

Improving Large Language Models Responses with Retrieval Augmented Generation in Animal Production Certification Platforms

Pedro Bilar Montero^a, Jonas Bulegon Gassen^b, Glênio Descovi^c, Vinícius Maran^d,
Tais Oltramari Barnasque, Matheus Friedhein Flores^e and Alencar Machado^f

Laboratory of Ubiquitous, Mobile and Applied Computing (LUMAC), Federal University of Santa Maria, Brazil
{pedro.bilar.montero, glenio.descovi, viniciusmaran, matheusfriedhein, alencar.comp}@gmail.com, jonas.gassen@ufsm.br;

Keywords: Retrieval-Augmented Generation, Large Language Models, Poultry Health, Sanitary Certification, PDSA-RS, Brazilian Regulations, Legal Texts, Animal Health, Natural Language Processing.

Abstract: This study explores the potential of integrating Large Language Models (LLMs) with Retrieval-Augmented Generation (RAG) to enhance the accuracy and relevance of responses in domain-specific tasks, particularly within the context of animal health regulation. Our proposal solution incorporates a RAG system on the PDSA-RS platform, leveraging an external knowledge base to integrate localized legal information from Brazilian legislation into the model's response generation process. By combining LLMs with an information retrieval module, we aim to provide accurate, up-to-date responses grounded in relevant legal texts for professionals in the veterinary health sector.


1 INTRODUCTION


The field of artificial intelligence (AI) has made significant advancements in recent years, encompassing a variety of subfields like computer vision, robotics, and, most notably, natural language processing (NLP). NLP has especially thrived with the emergence of Large Language Models (LLMs), which have achieved state-of-the-art results across a range of tasks, including text generation, summarization, and language translation. These models, based on deep neural networks and transformer architectures, learn from vast corpora and adapt to complex linguistic patterns with minimal supervision, allowing them to generate coherent and contextually relevant text in diverse applications (Brown et al., 2020). However, while LLMs demonstrate broad generalization abilities, they often encounter challenges when applied to specialized domains, as these areas require extensive, localized knowledge and precise understanding that is not always encapsulated within large, generic datasets


(Gururangan et al., 2020; Bommasani et al., 2021)


In industry, the integration of AI into operational systems has led to significant advancements across sectors such as healthcare, finance, and legal domains (Bommasani et al., 2021; Brown et al., 2020). Within the domain of animal health, for example, LLMs hold potential to streamline administrative and compliance processes by aiding in decision-making, regulatory adherence, and response generation. However, domain-specific applications introduce complexities that require accurate, context-aware responses to nuanced queries (Gururangan et al., 2020).


The regulatory frameworks governing animal health, particularly in Brazil, exemplify this challenge, as professionals must navigate intricate legal and sanitary guidelines to ensure compliance and protect public health. The Brazilian Ministério da Agricultura e Pecuária (MAPA) plays a key role in overseeing and enforcing animal health regulations. MAPA's responsibilities include regulating veterinary practices, approving vaccines, and monitoring the health status of livestock throughout the country. A specific challenge in animal health regulation in Brazil is ensuring compliance with the sanitary certification processes that guarantee the safety of animal products for both domestic consumption and international trade. The certification process, which verifies that a farm or animal facility meets required sanitary


^a  <https://orcid.org/0009-0002-9224-7694>

^b  <https://orcid.org/0000-0001-8384-7132>

^c  <https://orcid.org/0000-0002-0940-9641>

^d  <https://orcid.org/0000-0003-1916-8893>

^e  <https://orcid.org/0000-0003-4436-4327>

^f  <https://orcid.org/0000-0002-6334-0120>

standards, is particularly critical for industries such as poultry farming. These certifications ensure that animal products are free from diseases like avian influenza, salmonella, and mycoplasmosis, which are both economically damaging and potentially harmful to humans.

The Plataforma de Defesa Sanitária Animal do Rio Grande do Sul (PDSA-RS) is a platform designed to support the field of animal health regulation in Brazil by implementing an information system that integrates all stages of certification processes for poultry and swine farming in Rio Grande do Sul. This platform facilitates the organization of production activities while ensuring sanitary compliance with Brazilian animal health regulations. In the case of the PDSA-RS, veterinary certification processes depend on the model's capacity to interpret complex Brazilian legislation on animal health (Descovi et al., 2021; Ebling et al., 2024; Schneider et al., 2024).

To address these specialized needs, the Retrieval-Augmented Generation (RAG) framework has emerged as a solution for enhancing LLM capabilities by allowing the models to access external knowledge bases. This framework retrieves relevant information from a connected knowledge base during the response-generation process, making it a valuable approach for domains where specialized, dynamic knowledge is required. Studies show that RAG systems can effectively improve the accuracy of generated responses in specialized fields by supplementing LLMs with precise, domain-relevant information (Lewis et al., 2020; Karpukhin et al., 2020).

In this study, we aim to implement a RAG system that can be integrated into the PDSA-RS platform, enhancing its response-generation process with domain-specific knowledge pertinent to the animal health regulatory environment in Rio Grande do Sul. By integrating retrieval mechanisms that draw from a knowledge base of Brazilian legislation, our system can produce contextually grounded responses aligned with the requirements of the animal health regulation in Brazil. This RAG integration holds the potential to bridge the gap between general-purpose language models and the precise, regulatory-driven needs of professionals in the veterinary health sector, contributing to a more effective and reliable AI application within the industry.

Our study case will focus on the integration of this RAG system within the PDSA, specifically in the module responsible for poultry certification. Our objective is to create an assistant that will help the professional responsible for analyzing poultry certification processes to make decisions more quickly. The

purpose of this assistant is to validate all relevant data for these requests, which will be processed by our RAG system. Our focus will be to evaluate its performance in providing accurate responses to certification and regulatory questions.

Our paper is organized as follows: Section 2 covers the background of our research, showing all needed for this paper. In Section 3 we present our methodology describing how we organized our architecture and developed the RAG System. Section 4 is our study case about the implementation of the RAG System within the PDSA-RS and in Section 5 we show the conclusions of this work and future research possibilities.

2 BACKGROUND

This section provides context for the concepts utilized in this paper.

2.1 Artificial Intelligence (AI)

Artificial Intelligence (AI) focuses on systems capable of performing tasks requiring human intelligence. Over decades, AI has progressed remarkably, driven by machine learning (ML) and neural networks, which shifted from symbolic logic-based methods to data-driven approaches capable of identifying patterns in unstructured data (Bishop, 2006). The advent of deep learning enabled significant advancements, leveraging multilayered neural networks to process complex data representations effectively (LeCun et al., 2015). These breakthroughs led to practical AI applications in healthcare, finance, and legal compliance (Esteva et al., 2017; Sari and Indrabudiman, 2024). Among these advancements, Large Language Models (LLMs) have emerged as transformative tools for tasks like text generation, summarization, and translation, powered by innovations like the Transformer architecture (Vaswani et al., 2017). AI's potential lies in its adaptability to domains where contextual precision is critical.

2.2 Large Language Models and Specialized Domains

A Large Language Model (LLM) is a type of artificial intelligence (AI) designed to process and understand human language at scale. These models are trained on vast amounts of text data, enabling them to learn patterns, relationships, and nuances of language. LLMs have made a lot of progress in recent years, achieving state-of-the-art results in various NLP tasks such

as language translation, question answering, and text generation.

The development of the transformer architecture by (Vaswani et al., 2017) laid the groundwork for numerous advancements in large-scale language modeling, leading to the creation of influential models such as GPT by OpenAI. These models demonstrated that language generation could achieve unprecedented levels of fluency, coherence, and adaptability in tasks like translation, summarization, and question answering (Radford and Narasimhan, 2018; Brown et al., 2020). However, most early LLMs were proprietary, restricting their accessibility and limiting the potential for customization and improvement by the broader research community.

The release of the LLaMA (Large Language Model Meta AI) family by Meta marked a significant shift in this paradigm, as it offered high-performance, large-scale models with an open-access architecture. LLaMA's open-source nature allows researchers and developers to fine-tune and adapt the model for specific tasks, including specialized domains that require knowledge and expertise beyond general language capabilities. This openness has made LLaMA particularly valuable in academic and applied research settings, where access to large models with flexible adaptation options is essential for innovation (Touvron et al., 2023).

However, despite their impressive capabilities, LLMs often struggle with domain-specific tasks that require specialized knowledge and context. This limitation arises from several factors:

- **Lack of Domain Expertise.** While LLMs can be fine-tuned on specific domains, they may not fully grasp the intricacies of that domain without extensive training data (Zhang et al., 2023).
- **Domain-Specific Nuances.** Domains like law, healthcare, finance, or agriculture involve complex rules, exceptions, and subtleties that may be difficult for LLMs to capture without explicit training on these domains (Google Cloud, 2023).
- **Limited Contextual Understanding.** LLMs rely on statistical patterns in language to make predictions or generate text. However, this approach can lead to misinterpretation or misunderstanding of domain-specific context, leading to inaccurate results (Brown et al., 2020).

These limitations have led to the development of hybrid architectures, such as Retrieval-Augmented Generation (RAG), that aim to enhance LLMs by incorporating external knowledge sources to improve their performance in specific fields (Lewis et al., 2020; Karpukhin et al., 2020). Unlike fine-tuning,

which requires training the model on a domain-specific dataset and periodically retraining to update its knowledge, RAG offers a more dynamic approach by retrieving relevant information from an external database or knowledge source in real-time. This capability is particularly advantageous in fields with frequently updated information, such as legal, medical, and regulatory domains, where the model's responses need to reflect the latest standards and guidelines.

2.3 Retrieval-Augmented Generation (RAG) Architecture

Retrieval-Augmented Generation (RAG) is an architecture that enhances large language models (LLMs) by dynamically combining information retrieval with text generation, thus allowing LLMs to leverage external knowledge sources while generating responses. Unlike traditional LLMs that rely solely on pre-trained knowledge, RAG introduces a retrieval mechanism that fetches relevant external documents or data points, incorporating them into the generation process for more contextually accurate responses (Lewis et al., 2020; Karpukhin et al., 2020).

In RAG, the process starts with transforming user input into an embedding — a mathematical representation of the text. This embedding is then used to search a vector database, where documents are pre-processed and stored as embeddings. The vector database is essential for finding semantically relevant information in response to the user's query, as it allows efficient matching of queries with stored knowledge chunks. Upon retrieval, these documents are integrated into the generation component of the model, producing responses that are contextually relevant and informed by up-to-date knowledge sources (Lewis et al., 2020).

Figure 1 outlines the RAG workflow in four steps:

- **1:** The user inputs a query, for example, asking a question about a recent topic that isn't present on the training data of the LLM.
- **2:** This step illustrates the indexing of documents that are split into chunks, encoded into vectors, and stored in a vector database. The user query is then used to search this database for relevant content by a similarity search.
- **3:** After the search, the most relevant chunks are retrieved based on semantic similarity.
- **4:** After the search, the most relevant chunks are retrieved based on semantic similarity.

One of the significant advantages of RAG is its ability to integrate continuously updated information,

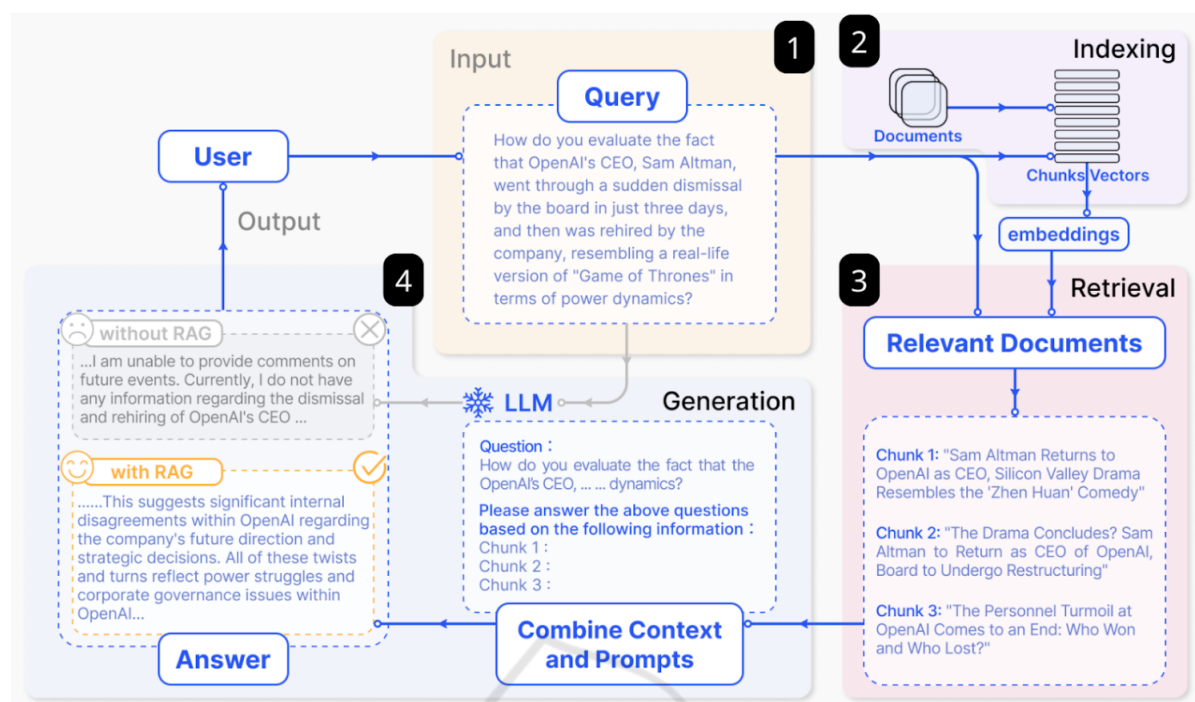


Figure 1: Illustration of the RAG Architecture: Information retrieval and generation process. Adapted from (Gao et al., 2023).

making it particularly valuable in fields that require timely data, such as regulatory environments where guidelines frequently change. By dynamically accessing relevant external content, RAG reduces the risk of outdated or inaccurate responses, improving both the accuracy and relevance of the model's output. Furthermore, RAG minimizes the computational demands associated with continuous fine-tuning, as it allows the model to access and integrate domain-specific knowledge on demand rather than retraining on static datasets (Izcard and Grave, 2021; Lewis et al., 2020).

In summary, RAG's approach of combining retrieval and generation enables LLMs to adapt to specific knowledge requirements, providing more robust support in specialized domains that benefit from up-dated and context-sensitive responses.

2.4 Fine-Tuning vs Retrieval-Augmented Generation (RAG)

In this section, we delve deeper into whether retrieval modules, as employed in RAG, provide a more robust benchmark than fine-tuning techniques for achieving domain-specific accuracy and efficiency.

2.4.1 Retrieval as a Benchmark for Domain-Specific Accuracy

As Section 2.3 outlines, RAG serve as a dynamic component by enabling LLMs to access external, up-to-date knowledge bases in real time. This capability contrasts with fine-tuning, which relies on embedding static domain knowledge into the model. Retrieval's real-time adaptability ensures that responses remain accurate in evolving fields such as legal or regulatory environments.

2.4.2 Efficiency in Computational Resource Utilization

Fine-tuning techniques, while effective, demand extensive computational resources and data curation. QLoRA, for instance, reduces these requirements by fine-tuning low-rank adaptation layers (Dettmers et al., 2024), yet it still requires periodic retraining to incorporate new domain knowledge. Retrieval, on the other hand, bypasses this need by separating knowledge storage from the generative process, thereby minimizing overhead and ensuring efficient use of resources (Lewis et al., 2020).

2.4.3 Comparative Evaluation Metrics

When considered as a tool for achieving domain-specific accuracy, retrieval modules excel in:

- **Adaptability.** Retrieval enables models to respond to domain-specific queries with real-time context, making it better suited for fields with dynamic knowledge requirements.
- **Scalability.** By offloading knowledge storage to external databases, retrieval reduces the need for model scaling, unlike fine-tuning which often requires larger model sizes to capture domain intricacies.
- **Benchmarking Potential.** Retrieval serves as an ongoing benchmark by continuously updating its knowledge base, allowing real-world validation of LLM performance in specialized domains.

From a theoretical standpoint, retrieval modules highlight a paradigm shift in LLM optimization by decoupling knowledge retrieval from generative capabilities. While fine-tuning such as QLoRA embeds domain-specific expertise into the model, retrieval treats knowledge as an external, modular component. This distinction positions retrieval as not merely a complement to fine-tuning but as a potential alternative benchmark for evaluating domain-specific effectiveness.

2.5 PDSA-RS and Animal Health Regulations in Brazil

Established in 2019, the Plataforma de Defesa Sanitária Animal do Rio Grande do Sul (PDSA-RS) supports animal health and production in Rio Grande do Sul through a real-time digital platform for managing certifications and ensuring compliance with sanitary regulations. Developed by the Federal University of Santa Maria and MAPA, with Fundesa's support, it enhances biosecurity, traceability, and export facilitation.

The system's modules, such as the poultry health certification feature, streamline data collection and certification issuance, linking veterinary inspections, laboratories, and the agricultural defense authorities. This interconnected system helps officials monitor disease control in flocks and facilitates efficient response to health risks. PDSA-RS allows inspectors and producers to follow up on health tests, sample processing, and the issuance of certificates required for both domestic and international movement of poultry, ensuring that health standards are met consistently.

As illustrated in figure 2, the platform adopts a microservices-oriented architecture. The front-end comprises several specialized portals tailored for different stakeholders: the State Veterinary Service

(SVE), technical managers (RTs), agricultural laboratories, and the Ministry of Agriculture (MAPA).

On the back-end, the architecture differentiates between two distinct types of REST APIs. The business APIs manage the core logic and processes associated with the platform's regulatory functions, ensuring that workflows and data management align with specific legal and procedural requirements. In contrast, the service APIs provide more generic functionality, supporting integration and interoperability with the business APIs by delivering reusable services across the platform.

This study introduces a dedicated service API to integrate the platform with a RAG System, facilitating retrieval and synthesis of regulatory knowledge, as detailed in subsequent sections.

3 METHODOLOGY

This section outlines the core components of our RAG System architecture, including the setup of the RAG Module and the tool used to run our LLM.

3.1 RAG Module

The RAG Module is a structured system designed to ingest, store, and retrieve documents content based on semantic similarity. It primarily functions to improve response accuracy by retrieving relevant document segments that align with user queries, utilizing vector embeddings for efficient similarity-based search to later be fed into the LLM.

3.1.1 Document Ingestion

The ingestion process prepares documents for storage in the vector database. This involves multiple steps:

- **Text Extraction.** Document content is extracted from various file types (PDFs, text files, etc.), preparing it for chunking and processing.
- **Chunking and Tokenization.** Documents are divided into smaller sections, such as sentences or paragraphs. Chunking improves retrieval precision and allows targeted access to specific information. Tokenization follows, breaking down text into discrete tokens for embedding..
- **Embedding Creation.** Each chunk is transformed into a vector embedding using a pre-trained embedding model, which captures the semantic content of the text. The embedding is a high-dimensional numerical representation of the chunk, enabling similarity-based retrieval within the vector database.

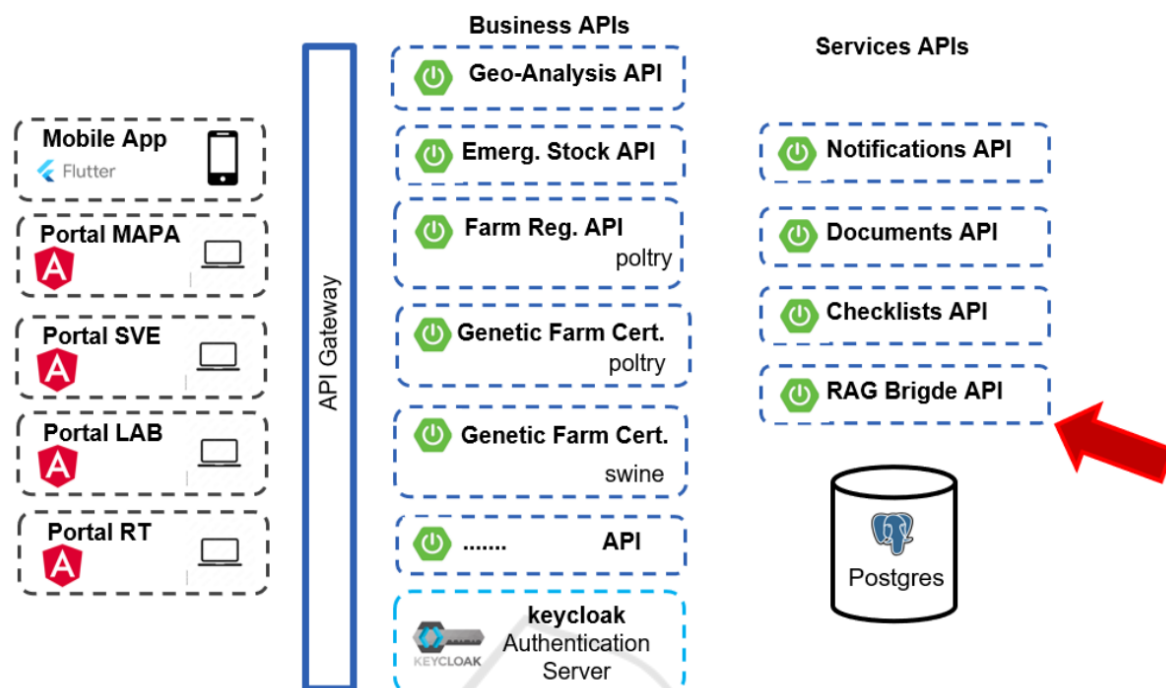


Figure 2: Overview of the PDSA-RS Architecture.

We choose to use the LlamaIndex data framework for this stage, leveraging the Python package llama-index. This package provides ready-to-use functions for document ingestion, which simplifies the process of embedding documents, indexing them, and storing them in a vector database for efficient retrieval.

3.1.2 Vector Storage

The RAG module utilizes a vector database to store embeddings. This type of storage allows efficient retrieval through similarity search, where embeddings with high semantic similarity to a query are located based on distance metrics. PostgreSQL was selected as the database solution, using the PGVector extension to store embeddings. PGVector is a PostgreSQL extension that provides efficient support for vector-based data storage, making it well-suited for handling the embeddings generated by the LlamaIndex framework.

3.1.3 Retrieval Process

The RAG module's retrieval component is used when a user submits a query. The query is converted into an embedding and used to locate relevant document chunks stored in the vector database:

- **Query Pre-Processing.** Before retrieval, the query may undergo filtering and rephrasing to enhance search relevance. Steps may include

expanding acronyms, removing stop words, and restructuring complex queries into simpler sub-queries. These adjustments help the RAG module better interpret the query's intent and improve retrieval accuracy.

- **Similarity Search.** Using the query embedding, a similarity search is conducted in the vector database. This search identifies chunks that closely align with the query's semantic content. Cosine similarity or other distance-based metrics are applied to find the most relevant chunks, ensuring that only document sections pertinent to the query are retrieved.

We are again utilizing LlamaIndex for the retrieval process, taking advantage of its built-in search functions. LlamaIndex provides efficient methods for performing semantic searches over document embeddings stored in the vector database. These search functions allow us to find the most relevant document chunks based on the similarity between the query embedding and the stored document embeddings.

3.1.4 Maintenance and Updates

To keep information relevant and accurate, the RAG module may undergo periodic updates:

- **Embedding Updates.** As new documents become available, they are embedded and stored in

the vector database to expand the scope of retrievable content.

- **Maintenance of Vector Store.** Regular maintenance helps optimize performance and relevance. Outdated information can be removed, and frequently accessed chunks may be optimized for faster retrieval, ensuring that the vector store remains responsive and accurate.

3.2 Large Language Model Setup

The LLM selected for use in our RAG system is the Llama 3.1 8B Instruct model. This model, part of Meta's LLaMA family, is a fine-tuned model from the base Llama 3.1 8B, designed for instruction-following tasks. We preferred the 8B version as it has a balance between computational efficiency and performance, offering a strong capacity for understanding and generating responses while also maintaining an efficient power consumption.

3.2.1 Model Selection and Configuration

To run our LLM, we have chosen to use privateGPT in conjunction with the Ollama framework. This combination allows us to maintain complete control over the model and data, ensuring privacy and security.

The privateGPT is a solution designed to run models locally, ensuring that the data and queries remain private without needing to send sensitive information to external servers. It is useful when working with confidential or proprietary data, like legal or medical information. In this setup, privateGPT operates as the interface to manage the LLM, which is hosted on a private server, offering full control over the model.

To run the model locally, privateGPT offers two frameworks: llama.cpp and Ollama. We chose Ollama based on the privateGPT documentation, which recommends it for its greater versatility in running across different computational environments. This framework is designed to simplify model deployment, providing an efficient and flexible setup that can be easily adapted to various hardware configurations.

3.2.2 API

To allow external services to interact with our system, we chose to use an API. PrivateGPT already includes an API as part of its structure, making it easy to integrate our system with external services. This API provides all the necessary endpoints for document ingestion, query handling, and response generation, abstracting the complexities of the RAG pipeline while allowing external systems to make requests and receive responses in a standardized way.

This API is divided into two logical blocks:

- **High-Level API.** Abstracts the complexities of the RAG pipeline by automating document processing and response generation. It manages tasks such as document parsing, splitting, metadata extraction, embedding generation, and storage, preparing documents for efficient retrieval. Additionally, the API handles chat and completion processes by retrieving relevant content, engineering prompts, and generating responses, allowing users to focus on querying the system and receiving contextually relevant answers based on the ingested documents, without needing to manage the intricate details of retrieval and generation.
- **Low-Level API.** Provides advanced users with the ability to generate embeddings for any given piece of text. It also includes an endpoint for contextual chunks retrieval, which, when given a query, searches the ingested documents and returns the most relevant text chunks.

3.3 Architecture Overview

Our architecture can be summarized as shown in figure 3. The system is composed of the RAG Module, which is responsible for the entire process of ingestion, storage, and retrieval of context, and the LLM, which processes user queries. The RAG module handles document parsing, embedding generation, and storing the relevant context in a vector database for efficient retrieval. The LLM, once the context is retrieved, generates responses based on the processed queries, ensuring the responses are contextually relevant and informative.

4 CASE STUDY

To evaluate the effectiveness of our RAG system on the PDSA-RS platform, we conducted a case study to integrate the RAG system within the platform and assess its performance in handling regulatory questions related to poultry certification.

In order for an establishment to obtain a sanitary certificate they must submit samples of birds, poultry products (such as eggs), among other materials for laboratory testing in institutions accredited by the MAPA. The purpose of these collections is to ensure that the batches are free from pathogenic agents such as Salmonella or other relevant infectious agents. For each age group of birds, there are different rules regarding the quantity of materials to be collected, the

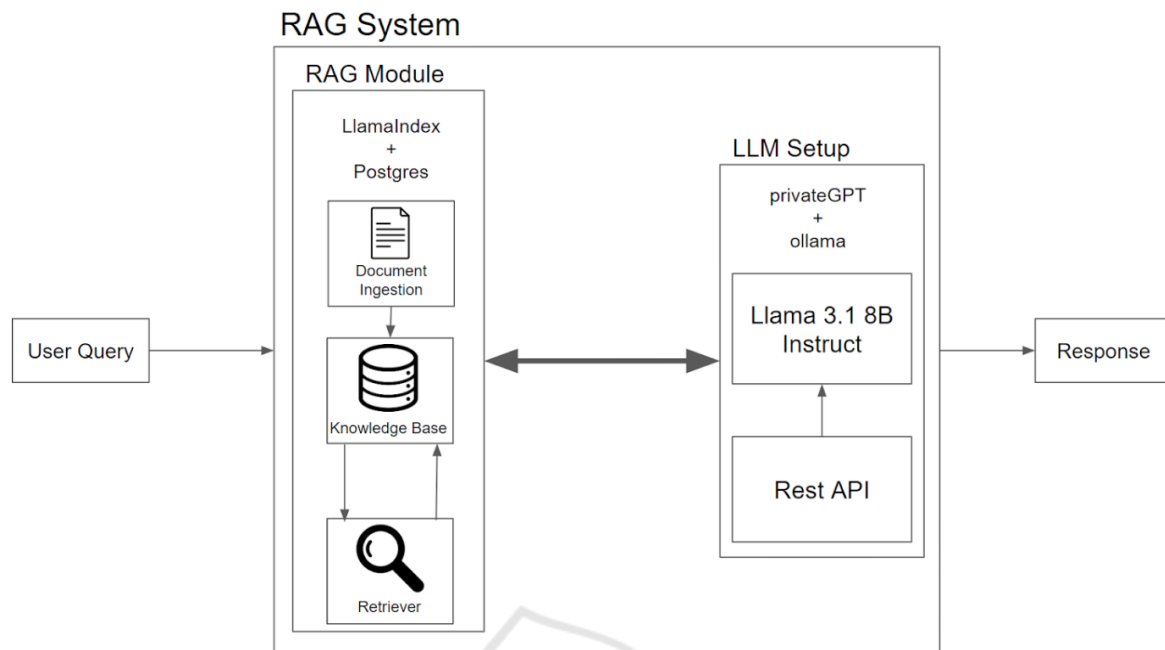


Figure 3: Overview of the RAG System Architecture.

types that must be collected, and the combinations of each material. Furthermore, the purpose of production for these birds must also be considered, as it influences the aforementioned parameters. We use these parameters to define the query that will be sent to the retrieval module to search for context in our knowledge base.

For the documents used in the analyses performed by the RAG system, we are utilizing IN 78/2003 (Ministério da Agricultura, 2003) and IN 44/2001 (Ministério da Agricultura, 2001). These documents outline the technical standards for disease control and certification of poultry establishments free or controlled from diseases like Salmonella and Mycoplasmosis.

4.1 Query Structure

We analyzed an object representing a completed certification request, which includes data on diseases monitored, sanitary conditions and laboratory testings. In this study, the focus was verifying whether all the laboratory testing for monitored diseases met the required standards.

The data structure from a sample certification object includes:

- **Nucleus Details.** Information about the facility, such as active status, purpose of production (e.g., Hatchery), and other relevant parameters.

- **Laboratory Exams.** The list of tested diseases, including Pullorum (SP), Gallinarum (SG), and others, containing dates of each test and also materials used for testing.
- **Sanitary Conditions.** Each disease status (e.g., Free or Vaccinated).

Our system was tasked to analyze the object, particularly the laboratory exams, to ensure that all elements were compliant with the required standards for certification.

As the context of our work is within a controlled environment, we have predictability of which data we want to feed into our LLM, thus we can use pre-established queries for the context where the module will be used. In the example of the health certification process for poultry establishments, we make choices according to the rules of each stage of the life cycle of a poultry establishment.

We did some experimentation and arrived at the following system prompt, which will be used to provide background instructions on how the model should answer. This prompt showed little incidence of hallucinations in the tests carried out:

"You must act as an expert in poultry farming and certification. Answer questions accurately about poultry certification and biosecurity. If unsure, indicate that the question cannot be answered."

For the condensed prompt we followed the structure below:


```

{{ Retrieved Context }}
Use the content above to respond to the query
below if relevant, or respond to the best of
your ability without it.
{{ Poultry Certification Request Data }}

```

The condensed prompt is designed aiming to boost contextual awareness and foster model behavior that prioritizes accuracy by only generating responses aligned with the provided information.

As shown in table 1, for the parameters of our LLM, we set the temperature value to 0.8. With this setting we were able to balance the model's ability to explore different possibilities while avoiding trivial or overly repetitive answers. Temperature is a hyperparameter that controls the randomness or softness of output probabilities in LLMs, allowing for more diverse or exploratory language generation rather than deterministic predictions. Additionally, we employed an adaptive context window size with a maximum capacity of 1024 tokens, accommodating a wide range of input lengths while ensuring efficient memory use. For output search strategies, we implemented a beam search with a width of 16, enabling the model to evaluate multiple possible answers and select the best output path based on probability scores.

Resource allocation was also a focus, with GPU memory dynamically set at 8GB to allow efficient processing on available hardware. CPU utilization was capped at 80%, with a maximum of 16 threads, mostly because of our limitation with the test hardware, to ensure that the model could make full use of our CPU power.

Table 1: LLM Hyperparameter Configuration.

Hyperparameter	Value
Temperature	0.8
Context Window Size	Adaptive (max 1024 tokens)
Beam Search Width	16
GPU Memory Allocation	8GB (dynamic allocation)
CPU Thread Utilization	80% (max 32 threads)

We encountered significant challenges in running inferences on our test hardware setup, which consists of a Ryzen 7 5800X3D CPU and a Radeon RX 7700XT 12GB GPU. Specifically, we had to carefully manage GPU memory allocation to prevent excessive memory usage, as the RX 7700XT's 12GB VRAM is limited compared to more powerful professional-grade GPUs.

The integration of the RAG System with PDSA-RS is done within one of the platform's several microservices, more specifically the service responsible for the poultry certification process, called Aves API. Within this API, an endpoint is responsible for making the call to the RAG system for chat completion,

containing all the information that is extracted from PDSA-RS according to the user's request. As shown in figure 4, this call is made through a button found on the front-end of the platform, within the component where certificate requests are analyzed.

4.2 Results

We then conducted tests using certification request data to evaluate the system's performance. For evaluation metrics, we used response accuracy and generation time. To measure the accuracy of the RAG system, we extracted real data from certifications that were produced by specialists that use the PDSA-RS. We used the condensed prompt described in section 4.1 to each evaluation step, baseline and RAG.

We take the laboratory exams and create a query to ask the LLM if, based on these exams, all the required materials were collected on the correct dates, according to the purpose of production for each stage. Based on this analysis, we ask the LLM to determine the sanitary condition of the farm for the tested disease and if it would be eligible for certification. We compare the results with the sanitary conditions that were made by a specialist to verify if the generation by the LLM was correct.

We categorized responses into two types:

- **Fully Correct Responses (Rc):** The model correctly identified compliance or non-compliance for all nuclei.
- **Partially Correct Responses (Rp):** The model provided partially accurate responses, missing or misinterpreting some details.

The formula used to calculate accuracy was:

$$\text{Accuracy} = \left(\frac{Rc + (W \times Rp)}{N} \right) \times 100$$

Where:

- N = Total queries in the benchmark
- W = Weight for partially correct responses (set to 0.5 in our case)

For comparison, we generated responses in two modes: with RAG, using the IN documents incorporated into our knowledge base, and without RAG, using only the base model. This approach allowed us to assess the impact of the RAG module on response quality and processing efficiency.

For the baseline model, which does not use the RAG system, the evaluation metrics were as follows:

- $N = 100$: Total queries related to certification requests.

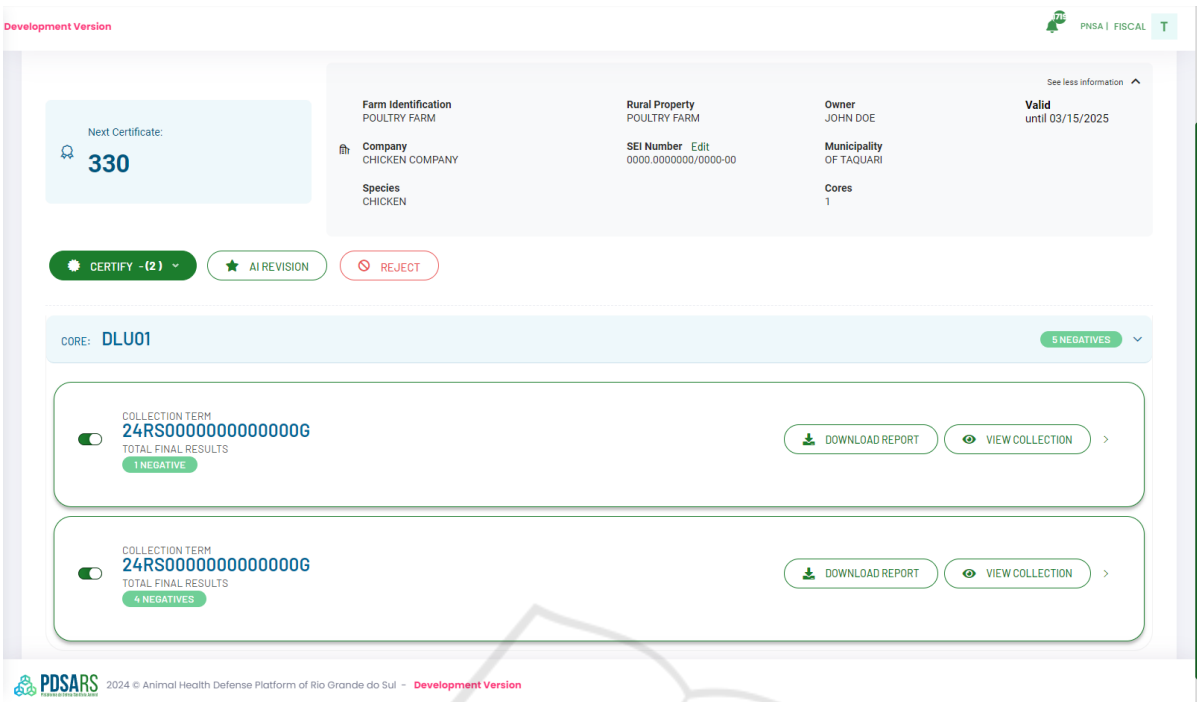


Figure 4: Screenshot of the poultry certification analysis from the PDSA-RS.

- $R_c = 2$: Fully aligned responses.
- $R_p = 20$: Partially correct responses.

Out of 100, there were 78 completely incorrect responses that were not accounted and therefore do not appear in the above parameters.

Applying the formula for the baseline:

$$\text{Accuracy} = \left(\frac{2 + (0.5 \times 20)}{100} \right) \times 100 = 12\%$$

For the RAG-enhanced system, the evaluation metrics were:

- $N = 100$: Total queries related to certification requests.
- $R_c = 40$: Fully aligned responses.
- $R_p = 30$: Partially correct responses.

Out of 100, there were 30 completely incorrect responses that were not accounted and therefore do not appear in the above parameters.

Applying the formula for the RAG-enhanced system:

$$\text{Accuracy} = \left(\frac{40 + (0.5 \times 30)}{100} \right) \times 100 = 55\%$$

As shown in table 2, baseline generation showed a low accuracy of 12%, which is a result of the LLM's

Table 2: Model Performance.

Phase	Response Accuracy	Generation Time (seconds)
Baseline	12%	5.2
RAG	55%	8.3

limited capacity to contextualize and apply domain-specific knowledge regarding poultry certification and internal monitoring standards.

In contrast, the RAG demonstrated a significant improvement in accuracy. The model's ability to retrieve relevant legal texts allowed it to generate responses that were more informed and contextually accurate.

The generation time is slightly longer when using RAG, as content retrieval is required before the response generation process can begin. This additional retrieval step, which involves searching for and loading relevant context from the knowledge base, naturally extends the time needed to generate each response. However, this trade-off enhances response accuracy and relevance by incorporating contextual information.

The results indicate that the integration of a RAG system substantially improves response accuracy, showing an accuracy increase of 43% when using the RAG approach.

The findings from this case study suggest that integrating Retrieval-Augmented Generation (RAG) into a platform like PDSA-RS offers a practical solution for enhancing the performance of LLMs in domain-

specific tasks as well as assisting and accelerating the process of analyzing the required standards necessary for poultry certification.

It is important to emphasize that the objective of this case study was not to replace human expertise in the poultry certification process. Rather, the focus was to assess how a LLM, integrated with a Retrieval-Augmented Generation (RAG) system, could assist in streamlining certain tasks by providing relevant legal information and context. While LLMs can offer valuable support in processing complex data and retrieving domain-specific knowledge, they still present limitations, such as the potential for generating inaccurate or incomplete responses. Therefore, human oversight remains crucial to ensure the reliability and precision of the certification process, particularly in areas where nuanced judgment and expert knowledge are required.

5 CONCLUSION

The integration of a Retrieval-Augmented Generation (RAG) system into the PDSA-RS platform has proven to be a valuable advancement in improving the accuracy and contextual relevance of large language models (LLMs) within the specialized domain of animal health regulation. By leveraging domain-specific retrieval to supplement the generative capabilities of LLMs, our system bridges the gap between general-purpose language models and the unique, nuanced needs of regulatory environments.

Our case study highlights the potential for RAG systems to streamline and enhance the way legal and regulatory queries are handled, especially in complex sectors like veterinary health. The performance gains observed through the incorporation of relevant regulatory texts into the LLM's output underscore the value of domain-adapted retrieval processes in increasing both the precision and usefulness of the generated responses. This approach demonstrates that RAG not only improves response quality but also provides a scalable solution to address domain-specific challenges, where accuracy and legal compliance are paramount.

Looking ahead, the continuous refinement of both the retrieval module and the underlying language model will be critical in ensuring that the PDSA-RS platform remains capable of delivering accurate and timely legal guidance.

From a theoretical perspective, this study emphasizes the adaptability of RAG in dynamic environments and its ability to overcome knowledge obsolescence—a limitation inherent to fine-tuned models.

Furthermore, the findings contribute to the broader discourse on hybrid AI systems, where generative and retrieval capabilities are combined to enhance domain-specific applications.

For future work, there is the potential to fine-tune the model with specific documents, which could further enhance the relevance and accuracy of content generation on this subject. This additional fine-tuning would allow the model to better handle complex, domain-specific queries, delivering more precise responses tailored to the chosen topic. Additionally, developing a graphical user interface (GUI) could significantly improve user interaction by allowing users to directly input various documents and interact with them. This interface would enable users to upload documents, ask questions, and receive contextually relevant responses, making the system more intuitive and accessible for end-users engaging with diverse content.

In conclusion, while the RAG-enhanced LLM system offers significant benefits, it is essential to maintain human oversight, particularly in legal and regulatory contexts where misinterpretation of guidelines could have serious consequences. The role of the LLM is not to replace human expertise but to augment decision-making by providing informed, contextually relevant suggestions. This hybrid approach—leveraging both cutting-edge AI and human expertise—represents a promising path forward for regulatory platforms like PDSA-RS, fostering innovation while ensuring the accuracy and integrity of the certification processes that underpin Brazil's animal health sector.

ACKNOWLEDGEMENT

This research is supported by FUNDESA, project “Combining Process Mapping and Improvement with BPM and the Application of Data Analytics in the Context of Animal Health Defense and Inspection of Animal-Origin Products in the State of RS” (UFSM/060496) and by MPA (Ministério da Pesca e Aquicultura), project “Use of Artificial Intelligence in the Systematization of Hygienic-Sanitary Certification Processes for Vessels and Accreditation of Legal Origin of Fish” (UFSM/060642). The research by Vinícius Maran is partially supported by CNPq grant 306356/2020-1 DT-2, CNPq PIBIC and PIBIT program and FAPERGS PROBIC program.

REFERENCES

- Bishop, C. (2006). *Pattern Recognition and Machine Learning*, volume 16, pages 140–155.
- Bommasani, R., Hudson, D., Adeli, E., Altman, R., Arora, S., Arx, S., Bernstein, M., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Creel, K., Davis, J., Demszky, D., and Liang, P. (2021). On the opportunities and risks of foundation models.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., and Dhariwal, e. a. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc. pp. 33-34.
- Descovi, G., Maran, V., Ebling, D., and Machado, A. (2021). Towards a blockchain architecture for animal sanitary control. In *Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 1: ICEIS*, pages 305–312. INSTICC, SciTePress.
- Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2024). Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.
- Ebling, D., Machado, F., Descovi, G., Cardenas, N., Machado, G., Maran, V., and Machado, A. (2024). A distributed processing architecture for disease spread analysis in the pdsa-rs platform. In *Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 2: ICEIS*, pages 313–320. INSTICC, SciTePress.
- Esteva, A., Kuprel, B., Novoa, R., Ko, J., Swetter, S., Blau, H., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542.
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., Wang, M., and Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997.
- Google Cloud (2023). A three-step design pattern for specializing llms. Accessed: 2024-09-13.
- Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- Izacard, G. and Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.
- Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521:436–44.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.
- Ministério da Agricultura, Pecuária e Abastecimento, S. d. D. A. (2001). Instrução normativa nº 44, de 23 de agosto de 2001. Accessed: 2024-10-23.
- Ministério da Agricultura, Pecuária e Abastecimento, S. d. D. A. (2003). Instrução normativa nº 78, de 3 de novembro de 2003. Accessed: 2024-10-23.
- Radford, A. and Narasimhan, K. (2018). Improving language understanding by generative pre-training.
- Sari, Y. and Indrabudiman, A. (2024). The role of artificial intelligence (ai) in financial risk management. *Formosa Journal of Sustainable Research*, 3:2073–2082.
- Schneider, R., Machado, F., Trois, C., Descovi, G., Maran, V., and Machado, A. (2024). Speeding up the simulation animals diseases spread: A study case on r and python performance in pdsa-rs platform. In *Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 2: ICEIS*, pages 651–658. INSTICC, SciTePress.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F., et al. (2023). Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.