2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC) | 979-8-3503-7519-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICAAIC60222.2024.10574972

Robust Multi Model RAG Pipeline For Documents Containing Text, Table & Images

Pankaj Joshi
Avasant intern Software Engineer (Labs)
Avasant Advisory India Pvt. Ltd.
Delhi, India
pankajjoshirpvv@gmail.com

Aditya Gupta
Avasant intern Software Engineer (Labs)
Avasant Advisory India Pvt. Ltd.
Delhi, India

Pankaj Kumar
Avasant Technical Lead (Labs)
Avasant Advisory India Pvt. Ltd.
Delhi, India

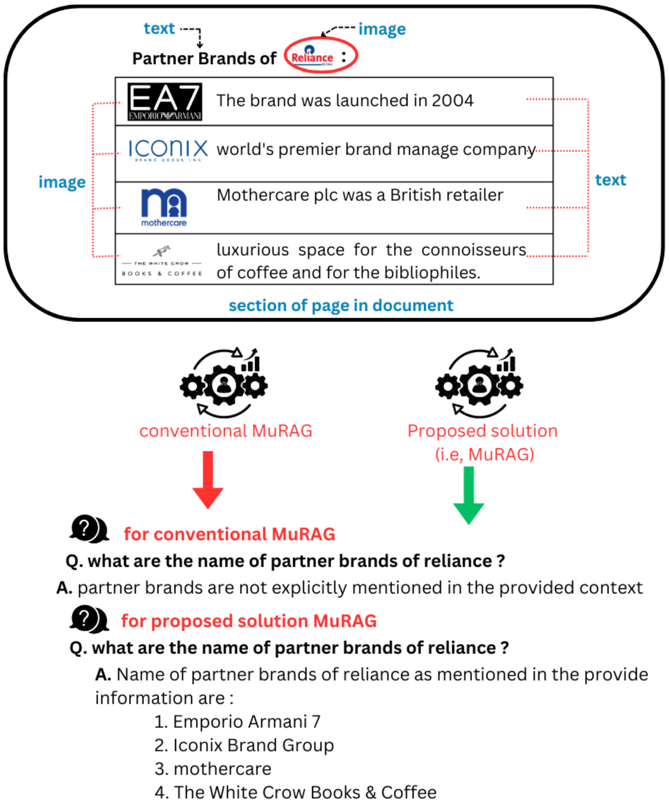
Manas Sisodia
Avasant Associate Software Engineer (Labs)
Avasant Advisory India Pvt. Ltd.
Delhi, India

***Abstract*—RAG (Retrieval Augmented Generation) is generally used for generating results from the existing knowledge-base. RAG refers to finding references (R), Adding references (A) and improving generation(i.e, answers to the question) (G). MultiModel-RAGs are used for generation of results over the documents which contain images and texts. There exists multiple different Multimodel-RAGs but these are not still efficient in generation of the results from the documents which contain relationships between images and texts. This study has proposed the solution to enable effective retrieval and generation of results, which includes the relationship between images and texts. The comparison of proposed Multimodal RAG with four different datasets (i.e., Short-form-type-QA, Long-form-type-QA, MCQ-type-QA, True-False-type-QA) shows the proposed solution improves the effectiveness of the existing Multimodal RAGs. Testing of proposed Multimodal RAG over two different other multimodal LLM i.e, Open-AI & Gemini helps in deciding whether the proposed solution fits best with LLM in different cases.**

***Index Terms*—MuRAG(Multimodal Retrieval Augmented Generation), LLM(Large Language Models), RAG(Retrieval Augmented Generation), GenAI(Generative AI).**

I. Introduction

Multimodal LLMs (Large Language Models) bring the cutting edge advancement in the field of generative AI. Multimodal LLM makes it easy to query over the documents which contain images and text [1]. There exists Multimodal RAGs which query over the document which contains images and texts [1] but they are still not effective in generating the results over the documents which contain the relationship between image and text (i.e, mentioned in fig. 1). The reason behind this is not because of hallucination exhibited by LLMs, maybe it is one of the reasons but it is not the only reason. Although there are various reasons for Hallucination in LLMs ([2], [4]). LLMs also hallucinate because the context which is provided to LLM for the generation of results are not relevant enough for generation of result of the query provided [6].


text
image
Partner Brands of
Reliance
EA7 EMPORIO ARMANI
The brand was launched in 2004
ICONIX BRAND GROUP INC.
world's premier brand manage company
mothercare
Mothercare plc was a British retailer
THE WHITE CROW BOOKS & COFFEE
luxurious space for the connoisseurs of coffee and for the bibliophiles.
image
text
section of page in document
conventional MuRAG
Proposed solution (i.e, MuRAG)
for conventional MuRAG
for proposed solution MuRAG

Q. what are the name of partner brands of reliance ?
A. partner brands are not explicitly mentioned in the provided context

Q. what are the name of partner brands of reliance ?
A. Name of partner brands of reliance as mentioned in the provide information are :
1. Emporio Armani 7
2. Iconix Brand Group
3. mothercare
4. The White Crow Books & Coffee

Fig. 1. This figure shows that the proposed solution works well with the documents which contain the images and text relationship, as compared to the existing Multimodal RAG's.

There are different conventional MuRAG pipelining for the documents containing images and text, which go through several steps [7] which includes first partitioning of the documents into images and text and then separately proceeds with both text and images, but this approach will not work. Because here the images and text are processed separately so it's very difficult to establish the relationship between the text and the images as they don't have anything in common between them. On account of the above issue this paper tries to propose a solution, which addresses the above problem, proposed

MuRAG will first extract the text from the document and process it as usual but for the images it will making the image of each page present in the document(i.e., convert each page of the document into a single image) and then proceed further and all the processing steps are further explained in-detail in the methodology. Here this proposed solution will be able to maintain the relationship between images and text present in the document. This paper helps the existing RAG to improve their effectiveness in the context of the relationship between images and text present in the documents. The results which are generated on Four dataset of Short-form-type-QA [3], Long-form-type-QA [8], MCQ-type-QA [9], True-False-type-QA [10] shows that the proposed solution changes the way of looking at the problem and also helps to improve the effectiveness of the generation. In order to facilitate the others to regenerate mention results, all source codes will be published later. Further this study will also show the comparative analysis of the Proposed MuRAG with two different LLMs (i.e, OpenAI, Gemini) which helps to understand which LLM performs best with the proposed MuRAG in different scenarios.

This research study contributes in the following ways like: 1) This paper proposes the scenarios where the conventional MuRAG is not able to perform well, and also propose the solution for that problem. 2) Testing of proposed MuRAG over four different datasets(i.e, Short-form-type-QA, Long-form-type-QA, MCQ-type-QA, True-False-type-QA) which shows that the proposed solution will be helpful to increase the effectiveness of the conventional MuRAG's. 3) Perform comparative analysis of the MuRAG's with two different LLMs(i.e, OpenAI and Gemini) which gives a better understanding of MuRAG with different LLMs in different scenarios. This research was carried out as part of an Internship research work at Avasant Advisory India Pvt. Ltd.

II. Related Work

Survey of Hallucination in LLM

Large Language Models gradually become popular as it leads to more coherent and fluent use of NLG (Natural Language Generation), leading to improvement and advancement in the downstream task such as data-to-text generation. However it's apparent that these LLMs are unintendedly prone to hallucinate ([2], [4]), which degrades the system performance and fails to generate accurate results in the field of generative AI. In order to address this issue there are many studies already done in the past which comprise all the aspects where the LLMs are prone to Hallucinate [3]. Hallucination in LLMs leads by various factors such as factual error by which LLM are struggling [5], outdated information and incorrect knowledge base & provided context information [12]. These are some serious factors which lead to hallucination in LLM. Hallucination in LLMs generates incorrect results which further lead to misguide the users. This can hurt the user's hope and trust from the most practical applications.

RAG (Retrieval Augmented Generation)

Language Models are pre-trained on some datasets and don't have the knowledge of present scenarios. Suppose, someone want to question-answer the LLMs on it's own knowledge base. In order to do that it will look for RAG based approaches which help us to question-answer the LLMs on it's own knowledge base [13]. RAG provides the methodology by which user can retrieve the relevant document-chunks from the vector-database using vector-similarity search [13] and then use these retrieved documents for the further generation of results from the LLMs [17]. RAGs are struggling with the semantic retrieval of the document-chunks. In-order to solve the problem of semantic retrieval of documents, the 'Advance-RAG approach' came into picture. But still not effective for the document which comprises images, texts & tables and their relationships with each other.

CRAG (Corrective Retrieval Augmented Generation)

CRAG [6] proposed the solution which enhances the context information which is retrieved by the retriever from the vector databases and further helps the LLMs to generate the results in retrieval augmented generation techniques. In this approach the retrieved documents are classified into three categories 'correct', 'incorrect' and 'ambiguous' [6]. In order to divide the retrieved documents into three categories T5-Large Model([15], [6]) is fine tuned and provide the score to the retrieved documents which is used for the further processes involved in the generation. This CRAG approach helps in enhancing the semantic retrieval of the document-chunks.

Multi-Modal Retrieval Augmented Generation

MuRAGs(Multimodal Retrieval Augmented Generation) are used with Multimodal LLMs which are capable of processing the images for the generation of results. Conventional Multimodal RAG processes the images and text separately for the generation of results. There are various past studies which are based on the experimental analysis of the existing multimodal-LLMs on various datasets. [1] Their study is based on the MuRAG, which fine-tune the LLM with encapsulated images and their related text from the web-search, for the better generation purpose. Their study is majorly focused on the images and fine-tuning tasks of language models on two different datasets(i.e, WebQA, MultimodalQA).

All the recent studies related to proposed work are mentioned(i.e, [13], [6], [1]) by highlighting their main difference. These approaches are aimed at enhancing the way of generation. This study mainly focuses on achieving effective and efficient retrieval. This study has proposed a solution which helps in solving the problem with generation of results from the documents which contain the relationship between images, text and tables(i.e, fig 1). This study makes the first attempt to solve the problem for the documents, which contain relationships between images, text and tables and helps in the effective retrieval of results from the documents.

III. Formulating Approach

Here, we consider an input document(D), which contains images, text and tables. I(θ) is the function which generates the image of the pages present in the document.

$$I(\theta = \text{Document (D)}) \Longrightarrow \sum_{i=1}^{n}[\text{Pg}_{\text{img}}]_i$$
$$= (\text{termed as } X)$$

Here Pg(img)i is the image of the ith page of the document(D). C(θ) extract the text chunks from the input document(D).

$$C(\theta = \text{Document (D)}) \Longrightarrow \sum_{i=1}^{n}[\text{Text - Chunk}]_i$$
$$= (\text{termed as } Y)$$

Hence Document(D) is converted into a sum of page-images and extracted text chunks.

$$D(\text{Document}) \Longrightarrow \sum_{i=1}^{n}[\text{Pg}_{\text{img}}]_i + \sum_{i=1}^{n}[\text{Text - Chunk}]_i$$
$$= (\text{X + Y})$$

Function En(θ) generates the base64 encoded image of the given input image. Sn(θ) this function takes images as an input and generates the summary of the encoded image. At(θ) this function takes the images as an input and attaches the summary with the Image Documents.

$$x' = \sum_{i=1}^{n}\text{At (Sn (En (Pg}_{\text{img}})))$$

or

$$x' = \sum_{\text{t}=Pg_{(\text{img})1}}^{Pg_{(\text{img})\text{n}}} \text{At (Sn (En (t)))}$$

Here in above equations Page-images are first encoded by En(θ) function then Sn(θ) function generates the summary of the page-images further At(θ) function is used for attaching the summary with the image-documents. so, the final images documents will becomes X'. Now, the processed documents will become D' as :
document(D') = X' + Y

$$D' = \sum_{\text{t}=Pg_{(\text{img})1}}^{Pg_{(\text{img})\text{n}}} \text{At (Sn (En (t)))} + \sum_{i=1}^{n}[\text{Text - Chunk}]_i$$

Here, D is the processed document.
Now, suppose the output result is denoted as "op" and inputs are X' and Y (i.e, using equation (4) and (2)) which are the components of the processed document(D')[i.e, equation(6)]. First retriever retrieves Top-k text-documents(D_1, D_2, ... , D_k) and top-k image-documents (I_1, I_2, I_3, ... , I_k) from the n-inputs which are X' and Y then from these top-k retrieved documents the language models produces the output "op" which is expressed as :

$$\text{P}\left(\frac{op}{X', Y}\right) =$$
$$\text{P}\left(\frac{D_1 \cdot D_2 \cdot D_3 \ldots D_k \cdot I_1 \cdot I_2 \cdot I_3 \ldots I_k}{X', Y}\right) .$$
$$\text{P}\left(\frac{\text{op}}{D_1 \cdot D_2 \cdot D_3 \ldots D_k \cdot I_1 \cdot I_2 \cdot I_3 \ldots I_k}\right)$$

Detailed explanation as well as experimental results of the approach is explained in the further sections.

IV. Methodology

Fig. 2. shows the overview of the architectural flow of proposed Mu-RAG which helps in solving the problem related with documents which contain the relationship between images and texts(i.e, as mentioned in fig. 1). In the first phase of proposed Mu-RAG pipeline, text chunks are extracted from the documents and for the images instead of extracting the images from the document it will convert pages into images which helps in maintaining the relationship between images and text present in the document. suppose a document(D) which contains n-number of images and text chunks and if Mu-RAG will extract the images and text separately then it's very difficult to maintain the relationship between two entities which are further processed individually and also if images are large in number then the process of encoding, storing and generate embeddings of each image individually is very heavy and costly process, so generating page-images will solve these problems.

Generating Knowledge-Base: Knowledge-Base is kind of a storage for own confidential data. Generally knowledge-Base contains embeddings of images and text chunks present in the documents, so it's important to generated the knowledge base properly and efficiently so that high accuracy can be achieved. Hence, In-order to generate own knowledge-base the next task is to store the images properly which helps in generating the efficient retrieval knowledge-base. so, proposed Mu-RAG try to solve this problem by encapsulating the encoded images with attached detailed summary.images are encoded because it's very crucial for the LLMs to provide the detailed & relevant context. Encoded images are easier for the LLMs to read and generate the desired results as they contain the detailed information about the image.

Now, the next thing is to generate the embeddings of the processed documents. Embeddings are the vector representation of image and text chunks in the vector store. Embeddings provides the simple and efficient way of using vector similarity search. Hence embeddings of the text-chunks and the embeddings of the processed image-docs(i.e, documents which contain the encoded images with detailed summary) are generated and then stored them into the vector-store(i.e., text-store and image-store) which becomes knowledge base. The vector-store which is used to generate the mention results is quadrant

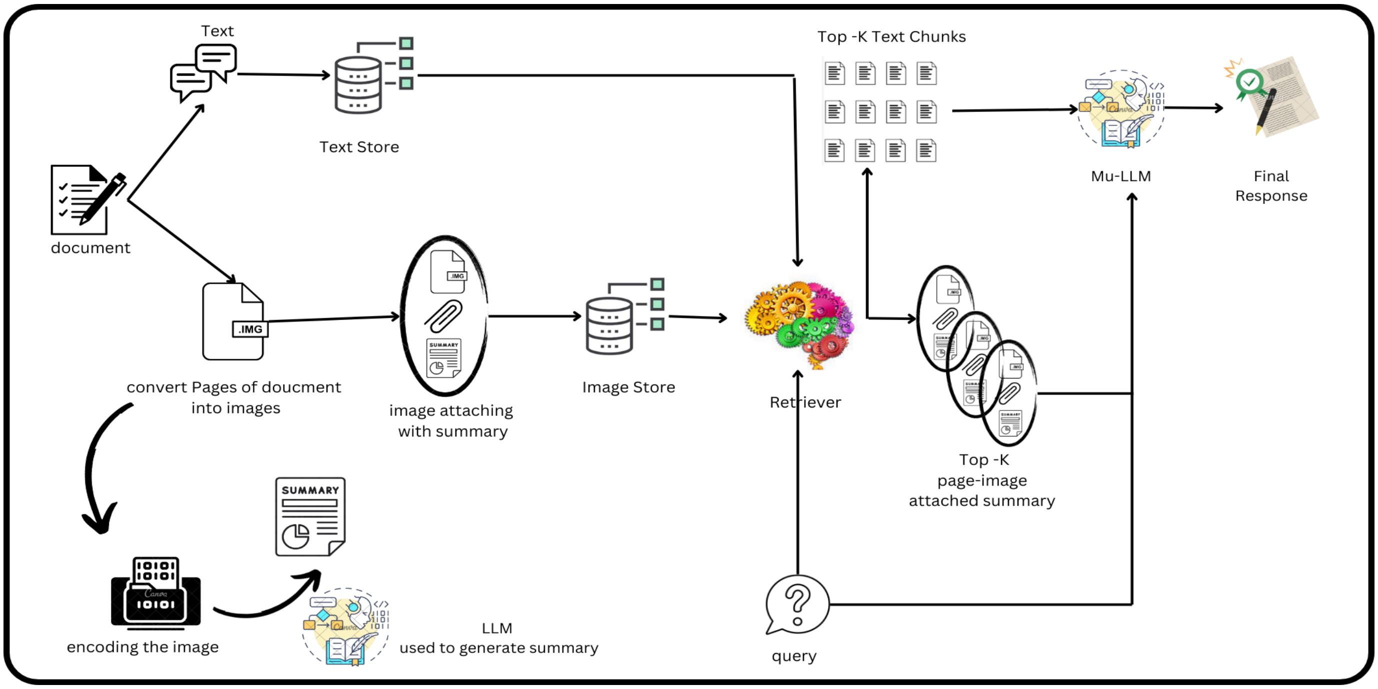

Text
Text Store
Top -K Text Chunks
Mu-LLM
Final Response
document
.IMG
convert Pages of doucment into images
image attaching with summary
Image Store
Retriever
Top -K page-image attached summary
SUMMARY
encoding the image
LLM used to generate summary
query

Fig. 2. This figure overview the architectural flow of the Multi modal RAG working of mention proposed solution. Text chunks are extracted first then pages of the documents are converted into images. images are further encoded and the summary of the images are attached with the page-image and then further processing is followed.

DB. Now, Generated Knowledge base contains the text chunks and processed image-docs of provided own documents. Now, this generated knowledge base is used further in the process of retrieval of related context with respect to users' query.

Retrieval and response generation: Retriever is used to retrieve the top-k relevant chunks from vector-store(i.e, knowledge base) which are further used for generation of the results. Retrieval extracts the similar text and image chunks from the knowledge base based on the similarity-search. There are different types of vector similarity search such as : dot-product, cosine similarity, Manhattan distance, euclidean distance. Here cosine similarity search is used to generate the given results, it is used for the retrieval of the top-k related document from the vector-store. Retrieved top-k results contain the processed image-docs(i.e, contain encoded image with summary attached) and text-chunks. Then these retrieved top-k documents and user query is further provided to Mu-LLM in-order to generate the final results.

In the above proposed solution the performance of Mu-RAG is improved in the area where the document contains the relationships between images and text. RAG first generates the page-images of the document with attached detailed summary so that relationship can be maintained and ultimately performance can be improved. By using the attached summary of page-images, RAG is able to connect the images with the provided text present in the document. Once relationship is maintained then the proposed Mu-RAG will be able to perform better than the conventional RAG's for the queries which needs the relationship b/w images and text.

This paper further evaluate proposed Mu-RAG into two phases: first is retrieval evaluation and second is response evaluation, Using four different datasets for retrieval and response evaluation. More detail about the experimentation and datasets are provided in the further section.

For the purpose of retrieval evaluation consider two parameters one is HR(i.e, hit-rate) and second is MRR(i.e, mean reciprocal rank) and for the purpose of response evaluation calculate mean-correctness-score, mean-relevancy-score and mean-faithfulness-score.

here llm-as-a-judge is used for response evaluation. evaluation and results are described in detail in the further sections. These results show that the proposed solution helps in proposing a methodology which effectively & efficiently helps in retrieval and generation of results and also solves the problem of maintaining the relationships between images and text present in the documents.

V. Experimentation

After finalizing the multi-modal RAG, this study has conducted the experimentation to evaluate the performance of RAG.

A. *Datasets Used*

Here, the OpenAI large language model is used to generate 5-6 questions per data chunk of the selected document. The questions were generated by the LLM following the very specific description as provided in the prompt to the LLM. mentioned four datasets are:

Short form type QA: This dataset is very similar to the Pop QA dataset, which contains 344 short form questions. [3]. Question and answer both are of approximately single line length aiming at improving the performance of question answering system. [3]

Long form type QA: This dataset contains 358 long form question and answers. Longform question datasets are essential for training and evaluating RAG pipeline that need to understand and generate detailed, nuanced responses to complex inquiries. [8]

True false type QA: Following previous work [10] this dataset contains 286 true-false type questions. This type of dataset helps in evaluating the rag for its fact-checking capabilities and turns out to be yet another important aspect to check proposed pipeline. [10]

MCQ type QA: Based on the previous work [9] create a dataset consisting 246 MCQ type questions, the model is tasked to select the correct option out of 4 choices. The dataset consists of single line questions and four options to each query. [9]

B. *Evaluation Target*

The multi-modal rag evaluation aims at two key components:

Retrieval Quality - Retrieval Quality is Determining the effectiveness and relevancy of the context sourced by the retriever component. here hit-rate and MRR metrics is used for measuring the retrieval quality.

Generation Quality- The generator's ability to combine meaningful and cogent responses from the recovered (or retrieved) context is the primary criterion for evaluating the quality of the generation.

C. *Evaluation Metrics*

In order to evaluate the efficiency of the retrievers, Hit Rate and Mean Reciprocal Rank metrics is used.

Hit Rate(HR): A binary score that indicate whether the right chunk is present in the retrieved chunks is called the hit rate. The hit-rate is considered as 1 if desired relevant document is present in the retrieved results otherwise 0. Typically, here the mean of hit-rates(HR) of all the queries is presented as :

$$HR = \frac{|U^{L}_{h_it}|}{|U_{all}|}$$

where $\left|\mathrm{U}^{\mathrm{L}}_{\mathrm{hit}}\right|$ is the number of queries for which the correct chunk containing answer is included in the top L retrieved recommendation list, $|\mathrm{U}_{\mathrm{all}}|$ is the total number of queries in the test dataset.

Mean Reciprocal Ratio(MRR): A measure used to assess the efficacy of recommendation engines, search engines, and other systems that rank a list of objects is called Mean Reciprocal Ranking, or MRR. It's very prevalent in information retrieval.

$$\text{MRR} = \left(\sum_{i=1}^{n} RR(i)\right)\frac{1}{n}$$

$$\text{RR(i)} = \left(\frac{1}{rank_i}\right)$$

Where, RR(u) is the reciprocal rank of correct document in the retrieved top K documents corresponding to the ith query of the test dataset and n is the total no of queries in the test dataset. MRR is simply the mean of reciprocal rank of all "n" no. of queries in the test dataset.

For the purpose of response evaluation here the following parameters are used such as: "mean-correctness-score", "mean-relevancy-score" and "mean-faithfulness-score". LLM-as-a-judge is used for the evaluation of responses [16].Mu-LLMs(i.e, "openai-gpt-4" and "openai-gpt-4-vision-preview") are used for evaluation of the responses.

mean-correctness-score: calculates correctness of the generated responses with respect to the actual reference outputs.

mean-relevancy-score: mean relevancy score calculates the relevancy of the retrieved images and text chunks from the knowledge base by considering the user input query.

mean-faithfulness-score: mean faithfulness score calculates the faithfulness of the Mu-RAG by considering the retrieved images and text chunks with reference to the input query. here obtained results are mention in the results section.

VI. Results and Discussion

Results corresponding to the evaluation of the Mu-RAG are divided into two phases : (1) retrieval evaluation results and (2) response evaluation results. For retrieval evaluation results, Hit-rates and MRR(mean reciprocal ratio) are used. here Hit-rate and MRR are calculated over four different datasets (i.e, Short-form-type-QA, Long-form-type-QA, MCQ-type-QA, True-False-type-QA). Mean hit rate is calculated as follow :

Hit Rate (i.e, mean hit rate) = (Number of Successful Recommendations / Total Number of Recommendations)×100%

Where:

- Number of Successful Recommendations: The number of recommendations that were correct or matched with the correct reference retrieval.
- Total Number of Recommendations: The total number of recommendations made to the user.

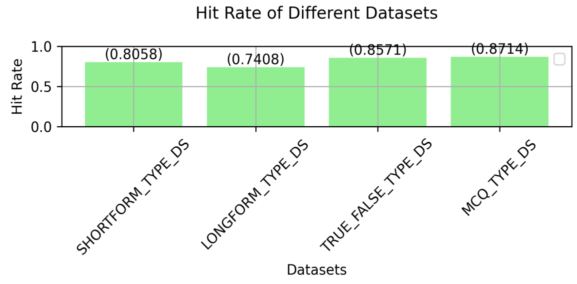


Fig. 3. shows the hit rates over four different datasets

TABLE I
RESPONSE EVALUATION RESULTS OF TWO DIFFERENT MU-LLMS OVER FOUR DIFFERENT DATASETS

	Mu-RAG-with-Gemini				Mu-RAG-with-gpt4-vision-preview			
	short-form	long-form	true-false	mcq-type	short-form	long-form	true-false	mcq-type
Mean correctness score (out of 5.00)	3.145	3.021	3.533	3.621	3.543	3.563	3.877	4.015
Mean relevancy score (out of 10.00)	6.550	6.940	7.650	7.770	7.420	8.230	8.510	8.991
Mean faithfulness score (out of 10.00)	7.770	7.230	8.320	7.620	8.450	8.770	8.650	8.830

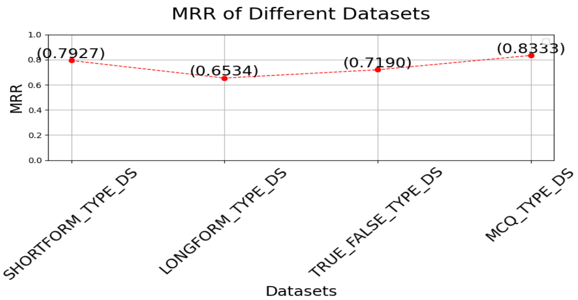


Fig. 4. shows the MRR scores of four different datasets.

TABLE II
VALUES OF HIT-RATE AND MRR OVER FOUR DIFFERENT DATASETS

Category	SF-QA	LF-QA	TF-QA	MCQ-QA
Hit Rate	0.8058	0.7408	0.8571	0.8714
MRR	0.7927	0.6534	0.7190	0.8333

fig.-4. gives us the measures of MRR of four different datasets. MRR here stands for Mean Reciprocal Rank also known as average reciprocal hit ratio. MRR can be calculated using below formulation:

$$\text{MRR} = \left(\sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \right) \frac{1}{|Q|}$$

Where:

- $\|Q\|$ is the total number of queries.
- $\text{rank}(i)$ is the rank of the first relevant item for the query.

Table-I gives the response evaluation results having mean correctness, mean relevancy and mean faithfulness as its key metric. The table also gives us an open comparison between the performance of proposed Mu-RAG using Google's Gemini as a multi-modal large language model and OpenAI's GPT-4-vision-preview as a multimodal large language model.

In mention table-[II] SF-QA, LF-QA, TF-QA, MCQ-QA refers to the short-form type QA dataset, long-form type QA dataset, true-false type QA dataset, MCQ-type QA dataset respectively.
Higher is the value of mean hit rate(HR) higher is the accuracy to retrieve the correct relevant chunk from the knowledge-base. Similarly higher is the MRR value higher is the accuracy of the retriever. Hence, In-order to validate the results we can simply check the values of mean hit-rate and the MRR corresponding to four different datasets(ie. table-[II]) and we get the Hit-rate and MRR which are very close to 1 hence our actual results are matching closely with the theoretical results.

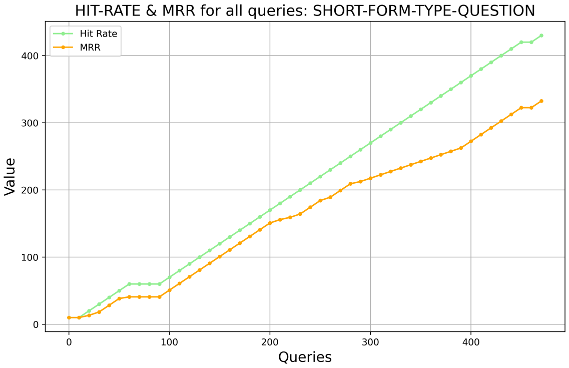


Fig. 5. This graphs shows the plot between queries and their corresponding summation of hit rate and MRR values for short-form type QA datasets

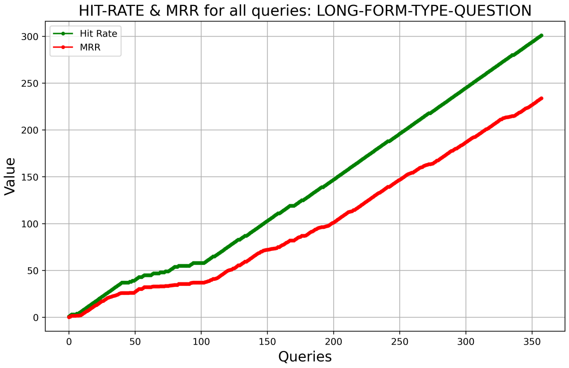


Fig. 6. This graphs shows the plot between queries and their corresponding summation of hit rates and MRR values for long-form type QA datasets

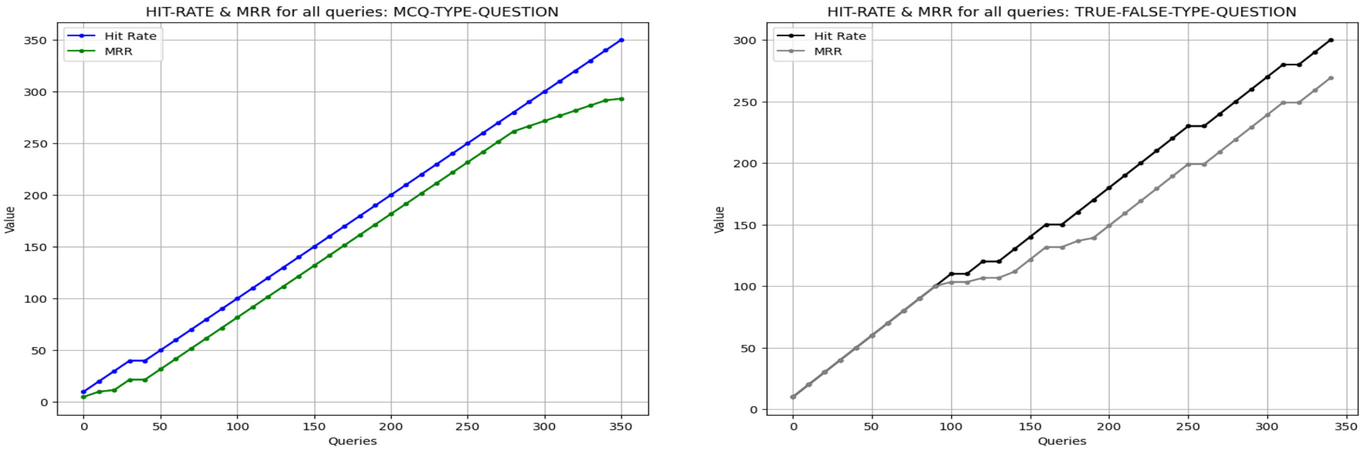


Fig. 7. These graphs shows the plot between queries and their corresponding summation of hit rate and MRR values for two different datasets: MCQ-type-QA and true-false-type-QA

VII. Conclusion

This research study has analyzed the problem where Mu-RAGs are facing challenges with effective retrieval of documents and tries to solve the problem of maintaining the relationship between images and text contained in the input documents. This study has proposed the solution, which helps in effective and efficient retrieval and generation of results from the document which contains the images-text relationships. Extensive experimentation of the proposed solution over four different datasets demonstrate the ability of the proposed Mu-RAG to significantly improve the process of generation. Experimentation also shows the accuracy, trust and relevancy scores of the two different Mu-LLM with proposed Mu-RAG, which helps to efficiently select the best LLM with different scenarios.

VIII. Acknowledgement

This research study was carried out as part of an Internship research at Avasant Advisory India Pvt. Ltd.

References

[1] Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, William W. Cohen, arXiv:2210.02928v2 [cs.CL] 20 Oct 2022, MuRAG: Multimodal Retrieval-Augmented Generator for Open Question Answering over Images and Text.

[2] Ziwei Ji , PictureNayeon Lee , PictureRita Frieske , PictureTiezheng Yu , PictureDan Su , PictureYan Xu , PictureEtsuko Ishii , PictureYe Jin Bang , PictureAndrea Madotto , PicturePascale Fung, [03 March 2023], Survey of Hallucination in Natural Language Generation.

[3] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, Hannaneh Hajishirzi, (Mallen et al., ACL 2023) [July 2023], When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories.

[4] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, Shuming Shi, arXiv:2309.01219, [Sun, 3 Sep 2023], Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models.

[5] Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, Yoav Shoham, arXiv:2307.06908 [Thu, 13 July 2023] , Generating Benchmarks for Factuality Evaluation of Language Models.

[6] Shi-Qi Yan1, Jia-Chen Gu2, Yun Zhu3 , Zhen-Hua Ling1, arXiv:2401.15884v2 [cs.CL] 16 Feb 2024, Corrective Retrieval Augmented Generation.

[7] Ruochen Zhao1 Hailin Chen1 Weishi Wang Fangkai Jiao, Xuan Long Do, Chengwei Qin1 Bosheng Ding1 Xiaobao Guo1 Minzhi Li, Xingxuan Li , Shafiq Joty, arXiv:2303.10868v3 [cs.CL] 1 Dec 2023, Retrieving Multimodal Information for Augmented Generation: A Survey.

[8] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, Hannaneh Hajishirzi, (Min et al., EMNLP 2023), [dec 2023], FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation.

[9] Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen Meng, James Glass, arXiv:2304.03728, [Fri, 7 Apr 2023], Interpretable Unified Language Checking.

[10] Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Peter Clark, arXiv:2102.03315 [Fri, 5 Feb 2021] , Think you have Solved Direct-Answer Question Answering? Try ARC-DA, the Direct-Answer AI2 Reasoning Challenge.

[11] Alon Talmor,1,2 Ori Yoran,1,2 Amnon Catav,2 Dan Lahav,2 Yizhong Wang3 Akari Asai3 Gabriel Ilharco3 Hannaneh Hajishirzi2,3 Jonathan Berant, arXiv:2104.06039v1 [cs.CL] 13 Apr 2021, MULTIMODAL QA: COMPLEX QUESTION ANSWERING OVER TEXT, TABLES AND IMAGES.

[12] Junyi Li1,3, Jie Chen1, Ruiyang Ren1, Xiaoxue Cheng1 , Wayne Xin Zhao1 , Jian-Yun Nie3 and Ji-Rong Wen, arXiv:2401.03205v1 [cs.CL] 06 Jan 2024, The Dawn After the Dark: An Empirical Study on Factuality Hallucination in Large Language Models.

[13] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela, Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.

[14] OpenAI, Open AI large Language Model, [September 25, 2023] GPT-4V(ision) system card.

[15] Mingye Wang , Pan Xie ,Yao Du and Xiaohui Hu , 14 June 2023, T5-Based Model for Abstractive Summarization: A Semi-Supervised Learning Approach with Consistency Loss Functions.

[16] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, Ion Stoica, arXiv:2306.05685, [Fri, 9 June 2023], Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.

[17] Garima Chhikara, Anurag Sharma, Kripabandhu Ghosh, Abhijnan Chakraborty, [Wed, 28 Feb 2024], arXiv:2402.18502, Few-Shot Fairness: Unveiling LLM's Potential for Fairness-Aware Classification