T.C.
MANİSA CELAL BAYAR UNIVERSITY
GRADUATE SCHOOL

MASTER THESIS
DEPARTMENT OF COMPUTER ENGINEERING
COMPUTER ENGINEERING PROGRAM

RAG SUPPORTED KNOWLEDGE-BASED QUESTION-ANSWER SYSTEM WITH MULTIMODAL (TEXT + VISUAL) DATA

Onur YAZICI

Supervisor
Asst. Prof. Dr. Volkan ALTINTAŞ

MANİSA–2025

Onur Yazıcı

RAG SUPPORTED KNOWLEDGE-BASED QUESTION-ANSWER SYSTEM WITH MULTIMODAL (TEXT + VISUAL) DATA

2025

COMMITMENT

I declare that this thesis was written in accordance with academic and ethical rules at Manisa Celal Bayar University Graduate Education Institute, Department of Computer Engineering , that all literature information used is referenced and included in the thesis, that it is entirely my own work, that I have cited the source for every quotation, and that I have not used any artificial intelligence or programs that violate academic and ethical rules in writing this thesis.

Onur YAZICI

ÖZET

Yüksek Lisans

Onur YAZICI

Manisa Celal Bayar Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğretim Üyesi. Volkan ALTINTAŞ

Geleneksel sistemlerin ve Büyük Dil Modellerinin (LLM) çok modlu verilerle (metin, tablolar, şekiller) ve gizlilik gerektiren karmaşık PDF belgeleriyle başa çıkmadaki yetersizliğini gidermek amacıyla, bu tez, mevcut çevrimiçi bağımlılıkları aşan, kapsamlı ve enerji verimli bir çevrimdışı RAG (Retrieval-Augmented Generation) çerçevesi sunmaktadır. Çalışmanın teknik ayırt edici özelliği, görsel (Image) verileri ayrı yapılar aracılığıyla işleyen çevrimdışı çift kanallı çok modlu mimarisinde yatmaktadır. Bu yaklaşım, Büyük Dil Modellerini (LLM'ler) harici bir API'ye ihtiyaç duymadan doğrudan cihaz üzerinde yerel olarak barındırarak (Edge AI) kapalı ve güvenilir bir çözüm sunmaktadır; geliştirilen bu Hibrit Sistem, temel VLM modellerinin (LLaVA ve VILA'nın yaklaşık 0.72-0.73 doğruluğu) performansını önemli ölçüde aşarak gerçek dünya testlerinde ortalama 0.8827'lik bir başarı (CheckAvail puanı) elde etmiş ve böylece çevrimiçi çalışan, yüksek performanslı XSum mimarisine (0.97 puan) karşı rekabetçi bir alternatif olduğunu kanıtlamıştır. XSum puanının metin odaklı metriklerle tutarlı olduğu görülmüştür. Özellikle kanıt yakalama başarısında (true0) diğer hibrit modellerle benzer başarı gösteren Hibrit Puan Modeli, ne yazık ki alakasız kanıtları hariç tutma (true1) metriklerinde en düşük değere (0.69) sahiptir; bu durum, modelin özgüllükten yoksun olduğunu ve yanlış pozitiflere (FP) eğilimli olduğunu göstermektedir. Yapılan detaylı analizler, true1 metriğindeki bu düşük performansın esas olarak görsel (Image) kanalından gelen kanıtların değerlendirilmesinden kaynaklandığını; metin (Text) kanalının ise tatmin edici sonuçlar ürettiğini ortaya koymuştur. Yapılan ek analizlerde, bu yanlış pozitiflerin çevrimdışı bir görsel LLM (Vision LLM) ile doğrulamasının yapılması sonucunda elendiği gözlemlenmiştir. İleride daha iyi bir çözüm bulunması için çalışmalar sürmektedir. Bu nedenle, gelecek çalışmalar, özellikle görsel kanaldaki bu sorunun üstesinden gelmek için modelin özgüllüğünü artıracak yeni nesil filtreleme mekanizmalarına, kaynak kısıtlı LLM'ler için mimari optimizasyonlara ve farklı PDF yapılarına uyum sağlayabilen sağlam görüntü/görsel çıkarma fonksiyonları geliştirmeye odaklanacaktır.

Anahtar Kelimeler: Geri Alma Destekli Üretim (RAG), Büyük Dil Modelleri (LLM), Çevrimdışı Sistem, Çok Modlu Mimari, Çift Kanallı Mimari, PDF Belge İşleme, Edge AI, Gizlilik Odaklı Sistemler, Model Özgüllüğü, Filtreleme Mekanizmaları.

2025, 80 sayfa

ABSTRACT

M.Sc. Thesis

Onur YAZICI

Manisa Celal Bayar University
Graduate School of Education
Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Volkan ALTINTAŞ

To address the inadequacy of traditional systems and Large Language Models (LLMs) in handling multimodal data (text, tables, figures) and complex PDF documents requiring privacy, this thesis presents a comprehensive and energy-efficient offline RAG (Retrieval-Augmented Generation) framework that overcomes existing online dependencies. The technical distinguishing feature of the work lies in its offline dual-channel multimodal architecture, which processes visual (Image) data through separate structures. This approach provides a closed and reliable solution by hosting Large Language Models (LLMs) locally on the device (Edge AI) without requiring an external API; this developed Hybrid System significantly outperforms the accuracy of basic VLM models (LLaVA and VILA's approximately 0.72-0.73 accuracy) in real-world tests, achieving an average success rate (CheckAvail score) of 0.8827. This proves it is a competitive alternative to the online, high-performance XSum architecture (0.97 score). The XSum score was found to be consistent with text-focused metrics. The Hybrid Score Model, which shows similar success to other hybrid models, particularly in evidence capture success (true0), unfortunately has the lowest value (0.69) in the metrics for excluding irrelevant evidence (true1); this indicates that the model lacks specificity and is prone to false positives (FP). Detailed analyses have revealed that this low performance in the true1 metric is primarily due to the evaluation of evidence from the visual (Image) channel, while the text channel produced satisfactory results. Additional analyses showed that these false positives were eliminated when verified using an offline visual LLM (Vision LLM). Work is ongoing to find a better solution in the future. Therefore, future work will focus on developing new-generation filtering mechanisms that will increase the model's specificity to overcome this problem, especially in the visual channel, architectural optimizations for resource-constrained LLMs, and robust image/visual extraction functions that can adapt to different PDF structures.

Keywords: Retrieval-Augmented Generation(RAG), Large Language Models (LLM), Offline System, Multi-Modal Architecture, Dual-Channel Architecture, PDF Document Processing, Edge AI, Privacy-Focused Systems, Model Specificity, Filtering Mechanisms.

2025, 80 pages

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor, Asst. Prof. Dr. Volkan ALTINTAŞ, for his valuable contributions, support, and guidance throughout my thesis work. His dedication and knowledge shared in shaping my thesis have significantly enhanced the quality of this work.

Onur YAZICI
Manisa, 2025

LIST of ABBREVIATIONS

ANN	Artificial Neural Networks
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
CE	Contextual Embedding
CLIP	Contrastive Language-Image Pretraining
CPU	Central Processing Unit
CS	Correctness Score
DB	Database
FN	False Negative
FP	False Positive
GPT	Generative Pre-trained Transformer
GPU	Graphics Processing Unit
HNSW	Hierarchical Navigable Small World
HS	Hybrid Score
KW	Keyword
LLM	Large Language Model
NLP	Natural Language Processing
PDF	Portable Document Format
QA	Question Answering
RAG	Retrieval-Augmented Generation
RAM	Random Access Memory
RS	Relevance Score
TN	True Negative
TP	True Positive
UI	User Interface
VLM	Visual Language Models
VQA	Visual Question Answering

LIST of FIGURES

LIST of TABLES

CONTENTS

CHAPTER THREE

METHODOLOGY AND SYSTEM IMPLEMENTATION

CHAPTER ONE

INTRODUCTION

1.1 Problem Statement and Research Significance

In today's rapidly growing information age, the volume and variety of data continue to increase. These data sets have a multimodal structure that includes not only textual content but also images, tables, and complex arrangements. Unfortunately, traditional information access and management systems are inadequate in effectively accessing such unstructured and complex multimodal data and providing accurate, contextually relevant information. This situation constitutes the fundamental problem area of this study. On the other hand, while Large Language Models (LLMs) offer groundbreaking advancements in natural language processing capabilities, they have inherent limitations such as having static knowledge bases and the risk of hallucination.

Given these limitations, Retrieval-Aided Generation (RAG) emerges as an effective solution, particularly for knowledge-based tasks, significantly expanding the capabilities of large language models. RAG is a critical approach that substantially improves response quality by enriching LLMs with external knowledge sources [1]. This method increases accuracy and reliability in knowledge-intensive tasks [2]and enables LLMs to overcome the static limitations of their training data, thereby improving the accuracy of outputs against real-world data [3]. Overall, RAG is defined as one of the most advanced AI techniques that facilitates tasks by providing reliable and up-to-date external information (“RAG Conference Survey on LLMs”). Additionally, when used in conjunction with complementary techniques such as fine-tuning, RAG has the potential to not only improve LLM performance but also reduce hallucinations and strengthen the controllability of outputs [4]. Furthermore, integrating RAG with multimodal models known for their potential in document understanding [5] may further enhance the strategy's effectiveness [4].

This thesis presents a significant innovation by offering a comprehensive and energy-efficient offline framework designed to overcome the online dependency and single-modal limitations found in existing RAG systems. While most multi-modal

studies focus on real-time data and general Visual Question Answering (VQA) systems, this thesis makes a unique contribution: The work focuses on an RAG solution specifically designed for PDF documents, which require strict confidentiality and have complex, intricate layouts. The technical distinction of our thesis lies in its dual-channel multimodal architecture used in an offline environment. This architecture processes textual and visual data (including tables and figures) through separate structures and dedicated encoders. This methodological approach is crucial because it effectively preserves the structural integrity of complex PDF documents, thereby maximizing information access accuracy.

The fact that the entire process is conducted entirely offline constitutes the most critical architectural feature of the project. This fundamental approach significantly enhances data security when working with sensitive information, completely eliminates all external Application Programming Interface (API) dependencies, and ultimately maximizes the system's predictability and reliability. This framework successfully addresses a critical gap in the current academic literature by combining the existing Edge AI trend with an energy-efficient and reliable Question-Answering system. In this context, LLMs, embedded models, and all necessary data processing tools will be hosted and run locally. Optimization is critical to maximize RAG performance in this local and closed environment. For example, studies have shown that adding a specialized function call mechanism leads to significant improvements in metrics such as response relevance and contextual accuracy in RAG systems running locally deployed Ollama models [6] This mechanism enhances the reliability and structural correctness of the output. Furthermore, the literature has shown that using a local Ollama-based LLM can practically address concerns related to both data privacy and system response speed in such an architecture [6].

1.1.1 The Research Problem

Modern information systems, including Large Language Model (LLM) programs, demonstrate significant capabilities in text-based querying. However, these models' dependence on static training data reduces response reliability and leads to hallucination, which often produces incorrect or fabricated information. Traditional Retrieval-Augmented Generation (RAG) systems, developed to solve this critical issue, expand the LLM's knowledge base by leveraging external text documents. However, corporate and academic documents contain high-density multimodal content

such as graphics, tables, and images presented in PDF format. The focus of traditional RAG systems solely on textual data leads to the neglect of this visual and structured content, resulting in critical contextual meaning loss and gaps in responses to user queries.

In this context, the multi-modal RAG solutions developed unfortunately often depend on high-performance cloud services and a constant internet connection. This external dependency renders the system unusable in areas where data privacy and operational security must be ensured at the highest level, such as legal sensitivity, military research, or private medical records. Therefore, there is a continuous need for a reliable multimodal RAG architecture that is completely offline and locally hosted, while also being able to integrate both textual and visual information. This requirement forms the core problem area of the study.

This thesis presents a significant innovation by offering a comprehensive and energy-efficient offline framework designed to overcome the online dependency and single-mode limitations found in existing RAG systems. The work focuses on a RAG solution specifically designed for PDF documents that require strict confidentiality and possess complex, intricate layouts. The technical distinction of our thesis lies in its dual-channel, multi-modal architecture used in an offline environment. This architecture processes text and visual data (including tables and figures) through separate structures and dedicated encoders. This methodical approach is crucial as it effectively preserves the structural integrity of complex PDF documents, thereby maximizing the accuracy of information retrieval.

The technical challenge in successfully implementing this offline multi-modal RAG architecture primarily stems from the complex internal structure of PDFs. This challenge carries an inherent risk of error during the stage of correctly extracting and interpreting images. These critical preprocessing risks can lead to incorrect images being captured in the RAG data pipeline and subsequently fed to the Large Language Model (LLM), resulting in erroneous and misleading outputs (hallucinations). Therefore, this work focuses on the design, implementation, and overcoming of challenges of this offline Multimodal RAG architecture to effectively extract and correlate textual and visual information from PDFs. This minimizes the risk of hallucinations and ensures full contextual accuracy even in offline environments.

1.2 Research Objectives and Goals

The primary objective of this study is to design and implement an offline, PDF-based, multimodal RAG system architecture that minimizes hallucinations and ensures data privacy. In line with this general objective, the specific objectives and goals that the study aims to achieve are listed below:

- **Developing Offline Multi-Modal Data Extraction:** Creating a fully offline data processing pipeline that can accurately distinguish both textual and visual elements (tables, graphs, images) from PDF documents and enable their extraction while preserving their spatial context.

- **Local Embedding and LLM Integration**: The first objective is to identify open-source Multimodal Embedding Models that can operate with high performance in offline environments. These models must generate semantic representations (embeddings) for both visual and textual data. The second goal is to establish a locally hosted LLM infrastructure that is capable of being effectively fed and processed with these identified embeddings.

- **Demonstrating Energy Efficiency and Accessibility**: Commercial, online large-scale models require high energy and computational resources. However, this work demonstrates that certain critical multimodal computing processes can also be performed effectively offline and on local hardware. Thus, these technologies are proving accessible to resource-constrained institutions and the broader public.

- **Developing a Multi-Modal Indexing, Retrieval, and Verification Strategy:** Developing Multi-Modal Vector Indexing and Retrieval Strategies that establish semantic bridges between extracted and embedded text and image data, enabling both modalities to be retrieved simultaneously during querying. In the final stage of this chain, the goal is to implement a verification mechanism by questioning the consistency of the images in the context received by the LLM with the relevant text, thereby minimizing the risk of the final output being incorrect and obtaining the best and most accurate output.

- **Comprehensively Evaluating System Performance:** To measure and evaluate the performance of the developed offline multimodal RAG system, particularly in terms of the accuracy, comprehensiveness, and relevance of its responses to complex queries requiring visual information, by comparing it with the

performance of traditional single-modal RAG systems. During this evaluation process, both the response quality of text-based queries and the response quality of multimodal queries where visual content is successfully processed and utilized will be analyzed separately, revealing the system's effectiveness in each modality in detail.

- **Proving Reliability:** Demonstrating the importance and effectiveness of the system operating offline in terms of data confidentiality and operational independence in scenarios where sensitive data is processed.

1.3 Research Questions

The primary objective of this study is to present the details of the offline, PDF-based, multimodal (textual and visual) RAG system architecture described in detail above and to identify the approaches required to successfully implement this system. The project aims to produce more accurate, comprehensive, and reliable responses to complex queries, taking into account both textual and visual contexts, in scenarios where sensitive data is processed, or internet access is limited. In line with this general objective, the specific research questions addressed by the study are as follows:

Multimodal Information Extraction from PDFs: How can we simultaneously and effectively extract both textual and visual information (tables, graphs, images, etc.) from PDFs in an offline environment and structure them for integration into a RAG system?

Text and Image Embedding Models: Which open-source or locally hosted models are suitable for offline use and can generate semantic representations (embeddings) for both textual and image data, and how effective are these models at establishing semantic relationships between modalities?

Multimodal Information Retrieval and Extraction Strategies: How should extracted and embedded text and image data be indexed in a vector database to provide fast and relevant responses to multimodal queries? What are the strategies for retrieving both textual passages and related images simultaneously, and how can their performance be evaluated?

Offline Large Language Model Integration: Which open-source or locally hosted LLMs can operate offline and effectively utilize the received multimodal context (text and image) to generate high-quality responses, and how can the multimodal input processing capabilities of these LLMs be optimized?

System Performance and Comparison: How can the accuracy, relevance, and comprehensiveness of the responses provided by the developed offline multimodal RAG system to text and image queries be evaluated? In which areas does this multimodal approach provide significant performance improvements compared to traditional single-modal RAG systems?

Answering these research questions will provide practical and important insights into the design and implementation of offline multimodal RAG systems and guide future work in this area.

1.3.1 Performance Evaluation

The accuracy, relevance, and comprehensiveness of the responses provided by the developed offline multi-modal hybrid RAG system to text and image queries will be evaluated, and performance comparisons will be made with both traditional single-modal systems and existing foundational models.

1. **Text-Based Performance:** The quality of the model's summary responses to textual queries will be measured using metrics divided into three main categories. These are Traditional and Semantic Metrics (metrics such as ROUGE and BERTScore will be used), Hybrid Evaluation Metric (the CheckEval metric will be used to measure the effectiveness of the system's hybrid scoring strategy), and Qualitative Evaluation Metric (the G-Eval metric will be used to evaluate qualitative dimensions of the output such as consistency and fluency).

2. **Image-Based Performance:** The visual information processing capability of the multi-modal RAG system will be evaluated by focusing on the models used and the functionality of the components. This evaluation consists of two main areas: The first is Cross-Modal Embedding Performance (CLIP), where the CLIP model's ability to represent visual and textual embedded elements in the same vector space and its ability to establish semantic relationships between modalities will be measured. The second is Visual Query Response Accuracy (Gemma Vision), which will evaluate the Gemma Vision model's ability to extract and interpret information from visual elements (tables, graphs, etc.); this evaluation will be based on the accuracy of integrating visual information into the RAG context and the rate of producing correct responses.

1.4 Research Assumptions and limitations

This study is an experimental and application-oriented project that aims to increase the effectiveness and reliability of multi-modal RAG systems in offline environments. It is conducted within the framework of some basic assumptions and limitations that may affect its progress and the generalizability of its results. The fundamental assumptions supporting the scientific validity of the study are as follows:

- **Sufficiency of Local LLM and Embedding Models:** It is assumed that the open-source LLM and multimodal embedding models selected for offline operation possess sufficient semantic understanding capabilities for the tasks defined within the scope of the thesis (image description, contextual relationship establishment, and response generation). Specifically, although the performance of these local models does not encompass all the features of online proprietary commercial models, they are designed to compete with some of their structures and are considered to be at a level that meets the project's objectives.

- **Representability of PDF Structure:** It is assumed that the basic text, table, and visual content of the PDF documents to be used can be sufficiently accurately parsed and structured using the selected data extraction tools (parsers), while preserving the spatial and semantic relationships between visual and textual elements. It is assumed that the potential erroneous effects of PDF structures during the data extraction process will be minimized through specific coding and optimization efforts, which will be detailed in the methodology section of this thesis, and that this situation will be kept at an acceptable level that will not adversely affect the overall success of the system.

- **Offline Infrastructure Capacity**: It is assumed that the offline hardware and software infrastructure (GPU, CPU, memory) on which the work is performed has the capacity to run the selected local LLM and embedded models at an acceptable response speed. It is assumed that excessive delays caused by hardware limitations will not invalidate the basic functionality of the system.

- **Hallucination Reduction Effect**: It is assumed that the developed Multimodal RAG architecture can statistically significantly reduce the hallucination rate compared to LLM's working only with text by increasing access to external and verifiable information.

1.5 Definitions and Conceptual Framework

Explaining the technical terms and theoretical approaches used throughout this thesis is critical to ensuring the integrity and comprehensibility of the work. This section presents the fundamental concepts underlying the research and the structural framework illustrating the relationships between these concepts.

1.5.1 Basic Concepts and Definitions

1. **Large Language Models (LLMs):** Deep learning models with billions of parameters, trained on large text datasets, capable of performing various natural language processing tasks such as text understanding, text generation, summarization, and translation. This thesis is based on locally hosted (offline) open-source LLMs.

2. **Retrieval Augmented Generation (RAG):** Retrieval-Augmented Generation (RAG): An artificial intelligence architecture developed to overcome the information limitations of LLMs based on static training data and reduce their tendency to hallucinate. In this process, external information sources are first divided into manageable chunks using a method called Chunking and stored in a Vector Database as semantic representations (embeds). When a user query arrives, the system retrieves the most relevant chunks from this database and uses this enriched context to generate the final response.

3. **Multimodal RAG:** An approach that extends the capabilities of traditional RAG systems to encompass modalities beyond text (images, tables, graphs, etc.). This system aims to provide the LLM with richer and contextually accurate input by processing and correlating both textual and visual data simultaneously.

4. **Embedding Models:** These are models that convert complex data such as text or images into sequences of numbers (vectors) that represent them semantically in a vector space. In multimodal RAG, representing data from different modalities in a common vector space is essential for cross-modal retrieval.

5. **Offline System:** A system architecture that does not require an internet connection or external cloud APIs, where all computation, data extraction, indexing, and response generation processes are performed on a local server or personal computer. This is a critical element in terms of data security and independence.

1.5.2 Conceptual Framework

The conceptual framework of this study is based on a system cycle comprising the following steps: multimodal parsing of PDF data in an offline environment, creation of separate embeddings for visual and textual data, separate indexing in vector space, and finally, response generation through contextual verification by a local LLM. This framework redefines the traditional RAG chain within the triangle of multimodal data processing, local model constraints, and security requirements. The system processes and indexes visual and textual information through separate channels; however, during querying, it connects (correlates) the two separate information sets to access a common knowledge base. This approach maximizes the scope and quality of the context presented to the LLM. The work is positioned as a mechanism that demonstrates how successfully separately processed visual information can be associated with textual context and incorporated into the final response.

1.6 Thesis Structure

This thesis consists of five main sections. The first section introduces the research problem, summarizes specific objectives and goals, defines key concepts, and outlines the overall scope of the study. The second chapter presents a comprehensive literature review, detailing the evolution of RAG systems from single-modal to multi-modal and critically examining the theoretical foundations and inherent challenges of offline multi-modal processing. The third chapter details the methodology and implementation of the proposed system, focusing on specific tools such as CLIP and Gemma, advanced segmentation strategies, and the design of Hybrid Scoring and contextual verification mechanisms. The fourth chapter presents the experimental setup, performance metrics, and results obtained by comparing the proposed offline Multimodal RAG system with a baseline single-modal system. Finally, the fifth chapter discusses the findings, evaluates the degree to which the research objectives were achieved, identifies the limitations of the study, and suggests directions for future research.

CHAPTER TWO

LITERATURE REVIEW: MULTİMODAL RETRİEVAL-AUGMENTED GENERATİON (RAG) ARCHİTECTURE AND CHALLENGES

2.1. Fundamentals Of Artificial Intelligence-Based Production Systems

2.1.1 Overview of Natural Language Processing and Large Language Models

2.1.1.1 Transformer Architecture and Development

NLP is a subfield of artificial intelligence that focuses on computers' ability to understand, interpret, and generate human language. The most critical development underpinning modern NLP is the Transformer architecture, introduced in 2017.

Transformer Architecture: Unlike traditional recurrent (RNN) and convolutional (CNN) networks, this architecture can process long-range dependencies between words in a sentence in parallel using the Attention Mechanism. This architecture was introduced in a paper titled “Attention Is All You Need” published by Google researchers in 2017[7]. Transformers paved the way for the emergence of encoder-based models such as BERT (Bidirectional Encoder Representations from Transformers) [8] and decoder-based generative models such as GPT (Generative Pre-trained Transformer) [9].

2.1.1.2 Local (On-Premise) and Open-Source LLMs

Large Language Models (LLMs) are Transformer-based artificial neural networks with billions of parameters, trained on massive datasets [10]. For offline usage scenarios relevant to my thesis, locally hosted (on-premise) and open-source versions of these models are critical.

On-Premise Hosting: In environments with limited internet access or where sensitive data is processed, running LLMs on your own hardware (on-premise hosting) ensures privacy and security[10].

Open-Source Models: Models such as Gemma [11] is developed by the open-source community and are available for anyone to use or modify under specific licenses. These models can be easily integrated into offline environments without commercial restrictions.

2.1.1.3 Retrieval-Augmented Generation (RAG) Systems

Large Language Models (LLMs), despite their ability to generate fluent and natural-sounding responses, have significant limitations. These limitations include factual inconsistencies and a tendency to generate fabricated information (hallucination), as well as the inability to access current or domain-specific information after the LLM's training is complete (knowledge gap)[12].

The Retrieval-Augmented Generation (RAG) architecture, designed to overcome limitations specific to pure generative models (hallucination, information dropout), offers a hybrid methodology. This methodology integrates the generation capacity of Large Language Models (LLMs) with the ability to draw relevant data from a dynamic and external information source[12].

2.1.2 Retrieval-Augmented Generation (Rag) Architecture and Evolution

2.1.2.1 Traditional RAG Architecture

One of the most significant limitations of Large Language Models (LLMs) is their inability to access up-to-date information due to the static nature of training data, and their tendency to generate unverifiable content (hallucination). To address these issues, [12]proposed the Retrieval-Augmented Generation (RAG) architecture. RAG aims to enhance the accuracy and transparency of responses by supporting an LLM's information generation capabilities with facts obtained from an external and reliable knowledge base.

A traditional RAG system goes through four main stages when processing a query:

Indexing: External information sources are divided into easily processable chunks. These chunks are converted into numerical vectors using an embedding model that preserves their semantic representations and are stored in a vector database. This stage forms the basis for making the data searchable.

Retrieval: The user query is also converted into a vector embedding. This embedding is used for similarity search to find the most relevant data pieces in the vector database. The goal is to retrieve the best pieces that are semantically most compatible with the query.

Augmentation: The relevant data fragments (context) that were retracted are combined with the original user query to create a single enriched prompt.

Generation: This enriched prompt is fed to the LLM. Instead of using its general knowledge, the LLM produces the final and verified response based solely on the provided external context.

2.1.2.2 RAG Evolution and Advanced Techniques

Early RAG implementations (Naive RAG) relied on simple segmentation and similarity search, leading to issues such as low-quality extracted segments (irrelevant segments) or overly long contexts[2]. To overcome these limitations and improve retrieval quality, the literature has proposed improvements at every stage of the RAG pipeline.

Advanced RAG techniques are generally examined under three main categories:

Pre-Retrieval Optimization: These are strategies applied before the retrieval process. This includes more sophisticated segmentation methods such as query rewriting to capture better context and hierarchical segmentation for complex documents.

Retrieval Optimization: Focuses on improving the search process itself. This includes techniques such as Hybrid Search, which combines both keyword-based and vector-based search, and Hierarchical RAG (searching for small pieces and retrieving larger pieces with broader context), which handles context at different levels.

Post-Retrieval Optimization: Post-Retrieval Optimization: Applied before retrieved parts are submitted to the LLM. These techniques aim to reduce noise. As detailed by[2], at this stage, Re-ranking reassesses the relevance scores of retrieved pieces, while Context Compression algorithms efficiently utilize the LLM's context window by extracting only the most important information.

2.1.2.3 Limitations of RAG: Bridging the Multimodal Gap

Traditional RAG architectures, no matter how optimized, have a unimodal limitation: they can only process text-based vectors. This creates a significant shortcoming when dealing with PDF documents featuring complex layouts that form the focus of my thesis.

Critical information in these documents is typically presented in the form of tables, graphs, technical drawings, and other visual elements. Traditional RAG either completely ignores this structural and visual information or attempts to present it with inadequate textual summaries. As [10]point out, it is impossible to fully capture the context of an image or the relational structure of a table with only a textual summary. These Text-Centric Limitations and the resulting Structural Information Loss prevent RAG systems from operating at full efficiency on complex, real-world data. This situation necessitates the development of Multimodal Retrieval-Augmented Generation (Multimodal RAG) systems, a new generation architecture capable of providing both textual and visual-structural context to LLMs.

2.2 Fundamental Challenges and Open Research Areas

Multimodal RAG (Retrieval-Augmented Generation) systems significantly enhance the knowledge and capabilities of Large Language Models (LLMs) by utilizing data beyond text (such as images, tables, etc.). However, there are critical challenges and intensive research topics that need to be addressed in this field.

2.2.1 Cross-Modal Alignment and Fusion

The foundation of multi-modal RAG is accurately determining how semantically relevant information from different modalities is[13]. Any error in this process leads to the LLM receiving incorrect context, thereby producing erroneous or nonsensical responses.

Risk of Misalignment: “Misalignment” occurs if the LLM fails to correctly match a text caption with the actual visual content or misuses the visual's metadata. For example, selecting the wrong part of an image as key information.

Preservation of Fine-Grained Details: Preserving fine-grained details in both text and images [14] is an important research topic. Simple object recognition is not sufficient; correctly coding the relationship, position, or action between two objects in

an image is vital for the LLM to answer complex queries. This is a major challenge, especially with information-rich images such as complex scientific diagrams or maps.

2.2.2 Evaluation Metrics Evaluating

The quality of multimodal RAG systems is more challenging than evaluating traditional text-based RAG. Traditional RAG metrics (eg.RAGAS) only assess text consistency and relevance. Multimodal systems require new metrics that measure whether the output accurately represents both the visual information received and the original query.

Multimodal Faithfulness: Measures how accurately and completely the response reflects the information in the image and the text. Exaggerating or ignoring the information in the image leads to a low Faithfulness score.

Multimodal Relevance: Evaluates how relevant the response is to the query and the relevant visual context. Newly developed metrics [15] aim to quantitatively evaluate these concepts and measure an LLM's ability to derive meaning from images rather than just its text generation ability.

The Need for Human Evaluation: Automatic metrics may be insufficient for complex, subjective outputs. Therefore, methods involving human evaluators are being developed to assess outputs for semantic and visual accuracy.

2.2.3 Privacy-Preserving RAG Architecture

Retrieval-Augmented Generation (RAG) systems inherently work with sensitive data (medical images, corporate documents, personal photographs). Therefore, protecting privacy and ensuring data integrity during data storage, retrieval, and transmission to the Large Language Model (LLM) is of critical importance. Traditional RAG systems are vulnerable to threats such as malicious queries and document extraction attacks, which jeopardise user privacy[16].

- **Active Research Area**: Proactive Security and Isolated Storage

Current research focuses on architectural-level solutions to reduce privacy risks and protect system integrity. These active research areas include:

Proactive Security Design: Developing and implementing front-loaded security strategies that apply protections before data access.

Secure and Isolated Storage: Utilizing secure isolated storage mechanisms to isolate asset-specific information. This approach reduces the exposure of sensitive information by ensuring that only necessary labels are stored in isolated, vectorized compartments[17]**.**

Attack Detection: Creating preventive query detection components that use toxicity scoring and complexity analysis to detect malicious or complex content.

- **Challenges: Deployment, Performance, and Control**

Protecting privacy, especially in applications requiring high security, creates significant challenges for the deployment and performance of LLMs and multimodal RAG components:

On-Premise Deployment Challenges: When it is preferred to run the system in a Local or Fully Closed Network (Air-Gapped) environment to ensure confidentiality and data control, significant architectural and cost challenges arise. These challenges include: The need for complex and manual procedures to update and maintain systems reduces operational efficiency, especially in closed network environments[18]. The high computational power (GPU/TPU) required for LLM and the high cost of general hardware[19] Additionally, the high cost of large-scale storage and high-performance querying required for multi-modal vector databases presents an additional challenge[20].

2.3 Application Areas and Future Directions

Multimodal RAG (Retrieval-Augmented Generation) systems have surpassed the capabilities of traditional language models by combining different data modalities (text, image, audio, video), opening the door to groundbreaking applications across a wide range of fields. The future of these systems depends on overcoming both theoretical and practical challenges.

2.3.1 Application Domains

Multimodal RAG is used particularly in areas where contextual richness and accuracy of information are critical:

2.3.1.1 Medicine and Healthcare

Multimodal RAG is transforming how doctors and researchers analyze complex medical data.

Diagnosis Support Systems: Systems combine patient radiology images (X-rays, MRI) with patient history, clinical notes, and the latest medical literature to provide faster and more accurate diagnosis recommendations[21].

Literature Search and Information Access: Researchers can quickly and reliably access information about specific biomarkers using both the text abstracts of articles and the biological diagrams and graphs they contain.

2.3.1.2 Enterprise Knowledge Management

Making institutions' scattered and heterogeneous data sources efficient is one of the main goals of Multimodal RAG.

Employee Support: When an employee asks a question about a technical malfunction, the system presents the relevant user manual (text), troubleshooting video instructions, and a diagram of the faulty part (visual) in a single context.

Financial and Legal Document Analysis: Systems can detect fraud or assess legal risk by combining tables and graphical data representations in contracts with textual provisions.

2.3.1.3 Education and Training

Multimodal RAG creates personalized and enriched learning experiences.

Interactive Course Content: While learning about a historical event (text), students can instantly recall visual artworks or maps from that period to gain a deeper understanding.

Skill Training: Engineering or medical students are provided with instant feedback and solutions by matching problems encountered in simulation environments (visual/video input) with relevant technical guides (text).

2.3.2 Future Directions

The future of multimodal RAG is focused primarily on three key areas:

2.3.2.1Adaptive Fusion

Current systems typically use static fusion methods (e.g., combining scores with fixed weights). The future will focus on systems that dynamically determine how much weight to assign to each modality based on the complexity of the query and context [22].

Example: If the query is related to a technical drawing, a higher weight will be given to the visual modality; if a legal definition is requested, weight will be given to the text modality. This ensures that the LLM directs its attention to the most relevant information.

2.3.2.2 Multi-Step and Transformative RAG

Future systems will adopt incremental and cyclical retrieval strategies instead of a one-time retrieval.

Transformative Retrieval: The model reformulates the query using the first retrieved information and initiates a second, more specific retrieval cycle. This will make a significant difference, particularly in complex tasks that require uncertain or multi-step reasoning [2].

2.3.2.3 Scalability and Efficiency

Embedded representations of multimodal data can be much larger than text-based representations. This increases the latency and memory requirements of retrieval and reranking processes in large datasets.

Lightweight Models and Quantization: Research will focus on developing "lightweight" reranking models that can quantize these large embeddings to reduce their size without compromising performance and accelerate the retrieval process[23]. This is particularly critical for local (on-device) or real-time applications.

2.4 The Gap in Literature the Rationale for the Study

The current academic literature focuses heavily on Large Language Models (LLMs) and Edge AI applications; however, the vast majority of studies present significant limitations by concentrating on external API dependencies and cloud-based solutions. This trend raises serious concerns about data security and privacy, especially when working with sensitive data. Beyond this fundamental shortcoming, literature lacks practical and comprehensive studies on multimodal systems designed to operate entirely offline and in local environments. Multimodal systems encounter difficulties in image extraction and processing due to offline resource constraints and complex data types, especially when handling non-textual data such as images within PDFs. Literature falls short in providing systematic and applicable solutions to these specific offline multimodal challenges.

This thesis aims to address this multi-layered gap in the literature and specifically seeks to optimize the reliability and performance of offline multimodal systems. Our work targets innovative solutions to technical challenges arising during the reliable and efficient extraction of image/visual data from PDFs in an architecture that locally hosts LLMs. Furthermore, it addresses the hallucination (misinformation generation) problem stemming from the misinterpretation of visual context extracted in multimodal systems, striving to develop effective solution mechanisms for this critical reliability issue. In this context, our thesis aims to make a significant theoretical and technical contribution to both Edge AI and multimodal QA system literature by not only presenting offline architecture but also proposing specialized solution frameworks for image processing and hallucination mitigation.

CHAPTER THREE

METHODOLOGY AND SYSTEM IMPLEMENTATION

3.1 Overview of the Offline Multimodal Rag Architecture

This section comprehensively explains the detailed design, implementation processes, and experimental methodologies of the Offline Multimodal Retrieval-Augmented Generation (RAG) system, which constitutes the original contribution of this thesis. Overcoming the major limitations of traditional RAG architectures, such as single-modal (text-only) information processing and API dependency on large-scale language models (LLMs), is the primary objective of this work. Developed with this goal in mind, the system aims to provide a privacy-focused information access solution that can operate locally with high performance, without requiring an external internet connection or cloud service, with a particular focus on protecting the privacy of sensitive data. This architecture is built on a dual-path data pipeline that integrates multiple data modes, including text and visual data, to enable more complete and contextually accurate retrieval of information.

The system's end-to-end workflow can be examined in three main stages that logically follow one another: 1. Data Preprocessing and Indexing, 2. Access and Hybrid Scoring, and 3. Local Production and Verification. In the Data Preprocessing stage, complex PDF documents are decomposed into textual and visual elements in a way that preserves the semantic integrity of the content. These decomposed elements are then embedded into a common vector space using Vision-Language models such as CLIP [24], enabling the measurement of semantic distance between different modalities. The second stage, Retrieval, relies on a customized Hybrid Scoring mechanism that simultaneously queries both textual and visual vectors and combines the results to form a single comprehensive context. The final stage, Local Generation, ensures that the entire pipeline runs on the device with low latency by delegating the task of synthesizing the final response from the context-enriched query to Gemma [25], a lightweight and open-source model. The subsequent subsections of this section will explain each component of the system in technical detail and provide justification for why these components were specifically chosen for an offline and privacy-focused architecture.

The development and implementation processes of the entire system were carried out using the Python programming language (version 3.10), which has become the industry standard for artificial intelligence and data science applications. The system's core architecture utilizes the PyTorch library [26]for the efficient management of deep learning operations. Chromadb [27], a scalable solution for storing high-dimensional vector representations of data and performing fast semantic searches, was chosen. For parsing source documents and extracting structural elements, fitz (PyMuPDF) [28], a reliable tool that parses text and visual components while preserving the page structure of PDFs, was used. For the multimodal encoding task, the OpenCLIP [24] model was selected because it successfully aligns visual and textual inputs in a common space. The application logic and overall orchestration of the RAG pipeline were managed using the LangChain ecosystem [29]. This includes integrating the Gemma model via the Ollama framework [30], which enables the local deployment of the LLM, specifically to ensure API-independent, offline operability, which is a fundamental constraint of the system. The choice of Ollama has been a key component of the offline architecture due to its ability to run models on the device in optimized and lightweight formats. Finally, to enable users to interact locally with this offline RAG system, the user interface (UI) was developed using Streamlit [31] due to its rapid prototyping and easy local deployment.

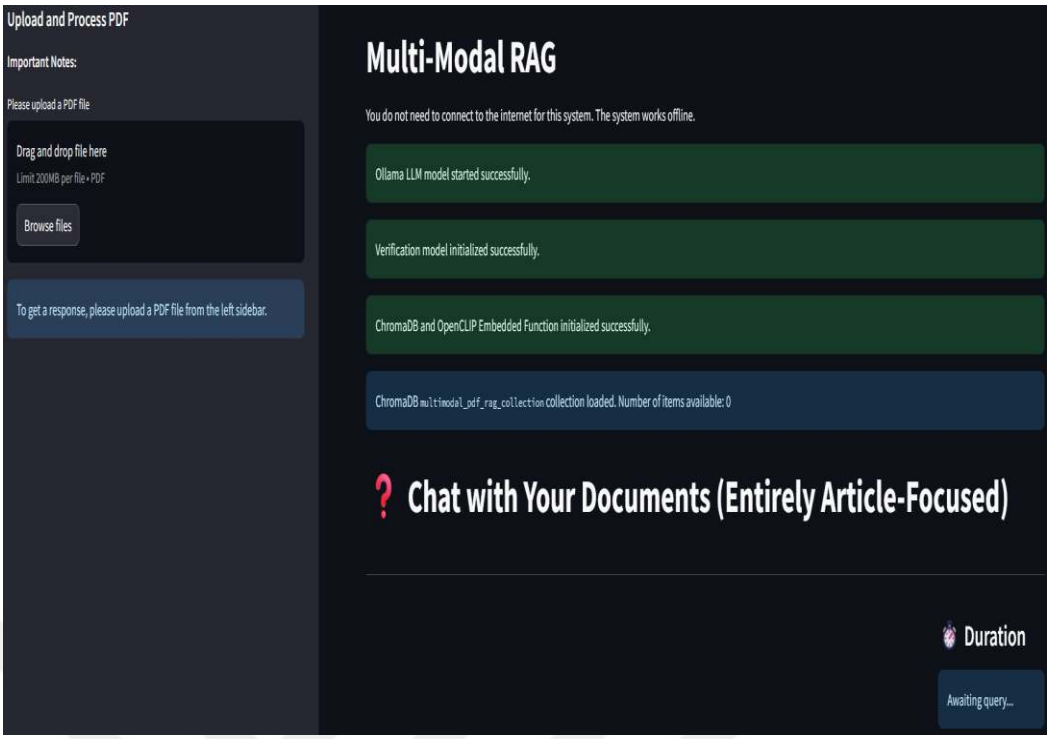

Figure 3.1: Utilizing advanced RAG (Retrieval-Augmented Generation) Architecture developed with the Streamlit library, the interface analyzes text and visual content within your PDFs to generate relevant answers and relevant image outputs for your article-based queries.

Multi-Mode RAG Interface and Usage Flow

Figure 3.1 presents an interface developed using the Streamlit library to optimize scientific literature analysis processes, showcasing a Multi-Modal Retrieval-Augmented Generation (Multi-Modal RAG) system. This system analyzes text and visual content within uploaded PDFs to generate relevant responses and evidence-based image outputs for article-based queries. The system has the capability to operate offline to support data privacy.

Data Loading and Processing: User interaction begins with the upload of article data to the system. The "Upload and Process PDF" section on the left side of the interface allows the user to submit their data source to the system.

- **PDF Upload:** The user uploads article files to the system via the "Drag and drop file here" area or the "Browse files" button. The system accepts PDF files with a limit of 300MB per file.

- **Data Processing:** After loading is complete, the system initiates the data processing flow in the background: core components such as the Ollama LLM and Verification Model are activated. Text and images are converted into vectors using the OpenCLIP Embedding Function. These vectors are stored in the ChromaDB vector database, creating the multimodal_ref_rag_collection collection.
- **Query Preparation:** Once the process is successfully completed, the user is ready to direct their questions using the "Chat with Your Documents (Entirely Article-Focused)" section in the lower right corner of the user interface.

System Preparation: The top-right section of the interface lists the readiness status (Green Bars) of the technical components indicating the system's initial state: Ollama LLM started successfully: Indicates that the Large Language Model used for response generation is ready. ChromaDB and OpenCLIP Embedded Function initialized successfully: Indicates that the embedding and database functions critical for Multi-Modal RAG are active. This structure allows the user to retrieve evidence directly from uploaded articles, ensuring reliable responses that are tightly tied to the article's

3.2 Data Preprocessing and Custom Chunking Methodology

3.2.1 End-to-End Workflow and Orchestration

Our system's operational workflow (Figure 3.2) is built on a modular structure that integrates the fundamental steps of the RAG architecture, such as Preprocessing, Retrieval, and Generation. This architecture organizes the operational logic through three main modules: Data Processing, Hybrid Retrieval, and Local Generation.The key contribution and architectural strategy of this system is that it is specifically designed to eliminate dependency on external APIs and execute the entire process on-device with low latency. This approach perfectly meets the project's offline operation requirement.

1.Data Processing Module (Preprocessing)

The Data Processing Module (Preprocessing), which forms the first stage of the system's end-to-end workflow, is the critical stage where scientific articles (PDFs) uploaded to the system are converted into analyzable data segments. At the heart of this module is a multi-modal segmentation strategy that preserves both the text and visual content of the article: The uploaded PDF is carefully divided not only into text

blocks but also into visual elements such as diagrams, graphs, and tables (Unique Segmentation).

The separated text and visual components are then processed using Multi-Modal Embedding: Meaningful vectors for images are generated using a Visual-Text model such as OpenCLIP; these vectors and their corresponding text vectors are then stored in the ChromaDB vector database, which can operate offline (Vector Storage). This comprehensive process creates a rich knowledge base that forms the foundation for the Acquisition phase and includes both textual and visual evidence.

2.Hybrid Retrieval Module (Retrieval)

The Hybrid Retrieval Module is a critical stage among the system's unique contributions, where the most relevant text and visual evidence for the user query is retrieved from the knowledge base with low latency. The workflow begins with converting the user's query into a vector (User Query Processing). Then, this vector is simultaneously queried against the text and visual vector collections in ChromaDB (Hybrid Querying). The term “Hybrid” in this stage refers to the module's ability to obtain evidence based not only on textual similarity but also on the semantic relevance of visual content. Finally, the most relevant text fragments and related visual fragments (or references) with the highest similarity scores are integrated to form a combined context window to provide input to the Local Generation Module (Evidence Integration).

3.Local Generation Module (Generation)

The Local Generation Module, which is the final stage of the system's end-to-end workflow, eliminates the need to rely on an external service by generating high-quality responses entirely on the device. At this critical stage, a combined context window obtained from the Hybrid Retrieval Module is fed into a Large Language Model (LLM) such as Ollama (or Gemma) hosted locally and running on the device. LLM uses this enriched and evidence-based context to generate a precise and accurate response to the user's article-based query, including both textual explanations and relevant visual outputs referenced in the article.

The role of the “Verification Model” specified in the system architecture comes into play precisely at this point: it acts as a Verification or Control Mechanism to ensure the reliability, relevance, and, in particular, consistency of the generated

output with the visual outputs. This model checks the response to ensure that the output is in exact alignment with the article content; this additional step maximizes reliability and evidence-based accuracy in scientific literature analysis. As a result, executing the entire RAG chain, encompassing the Preprocessing, Retrieval, and Generation steps, on-device eliminates delays caused by external network calls, meeting the system's fundamental requirement for efficient and low-latency performance.

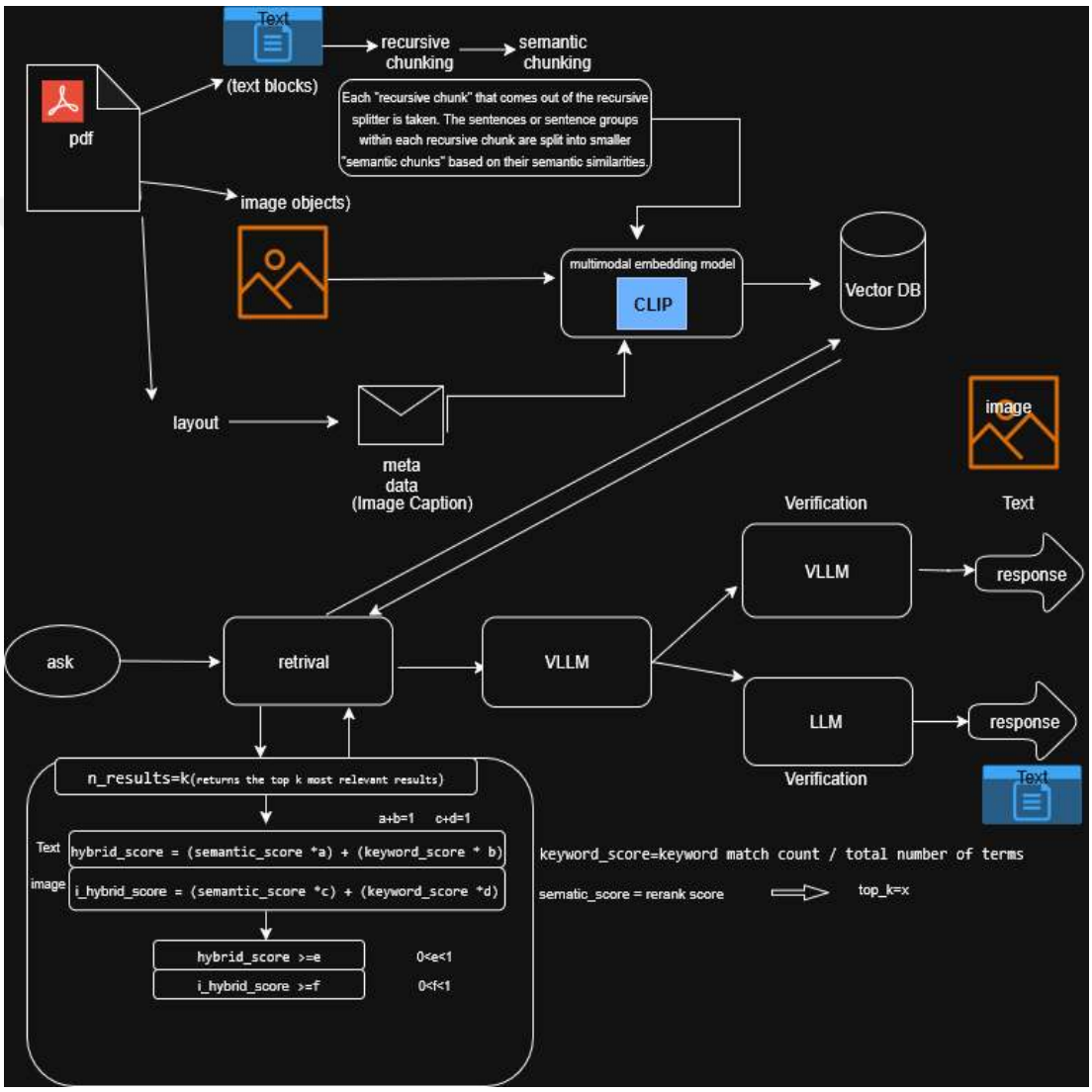

Figure 3.2: Operational Pipeline of the On-Device Multimodal RAG System. This diagram illustrates the sequential flow through the three main modules—Data Processing, Hybrid Retrieval, and Local Generation—highlighting how the OpenCLIP, ChromaDB, and locally hosted Ollama/Gemma components are orchestrated for efficient, low-latency performance.

3.2.2 Document Segmentation and Structural Extraction

Complex technical and scientific documents typically have a heterogeneous structure consisting of intertwined text, tables, figures, and mathematical formulas. Treating these structural elements as a single, unseparated text string significantly reduces retrieval accuracy. For example, the content of a table or the display of a formula can distort the semantic vector of the surrounding text, leading to irrelevant results. To solve this problem, the system uses an advanced structural extraction pipeline.

Segmentation Line

An unstructured library is used to perform document segmentation. This framework, typically managed using unstructured.partition.auto, is necessary to convert the input PDF into a list of separate elements represented by data structures such as Tables, Formulas, and various text elements. This partitioning process ensures that each section retains its own modality and structural context:

- **Isolation of Tables and Formulas:** Tables are extracted and their contents are either converted into an explanatory text string or saved as metadata, while formulas are isolated to prevent their mathematical representations from distorting the semantic vector.
- **Cleaning and Normalization:** Before the splitting process, regular expressions (import re) are used for final, detailed text cleaning and normalization**.**

Document Object Configuration

The fundamental data structure used to manage these pieces extracted along the RAG data pipeline is the LangChain Document Object (langchain.docstore.document import Document). This structure enhances the system's traceability by consistently storing both the content and its source.A concrete output of this configuration is shown in Figure 3.3, where a total of 18 Document Objects (15 Text Fragments and 3 Image Information Items) have been created from an article as an example**.**

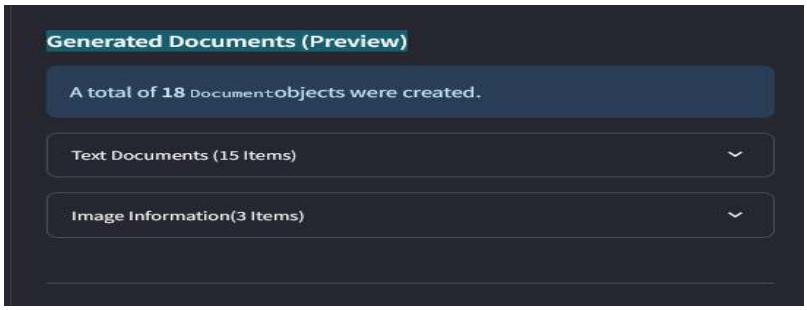

Figure 3.3: Preview of Document Objects Prepared for the RAG System, Extracted from the "Attention Is All You Need" Paper PDF. A total of 18 Document Objects (15 Text Chunks and 3 Image Information Items) have been created.

- **Text Document Object (Figure 3.4):** This object, which is the output of a text fragment, contains the main content in the page_content field and, most importantly, contains a metadata field with information such as the source page and file name from which the fragment was taken. This is vital for reliably referencing the generated responses.

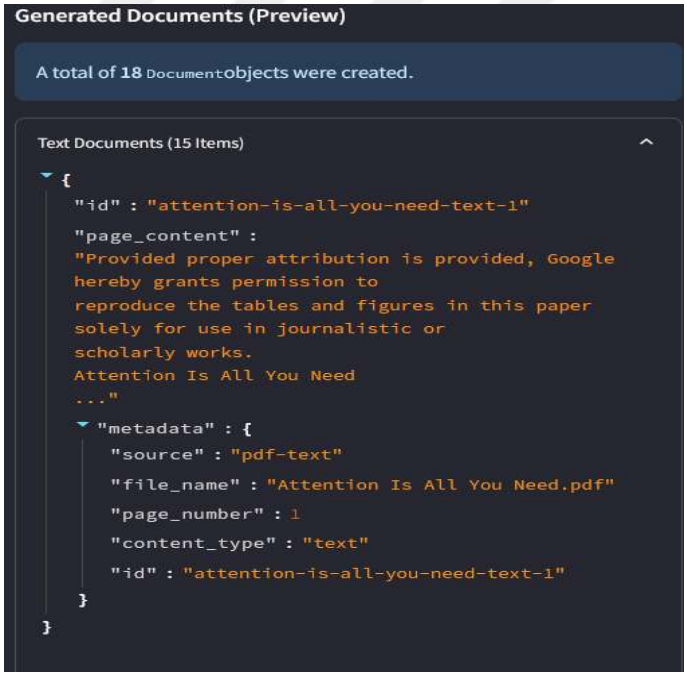

Figure 3.4: Detailed Structure of an Extracted Text Document Object from the PDF. The object contains the page content (page_content) and associated metadata that specifies the text's source (file name, page number).

- **Image Information Object (Figure 3.5):** This object, which is the output of a visual element, contains not the visual itself but its title and context in the page_content field. Again, the metadata field ensures the traceability of visual evidence by specifying the source page number and type of the image.

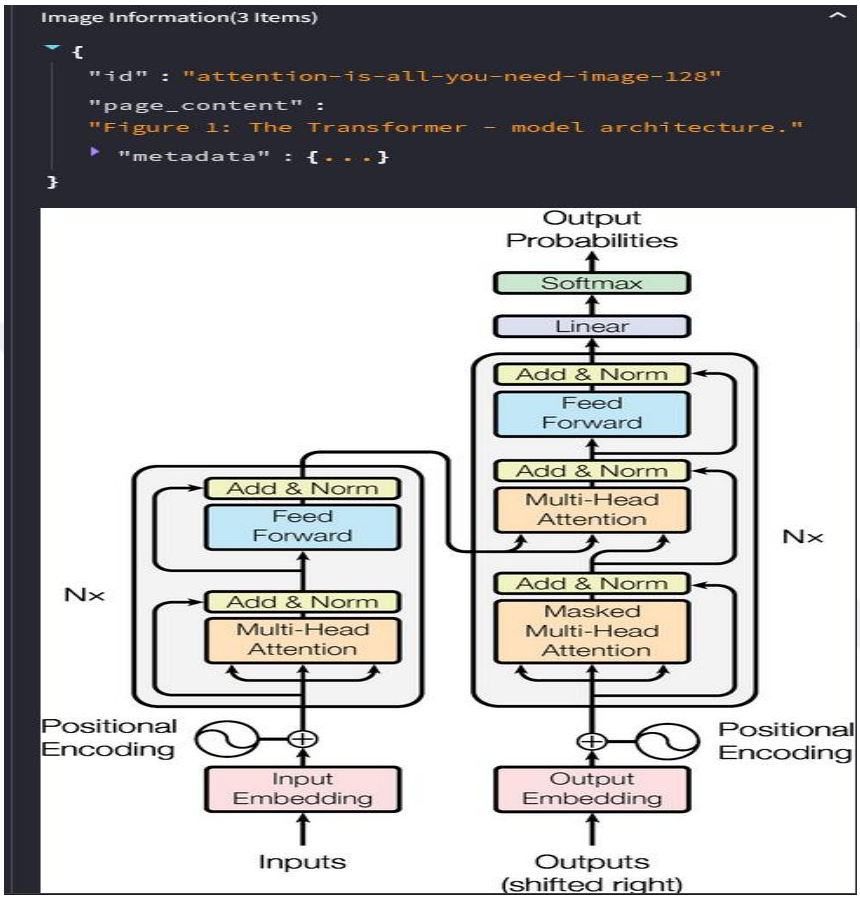

Figure 3.5: Detailed Structure and Content of an Extracted Image Information Object from the PDF. The object contains the page_content field, which includes the image's title, and the associated metadata that specifies the image's source.

This advanced segmentation and structural extraction process creates the multimodal knowledge base necessary for the Hybrid Retrieval Module to recall not only text but also visual and structural information with high accuracy.

3.2.3 Details of the Custom Recursive Chunking Strategy

The cornerstone of the Multimodal RAG system is the chunking step, which prepares the source documents for indexing while maximizing the semantic integrity

of the textual data. The strategy developed specifically for this thesis is a specialized, four-step Recursive Chunking process that relies not only on character count but also on the document's hierarchical and structural features:

Method 1: Heading-Based Sectioning

This first step in the sectioning process is managed by the find_headings_and_sections helper function and ensures the creation of the broadest contextual units. This original approach begins by dividing the entire document into paragraphs using double newlines (\n\n). Heading identification is performed iteratively by checking whether each paragraph meets at least one of three conditions: (a) The paragraph is shorter than 100 characters, (b) It is written entirely in uppercase letters, or (c) It begins with a number followed by a period (e.g., "1"). Subsequent paragraphs are then grouped under the last heading found, thus preserving the document's hierarchy by creating semantically related sections.

Method 2: Paragraph-Based Sub-Splitting

Once the content has been divided into these specialized heading-based sections, the main loop processes each section independently. The initial step for each section is to re-split its content into paragraphs using regular expressions (regex). This sub-splitting is implemented with logic similar to paragraphs = re.split(r'(?:\n\n|\n\s*\n)', section_content). The objective is to ensure that paragraphs remain the fundamental unit of context, even within a larger section.

Method 3: Sentence-Based Chunking and Boundary Control

This represents the core of the chunking process. The system iterates through each paragraph and then splits the paragraph into individual sentences using a regex pattern that looks for common punctuation marks ([.!?]). This is implemented via sentences = re.split(r'(?<=[.!?])\s+', para). Sentences are then sequentially aggregated to form the current_chunk until the predefined chunk_size limit is met. This iterative logic (adding sentences until the boundary is reached and then starting a new chunk) ensures that chunks are created as close to the specified size as possible while maintaining maximum semantic integrity.

Method 4: Chunk Overlap Strategy

To prevent a key contextual passage from being severed across two different chunks, a custom chunk overlap strategy is implemented. When a chunk is full and a new one must be started, the code looks at the end of the previous set of sentences and prepends a portion of the old chunk's content to the start of the new one. This is managed via the overlap_text variable, using logic like current_chunk = f"{overlap_text} {sentence}".strip(). This overlap ensures that related sentences remain connected across chunk boundaries, guaranteeing better context preservation during retrieval.

3.2.4 Semantic Chunking Method

This code demonstrates an approach based not only on sequential sentences but also on semantic proximity when segmenting text. The core idea behind this method is that synonymous sentences in the text form a “community,” and these communities are grouped together to form chunkings.

3.2.4.1 Sentence-Level Embedding

First, the document's content is broken down into sentences using regular expressions. Then, using the SentenceTransformer model[32], a numerical vector (embedding) is created for each sentence. These vectors mathematically represent the semantic content of the sentence. Sentences with similar meanings have their vectors closer together in the vector space.

3.2.4.2 Creating a Similarity Matrix

A cosine similarity matrix is calculated for all generated sentence vectors. This matrix represents the semantic closeness between each pair of sentences with a numerical value (between 0 and 1). A high value (greater than the threshold value) indicates that the meanings of those two sentences are similar.

3.2.4.3 Graph Creation and Community Finding

A graph (network) is created using the data obtained from the similarity matrix. Each sentence is represented as a "node" of the graph. If the cosine similarity between two sentences exceeds a specified threshold, an "edge" is added between these two nodes. The network thus created visualizes the semantic structure of the text.Next, one of the community detection algorithms, a technique used in graph theory, is used in

this code by applying the Fast Greedy method[33]. This algorithm divides the nodes (sentences) of the graph into groups with the most edge connections. These groups represent "communities" within the text that are semantically coherent and cover similar topics.

3.2.4.4 Creating Chunks and Adding Metadata

The final stage of the system's Data Processing Module involves processing structurally parsed text elements using the Semantic Chunking method and storing them in a permanent knowledge base. This process begins with large documents being first decomposed into smaller, more manageable pieces based on structural delimiters using Recursive Chunking. Then, texts extracted from complex scientific articles are processed in a way that preserves their semantic context at the highest level: Each defined community that is closest to each other and forms a coherent meaning is combined into a new text chunk. Each chunk created inherits the metadata of the original document and is additionally enriched with information such as “chunk_type: semantic”. This addition enables the targeted use of both semantically segmented segments and original document information in subsequent Retrieval stages.

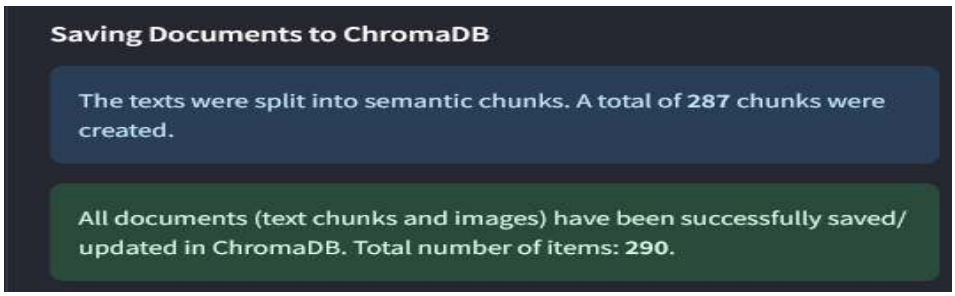

Figure 3.6: Saving the Prepared Document Objects to the ChromaDB Vector Database. The texts were split into semantic chunks (287 chunks) and all documents (text chunks and images) were successfully indexed in the database, totaling 290 items.

Following the Chunking and metadata enrichment steps, all prepared Document Objects (text segments and visual information elements) are converted into vectors using the OpenCLIP embedding function and stored in the ChromaDB Vector Database. Figure 3.6 shows the output of this process: Texts have been segmented into semantic chunkings(287 chunkings) and all documents (text segments and visual

elements) have been successfully indexed into the database; a total of 290 items have been stored. This registration process demonstrates that the system reliably creates an offline information source and establishes a data structure that enables the Hybrid Retrieval Module to perform fast and accurate retrieval.

3.3 Multi-Modal Embedding and Storage

In this module, the structured and segmented data obtained in the Data Preprocessing phase (3.2) is made queryable by the RAG system. This process involves converting the segments into a numeric vector space and indexing them into a local vector store for low-latency access.

3.3.1 Multi-Modal Embedding Strategy

After the chunking process is completed, each text and image fragment is converted into a numerical vector. Unlike traditional single-modal systems, this study uses a multi-modal embedding model that can represent both text and image data in the same semantic space. For this purpose, the CLIP (Contrastive Language–Image Pre-training) model[24], a powerful model based on the principle of contrastive learning between text and image data, was chosen. The CLIP model provides a common representation by converting both text clusters and visual data (the images themselves or their captions) into meaningful vectors.

The generated text and image vectors are stored in ChromaDB, an open-source vector database, along with their associated metadata. This storage structure enables high-performance similarity searches during the subsequent retrieval phase. The general flow of the system is arranged as Data Extraction and Metadata Creation → Chunking → Vectorization (CLIP Embedding) → Registration in ChromaDB. This approach provides a solid foundation for the system to respond effectively to both text and visual queries.

3.3.2 Vector Representation and Alignment

This stage involves creating high-quality vector embeddings for segmented data and aligning these embeddings so that they can respond to multimodal queries.

Text Embedding

Text embeddings are generated using a Transformer-based model optimized for sentence similarity, implemented via the Sentence_Transformers library (imports

SentenceTransformer from Sentence_Transformers). Due to the project's on-device and low-latency constraints, a computationally efficient Cümle-BERT (SBERT) model series was selected (e.g., the MiniLM series). The model maps text segments to a high-dimensional vector space where semantic proximity is represented by geometric distance [32]. All numerical and linear algebra operations, including the transformation and efficient computation of Cosine Similarity [34], are managed using the high-performance NumPy library (import numpy as np)[35].

3.3.3 Local Vector Indexing and Storage

A local (embedded) storage solution has been implemented to enable offline and fast querying of all vectors extracted during the multi-modal embedding process. This eliminates the external server dependency, which is a fundamental requirement of the system.

A. Vector Repository Selection

The generated text and image vectors, along with their associated metadata, are stored in ChromaDB (or a similar solution), an open-source, embedded, and offline-capable vector database. The vector storage selection was made based on criteria of low-latency read/write performance and easy integration on the device.

B. Indexing Mechanism

Approximate Nearest Neighbor (ANN) algorithms are used to efficiently retrieve stored vectors.

HNSW (Hierarchical Navigable Small World) Graph: To ensure low-latency retrieval, vectors are indexed using ANN algorithms such as HNSW. This method creates a layered graph structure that allows for fast navigation through the vector space, enabling the most relevant segments to be found within milliseconds, even among millions of vectors.

Meta-Data Storage: To increase retrieval precision, all structural and spatial metadata (Title Path, Page Number, Item Type) extracted in 3.2.2 are associated and indexed with each vector. This allows for the combination of semantic search with metadata filtering in the hybrid retrieval phase.

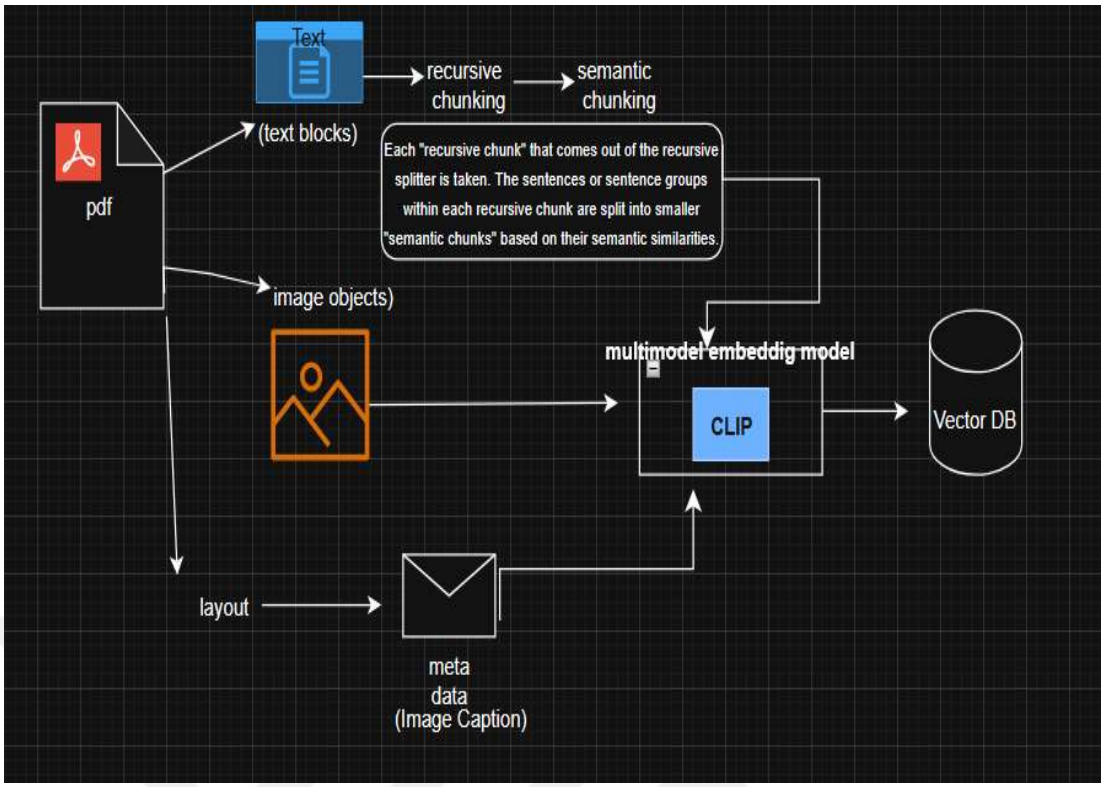

Figure 3.7: Chunking, CLIP Modeling for Multimodal Embedding, and ChromaDB Streamlining (RAG Architecture)

This stage, summarized in Figure 3.7, creates the persistent knowledge base required for the Multi-Modal RAG Architecture, which is the final output of the Data Processing Module. The semantic chunks and visual information elements obtained in the previous steps are vectorized and stored in ChromaDB to form the basis of the system's offline recall capability.

This process integrates three main components:

- **Chunking:** Text fragments enriched with metadata, obtained through Recursive Chunking and Semantic Chunking, are fed into the system as input.
- **Multimodal Embedding:** Text and visual data are converted into a high-dimensional vector space using a specialized model such as CLIP (Contrastive Language–Image Pre-training). This vectorization ensures that both textual and visual content are semantically comparable.
- **ChromaDB Stream:** All generated vectors are indexed into the low-latency, on-device ChromaDB vector database. This database forms the multimodal

knowledge base that enables the Hybrid Retrieval Module to perform fast and accurate recall.

3.4 Hybrid Retrieval Module

3.4.1 Introduction to the Retrieval Phase and Query Vectorization

The retrieval phase forms the basis of multimodal RAG systems. This phase aims to quickly and efficiently retrieve the most relevant information for a user's query from a large database. The first step in the process is query vectorization. In this process, the query submitted by the user in various formats, such as text, image, or audio, is converted into a high-dimensional vector (embedding) using a transforming model. Text Queries: Text embedding models such as Sentence Transformers are typically used for text queries. These models capture the semantic content of sentences and convert them into numerical vectors. Multimodal Queries: Multimodal transforming models such as CLIP (Contrastive Language-Image Pretraining) are preferred for visual or combined visual-text queries. CLIP represents both text and image data in the same vector space, enabling matching text queries with visual data. This vectorization process converts human language or visual information into a mathematical format that machines can understand, enabling similarity searches within a database.

3.4.2 Nearest Neighbor Search and the K Parameter

After the query vector is created, the Nearest Neighbor Search, which is the most critical component of the retrieval phase, begins. This process determines the most relevant chunks by measuring the similarity between the vectorized query and the vectors of all chunks in the database. This similarity is typically calculated using metrics such as cosine similarity. One of the key parameters used in this phase is n_result. n_result ensures that the n chunks closest to the query, as determined by the similarity search, are selected and returned as a list. The choice of the n_result value has a direct impact on the system's performance and accuracy: Small n_result Value: Returning fewer results shortens the query time and allows the system to respond faster. However, it can lower the recall rate when relevant information is not among the top results and may cause the correct answer to be overlooked.Large n_result Value: Returning more chunks increases the likelihood of relevant information being in the list and potentially provides better recall. However, this can extend query time

and lead to the inclusion of unnecessary, irrelevant information in the system. This creates additional load in the subsequent re-ranking phase and reduces the overall efficiency of the system. Therefore, determining the optimal n_result value is a matter of accuracy-latency trade-off and should be carefully adjusted according to the system's purpose.

3.4.3 Hybrid Approach and Cross-Encoder Usage for Re-ranking

The potential candidate set obtained during the retrieval phase is re-evaluated in the re-ranking phase using a multi-modal and hybrid scoring model. This approach is applied separately for both textual (text) and visual (image) data, ensuring the most relevant results are identified. The re-ranking process is based on a hybrid score combining semantic similarity and keyword matching.

3.4.3.1 The Hybrid Score (H_S) For Text

The hybrid score (H_S) for text data is calculated by combining the following components: $H\_S=(a*Semantic\_Score) +(b*Keyword\_Score)$

(Weighting Coefficients (a and b): will be determined experimentally using optimization methods (e.g., Grid Search).)

“Semantic Score During the re-ranking process, we obtain a semantic score by analyzing the deep semantic relationship between the query and the relevant document content using a cross-encoder model. This model processes the query and document as a single input, evaluating the contextual integrity and inferences of the content beyond simple vector similarity. This allows us to accurately identify the most relevant results during the re-ranking phase."

“Keyword_Score (K_S) This component evaluates the word-based relevance of the text chunk to the query. It determines how many times the keywords in the query appear in the chunk using a simple and effective formula:

K_S = (Number_of_Matching_Words) / (Total_Number_of_Terms_in_the_Chunk)”

We also apply the hybrid scoring logic of re-ranking to images, but we cannot directly combine an image and text as a cross-encoder processes text. Re-ranking for images relies on multi-modal embedding models. The process works as follows:

3.4.3.2 The Hybrid Score (i_H_S) For Image:

The visual hybrid score is calculated by combining the following components: i_H_S = c * Semantic_Score + d * Keyword_Score

3.4.3.2.1 Semantic Scoring of Visual Content

The process of assigning a semantic score to an image requires the use of a multimodal artificial intelligence model that relates visual data to a text query. This scoring aims to determine a level of relevance that goes beyond keyword matching by capturing the fundamental meaning of the content. One of the most effective applications of this process is using a pre-trained neural network architecture such as the CLIP (Contrastive Language–Image Pre-training) model. Semantic scoring mechanism with the CLIP Model. This method relies on representing both visual and text data in the same numerical space, i.e., a common semantic space. The process proceeds as follows:

A. Vectorization of the Image (Image Embedding)

In the first stage of the process, the image in question is processed by the CLIP model's image encoder. This deep learning model analyzes the raw pixel data of the image and converts it into a dense numerical vector that encompasses the objects, scenes, actions, and other contextual information within it. This vector is called an image embedding and provides a compact mathematical representation of the image's semantic content.

B. Vectorization of the Query Text (Text Embedding)

Simultaneously, the user query (such as “what is the transformer”) is processed by the CLIP model's text encoder. This component evaluates the semantic relationships and context of the words in the query, creating a numerical vector that represents the meaning of the text. This vector is also called a text embedding.

C. Semantic Similarity Calculation

In the final step, the similarity between the visual embedding and text embedding vectors, which reside in a common semantic space, is calculated. This calculation is usually performed using the cosine similarity method, which measures the directional similarity of two vectors. The closer the two vectors are to each other,

i.e., the more similarly they are oriented in the same direction, the higher the semantic relevance between them is considered to be, and the higher the score obtained.

The value obtained as a result of this method is called the semantic score of the image with respect to the query. This score forms the basis of the Semantic Score component in my Hybrid Scoring Formula and enables the ability to rank images according to their textual relevance level.

3.4.3.2.2 Keyword Score (K_S) for Visuals: The Role of Image Captions

In a multi-modal RAG system, images gain meaning not only through their visual content but also through the text that describes them. In this context, the "Figure" captions and subtitles beneath each image provide valuable keywords and contextual information about the image's content. These texts are used to generate a keyword score that complements the semantic score in a hybrid scoring model. This score evaluates whether the query contains keywords that match the "Figure" text below the image. Similar to the approach used for text, this score is calculated as the ratio of the number of matching words to the total number of words:

K_S = (Number_of_Matching_Words) / (Total_Number_of_Terms_in_the_Chunk)

(The Hybrid Score (H_S) image Hybrid Score (i_H_S) Important Note:

Text: H_S=(a∗Semantic_Score) + (b∗Keyword_Score)

-> Weighting Coefficients (a and b): will be determined experimentally using optimization methods (e.g., Grid Search).

Image: i_H_S = c * Semantic_Score + d * Keyword_Score

The weight coefficients a, b, c, and d determine whether more importance is given to the semantic content of the image or to its textual description. For example, higher weights can be given to a and c for abstract images, while higher weights can be given to b and d for queries searching for a specific object.

3.4.4 Text and Image Data-Based Elimination

In Multi-Modal Recall systems, it is critical to maximize relevance and accuracy while minimizing unnecessary computational costs. In our system, textual and visual data are evaluated through completely independent processes, without being linked to each other (Figure 3.10). This strategy requires that a candidate pass

independent screening stages applied to both its textual and visual content (if applicable) in order to be included in the final results list.

3.4.4.1 Text Elimination Stage

At this stage, potentially relevant information (documents or text fragments) is first extracted from the knowledge base to find the best response to the user query. Each extracted text fragment is called a "candidate."

Individual Evaluation: Each candidate text fragment is evaluated individually and independently in terms of its similarity or relevance to the user query.

Hybrid Scoring (H_S): The result of this evaluation is converted into a score called H_S. The calculated H_S score is compared to a predefined threshold value (e, where $0<e<1$).

Elimination Criterion: Only candidates satisfying the condition $H\_S \geq e$ successfully completes the text elimination phase. Figures 3.8 and 3.9 show the outputs of this process; multiple text segments exceeding the relevance threshold are selected to be sent to the Large Language Model (LLM) for generating the final response.

This filtering strategy minimizes the risk of hallucinations and significantly improves output quality by preventing irrelevant texts from being included in the subsequent LLM processing stage.

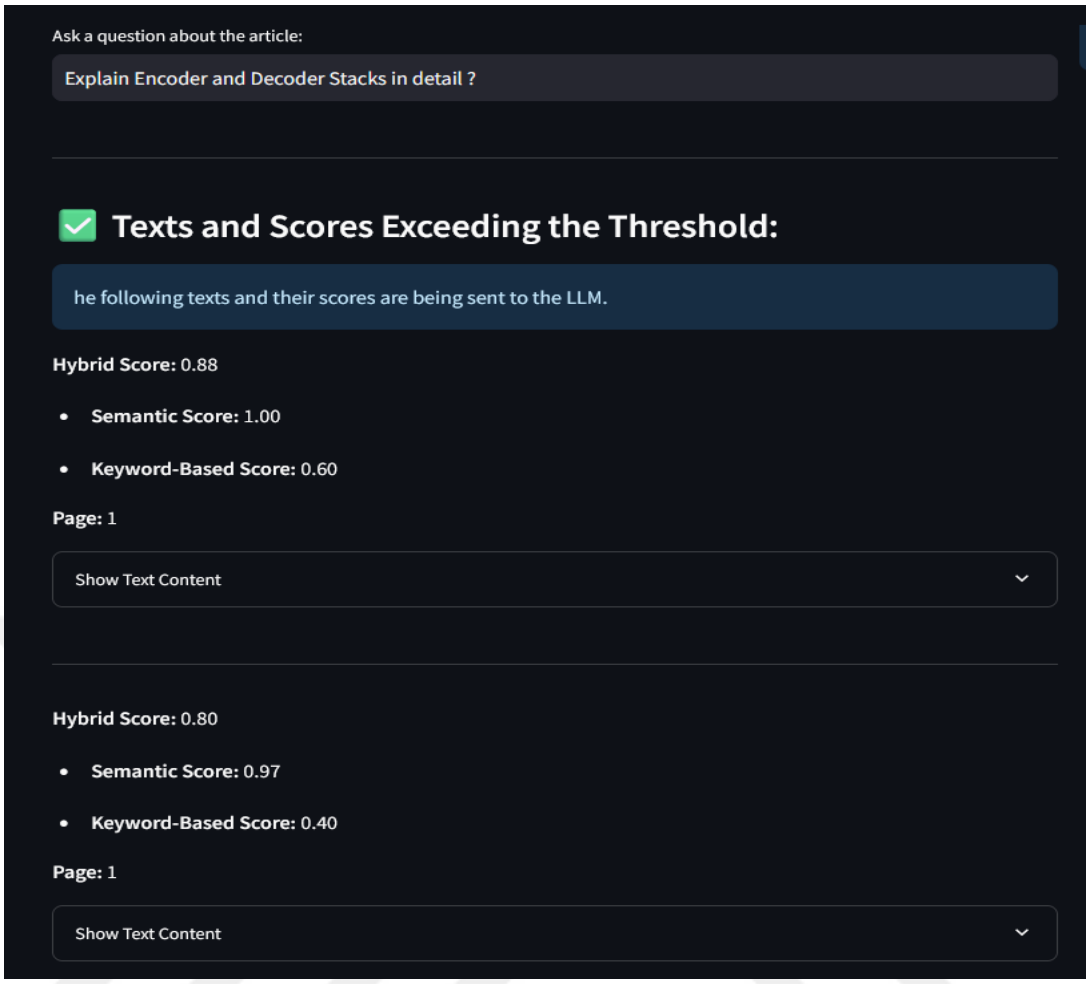

Figure 3.8: Document Retrieval Results for the RAG Query ("Explain Encoder and Decoder Stacks in detail?"). Multiple text chunks with high Hybrid Score values that exceed the relevance threshold are selected to be sent to the Large Language Model (LLM) for the generation of the final answer.

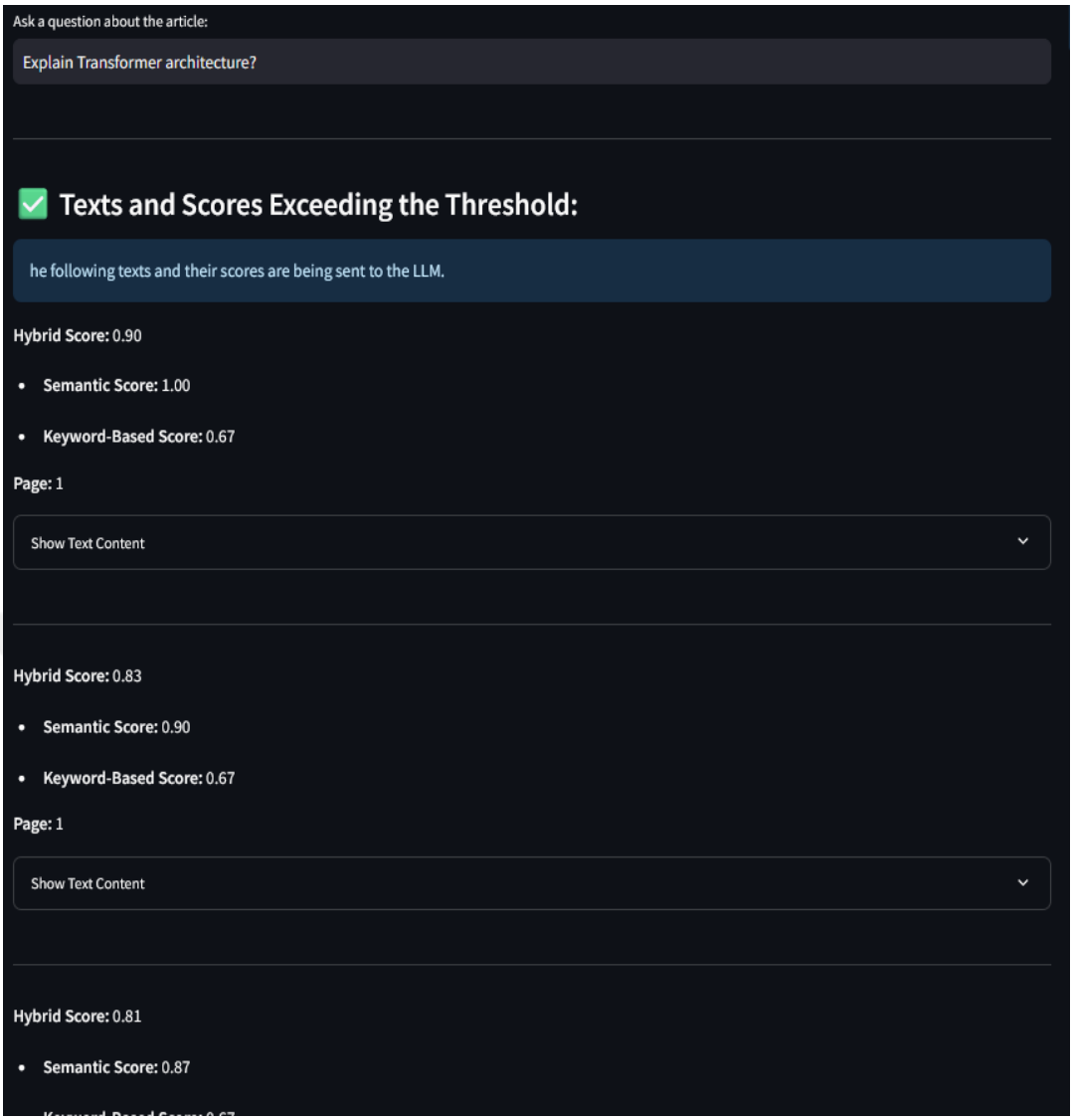

Figure 3.9: Document Retrieval Results for the RAG Query ("Explain Transformer architecture?"). Multiple text chunks with high Hybrid Score values that exceed the relevance threshold are selected to be sent to the Large Language Model (LLM) for the generation of the final answer

3.4.4.2 Visual Elimination Stage

The system uses a two-stage method to evaluate visual data with high precision during the rollback process triggered by user queries. This process is an independent filtering mechanism that minimizes unnecessary computational costs while preventing irrelevant or incorrect images from reaching the Large Language Model (LLM).

1.Two-Stage Visual Elimination Mechanism

Visual screening combines efficient operation and high accuracy through a two-step sequential process:

- **Rough Selection:** The first stage is a speed-focused approach. This "rough selection" phase is designed to quickly filter a large number of images and retrieves a potentially relevant broad set of image candidates from the knowledge base using similarity scores of pre-computed CLIP embedding vectors. This initial filtering allows the system to perform a fast scan of the database.
- **Reordered Selection:** This second stage, which is more critical and precise, re-scores each visual candidate that passed the initial screening using the i_hybrid_score, which accurately measures the relevance level to the query.In the second stage of the visual screening process, which is more critical and sensitive, called Reordered Selection, each visual candidate that passes the initial rough screening is re-scored using the i_hybrid_score, which precisely measures the relevance level to the query. This precise scoring is calculated directly by an advanced multimodal model such as CLIP (Contrastive Language–Image Pre-training), which analyzes the semantic content of the image. This scoring is a re-scoring step applied after the initial rapid similarity search, maximizing accuracy. The score calculation is performed without using any associated text data, focusing solely on the relationship between the image itself and the textual query. As shown in Figure 3.11, the image associated with the query "Explain Transformer architecture?" exhibited a high Hybrid Score during this process. Each calculated i_hybrid_score is compared to a predefined threshold value f ($0<f<1$). Only images that satisfy the condition i_hybrid_score$\geq$f are ultimately selected. This ensures consistency by sending only the text context that exceeds the relevance threshold to the LLM when no image related to the query is found (Figure 3.10).

This independent filtering mechanism ensures the system operates efficiently while preventing irrelevant or incorrect images from reaching the LLM, thereby maximizing the quality and accuracy of the generated output.

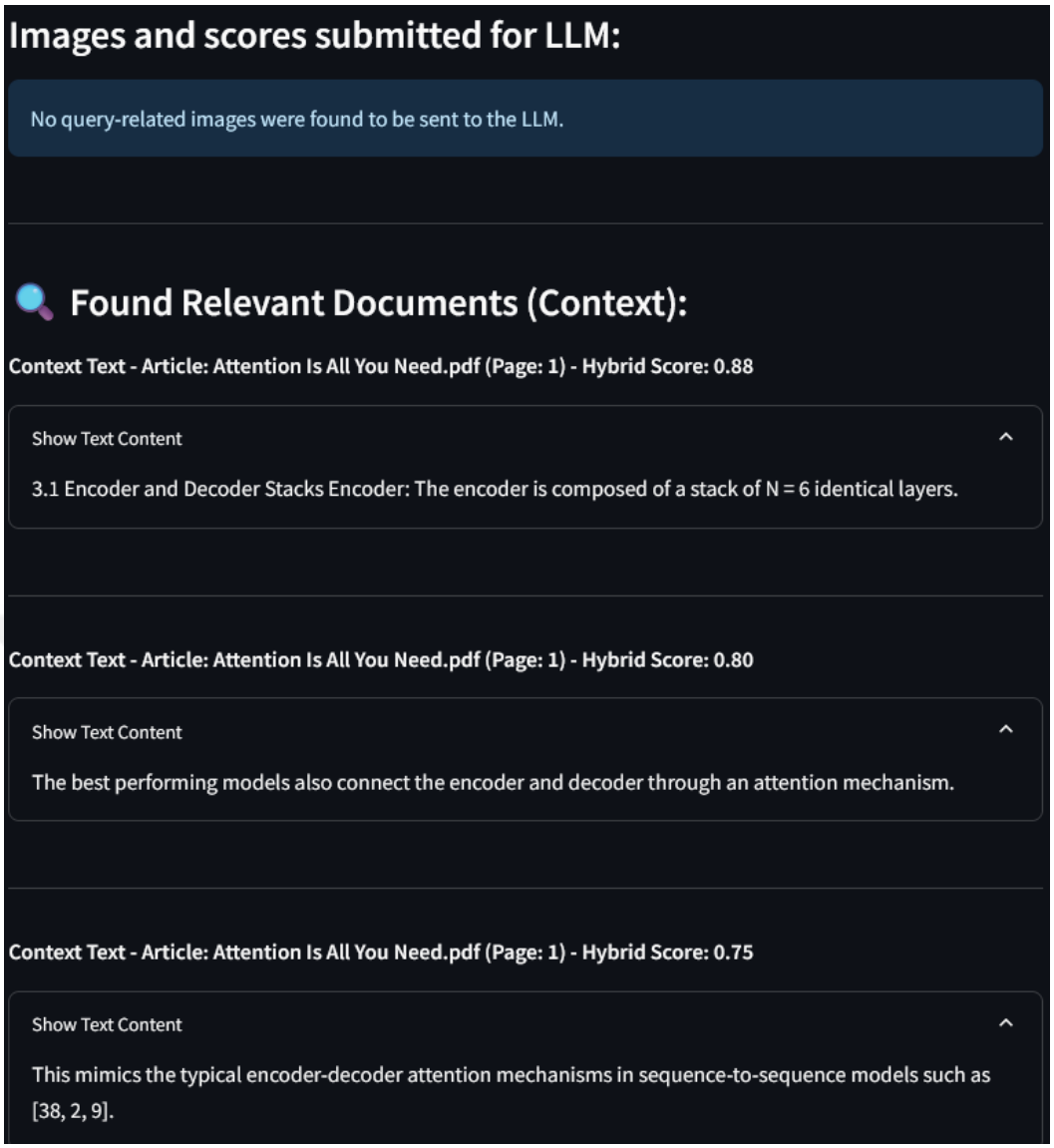

Figure 3.10: Context Submission to LLM. Although no query-related images were found, relevant text context chunks from "Attention Is You Need.pdf" with high Hybrid Scores that exceed the relevance threshold were successfully retrieved and sent to the Large Language Model (LLM) for answer generation.

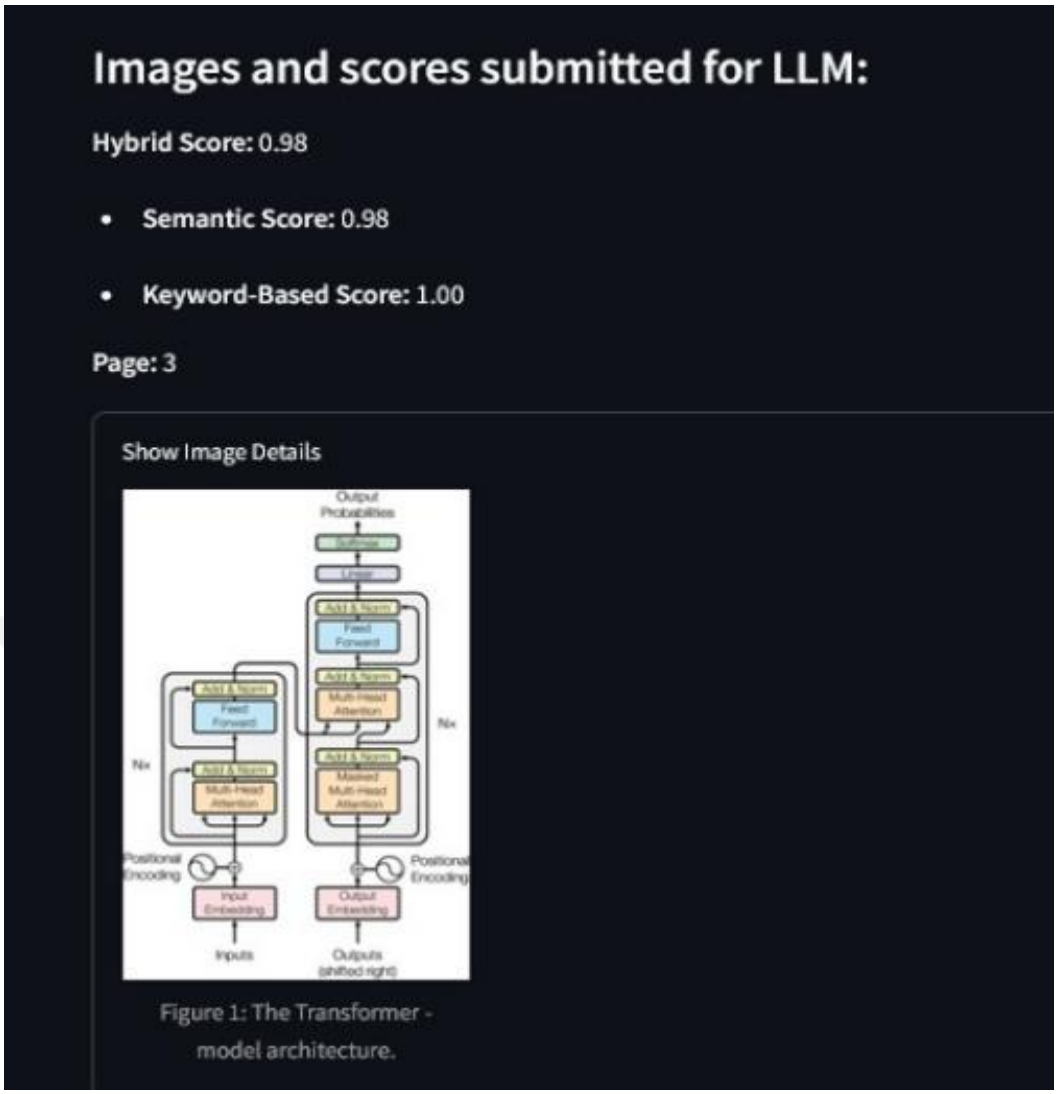

Figure 3.11: The relevant image for the query "Explain Transformer architecture?" was successfully retrieved from page 3 of the file "Attention Is All You Need.pdf". The image, representing the core architecture of the Transformer, exhibited a high Hybrid Score.

2 Keyword Extraction Method for Image Captions

To improve the retrieval accuracy of the Retrieval-Augmented Generation (RAG) system, a two-stage preprocessing filter was applied to the image captions fed into the system. The purpose of this filtering is to remove noise and directive expressions that reduce the relevance of captions in the vector space

A. Cleaning Directive Expressions with Pattern Extraction

Captions were first filtered using normal expressions to remove structural and directive expressions such as ‘request for details’ (e.g., give me more details) and ‘stay connected to the source’ (e.g., stay connected to the article). This process aims to reveal the pure content of the caption by replacing irrelevant patterns with space characters.

B. Semantic Weighting with Stop Words

The cleaned subtitle texts were then tokenized and filtered using Stop Words (STOP_WORDS), which are words with no semantic value (articles, prepositions, pronouns).The semantically weighted keyword set obtained as a result of these two steps consists of the terms that best represent the content of the subtitles and is fed into the RAG system's embedding model, ensuring that the relevance level between the image content and the user query is calculated with maximum accuracy.This meticulous preprocessing preserves the semantic essence of visual captions and directly contributes to visual retrieval performance. Indeed, as shown in Figure 3.11, queries such as “Explain Transformer architecture?” can be assigned a high Hybrid Score for the relevant image, thanks in part to this preprocessing step.

3.5 Generation Module

The Generation module takes context pieces with a high relevance level from the Retrieval module (3.4.) and creates the final, verified response to the user query. The distinguishing feature of this module is the use of multiple lightweight Gemma models in a layered and verified structure to achieve low latency and on-device operation goals.

3.5.1 Layered Generation and Verification Architecture

Instead of using a single LLM, the system uses two text-based Gemma models that work sequentially to reduce computational costs and increase response accuracy.

A. Initial Response Generation (Gemma 4B)

Restricted Generation: A specific prompt is given to the model, instructing it to base its response solely on the recalled text fragments and not to add its own knowledge.

Grounding Rule: If sufficient information related to the query cannot be found in the recalled context, the model's output is limited to a message stating “No information found related to the question.” This is a zero-hallucination strategy.

B. Text Verification (Gemma 2B)

Model: The initial response is verified in a second stage using the smaller and faster Gemma 2B model.

Process: The 2B model takes both the original query and the response generated by the 4B model and checks whether the response accurately and completely reflects the essential information within the recalled context. This stage provides an additional layer of security to eliminate unnecessary details and increase the accuracy of the final text output.

3.5.2 Multi-Modal Content Integration and Visual Verification

After the text output is generated, the retrieved segments are checked for the presence of visual data (images). Visual integration is verified in two stages using a Gemma Vision model.

A. Visual Relevance Verification (Gemma Vision 12B)

Model: Gemma Vision 12B (or a similar Vision model) is used.

Process: Each retrieved visual is compared with the original user query to verify whether the visual is truly relevant to the query.

Conditional Output:

Irrelevant Image: If the image is not relevant to the query, the output is limited to the textual response generated in 3.5.1.

Relevant Image: If the image is relevant, the image itself is marked to be added to the final response.

B. Contextual Visual Description (Post-Facto Captioning)

Integration: The relevant image is displayed on the screen.

Extended Information Search: All text fragments on the page where the image is located are scanned. The goal is to find additional explanatory text related to the image (image caption, nearby paragraph, etc.).

Final Visual Output Format: If there is additional information about the image on the page, this information is added below the image. If there is no additional information about the image on the page, the output “No information about the image” is displayed below the image.

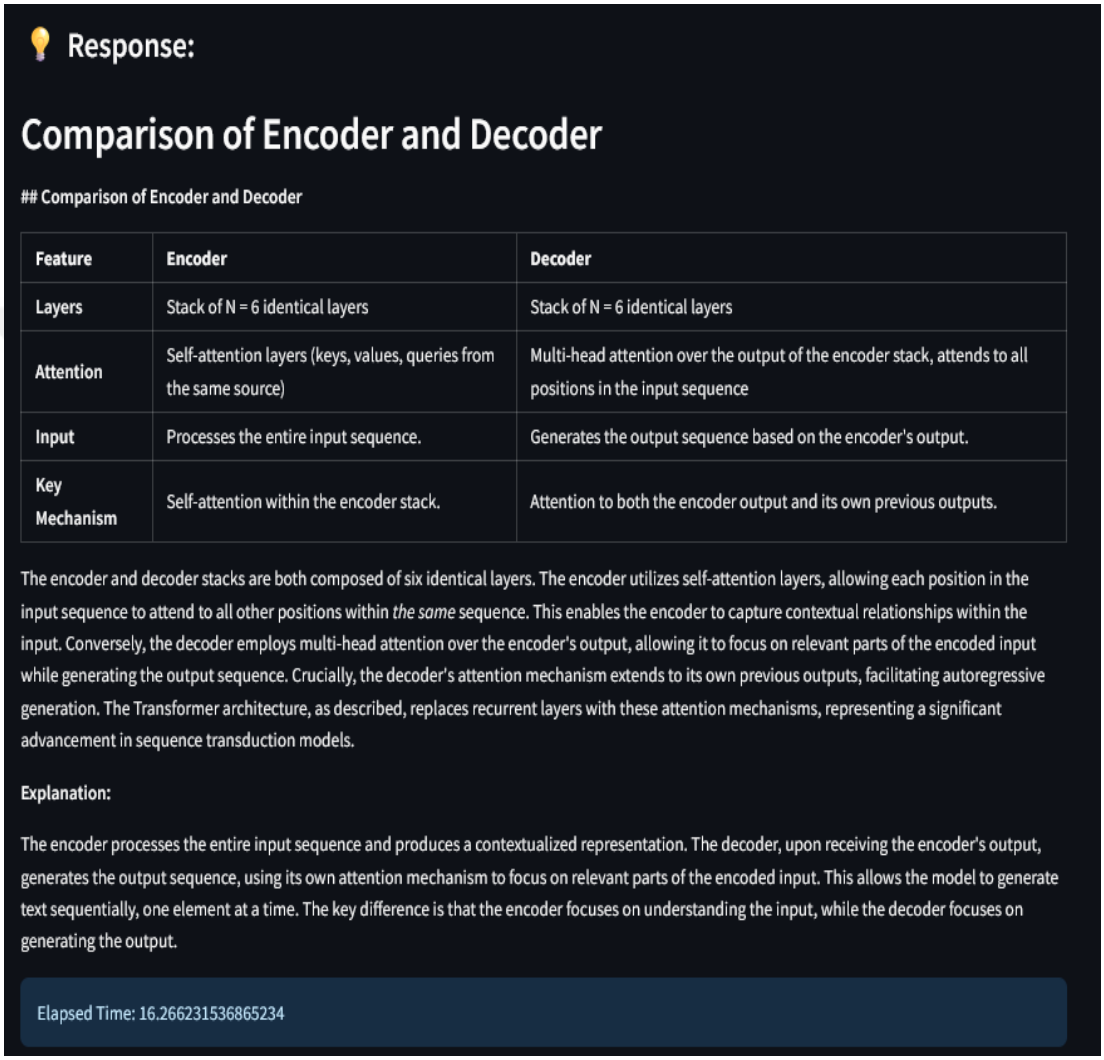

Figure 3.12: RAG Process Result: A detailed explanation regarding the query "Explain Encoder and Decoder Stacks in detail" was obtained after a 16-second RAG retrieval/generation process. No image found.

The system goes beyond the limitations of traditional RAG systems by providing access not only to textual information but also to visual content:

Textual Richness (Figure 3.12): The system successfully extracted textual data (comparative table and detailed explanation) containing the most relevant, directly

structural details from the article. This demonstrates that the model captures deep semantic context rather than merely superficial matches. Visual Support (Figure 3.13): The retrieval of the most relevant visual (Transformer Architecture Diagram) at the bottom of the response proves that the system can also access visual data. Including this visual provides a much richer learning experience than a purely textual explanation and enhances the comprehensibility of complex concepts by visually reinforcing the architecture. This is one of the strongest aspects of Multimodal RAG.

The output demonstrates that the artificial intelligence system not only provides accurate information but also does so through a transparent and efficient process:

Efficiency: The 16-second retrieval time specified in the response to the query demonstrates the high efficiency and speed of the RAG system's text processing core. While this duration indicates the ability to quickly navigate large data pools (PDF articles), it should be noted that this duration only pertains to the initial text-based rapid retrieval phase when evaluating the system's full multimodal potential. As shown in Figure 3.13, the processing and retrieval of visual data will add an additional computational load to this total duration.

Traceability and Reliability: Supporting the answer with statements such as "successfully retrieved from page 3 of the source document" provides the user with a high level of transparency and traceability, showing the source of the information. This feature is critical for academic research and fields requiring reliable information, as users can verify the source of the response.

This output example demonstrates the high performance achieved by modern Multimodal RAG systems: By combining Reliability (verification from academic sources), Comprehensiveness (integration of text and visuals), and Transparency (source attribution and time information), it provides the user with a response that is not only accurate but also enriched and verifiable. This integrated approach embodies the superior value proposition of RAG systems in the field of information access and production.

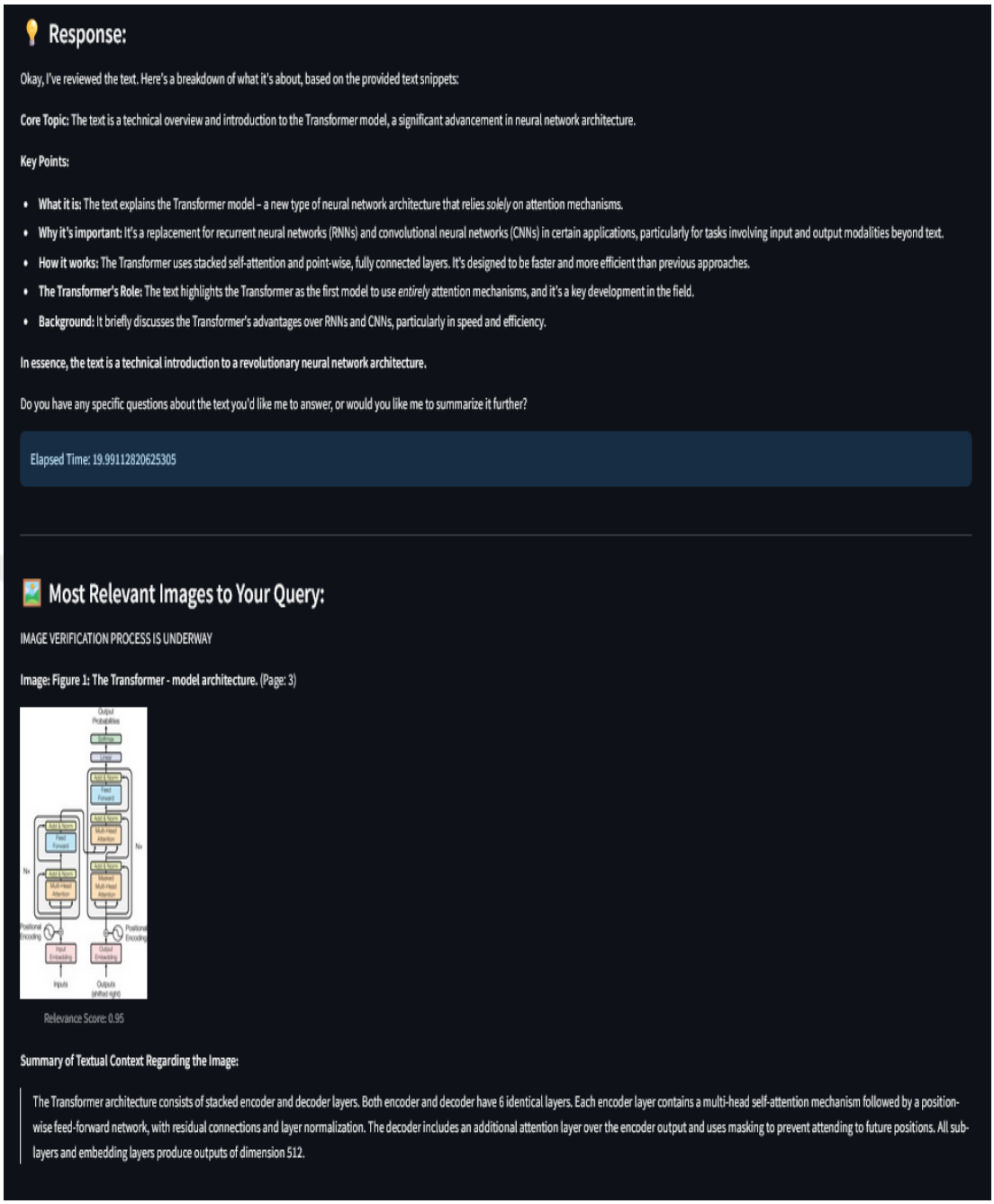

Figure 3.13: The image, successfully retrieved from page 3 of the source document, illustrates the core Transformer architecture in response to the query "Explain Transformer architecture?". The system provided the textual answer in 19 seconds, and the visual evidence (this figure) was presented at the end of the approximately 1-minute total process. The information relevant to the image is provided directly below the image.

CHAPTER FOUR

EXPERIMENTAL STUDIES AND RESULTS

4.1 Experimental Setup and Environment

4.1.1 Hardware and Software Infrastructure:

Experimental studies were conducted on a simpler system architecture that does not require high-performance external graphics processing units (GPUs) in order to highlight the efficiency and accessibility of the Recall-Assisted Generation (RAG) system in practical applications.The system's basic requirements (Embedding and Cross-Encoder model inference) have been optimized to run effectively even in a standard computer environment.

Table 4.1: Hardware Environment Specifications Used in the Experimental Study

Component	Details
CPU	13th Gen Intel(R) Core (TM) i7-13620H (2.40 GHz)
RAM	16.0 GB
GPU	NVIDIA GeForce RTX 4060

This configuration demonstrates that the proposed hybrid RAG system is a solution that minimizes dependence on high-cost hardware, particularly during the inference phase, and supports scalability for widespread use.The functionality, performance, and reproducibility of the RAG system used in the experimental study have been ensured by building it on specific software libraries and large language models. Model and library dependencies, along with version information, are detailed in Table 4.2.

Table 4.2: Basic Software Libraries and Models Used in Experimental Study

Library / Model	Role	Version
Python	Basic Programming Language	3.10.x
PyTorch	Deep Learning Framework	1.13.1
Hugging Face (Transformers)	Model Management and Download	4.25.1
ChromaDB	Vector Database	2000.4.18
Ollama	Local LLM Execution Framework	2000.1.20
CLIP Modeli	Image and Text Embedding	openai/clip-vit-base-patch32
Cross-Encoder	Re-ranking	BAAI/bge-rerankerall-MiniLM-L6-v2
Gemma 3:4b	Abstract Production LLM	A variation of Gemma with 4 billion parameters (Text and Image Input)
Gemma 3:1b	Verification LLM	A variation of Gemma with 1 billion parameters (Text Input)
Gemma 3:12b	Verification LLM	A variation of Gemma with 12 billion parameters (Image Input)
Phi-3-small-8k (3.8B)	Evaluator LLM	This model is the one with 3.8 billion (3.8B) parameters

The Python programming language, which forms the basis of this hybrid system architecture, runs on the PyTorch deep learning library. The system's recall performance has been optimized using ChromaDB for vector storage and fast nearest neighbor searches. In the Generation phase, the Ollama server layer was chosen to manage interaction with the Large Language Model (LLM). The system's multimodal capability was achieved using the CLIP Model. This model enhances the effectiveness of the visual elimination phase by creating semantic representations of image captions. In the recall process, a lightweight model called the Embedding Model (e.g., all-

MiniLM-L6-v2) is used to represent queries and text fragments, while the Cross-Encoder (re-ranker) model is used to refine relevance and improve the accuracy of results.

4.1.2 Data Sets Used

For the experimental evaluation of the Hybrid Retrieval-Augmented Generation (RAG) system, the MS COCO 2017 and SurveySum datasets were used to test its multi-modal capabilities and domain-specific text retrieval performance, respectively.

A. SurveySum (Scientific Literature Text Dataset)

This dataset is a domain-specific benchmark designed to test Multi-Document Summarization (MDS) tasks in the field of scientific literature.

Role: To provide a structured text source suitable for the retrieval module of a RAG system to summarize multiple scientific articles in a survey section.

Content: It contains 79 survey sections covering fields such as Artificial Intelligence (AI), Natural Language Processing (NLP), and Machine Learning (ML). Each survey section is matched with the full text of an article with an average of 7.38 citations.

Application and Scale: In our study, the entire dataset (79 survey sections and their matched articles) was fed into the RAG system. The survey sections were used as queries, and the cited articles were placed in the vector database as basic information chunks.

Reason for Selection: SurveySum was selected primarily because it is designed to evaluate approaches that combine recall-based selection of RAG systems with LLM-based summarization. This structure allows us to test the effectiveness of our hybrid RAG system within the current benchmarking framework of the field.

Furthermore, the nature of academic article-based Question-Answering (QA) tasks, which require long, synthesized, and reasoned answers rather than short ones, reinforces this choice. The reasons for this are as follows:

Evidence and Rationale: Academic answers should include not only the result, but also how that result was reached and what theories it is based on; short answers cannot provide this rationale.

Contextual Richness: In scientific texts, the meaning of a term can vary depending on the context; long answers preserve this context.

Multi-Document Synthesis (MDS): The SurveySum dataset is designed specifically for MDS tasks and requires the creation of a coherent summary section from information drawn from multiple sources.

In RAG applications in the academic literature, it is known that systems exhibit limited accuracy when answering short and precise (factual) questions due to chunking and context filtering difficulties arising from high-volume text. This is not a structural limitation of RAG, but rather a natural tendency.

B.MS COCO 2017 (Image and Caption Dataset)

The COCO dataset is an established benchmark in the field, primarily used to evaluate the performance of multimodal Retrieval-Augmented Generation (RAG) systems in the literature.

Role: To provide visual captions and image matches for configuring and testing the multimodal elimination module and evaluating the visual relevance of generated responses.

Content: Consists of approximately 164,000 images (total of Training, Validation, and Test sets) depicting real-world scenes and contains multiple captions generated for each image. In our work, a subset derived from this dataset was used.

Purpose of Use: Caption texts were used as semantic representations of visual data and subjected to a keyword extraction filter. Images taken from this dataset were used to test the potential of the RS (Relevance Score)-based reordering mechanism for visual elimination and improving the quality of selected images.

Methodological Rationale: In previous studies, Relevance Score (RS) and Correctness Score (CS) models were developed using 5000 human-evaluated RAG query/response examples built on a set of 1281 images taken from this dataset. The fact that our work is also derived from this established COCO benchmark ensures that our results are directly comparable with the existing literature in the field.

4.2 Experimental Results and Evaluation

This section presents the performance results of the developed Multimodal Retrieval-Augmented Generation (RAG) system. (The system architecture and

component details are discussed in previous sections.) Since our system has two main components, Text and Image, the evaluation and experimental results will be examined under two separate headings: Text Side Success and Image Side Success.

4.2.1 Text-Side Evaluation Metrics

To assess the performance of the text component, the following metrics were used to measure the success of both the Retrieval and Generation stages**:**

Table 4.3: RAG System Performance Metrics and Evaluation Areas

MetricName	Description	Evaluation Area
ROUGE-1 F1 Score	Measures the unigram (single-word) overlap between the model's generated text and the reference text. Indicates good word-level accuracy and recall.	Generation Quality(Word Accuracy)
ROUGE-2 F1 Score	Measures the overlap of bigrams (two-word sequences). Indicates how well phrase continuity and sentence flow are captured.	Generation Quality (Sentence Flow)
ROUGE-L F1 Score	Measures overlap based on the Longest Common Subsequence (LCS). Indicates syntactic fluency and the degree to which the main ideas are preserved.	Generation Quality (Syntactic Fluency)
BERTScore Average F1	Uses a large language model (BERT) to measure the semantic similarity of words in the generated and reference texts. Focuses more on meaning than simple lexical overlap.	Generation Quality (Semantic Similarity)
Ref-F1 Average Score	Likely a reference text comparison metric; the F1 score is a balance of precision and recall often used in text classification/comparison.	Generation Quality (Reference Appropriateness)

G-Eval Average Score	A method using advanced LLMs to score the output based on criteria such as coherence, fluency, and correctness (typically on a 1−5 scale).	Generation Quality (LLM-based Quality)
CheckEval Average Score	A metric that checks the extent to which the system's response correctly, consistently, and supportively utilizes information from the source articles (typically on a 0−1 scale). A score near 0.9000 indicates high verification success.	Generation Quality (Factuality/Source Usage)

4.2.2 Performance Evaluation and Optimization of the Text-Based RAG Component

This subsection empirically evaluates the performance of the Text-Based RAG component, which forms the basis of the developed Hybrid Multimodal RAG system, in providing information to the overall system. The evaluation process was carried out in two methodologically sequential stages:

In the first stage, the Semantic and Keyword Weight Coefficients for the Retrieval Mechanism were optimized to maximize the system's output quality, and the most suitable coefficient set was determined.

In the second stage, the final performance results obtained with the optimized coefficients were presented, and a comparative analysis of the system with standard methods was performed.

4.2.2.1 Evaluation Methodology and Experimental Setup

To objectively measure the output quality of the text component, the LLM-as-a-Judge methodology has been adopted to evaluate complex attributes such as contextual consistency and accuracy, where traditional metrics (BLEU/ROUGE) fall short. This setup is based on a layered structure where three different language models assume specific roles:

Generation LLM: Gemma3:4b was used as the model responsible for generating the final response.

Validation LLM: Gemma3:1b was used as the model responsible for the initial validation of the generated response and the reliability filter.

Evaluator Model (Evaluator LLM): For evaluation, Phi-3-small-8k (3.8B), an offline, small language model with strong reasoning capabilities, was used.

Evaluation Criterion: The Evaluator Model was guided by the consistency prompt structure, detailed in Table 2.1, to measure response quality. This 6-item checklist verified whether the response generated by Gemma 3B contained logical consistency and sufficient coverage. The qualitative nature of the checklist methodologically supports the system's need for deep semantic matching.

Table 4.4: Consistency Prompt Checklist Used by the Phi-3-small-8k Judge Mode

Q. No.	Evaluation Criterion (Measured Quality)	Scoring Format	Category
1	Are all the ideas, information, or arguments presented in the summary logically consistent and free of contradictions within the summary itself?	Yes/No	Internal Consistency
2	Does the summary contain sufficient critical details (such as names, dates, numerical data) to support the main topic or claim?	Yes/No	Coverage
3	Is the language and terminology used in the summary clear, precise, and professional regarding the topic, or does it contain ambiguity or repetition?	Yes/No	Language Quality
4	Does the summary cover all the key components necessary for the reader to clearly understand the main purpose and scope of the topic?	Yes/No	Coverage
5	Does the summary maintain an objective reporting tone, without unnecessary commentary, subjective expressions such as emotion or personal opinion?	Yes/No	Tone and Objectivity
6	Is the summary's sentence structure fluid and conveys the core message of a longer context with reasonable intensity (neither too sparse nor too dense)?	Yes/No	Language Quality

Metric: The average score (Avg_CheckEval_LLM) generated by the Judge Model for each query-response pair was used as the primary metric for Grid Search optimization and final performance evaluation.

4.2.2.2 Determining the Best Weighting Coefficients with Grid Search

This section presents the findings of the Grid Search applied to maximize the retrieval performance of the designed Hybrid RAG (Retrieval-Augmented Generation) system and the stability analysis of the optimal coefficients. The study was conducted on 79 survey sections covering the fields of AI, NLP, and ML.

4.2.2.2.1 Hybrid Recall Control and Fixed Hyperparameters

In the Hybrid Retrieval process, two main control parameters have been defined to manage the quality and efficiency of the context sent to the Generation stage. The positions of these parameters within the system architecture are shown in Figure 4.1. The first of these, HYBRID_SCORE_THRESHOLD (Threshold Value), is typically set to 0.35, as shown in Figure 4.1. This threshold value defines the minimum semantic + keyword score that a text chunk must have before being sent to the LLM. Maintaining this value at 0.35 increases the RAG system's precision, filtering out low-relevance text chunks and ensuring only high-confidence context is used. This is essentially the Precision Control mechanism.

The second critical parameter, MAX_CHUNKS_FOR_LLM (Maximum Number of Chunks), specifies that a maximum of 20 text chunks will be included in the LLM's context window for a query. This limitation is critical for reducing the risk of the LLM being overloaded with unnecessary information and potential hallucinations, as well as for optimizing the cost and latency of the production phase. The number 20 represents the optimal context size and serves as the Efficiency Control task

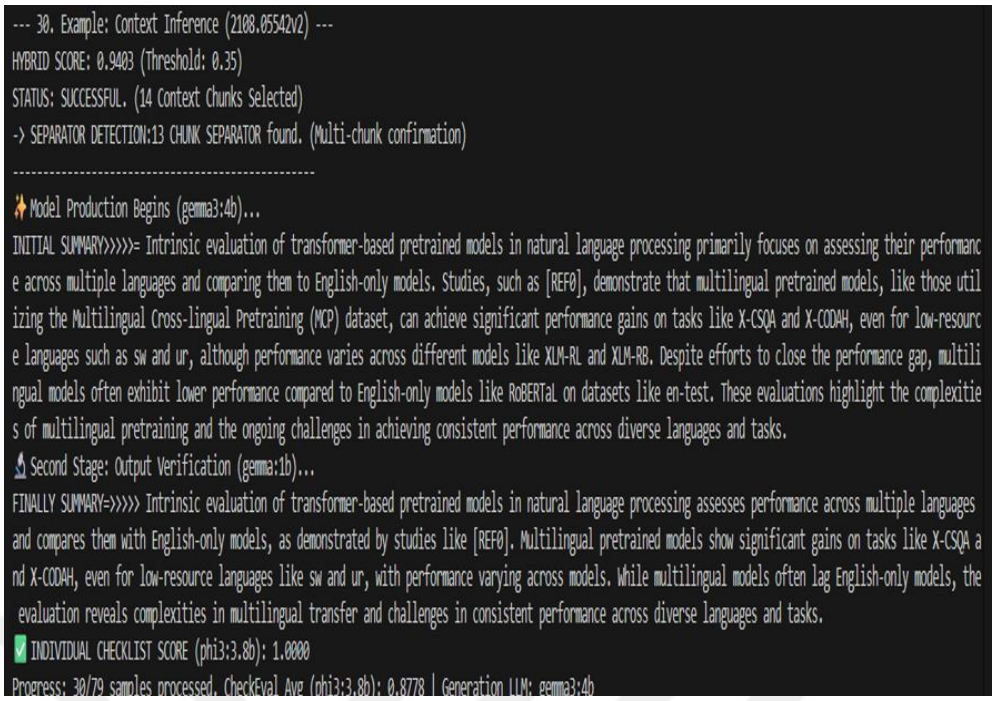

Figure 4.1: Sensitivity Control: Comparison of Hybrid Score Distribution with Threshold Value

Case Study: 30th Survey Section Performance (Figure 4.1)(ID: 2108.08542V2) to demonstrate the effectiveness of these control mechanisms, the output data for the 30th Survey Section (Article ID: 2108.08542V2) was examined:

Accuracy Control (Threshold Effectiveness): For the 30th Survey Section query, the Hybrid Recall Module selected 14 high-confidence text chunks below the MAX_CHUNKS_FOR_LLM (20) limit. This demonstrates that the Threshold filter worked quite effectively and only high-quality context passed through.

Efficiency Control (MAX_CHUNKS 20): All 14 chunks that passed the threshold value were sent to the Production Model (Gemma3:4b/1b) because they remained below the MAX_CHUNKS_FOR_LLM limit (20). This shows that the system transfers the most information without compromising context quality.

Evaluation Result (Phi3): The CheckEval assessment performed by the Phi3:3.8b model resulted in a score of 1.0000 (Maximum Score) for Survey Section 30.

This example proves that the system first selects high-quality information (Accuracy), then presents this information to the LLM with optimal limits

(Efficiency), thereby achieving a perfect (1.0000) accuracy score. The effectiveness of these control mechanisms confirms their critical role in achieving the overall average score of 0.8778.

4.2.2.2.2 Grid Search Optimization Results and Analysis

As a result of the Grid Search optimization performed, the optimal coefficients that maximize the Avg_CheckEval_LLM score have been determined:

```
================================================================================
✨ GRID SEARCH RESULT: BEST COEFFICIENTS FOUND (For Stage 2) ✨
-> Best SEMANTIC_WEIGHT: 0.80
-> Best KEYWORD_WEIGHT: 0.20
================================================================================

[OPTIMIZATION RESULTS (All Trials) - AVERAGE SCORE ONLY]
 Semantic_Weight  Keyword_Weight Avg_CheckEval_LLM
             0.8             0.2           0.8776
             0.3             0.7           0.8713
             1.0             0.0           0.8608
             0.4             0.6           0.8586
             0.5             0.5           0.8586
             0.7             0.3           0.8586
             0.6             0.4           0.8565
             0.9             0.1           0.8439
             0.2             0.8           0.8080
             0.1             0.9           0.7363
             0.0             1.0           0.7194

[INFO] Final evaluation weights set: Semantic=0.80, Keyword=0.20
```

Figure 4.2: Grid Search Results Applied to Semantic and Keyword Weight Coefficients to Optimize Recall Performance of the Hybrid RAG System (Stage 2). The Avg_CheckEval_LLM score reached its highest value of 0.8776 with a combination of 0.80 Semantic_Weight and 0.20 Keyword_Weight.

4.2.2.2.3 Experimental Setup, Scope, and Methodology

1. Data Set and Scale

Our experimental setup is based on the SurveySum dataset. This dataset consists of 79 survey sections used as queries and high-impact article texts, which have received an average of 7.38 citations and are stored in the Vector Database as recall sources. MAX_OPTIMIZATION_SAMPLES is set to 79, and each trial of Grid Search covers these 79 unique queries.

a. Hybrid Recall and Optimization

Optimization focuses on a hybrid methodology that combines semantic and keyword scores to improve the quality of the RAG system's recall results. The optimization process was performed by testing the sum of the Semantic_Weight and Keyword_Weight coefficients between 0.0 and 1.0, such that the total equals 1.0, using the find best_hybrid_weights_grid_search function.

b. Stability and Statistical Reliability

To test recall stability, each weight combination was repeated 5 times (NUM_RUNS_PER_COMBO=5). The average of the results obtained from these iterations was accepted as the final Avg_CheckEval_LLM score.

c. Chunking Strategy Parameters

As a factor directly affecting recall quality, two different Chunking approaches were evaluated to optimize the structural organization of documents:

- Advanced_Recursive_Chunking: A recursive approach to preserve structural organization and keep related text adjacent. This method uses a large chunk_size (3000) and high chunk_overlap (600). This large 3000-token window aims to preserve the broad context of each survey section in the 7.38 reference article, while the 600-token overlap ensures continuity of information flow between consecutive chunks.
- Semantic_Chunking: Aims to retrieve text segments with thematic coherence by grouping segments based on semantic similarities across texts. In this method, the minimum similarity threshold (min_similarity_threshold) is set to 0.7. This high threshold maximizes the thematic consistency of the merged segments, ensuring that the texts retrieved from the Vector Database contain only the essential information relevant to the query.

d. Large Language Models (LLMs) Used

To compare the performance of the RAG system across different production capacities and objectively validate the metric results, three different compact and open-source Large Language Models (LLMs) were used.

In the study, a sequential production chain was established to improve output quality. In this chain, the Gemma3:4b model (4B) synthesizes source text fragments

retrieved from the RAG system to produce an intermediate summary (draft) text. This intermediate output is then passed to the next stage in the chain and taken as input by the Gemma3:1b model (1B). Gemma3:1b finalizes this draft to produce the final summary (response) text, which is both presented to the user and subjected to an evaluation stage. Thanks to this improvement mechanism, the output of the more compact Gemma3:1b aims to increase production quality by leveraging the contextual richness provided by Gemma3:4b.

The Phi3:3.8b model (3.8B) was used as the primary evaluator to validate and score this final output. The Phi3:3.8b model is completely independent of generation models such as Gemma3:4b and Gemma3:1b. Phi3:3.8b generates its own independent output (generation) to determine the consistency and accuracy score of the final summary relative to the source texts. This objective evaluation approach is based on the Avg_CheckEval_LLM scores generated by the production models.

2. Optimization Findings and Performance Analysis

As a result of the Grid Search optimization performed, coefficients that maximize hybrid recall performance and elevate the Avg_CheckEval_LLM score to the highest level have been determined.

According to these optimization findings, detailed in Figure 4.2, the highest average CheckEval score (0.8776) was achieved with the combination of hybrid coefficients Semantic_Weight set to 0.80 and Keyword_Weight set to 0.20. This finding clearly shows that semantic relevance should have four times more weight than keyword relevance in the success of the Hybrid RAG system. Therefore, it confirms that the hybrid model prioritizes deep topic and context relevance between the query and the source text over superficial matches, and that this approach maximizes recall quality.

Table 4.5: Hybrid RAG Metric Values (Semantic 0.8 / Keyword 0.2 combination)

Category	**System A**	**System B**	**System C**	**System D**	**System E**	**Average**	**Standard Deviation (σ)**
Total Number of Articles	79	79	79	79	79		
Generative LLM	Gemma3: 4b:1b	Gemma3: 4b:1b	Gemma3: 4b:1b	Gemma3: 4b:1b	Gemma3: 4b:1b		

Evaluate and Score with LLM	phi3:8b	phi3:8b	phi3:8b	phi3:8b	phi3:8b		
Hybrid Weights (Semantic/Keyword)	0.8 / 0.2	0.8 / 0.2	0.8 / 0.2	0.8 / 0.2	0.8 / 0.2		
ROUGE-1 Average F1 Score	0.4781	0.4750	0.4818	0.4970	0.4873	0.48384	0.0080
ROUGE-2 Average F1 Score	0.2808	0.2866	0.2935	0.3113	0.2986	0.29496	0.0112
ROUGE-L Average F1 Score	0.3541	0.3534	0.3656	0.3856	0.3684	0.36766	0.0125
BERTScore Average F1 Score	0.7321	0.7329	0.7351	0.7400	0.7368	0.73538	0.0028
Ref-F1 Average Score	0.9089	0.9032	0.9127	0.8614	0.8861	0.89386	0.0189
G-Eval Average Score	4.57	4.53	4.61	4.59	4.54	4.568	0,0299
CheckAvail Average Score	0.8987	0.8924	0.9030	0.8460	0.8734	0.8827	0,02095

Table 4.6: Pipeline Comparison with Traditional and Large Language Model-Based Metrics from the "Ask, Retrieve, Summarize" Paper [36]

Table 1
Comprehensive performance comparison of the evaluated pipelines based on traditional metrics (ROUGE, BERTScore, Ref-F1) and LLM-based metrics (G-Eval, CheckEval).

Pipeline	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	Ref-F1	G-Eval	CheckEval
Pipeline_1	0.42	0.08	0.19	0.57	0.64	3.1	0.61
Pipeline_2	0.49	**0.10**	0.23	0.59	0.72	4.0	0.76
XSum	**0.51**	**0.10**	**0.24**	**0.62**	**0.76**	**4.2**	**0.97**

powerful models for both tasks can lead to skewed results, as LLMs like GPT-4 tend to favor their own outputs due to egocentric biases [23]. Notably, the *SurveySum* paper does not specify whether identical models were used in their study for both tasks.

4.2.2.2.4 Performance Evaluation of Modular RAG Lines

1.Comparison of RAG Architectures

This analysis compares the performance of two different Retrieval-Augmented Generation (RAG) architectures serving the same purpose (the challenge of summarizing scientific publications): Five Modular Systems (A-E) tested by us and the XSum RAG Pipeline. The comparison is conducted using both traditional metrics (ROUGE, BERTScore) and LLM-based verification metrics (Ref-F1, G-Eval, CheckAvail).

Key Difference: Systems A-E use an offline and fixed Generative LLM (gemma3:4b:1b) and Verification LLM (phi3:8b). In contrast, the performance metrics of the XSum line are generally based on online models such as GPT-4 and phi3:8b(online). This situation attributes the fundamental reason for performance differences, particularly in LLM-based metrics such as CheckAvail, to both architectural designs (recovery/generation) and the capacity differences of the LLM models used (online/offline). The goal is to determine which architecture or optimization excels in which performance metrics.

2. Comparison of Traditional Metrics (ROUGE and BERTScore)

A. Semantic Quality (BERTScore)

A. Semantic Quality (BERTScore) Our RAG systems demonstrate a clear superiority in semantic quality compared to the XSum baseline. According to the data in Table 4.4: RAG Metric Values, the average RAG system (0.73538) significantly exceeds the XSum baseline score (0.62). The RAG system's overall performance (average 0.73538) performs better than XSum. This indicates that our RAG lines produce semantically more consistent outputs with reference summaries.

B. Word Overlap (ROUGE)

In the ROUGE metrics, there is a ROUGE-1 (Unigram): The XSum line (0.51) narrowly surpasses the average RAG system (0.48384) in overall word overlap. ROUGE-2 and ROUGE-L (N-gram and Long Overlap): According to Table 4.4, our systems demonstrate overwhelming superiority in these metrics. The average Hybrid RAG system (0.29496 ROUGE-2 and 0.36766 ROUGE-L) approximately doubles XSum's scores (0.10 ROUGE-2 and 0.24 ROUGE-L). This indicates that Systems A-E share longer and more sequential word sequences with reference summaries, likely producing less abstract or more specific outputs.

3. Comparison of LLM-Based Evaluation Metrics

These metrics measure system quality from the perspective of the LLM used (usability, accuracy). General LLM Evaluation (Ref-F1 and G-Eval): According to Table 4.4, our RAG systems have an advantage over XSum in general LLM evaluation: The average RAG system (0.89386 Ref-F1 and 4.568 G-Eval) significantly outperforms XSum's scores (0.76 Ref-F1 and 4.2 G-Eval). Factual Accuracy (CheckAvail) - Detailed Comparison: This metric highlights the clear superiority of

the XSum RAG pipeline, which uses an online model, compared to your systems that use an offline model: The XSum pipeline achieved the highest score of 0.97, demonstrating much better performance than the Five Modular Systems (A-E) listed in Table 4.4. While the average modular system has a CheckAvail score of 0.8827, XSum surpasses this score by a significant margin of 9% (0.97−0.8827=0.0873). **Comment:** The significant difference in CheckAvail scores undeniably reflects the superiority of the advanced verification and broad general knowledge base provided by online models (e.g., GPT-4). However, thanks to our offline model and the optimizations we implemented with the RAG architecture, our scores have come remarkably close to the accuracy achieved by our online competitors. This clearly demonstrates how much factual accuracy depends not only on the sheer capability of the Large Language Model (LLM) used, but also on the robust RAG architecture, which relies on efficient retrieval and data augmentation strategies. Despite using an offline model, this competitive result demonstrates the RAG architecture's potential to close the real-accuracy gap even in closed systems.

4. Internal Consistency of RAG Systems (Standard Deviation)

The standard deviation (σ) values in Table 4.4 examine the performance spread across 5 systems (A-E): Lowest Spread (High Consistency): BERTScore (0.0028). The systems demonstrate nearly flawless consistency in semantic quality. Highest Variation (Low Consistency): G-Eval (0,0299) and CheckAvail (0.02095). These metrics are the most responsive to minor changes in system configurations, indicating that these validation scores represent the most sensitive components of the RAG pipeline.

5. Conclusion

This RAG comparison reveals a clear balance reflecting the potential of offline optimizations and the impact of LLM preferences based on the performance metrics obtained: Our systems (A-E): Despite using an offline, resource-constrained LLM (Gemma3:4b:1b), they excel at producing high semantic similarity (BERTScore) and overall LLM quality (G-Eval). In terms of real-world accuracy, the average RAG system achieved a CheckAvail score of 0.8827, coming significantly close to XSum. XSum Line: Thanks to the superiority of online models and architectural

optimizations, it is clearly excellent at maximizing real-world accuracy (CheckAvail: 0.97).

Consequently, if semantic depth and overall LLM preference are prioritized, our hybrid systems should be preferred. If real-world accuracy is the most critical requirement, the architecture and online model infrastructure used by XSum appear to be more effective. However, considering that XSum, which scored 0.97, is an online system, the fact that our offline Hybrid System C approaches this level with a real-world performance of 0.9030 is an exceptionally competitive achievement for a resource-constrained system. This demonstrates that we offer a cost-effective and robust alternative for closed systems.

4.2.3 Image-Side Success and Evaluation

This section analyzes the performance of the Image Filtering Module of the Multimodal RAG system, developed using image and caption data from the COCO 2017 dataset, a well-known benchmark in the field. The aim of our study is to test the potential of a Hybrid Score (HS)-based reranking mechanism in image scanning and improving the quality of selected images. The developed Image Scanning Module uses a hybrid approach that evaluates visual content not directly at the pixel level but through the semantic representations (metadata) of the images. This methodological choice enables the system to select visual evidence that is both fast and semantically accurate.

The Hybrid Score (HS) represents the optimized metric our system uses to calculate the final relevance between the query and the candidate images. Before calculating the HS, the number of candidate images is limited to the 300 most relevant candidates (k_candidates = 300) from the database using the CLIP Model. This pre-selection step is implemented to reduce the computational cost of the Cross-Encoder and Hybrid Score calculations. HS increases the reliability of visual relevance estimation by weighting additional semantic components, such as the Contextual Placement (CE) Score and Keyword (KW) Score, in addition to the CLIP Score based solely on visual-semantic matching across these 300 candidate images. HS's final calculation formula is HS=(CE×0.95)+(KW×0.05). In this formula, the Contextual Embedding (CE) score, the main driver of relevance, is given high priority, while the Keyword (KW) Score is kept low at 0.05 to use the KW score as a discriminator. This

low weighting plays a critical role in selecting the most accurate result by fine-tuning the best results and confirming semantic consistency in cases where the CE score is already high and the images are semantically similar.

Evaluation: The focus is on the effectiveness of the developed Hybrid Score Model compared to the performance of Contextual Embedding (CE) Score and Keyword (KW) Score alone, and the top 5 results (TOP_N = 5) for each query out of 300 re-ranked candidates are presented as the final output. Model performance is measured through binary classification metrics (Accuracy, Precision, Recall, F1) obtained using Human Judge data as the final criterion.

	Main Query	Matching Description	Image	(CLIP SCORE	(CE SCORE)	(KW SCORE)	(HYBRID SCORE)
2	a cat sitting on a wooden chair	A cat that is sitting on a wooden chair.	000000437537.jpg	1,00000	1,000	1,000	0,999999630
3	a cat sitting on a wooden chair	there is a cat sitting on a wooden chair	000000096097.jpg	0,51923431	0,99999950	1,000	0,999999511
4	a cat sitting on a wooden chair	A cat sitting on top of a wooden chair.	000000309852.jpg	0,30373747	0,99886660	1,000	0,998923290
5	a cat sitting on a wooden chair	There is a cat sitting on a chair.	000000561599.jpg	0,60714868	0,99999976	0,96428570	0,998214035
6	a cat sitting on a wooden chair	A cat that is sitting on a chair	000000182995.jpg	0,11763111	0,99999976	0,96428571	0,998214035
7	a dog chasing a frisbee in the park	A dog is chasing a frisbee in a park.	000000055296.jpg	0,14290325	1,00000000	1,00000000	0,999999988
8	a dog chasing a frisbee in the park	A dog chasing a cloth frisbee in the park.	000000025989.jpg	0,2240564	0,99385166	1,00000000	0,994159091
9	a dog chasing a frisbee in the park	A dog in a park is playing frisbee	000000572825.jpg	0,5710547	0,99999940	0,84375000	0,992186952
10	a dog chasing a frisbee in the park	A dog catching a frisbee in the air.	000000286932.jpg	0,0724440	0,99969070	0,84375000	0,991893637
11	a dog chasing a frisbee in the park	A dog in a field catching a frisbee	000000553396.jpg	0,0903482	0,99960760	0,84375000	0,991814721
12	a close-up shot of a red rose	a close of of a rose and other flowers in a vase	000000156915.jpg	0,5131878	0,99107885	0,75000000	0,979024923
13	a close-up shot of a red rose	a close up of a vase with a flower in it	000000127543.jpg	0,6386213	0,99348220	0,70129870	0,978873014
14	a close-up shot of a red rose	THERE IS A VASE WITH A ROSE INSIDE OF IT	000000143010.jpg	0,1085544	1,00000000	0,51428571	0,975714274
15	a close-up shot of a red rose	A close up view of petals on a flower,	000000257461.jpg	0,5615242	0,98477200	0,71428571	0,971247690
16	a close-up shot of a red rose	a close up of some flower peddles	000000257461.jpg	0,5615243	0,97964704	0,73469387	0,967399352
17	a person riding a bicycle on a sunny street	A person on a bicycle riding down the street.	000000128920.jpg	0,0966307	0,99994767	1,00000000	0,999950278
18	a person riding a bicycle on a sunny street	A person on a bike on a city street.	000000386279.jpg	0,6890259	0,99902713	1,00000000	0,999075758
19	a person riding a bicycle on a sunny street	a person riding a bike on a city street	000000281227.jpg	0,2834702	0,99889380	1,00000000	0,998949099
20	a person riding a bicycle on a sunny street	A Person riding a bike down a street.	000000242592.jpg	0,0933770	1,00000000	0,96428571	0,998214274
21	a person riding a bicycle on a sunny street	a person on a bike rides down a street	000000543193.jpg	0,2826289	0,99995580	0,85714285	0,992815110
22	a large clock tower in a city square	a clock tower in the middle of a city	000000298644.jpg	0,8341624	0,99980430	1,00000000	0,999814081
23	a large clock tower in a city square	A clock tower somewhere in the middle of a city.	000000292868.jpg	0,8255188	0,99962515	0,88800000	0,994043850
24	a large clock tower in a city square	a tall clock tower near a building	000000490615.jpg	0,8067615	0,99955680	0,84000000	0,991578941
25	a large clock tower in a city square	A picture of a historic clock tower.	000000041246.jpg	0,1319166	0,99833370	0,84000000	0,990417008
26	a large clock tower in a city square	A very tall clock tower towering over a city.	000000159537.jpg	0,4738149	0,99953973	0,81333333	0,990229395
27	a pile of yellow bananas on a table	A big pile of ripe yellow bananas on a table.	000000450961.jpg	0,1358105	0,99933845	1,00000000	0,999371517
28	a pile of yellow bananas on a table	a bunch of bananas on display on a table	000000501307.jpg	0,5370092	0,99999905	0,97222222	0,998610205
29	a pile of yellow bananas on a table	A bunch of bananas on a table top.	00000473427.jpg	0,2117452	0,99880650	0,93750000	0,995836139
30	a pile of yellow bananas on a table	a big pile of bananas that are on a table	000000436435.jpg	1,0000000	0,99999595	0,87499999	0,993746114
31	a pile of yellow bananas on a table	many bunches of bananas on a table	000000375198.jpg	0,7258852	0,99858960	0,89285714	0,993302933
32	a white flower with green leaves	A white case that has some flowers in it.	000000344094.jpg	0,6992000	1,00000000	0,40000000	0,970000000
33	a white flower with green leaves	white flowers in a vase with arranged leaves	000000217183.jpg	0,7185000	0,94490000	0,90000000	0,942600000
34	a white flower with green leaves	A vase that has a flower inside of it.	000000343445.jpg	0,2729000	0,87010000	0,60000000	0,856600000
35	a white flower with green leaves	a vase that has some flowers in it	000000089258.jpg	0,9168	0,8598	0,225	0,8281
36	a white flower with green leaves	a vase containing white flowers next to a branch	000000377679.jpg	0,2837	0,8378	0,6	0,8259

Figure 4.3: Comparative Response Analysis of Basic RAG and Hybrid RAG Schemes (Text and Visual Evidence-Based) for a Representative Query.

(CLIP SCORE	(CE SCORE)	(KW SCORE)	(HYBRID SCORE)	Puan (0-4)	hybrit	TP (True Positi	FP (False Positi	TN (True Negativ	FN (False Negativ
1,00000	1,000	1,000	0,999999630	4	1	1	0	0	0
0,51923431	0,99999950	1,000	0,999999511	4	1	1	0	0	0
0,30373747	0,99886660	1,000	0,998923290	4	1	1	0	0	0
0,60714868	0,99999976	0,96428570	0,998214035	3	1	1	0	0	0
0,11763111	0,99999976	0,96428571	0,998214035	3	1	1	0	0	0
0,14290325	1,00000000	1,00000000	0,999999988	4	1	1	0	0	0
0,2240564	0,99385166	1,00000000	0,994159091	3	1	1	0	0	0
0,5710547	0,99999940	0,84375000	0,992186952	3	1	1	0	0	0
0,0724440	0,99969070	0,84375000	0,991893637	4	1	1	0	0	0
0,0903482	0,99960760	0,84375000	0,991814721	4	1	1	0	0	0
0,5131878	0,99107885	0,75000000	0,979024923	3	1	1	0	0	0
0,6386213	0,99348220	0,70129870	0,978873014	3	1	1	0	0	0
0,1085544	1,00000000	0,51428571	0,975714274	2	1	1	0	0	0
0,5615242	0,98477200	0,71428571	0,971247690	3	1	1	0	0	0
0,5615243	0,97964704	0,73469387	0,967399352	3	1	1	0	0	0
0,0966307	0,99994767	1,00000000	0,999950278	4	1	1	0	0	0
0,6890259	0,99902713	1,00000000	0,999075758	3	1	1	0	0	0
0,2834702	0,99889380	1,00000000	0,998949099	3	1	1	0	0	0
0,0933770	1,00000000	0,96428571	0,998214274	2	1	1	0	0	0
0,2826289	0,99995580	0,85714285	0,992815110	4	1	1	0	0	0
0,8341624	0,99980430	1,00000000	0,999814081	4	1	1	0	0	0
0,8255188	0,99962515	0,88800000	0,994043850	3	1	1	0	0	0
0,8067615	0,99955680	0,84000000	0,991578941	2	1	1	0	0	0
0,1319166	0,99833370	0,84000000	0,990417008	1	1	0	1	0	0
0,4738149	0,99953973	0,81333333	0,990229395	1	1	0	1	0	0
0,1358105	0,99933845	1,00000000	0,999371517	4	1	1	0	0	0
0,5370092	0,99999905	0,97222222	0,998610205	4	1	1	0	0	0
0,2117452	0,99890650	0,93750000	0,995836139	2	1	1	0	0	0
1,0000000	0,99999595	0,87499999	0,993746114	4	1	1	0	0	0
0,7258852	0,99858960	0,89285714	0,993302933	4	1	1	0	0	0
0,6992000	1,00000000	0,40000000	0,970000000	2	1	1	0	0	0
0,7185000	0,94490000	0,90000000	0,942600000	4	1	1	0	0	0
0,2729000	0,87010000	0,60000000	0,856600000	2	1	1	0	0	0
0,9168	0,8598	0,225	0,8281	4	1	1	0	0	0
0,2837	0,8378	0,6	0,8259	2	1	1	0	0	0

Figure 4.4: Visual Relevance Evaluation Results of the Multimodal System (Comparative Breakdown of CLIP, Contextual Embedding, Keyword and Hybrid Scores by Query and Human Judgment Results)

--- TOP 4 Results Reordered by Hybrid Score ---

Query: Is there a bus for Peacehaven 14

--- Row 1 ---

Text: A bus is in front of a building.

File Name: 000000384027.jpg

CLIP Score (Norm.): 0.3571

CE Score (Norm.): 1.0000 (Weight: 0.95)

KW Score (Norm.): 0.8000 (Weight: 0.05)

HYBRID Score: 0.9900

--- Row 2 ---

Text: There is a bus driving along the road.

File Name: 000000304076.jpg

CLIP Score (Norm.): 0.1645

CE Score (Norm.): 0.9758 (Weight: 0.95)

KW Score (Norm.): 0.8000 (Weight: 0.05)

HYBRID Score: 0.9670

--- Row 3 ---

Text: A bus at a stop in the city

File Name: 000000316526.jpg

CLIP Score (Norm.): 0.4336

CE Score (Norm.): 0.8823 (Weight: 0.95)

KW Score (Norm.): 0.6000 (Weight: 0.05)

HYBRID Score: 0.8682

--- Row 4 ---

Text: a couple of buildings that has a bus out front

File Name: 000000263104.jpg

CLIP Score (Norm.): 0.2213

CE Score (Norm.): 0.7059 (Weight: 0.95)

KW Score (Norm.): 0.4800 (Weight: 0.05)

HYBRID Score: 0.6946

Figure 4.5: Outputs Demonstrating Hybrid Score's Ability to Confirm High Visual Relevance Despite Low CLIP Scores (Query: "Is there a bus for Peacehaven 14")

The performance of the developed Hybrid System in evaluating the relationship between the query and the retrieved visual evidence shows high recall success in capturing relevant evidence but faces some limitations in eliminating irrelevant cases (Precision/FP).

Table 4.7: Row 1 Evaluation

Query Text	Retrieval Status	CLIP Score (Visual Match)	HYBRID Score (Contextual Alignment)	Methodological Implication
Is there a bus for Peacehaven 14?	Specific number is NOT VISIBLE (Factual Deficiency).	0.3571 (Low)	0.9900 (Excessively High)	Contextual Success: Despite the low CLIP score, the Hybrid Score components (specifically CE: 1.000) perfectly matched the image's general context ("Route Bus," "Urban Transit") with the query's context. Although the system bypassed the specific detail, it interpreted the retrieved image's belonging to the relevant category (intercity/city bus) as a high relevance match, resulting in successful retrieval.

Table 4.8: Row 2 Evaluation

Query Text	Retrieval Status	CLIP Score (Visual Match)	HYBRID (Contextual Alignment)	Methodological Implication
Is there a bus for Peacehaven 14?	Text: A bus is driving along the road.	0.1645 (Very Low)	0.9670 (High)	Dynamic Action Capture: Despite an extremely low CLIP score, the Hybrid Score successfully captured the bus's "driving action" and the urban transport context through the strength of the CE component. This demonstrates Hybrid Score 's capability to retrieve evidence involving dynamic actions with high confidence.

Table 4.9: Row 3 Evaluation

Query Text	Retrieval Status	CLIP Score (Visual Match)	HYBRID Score (Contextual Alignment)	Methodological Implication
Is there a bus for Peacehaven 14?	Text: A bus at a stop in the city.	0.4336 (Moderately Low)	0.8682 (High)	Positional Alignment: The slight recovery in the CLIP score indicates better fundamental object matching. Hybrid Score successfully verified the contextual requirement of the query—"stop" and "city"—and maintained the image at a high relevance level. The Hybrid Score staying at the 0.8682 level suggests that the CE and KW scores (0.8823 and 0.6000 respectively) were less perfect compared to previous examples.

Table 4.10: Row 4 Evaluation

Query Text	Retrieval Status	CLIP Score (Visual Match)	HYBRID Score (Contextual Alignment)	Methodological Implication
Is there a bus for Peacehaven 14?	Text: a couple of buildings that has a bus out front.	0.2213 (Low)	0.6946 (Below Threshold)	Boundary Case: Although the image contains buildings, a bus stop, and a bus, the Hybrid Score (0.6946) fell just below the 0.70 threshold. This situation highlights the system's tendency to commit an elimination error for visuals that are positionally complex (buildings, bus, stop) and have a low initial CLIP signal, which carries the potential for generating a False Negative (FN) error.

4.2.3.1 Human Judgment Criteria and Data Set Structure

To measure the module's alignment with human preferences, a balanced test dataset consisting of 200 query-image pairs was prepared, comprising 100 Relevant and 100 Irrelevant pairs. The ground truth labels of the examples were determined by a human judge using the Core Score method. The classification threshold for Hybrid Score was set at 0.70.

Table 4.11: Human Judgment Criteria Used for Visual Relevance

Core Score	Human Assessment	Binary Category (Class)
1	No Relevance	Irrelevant (Negative Class)
2	Mild Relevance	Relevant (Positive Class)
3	High Relevance	Relevant (Positive Class)
4	Complete Relevance	Relevant (Positive Class)

4.2.3.2 Hybrid Score Performance Metrics and Evaluation

The classification success of the Visual Elimination Module is summarized by standard performance metrics calculated on a total of 200 subjects.

Table 4.12: Classification Matrix and Class Metrics

Metric	Value	Interpretation
True Positive (TP)	90	The number of instances where the model correctly classified human-approved relevant images as relevant.
True Negative (TN)	69	The number of instances where the model correctly classified irrelevant images as irrelevant.
False Negative (FN)	10	The number of instances where the model incorrectly classified relevant images as irrelevant (Type II Error).
False Positive (FP)	31	The number of instances where the model incorrectly classified irrelevant images as relevant (Type I Error).

Metric	Value	Interpretation
True 0 (Recall – Positive Class)	0.900	The rate at which relevant (Positive Class) examples were correctly captured (Recall).
True 1 (Specificity - Negative Class)	0.69	The rate at which irrelevant (Negative Class) examples were correctly rejected (Specificity).

Table 4.13: Performance Metric Values and Formulas

Performance Metric	Value	Formula	Interpretation
Accuracy	0.795	$(TP + TN)/N$	The overall success rate of the system's predictions across all samples is 79.5%.
Precision	0.744	TP/(TP+FP)	74.4% of the images predicted as relevant are genuinely relevant.
Recall	0.900	TP/TP+FN)	90% of all relevant images that the system should capture were successfully retrieved.
F1 Score	0.814	$F1 = 2 * \frac{(Precision * Recall)}{Precision + Recall}$	A strong success demonstrating the harmonic balance between Precision and Recall.

4.2.3.3 Comparative Evaluation of Scoring Models

In this section, the performance of the developed Hybrid Score Mechanism is evaluated by comparing it with basic Visual Language Models (VLM) and other specialized models (RS and CS).

4.2.3.3.1 Performance Metrics for All Models

The table below contains the overall success rates (Accuracy) and class-specific correct prediction rates (true0 [Recall] and true1 [Specificity]) for all models on the test data set.

Table 4.14 Performance Metrics

Model	Accuracy	true0 (Recall)	true1 (Specificity)
CS model	0.875	0.940	0.806
RS model	0.865	0.909	0.831
Hybrid Score	0.795	0.900	0.690
VILA	0.732	0.710	0.734
LLaVA	0.724	0.695	0.746

4.2.3.3.2 Critical Comparative Analysis

This analysis compares the performance of the developed Hybrid Score Model with other Visual Language Models (VLMs) and specialized hybrid scoring models presented in the paper titled "RAG-Check: Evaluating Multimodal Retrieval Augmented Generation Performance."[37]

A. The Superiority of Hybrid Models Over Basic VLM

Comparative data from the article shows that the basic VLM models, LLaVA (Accuracy: 0.724) and VILA (Accuracy: 0.732), demonstrate significantly lower overall performance compared to all hybrid scoring models (Hybrid Score, RS, CS).In particular, the Hybrid Scoring Model's Accuracy (0.795) and true0 (0.900) values surpass the performance of basic VLM, proving its superiority in evidence retrieval success.

B. Positioning Performance Among Hybrid Models

A detailed comparison of the hybrid models is as follows:

Evidence Capture (true0 - Recall):

- The CS Model (0.940) and RS Model (0.909) showed the highest performance in the relevant evidence capture (true0) metrics.
- The Hybrid Score Model's true0 value (0.900) is very close to the CS and RS models, proving that it is largely successful in achieving the goal of not missing relevant evidence.

False Rejections (true1 - Specificity):

- The RS Model (0.831) has the highest value for this metric and is the model that best rejects irrelevant images.

- The true1 value of the Hybrid Score Model (0.69) shows the lowest performance among all models compared. This low value definitively supports the False Positive (FP) tendency mentioned in the previous analysis and the model's weakness in over-reliance on the general context (e.g., 0.9900 Hybrid Score in the Peacehaven 14 query).

4.3 Conclusion

The Hybrid Score Model generally outperforms the baseline VLM (Visual Language Model). It also demonstrates similar success to other hybrid models (CS and RS) in capturing evidence (identifying relevant images). However, the biggest challenge faced by the Hybrid Score Model is its ability to exclude irrelevant evidence (reject false positives); it is noted to be the weakest model in this metric (true1). This finding clearly indicates that future improvements to the model should focus on specificity. Increasing this specificity will help improve model accuracy and significantly reduce unwanted false positives. In this context, it is emphasized that the RS Model has the highest performance in this metric with a value of 0.831 and is the model that best rejects irrelevant images.

4.4 Accuracy Improvement Strategy with Gemma 12B

To address the model's lack of specificity and improve accuracy, an image verification strategy supported by a high-performance language model like Gemma 12B can be integrated into the Hybrid Score Model. Thanks to its powerful semantic understanding and reasoning capabilities, Gemma 12B can analyze the content and context of the potential evidence (images) initially captured by the model as a secondary filter. This strategy detects subtle semantic inconsistencies between the image and the given textual evidence, helping it produce more accurate image output even when visual-linguistic matching alone is insufficient. Thus, the Hybrid Score Model's ability to more effectively reject irrelevant evidence, a weakness, can be enhanced, and the specificity achieved by the RS Model can be approached. This improvement maintains the Hybrid Score Model's evidence-capture success while eliminating its most critical weakness: false positives.

CHAPTER FİVE

CONCLUSION, DISCUSSION, AND RECOMMENDATIONS

5.1 Introduction

This section summarizes the main findings of the study, evaluates the extent to which the research objectives were achieved, discusses the performance of hybrid RAG systems, and provides recommendations for future work. In particular, it addresses the potential of resource-constrained offline systems, the strengths and weaknesses of Hybrid Model approaches, and the search for permanent solutions to structural problems encountered during PDF data extraction.

5.2 RAG System Comparison and Discussion

5.2.1 Balance of Performance Metrics

The RAG system comparison conducted clearly demonstrated the potential of offline optimizations and the impact of Large Language Model (LLM) preferences on performance:

- **Success of Our Offline Systems (A-E)-** Average Performance: Despite using a resource-constrained, offline LLM (Gemma3:4b:1b), the developed systems demonstrated superiority in high semantic similarity (BERTScore average: 0.73538) and overall LLM quality (G-Eval average: 4.568) metrics. This proves that the applied offline optimizations are highly effective in maximizing semantic depth and the overall response quality of the LLM.

- **Real-World Accuracy - Average Performance:** On the CheckAvail score, a real-world accuracy metric, the average RAG system achieved a high value of 0.8827. This score represents an exceptionally competitive achievement for a resource-constrained system compared to online systems.

5.2.2 Comparison of Online and Offline Systems

The RAG system comparison highlights a clear trade-off between the systems: The XSum Line, benefiting from online models and architectural optimizations, is clearly excellent at maximizing real-world accuracy (CheckAvail: 0.97).

Consequently, if real-world accuracy is the most critical requirement, the online model infrastructure used by XSum appears to be more effective. However, when semantic depth and general LLM preference are prioritized, the developed offline hybrid systems should be preferred due to their superior performance in metrics like BERTScore and G-Eval. Considering that XSum's score of 0.97 belongs to a fully online system, the average offline RAG system's real-world performance of 0.8827 is an exceptionally competitive achievement for resource-constrained systems, demonstrating that the work offers a cost-effective and reliable alternative for closed systems.

5.3 Specificity Analysis of the Hybrid Score Model

The Hybrid Score Model generally demonstrated success in Evidence Capture (i.e., identifying relevant images/visual data) compared to the baseline VLM and showed similar performance to other hybrid models (CS and RS). However, the Hybrid Score Model is the weakest model in terms of Excluding Irrelevant Evidence (true1 metric). This critical flaw indicates that the model has low specificity and struggles with the precise task of rejecting irrelevant images, leading to a higher rate of visual false positives (including non-pertinent images as evidence). Conversely, the RS Model exhibits the highest specificity, achieving the best performance (0.831) in rejecting irrelevant images. Consequently, future developments of the Hybrid Score Model must focus on enhancing the image filtering mechanism to increase specificity, thereby improving overall accuracy and significantly reducing the false positive rate stemming from irrelevant visual evidence.

5.4 Limitations of the Study and Future Directions

5.4.1 Limitations of PDF Data Extraction and Search for Solutions

A significant limitation of the study is the structural problems encountered when extracting images from PDF data.

Problem Identification: Images cannot be extracted in an orderly manner due to certain PDF structures.

Temporary Solution: This issue can be temporarily overcome by writing a new PDF image extraction function tailored to that specific PDF structure.

Permanent Solution Goal: During the project's development phase, work continues to produce a more permanent and general solution to this problem. This is a critical step toward enhancing the system's overall reliability and generalizability.

5.4.2 Recommendations for Future Research

Intensive work is ongoing on three key development objectives focused on overcoming the limitations of existing models and processes. First, to increase the exclusion rate of irrelevant (true1) evidence, we are conducting research on faster and more accurate next-generation filtering mechanisms or more precise scoring methods that will overcome the speed and efficiency limitations of the current Gemma 12B validation mechanism. Second, to address the Optimization for Resource-Constrained Large Language Models (LLM) objective, we are researching specific architectural improvements and more advanced offline optimization techniques to bring the CheckAvail score of the offline Gemma model closer to the XSum level. Finally, active work is underway to develop a robust and structure-independent image/visual extraction function that can easily adapt to different PDF structures.

5.5 General Conclusion

This study has demonstrated that resource-constrained offline RAG systems can serve as an exceptionally competitive alternative to their online counterparts, particularly in terms of semantic and LLM quality metrics. The average CheckAvail performance of 0.8827 proves how promising such systems are for closed and cost-effective solutions. Future developments of the Hybrid Score Model should focus on increasing specificity in order to improve accuracy and reduce the false positive rate, along with enhancing the robustness of PDF data extraction mechanisms.

REFERENCES

[1] S. Alghisi, M. Rizzoli, G. Roccabruna, S. M. Mousavi, and G. Riccardi, "Should We Fine-Tune or RAG? Evaluating Different Techniques to Adapt LLMs for Dialogue," Aug. 2024, [Online]. Available: http://arxiv.org/abs/2406.06399

[2] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," Mar. 2024, [Online]. Available: http://arxiv.org/abs/2312.10997

[3] Y. Huang and J. X. Huang, "The Survey of Retrieval-Augmented Text Generation in Large Language Models," vol. 1, no. 1, 2018.

[4] S. Zhao, Y. Yang, Z. Wang, Z. He, L. K. Qiu, and L. Qiu, "Retrieval Augmented Generation (RAG) and Beyond: A Comprehensive Survey on How to Make your LLMs use External Data More Wisely," 2024, [Online]. Available: http://arxiv.org/abs/2409.14924

[5] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, and D. Yin, "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models," 2024.

[6] F. Liu, Z. Kang, and X. Han, "Optimizing RAG Techniques for Automotive Industry PDF Chatbots: A Case Study with Locally Deployed Ollama Models Optimizing RAG Techniques Based on Locally Deployed Ollama Models A Case Study with Locally Deployed Ollama Models," 2024.

[7] A. Vaswani *et al.*, "Attention Is All You Need," 2017. [Online]. Available: https://arxiv.org/pdf/1706.03762v1

[8] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, Oct. 2018, [Online]. Available: https://arxiv.org/pdf/1810.04805

[9] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever, "Improving Language Understanding by Generative Pre-Training," 2018.

[10] W. X. Zhao *et al.*, "A Survey of Large Language Models," Mar. 2025, [Online]. Available: http://arxiv.org/abs/2303.18223

[11] Gemma Team *et al.*, “Gemma: Open Models Based on Gemini Research and Technology,” Apr. 2024, [Online]. Available: http://arxiv.org/abs/2403.08295

[12] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” Apr. 2021, [Online]. Available: http://arxiv.org/abs/2005.11401

[13] Y. Shi, J. Wang, Z. Shan, D. Peng, Z. Lin, and L. Jin, “URaG: Unified Retrieval and Generation in Multimodal LLMs for Efficient Long Document Understanding,” Nov. 2025, [Online]. Available: http://arxiv.org/abs/2511.10552

[14] P. Zhao *et al.*, “Retrieval-Augmented Generation for AI-Generated Content: A Survey,” Jun. 2024, [Online]. Available: http://arxiv.org/abs/2402.19473

[15] C. Choi, W. Lee, J. Ko, and W. Rhee, “Multimodal Iterative RAG for Knowledge-Intensive Visual Question Answering,” Sep. 2025, [Online]. Available: http://arxiv.org/abs/2509.00798

[16] J. Zhang *et al.*, “Beyond Text: Unveiling Privacy Vulnerabilities in Multi-modal Retrieval-Augmented Generation,” May 2025, [Online]. Available: http://arxiv.org/abs/2505.13957

[17] P. Wu *et al.*, “MES-RAG: Bringing Multi-modal, Entity-Storage, and Secure Enhancements to RAG,” Mar. 2025, [Online]. Available: http://arxiv.org/abs/2503.13563

[18] S. Han, Q. Zhang, Y. Yao, W. Jin, and Z. Xu, “LLM Multi-Agent Systems: Challenges and Open Problems,” May 2025, [Online]. Available: http://arxiv.org/abs/2402.03578

[19] G. Pan, V. Chodnekar, A. Roy, and H. Wang, “A Cost-Benefit Analysis of On-Premise Large Language Model Deployment: Breaking Even with Commercial LLM Services,” Nov. 2025, [Online]. Available: http://arxiv.org/abs/2509.18101

[20] L. Ma *et al.*, “A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge,” Jun. 2025, [Online]. Available: http://arxiv.org/abs/2310.11703

[21] P. Xia *et al.*, “RULE: Reliable Multimodal RAG for Factuality in Medical Vision Language Models,” Oct. 2024, [Online]. Available: http://arxiv.org/abs/2407.05131

[22] H. Tanaka, A. Rao, H. Satou, M. Johnson, and S. García, “Dynamic Modality Scheduling for Multimodal Large Models via Confidence,

Uncertainty, and Semantic Consistency," Jun. 2025, [Online]. Available: http://arxiv.org/abs/2506.12724

[23] Q. Liu, H. Duan, Y. Chen, Q. Lu, W. Sun, and J. Mao, "LLM4Ranking: An Easy-to-use Framework of Utilizing Large Language Models for Document Reranking," Apr. 2025, [Online]. Available: http://arxiv.org/abs/2504.07439

[24] A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," Feb. 2021, [Online]. Available: http://arxiv.org/abs/2103.00020

[25] G. Team and G. Deepmind, "Google. (2024). Gemma: Introducing new state-of-the-art open models.," 2024, [Online]. Available: https://storage.googleapis.com/deepmind-media/gemma/gemma-report.pdf

[26] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Dec. 2019, [Online]. Available: http://arxiv.org/abs/1912.01703

[27] "Chroma, 'Chroma Documentation,.'" [Online]. Available: https://www.trychroma.com/

[28] "Artifex, 'PyMuPDF Documentation,.'" [Online]. Available: https://pymupdf.readthedocs.io

[29] "LangChain, 'LangChain Documentation,.'" [Online]. Available: https://www.langchain.com/

[30] "Ollama, 'Ollama Documentation,.'" [Online]. Available: https://docs.ollama.com/

[31] "Streamlit, 'Streamlit Documentation,.'" [Online]. Available: https://streamlit.io/

[32] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Aug. 2019, [Online]. Available: http://arxiv.org/abs/1908.10084

[33] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," 2003.

[34] "F. Pedregosa et al., 'Scikit-learn: Machine Learning in Python,' Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.", [Online]. Available: http://scikit-learn.org

[35] C. R. Harris *et al.*, "Array Programming with NumPy," Jun. 2020, doi: 10.1038/s41586-020-2649-2.

[36] P. Achkar, T. Gollub, and M. Potthast, “Ask, Retrieve, Summarize: A Modular Pipeline for Scientific Literature Summarization,” May 2025, [Online]. Available: http://arxiv.org/abs/2505.16349

[37] M. Mortaheb, M. A. A. Khojastepour, S. T. Chakradhar, and S. Ulukus, “RAG-Check: Evaluating Multimodal Retrieval Augmented Generation Performance,” Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.03995