iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** VIII **Month of publication:** August 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.84752

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 VIII Aug 2026- Available at www.ijraset.com*

# Large Language Models Hallucinate and How Retrieval- Augmented Generation Mitigates It

Shyalaja L N[1], Shantinath Patil[2], Prithviraj S R[3], Shreyas Bharadwaj B M[4], Harshitha P[5]
[1]*Associate Professor,* [2, 3, 4, 5]*Student, 7th Sem,*
*Department of Artificial Intelligence and Machine Learning, Bahubali College of Engineering, Shravanabelagola, India*

***Abstract:*** ***Large Language Models (LLMs) can generate fluent and convincing responses, but fluency does not guarantee factual correctness. Hallucination occurs when a model produces information that is false, unsupported, or inconsistent with available evidence. This paper reviews why hallucinations arise andexamine Retrieval-Augmented Generation (RAG) as a practical mitigation strategy. The discussion focuses on prediction-based generation, missing or outdated knowledge, ambiguous queries, and the absence of automatic verification. RAG addresses these limitations by retrieving relevant information from an external knowledge source and supplying it to the generator as contextual evidence. The paper describes the major stages of a RAG pipeline, including document ingestion, chunking, embeddings, vector storage, retrieval, context augmentation, and answer generation. It also examines the conditions under which RAG can fail, including poor retrieval, incomplete knowledge bases, unreliable sources, weak ranking, and incorrect interpretation of retrieved context. Finally, the paper discusses evaluation dimensions such as context relevance, answer faithfulness, and answer relevance, and outlines applications of RAG in education, enterprise search, technical support, research assistance, and document question answering. The analysis concludes that RAG should be viewed as a grounding and evidence-access mechanism rather than a guarantee of hallucination-free generation.***

***Keywords: Large Language Models, Hallucination, Retrieval-Augmented Generation, RAG, Embeddings, Vector Database, Information Retrieval, Factuality.***

## I. INTRODUCTION

Generative artificial intelligence has made large language models a central component of modern natural language processing. LLMs can answer questions, summarize documents, generate code, translate text, and explain technical concepts. However, the same generative mechanism that enables flexible language production can also produce statements that are plausible in form but incorrect in substance. The literature commonly describes this phenomenon as hallucination, and surveys show that hallucination remains an important reliability problem across multiple natural language generation tasks [2].

A key reason is that an LLM is fundamentally a probabilistic sequence generator. It learns statistical and semantic patterns from training data and generates likely continuations of an input. A likely continuation is not necessarily a verified fact. The problem becomes more significant when a question requires information that is absent from the model's learned knowledge, is ambiguous, or has changed after training. In such cases, a model may still produce a confident response rather than explicitly stating that evidence is unavailable.

Retrieval-Augmented Generation was introduced as a way to combine parametric model knowledge with non-parametric external memory. The foundational RAG work by Lewis et al. demonstrated a framework in which a retriever obtains relevant passages from a dense index and the generator conditions its response on those passages [1]. Later surveys have expanded the RAG paradigm into retrieval, augmentation, and generation components and have highlighted its usefulness for dynamic and domain-specific knowledge [3].

This paper presents a focused review of the relationship between hallucination and RAG. The objective is not to claim that RAG eliminates hallucination, but to explain the mechanism by which evidence retrieval can reduce unsupported generation and to identify the failure modes that remain.

## II. UNDERSTANDING LLM HALLUCINATIONS

An LLM hallucination is a generated statement that appears coherent or confident but is false, fabricated, unsupported, or inconsistent with the available evidence. Hallucinations are broader than simple spelling or formatting errors: a model may present an incorrect date, person, citation, statistic, explanation, or event with high linguistic confidence. The hallucination literature distinguishes different forms and task-specific manifestations, including factual inconsistency and unsupported content [2].

### A. *Major Causes of Hallucination*

1) Prediction-based generation: LLMs generate tokens according to learned probability distributions. The most probable continuation can be linguistically appropriate without being factually correct.
2) Missing information: When the required evidence is not available in the model's learned knowledge, the model may attempt to construct an answer instead of refusing the question.
3) Outdated knowledge: A model's parametric knowledge can become stale as facts, policies, products, regulations, and events change. This creates a mismatch between learned knowledge and current reality.
4) Ambiguous queries: Questions with unclear entities, dates, terminology, or user intent can cause the model to select an incorrect interpretation and generate a response consistent with that interpretation.
5) Lack of automatic verification: Standard generation does not inherently require every claim to be checked against an authoritative source before the response is returned.

## III. RETRIEVAL-AUGMENTED GENERATION

RAG is a framework that retrieves relevant information from an external knowledge source and provides that information to an LLM before generation. The central principle is simple: retrieve evidence first, then generate an answer conditioned on that evidence. In the original formulation, RAG combines a pre-trained sequence-to-sequence generator with a dense vector index accessed by a neural retriever [1]. Modern RAG systems generalize this idea to enterprise documents, websites, databases, manuals, and other knowledge repositories [3].

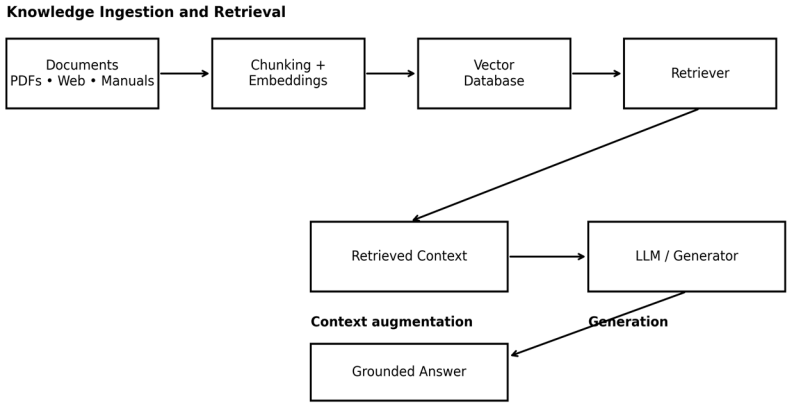



Fig. 1. General Retrieval-Augmented Generation Architecture.

### A. *RAG Pipeline*

A typical RAG pipeline contains five logical stages. First, documents are collected and transformed into manageable chunks. Second, the chunks are converted into embeddings that represent their semantic content. Third, the embeddings are stored in a vector index or vector database. When a user submits a query, the retriever searches for semantically relevant passages. Finally, the selected context is inserted into the generation prompt and the LLM produces a response grounded in the retrieved information.

This architecture separates knowledge access from language generation. The retriever is responsible for finding evidence, while the generator is responsible for interpreting the evidence and producing a natural-language response. This separation is one reason RAG can incorporate new or private knowledge without retraining the entire language model [1], [3].

## IV. EMBEDDINGS, VECTOR DATABASES, AND RETRIEVAL

Embeddings map text into numerical vector representations so that semantically related pieces of text can be compared in a continuous vector space. For example, a user query asking how to submit leave may be matched with a document titled 'Procedure for submitting a leave application' even when the exact words differ. The vector database stores these representations and supports similarity search. Depending on the system design, retrieval can use dense vector similarity, sparse lexical methods, hybrid retrieval, metadata filtering, or a combination of these techniques. The retrieved passages are then ranked or filtered so that the generation model receives context that is both relevant and sufficiently focused.

Retrieval quality is critical because RAG can only ground an answer in the evidence it successfully retrieves. A system that retrieves irrelevant, incomplete, or contradictory passages can still produce an incorrect response. Therefore, improving retrieval and ranking is not merely an infrastructure concern; it directly affects factual reliability.

Table I. Major Rag Components

| Component | Primary Function | Reliability Role |
|---|---|---|
| Documents | Provide source knowledge | Defines evidence quality |
| Chunking | Split documents into retrievable units | Affects context completeness |
| Embeddings | Represent semantic meaning | Affects semantic matching |
| Vector Store | Index and search representations | Affects retrieval efficiency |
| Retriever / Ranker | Select relevant passages | Controls evidence quality |
| LLM Generator | Interpret context and answer | Controls faithfulness and fluency |

## V. HOW RAG MITIGATES HALLUCINATION

RAG mitigates hallucination primarily through grounding. Instead of relying only on information encoded in model parameters, the generator receives explicit passages retrieved at answer time. If the retrieved source is authoritative and relevant, the model has stronger evidence from which to formulate the response.

RAG also provides a mechanism for external and domain-specific knowledge. A college policy document, company handbook, product manual, or technical knowledge base can be indexed and queried without modifying the underlying LLM. This is particularly useful for private information that would not normally exist in the model's public training corpus.

Another benefit is traceability. A RAG application can retain the retrieved passages and expose citations or source references alongside the generated answer. This does not automatically prove that the answer is correct, but it gives users a practical path for verification and makes unsupported claims easier to investigate.

TABLE II. HALLUCINATION SOURCES AND RAG MITIGATION

| Hallucination Source | RAG Response | Remaining Risk |
|---|---|---|
| Missing knowledge | Retrieve external evidence | Evidence may not exist |
| Outdated model knowledge | Use updated source | Source may also be stale |
| Private/domain data | Connect a controlled knowledge base | Access and indexing errors |
| Unsupported claims | Condition generation on context | Model may misuse context |
| Poor source traceability | Expose retrieved citations | Citation does not guarantee correctness |

## VI. LIMITATIONS AND FAILURE MODES OF RAG

RAG is a mitigation technique, not a guarantee of zero hallucinations. The quality of the final answer depends on the complete retrieval-and-generation chain. A useful way to express the dependency is: poor retrieval leads to poor context, which can lead to a poor answer.

Poor retrieval occurs when the retriever fails to identify the passage that contains the answer. This may happen because of weak embeddings, poor chunking, lexical mismatch, ambiguous queries, inadequate indexing, or an inappropriate retrieval depth.

Missing information is another failure mode. If the knowledge base does not contain the required fact, retrieving additional passages cannot solve the problem. The system should ideally recognise the evidence gap and communicate uncertainty rather than inventing an answer. Source quality also matters. RAG can faithfully retrieve an incorrect document, and an LLM can then generate an answer that is well grounded in a bad source. Conflicting documents create an additional challenge because the generator must determine which evidence should be trusted.

Finally, the LLM may misinterpret or misuse retrieved context. Therefore, retrieval alone is insufficient; prompt design, context selection, reranking, source quality, answer verification, and appropriate refusal behaviour are all relevant to reliable RAG systems

## VII. NORMAL LLM VERSUS RAG-BASED LLM

A conventional LLM primarily relies on knowledge encoded in its parameters. A RAG-based system combines that parametric knowledge with information retrieved from an external source at inference time. The distinction is important because RAG changes the knowledge-access mechanism rather than simply increasing the model size.

TABLE III. COMPARISON OF NORMAL LLM AND RAG-BASED LLM

| Feature | Normal LLM | RAG-Based LLM |
|---|---|---|
| Knowledge | Mostly parametric | Parametric + retrieved |
| Recent information | May be limited | Can use updated sources |
| Private documents | Not inherently available | Can query connected sources |
| Traceability | Often limited | Can expose retrieved evidence |
| Hallucination | Can occur | Can be reduced, not eliminated |

## VIII. EVALUATION OF RAG SYSTEMS

A reliable RAG system should be evaluated as a pipeline rather than only by inspecting final answers. Recent evaluation frameworks emphasize multiple dimensions. RAGAS proposes reference-free evaluation for retrieval-augmented systems and focuses on dimensions such as the relevance of retrieved context, the faithfulness of generated answers to that context, and the relevance of the final answer [4]. ARES similarly evaluates RAG systems using context relevance, answer faithfulness, and answer relevance, and demonstrates how automated evaluation can reduce the manual effort required to assess large numbers of generated responses [5]. These approaches highlight an important research principle: an apparently good answer can hide a poor retriever, and a good retriever can still be followed by an unfaithful generator. For practical evaluation, researchers should construct representative queries, inspect retrieval recall and ranking quality, measure faithfulness and answer relevance, and test adversarial or out-of-domain questions. Evaluation should also include cases where the correct answer is absent, because a robust system should distinguish 'not found' from 'found and answered.'

## IX. APPLICATIONS

RAG is applicable wherever users need natural-language access to a changing or specialized document collection. Educational institutions can use RAG for college policy assistants, course-material question answering, and student support. Enterprises can build assistants over internal documentation, standard operating procedures, and product knowledge bases. Technical support systems can retrieve manuals and troubleshooting instructions before generating responses.

Research assistants can retrieve papers and evidence relevant to a question, while legal and compliance systems can use controlled document collections to support document question answering. In all of these settings, the principal value of RAG is not merely conversational fluency; it is the ability to connect generation with an explicit evidence source.

## X. RESEARCH CHALLENGES AND FUTURE DIRECTIONS

Current RAG research is moving beyond basic retrieve-and-generate pipelines toward better query transformation, hybrid retrieval, reranking, adaptive retrieval, multimodal retrieval, and more rigorous evaluation [3]. Future systems are likely to place greater emphasis on evidence selection and verification rather than treating retrieval as a single similarity-search step.

A major research challenge is distinguishing retrieval failure from generation failure. When an answer is wrong, the system should determine whether the necessary evidence was never retrieved, whether the retrieved evidence was insufficient, or whether the generator failed to use correct evidence. This decomposition can support targeted improvements and more transparent reliability analysis.

Another direction is calibrated uncertainty. A high-quality RAG system should not only produce grounded answers when evidence exists, but should also abstain or request clarification when evidence is weak, conflicting, or absent. Such behavior can make RAG systems more dependable in high-stakes or knowledge-intensive applications.

## IX. CONCLUSION

LLM hallucination is a fundamental reliability challenge caused in part by the difference between fluent text generation and verified factual reasoning. An LLM can generate a convincing answer even when the required evidence is missing, outdated, ambiguous, or incorrectly represented in its learned knowledge.

Retrieval-Augmented Generation mitigates this problem by connecting the generator to an external knowledge source. Through document ingestion, chunking, embeddings, retrieval, context augmentation, and generation, RAG gives the LLM explicit evidence that can improve factual grounding and domain relevance [1], [3]. However, RAG does not eliminate hallucinations. Poor retrieval, incomplete or incorrect sources, weak ranking, and misinterpretation of context can still produce unreliable answers.

The most accurate conclusion is therefore that RAG is an evidence-access and grounding mechanism, not a truth guarantee. Its effectiveness depends on the quality of the knowledge source, retrieval pipeline, context selection, generation behavior, and evaluation methodology. Combining strong retrieval with explicit source attribution, faithfulness evaluation, uncertainty handling, and continuous testing provides a more credible path toward reliable GenAI systems.

## X. ACKNOWLEDGMENT

The author acknowledges the guidance and technical training that supported the preparation of this paper. The paper is prepared as a literature-based academic review and does not claim original experimental results.

## REFERENCES

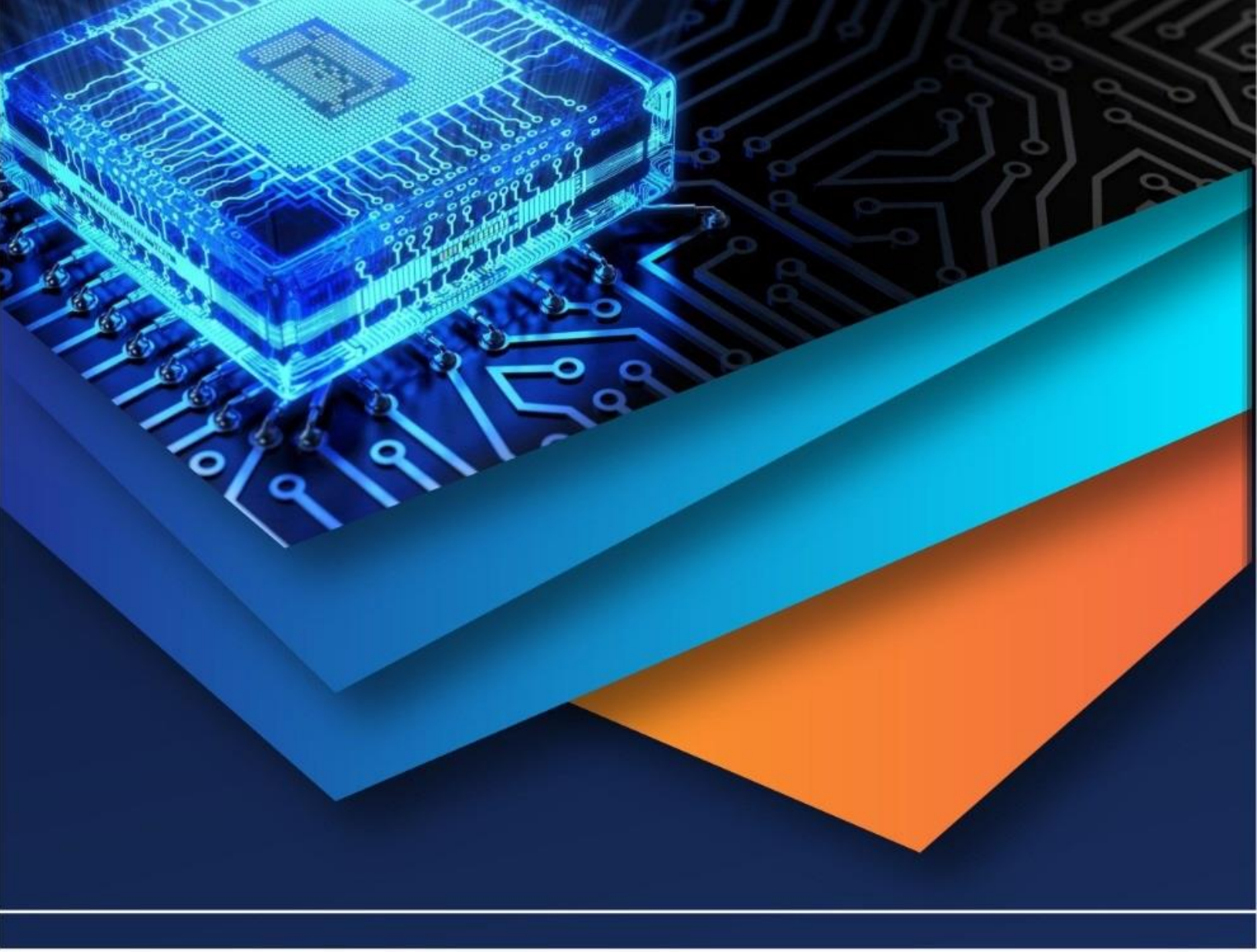
[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in Advances in Neural Information Processing Systems, vol. 33, 2020.

[2] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, “Survey of Hallucination in Natural Language Generation,” ACM Computing Surveys, vol. 55, no. 12, Art. no. 248, 2023, doi: 10.1145/3571730.

[3] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, “Retrieval-Augmented Generation for Large Language Models: A Survey,” arXiv preprint arXiv:2312.10997, 2023.

[4] S. Es, J. James, L. Espinosa Anke, and S. Schockaert, “RAGAs: Automated Evaluation of Retrieval Augmented Generation,” in Proc. 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, 2024.

[5] sJ. Saad-Falcon, O. Khattab, C. Potts, and M. Zaharia, “ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems,” arXiv preprint arXiv:2311.09476, 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY