowledge graphs/ontologies and fine-grained tailoring will be illustrated using flexibly and transparently applying modular RAG. The high computation cost and training cost are among the challenges of deployment of Modular RAG. The advanced RAG is expected to be well suited for piloting based on latest research, the modular RAG is expected to be very scalable for production, due to the uniformity of learning pipelines, and maturity of infrastructure.

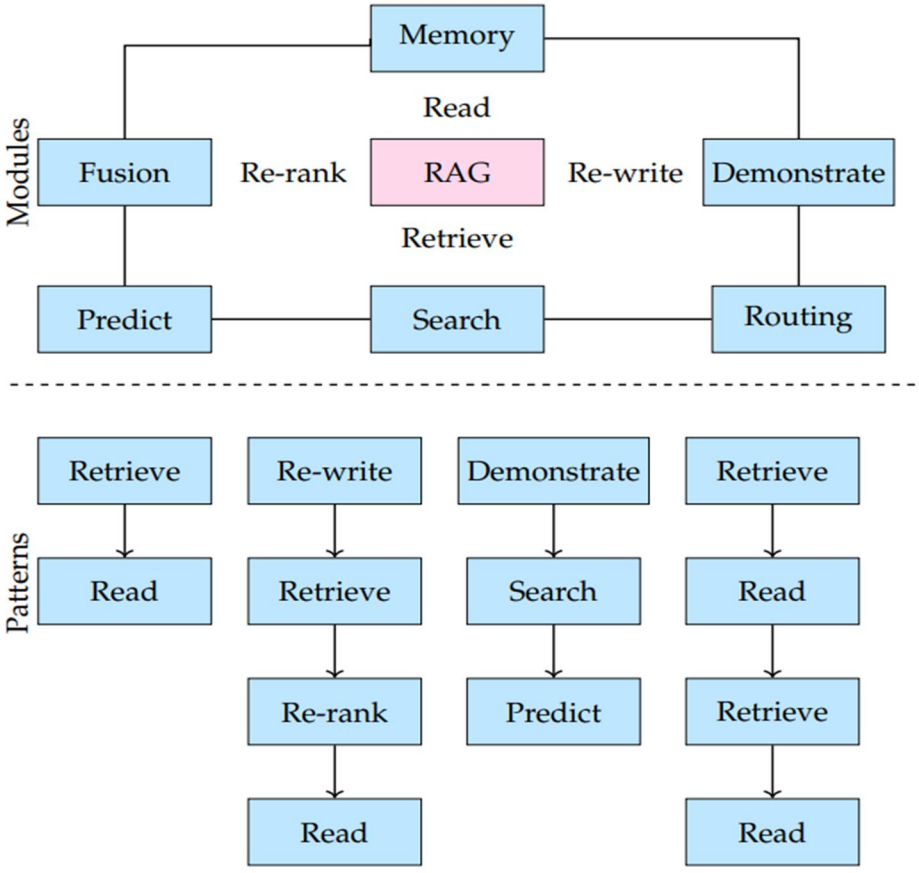


Figure 7 – Architecture of Modular RAG
Source – Neha et al (2025)

RAG can no longer be restricted to just a first pass pre-processing step as it was in days gone by; rather, RAG should be a smart and holistic aspect of reasoning. But Naïve RAG is feeble, but effective in “retrieve then read” paradigm. They are directly referred by poorly indexed documents or complicated/complex queries, appropriate indexing (and chunking) to get high quality inputs, reranking, hybrid search to leverage retrieval, filtering and compression to leverage the context. While it is important to make improvements, refine and develop in order to create a solid and correct pipeline, it is not a change in the sequence nature.

Table 1 – Comparison between Naïve, Advanced and Modular RAG

Characteristics	Naïve RAG	Advanced RAG	Modular RAG
Core concept	Sequential, simple generation and retrieval	Refining and optimizing core stages of the pipeline	Modular system for agency and orchestration
Key approaches	Core semantic search	Re-ranking, hybrid search, dynamic chunks, expanding queries, context filtering	Planning queries, reasoning in different steps, self-correction, iterative verification
Architecture	Follows Query → Retrieve → Generate pipeline	Improved sub-components with linear pipeline	Cyclical, non-linear graph of modules managed by planner or agent
Retrieval	Static, one-shot retrieval	2-stage, single-stage retrieval	Multi-turn, adaptive retrieval. It is up to the agent what to retrieve
Core strengths	Low latency, solid baseline and easy to implement	Reliability and accuracy are improved drastically over Naïve model	Manages multi-dimensional, complex queries. Adopts advanced tools.
Managing context	All recovered chunks are passed to LLMs in a naïve manner	Optimizes and processes context before it passes to the LLM	Synthesizes data from various modules and retrievals before final output
Weaknesses	Context overload, failing in complex challenges and irrelevant retrieval	Majorly reactive to first query with high computation cost	High latency, complexity, and development cost
Where to use?	Simple Q&A on well-planned masses	Needs high quality and accuracy of queries to apply production	Mission-critical, complex systems needing research, verification, and reasoning

Source – Fezari and Al-Dahoud (2025)

Agentic and Modular RAG seemed to be a revolution in making decisions and orchestration, though. It breaks it down the colossal pipeline into matrix of modules which are controlled by agentic controller: validation, synthesis and search. It offers loops, dynamics and conditionals which enable the system to think for itself and respond to a question instead of just recalling the answer. It's in the agency. This process is streamlined using AI-powered RAG, with modular picking the process from meta reasoning. Understands how to use a reactive tool, and creates a very powerful engine and adaptive agent/engine. A comparison of the different components of Naïve RAG, Advanced RAG and Modular RAG (Fezari and Al-Dahoud, 2025) is presented in Table 1 below:

Be aware if it is going to be a changeover towards either this module or another one or a changeover between these two – this will be determined at the moment, NOT in stages. Advanced RAG can be used as a “plus” to the existing system to enhance performance and predictability – while maintaining the security and integrity of the system. Although the landscape system is a complex system, the following challenges could be faced: Improving the capabilities of smartest AI assistants and agents (Fezari and Al-Dahoud, 2025), lack of query direction and navigation, and system landscape complexity.

Open RAG will look for information in typical data sources (Wikipedia, or the data on the web). Some brief descriptions of some of the architectures that we developed in this class. Several very basic architectures have been implemented such as Atlas, REALM and RETRO described by Izacard et al (2023) and Borgeaud et al (2022) respectively. There are a variety of ways of incorporating retrieval. The emphasis of REALM is on the pre-training. Chunk based retrieval is performed by RETRO which performs compression in local memory, and fine tuning by Atlas which is used to update the retriever. Neha et al (2025) have introduced open domain RAG which can generalize and explain most of the domains, but not able to provide factual information in health care domain.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IX Sep 2026- Available at www.ijraset.com

In the meantime, domain-oriented RAG has an influence on the retrieval to curated biomedical sources to improve the quality of the retrieved biomedical terminology, facts and trustworthiness. Any of these models can be BioRAG (Wang et al, 2024), MedRAG (Wang et al, 2024) or ClinicalGPT (Wang et al, 2024). Examples of leveraging of these systems are the use of domain-based knowledge, biomedical research and specific corpora. The medical language generation and retrieval system is developed specifically for medical tasks such as summarization of different parts of medical reports, assistance in diagnosis and decision making, interaction with doctors and/or patients to provide an answer to their queries. (Neha et al, 2025)

VI. DISCUSSION

There is no reference to a Technical RAG. This would be more of an objective than plan. The question shouldn't be whether or not an organization should be implementing RAG, it should be how a knowledge infrastructure would be sustainable for them in the years ahead until 2030 and beyond as they also have other problems to face such as the loss of institutional knowledge and regulatory requirement. Those that will be successful, will be the ones that are using RAG and more than the technique. They consider evaluation models and then start to write them, design an evaluation platform for them to develop and are involved in initial platform governance. The difference between leaders and the strugglers would be gigantic. Those early birds who embarked on the road to custom cycles using AI are off to a good start in the coming few weeks and others are building their own custom cycles (6-12 months). RAG will become 100% transformed until 2030. What will change, however, will be that the static top-K search will get dynamic as well as user adaptive orchestration and dynamically adapt search strategy based on user's requirement and complex queries will be added to the game. The knowledge structure that will be evolved from documents to multi-modal rich suggestions will be a hybrid of entity graph, embedding of vector and hierarchical indexes. For complex tasks, multi-step data collection strategies will be planned with the agents that can reflect on “quality of findings”, with agentic systems.

Compliance building blocks will continue in the system, new use cases will be added to address privacy concerns (whether or not they involve RAG will be determined), and perhaps how retrieval can further expand in the years to come if it can't be made illegal through RAG. Organizations need to be flexible, in order to cater to this trend. Other issues such as necessity of changes in shifts, necessity of changing components if there will be technology changes and necessity of changing level separation between the logic system and knowledge system (Fezari and Al-Dahoud, 2025) are also very important. All the competitive advantages will be transiting at the knowledge system by 2030. There will be models to choose from and will be presented to winners: correct models. Transition to governance models that have smart retrieval systems that can be safely deployed, scaled and are winners with institutional knowledge through smart retrieval systems as they implement them. Some organizations have been set up which have runtimes based on knowledge of RAG, such as Fezari and Al-Dahoud (2025).

A. Limitations

Though RAG framework is implemented on a large scale as evident from this study, there are some constraints of RAG framework.

1) System Noise

In data retrieval fields, data depiction during searching data has integral effect on data retrieval process, data loss is an integral part of data retrieval process. The system noise is the cause of failure points in RAG (Barnett et al, 2024) since it is in the form of misleading and irrelevant data. The accuracy of the retrievals can be further enhanced in the effective RAG, but recent studies demonstrated that the noisy retrievals (Cuconasu et al., 2024) can be generated to boost the quality of the generation in the effective RAG. There are a number of retrievals (Qiu et al., 2022) which can trigger the speedy building. Whether retrieval noise has an effect on or whether choice of metrics and communication of the retriever to the other person are in conflict in real-world settings has yet to be determined.

2) Retriever-Generator Gaps

Optimisation and design of every generator/retriever interaction might be required as the goals of the generators and retrievers do not necessarily match and the latent spaces of generators may be different from that of the retrievers. The strategies now being used today involve either disentangled generation and retrieval or having intermediate steps of using disentangled generation and retrieval. The former effect might be detrimental to the latter effect – combined training and generalizability might be affected. However in reality, it would be difficult to determine which of these solutions would be the most effective and would require discussion and consultation to determine that.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

3) Additional Overhead

This means that depending on the situation, part of this generation could be recovered which lower the generation cost, or if necessary, an additional overhead must be added in order to recover the generation (Izacard et al, 2023). The interaction and retrieval technique, however, does add latencies, as do the amount of socializing. This can also be further magnified when coupled with a complex improvement, like iterative RAG and recursive retrieval (Zhang et al, 2023). Another is that the larger the retrieval sources is the more complex will be retrieval and storage (Zhang et al, 2023). In RT services overhead is not a problem from the point of view of the practicability of RAG.

4) Complex System

Retrieval also inevitably entails tuning some hyper-parameters and the complexity of the system is increased. For instance, a recent work has demonstrated the effectiveness of the top-k in improving the attribution, in comparison with a single retrieval. It also impacts query-based fluency and more metrics need to be investigated (Aksito et al, 2023). As the level of expertise required to "tune" the service increases so can it rely on RAG.

B. Future Scope

There is however a great deal more research which could be carried out by RAGs:

1) Flexible Pipelines

Also, RAG pipelines should be flexible and adaptable as RAG which is recursive, iterative and adaptive. A set of unique versions of retrievers, sources, RAG subsystems and generators can be chosen to provide a mixture of smart engineering and tuning to optimize system performance to achieve complex tasks. Additional studies will continue to be conducted, such as new innovative RAG programs.

2) Augmentation Designing

There are a number of patterns which have been recognized, based on previous research on generator-retriever interaction. Some influence of the practical application on the end generation is present, but it's in different ways. Need to have more sophisticated frameworks to capture the benefits of RAG more advanced.

3) Efficient Processing

The deployment of a query-oriented RAG and LLM can be achieved in various ways, such as PipeRAG (Liu, 2022), LLAMA-Index (Jiang et al, 2024) etc. What is needed is to come up with a plug an' play solution for other tasks according to RAG generation and foundational. As the complexities and overhead of generators and retrievers grow, it is still difficult to realize smooth RAG, and optimization of the system-level (Hu et al, 2025) is needed.

4) Large-scale applications

Numerous options for use of R-A-G. But there are tasks that are generative and for RAG, it is essential that these tasks are explored. However, if there are any properties of the domain that are not considered to be required then RAG can be applied without referring to that property. RAGs should be designed to suit the domain, and be designed for long time uses.

5) Real-time

Long-tail knowledge RAG can be used to give long term and real time data. This has been previously explored based on knowledge expansion pipeline. While many past studies have concentrated on the information that can be delivered by a generator's training data, they have not considered the flexible and dynamic details that might be delivered by retrieval. This has resulted in a lot of research on RAG systems with flexible and continuously evolving information. RAG will also play a key role in the personalization/customization of information in today's Web services.

6) Use of Agentic RAG

The most crucial innovation of Agentic RAG is that it also introduces AI agents that have planning and decision-making abilities, tackling the drawbacks of traditional RAG (Krishnan, 2025). It's hard for conventional RAG to process information from multiple sources in a multi-step reasoning process, compared to the information flow in a traditional RAG, which flows from one step to the next and is static. Lack of skills to enhance the looks /noise of extra tools in context.

Agentic RAG is an agent that is a smart coordinator to create loop of reasoning, planning, acting and iterating (Wu et al, 2025). A few of the crucial ones are the dynamic tool, decomposition of tasks and iterative refinement of context. The advantages of this system are in relation to its dynamism, autonomy and strength modularity of the agents. These characteristics provide add a lot of improvements in reliability and precision of outputs (Fu et al, 2022). In addition, they can develop skills to deal with complex tasks by using and applying tool and dynamic planning towards proactive problem solving.

VII. CONCLUSION

RAG has been bred to be bigger, smarter and more realistic. RAG has solved the biggest problem with LLM reliability – that of pipelines. RAG can be used in conjunction with real-time streams of data and long-context windows, and this is more likely to yield an intelligence and active memory, capable of navigating large swathes of complexity and intellectualism with unexpected accuracy. From Naïve RAG to smart Multi-agent RAG, the RAG has been matured significantly by 2025. These approaches are more than just concepts of research. They play a key role in the production-level AI programmes. When creating a virtual assistant, chatbot or knowledge engine, it's crucial to select the right RAG framework to address the needs of the organization.

This study provides comprehensive knowledge about RAG framework when transforming AI models, with special emphasis on improvements and applications. This study has summarized and organized the basic changes in RAG, and provided insight into interaction among the generators and retrievers. This study offers possible practical tips on RAG for various tasks and modalities. Lastly, existing RAG benchmarks are also introduced with some limitations of RAG and highlighted with some promising areas of research.

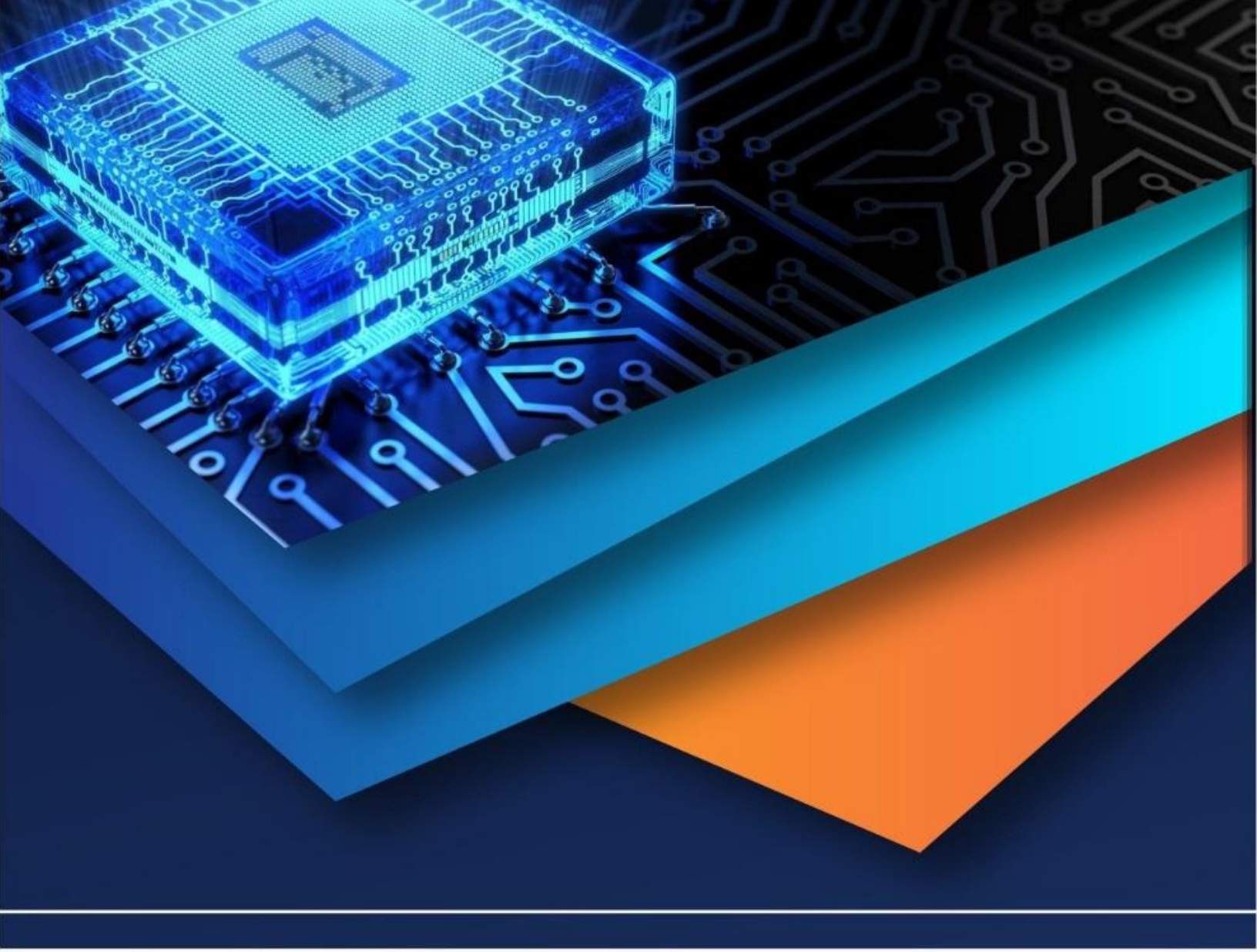
REFERENCES

[1] Husain, H., Wu, H. H., Gazit, T., Allamanis, M., & Brockschmidt, M. (2019). Codesearchnet challenge: Evaluating the state of semantic code search. arXiv preprint arXiv:1909.09436.
[2] Berant, J., Chou, A., Frostig, R., & Liang, P. (2013, October). Semantic parsing on freebase from question-answer pairs. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 1533-1544).
[3] Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., ... & Fung, P. (2023, November). A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In Proceedings of the 13th international joint conference on natural language processing and the 3rd conference of the asia-pacific chapter of the association for computational linguistics (volume 1: Long papers) (pp. 675-718).
[4] Wang, Y., Lipka, N., Rossi, R. A., Siu, A., Zhang, R., & Derr, T. (2024, March). Knowledge graph prompting for multi-document question answering. In Proceedings of the AAAI conference on artificial intelligence (Vol. 38, No. 17, pp. 19206-19214).
[5] Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., ... & Kaplan, J. (2022). Constitutional ai: Harmlessness from ai feedback. arXiv preprint arXiv:2212.08073.
[6] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., ... & Yih, W. T. (2020, November). Dense passage retrieval for open-domain question answering. In Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP) (pp. 6769-6781).
[7] Saha, S., Junaed, J. A., Saleki, M., Sharma, A. S., Rifat, M. R., Rahouti, M., ... & Amin, M. R. (2023, December). Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In Proceedings of the First Workshop on Bangla Language Processing (BLP-2023) (pp. 72-84).
[8] Levesque, H. J. (1984, August). A logic of implicit and explicit belief. In AAAI (Vol. 84, pp. 198-202).
[9] Saha, A., Pahuja, V., Khapra, M., Sankaranarayanan, K., & Chandar, S. (2018, April). Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).
[10] NVIDIA (2025). Spectrum-X: End-to-End Networking for AI and High-Performance Computing. Available at https://www.nvidia.com/en-us/networking/spectrumx/.
[11] Ma, L., Zhang, R., Han, Y., Yu, S., Wang, Z., Ning, Z., ... & Lu, C. T. (2023). A comprehensive survey on vector database: Storage and retrieval technique, challenge. arXiv preprint arXiv:2310.11703.
[12] Neha, F., Bhati, D., Shukla, D. K., Guercio, A., & Ward, B. (2025, January). Exploring AI text generation, retrieval-augmented generation, and detection technologies: A comprehensive overview. In 2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 00633-00639). IEEE.
[13] Han, B., Susnjak, T., & Mathrani, A. (2024). Automating systematic literature reviews with retrieval-augmented generation: A comprehensive overview. Applied Sciences, 14(19), 9103.
[14] Parekh, K. V., Saxena, N., & Ansari, M. A. (2025, June). A Comparative Study of Retrieval-Augmented Generation (RAG) Chatbots. In 2025 International Conference Automatics, Robotics and Artificial Intelligence (ICARAI) (pp. 1-6). IEEE.
[15] Singh, A., Ehtesham, A., Kumar, S., Khoei, T. T., & Vasilakos, A. V. (2025). Agentic retrieval-augmented generation: A survey on agentic rag. arXiv preprint arXiv:2501.09136.
[16] Ramdurai, B. (2025). Large language models (LLMs), retrieval-augmented generation (RAG) systems, and convolutional neural networks (CNNs) in application systems. International Journal of Marketing and Technology, 15(01), 2249-1058.
[17] FEZARI, M., & Al-Dahoud, A. (2025). The Evolution of Retrieval-Augmented Generation (RAG) in AI.
[18] Kalotra, S. (2025). Trends in Active Retrieval Augmented Generation. Signity. Available at https://www.signitysolutions.com/blog/trends-in-active-retrieval-augmented-generation

[19] Dogan, E. (2024). RAG, or Retrieval Augmented Generation: Revolutionizing AI in 2025. Glean. Available at https://www.glean.com/blog/rag-retrieval-augmented-generation
[20] Naresh (2025). Beyond Vanilla RAG: The 7 Modern RAG Architectures Every AI Engineer Must Know. Available at https://dev.to/naresh_007/beyond-vanilla-rag-the-7-modern-rag-architectures-every-ai-engineer-must-know-4l0c
[21] Karlsson, D. (2025). Evaluating modular RAG with reasoning models. Available at https://www.kapa.ai/blog/evaluating-modular-rag-with-reasoning-models
[22] Naminas, K. (2025). RAG Evaluation: Metrics and Benchmarks for Enterprise AI Systems. Label Your Data. Available at https://labelyourdata.com/articles/llm-fine-tuning/rag-evaluation
[23] Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., ... & Clark, P. (2023). Self-refine: Iterative refinement with self-feedback. Advances in neural information processing systems, 36, 46534-46594.
[24] Huang, J., & Chang, K. C. C. (2023, July). Towards reasoning in large language models: A survey. In Findings of the association for computational linguistics: ACL 2023 (pp. 1049-1065).
[25] Saad-Falcon, J., Khattab, O., Potts, C., & Zaharia, M. (2024, June). Ares: An automated evaluation framework for retrieval-augmented generation systems. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers) (pp. 338-354).
[26] Zhang, J. (2023). Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt. arXiv preprint arXiv:2304.11116.
[27] Jiang, X., Zhang, R., Xu, Y., Qiu, R., Fang, Y., Wang, Z., ... & Wang, Y. (2025, July). HyKGE: A hypothesis knowledge graph enhanced RAG framework for accurate and reliable medical LLMs responses. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 11836-11856).
[28] Ye, D., Lin, Y., Du, J., Liu, Z., Li, P., Sun, M., & Liu, Z. (2020, November). Coreferential reasoning learning for language representation. In Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP) (pp. 7170-7186).
[29] Zeng, H. (2023). Measuring massive multitask chinese understanding. arXiv preprint arXiv:2304.12986.
[30] Jiang, H., Wu, Q., Lin, C. Y., Yang, Y., & Qiu, L. (2023, December). Llmlingua: Compressing prompts for accelerated inference of large language models. In Proceedings of the 2023 conference on empirical methods in natural language processing (pp. 13358-13376).
[31] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., ... & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.
[32] Yang, A., Nagrani, A., Seo, P. H., Miech, A., Pont-Tuset, J., Laptev, I., ... & Schmid, C. (2023). Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10714-10726).
[33] Mallen, A., Asai, A., Zhong, V., Das, R., Khashabi, D., & Hajishirzi, H. (2023, July). When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers) (pp. 9802-9822).
[34] Xia, M., Huang, G., Liu, L., & Shi, S. (2019, July). Graph based translation memory for neural machine translation. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 7297-7304).
[35] Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., ... & Wang, T. (2016). MS MARCO: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268.
[36] Seo, M., Baek, J., Thorne, J., & Hwang, S. J. (2024). Retrieval-augmented data augmentation for low-resource domain tasks. arXiv preprint arXiv:2402.13482.
[37] Gao, Y., Xiong, Y., Wang, M., & Wang, H. (2024). Modular rag: Transforming rag systems into lego-like reconfigurable frameworks. arXiv preprint arXiv:2407.21059.
[38] Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., ... & Grave, E. (2023). Atlas: Few-shot learning with retrieval augmented language models. Journal of Machine Learning Research, 24(251), 1-43.
[39] Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., ... & Sifre, L. (2022, June). Improving language models by retrieving from trillions of tokens. In International conference on machine learning (pp. 2206-2240). PMLR.
[40] Wang, C., Long, Q., Xiao, M., Cai, X., Wu, C., Meng, Z., ... & Zhou, Y. (2024). Biorag: A rag-llm framework for biological question reasoning. arXiv preprint arXiv:2408.01107.
[41] KHAPRA, C. (2024). A survey on end-to-end speech recognition systems. INTERNATIONAL JOURNAL OF COMMUNICATION, 5(2), 122-127.
[42] Shukla, S., Mishra, S., Singh, J., Mishra, K., Khapra, C., & Morey, P. (2024, August). Deep Belief Networks for Unsupervised Feature Learning and Improved Image Recognition. In International Conference on Artificial Intelligence on Textile and Apparel (pp. 133-145). Singapore: Springer Nature Singapore.
[43] Khapra, P. (2022). Evolution of cybercrime and deepfakes–Exploring intervention strategies of international organizations against AI threats. NeuroQuantology, 20(22), 1425-1434.
[44] Barnett, S., Kurniawan, S., Thudumu, S., Brannelly, Z., & Abdelrazek, M. (2024, April). Seven failure points when engineering a retrieval augmented generation system. In Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI (pp. 194-199).
[45] Cuconasu, F., Trappolini, G., Siciliano, F., Filice, S., Campagnano, C., Maarek, Y., ... & Silvestri, F. (2024, July). The power of noise: Redefining retrieval for rag systems. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 719-729).
[46] Qiu, L., Shaw, P., Pasupat, P., Shi, T., Herzig, J., Pitler, E., ... & Toutanova, K. (2022, December). Evaluating the impact of model scale for compositional generalization in semantic parsing. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (pp. 9157-9179).
[47] Zhang, H., Zhao, P., Miao, X., Shao, Y., Liu, Z., Yang, T., & Cui, B. (2023). Experimental analysis of large-scale learnable vector storage compression. arXiv preprint arXiv:2311.15578.
[48] Aksitov, R., Chang, C. C., Reitter, D., Shakeri, S., & Sung, Y. (2023). Characterizing attribution and fluency tradeoffs for retrieval-augmented large language models. arXiv preprint arXiv:2302.05578.
[49] Liu, J. (2022). LlamaIndex [Computer software]. https://doi.org/10.5281/zenodo.1234
[50] Jiang, W., Zhang, S., Han, B., Wang, J., Wang, B., & Kraska, T. (2024). Piperag: Fast retrieval-augmented generation via algorithm-system co-design. arXiv preprint arXiv:2403.05676.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[51] Hu, Z., Murthy, V., Pan, Z., Li, W., Fang, X., Ding, Y., & Wang, Y. (2025, October). Hedrarag: Co-optimizing generation and retrieval for heterogeneous rag workflows. In Proceedings of the ACM SIGOPS 31st symposium on operating systems principles (pp. 623-638).
[52] Krishnan, N. (2025). Ai agents: Evolution, architecture, and real-world applications. arXiv preprint arXiv:2503.12687.
[53] Wu, P., Zhang, M., Wan, K., Zhao, W., He, K., Du, X., & Chen, Z. (2025). Hiprag: hierarchical process rewards for efficient agentic retrieval augmented generation. arXiv preprint arXiv:2510.07794.
[54] Fu, Y., Peng, H., Sabharwal, A., Clark, P., & Khot, T. (2022). Complexity-based prompting for multi-step reasoning. arXiv preprint arXiv:2210.00720.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)