Journal of the Brazilian Computer Society, 2025, 31:1, doi: 10.5753/jbcs.2025.5806

Complex Interactions in Dialog Systems for Brazilian Portuguese: A Comparison of RAG Approaches

Felipe Coelho de Abreu Pinna [**Universidade de São Paulo** | ***fpinna@larc.usp.br***]
Victor Takashi Hayashi [**Universidade de São Paulo** | ***victor.hayashi@usp.br***]
João Carlos Néto [**Universidade de São Paulo** | ***joaocarlos@larc.usp.br***]
Isabella Sadakata Takara [**Universidade de São Paulo** | ***isabella.takara@usp.br***]
Stephan Kovach [**Universidade de São Paulo** | ***skovach@larc.usp.br***]
Lucas Gaspar Mendonça [**Universidade de São Paulo** | ***lucas.gm@usp.br***]
Romeo Bulla Junior [**Universidade de São Paulo** | ***romeo@larc.usp.br***]
João Victor Sá [**Universidade de São Paulo** | ***sajoaovictor@usp.br***]
Wilson Vicente Ruggiero [**Universidade de São Paulo** | ***wilson@larc.usp.br***]

Laboratory of Computer Architecture and Networks (LARC), Polytechnic School (EPUSP), Universidade de São Paulo, Av. Prof. Luciano Gualberto, 380, Butantã, São Paulo, SP, 05508-010, Brazil.

Received: 31 March 2025 • **Accepted:** 29 July 2025 • **Published:** 31 October 2025

Abstract Retrieval-Augmented Generation (RAG) has emerged as a key technique in enhancing the capabilities of Large Language Models (LLMs) by incorporating external knowledge sources into the response generation process. This paper presents a comparative analysis of various RAG approaches applied to dialog systems in Brazilian Portuguese. The study explores multiple retrieval strategies, including VectorRAG, GraphRAG, MemoRAG, HybridRAG, and HippoRAG, assessing their performance in handling complex queries, multi-turn conversations, and contextual disambiguation.We evaluate these models in the banking context using real-world datasets from two case studies. The analysis highlights the strengths and limitations of each method.Experimental results indicate that context-aware retrieval strategies improve response accuracy when addressing ambiguous or multi-faceted user queries. However, trade-offs in computational efficiency and response time remain critical challenges. Our findings provide insights into optimizing dialog systems for Brazilian Portuguese, paving the way for domain-specific conversational agents in financial and other specialized applications.

Keywords: Banking, Dialog Systems, Finance, Generative AI, LLM, RAG, Brazilian Portuguese.

1 Introduction

The emergence of Large Language Models (LLM) with remarkable language processing and generation capabilities have attracted enormous academic and industry interest. These deep learning language models are trained in large textual datasets and many of them are available as open-source models that can be used in various scenarios, such as dialog systems Hadi *et al.* [2023]; Liu *et al.* [2023].

In this context, Retrieval-Augmented Generation (RAG) has emerged as a key technique in enhancing the capabilities of Large Language Models (LLMs) by incorporating external knowledge sources into the response generation process. While the alternative approach finetuning requires model modification based on a specialized dataset, RAG uses external databases to specialize in specific knowledge domains without altering the LLM itself Zhang *et al.* [2024]; Zhao *et al.* [2024].

When working with small textual resources to answer simple queries in a dialog system, RAG is an efficient way to incorporate particular information of specific domains to a pre-trained LLM, and it is more cost-effective when comparing to the finetuning process Zhang *et al.* [2024]. However, when there are several textual resources of a specific knowledge domain, the standard RAG process based solely on vector representations is not enough to support complex interactions. More advanced RAG approaches are required to support short-term and long-term memory aspects to LLMs.

Among the specific Natural Language Processing challenges, there are significant regional variations, frequent idiomatic expressions, and difficulties in dealing with complex financial contexts due to specific terminology and regulatory nuances. Therefore, a detailed analysis comparing different RAG approaches becomes essential to validate their practical effectiveness in real scenarios. Therefore, the Research Question considered in the present work is: ‘Which RAG approach offers the best performance in complex queries in Brazilian Portuguese, considering the criteria of accuracy, efficiency and adaptability to the banking domain?’. To support work in a feasible research scope, the proof of concept was designed for one specific language and the banking specific domain.

This paper presents a comparative analysis of various RAG approaches applied to dialog systems development. The study explores multiple retrieval strategies, including VectorRAG, GraphRAG, MemoRAG, HybridRAG, and HippoRAG, assessing their performance in handling complex queries, multi-turn conversations, and contextual disambiguation. While most complex RAG approaches are applied in English, this is one of the first works to present results of applications in Brazilian Portuguese Edge *et al.* [2024]; Peng *et al.* [2024];

Qian *et al*. [2024]; Gutiérrez *et al*. [2024].

We evaluate these models in the banking context using real-world datasets from two case studies. The analysis highlights the strengths and limitations of each method. Experimental results indicate that context-aware retrieval strategies improve response accuracy when addressing ambiguous or multi-faceted user queries. However, trade-offs in computational efficiency and response time remain critical challenges. Our findings provide insights into optimizing dialog systems for Brazilian Portuguese, paving the way for domain-specific conversational agents in financial and other specialized applications.

The text is organized as follows: section 2 presents the main concepts of Information Retrieval, Large Language Models and Retrieval-Augmented Generation; section 3 contains the literature review with a critical analysis of the most relevant related work. Section 4 presents the dialog system framework based on a previous study, and how LLM and different RAG approaches were used in the present work. The case studies in the banking domain are presented in section 5. Based on the experiments and dialog system development, the final proposed architecture to support complex queries is presented in section 6. Lessons learned and research limitations are presented in section 7. The final considerations and directions for future work are included in section 8.

2 Research Background

This research was conducted to understand the context and features involved in language models, with a focus on Large Language Models (LLMs). Advances in this area have allowed the extraction of concepts, facts, and relationships from texts, facilitating the decomposition and analysis of complex dialogues, which are fundamental for the interaction between humans and machines Ozdemir [2023]. Given these advances, it is essential to analyze how language models can be integrated into efficient information search and organization mechanisms.

In this context, three fundamental elements stand out: Information Retrieval – which allows locating and extracting relevant knowledge from large volumes of data –, Large-Scale Language Models (LLMs) – which process and generate text based on deep learning – and Retrieval-Augmented Generation (RAG) – a hybrid approach that combines information retrieval and text generation to improve the accuracy and contextualization of responses). These concepts are rapidly explored in the following sections.

2.1 Fundamental Concepts

Information Retrieval (IR) can be defined as finding unstructured data within a large volume of data to satisfy a specific information need. Traditionally, IR systems, such as search engines and textual databases, use methods based on indexing, vector retrieval and relevance ranking, according to Manning [2009].

Figure 1 shows an overview of an IR process:

1. **Query:** a user's query is submitted to an IR system that contains a collection of documents;
2. **First Pre-Processing:** these documents are previously analised for standardizing and cleaning data. It goes through processes that include tokenization, specific selected stop-words removal and stemming;
3. **Indexing:** the textual informations are so indexed, and then inserted into the system's Index;
4. **Second Pre-Processing:** when executing the query, the same pre-processing operations are performed on the text;
5. **Searching and Ranking:** the pre-processed text of the query is then used to search the index for the documents that best meet the query;
6. **Results:** the query results are then returned to the user, according to Moreira [2023].

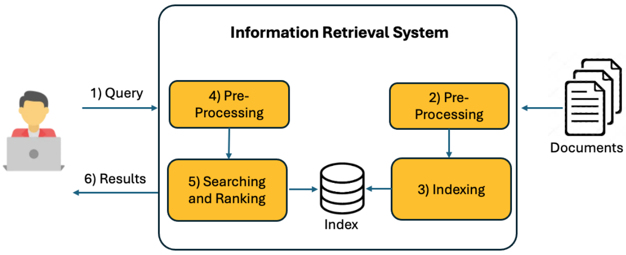


Figure 1. Overview of the Information Retrieval process, illustrating key stages from query submission to results presentation. Highlights preprocessing, indexing, and ranking as core components influencing retrieval effectiveness (based on Moreira [2023]).

In addition, Information retrieval models can define methodologies for comparing queries with documents and determining their relevance, such as Boolean, Vector and Probabilistic Models. Due to the Large Language Models having driven profound transformations in NLP, recent research has explored strategies to improve IR systems by incorporating LLMs, expanding their capabilities and redefining the boundaries of the field, according to Zhu *et al*. [2023].

Large Language Models (LLMs) represent a significant advance in the field of Natural Language Processing. They have billions of parameters and are capable of performing a wide range of NLP tasks, according to Ozdemir [2023]. Furthermore, the development of various new architectures based on Transformers allowed training models on large volumes of textual data.

In the article "Attention Is All You Need" from Vaswani *et al*. [2017], the Transformers were introduced with the innovation of the **self-attention**. They examine all words in parallel, assigning different weights to each one based on contextual relevance.

Transformer-based LLM models such as BERT and GPT are trained in two phases: pre-training and fine-tuning. In pre-training, models are trained on a massive internet dataset to predict the next word in a text sequence; in the fine-tuning phase, the model processes user messages along with trainer responses to improve the model's results, according to Zhang *et al*. [2024].

Currently, there are several private and open-source LLMs with varied applications, according to Radford *et al*. [2019]. In the private segment, we can mention Claude 3.5 from Anthropic, GPT-4o from OpenAI and Gemini Advanced from

Google. In the open-source segment, models such as LLaMA 3.1 from Meta and Falcon 180B from TII offer accessible and customizable alternatives for researchers and companies, according to Almazrouei *et al*. [2023]. Each model has specific advantages, from greater control over data to better computational efficiency, depending on the user's needs. In this way, LLMs are redefining the limits of Information Retrieval (IR), enabling more fluid and personalized interactions.

Despite the advances in LLMs, there are significant challenges in effectively deploying them in specialized domains. This is because the knowledge of LLMs is limited to public data up to a specific point in time when it was trained. To reason on private data or data introduced after a model was trained, the model's knowledge must be augmented with the specific information it needs.

In that way, **Retrieval-Augmented Generation (RAG)** addresses this limitation by allowing one to extend the capabilities of LLMs to specific domains or to an organization's internal knowledge base without the need to retrain the model. It is a proposed approach that aims to improve LLM output so that it remains accurate in multiple contexts. The main differentiator of RAG is its ability to search for information in external knowledge bases before generating a response, according to Zhao *et al*. [2024].

RAG can be understood through the following steps, as shown in Figure 2:

1. Given a query, for example, a user's question;
2. The Retrieval System searches for relevant information in an external knowledge base;
3. This search returns relevant information to obtain additional context;
4. The query and additional context are passed to the language model. This step aims to prevent the model from relying solely on pre-trained knowledge, thus contributing to reduce hallucinations;
5. With the information received, the language model generates a contextualized response based on the retrieved data;
6. This response is passed to the user's terminal.

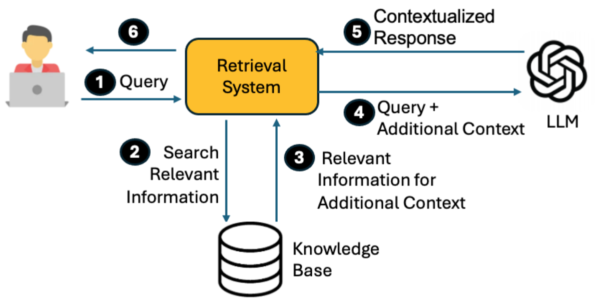

6
1 Query
Retrieval System
5 Contextualized Response
4 Query + Additional Context
LLM
2 Search Relevant Information
3 Relevant Information for Additional Context
Knowledge Base

Figure 2. Diagram of the Retrieval-Augmented Generation (RAG) process, showing how context is retrieved and incorporated into LLM-based answer generation. Emphasizes the modular design enabling updated, accurate responses without retraining the model.

However, RAG has some limitations, among which it is important to mention: how to deal with noisy retrieval results, manage computational overhead, and address the inherent complexity of these systems. Despite this, the Retrieval-Augmented Generation technique represents a significant advance in the area of Natural Language Processing by combining prior learning with updated facts.

2.2 Evaluation Metrics

In a RAG system, there are two fundamental components: Retrieval and Generation. The Retrieval component is responsible for searching external knowledge sources, identifying and retrieving the most relevant information, while the Generation component processes this retrieved data to construct the responses. One key objective of a RAG system is to generate coherent and contextually relevant responses.

The evaluation of a RAG system aligns with these two components and can be divided into two phases: the first focused on assessing the retrieval quality, analyzing how well the system selects relevant and useful information, and the second phase measuring the accuracy and relevance of the generated content, ensuring that the final response is both reliable and contextually appropriate Yu *et al*. [2024].

For this, the metrics below can be applied to measure de quality of the information retrieved and the answer generated, based on the RAGAS framework Shahul Es [2023]:

- **Context Precision**: it measures the proportion of retrieved documents that are actually relevant to answering the question. It is calculated as the mean of the P_n (ratio of the number of relevant chunks at rank n to the total number of chunks at rank n) for each chunk in a given context:

$$CP = \frac{\sum_{n=1}^{N} P_n \times v_n}{\text{Total of relevant items in the top N results}} \tag{1}$$

, where

$$P_n = \frac{\text{Relevant chunks at rank k}}{\text{Total number of chunks at rank k}} \tag{2}$$

N is the total number of chunks in the all query, and v_n is the relevance indicator at rank n.

- **Faithfulness**: this metric assesses whether the response generated by the model is faithful to the information retrieved. It is a real value from 0 to 1: higher scores correspond to a better consistency. In this context, an response is so faithful as all its claims are supported by the retrieved context.
To calculate this metric, it is necessary to identify all the statements that make up the answer and check, for each statement, whether it can be inferred from the context retrieved. Then, it can be calculate the Faithfulness score Fa.

$$Fa = \frac{\text{Number of claims supported by the context}}{\text{Total number of claims in the response}} \tag{3}$$

- **Answer Relevance**: this measures whether the response actually answers the user's question in a useful and relevant way. The calculation is the average of the cosine similarity scores:

$$AR = \frac{1}{n}\sum_{n=1}^{N} sim(c, c_i) \quad (4)$$

For each E_{gi} embedding question generated based on the answer to the original question E_O.

It is possible to manually count the number of claims to calculate this metric, but in this work a LLM was used to facilitate automatic and reproducible testing.

3 Related Work

The exploratory research deepened the understanding of LLM models and their applications in natural language processing, particularly in problem-solving scenarios related to research, specialization, and disambiguation. This was achieved through the ingestion of large volumes of data in various formats extracted from PDF files. In this context, Retrieval-Augmented Generation (RAG) emerged as a key technique to address the limitations of traditional generative AI models by incorporating external knowledge sources into the generation process.To enhance traditional RAG methods and develop specialized Knowledge Bases (KBs) across different domains, various RAG architectures were explored. Among the retrieval strategies relevant to the complementary exploratory research, notable approaches include VectorRAG, GraphRAG, MemoRAG, HybridRAG, and HippoRAG Zhao *et al.* [2024]; Wang *et al.* [2024]; Phan *et al.* [2024]. These methodologies are presented in this section.

3.1 VectorRAG

In VectorRAG Sarmah *et al.* [2023], documents are divided into chunks, converted into embeddings, and stored in a vector database, as illustrated in Figure 3. When a user submits a query, the system generates an embedding for it and searches the vector database for semantically similar chunks, which are then used as context for the LLM along with the original query. The language model then generates a response by combining the retrieved information with its prior knowledge. This approach ensures that responses are detailed and contextually accurate by leveraging external information in real time.

The main advantages of VectorRAG are its simplicity and efficiency in handling large datasets. However, it may struggle with tasks requiring complex reasoning or analyzing relationships between different pieces of information, especially when dealing with implicit or ambiguous queries Sarmah *et al.* [2023]; Peng *et al.* [2024].

3.2 GraphRAG

GraphRAG Edge *et al.* [2024] leverages Knowledge Graphs (KG) to enhance LLMs by retrieving relevant structured information for a given query. A Knowledge Graph (KG) is a structure that organizes real-world entities, their attributes, and relationships, stored in graph databases. Each fundamental unit (triple) consists of a subject, predicate, and object.

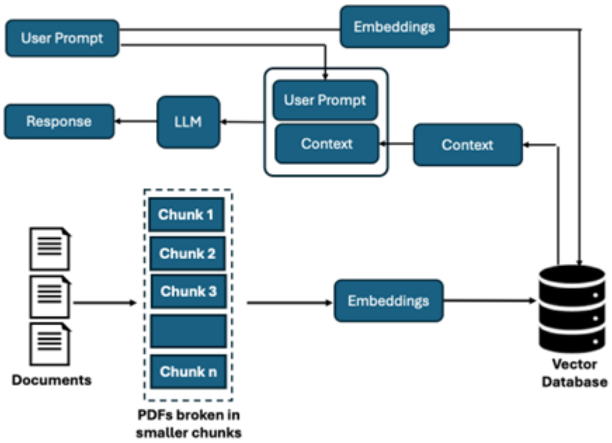


Figure 3. VectorRAG architecture for document ingestion and query resolution. Demonstrates how document chunks are embedded and matched semantically with user queries, enabling retrieval of relevant content from vector databases Sarmah *et al.* [2023]; Peng *et al.* [2024].

.

Using this methodology, a KG is constructed from the data, where nodes represent entities and edges represent relationships between them. When a user query is presented, GraphRAG searches the KG for relevant subgraphs and uses them to generate the response. For temporal queries, temporal information can be incorporated into the KG by adding temporal nodes and edges—for example, an edge can indicate that an event occurred during a specific period.

The main benefit of GraphRAG is its ability to handle complex relationships and reasoning tasks, especially in queries requiring multihop reasoning. However, constructing and maintaining large KGs can be challenging and time-consuming. Additionally, performance may suffer when dealing with ambiguous queries or tasks that do not explicitly mention entities present in the KG Edge *et al.* [2024]; Peng *et al.* [2024].

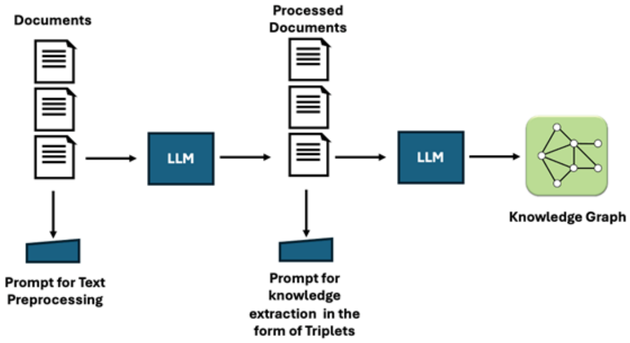

Documents
Processed Documents
LLM
LLM
Knowledge Graph
Prompt for Text Preprocessing
Prompt for knowledge extraction in the form of Triplets

Figure 4. GraphRAG structure leveraging knowledge graphs to support reasoning. Depicts how entity relationships are encoded and used for advanced semantic queries, particularly beneficial in complex tasks Edge *et al.* [2024]; Peng *et al.* [2024].

3.3 HybridRAG

HybridRAG Sarmah *et al.* [2024] combines the strengths of VectorRAG and GraphRAG to improve the quality of LLM responses when retrieving information from vector databases and KGs, providing a more comprehensive and accurate context for generating responses. To support temporal queries, temporal metadata can be embedded in the vector database and KG, allowing HybridRAG to filter and retrieve information relevant to a specific time period (see Figure 5).

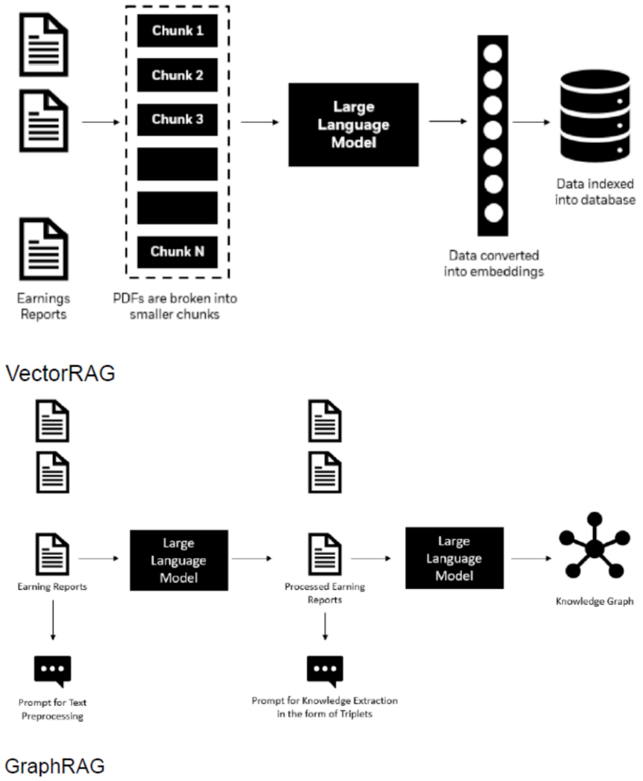

Chunk 1
Chunk 2
Chunk 3
Chunk N
Large Language Model
Data indexed into database
Data converted into embeddings
Earnings Reports
PDFs are broken into smaller chunks
VectorRAG
Earning Reports
Large Language Model
Processed Earning Reports
Large Language Model
Knowledge Graph
Prompt for Text Preprocessing
Prompt for Knowledge Extraction in the form of Triplets
GraphRAG

Figure 5. HybridRAG strategy combining Vector and Graph approaches. Captures the advantage of dual-retrieval methods, balancing semantic similarity and relational context to enhance response accuracy in complex questions Sarmah *et al*. [2024].

This method aims to benefit from the strengths of VectorRAG and GraphRAG, providing a more balanced approach to information retrieval. However, it may suffer from limitations of both techniques, such as the complexity of constructing and maintaining KGs and the difficulties in dealing with ambiguous or implicit queries Sarmah *et al*. [2024].

3.4 MemoRAG

Another relevant approach in the literature is MemoRAG Qian *et al*. [2024], a variation of RAG that adds short-term memory to maintain context between interactions. This allows the model to remember relevant information within a conversation, avoiding repetition and making responses more coherent Qian *et al*. [2024].

MemoRAG works like a regular RAG, retrieving information from an external database (vector database, documents or structured knowledge). Unlike a traditional RAG, MemoRAG saves the context of the conversation for future reference Qian *et al*. [2024]. Thus, if the user continues the conversation with a question connected to the previous topic, MemoRAG uses its short-term memory to contextualize the response. The model combines the retrieved information with its memory and generates a more coherent response. After responding, MemoRAG updates its memory with the new interaction. This maintains the context for future questions without having to process everything again (Figure 6).

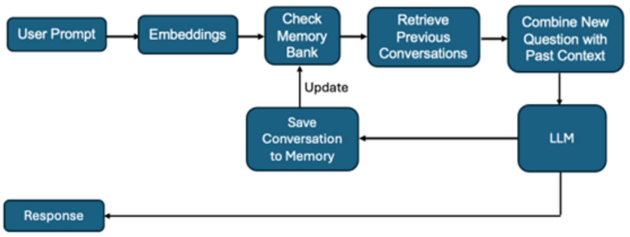

User Prompt
Embeddings
Check Memory Bank
Retrieve Previous Conversations
Combine New Question with Past Context
Update
Save Conversation to Memory
LLM
Response

Figure 6. MemoRAG design showcasing short-term memory integration in retrieval. Enables continuity across turns in multi-turn conversations by maintaining conversational history for contextualized responses Qian *et al*. [2024].

The main benefits of MemoRAG include its ability to handle complex tasks and long contexts, especially when dealing with implicit or ambiguous queries. MemoRAG can use one or two language models (LLMs), depending on the specific architecture Qian *et al*. [2024]:

For single LLM:

- A single LLM manages both information retrieval and memory;
- It receives the context stored in memory and generates contextualized responses.
- Single-LLM is simpler and more efficient for short conversations;
- Examples: GPT-4 and LLaMA.

for Dual-LLM (Two Language Models):

- Main LLM → Responsible for generating responses;
- o Auxiliary LLM (or external memory) → Manages the storage and retrieval of the conversation context;
- The second LLM can be a smaller model, specialized in organizing memory and optimizing searches;
- Dual-LLM improves accuracy and scalability in more advanced systems;
- Requires the training of two LLMs (memory module and response generation module), which can be computationally expensive Qian *et al*. [2024].

3.5 HippoRAG

HippoRAG (Hierarchical Persistent Personalized Online RAG) Gutiérrez *et al*. [2024] is an evolution of MemoRAG that incorporates persistent and hierarchical memory to maintain the context of conversations over the long term. Unlike MemoRAG, which focuses on short-term memory, HippoRAG can remember information for days, weeks, or even months, organizing it efficiently Yang [2024]. HippoRAG takes an approach inspired by the hippocampal indexing theory of human memory (Figure 7).

HippoRAG saves the interaction in a hierarchically organized memory, that is, information is stored at different levels of importance and relevance. This allows the model to:

- Remember relevant information over time;
- Automatically forget less important details;
- Retrieve responses more efficiently.

To support temporal queries, temporal information can be incorporated into the KG triples, such as including timestamps or time intervals associated with entities and relationships, and thus allowing the analysis of information from a specific period, for example. The main benefits of HippoRAG include its ability to continuously integrate new information, perform multi-hop computation in a single step, and achieve high efficiency in information retrieval. However, it requires significant initial processing for the construction of the KG, as well as being sensitive to the accuracy of the process used to extract triples from the data and having potential difficulties in scaling the database Gutiérrez *et al.* [2024].

3.6 Alternatives to RAG

In addition to **Retrieval-Augmented Generation (RAG)**, more recent approaches have emerged, such as **Cache-Augmented Generation (CAG)** Chan *et al.* [2025] and **Knowledge-Augmented Generation (KAG)** Liang *et al.* [2025].

Cache-Augmented Generation (CAG) enhances the generation process by preloading all relevant documents into the context of a large language model (LLM) using a precomputed key-value (KV) cache. This approach reduces latency and computational overhead by eliminating the need for real-time retrieval, allowing the model to respond more quickly and consistently. CAG simplifies the system architecture by transitioning from a query-retrieve-generate pipeline to a more efficient query-generate workflow. It is particularly effective in well-defined domains, ensuring reliable and repeatable outputs. However, CAG is constrained by the LLM's context window, limiting its scalability for large or rapidly changing knowledge bases. Additional challenges include static knowledge representation, higher upfront costs for preprocessing and storage, reduced flexibility in handling unexpected queries, and potential security concerns due to persistent in-memory data Chan *et al.* [2025].

Knowledge-Augmented Generation (KAG) is a framework that integrates the retrieval mechanisms of Retrieval-Augmented Generation (RAG) with the structured logic provided by knowledge graphs (KGs). While RAG extends large language models (LLMs) by incorporating external knowledge, it faces several limitations: (a) Its similarity-based retrieval process often struggles to capture logical relationships such as temporal dependencies, numerical operations, or causal links; (b) It can also lead to redundant, repetitive, or irrelevant outputs, making it difficult to extract meaningful insights for downstream tasks.

KAG addresses these limitations by incorporating structured reasoning capabilities through the semantic relationships encoded in knowledge graphs, enabling deeper, more contextualized analysis and improved relevance in generation tasks. However, its effectiveness is limited by the coverage and quality of the underlying knowledge graph, and scaling such graphs to broad or dynamic domains poses significant technical challenges Liang *et al.* [2025].

In addition to CAG and KAG, other alternatives to RAG include **Fine-Tuning** Soudani *et al.* [2024] and **In-Context Learning (ICL)** Guo *et al.* [2025].

Fine-Tuning involves training a pre-trained LLM on domain-specific data for a particular task. For example, a model can be fine-tuned using financial documents to enhance its performance in finance-related tasks. This process enables the model to transfer general knowledge acquired during pre-training to the specific domain. Fine-tuning is particularly effective when the target domain is well-defined and relatively stable. However, while it is more resource-efficient than training from scratch, it still demands significant computational power and data volume, and is susceptible to overfitting if not properly managed Soudani *et al.* [2024].

In-Context Learning (ICL), by contrast, involves providing relevant information directly within the prompt. This approach is straightforward to implement and avoids the complexity of external retrieval systems. ICL allows models to adapt in real time using examples embedded in the prompt, without modifying the model's weights. While fast and flexible, it is constrained by the context window size and heavily dependent on the quality of the prompt Guo *et al.* [2025].

Despite the availability of these alternative methods, the present work focused exclusively on RAG. The goal was not to compare all existing strategies but to develop a system based on the most readily available approach at the time—namely, RAG.

3.7 Comparison of RAG Approaches

With the research carried out, a set of possible approaches was consolidated, along with their main benefits and limitations. Table 1 presents the comparative analysis of the relevant approaches considered for the development of the search strategy for the Proof of Concept (PoC).

By analyzing the RAG approaches for the development of the PoC, the VectorRAG, GraphRAG and MemoRAG techniques were prioritized in order to support the search problem-situation.

4 Materials and Methods

The current work uses a modular conversational agent framework for specific domains Pinna *et al.* [2024], which is an advancement in the common architecture of task-oriented conversational agents Ultes *et al.* [2017]. In this work, we complement this framework by using the latest generative AI techniques and Large Language Models (LLMs). A diagram of the employed architecture is shown in Figure 8, where the modules of the conversational system from previous works are highlighted in yellow, and the points where language models are applied in the current work are shown.

To enhance system interaction and ensure greater accuracy in the responses provided, we employ LLMs for input and output message filtering. These models are used to identify and mitigate hallucinations, helping that generated responses remain aligned with the system's validated knowledge. Furthermore, we apply domain filters to ensure interactions remain within specified topics, reducing noise and irrelevant information. We also incorporate toxicity detection mechanisms, aiming to prevent inappropriate or offensive responses and promoting a safe and responsible interaction environment. If a question is blocked by one of the filters, the system does not provide a response, as the query is considered outside the agent's scope.

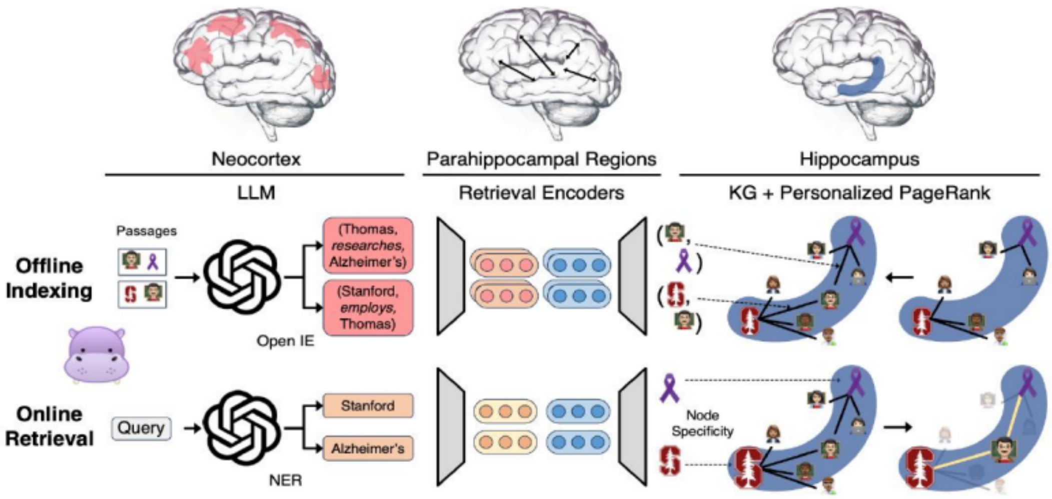


Figure 7. HippoRAG architecture incorporating hierarchical long-term memory. This approach supports persistent memory, allowing retrieval of past information Gutiérrez *et al*. [2024].

Search Approach	Related Work	Benefits	Limitations
VectorRAG	Sarmah *et al*. [2023]	Captures relationships between documents; useful for domains with interconnections	High computational complexity; graph construction and maintenance.
GraphRAG	Edge *et al*. [2024]	High precision in semantic queries;easy implementation with pre-trained models.	Limited by fixed dimensionality; does not capture complex relationships well.
MemoRAG	Qian *et al*. [2024]	Reuses knowledge from previous queries; better for global queries or implicit information	Difficult to adapt memory module to different domains; requires efficient memory management
HybridRAG	Sarmah *et al*. [2024]	Combines vectors and graphs; flexible for different query types	Complex to implement; more computational resources required
HippoRAG	Gutiérrez *et al*. [2024]	Easy to update long-term memory; possibility of reasoning through multiple graphs edges	Low efficiency when scaling a database; difficult to optimize for different domains

Table 1. Comparison of RAG Approaches

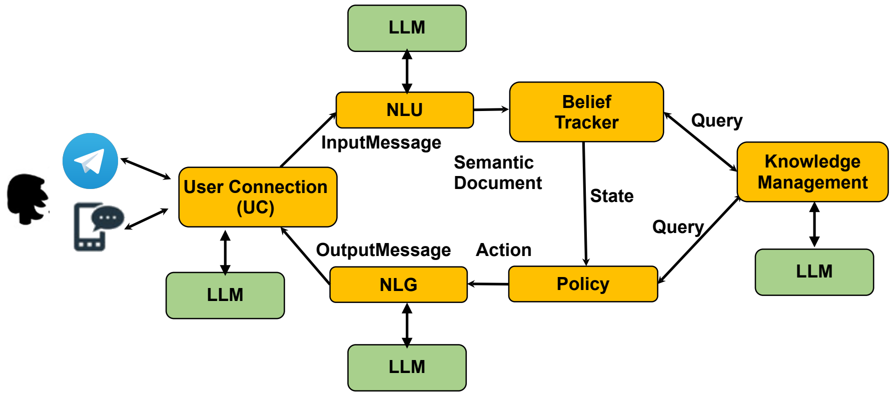


Figure 8. Architecture of the modular dialog system, indicating where LLMs are applied in input/output processing, knowledge management, and generation. Highlights flexibility to switch LLMs and support domain-specific interactions.

Another major architectural modification was replacing traditional Natural Language Understanding (NLU) modules with LLMs. This approach enables the system to interpret complex queries more effectively, identifying intents and entities with high precision. As a result, the process of interpreting user inputs becomes more flexible and adaptable to different types of questions, enhancing the interactive experience and reducing ambiguities in meaning extraction. Moreover, with the adoption of LLMs, it is no longer necessary to train traditional AI classification and labeling models, which previously required domain-specific annotated datasets. This eliminates a friction point for adding new domains to the agent, facilitating its reuse.

Regarding response generation (NLG), LLMs were applied to create personalized and contextually coherent responses. These models enable the elaboration of more natural responses, adjusting to the user's needs and the interaction context. Additionally, we allow the use of different LLM models as needed.

The LLM used in the project can be replaced at any point, allowing flexibility in model selection according to the specific requirements of the application. We tested this approach with different models, including GPT-4o and GPT-4o-mini accessed via API OpenAI [2025], and open-source models such as Llama 3.2 Meta [2025], which can be run locally. Moreover, other alternatives can be integrated as needed, ensuring adaptability throughout development. Considering that the objective of evaluating RAG approaches, ChatGPT-4o was also used as the evaluator model when using RAGAS framework library (associated metrics presented in subsection 2.2). Possible evaluations with other models are subject to future investigations and are considered out of scope of the present work.

Advanced data ingestion is an essential component of the system, ensuring that the utilized information is well-structured and organized. To this end, we implemented mechanisms for extracting data from PDF documents, chunk fragmentation, metadata extraction, and chunk clustering, enabling efficient data indexing. Additionally, we use vectorstore generation for vector-based information storage and graphstores for organizing semantic relationships between data, allowing efficient queries through the GraphRAG technique.

We utilized a PDF document loader capable of extracting textual information directly from PDFs, as well as parsing tables and employing Optical Character Recognition (OCR) to retrieve text embedded in images within the documents Team [2024]. Subsequently, we applied a semantic chunking technique to partition the document content into coherent chunks, ensuring sentences with similar semantic content remained grouped together, while separating sentences expressing distinct meanings into different chunks. These chunks were then clustered using hierarchical agglomerative clustering based on their embeddings, a method chosen to optimize representation efficiency as the number of processed documents increased.

Throughout this process, each chunk was annotated with metadata, which was subsequently stored alongside the chunk vectors in the vector store. This metadata serves multiple purposes: it can filter search query results and assist the Large Language Model (LLM) in generating responses by providing accurate source references, such as the original document and exact page number where specific facts were mentioned.

The following metadata attributes were collected during our experiments:

- **Filename**: Name of the original source file.
- **Page**: Page number within the original file.
- **Paragraph**: Paragraph number indicating the chunk's starting point.
- **Published Date**: Publication date of the source document.
- **Keywords**: List of relevant keywords within the chunk.
- **Title**: Title of the original document.
- **Section Titles**: Hierarchical list of the section headings associated with the chunk.
- **Authors**: Names of the document authors.
- **Source URL**: URL linking to the original document source.
- **Context**: Surrounding textual context for each chunk.

In the ingestion interface and developed APIs, users can choose to create the vector store, the graph store, or both. The indexing processes are executed in parallel, and their results are stored independently, allowing separate queries if necessary. Each result is registered in the triplestore and associated with the corresponding domain, so they can be queried later by the Knowledge Manager (KM). A diagram of this process can be seen in Figure 9.

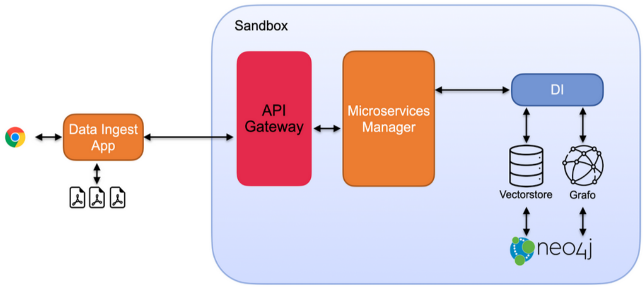


Figure 9. Diagram of the data ingestion pipeline. It shows how documents are processed into vector and graph stores, with metadata extraction supporting accurate retrieval, source tracing, and hybrid querying.

The Retrieval-Augmented Generation (RAG) mechanism plays a crucial role in obtaining accurate responses to complex questions related to specific domains. We use this approach to seek responses based on ingested documents, ensuring the returned information is well-founded and relevant. The system allows flexible searches, consulting exclusively the vectorstore, the graphstore, or both, depending on the type of query performed. We also implemented strategies to handle dependent and compound queries, ensuring even complex questions can be answered with high accuracy and contextual coherence.

To minimize GraphRAG usage and make searches efficient and effective, we adopted a selective query strategy for graph stores, as it demands higher computational costs. "Simple" questions, which can be answered with specific text excerpts from the vector stores, will be handled solely by VectorRAG, ensuring faster and more efficient responses. In contrast, "Complex" questions—those requiring information gathering from multiple excerpts or exploring relationships between various entities (e.g., follow-up questions, specific timeframes) —will be answered in a hybrid manner, combining vector store and graph store queries.

In this hybrid approach, the system independently queries both stores, and the results are integrated into a single context, provided to the LLM to generate the final response. Thus, additional computational resources are used only when necessary for more complex questions.

The NLU service performs the question complexity classification, identifying whether the query should be made only in VectorRAG or in both stores. This way, the system dynamically adapts to the question's complexity level and data availability for the relevant domain. Figure 10 illustrates this process.

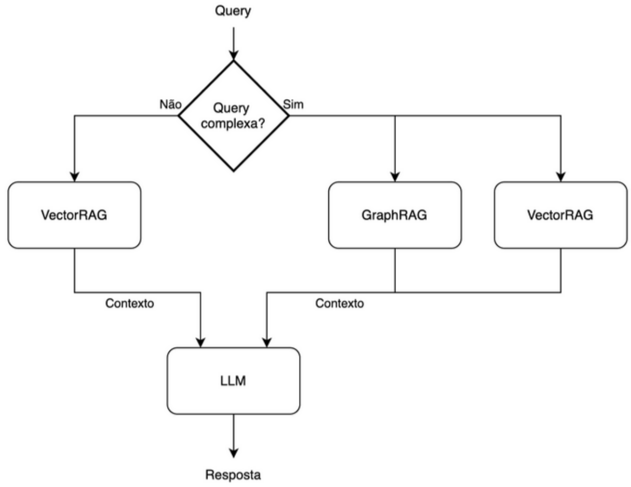


Figure 10. System's decision logic for determining question complexity. Enables dynamic selection of retrieval methods—simple queries use VectorRAG, while complex ones combine Vector and Graph retrievals.

The criteria adopted for selecting the datasets involved domain relevance, public availability of data and complexity of typical queries. However, it is important to recognize possible biases arising from the selection of limited sources, which may influence the generalization of the results obtained.

5 Case Studies

This section presents both Banking case studies used to evaluate the advanced RAG approaches to support complex user queries.

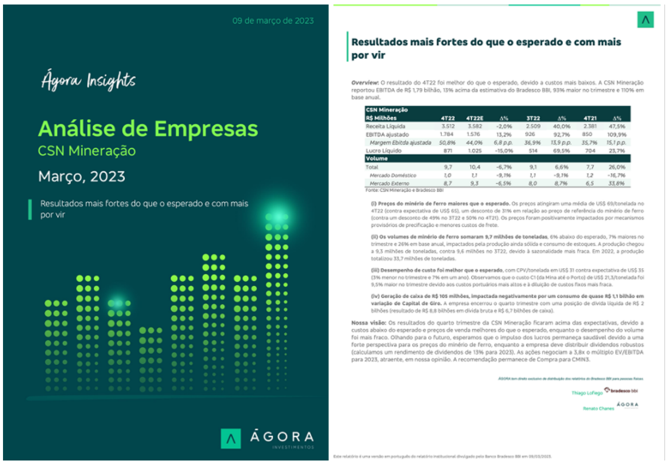
5.1 Ágora Broker

Ágora is an investment brokerage firm owned by Bradesco. The company regularly publishes financial reports and market insights authored by its team of analysts. In this case study, we evaluate the performance of the proposed RAG system using a dataset composed of client queries and financial analysis documents provided by Ágora.

The dataset includes 100 chat messages based on 56 financial documents in PDF format. These messages were manually created by domain specialists at the bank, simulating realistic information needs that might be posed to a conversational agent. Each message references a snippet from the source document that contains the relevant answer, facilitating evaluation. Table 2 presents representative samples from the dataset. Multi-turn interactions are grouped by a shared *Chat ID*.

For each user query, a corresponding reference snippet was selected from the associated source document. These snippets are used as ground truth during recall-based evaluations of the system. Additionally, metadata such as the source file name and its download link are stored for each interaction. These documents were preprocessed by the system and indexed to form the domain-specific retrieval database.

Figure 11 presents an example of a financial insights report published by Ágora. These reports are publicly available on the company's website.


09 de março de 2023

Ágora Insights

Análise de Empresas

CSN Mineração

Março, 2023

Resultados mais fortes do que o esperado e com mais por vir

ÁGORA INVESTIMENTOS

Resultados mais fortes do que o esperado e com mais por vir

Overview: O resultado do 4T22 foi melhor do que o esperado, devido a custos mais baixos. A CSN Mineração reportou EBITDA de R$ 1,79 bilhão, 13% acima da estimativa do Bradesco BBI, 93% maior no trimestre e 110% em base anual.

CSN Mineração **R$ Milhões**	**4T22**	**4T22E**	**Δ%**	**3T22**	**Δ%**	**4T21**	**Δ%**
Receita Líquida	3.512	3.582	-2,0%	2.509	40,0%	2.381	47,5%
EBITDA ajustado	1.784	1.576	13,2%	926	92,7%	850	109,9%
Margem Ebitda ajustada	*50,8%*	*44,0%*	*6,8 p.p.*	*36,9%*	*13,9 p.p.*	*35,7%*	*15,1 p.p.*
Lucro Líquido	871	1.025	-15,0%	514	69,5%	704	23,7%
Volume							
Total	9,7	10,4	-6,7%	9,1	6,6%	7,7	26,0%
Mercado Doméstico	*1,0*	*1,1*	*-9,1%*	*1,1*	*-9,1%*	*1,2*	*-16,7%*
Mercado Externo	*8,7*	*9,3*	*-6,5%*	*8,0*	*8,7%*	*6,5*	*33,8%*

Fonte: CSN Mineração e Bradesco BBI

(i) Preços do minério de ferro maiores que o esperado. Os preços atingiram uma média de US$ 69/tonelada no 4T22 (contra expectativa de US$ 65), um desconto de 31% em relação ao preço de referência do minério de ferro (contra um desconto de 49% no 3T22 e 50% no 4T21). Os preços foram positivamente impactados por mecanismos provisórios de precificação e menores custos de frete.

(ii) Os volumes de minério de ferro somaram 9,7 milhões de toneladas, 6% abaixo do esperado, 7% maiores no trimestre e 26% em base anual, impactados pela produção ainda sólida e consumo de estoques. A produção chegou a 9,3 milhões de toneladas, contra 9,6 milhões no 3T22, devido à sazonalidade mais fraca. Em 2022, a produção totalizou 33,7 milhões de toneladas.

(iii) Desempenho de custo foi melhor que o esperado, com CPV/tonelada em US$ 31 contra expectativa de US$ 35 (3% menor no trimestre e 7% em um ano). Observamos que o custo C1 (da Mina até o Porto) de US$ 21,3/tonelada foi 9,5% maior no trimestre devido aos custos portuários mais altos e à diluição de custos fixos mais fraca.

(iv) Geração de caixa de R$ 105 milhões, impactada negativamente por um consumo de quase R$ 1,1 bilhão em variação de Capital de Giro. A empresa encerrou o quarto trimestre com uma posição de dívida líquida de R$ 2 bilhões (resultado de R$ 8,8 bilhões em dívida bruta e R$ 6,7 bilhões de caixa).

Nossa visão: Os resultados do quarto trimestre da CSN Mineração ficaram acima das expectativas, devido a custos abaixo do esperado e preços de venda melhores do que o esperado, enquanto o desempenho do volume foi mais fraco. Olhando para o futuro, esperamos que o impulso dos lucros permaneça saudável devido a uma forte perspectiva para os preços do minério de ferro, enquanto a empresa deve distribuir dividendos robustos (calculamos um rendimento de dividendos de 13% para 2023). As ações negociam a 3,8x o múltiplo EV/EBITDA para 2023, atraente, em nossa opinião. A recomendação permanece de Compra para CMIN3.

ÁGORA tem direito exclusivo de distribuição dos relatórios do Bradesco BBI para pessoas físicas.

Thiago Lofiego bradesco bbi

Renato Chanes ÁGORA

Este relatório é uma versão em português do relatório institucional divulgado pelo Banco Bradesco BBI em 09/03/2023.

Figure 11. Sample report from Ágora Brokerage in Brazilian Portuguese used in the case study. Highlights typical document structure and financial language used as input for conversational system testing.

The evaluation questions are categorized into groups based on distinct problem situations that challenge conventional dialogue systems. Table 3 presents the frequency distribution of these situations in the evaluation dataset.

After ingestion, the processed documents become available to various components of the architecture. Figure 12 illustrates the data querying mechanism within the system, which enables response generation through document retrieval and answer synthesis.

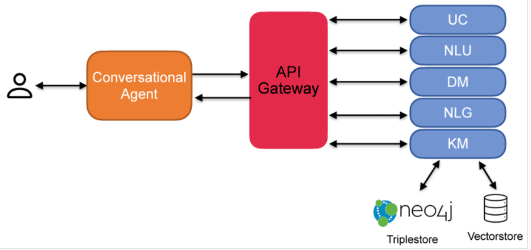


Figure 12. Internal query mechanism of the dialog system. Depicts how user input is matched to retrieved content and processed by LLMs for contextualized answer generation.

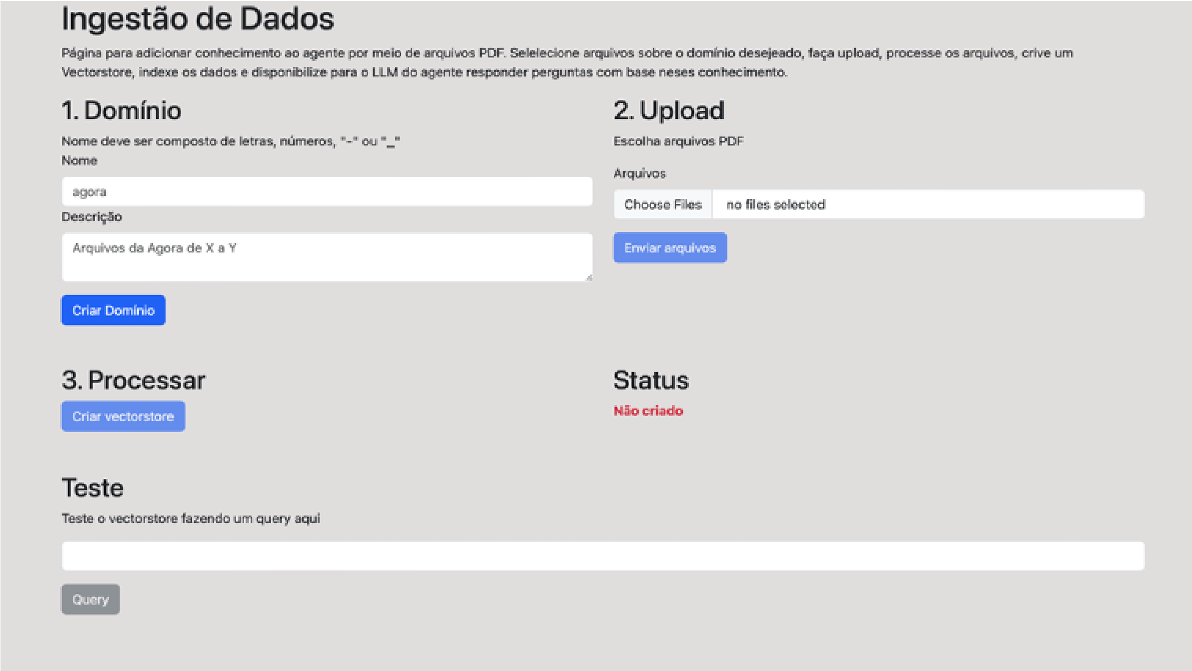
The ingestion process is facilitated through a simple user interface, as demonstrated in Figure 13. The developed interface allows a business user to create a new domain, upload its documents, trigger the data ingestion pipeline, and test the retrieval from the resulting databases. This domain is then ready to be used by the dialog system.

After data ingestion, the conversational agent utilizes a Large Language Model (LLM) that queries information from the vector store to answer specific questions within the context, as depicted in Figure 14.

We focused our evaluation on the VectorRAG and GraphRAG search approaches, as they represent widely adopted and well-established techniques in the RAG ecosystem. Other methods were excluded from this study due to

Chat ID	Message	Reference text	Source
0	Dado o investimento de R$ 2,0 bilhões em 2023 e aproximadamente R$ 3,1 bilhões/ano entre 2024-26, qual foi a ação da CTEEP?	Com previsão de investimento de R$ 2,0 bilhões em 2023 e aproximadamente R$ 3,1 bilhões/ano entre 2024-26, a CTEEP provavelmente terá que manter seu pagamento de	10013899.pdf
0	E qual foi o seu novo preço alvo?	Com previsão de investimento de R$ 2,0 bilhões em 2023 e aproximadamente R$ 3,1 bilhões/ano entre 2024-26, a CTEEP provavelmente terá que manter seu pagamento de	10013899.pdf
1	Qual foi o valor pago pela Multiplan para adquirir 49,9% do DiamondMall?	A Multiplan divulgou ontem (02) à noite um fato relevante informando que exerceu o direito de preferência na aquisição de 49,9% do DiamondMall, em Belo Horizonte (MG), por R$ 340 milhões, aser pago da seguinte forma: (i) R$ 136 milhões na assinatura do contrato; e (ii) os R$ 204 milhões restantes em 12 parcelas mensais, indexadas ao IPCA.	10372155.pdf

Table 2. Ágora domain dataset sample


Ingestão de Dados

Página para adicionar conhecimento ao agente por meio de arquivos PDF. Selelecione arquivos sobre o domínio desejeado, faça upload, processe os arquivos, crive um Vectorstore, indexe os dados e disponibilize para o LLM do agente responder perguntas com base neses conhecimento.

1. Domínio

Nome deve ser composto de letras, números, "-" ou "_"

Nome: agora

Descrição: Arquivos da Agora de X a Y

2. Upload

Escolha arquivos PDF

Arquivos: no files selected

3. Processar

Status

Não criado

Teste

Teste o vectorstore fazendo um query aqui

Figure 13. Web interface in Brazilian Portuguese for users to upload documents and initiate ingestion. Facilitates domain creation and allows manual validation of search and retrieval operations with specific domain selection, document upload, processing and tests in natural language.

Situation	Count
Dependent question	31
Direct question	22
Acronym interpretation	13
Temporal reference understanding	10
Questions with multiple sentences	9
Ability to compare and differentiate	7
Context switch	4
Long-term context	2
Dependent question with multiple sentences	2

Table 3. Frequency of problem situations in the Ágora domain evaluation dataset

implementation complexity and inefficiency within the constraints of this domain. Additionally, we explored the combination of both techniques by aggregating their results into a single context window for final response generation by the LLM. This combination was tested in two configurations: (i) querying both the vector store and the graph store for every input (VectorRAG + GraphRAG), and (ii) querying the graph store only when a question is classified as complex, as described in Section 4 (VectorRAG + GraphRAG complexity).

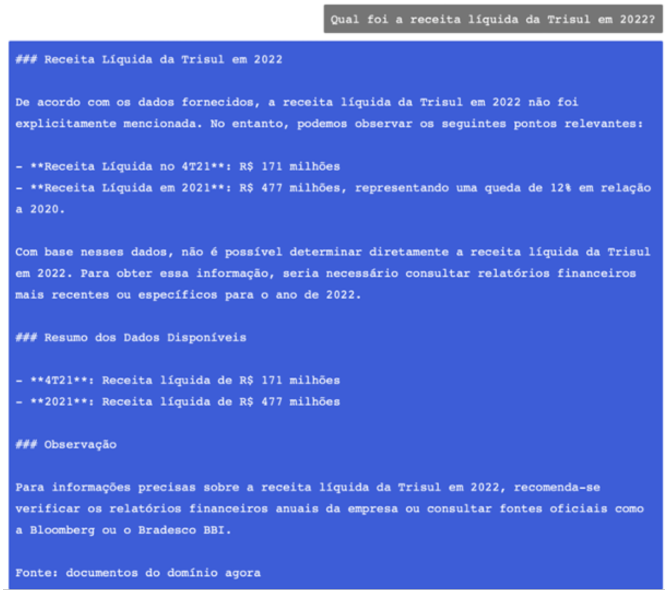
The evaluation results for the Ágora dataset are shown in Table 4. We report the metrics of faithfulness, context precision (CP), and relevance, each on a scale from 0 to 1, where higher values indicate better performance. The CP metric is reported both with and without explicit reference context. All evaluation metrics were computed using GPT-4o as the evaluator model.

To account for the cost of evaluation, we report token usage in both the input prompts ("Eval in tokens") and the generated outputs ("Eval out tokens") processed by the LLM evaluator. We also include generation time and evaluation time, measured as the total time required to process the entire dataset.

The results indicate that VectorRAG outperformed

Search Approach	Faithfulness	CP w/o reference	CP w/ reference	Relevance	Eval in tokens	Eval out tokens	Gen time	Eval time
VectorRAG	0.66	0.50	0.24	0.56	3.79M	176k	697s	328s
GraphRAG	0.62	0.51	0.34	0.25	771k	73k	599s	309s
VectorRAG + GraphRAG	0.66	0.46	0.22	0.49	4.07M	197k	789s	530s
VectorRAG + GraphRAG complexity	0.64	0.47	0.24	0.53	3.88M	183k	611s	533s

Table 4. Evaluation result for search approaches in the Ágora Broker domain



```
Qual foi a receita líquida da Trisul em 2022?

### Receita Líquida da Trisul em 2022

De acordo com os dados fornecidos, a receita líquida da Trisul em 2022 não foi
explicitamente mencionada. No entanto, podemos observar os seguintes pontos relevantes:

- **Receita Líquida no 4T21**: R$ 171 milhões
- **Receita Líquida em 2021**: R$ 477 milhões, representando uma queda de 12% em relação
a 2020.

Com base nesses dados, não é possível determinar diretamente a receita líquida da Trisul
em 2022. Para obter essa informação, seria necessário consultar relatórios financeiros
mais recentes ou específicos para o ano de 2022.

### Resumo dos Dados Disponíveis

- **4T21**: Receita líquida de R$ 171 milhões
- **2021**: Receita líquida de R$ 477 milhões

### Observação

Para informações precisas sobre a receita líquida da Trisul em 2022, recomenda-se
verificar os relatórios financeiros anuais da empresa ou consultar fontes oficiais como
a Bloomberg ou o Bradesco BBI.

Fonte: documentos do domínio agora
```

Figure 14. Conversation example illustrating VectorRAG usage. Demonstrates how semantically aligned responses are generated from vector-based retrieval in response to financial queries.

GraphRAG in this domain, achieving higher scores in both faithfulness and relevance. Combining both methods (VectorRAG + GraphRAG) resulted in similar performance to VectorRAG alone but introduced a increase in token usage and computational overhead. The hybrid strategy that selectively queries the graph store for complex questions achieved a favorable balance: maintaining comparable performance while reducing resource usage compared to always querying both sources.

First, VectorRAG achieved the highest scores in both faithfulness and relevance. This suggests that the semantic embeddings were well-aligned with the language and structure of the documents, which typically feature consistent terminology and well-formed financial narratives.

In contrast, GraphRAG demonstrated lower performance across most metrics—most notably in relevance, which dropped considerably compared to the other approaches. This underperformance can likely be attributed to several domain-specific factors: (i) sparse or non-standard graph structure, the reports are structured primarily as prose documents with limited use of explicit entities or relational patterns, (ii) lack of fine-grained entities, key concepts may not be captured as discrete nodes in the knowledge graph, and (iii) financial queries from the dataset mostly requires interpretation of specific sentences or paragraphs.

The combined approach (VectorRAG + GraphRAG) performed similarly to VectorRAG in terms of accuracy, but incurred a higher computational cost, as shown by increased token usage and evaluation time. Interestingly, the complexity-aware hybrid strategy (VectorRAG + GraphRAG complexity) yielded a balanced result: it maintained performance close to VectorRAG while reducing overall computational cost by selectively invoking the graph-based retrieval only when the query was classified as complex. This approach effectively leverages the strengths of both methods while mitigating unnecessary overhead.

The final architecture of the Ágora Case is presented in Figure 15, where the newly developed components are integrated with the pre-existing architecture.

This case study highlights the importance of integrating conversational agents with financial data sources, ensuring accurate and context-aware responses. The use of LLMs in conjunction with structured data ingestion and retrieval mechanisms provides a scalable solution for automating financial insights delivery. RAGAS metrics allowed the comparison of the different approaches, but the overall low values indicate that using other techniques such as In-Context Learning and Finetuning may be required before deployment of the proposed system with real users.

5.2 Investment Recommendation Letters

The Investment Recommendation Letters dataset is made up of reports released by hedge funds from Brazil. These documents contain financial market analyses, investment strategies, and the performance of the funds over time. The letters are mostly available in PDF format, published by different managers, and exhibit varying structures and formatting standards.

Table 5 below presents the conversation database built for this domain, following a structure similar to that used in the Ágora domain, previously described. This set of questions was designed to evaluate the system's accuracy and robustness in retrieving information within the context of Investment Recommendation Letters. This dataset includes 69 chat messages based on 40 investment letters. These messages were created the same manner as the previous domain shown. As letters are published by multiple investment groups, each one can have particular availability, mostly the dataset are composed of online letters to investors from groups like BNP Paribas, ARX Investimentos, V8 Capital, Legacy Capital, etc.

The categorization for this evaluation dataset is presented

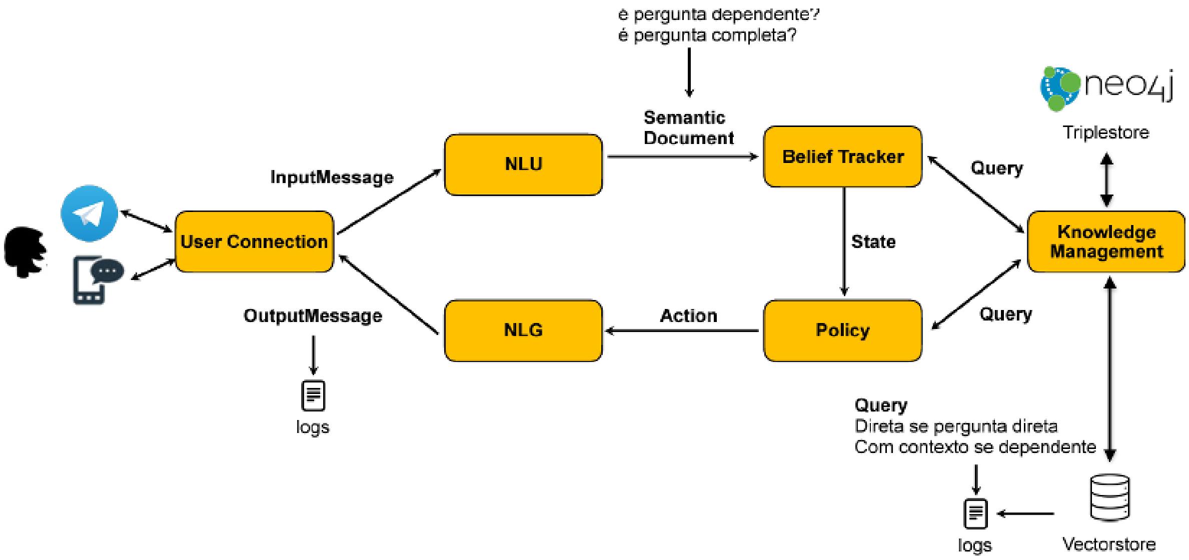


Figure 15. Final integrated architecture used in the Ágora case study. Shows how conversational agents interface with LLMs and retrieval modules to support financial insights queries.

ChatID	Message	Text of Reference	Source
2	Quais foram os ganhos da Ace Capital segundo a carta de 06/23?	O Ace Capital FIC FIM registrou ganho de 1,10% em julho; ganho de 9,38% no acumulado do ano (123% do CDI ou CDI+2,79% a.a.); ganho de 17,33% nos últimos 12 meses (128% do CDI ou CDI+3,34 a.a.); e acumula retorno de 50,68% desde seu início em 30/09/2019 (162% do CDI ou CDI+3,66% a.a.).	Carta-Julho-2023.pdf
5	Como foi o desempenho da economia dos EUA durante 05/23?	Os sinais de resiliência da atividade econômica nos Estados Unidos continuaram evidentes ao longo do mês de maio, com os indicadores de consumo e de mercado de trabalho seguindo como os principais destaques positivos. Além disso, a inflação continua se mostrando mais resistente do que o desejado pelo Fed, sem ainda dar sinais claros de que está arrefecendo.	Carta-Maio-2023.pdf
5	e na Europa?	Na Europa, a rápida recuperação dos indicadores de confiança industrial do início do ano apresentou uma pausa em maio, com a Alemanha sendo o principal destaque negativo do mês	Carta-Maio-2023.pdf

Table 5. Test Cases Dataset

in Table 6, showing the frequency of each problem situation.

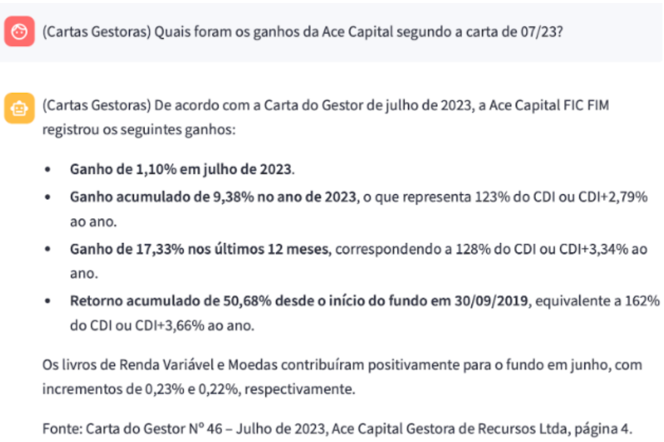
Situation	Count
Dependent question	17
Questions with multiple sentences	17
Direct question	13
Temporal reference understanding	12
Dependent question with multiple sentences	10

Table 6. Frequency of problem situations in the investment recommendation letters domain evaluation dataset

Data ingestion in this case followed a similar approach to that adopted in the Ágora domain, processing the available documents and storing the extracted information in appropriate structures for efficient retrieval. However, an additional challenge was identified during this stage: some of the analyzed documents contained only images, with no embedded text, making direct information extraction through conventional PDF processing techniques unfeasible.

To overcome this limitation, the ingestion pipeline was enhanced with optical character recognition (OCR) functionality, enabling the conversion of images containing text into processable data. This functionality was implemented using the Docling library, which, in addition to OCR, enabled: extraction of additional metadata from documents, expanding the available information for search; increased processing speed, making the pipeline more efficient; and structured capture of tables, allowing correctly organized financial information to be extracted.

After processing this domain documents with OCR enabled, the test queries from the were used to test the system and evaluate the recovery of information in the Investment Recommendation Letters dataset. Some sample interaction are shown in Figures 16, 17, and 18.


(Cartas Gestoras) Quais foram os ganhos da Ace Capital segundo a carta de 07/23?

(Cartas Gestoras) De acordo com a Carta do Gestor de julho de 2023, a Ace Capital FIC FIM registrou os seguintes ganhos:

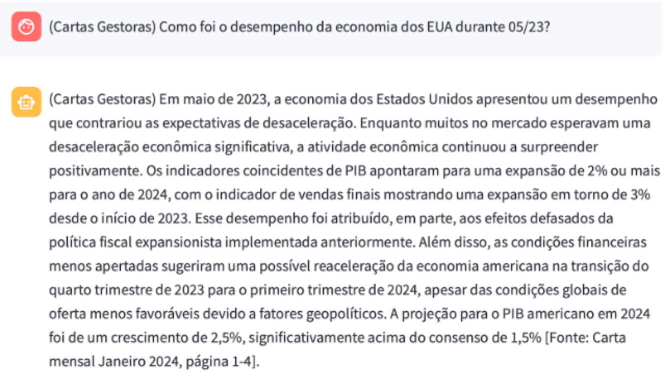
- **Ganho de 1,10% em julho de 2023.**
- **Ganho acumulado de 9,38% no ano de 2023,** o que representa 123% do CDI ou CDI+2,79% ao ano.
- **Ganho de 17,33% nos últimos 12 meses,** correspondendo a 128% do CDI ou CDI+3,34% ao ano.
- **Retorno acumulado de 50,68% desde o início do fundo em 30/09/2019,** equivalente a 162% do CDI ou CDI+3,66% ao ano.

Os livros de Renda Variável e Moedas contribuíram positivamente para o fundo em junho, com incrementos de 0,23% e 0,22%, respectivamente.

Fonte: Carta do Gestor Nº 46 – Julho de 2023, Ace Capital Gestora de Recursos Ltda, página 4.

Figure 16. Successful retrieval of financial data from PDF table. The system accurately identifies and extracts performance metrics from tabular content using OCR and chunk-based retrieval.

Comparing the answer in Figure 16 with the respective reference text shown in Table 5, it is possible to see that the system was able to not only retrieve the exact part it was supposed to, but also correctly extract the data from the table.

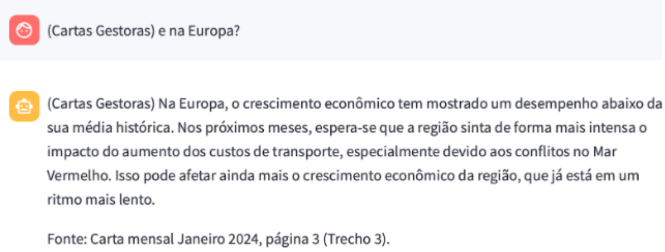


(Cartas Gestoras) Como foi o desempenho da economia dos EUA durante 05/23?

(Cartas Gestoras) Em maio de 2023, a economia dos Estados Unidos apresentou um desempenho que contrariou as expectativas de desaceleração. Enquanto muitos no mercado esperavam uma desaceleração econômica significativa, a atividade econômica continuou a surpreender positivamente. Os indicadores coincidentes de PIB apontaram para uma expansão de 2% ou mais para o ano de 2024, com o indicador de vendas finais mostrando uma expansão em torno de 3% desde o início de 2023. Esse desempenho foi atribuído, em parte, aos efeitos defasados da política fiscal expansionista implementada anteriormente. Além disso, as condições financeiras menos apertadas sugeriram uma possível reaceleração da economia americana na transição do quarto trimestre de 2023 para o primeiro trimestre de 2024, apesar das condições globais de oferta menos favoráveis devido a fatores geopolíticos. A projeção para o PIB americano em 2024 foi de um crescimento de 2,5%, significativamente acima do consenso de 1,5% [Fonte: Carta mensal Janeiro 2024, página 1-4].

Figure 17. Retrieval of extended financial context from related document. System correctly recognizes the month but includes additional information by retrieving a more general context.

In Figure 17, another example was tested. Although the system correctly answered the query by referencing the month of May, it retrieved a different document than expected, thus providing more information than requested.



(Cartas Gestoras) e na Europa?

(Cartas Gestoras) Na Europa, o crescimento econômico tem mostrado um desempenho abaixo da sua média histórica. Nos próximos meses, espera-se que a região sinta de forma mais intensa o impacto do aumento dos custos de transporte, especialmente devido aos conflitos no Mar Vermelho. Isso pode afetar ainda mais o crescimento econômico da região, que já está em um ritmo mais lento.

Fonte: Carta mensal Janeiro 2024, página 3 (Trecho 3).

Figure 18. System follows up on multi-turn conversation with expanded temporal scope. Highlights ability to maintain context but also shows tendency to support broader timeframes.

Figure 18 shows the following interaction made in Figure 17. It is possible to see that the system identified the query's meaning, recovering the context from the previous question. However, it still retrieved a different document than expected, considering the entire year of 2023 instead of just the month of May.

The same evaluation procedure described previously was applied to a second domain: investment recommendation letters using ChatGPT-4o as the evaluator model. The results, shown in Table 7, present a contrasting outcome to the Ágora domain. In this case, GraphRAG outperformed VectorRAG across all evaluation metrics, emphasizing the importance of domain adaptation when selecting retrieval strategies.

This performance difference can be attributed to the increased complexity of the documents and queries in this domain. Many evaluation questions require aggregating information from multiple sections or even across different documents. Graph-based retrieval is better suited to this scenario, as it can model semantically related content through interconnected nodes and communities in the generated graph structure.

Documents in this domain are also more structurally complex. In addition to dense text, they include relevant content embedded in images and tables, which adds complexity to the information extraction process. Document length is another contributing factor: while Ágora reports are typically 3–4 pages long, recommendation letters span 7–11 pages. These characteristics increase the benefit of a graph-based representation, which can better capture relationships across dispersed and heterogeneous content.

The combined approaches yielded intermediate performance results, balancing the strengths and weaknesses of each individual strategy. However, they also incurred higher token usage and computational costs, as expected. Among them, the complexity-aware hybrid strategy achieved higher faithfulness and slightly improved context precision and relevance, while reducing costs compared to always querying both retrieval systems. This demonstrates the efficacy and efficiency of the proposed complexity-based decision mechanism.

It is also worth noting that the combined strategies used a fixed number of documents (10) retrieved from VectorRAG, whereas the number retrieved from GraphRAG varied depending on the number of communities identified in the graph. This imbalance may have influenced the combined method's performance, and suggests a tunable parameter that could be optimized in future work to better balance the contributions of each retrieval source.

Therefore, the results demonstrate that integrating OCR enhanced the system's retrieval capabilities, enabling accurate extraction of information from images and tables. This advancement not only expanded the scope of searchable content but also ensured more comprehensive and precise data processing, contributing to a more robust and effective information retrieval system.

Finally, the inclusion of OCR in the document ingestion pipeline was crucial for this domain. It improved retrieval effectiveness by enabling accurate extraction of information from non-textual sources such as images and tables. This enhancement expanded the scope of searchable content, contributing to more robust and comprehensive system performance.

6 Architecture for Complex Interactions

Based on the experiments performed and the development of the conversational agent for the financial domain, the final architecture presented in Figure 19 was consolidated.

In Figure 19, the following components are highlighted, which are the main modules of the system (highlighted in dark blue):

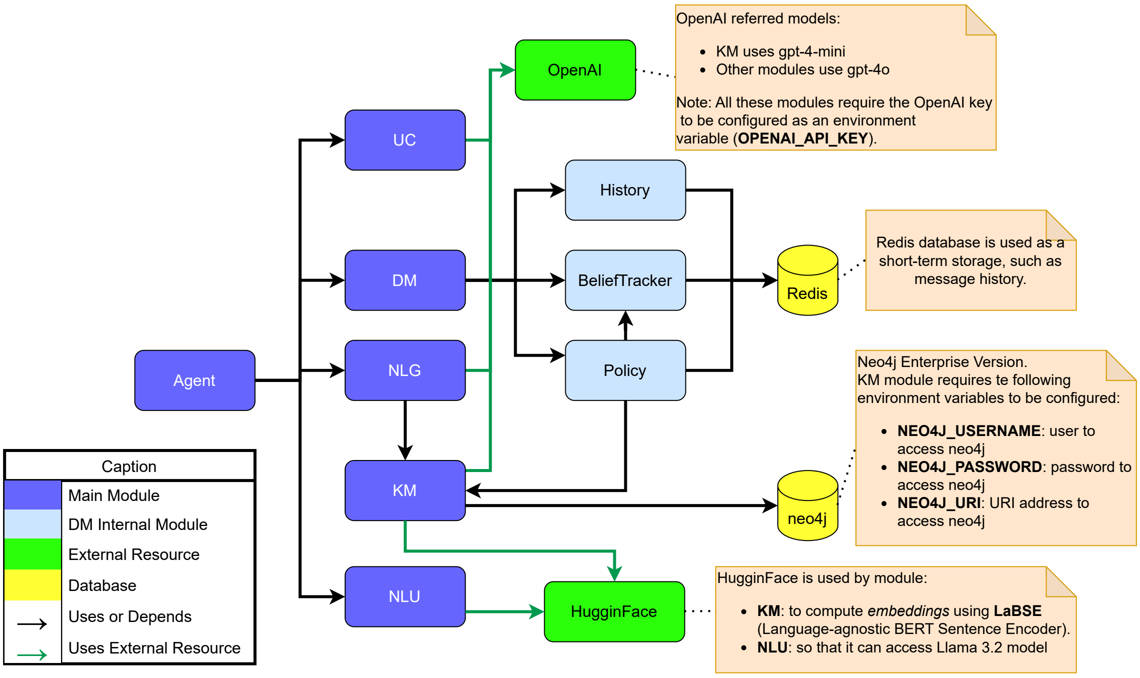
- Agent: module that aggregates the main functionalities of the conversational system;
- UC: User Connection functionalities;
- DM: Dialog Manager functionalities;
- NLG: Natural Language Generator functionalities;
- KM: Knowledge Management functionalities;
- NLU: Natural Language Understanding functionalities.

The components below, highlighted in light blue, are aggregated in the DM module. However, their visual separation in the diagram is important so that they can be identified by their main functionalities. These components are: History, Belief Tracker and Policy.

Still in Figure 19, the databases used by the conversational system are highlighted in yellow:

Search Approach	Faithfulness	CP w/o reference	CP w/ reference	Relevance	Eval in tokens	Eval out tokens	Gen time	Eval time
VectorRAG	0.67	0.24	0.12	0.33	3.03M	146k	661s	321s
GraphRAG	0.85	0.66	0.34	0.47	199k	180k	614s	329s
VectorRAG + GraphRAG	0.73	0.26	0.15	0.31	4.49M	246k	674s	501s
VectorRAG + GraphRAG complexity	0.78	0.29	0.15	0.32	4.39M	230k	669s	359s

Table 7. Evaluation result for search approaches in the Investment Recommendation Letters domain



OpenAI referred models:

- KM uses gpt-4-mini
- Other modules use gpt-4o

Note: All these modules require the OpenAI key to be configured as an environment variable (**OPENAI_API_KEY**).

Redis database is used as a short-term storage, such as message history.

Neo4j Enterprise Version.
KM module requires te following environment variables to be configured:

- **NEO4J_USERNAME**: user to access neo4j
- **NEO4J_PASSWORD**: password to access neo4j
- **NEO4J_URI**: URI address to access neo4j

HugginFace is used by module:

- **KM**: to compute *embeddings* using **LaBSE** (Language-agnostic BERT Sentence Encoder).
- **NLU**: so that it can access Llama 3.2 model

Figure 19. Full system architecture for complex, domain-specific dialog systems. Modular design supports vector/graph retrieval, multiple LLMs, metadata-aware ingestion, and conversation management using belief tracking and memory.

- neo4j: used to store the vector store and the graph store;
- Redis: used to store short-term memory data (dialog).

All of the aforementioned modules and components are interconnected with each other with black arrows, indicating which component/module uses the other component/module.

However, there is a dependency on resources external to the system, hosted on the internet. There are two such resources (the modules that use these resources are indicated by green arrows):

- OpenAI: resource or service required to use the gpt-4o model;
- HuggingFace: resource or service required to access the Llama 3.2 model and access LaBSE (Language-agnostic BERT Sentence Encoder).

The proposed technical architecture was designed to enable modular integration with multiple AI services in the cloud and support for different LLMs, including OpenAI, Azure OpenAI and local models such as Llama. The choice of technologies such as Neo4j, Redis, and Streamlit aimed to provide fast retrieval and flexible interface development. In addition, the migration of microservices to a component-based approach supports easy maintenance and reuse of code. This architecture was used to run the experiments reported in Section 5, which covered diverse problem situations described below.

The documents used by RAG contained detailed information about companies, including financial results and data on expansion. The objective of the test was to evaluate the agent's ability to extract information from these files and respond accurately and contextually, according to the content available in the documents. However, the questions asked were not restricted to direct questions. Questions that required temporal interpretation, inference of information not explicitly mentioned in the PDFs, dependence on previous answers, analysis of bank transactions, the ability to compare multiple documents, and long-term context maintenance were included. In addition, complex questions were tested, such as those composed of multiple sentences, containing acronyms not previously defined, and with abrupt changes in context.

The complexity of the tests was increased by the large volume of documents analyzed and the need for computing power to process the information efficiently. The variability of the questions required the system to be able to handle multiple scenarios and adapt to different contexts, highlighting the challenges involved in the automated extraction of information from extensive and diverse documents.

6.1 Temporal Reference Extraction

Among the tests carried out during the project, it is possible to mention the test of the process of extracting temporal references contained in the chunks stored in the vectorstore. Extracting times contained in a text is a difficult process, since times can be expressed in different ways, such as explicitly (day/month/year, day/month, month/year, 3Q22, etc.) and relatively (yesterday, today, tomorrow, annually, at the beginning of..., etc.). Because of this, it becomes impractical to develop a code that handles all the possibilities of expressing time. To get around this situation, it was decided to use LLM to extract the time information contained in a chunk.

The test procedure consisted of generating a file containing queries from Àgora Broker domain and running a program that:

- For each query contained in the query file;
- Searches for k results (chunks) most similar to the query stored in the vector store;
- Using LLM, extracts all time references from both the query and the results obtained from the search.

The LLM successfully identified explicit and relative temporal references across tested queries, manual inspection of the outputs confirmed correct extraction in the sampled cases. This experiment also served as an initial check of the retrieval process from both the vector store and the graph store, confirming that relevant chunks were returned for temporal queries.

6.2 Direct Information Extraction

This test focused on extracting information that was explicitly available in one of the Ágora Brokerage PDFs. In this case, there was no need to interpret the context, correlate different excerpts or infer non-explicit data. The extraction of the answers combined data from the vector store and the graph store, which allowed verification that both sources could return the correct document segments for direct questions. In this scenario, the agent only needed to locate the exact answer in the document, making the process straightforward and less demanding in terms of reasoning steps compared to tasks involving inference or long-context analysis.

6.3 Dependent Questions

To deal with dependent questions, it was necessary to implement a chat history mechanism based on the chat ID. This system allowed the agent to store all interactions associated with the same identifier, enabling the recovery of the complete history of previous questions and answers. This mechanism enabled the agent to access the full record of previous questions and answers whenever needed, maintaining continuity in the dialog.

Additionally, processing these questions required a more advanced level of contextual information. In many cases, the agent needed to identify implicit connections between different parts of the conversation by recognizing previously discussed terms and concepts. This was achieved by extracting relevant information from the PDFs by combining data from the vector store and the graph store, basing the answers on the available content.

By integrating the chat history with the information extraction mechanisms, the system was able to consider previous context when selecting document segments and formulating responses. This solution allowed the model to respond contextually, adapting to the nuances of the conversation and providing complete information even when the question did not directly mention all the necessary details.

6.4 Comparison Capability

Comparison capability represented one of the most complex challenges for the agent, as it required not only extracting information from multiple PDFs, but also correctly interpreting the question. Comparisons involve identifying similarities and differences between financial data, company performance, or trends over time. The first challenge was recognizing that the question asked for a comparative analysis, which was not always explicit. In addition, it was necessary to locate the correct information in different sources, ensuring that the extracted data was truly comparable.

To deal with these difficulties, the agent used a combination of the vector store and the graph store to retrieve the relevant information from different documents. The global search capability in the graph store allows the retrieval of information from several different documents simultaneously,which supports comparisons that span multiple sources. The search in the vector store allows for easy finding of comparisons made in nearby sections or in the same document, when the original text presents comparative analyses. By integrating these two mechanisms, the agent was designed to select and present information in a way that reflects the relationships between different data points without requiring manual intervention.

6.5 Question with Multiple Sentences

Questions composed of multiple sentences often present varied demands within a single message, requiring the system to interpret and respond appropriately to each of them. To deal with this type of query, the NLU module was configured to interpret multiple components in a single interaction turn, enabling the system to generate responses that address each part of the input.

The retrieval process combines information from both the vector store and the graph store. Initially, the search in the vectorstore returns a set of documents closely related to the user query, while the graph store enables a global query for semantic connections between the data. The content retrieved from these sources is then incorporated into the LLM prompt, providing the necessary context for it to generate a comprehensive response. This method allows the model to connect scattered information and provide structured and coherent responses, even in queries that require multiple responses within a single interaction.

6.6 Sudden Context Switching

The maintenance of the conversation context is done by the Belief Tracker (BT) module of the developed system. It has the ability to keep the conversation state updated, complementing it with new information at each conversation turn and ensuring that different contexts are managed separately. When there is a context switch in the conversation, a new state is created for that specific context, allowing for the organization of the history. This design enables the system to manage multiple contexts separately during a conversation.

7 Discussion

The results from both domains offer valuable insights into the performance dynamics of different retrieval strategies in RAG systems.

Across domains, VectorRAG and GraphRAG exhibited complementary strengths, reinforcing the notion that no single retrieval method universally outperforms the other. In the Ágora domain, which features shorter, well-structured, and text-heavy documents, VectorRAG consistently delivered higher faithfulness and relevance. This suggests that dense vector embeddings are well-suited for domains with consistent vocabulary and self-contained content blocks.

In contrast, the investment recommendation letters domain favored GraphRAG, which outperformed VectorRAG across all metrics. This can be attributed to the greater structural and semantic complexity of the documents in this domain, where relevant information is dispersed across longer texts, and often embedded in images and tables. The graph-based retrieval method was more effective at modeling these cross-sectional and cross-document relationships, offering improved coverage and contextual linkage.

The combined retrieval strategies produced intermediate results in both settings, validating their role as compromise solutions. However, their increased computational cost, reflected in higher token usage and longer evaluation times, highlights the importance of strategic resource allocation in practical deployments. The complexity-aware hybrid approach, which selectively invokes GraphRAG for difficult queries, showed particular promise by delivering strong performance with lower cost—demonstrating that adaptive retrieval mechanisms can optimize the trade-off between effectiveness and efficiency.

These findings underscore the critical role of domain characteristics, such as document length, structure, and content heterogeneity, in guiding the selection and configuration of retrieval components in RAG systems. Future work could explore dynamic parameter tuning (e.g., varying the number of retrieved documents per method) and more advanced query classification strategies to further enhance adaptability and performance.

7.1 Lessons Learned

The development and evaluation of complex RAG architectures in dialog systems for Brazilian Portuguese led to several practical and technical insights:

- **Vector vs. Graph Trade-offs**: VectorRAG provided faster responses and simpler setup, while GraphRAG enhanced accuracy in multi-hop queries and temporal disambiguation;
- **Semantic Chunking Improves Relevance**: The use of semantic chunking and hierarchical clustering helped avoid context fragmentation and improved retrieval relevance, especially when dealing with documents containing dense financial knowledge;
- **Human-in-the-Loop Remains Key**: Manual evaluation was still necessary to validate subtle aspects of accuracy, especially in ambiguous or dependent questions. This suggests the need for hybrid evaluation methods combining automatic metrics with expert reviews;
- **Modularity Enables Scalability**: The modular architecture—combining open-source tools, local and API-based LLMs, and flexible microservices—proved highly reusable and extensible for other specialized domains beyond finance;
- **Importance of Metadata**: Proper metadata tagging during ingestion (e.g., titles, dates, sections) played a important role in improving retrieval precision and grounding responses with source references.

These lessons will guide future developments in building robust domain-specific conversational agents, not only in Brazilian Portuguese for Banking, but also in other application areas.

7.2 Research Limitations

While the presented approach achieved promising results in handling complex interactions in dialog systems using Retrieval-Augmented Generation (RAG) for Brazilian Portuguese, some limitations must be acknowledged:

- **Dataset Size and Diversity**: The case studies were based on specific financial domains, particularly investment brokerage reports and recommendation letters. The case studies do not fully capture the diversity of language and complexity found in broader financial contexts or in other specialized domains;
- **Dataset for RAG Only**: The case studies provided real-world datasets for the construction of the databases to test different RAG approaches, but no real-world dataset of user conversations was available. For a in-depth evaluation, user interactions datasets are required;
- **Language-Specific Evaluation**: Most RAG benchmarks and evaluation tools are originally developed for English. The adaptation of metrics and tools to Brazilian Portuguese may have introduced limitations in standardization and comparison across different languages;
- **Ground Truth Alignment**: The evaluation relied on human-curated responses and context references, which, while accurate, may introduce bias or variability in interpretation. The lack of standardized ground truth datasets in Portuguese for RAG evaluation presents challenges in reproducibility;
- **Computational Constraints**: Some approaches like HippoRAG and HybridRAG, despite being more powerful for complex reasoning, demonstrated higher latency

and resource consumption, limiting their scalability in real-time environments;
- **Model Dependency**: The accuracy and coherence of generated responses are dependent on the underlying LLM. Although flexible, this introduces variability when switching between models (e.g., GPT-4o and LLaMA 3.2), especially when dealing with subtle linguistic features in Brazilian Portuguese;
- **Long-Term Evaluation**: The persistent memory mechanisms (e.g., HippoRAG) were tested in controlled environments. However, long-term use in production scenarios—where memory evolves continuously and must handle outdated or contradictory information—remains an open challenge.

These limitations underscore the importance of further studies involving multilingual benchmarking, broader domain applicability, and long-term deployment constraints. Additionally, it is crucial to discuss how such technical limitations can directly affect the practical application of these approaches in real contexts, such as financial institutions, and to propose ways to mitigate these challenges in future work.

8 Conclusion

This paper presented different complex RAG approaches to use LLM to support complex interactions in a specific domain for a dialog system in Brazilian Portuguese. The models were evaluated in the banking context using real-world datasets from two case studies. The presented analysis highlights the strengths and limitations of each method. Experimental results indicate that context-aware retrieval strategies improve response accuracy when addressing ambiguous or multi-faceted user queries.

In the Ágora case study, which involved concise financial reports with structured tabular data, VectorRAG proved to be the most effective technique. It achieved the highest faithfulness (0.66) and Relevance (0.56) with a reasonable response time (697 seconds). In the Investment Recommendation Letters case study, where documents were longer, textual, and semantically interlinked, GraphRAG outperformed all other techniques. It reached faithfulness of 0.85 and relevance of 0.47, demonstrating its strength in capturing entity relationships and discourse structure. The combined approach presents comparable metrics to each best technique in both scenarios, showing its adaptability.

However, trade-offs in computational efficiency and response time remain critical challenges. Our findings provide insights into optimizing dialog systems for Brazilian Portuguese, paving the way for domain-specific conversational agents in financial and other specialized applications. Considering that RAG-based approaches do not require model modification with costly training processes, the associated environmental impact is considered low when compared to finetuning-based alternatives.

During implementation, various practical challenges emerged — including inconsistencies in document formats (e.g., scanned PDFs), difficulties in extracting temporal information, and memory management in long-term dialogues. The integration of OCR, chunking strategies, and metadata-driven retrieval helped mitigate some of these issues, but adapting each RAG approach required iterative tuning and domain-specific adjustments. These practical insights reaffirm the need for flexible and modular architectures when dealing with real-world data.

The main scope of this work was specialization, contextual search and disambiguation, and advanced solutions in RAG. Among the main developments suggested for future work, we highlight the implementation of Specialized Semantic Search for large volumes of data and the deepening of advanced RAG approaches, such as Re-Ranking and the incorporation of Relevance and Groundness metrics. In addition, fundamental challenges remain open, including the integration of multimodality, the application of Chain of Thought (CoT) techniques, and the orchestration of multi-agent systems, which can enhance efficiency. The prioritized next step is Specialized Semantic Search considering the research roadmap.

Applying advanced RAG systems in sensitive domains such as Finance also raises ethical challenges — including potential misinformation, hallucinations, and over-reliance on AI-generated content. To address this, the authors implemented data curation mechanisms. Future work may focus on explainability, human-in-the-loop oversight, and robust auditing to ensure responsible deployment in real financial environments. Furthermore, it is also important to consider the existing literature on ethical and social impacts, especially in banking applications, where the use of Artificial Intelligence can influence critical decisions that directly affect the general public. In particular, it is also recommended that future research explore detailed analyses of computational costs, and studies of direct applicability in educational and commercial contexts.

Declarations

Authors' Contributions

Conceptualization, J.C.N. and W.V.R.; methodology, J.C.N. and W.V.R.; software, F.C.d.A.P., V.T.H., J.C.N., R.B.Jr, S.K., I.S.T., J.V.S. and L.G.M.; validation, J.C.N. and W.V.R.; investigation, F.C.d.A.P., V.T.H., J.C.N., R.B.Jr, S.K., I.S.T., J.V.S. and L.G.M.; resources, F.C.d.A.P., V.T.H., J.C.N., R.B.Jr and S.K.; data curation, F.C.d.A.P., V.T.H., J.C.N., R.B.Jr, S.K., I.S.T., J.V.S. and L.G.M.; writing—original draft preparation, F.C.d.A.P., V.T.H., R.B.Jr, S.K., I.S.T., J.V.S. and L.G.M.; writing—review and editing, F.C.d.A.P., V.T.H., R.B.Jr, S.K., I.S.T., J.V.S. and L.G.M.; supervision, J.C.N. and W.V.R.; project administration, J.C.N. and W.V.R.; funding acquisition, J.C.N. and W.V.R. All the authors have read and agreed to the published version of the manuscript. F.C.d.A.P. is the main contributor and V.T.H. is the main writer of this manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The authors thank the support provided by Bradesco Bank and the Laboratory of Computer Architecture and Networks (LARC) at the Computing and Digital Systems Engineering Department (PCS), Polytechnic School, Universidade de São Paulo, Brazil.

Funding

This research was funded by the Graduate Program in Electrical Engineering (PPGEE) from the Polytechnic School of the Universidade de São Paulo.

Availability of data and materials

The datasets (and/or softwares) generated and/or analysed during the current study will be made upon request.

References

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Goffinet, É., Hesslow, D., Launay, J., Malartic, Q., *et al.* (2023). The falcon series of open language models. *arXiv preprint arXiv:2311.16867*. DOI: 10.48550/arxiv.2311.16867.

Chan, B. J., Chen, C.-T., Cheng, J.-H., and Huang, H.-H. (2025). Don't do rag: When cache-augmented generation is all you need for knowledge tasks. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 893–897. DOI: 10.1145/3701716.3715490.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., and Larson, J. (2024). From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*. DOI: 10.48550/arxiv.2404.16130.

Guo, Y., Tao, Y., Ming, Y., Nowak, R. D., and Liang, Y. (2025). Retrieval-augmented generation as noisy in-context learning: A unified theory and risk bounds. *arXiv preprint arXiv:2506.03100*. DOI: 10.48550/arXiv.2506.03100.

Gutiérrez, B. J., Shuv, Y., Gu, Y., Yasunaga, M., and Su, Y. (2024). Hipporag: Neurobiologically inspired long-term memory for large language models. *arXiv preprint arXiv:2405.14831*. DOI: 10.48550/arxiv.2405.14831.

Hadi, M. U., Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M. B., Akhtar, N., Wu, J., Mirjalili, S., *et al.* (2023). A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*, 3. DOI: 10.36227/techrxiv.23589741.v1.

Liang, L., Bo, Z., Gui, Z., Zhu, Z., Zhong, L., Zhao, P., Sun, M., Zhang, Z., Zhou, J., Chen, W., *et al.* (2025). Kag: Boosting llms in professional domains via knowledge augmented generation. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 334–343. DOI: 10.1145/3701716.3715240.

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., *et al.* (2023). Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-radiology*, 1(2):100017. DOI: 10.1016/j.metrad.2023.100017.

Manning, C. D. (2009). *An introduction to information retrieval*. Cambridge University Press. Book.

Meta (2025). Llama3.2. Available at: `https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_2/`.

Moreira, V. P. (2023). Capítulo 19 recuperação de informação. *Brasileiras em PLN*. Available at: `https://brasileiraspln.com/livro-pln/2a-edicao/parte-aplicacoes/cap-ir/cap-ir.pdf`.

OpenAI (2025). Models. Available at: `https://platform.openai.com/docs/models`.

Ozdemir, S. (2023). *Quick start guide to large language models: strategies and best practices for using ChatGPT and other LLMs*. Addison-Wesley Professional. Book.

Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C., Zhang, Y., and Tang, S. (2024). Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*. DOI: 10.48550/arxiv.2408.08921.

Phan, H., Acharya, A., Meyur, R., Chaturvedi, S., Sharma, S., Parker, M., Nally, D., Jannesari, A., Pazdernik, K., Halappanavar, M., *et al.* (2024). Examining long-context large language models for environmental review document comprehension. *arXiv preprint arXiv:2407.07321*. DOI: 10.48550/arXiv.2407.07321.

Pinna, F. C. d. A., Hayashi, V. T., Néto, J. C., Marquesone, R. d. F. P., Duarte, M. C., Okada, R. S., and Ruggiero, W. V. (2024). A modular framework for domain-specific conversational systems powered by never-ending learning. *Applied Sciences*, 14(4). DOI: 10.3390/app14041585.

Qian, H., Zhang, P., Liu, Z., Mao, K., and Dou, Z. (2024). Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591*. DOI: 10.48550/arxiv.2409.05591.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., *et al.* (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9. Available at:`https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe`.

Sarmah, B., Hall, B., Rao, R., Patel, S., Pasquali, S., and Mehta, D. (2024). Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. *arXiv preprint arXiv:2408.04948*. DOI: 10.1145/3677052.3698671.

Sarmah, B., Zhu, T., Mehta, D., and Pasquali, S. (2023). Towards reducing hallucination in extracting information from financial reports using large language models. in: Proceedings of the third international conference on aiml systems. In *In: Proceedings of the Third International Conference on AI-ML Systems*, pages 1–5. DOI: 10.48550/arXiv.2310.10760.

Shahul Es, Jithin James, L. E.-A. S. S. (2023). Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*. DOI: 10.18653/v1/2024.eacl-demo.16.

Soudani, H., Kanoulas, E., and Hasibi, F. (2024). Fine tuning vs. retrieval augmented generation for less popular knowledge. In *Proceedings of the 2024 Annual International*

ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, pages 12–22. DOI: 10.1145/3673791.3698415.

Team, D. S. (2024). Docling technical report. DOI: 10.48550/arXiv.2408.09869.

Ultes, S., Barahona, L. M. R., Su, P.-H., Vandyke, D., Kim, D., Casanueva, I., Budzianowski, P., Mrkšić, N., Wen, T.-H., Gasic, M., *et al.* (2017). Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78. DOI: 10.18653/v1/p17-4013.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30. DOI: 10.48550/arxiv.1706.03762.

Wang, J., Ma, W., Sun, P., Zhang, M., and Nie, J.-Y. (2024). Understanding user experience in large language model interactions. *arXiv preprint arXiv:2401.08329*. DOI: 10.48550/arXiv.2401.08329.

Yang, A. (2024). Old wine in a new bottle: How hipporag revolutionizes retrieval with knowledge graphs. Available at: `https://angelina-yang.medium.com` March 30, 2025.

Yu, H., Gan, A., Zhang, K., Tong, S., Liu, Q., and Liu, Z. (2024). Evaluation of retrieval-augmented generation: A survey. In *CCF Conference on Big Data*, pages 102–120. Springer. DOI: 10.1007/978-981-96-1024-2_8.

Zhang, B., Liu, Z., Cherry, C., and Firat, O. (2024). When scaling meets llm finetuning: The effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*. DOI: 10.48550/arXiv.2402.17193.

Zhao, S., Yang, Y., Wang, Z., He, Z., Qiu, L. K., and Qiu, L. (2024). Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*. DOI: 10.48550/arXiv.2409.14924.

Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Chen, H., Liu, Z., Dou, Z., and Wen, J.-R. (2023). Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*. DOI: 10.1145/3748304.