NOVA IMS Information Management School

MDSAA

Master Degree Program in

Data Science and Advanced Analytics

Malus Chatbot

A Chatbot for Apple Tree Cultivation in Portugal

Zenan Chen

Project Work

presented as partial requirement for obtaining a Master's Degree in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Malus Chatbot

A Chatbot for Apple Tree Cultivation in Portugal

by

Zenan Chen

Project Work presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a specialization in Data Science

Supervised by

Supervisor: Bruno Jardim, PhD, NOVA Information Management School

July, 2025

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Lisbon, 15 of July 2025

Zenan Chen

DEDICATION

I wish to dedicate this to my grandparents Cuifen and Dalin. Thank you for all the love.

ACKNOWLEDGMENTS

I would like to thank my supervisor Professor Bruno for his immense support throughout this thesis. I would also like to thank my friends and family for pushing me through and checking on me during all this time. Love you all.

ABSTRACT

This thesis presents the development of the Malus Chatbot, a Retrieval-Augmented Generation conversational agent designed to support apple cultivation in Portugal. Apple growers and stakeholders frequently face challenges accessing timely and reliable information due to the fragmented and unstructured nature of agricultural knowledge sources. While advances in Large Language Models have enabled significant improvements in natural language understanding, most existing agricultural chatbots remain limited by static or rule-based approaches, resulting in outdated or generic responses. The Malus Chatbot addresses this gap by combining a curated, domain-specific knowledge base with advanced retrieval and generation techniques to provide contextually relevant, evidence-based answers. Uniquely, the system can extract and present visual information like the figures and tables from source documents and offers direct links to these sources alongside each answer, supporting transparency and user verification. The system employs state-of-the-art embeddings, a hybrid retrieval strategy, and the GPT-4.1-mini language model to generate accurate and informative responses. Evaluation of the chatbot was carried out using RAGAS metrics and an expert provided question-answer dataset. Results demonstrate that the RAG-based approach substantially enhances answer quality, reliability, and source traceability compared to traditional methods. This work highlights the potential of retrieval-augmented conversational AI for advancing specialized knowledge access in agriculture and lays the foundation for future research in domain-adapted chatbot systems.

KEYWORDS

Retrieval-Augmented Generation; Chatbot; Generative AI; Large Language Models; Natural Language Processing; Agricultural Informatics; Apple Cultivation

Sustainable Development Goals

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
AIML	Artificial Intelligence Markup Language
ANN	Approximate Nearest Neighbour
BERT	Bidirectional Encoder Representations from Transformers
CRISP-DM	Cross-Industry Standard Process for Data Mining
ER	Ensemble Refinement
GPT	Generative Pre-trained Transformer
HMM	Hidden Markov Model
IoT	Internet of Things
ISA	Instituto Superior de Agronomia
KMS	Knowledge Management System
LLM	Large Language Model
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
OCR	Optical Character Recognition
RAG	Retrieval-Augmented Generation
RAGAS	Retrieval-Augmented Generation Assessment Suite
RNN	Recurrent Neural Network
SVM	Support Vector Machine
WSN	Wireless Sensor Network

1. INTRODUCTION

Apple cultivation stands as one of the most significant fruit sectors in mainland Portugal, contributing substantially to national fresh fruit production (Compal, n.d). Despite its importance, the sector faces persistent challenges such as market fluctuations, climate variability, disease management, and complex post-harvest processes (USDA GAIN Report, 2021). Growers and stakeholders increasingly rely on timely, trustworthy information to make critical decisions about cultivation, pest control, and market strategy. However, accessing relevant and up-to-date information remains a significant barrier due to the fragmented nature of agricultural knowledge sources (Fraga & Santos, 2021).

Recent advances in artificial intelligence (AI), particularly in natural language processing (NLP) and large language models (LLMs), have opened new possibilities for transforming how domain knowledge is accessed and delivered. LLM-based chatbots, powered by cutting-edge neural architectures, are now capable of understanding nuanced questions and providing human-like, context-aware responses (Bommasani et al., 2021; Brown et al., 2020). However, while LLMs demonstrate impressive fluency, they also suffer from limitations such as outdated or hallucinated answers due to their fixed training data (Ji et al., 2023; Roberts et al., 2020).

To overcome these challenges, retrieval-augmented generation (RAG) has emerged as a promising approach that grounds chatbot responses in up-to-date, authoritative sources by combining information retrieval with generative models (Lewis et al., 2020). By integrating a retriever that fetches relevant documents and a generator that synthesizes answers, RAG-based chatbots offer both the conversational ability of LLMs and the factual accuracy of structured knowledge bases.

This project introduces the Malus Chatbot: a RAG-based conversational agent designed to support apple growers in Portugal by providing accurate, context-aware answers on cultivation, plant protection, and market trends. Beyond text-based responses, the Malus Chatbot is capable of presenting visual information such as figures and tables from its knowledge base, and it provides direct access to source documents alongside each answer, supporting transparency and verification. The Malus Chatbot leverages a curated domain-specific corpus, advanced embedding and retrieval techniques, and state-of-the-art LLMs to address real-world information needs in agriculture.

The following chapters of this thesis develop this work in detail: the next chapter provides a comprehensive literature review, covering the evolution of NLP, chatbots, large language models, and retrieval-augmented generation, and critically assesses existing agricultural chatbots to clarify the research gap. The methodology chapter details the development of the Malus Chatbot, including data collection, knowledge base construction, and system implementation. Results and discussion are then presented, evaluating the chatbot using both automated metrics and expert human feedback. The thesis concludes with a dedicated

chapter summarizing the main findings, followed by a final chapter discussing limitations and outlining directions for future research and development.

By bridging the latest advances in AI and domain-specific knowledge retrieval, this thesis aims to demonstrate the practical value and challenges of deploying LLM-powered, retrieval-augmented chatbots in specialized, real-world contexts.

2. LITERATURE REVIEW

2.1. TECHNICAL BACKGROUND

2.1.1. NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP) is a subdiscipline of computer science and artificial intelligence that uses machine learning to facilitate computers' comprehension and interaction with human language. It empowers computers and digital devices to identify, comprehend, and produce text and speech by integrating computational linguistics, which involves rule-based modelling of human language, with statistical modelling, machine learning, and deep learning (IBM, 2024). NLP connects human communication with machine comprehension, enabling applications such as machine translation, sentiment analysis, and conversational agents, hence improving interactions between humans and technology.

Original concepts of NLP can be found in the works of Alan Turing, who proposed the idea of machines understanding and generating language in his paper “Computing Machinery and Intelligence.” These early systems were primarily rule-based models, meaning they relied on explicitly defined linguistic rules and patterns created by human experts to process and analyze language. For example, they used pre-determined grammar rules, dictionaries, and syntax structures to perform tasks like parsing sentences or translating languages. A notable example is the Georgetown-IBM experiment in 1954, which successfully translated Russian sentences into English, showcasing the potential of machine translation (Hutchins, 2005). However, these systems were limited by their reliance on handcrafted rules, which made them inflexible, unable to learn from data, and ineffective in handling the complexity, nuance, and ambiguity inherent in natural language.

The limitations of rule-based approaches led to the exploration of statistical methods, using probabilistic models and large corpora to capture language patterns. Hidden Markov Models (HMMs) were introduced for tasks such as speech recognition and part-of-speech tagging, modeling language as probabilistic sequences (Rabiner, 1989). Statistical machine translation also emerged during this period, with researchers at IBM developing models that treated translation as a statistical process based on bilingual corpora (Brown et al., 1990; Brown et al., 1993).

Building upon these statistical foundations, the following period saw the rise of machine learning approaches in NLP with algorithms like Support Vector Machines (SVMs) and Conditional Random Fields (CRFs), enabling models to learn complex patterns from data without explicit linguistic programming (Joachims, 1998; Lafferty et al., 2001). Feature engineering played a critical role during this era, with the performance of models heavily dependent on the quality of features extracted from text, including n-grams and syntactic

structures. Despite advancements, these approaches required extensive computational resources and faced challenges in generalizing to unseen data.

The advent of deep learning marked a significant transformation in NLP, enabling models to learn hierarchical representations of language data directly from raw text (Bengio et al., 2013). Traditional NLP approaches relied heavily on manual feature engineering and statistical methods, which often struggled to capture the complexities and nuances of human language. Deep learning models, particularly neural networks, have overcome many of these limitations by learning feature representations automatically from large datasets.

One of the most important concepts in deep learning for NLP is the use of word embeddings (Mikolov et al., 2013; Pennington et al., 2014). They are dense vector representations of words in a continuous vector space, where semantically similar words are mapped to nearby points. This representation allows models to capture syntactic and semantic relationships between words, helping with the better understanding and processing of language.

Mikolov et al. (2013) introduced the Word2Vec model, which learns word embeddings using shallow neural networks. The model employs two architectures, the first being the Continuous Bag-of-Words, which predicts a target word based on its surrounding context words, and the other being the skip-gram, which predicts surrounding context words given the target word. Another significant contribution is GloVe (Global Vectors) (Pennington et al., 2014), combining global word co-occurrence statistics to produce word vectors that capture meaning.

Another important concept is the use of neural networks, one of which being the Recurrent Neural Networks (RNNs). They are a class of neural networks designed to handle sequential data by maintaining a hidden state that captures information about previous inputs. In NLP, RNNs are particularly effective because they can process sequences of arbitrary length, making them suitable for tasks like language modeling, text generation, and sequence labeling. However, standard RNNs face challenges with long-term dependencies due to issues like vanishing and exploding gradients (Bengio et al., 1994). These problems make it difficult for RNNs to capture dependencies over long sequences, which is critical for understanding context in language.

To address the limitations of standard RNNs, Hochreiter and Schmidhuber (1997) introduced Long Short-Term Memory (LSTM) networks. They are a type of RNN that incorporate memory cells and gating mechanisms to better capture long-term dependencies. The key components of an LSTM cell include the Input Gate, controlling the extent to which new information flows into the cell; the Forget Gate, determining what information to discard from the cell state; and the Output Gate, regulating the information output from the cell. These gates allow LSTMs to maintain and update the cell state effectively, mitigating the vanishing gradient problem and enabling the modeling of long-range dependencies in text.

Sequence-to-sequence (Seq2Seq) models extend the capabilities of RNNs and LSTMs to handle tasks where both the input and output are sequences, potentially of different lengths. Introduced by Sutskever et al. (2014), the Seq2Seq framework consists of an Encoder that processes the input sequence and encodes it into a fixed-length context vector capturing the input's semantic information and a Decoder that generates the output sequence by decoding the context vector, producing one token at a time.

A limitation of early Seq2Seq models is the fixed-size context vector, which may not effectively capture all necessary information from long input sequences. Chorowski et al. (2015) proposed the attention mechanism to overcome this limitation. Attention allows the decoder to focus on different parts of the input sequence at each decoding step, dynamically weighting the importance of each input token.

The quick evolution of NLP, from early rule-based systems with statistical models into the era of deep learning, has dramatically increased the potential for machines to interact with human language in a more sophisticated way (Young et al., 2018). One of the most promising and impactful applications of these advances is in the development of conversational agents, commonly known as chatbots (Adamopoulou & Moussiades, 2020).

2.1.3. CHATBOTS

Chatbots serve as a practical and highly visible example of NLP techniques in real-world applications. By using the latest advances in machine learning and deep learning, chatbots use NLP methods to interpret user intent, manage ongoing conversations, and deliver responses that are not only relevant but also contextually appropriate (Adamopoulou & Moussiades, 2020).

The concept of chatbots dates back to the mid-20th century with the development of early systems like ELIZA (Weizenbaum, 1966), which used pattern matching and scripted responses to mimic a Rogerian psychotherapist. These chatbots were mostly rule-based, relying on predefined rules, templates, and decision trees to determine the system’s response. They worked well for small domains with predictable interactions, like FAQs or menu-driven customer service, but were limited in scope and unable to truly understand user intent or manage complex conversations. As a result, these systems were difficult to scale, inflexible to unexpected queries, and often produced repetitive or unnatural interactions (Shawar & Atwell, 2007).

The limitations of rule-based systems forced a shift to data-driven models that use machine learning and NLP techniques to learn conversational patterns from large datasets (Jurafsky & Martin, 2023). Retrieval-based chatbots select the best response from a repository of pre-written candidates based on similarity to the user, while generative chatbots construct responses word by word using neural network models.

The advances in deep learning and NLP mentioned earlier allowed for the development of highly sophisticated chatbot systems capable of handling diverse, open-domain conversations as well as specialized, domain-specific tasks. Generative models based on sequence-to-sequence architectures, often enhanced with attention mechanisms, can generate fluent and contextually relevant responses, even to previously unseen queries (Bahdanau et al., 2015).

A significant improvement in chatbot capability was made possible due to the adoption of transformer-based models and large language models (Vaswani et al., 2017; Brown et al., 2020). These models are pre-trained on massive corpora and can be fine-tuned for specific dialogue tasks, enabling chatbots to maintain context over multiple turns, understand nuanced user intent, and produce more natural, human-like dialogue.

2.1.4. LARGE LANGUAGE MODELS

Large Language Models (LLMs) are a class of artificial intelligence models designed to understand and generate human-like text by learning from large amounts of data. They have revolutionized NLP by achieving state-of-the-art results in various tasks such as translation, summarization, and conversational agents (Bommasani et al., 2021).

A defining feature of modern LLMs is their reliance on the transformer architecture (Vaswani et al., 2017). Transformers marked a turning point in NLP by introducing a self-attention mechanism, allowing models to capture complex dependencies between words, regardless of their positions in a sequence. Unlike earlier recurrent or convolutional models, transformers enable parallel processing of input data, leading to more efficient training and better performance on long-range context tasks.

LLMs are typically pretrained on large corpora using self-supervised learning objectives. Models like BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) introduced masked language modeling and next sentence prediction tasks to learn deep bidirectional representations, leading to significant improvements in understanding context and semantics, making it a cornerstone for many NLP applications.

Following BERT, models such as GPT (Generative Pretrained Transformer) series by OpenAI focused on unidirectional language modeling to generate coherent and contextually relevant text. GPT-3, with 175 billion parameters, showcased remarkable abilities in zero-shot and few-shot learning scenarios, effectively performing tasks without task-specific training data (Brown et al., 2020).

Research has shown that scaling up model size, data quantity, and computational resources leads to improved performance across a range of tasks (Kaplan et al., 2020). However, larger models also present challenges, including increased computational costs, difficulty in deployment, and the need for more sophisticated training techniques to manage issues like overfitting and catastrophic forgetting.

While LLMs have demonstrated impressive capabilities, they inherently suffer from limitations in capturing dynamic world knowledge. Since they learn statistical patterns from training data, their knowledge is fixed up to the cutoff date of the data they were trained on. This static nature leads to challenges in providing up-to-date information and handling domain-specific queries that require specialized knowledge (Roberts et al., 2020).

Moreover, LLMs are prone to generating plausible sounding but incorrect or nonsensical answers, a phenomenon known as hallucination. This issue raises concerns about the reliability of these models, especially in applications where accuracy is critical (Ji et al., 2023).

To address these limitations, recent research has focused on integrating LLMs with external knowledge sources through retrieval mechanisms. By augmenting LLMs with retrieval capabilities, models can access and incorporate relevant information at inference time, improving their ability to provide accurate and contextually appropriate responses (Lewis et al., 2020). This integration is particularly beneficial in domain-specific applications, such as specialized chatbots, where the model needs to access up-to-date and detailed information that may not be sufficiently represented in the pretraining data.

In the context of chatbot development, LLMs serve as the foundational models that generate responses based on user input. Their ability to understand context, manage dialogue flow, and produce human-like text makes them well-suited for conversational applications (Adiwardana et al., 2020). However, without augmentation, they may lack the specificity and accuracy required for domain-specific interactions.

2.1.5. RETRIEVAL AUGMENTED GENERATION

Retrieval-Augmented Generation (RAG) is an advanced framework that enhances the capabilities of generative models by integrating information retrieval mechanisms directly into the generation process (Lewis et al., 2020). In the context of chatbot implementation, RAG enables the creation of conversational agents that can produce responses that are not only contextually appropriate but also enriched with accurate and up-to-date information.

Traditional chatbot models, especially those based on sequence-to-sequence architectures, generate responses solely based on the data they were trained on. This limitation often results in outputs that lack specificity or contain outdated information, especially when dealing with open-domain conversations that require a vast and current knowledge base (Vinyals & Le, 2015). RAG addresses this challenge by incorporating a retrieval component that allows the chatbot to access and utilize external knowledge sources during inference.

The RAG framework consists of two main components: a retriever and a generator. The retriever searches through an external corpus, such as a database of documents, articles, or web pages, to find information relevant to the user's input. This is typically achieved using dense vector representations, where both queries and documents are embedded into a

shared semantic space using neural network encoders (Karpukhin et al., 2020). Similarity metrics, like cosine similarity, are then employed to identify the most pertinent documents efficiently.

Once relevant information is retrieved, the generator conditions its response on both the user's input and the retrieved content. This conditioning allows the chatbot to ground its responses in factual data, significantly improving the accuracy and relevance of the conversation (Lewis et al., 2020). For instance, if a user asks about recent events or specific factual information, the chatbot can provide precise answers by retrieving the latest data from the external corpus.

Implementing a RAG-based chatbot involves several key design decisions. These include selecting the language model, such as GPT-4, Llama, or other open-source and proprietary large language models, each offering different trade-offs in performance, cost, and domain suitability. Another important aspect is the type of retriever used, which may be dense (embedding-based), sparse (keyword-based, such as BM25), or a hybrid approach that combines both methods; each option has implications for retrieval accuracy and efficiency. The choice of vector database, for example ChromaDB (Chroma, 2023), FAISS (Johnson et al., 2017), or similar systems, affects scalability, speed, and integration with existing data sources. Additionally, retrieval strategies like the number of documents to retrieve, chunk size, context window, and the use of reranking techniques can all influence the final performance and relevance of the chatbot’s responses.

The application of RAG in chatbot development has demonstrated significant improvements over traditional models. Shuster et al. (2021) showed that incorporating retrieval mechanisms reduces instances of hallucination in conversational agents, where the model might otherwise generate incorrect or nonsensical information. By grounding responses in retrieved knowledge, the chatbot provides users with reliable and informative interactions.

Moreover, RAG enables chatbots to adapt to new information without the need for extensive retraining. Since the external knowledge base can be updated independently, the chatbot can access the latest data simply by refreshing the retrieval corpus.

2.1.6. Evaluation Metrics

To ensure a chatbot's effectiveness, accuracy, and user satisfaction, we must define comprehensive evaluation metrics. This evaluation must contain both quantitative metrics and qualitative assessments to comprehensively measure the chatbot's performance.

One primary method of evaluation involves automatic metrics that quantitatively assess the chatbot's responses. Perplexity is commonly used to measure the fluency of language models by evaluating how well the model predicts a sequence of words since a lower perplexity indicates better predictive performance and more fluent responses (Chen & Goodman, 1999).

However, perplexity alone may not capture the adequacy of responses in a conversational context, especially within specialized domains.

To evaluate the relevance and content quality, metrics such as BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) can be used. BLEU measures the n-gram overlap between the generated responses and a set of reference responses, focusing on precision (Papineni et al., 2002). ROUGE emphasizes recall by calculating the overlap of n-grams and sequences between the generated text and reference texts (Lin, 2004). While useful, these metrics may not fully capture the semantic meaning of responses in domain-specific contexts due to potential variations in terminology and expression.

To address semantic evaluation, BERTScore utilizes contextual embeddings from pretrained models like BERT to assess the similarity between generated responses and reference texts (Zhang et al., 2019). This metric goes beyond surface-level text comparison by considering the meaning of words in context, making it particularly suitable for evaluating responses in specialized domains where synonyms and paraphrases are common.

Evaluating the effectiveness of the retrieval component is equally important, as it directly influences the chatbot's ability to provide accurate and relevant information. Metrics such as Recall@K measure the proportion of relevant documents retrieved within the top *K* results, reflecting the retrieval system's capacity to supply necessary information for response generation (Karpukhin et al., 2020).

However, these traditional metrics may not fully capture the unique challenges of evaluating retrieval-augmented chatbots. High n-gram overlap does not necessarily indicate that a response is grounded in retrieved documents, and standard metrics may overlook the faithfulness of an answer to its context or the quality of document selection (Shuster et al., 2021). To address these limitations, specialized evaluation frameworks such as RAGAS (Retrieval-Augmented Generation Assessment Suite) have emerged (Es et al., 2024). RAGAS provides a suite of metrics such as faithfulness, answer relevance, context precision, and context recall, which are designed to evaluate both the retrieval and generation stages of RAG systems. These metrics offer a more comprehensive assessment of retrieval-augmented chatbots, especially in knowledge-intensive domains.

In addition to automatic metrics, human evaluation provides qualitative insights into the chatbot's performance. Domain experts or target users assess the chatbot's responses based on criteria such as relevance, accuracy, fluency, and coherence (Liu et al., 2016). Relevance and accuracy focus on whether the responses appropriately address the user's queries and contain correct domain-specific information. Fluency evaluates the grammatical correctness and naturalness of the language used, while coherence examines the logical flow and consistency of the conversation.

Furthermore, user satisfaction surveys and interaction studies can offer valuable feedback on the chatbot's usability and effectiveness in real-world scenarios. By gathering data on user engagement, perceived usefulness, and overall satisfaction, developers can identify areas for improvement and better align the chatbot with user needs.

2.2. Related Works

The development of agricultural chatbots has progressed through distinct technological generations, from early rule-based systems to modern AI driven approaches. To provide a comprehensive overview of the field, this section reviews the most relevant works in two categories. First, it examines rule-based chatbots, which rely on scripted responses and fixed decision trees to address user queries. Second, it explores AI-based chatbots, which employ machine learning, natural language processing, and, in some cases, large language models and retrieval-augmented generation to deliver more adaptive and context-aware support.

2.2.1. Rule-Based Chatbots

One notable early example of a rule-based agriculture chatbot is AgronomoBot, proposed by Mostaço et al. (2018), which was designed to assist vineyard management through a Telegram-based interface integrated with Wireless Sensor Networks (WSNs). By leveraging IBM Watson's NLP capabilities, AgronomoBot enabled users to access real-time environmental data such as soil moisture and temperature, offering basic decision-making support. However, its functionality was limited to predefined scenarios and lacked the flexibility to address complex, unstructured queries.

Ekanayake and Saputhanthri (2020) developed “E-AGRO”, an AI-powered chatbot aimed at assisting Sri Lankan farmers in making timely decisions through real-time interaction. The chatbot, built using Artificial Intelligence Markup Language (AIML), was designed to identify user intents based on predefined examples, addressing issues like crop selection and farming practices. While the chatbot demonstrated user-friendly functionality, its scope was limited to atomic questions, requiring further refinement for complex queries and broader geographical coverage.

Ong et al. (2021) developed a chatbot integrated into a Knowledge Management System (KMS) to assist small-scale farmers in Malaysia. The chatbot used NLP components to interpret user queries and provide responses based on a centralized knowledge repository containing agricultural best practices and expert contributions. While the chatbot improved accessibility and supported real-time interactions, it struggled to address nuanced or diverse queries due to its reliance on predefined responses and static databases.

Momaya et al. (2021) introduced “Krushi—The Farmer Chatbot”, an AI-driven conversational system designed to assist Indian farmers. Built on the RASA X framework, the chatbot used NLP for intent detection and query classification, enabling it to address diverse farming-

related inquiries. These included crop diseases, weather conditions, animal husbandry, and government schemes. The chatbot integrated external APIs, such as OpenWeatherMap, to provide real-time weather updates and was deployed via WhatsApp using Twilio for enhanced accessibility. Despite achieving a 96.1% accuracy in query resolution, Krushi relied heavily on predefined intents and responses, limiting its ability to handle more complex, context-aware interactions. The study underscores the need for advanced techniques, such as machine learning models and RAG, to enable more dynamic and personalized advisory systems for farmers.

Suebsombut et al. (2022) proposed a LINE chatbot integrated with Internet of Things (IoT) technologies to support smart agriculture in Thailand. The chatbot assists farmers by providing crop cultivation recommendations, monitoring environmental data, and controlling irrigation systems. Using a rule-based DialogFlow model, the system delivers insights on irrigation, fertilization, pest control, and weed management. Although it achieved a 96% user satisfaction rate, its reliance on exact keyword input limited usability. This study demonstrates the potential of combining chatbots with IoT for enhanced decision-making in agriculture while underscoring the need for more flexible and intelligent conversational models.

Collectively, these rule-based systems provided foundational solutions for digital agricultural support, enabling structured information access and basic decision-making for farmers. However, they were constrained by inflexibility, reliance on predefined scenarios, and limited capability to manage complex, open-ended, or context-dependent queries. These limitations motivated a transition toward AI-driven and more adaptive chatbot architectures in the agricultural sector.

2.2.2. AI Based Chatbots

AgroBot (Chandolikar et al., 2022) presents a chatbot designed to address the information gap among Indian farmers. The system processes user queries using NLP techniques such as tokenization, stemming, and the bag-of-words model to convert input into a structured format. For intent classification and response selection, AgroBot employs a Gaussian Naive Bayes classifier trained on labeled examples of typical farmer queries, allowing the chatbot to categorize incoming questions (e.g., on crops, weather, or farming techniques) and answer with predefined, contextually appropriate answers. While effective at resolving common and well-defined queries, AgroBot's architecture is restricted by its reliance on fixed datasets and pre-specified categories, limiting its ability to address more complex, open-ended, or novel user inputs.

CropBot (Banerjee et al., 2024) advances the field by integrating machine learning and NLP to deliver personalized crop recommendations based on soil and environmental parameters. The system allows farmers to input information in a flexible, conversational way, such as listing soil nutrients or describing environmental factors. CropBot automatically extracts the relevant values from the user's message using pattern-matching rules. These values are then processed

by a machine learning model, specifically Random Forest, which has been trained on a large dataset of crop and environmental data to identify the best crop choice for given conditions. Notably, the system can still provide recommendations even if the user does not supply all possible information by utilizing specialized models for partial input. For general farming queries, CropBot uses intent recognition to select appropriate responses, with this design enabling CropBot to deliver tailored, data-driven advice through an intuitive and accessible chat interface, although its recommendations are ultimately limited to the crops and scenarios represented in its training data.

Gaddikeri et al. (2023) explored the application of ChatGPT in agriculture, highlighting its potential to revolutionize precision farming, crop monitoring, livestock management, and food processing. ChatGPT's advanced natural language capabilities enable tasks such as generating crop yield predictions, advising on water and resource management, and assisting in farm mechanization. However, the study also acknowledges limitations, including the model's reliance on large datasets, potential biases, and computational constraints.

Silva et al. (2023) evaluated the performance of large language models, including GPT-4, in answering agriculture-related questions using datasets from Brazil, India, and the USA. The study employed standardized question-answer datasets, such as certification exam questions and structured multiple-choice items, to benchmark the models' capabilities. Results showed that GPT-4 significantly outperformed earlier models, achieving high accuracy rates, including 93% on Certified Crop Adviser exam-style questions. The research also incorporated advanced approaches like RAG and Ensemble Refinement, both of which further improved accuracy. While the study highlights the strong potential of LLMs for applications in agricultural education, certification, and crop management, the authors note the importance of addressing challenges related to factual accuracy and contextual understanding. Overall, the findings show the value of LLMs, particularly when combined with retrieval techniques, for advancing domain-specific problem-solving in agriculture.

Balaguer et al. (2024) investigated the trade-offs between RAG and fine-tuning for domain-specific applications of LLMs in agriculture. Using models like GPT-4 and Llama2-13B, they demonstrated how RAG enhances context relevance and retrieval efficiency, while fine-tuning enables models to acquire new skills for precise responses. Their comprehensive pipeline for generating agriculture-specific question-answer datasets highlighted cumulative accuracy gains of over 11 percentage points through combined RAG and fine-tuning approaches. This study underscores the potential of tailoring LLMs with RAG and fine-tuning to address industry-specific challenges, offering insights for scalable and efficient chatbot development in agriculture.

These AI-based chatbots represent a substantial advancement over rule-based systems by leveraging machine learning, natural language processing, and, in recent studies, large language models and retrieval-augmented generation techniques. They demonstrate

improved accuracy, adaptability, and the capacity for more personalized and domain-specific advisory services. However, limitations still remain, as most systems still rely on predefined datasets, face challenges with dynamic knowledge integration, and must address issues of factual accuracy, context awareness, and scalability, especially in specialized domains such as agriculture.

While recent research has begun to explore the combination of large language models with retrieval techniques, as seen in studies that apply RAG to agricultural question answering, these efforts are typically evaluated on standardized question-answer datasets such as certification exams or curated multiple-choice questions. As a result, their performance may not fully reflect the complexity and diversity of real-world agricultural information needs, where users often require answers grounded in open, heterogeneous document collections rather than narrowly scoped test questions.

Overall, despite clear progress in agricultural chatbot technology, the literature still reveals a gap in solutions that truly combine the conversational fluency and flexibility of large language models with robust, up-to-date knowledge retrieval from unstructured sources. Addressing this need for domain-specific, contextually aware, and reliable advisory chatbots that operate over real and diverse agricultural documents forms the central focus of the present thesis.

3. METHODOLOGY

For the development of the Malus Chatbot, the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework was adopted to provide a robust and transparent structure throughout the project lifecycle (Wirth & Hipp, 2000). This methodology offers a clear, stepwise approach to handling complex, data-centric projects, making it particularly well-suited for the iterative and multidisciplinary nature of building a RAG chatbot.

In this project, the methodology was adapted to fit the requirements of a RAG system, with each phase addressing the specific challenges of developing a domain-focused chatbot, ensuring that steps such as business understanding, data understanding, data preparation, RAG system development, deployment, and evaluation were carried out in a systematic and repeatable way.

The following sections describe how each phase was implemented in the context of the Malus Chatbot.

3.1. BUSINESS UNDERSTANDING

Apple cultivation is one of the most important fruit sectors in mainland Portugal, representing a substantial portion of the country's fresh fruit production. While growing, it has historically operated at a trade deficit, contributing only to about 2.5% of the total European Union apple production (Compal, n.d.).

Despite its relevance, apple growers in Portugal face challenges related to market fluctuations, climate variability, disease management, and post-harvest handling (USDA GAIN Report, 2021). Access to timely and trustworthy information is essential for effective decision-making in areas such as cultivation practices, pest and disease control, and marketing strategies. However, farmers and stakeholders often find it time-consuming and complex to retrieve relevant information from diverse sources, including scientific publications, government reports, and agricultural guidelines (Fraga & Santos, 2021). In this context, optimizing productivity and market access through knowledge dissemination is crucial.

The Malus Chatbot is designed to address this challenge by providing a conversational AI tool that assists users in retrieving information about apple cultivation efficiently. The chatbot is developed using RAG, which combines a knowledge base with a language model, ensuring that responses are based on authoritative and structured sources rather than relying solely on generative text predictions. The goal is to provide accurate, evidence-based answers to common questions in the domain of apple production, including cultivation techniques, market trends, pest control, post-harvest treatments, and economic factors.

3.2. DATA UNDERSTANDING

The Malus Chatbot relies on a diverse and structured dataset, sourced from Instituto Superior

de Agronomia (ISA) and other agricultural research institutions. The dataset consists of 131 documents, categorized into areas essential to apple cultivation, trade, post-harvest, and treatments, with the distribution shown in Table 1. These documents are stored in PDF, Excel, and CSV formats, containing a mix of structured data (tabular statistics), semi-structured content (reports, manuals), and unstructured data (research articles, scanned documents). The language of the documents also varies, there being English, Portuguese, and Spanish.

Table 1 – List of Training Documents

Document Subject	**Count**
Cultivation	18
Plant Protection Treatment	33
Post-Harvest	9
Trade	15
Statistics	7
Varieties	27
Projects	14
Other Resources	8

The dataset includes research papers, technical manuals, financial reports, phytosanitary guidelines, and market studies. Given the variety of document formats, an extensive preprocessing step is required to extract relevant textual, tabular, and visual data. PDFs contain both machine-readable text and scanned content that necessitate Optical Character Recognition (OCR) for text extraction. Excel and CSV files store structured tabular data, primarily used for financial records, production statistics, and market trends.

A key challenge in data processing is ensuring that structured numerical data and visual elements are meaningfully incorporated into the retrieval system. Since tabular data cannot be embedded directly for retrieval, it is converted into descriptive text while preserving key numerical relationships. Similarly, graphs and charts are indexed separately, with their captions and surrounding text stored to enhance search accuracy.

The processed dataset serves as the knowledge base for the chatbot, enabling it to retrieve domain-specific, high-quality information in response to user queries. By using a vector database and embedding techniques, the system ensures that relevant content is retrieved based on semantic similarity rather than simple keyword matching (AWS, n.d.). This structured approach guarantees that the chatbot provides contextually accurate responses, supporting farmers, agronomists, and researchers in their decision-making processes.

3.3. DATA PREPARATION

The Malus Chatbot requires an efficient data processing pipeline to transform diverse document formats into a structured, searchable knowledge base. Given the heterogeneous nature of the dataset, including PDFs, Excel files, and CSV files, the processing methodology

ensures that text, tables, and images are systematically extracted, structured, and embedded for retrieval.

To achieve this, the pipeline integrates metadata enrichment, content extraction, vectorization, and storage in ChromaDB, a high-performance vector database optimized for semantic search and retrieval.

3.3.1. METADATA DATA COLLECTION AND ENRICHMENT

Metadata is an essential aspect of the Malus Chatbot system, providing basic descriptive information about each document used in the knowledge base. In this project, metadata is sourced directly from an Excel file provided by domain experts. This file includes essential details for each document, such as the filename, title, institution source, publication year, document type, and topic. The filename serves as a unique identifier, ensuring that each document is accurately linked to its metadata. The title offers a brief description of the document's content, while the institution source identifies the organization responsible for its creation. The publication year provides a temporal reference, distinguishing between older and more recent documents. The document type categorizes the file based on its nature, such as research paper or guideline, and the topic indicates the primary subject it addresses.

In addition to this expert-provided metadata, the system also generates and stores basic file-related metadata during the data preparation process. This includes the source file path, the filename, the folder containing the document, and the type of file (e.g., "text", "excel", "csv").

Although this metadata is relatively basic, it can help in organizing documents and improving search efficiency. It also provides some useful contextual information that can enhance the RAG, allowing the system to better understand and filter documents based on their type, topic, or source. Therefore, it was integrated into the system during the data preparation process.

3.3.2. CONTENT EXTRACTION AND STRUCTURING

Content extraction is designed to systematically parse textual, tabular, and visual data, ensuring that all document elements are properly indexed for retrieval. Due to the existence of different document types, different strategies were employed.

3.3.2.1. PDF

Text extraction from PDFs was implemented using a hybrid approach, leveraging both *PyPDFLoader* (Langchain, 2025) and *pdfplumber* (Singer-Vine, 2025) with OCR via *Tesseract* (Google, 2025a). This dual strategy ensures that all text content, regardless of document type, is accurately extracted. The process begins with *PyPDFLoader*, which directly reads text from each page of standard text-based PDFs. Extracted text is enriched with metadata, including the source file path, filename, folder name, and page number. If *PyPDFLoader* detects empty

or near-empty pages (indicating scanned or image-based content), the system automatically applies OCR using *pdfplumber* and *Tesseract*. This fallback mechanism processes each page as an image, extracting text content from the images, ensuring that even scanned documents are accurately processed.

For table and figure extraction, a specialized method using *layoutparser* (Shen et al., 2021) and *Tesseract* OCR was employed. This approach is designed to detect and extract visual elements, including tables and figures, from PDF pages. The process begins by converting each page of the PDF into a high-resolution image using *pdf2image* (Belval, 2024) (Figure 1). These images are then processed by a pre-trained layout detection model, *Detectron2LayoutModel* (Wu et al., 2025), which identifies visual regions corresponding to tables and figures (Figure 2). Once identified, these regions are cropped and saved as separate images.

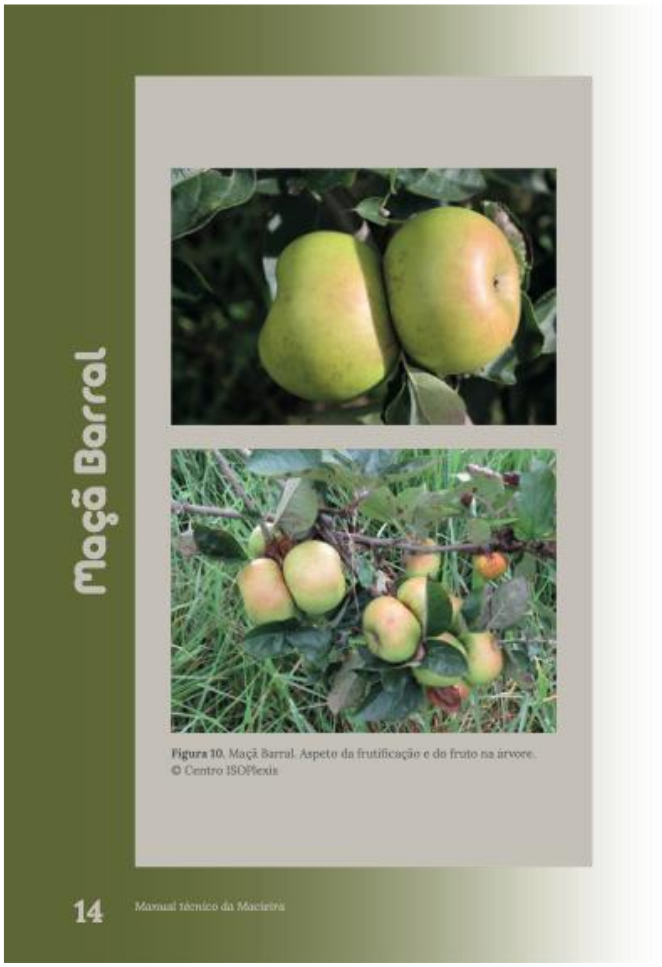

Figure 1 – Page with figure identified

Figure 2 – Automatically cropped image

In addition to extracting the image regions, the system also identifies and captures any accompanying captions. Using a dynamic cropping approach, an area below (for figures) or above (for tables) is extracted, capturing any descriptive text (Figure 3). *Tesseract* OCR is applied to this region to extract the caption text. If the caption text includes keywords such as "Figure," "Table," "Figura," or "Tabela," these labels are detected and used to properly categorize the visual. Each image and its corresponding caption are stored with metadata that includes the source file, page number, object type (table or figure), and a unique identifier.

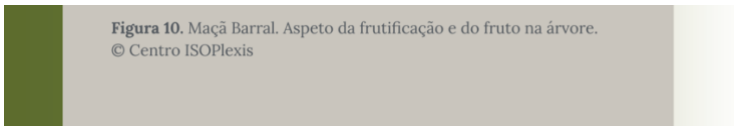

Figure 3 – Automatically cropped caption

Since RAG systems are designed to operate primarily with text, they cannot directly handle visual elements such as figures and tables. Therefore, it is necessary to convert these visual elements into textual descriptions to make them compatible. To achieve this, we leverage GPT's vision capabilities to generate detailed descriptions of the visual: each visual (figure or table), together with its caption, is sent to GPT-4o along with a carefully designed prompt that guides the model to identify the type of visual and generate a clear, informative description (OpenAI, 2024a). The generated descriptions are stored in the same format and structure as standard text chunks, allowing them to be indexed and retrieved like any other content. Furthermore, they are enriched with specific metadata such as the image path and object identifier, ensuring that the Malus Chatbot can reference and explain visual elements effectively during user interactions.

3.3.2.1. EXCEL

The process of extracting data from Excel files involves parsing each sheet, cleaning the data, and converting it into a structured format suitable for the Malus Chatbot. The system begins by loading the Excel file and iterating through its sheets. Each sheet is processed by removing empty rows and columns. If the sheet contains useful data (more than one column), it is converted into a markdown format for easy handling. A Document object is created for each valid sheet, which includes the table content and metadata such as sheet name, number of rows, and number of columns.

3.3.2.2. CVS

The process for handling CSV files involves reading the CSV, splitting it into manageable chunks, and converting each chunk into a markdown-based Document object. This allows for large CSV files to be processed efficiently by breaking them into smaller pieces, each representing a section of rows. Each chunk is then stored with relevant metadata, such as the file name, source path, and a range of rows.

3.3.3. CHUNKING

After processing the documents, a chunking strategy is applied to all the processed documents in order to break down the content into smaller, manageable pieces. This involves splitting documents into smaller sections based on character count, with a defined chunk size and overlap between consecutive chunks, ensuring that long documents or sections with complex content are divided into coherent, contextually meaningful parts while maintaining smooth transitions between chunks. In this implementation, a chunk size of 500 characters with an overlap of 50 characters was used to preserve contextual flow between adjacent chunks.

By breaking down the documents into smaller chunks, the system can index these chunks for efficient retrieval of the most relevant sections based on user queries, rather than processing the entire document, allowing for more precise and efficient responses, especially when dealing with larger volumes of data.

Additionally, each chunk is assigned a unique identifier, which makes it easier to locate neighboring chunks for future reference or retrieval, ensuring smooth transitions and context preservation between chunks when needed.

3.3.4. TEXT EMBEDDING AND VECTORIZATION

Following the chunking process, each chunk of text was transformed into a numerical representation called a vector embedding. Embeddings are dense vectors of floating-point numbers that capture the semantic meaning of the text. In simpler terms, they translate words, phrases, or entire documents into a format that machine learning models can understand. Each dimension in the vector represents a different aspect of the text's meaning,

and the combination of these values creates a unique representation for that specific piece of text.

The utility of vector embeddings stems from their ability to represent semantic relationships. Texts with similar meanings are located close to each other in the high-dimensional space defined by the vectors, while texts with dissimilar meanings are further apart. When a user poses a query for our chatbot, we can also convert this query into an embedding, and the system can then efficiently retrieve the most relevant text chunks from the knowledge base by comparing the similarity of their embeddings. This allows the chatbot to provide responses that are grounded in the most pertinent information.

We chose to use *text-embedding-3-large* (OpenAI, n.d.) model from OpenAI to generate the vector embeddings. This model was selected over the regular *text-embedding-ada-002* (OpenAI, n.d.) model due to its superior performance in handling multilingual data (OpenAI, 2024b), which is particularly important since our chatbot is primarily designed for Portuguese-speaking users, and the used dataset contains a significant portion of text in Portuguese, along with a considerable amount of English and some Spanish. The *text-embedding-3-large* model, specifically designed for enhanced multilingual capabilities, allows for more accurate and consistent representation of semantic meaning across these different languages (OpenAI, 2024b). This enhanced multilingual performance is crucial for ensuring the effectiveness of our chatbot, as it enables the retrieval of relevant information across the diverse linguistic content of our dataset, leading to more accurate and comprehensive responses.

3.3.5. Integration Into ChromaDB For Retrieval

Following the generation of vector embeddings, these numerical representations must be stored in a manner that allows for efficient retrieval. Several options exist for this purpose, including specialized vector databases, which are designed to handle the unique demands of similarity search, offering significant advantages over traditional databases when working with embeddings. While traditional databases excel at exact match queries on structured data, vector databases are optimized for approximate nearest neighbor (ANN) search (Johnson et al., 2019). ANN search enables the efficient retrieval of vectors that are most similar to a given query vector, even within massive datasets.

In the Malus Chatbot system, we utilize ChromaDB as the core storage and retrieval engine for these vector embeddings. ChromaDB is a high-performance, open-source vector database that excels at handling large-scale semantic search. It was chosen for its ease of use, scalability, and strong performance in retrieving relevant information based on similarity (Jing et al., 2025).

Specifically, ChromaDB functions within the Malus Chatbot system as follows: once the text, tables, and images are transformed into numerical embeddings, they are indexed and stored within ChromaDB. Each stored embedding is associated with its original source document or

data segment. When a user submits a query, that query is also converted into an embedding vector. ChromaDB then efficiently compares this query embedding to the stored embeddings using similarity search algorithms to find the most similar vectors. The most similar embeddings, along with their associated source data, are then retrieved. This retrieved information is then passed to the LLM, enabling it to generate a response grounded in the most relevant information from the knowledge base.

3.4. RAG

The section focuses on designing and implementing a question-answering chatbot capable of retrieving and generating responses based on structured and unstructured data related to apple cultivation. The system follows a RAG approach, integrating ChromaDB for efficient document retrieval and OpenAI's GPT-4.1-mini (OpenAI, 2025a) model for response generation. This architecture ensures that responses are accurate, context-aware, and supported by relevant documents, tables, and images.

3.4.1. Query Processing And Context Management

In the Malus Chatbot, the emphasis on rigorous query preprocessing is intentionally reduced. While traditional NLP systems often rely heavily on techniques like stop word removal, stemming, and normalization, the capabilities of modern LLMs offer a different perspective. LLMs possess a strong inherent ability to understand natural language, including variations in phrasing, grammar, and even minor errors. These models can effectively interpret the user's intent even with slight imperfections in the query.

However, even though LLMs are very good at understanding and answering single questions, real conversations often involve more than just one message at a time. In these cases, managing context is important for the Malus Chatbot to have meaningful back-and-forth interactions. Users may refer to things they said earlier, ask follow-up questions, or add new information as the conversation goes on. To respond accurately, the chatbot needs to remember and use information from previous messages in the conversation.

To achieve this, the system feeds the LLM a prompt that includes the conversation history and the user's latest query. The LLM then analyzes the query to determine its dependence on the preceding context. If it is identified that the query requires context, the LLM rewrites the query to incorporate the necessary information from the history, making it self-contained. This ensures that the retrieval system receives a clear and complete query, regardless of its original form. If the query is deemed independent or if the previous context is irrelevant, the original query is returned unchanged. This selective context incorporation optimizes retrieval accuracy and prevents the inclusion of extraneous information.

The returned query, whether rewritten or original, is then embedded using the same model used to generate embeddings for the knowledge base, enabling the system to compare them in a later stage and retrieve the most relevant information.

3.4.2. RETRIEVAL

Retrieval plays a central role in RAG systems by determining which information is selected and passed to the language model, since the relevance and quality of this retrieved context directly affect the factual accuracy and usefulness of the generated answers (Lewis et al., 2020). In the Malus Chatbot, both dense and hybrid retrieval strategies are employed. The hybrid approach combines dense vector-based retrieval with sparse keyword-based methods such as BM25, allowing the system to leverage both semantic understanding and exact keyword matching to identify the most relevant information.

3.4.2.1. DENSE RETRIEVAL ALGORITHMS

Dense retrieval algorithms utilize vector embeddings to semantically represent textual data. Recent advancements in neural embeddings, such as BERT-based models (Devlin et al., 2018), have enabled more contextually relevant retrieval by capturing semantic relationships beyond mere keyword matching. In dense retrieval, each document chunk and user query are transformed into numerical embeddings, facilitating semantic similarity comparisons. ChromaDB, a popular vector database, enables efficient dense retrieval (Chroma, 2023), typically employing either cosine similarity or L2 distance as similarity metrics.

Cosine similarity assesses the similarity between two embedding vectors based on the angle between them. The formula is:

$$Cosine\ Similarity = \frac{A \cdot B}{\|A\|\|B\|}$$

This metric effectively captures semantic similarities by focusing solely on the direction rather than magnitude.

L2 distance measures the Euclidean distance between vectors, considering both direction and magnitude. The formula is:

$$L2\ Distance = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$

It captures semantic nuances that cosine similarity might overlook due to its sensitivity to vector magnitude.

3.4.2.2. SPARSE RETRIEVAL ALGORITHMS BM25

Sparse retrieval algorithms primarily rely on term frequency and inverse document frequency metrics, assessing relevance based on keyword occurrences. BM25, a well-established sparse retrieval algorithm introduced by Robertson and Zaragoza (2009), scores documents according to the likelihood of relevance derived from term frequency statistics. The BM25 formula is:

$$\mathrm{BM25}(Q,D) = \sum_{i=1}^{n} \mathrm{IDF}(q_i) \frac{f(q_i,D)\cdot(k_1+1)}{(q_i,D) + k_1\left(1 - b + b\frac{|D|}{\mathrm{avgdl}}\right)}$$

where:

Q is the query containing terms q_i;

D is the document;

$f(q_i, D)$ is the frequency of term q_i in document D;

$|D|$ is the length of document D;

avgdl is the average document length in the corpus;

k_1 and b are parameters typically set to 1.2 and 0.75, respectively, optimizing term frequency saturation and document length normalization;

$\mathrm{IDF}(q_i)$ calculates term scarcity across documents as:

$$\mathrm{IDF}(q_i) = \log\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right)$$

where N is the total number of documents, and $n(q_i)$ is the number of documents containing term q_i.

3.4.2.3. HYBRID RETRIEVAL

Hybrid retrieval strategies combine dense and sparse methodologies to harness their complementary strengths. EnsembleRetriever, an established hybrid retrieval method, integrates semantic-based dense retrieval and keyword-based sparse retrieval. We use Langchain's EnsembleRetriever to aggregate the outputs from both retrieval methods by applying weighted combinations. Typically, the dense and sparse retrieval results are equally weighted, ensuring balanced consideration of both semantic similarity and keyword specificity. Recent studies have shown that hybrid retrieval approaches often outperform single-method retrieval systems, particularly in domains where both semantic understanding and precise keyword matching are necessary (Izacard & Grave, 2020).

3.4.3. CONTEXT ASSEMBLY AND PROMPTING

This section details the process of constructing prompts to effectively guide the LLM in generating accurate and informative responses. The prompt engineering is designed to provide the LLM with both the user's query and relevant context retrieved from the knowledge base while also instructing the LLM on how to structure its output for efficient processing and display by the chatbot application.

The prompt construction involves assembling context blocks from the retrieved documents. For each document, the system includes relevant metadata such as the source path, file name, and page number. The full system prompt can be found in Appendix A. The inclusion of the source path and file name is crucial because it allows the chatbot to provide users with direct access to the original source documents, enhancing transparency and enabling users to verify the information. If the document is a figure or table, an image block is included, containing the display name and image path. Specifying the image path is essential for the chatbot to display the referenced images within the response, providing a richer and more visually informative user experience. The document's content is then appended to this block. These context blocks are concatenated and presented to the LLM as the “Context”.

Furthermore, a detailed system message is provided to the LLM, instructing it to act as a scientific assistant specialized in apple cultivation. The system message outlines specific requirements for the LLM's response, most notably the requirement to structure the response as a single, valid JSON object. This design choice is critical for efficient post-processing. By receiving the output in a structured JSON format, the system can easily parse the response and access specific elements such as the answer text, image details, and source document information, eliminating the need for complex text parsing and reducing the risk of errors in extracting the required data.

After several trials with different output formats, it was found that requesting a JSON format was the most reliable way to ensure the LLM consistently included all necessary information, such as source files and mentioned images, without omissions or formatting errors. The JSON object includes specific keys: "answer" (containing the answer in Markdown format), "images" (a list of image details with display names and image paths), and "sources" (a list of source document details with file names, page numbers, and source paths). This formatting allows the chatbot to display the answer, present any referenced images, and provide clickable links or references to the source documents, as the JSON structure is easily parsed and utilized by the application. See Appendix B for an example of the unparsed LLM response.

This structured prompting approach ensures that the LLM receives clear instructions, relevant information, and formatting guidelines, leading to more accurate, informative, and usable responses, while also significantly simplifying the post-processing of the LLM's output for seamless integration with the chatbot's display functionalities.

3.4.4. ANSWER GENERATION MODEL

The choice of the LLM for answer generation is one of the most important design decisions in a RAG-based chatbot, as it directly influences the fluency, factual accuracy, and overall user experience of the system. Even with a well-tuned retriever, the model must be capable of synthesizing retrieved content into clear and relevant answers. In practice, the quality of the final response often depends more on the capabilities of the LLM than on retrieval alone, making this component central to overall performance (Lewis et al., 2020).

A first consideration involves choosing between open-source and proprietary language models. Open-source models such as Mistral, or LLaMA offer transparency, offline control, and cost advantages in the long term, but they are generally harder to configure, especially in a multilingual domain-specific setting like this one. Their performance also tends to be lower without extensive fine-tuning, and their inference speed and stability are highly dependent on local hardware resources (Mistral AI, 2024; Touvron et al., 2023). Given these limitations, we opted for a proprietary solution.

The most advanced large language models available today are so-called “reasoning” models, designed to excel at multi-step inference, complex problem-solving, and generalization across domains. Models such as Google’s Gemini 2.5 Pro and OpenAI’s latest reasoning variants have been demonstrated to achieve state-of-the-art performance in complex reasoning tasks and long-context understanding, but they are also significantly more costly to operate and require considerably longer inference times compared to more efficient alternatives (Google, 2025b; OpenAI, 2025b). While they are recognized as setting new benchmarks in language understanding and reasoning, they are not always the most appropriate choice for RAG. In the context of RAG, where the main requirement is to generate responses grounded in retrieved documents, the advanced reasoning capabilities of these models are often underutilized. Given their high computational cost and latency, and the limited benefit these features offer in grounded generation tasks, we opted against deploying the latest reasoning models, instead selecting a model that offers a stronger balance of performance, efficiency, and practical deployment.

Among proprietary providers, OpenAI is widely recognized by the wider AI research and industry community as the leading developer of large language models. OpenAI’s flagship model, GPT-4.1, provides state-of-the-art fluency and factual accuracy, yet maintains lower inference costs and faster response times compared to the newest and most resource-intensive models (OpenAI, 2025a). It also demonstrates robust performance on multilingual and domain-specific queries, which are essential for our application. Another significant advantage is the seamless integration offered by the OpenAI API, which was also used for embedding generation in this project. By using the same provider for both language modeling and embedding, system integration and maintenance were streamlined, and the overall development process was simplified. In addition, the maturity of OpenAI’s infrastructure and its stable API ecosystem further reduce integration and maintenance overhead. For these reasons, we chose GPT-4.1 as the LLM for the Malus Chatbot.

Figure 4 illustrates the full system architecture described until now.

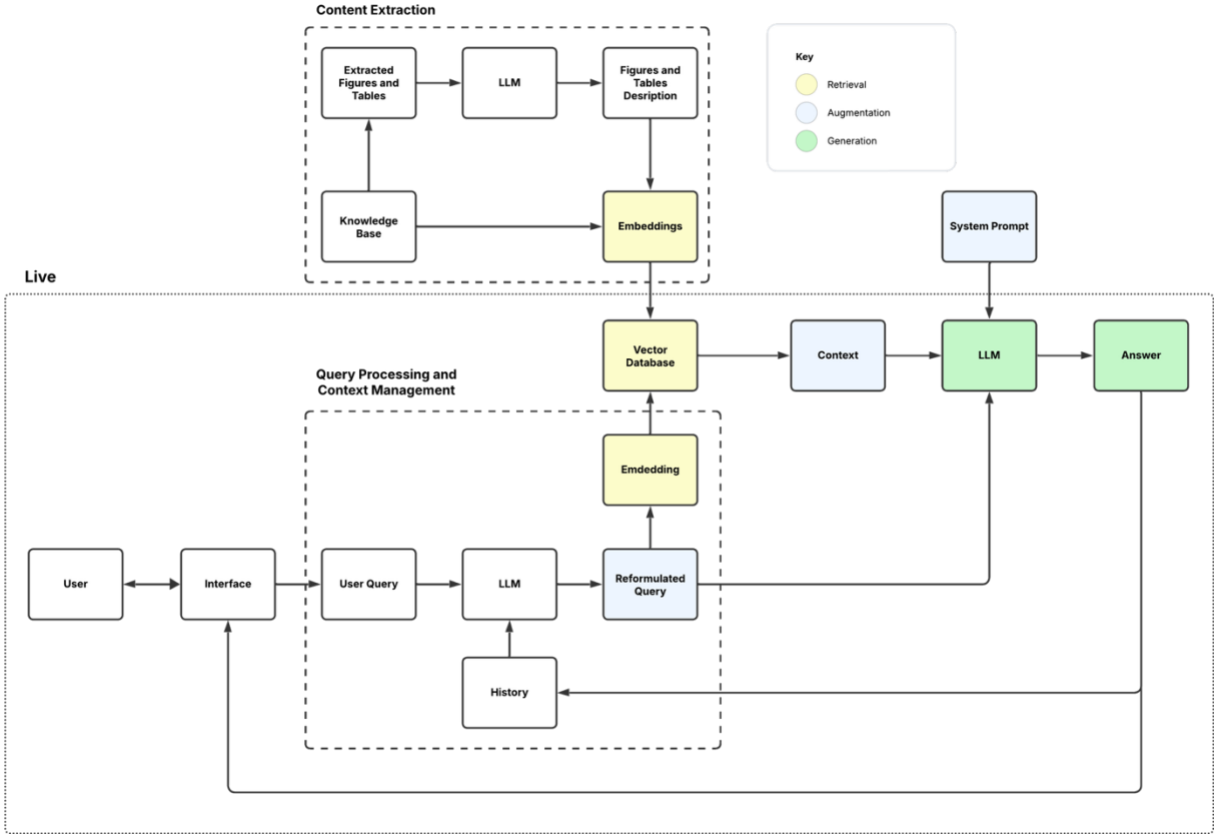

Figure 4 – Malus Chatbot system architecture

3.5. DEPLOYMENT

To provide users with a way to interact with the Malus Chatbot, a user-friendly deployment strategy was essential. Streamlit, a Python library for creating interactive web applications, was chosen for its ability to streamline the development of data-driven interfaces. With Streamlit, users can easily input their questions and receive the chatbot's responses in a conversational format, while the application can dynamically display various types of content, including text, images, and hyperlinks, all within a single Python script, enabling the rapid development of an engaging and functional chatbot interface without the complexities of traditional web development.

For hosting and sharing the application online, Streamlit Cloud was used. Streamlit Cloud is a managed platform that allows developers to deploy, host, and share Streamlit applications without the need for complex server setup or ongoing maintenance. However, it is limited by not supporting persistent local file storage, which is necessary for handling the chatbot's vector database as well as supplementary files and images. To overcome this, Hugging Face's cloud platform was integrated into the solution. We use it to host the vector database along with all relevant document files and images, ensuring reliable, scalable, and efficient access to these resources during chatbot operation.

The structured JSON output from the LLM, as described in Section 3.4.3, is fundamental to how information is presented to the user. This format enables the Streamlit application to efficiently parse and display the different components of the LLM's response. Specifically, the answer text is presented as the chatbot's response within the chat interface. Furthermore, image URLs and display names from the JSON are used to display images directly in the conversation, enriching the user's understanding when visual information is relevant. Finally, source file names, page numbers, and URLs are used to generate clickable links or references, allowing the users to download the original documents and images, enabling them to verify the information provided by the chatbot. This deployment strategy, which combines Streamlit's ease of use with the structured JSON output, delivers an intuitive and informative chatbot experience, empowering users to effectively explore and utilize the knowledge base.

Figures 5, 6 and 7 show the deployed Malus Chatbot with an example interaction, showcasing all the capabilities mentioned.

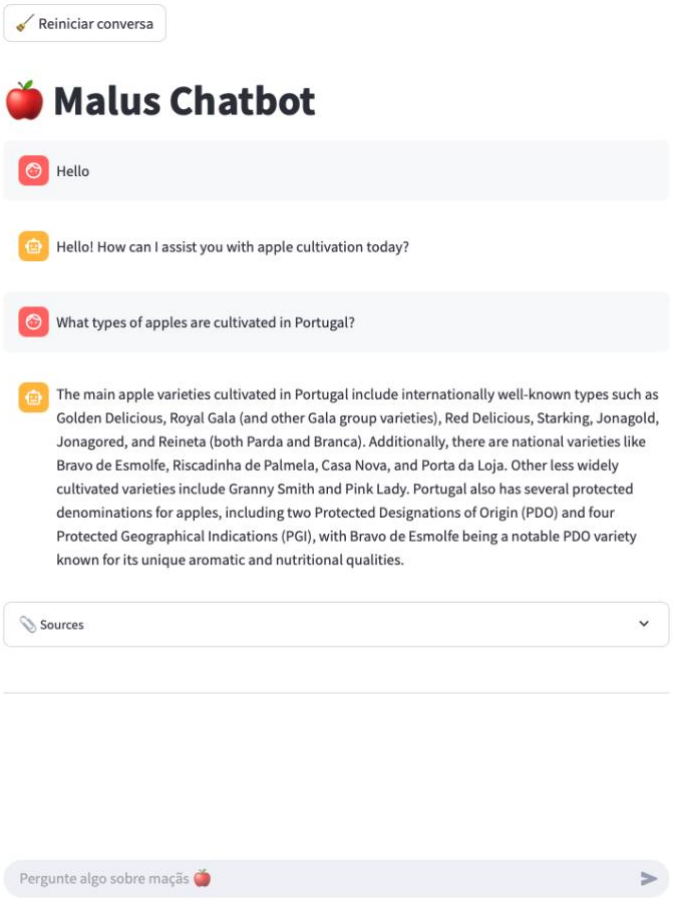

Figure 5 – Malus Chatbot example interaction

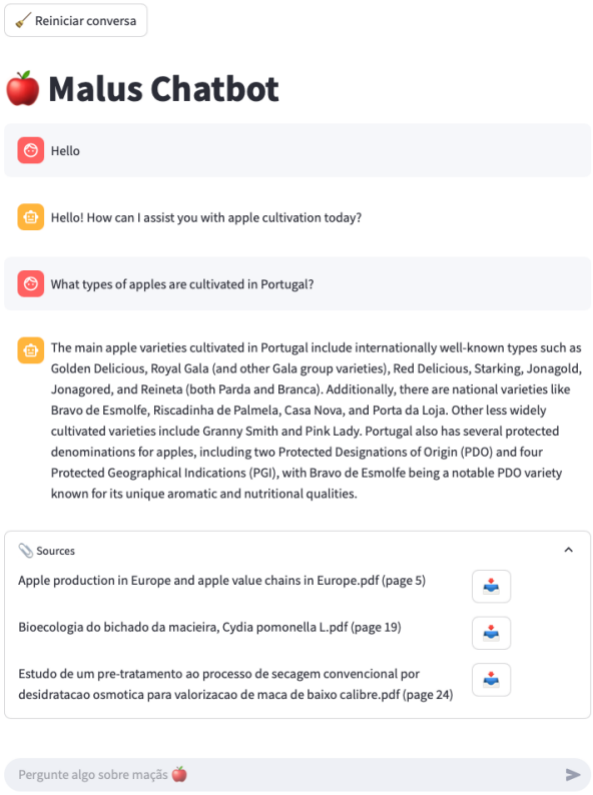

Figure 6 – Conversation with sources displayed and available for download

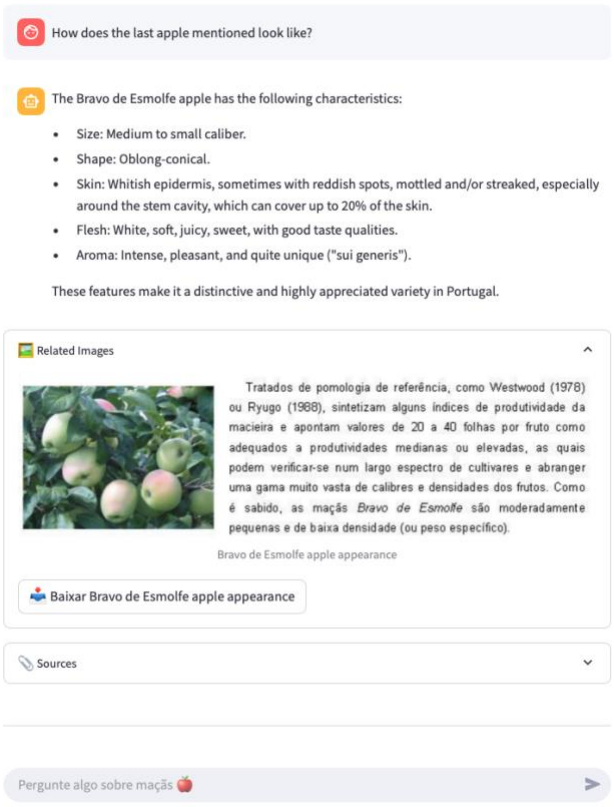

Figure 7 – Conversation showing context retention with image displayed and available for download

3.6. EVALUATION

Evaluating the performance of a retrieval-augmented chatbot requires a comprehensive approach that considers both quantitative and qualitative perspectives. Automated evaluation enables efficient, large-scale, and objective assessment, but it may not fully capture contextual accuracy or domain-specific relevance (Shuster et al., 2021). Human expert evaluation, on the other hand, provides deeper insights into the quality, trustworthiness, and practical usefulness of the chatbot's answers (Dinan et al., 2020). By combining these complementary methods, a more holistic understanding of the chatbot's strengths and limitations can be achieved, ensuring it meets the needs of users in real-world settings.

3.6.1. AUTOMATED EVALUATION

For the automated evaluation, the central resource used is a dataset consisting of 86 domain-specific questions and expert-validated answers. Developed by specialists from ISA, FNOP (Federação Nacional das Organizações de Produtores de Frutas e Hortícolas), and DGADR (Direção-Geral de Agricultura e Desenvolvimento Rural), this dataset covers a wide range of topics relevant to apple cultivation, including agronomic practices, pest and disease management, post-harvest, treatments, market trends, and apple variety characteristics. Each question in the dataset is directly linked to information found within the documents of the chatbot's knowledge base, ensuring that the evaluation reflects real-world inquiries and the actual scope of the available resources. The expert-validated answers serve as a benchmark against which the chatbot's generated responses can be compared, aligning the evaluation process with scientifically accurate and practically relevant knowledge.

The framework used for automated evaluation is RAGAS (Retrieval-Augmented Generation Assessment Suite), a specialized evaluation suite designed for retrieval-augmented generation systems. RAGAS operates by evaluating both the retrieval and answer-generation stages of the chatbot. For each question, RAGAS considers the user prompt, the chatbot's generated answer, the set of documents retrieved from the knowledge base, and the corresponding expert reference answer. It then uses natural language processing models and specialized algorithms to compare the generated answer to the expert reference and analyze how well the retrieved documents support the chatbot's response. This enables RAGAS to provide transparent and detailed feedback on both the retrieval and generation components, offering actionable insights into system performance (Es et al., 2024).

For the evaluation of the Malus Chatbot, six key RAGAS metrics were selected:

- Context Precision: Measures how much of the retrieved context actually contains information that is relevant to answering the user's question. A high context precision score means that most of the context provided is useful and directly related to the query, while a lower score suggests that the context includes a lot of unrelated or unnecessary information. It is determined by comparing the relevant parts of the

retrieved context to the total amount of context retrieved. The score reflects the proportion of the context that was truly helpful for generating an accurate response, ranging from 0 to 1.

$$\text{Context Precision@K} = \frac{\sum_{i=1}^{K}(\text{Precision@k} \times v_k)}{\text{Total number of relevant intems the top } K \text{ results}}$$

$$\text{Precision@K} = \frac{\text{true positives@k}}{\text{true positives@k} + \text{false positives@k}}$$

- Context Recall: Measures how much of the relevant information from a reference answer is present in the retrieved context. This metric focuses on completeness, ensuring that important details are not missed. Calculating context recall requires a reference answer, which is divided into individual claims. Each claim is then checked to determine if it is supported by the retrieved context. Perfect context recall is achieved when all claims from the reference answer can be found in the retrieved context.

$$\text{Cotext Recall} = \frac{\text{Number of claims in the reference supported by the retrieved context}}{\text{Total number of claims in the reference answer}}$$

- Faithfulness: Assesses how factually accurate a response is compared to the retrieved context, with scores ranging from 0 to 1, with a higher score indicating a greater consistency. A response is considered faithful if every claim it makes is supported by the retrieved context. To determine this, each claim is checked against the provided information, and the score is calculated based on how well the response aligns with the context.

$$\text{Faithfullness Score} = \frac{\text{Number of claims in the response supported by the retrieved context}}{\text{Total number of claims in the response}}$$

- Answer Relevancy: Evaluates how well a response addresses the user's input. Higher scores indicate that the response closely matches the user's intent, while lower scores reflect answers that are incomplete or contain unnecessary information. To calculate this metric, artificial questions are generated based on the content of the response. The cosine similarity between the embedding of the user input and the embedding of each generated question is then measured. The final Answer Relevancy score is obtained by averaging these similarity values.

$$\text{Answer Relevancy} = \frac{1}{N}\sum_{i=1}^{N}\frac{E_{gi} \cdot E_o}{\|E_{gi}\|\|E_o\|}$$

where E_{gi} is the embedding of the i^{th} generated question E_o is the embedding of the user input, and N is the number of generated questions (default is 3).

- Semantic Similarity: evaluates how closely the meaning of the generated answer

matches the ground truth answer. To calculate semantic similarity, both the generated answer and the ground truth are converted into vector representations using an embedding model. The cosine similarity between these two vectors is then computed, producing a score between 0 and 1, where higher values indicate stronger semantic alignment between the two answers.

- Answer Correctness: measures how accurately a generated answer matches the ground truth or reference answer, with scores ranging from 0 to 1. Higher scores indicate a closer match between the response and the ground truth, reflecting better correctness. To assess answer correctness, the evaluation considers both factual overlap and semantic similarity between the generated answer and the reference answer. Factual correctness is determined by identifying true positives (facts present in both answers), false positives (facts present only in the generated answer), and false negatives (facts present only in the ground truth). The F1 score is then used to quantify the factual similarity based on these values.

$$\text{F1 Score} = \frac{|\text{TP}|}{(|\text{TP}| + 0.5 \times (|\text{FP}| + |\text{FN}|))}$$

The final answer correctness score is a weighted average of the factual similarity (F1 score) and semantic similarity.

3.6.2. HUMAN EXPERT EVALUATION

While automated metrics offer valuable quantitative benchmarks, they cannot fully address how well the chatbot meets the nuanced and practical needs of its intended audience. To complement automated assessment, six domain experts are invited to interact directly with the Malus Chatbot, posing their own domain-specific questions related to apple cultivation. After using the chatbot, they complete a structured survey evaluating the system's performance across a range of criteria.

Experts are asked to rate each response based on the following aspects:

- Relevance: Whether the response effectively addresses the user's query.
- Accuracy: The correctness of the information provided and its alignment with established agricultural practices.
- Fluency: The grammatical correctness, clarity, and readability of the answer.
- Coherence: The logical flow and completeness of the response.
- Usefulness: How helpful the chatbot is in enabling users to find and understand information about apple cultivation.

- Efficiency: Whether the chatbot reduces the time and effort required to retrieve agricultural information compared to traditional methods.
- Overall Satisfaction: The general satisfaction of the evaluator with the chatbot's performance and user experience.

Each aspect is rated using a standardized scale, from 1 to 5, with experts also providing qualitative feedback to highlight strengths or suggest areas for improvement. The survey questions can be found in Appendix C.

4. RESULTS AND DISCUSSION

Following the methodology outlined in the previous chapter, this section presents the results of evaluating the Malus Chatbot using both automated metrics and human expert feedback. The dual approach ensures a balanced assessment, combining objective quantitative indicators with subjective, domain-informed insights.

4.1. AUTOMATED EVALUATION

As mentioned, the automated evaluation of the Malus Chatbot was conducted using the dataset containing 86 sets of expert-validated, domain-specific questions and reference answers. For all experiments in this section, the evaluation model was kept constant: RAGAS metrics were calculated using GPT-4.1-mini as the scoring model, ensuring consistency across setups.

While there are many parameters that can be tuned in a retrieval-augmented generation system, this evaluation focuses on those that are expected to have the greatest impact on answer quality and overall system effectiveness. In particular, we systematically evaluate the following main parameters:

- Retrieval algorithm: Determines the method used to select relevant context chunks. This is a foundational parameter, as it directly controls which information is provided to the language model.
- Number of retrieved documents (k): Sets how many context chunks are supplied to the model, affecting coverage and precision.
- Temperature: Controls the variability of the language model's output, balancing determinism and creativity.
- Embedding model: Impacts semantic retrieval quality, particularly important for multilingual and technical content.
- Answer generation model: Determines the fluency, factuality, and synthesis ability of the chatbot's responses.

The evaluation follows a sequential approach, beginning with an initial base configuration and systematically varying one parameter at a time to assess its effect on chatbot performance. After identifying the best-performing value for a parameter, this value is adopted for subsequent experiments as the new default. This process is repeated for each parameter in turn, enabling a stepwise optimization and fair assessment of each factor's impact.

The initial base configuration is shown in Table 2.

Table 2 – Initial parameter configuration

Parameter	Value
Retrieval algorithm	Cosine similarity (ChromaDB default)
No. of Retrieved Documents	5
Temperature	1 (OpenAI default)
Embedding model	text-embedding-3-large
Generation model	GPT-4.1

In the sections that follow, each parameter is optimized in order, beginning with the retrieval algorithm, to systematically identify the configuration that delivers the best performance for the Malus Chatbot.

4.1.1. RETRIEVAL ALGORITHM

To assess the impact of retrieval strategy, we compared three algorithms: cosine similarity, L2 distance, and a hybrid approach combining dense retrieval with sparse retrieval, with equal weight. For the hybrid approach, the dense retrieval algorithm used is the one that yields better results between cosine similarity and L2, and the sparse retrieval algorithm used is BM25.

In the context of retrieval algorithm comparison, Context Precision, and Context Recall are the most informative metrics, as they directly reflect the quality of the documents retrieved in the answer generation step. The other metrics capture the overall effectiveness of the pipeline but may be more strongly influenced by the language model's generation capabilities.

The results are shown in Table 3.

Table 3 – Retrieval algorithm results comparison

Metric	Cosine	L2	Hybrid w/ Cosine
Context Precision	0.653	0.631	0.675
Context Recall	0.863	0.822	0.880
Faithfulness	0.948	0.930	0.958
Answer Relevancy	0.865	0.858	0.875
Semantic Similarity	0.919	0.909	0.920
Answer Correctness	0.511	0.507	0.557

Comparing cosine and L2 retrieval, cosine similarity performed better in all the evaluation metrics, so we select it as the dense algorithm for the hybrid approach.

Using the hybrid approach with cosine, we see that it achieves even higher scores across all metrics, confirming the effectiveness of combining semantic understanding with precise keyword matching. Thus, we select hybrid with cosine as our default retrieval strategy moving forward.

4.1.2. NUMBER OF RETRIEVED DOCUMENTS

The number of retrieved documents (*k*) determines how much context is provided to the language model for each question. This parameter directly affects the balance between answer completeness and the risk of including irrelevant information. In this section, we compare the performance of the retrieval pipeline using three commonly used values for *k*: 3, 5, 10, and 15. These values were selected because a smaller *k* may result in insufficient context, while a larger *k* could introduce more irrelevant content or exceed the model's context window.

The results are shown in Table 4.

Table 4 – Number of retrieved documents results comparison

Metric	**k = 3**	**k = 5**	**k = 10**	**k = 15**
Context Precision	0.703	0.675	0.647	0.614
Context Recall	0.845	0.880	0.929	0.933
Faithfulness	0.969	0.958	0.966	0.960
Answer Relevancy	0.867	0.875	0.917	0.921
Semantic Similarity	0.918	0.920	0.922	0.922
Answer Correctness	0.492	0.557	0.601	0.598

We observe that increasing k leads to notable improvements in context recall (from 0.845 at k = 3 to 0.933 at k = 15) and answer relevancy (from 0.867 at k = 3 to 0.928 at k = 15). This makes sense since retrieving more documents increases the likelihood of capturing all relevant context and resulting in more relevant answers.

As k increases, we also witness a decrease in context precision (dropping from 0.703 at k = 3 to 0.614 at k = 15), which is also expected, since retrieving more documents inherently leads to more irrelevant documents. However, despite more irrelevant documents being present, faithfulness and semantic similarity values remain consistently high across all tested k values, leading us to believe that the language model is good at keeping hallucinations low even with the increase in information.

Answer correctness, on the other hand, while increasing with k initially, slightly decreases from k = 10 to k = 15. This, along with the observation that context recall and answer relevancy also plateau when reaching these two k values, leads us to conclude that k = 10 is the best value going forward, offering an optimal balance between retrieval coverage, answer quality, and efficiency.

4.1.3. EMBEDDING MODEL

As detailed in Section 3.4, the text-embedding-3-large model from OpenAI was initially chosen over the text-embedding-ada-002 model due to its reported superior multilingual capabilities, critical for the multilingual nature (Portuguese, English, Spanish) of our dataset. Given this

initial justification, it remains important to confirm the performance benefits of the selected embedding model. This section therefore provides a direct comparison between text-embedding-3-large and the earlier, more cost-effective text-embedding-ada-002 model to validate whether the theoretically anticipated advantages are reflected in practical retrieval and answer-quality metrics.

The results are shown in Table 5.

Table 5 – Embedding model results comparison

Metric	text-embedding-ada-002	text-embedding-3-large
Context Precision	0.518	0.647
Context Recall	0.859	0.929
Faithfulness	0.949	0.966
Answer Relevancy	0.866	0.917
Semantic Similarity	0.918	0.922
Answer Correctness	0.537	0.601

The results confirm that the text-embedding-3-large model outperforms text-embedding-ada-002 across all evaluated metrics. These findings validate the initial methodological decision to use text-embedding-3-large, and this model will therefore be retained as the default embedding configuration in all subsequent experiments.

4.1.4. TEMPERATURE

The temperature parameter in language generation controls the randomness of the model's output. Lower temperature values (e.g., 0.0 to 0.3) produce more deterministic and consistent responses by favoring high-probability tokens. Higher temperatures (e.g., above 0.7) introduce more variation, potentially improving fluency or creativity at the cost of increased unpredictability and factual drift.

In RAG, lower temperatures are generally expected to yield better performance. The objective is not to generate creative or varied text but to produce responses that are grounded in retrieved documents. Higher temperatures may increase the risk of hallucination or introduce deviations from the source context. This assumption is reflected in previous work, where deterministic or low-temperature decoding is commonly used to preserve factual consistency (Lewis et al., 2020; Guu et al., 2020).

The results are shown in Table 6.

Table 6 – Temperature results comparison

Metric	0.00	0.25	0.50	0.75	1.00
Context Precision	0.645	0.652	0.644	0.649	0.647
Context Recall	0.929	0.933	0.927	0.932	0.929
Faithfulness	0.964	0.962	0.959	0.962	0.966
Answer Relevancy	0.915	0.918	0.920	0.913	0.917
Semantic Similarity	0.923	0.925	0.922	0.927	0.922
Answer Correctness	0.604	0.602	0.596	0.600	0.601

Surprisingly, the results show no noticeable difference or trends in performance across temperature settings, indicating that higher temperatures do not lead to a measurable increase in hallucination or reduction in answer quality. This suggests that, within this RAG configuration, the model remains stable across a range of temperature values.

Given these findings, and following established practices in prior work where lower temperatures are commonly used to promote factual consistency and reduce hallucination, we choose temperature = 0.25 as the default. It offers a balanced setting that aligns with standard recommendations for retrieval-augmented generation systems.

4.1.5. Answer Generation Model

After optimizing the retrieval pipeline, we compared three variants of the GPT-4.1 language model offered by OpenAI to assess their impact on answer generation quality: the standard GPT-4.1, the smaller and more cost-effective GPT-4.1-mini, and the lightweight GPT-4.1-nano. The goal was to evaluate the trade-offs between output quality and computational efficiency and to determine whether the smaller models could deliver acceptable performance for domain-specific question answering. GPT-4.1-mini is approximately one-fifth the cost of the full GPT-4.1 model, while GPT-4.1-nano is just one-quarter the cost of GPT-4.1-mini, making them appealing options for scalable or resource-constrained applications if their lower cost does not come at the expense of a significant drop in performance.

The results are shown in Table 7.

Table 7 – Answer generation model results comparison

Metric	GPT-4.1	GPT-4.1-mini	GPT-4.1-nano
Context Precision	0.652	0.650	0.652
Context Recall	0.933	0.934	0.931
Faithfulness	0.962	0.955	0.888
Answer Relevancy	0.918	0.911	0.857
Semantic Similarity	0.925	0.922	0.915
Answer Correctness	0.602	0.597	0.524

Surprisingly, GPT-4.1-mini shows no meaningful decline in performance compared to GPT-4.1 on the most relevant generation metrics, with differences being marginal and not justifying the substantially higher cost of using the full GPT-4.1 model. One possible explanation is that, for many RAG tasks, the retrieval process delivers sufficiently informative context, allowing even a lighter model to generate high-quality, well-grounded answers without significant loss of fluency or factual accuracy. Additionally, model compression and distillation techniques used in developing the mini variants may effectively preserve core capabilities essential for most RAG scenarios, especially when the complexity of reasoning required is modest.

In contrast, GPT-4.1-nano demonstrates a noticeable drop in performance in areas where the generation model plays a more central role, making it less suitable for applications that demand reliable, well-grounded answers. This could be due to further reductions in model capacity, which limit its ability to synthesize nuanced responses or handle more complex queries, especially when the retrieved context is less explicit or requires greater language understanding to interpret.

Given its strong performance, nearly matching GPT-4.1 at a much lower cost, GPT-4.1-mini was selected as the default generation model.

The optimal configuration identified through these experiments was adopted for deploying the Malus Chatbot on Streamlit and are displayed in Table 8.

Table 8 – Final parameters configuration

Parameter	**Value**
Retrieval algorithm	Hybrid w/ Cosine
No. of Retrieved Documents	10
Temperature	0.25
Embedding model	text-embedding-3-large
Generation model	GPT-4.1-mini

Overall, our system achieved consistently high values across most evaluation metrics, demonstrating strong overall performance, with the exceptions being context recall and answer correctness. However, context recall is less critical in this scenario, since the model can correctly identify relevant documents. For answer correctness, further analysis was conducted through manual comparison between the generated answers and the ground truth. This closer inspection revealed two key scenarios contributing to the lower scores. In many cases, the generated answers included the facts from the ground truth but also contained additional information. These extra facts were relevant to the question and verifiably present in the retrieved documents. In other cases, the generated answer did not exactly match the ground truth, but the information provided was still well supported by the retrieved sources and relevant.

This discovery highlights an important consideration: if the ground truth is based on a single document, but there are several other correct and relevant facts distributed across different documents, it is inherently challenging for the retrieval component to always surface the exact fact the ground truth expects. In practice, there may be multiple valid answers to a question, and a robust retrieval system will often return a broader set of relevant and accurate facts rather than strictly matching the ground truth. Therefore, even when some ground truth facts are missing from the generated answer, the retrieval may still be performing well by presenting a diverse and accurate set of facts from the available corpus. This reveals a limitation of evaluation methods that rely on single-reference ground truths, as they may not fully capture the range of acceptable answers produced by an effective RAG system.

As a result, automated answer correctness metrics in RAGAS, which primarily measure overlap with the reference answer, may underestimate the true performance of the model when the ground truth is incomplete or overly narrow. However, since both faithfulness and answer relevancy metrics remain high, this strongly indicates that the generated answers are well grounded in the retrieved context and directly relevant to the posed questions. Therefore, despite the lower answer correctness scores, the outputs can still be considered accurate and reliable for practical use. This finding highlights the importance of supplementing automated evaluation with real-time expert assessment to provide a more comprehensive and trustworthy judgment of answer quality and factual consistency.

4.2. HUMAN EXPERT EVALUATION

To complement the automated evaluation and address the limitations of single-reference answer metrics, a human expert assessment was conducted. Six agricultural specialists with expertise in apple cultivation in Portugal were invited to interact directly with the deployed Malus Chatbot, with the evaluation as described in Section 3.6.2. This evaluation not only assesses the quality and relevance of the chatbot's answers, but also measures user experience and satisfaction with the interface, including aspects such as conversation coherence, response speed, and the helpfulness of images and source attributions. The results are shown in Table 9.

Table 9 – Human expert evaluation results

Question	Average Score (1-5)
The answers provided by the chatbot are relevant to the questions asked	3.7
I am confident in the information provided by the chatbot	3.5
The answers provided are sufficiently detailed and cover all important aspects of the question	3.5
The sources provided contain information that supports the answers	3.5
The images provided by the chatbot help with the answer	3.5
The chatbot is able to maintain a coherent conversation, understanding the context between questions	3.7
The chatbot responds quickly to questions	3.8
I am very satisfied with the experience of this chatbot	3.7
I recommend this chatbot to other users	3.7

With six expert participants, the evaluation results indicate a generally positive user experience. The highest scores were recorded for response speed (3.8), coherence of conversation (3.7), relevance of answers (3.7), overall satisfaction (3.7), and likelihood of recommending the chatbot to others (3.7). Scores for confidence in the information, detail and completeness of answers, supporting sources, and the helpfulness of images all reached 3.5, suggesting that while the chatbot performs well, there is some room for further improvement in these areas.

On the general comments, experts noted that the chatbot was able to maintain relevant and coherent conversations, effectively utilizing supporting images and source documents. They also appreciated the clear and interactive interface, which contributed positively to the overall user experience. However, they observed that some responses could be further enhanced in detail and coverage, and suggested that expanding the knowledge base would help address certain gaps in the information provided.

Overall, the human evaluation corroborates the findings from automated metrics, demonstrating that the Malus Chatbot delivers relevant, well-supported, and timely answers for apple cultivation in Portugal. The results also confirm the value of integrating images and source attributions within chatbot responses, as well as the importance of an accessible and responsive user interface. These findings highlight the system's practical utility for real-world advisory applications, while also pointing to future opportunities for enhancing answer depth, expanding the knowledge base, and improving source transparency

Comparing these results to previous work, it is important to recognize the significant differences in scope, methodology, and evaluation design. Most earlier agricultural chatbots, especially those based on rule-based or intent-driven architectures (such as Mostaço et al., 2018; Ong et al., 2021; Momaya et al., 2021), are typically evaluated on closed datasets with a fixed set of questions and limited domain coverage. These systems often report very high accuracy, but this is mainly because they are evaluated under very controlled and predictable conditions, meaning they are not designed to handle more open-ended or complex questions, nor can they adapt to new or evolving information outside of what has already been programmed.

More recent works involving AI-based and RAG systems, including studies by Silva et al. (2023) and Balaguer et al. (2024), have taken important steps forward. However, these systems are generally trained and tested on structured datasets such as multiple-choice or exam-style question sets. Although the reported accuracy in these cases can be as high as 93 percent, the questions themselves are generally standardized and the evaluation criteria are quite rigid, with answers matched to a single reference, meaning these studies may not fully capture the realities and unpredictability of actual information needs in agriculture.

In contrast, this project evaluated the Malus Chatbot under conditions designed to closely mirror real-world agricultural information needs. Both the knowledge base and the expert-validated evaluation dataset are open-domain, reflecting the practical and varied questions encountered by apple growers in Portugal. This approach goes beyond the narrowly defined and standardized datasets used in much of the prior literature, making direct comparison of accuracy metrics less straightforward but providing a much more realistic test of system performance.

Furthermore, the Malus Chatbot was deployed as an interactive web application, enabling domain experts to engage with it directly and assess its effectiveness in genuine advisory situations. The system also introduces several features not found in earlier agricultural chatbots, such as the integration of images within conversations and the ability for users to download and verify the original documents and source files. These capabilities enhance transparency, support verification, and contribute to a richer and more practical user experience.

Taken together, the use of open-domain training and evaluation data, real-world deployment, and an emphasis on user interaction, along with the integration of images and direct access to original sources, distinguish the Malus Chatbot from prior systems. These innovations highlight the system's adaptability and practical value for agricultural advisory applications, setting a new benchmark for realistic, user-focused chatbot development and evaluation in this field.

5. CONCLUSIONS

This thesis presented a comprehensive evaluation and optimization of the Malus Chatbot, a RAG system designed for domain-specific question answering in the field of apple tree cultivation in Portugal. Through systematic experimentation and parameter tuning, the final configuration of the chatbot demonstrates a robust balance between answer quality, efficiency, and cost, supporting its practical value for agricultural stakeholders and practitioners.

One of the most important findings of this work is that the performance of a RAG system relies heavily on optimizing the retrieval process. The adoption of a hybrid retrieval algorithm, combining dense semantic and sparse lexical methods, consistently yielded superior results, emphasizing the benefit of leveraging multiple retrieval approaches for improved recall and precision. The study of the number of retrieved documents (k) revealed a classic trade-off: while increasing k led to higher context recall and answer relevancy, it also reduced context precision and slightly diminished answer correctness at the highest values. The optimal value was found at k = 10, which balances sufficient context coverage with minimal introduction of irrelevant information, providing a practical guideline for future RAG deployments.

Another important parameter explored was the temperature setting for language model decoding. Contrary to common assumptions, varying the temperature within a reasonable range did not lead to significant changes in answer quality or hallucination rates. This finding suggests that, at least within the tested RAG configuration, the language model maintains stable performance across a range of temperature values, allowing practitioners some flexibility in tuning this parameter without risk of degrading factuality. Nevertheless, the adopted default of temperature = 0.25 aligns with best practices for factual consistency in retrieval-augmented generation.

The comparison of answer generation models identified GPT-4.1-mini as the optimal choice for this application, achieving nearly the same output quality as the full GPT-4.1 model at a fraction of the cost and computational requirements. In contrast, the lighter GPT-4.1-nano variant exhibited a more pronounced drop in performance, especially in scenarios requiring nuanced synthesis or when the retrieved context was less explicit.

The automated evaluation of the optimized system produced high scores across most metrics. Manual analysis showed that lower answer correctness scores were often due to the limitations of single-reference ground truth, not to errors in the chatbot's responses. The chatbot frequently provided additional, accurate, and contextually supported facts, highlighting the strengths of retrieval-based aggregation in agricultural domains where multiple valid answers from different sources may exist.

The role of human expert evaluation further validated these results. Six specialists interacted directly with the deployed Malus Chatbot, confirming its ability to deliver relevant, well-

supported answers and a positive user experience. They also highlighted the value of integrated images within conversations and direct access to original source documents, both of which enhanced clarity, transparency, and user trust, features not present in previous agricultural chatbots.

In summary, this work demonstrates that careful optimization of retrieval and generation parameters can yield a reliable, efficient, and context-aware RAG-based chatbot tailored to apple tree cultivation in Portugal. The combination of open-domain knowledge, real-world deployment, user-focused interface, and advanced features such as image integration and source transparency sets a new benchmark for practical chatbot applications in agriculture. The Malus Chatbot offers a promising tool for improving access to agricultural expertise and supporting informed decision-making among Portuguese apple growers, while also providing guidance for future conversational AI deployments in specialized domains.

6. LIMITATIONS AND FUTURE WORK

Despite the strong performance demonstrated by the Malus Chatbot, several important limitations remain that present opportunities for further development and research.

A key limitation arises from the current evaluation methodology, particularly the reliance on single-reference ground truth answers for assessing answer correctness. In many cases, the chatbot's responses included additional, accurate, and relevant information not present in the ground truth, or provided alternative facts that were still well supported by the retrieved documents. As a result, answer correctness scores can be artificially low, underestimating the system's true capabilities. This limitation highlights the need for more nuanced evaluation protocols, such as multi-reference ground truths or metrics that recognize the validity of diverse, contextually grounded answers. Implementing these approaches would provide a more realistic assessment of performance, especially in domains like agriculture, where expertise is distributed across numerous and sometimes inconsistent sources.

However, this challenge goes beyond evaluation and reflects an inherent feature of RAG systems using large, diverse knowledge bases: information from different sources may naturally yield a range of answers, some complementary and others potentially contradictory. While this diversity enriches the system's responses, it also creates the need to prioritize, reconcile, or clearly communicate uncertainties and conflicts to users. In this work, source attribution was implemented to increase transparency and help users assess the credibility of information. Nonetheless, further development is required to systematically detect conflicting statements, highlight areas of consensus or disagreement, and convey the degree of certainty present. Enhancing these capabilities will be essential for building user trust and supporting informed decisions when working with complex and sometimes inconsistent knowledge sources.

Another area for improvement lies in the integration of structured data such as Excel and CSV files. While the current approach of converting sheets and chunks into markdown-formatted Document objects enables their inclusion in the knowledge base, it does not fully preserve the semantics and relationships within complex tables. Advanced table understanding methods, such as specialized embeddings for tabular structures (Hegselmann et al., 2023) or hybrid pipelines combining language models with structured data parsers (Yin et al., 2020), should be explored to address these challenges.

Image retrieval and integration is a further area where enhancements can be made. While the current system can present images relevant to the user's query, there is potential to improve the selection, ranking, and contextual relevance of retrieved images. Techniques such as multimodal embeddings, advanced image-text alignment (Guo et al., 2023), or incorporating visual question answering (VQA) models (Huynh et al., 2025) could be explored to ensure that images presented alongside answers are both accurate and truly informative for the user.

Improving image retrieval in this way would further enrich the user experience and strengthen the practical value of the chatbot.

Additionally, the project has relied exclusively on OpenAI's language models for both answer generation and embeddings. Although these models offer strong performance, future work should include benchmarking of alternative solutions from other leading AI and generative language model providers, as well as exploring cutting-edge open-source models. Although earlier sections discussed the current challenges of open-source LLMs, the ongoing improvements in these models may soon make them more competitive, particularly for teams seeking greater transparency, lower operational costs, or increased control over data privacy. As agricultural AI applications scale or move into public-sector or privacy-sensitive contexts, these trade-offs will become increasingly important to revisit. Comparative experiments with a broader range of models, using the same rigorous evaluation pipeline established here, will be essential for identifying the most suitable solutions for future deployments.

In summary, future research should focus on more comprehensive evaluation strategies, improved handling of conflicting and redundant information, enhanced integration of structured data, broader exploration of language model options, and human expert-driven assessment. Addressing these areas will further strengthen the Malus Chatbot and advance retrieval-augmented conversational AI for complex domains like agriculture.

BIBLIOGRAPHIC REFERENCES

Adamopoulou, E., & Moussiades, L. (2020). An overview of chatbot technology. In Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16 (pp. 373-383). Springer International Publishing.

Adiwardana, D., Luong, M. T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., ... & Le, Q. V. (2020). Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

AWS, (n.d.). *What is retrieval-augmented generation?* Retrieved June 17, 2025, from https://aws.amazon.com/what-is/retrieval-augmented-generation/

Balaguer, A., Benara, V., de Freitas Cunha, R. L., Estevão Filho, R. D. M., Hendry, T., Holstein, D., ... & Chandra, R. (2024). RAG vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. *arXiv e-prints*, arXiv-2401.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Banerjee, S., Akhila, P., Elakiya, E., & Surendiran, B. (2024, July). CropBot: A Chatbot for Agriculture Advancement. In *2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)* (pp. 1-6). IEEE.

Belval, M. (2024). pdf2image. https://github.com/Belval/pdf2image

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157-166.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1798-1828.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J., ... & Roossin, P. S. (1990). A statistical approach to machine translation. *Computational linguistics*, *16*(2), 79-85.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, *19*(2), 263-311

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

Compal. (n.d.). *Maçã*. Retrieved June 17, 2025, from https://www.compal.pt/origem-das-frutas/maca/

Chandolikar, N., Dale, C., Koli, T., Singh, M., & Narkhede, T. (2022, March). Agriculture assistant Chatbot using artificial neural network. In *2022 International Conference on Advanced Computing Technologies and Applications (ICACTA)* (pp. 1-5). IEEE.

Chen, S. F., & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, *13*(4), 359-394.

Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. *Advances in neural information processing systems*, *28*.

Chroma. (2023). Chroma: The AI-native open-source embedding database. https://www.trychroma.com/

Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dinan, E., Logacheva, V., Malykh, V., Miller, A., Shuster, K., Urbanek, J., ... & Weston, J. (2020). The second conversational intelligence challenge (convai2). In The NeurIPS'18 Competition: From Machine Learning to Intelligent Conversations (pp. 187-208). Springer International Publishing.

Es, S., James, J., Anke, L. E., & Schockaert, S. (2024, March). Ragas: Automated evaluation of retrieval augmented generation. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (pp. 150-158)

Ekanayake, J., & Saputhanthri, L. (2020). E-AGRO: Intelligent chat-bot. IoT and artificial intelligence to enhance farming industry. *Agris on-line Papers in Economics and Informatics*, *12*(1), 15-21.

Fraga, H., & Santos, J. A. (2021). Assessment of climate change impacts on chilling and forcing for the main fresh fruit regions in Portugal. *Frontiers in plant science*, *12*, 689121.

Gaddikeri, V., Jatav, M. S., & Rajput, J. (2023). Revolutionizing agriculture: unlocking the potential of ChatGPT in agriculture. *Food and Scientific Reports*.

Google (2025a). Tesseract OCR engine. https://github.com/tesseract-ocr/tesseract

Google (2025b). Gemini 2.5 Pro. https://deepmind.google/models/gemini/pro/

Guo, R., Wei, J., Sun, L., Yu, B., Chang, G., Liu, D., ... & Bu, L. (2023). A survey on image-text multimodal models. *arXiv preprint arXiv:2309.15857*.

Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020, November). Retrieval augmented language model pre-training. In *International conference on machine learning* (pp. 3929-3938). PMLR.

Hegselmann, S., Buendia, A., Lang, H., Agrawal, M., Jiang, X., & Sontag, D. (2023, April). Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics* (pp. 5549-5581). PMLR.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Hutchins, J. (2005). The first public demonstration of machine translation: the Georgetown-IBM system, 7th January 1954. *noviembre de*.Torfi, Amirsina, et al "Natural language processing advancements by deep learning: A survey." *arXiv preprint arXiv:2003.01200* (2020).

Huynh, N. D., Bouadjenek, M. R., Aryal, S., Razzak, I., & Hacid, H. (2025). Visual question answering: from early developments to recent advances--a survey. *arXiv preprint arXiv:2501.03939*.

IBM. (2024). What is Natural Language Processing? https://www.ibm.com/topics/natural-language-processing

Izacard, G., & Grave, E. (2020). Leveraging passage retrieval with generative models for open domain question answering. arXiv preprint arXiv:2007.01282.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., ... & Fung, P. (2023). Survey of hallucination in natural language generation. ACM computing surveys, 55(12), 1-38.

Jing, Z., Su, Y., & Han, Y. (2025, February). When large language models meet vector databases: A survey. In *2025 Conference on Artificial Intelligence x Multimedia (AIxMM)* (pp. 7-13). IEEE.

Jurafsky, D., & Martin, J. H. (2023). Speech and language processing (3rd ed., draft). Stanford University. https://web.stanford.edu/~jurafsky/slp3/

Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (pp. 137-142). Berlin, Heidelberg: Springer Berlin Heidelberg.

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535-547.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., ... & Yih, W. T. (2020). Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Lafferty, J., McCallum, A., & Pereira, F. (2001, June). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Icml* (Vol. 1, No. 2, p. 3).

LangChain (2025). LangChain Python library. https://github.com/langchain-ai/langchain

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, *33*, 9459-9474.

Liu, C. W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., & Pineau, J. (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*(pp. 74-81).

Mikolov, T., Chen, K., Corrado, G., Dean, J. (16 January 2013).Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, *3781*.

Mistral AI. (2024). Mixtral 8x22B Technical Report. Mistral AI Documentation. https://mistral.ai/news/mixtral-8x22b

Momaya, M., Khanna, A., Sadavarte, J., & Sankhe, M. (2021, June). Krushi–the farmer chatbot. In *2021 International Conference on Communication information and Computing Technology (ICCICT)* (pp. 1-6). IEEE.

Mostaco, G. M., De Souza, I. R. C., Campos, L. B., & Cugnasca, C. E. (2018, June). AgronomoBot: a smart answering Chatbot applied to agricultural sensor networks. In *14th international conference on precision agriculture* (Vol. 24, pp. 1-13).

Ong, R. J., Raof, R. A. A., Sudin, S., & Choong, K. Y. (2021, February). A review of chatbot development for dynamic web-based knowledge management system (KMS) in small scale agriculture. In *Journal of Physics: Conference Series* (Vol. 1755, No. 1, p. 012051). IOP Publishing.

OpenAI. (n.d.). Text embedding models. https://platform.openai.com/docs/guides/embeddings

OpenAI. (2024a). *Introducing GPT-4o: Faster, cheaper, smarter*. https://openai.com/index/hello-gpt-4o/

OpenAI. (2024b). *New embedding models and API updates*. https://openai.com/blog/new-embedding-models-and-api-updates

OpenAI. (2025a). *Introducing GPT-4.1 in the API*. https://openai.com/index/gpt-4-1/

OpenAI. (2025b). *Reasoning models*. https://platform.openai.com/docs/guides/reasoning?api-mode=responses

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257-286.

Roberts, A., Raffel, C., & Shazeer, N. (2020). How much knowledge can you pack into the parameters of a language model?. *arXiv preprint arXiv:2002.08910*.

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval, 3(4), 333-389.

Shawar, B. A., & Atwell, E. (2007). Chatbots: are they really useful?. Journal for Language Technology and Computational Linguistics, 22(1), 29-49.

Shen, Z., Zhang, R., Dell, M., Lee, B. C. G., Carlson, J., & Li, W. (2021). Layoutparser: A unified toolkit for deep learning based document image analysis. In Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I 16 (pp. 131-146). Springer International Publishing.

Shuster, K., Poff, S., Chen, M., Kiela, D., & Weston, J. (2021). Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Silva, B., Nunes, L., Estevão, R., Aski, V., & Chandra, R. (2023). GPT-4 as an agronomist assistant? Answering agriculture exams using large language models. *arXiv preprint arXiv:2310.06225*.

Singer-Vine, J. (2025). pdfplumber: Python PDF text extraction. https://github.com/jsvine/pdfplumber

Suebsombut, P., Sureephong, P., Sekhari, A., Chernbumroong, S., & Bouras, A. (2022, January). Chatbot application to support smart agriculture in Thailand. In *2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)* (pp. 364-367). IEEE.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27.

Turing, A. M. (2009). Computing machinery and intelligence (pp. 23-65). Springer Netherlands.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

USDA, Foreign Agricultural Service. (2021, June 21). *Portuguese fruit sector aims to increase investments efficiency and exports* (GAIN Report PT20210017)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems* (p./pp. 5998--6008)

Vinyals, O., & Le, Q. (2015). A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Wirth, R., & Hipp, J. (2000, April). CRISP-DM: Towards a standard process model for data mining. In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (Vol. 1, pp. 29-39).

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36-45.

Yin, P., Neubig, G., Yih, W. T., & Riedel, S. (2020). TaBERT: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine, 13(3), 55-75.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2025). Detectron2. https://github.com/facebookresearch/detectron2

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

APPENDIX A

System prompt

"You are a scientific assistant specialized in apple cultivation.

Use the following documents as context to answer the question accurately and clearly.

The documents may be written in different languages, but you MUST respond only in the same language as the user's question (mostly English or European Portuguese).

If the context does not contain a complete answer, say so — do not make up information.

Your response MUST be a single, valid JSON object containing the following keys:

- "answer": A string containing the answer in Markdown format.

- "images": A list of JSON objects, where each object has a "display_name" (string) and "image_path" (string). This list should be empty if no figures/tables are explicitly mentioned in the answer.

- "sources": A list of JSON objects, where each object has a "file_name" (string), "page" (string or null, containing page specifier like "page 5" or "pages 5, 6", or null if none), and "source_path" (string). This list should be empty if no documents were directly used.

IMPORTANT: Output only the JSON object. Do not include any extra text, explanations, or formatting before or after the JSON.

Example JSON output:

```
{ "answer": "Answer text...",
  "images": [ {"display_name": "figure_001", "image_path": "images/figure_001.png"} ],
  "sources":[
     {"file_name": "manual.pdf", "page": "page 4", "source_path": "pdfs/manual.pdf"},
    {"file_name": "apples_guide.pdf", "page": "pages 4, 10", "source_path": "pdfs/apples_guide.pdf"},
    {"file_name": "apple_stats.csv", "page": null, "source_path": "csvs/apple_stats.csv"}
]
}
```

You MUST follow this exact 3-part structure using triple-bracket tags to mark each section clearly: "

APPENDIX B

Example of unparsed response

"{

"answer": "The Bravo de Esmolfe apple has the following characteristics:\n- Size: Medium to small caliber.\n- Shape: Oblong-conical.\n- Skin: Whitish epidermis, sometimes with reddish spots, mottled and/or streaked, especially around the stem cavity, which can cover up to 20% of the skin.\n- Flesh: White, soft, juicy, sweet, with good taste qualities.\n- Aroma: Intense, pleasant, and quite unique (\"sui generis\").\n\nThese features make the Bravo de Esmolfe apple highly appreciated by consumers and contribute to its Protected Designation of Origin (PDO) status.",

"images": [

{

"display_name": "Bravo de Esmolfe apple appearance",

"image_path": "extracted_images/produtos_tradicionais_principais_carateristicas_maca_bravo_esmolfe/figure_001.png"

},

{

"display_name": "Bravo de Esmolfe apples on tree",

"image_path": "extracted_images/relacoes_de_produtividade_area_folhear_e_alternancia_na_macieira_bravo_de_esmolfe/figure_004.png"

}

],

"sources": [

{

"file_name": "Produtos Tradicionais - Principais carateristicas Maca Bravo Esmolfe.pdf",

"page": "page 1",

"source_path": "Dados/Variedades/Produtos Tradicionais/Produtos Tradicionais - Principais carateristicas Maca Bravo Esmolfe.pdf"

```
    },
    {
      "file_name": "Relacoes de produtividade, area folhear e alternancia na macieira Bravo de Esmolfe.pdf",
      "page": "page 1",
      "source_path": "Dados/Cultivo/Relacoes de produtividade area folhear e alternancia na macieira Bravo de Esmolfe.pdf"
    }
  ]
}"
```

APPENDIX C

Survey Questions

Question	Type	Options
The answers provided by the chatbot are relevant to the questions asked	Scale	1 (Strongly Disagree) to 5 (Strongly Agree)
I am confident in the information provided by the chatbot	Scale	1 (Strongly Disagree) to 5 (Strongly Agree)
The answers provided are sufficiently detailed and cover all important aspects of the question	Scale	1 (Strongly Disagree) to 5 (Strongly Agree)
The chatbot provides sources for its answers	Scale	1 (Strongly Disagree) to 5 (Strongly Agree)
The sources provided contain information that supports the answers	Scale	1 (Strongly Disagree) to 5 (Strongly Agree)
The images provided by the chatbot help with the answer	Scale	1 (Strongly Disagree) to 5 (Strongly Agree)
The chatbot is able to maintain a coherent conversation, understanding the context between questions	Scale	1 (Strongly Disagree) to 5 (Strongly Agree)
The chatbot responds quickly to questions	Scale	1 (Strongly Disagree) to 5 (Strongly Agree)
I am very satisfied with the experience of this chatbot	Scale	1 (Strongly Disagree) to 5 (Strongly Agree)
I recommend this chatbot to other users	Scale	1 (Strongly Disagree) to 5 (Strongly Agree)
General comments	Open-ended	N/A

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa