Beyond Text: Optimizing RAG with Multimodal Inputs for Industrial Applications

Monica Riedler[1,2], Stefan Langer[1,2]

[1]Center for Information and Language Processing, LMU Munich,
[2]Siemens AG
monica.riedler@campus.lmu.de, stefan.langer@cis.lmu.de

Abstract

Large Language Models (LLMs) have demonstrated impressive capabilities in answering questions, but they lack domain-specific knowledge and are prone to hallucinations. Retrieval Augmented Generation (RAG) is one approach to address these challenges, while multimodal models are emerging as promising AI assistants for processing both text and images. In this paper we describe a series of experiments aimed at determining how to best integrate multimodal models into RAG systems for the industrial domain. The purpose of the experiments is to determine whether including images alongside text from documents within the industrial domain increases RAG performance and to find the optimal configuration for such a multimodal RAG system. Our experiments include two approaches for image processing and retrieval, as well as two LLMs (GPT4-Vision and LLaVA) for answer synthesis. These image processing strategies involve the use of multimodal embeddings and the generation of textual summaries from images. We evaluate our experiments with an LLM-as-a-Judge approach. Our results reveal that multimodal RAG can outperform single-modality RAG settings, although image retrieval poses a greater challenge than text retrieval. Additionally, leveraging textual summaries from images presents a more promising approach compared to the use of multimodal embeddings, providing more opportunities for future advancements.

1 Introduction

The release of Large Language Models (LLMs), such as Llama3 (Meta LLaMA Team, 2024) and GPT-4 (OpenAI, 2023a), has significantly advanced the field of Natural Language Processing (NLP), enabling a wide range of applications, including automated content generation and conversational agents. However, LLMs often still lack domain-specific knowledge and are prone to hallucinations (Kandpal et al., 2023; Rawte et al., 2023). Retrieval Augmented Generation (RAG) addresses these limitations by combining document retrieval with generative language models.

Recently, Multimodal Large Language Models (MLLMs) have emerged, extending LLM capabilities to include modalities like images and videos (Zhang et al., 2024; Yin et al., 2023). This development holds significant potential for industrial settings such as manufacturing, engineering, and maintenance, where documents like manuals, software guides, and product brochures frequently combine complex technical text with detailed visuals, such as diagrams, schematics and screenshots. This combination of modalities makes the industrial domain particularly challenging for AI systems, as they must accurately interpret both textual and visual information to provide meaningful insights.

While extensive research has been conducted on text-only RAG systems and their optimization (Gao et al., 2023; Siriwardhana et al., 2023), the application of multimodal RAG to the industrial domain is less documented in academic literature. Existing examples mainly target general-domain datasets (Chen et al., 2022; Lin and Byrne, 2022) and medical applications (Sun et al., 2024; Xia et al., 2024; Zhu et al., 2024).

In our paper, we explore the integration of multimodal models into RAG systems for the industrial domain. Specifically, we investigate whether incorporating images alongside text enhances RAG performance and we identify optimal configurations for such systems. We use two MLLMs, GPT-4Vision (OpenAI, 2023b) and LLaVA (Liu et al., 2024), for answer synthesis and evaluate two image processing strategies: multimodal embeddings and textual summaries from images.

Our research addresses two primary questions: (1) Does the inclusion of both text and images improve the performance of RAG systems in the industrial domain compared to single-modality RAG? (2) How can the performance of multimodal

RAG be optimized for this domain? To answer these questions, we compare the performance of single-modality (text-only or image-only) and multimodal (text and image) RAG systems using a set of 100 domain-specific questions. Additionally, we explore various retrieval methods to optimize the performance of the multimodal RAG pipelines.

Our paper makes the following contributions:

- We integrate multimodal models into RAG systems for the industrial domain, demonstrating that multimodal RAG can outperform single-modality RAG.
- We compare multimodal embeddings and image summaries for image processing, employing GPT-4V and LLaVA for answer synthesis, and find that image summaries offer greater flexibility and potential for advancement.

2 Related Work

Multimodal LLMs In recent years, LLMs have shown emergent abilities such as in-context learning, instruction following, and chain-of-thought reasoning (Brown et al., 2020; Ouyang et al., 2022; Wei et al., 2024), making them suitable for various NLP tasks. Multimodal LLMs extend these capabilities to understand and generate multiple modalities, enabling AI assistants to process text, images, videos, and audio. Notable examples include the integration of pre-trained unimodal models into multimodal systems, which consist of a modality encoder, a pre-trained LLM, and a modality generator, connected by projectors to transform inputs and outputs across different modalities (Zhang et al., 2024; Yin et al., 2023).

Retrieval Augmented Generation RAG has emerged as an effective approach to address the limitations of LLMs in domain-specific question answering. By combining an LLM for answer generation with an external vector database accessed via a retriever, RAG has been effectively applied to various tasks, including question answering, fact verification, and question generation, achieving state-of-the-art results in open-domain question answering (Lewis et al., 2020; Guu et al., 2020; Izacard and Grave, 2021; Borgeaud et al., 2022). Key optimizations to the initial approaches are extensively described by Gao et al. (2023). Despite significant advancements, challenges such as retrieval quality (Ma et al., 2023; Carpineto and Romano, 2012), the reliability of the generation component (Wu et al., 2024; Niu et al., 2024), and RAG robustness remain active areas of research (Cuconasu et al., 2024).

Multimodal RAG Chen et al. (2022) introduced MuRAG, the first multimodal Retrieval-Augmented Transformer, which enhances model capabilities using an external non-parametric multimodal memory. An alternative approach by Lin and Byrne (2022) involves transforming images into textual representations through OCR, image captioning, and object detection, followed by dense passage retrieval (Karpukhin et al., 2020). Other works apply multimodal RAG scenarios to medical and healthcare applications (Sun et al., 2024; Xia et al., 2024; Zhu et al., 2024), highlighting the potential of leveraging images as additional context.

RAG Evaluation Evaluating RAG systems involves assessing both retrieval and generation components. Frameworks like RAGAs (Es et al., 2024) incorporate metrics such as Faithfulness, Answer Relevance, and Context Relevance to holistically assess RAG performance. Automated benchmarking methods, using datasets like TruthfulQA (Lin et al., 2022) and MMLU (Hendrycks et al., 2021), also evaluate specific RAG capabilities. For our experiments, we adopt an LLM-as-a-Judge approach (Chiang et al., 2024), where LLMs evaluate their own generated outputs. While scalable and often aligning with human evaluations, automated methods remain approximations. Human annotations are the gold standard for accuracy, especially in domain-specific tasks, but are costly and time-intensive, making automated methods a practical alternative. For multimodal evaluation, Zhang et al. (2023) showed GPT-4V's effectiveness in vision-language tasks, aligning well with human assessments.

3 Approach

3.1 Data

Due to the lack of annotated context-question-answer triplets in the industrial domain, and even more so for a multimodal setting requiring quadruples of text context, image context, question, and answer, we created a manually annotated dataset[1].

We used 20 PDF documents from the industrial domain, such as manuals and software documentation for devices like programmable controllers,

[1] Unfortunately we cannot release this dataset due to copyright concerns.

circuit breakers, and robots. Text and images were extracted from these documents, resulting in 8540 text chunks (with an average length of 225 words per chunk) and 8377 images, aligned by page to maintain contextual accuracy.

To create the RAG test set, we manually annotated 100 question-answer pairs. Each annotation includes a question, reference answer, and page number used to retrieve both text and image context from the corresponding page, forming multimodal quadruples. The questions were designed to address typical industrial tasks, such as operational procedures, device configurations, and troubleshooting, where visual context plays a key role. The annotation process involved two annotators: one with basic domain knowledge and the other with extensive industrial experience. An example quadruple, along with further details on the extraction and annotation, are provided in Appendix C.

3.2 Experiments

In this section, we outline the experiments we conducted to investigate two primary questions: (1) Does using both text and image modalities improve performance? (2) What is the optimal configuration for a multimodal RAG system in the industrial domain? We categorize the experiments into three RAG settings: Text-Only RAG, Image-Only RAG, and Multimodal RAG. Additionally, we implemented two reference settings for comparison: (1) a Baseline, where we feed questions directly to an LLM without retrieval, and (2) a Gold Standard Context setting, which serves as an upper bound. We used a prompt to instruct the model to answer based on the retrieved context (Figure 3 in Appendix). To ensure consistency across experiments, we ran all settings with both GPT-4V and LLaVA, including the text-only settings, which do not require multimodal content processing. We report implementation details on vector databases, retrieval, model versions, and hyperparameters in Appendix A.

3.2.1 Baseline

Our baseline consists in feeding questions from the test set directly to the LLM, without any retrieval step. This allowed us to test the LLM's internal knowledge and gain insight into its performance on domain-specific industrial questions.

3.2.2 Text-Only RAG

In the Text-Only RAG setting, we used only the texts extracted from the PDF collection. We embedded the text chunks using OpenAI's text-embedding-3-small[2] model and stored them in a vector store. We then performed a similarity search on the vector store for each embedded question to retrieve the most relevant texts, which were concatenated with the user query and passed to the multimodal LLM for answer generation. This setup allowed us to evaluate the performance of text-based retrieval and answer synthesis.

3.2.3 Image-Only RAG

In this setting, we only used images from the PDF documents. For image retrieval, we explored two distinct approaches:

Multimodal Embeddings We used CLIP (Radford et al., 2021) to jointly embed both images and questions. CLIP was selected for its ability to align image and text modalities in a shared embedding space, which allows to easily compute similarities between different types of data. This alignment is crucial for multimodal retrieval tasks, where understanding the relationship between image content and textual queries is key. We stored the obtained embeddings in a vector store, and performed a similarity search to retrieve the most relevant images based on the embedded query.

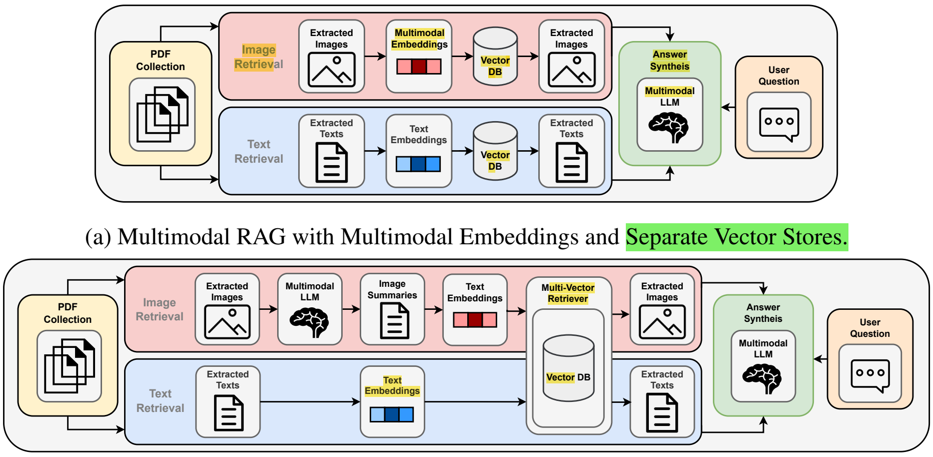
Text Embeddings From Image Summaries We summarized the images into text using a multimodal LLM (see Figure 4 in Appendix) and then embedded these summaries using text-embedding-3-small. We employed LangChain's Multi-Vector Retriever[3], which allows to decouple the retrieval and generation sources. Summaries were stored in a vector store, while the original images were stored in a document store, allowing retrieval through textual summaries while preserving the original images for answer generation to reduce potential information loss.

3.2.4 Multimodal RAG

Figure 1 provides an overview of our multimodal RAG approaches, where we combined text and image modalities. We reused the two image retrieval methods from the Image-Only RAG setting,

[2] https://platform.openai.com/docs/guides/embeddings

[3] https://python.langchain.com/v0.2/docs/how_to/multi_vector/



(a) Multimodal RAG with Multimodal Embeddings and Separate Vector Stores.

(b) Multimodal RAG with Image Summaries and Combined Vector Store.

Figure 1: Overall architecture of our proposed multimodal RAG pipelines. For the Text-Only RAG, we only use the Text Retrieval component. Conversely, in the Image-Only RAG, we only employ the Image Retrieval component, either with multimodal embeddings or image summaries.

adding text retrieval. We explored two configurations within this setup:

Multimodal Embeddings and Separate Vector Stores We embedded images using CLIP and texts using text-embedding-3-small, storing them in separate vector stores. We embedded the query for both modalities and performed a separate similarity search in each store, ensuring both text and image results are retrieved.

Image Summaries and Combined Vector Store We converted images into text summaries and embedded these, along with text chunks extracted from the PDF documents, using text-embedding-3-small. Both were stored in a single vector store. A similarity search was then performed to retrieve the most relevant documents, whether text or image, based on the query embedding.

3.2.5 Gold Standard Context Prompting

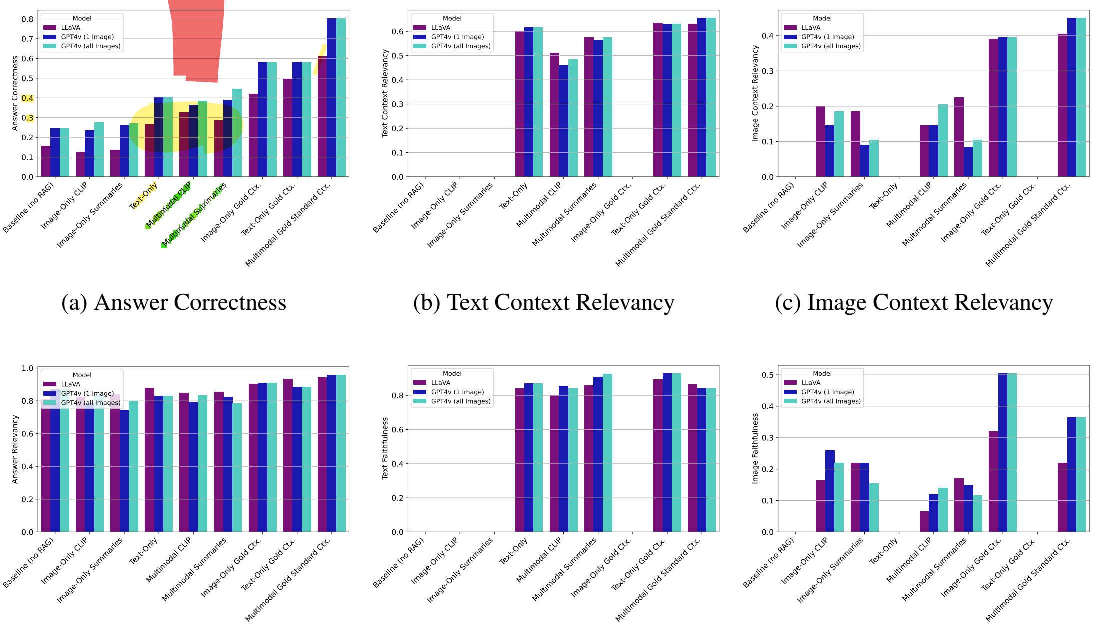
In the Gold Standard Context setting, we directly provided the annotated context from the test set to the LLM along with the question, skipping the retrieval step. This setup serves as an upper bound, demonstrating the performance achievable with perfect retrieval and enabling a direct comparison between the generating models (GPT-4V and LLaVA) independently of retrieval performance.

4 Evaluation Framework

We evaluate the performance of the RAG pipelines using an LLM-as-a-Judge approach (Chiang et al., 2024; Zhang et al., 2023), developing a custom evaluation framework tailored to multimodal data. This framework incorporates metrics similar to those in existing text-only RAG evaluation frameworks, such as RAGAs (Es et al., 2024). However, our framework enables evaluating multimodal content to handle both text and images, ensuring a comprehensive assessment of the RAG system's performance. We make the code for all experiments and the evaluation framework available at the following URL: `https://github.com/riedlerm/multimodal_rag_for_industry`.

The framework is designed to be modular and can be used with multiple models as evaluators, including GPT-4V and LLaVA. The core of the framework consists of an evaluation module that includes specialized evaluators for each metric. These evaluators construct prompts for the model, execute the evaluation, and parse the model's output to ensure it meets the required format. The delivered output for each metric includes a binary grade (either 1 or 0) and a reason for the judgment to facilitate easy aggregation and analysis of the results. The final score for each metric is obtained by averaging all binary evaluations over the dataset.

Evaluation Metrics The framework employs six evaluation metrics: **Answer Correctness** uses reference-guided pairwise comparison to evaluate the correctness of the generated answer compared to a reference answer and is the only metric relying on the presence of ground truth answers; **Answer Relevancy** assesses whether the generated answer is relevant to the question; **Text Faithfulness** meaeven upper bounds are very low



(a) Answer Correctness

(b) Text Context Relevancy

(c) Image Context Relevancy

(d) Answer Relevancy

(e) Text Faithfulness

(f) Image Faithfulness

Figure 2: RAG evaluation results for GPT-4V prompted with either a single or multiple images, and LLaVA (always single image) across six metrics. The results show the performance of each RAG setting in generating accurate, relevant, and faithful responses based on both text and image inputs.

sures the consistency between the generated answer and the retrieved textual context; **Image Faithfulness** evaluates whether the generated answer aligns with the content of the retrieved images; **Text Context Relevancy** evaluates the relevancy of the retrieved text context in answering the question; and **Image Context Relevancy**, assesses the relevancy of the retrieved images to the question. We summarize these metrics in Table 1 and report the prompts to calculate each metric in Appendix B.2.

Metric	Required Inputs
Ans. Correctness	Q, GA, RA
Ans. Relevancy	Q, GA
Text Faithfulness	GA, Text Ctx.
Img. Faithfulness	GA, Img. Ctx.
Text Ctx. Relevancy	Q, Text Ctx.
Img. Ctx. Relevancy	Q, Img. Ctx.

Table 1: Evaluation metrics and their required inputs: Q is the question, GA is the generated answer, RA is the reference answer, and Text/Img. Ctx. are the context provided as text/image respectively.

5 Results

We summarize the performance of GPT-4V and LLaVA in 9 different settings including single-modality and multimodal RAG approaches in Figure 2. Unlike LLaVA, which was limited to processing a single image per prompt during our experiments[4], GPT-4V has the capability to handle multiple images and interleaved text-image sequences. We assessed its performance both using one image, for comparison with LLaVA, and multiple images to inspect the effect of using multiple images as context. In settings utilizing image summaries for retrieval, the summarizing model (LLaVA or GPT-4V) is consistently used for answer synthesis. Both models, i.e., LLaVA and GPT-4V, are used for evaluation. This means that the final score is derived from averaging evaluations conducted with both models, regardless of the generator model chosen. This approach helps mitigate self-enhancement bias (Chiang et al., 2024) and avoids single-judge evaluations. We report the full results of our experiments in Appendix D.

5.1 Single-Modality vs. Multimodal RAG

In Figure 2a, we show the Answer Correctness for single-modality and multimodal settings to investigate whether a combination of text and images improves performance. The upper bound results, obtained by prompting with the gold standard context, reveal that using both text and images sig-

[4] For our experiments we employed `llava-hf/llava-v1.6-mistral-7b-hf`

nificantly outperforms single-modality approaches. This suggests that integrating images with text is beneficial for this domain. In RAG settings, results show that multimodal RAG can outperform single-modality approaches. Image-only RAG tends to yield the lowest scores, only slightly outperforming the baseline, highlighting the need for improved image retrieval mechanisms. Conversely, multimodal RAG using image summaries slightly outperforms text-only RAG, although the difference is smaller compared to the gold standard context setting. The multimodal setting using CLIP embeddings shows mixed results: LLaVA performs better with multimodal inputs, while GPT-4V performs better with text-only RAG. Overall, while multimodal RAG offers an advantage over text-only RAG, particularly in scenarios with effective text and image retrieval, image retrieval still requires further improvement.

5.2 Prompting with Gold Standard Context

When provided with the gold standard context, using both text and images yields significantly higher Answer Correctness scores compared to single-modality approaches (Figure 2a). Around 60% of the questions can be answered using a single modality, given the correct context. However, using both modalities increases the Answer Correctness to approximately 80%, suggesting that a combination of text and image is often required for a correct answer. Despite these gains, image retrieval appears more challenging compared to text retrieval, as evidenced by the pronounced gap between the single-modality gold context and their respective single-modality RAG settings.

5.3 Prompting with Multiple Images

Employing multiple images in prompts generally improves performance across all metrics, except for Image Faithfulness, in both Image-Only and Multimodal RAG settings. This enhancement suggests that additional images increase the chance of incorporating the relevant context in the prompt, thereby improving Answer Correctness and Relevancy. However, the model's tendency to focus on one image might explain lower Image Faithfulness scores. Overall, the results highlight the benefits of processing multiple images within a single prompt.

5.4 GPT-4V vs. LLaVA

In our experiments GPT-4V consistently outperforms LLaVA in terms of Answer Correctness, often by a significant margin. The same trend is observed for Text Faithfulness, albeit less pronounced. The image-related metrics present a mixed picture. While some data points show LLaVA substantially surpassing GPT-4V in Image Context Relevancy, other results suggest the opposite trend. In terms of Answer Relevancy, LLaVA slightly outperforms GPT-4V in most settings.

5.5 Multimodal Embeddings vs. Image Summaries

In both Image-Only and Multimodal RAG settings, our two image processing strategies show comparable performance. However, the image summaries setup slightly outperforms multimodal embeddings across most metrics, except for Image Context Relevancy. The image summaries approach appears to be more promising, as it offers greater potential for future advancements. For instance, the summarization prompt can be tailored to focus on specific image aspects and include few-shot examples, options not possible with multimodal embeddings. Additionally, the summarization model can be further optimized with task-specific models, and there are more choices for text embedding models compared to multimodal ones. In contrast, the multimodal embedding approach relies heavily on the quality of the embedding model, limiting its potential for improvement.

6 Conclusion

Our research demonstrates the potential of integrating multimodal models into RAG systems for the industrial domain. By incorporating both text and images, we observed significant improvements in performance, particularly when the retrieval process successfully identifies relevant texts and images. Leveraging textual summaries from images provides greater flexibility and optimization opportunities compared to multimodal embeddings. Despite the challenges associated with image retrieval, our findings underscore the importance of multimodal data in enhancing the quality of generated answers. Future work will focus on refining image retrieval, comparing our results with fine-tuning-based approaches, and combining RAG with multimodal LLMs fine-tuned for the industrial domain. Additionally, we plan to evaluate each pipeline step independently on domain-specific data to better identify potential failure points, especially in image retrieval, even though our current analysis focused on end-to-end metrics.

7 Limitations

While our study demonstrates promising results, several areas require further investigation. First, the lack of publicly available, domain-specific datasets restricts the reproducibility and generalizability of our findings, highlighting the need for future work to develop such resources.

While GPT-4V and LLaVA were effective, they share common LLM limitations, including inaccuracies, hallucinations, and difficulty handling complex multimodal inputs. Additionally, our evaluation relies on LLMs, which may introduce biases. While we mitigated this by using both GPT-4V and LLaVA, human evaluations remain essential for a reliable assessment. New multimodal models, such as GPT-4o (OpenAI, 2024), emerged during our study but could not be included. Continuous evaluations with the latest models are necessary.

Although we focused on the industrial domain, we believe our approach can be applied to other domains as well, since no domain-specific components were used at any stage of our pipeline. Additional experiments can help confirm the applicability of our findings in other fields, such as healthcare or finance.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1).

Harrison Chase. 2022. Langchain. `https://github.com/langchain-ai/langchain`. Accessed: September 09, 2024.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. 2022. MuRAG: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5558–5570, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. 2022. Reproducible scaling laws for contrastive language-image learning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2829.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *Preprint*, arXiv:2403.04132.

Chroma. 2022. Chroma: the open-source embedding database. `https://github.com/chroma-core/chroma`. Accessed: September 09, 2024.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 719–729, New York, NY, USA. Association for Computing Machinery.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the*

37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Weizhe Lin and Bill Byrne. 2022. Retrieval augmented visual question answering with outside knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11238–11254, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Yu A. Malkov and D. A. Yashunin. 2020. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(4):824–836.

Meta LLaMA Team. 2024. Introducing meta llama 3: The most capable openly available llm to date. `https://ai.meta.com/blog/meta-llama-3/`. Accessed: 2024-06-30.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI. 2023a. GPT-4 technical report.

OpenAI. 2023b. Gpt-4v(ision) system card.

OpenAI. 2024. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`. Accessed: September 22, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. 2023. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2541–2573, Singapore. Association for Computational Linguistics.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Liwen Sun, James Zhao, Megan Han, and Chenyan Xiong. 2024. Fact-aware multimodal retrieval augmentation for accurate medical radiology report generation. *ArXiv*, abs/2407.15268.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Kevin Wu, Eric Wu, and James Zou. 2024. Clasheval: Quantifying the tug-of-war between an llm's internal prior and external evidence. *Preprint*, arXiv:2404.10198.

Peng Xia, Kangyu Zhu, Haoran Li, Hongtu Zhu, Yun Li, Gang Li, Linjun Zhang, and Huaxiu Yao. 2024. Rule: Reliable multimodal rag for factuality in medical vision language models. *ArXiv*, abs/2407.05131.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *ArXiv*, abs/2306.13549.

Duzhen Zhang, Yahan Yu, Jiahua Dong, Chenxing Li, Dan Su, Chenhui Chu, and Dong Yu. 2024. MM-LLMs: Recent advances in MultiModal large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 12401–12430, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Xinlu Zhang, Yujie Lu, Weizhi Wang, An Yan, Jun Yan, Lianke Qin, Heng Wang, Xifeng Yan, William Yang Wang, and Linda Ruth Petzold. 2023. Gpt-4v(ision) as a generalist evaluator for vision-language tasks. *ArXiv*, abs/2311.01361.

Yinghao Zhu, Changyu Ren, Zixiang Wang, Xiaochen Zheng, Shiyun Xie, Junlan Feng, Xi Zhu, Zhoujun Li, Liantao Ma, and Chengwei Pan. 2024. Emerge: Integrating rag for improved multimodal ehr predictive modeling. *ArXiv*, abs/2406.00036.

A Implementation Details

A.1 Vector Databases and Retrievers

For our RAG pipelines, we chose ChromaDB (Chroma, 2022) as our vector database because of its open-source nature, local usability, and seamless integration with LangChain (Chase, 2022), which we used as the framework for our experiments. We employed the Hierarchical Navigable Small World (HNSW) search method (Malkov and Yashunin, 2020) along with L2 similarity search.

Embeddings were generated using either `CLIP ViT-L/14` via the OpenCLIP implementation (Cherti et al., 2022) for settings with multimodal embeddings, or OpenAI's `text-embedding-3-small`[5] for text and image summaries.

For retrieval, we employed LangChain's VectorStoreRetriever for CLIP embeddings and the MultiVectorRetriever for text and image summaries embedded with `text-embedding-3-small`.

A.2 Number of Retrieved Documents

We maintained the parameter k, representing the number of documents retrieved during the pipeline's retrieval step, at LangChain's default value of 4 across all experiments to ensure consistency. In settings with separate vector stores and retrievers for text and image embeddings, k was set to 2 for each modality, ensuring a total of 4 retrieved documents. Consequently, in these settings, the top 2 images and the top 2 texts were always retrieved. Conversely, in settings with a single vector store for both texts and textual summaries, the distribution of retrieved texts and summaries varied depending on the relevance of the documents. This variation implies that if the top 4 retrieved documents were all texts, there could be instances where no images were retrieved, and vice versa. We acknowledge that this approach might not be ideal, as the fixed value of k may not always capture the most relevant documents, especially in cases where the relevance distribution between text and images is uneven. However, we adopted this strategy to maintain uniformity and simplicity across all experiments, thereby facilitating a controlled comparison of the retrieval mechanisms.

A.3 Model Versions and Hyperparameters

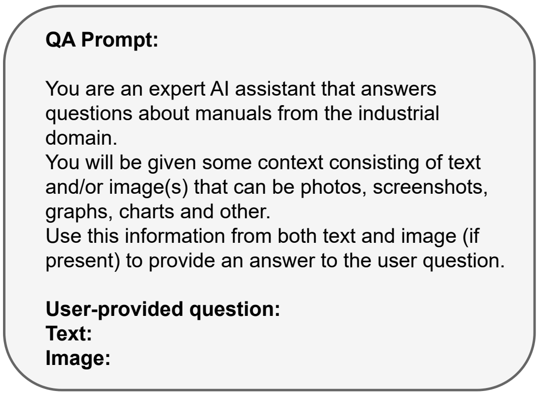
Configuration	**GPT-4V**	**LLaVA**
version	2024-02-15-preview	llava-v1.6-mistral-7b
temperature	0.7	1
top_p	0.95	1
max_tokens	300	300

Table 2: Model Versions and Hyperparameters

[5] https://platform.openai.com/docs/guides/embeddings

B Prompt Templates

B.1 RAG Prompt Templates



QA Prompt:

You are an expert AI assistant that answers questions about manuals from the industrial domain.
You will be given some context consisting of text and/or image(s) that can be photos, screenshots, graphs, charts and other.
Use this information from both text and image (if present) to provide an answer to the user question.

User-provided question:
Text:
Image:

Figure 3: Question Answering Prompt.

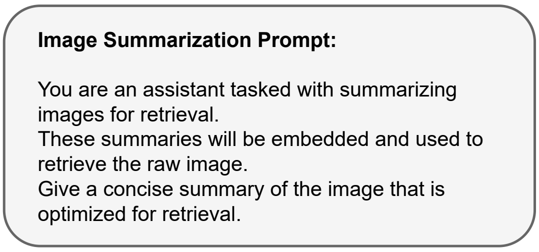
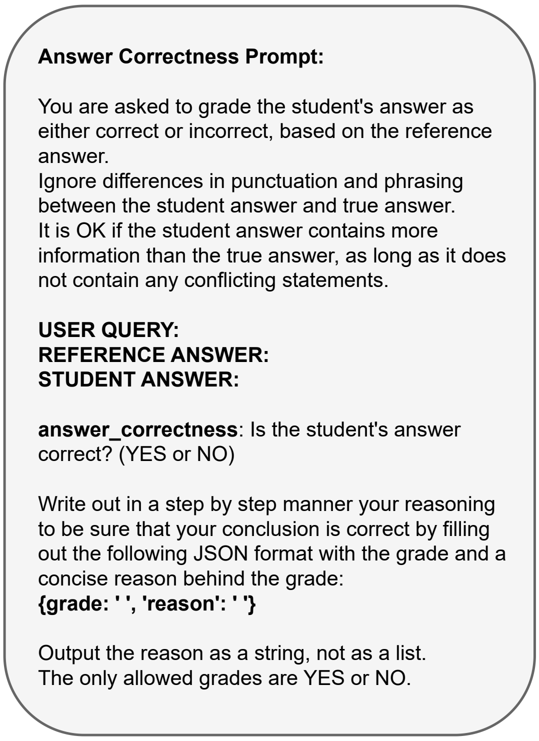


Image Summarization Prompt:

You are an assistant tasked with summarizing images for retrieval.
These summaries will be embedded and used to retrieve the raw image.
Give a concise summary of the image that is optimized for retrieval.

Figure 4: Image Summarization Prompt.



Answer Correctness Prompt:

You are asked to grade the student's answer as either correct or incorrect, based on the reference answer.
Ignore differences in punctuation and phrasing between the student answer and true answer.
It is OK if the student answer contains more information than the true answer, as long as it does not contain any conflicting statements.

USER QUERY:
REFERENCE ANSWER:
STUDENT ANSWER:

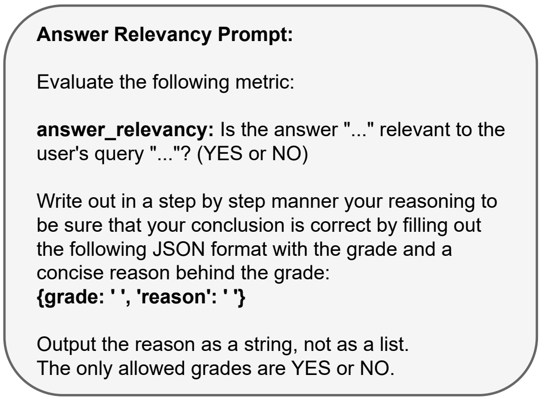
answer_correctness: Is the student's answer correct? (YES or NO)

Write out in a step by step manner your reasoning to be sure that your conclusion is correct by filling out the following JSON format with the grade and a concise reason behind the grade:
{grade: ' ', 'reason': ' '}

Output the reason as a string, not as a list.
The only allowed grades are YES or NO.

Figure 5: Answer Correctness Evaluation Prompt.

B.2 Evaluation Prompt Templates



Answer Relevancy Prompt:

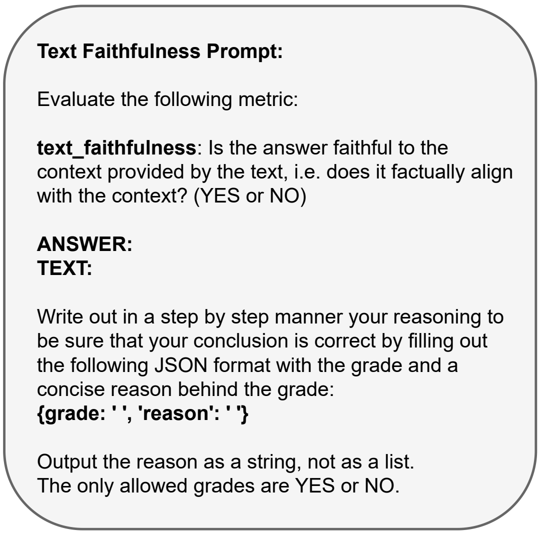
Evaluate the following metric:

answer_relevancy: Is the answer "..." relevant to the user's query "..."? (YES or NO)

Write out in a step by step manner your reasoning to be sure that your conclusion is correct by filling out the following JSON format with the grade and a concise reason behind the grade:
{grade: ' ', 'reason': ' '}

Output the reason as a string, not as a list.
The only allowed grades are YES or NO.

Figure 6: Answer Relevancy Evaluation Prompt.



Text Faithfulness Prompt:

Evaluate the following metric:

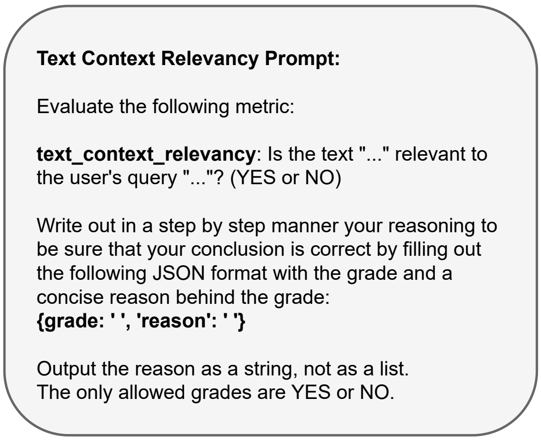
text_faithfulness: Is the answer faithful to the context provided by the text, i.e. does it factually align with the context? (YES or NO)

ANSWER:
TEXT:

Write out in a step by step manner your reasoning to be sure that your conclusion is correct by filling out the following JSON format with the grade and a concise reason behind the grade:
{grade: ' ', 'reason': ' '}

Output the reason as a string, not as a list.
The only allowed grades are YES or NO.

Figure 7: Text Faithfulness Evaluation Prompt.



Text Context Relevancy Prompt:

Evaluate the following metric:

text_context_relevancy: Is the text "..." relevant to the user's query "..."? (YES or NO)

Write out in a step by step manner your reasoning to be sure that your conclusion is correct by filling out the following JSON format with the grade and a concise reason behind the grade:
{grade: ' ', 'reason': ' '}

Output the reason as a string, not as a list.
The only allowed grades are YES or NO.

Figure 8: Text Context Relevancy Evaluation Prompt.

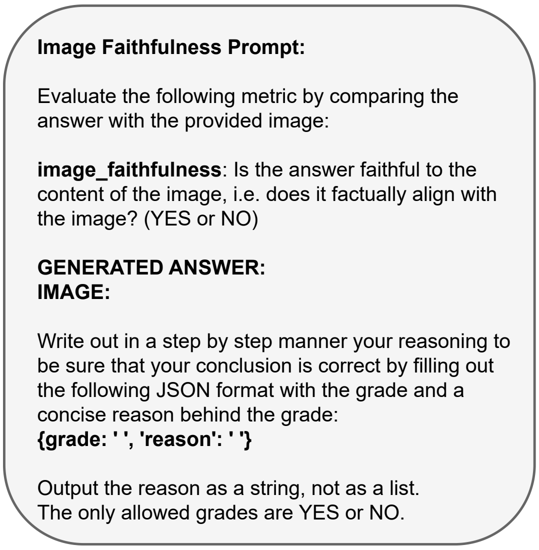


Image Faithfulness Prompt:

Evaluate the following metric by comparing the answer with the provided image:

image_faithfulness: Is the answer faithful to the content of the image, i.e. does it factually align with the image? (YES or NO)

GENERATED ANSWER:
IMAGE:

Write out in a step by step manner your reasoning to be sure that your conclusion is correct by filling out the following JSON format with the grade and a concise reason behind the grade:
{grade: ' ', 'reason': ' '}

Output the reason as a string, not as a list.
The only allowed grades are YES or NO.

Figure 9: Image Faithfulness Evaluation Prompt.

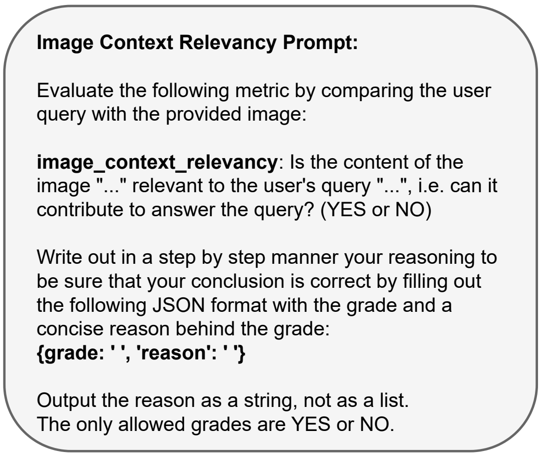


Image Context Relevancy Prompt:

Evaluate the following metric by comparing the user query with the provided image:

image_context_relevancy: Is the content of the image "..." relevant to the user's query "...", i.e. can it contribute to answer the query? (YES or NO)

Write out in a step by step manner your reasoning to be sure that your conclusion is correct by filling out the following JSON format with the grade and a concise reason behind the grade:
{grade: ' ', 'reason': ' '}

Output the reason as a string, not as a list.
The only allowed grades are YES or NO.

Figure 10: Image Context Relevancy Evaluation Prompt.

C Dataset Details

C.1 Text and Image Extraction

We used the PyMuPDF[6] library to extract text and images from industrial domain PDFs, creating a structured dataset stored in a parquet file. Each entry represents a page, with text and corresponding images aligned by page. If a page contained multiple images, each image was stored as a separate entry, along with the text of the page.

C.2 Question and Answer Annotation

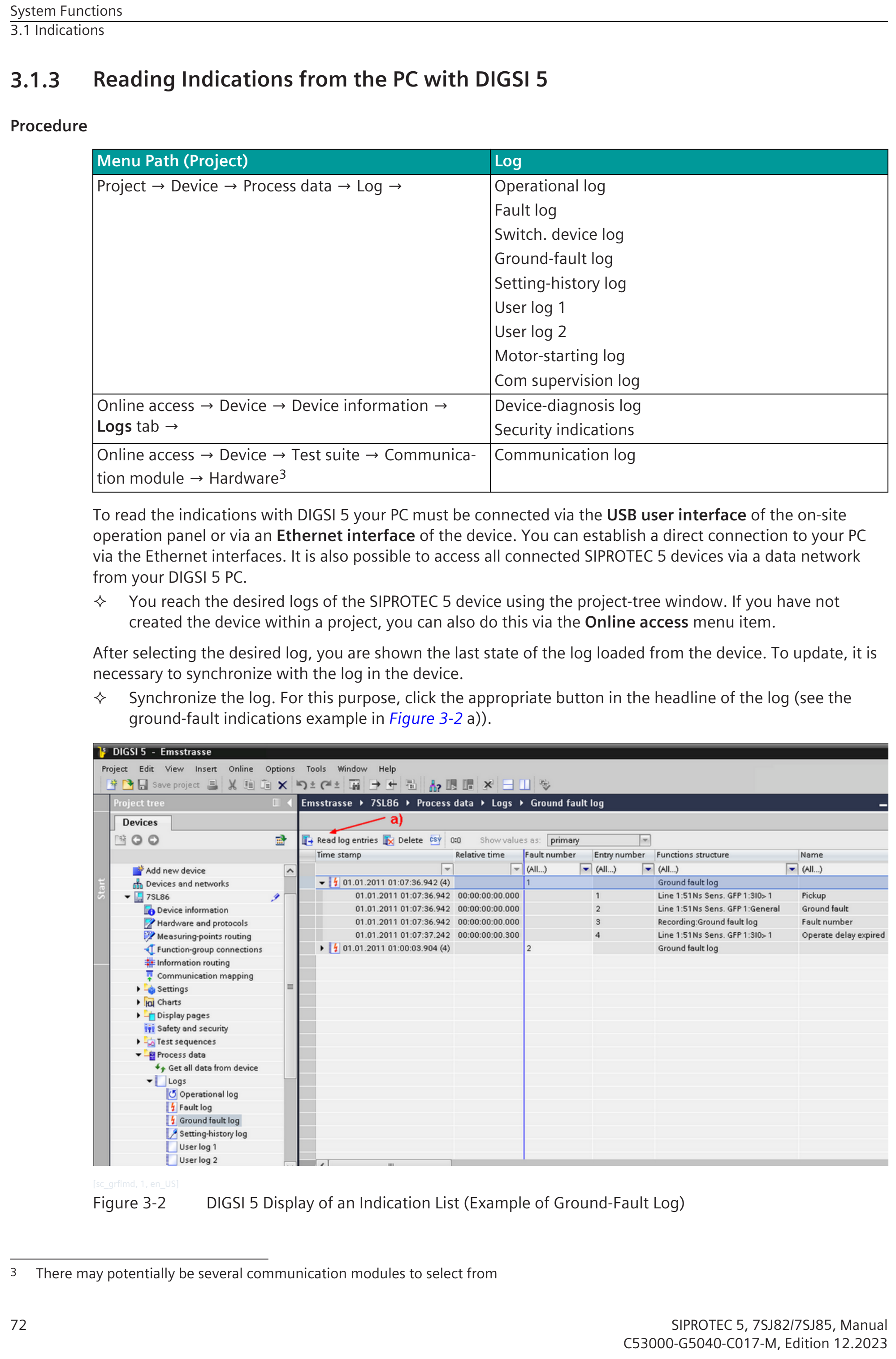
To generate the RAG test set, we manually annotated 100 question-answer pairs. The questions were designed to reflect typical queries in the industrial domain, such as operational procedures, device configurations, or troubleshooting guidance. These questions were inspired by existing text-only question-context-answer triples from industrial copilot systems, AI assistants that support factory personnel with automation and diagnostics. We introduced the multimodal aspect, incorporating image context to reflect the visual nature of the documents.

The annotation process involved two annotators: one with a base level of understanding of the domain and the other with extensive experience, having worked for many years in the industrial sector, particularly in software development. The annotators used the industrial PDFs as a starting point, manually creating a question based on the content of the document, along with the corresponding answer. For each annotated pair, the annotators also recorded the page number, which was used to extract both the relevant text and image context from the document, forming the quadruples.

C.3 Example from the Dataset

In this section, we present an example from our document collection and RAG test set. First, we show a page from one of the industrial manuals (Figure 11), followed by the corresponding annotated quadruple, consisting of a question, an answer, the extracted text, and the extracted image (Table 3).

[6] https://github.com/pymupdf/PyMuPDF



System Functions
3.1 Indications

3.1.3 Reading Indications from the PC with DIGSI 5

Procedure

Menu Path (Project)	Log
Project → Device → Process data → Log →	Operational log Fault log Switch. device log Ground-fault log Setting-history log User log 1 User log 2 Motor-starting log Com supervision log
Online access → Device → Device information → **Logs** tab →	Device-diagnosis log Security indications
Online access → Device → Test suite → Communication module → Hardware[3]	Communication log

To read the indications with DIGSI 5 your PC must be connected via the **USB user interface** of the on-site operation panel or via an **Ethernet interface** of the device. You can establish a direct connection to your PC via the Ethernet interfaces. It is also possible to access all connected SIPROTEC 5 devices via a data network from your DIGSI 5 PC.

✧ You reach the desired logs of the SIPROTEC 5 device using the project-tree window. If you have not created the device within a project, you can also do this via the **Online access** menu item.

After selecting the desired log, you are shown the last state of the log loaded from the device. To update, it is necessary to synchronize with the log in the device.

✧ Synchronize the log. For this purpose, click the appropriate button in the headline of the log (see the ground-fault indications example in *Figure 3-2* a)).

DIGSI 5 - Emsstrasse

Emsstrasse ▸ 7SL86 ▸ Process data ▸ Logs ▸ Ground fault log

Show values as: primary

Time stamp	Relative time	Fault number	Entry number	Functions structure	Name
		(All...)	(All...)	(All...)	(All...)
01.01.2011 01:07:36.942 (4)		1		Ground fault log	
01.01.2011 01:07:36.942	00:00:00:00.000		1	Line 1:51Ns Sens. GFP 1:3I0> 1	Pickup
01.01.2011 01:07:36.942	00:00:00:00.000		2	Line 1:51Ns Sens. GFP 1:General	Ground fault
01.01.2011 01:07:36.942	00:00:00:00.000		3	Recording:Ground fault log	Fault number
01.01.2011 01:07:37.242	00:00:00:00.300		4	Line 1:51Ns Sens. GFP 1:3I0> 1	Operate delay expired
01.01.2011 01:00:03.904 (4)		2		Ground fault log	

[sc_grflmd, 1, en_US]

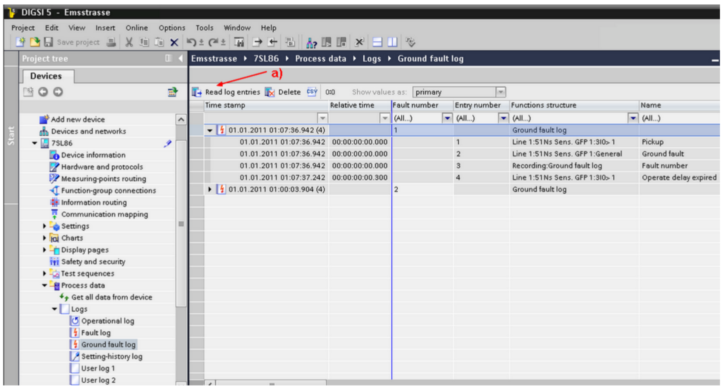
Figure 3-2 DIGSI 5 Display of an Indication List (Example of Ground-Fault Log)

[3] There may potentially be several communication modules to select from

72

SIPROTEC 5, 7SJ82/7SJ85, Manual
C53000-G5040-C017-M, Edition 12.2023

Figure 11: Example page of a PDF file from our document collection.

Question	How can I synchronize the log to read the indications with DIGSI 5?
Answer	To synchronize the log, click the button ‘Read log entries’ in the headline of the log.
Text Context	System Functions 3.1 Indications 3.1.3 Reading Indications from the PC with DIGSI 5 Procedure Menu Path (Project) Log Project → Device → Process data → Log → Operational log Fault log Switch. device log Ground-fault log Setting-history log User log 1 User log 2 Motor-starting log Com supervision log Online access → Device → Device information → Logs tab → Device-diagnosis log Security indications Online access → Device → Test suite → Communication module → Hardware Communication log To read the indications with DIGSI 5 your PC must be connected via the USB user interface of the on-site operation panel or via an Ethernet interface of the device. You can establish a direct connection to your PC via the Ethernet interfaces. It is also possible to access all connected SIPROTEC 5 devices via a data network from your DIGSI 5 PC. You reach the desired logs of the SIPROTEC 5 device using the project-tree window. If you have not created the device within a project, you can also do this via the Online access menu item. After selecting the desired log, you are shown the last state of the log loaded from the device. To update, it is necessary to synchronize with the log in the device. Synchronize the log. For this purpose, click the appropriate button in the headline of the log (see the ground-fault indications example in Figure 3-2 a)). Figure 3-2 DIGSI 5 Display of an Indication List (Example of Ground-Fault Log) There may potentially be several communication modules to select from SIPROTEC 5, 7SJ82/7SJ85, Manual C53000-G5040-C017-M, Edition 12.2023
Image Context	

Time stamp	Relative time	Fault number	Entry number	Functions structure	Name
		(All...)	(All...)	(All...)	(All...)
01.01.2011 01:07:36.942 (4)		1		Ground fault log	
01.01.2011 01:07:36.942	00:00:00:00.000		1	Line 1:51Ns Sens. GFP 1:3I0> 1	Pickup
01.01.2011 01:07:36.942	00:00:00:00.000		2	Line 1:51Ns Sens. GFP 1:General	Ground fault
01.01.2011 01:07:36.942	00:00:00:00.000		3	Recording:Ground fault log	Fault number
01.01.2011 01:07:37.242	00:00:00:00.300		4	Line 1:51Ns Sens. GFP 1:3I0> 1	Operate delay expired
01.01.2011 01:00:03.904 (4)		2		Ground fault log	

Table 3: An example of a multimodal quadruple, incorporating both text and image context. The text highlighted in blue provides useful information to answer the question; however, it is insufficient to identify the ‘Read log entries’ button, which is highlighted in the image and also required for a correct answer.

D Detailed Results

Approach	Generator	Evaluator	Ans. Corr.	Ans. Rel.	Text Faith.	Text Ctx. Rel.	Img. Faith.	Img. Ctx. Rel.
Baseline	GPT-4V	GPT-4V	0.18	0.96	–	–	–	–
	GPT-4V	LLaVA	0.31	0.78	–	–	–	–
	LLaVA	GPT-4V	0.12	0.92	–	–	–	–
	LLaVA	LLaVA	0.19	0.70	–	–	–	–
Text-Only RAG	GPT-4V	GPT-4V	0.39	0.91	0.76	0.63	–	–
	GPT-4V	LLaVA	0.42	0.75	0.98	0.60	–	–
	LLaVA	GPT-4V	0.29	0.95	0.69	0.63	–	–
	LLaVA	LLaVA	0.24	0.81	0.99	0.57	–	–
Image-Only RAG Clip	GPT-4V MI	GPT-4V	0.24	0.89	–	–	0.20	0.36
	GPT-4V MI	LLaVA	0.31	0.79	–	–	0.24	0.01
	GPT-4V SI	GPT-4V	0.19	0.86	–	–	0.32	0.28
	GPT-4V SI	LLaVA	0.28	0.71	–	–	0.20	0.01
	LLaVA	GPT-4V	0.11	0.82	–	–	0.04	0.39
	LLaVA	LLaVA	0.14	0.83	–	–	0.29	0.01
Image-Only RAG Summaries	GPT-4V MI	GPT-4V	0.22	0.85	–	–	0.20	0.20
	GPT-4V MI	LLaVA	0.32	0.75	–	–	0.11	0.01
	GPT-4V SI	GPT-4V	0.21	0.80	–	–	0.30	0.18
	GPT-4V SI	LLaVA	0.31	0.69	–	–	0.14	0.00
	LLaVA	GPT-4V	0.10	0.86	–	–	0.11	0.36
	LLaVA	LLaVA	0.17	0.82	–	–	0.33	0.01
Multimodal RAG Clip	GPT-4V MI	GPT-4V	0.37	0.91	0.14	0.40	0.74	0.59
	GPT-4V MI	LLaVA	0.40	0.76	0.14	0.01	0.94	0.38
	GPT-4V SI	GPT-4V	0.35	0.86	0.11	0.29	0.76	0.57
	GPT-4V SI	LLaVA	0.38	0.73	0.13	0.00	0.95	0.35
	LLaVA	GPT-4V	0.30	0.94	0.05	0.28	0.62	0.63
	LLaVA	LLaVA	0.35	0.76	0.08	0.01	0.98	0.39
Multimodal RAG Summaries	GPT-4V MI	GPT-4V	0.43	0.86	0.17	0.21	0.88	0.72
	GPT-4V MI	LLaVA	0.46	0.71	0.06	0.00	0.97	0.43
	GPT-4V SI	GPT-4V	0.40	0.89	0.17	0.17	0.86	0.69
	GPT-4V SI	LLaVA	0.38	0.76	0.13	0.00	0.96	0.44
	LLaVA	GPT-4V	0.29	0.91	0.05	0.45	0.73	0.65
	LLaVA	LLaVA	0.28	0.80	0.29	0.00	0.99	0.50
Text-Only GSC	GPT-4V	GPT-4V	0.57	0.93	0.93	0.84	–	–
	GPT-4V	LLaVA	0.59	0.84	0.93	0.42	–	–
	LLaVA	GPT-4V	0.47	0.97	0.84	0.85	–	–
	LLaVA	LLaVA	0.52	0.90	0.95	0.42	–	–
Image-Only GSC	GPT-4V	GPT-4V	0.53	0.94	–	–	0.68	0.75
	GPT-4V	LLaVA	0.63	0.88	–	–	0.33	0.04
	LLaVA	GPT-4V	0.21	0.95	–	–	0.24	0.74
	LLaVA	LLaVA	0.63	0.86	–	–	0.40	0.04
Multimodal GSC	GPT-4V	GPT-4V	0.78	1.00	0.50	0.86	0.78	0.89
	GPT-4V	LLaVA	0.83	0.92	0.23	0.04	0.90	0.42
	LLaVA	GPT-4V	0.59	1.00	0.26	0.77	0.80	0.85
	LLaVA	LLaVA	0.63	0.89	0.18	0.04	0.93	0.41

Table 4: Full evaluation scores across all experiments and metrics. Each setup was run with LLaVA, GPT-4V prompted with a single image (GPT-4V SI), and GPT-4V prompted with multiple images (GPT-4V MI). The quality of the generated textual answer was evaluated with both LLaVA and GPT-4V. GSC refers to prompting with the gold standard context.