

Retrieval Augmented Generation for Document Query Automation using Open source LLMs

Khant Ko
University of Information Technology
Yangon, Myanmar
khantko@uit.edu.mm

Thwet Yin Nyein
University of Information Technology
Yangon, Myanmar
thwetyinnyein@uit.edu.mm

Khine Khine Oo
University of Information Technology
Yangon, Myanmar
khinekhineoo@uit.edu.mm

Thant Zin Oo
University of Information Technology
Yangon, Myanmar
thantzinoo@uit.edu.mm

Thet Thet Zin
Faculty of Computer Science,
University of Information Technology
Yangon, Myanmar
thetthetzin@uit.edu.mm

Abstract—Ollama provides access to powerful open-source language models that can be integrated into various applications. It supports local hosting, controlling the model's usage and data privacy. LLMs are large language models also known as deep learning models which are pre-trained on a vast amount of data. Integrating with retrieval augmented generation (RAG) can improve the efficiency of the LLM applications by retrieving custom data. The specific website or input of a particular document file can be integrated into the system. The document queries are automatically added to the Excel file using UiPath Automation. Firstly, the proposed system prompts by directly passing through the two LLM models: the Phi3 model by Microsoft with 3 billion parameters and the Llama 3.1 model by Meta with 8 billion parameters for text input and output, which can be accessed from Ollama released by Meta. To achieve a desired output, the system can also prompt by passing through retrieval augmented generation (RAG). Finally, analyze the results of the two outputs by directly using LLM models and embedding them with RAG. According to the evaluation result, RAG integration with llama3.1 improves response quality and relevance for custom data. Phi3 is better in the latency evaluation result.

Keywords— Large Language Model (LLM), Retrieval Augmented Generation (RAG), UiPath, Ollama, Phi3

I. INTRODUCTION

Document search capabilities have advanced in the way to retrieve and extract information from a large amount of data. In today's digital age, the importance of fast and accurate searching for a specific document can affect the decision-making in every organization. Document query system means finding the relevant documents and giving a response to a query. Traditionally, documents are stored in a specific file on the local desktop or a cloud. When organizations have a large amount of data, manual searching is not sufficient, and it takes several minutes or hours. To solve this problem, a large business organization uses the document query system.

To build an efficient document query system, large language models (LLMs) are impressive for their ability to recognize, translate, predict, and generate text on a huge set of data. An LLM function is like a black box that receives a text as a prompt and generates a text as an output. The box contains a neural network called a transformer network, which is a computing system representing layers of interconnected neurons to loosely mimic the human brain. The weights between two neurons are calculated and adjusted during the training phase and form parameters. The input text is transformed into a token as a word or part of a word for a single character. LLMs can predict the next word called

tokens based on the text it has observed. While LLMs can generate original content, the quality, relevance, and innovativeness of their output can vary and require human oversight and refinement. To improve the prompt, and get relevant responses add Retrieval Augmented Generation (RAG) and fine-tune to the LLM.

RAG is used to optimize LLM outputs by referencing external knowledge bases beyond the model's training data before generating a response. While LLMs are trained on vast datasets to perform tasks like answering questions and language translation, RAG enhances their ability to provide accurate, domain-specific information without retraining. It combines an LLM with a document search module, enabling human-like, fact-checked responses even with sparse data. For tasks requiring specific outputs, fine-tuning adjusts the LLM's parameters using smaller datasets, improving the model's performance in specialized domains. This boosts retrieval and overall LLM accuracy.

In the proposed system UiPath automation is used for data collection for RAG. Data collection for RAG plays an important role in this system. By using UiPath automation the system is able to extract the required amount of data from the website in a short period of time. The document queries from users are passed to two LLMs and RAG. Then responses of two LLMs without RAG and with RAG are compared and analyzed.

The remaining section of this paper is that related works for LLMs and RAG are presented in section II and LLMs are described in section III. section IV and section V present RAG and UiPath automation systems. Section VI describes the proposed system and evaluation and analysis and conclusion will be presented in section VII and IX.

II. RELATED WORKS

Since the announcement of ChatGPT in November 2022, a lot of attention has been directed to the Large Language Models (LLMs) due to their strong performance on a range of natural language tasks. The research area of LLMs is growing rapidly in many different ways. Several LLMs were released in 2023, gaining significant popularity. Notable examples include OpenAI's ChatGPT [8], Meta AI's LLaMA [9], and Databricks' Dolly 2.0 [10]. Brown, T. B and researchers [1] introduced the GPT-3 language model, which demonstrates impressive few-shot learning capabilities and can be used for various natural language processing tasks, including document query answering. Chen, D., and researchers [2] also provided various approaches to existing document query answering systems, including their

architectures, techniques, and evaluation metrics. Meta released Llama 3.1 April 2024. Llama 3.1 is the first available model that comes up with the top AI models [11].

RAG is an AI framework for retrieving contents from an external knowledge base to ground large language models (LLMs) on the most accurate, up-to-date information and to give users insight into LLMs' generative process. Lewis, P., and researchers [3] introduced a novel approach to combining retrieval and generation tasks, where a pre-trained language model retrieves relevant information from a large corpus and then generates a response based on the retrieved information. Wang, H., Yang, Z., & Zhang, J. [4] proposed a RAG model that leverages a large scaled knowledge-based response to improve the quality and accuracy of generated responses based on knowledge-intensive tasks. Radford and researchers [5] presented the GPT-2 language model, which is a powerful language model that can generate human-quality text. Zhang, Y., and researchers [6] proposed a neural network-based document query answering system that can effectively retrieve and answer questions from large-scale document collections.

Karpukhin, D., and researchers [7] introduced a dense passage retrieval model that can efficiently retrieve relevant passages from a large corpus of documents. Patrick Lewis, Ethan Perez, and group [12] presented retrieval-augmented generation for knowledge-intensive NLP tasks. They considered two models: the RAG sequence model and the RAG token model. During training, they retrieved the top k documents for each query. They showed that their RAG models obtain state-of-the-art results of an open-domain QA.

Automation has profoundly transformed the operational landscape of companies across various industries. Robotic Process Automation (RPA) has obtained growing attention within the digital transformation. [13] presented the effectiveness of RPA for data mining using UiPath. For small-scale organizations, it is very easy to implement and easy to export data in the required format.

In this system, UiPath is used for data collection and integration for LLMs and RAG.

III. LARGE LANGUAGE MODELS

Today, Large Language Models (LLMs) have been the majority trend in the technology field, especially in Natural Language processing. Their ability to understand and generate human-like text has made them attractive for applications in document query systems. The proposed system combines the capabilities of LLMs with retrieval techniques to improve the efficiency and accuracy of document queries. In this paper, two prominent open-source LLMs, Llama 3.1 and Phi-3 are compared to evaluate their suitability for our proposed system.

A. Llama 3.1

The Llama 3.1 open-source family was released in July 2024 by Meta AI. This latest family includes three models: larger 450B, medium 70B, and lightweight 8B, which are currently the most advanced models available. It is designed to be a foundational model capable of a wide range of natural language tasks, including text generation, translation, and summarization. Key features of Llama 3.1 include large-scale

TABLE I. KEY DIFFERENCES OF TWO LLMs

Model	Parameters	Context Length	Performance	Suitability
Llama 3.1 (8B)	8 billion	128K tokens	Good balance of performance and efficiency	-Suitable for deployment on devices with limited resources
Phi-3 (3B)	3 billion	4096 tokens	Basic performance	-Suitable for devices with low computational resources - Suitable for domain-specific

training, efficiency, and flexibility. The model is trained on a massive dataset of text and code which results in the ability to generate human-quality text. A context window of 128,000 tokens can handle much longer inputs and maintain context over extended conversations or documents. Responsible AI is designed to mitigate risks and reduce biases, ensuring safer and more reliable outputs. The family features a decoder-only transformer architecture, which enhances context understanding and enables the models to generate more human-like text.

While the larger models (240B and 70B) may provide even higher performance, the 8B model is selected for the proposed system. It has lower latency than larger models and is suitable for real-time applications. Lastly, due to its smaller size and simpler architecture, it can be more easily fine-tuned based on the scope and requirements of the proposed system.

B. Phi 3

Phi-3 is another open-source LLM introduced by Microsoft Research in 2023. It is a general-purpose language model capable of performing various natural language tasks such as text generation and question answering. Some of the models in the Phi-3 family include 175B which is the largest model in the Phi-3 family, 12B medium-sized model, and the mini Phi-3 model.

The key differences between the two LLMs applied in the system are shown in TABLE I.

IV. RETRIEVAL AUGMENTED GENERATION (RAG)

Retrieval Augmented Generation (RAG) is an AI framework that combines the strengths of retrieval and generation techniques to enhance the capabilities of LLMs. In our proposed system, we employ a RAG framework to effectively retrieve relevant documents from a corpus and generate informative responses based on the retrieved knowledge. The RAG framework is mainly composed of two components: embedding and similarity search.

A. Embedding and Vector Database

RAG (Retrieval-Augmented Generation) enhances the retrieval process by extracting relevant information based on user queries. Unlike simple keyword searching, RAG uses a vector database and embeddings to capture semantic meaning. Words or text chunks are transformed into numerical vectors, with similar meanings resulting in close vector values. For instance, vectors about "technology" will have vectors close to those related to "computers," but distant

from topics like "football" or "art." These vectors, along with their source indices, are stored in a vector database optimized for similarity searches. This enables more contextually accurate results.

B. Similarity Search

Similarity search is the operation used to retrieve relevant information in Retrieval Augmented Generation (RAG) systems. One of the widely used similar search algorithms is cosine similarity. It measures the similarity between the incoming query that had been embedded into a vector, and the existing document chunks vectors in the vector database by calculating the cosine of the angle between two vectors. The result of the cosine similarity of the vectors reveals how close the semantic meanings of the two are. The cosine similarity equation is:

$$\text{cosine similarity}(A, B) = \frac{A \cdot B}{||A|| ||B||}$$

Where A and B are the vectors of the user query and document chunk. $A \cdot B$ denotes their dot product, and $||A||$ and $||B||$ are the magnitudes of these vectors. The similarity value ranges between -1 and 1. A high cosine similarity value, closer to 1, indicates that they are relevant to each other.

In the proposed systems, the top ten chunks most relevant to incoming queries are extracted as context. The context will then be sent along with the query to LLM for further processing.

Example: There are two texts (snack and book) for the vector database and one incoming query text is (food). After they have been embedded, their vector will look like this:

```
food    = [0.039130427, 0.092805766, -0.20178486, ...,
           -0.072320595, 0.0027884755]
snack   = [0.05322736, 0.08168169, -0.17580228, ...,
           -0.018530631, 0.011965672]
book    = [0.015301478, 0.052100748, -0.21122386, ...,
           -0.046694092, -0.007191828]
```

The incoming query text vector (food) will be paired with the vectors in the database. The cosine similarity of two pairs is calculated and the pair with a higher similarity value can be said that they are more relevant.

```
Cosine similarity (food, snack)    = 0.7013791020298251
Cosine similarity (food, book)     = 0.5715535873036892
```

According to the result, the pair (food, snack) have a higher similarity value because the semantic meaning of food is closer to that of snack than to that of book.

V. UIPATH AUTOMATION

This section discusses the integration of UiPath with Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs) to automate repetitive tasks in document query processing. In the system, we streamline data extraction, query processing and response representation, improving efficiency and resource management through RPA.

A. Extracting Data

The automation process begins with UiPath's web scraping capabilities, targeting a specified online store. The extracted data is systematically stored in a dynamically structured DataTable with three key columns: Username, Comments/Reviews, and AI Response. Automating data extraction through UiPath enhances system scalability, reduces manual intervention, and ensures consistency in data collection.

B. User Interaction and Mode Selection

After the data extraction process, there is a popup on the user interface that allows the user to choose between two operating modes, Passthrough and RAG, which specify the next processing path. The decision was made in response to the need to either use LLMs capabilities only (Passthrough) or add document-derived context (RAG) to responses. This step ensures flexibility in AI processing by allowing users to either employ a Passthrough or RAG mode.

C. System Execution

Using CLI, the UiPath bot triggers the Flask server for the RAG application, establishing communication between the AI backend and UiPath workflow. The operation dynamically adjusts based on the selected mode.

In Passthrough Mode, the review text is directly delivered to the LLMs which only uses its internal knowledge to generate responses.

In RAG Mode, to enhance the AI's response capabilities, it involves extra stages where the user selects a document, enters text into the system, and forwards the document's path to the RAG program.

D. Handling HTTP Requests

UiPath facilitates HTTP-based communication between the RPA workflow and the AI processing backend. For RAG mode, the user selects a document, and UiPath sends the file path to the Flask server, which processes the document for retrieval augmentation. For both modes, reviews or queries are sent via HTTP requests, and responses are returned to the DataTable. This automated exchange allows the RPA to coordinate complex and multi-stage processes between user inputs and query processing.

E. Storing Results

As the final step of automation, the DataTable is exported to an Excel sheet containing the AI's responses, which, depending on the mode selected, either represent the responses from straightforward AI understanding or a document-assisted comprehension. Automating the storage of results enhances traceability and ensures that the system can function in a resource-efficient manner, preserving important data for future audits or evaluations.

By automating the processing pipeline with repetitive tasks such as data extraction, mode selection, HTTP requests, and result storage, UiPath eliminates the need for manual intervention and ensures a consistent, scalable process. This is particularly valuable in document query processing systems like the proposed system, where the volume of data and user interactions can be large and dynamic.

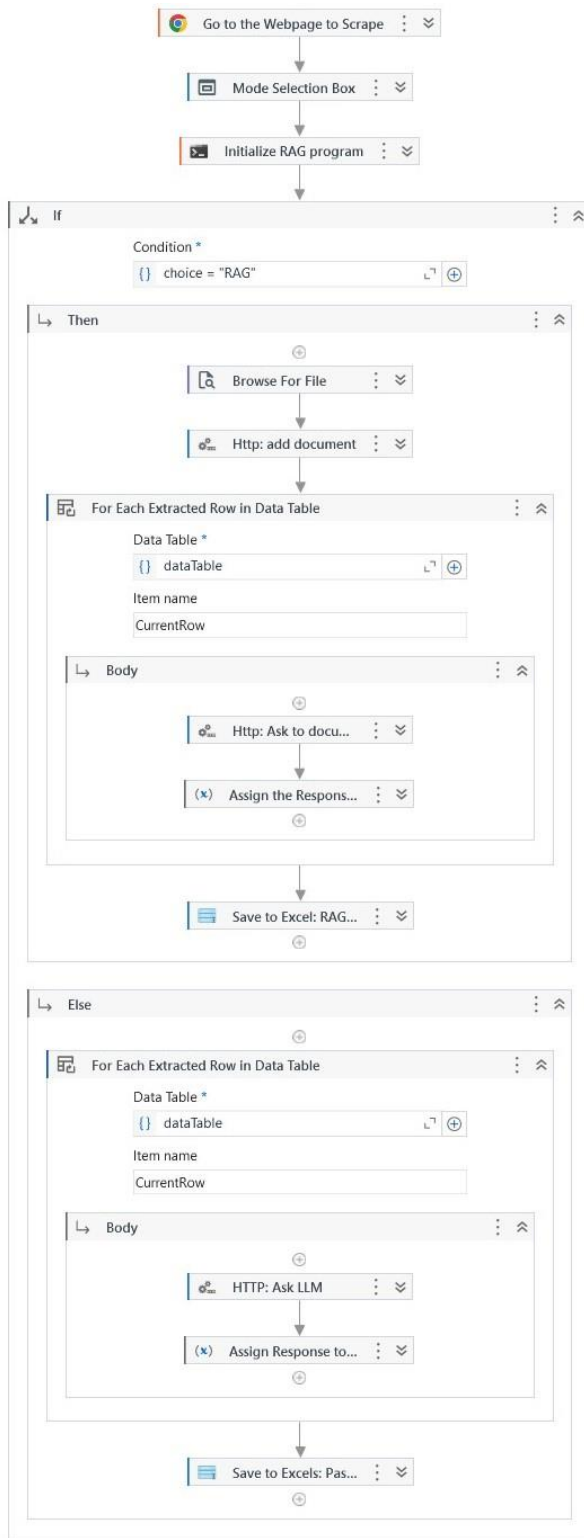


Fig. 1. UiPath Automation Workflow

VI. PROPOSED SYSTEM

It is now essential to be able to evaluate user-generated content, such as online reviews and comments, and draw insightful conclusions from it. Using extracted comments from online platforms and cutting-edge Large Language Models (LLMs), the proposed system processes and interprets user inquiries. The system has two different operational modes: Passthrough and Retrieval-Augmented Generation (RAG) to improve the precision and applicability

of replies. While the RAG mode adds pertinent documents retrieved in response to the query to enhance the LLAMA's processing, the Passthrough mode interacts with the LLM directly with raw comments and user queries. With this hybrid method, the system can save AI-generated responses for further analysis or record-keeping, while still being able to adapt to a variety of information needs. The main processes in this system are outlined below.

- Extract Comments: The system extracts comments from the specified website.
- Run LLM: The LLM is used to process the extracted comments and user queries.
- Select Mode: The system determines whether to use RAG or pass through LLM based on the user's preference.
- Passthrough LLM: If passthrough mode is selected, the query and extracted comments are directly passed to the LLM for processing.
- RAG: If RAG mode is selected, the system requests the most relevant documents based on the query and passes them to the LLM for processing.
- Source Responses: The LLM's responses are stored in an Excel file.

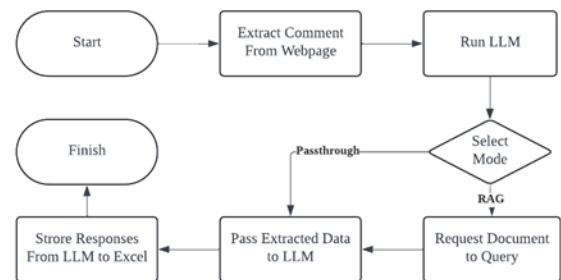


Fig. 2. Process Flowchart of the Proposed System

This document query system leverages Retrieval Augmented Generation (RAG) and the LLMs (Llama 3.1 and Phi3) to provide informative and relevant responses to user queries based on a given corpus of documents. RAG operates in six phases.

1. Document Chunking: The text splitter divides the given documents into smaller, more manageable chunks to facilitate efficient processing.
2. Embedding: The chunk and user's query are embedded into the same vector space using an embedding model. These embeddings capture the semantic meaning of the text.
3. Storing in Vector Database: The embedded texts are stored in a vector database, which allows for efficient similarity searches.
4. Similarity Search: The system performs a similarity search using Cosine similarity in the vector database to retrieve the most relevant document chunks based on their embeddings.
5. Prompt Generation: A prompt is constructed for the LLM, incorporating the original query, the retrieved relevant chunks, and any additional context that might be helpful.
6. LLM Processing: The LLM processes the prompt and generates a response.

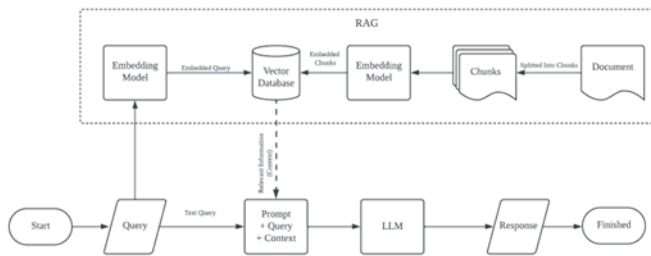


Fig. 3. Process Flow of RAG System

By combining the power of RAG and the LLM, this document query system offers a robust and effective solution for answering user questions based on a given corpus of documents. The below diagram describes the process flowchart of how RAG and LLMs are integrated into query embedding and prompt processing to source query- related knowledge.

VII. EVALUATION

In this evaluation, the performance of document retrieval and query-answering systems using Retrieval-Augmented Generation (RAG) with the LLMs are assessed, specifically Llama 3.1 and Phi 3. The effectiveness of these models is compared with and without the RAG framework to understand how RAG influences query performance and overall system capability. The evaluation metrics are measured in the below three criteria.

Relevance: The pertinence of responses and the retrieved documents to the queries

Response Quality: The coherence and informativeness of the generated responses

Latency: The time taken to generate responses

Due to the limited computation resources of the host machine, for each criterion, 100 queries are used to get an initial sense of how the system performs across two LLMs and selected modes. Promptfoo from promptfoo.dev is used as an evaluation tool.

A. Response Quality

To evaluate the response quality of the query system, customer reviews from the eBay website are passed to the candidates checking their quality of responses including informativeness and completeness of instructions in a given prompt. The following table shows the first three rows of the Excel sheets extracted in “Passthrough responses” and “RAG responses” sourced by the llama 3.1 model.

The table clearly shows that the response quality of RAG is better than the normal LLM passthrough. RAG systems not only can role-play customer service but also include relevant points about the shop according to customer comments. As for the comparisons between two RAG-integrated LLMs, the Llama 3.1 RAG has greater-quality responses than phi3 RAG. When the context is long, RAG with Phi3 can’t absorb the whole context and misses some information in responses.

B. Relevance

To measure the relevance of responded answers, 100 questions, answers and their source files to search for answers

TABLE II. PASSTHROUGH AND RAG RESPONSES

User Comments	LLM	LLM and RAG
This phone works wonders compared to another phones. I got to this phone at good quality. It seems better then good....	It sounds like you're very pleased with your phone purchase! You've highlighted several positives:1. **Good quality** : ...	Thank you so much for your kind review! We're thrilled that our phone exceeded your expectations and that you were...
Phone needed a few restarts to get up and running but seems to working properly now aside from one or two browser crashes. There ...	It sounds like you've got a phone that's been well-used, but still gets the job done. The fact that it needed a few restarts to get going suggests...	Hi, thank you for sharing your experience with our phone. We apologize for the initial issues with the device and the imperfections ...
Super fast shipping! Physical Condition is exactly as described. The phone works great! Once I put a cover on it, you won't even know it's a used phone! Also, let's ...	It sounds like you're extremely satisfied with your purchase from Cellfee! You've mentioned several key points that suggest a positive experience:* Super...	Dear valued customer,Thank you for your wonderful review! We're thrilled to hear that our shipping was fast and that the phone met your expectations. ...

are collected from TriviaQA dataset, which is publicly available. Collected answers and text from the source files are passed to Llama 3.1 and Phi-3 as queries and contexts. Resultant responses from LLMs are compared to their answers from TriviaQA using similarity comparison. Responses with similarity scores larger than 0.8 are considered relevant. The following graph is the result of the comparison of the relevance of the response. The red bar indicates the total number of answers relevant to the context in Llama 3.1 using RAG. The blue bar, the relevance of Phi3 RAG responses is low compared to Llama 3.1.

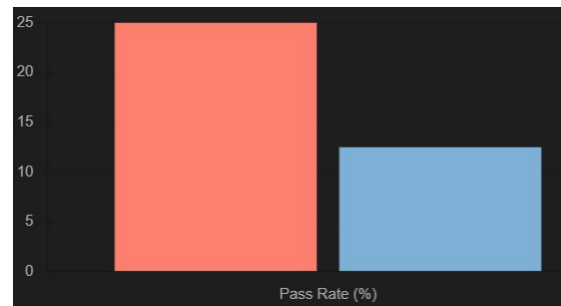


Fig. 4. Comparison of Relevance with Context: Llama 3.1 RAG (red) and Phi3 RAG (blue)

Phi 3 with RAG has a relevance score of 13% and Llama 3.1 with RAG has a relevance score of 25%. The results proved that Llama 3.1 achieved an even higher degree of relevance compared to Phi-3. The RAG framework significantly improves the relevance of the retrieved documents in both models. Llama 3.1 demonstrated a slightly better performance in relevance compared to Phi3. This suggested that llama 3.1 may have more efficient retrieval mechanisms or better integration with RAG, allowing it to provide documents that are even more closely aligned with the user’s queries.

C. Latency

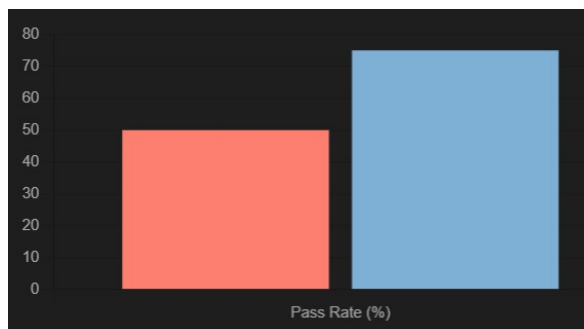


Fig. 5. Latency of Generating Responses: Llama 3.1 RAG (red) and Phi3 RAG (blue)

The latency evaluation is measured with the threshold 30s to compare the pass rate of both models. The queries and context used for evaluation are the same as that used in the relevance evaluation. Due to the limited computational resources, 30s is set as the optimal threshold value for the system currently. Compared to Llama 3.1 RAG, Phi3 RAG has more pass rate, meaning that RAG with the Phi3 model has faster response time than with Llama 3.1. Both models experience increased latency when RAG is employed due to the document retrieval process.

RAG substantially enhances the relevance of retrieved documents. By incorporating external knowledge, RAG allows the models to access more pertinent information, which directly contributes to more relevant and contextually appropriate responses.

The integration of RAG with LLMs (Llama 3.1 and Phi 3) demonstrates substantial improvements in accuracy, relevance, and response quality. The added retrieval component provides valuable context, leading to more precise and informative answers. However, this comes at the cost of increased latency. For applications where response time is less critical, the benefits of RAG in enhancing response quality and accuracy make it a compelling choice. Conversely, in scenarios where latency is a major concern, the performance gains from RAG should be weighed against the impact on response time.

VIII. LIMITATION

Hallucination is a key challenge in LLM. LLM hallucination occurs when the models generate incorrect or nonsensical information that is not related to the input query. There are confusingly different meanings in the retrieval phase. To solve this problem, the principles of garbage in garbage out are required. Retrieve relevant knowledge to get accurate or trusted answers. The training of LLMs is a computationally intensive procedure and requires more energy resources like supercomputing clusters or specialized hardware such as high-end professional GPUs. In LLM training, a large dataset is divided into smaller chunks and distributed across multiple GPUs. To hold the model,

parameters, and data parallelism, a large amount of GPU memory is required.

IX. CONCLUSION

The proposed system is applied by two LLMs and one embedding model for automatic document query systems. To obtain accurate responses for the users, RAG is also applied in the system. According to the evaluation result, RAG integration with llama 3.1 improves response quality and relevance. Phi3 is better than the latency evaluation result. Moreover, UiPath is used in the data extraction process. Thus, data collection time and cost can be very effective for the system. In the future, the system can be improved by integrating RAG with fine-tuning for specific target languages.

REFERENCES

- [1] Brown, T. B., Mann, B., Ryder, N., Subbiah, J., Dhariwal, P., Neelakantan, A., ... & Kaplan, J. (2020), "Language models are few-shot learners", arXiv preprint arXiv:2005.14165.
- [2] Chen, D., He, H., Wang, L., & Liu, T. (2020), "A survey of document query answering systems. IEEE Transactions on Knowledge and Data Engineering", 32(10), 2423-2440
- [3] Lewis, P., Liu, Y., Lapata, M., & Neubig, G. (2020), "Retrieval-augmented sequence generation", In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL) (pp. 6365-6375).
- [4] Wang, H., Yang, Z., & Zhang, J. (2022), "Retrieval-augmented response generation for knowledge-intensive tasks", In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL) (pp. 8321-8331).
- [5] Radford, A., Wu, J., Child, R., Luu, D., Amodei, D., Sutskever, I., ... & Chrabaszcz, P. (2018), "Improving language understanding by generative pre-training", OpenAI Blog.
- [6] Zhang, Y., Tang, J., & Zhang, J. (2020), "Document query answering with neural networks", In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 6671-6680).
- [7] Karpukhin, D., Lee, K., Wu, Y., Tsiroakis, V., Scialo, S., & Min, Y. (2020), "Dense passage retrieval for question answering. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)", pp. 4044-4055.
- [8] OpenAI, GPT-4 technical report, 2023, <https://arxiv.org/abs/2303.08774>
- [9] Meta AI, Introducing llama: A foundational, 65-billion-parameter language model, 2023. <https://ai.meta.com/blog/large-language-model-llama-meta-ai/>
- [10] Databricks J. Free dolly: Introducing the world's first open and commercially viable instruction-tuned LLM. <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>
- [11] Introducing Llama 3.1: Our most capable models to date (meta.com)
- [12] Patrick Lewis, Ethan Perez and group, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks", 34th Conference on Neural Information Processing Systems (NeurIPS 2020) Vancouver, Canada.
- [13] Yashodhan Ketkar and Sushopti Gawade, "Effectiveness of Robotic Process Automation for data mining using UiPath", 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), March 2021.
- [14] Mandar Joshi, Eunsol Choi, Daniel Weld, Luke Zettlemoyer. "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension", In Association for Computational Linguistics (ACL) 2017, Vancouver, Canada.