2024 16th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)
ICUMT 2024 · 26. – 28.11.2024 · Meloneras, Gran Canaria, Spain

Semantic Fusion of Text and Images: A Novel Multimodal-RAG Framework for Document Analysis

Tuhina Nandi*, Sidharth Gupta*, Abhishek Kaushal**, Radim Burget#, Stepan Jezek#, Malay Kishore Dutta**
*Amity Institute of Biotechnology, Amity University, Noida, India
**Amity Centre for Artificial Intelligence, Amity University, Noida, India
Dept of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czech Republic
tuhina.nandi@s.amity.edu, sidharth.gupta@s.amity.edu, abhishekkaushal401@gmail.com, malaykishoredutta@gmail.com, burgetrm@vut.cz, Stepan.Jezek1@vut.cz

***Abstract*— The presented multimodal Retrieval-Augmented Generation (RAG) model combines FAISS with Gemini 1.5 Flash to quickly recover and synthesize information from text and image data in PDFs. This methodology allows for high-accuracy searches within a unified vector space by using distinct FAISS indices for text and images, while text-embedding-004 processes queries and stored data. Following a user query submission, FAISS retrieves the most relevant text chunks and images based on similarity metrics. These are then submitted to the Gemini model, which produces a coherent and informative response. A recursive text-chunking technique optimizes the handling of token limitations, increasing processing performance and reducing redundancy. This framework excels in multimodal synthesis, extracting valuable insights from both textual and visual content while remaining computationally efficient. This method focuses on the model's utility in domains needing integrated analysis from several data sources, providing a comprehensive but simplified information retrieval solution.**

Keywords— Document Analysis, Multimodal Retrieval-Augmented Generation, Semantic Representation, Transformer-Based Models, Unified Embedding Space, Vision-Language Models

I. Introduction

The proposed work presents a multimodal Retrieval-Augmented Generation (MM-RAG) framework, designed to handle both multimodality of inputs, i.e. both texts and images for comprehensive multiple document analysis. The system in this study employs a sophisticated Vision-Language Model (VLM), specifically the Gemini model, to convert visual content into concise textual descriptions [1], [2]. This framework provides a seamless way to unify text and image modalities retaining their semantics. Upon querying, the proposed model has demonstrated exceptional capabilities in both textual and visual data analysis from multiple PDF documents.

In recent years, the Multimodal Retrieval-Augmented Generation (RAG) system has emerged as a ground-breaking advancement in the realm of intelligent information retrieval and generation [3], [4], [5]. Unlike traditional RAG models that focus solely on text, which, are limited by their inability to capture the rich contextual information available across different types of data. For instance, a text-based model may miss visual cues that could be critical for understanding context, leading to incomplete or less accurate predictions this innovative system leverages the power of multimodal data to address challenges that have been difficult to overcome with single-modality approaches [6]. On the other hand, while multimodal RAGs have made progress over the years, they often struggle with the alignment and fusion of heterogeneous data, leading to issues such as the dominance of one modality over others or difficulties in scaling across domains [7], [8].

A key innovation of this research is its dual-modality framework. By integrating large language models (LLMs) for text summarization with advanced vision models for image understanding, this system operates within a unified architecture that enhances both retrieval and generation capabilities. This integrated approach addresses limitations observed in traditional RAG systems that treat text and images independently. The balanced 768-dimensional embedding space used by the system captures both textual and visual nuances, ensuring detailed and meaningful vector representations of multimodal data. Moreover, the incorporation of Google Generative AI elevates the system's performance by providing more contextually aware and coherent responses. This enhancement aligns with recent research showing that advanced AI models can significantly improve language comprehension and generation, making them more effective for complex queries [9], [10]. The system's ability to seamlessly process both text and images allows it to excel in real-world applications where mixed-media content is prevalent, positioning it as a versatile tool for comprehensive document analysis [11].

This work represents a significant leap forward in the integration of text and image data, demonstrating a powerful and adaptable approach to understanding and generating insights from complex documents. The manuscript outlines the experimental setup and the PDFs that have been used to generate the sample responses as a form of benchmark. Followed by the description of the proposed Model Architecture explaining each component of the model succinctly. Lastly, the results and conclusion are discussed at the end, taking a futuristic perspective for the further research that can be done in this domain.

II. Experimental setup

A. Data Preparation

An open-source collection of PDF documents was utilized in this study. These PDFs were sourced from publicly available resources on the internet and contain both textual and visual content relevant to various subjects.

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 05:13:17 UTC from IEEE Xplore. Restrictions apply.

1. **PDF Documents:**

   “Neurobiology Introduction" by MUK Publications [13]**:** This document features extensive textual content along with various diagrams and charts pertinent to the field of neurology. It provides comprehensive coverage of neurological concepts, making it a valuable source for extracting textual and visual information. **“Animal Cell Culture and Technology”** [14]**:** This PDF includes detailed technical descriptions and images related to cell culture techniques. It covers methodologies, technologies, and protocols used in animal cell culture, with textual explanations and technical illustrations.

2. **Images:**

   Images were extracted from these PDFs to augment the multimodal capabilities of the Retrieval-Augmented Generation (RAG) system. Integrating these images into the system enhances its ability to manage and retrieve complex, multimodal information.

B. Data Processing

The whole process is depicted in Fig. 1 The PDFs are stored in a separate directory. Each PDF is processed to extract both text and images. The partition-pdf function from the unstructured library is employed to split the PDF into chunks, extracting text and images from every page. Images are saved in the /content/images directory, where they are processed and encoded in base64 format for integration into the multimodal system, with the encode-image function being used to read and encode these image files.

To generate vector representations of the extracted text, the 004 model is employed, enabling efficient retrieval and relevance matching for users' queries. Furthermore, images extracted from the PDFs are summarized using the Gemini-1.5 flash model from the Google-generative.ai library. The summaries are embedded using the same model that is applied for text embeddings, ensuring consistent representation of textual and visual data. Each image, encoded in base64 format, is directly associated with its corresponding textual description, thereby enhancing the multimodal capabilities of the RAG system.

For managing and efficiently storing these embeddings, FAISS (Facebook AI Similarity Search) indices are created, with separate indices for text and image embeddings. This design optimizes retrieval performance by allowing targeted searches based on data type. The embeddings for text and images are systematically added to their respective FAISS indices using the FAISS library [12], enabling quick and accurate retrieval of relevant information in response to user queries. The FAISS indices are saved to files, such as text-embeddings-index and image-embeddings-index, ensuring persistent storage. This allows for the reuse of indices in future sessions without the need for recompilation. When necessary, the saved FAISS indices are loaded from these files, maintaining continuity and efficiency in the retrieval process.

III. Model Architecture

The architecture allows the system to combine information from text and images to provide more comprehensive and informative answers to user queries.

A. PDF Parser

The proposed Model uses the unstructured library to parse PDF files. This library is designed to handle both structured and unstructured data, processing regardless of its storage format, whether from structured databases or unstructured text files. The library performs the necessary transformations and preprocessing steps to convert PDF files into a format that is suitable for analysis, such as JSON, facilitating further downstream processes. When the library processes each PDF, it breaks the document into discrete “elements”.

These elements could be pieces of text, images, or other content. Each element is defined as an object, and the architecture assigns specific properties to it. For instance, each element has a “Category” label indicating whether it is “Text” or “Image”. If the element is a text section, it includes the “Text Content” property that contains extracted text. Some elements may have further metadata, such as bounding boxes for images, which provide spatial context and layout information. A try-expect block is incorporated around the image processing function to handle potential issues gracefully even if certain images are problematic, it works efficiently without hindering the workflow.

B. Image Encoder

The image encoder provides an innovative technique, departing from the typical method of encoding images via fundamental format conversions or compression. Instead, it takes a more advanced approach, creating textual summaries of images that are then stored as vector embeddings for optimal storage and retrieval.

The Gemini-1.5-flash model summarises each image taken from the PDFs, generating a short, descriptive text suitable for database integration. This summary is explicitly labelled as "image description" to indicate its function. Once formed, the text-embedding-004 model converts this description into a vector embedding. This embedding captures the semantic meaning of the image's description in a numerical form, allowing the system to retrieve relevant information quickly. The data stored for each image includes key components: a type marker identifying it as "image"; the image data encoded in base64 format to preserve its original integrity; the name of the PDF file from which the image was taken; the descriptive text summary produced by the Gemini-1.5-flash model; and the vector embedding derived from this summary.

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 05:13:17 UTC from IEEE Xplore. Restrictions apply.

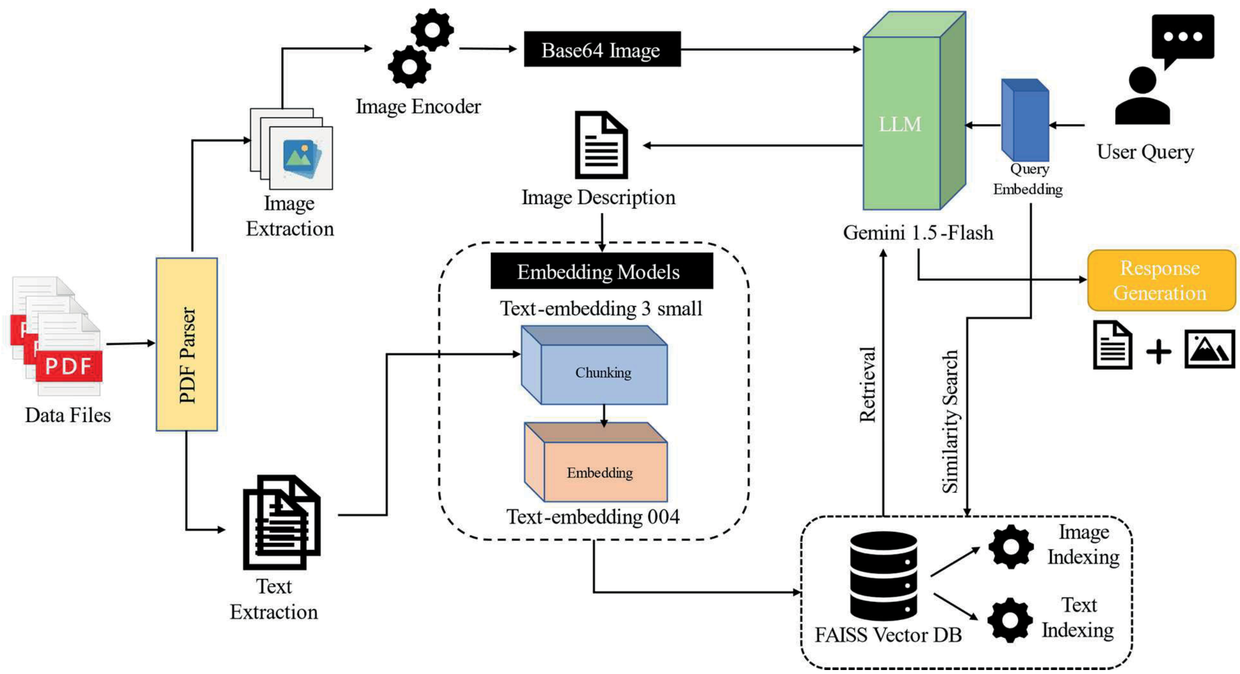


Fig 1. Proposed Model Architecture

C. *Text Splitter*

The model applies the RecursiveCharacterTextSplitter from the langchain library to divide the text into smaller, manageable chunks. This step ensures that text embeddings capture essential information without surpassing the input limits of the embedding model.

The splitter works recursively, first attempting to separate the text at higher levels, such as paragraphs or headings, to preserve as much context as possible. If the chunks are still too large, it separates them at a lower level—by sentences or individual words—to keep each chunk inside the token limit while preserving meaning. The model makes use of the Tik-Token library to estimate the number of tokens in the text accurately. Because different models use different tokenization methods, using the same tokenization as the embedding model assures accurate chunk sizing.

Finally, each chunk of the PDF is divided into smaller chunks that each fall under the stated token limit. This strategy maximizes contextual information in each chunk, allowing for more accurate and relevant retrieval within the limits of the embedding model.

D. *Embedding Model*

The proposed model uses two embedding models from Google

- **text-embedding-004**: This more advanced embedding model creates embeddings for both the main text and image descriptions, which are then saved in the vector database.
- **text-embedding-3-small**: The Recursive-Character-Text-Splitter uses a smaller, quicker embedding model to efficiently split the text into pieces of a certain token size.

Initially, the text is broken into manageable bits with text- text-embedding-3-small. The text-embedding-004 model then creates embeddings for every text chunk and image description. These embeddings for a given text or image are expressed as high-dimensional vectors indicated as e(x), which capture the semantic meaning of the content. (1).

$$\mathbf{e}(x) = \text{text-embedding-004}(x) \tag{1}$$

The embeddings are saved in an FAISS index, which is a specialized data format designed to facilitate similarity searches. When a user submits a query, text-embedding-004 generates an embedding for the query, which is then compared to the FAISS index to identify the most similar stored embeddings. The matching text and images are subsequently obtained to respond to the user's query, allowing for exact and relevant information retrieval.

E. *Vector Database*

The system uses FAISS (Facebook AI Similarity Search) as its vector database [12]. FAISS is an open-source library that focuses on efficient similarity search and clustering of dense vectors.

The proposed model initially generates two different FAISS indices, one for text embeddings and one for image embeddings. Each index is a data structure that organizes the embeddings in a way that enables quick similarity searches. After creating embeddings for text chunks and image descriptions, they are assigned to their corresponding FAISS

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 05:13:17 UTC from IEEE Xplore. Restrictions apply.

indexes. The indices are then saved to disc, so they can be loaded and reused later without having to recompute the embeddings.

When a user query is submitted, the embedded data is used to search the FAISS indices. FAISS quickly finds the most similar embeddings that correlate to the most relevant text and image data. The choice of FAISS is critical because of its efficiency in handling large-scale datasets and its capacity to do quick similarity searches. This efficiency is critical for promptly retrieving the most relevant information in response to user inquiries. There is also functionality to store and load the FAISS indices, which is necessary for maintaining the vector database and preventing repeated calculations.

F. *Query Processor*

The query processor handles user inquiries and obtains pertinent data from the vector database. This procedure consists of multiple steps. First, the user's question is encoded in an embedding vector using the same embedding model (text-embedding-004) as the text and picture data. The query embedding is then used to search two FAISS indices: one for text and one for images. FAISS quickly selects the most comparable embeddings to the query embedding using a distance measure, most (likely L2 distance (2) or cosine similarity (3)).

$$d(\boldsymbol{v_1}, \boldsymbol{v_2}) = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{v}_{1\mathrm{i}} - \mathrm{v}_{2\mathrm{i}})^2} \tag{2}$$

$$\text{cosine similarity}(\mathrm{a}, \mathrm{b}) = \frac{\mathrm{a} \cdot \mathrm{b}}{|\mathrm{a}||\mathrm{b}|} \tag{3}$$

The indices of the most similar embeddings are used to extract text chunks and image descriptions from the stored data. The recovered text and image data, along with the original query, are fed into a large language model (LLM). The LLM uses this data to provide a thorough and informative response to the user's query.

G. *Retreival Mechanism*

The retrieval method finds the most relevant information in the vector database depending on a user's query. It uses the FAISS package and pre-computed embeddings to efficiently find related vectors. When a user enters a question, it is first converted into an embedding vector with the same embedding model as the text and image data (text-embedding-004).

This assures that the query and the stored data share the same vector space. The query embedding is then used to search the FAISS indices—one for text and one for images. FAISS employs efficient algorithms to find the nearest neighbors to the query embedding in the high-dimensional vector space. The search process returns the indices of the most similar embeddings within the FAISS indices. These indices correspond to the locations of the most relevant text chunks and image descriptions in the original data lists.

The system obtains the actual text and image data associated with the retrieved indices, which contain the text content of the relevant text chunk, the relevant image's base64 representation, and the created summary. In this implementation, the best match from each index (text and image) is chosen. However, with a more advanced system, numerous top matches might be collected and sorted based on similarity scores or other factors.

The acquired text and image data are then fed into a large language model (LLM), which generates a detailed response to the user's query. This retrieval strategy guarantees that the LLM has access to the most relevant information from the vector database, allowing it to provide more accurate and informative results.

H. *Large Language Model*

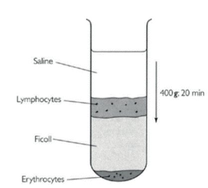
The Gemini 1.5 Flash model is used as the system's Large Language Model (LLM). Gemini is a family of big language models created by Google that are notable for their high performance in a variety of tasks such as text generation, translation, and code generation.

This approach is used to create succinct descriptions of images. It accepts an image as input and returns a written description within a defined token limit. This method enables effective storage and retrieval of image information in the vector database. After obtaining relevant text and image information, it is used to provide a prompt for the Gemini model. The model then responds to the user's query using this prompt, combining both the collected information and its underlying expertise. The Gemini-1.5-flash model is used specifically in the system due to its performance and efficiency balance. It provides faster response times than larger Gemini models while maintaining high-quality outcomes. The Google-generative-ai library serves as a handy interface for sending requests to the Gemini API and returning answers. By exploiting Gemini's capabilities, the system gains superior natural language understanding and generation, allowing it to respond to user inquiries in an informative and logical manner based on both textual and visual information collected from PDF files.

IV. Result & Conclusions

The Multimodal Retrieval-Augmented Generation (RAG) model performed well in digesting and synthesizing data from the supplied PDF document. When queried, the model displayed excellent capabilities in both textual and visual data analytics. Table 1 shows some sample query and response outcomes obtained from the suggested model utilizing PDFs.

Table 1: Sample response from the proposed model

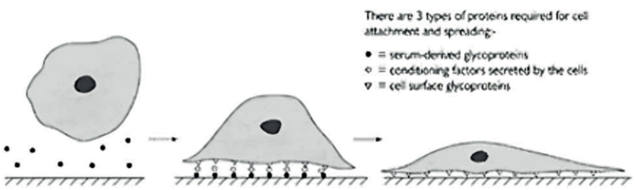
Query1- How to Separate cells by using gradient centrifugation?
Image Response

Text Response
Gradient centrifugation uses density gradients, often made with Ficoll or Percoll, to separate cells based on their density. Cells sediment to a position in the gradient matching their

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 05:13:17 UTC from IEEE Xplore. Restrictions apply.

density (isopycnic sedimentation). This method is particularly useful for isolating specific cell types, like lymphocytes, in sterile conditions.

Query2- What are the key factors involved in anchorage-dependence of cells, and how do proteins such as fibronectin and divalent cations contribute to the process of cell adhesion to a solid substratum?

Image Response

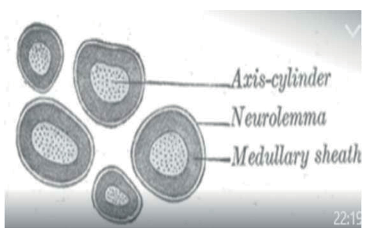


Text Response

Anchorage-dependent cells need to attach to a solid surface (like a Petri dish) to grow. This attachment happens through a combination of:
- Van der Waals forces: These weak interactions help the cell membrane initially adhere to the surface.
- Ca2+: These ions help strengthen the initial attachment.

Query3 - What do you understand by Transverse sections of medullated nerve fibers?

Image Response



Text Response

A transverse section of a medullated nerve fiber is a cross-section view of the fiber, of the fiber, revealing its internal structure. You'd see the axis-cylinder (the central core containing the axon), surrounded by the medullary sheath (the fatty myelin layer), and the outermost neurolemma (the Schwann cell membrane)

In terms of textual data processing, the model successfully extracted and summarised key information from the PDF. It included a detailed summary of the main issues, important observations, and relevant results, ensuring a clear and thorough comprehension of the document's content. For visual data analysis, the model carefully investigated visual features in the PDF, such as images, charts, and diagrams. It correctly discovered and defined these visual components, extracting useful insights and elegantly combining them with the textual material. Moreover, the model showcased an advanced ability to correlate and integrate both textual and visual data. This integrative approach resulted in a cohesive and detailed summary, adeptly combining insights from both types of information. This holistic synthesis underscored the model's effectiveness in delivering a unified and comprehensive overview of the document's content.

Acknowledgment

This work was supported by MVCR reg. no. VK01010153, Czechia.

References

[1] Mondillo, G., Frattolillo, V., Colosimo, S. et al. Basal knowledge in the field of pediatric nephrology and its enhancement following specific training of ChatGPT-4 “omni” and Gemini 1.5 Flash. Pediatr Nephrol (2024). https://doi.org/10.1007/s00467-024-06486-3

[2] Reid, Machel, et al. "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context." arXiv preprint arXiv:2403.05530 (2024).

[3] Ghosh, Akash, et al. "Exploring the frontier of vision-language models: A survey of current methodologies and future directions." arXiv preprint arXiv:2404.07214 (2024).

[4] Chen, Wenhu, et al. "Murag: Multimodal retrieval-augmented generator for open question answering over images and text." arXiv preprint arXiv:2210.02928 (2022).

[5] R. Zhou, “Advanced Embedding Techniques in Multimodal Retrieval Augmented Generation A Comprehensive Study on Cross Modal AI Applications,” *Journal of Computing and Electronic Information Management*, vol. 13, no. 3, Art. no. 3, Jul. 2024, doi: 10.54097/h8wf8vah.

[6] “Retrieval-Augmented Generation for Large Language Models: A Survey.” Accessed: Sep. 10, 2024. Available: https://arxiv.org/html/2312.10997v5

[7] Yuan, Ye, et al. "A Hybrid RAG System with Comprehensive Enhancement on Complex Reasoning." arXiv preprint arXiv:2408.05141 (2024).

[8] R. Zhao *et al.*, “Retrieving Multimodal Information for Augmented Generation: A Survey,” in *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore: Association for Computational Linguistics, 2023, pp. 4736–4756. doi: 10.18653/v1/2023.findings-emnlp.314.

[9] Doris, Anna C., et al. "DesignQA: A Multimodal Benchmark for Evaluating Large Language Models' Understanding of Engineering Documentation." arXiv preprint arXiv:2404.07917 (2024).

[10] Lu, J., et al. (2021). VL-BERT: Pre-training of Generic Visual-Linguistic Representations. Proceedings of the International Conference on Learning Representations (ICLR).

[11] P. Omrani, A. Hosseini, K. Hooshanfar, Z. Ebrahimian, R. Toosi and M. Ali Akhaee, "Hybrid Retrieval-Augmented Generation Approach for LLMs Query Response Enhancement," 2024 10th International Conference on Web Research (ICWR), Tehran, Iran, Islamic Republic of, 2024, pp. 22-26, doi: 10.1109/ICWR61162.2024.10533345.

[12] Herbrich, J., Kreymer, I., & van der Maaten, L. (2017). FAISS: A Library for Efficient Similarity Search and Clustering of Dense Vectors. Facebook AI Research.

[13] Dr. Zhang Peilan, (2016) “Neurobiology Introduction" by MUK Publications. ISBN- 978-81-89630-98-0

[14] Butler, M. (2004). Animal Cell Culture and Technology (2nd ed.). Routledge. https://www.routledge.com/Animal-Cell-Culture-and-Technology/Butler/p/book/9781859960493.

Authorized licensed use limited to: DePaul University. Downloaded on July 02,2025 at 05:13:17 UTC from IEEE Xplore. Restrictions apply.