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 10, June 2026
Impact Factor: 8.2

A Comparative Study of Retrieval-Augmented Generation and Standard Large Language Models for Factual Question Answering

Nutan Magar[1] and Aleena Thomas[2]
MCA Department[1,2]
Navinchandra Mehta Institute of Technology and Development, Dadar, India
magannutan@gmail.com[1], aleenat582@gmail.com[2]

Abstract: *It is well known that LLMs tend to hallucinate and generate plausible yet absolutely fictitious data when they try to respond to certain queries based on the given document. This study seeks to solve this problem. We tested two setups: one where we had a standalone LLM that worked exclusively on its own internal knowledge base and another setup where we used RAG system. For both setups we chose the Mistral-7B-Instruct-v0.1 model as the base LLM. There was only one difference in these two approaches – presence or absence of the retrieval part.*

To implement the RAG approach, we chose the following tools: BAAI/bge-small-en embeddings, FAISS vector database, and LangChain. We compared both models using 25 fact-based questions extracted from specialized PDF documents. We used five metrics: Exact Match, F1 Score, Accuracy, Hallucination Rate, and Latency.

The results speak for themselves: the RAG model did not deviate from the factual data in any way, while standalone LLM failed to correctly handle queries based on the document content. Thus, one can say that adding a retrieval part is not merely helpful for an LLM but necessary.

Keywords: Retrieval-Augmented Generation, Large Language Models, Mistral-7B, FAISS, Hallucination, Factual QA, NLP Evaluation, LangChain, Sentence Transformers

I. INTRODUCTION

Nowadays, large language models have become so advanced in producing convincing responses that it becomes difficult to tell whether their answers are true or not. If you pose any question to a modern-day language model, it will give you a smooth and confident response. However, sounding right and being right are two different things, particularly if the model does not have information about the specific documents it was asked about. [1]. We came across this issue during our experimentation phase. While testing the Mistral-7B model, we found out that if the model is asked questions regarding the contents of our uploaded PDFs, it would respond based on its general knowledge, not the document itself. Some of the time, it was spot-on; some of the time, it wasn't; and other times, it would be totally off. This is actually well-known in natural language processing (NLP) studies and referred to as "hallucination." [2].

As a result of that, RAG became an effective way of solving this problem. Since we provide the model with answers from the document, it means that we help the model not to rely on everything that it has learned throughout the entire training period. When the correct answer is available to the model, then it becomes extremely difficult for the model to fabricate anything [3].

There were several examples in the past when researchers have proven the effectiveness of RAG [3,4], which made us sure that we do not need to think of any disadvantages. What we should pay attention to was the possibility of using RAG in our case.

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 10, June 2026
Impact Factor: 8.2

The following experiment was based on a comparative test under controlled conditions. In other words, everything but retrieval should remain unchanged. Here are all the details:

II. LITERATURE REVIEW

2.1 Birth of Transformers

It has become possible to make the most recent AI revolution due to the attention mechanism developed by Vaswani et al. [5] that allows handling all sequences of texts simultaneously without the usage of any sequential calculations. Therefore, gigantic AI models, including GPT and LLaMA, have appeared. Such models may be trained to learn absolutely everything known to humanity. Our research features the example of an amazing development called Mistral-7B based on the Sliding Window Attention concept.

2.2 How to Fight Hallucinations

As mentioned in the work of Ji et al. [2], hallucinations may be divided into intrinsic (that contradict data) and extrinsic (that include unverifiable facts). One needs to bear in mind that, while thinking about the use of the colossal model as the solution, the problem with hallucinations will not disappear because, regardless of size, a model will never get any data that has not been included in the training set. Thus, RAG relies on the extension of model memory.

2.3 Evolution of RAG

Lewis et al. [3] have demonstrated that a system comprising of a dense retriever and a generator always performs better than a closed-book system. Also, the use of dense embeddings such as BAAI/bge model we selected has proven effective in understanding the context of a query and thus performing better than keyword-based search [6]. Although there exist sophisticated RAG systems with advanced re-ranking techniques, our research was focused on naive RAG.

2.4 Contribution of This Study

In almost all existing papers on RAG, researchers use common benchmarks such as SQuAD or TriviaQA. These data sets are good for evaluation purposes; however, the model can have come across them at least in some form during the process of training. The models evaluated in this paper are sure to not have encountered any part of the data set used, meaning the results will reflect pure retrieval and pure ignorance on the part of the LLM.

III. RESEARCH METHODOLOGY

3.1 Research Design

This research has adopted an experimental design where the retrieval-augmented generation is evaluated through experimentation using a single large language model. Using the same base model, namely Mistral-7B-Instruct-v0.1, will assist in ensuring that only the effect of retrieval-augmented generation is being analyzed.

3.2 Overview of System Architecture

Two systems have been developed for this study, which are:

3.2.1 Retrieval-Augmented Generation System:

In this system, the mistral-7B-Instruct-v0.1 model is employed alongside the retrieval pipeline, which uses the bge-small-en embeddings, faiss vector database, and langchain retriever. The documents chunks most relevant to the question are retrieved and served as input.

3.2.2 Large Language Model (LLM):

In this system, the mistral-7B-Instruct-v0.1 model is employed alone without any retrieval process.

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 10, June 2026
Impact Factor: 8.2

3.3 External Knowledge Base Building

The external knowledge base includes four PDF files that have information regarding historical events, scientific topics, endangered animals, and solar system. These files were uploaded and preprocessed in a Google Colab environment. Text data was extracted from these PDF files using the LangChain's PyPDFLoader module and then divided into smaller chunks. Each chunk of text was then embedded using the BAAI/bge-small-en model and saved in a FAISS vector database.

3.4 Evaluation Criteria

The efficiency of both the systems was tested based on the following five criteria:

3.4.1 Exact Match (EM)

Checks if there is an exact match between the predicted response and the correct answer.

3.4.2 F1 Score

Calculates the similarity between the predicted response and the correct answer based on their token-level similarities. This similarity score is calculated by taking the harmonic mean of precision and recall.

3.4.3 Accuracy

Checks if the correct answer is present in the predicted answer.

3.4.4 Hallucination Rate

Checks if there is any hallucination in the generated answer. An answer is said to be hallucinated when it does not share any lexical similarity with the retrieved context.

3.4.5 Response Time

Checks the generation time of the response.

3.5 Experiment Methodology

All queries from the dataset were treated independently by each of the two models. In the case of RAG model, the related document passages were extracted and used for generating a response while in the case of LLM model, responses were generated without any context. Responses obtained from both methods were evaluated using a set of metrics that were predetermined as the standard answer set.

IV. IMPLEMENTATION

4.1 Environment and Dependencies

All experiments were performed on the Google Colab platform used for accelerating computations. The system utilized LangChain. Sentence-Transformers, FAISS, PyPDF, NLTK, Matplotlib libraries.

4.2 RAG Pipeline

4.2.1 Document Loading

Four PDF files were imported into Google Colab. Text extraction from each file was done using LangChain's PyPDFLoader, which extracts text page-by-page and keeps metadata for possible referencing purposes.

4.2.2 Text Chunks Creation

The extracted text was divided into chunks using RecursiveCharacterTextSplitter, with 400 tokens per chunk and a 50-token overlap. The overlap ensures that the context of the information is not lost between chunks.

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 10, June 2026
Impact Factor: 8.2

4.2.3 Embeddings Creation

The text chunks were embedded using BAAI/bge-small-en embedding model, which generates 384-dimensional vectors. The model is well suited for semantic tasks due to its small size and efficiency, which made it suitable for running on the limited resources of the Google Colab cloud environment.

4.2.4 Building and Searching the Vector Index

The embeddings were indexed with the help of the FAISS library. At query time, the query itself was encoded with the same model, and the corresponding chunks were returned based on similarity.

4.3 Baseline System Pipeline

The baseline pipeline employed the same Mistral-7B-Instruct-v0.1 model without retrieval augmentation. The prompt template was considerably simplified. There was no additional context at all. The generation parameters were kept constant for both pipelines.

4.4 Evaluation Pipeline

Every query was launched simultaneously on both pipelines under the same conditions. The exact match score, F1 score, accuracy, hallucination rate, and response time were evaluated for each query. The metrics were aggregated into a Pandas DataFrame, and the summary statistics were visualized with Matplotlib.

V. RESULTS AND ANALYSIS

The below table presents the mean scores across all questions for both systems.

Experimental results

Metric	Standard LLM (Mistral-7B)	RAG System (Mistral-7B + FAISS)
Exact Match (EM)	0.000	0.136
F1 Score	0.106	0.462
Accuracy	0.360 (36%)	0.540 (54%)
Hallucination Rate	1.000 (100%)	0.680 (68%)
Avg. Response Time	9.87 seconds	6.44 seconds

Exact Match (EM):

The LLM achieved zero exact matches across all 25 questions, while the RAG system produced exact matches on approximately 3 out of every 25 questions. EM is an extremely strict metric the predicted string must match the ground truth character-for-character after normalisation so even a score of 0.136 represents meaningful precision. The LLM's score of 0.000 indicates that its responses, while possibly containing relevant information, consistently differ in phrasing, specificity, or completeness from the expected answers. This is consistent with LLM responses being verbose and paraphrastic rather than targeted.

F1 Score:

F1 score measures token-level overlap between prediction and ground truth, providing a more lenient measure of partial correctness. The gap here is the largest across all metrics: RAG's F1 score is 0.462 and the LLM's F1 score is 0.106. This indicates that RAG answers share substantially more content words with the correct answer, even when the exact string does not match.

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 10, June 2026
Impact Factor: 8.2

Accuracy:

Using the substring-matching definition of accuracy, the RAG system achieved an accuracy of 54%, compared to 36% for the standalone LLM, making RAG approximately 1.5× more accurate. The LLM's baseline performance is not insignificant, as it reflects cases where answers are well represented in its pre-training data. However, the improvement observed with RAG becomes more pronounced for document-specific queries, where the LLM lacks direct exposure to the provided data and is therefore unable to reliably retrieve correct information.

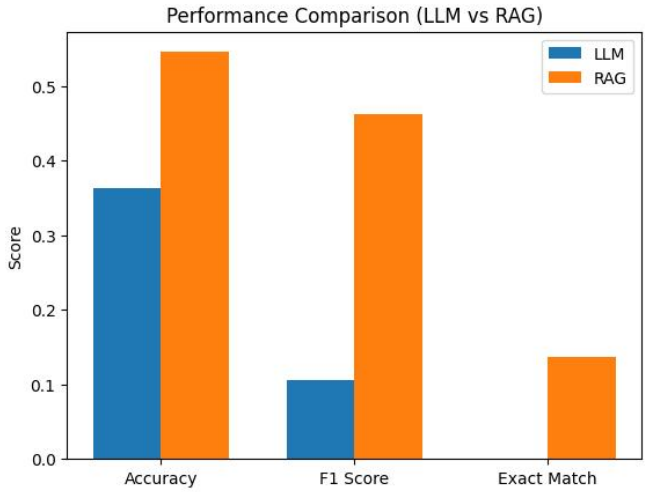

Fig 1. Accuracy, F1 score and Exact Match result comparison for LLM vs RAG

Hallucination Rate:

This represents one of the most critical findings of the study. The standalone LLM exhibited a hallucination rate of 100%, indicating that for every query, the generated response contained zero tokens drawn from any retrievable context. This outcome is expected, as the LLM has no mechanism to access the uploaded PDFs and therefore relies entirely on its parametric knowledge, which cannot align with document-specific content. The hallucination metric, as defined in this study, effectively captures this limitation.In comparison, the RAG system reduced the hallucination rate to 68%, reflecting a 32-percentage-point improvement over the baseline. While this demonstrates the effectiveness of retrieval augmentation, the hallucination rate remains relatively high. Residual hallucinations in RAG can be attributed to two primary factors: (a) retrieval failure, where the relevant information is not included in the top retrieved passages, and (b) partial grounding, where the model combines retrieved context with its internal knowledge.

Average Response Time:

Counter-intuitively, the RAG system was faster, with an average response time of 6.44 seconds, compared to 9.87 seconds for the standalone LLM. This difference can be explained by variations in response length and prompt constraints. The RAG prompt explicitly enforces concise, single-line answers grounded in the provided context. In contrast, the standalone LLM prompt provides weaker constraints, and the model often produces longer, less focused responses. This observation challenges the common assumption that retrieval overhead necessarily results in slower system performance.

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 10, June 2026
Impact Factor: 8.2

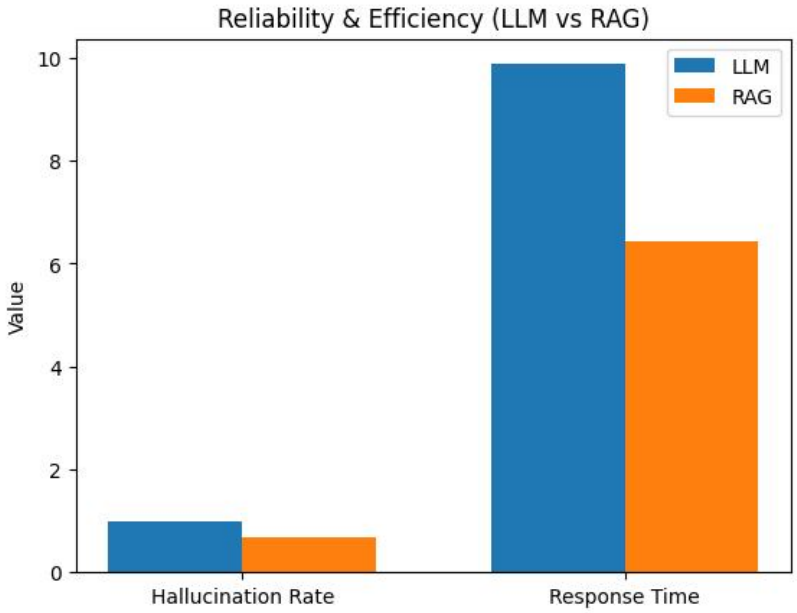

Fig 2. Hallucination Rate and Response time result comparison for LLM vs RAG

The LLM performed relatively well on questions where general world knowledge suffices . Its failure was systematic and complete on document-specific questions those requiring knowledge of content within the uploaded PDFs. Since these document-specific questions are precisely the use case that motivates building a custom QA system in the first place, the LLM's 100% hallucination rate makes it categorically unsuitable for this application.

VI. CONCLUSION

In this paper, we have presented an implementation and evaluation of the Retrieval-Augmented Generation System, along with its comparison to the Large Language Model (LLM). Both of these models have been built based on the same base - Mistral-7B-Instruct-v0.1, so the difference is only caused by implementing the retrieval system. For the purpose of implementing RAG, we have used the following components: BAAI/bge-small-en embeddings, FAISS vector index, and LangChain-based retrieval framework. Evaluation of both methods was carried out with a set of 25 questions. According to the obtained results, RAG outperformed LLM in all aspects. RAG showed the following scores: accuracy - 54%; F1 score - 46%. On the contrary, LLM demonstrated the following results: accuracy - 36%; F1 score - 10%. Additionally, it is important to note that RAG model was faster than LLM model. In summary, the results provided above demonstrate the effectiveness of the RAG technique regarding improving the accuracy and efficiency of the generation process. Most likely, the most striking conclusion drawn during the research process is the fact that the standalone model is entirely reliant on parametric knowledge (the hallucination rate is 100%). The conclusion is that despite the fact that the model has the ability to produce coherent texts and answer general questions properly, the standalone language model cannot extract information from outside sources or the input text provided by the user. The implementation of the RAG technique resolves this problem.

However, as one can see from the experiments conducted, there are still some problems related to the efficiency of the retrieval augmentation generator technique that need to be considered. To begin with, the fact that some inaccuracies remain and the existence of hallucination proves that the retriever does not manage to retrieve all the necessary documents and consider their entire content. There are several possibilities for further study suggested by this

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 10, June 2026
Impact Factor: 8.2

experiment. For instance, it is possible to refine the retrieval results through cross-encoder re-ranking. Besides, it is possible to use other retrieval methods to enhance the retrieval process and minimize hallucinations.

To conclude, it should be stated that this experiment proves that the application of the retrieval augmented generation technique can be an effective method of dealing with the issue of hallucination and lack of knowledge in large language models.

REFERENCES

1. Brown, T., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901. https://arxiv.org/abs/2005.14165
2. Ji, Z., Lee, N., Frieske, R., et al. (2023). Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12), 1–38. https://arxiv.org/abs/2302.03629
3. Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474. https://arxiv.org/abs/2005.11401
4. Gao, Y., Xiong, Y., Gao, X., et al. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997. https://arxiv.org/abs/2312.10997
5. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30. https://arxiv.org/abs/1706.03762
6. Karpukhin, V., Oguz, B., Min, S., et al. (2020). Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906. https://arxiv.org/abs/2004.04906
7. Shuster, K., Poff, S., Chen, M., et al. (2021). Retrieval augmentation reduces hallucination in conversation. arXiv preprint arXiv:2104.07567. https://arxiv.org/abs/2104.07567
8. Creswell, J. W., & Creswell, J. D. (2018). Research design: Qualitative, quantitative, and mixed methods approaches (5th ed.). SAGE Publications. https://us.sagepub.com/en-us/nam/research-design/book255675
9. Jurafsky, D., & Martin, J. H. (2024). Speech and language processing (3rd ed. draft). Stanford University. https://web.stanford.edu/~jurafsky/slp3/
10. Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250. https://arxiv.org/abs/1606.05250