Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Design and Evaluation of a Retrieval-Augmented Question Answering System for Education Managers: Bringing Research Results to Professionals in the Field

Abraão de Paula Carolino

JUIZ DE FORA
JANEIRO, 2026

Design and Evaluation of a Retrieval-Augmented Question Answering System for Education Managers: Bringing Research Results to Professionals in the Field

Abraão de Paula Carolino

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Jairo Francisco de Souza

Coorientador: César Pedrosa Soares

JUIZ DE FORA

JANEIRO, 2026

Design and Evaluation of a Retrieval-Augmented Question Answering System for Education Managers: Bringing Research Results to Professionals in the Field

Abraão de Paula Carolino

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Jairo Francisco de Souza
Professor do Departamento de Ciência da Computação da Universidade Federal de Juiz de Fora

César Pedrosa Soares
Analista de dados no Centro de Políticas Públicas e Avaliação da Educação (CAEd/UFJF)

Marcelo de Oliveira Costa Machado
Professor do Departamento de Ciência da Computação da Universidade Federal de Juiz de Fora

Jorão Gomes Jr.
Professor da Vienna University of Economics and Business

JUIZ DE FORA

20 DE JANEIRO, 2026

To my friends and brothers.

To my parents, for their support and sustenance.

Abstract

Educational decision-making increasingly requires access to reliable, research-based information. However, the vast amount of academic production often remains distant from those who could most benefit from it: teachers, managers, and policymakers, due to barriers of access. As a result, insights that could improve public education are frequently underused in practice.

This study proposes a technological approach to bridge this gap between research and practice by developing a question answering system capable of transforming academic repositories into accessible sources of knowledge. Built on the Retrieval-Augmented Generation (RAG) paradigm, the system integrates large language models (LLMs) with document retrieval, enabling users to explore research findings through natural dialogue while maintaining factual grounding.

Three configurations of the system were developed and tested to assess their ability to provide accurate and contextually relevant responses to questions in the educational field. The results demonstrate that such systems can make complex academic knowledge more approachable for professionals engaged in evidence-based decision-making.

Ultimately, this research contributes not only a functional system but also an example of how to apply artificial intelligence to democratize access to scientific knowledge in education, transforming static academic repositories into dynamic environments that support learning and policy innovation.

Keywords: Retrieval-Augmented Generation, RAG, Large Language Models, LLM, Education, Academic Knowledge, Decision Support.

Resumo

A tomada de decisão na educação exige cada vez mais o acesso a informações confiáveis e baseadas em evidências. No entanto, o grande volume de produção acadêmica permanece distante daqueles que mais poderiam se beneficiar desse conhecimento: professores, gestores e formuladores de políticas, devido a barreiras de acesso. Como resultado, evidências que poderiam orientar melhorias na educação pública acabam sendo subutilizadas na prática.

Este estudo propõe uma abordagem tecnológica para reduzir essa distância entre pesquisa e prática, por meio do desenvolvimento de um sistema de perguntas e respostas capaz de transformar repositórios acadêmicos em fontes acessíveis de conhecimento. Baseado no paradigma de Geração Aumentada por Recuperação (RAG), o sistema integra Modelos de Linguagem de Grande Escala (LLMs) com mecanismos de recuperação de documentos, permitindo que o usuário explore resultados de pesquisas por meio de diálogo natural e fundamentado em evidências.

Três configurações do sistema foram desenvolvidas e testadas, avaliando sua capacidade de fornecer respostas precisas e contextualmente relevantes a perguntas do campo educacional. Os resultados demonstram que sistemas desse tipo podem tornar o conhecimento acadêmico mais acessível e útil para profissionais engajados em decisões baseadas em evidências.

Por fim, a pesquisa contribui não apenas com um sistema funcional, mas também com um exemplo de como aplicar a inteligência artificial na democratização do acesso ao conhecimento científico em educação, transformando repositórios acadêmicos estáticos em ambientes dinâmicos de diálogo que apoiam o aprendizado e a inovação em políticas públicas.

Palavras-chave: Geração Aumentada por Recuperação, RAG, Modelos de Linguagem de Grande Escala, LLM, Educação, Conhecimento Acadêmico, Apoio à Decisão.

Acknowledgments

To Professor Jairo, who is at this very moment reading this text and probably correcting the various errors I left behind. Thank you for your patience, guidance, and for all the support throughout this work.

To Professor César, co-supervisor of this project, whose previous developments served as the foundation for the system implemented here. Thank you for the valuable guidance, for clarifying doubts during the process, and especially for the discussions and insights that helped shape the qualitative evaluation system.

To my family, for their constant care, encouragement, and love throughout my life, and especially during these years of university. None of this would have been possible without your support.

To my friends, who made this journey lighter and helped me through the difficult moments, reminding me that even during the toughest times, there was always room for laughter and hope.

This work was supported by UFJF's High-Speed Integrated Research Network (RePesq): ⟨https://www.repesq.ufjf.br/⟩.

Contents

List of Figures **8**

List of Tables **9**

List of Abbreviations **10**

1 Introduction **11**

2 Theoretical Framework **14**
2.1 RAG Architectures . . . 15
2.2 Retrieval Methods . . . 16
2.3 Embeddings and Semantic Representations . . . 18
2.4 Data Preparation and Refinement . . . 20
2.4.1 Pre-retrieval Methods . . . 20
2.4.2 Post-Retrieval Methods . . . 21
2.5 Evaluation Metrics . . . 22
2.5.1 Retrieval Metrics . . . 22
2.5.2 Generation Metrics . . . 23
2.5.3 LLM-as-a-Judge Evaluation . . . 23
2.5.4 Overall RAG System Evaluation Metrics . . . 26
2.6 Vector Databases . . . 26
2.7 Related Work . . . 27
2.7.1 Domain-Specific RAG Systems . . . 28
2.7.2 RAG Frameworks and General-Purpose Tools . . . 28
2.7.3 Contribution . . . 29

3 Proposed System and Architecture **30**
3.1 Design Decisions and System Motivation . . . 30
3.2 Academic Corpus and Data Processing Pipeline . . . 31
3.2.1 Corpus Composition . . . 31
3.2.2 Text Extraction and Segmentation . . . 31
3.2.3 Embeddings and Vector Database . . . 32
3.2.4 Language Model Selection . . . 32
3.3 Retrieval-Augmented Generation Architectures . . . 32
3.3.1 Naive RAG Architecture . . . 32
3.3.2 Advanced RAG Architecture . . . 33
3.3.3 Modular RAG Architecture . . . 34
3.4 Design Trade-offs and Limitations . . . 36

4 Materials and Methods **37**
4.1 Computational Infrastructure . . . 37
4.2 Experimental Methodology . . . 38
4.2.1 Development of RAG Architectures . . . 38
4.2.2 Models and Vector Database . . . 39
4.2.3 System Architecture . . . 40

4.3 User-Based Evaluation Design 40
4.4 Automated System Evaluation Design 46
4.4.1 Retrieval and Query Fidelity Metrics 46
4.4.2 Response-Quality Metrics 47
4.4.3 Latency Measurement 48

5 Results and Analysis **49**
5.1 User Study Design 49
5.1.1 Recruitment 49
5.1.2 Participant Profile 50
5.2 Quantitative Analysis 50
5.2.1 Automated System-Level Metrics 50
5.2.2 Quantitative User-Based Ratings 51
5.3 Qualitative Analysis 55
5.3.1 Qualitative Feedback Analysis 55
5.4 Interpretation 55

6 Conclusion **57**

Bibliography **60**

List of Figures

2.1 Standard pipeline of a baseline RAG system (Naive RAG) 15
2.2 Overview of RAG optimization modules and processing stages used in Advanced and Modular architectures 16

3.1 Naive RAG architecture. 33
3.2 Advanced RAG architecture. 34
3.3 Modular RAG architecture. 35

4.1 Overview of the qualitative evaluation process. 41
4.2 Screens from the identification and profile stages of the qualitative evaluation. 41
4.3 Screens illustrating the fixed-question evaluation phase, including examples of the response evaluation interface. 42
4.4 Screen from the interactive chat showing message evaluation components. . 44
4.5 Feedback screen from the final stage of the qualitative evaluation 45

5.1 Distribution of Familiarity with Educational Assessment by Educational Level 50

List of Tables

5.1 Quantitative evaluation results for RAG architectures. 51
5.2 Mean and Standard Deviation of Ratings by Architecture for Fixed Questions 52
5.3 Mean and Standard Deviation of Ratings by Architecture for Fixed Questions (Expert Participants Only) 53
5.4 Mean Chat Ratings and Interaction Counts by Architecture 54
5.5 Mean Score by Message Index and Architecture in Chat Evaluation 54

List of Abbreviations

ANNS	Approximate Nearest Neighbor Search
CAEd	Center for Public Policies and Educational Evaluation
DCC	Department of Computer Science
HNSW	Hierarchical Navigable Small World
HyDE	Hypothetical Document Embedding
IDEB	Basic Education Development Index
LLM	Large Language Model
LTS	Long-Term Support
LSH	Locality-Sensitive Hashing
MMR	Mean Reciprocal Rank
NDCG	Normalized Discounted Cumulative Gain
PAE	Educational Action Plan
PPGP	Professional Graduate Program in Public Education Management and Evaluation
RAG	Retrieval-Augmented Generation
RRF	Reciprocal Rank Fusion
SBERT	Sentence-BERT
SD	Standard Deviation
UFJF	Federal University of Juiz de Fora
VDB	Vector Database
VDBMS	Vector Database Management System

1 Introduction

The rapid growth in the volume of digitally produced and stored information, driven by advances in information and communication technologies, has created substantial challenges related to data organization and effective use. In academic contexts, this phenomenon is particularly evident in the increasing production of articles, theses, technical reports, and research datasets, which collectively form extensive repositories of knowledge that are often difficult to navigate and synthesize (BORGMAN, 2015; TENOPIR et al., 2015).

In the field of education, these challenges directly affect teachers, school leaders, and policymakers, who often struggle to incorporate research evidence into their professional practice. Studies have highlighted the persistent gap between academic research and educational routines. According to Levin (2013), this disconnect is not merely due to a lack of information, but is rooted in the complex social and organizational nature of knowledge mobilization, which requires more than simple dissemination to translate findings into actionable insights.

Recent advances in artificial intelligence, particularly Large Language Models (LLMs), have enabled new forms of interaction with large-scale textual data. These models are capable of processing and generating natural language, supporting tasks such as summarization, question answering, and document exploration. However, despite their expressive capabilities, LLMs present important limitations for educational and professional use. Their reliance on static training data leads to knowledge cutoff issues (MALLEN et al., 2023; ZHANG et al., 2024), and their generative nature may result in inaccurate or unverifiable outputs—commonly referred to as hallucinations—reducing trust in educational settings where factual accuracy is essential (JI et al., 2023; ZHAO et al., 2024).

To mitigate these limitations, Retrieval-Augmented Generation (RAG) has been proposed as an architectural approach that integrates information retrieval mechanisms with language model generation (LEWIS et al., 2021). By retrieving documents from external repositories at query time and grounding responses in verifiable sources, RAG-based systems improve transparency and factual consistency. Recent reviews indicate

that this approach is particularly suitable for educational applications, as it allows for dynamic knowledge updates and mitigates the risks of misinformation (LI et al., 2025; GAO et al., 2024).

Within this context, this study investigates the design and evaluation of a RAG-based question answering system applied to an academic repository in education. The system was developed using a collection of 1,065 master's dissertations from the *Professional Graduate Program in Public Education Management and Evaluation* (PPGP) at the *Center for Public Policies and Educational Evaluation* (CAEd/UFJF). The objective is to facilitate exploratory access to academic research by education managers, researchers, and students, enabling structured interaction with complex scholarly content.

Three RAG architectures were implemented and compared, differing in their retrieval, reranking, and generation strategies. The evaluation combined automated quantitative metrics with an exploratory user study involving 13 participants with diverse educational backgrounds.

The central hypothesis of this work is that RAG-based question answering systems can serve as effective technological intermediaries, capable of transforming static academic repositories into interactive research tools. It is hypothesized that such systems can lower the barriers to information access, allowing professionals to locate, interpret, and synthesize relevant academic evidence through natural language, thereby enhancing the utility of existing educational collections.

Overall, the results suggest that while simpler RAG configurations perform well in retrieval-oriented metrics, more modular architectures tend to offer advantages in contextual grounding and response quality, albeit at the cost of increased latency. These findings highlight important trade-offs between efficiency and interpretability in educational applications of RAG.

This dissertation is organized as follows. Chapter 2 presents the theoretical background, covering Large Language Models, Retrieval-Augmented Generation, embeddings, evaluation metrics, and vector databases. Chapter 3 describes the proposed system and the design of the three RAG architectures. Chapter 4 details the experimental methodology, including dataset construction and evaluation procedures. Chapter 5 presents and

discusses the results from both quantitative metrics and user feedback. Finally, Chapter 6 concludes the study, outlining its limitations and directions for future research.

2 Theoretical Framework

With the growing demand for systems that understand and generate natural language, language models have evolved substantially. Traditional techniques, such as the use of n-grams and probabilistic models (GOODMAN, 2001), have been replaced by deep neural architectures with attention mechanisms, culminating in the Transformer model (VASWANI et al., 2023). This innovation enabled the creation of LLMs, such as GPT (OPENAI, 2023) and BERT (DEVLIN et al., 2019), which marked a turning point in natural language processing by significantly improving the ability to interpret and generate coherent text.

Even though large language models are powerful, they come with notable drawbacks like their knowledge cutoff date, which prevents them from accessing current or niche data not included during training (GAO et al., 2024). Because of this gap, answers can drift into error, what many call “hallucinations”. Such issues make these models less reliable when used in fast-moving fields. Academic work, troubleshooting tasks, and tools meant to guide choices may suffer as a result.

To tackle such constraints, a method known as Retrieval-Augmented Generation (RAG) emerged (LEWIS et al., 2021). RAG is a hybrid approach that combines information retrieval techniques with generation capabilities of LLMs. It allows the introduction of external information, retrieved in real time, into the generation pipeline. This strategy not only mitigates hallucinations but also expands the applicability of LLMs to domains requiring up-to-date or domain-specific information (YU et al., 2024).

RAG operates by retrieving pertinent documents to a user query from a designated source, such as a database, search engine, or document repository, and then adding this information into the prompt context before generation. Making it so the LLMs have access to relevant data related to the prompt to inform their generation and produce more data-accurate responses in return.

2.1 RAG Architectures

Different RAG architectures have been developed to meet specific scenarios and demands, each with its own advantages and limitations. Below, we discuss three approaches defined by Gao et al. (2024).

The first approach, *Naive RAG*, adopts a basic sequential flow of operations: indexing, retrieval, and generation. Initially, documents are chunked and indexed; then, information is retrieved based on the similarity to user queries; finally, the model generates a response using the retrieved data. As illustrated in Figure 2.1, this architecture follows a simple "Retrieve-Read" framework. However, Gao et al. (2024) highlight significant limitations in this approach, categorized into retrieval, generation, and augmentation challenges. Specifically, the retrieval phase often suffers from low precision and recall, selecting misaligned or irrelevant chunks. This occurs because the system relies solely on the initial similarity calculation without further verification, making it vulnerable to noise. Consequently, the generation phase may produce hallucinations or incoherent outputs when the retrieved context is disjointed or insufficient.

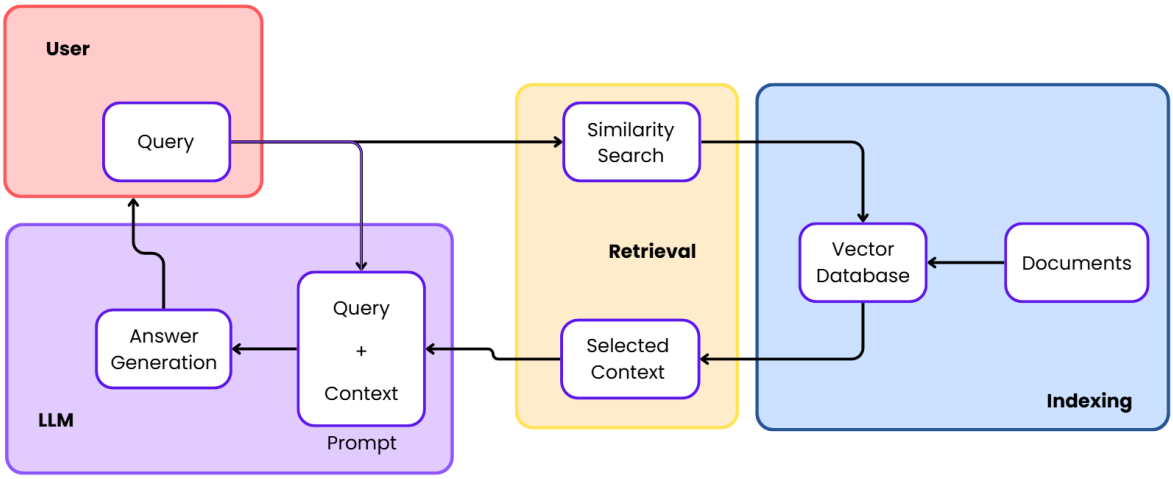

Figure 2.1: Standard pipeline of a baseline RAG system (Naive RAG)

To overcome these drawbacks, *Advanced RAG* introduces specific improvements through pre-retrieval and post-retrieval strategies. As depicted in Figure 2.2, which illustrates the broader ecosystem of RAG optimizations, these strategies aim to refine the process without necessarily altering the sequential nature of the pipeline. Pre-retrieval methods, such as query rewriting and query expansion, aim to refine the user input to

improve search relevance. Post-retrieval techniques, such as *re-ranking* and context compression, act as a filter to ensure that the most relevant information is prioritized and fits within the model's context window, thereby increasing the clarity and accuracy of generated responses (GAO et al., 2024).

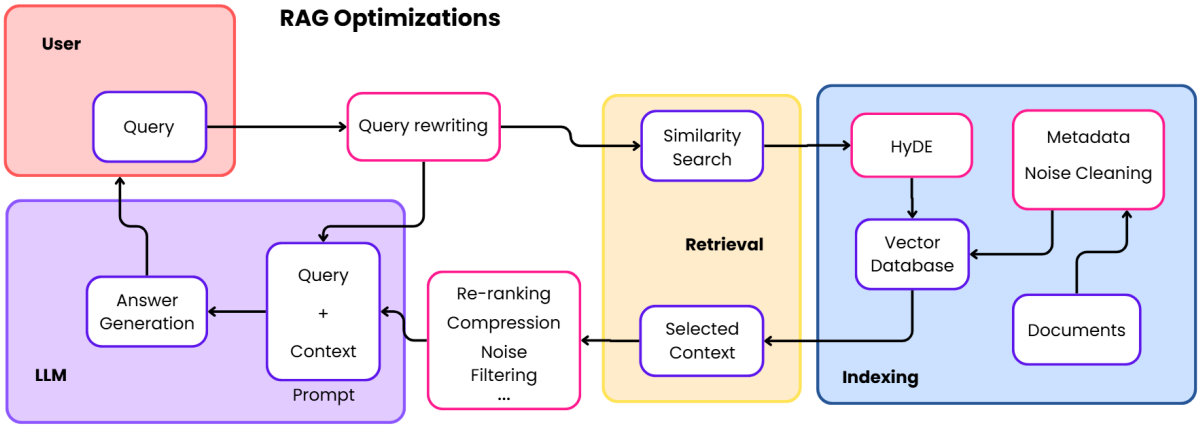

Figure 2.2: Overview of RAG optimization modules and processing stages used in Advanced and Modular architectures

Finally, *Modular RAG* offers a more flexible architecture where retrieval and generation are not limited to a rigid sequential pipeline. This paradigm utilizes the functional modules presented in Figure 2.2—such as search, memory, and routing—but allows for their substitution or dynamic reconfiguration to adapt to diverse tasks. Rather than following a fixed path, Modular RAG can orchestrate these components to enable the integration of sophisticated retrieval strategies, such as *Hybrid Retrieval* and *Recursive Retrieval*, which are discussed in detail in the following section.

2.2 Retrieval Methods

The performance of RAG-based systems is directly related to the retrieval methods employed. These methods play a crucial role in locating and returning relevant information that will be used in the response generation stage. The choice of the most appropriate method depends on the application domain and the specific needs of the problem being addressed.

Sparse Retrieval is based on exact keyword matching and uses algorithms such as BM25 (ROBERTSON; ZARAGOZA, 2009). This approach grounds the retrieval process

in lexical overlap, measuring the frequency of query terms within documents to determine relevance (LI et al., 2025; GAO et al., 2024). This approach is efficient in contexts where the exact terms of the queries are sufficiently representative to locate the desired information. Despite its simplicity and efficiency, *sparse* methods may have limitations in scenarios requiring more complex semantic interpretation.

Dense Retrieval, on the other hand, adopts a different approach, using vector representations (*embeddings*) to capture deeper semantic relationships between queries and documents (KARPUKHIN et al., 2020; GAO et al., 2024). These methods are usually based on Transformer models like BERT (DEVLIN et al., 2019). This technique is especially useful in scenarios where the language used in queries may differ significantly from that present in the data (e.g., synonyms or paraphrasing), providing greater flexibility and accuracy than lexical matching alone (LI et al., 2025).

Graph-Based Retrieval excels in highly structured scenarios where explicit relationships between data play an important role. This method uses graphs to connect related information, allowing the system to navigate complex relationships and locate content in a highly contextualized manner (GAO et al., 2024). Typical applications include recommendation systems and highly interconnected databases.

To maximize performance, *Hybrid Retrieval* combines the advantages of both approaches—*sparse* and *dense*—using keywords for efficiency and vector embeddings for greater semantic understanding. These two types coexist through fusion mechanisms, such as weighted scoring or Reciprocal Rank Fusion (RRF), where the system normalizes and combines the rankings from both retrievers to produce a final, more robust list of documents (GAO et al., 2024). This method is particularly effective in open-ended queries or cases where the context of the information varies widely, offering a balance between accuracy and coverage.

Additionally, *Recursive Retrieval* strategies can be employed to enhance depth. This method works by iteratively refining search queries based on previous results or by traversing hierarchical data structures. By breaking down complex problems into manageable retrieval steps, it improves the overall richness and granularity of the context provided to the LLM (GAO et al., 2024).

2.3 Embeddings and Semantic Representations

Embeddings are dense numerical representations of textual data that encode semantic and syntactic properties of language into continuous vector spaces. By mapping words, sentences, or entire documents to fixed-length vectors, embeddings enable computational models to measure semantic similarity, perform clustering, and retrieve related content based on meaning rather than exact lexical overlap. This representation paradigm is foundational to modern natural language processing and underpins dense retrieval, vector databases, and RAG systems.

Early approaches to distributed representations were introduced with neural language models such as Word2Vec (MIKOLOV et al., 2013), which demonstrated that words appearing in similar contexts could be represented as nearby vectors in an embedding space. While effective for capturing local semantic relationships, these models generate static word embeddings, assigning a single vector to each word regardless of context. This limitation becomes problematic in natural language, where word meaning is often context-dependent.

The emergence of Transformer-based architectures (VASWANI et al., 2023) enabled the development of contextualized embeddings, where the representation of a token varies according to its surrounding context. Models such as BERT (DEVLIN et al., 2019) produce embeddings that reflect both syntactic structure and semantic nuance, substantially improving performance in tasks such as question answering, information retrieval, and textual inference. Contextual embeddings are particularly important in domains like education and social sciences, where meaning is frequently shaped by discourse, institutional context, and specialized terminology.

Building upon contextual language models, Sentence-BERT (SBERT) (REIMERS; GUREVYCH, 2019) introduced a Siamese architecture that enables efficient generation of sentence-level embeddings optimized for semantic similarity tasks. Unlike vanilla BERT, which is computationally expensive for pairwise comparisons, SBERT allows embeddings to be computed independently and compared using vector similarity measures such as cosine similarity. This innovation made large-scale semantic search and dense retrieval feasible and has since become a standard approach in embedding-based information re-

trieval pipelines.

Recent research has further explored the use of large language models to generate high-quality embeddings. Models such as SGPT (MUENNIGHOFF, 2022) demonstrate that generative pre-trained transformers can be adapted to produce sentence embeddings that perform competitively in semantic search tasks. These approaches blur the boundary between generative and representational models, highlighting the increasing versatility of LLMs across both retrieval and generation stages.

In Retrieval-Augmented Generation systems, embeddings play a central role in the retrieval phase. User queries are embedded into the same vector space as document chunks, and similarity search is used to identify the most relevant pieces of external knowledge (LEWIS et al., 2021). The effectiveness of this process depends critically on the quality of the embeddings, as poorly aligned representations can lead to irrelevant retrieval and, consequently, degraded generation quality. As emphasized by Gao et al. (2024), embedding quality directly influences recall, contextual relevance, and faithfulness in RAG systems.

Embeddings also form the backbone of vector databases, where they enable approximate nearest neighbor search over large corpora. Vector database management systems rely on embedding representations to support efficient similarity-based retrieval, allowing RAG pipelines to scale to thousands or millions of documents (HAN; LIU; WANG, 2023). The combination of high-quality embeddings with optimized indexing structures ensures that relevant context can be retrieved in real time, even under strict latency constraints.

In multilingual and educational settings, embedding selection becomes especially critical. Multilingual embedding models allow semantic alignment across languages, enabling retrieval in Portuguese corpora using natural language queries without reliance on exact keyword matching. In this work, sentence-level multilingual embeddings were adopted to support semantic search over academic dissertations, ensuring that conceptual similarity—rather than surface-level lexical overlap—guided the retrieval process.

Overall, embeddings constitute a foundational layer in modern RAG architectures, mediating between unstructured textual data and downstream reasoning capa-

bilities of language models. Their role extends beyond retrieval efficiency, shaping the semantic grounding, interpretability, and reliability of generated responses. As RAG systems continue to evolve, advances in embedding models are expected to play a decisive role in improving both retrieval fidelity and generation quality across knowledge-intensive applications.

2.4 Data Preparation and Refinement

The effectiveness of *RAG*-based systems depends heavily on how clean and useful the data are, right from the start through every step after retrieval. To ensure accuracy, relevance, and efficiency in response generation, it is essential to employ robust pre-retrieval and post-retrieval techniques that maximize the potential of available data and prevent errors or redundant information.

2.4.1 Pre-retrieval Methods

Pre-processing plays a crucial role in preparing data before it is used in the RAG pipeline because it aims to eliminate noise, structure information, and optimize conditions for efficient and relevant retrieval.

- *Data Cleaning:* Removing redundant, inconsistent, or irrelevant information ensures that only useful content is considered. This step may include eliminating duplicate data, correcting spelling errors, and normalizing formats, reducing the likelihood of retrieving irrelevant information. With cleaner inputs, search results stay focused without unnecessary noise creeping in.

- *Use of Metadata:* Organizing information through metadata, such as author, date, title, or keywords, facilitates more precise and specific searches as structured metadata allow retrieval systems to filter results based on defined criteria, increasing relevance.

- *Chunking:* This technique divides long documents into smaller, manageable blocks called *chunks*. By segmenting the data, chunking enables the system to analyze

and retrieve only the most relevant parts, optimizing the accuracy of the generated response. As a result, systems can pinpoint pertinent information more effectively.

- *Query Rewriting:* Adjusting how questions are phrased significantly improves search accuracy by modifying user questions to make them more compatible with available data. This technique may involve synonyms, semantic expansion, or the inclusion of specific terms.

- *HyDE (Hypothetical Document Embedding):* This technique generates hypothetical documents based on the user's query. Instead of merely searching for existing data, the system creates hypothetical representations that help capture the deeper semantics of the query, improving retrieval in complex scenarios or cases with little direct data correspondence (GAO et al., 2022). These fictional, yet plausible, texts serve as guides, pointing to more meaningful results when it is difficult to find exact matches.

2.4.2 Post-Retrieval Methods

Once the system pulls information matching the query, certain techniques verify its accuracy while refining what eventually reaches the model.

- *Re-ranking:* This technique reorders retrieved documents based on contextual relevance criteria. *Re-ranking* algorithms, such as MMR (CARBONELL; GOLDSTEIN, 1998), evaluate the alignment between retrieved data and the user's query, prioritizing the most relevant results in the final response.

- *Response Compression:* Compression techniques condense retrieved information, eliminating superfluous details and highlighting key points to improve the responses generated by the LLM.

- *Noise Filtering:* During post-processing, noise that may have been incorporated during retrieval is identified and eliminated. Any mistaken or off-topic parts can weaken answers, therefore removing them helps keep things clear. Making it so what remains aligns better with accurate results

The combination of *pre-processing* and *post-processing* techniques is essential for RAG systems to achieve superior performance. While *pre-processing* prepares data for efficient retrieval, *post-processing* ensures that generated responses are refined, relevant, and useful in the user's context. In contexts like education guidance, where mistakes carry weight, refining both input and output shapes outcomes for the better.

2.5 Evaluation Metrics

The evaluation of RAG systems requires a multifaceted approach that considers both retrieval quality and response generation, because these systems present unique challenges as they integrate two distinct yet interdependent stages, making the evaluation process complex and comprehensive. Yu et al. (2024) proposes the Auepora framework, a unified process aimed at standardizing and systematizing the evaluation of RAG systems. This framework encompasses retrieval, generation, and overall system evaluation, highlighting the importance of considering aspects such as relevance, accuracy, and response faithfulness, along with specific metrics that capture system efficiency.

2.5.1 Retrieval Metrics

The retrieval phase in RAG systems is critical, as it determines the quality of the information used in response generation. Traditionally, metrics such as precision and recall are widely adopted. Precision evaluates the proportion of relevant documents among those retrieved, while recall measures the system's ability to identify all relevant documents available for a given query. However, classical metrics do not always capture the complexities present in RAG systems.

Beyond these traditional metrics, Mean Reciprocal Rank (MRR) (CRASWELL, 2009) and Normalized Discounted Cumulative Gain (NDCG) (JäRVELIN; KEKäLäINEN, 2000) have been used to assess not only the presence of relevant documents but also their ranking positions. MRR considers the position of the first relevant document in the retrieved list, favoring cases where this document appears at the top. Meanwhile, NDCG accounts for document relevance based on their positions, assigning greater weight to

higher-ranked documents—an essential factor in scenarios where result ordering directly impacts the user experience.

Salemi e Zamani (2024) propose an innovative method for evaluating the retrieval phase in RAG systems, called eRAG. This approach involves applying the language model individually to each retrieved document and assessing its impact on response generation. Unlike traditional methods that evaluate retrieval independently, eRAG directly correlates retrieved documents with generated response quality, yielding more precise metrics for RAG systems. The authors demonstrate that this approach offers a higher correlation with downstream performance while also being more computationally efficient.

2.5.2 Generation Metrics

In the generation phase, evaluation must go beyond merely verifying the completeness of the generated response. Faithfulness is a central metric as it assesses whether the response remains true to the retrieved information, minimizing the occurrence of hallucinations generated by the model. This is especially relevant in critical domains such as healthcare and law, where incorrect information can have serious consequences. (YU et al., 2024) highlight that faithfulness becomes one of the main challenges in RAG system evaluation due to the dynamic nature of external information sources.

Another important metric is relevance, which evaluates how well the response aligns with the original query. To complement this analysis, classical metrics such as ROUGE and BLEU are used to measure the similarity of the generated response to a predefined reference. However, Salemi e Zamani (2024) emphasize that these metrics have limitations in scenarios where no single correct answer exists. And thus, incorporating human evaluations into the process is recommended to capture subjective aspects of response quality.

2.5.3 LLM-as-a-Judge Evaluation

The evaluation of Large Language Models (LLMs) presents unique challenges that differ substantially from traditional natural language processing tasks. Unlike classification or retrieval problems with clearly defined ground truth, many LLM applications—such

as open-ended question answering, summarization, and conversational systems—lack a single objectively correct response. This ambiguity becomes even more pronounced in RAG systems, where output quality depends not only on linguistic fluency but also on the relevance, faithfulness, and factual alignment with retrieved documents.

To address these challenges, recent research has increasingly adopted the paradigm known as *LLM-as-a-Judge*, in which a language model itself is used to evaluate the outputs of another model (or even its own outputs) according to predefined criteria. Instead of relying solely on surface-level similarity metrics such as BLEU or ROUGE, this approach leverages the reasoning capabilities of LLMs to perform nuanced assessments that approximate human judgment (ZHENG et al., 2023).

The core idea of LLM-as-a-Judge is to prompt an evaluation model with explicit instructions that define the evaluation dimensions—such as relevance, coherence, faithfulness, or completeness—and ask it to assign scores or structured feedback to a generated response. These evaluations may take the form of scalar ratings, pairwise comparisons, or categorical judgments. By formalizing evaluation criteria in natural language prompts, this paradigm enables flexible and task-specific assessment without the need for large, manually annotated datasets.

Zheng et al. (2023) demonstrate that strong language models, such as GPT-4, exhibit high agreement with human evaluators across a wide range of tasks, including dialogue quality, summarization, and instruction following. Their findings suggest that LLM-based judges can serve as reliable proxies for human judgment, particularly when evaluation prompts are carefully designed and when models are used consistently across comparisons.

In the context of RAG systems, LLM-as-a-Judge plays a particularly important role. Traditional retrieval metrics—such as Recall@k or MRR—evaluate only the retrieval stage and fail to capture whether retrieved documents were actually used correctly during generation. To address this gap, recent frameworks propose LLM-based evaluation metrics that assess the alignment between retrieved context and generated answers (YU et al., 2024). These metrics explicitly measure properties such as *faithfulness*, which evaluates whether claims made in the response are supported by the retrieved documents, and

answer relevance, which assesses whether the response adequately addresses the user's query.

One notable example is the DeepEval framework (AI, 2024), which operationalizes LLM-as-a-Judge evaluation for RAG pipelines. DeepEval provides a set of structured evaluation modules in which an LLM is prompted to judge dimensions such as context relevance, answer relevance, and faithfulness. By standardizing prompt templates and evaluation logic, the framework seeks to improve reproducibility and comparability across experiments. This approach allows researchers to move beyond purely retrieval-centric metrics and evaluate RAG systems in a manner that reflects downstream user-facing quality.

Despite its advantages, LLM-as-a-Judge evaluation also presents limitations that must be acknowledged. First, evaluation results are inherently dependent on the judge model used. Differences in model size, training data, or alignment strategies can lead to variability in judgments. Second, LLM-based evaluators may themselves exhibit biases, hallucinations, or overconfidence, particularly when evaluation prompts are ambiguous or underspecified. As noted by Liu et al. (2023), while LLM-based evaluation correlates well with human judgment, it should be interpreted as an approximation rather than an absolute measure of quality.

Another important concern is circularity: when the same or similar models are used both for generation and evaluation, there is a risk that the judge favors stylistic patterns or reasoning strategies it implicitly recognizes. To mitigate this issue, prior work recommends separating generator and evaluator models when possible, or combining LLM-based evaluation with complementary human assessments (YU et al., 2024).

In educational and decision-support contexts, where interpretability and trust are critical, LLM-as-a-Judge metrics offer a pragmatic balance between scalability and semantic depth. While human evaluation remains the gold standard for assessing pedagogical quality and real-world usefulness, LLM-based judges enable systematic, repeatable, and fine-grained analysis of system behavior across multiple dimensions. When used transparently and in conjunction with qualitative feedback, this paradigm provides a robust methodological foundation for evaluating RAG systems in complex, knowledge-intensive

domains.

Overall, LLM-as-a-Judge represents a significant shift in how language models are evaluated, aligning evaluation practices more closely with the open-ended and interpretive nature of modern AI systems. In RAG architectures, this approach is particularly valuable for capturing the interplay between retrieval and generation, offering insights that would be difficult to obtain through traditional automatic metrics alone.

2.5.4 Overall RAG System Evaluation Metrics

Evaluating the RAG system as a whole presents additional challenges due to the complex interaction between retrieval and generation components. Latency is a fundamental metric as it assesses the time required to complete the retrieval and generation process. In real-time applications, system agility is crucial to ensuring a good user experience.

Other important metrics include noise robustness, which measures the system's ability to handle irrelevant information without compromising response quality, and negative rejection, which evaluates the system's ability to refrain from providing answers when insufficient information is available. The CRUD-RAG benchmark, proposed by Lyu et al. (2024), offers a comprehensive approach to evaluating RAG systems by categorizing their applications into scenarios of information creation, reading, updating, and deletion. The authors argue that integrating these metrics is essential for assessing not only the system's technical performance but also its adaptability to different scenarios and demands.

2.6 Vector Databases

Vector databases (VDBs) are specialized systems designed to store and retrieve high-dimensional vectors—dense numeric representations of complex data such as texts, images, and audio. These vectors are typically generated through embedding models and serve as semantic encodings of unstructured content (HAN; LIU; WANG, 2023). Unlike traditional databases that rely on exact matches, VDBs enable similarity-based queries using vector distance metrics (e.g., cosine similarity, Euclidean distance), which makes them essential in tasks involving semantic search and information retrieval.

In the context of RAG, vector databases serve a critical role by enabling real-time search for the most semantically relevant documents based on the user's query embedding. This allows the system to retrieve meaningful external context to augment the language model's generation process, mitigating issues such as hallucinations and stale knowledge.

Vector databases are optimized for approximate nearest neighbor search (ANNS), a performance-critical operation that balances retrieval accuracy and speed in high-dimensional spaces. Common indexing techniques include hashing (e.g., locality-sensitive hashing), tree-based partitioning (e.g., KD-trees), and graph-based methods (e.g., HNSW), all aimed at accelerating similarity search while managing large volumes of data (HAN; LIU; WANG, 2023). Storage optimizations—such as quantization, compression, and partitioning—are also widely adopted to ensure scalability and efficiency.

Modern Vector Database Management Systems (VDBMSs) are composed of a query processor and a storage manager. These components handle hybrid queries (e.g., combining vector similarity with structured filters), manage index updates, and integrate with systems like RAG pipelines seamlessly. Popular systems such as FAISS, Milvus, and Weaviate exemplify the practical adoption of VDBs in real-world AI applications (PAN; WANG; LI, 2023).

Thus, vector databases are a foundational component for enabling RAG architectures, providing both the infrastructure for fast and relevant document retrieval and the flexibility to scale across diverse data domains.

2.7 Related Work

RAG has been increasingly applied to facilitate access to textual repositories across various domains, enhancing how users interact with large-scale information. These systems combine traditional information retrieval with generative models, allowing users to pose natural language questions and receive synthesized responses grounded in retrieved documents. In healthcare, for example, RAG has been used to assist clinicians in accessing medical literature and ensuring traceability of sources, as seen in systems like Med-R2 (LU et al., 2025). In the educational domain, applications range from supporting learning analytics (LóPEZ-PERNAS et al., 2024) to enhancing student engagement with academic

content (THüS; MALONE; BRüNKEN, 2024).

2.7.1 Domain-Specific RAG Systems

Lu et al. (2025) system, Med-R2, exemplify domain-specific RAG tailored to high-stakes, codified environments like healthcare. It integrates evidence-based medical documents, applies hierarchical reranking, and emphasizes citation fidelity. However, these characteristics reflect the structured nature of clinical data and may not transfer directly to more heterogeneous domains like educational policy.

In education, RAG applications span several goals. López-Pernas et al. (2024) introduced LARAG, a system that supports educators by extracting insights from a local collection of learning analytics literature. LARAG is built on the Kotaemon framework[1] and enables conversational queries. However, its implementation focused on a fixed dataset of 136 studies. Unlike our proposed architecture, which prioritizes modularity and hybrid retrieval to handle growing repositories, LARAG represents a more static approach to document interaction.

Other educational RAG systems focus on pedagogical engagement. Thüs, Malone e Brünken (2024) developed a tool to simplify access to academic content for students, aiming to reduce cognitive load. While these approaches contribute to the broader theme of knowledge access, their design is centered around the learning process, distinct from the organizational and strategic needs of school managers targeted in this work.

2.7.2 RAG Frameworks and General-Purpose Tools

Beyond specific academic implementations, the development of RAG systems is supported by an evolving ecosystem of frameworks and commercial tools. As highlighted in recent surveys, technology stacks like LangChain and LlamaIndex have become standard for orchestrating retrieval and generation pipelines (LI et al., 2025). More recently, frameworks such as RAGFlow have emerged, focusing on "Deep Document Understanding" to better parse complex unstructured data before retrieval. These frameworks provide the building

[1]Kotaemon is an open-source RAG UI for querying documents. Available at: https://github.com/Cinnamon/kotaemon

blocks for constructing custom solutions but require architectural decisions to adapt to specific domains like educational research.

In the realm of end-user applications, Google's NotebookLM has popularized the concept of "grounded generation" by allowing users to interact with personal documents through a conversational interface. While NotebookLM demonstrates the potential of LLMs to synthesize uploaded content effectively, it operates as a general-purpose tool for individual use. In contrast, institutional repositories require systems designed for scalability, diverse user queries, and specific domain alignment, which off-the-shelf tools may not fully address.

2.7.3 Contribution

Our work contributes to this landscape by targeting school leaders and educational policymakers, a group often overlooked in favor of student-centric solutions. We propose and evaluate three distinct RAG architectures—Naive, Advanced, and Modular—designed to help non-academic stakeholders interact with complex research through synthesized and verifiable answers.

In contrast to static or purely conversational implementations found in related work, our most robust architecture (Modular RAG) introduces active control mechanisms, including query rewriting, cross-encoder reranking, and citation verification. These components are specifically designed to address the challenges of hallucination and context loss in long academic documents.

Furthermore, we provide an empirical evaluation of the trade-offs between system complexity, latency, and response quality in the educational domain. Rather than claiming to automate decision-making—a complex social process—our system aims to reduce the friction in information retrieval, fostering exploratory analysis and professional reflection in school governance.

Future research should continue to explore scalable, transparent, and interactive RAG pipelines, potentially incorporating the hybrid retrieval methods discussed in the theoretical framework to further enhance recall.

3 Proposed System and Architecture

This chapter describes the RAG system proposed in this research as a computational artifact. The objective is to document, at an algorithmic level, the design decisions, data processing pipeline, and retrieval-generation architectures developed in this work. Rather than focusing on evaluation outcomes, the emphasis is on how the system was constructed, why specific choices were made, and how its components interact, providing sufficient detail to support reproducibility.

3.1 Design Decisions and System Motivation

The system was designed to reduce the distance between academic research and educational practice by enabling natural language interaction with a corpus of academic dissertations in public education management and evaluation. Unlike generic conversational agents, the proposed system explicitly prioritizes grounded responses supported by verifiable academic sources.

This motivation led to three central design decisions. First, Retrieval-Augmented Generation was adopted instead of a standalone language model, ensuring that responses are conditioned on an external and auditable knowledge base. Second, citation control mechanisms were treated as first-class components of the system, rather than as a post-processing feature, enforcing explicit links between generated statements and source documents. Third, multiple RAG architectures were implemented to explore different trade-offs between simplicity, contextual control, and computational cost.

To isolate architectural effects, all configurations share the same corpus, embedding model, vector database, and language model. Differences between systems arise solely from how retrieval, memory, prompting, and verification stages are orchestrated.

3.2 Academic Corpus and Data Processing Pipeline

3.2.1 Corpus Composition

The knowledge base used by the system consists of 1,065 academic dissertations produced within the Professional Graduate Program in Public Education Management and Evaluation (PPGP) at the Center for Public Policies and Educational Evaluation (CAEd/UFJF). These dissertations are publicly available and follow a relatively standardized academic structure, including sections such as introduction, contextual description, theoretical analysis, Educational Action Plan (PAE), and conclusions.

This corpus was selected because it represents a coherent domain with shared terminology, methodological assumptions, and practical orientation toward educational policy and management, making it suitable for evaluating retrieval-augmented educational assistants.

3.2.2 Text Extraction and Segmentation

Each dissertation was processed through a standardized preprocessing pipeline. Raw text was extracted from PDF files, while essential metadata—such as title, author, year, and source URL—was preserved to enable traceability and citation.

The extracted text was segmented into semantically coherent chunks of approximately 250 to 500 tokens. This range was chosen as a compromise between semantic completeness and retrieval granularity: smaller chunks improve recall and precision during dense retrieval, while larger chunks preserve argumentative coherence necessary for interpretative questions.

Although the documents follow a common internal structure, all text segments were treated uniformly during indexing. This design choice ensured architectural neutrality during evaluation and avoided introducing section-specific biases into retrieval behavior.

3.2.3 Embeddings and Vector Database

Each text chunk was converted into a dense vector representation using the `paraphrase-multilingual-mpnet-base-v2` model from the SentenceTransformers library. This model was selected due to its strong performance in semantic similarity tasks and explicit support for Portuguese.

All embeddings were stored in a `ChromaDB` [2] vector database, which serves as the sole retrieval layer for all system configurations. `ChromaDB` was chosen for its simplicity, reproducibility, and efficient support for dense similarity search in medium-scale academic corpora.

3.2.4 Language Model Selection

Text generation across all architectures was performed using the `mistralai/mistral-small-3.2-24b-instruct` model, accessed via the `OpenRouter API`. This model offers a balance between linguistic competence and computational feasibility, enabling multi-stage prompting (e.g., summarization, verification, and decision agents) within realistic latency constraints.

By fixing the language model across architectures, the study isolates the impact of retrieval and control mechanisms from underlying model capabilities.

3.3 Retrieval-Augmented Generation Architectures

This section describes the three RAG architectures developed in this work, progressing from a minimal baseline to a fully modular pipeline. Figures 3.1, 3.2, and 3.3 illustrate the respective architectures.

3.3.1 Naive RAG Architecture

The Naive RAG architecture constitutes the baseline system and implements a single-stage retrieval–generation pipeline. For each user query, the system performs dense similarity

[2] ChromaDB is an open-source vector database designed for building AI applications with embeddings. Available at: ⟨https://www.trychroma.com/⟩

search over the vector database and retrieves the top 20 most similar text chunks.

These chunks are concatenated into a single context block and passed directly to the language model, together with a minimal instruction prompt:

"Responda com base no contexto fornecido."

Conversational memory is managed exclusively through trimming: when the message history exceeds a predefined limit, older turns are discarded. No summarization, query rewriting, reranking, or verification is applied.

As illustrated in Figure 3.1, this architecture prioritizes transparency and low latency, serving as a reference point for evaluating more complex designs.

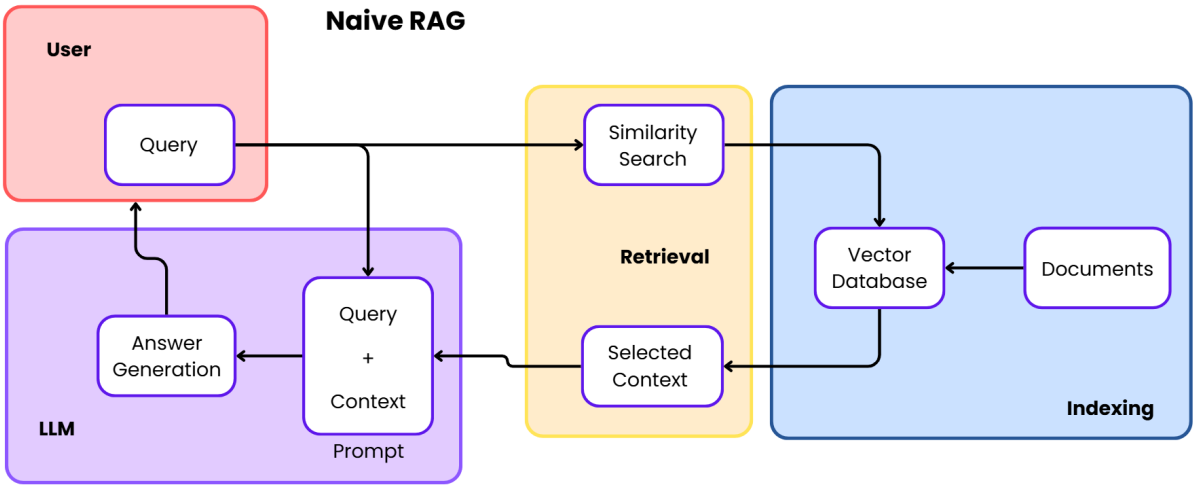

Figure 3.1: Naive RAG architecture.

3.3.2 Advanced RAG Architecture

The Advanced RAG architecture extends the baseline by introducing two mechanisms aimed at improving contextual continuity and retrieval quality: conversational summarization and document reranking.

When the conversation history exceeds a fixed length, older turns are summarized into a compact system message using the language model itself. The summarization is performed with the explicit prompt:

"Resuma a conversa acima em 2–3 frases, extraindo fatos e contextos que seriam úteis para responder perguntas futuras."

This summary replaces the older messages and is prepended to subsequent interactions, allowing relevant context to be preserved while controlling token usage.

Retrieval initially returns the top 20 candidates via dense similarity search. These candidates are then reranked using a cross-encoder model (`cross-encoder/ms-marco-MiniLM-L6-v2`), and only the top 10 passages are included in the generation context.

The generation prompt is further refined by introducing a domain-specific role instruction:

> *"Você é um assistente especializado na área de educação. Responda com base no contexto fornecido."*

Figure 3.2 depicts this architecture, highlighting the added summarization and reranking stages.

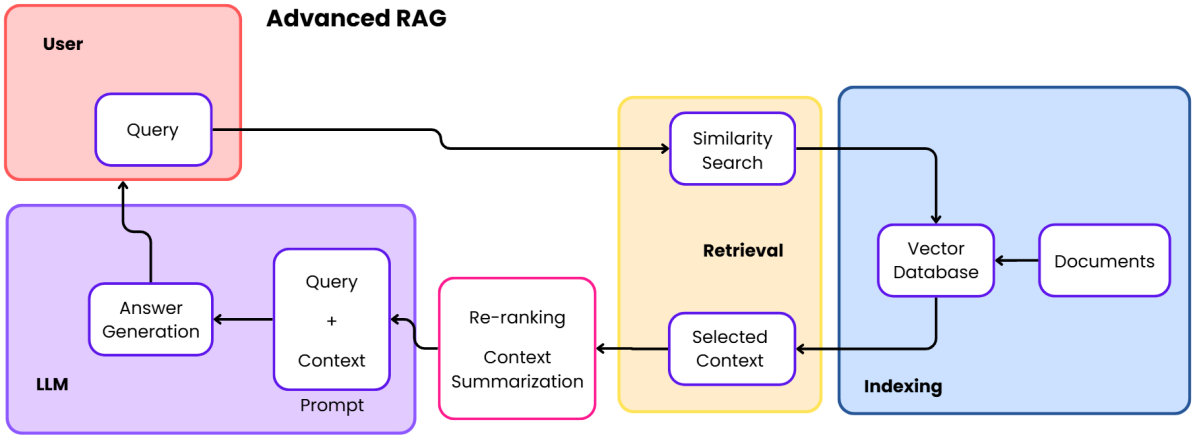

Figure 3.2: Advanced RAG architecture.

3.3.3 Modular RAG Architecture

The Modular RAG architecture represents the most elaborate configuration and decomposes the RAG pipeline into explicit, independently controlled stages.

The process begins with a query rewriting stage, in which the language model generates up to two alternative formulations of the user question to improve recall. The rewriting prompt is defined as:

"Reescreva a pergunta abaixo em até 2 variações mais claras e completas. Responda apenas com uma lista simples."

Next, a retrieval decision agent evaluates whether new document retrieval is necessary, based on the current question, the conversation history, and the accumulated context. This agent is also implemented using the language model and produces a binary decision.

If retrieval is triggered, documents are fetched using all query variations and reranked with the same cross-encoder employed in the Advanced architecture. The top-ranked passages form the generation context.

After response generation, a verification stage prompts the language model to assess whether the generated answer is consistent with the retrieved context. Finally, a citation validation module enforces coherence between in-text citations and the reference list.

As shown in Figure 3.3, this architecture emphasizes control, interpretability, and factual grounding, at the cost of increased computational complexity and latency.

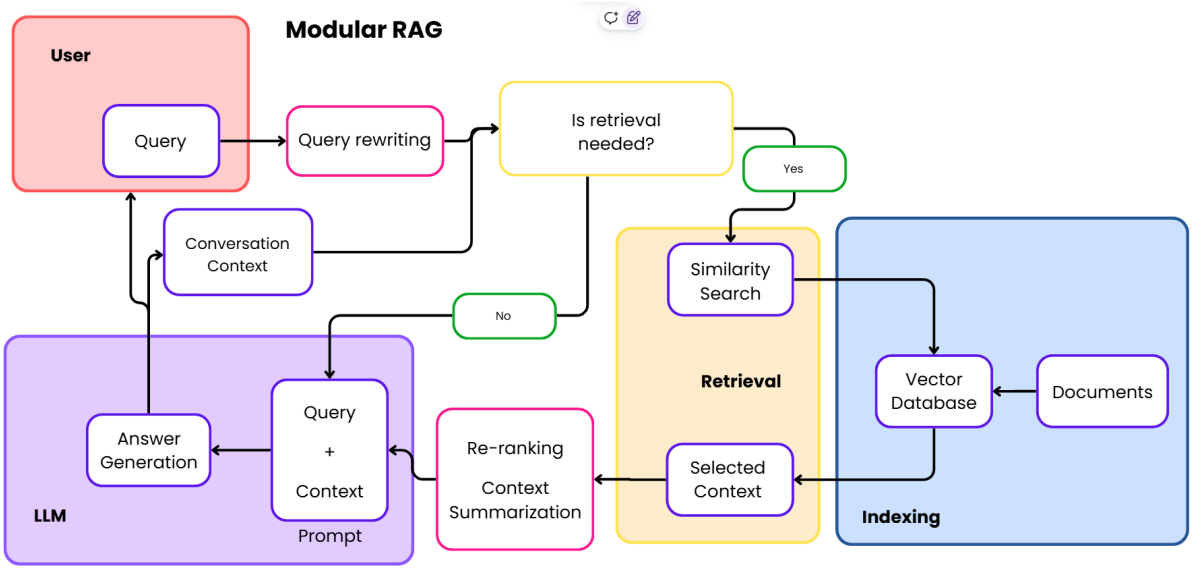

Figure 3.3: Modular RAG architecture.

3.4 Design Trade-offs and Limitations

Each architecture embodies distinct trade-offs. The Naive RAG favors simplicity and efficiency but offers limited support for long-term contextual coherence. The Advanced RAG improves continuity and retrieval precision with moderate additional complexity. The Modular RAG provides the highest level of control and transparency, but introduces additional latency and system overhead.

Although the corpus exhibits internal structural regularities, all text segments were treated uniformly during retrieval. Leveraging document structure—such as prioritizing abstracts or Educational Action Plans—represents a promising direction for future work.

4 Materials and Methods

This chapter describes the methodological and technical foundations that guided the development, implementation, and evaluation of the proposed RAG system. The goal is to provide a transparent and reproducible account of the procedures, tools, and configurations used throughout the study, as well as to explain how each experimental stage was designed to address the research objectives.

The chapter is organized into four main sections. The first section presents the computational infrastructure used for implementation and testing. The second section details the development of the three RAG architectures—*Naive*, *Advanced*, and *Modular*—describing their design principles, pipelines, and generation mechanisms. The third section explains the experimental methodology, the construction of the vector database and how the qualitative evaluation was carried out with the participants. Finally, the last section introduces the quantitative evaluation framework, and outlines the metrics used to assess retrieval performance, response quality, and system latency.

Together, these materials and methods establish the empirical and computational basis for the analyses discussed in the following chapters, ensuring that the study's findings can be interpreted within a clear methodological context.

4.1 Computational Infrastructure

All experiments were conducted in a virtualized computing environment configured for the development, evaluation, and deployment of the RAG architectures. The environment operated on a virtual machine hosted on an Intel® Core™ i9-9900K processor with four virtual CPUs running at 3.60 GHz. All computations were performed on the CPU, as no dedicated GPU was used.

The system ran on Ubuntu 16.04.7 LTS and used Python 3.12 for all experimental workflows. Key software components included the Hugging Face `Transformers` library for model implementation, `ChromaDB` for vector storage and retrieval, and `Streamlit` for

the web-based evaluation interface.

All modules were containerized using Docker and orchestrated with Docker Compose to ensure reproducibility and environment isolation. Large language model inference was performed through the `OpenRouter API`[3], a service that provides a standardized interface to access multiple model backends, including those used for retrieval and generation stages. Version control was managed with GitHub to maintain code traceability and enable collaborative development.

4.2 Experimental Methodology

This section details the methodological procedures adopted to construct and validate the proposed question-answering system. The methodology is structured into three main stages: the development of distinct RAG architectures to allow for comparative analysis; the selection and configuration of the underlying language models and vector database; and the integration of these components into a functional full-stack system architecture.

4.2.1 Development of RAG Architectures

Three RAG architectures were developed and evaluated to compare different configurations of retrieval and generation processes: *Naive RAG*, *Advanced RAG*, and *Modular RAG*. All architectures were implemented in Python using the `LangChain` and `Hugging Face` libraries, integrated into a FastAPI back-end and a Streamlit front-end interface. Their detailed design and internal pipelines are described in Chapter 3.

The purpose of this phase was to evaluate how architectural differences—particularly in context handling, query processing, and document selection—affect the quality, faithfulness, and perceived usefulness of generated responses under controlled and interactive conditions.

During the experiments, each architecture was exposed to the same set of queries, documents, and model configurations. Differences in behavior therefore stem exclusively from architectural design choices rather than variations in data, embeddings, or language

[3]OpenRouter is a unified interface that aggregates access to various large language models from different providers. Available at: ⟨https://openrouter.ai/⟩

models.

4.2.2 Models and Vector Database

All three RAG architectures employed the same underlying language and embedding models to ensure comparability across experiments. Text generation was carried out using the `mistralai/mistral-small-3.2-24b-instruct` model, accessed through the `OpenRouter API`. This model was selected because it offers an adequate balance between computational efficiency and linguistic performance, producing coherent and contextually appropriate responses while remaining feasible to deploy within the hardware constraints available for this project.

For semantic representation, the `paraphrase-multilingual-mpnet-base-v2` model from the `SentenceTransformers` library was adopted. This multilingual transformer-based model supports Portuguese and has been optimized for semantic similarity and retrieval tasks in multiple languages.

The vector database was implemented using `ChromaDB`, an open-source vector store optimized for dense retrieval and semantic search. This database constituted the core retrieval layer for all three architectures. It was populated with academic dissertations from the *Professional Graduate Program in Public Education Management and Evaluation* (PPGP) at CAEd/UFJF, publicly available at ⟨https://mestrado.caedufjf.net/dissertacoes/⟩. In total, 1,065 dissertations were collected in PDF format and processed through a structured data preparation pipeline designed to ensure both semantic coherence and retrieval efficiency.

The data preprocessing process began with the extraction of raw text from each PDF, during which essential metadata such as title, author, and year of publication were preserved to maintain traceability and enable citation in responses. The extracted text was then segmented into semantically coherent units, typically ranging from 250 to 500 tokens. Following segmentation, each text fragment was converted into a dense vector representation using the embedding model and stored in `ChromaDB`.

Although the dissertations followed a relatively standardized internal structure, all textual segments were treated uniformly during indexing in this study. This design

choice ensured architectural neutrality during evaluation, while also highlighting opportunities for future work to incorporate structure-aware retrieval strategies.

4.2.3 System Architecture

The back-end system was implemented in FastAPI[4] and managed conversational history, routing of user inputs to the corresponding RAG architecture, integration with the language model, and communication with the vector database. The front-end was developed in Streamlit[5] and guided participants through a structured evaluation workflow.

This interface supported both fixed-question assessments and open-ended interaction, enabling controlled data collection while preserving a realistic conversational setting for qualitative evaluation.

4.3 User-Based Evaluation Design

A user-based evaluation was conducted to assess the practical performance of the proposed RAG architectures in an educational tutoring context. This evaluation combined quantitative perception-based measures with qualitative feedback. The objective of this stage was to determine which architecture best supports the intended use case: providing accurate, pedagogically relevant, and user-friendly responses to educators engaging in queries about educational assessment.

The qualitative experiment was implemented as an interactive web application. Data generated during the experiment were automatically logged to both a local `SQLite` database (for conversational interactions) and synchronized to `Google Sheets` (for aggregated analysis).

The experiment was structured into five stages, as described on Figure 4.1.

In the identification stage, each participant informed their name or pseudonym to allow organizational tracking of participation and no personally sensitive or identifying data were collected beyond this field. The unique identifier allowed subsequent linkage be-

[4]FastAPI is a modern, high-performance web framework for building APIs with Python 3.8+ based on standard Python type hints. Available at: ⟨https://fastapi.tiangolo.com/⟩

[5]Streamlit is an open-source Python library that turns data scripts into shareable web apps. Available at: ⟨https://streamlit.io/⟩

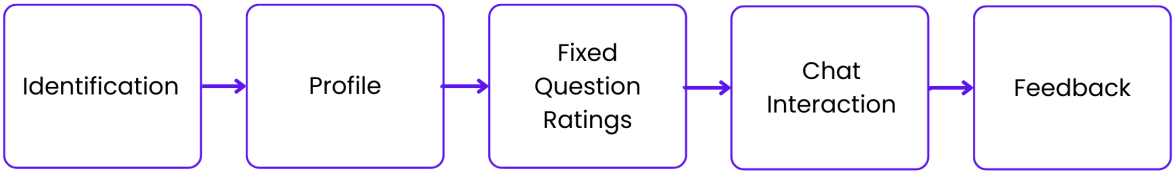

Figure 4.1: Overview of the qualitative evaluation process.

tween profile data, evaluation responses, and chat interactions, enabling intra-participant analysis.

Figure 4.2: Screens from the identification and profile stages of the qualitative evaluation.

The second stage collected basic profile information from each participant, including their level of formal education and prior familiarity with educational evaluation concepts. These variables serve as contextual indicators for interpreting the results: they help determine whether the perceived quality of the responses varied according to participants' domain knowledge. For instance, a participant with professional experience in school assessment may judge factual precision more rigorously than a participant with limited prior exposure.

The third stage of the experiment involved a controlled evaluation of predefined responses generated by each of the three RAG architectures. Participants were presented with five fixed questions related to themes in educational evaluation. For each question, three distinct responses were shown, each produced independently by one of the architectures—*Naive*, *Advanced*, or *Modular*. No indication was given as to which model had generated each one, ensuring that the evaluation process remained fully blind and unbiased.

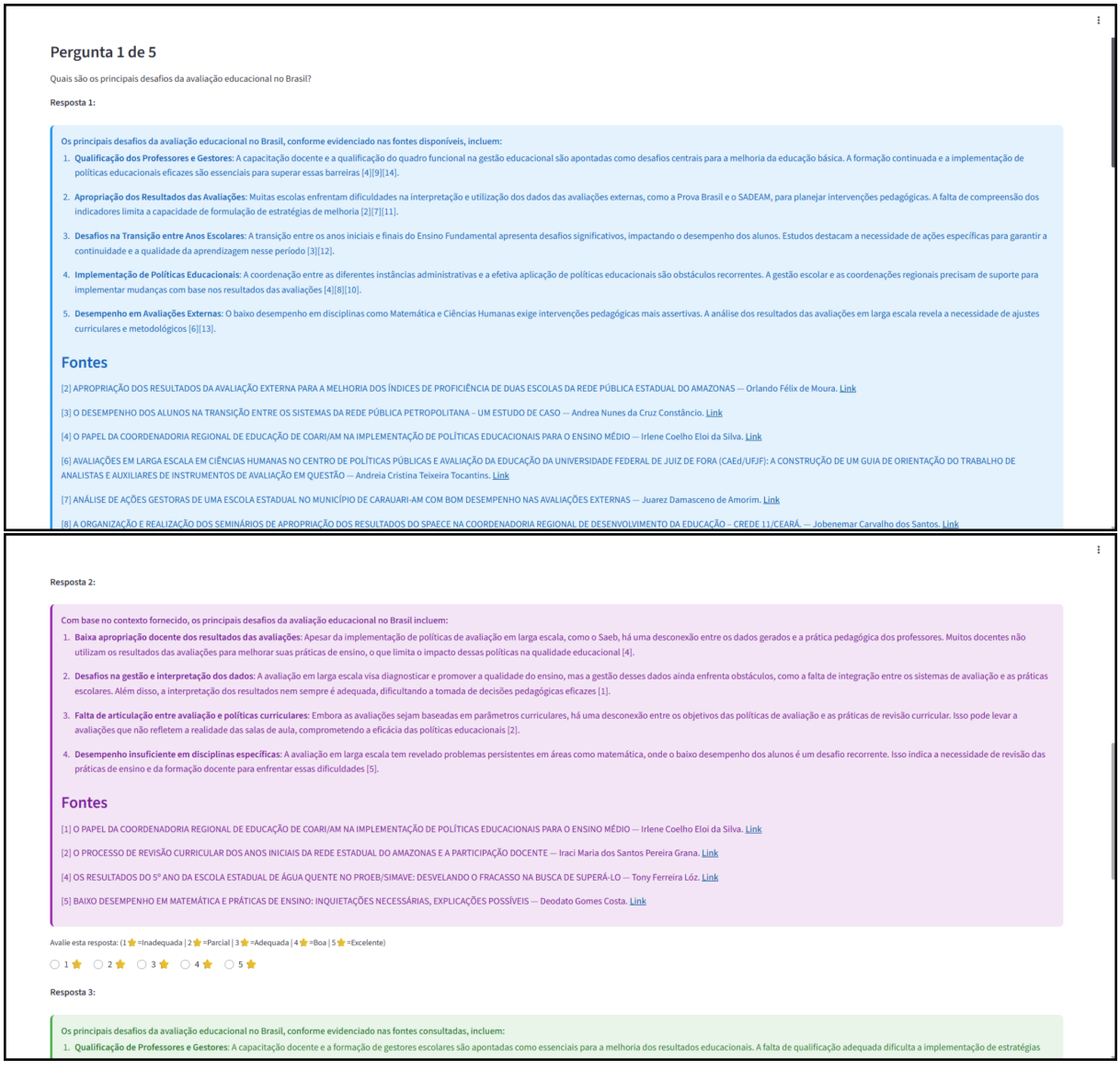

Figure 4.3: Screens illustrating the fixed-question evaluation phase, including examples of the response evaluation interface.

Participants rated each response on a five-point Likert scale:

- 1 – Inadequate or incorrect;

- 2 – Partially correct, with conceptual or linguistic issues;
- 3 – Adequate, but incomplete or lacking depth;
- 4 – Good and useful;
- 5 – Excellent and highly relevant.

This scoring system was designed to capture perceptions of factual accuracy, conceptual depth, linguistic clarity, and overall usefulness of each answer. By applying the same set of questions across all architectures, this stage established a standardized baseline for comparative analysis of output quality.

The set of predefined questions was carefully selected to cover a range of levels of specificity and thematic scope within the domain of educational assessment. The questions included both broad, general topics and highly focused queries derived from particular academic dissertations. This design aimed to test the systems' ability to generate coherent and informative responses across different levels of complexity and contextual grounding.

The five original questions in Portuguese and a English translation for each are presented below:

1. *Quais são os principais desafios da avaliação educacional no Brasil?*

What are the main challenges of educational assessment in Brazil?

2. *De que forma o uso de tecnologias digitais tem contribuído para aprimorar os processos de ensino, aprendizagem e avaliação na educação básica?*

How has the use of digital technologies contributed to improving teaching, learning, and assessment processes in basic education?

3. *Quais fatores têm sido identificados pela literatura como determinantes para as elevadas taxas de reprovação em escolas públicas estaduais?*

Which factors have been identified in the literature as determinants of high failure rates in state public schools?

4. *Quais foram as principais causas do abandono escolar no Ensino Médio noturno em escolas públicas estaduais, segundo a literatura?*

According to the literature, what were the main causes of school dropout in night-shift upper secondary education in state public schools?

5. *Que conclusões podem ser extraídas acerca da eficácia escolar em escolas públicas estaduais, com base nos indicadores de desempenho (por exemplo, IDEB, taxas de aprovação/reprovação e fluxo)?*

What conclusions can be drawn about school effectiveness in state public schools based on performance indicators (e.g., IDEB, approval/failure rates, and student flow)?

Each participant's evaluation was stored with metadata including the question text, the hidden architecture label, the numerical rating, and a timestamp, forming the dataset later used for statistical and interpretative analysis.

Following the fixed-response evaluations, participants proceeded to an open-ended chat environment where they could freely interact with a tutor powered by one of the three RAG architectures. This stage was designed to capture the systems' performance under natural conversational conditions, allowing participants to ask questions related to educational evaluation and assess the system's fluency, coherence, and perceived reliability.

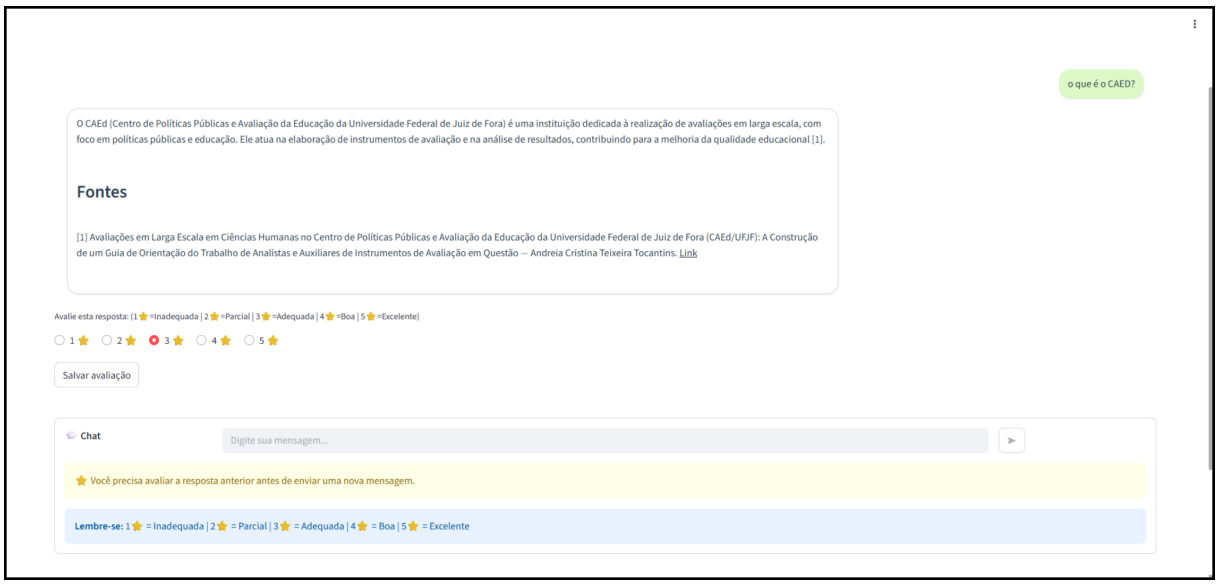

Figure 4.4: Screen from the interactive chat showing message evaluation components.

Each participant interacted with only one of the three architectures—*Naive*, *Advanced*, or *Modular*—to ensure that qualitative impressions could be analyzed indepen-

dently. The assignment of participants to architectures followed a mixed procedure combining controlled distribution and partial randomization. Invitations containing unique system links were distributed among selected participants, each link pre-configured to activate one of the architectures via URL parameters. This method ensured that the three systems could be tested under comparable conditions, while maintaining flexibility for broader participation.

In this interactive phase, participants could ask any number of questions related to educational evaluation, mirroring realistic use of an intelligent tutoring system. After each response, they rated the output using the same 1–5 scaled applied previously. These data provide insight into how well each architecture maintains coherence, relevance, and factual grounding in dynamic conversational settings. Moreover, this stage captures how architectural differences manifest in user experience: latency tolerance, fluency, adaptability to follow-up questions, and perceived reliability.

All conversations, including user messages, generated responses, ratings, and timestamps, were logged locally via a `SQLite` database and backed up to a `Google Sheets` table.

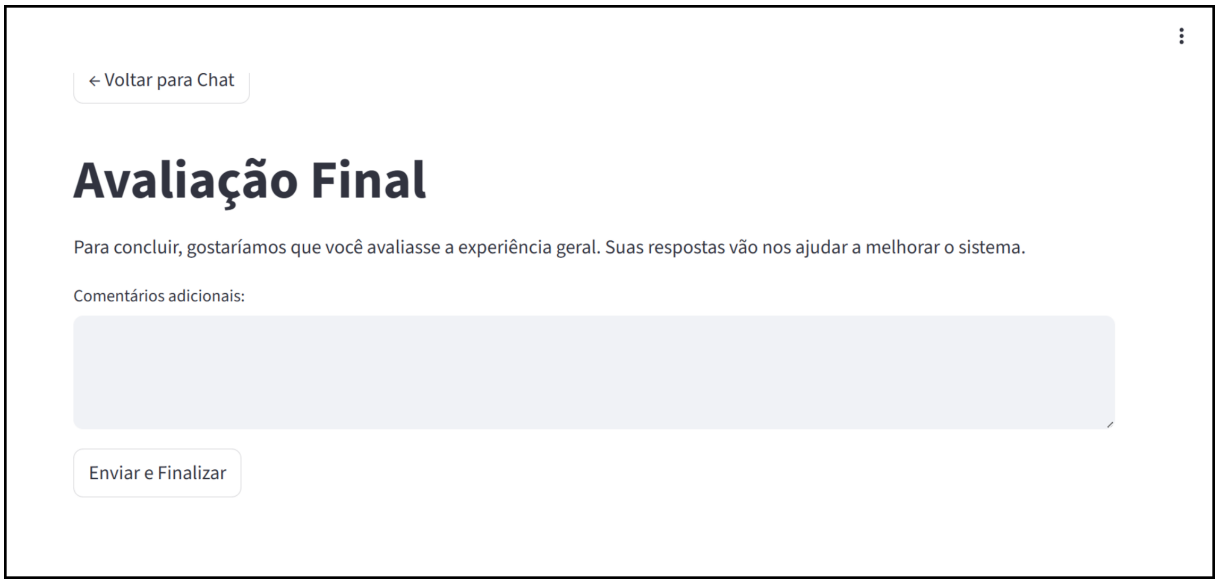

Figure 4.5: Feedback screen from the final stage of the qualitative evaluation

The final stage collected open-ended qualitative feedback through written text. Participants were invited to share their impressions regarding clarity, usefulness, and overall satisfaction with the tutor's responses, as well as any perceived strengths or weaknesses.

This information provides crucial qualitative evidence to contextualize quantitative ratings.

4.4 Automated System Evaluation Design

Quantitative system evaluation was conducted using an automated assessment pipeline that queried each RAG architecture through a dedicated API endpoint and computed retrieval behavior, response quality, and latency metrics. Each query was evaluated independently for the Naive, Advanced, and Modular architectures, and results were aggregated by architecture type. Response-quality metrics were computed using DeepEval (AI, 2024), an open-source framework for LLM evaluation.

Crucially, retrieval-based metrics serve more as signs of how well a system keeps to the original query, instead of strict judgments on relevance. Because certain architectures intentionally rephrase or break down queries, reduced metric scores might reflect shifts in meaning - rather than an inability to retrieve results.

4.4.1 Retrieval and Query Fidelity Metrics

To analyze how effectively each architecture retrieves information aligned with the user query, two complementary metrics were employed: Recall@5 and Mean Reciprocal Rank (MRR).

Recall@5 quantifies the proportion of relevant text segments—also referred to as "oracle-relevant chunks"—that appear among the top five retrieved results for each query. In the context of this study, a higher Recall@5 value indicates that the system consistently retrieves portions of dissertations semantically related to the user's question. This measure is particularly useful for evaluating whether an architecture maintains the original semantic focus of the query during the retrieval process. However, architectures that deliberately reformulate queries, such as the Modular RAG, may show lower Recall@5 scores due to the expansion of semantic scope, even while retrieving contextually meaningful evidence.

Mean Reciprocal Rank (MRR), in turn, measures how early the first oracle-

relevant document appears within the retrieved list. It captures the ability of the retrieval component to prioritize the most relevant information near the top of the ranking. High MRR values suggest that the system's retrieval ordering is semantically faithful to the user's intent.

Together, Recall@5 and MRR provide a comprehensive picture of each architecture's query fidelity, showing not only how much relevant content is retrieved but also how well it is ranked for immediate use by the generator.

4.4.2 Response-Quality Metrics

While retrieval metrics focus on the search stage, response-quality metrics evaluate the quality of the generated text itself; specifically, how accurately, coherently, and faithfully the model integrates retrieved information into its answers. These metrics are essential for understanding whether architectural enhancements in RAG systems actually lead to more informative and trustworthy responses, rather than merely altering retrieval behavior.

Three complementary indicators were used in this analysis: Context Relevancy, Faithfulness, and Answer Relevancy.

Context Relevancy, computed using DeepEval's `ContextualRelevancyMetric`, measures the semantic relationship between the retrieved context and the user query. It indicates whether the model's evidence base truly aligns with the informational need expressed in the question. In practical terms, high Context Relevancy suggests that the system retrieves material that meaningfully contributes to answering the query rather than tangential or unrelated passages.

Faithfulness, computed using DeepEval's `FaithfulnessMetric`, assesses factual consistency between the generated answer and the supporting context. This metric plays a critical role in reducing the risk of "hallucinations". A high Faithfulness score indicates that the model remains grounded in verifiable evidence, a particularly important property for educational applications where factual accuracy underpins trust and pedagogical value.

Finally, Answer Relevancy, derived from DeepEval's `AnswerRelevancyMetric`, evaluates how directly and completely the generated response addresses the user's question. This metric captures the communicative effectiveness of the system, combining

linguistic adequacy with semantic focus. High Answer Relevancy reflects that the model not only generates correct information but does so in a way that is clear and useful.

When analyzed together, these three response-quality metrics provide a holistic view of system performance, revealing how retrieval accuracy translates into educationally valuable output.

4.4.3 Latency Measurement

Latency was measured as the full round-trip time of each API request, including retrieval, reasoning, and generation steps. Reported values correspond to the mean latency per architecture across the evaluation set.

5 Results and Analysis

The analysis of results was divided into two complementary dimensions: (i) quantitative system-level metrics derived from automated evaluation scripts and (ii) user-based evaluation results, which include both quantitative perception-based ratings (Likert-scale responses) and qualitative feedback

The results were obtained from the mean of the execution and testing of five questions corresponding to the same five questions present in the fixed step of qualitative evaluation.

5.1 User Study Design

5.1.1 Recruitment

Participants were recruited through two complementary strategies. Expert participants were selected from the Center for Public Policies and Educational Evaluation (CAEd/UFJF), based on their professional experience with educational assessment and research. Invitations were sent directly to selected CAEd researchers and practitioners, and participation was voluntary, resulting in a subset of experts who agreed to take part in the study.

The remaining participants were recruited from UFJF and consisted primarily of undergraduate and graduate students involved in projects related to education, language technologies, or information retrieval, many of whom collaborate with CAEd or participate in adjacent research initiatives. These participants were invited during an academic meeting in which the project was presented, after which interested students volunteered to participate in the evaluation.

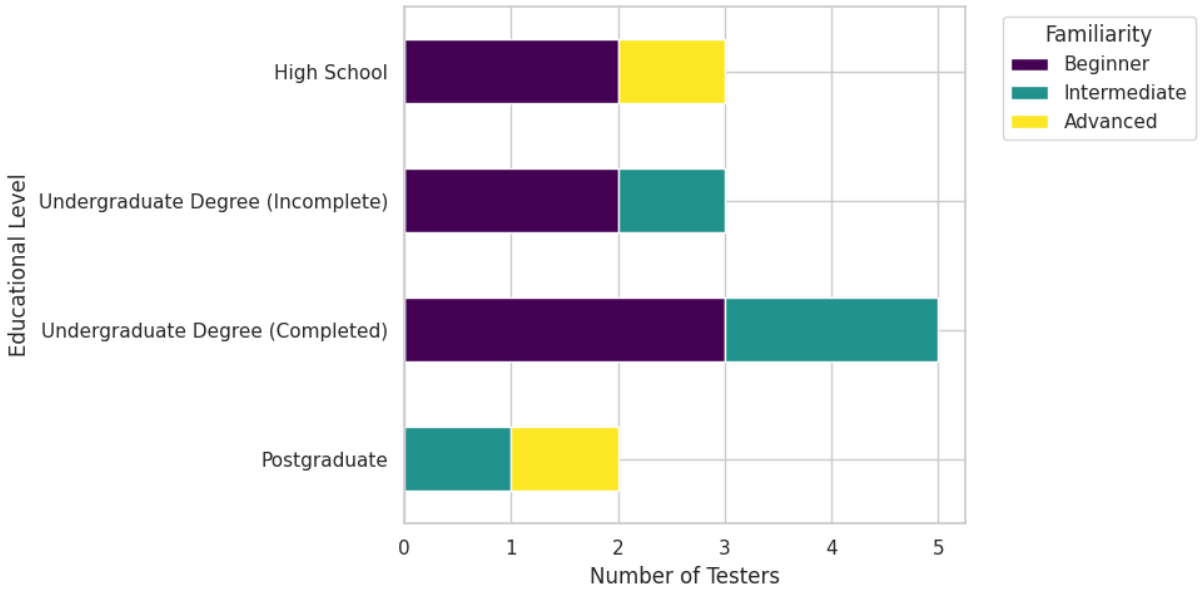

Figure 5.1: Distribution of Familiarity with Educational Assessment by Educational Level

5.1.2 Participant Profile

Of the 13 participants, 3 had completed secondary education, 5 had completed higher education, 3 were enrolled in undergraduate programs, and 2 held postgraduate degrees. Regarding familiarity with educational evaluation, 7 identified as beginners, 4 as intermediate, and 2 as advanced. This range of profiles ensured coverage of both expert and novice user perspectives.

The distribution of participants' educational levels and familiarity with educational evaluation is presented in Figure 5.1. As shown, the sample includes both novice and expert users, allowing the analysis to capture a broad range of perspectives.

5.2 Quantitative Analysis

5.2.1 Automated System-Level Metrics

Table 5.1 summarizes the mean performance of each architecture across all metrics.

The Naive and Advanced architectures achieve identical Recall@5 and MRR values, indicating strong preservation of the semantic ranking induced by the original user query. This result is expected, as both architectures rely primarily on dense retrieval over the original query, with the Advanced architecture improving efficiency and ranking

Table 5.1: Quantitative evaluation results for RAG architectures.

Architecture	Recall@5	MRR	Context Relevancy	Faithfulness	Answer Relevancy	Latency (s)
Advanced	0.50	1.000	0.439	0.600	0.600	30.82
Modular	0.18	0.573	0.660	0.933	0.925	34.86
Naive	0.50	1.000	0.640	0.800	0.800	31.16

through query refinement and cross-encoder reranking without substantial semantic drift.

In contrast, the Modular architecture exhibits lower Recall@5 and MRR, reflecting deliberate semantic reformulation introduced by query rewriting. Rather than indicating retrieval degradation, this behavior highlights a strategic divergence from the original query representation.

Despite lower query fidelity metrics, the Modular architecture substantially outperforms the other systems in response-quality measures. It achieves the highest context relevancy, faithfulness, and answer relevancy scores, demonstrating superior grounding and alignment between retrieved evidence and generated responses. These results suggest that the modular design enables more effective selection and utilization of supporting information, even when that information does not closely match the original query's semantic neighborhood.

Latency differences between architectures were small, with the Modular system incurring a moderate overhead due to additional reasoning stages. However, this increase is accompanied by a significant gain in answer quality, indicating a favorable tradeoff between reasoning depth and responsiveness.

Overall, the results demonstrate that while retrieval fidelity metrics favor simpler architectures, response-quality metrics strongly favor the Modular RAG. This highlights a fundamental tradeoff between semantic preservation and reasoning effectiveness, reinforcing the importance of evaluating RAG systems beyond retrieval-centric metrics alone.

5.2.2 Quantitative User-Based Ratings

The user-based evaluation complements the automated quantitative findings by examining user perceptions of response quality, clarity, and interaction experience through numerical

ratings. Thirteen participants with diverse educational backgrounds and levels of expertise in educational assessment participated in the study.

Fixed Question Ratings

Participants evaluated five standardized questions, each accompanied by three unlabeled responses (one from each architecture) and rated them on a five-point Likert scale from 1 (very poor) to 5 (excellent). This stage provided a controlled comparison of response quality across architectures under identical prompts.

Table 5.2 summarizes the mean rating, standard deviation, and number of evaluations for each architecture. The discrepancy between the total recruited participants ($N = 13$) and the effective sample size ($N = 7$) resulted from the study's data persistence mechanism. Ratings were only committed to the database upon the user's explicit transition to the next question block. Consequently, participants who engaged with the evaluation but exited the session without clicking the navigation button resulted in unrecorded interactions. Furthermore, since the interface permitted partial evaluations within a block, slight variations in total response counts occurred across architectures (e.g., 28 vs. 30).

The results show comparable performance among the three systems. While the *Naive RAG* configuration achieved a slightly higher mean, the high standard deviations relative to the means suggest substantial variability in user satisfaction across individual questions. Consequently, no single architecture demonstrated a definitive advantage in this controlled setting.

Table 5.2: Mean and Standard Deviation of Ratings by Architecture for Fixed Questions

Architecture	**Mean Rating**	**Standard Deviation**	**N (responses / participants)**
Advanced	3.50	0.90	28 / 7
Modular	3.29	1.30	30 / 7
Naive	3.57	1.14	28 / 7

The data in Table 5.2 indicates that participant evaluations were fairly homogeneous across architectures. The proximity of the mean scores suggests that, on average, the added complexity of the Modular and Advanced architectures did not translate into a perceived improvement in response quality for this specific set of fixed questions when

evaluated by the general sample.

When considering only the evaluations provided by participants affiliated with CAEd/UFJF, who possess advanced expertise in educational assessment, a different pattern emerges, as shown in Table 5.3. First, the mean ratings were notably lower across all architectures compared to the general sample, indicating that expert evaluators tended to be more critical of the system's outputs. Second, unlike the general sample where the Naive architecture scored highest, the *Advanced RAG* received the highest mean rating among experts.

Table 5.3: Mean and Standard Deviation of Ratings by Architecture for Fixed Questions (Expert Participants Only)

Architecture	**Mean Rating**	**Standard Deviation**	**N (responses / participants)**
Advanced	3.20	0.87	10 / 2
Modular	2.60	1.11	10 / 2
Naive	2.80	0.75	10 / 2

It is important to note the limited sample size for the expert group ($N = 10$ responses from 2 participants), which prevents statistical generalization. However, the observational data suggests that domain expertise may influence which architectural traits are valued, with experts showing a slight preference for the Advanced configuration in this limited test, diverging from the general preference for the Naive baseline.

Chat Interaction Ratings

In the open chat phase, participants interacted freely with one assigned architecture. This stage aimed to evaluate the systems' behavior in natural conversational settings.

Although the experiment was designed to balance participation across architectures, actual adherence varied significantly. Of the participants who received direct invitations, only those assigned to the *Modular RAG* completed the full chat interaction stage. To compensate for this, open access links were shared publicly, but this resulted in an uneven distribution of responses, with the majority of interaction data concentrated in the Modular configuration.

These deviations resulted in very small sample sizes for the Naive and Advanced architectures in the chat phase, constraining the comparability of results. The data pre-

sented below should therefore be interpreted as descriptive of specific interaction sessions rather than indicative of overall system performance.

Overall, the mean chat rating across all interactions was **3.91** (SD = 0.46), with participants sending an average of **5.5 messages** per session. Table 5.4 summarizes the descriptive statistics for the recorded interactions.

Table 5.4: Mean Chat Ratings and Interaction Counts by Architecture

Architecture	**Mean Rating**	**Standard Deviation**	**Interactions (Users)**
Naive	3.50	1.73	8 (2)
Advanced	4.33	0.47	3 (1)
Modular	4.09	0.79	11 (1)

Table 5.5 shows the rating assigned to each message across architectures. While satisfaction levels appear high for the Advanced and Modular sessions, the low number of unique users (indicated in Table 5.4) means these scores may reflect the experience of single individuals rather than a general trend.

Table 5.5: Mean Score by Message Index and Architecture in Chat Evaluation

Message	**Naive**	**Advanced**	**Modular**
1	2.50 ± 2.12	4.00 ± 0.00	5.00 ± 0.00
2	5.00 ± 0.00	4.00 ± 0.00	5.00 ± 0.00
3	2.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
4	5.00 ± 0.00	–	4.00 ± 0.00
5	5.00 ± 0.00	–	4.00 ± 0.00
6	1.00 ± 0.00	–	4.00 ± 0.00
7	–	–	3.00 ± 0.00
8	–	–	3.00 ± 0.00
9	–	–	5.00 ± 0.00
10	–	–	4.00 ± 0.00
11	–	–	3.00 ± 0.00

5.3 Qualitative Analysis

5.3.1 Qualitative Feedback Analysis

Open-ended responses were analyzed qualitatively to gain insight into how participants perceived the system's usefulness, response quality, and usability. A lightweight thematic analysis was conducted by identifying recurring patterns and concerns across participant comments.

Overall, participants viewed the tool as promising for supporting academic research and facilitating the exploration of dissertation repositories, particularly in making complex academic information more accessible. Several users highlighted the potential of such a system to serve as a bridge between academic production and practical application in education. However, some participants observed that the system occasionally produced overly verbose answers or diverged slightly from the main topic, issues reported more frequently in the *Naive RAG* configuration. In addition, a few users mentioned minor interface inconsistencies and navigation bugs, suggesting that future versions could benefit from improvements in layout and interaction design. One recurrent suggestion was to make the source references collapsible, thereby enhancing readability and reducing visual clutter during extended interactions. Together, these comments emphasize both the perceived value of the system as a research and learning aid and the need for refinements to optimize user experience and interaction clarity.

5.4 Interpretation

The combination of quantitative system metrics, quantitative user-based ratings, and qualitative feedback provides an initial, though limited, understanding of how the different RAG architectures behave in both controlled and interactive settings. While the Naive system displayed consistent efficiency and clarity in producing straightforward answers, the Modular architecture appeared to offer richer, contextually grounded responses during extended interactions. However, these observations should be interpreted with caution, as the qualitative phase—particularly the open chat evaluation—was exploratory in nature

and based on a small and uneven number of participants.

The differences observed across architectures may therefore reflect individual interaction styles or varying familiarity with the system rather than intrinsic differences in model design. Moreover, the sample included participants with highly diverse profiles, ranging from undergraduate students to experts in educational assessment. The small number of experts who completed the full evaluation also limits the generalizability of the findings. Notably, their stricter ratings, compared to the general participant pool, suggest that professional evaluators focus more on factual consistency and conceptual rigor, whereas less experienced users may prioritize clarity and fluency.

These constraints mean that the qualitative component should not be interpreted as a definitive comparison of architectures, but rather as an exploratory step toward understanding how different system designs shape user experience and perceived reliability. Despite these limitations, the results still offer valuable insights into user expectations and highlight important design trade-offs: while simpler RAG systems are efficient and easy to interpret, modular approaches seem more suitable for deeper, tutoring-oriented interactions that demand higher contextual retention.

Overall, these findings suggest that RAG architectures hold promise for bridging the gap between academic research and educational practice, but that future evaluations must involve larger, more balanced participant groups and more structured task protocols. Expanding participation among education professionals and policy practitioners would be especially valuable to assess how such systems might support evidence-based decision-making in real institutional contexts.

6 Conclusion

This research aimed to tackle a central challenge in education: the difficulty of effectively accessing and interpreting the vast volume of academic research produced in the field. In a context where scientific production continues to expand rapidly, much of this knowledge remains underused by those who could most benefit from it, such as teachers, school leaders, and policymakers. By developing and evaluating a Retrieval-Augmented Generation (RAG) system, this study explored how artificial intelligence can be used to bridge this gap, transforming static academic repositories into interactive tools for inquiry.

Three architectures were designed and implemented—*Naive RAG*, *Advanced RAG*, and *Modular RAG*—each representing a distinct pipeline of retrieval and generation. Their comparative evaluation aimed to understand how technical design choices affect not only retrieval performance but also the utility and quality of the generated responses as a support for professional study.

Quantitative analyses revealed a trade-off between retrieval fidelity and interpretative depth. The *Naive RAG* achieved the highest recall and ranking metrics, showing efficiency in preserving the semantic intent of queries through direct dense retrieval. The *Modular RAG*, on the other hand, excelled in metrics related to faithfulness, contextual relevance, and answer quality, demonstrating stronger alignment between retrieved evidence and generated responses. The *Advanced RAG* showed balanced behavior, benefiting from summarization and reranking mechanisms.

These results provide partial support for the study's hypothesis regarding the system's utility as a research tool. Architectural complexity appears to enhance contextual reasoning and faithfulness, suggesting that while simpler architectures suffice for quick factual lookups, modular configurations are more appropriate for dialogic or interpretative uses, such as tutoring or synthesizing academic concepts.

The qualitative evaluation complemented these findings by revealing how users experienced the system in practice. Among the participants, the overall perception was that the RAG tutor effectively supported the exploration of academic themes and im-

proved the understanding of complex topics found in the dissertations.

User feedback offered valuable design lessons. Participants appreciated the system's potential to democratize access to the dissertation collection but pointed out usability issues, such as verbosity in some answers. These observations indicate that in educational AI tools, clarity and ease of interaction are as vital as retrieval precision.

The evaluation process faced specific limitations. The work focused on the system's technical and perceived ability to retrieve and synthesize information, but it did not measure the downstream impact on actual administrative decisions or policy implementation. Furthermore, the small and unbalanced sample in the user study limits the generalizability of the qualitative findings. Differences in expertise between specialists and general participants also influenced how response quality was perceived.

From a development standpoint, several technical insights emerged. Query rewriting in the *Modular RAG*, though designed to enhance recall, sometimes led to semantic drift. Similarly, the dense retrieval strategy did not fully exploit the structured nature of the repository. Future systems could leverage document structure to prioritize more informative segments, such as abstracts and Educational Action Plans (PAEs).

Building on these findings, future work should expand both the system and its evaluation. Larger and more balanced participant samples would allow for more robust statistical analysis. Integrating larger models could improve fluency, while hybrid retrieval could increase efficiency. On the interface side, dynamic reference visualization may help improve readability.

Beyond technical enhancements, future studies should focus on the practical integration of such tools in educational settings. Integrating systems like this into professional development programs could enable educators to access and interpret academic findings more easily. This step would realize the broader goal of democratizing access to academic knowledge through AI.

Rather than claiming to solve the complexity of educational decision-making, this study provides evidence that RAG-based systems can play a meaningful role in connecting academic research to professional practice. By grounding responses in verifiable sources and offering conversational access to complex research, they present a promising pathway

for transforming academic repositories into active environments for learning and reflection.

In the end, the process of building and testing this system demonstrated that facilitating access to knowledge is not only a technical challenge but a human one. It requires designing systems that respect the pace, curiosity, and constraints of real users. Artificial intelligence in education should not aim merely to automate information retrieval, but to foster understanding and evidence-informed inquiry. Through thoughtful design, tools like the one developed here can help ensure that the growing body of academic knowledge truly reaches those who shape the future of education.

Bibliography

AI, C. *DeepEval: An open-source evaluation framework for LLMs*. 2024. Version 0.21.2 (accessed December 2025). Disponível em: ⟨https://github.com/confident-ai/deepeval⟩.

BORGMAN, C. L. *Big Data, Little Data, No Data: Scholarship in the Networked World*. The MIT Press, 2015. ISBN 9780262327862. Disponível em: ⟨https://doi.org/10.7551/mitpress/9963.001.0001⟩.

CARBONELL, J.; GOLDSTEIN, J. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: . [S.l.: s.n.], 1998.

CRASWELL, N. Mean reciprocal rank. In: _____. *Encyclopedia of Database Systems*. Boston, MA: Springer US, 2009. p. 1703–1703. ISBN 978-0-387-39940-9. Disponível em: ⟨https://doi.org/10.1007/978-0-387-39940-9_488⟩.

DEVLIN, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. Disponível em: ⟨https://arxiv.org/abs/1810.04805⟩.

GAO, L. et al. Precise zero-shot dense retrieval without relevance labels. In: . [S.l.: s.n.], 2022.

GAO, Y. et al. *Retrieval-Augmented Generation for Large Language Models: A Survey*. 2024. Disponível em: ⟨https://arxiv.org/abs/2312.10997⟩.

GOODMAN, J. *A Bit of Progress in Language Modeling*. 2001. Disponível em: ⟨https://arxiv.org/abs/cs/0108005⟩.

HAN, Y.; LIU, C.; WANG, P. *A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge*. 2023. Disponível em: ⟨https://arxiv.org/abs/2310.11703⟩.

JäRVELIN, K.; KEKäLäINEN, J. Ir evaluation methods for retrieving highly relevant documents. In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, 2000. (SIGIR '00), p. 41–48. ISBN 1581132263. Disponível em: ⟨https://doi.org/10.1145/345508.345545⟩.

JI, Z. et al. Survey of hallucination in natural language generation. *ACM Computing Surveys*, Association for Computing Machinery (ACM), v. 55, n. 12, p. 1–38, mar. 2023. ISSN 1557-7341. Disponível em: ⟨http://dx.doi.org/10.1145/3571730⟩.

KARPUKHIN, V. et al. *Dense Passage Retrieval for Open-Domain Question Answering*. 2020. Disponível em: ⟨https://arxiv.org/abs/2004.04906⟩.

LEVIN, B. To know is not enough: research knowledge and its use. *Review of Education*, v. 1, n. 1, p. 2–31, 2013. Disponível em: ⟨https://bera-journals.onlinelibrary.wiley.com/doi/abs/10.1002/rev3.3001⟩.

LEWIS, P. et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 2021. Disponível em: ⟨https://arxiv.org/abs/2005.11401⟩.

LI, Z. et al. Retrieval-augmented generation for educational application: A systematic survey. *Computers and Education: Artificial Intelligence*, Elsevier, v. 8, p. 100417, 2025.

LIU, Y. et al. *G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment*. 2023. Disponível em: ⟨https://arxiv.org/abs/2303.16634⟩.

LU, K. et al. *Med-R^2: Crafting Trustworthy LLM Physicians through Retrieval and Reasoning of Evidence-Based Medicine*. 2025. Disponível em: ⟨https://arxiv.org/abs/2501.11885⟩.

LYU, Y. et al. *CRUD-RAG: A Comprehensive Chinese Benchmark for Retrieval-Augmented Generation of Large Language Models*. 2024. Disponível em: ⟨https://arxiv.org/abs/2401.17043⟩.

LóPEZ-PERNAS, S. et al. Augmenting ai with curated learning analytics literature: Building and initial exploration of a local rag for supporting teachers (larag). ceur-ws.org, 2024. Query date: 2025-04-16 02:24:56. Disponível em: ⟨https://ceur-ws.org/Vol-3938/Paper_1.pdf⟩.

MALLEN, A. et al. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. [S.l.: s.n.], 2023. p. 9802–9822.

MIKOLOV, T. et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. Disponível em: ⟨https://arxiv.org/abs/1301.3781⟩.

MUENNIGHOFF, N. *SGPT: GPT Sentence Embeddings for Semantic Search*. 2022. Disponível em: ⟨https://arxiv.org/abs/2202.08904⟩.

OPENAI. Gpt-4 technical report. In: . [s.n.], 2023. Disponível em: ⟨https://cdn.openai.com/papers/gpt-4.pdf⟩.

PAN, J. J.; WANG, J.; LI, G. *Survey of Vector Database Management Systems*. 2023. Disponível em: ⟨https://arxiv.org/abs/2310.14021⟩.

REIMERS, N.; GUREVYCH, I. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. Disponível em: ⟨https://arxiv.org/abs/1908.10084⟩.

ROBERTSON, S.; ZARAGOZA, H. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, Now Publishers Inc., Hanover, MA, USA, v. 3, n. 4, p. 333–389, abr. 2009. ISSN 1554-0669. Disponível em: ⟨https://doi.org/10.1561/1500000019⟩.

SALEMI, A.; ZAMANI, H. *Evaluating Retrieval Quality in Retrieval-Augmented Generation*. 2024. Disponível em: ⟨https://arxiv.org/abs/2404.13781⟩.

TENOPIR, C. et al. Changes in data sharing and data reuse practices and perceptions among scientists worldwide. *PLOS ONE*, Public Library of Science, v. 10, n. 8, p. 1–24, 08 2015. Disponível em: ⟨https://doi.org/10.1371/journal.pone.0134826⟩.

THüS, D.; MALONE, S.; BRüNKEN, R. Exploring generative ai in higher education: a rag system to enhance student engagement with scientific literature. *Frontiers in Psychology*, Volume 15 - 2024, 2024. ISSN 1664-1078. Disponível em: ⟨https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2024.1474892⟩.

VASWANI, A. et al. *Attention Is All You Need*. 2023. Disponível em: ⟨https://arxiv.org/abs/1706.03762⟩.

YU, H. et al. *Evaluation of Retrieval-Augmented Generation: A Survey*. 2024. Disponível em: ⟨https://arxiv.org/abs/2405.07437⟩.

ZHANG, N. et al. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*, 2024.

ZHAO, H. et al. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, ACM New York, NY, v. 15, n. 2, p. 1–38, 2024.

ZHENG, L. et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. 2023. Disponível em: ⟨https://arxiv.org/abs/2306.05685⟩.