information

MDPI

Article

Integrating RAG for Smarter Animal Certification Platforms

Pedro Bilar Montero [1], Jonas Bulegon Gassen [1], Glênio Descovi [1], Tais Oltramari Barnasque [2], Gabriel Rodrigues da Silva [1], Felipe Amadori Machado [1], Gabriel Vieira Casanova [1], Vinícius Maran [1] and Alencar Machado [1,*]

[1] Laboratory of Ubiquitous, Mobile and Applied Computing (LUMAC), Polytechnic School, Federal University of Santa Maria, Av. Roraima, 1000, Camobi, Santa Maria 97105-900, RS, Brazil; pedro.montero@acad.ufsm.br (P.B.M.); jonas.gassen@ufsm.br (J.B.G.); glenio.descovi@politecnico.ufsm.br (G.D.); gabriel-silva.3@acad.ufsm.br (G.R.d.S.); famachado@inf.ufsm.br (F.A.M.); gabriel.casanova@acad.ufsm.br (G.V.C.); vinicius.maran@ufsm.br (V.M.)
[2] Ministry of Agriculture, Livestock and Food Supply (MAPA), Brasília 70043-900, DF, Brazil; tais.barnasque@agro.gov.br
* Correspondence: alencar.machado@ufsm.br

Abstract

Large Language Models (LLMs) encounter significant challenges when applied in specialized domains that require precise and localized information. This problem is particularly critical in regulatory sectors, such as the animal health sector in Brazil, where professionals depend on complex and constantly updated legal norms to perform their work. The generic knowledge encapsulated in traditional LLMs is often insufficient to provide reliable support in these contexts, which can lead to inaccurate or outdated responses. To address this gap, this work presents a practical implementation of a Retrieval-Augmented Generation (RAG) system. We detail the integration of this system with the Plataforma de Defesa Sanitária Animal do Rio Grande do Sul (PDSA-RS), a real platform used for animal production certification. Our solution connects an LLM to an external knowledge base containing specific Brazilian legislation, allowing the model to retrieve relevant legal texts in real time to generate its responses. The principal objective is to demonstrate how this approach can produce accurate and contextually grounded answers for professionals in the veterinary field, assisting in decision-making processes for sanitary certification.

Keywords: Retrieval-Augmented Generation; Large Language Models; poultry health; sanitary certification; PDSA-RS; Brazilian regulations; legal texts; animal health; natural language processing

check for updates

Academic Editor: Katsuhide Fujita

Received: 25 August 2025
Revised: 18 September 2025
Accepted: 22 September 2025
Published: 30 September 2025

Citation: Montero, P.B.; Bulegon Gassen, J.; Descovi, G.; Barnasque, T.O.; da Silva, G.R.; Machado, F.A.; Casanova, G.V.; Maran, V.; Machado, A. Integrating RAG for Smarter Animal Certification Platforms. *Information* **2025**, *16*, 843. https://doi.org/10.3390/info16100843

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

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 [1]. However, while LLMs demonstrate broad generalization abilities, they often encounter challenges when applied to specialized domains, as these areas require extensive localized knowledge and a precise understanding that is not always encapsulated within large generic datasets [2,3].

In industry, the integration of AI into operational systems has led to significant advancements across sectors such as healthcare, finance, and law [1,3]. Within the domain of animal health, for example, LLMs hold the 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 [2].

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. The 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 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 [4,5].

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 [6,7].

In this study, we aim to implement an RAG system that can be integrated into the PDSA-RS, 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 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.

To this end, our study focuses on the practical application of this approach. We present a documented integration of an RAG system into a Brazilian governmental platform, the PDSA-RS, specifically targeting the poultry health certification module. This work demonstrates the feasibility of using locally hosted Large Language Models (LLMs) to navigate complex national regulations, thereby ensuring both regulatory compliance and data privacy. The primary objective is to develop and evaluate an AI-powered assistant capable of providing veterinary professionals with accurate, contextually grounded, and verifiable answers to support their decision-making in the certification process.

Our paper is organized as follows: Section 2 covers the background of our research, showing everything needed for this paper. In Section 3, we present our methodology and describe how we organized our architecture and developed the RAG system. Section 4 is our case study of the implementation of the RAG system within the PDSA-RS. In Section 5, we discuss the findings, implications, limitations, and future directions for our work. Finally, in Section 6, we provide the conclusions of this work. A glossary of key terms and acronyms used throughout this paper is provided in Appendix B.

2. Background

2.1. Artificial Intelligence (AI)

Artificial intelligence (AI) is a field of computer science dedicated to creating systems that can perform tasks normally requiring human intelligence [8]. Historically, AI research evolved from symbolic, logic-based systems to data-driven approaches, a paradigm shift largely propelled by the rise of Machine Learning (ML). ML algorithms learn to identify patterns and make predictions from data without being explicitly programmed for the task [9,10].

A subfield of ML, Deep Learning (DL), has been particularly transformative. DL utilizes deep neural networks—architectures with multiple layers of interconnected nodes—to learn hierarchical representations directly from raw data [11]. This layered approach allows models to capture complex, high-level abstractions, leading to breakthroughs across various domains. For instance, deep neural networks have achieved expert-level performance in tasks ranging from image recognition and medical diagnostics to financial modeling [12–14].

The ability of these models to process not just static data but also sequential information, such as text and time series, has been a crucial development [15]. This progress in handling complex data structures has been fundamental to the growth of many specialized AI fields. Among the most prominent of these is natural language processing (NLP), which focuses on the interaction between computers and human language and has been revolutionized by these Deep Learning advancements.

2.2. Natural Language Processing and Language Models

Natural language processing (NLP) is a specialized subfield of AI focused on enabling computers to understand, interpret, and generate human language. The core of NLP is the language model, a computational model that assigns probabilities to sequences of words. Early language models relied on statistical methods like n-grams, which predict the next word based on a small fixed window of preceding words. While foundational, these models struggled to capture long-range dependencies and complex linguistic structures.

The advent of neural networks brought significant progress. Recurrent Neural Networks (RNNs) and their more sophisticated variant, Long Short-Term Memory (LSTM) networks, introduced the concept of a hidden state, allowing them to maintain a memory of past information and better model sequential data [15]. However, these architectures face challenges in processing very long sequences and are inherently sequential, limiting parallelization and training efficiency.

A revolutionary breakthrough came with the introduction of the Transformer architecture by Vaswani et al. [16]. The Transformer dispenses with recurrence entirely, instead relying on a self-attention mechanism that allows the model to weigh the importance of different words in the input sequence simultaneously. This parallelizable design enabled the training of much larger and deeper models.

2.3. Large Language Models

A Large Language Model (LLM) is a type of model that learns patterns in human language by reading large volumes of text. It is trained with a method called self-supervised learning, where the model tries to predict parts of the text based on the context. Once trained, it can be used to understand and generate human-like language. The development of LLMs became much more effective after the introduction of the Transformer architecture, proposed by Vaswani et al. [16], which brought a new way to deal with text data.

The main difference between the Transformer and older models like RNNs and LSTMs is that it does not process the text word by word in order. Instead, it works with the entire sentence at once, using what is called attention mechanisms [16]. As we can see in Figure 1, the Transformer is divided into two parts: an encoder and a decoder. The encoder takes in the input text and creates a representation of it that captures the meaning and context of the words. The decoder then uses this representation to generate a response or continue the text.

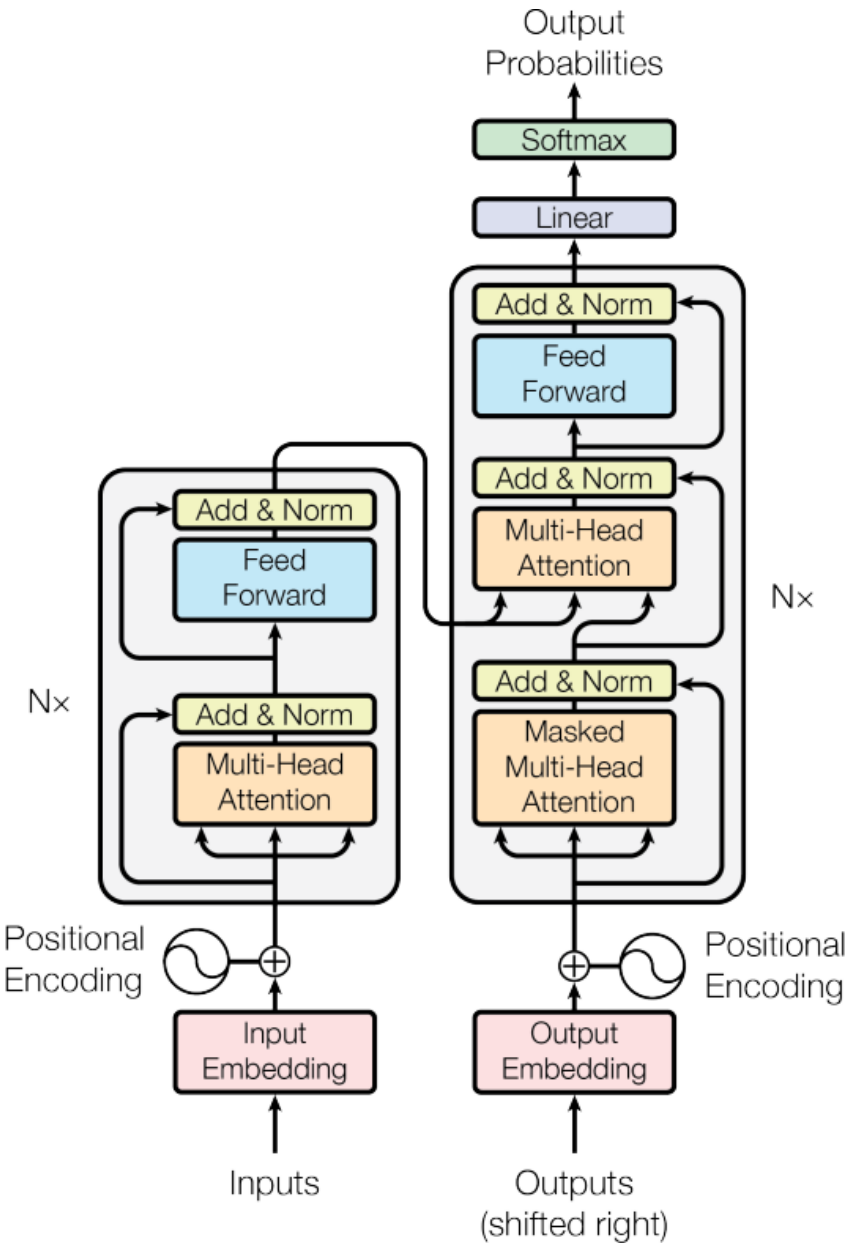


Figure 1. Overview of the Transformer architecture, as proposed by Vaswani et al. in 'Attention is All You Need' [16].

A key idea in this architecture is self-attention. This allows the model to figure out how each word in a sentence relates to the others [16]. For example, in the sentence "The animal didn't cross the street because it was too tired," the model needs to know that "it" refers to "the animal" and not "the street." The self-attention mechanism helps with that. The model also uses something called multi-head attention, which checks the relationships between words from different angles, helping it to understand the structure and meaning more deeply. Since the model does not read the text in order, it adds some extra information called positional encoding to keep track of word positions [16].

The training process of an LLM usually happens in two steps: First is pre-training, where the model reads a huge amount of text and learns how language works in general—how sentences are structured, how words relate, and so on. This step creates a very general model, sometimes called a foundation model. The second step is fine-tuning, where the model is trained on a smaller and more specific dataset for a particular task, like answering questions, summarizing texts, or translating. This makes the model more useful for real applications. OpenAI's GPT models are famous examples of this two-step approach [1,17,18].

At first, most high-performing models were closed-source, meaning only a few companies could use or modify them. But this started to change with open-source alternatives, like Meta's LLaMA (Large Language Model Meta AI) [19,20]. These models are freely available, making it easier for researchers and developers to experiment and adapt them to specific needs. This is especially useful when working in specialized areas where we need the model to understand a particular vocabulary or context [21].

2.4. Specialized Domains

Despite their impressive general capabilities, Large Language Models often fall short when deployed in specialized, high-stakes domains such as law, finance, healthcare, and regulated agriculture. Their performance in these fields is constrained by several fundamental factors that stem from the nature of their training and architecture. Simply scaling up the model size does not resolve these core issues, making the direct application of general-purpose LLMs in these contexts both impractical and risky. These challenges can be categorized as follows:

- **Knowledge Gaps and Hallucinations:** LLMs are trained on vast but general corpora like Common Crawl, which disproportionately represent conversational language over niche, technical literature. As a result, the model's understanding of domain-specific terminology, concepts, and reasoning patterns is often shallow or incomplete. This "knowledge gap" can lead to factual inaccuracies or, more dangerously, "hallucinations," where the model generates plausible-sounding but entirely fabricated information to fill the void in its knowledge [2,22]. For instance, it might invent a non-existent legal precedent or incorrectly describe a technical standard with high confidence.
- **Lack of Domain-Specific Nuance and Reasoning:** Specialized fields are characterized by complex rules, subtle exceptions, and context-dependent interpretations that are difficult to learn from a general corpus. A legal term like "liability" has a precise, multi-part definition that differs starkly from its colloquial usage. Similarly, a medical diagnosis relies on structured, causal reasoning, a known weakness of LLMs, which excel at identifying statistical correlations but not at understanding underlying mechanisms. This inability to grasp deep semantic nuance means models can easily misinterpret a query or provide a technically correct but contextually inappropriate response [23,24].
- **Static Knowledge and Information Obsolescence:** The knowledge embedded within a pre-trained LLM is static, frozen at the time of its last training run. This makes it inherently unreliable for dynamic domains where information evolves rapidly. Fields

like regulatory compliance, medical research, and financial markets require access to real-time updates. An LLM trained in early 2024 would be oblivious to new legislation passed later that year, making its advice not just outdated but potentially illegal [1]. This "knowledge cutoff" is a critical vulnerability that prevents the model from being a trustworthy source for current information.

- **High Cost of Errors and Lack of Verifiability:** In consumer chatbot applications, an error is often a trivial inconvenience. In specialized domains, the consequences can be severe; a misinterpreted compliance rule can lead to financial penalties, a flawed engineering specification can cause catastrophic failure, and an incorrect medical suggestion can endanger lives. These high-stakes environments demand an exceptional level of accuracy and, crucially, verifiability—the ability for a user to trace an answer back to its source document. Standard LLMs act as "black boxes," making it impossible to verify the origin or factual basis of their generated output.

To address these significant limitations, researchers have pursued two primary strategies: The first involves creating domain-specific LLMs by training them from scratch or continuing pre-training on large curated datasets from the target field. Models like BloombergGPT for finance [25] and GatorTron for clinical medicine [26] are powerful examples. While effective, this approach is extremely resource-intensive, requiring massive computational power and extensive data curation.

A more flexible and efficient alternative is the use of hybrid architectures like Retrieval-Augmented Generation (RAG) [6,7]. RAG enhances a general-purpose LLM by dynamically retrieving relevant information from an external, domain-specific knowledge base at the time of inference. This approach allows the model to ground its responses in current, factual, and contextually appropriate information without the need for constant re-training, making it particularly suitable for domains with evolving knowledge and a high requirement for verifiability.

2.5. Retrieval-Augmented Generation (RAG) Architecture

Retrieval-Augmented Generation (RAG) is an architecture that combines language models with external information retrieval to improve the relevance and precision of generated responses [6]. This approach is especially useful when the language model needs to provide information that may not be present in its training data or that changes frequently, such as legal norms, technical documentation, or information related to current events.

Traditional language models rely only on the knowledge they learn during pre-training, which is fixed and can become outdated. In contrast, RAG introduces a retrieval mechanism that allows the model to query external sources (usually a set of documents indexed in a vector database) at the time of inference. The general idea is to enable the model to "look up" relevant pieces of information when answering a user's question, instead of depending solely on what it has memorized.

The RAG workflow typically involves two main components: a retriever and a generator. The retriever is responsible for identifying documents that are semantically similar to the user query. These documents are stored in a vector database, where each document or chunk of text is pre-processed and encoded into a high-dimensional vector using a dense embedding model (often based on Transformers, like Sentence-BERT or similar) [7]. When a query is received, it is also converted into an embedding and compared against the database using similarity metrics (like cosine similarity). The top-k most similar chunks are returned to the generation component.

The generator, usually a pre-trained language model like LLaMA or GPT, receives both the original query and the retrieved documents as input. It then generates a response conditioned on this additional context. This allows the model to produce more accurate, up-

to-date, and context-aware answers, especially in cases where domain-specific knowledge is needed [27].

Figure 2 outlines the RAG workflow in four steps:

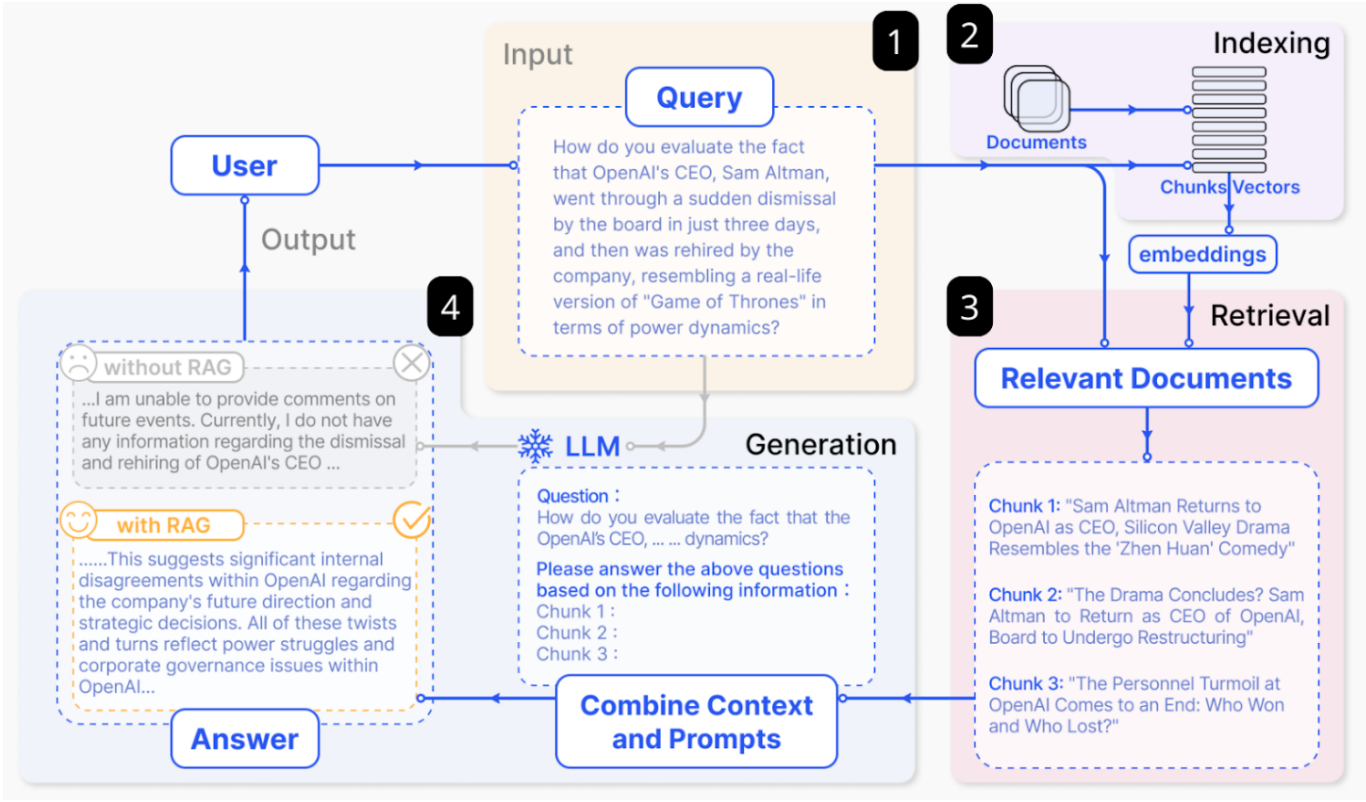


Figure 2. Illustration of the RAG architecture: information retrieval and generation processes. Adapted from [28].

- **1:** The user sends a query—for example, a legal question or a request for information about a specific topic.
- **2:** The query is transformed into an embedding and matched against a vector database containing pre-encoded chunks of documents.
- **3:** The retriever selects the most relevant chunks based on semantic similarity.
- **4:** These chunks are passed along with the query to the language model, which generates a response based on both the query and the retrieved content.

A practical advantage of this setup is that the retrieval step allows access to updated or domain-specific information without having to re-train the model. This reduces computational costs and makes the system more flexible, particularly in domains where the information changes often, like law or health regulation.

In addition, RAG architectures can be improved further with techniques such as query rewriting, document reranking, and feedback loops to refine results. There are also variations that use multi-stage retrieval or hybrid search (dense + keyword-based) to enhance performance in more complex scenarios [29,30].

To summarize, RAG offers a way to complement the static knowledge of Large Language Models with dynamic external information. This makes it a useful architecture for applications that require updated, trustworthy, and domain-sensitive responses—especially in environments where decisions depend on accurate and verifiable data.

2.6. 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.

As Section 2.3 outlines, RAG serves 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. The real-time adaptability of retrieval ensures that responses remain accurate in evolving areas such as legal or regulatory environments.

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 [31], yet it still requires periodic re-training 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 [6].

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

- **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 a potential alternative benchmark for evaluating domain-specific effectiveness.

2.7. 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 the 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 responses to health risks. The 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 3, 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).

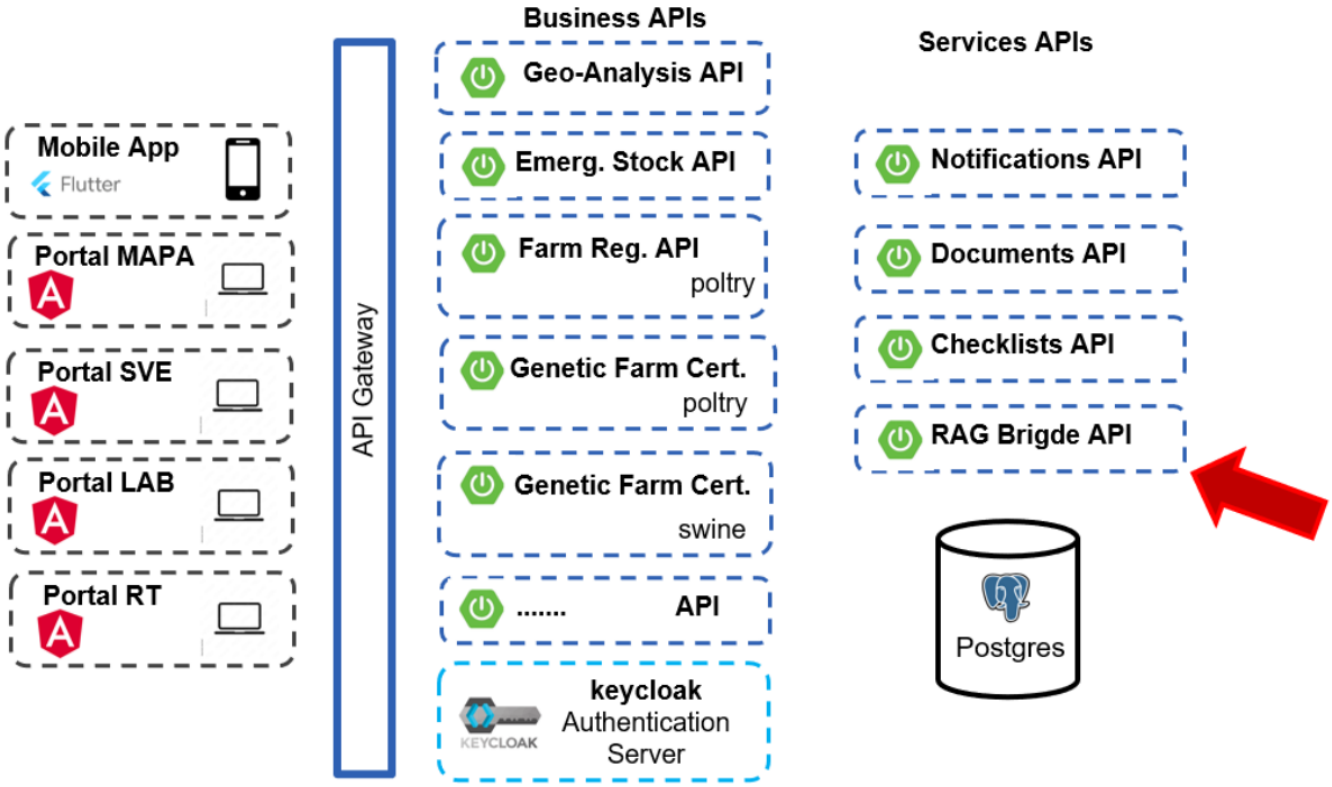


Figure 3. Overview of the PDSA-RS architecture.

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. Materials and Methods

This section presents the main components of the system developed for the RAG architecture. It includes details about the design and implementation of the RAG Module, the integration with the LLM, and how the overall system operates through an API interface.

3.1. RAG Module

The RAG (Retrieval-Augmented Generation) Module is responsible for organizing the document-processing pipeline, from ingestion to semantic retrieval. The goal is to enable the system to answer user queries using contextually relevant information retrieved from a structured knowledge base. The implementation relies on the LlamaIndex (v0.10.17) framework to simplify the ingestion and retrieval processes and uses PGVector (v0.5.1) for vector storage within PostgreSQL 15.

The ingestion process starts with the extraction of textual content from documents, such as PDFs and plain-text files. Once the raw text is available, it is segmented using a recursive character-splitting strategy. This method attempts to split text along semantically meaningful boundaries—such as paragraphs and sentences—to keep related content to-

gether. We configured a chunk size of 1024 characters with an overlap of 200 characters to ensure contextual continuity between adjacent chunks. After segmentation, the text is tokenized and transformed into vector embeddings using the bge-large-en-v1.5 model, which is optimized for high-performance semantic retrieval. These embeddings represent the semantic content of each chunk in a numerical format.

To store and manage the embeddings, we use PostgreSQL with the PGVector extension. This choice was a strategic decision based on several technical advantages: First, it allows us to unify our operational and vector data storage within a single database, which simplifies the system architecture and reduces maintenance overhead. Second, this integrated approach enables hybrid queries that combine semantic (vector) searches with traditional metadata filters using standard SQL. This means we can retrieve semantically relevant text chunks that also meet specific criteria, such as document source or creation date criteria, within a single operation. Finally, by building upon PostgreSQL, we leverage the maturity, scalability, and robust security features of a world-class DBMS, ensuring a reliable and enterprise-grade foundation for our vector database.

When a user sends a query, the system follows a retrieval process: The query is first embedded using the same bge-large-en-v1.5 model used for document embeddings. This embedding is then compared to the stored vectors in the database to find the most semantically similar chunks. These retrieved chunks are later used as context for the language model to generate more informed responses.

We also implement several query pre-processing steps to improve retrieval accuracy. A key step is acronym expansion, which is implemented using a custom dictionary lookup. This dictionary maps common acronyms found in the documents to their full-text descriptions. For example, a user query containing "What is the policy on LLMs?" would be transformed into "What is the policy on Large Language Models?" before being embedded. This normalization ensures that the query's semantic representation aligns more closely with the document content, which may use the expanded form. Other steps include the filtering of stop words and restructuring complex questions to simplify the retrieval task.

The RAG Module is designed to be updatable. New documents can be ingested at any time, and the embeddings can be recalculated and added to the database. Additionally, we can remove outdated content or re-process documents to improve retrieval performance.

3.2. Large Language Model Setup

The LLM chosen for this system is the LLaMA 3.1 8B Instruct model, which is a fine-tuned variant of Meta's LLaMA 3.1 family. We selected the 8B version because it provides a reasonable balance between performance and computational cost, making it suitable for running on local infrastructure.

To deploy the model, we used the Ollama (v0.1.15) framework, which simplifies the management and execution of Large Language Models in local environments. Ollama allowed us to run the LLaMA 3.1 model with good performance across different hardware setups without requiring access to cloud services or external APIs. This is an important consideration, especially when working with sensitive data where privacy is a concern.

The language model is integrated into the system in a way that allows it to receive the retrieved context from the RAG Module and generate a response based on both the context and the original user query. This architecture helps reduce hallucinations and improves the factual accuracy of the generated responses, especially in domain-specific applications.

3.3. API and System Integration

The system includes an API layer built with ExpressJS to allow interaction with external services. This API exposes endpoints that handle document ingestion, query

processing, and response generation. It acts as the interface between the user-facing application and the internal components of the RAG architecture.

The API is divided into two parts: The high-level endpoints are responsible for automating the full pipeline: document upload, parsing, embedding, and storage. These endpoints also manage user queries by running the retrieval process and sending the relevant context to the LLM for response generation.

In addition, we provide low-level endpoints that offer more direct access to system functions. For example, users can manually generate embeddings for a specific piece of text or perform a similarity search to retrieve the most relevant chunks. This design provides flexibility for both typical use cases and advanced usage scenarios.

This modular API structure ensures that the system can be extended or integrated into larger platforms in the future. It also abstracts most of the complexity behind a simple interface, allowing developers to work with it without needing to understand all the internal details of the RAG pipeline.

3.4. Architecture Overview

Figure 4 shows the general architecture of the system. The pipeline starts with document ingestion, where raw text is processed and converted into vector embeddings. These embeddings are stored in a PostgreSQL database using PGVector. When a user submits a query, the system retrieves relevant context from the vector database using LlamaIndex's semantic search functions. The context is then passed to the LLaMA 3.1 model running via Ollama, which generates a response that is sent back through the API.

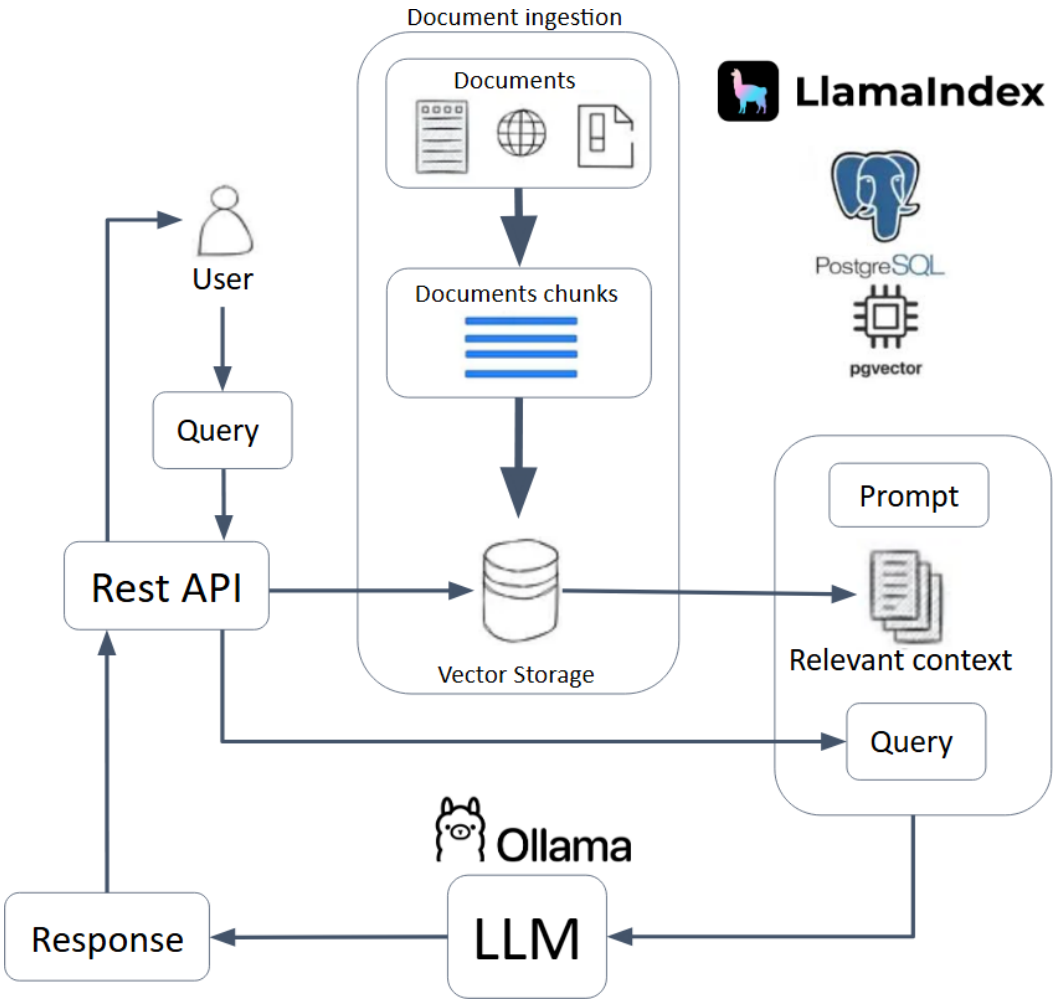


Figure 4. Overview of the RAG system architecture.

To illustrate this workflow, we consider the query "What is the required number of samples for Mycoplasma in 12-week-old birds?" Upon receiving this query via the API, LlamaIndex generates a vector embedding of the question. This vector is used to perform a cosine similarity search within the PGVector database to identify the most relevant document chunks. For this specific query, the search retrieves excerpts from regulation IN 44/2001 detailing sampling requirements. These retrieved text chunks, containing key phrases like "...for *Mycoplasma gallisepticum*: at least 300 samples..." and "...for *Mycoplasma synoviae*: at least 100 samples...", are then systematically formatted into a prompt. This context-augmented prompt is passed to the LLaMA 3.1 model, which synthesizes the provided facts to generate a precise answer, such as "For 12-week-old breeder birds, at least 300 samples are required for *Mycoplasma gallisepticum* and 100 samples for *Mycoplasma synoviae*." The final response is then returned to the user.

4. Results

To evaluate the effectiveness of our RAG system on the PDSA-RS, 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 and poultry products (such as eggs), among other materials, for laboratory testing at 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 types that must be collected, and the combinations of materials. Furthermore, the purpose of the production of these birds must also be considered, as it influences the aforementioned parameters. We use these parameters to define the query that is 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 utilize IN 78/2003 [32] and IN 44/2001 [33]. These documents outline the technical standards for disease control and certification of poultry establishments free from or controlled for diseases like Salmonella and Mycoplasmosis.

4.1. Data Curation and Knowledge Base Construction

A preliminary phase involves transforming the structured JSON data from the PDSA-RS into a format suitable for natural language processing.

The raw data, consisting of isolated attributes from the poultry farm certification process, are processed by a new internal feature in the poultry certification module. This feature generates a coherent, human-readable report summarizing the history of a specific certificate.

The transformation preserves the full context and traceability of the certification process. Each generated report includes the certificate's ID, issue/expiration dates, the veterinarian's name, farm details, and a breakdown of the sanitary status for each production unit (nucleus). This description details monitoring events chronologically, including sample collection dates, bird age, materials collected, laboratory information, test types (e.g., ELISA, RSA), and diagnostic results. Once generated, this natural language report, along with the foundational legal texts (IN 78/2003 and IN 44/2001), forms our domain-specific knowledge base. Both the legal documents and the case-specific reports are segmented into logical semantic chunks.

Each chunk is then passed through an embedding model and stored as a vector in our PostgreSQL database using the PGVector extension. This process creates a unified, searchable knowledge base where both general regulations and specific case details are co-located and indexed for efficient retrieval.

4.2. Query Structure and Retrieval Process

With the knowledge base constructed, the system is prepared to handle queries. The process leverages the core RAG architecture; instead of providing the entire report as context, the system retrieves only the most relevant information in response to a query.

Each query is tied to a specific certification validation task. For instance, a veterinarian might need to verify if a farm's monitoring activities comply with regulations for a specific disease. The query is structured to be precise, such as "Verify if the sample collection for Mycoplasma gallisepticum in nucleus NL01 complies with the standards for 16-week-old birds, based on the provided records."

Upon receiving the query, the system performs the following steps: The query is converted into an embedding vector using the same model that indexed the knowledge base.

A similarity search is executed on the PGVector database to find the top-k most relevant chunks. These chunks can come from the legal texts (detailing the official requirement) and the specific certification report (detailing what was actually carried out).

These retrieved chunks are compiled into a context block, which is then prepended to the original query in a structured prompt for the LLM. The final prompt structure is as follows:

Context:

Retrieved Chunks from Legal Docs and Certification Report

Question:

Original User Query

Instructions:

Based only on the context provided above, answer the question.

This structure ensures the LLM's response is grounded in the retrieved factual information, minimizing ambiguity and reducing the likelihood of hallucinations.

4.3. LLM Configuration

The quality and relevance of the model outputs in this case study depend not only on the prompt structure but also on the configuration of key model parameters. For this work, we adopt the LLaMA 3.1 8B Instruct model, running locally through the Ollama framework. Retrieval is performed using LlamaIndex, and vector embeddings are stored in a PostgreSQL database using the PGVector extension.

A brief sensitivity analysis was conducted to support the selection of each hyperparameter:

- **Temperature:** Set to 0.4.
  - Lower temperatures are generally preferred in tasks that require factual consistency and reduced variability.
  - We experimented with values of 0.2, 0.4, and 0.6.
  - At 0.2, responses were precise but overly rigid and often lacked fluency.
  - At 0.6, output became more natural but occasionally introduced minor inaccuracies or less formal wording.
  - Temperature 0.4 provided a good trade-off, generating outputs that were both accurate and clear while preserving the formal tone required in veterinary reports.
- **Context Window Size:** Limited to 1024 tokens.
  - Although relatively small, this limit was suitable due to the focused and preprocessed nature of the retrieved content.
  - A smaller context helped reduce inference latency during local execution.
- **Beam Search Width:** Set to 16.
  - Greedy decoding (width 1) often led to less complete answers.

  - Width 8 improved quality slightly, but width 16 consistently yielded better results for structured outputs such as certificate summaries.
- **Hardware Adaptation:**
  - We ran the system on a machine equipped with a AMD Ryzen 7 5800X3D CPU and a AMD Radeon RX 7700XT (12 GB) GPU (Santa Maria, RS, Brazil).
  - GPU memory allocation was handled dynamically, with a cap of 8 GB to avoid contention with other services.
  - CPU usage was limited to 80%, with a maximum of 16 threads, ensuring system responsiveness under concurrent load.

These settings, chosen to balance performance and accuracy for our specific hardware, are summarized in Table 1.

Table 1. LLM hyperparameter configuration.

Hyperparameter	Value
Temperature	0.4
Context Window Size	1024 tokens (max)
Beam Search Width	16
GPU Memory Allocation	8 GB (dynamic)
CPU Thread Utilization	80% (16 threads max)

The integration between the RAG system and the PDSA-RS occurs within the Aves API, one of the microservices responsible for handling poultry certification workflows. A dedicated endpoint in this API performs the request to the RAG system for generating chat completions. This endpoint packages all relevant information extracted from the PDSA-RS according to the user's query.

4.4. Experimental Evaluation

The evaluation was conducted using a benchmark dataset composed of 100 certification requests randomly sampled from the PDSA-RS. With just over 300 registered poultry genetics farms in the state of Rio Grande do Sul, this sample represents approximately one-third of the certification requests submitted, ensuring a high degree of representativeness. This sample size was chosen to provide a statistically meaningful yet manageable set of scenarios, encompassing both routine compliance checks and more complex edge cases involving ambiguous regulatory language or incomplete documentation. This approach ensured the evaluation reflects the practical challenges faced by regulatory analysts.

To move beyond a simple binary assessment, we adopted a three-tiered evaluation metric to provide a more granular view of the system's capabilities. Each of the 100 responses was analyzed by a domain expert, who classified them into one of the following categories:

- **Correct (C):** The model's conclusion perfectly matched the ground truth, correctly identifying the compliance status (e.g., compliant/non-compliant) and citing the specific evidence from the retrieved context to justify its reasoning.
- **Partially Correct (PC):** The response showed promise but was ultimately flawed. This category includes cases where the system retrieved the correct evidence but failed to synthesize it into the right conclusion, or where the reasoning was sound but based on incomplete or slightly irrelevant context.
- **Incorrect (I):** The response failed completely, providing the wrong conclusion based on faulty reasoning, hallucinated information, or irrelevant retrieved context.

This nuanced evaluation allowed us to distinguish between catastrophic failures and near misses, offering deeper insight into the system's primary challenges. For detailed examples of queries and responses corresponding to these categories, please see Appendix A.

4.4.1. Baseline Model Performance

The baseline model, which lacked access to our RAG system, struggled significantly. Out of 100 queries, it produced only 12 correct responses, resulting in a strict accuracy of 12%. The remaining 88 responses were incorrect, typically failing because the model resorted to generating plausible-sounding but factually inaccurate information based on its general training data. Without the specific legal texts or case data, it was fundamentally incapable of performing a valid compliance check.

4.4.2. RAG-Enhanced System Performance

The integration of the RAG system yielded a dramatic improvement. The system provided 55 fully correct responses, achieving a strict accuracy of 55%. While this figure may seem modest in absolute terms, it represents a transformative step in practical utility. A 55% success rate means that, for over half of the queries, analysts can trust the system's output completely, significantly reducing their manual workload and allowing them to focus their expertise on more ambiguous or complex scenarios. This result elevates the system from a simple search tool to a reliable assistant.

In addition to the correct responses, 20 responses (20%) were classified as "Partially Correct." These cases often demonstrated that the RAG system was on the right track but failed in the final reasoning step. The remaining 25 responses (25%) were fully incorrect. An analysis of these failures pointed to two primary causes:

- **Retrieval Errors (15/25 cases):** The retriever pulled irrelevant or conflicting context, especially for highly ambiguous queries with overlapping terminology.
- **Synthesis Failures (10/25 cases):** The LLM struggled to reason correctly when the answer required synthesizing information from multiple complex legal document chunks, even when the correct context was provided.

  Consider the following example of a **Retrieval Error**:

  Query: "What are the specific sampling and testing requirements for Mycoplasma in 12-week-old breeder hens?" **Retrieved Context (Incomplete):** The retriever pulled a general clause regarding flock health, stating, *"The establishment must implement a sanitary monitoring and control program to ensure the health of the poultry flock and prevent the spread of diseases."* **Missed Critical Context:** The retriever failed to pull the specific binding criteria for Mycoplasma testing from IN 44/2001 from the MAPA, which states the following for breeder birds aged 12 weeks:

  - **Initial Test:** A Rapid Serum Agglutination (SAR) test must be conducted.
  - **Sampling for *Mycoplasma gallisepticum*:** At least 300 samples must be tested.
  - **Sampling for *Mycoplasma synoviae*:** At least 100 samples must be tested.
  - **Sampling Method:** Samples must be selected randomly, with representation from each shed.
  - **Confirmatory Testing:** If any samples are reactive in the SAR test, they must be complemented with either a Hemagglutination Inhibition (HI) or an ELISA test.

Incorrect LLM Response: "According to the regulations, the establishment is required to have a sanitary monitoring program for its flocks. However, specific details on the number of samples or the types of tests required for Mycoplasma in breeder hens are not specified in this particular guideline."

This example highlights how the retrieval of incomplete context directly leads to an incorrect conclusion, demonstrating the critical importance of retrieval accuracy.

As detailed in Table 2, the RAG system not only achieved a 43-point improvement in strict accuracy but also demonstrated partial understanding in an additional 20% of cases. The modest increase in generation time is a necessary trade-off for transforming the LLM from an unreliable generator of generic text into a functionally useful assistant for domain-specific regulatory analysis.

Table 2. Model performance comparison with nuanced metrics, where C (Correct) indicates a fully correct response; PC (Partially Correct) indicates a response with the correct evidence but a flawed conclusion; and I (Incorrect) indicates a complete failure in retrieval or reasoning.

Model	C	PC	I	Avg. Generation Time (s)
Baseline	12%	0%	88%	5.2
RAG-Enhanced	55%	20%	25%	8.3

4.4.3. Quantitative Metrics and Statistical Significance

To supplement our qualitative analysis and provide a standardized benchmark, we calculated traditional performance metrics based on the expert evaluations. For this calculation, a response was considered successful only if it was classified as 'Correct' (C). Given that the system's task is to produce a correct analysis for each given scenario, we define Accuracy, Precision, Recall, and F1-Score where the positive class is a 'Correct' response. In this context, Precision, Recall, and F1-Score are mathematically equivalent to the strict accuracy rate.

We report these metrics alongside the 95% confidence interval for the accuracy of each model in Table 3.

Table 3. Standard performance metrics comparison. All metrics were calculated based on the 'Correct' classification as the positive outcome. The 95% confidence interval (CI) is provided for the accuracy metric.

Model	Accuracy	Precision	Recall	F1-Score	Accuracy 95% CI
Baseline	12.0%	12.0%	12.0%	12.0%	[5.6%, 18.4%]
RAG-Enhanced	55.0%	55.0%	55.0%	55.0%	[45.3%, 64.8%]

To determine if the observed improvement was statistically significant, we performed a chi-squared test of independence on the counts of correct versus incorrect (including partially correct) responses for both the baseline and RAG-enhanced models. The results showed a highly significant difference in performance ($\chi^2(1, N = 200) = 41.48$, $p < 0.001$). This indicates that the substantial improvement in accuracy achieved by the RAG-enhanced system was not due to random chance. The confidence intervals for the two models do not overlap, further supporting the conclusion that the RAG-enhanced system provided a significant and reliable performance gain over the baseline.

5. Discussion

5.1. Summary of Findings

The integration of a Retrieval-Augmented Generation (RAG) system into the PDSA-RS has proven to be a valuable advancement for 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.

5.2. Theoretical and Practical Implications

Beyond the specific application in animal health, the architecture presented in this study offers a generalizable and replicable model for developing domain-specific query systems in other regulated fields, such as finance or human healthcare. The modular stack—comprising a FastAPI endpoint, LlamaIndex for retrieval orchestration, PGVector for scalable vector storage, and Ollama for local model serving—provides a robust framework that can be adapted to various knowledge domains.

A significant advantage of this design is its reliance on locally hosted models, which ensures that sensitive or proprietary data remain within a secure infrastructure. This addresses data privacy and confidentiality concerns, making the approach a viable solution for organizations handling confidential regulatory information.

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.

5.3. Future Directions

Building upon the insights from this study, future development could focus on a more sophisticated hybrid approach. While the current system relies on RAG to provide timely and relevant responses using retrieved context, there is significant potential in complementing this with a selectively fine-tuned model. Specifically, fine-tuning the base LLM on legal texts, administrative proceedings, and veterinary-specific language could equip it with a deeper understanding of normative phrasing, formal structures, and domain reasoning patterns. When paired with RAG's access to real-time case data and evolving legislation, this would create a robust synergy; the model would not only retrieve the right documents but also interpret them with greater legal and contextual fidelity.

Additionally, introducing a feedback mechanism into the retrieval process—where expert users can rate or comment on the relevance of retrieved segments—would help iteratively improve the quality of retrieval over time. This would allow the system to learn from real-world usage patterns and evolve to better meet practitioners' needs.

5.4. User Interface Considerations

On the user interface front, while a simple Q&A format provides a good starting point, a more functional GUI could significantly enhance the platform's utility for veterinary professionals. Key features might include the following:

- A document ingestion panel where users can upload reports, legal documents, or certification requests and link them to specific certification workflows;
- Highlighted response explanations, showing which parts of the document were used in generating the answer, to promote transparency;
- Interactive forms for generating and validating certification drafts with contextual validation based on regulatory criteria;
- A versioning system to track document changes and recommendations over time, supporting auditability and compliance;
- Offline capabilities or mobile support for field veterinarians who operate in rural or low-connectivity areas.

Such features would not only make the tool more responsive to the everyday tasks of veterinary inspectors but also open the possibility of integrating it into broader workflows related to disease notification, certification issuance, and audit reporting.

5.5. Security, Privacy, and Ethical Considerations

The integration of AI systems into governmental platforms like the PDSA-RS, which handle sensitive information about producers, their properties, and sanitary status, necessitates a robust approach to security and privacy. The primary risk associated with such systems is the potential leakage of confidential data, especially when leveraging third-party AI services that process data externally.

Our architecture was designed with a "privacy-by-design" principle to mitigate this risk. The core mitigation mechanism is the deployment of the entire RAG pipeline within a controlled local infrastructure. By using the Ollama framework to serve the LLaMA 3.1 model, we ensure that no producer data ever leave the secure confines of the PDSA-RS environment. Queries and retrieved context are processed in-memory and are never transmitted to external APIs. This self-contained setup allows the system to operate in an effectively air-gapped manner, separate from the public internet, drastically reducing the attack surface for data exfiltration.

This approach is in direct alignment with the requirements of the Brazilian General Data Protection Law (LGPD). A critical aspect of our compliance strategy is that producer-specific data are used strictly for inference—that is, to generate a response for a single, specific query. The data are never used to train or fine-tune the LLM. This ensures that sensitive information is not incorporated into the model's parameters, upholding the LGPD principles of purpose limitation and data minimization.

Ethically, the system is positioned as a decision-support tool, augmenting the capabilities of human experts rather than replacing them. The final authority for any regulatory decision remains with the veterinary analyst, ensuring human oversight and accountability in this high-stakes domain.

5.6. Limitations

Despite the promising results, this study has several limitations that warrant consideration: First, the evaluation was conducted on a benchmark dataset limited to 100 scenarios from a single regulatory domain. While selected to be representative, this sample size may not capture the full spectrum of edge cases and complexities present in nationwide regulatory practices. Second, the efficacy of the entire system is critically dependent on the quality of the initial document retrieval stage. If the retriever fails to pull the most

relevant legal clauses for a given query, the generator is deprived of the necessary context, inevitably leading to an incorrect or incomplete response. Finally, while the RAG architecture significantly mitigates the risk of factual inaccuracies, it does not entirely eliminate the possibility of hallucinations. The LLM may still misinterpret complex legal language or synthesize information in a subtly flawed manner, reinforcing the necessity of expert human oversight in all final decision-making processes.

6. Conclusions

This study demonstrated that integrating Retrieval-Augmented Generation (RAG) into the PDSA-RS significantly improves the precision, transparency, and usability of legal and regulatory query handling in the veterinary health domain. The architecture proved effective not only for this specific case but also as a generalizable model for other regulated sectors requiring accuracy and compliance.

Ultimately, the proposed system highlights the value of hybrid AI approaches, combining retrieval with generation, technical robustness with user-centric design, and automation with human oversight. This balance ensures that innovation in regulatory platforms can progress without compromising legal integrity, privacy, or accountability.

Author Contributions: Conceptualization, P.B.M. and A.M.; methodology, P.B.M.; software, P.B.M., J.B.G., G.D., G.V.C. and G.R.d.S.; validation, T.O.B. and A.M.; formal analysis, P.B.M.; investigation, P.B.M.; resources, A.M.; data curation, P.B.M. and F.A.M.; writing—original draft preparation, P.B.M.; writing—review and editing, P.B.M., A.M. and V.M.; visualization, P.B.M.; supervision, A.M.; project administration, A.M.; funding acquisition, A.M. and V.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Institutional Postgraduate Scholarship Program (PIBPG) of CNPq, Call No. 50/2024, Master's scholarship modality (GM). This research is supported by FUNDESA, project UFSM/060496, by MPA (Ministério da Pesca e Aquicultura), project UFSM/060642 and FAPERGS, grant n. 24/2551-0001401-2. The research by Vinícius Maran is partially supported by CNPq grant 306356/2020-1 DT-2, the CNPq PIBIC and PIBIT program, and the FAPERGS PROBIC and PROBITI program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code is proprietary as it is integrated into the closed-source governmental platform the PDSA-RS. Additionally, the data are confidential and protected under the Brazilian General Data Protection Law (LGPD). To ensure maximum transparency and allow for conceptual replication, we provide (a) a description of the architecture and open-source components; (b) a detailed appendix with examples, including prompts, retrieved public legal texts, and model outputs.

Acknowledgments: We thank the members of the LUMAC laboratory for their insightful discussions and support.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Workflow Examples

Table A1. Workflow examples for RAG system evaluation.

Example 1: Correct Mycoplasma Sampling Verification	
Scenario	An analyst needs to verify if the number of samples collected for Mycoplasma testing in a flock of 16-week-old breeder hens (Nucleus NL01) adheres to national regulations. The farm's report states that 300 samples were tested for *Mycoplasma gallisepticum* and 100 for *Mycoplasma synoviae*.
Generated Query	Verify if the sample collection for *Mycoplasma gallisepticum* and *Mycoplasma synoviae* in nucleus NL01 complies with the standards for 16-week-old breeder birds, based on the provided records showing 300 and 100 samples, respectively.
Retrieved Context (IN 44/2001)	*"For breeder birds, monitoring for Mycoplasma gallisepticum and Mycoplasma synoviae is mandatory. At the 16th week of age, a minimum sample size is required. For M. gallisepticum, at least 300 samples must be collected for RSA testing. For M. synoviae, at least 100 samples must be collected..."*

Final Prompt (Template)	`Context:`
	`Retrieved Legal Text from IN 44/2001`
	`Question:`
	`Generated Query`
	`Instructions:`
	`Based only on the context provided, answer the question.`

LLM's Response	Based on the provided context from IN 44/2001, the sample collection for nucleus NL01 is compliant. The regulation requires a minimum of 300 samples for *M. gallisepticum* and 100 for *M. synoviae*. The farm's report meets these requirements.
Expert's Assessment	**Correct.** The system accurately identified the clause, interpreted the requirements, and confirmed compliance.
Example 2: Correct Salmonella Testing Verification	
Scenario	A certification request for a poultry farm indicates that for a flock aged 24 weeks, environmental drag swabs were collected from 4 sheds for Salmonella testing. The analyst needs to confirm compliance with the PNSA.
Generated Query	Does the collection of environmental drag swabs from 4 sheds for a 24-week-old flock comply with PNSA regulations for Salmonella monitoring?
Retrieved Context (IN 78/2003)	*"For Salmonella monitoring in breeder establishments, official sampling must occur between the 22nd and 26th week. The procedure involves collecting one pair of drag swabs per shed. The collected material from up to four sheds can be pooled for analysis..."*

Final Prompt (Template)	`Context:`
	`Retrieved Legal Text from IN 78/2003`
	`Question:`
	`Generated Query`
	`Instructions:`
	`Based only on the context provided, answer the question.`

LLM's Response	Yes, the procedure is compliant with IN 78/2003. The regulation specifies monitoring between the 22nd and 26th week and allows pooling of up to four sheds. Therefore, sampling from 4 sheds at 24 weeks is correct.
Expert's Assessment	**Correct.** The model correctly identified the age bracket, shed pooling rule, and confirmed compliance.
Example 3: Partially Correct Response due to Ambiguous Context	
Scenario	An analyst reviews a case where a farm tested for Mycoplasma at 10 weeks. The query is whether this test is sufficient for certification, without specifying production type (breeder vs. broiler).
Generated Query	Is a Mycoplasma test conducted at 10 weeks of age sufficient for sanitary certification?
Retrieved Context (IN 44/2001, IN 78/2003)	*"[Chunk 1: IN 44/2001] For breeder birds, mandatory testing is at 16 weeks... [Chunk 2: IN 78/2003] For commercial broilers, a sanitary program is required, but mandatory testing ages are not specified... [Chunk 3: IN 44/2001] Additional tests may be performed at any age if clinical signs are present..."*

Final Prompt (Template)	`Context:`
	`Retrieved Chunks from IN 44/2001 and IN 78/2003`
	`Question:`
	`Generated Query`
	`Instructions:`
	`Based only on the context provided, answer the question.`

LLM's Response	A Mycoplasma test at 10 weeks can be part of a control program, especially if clinical signs are present. For breeders, however, the mandatory test is at 16 weeks. Sufficiency depends on production type, which is not specified.
Expert's Assessment	**Partially Correct.** The system distinguished production types and mandatory 16-week testing but did not explicitly conclude "no" for breeder certification. The synthesis lacked decisiveness.

Appendix B. Glossary of Terms and Acronyms

AI (Artificial Intelligence) A field of computer science focused on creating systems capable of performing tasks that typically require human intelligence.

DBMS (Database Management System) Software used to create, manage, and interact with databases, ensuring data integrity, security, and accessibility.

DL (Deep Learning) A subfield of Machine Learning that uses multi-layered neural networks (deep neural networks) to learn hierarchical representations from data.

ELISA (Enzyme-Linked Immunosorbent Assay) A common laboratory test used to detect the presence of antibodies or antigens in a sample, widely used in veterinary diagnostics.

FUNDESA (Fundo de Desenvolvimento e Defesa Sanitária Animal) : The Fund for the Development and Animal Health Defense, a private entity in Rio Grande do Sul that supports actions to improve animal health in the state.

IN (Instrução Normativa, or Normative Instruction) A type of legal regulation issued by Brazilian government agencies, such as the MAPA, to establish technical standards and procedures.

LGPD (Lei Geral de Proteção de Dados) The Brazilian General Data Protection Law, a legal framework that regulates the collection, use, processing, and storage of personal data in Brazil.

LLaMA (Large Language Model Meta AI) A family of open-source Large Language Models developed by Meta AI, known for their performance and accessibility.

LLM (Large Language Model) A type of artificial intelligence model trained on vast numbers of text data to understand and generate human-like language.

LSTM (Long Short-Term Memory) A type of Recurrent Neural Network (RNN) architecture designed to effectively learn and remember long-range dependencies in sequential data.

MAPA (Ministério da Agricultura e Pecuária) The Brazilian Ministry of Agriculture and Livestock, the federal body responsible for formulating and implementing policies for agribusiness and regulating the agricultural sector.

ML (Machine Learning) A subset of AI where algorithms are trained on data to identify patterns and make predictions without being explicitly programmed.

NLP (Natural Language Processing) A specialized subfield of AI focused on enabling computers to understand, interpret, and generate human language.

PDSA-RS (Plataforma de Defesa Sanitária Animal do Rio Grande do Sul) The Animal Health Defense Platform of Rio Grande do Sul, a digital system for managing animal health certifications and regulatory compliance in the state.

PGVector An open-source extension for the PostgreSQL database that enables the storage and querying of vector embeddings, facilitating semantic search and similarity tasks.

PNSA (Plano Nacional de Sanidade Avícola) The National Poultry Health Plan, a program by the MAPA that establishes measures for the prevention, control, and eradication of diseases affecting poultry.

QLoRA (Quantized Low-Rank Adaptation) An efficient fine-tuning technique for Large Language Models that reduces memory usage by quantizing the model and using low-rank adapters.

RAG (Retrieval-Augmented Generation) An AI architecture that enhances the responses of a Large Language Model by retrieving relevant information from an external knowledge base at the time of inference.

REST API (Representational State Transfer Application Programming Interface) A standardized architecture for designing networked applications, allowing different systems to communicate over the web.

RNN (Recurrent Neural Network) A class of neural networks designed to work with sequential data, where connections between nodes form a directed graph along a temporal sequence.

RSA (Rapid Serum Agglutination) A serological test used for the rapid screening of certain diseases in poultry, such as Mycoplasmosis, by detecting antibodies in blood serum.

RT (Responsável Técnico, or Technical Lead) A licensed professional (typically a veterinarian) legally responsible for overseeing the sanitary procedures and compliance of a farm or establishment.

SVE (Serviço Veterinário Estadual, a.k.a. State Veterinary Service) The state-level governmental body responsible for executing animal health defense policies and inspections.

References

1. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 1877–1901.
2. Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N.A. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8342–8360.
3. Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. On the Opportunities and Risks of Foundation Models. *arXiv* **2021**, arXiv:2108.07258. [CrossRef]
4. Descovi, G.; Maran, V.; Ebling, D.; Machado, A. Towards a Blockchain Architecture for Animal Sanitary Control. In *Proceedings of the 23rd International Conference on Enterprise Information Systems–Volume 1: ICEIS, Online, 26–28 April 2021*; INSTICC, SciTePress: Setúbal, Portugal, 2021; pp. 305–312.
5. Schneider, R.; Machado, F.; Trois, C.; Descovi, G.; Maran, V.; Machado, A. 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, Angers, France, 28–30 April 2024*; INSTICC, SciTePress: Setúbal, Portugal, 2024; pp. 651–658.
6. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Kiela, D.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 9459–9474.
7. Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; Yih, W. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6769–6781.
8. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 4th ed.; Pearson: London, UK, 2020.
9. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
10. Machado, A.; Maran, V.; Augustin, I.; Wives, L.K.; de Oliveira, J.P.M. Reactive, proactive, and extensible situation-awareness in ambient assisted living. *Expert Syst. Appl.* **2017**, *76*, 21–35. [CrossRef]
11. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*; Curran Associates, Inc.: Red Hook, NY, USA, 2012.
13. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [CrossRef] [PubMed]
14. Heaton, J.B.; Polson, N.G.; Witte, J.H. Deep learning for finance: Deep portfolios. *Appl. Stoch. Model. Bus. Ind.* **2017**, *33*, 3–12. [CrossRef]
15. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
16. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30*; Curran Associates, Inc.: Red Hook, NY, USA, 2017.
17. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. *Improving Language Understanding by Generative Pre-Training*; Technical Report; OpenAI: San Francisco, CA, USA, 2018. Available online: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 16 November 2024).

18. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* **2019**, *1*, 9.
19. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:2302.13971. [CrossRef]
20. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv* **2023**, arXiv:2307.09288. [CrossRef]
21. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A Survey of Large Language Models. *arXiv* **2024**, arXiv:2303.18223. [PubMed]
22. Zhang, S.; Dong, L.; Li, X.; Zhang, S.; Sun, X.; Wang, S.; Li, J.; Hu, C.; Liu, Y.; Wang, H.; et al. Instruction Tuning for Large Language Models: A Survey. *arXiv* **2024**, arXiv:2308.10792. [CrossRef]
23. Thirunavukarasu, A.J.; Ting, D.S.J.; Elangovan, K.; Gutierrez, L.; Tan, T.F.; Ting, D.S.W. Large language models in medicine. *Nat. Med.* **2023**, *29*, 1930–1940. [CrossRef] [PubMed]
24. Cadena-Bautista, Á.; López-Ponce, F.F.; Ojeda-Trueba, S.L.; Sierra, G.; Bel-Enguix, G. Exploring the Behavior and Performance of Large Language Models: Can LLMs Infer Answers to Questions Involving Restricted Information? *Information* **2025**, *16*, 77. [CrossRef]
25. Wu, S.; Irsoy, O.; Lu, S.; Dabravolski, V.; Dredze, M.; Gehrmann, S.; Kambadur, P.; Rosenberg, D.; Mann, G. BloombergGPT: A Large Language Model for Finance. *arXiv* **2023**, arXiv:2303.17564. [CrossRef]
26. Yang, X.; Chen, A.; Pour, N.Z.; Rossi, A.; Li, C.; Al-Garadi, M.A.; Sarker, A.; Magge, A.; Liu, Y.; Bian, J. GatorTron: A Large Clinical Language Model to Unlock Patient Information from Unstructured Electronic Health Records. *arXiv* **2022**, arXiv:2203.03540. [CrossRef]
27. Izacard, G.; Grave, E. Leveraging Passage Retrieval with Generative Models for Open-Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 876–882.
28. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, H. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv* **2024**, arXiv:2312.10997. [CrossRef]
29. Zhao, P.; Liu, Z.; Wang, W.; Wu, Z.; Yang, S.; Zhang, Z.; Zhou, M. A Survey on Retrieval-Augmented Text Generation. *arXiv* **2023**, arXiv:2202.01110.
30. Khattab, O.; Zaharia, M. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 39–48.
31. Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv* **2023**, arXiv:2305.14314. [CrossRef]
32. Ministério da Agricultura, Pecuária e Abastecimento (MAPA). Normas técnicas para o controle e a certificação de núcleos e estabelecimentos avícolas para a micoplasmose aviária (Mycoplasma gallisepticum, M. synoviae e M. meleagridis). In *Diário Oficial da União. Instrução Normativa no 78, de 03 de novembro de 2003*; Ministério da Agricultura, Pecuária e Abastecimento (MAPA): Brasília, Brazil, 2003.
33. Ministério da Agricultura, Pecuária e Abastecimento (MAPA). Plano Nacional de Sanidade Avícola–PNSA. In *Diário Oficial da União. Instrução Normativa no 44, de 02 de outubro de 2001*; Ministério da Agricultura, Pecuária e Abastecimento (MAPA): Brasília, Brazil, 2001.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.