Querying Core Procedures Documents with Generative AI: A Case Study with benchmarks

ABDESSLAM, S. [(1)]; MECHERGUI, A. [(1)]; LE MESTRE, S.[(2)]; LUZ, D. [(2)]
(1) Forvia Carthage FIT, Tunis, Tunisia
(2) Forvia Paris Tech Center, Paris, France

Introduction

In today's fast-paced and dynamic business environment, easy and effective access to company core procedures documents is crucial for the success of any organization. The sheer volume and complexity of these documents, including such domains as safety and environment, purchasing, HR, make manual management and access processes error-prone and time-consuming. To address these challenges, the integration of artificial intelligence into core procedures documents querying presents a compelling solution. This article explores the innovative approach of leveraging generative AI to facilitate access and understanding of core procedures documents.

Basic concepts

Generative Artificial Intelligence (Sengar et al., 2024) is a type of AI that can create content, such as conversations, images, videos, music, and more. The key to its success lies in its capacity to capture the essence of the training data and use that understanding to create new, unique content. This generative process sets it apart from traditional AI techniques that are primarily focused on regression analysis, classification, or optimization tasks. By tapping into the creative potential of AI, generative models open new possibilities for innovation, artistic expression, and problem-solving across a wide range of domains.

Large Language Models (LLMs) are a class of artificial intelligence models designed to understand and generate human-like language. Characterized by a huge number of parameters, LLMs can capture intricate patterns and relationships within vast amounts of textual data. Their development has progressed from basic n-gram (Krishnan, A., 2023) models to complex architectures like transformers (Vaswani et al., 2017), which have significantly improved language understanding and generation. Leveraging deep learning, LLMs perform a wide range of natural language processing tasks, including language translation, text summarization, question answering, and conversational AI.

In addition to general capabilities, LLMs can be tailored for specific tasks using different methods, including Retrieval-Augmented Generation (Lewis et al., 2020). RAG enhances LLMs by allowing them to access external, domain-specific knowledge bases, extending model knowledge and improving accuracy and relevance when generating responses. This approach combines traditional language generation with information retrieval, making LLMs more adaptable and capable of delivering up-to-date, contextually rich answers in specialized domains such as contract analysis, customer support, and even scientific research, based on external data not originally used in the model training.

Proposed Solution and Implementation

In this article we present a framework for benchmarking RAG .

The project's goal is twofold:

1. Leverage the organization's proprietary documentation to develop a RAG-based conversational agent that answers questions related to core procedures across all domains. This agent can be deployed company wide. When a query is sent to the LLM, it is transformed into a vector embedding. Using this vector embedding, a function performs a semantic search based on the k-nearest neighbors (KNN) algorithm, returning the k most relevant information sources from the company's specific data. The LLM then synthesizes the information from these sources to generate a coherent and accurate answer to the user's query.
2. To develop a repeatable benchmark, including architecture, evaluation datasets and evaluation metrics, that can be reused for model performance evaluation and intercomparison once new models are released. By harnessing the best performing LLMs based on these data and metrics, the organization can optimize the agent's performance, enhance its capabilities, and evaluate how much it captures of the nuances of the company's documents, context, and specific language.

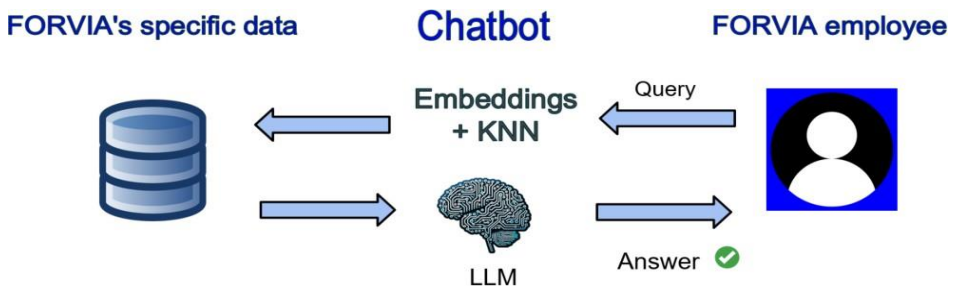

Figure 1: RAG-based conversational agent that embeds a user query, performs a semantic search for relevant information retrieval in order to generate a coherent response.

Project development

To achieve our objective, we utilized the Retrieval-Augmented Generation (RAG) technique and adopted a multimodal approach that integrates both text and image data.

The project is organized into several key phases:

1. Data Collection and Preprocessing :

The process begins with gathering a comprehensive dataset of internal core procedure documents. This data is then cleaned and prepared for model training. Text and image data are processed separately; the text undergoes chunking to break it into manageable segments, while irrelevant information is removed, errors are corrected, and formatting is standardized. Images are converted into a suitable format and then fed into a multimodal large language model to generate descriptions of the visual elements. These descriptions are preprocessed in a manner similar to the textual data to maximize the extraction of relevant information. We also embed the chunks into vector representations, converting them into a numerical format suitable for

Retrieval-Augmented Generation . Finally, these vector embeddings are stored in specific datasets. By incorporating both text and image modalities, we aim to enhance the model's ability to understand and generate contextually accurate outputs, thereby improving overall performance.

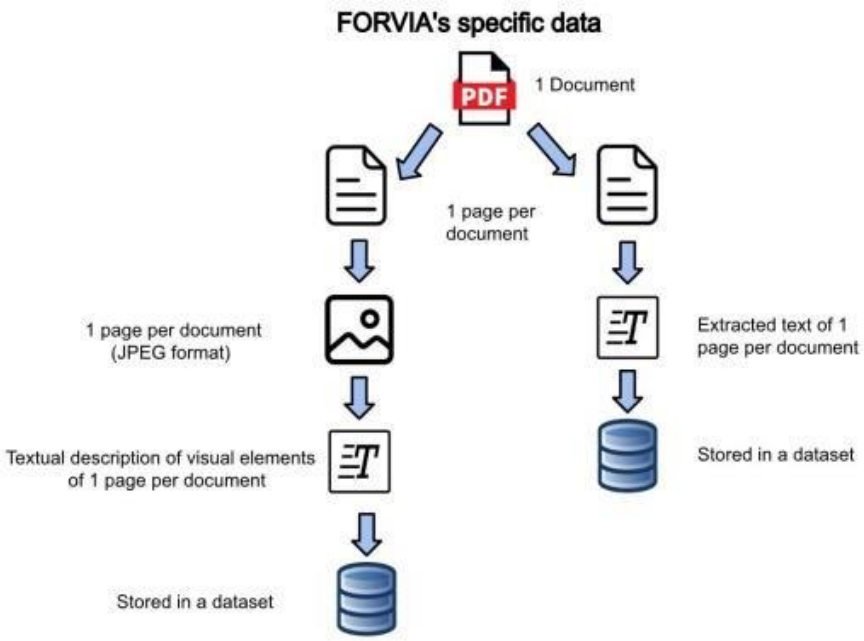

Figure 2 : Data preprocessing pipeline illustration: extracting and chunking text, converting to vector embeddings, processing images, and generating descriptions with a multimodal LLM which are then preprocessed like the text.

2. Model choice:

For this project, we aimed to explore a model capable of processing both text and images. We first developed a solution based on GPT4-turbo-with-vision but, ultimately, we leaned towards GPT-4o, as it is now considered OpenAI's flagship model. GPT-4o offers several advantages over GPT4-turbo-with-vision, including enhanced speed, lower costs, and multimodal capabilities, making it a more suitable option for our needs

GPT-4 Turbo-with vision and GPT-4o differ in their capabilities and focus. While both models share a large input context window of 128,000 tokens, GPT-4 Turbo stands out with its vision capabilities, allowing it to process and interpret images, as well as support JSON mode and function calling for more structured interactions. In contrast, GPT-4o is optimized for complex, multi-step tasks, and is faster and cheaper than GPT-4 Turbo-with vision, with a higher maximum output of 16,384 tokens, compared to GPT-4 Turbo-with vision's 4,096 tokens. However, GPT-4 Turbo-with vision's benefits from slightly more recent training data (up to December 2023), while GPT-4o's data is up to October 2023. Thus, GPT-4 Turbo is ideal for applications needing vision or system integration, whereas GPT-4o is better suited for tasks requiring extended, detailed responses with cost and speed efficiency (see OpenAI documentation).

3. Benchmarking study:

To comprehensively assess the capabilities of LLMs we chose to evaluate their proficiency in two distinct areas: understanding extracted text and providing descriptions for images using the same approach. This enables a more targeted and thorough analysis of the models' performance across different tasks.

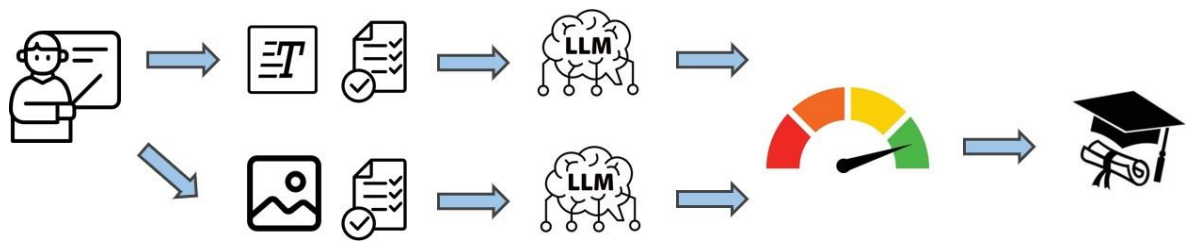

Figure 3.1 Evaluating LLMs: assessing their proficiency in understanding extracted text and generating image descriptions.

3.1. Creating Reference Questions and Answers (Test Development)

We constructed two specialized datasets, one for benchmarking the understanding of the extracted text and another one for benchmarking the LLMs' capability in providing useful descriptions for images. As depicted in Figure 3.3, these datasets serve as a standardized test for evaluating LLMs. This dataset comprises questions and corresponding "reference answers" generated through a structured process. First, Microsoft Copilot was used to produce questions and answers based on the provided text from a sample of documents comprising 23 core procedures documents. These initial outputs were then thoroughly reviewed and validated by human evaluators to ensure their quality and accuracy.

For benchmarking the descriptions of images, another dataset containing questions and answers about the images was entirely developed and validated by us without using AI. Indeed, when we attempted to use MS Copilot to generate questions and answers about images, it struggled to comprehend them due to the higher complexity of image data compared to text data.

3.2. LLM Evaluation

After preparing the datasets of questions and expected, or reference, answers, we assessed the LLM's ability to accurately interpret and answer questions. A specific prompt was designed to include the question and the corresponding extracted text, which contained the answer. This served as a context for the LLM.

For benchmarking the image understanding part, the text is the description of visual elements in the documents.

This setup, detailed in Figure 3.2, aimed to rigorously evaluate the model's contextual understanding and response accuracy.

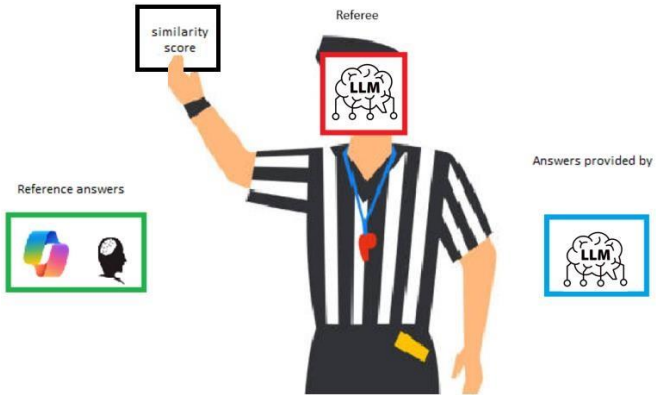

Figure 3.2 Evaluating LLMs using an LLM as a referee.

3.3. Comparative Analysis of Results Using LLM as a "Referee"

To assess the accuracy of the LLM's responses, we compared them with the reference answers using two metrics: a similarity index and the distance. The first comparison involved employing another LLM (GPT-3.5) as a referee, which assigned similarity scores on a scale from 0 to 1, based on the degree of alignment between the LLM-generated answers and the reference answers.

In addition to the referee LLM's scores, we established various KPIs to provide quantitative metrics for evaluating the models' performance. Initially, we calculated the Levenshtein distance, which determines the number of character-level edits needed to convert one response into another. While useful for identifying differences at a surface level, the Levenshtein distance does not capture the semantic content of the answers.

To have a better assessment of the response's semantics, we devised two metrics. For metric 1 we used the similarity scores generated by the LLM referee, as depicted in Figure 3.3. For metric 2 we computed the cosine similarity between the embeddings of the reference and LLM-generated answers. These metrics leverage the semantic encoding power of embeddings to offer a more meaningful evaluation by considering the conceptual alignment between texts, rather than just their surface similarity in terms of characters in the sentence.

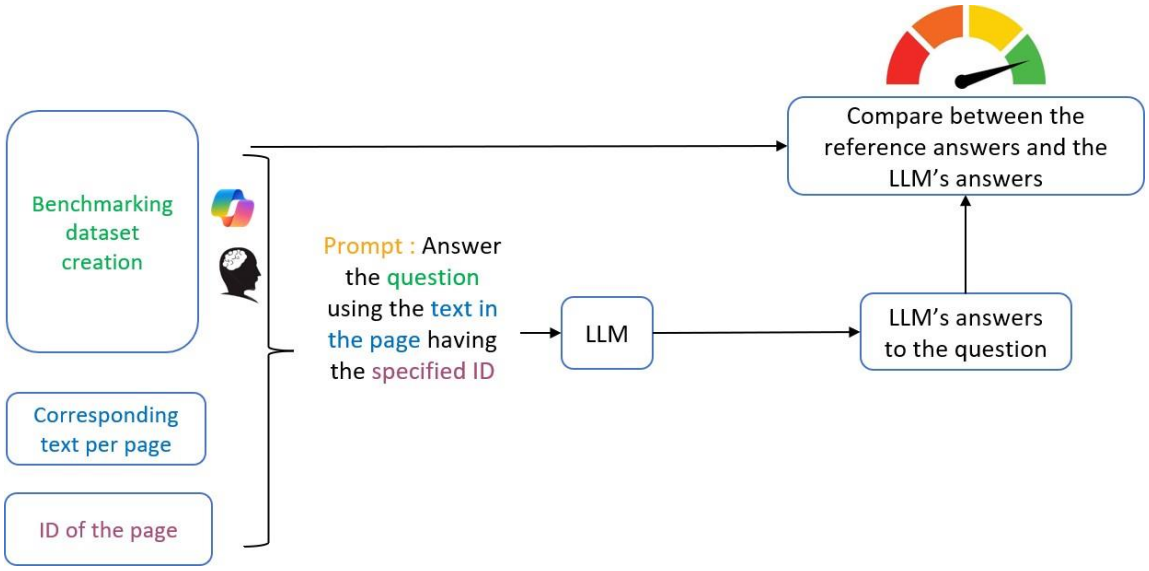

Figure 3.3: Evaluating LLMs: assessment setup for contextual understanding and response accuracy with extracted text and image descriptions.

Results and Discussion

The utilization of an LLM as a referee may seem unconventional, given the inherent variability in LLM responses. However, this automated evaluation approach significantly reduces the time and effort required for large-scale assessments, providing a practical solution to the laborious task of manual grading. The trade-off lies in balancing the variability of LLMs' responses with the need for efficiency, particularly when evaluating hundreds or thousands of responses. We maximized model consistency by setting the LLM temperature to zero.

The benchmarking process highlighted both the strengths and challenges associated with using LLMs for automated evaluations. While automated methods such as referee models enhance evaluation efficiency, they also introduce variability in scoring due to the dynamic nature of LLMs. Careful interpretation of similarity scores is essential, recognizing the delicate balance between the comfort of automated evaluations and the inherent complexity of LLM behavior.

Conclusion

The utilization of RAG-supported generative AI is now becoming an off-the-shelf solution to access information from core procedures documents and other company-specific knowledge bases, with the potential to enhance operational efficiency and comfort and improve overall productivity. Future work aims to extend the application of this solution to different knowledge bases within the organization and continuously update and benchmark the performance of new LLMs as they become available.

References

Sengar, S. S., Hasan, A. B., Kumar, S., & Carroll, F. (2024). *Generative artificial intelligence: A systematic review and applications*. arXiv:2405.11029 [cs.LG]. https://doi.org/10.48550/arXiv.2405.11029

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention is all you need*. In *Advances in Neural Information Processing Systems* (Vol. 30). Retrieved from https://arxiv.org/abs/1706.03762

Lewis, P., Oguz, B., Riedel, S., & Kwiatkowski, T. (2020). *Retrieval-augmented generation for knowledge-intensive NLP tasks*. arXiv. https://doi.org/10.48550/arXiv.2005.11401

Krishnan, A. (2023). *On the n-gram approximation of pre-trained language models*. arXiv. https://doi.org/10.48550/arXiv.2306.06892

OpenAI. (n.d.). *Models*. OpenAI. https://platform.openai.com/docs/models