Enhancing Crop Forecasting with Retrieval-Augmented Generation Systems

Stefan Stefanov

I. Abstract

Sustainable food production is a critical challenge of our era, driven by the increasing demand for plant-based diets, the impact of climate change on traditional agriculture, and the need for biodiversity in farming systems. This internship project focused on developing a robust Retrieval-Augmented Generation (RAG) database and implementing Extract, Transform, Load (ETL) processes to facilitate the forecasting of resilient and nutritionally valuable crops within the Brassicaceae family. By integrating data from a variety of sources, such as scientific literature. The project aimed to assist in the creation of a predictive analytics framework for identifying future crops suitable for sustainable agriculture in different locations. This report details the available technology, the methodologies used, and the results from the experiments with the newly developed software. The outcomes provide a helping hand for advanced analytics and decision-making in the domain of crop forecasting, contributing to the global pursuit of sustainable agricultural practices.

II. Introduction

The sustainable production of food has become one of the most pressing global challenges in the 21st century. Conventional agricultural practices, heavily reliant on monocultural farming and animal farming, face growing criticism due to their environmental impacts, including soil degradation, reduced biodiversity, and vulnerability to climate change. Addressing these issues requires a paradigm shift toward identifying new crops and farming methods that are resilient, sustainable, and adaptable to changing environmental conditions.

The Brassicaceae family, with its nearly 4,000 species adapted to a wide range of climates, presents a promising avenue for addressing these challenges. This project aimed to help harness the potential of advanced data integration and analytics to identify and forecast the success of crops within this family under future climatic scenarios. A key component of this effort involved the creation of a Retrieval-Augmented Generation (RAG) database, supported by robust ETL processes, to systematically collect, clean, and integrate data from diverse sources, such as available scientific literature.

This report outlines the research objectives, methodologies, and outcomes of the internship project. It showcases the technical implementation of the RAG database and ETL pipelines, provides insight into the state of the art on the topic at hand, and highlights the significance of advancing data-driven approaches for sustainable agriculture. The findings show an example of how data science and artificial intelligence can be used to address critical challenges in food production, fostering innovation in crop forecasting and agricultural resilience.

III. The approach

To build a robust system capable of addressing the goals of this project, a multi-phased approach was adopted. Each phase was designed to ensure systematic development, implementation, and evaluation of the key components, namely the Retrieval-Augmented Generation (RAG) database and the Extract, Transform, Load (ETL) processes to feed into it. To ensure each step is implemented in the best way possible at this time the following methodology was applied:

A. Research and Requirement Analysis

The initial phase of the project involved extensive research to define the problem scope and identify the system requirements. The focus was on understanding the agricultural challenges and what help is needed in the fight against them, as well as finding out what kind of technology out there can be used to efficiently create a RAG system, which would be well equipped to assist in solving the said challenges with the lowest cost possible. The discovered available technology is discussed further in the next section.

B. Data Acquisition and Integration

In this phase, the focus was on finding out what the best resources of data are and what kind of data is useful, as well as what formats of files are we most interested in. In order to get a working software prototype going, the focus had to be on just a few file types that were most common, while allowing for future improvements and additional implementations that would widen that scope. The file types I settled on are pdf files (most common), docx files (somewhat common), HTML files (useful for extracting knowledge from web pages). The next needed step was to gather data for tests during development and the later experiments of the system. To do so in an efficient manner, the Selenium web driver was used. This is an open-source tool used for web browser automation, perfect for creating a crawler that collects free literature regarding the Brassicaceae family, that is either useful for the final product or close enough to the type of useful literature that the RAG system might have in the future, in order to get accurate results from testing it in the current state and improving upon it further. The primary source of the literature was the Maastricht University Library, which covers a wealth of academic resources in a wide range, such as the

scientific papers and research articles we are interested in. To of course limit the amount of irrelevant files collected by the crawler, some keywords are used to filter the library's shown results, only retrieving files connected with Brassicaceae, plant taxonomy, genetics of Brassicaceae and ecological studies.

C. Develop RAG solution

This involves utilizing the best available technology to extract useful information from the accumulated data and structure it in a way that would be easily digestible by an LLM, once it is needed. Of course here is where we need a multitude of solutions for the different file types as well as the different structured files and the possible types of data stored within them, such as tables, images, unstructured text, lists, etc. As well as finding an appropriate vector database and embeddings generator to create and store the knowledge until it is needed and then efficiently retrieve only the useful information. The final step, of course involves the use of an LLM to answer a specific question by using the aquired knowledge from all of the literature stored in the vectorstore. This not only requires having a good solution but one that can easily be built upon. The LLM should be a modular part of the whole software and you should be able to replace it with another model in the future, should it become necessary. The final step involves a lot of fine tuning both for the creation, storage and retrieval of data, as well as prompt engineering to get the best results possible from the large language model.

D. Experiment and validate

This phase focused on testing the RAG system to evaluate its performance and validate its capabilities. The experiments aimed to measure how effectively the system could retrieve relevant information and provide accurate, contextually appropriate answers to specific queries related to the Brassicaceae family. The following steps were taken during this phase:

- Testing the ETL Pipeline: The pipeline's ability to extract, transform, and load data from various file types (PDF, DOCX, and HTML) was tested using a sample set from the accumulated files, ensuring that it works in a correct and efficient manner.
- Evaluating the Vector Database and the ability to retrieve the useful information that is needed for the correct answer of a prompt.
- LLM Validation: The large language model (LLM) was integrated into the system and tested with a set of predefined and random questions. This ensured the model effectively utilized the stored data to produce insightful and accurate responses without answering questions it has no specific sources for and sticking closely to the given context, limiting any attempts for hallucinations.
- Testing of the GUI and APIs: Ensuring that the software's GUI for interacting with the system and all the utilized APIs work as expected and bugs don't emerge.
- User Simulation Tests: Simulated user scenarios were conducted to replicate real-world use cases, such as researchers querying the system for specific information on Brassicaceae genetics or ecological adaptations.

E. Refine and improve

During the whole development phase as well as following the experimental phase, the system underwent iterative refinements to address arising issues, found during validations. The focus was on optimizing performance, enhancing usability, reducing cost, improving on user experience and ensuring scalability for future needs.

IV. Available technology

A. LLMs

Large Language Models (LLMs) represent a pivotal advancement in natural language processing, offering powerful capabilities for understanding and generating text. In recent years, models such as OpenAI's GPT series have set the benchmark for contextual understanding and high-quality output. For this project, GPT-4 was chosen as the primary LLM due to its consistent performance in handling complex queries, offering accurate and relatively expected outputs.

The decision to focus on GPT-4 was driven by its extensive pre-training, ability for contextual depth, and ability to process domain-specific questions effectively. However, recognizing the rapid evolution of AI, the system was designed with modularity in mind. This architecture ensures the seamless integration of other LLMs, such as newer GPT versions or alternative models like LLaMA or LLaVA, if required in the future. The modular design makes it easy to replace or supplement the current implementation, allowing the system to remain adaptable to advancements in LLM technology and support better or cheaper solutions as needed.

B. RAG tools

Retrieval-Augmented Generation (RAG) tools form the backbone of the system, enabling efficient data extraction, storage, and retrieval. Several key tools were employed to build and manage the RAG system:

- LangChain: (*LangChain*, n.d.) was utilized as the orchestration framework for integrating the LLM with the vector database. Its robust capabilities for chaining prompts and retrieval processes make it an ideal choice for RAG systems. It ensures seamless communication between components while allowing for the integration of custom logic tailored to the project's needs. In the current state of the software, the GPT model is integrated through langchain's library, and the vector database, though separated in its own module and being easily swappable, is also currently implemented through langchain's vector stores package. This way both parts are ensured to work together efficiently and synergistically.
- Unstructured.io: This library (*Unstructured.io*, n.d.) was crucial for parsing and processing documents of various formats, such as PDFs, DOCX files, and HTML. Its ability to decompose Unstructured data into meaningful components enabled the ETL pipeline to extract and

prepare data efficiently for vectorization and storage. Each extracted element is tagged with relevant metadata such as what type it is (title, narrative text, header, footer, figure footer, etc), what is the source of the text, and the ability to add additional information to the metadata if needed for a specific task. This metadata is stored within the embeddings of each element, ensuring it is available for further processing when it is retrieved from the vector store and utilized by the LLM. This metadata also allows for easier chunking, for example, the ability to chunk elements together by title of their paragraph or segment. Unstructured.io also allows for an easy implementation of OpenAI embeddings on the already extracted elements from the text, preparing the information for its final storage in the database.

- ChromaDB: (*ChromDB*, n.d.) served as the vector database, providing a free, efficient, and scalable solution for embedding storage and retrieval. With its focus on high performance and ease of use, ChromaDB allowed the system to handle complex queries with precision, retrieving only the most relevant information for use by the LLM. Currently, the provided software solution utilizes a retrieval strategy of 'k similar' documents, where k can be adjusted easily but by default is set to 5, which in testing gave the best results for the tried questions. The retrieval process itself is once again done through LangChain, using the ConversationalRetrievalChain method.

C. Object detection models and high-res extraction

High-resolution object detection models played a vital role in processing and understanding the structural components of PDF documents that might not be super straightforward or could even be scanned images of a document. For this purpose, two solutions were incorporated:

- YOLOX through Unstructured.io's High-Res Functionality: The Unstructured library provided seamless integration with YOLOX for object detection within high-resolution PDF documents. This approach involved segmenting pages into their components, such as text blocks, tables, and figures, and tagging them for further processing. After segmentation, the Tesseract OCR engine was used to extract text from these components accurately. At the end, the result is provided back into 'Unstructured elements' that are uniform throughout the suite of functions for extraction by Unstructured and can be further processed exactly the same as any simpler extraction process.
- Detectron2: As an alternative high-resolution object detection model, Detectron2 was employed to provide additional flexibility and precision at a lower "free" price. It is free in the sense that there is no third-party company charging you for the access to the model and its output. Detectron2 is trained specifically on segmenting document pages into their components and allowed for more complex detections, independant of the PDF document. Similar to the YOLOX setup, once the model extracts the separate parts of a page and tags them, Tesseract was used for OCR to ensure accurate text extraction. At the end, a custom solution is applied to convert all the extracted data into the unified elements that are expected by the rest of the system for further processing and final conversion into embeddings that can be stored and retrieved as needed.

Both solutions provided reliable methods for breaking down and analyzing PDF pages, ensuring the system could handle a variety of document formats and structures. These models contributed significantly to the accuracy of the ETL process by ensuring that all data, regardless of its layout and origin, was processed successfully and effectively.

V. Created software

The final software available for use consists of a GUI that is a simple chat window to talk to the bot, as shown in Fig. 1.

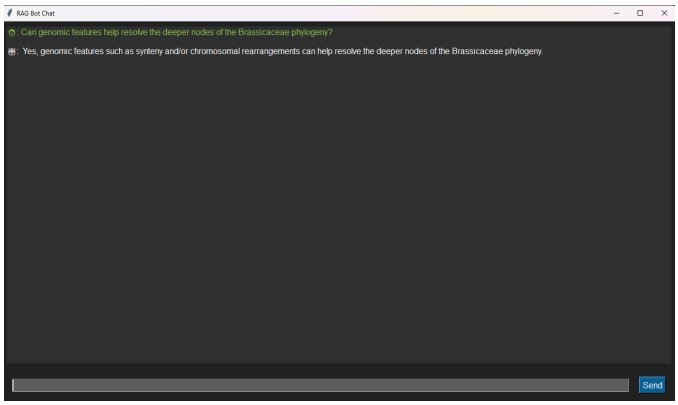

Fig. 1. Chat window

Here the code is separated into 2 major components. One is the GUI running on the local machine where the user is and the other is the RAG system running in a separate docker container that can be hosted on a dedicated server and accessed through an API. The API can then be used to produce a vector store on the user's local machine to ensure safety of the data by not storing anything on the server itself. After the creation of the initial vector database, it can later be updated with more knowledge as more files become relevant/available. The GUI application only needs that locally stored vector store to function. Of course in a realistic scenario, this could also be moved to a server accessible by multiple authorized users. In the test case shown, the container is also locally hosted, but the access through the API is the same regardless of the location of the RAG service. To host an API the flaskAPI python library (*FlaskAPI*, n.d.) is used. The API itself has endpoints only for PDFs, since all of the relevant literature in the experiments were PDFs, but the codebase for docx and HTML files is still present in the docker container and a simple addition of HTML and Docx endpoint functions will allow the API's functionality to widen.

VI. Experiments

A. Text extraction

In this experiment the same 10 pdfs are used for text extraction using Unstructured's fast method, their high-res YoloX method, a custom-made method utilizing the python library 'pdfminer' and a custom method utilizing locally running Detectron2 model, and tesseract-ocr. For context, the pdfminer library is largely what is used by the fast Unstructured method for the information extraction, but this separate implementation allows for more fine-tuning and control over what exactly is done and how the data is handled. For the Unstructured methods tests were conducted both by using their paid API service and the free open-source code running locally and the results between the two are identical. All methods had no issues extracting plain text from normal PDFs and of course, for scanned documents or images with text, only YoloX and Detectron2 paired with tesseract-ocr were able to extract the information.

B. Realistic scenario

In this experiment 15 questions are asked by looking at the input PDF files that are used to create embeddings for the vector store, making sure that the answer of the question is in the files. On top of that 15 other questions are asked about simple topics that are not part of the input files, to make sure that the bot does not make up answers without having context for them inside the database. To make sure that the input PDFs carry information within them that is unlikely to be stored in the LLM's pre-trained knowledge all of the text is synthetically generated using Gemini and a prompt asking it for imaginary scientific information about a made-up topic as well as questions for each paragraph of text and the correct answer. The paragraphs are then randomized in order and placed into different PDFs, to make sure that the RAG bot does not rely on all of the knowledge coming from the same file. Also, each paragraph has a title, the same as in any scientific paper. The prompt for the test data generation is the following:
"I want to test out a RAG system's ability to answer questions based on knowledge within the context only. For this, I need made-up scientific topics and information about them in a relatively short paragraph. Give me 15 of those paragraphs and for each paragraph provide a question that can be answered by the information within it as well as what is the correct answer for the question. The topics of the paragraphs can vary."
The results you can see in table I, all of the questions were answered correctly. Most of the time the chatbot's response would even be a bit more conversational and in-depth while still providing all of the expected facts.

VII. Discussion

A. Table extraction

PDF files and especially scientific papers commonly have tables in them and sometimes to properly answer a question, the LLM might need context from the information within one. All of Unstructured's functions for PDF extraction have the ability to tag extracted elements as tables and even return you the table as HTML, which makes it easy and convenient for an LLM to understand. However, these functionalities are not perfect and for more complicated tables they have either not extracted everything from the table or ruined its intended structure which could potentially feed wrong information into the LLM. To try and find a better solution I looked at pdf libraries such as pdfPlumber and tabula, both of which have methods for detecting tables and parsing them, however in my tests both have failed on every table found and tested from scientific biology papers. I then found out there is a fine-tuned version of the multi-modal model LLaVA specifically designed to extract information from tables, called Table-LLaVA. This inspired me to try and combine a detection model, such as Detectron for identifying and cropping tables from PDFs and the fine tuned Table-LLaVA model in order to produce a solution capable of dealing with all kinds of tables. The idea here is for Detectron to find the bounding box of a table, after which an image can be automatically cropped of just the table and given to the LLaVA model to read, understand and recreate in PDF, similar to how the Unstructured library does it. Event hough this approach does produce tables every time and some of them have meaningful information in them, oftentimes the LLaVA model fails to correctly extract the values of the tables and would hallucinate either some values, all or even sometimes even change the table altogether. It is possible that further fine-tuning and experimenting with the model could lead to more substantial results. You can read more about Table-LLaVA from (Zheng et al., 2024). As a final attempt to deal with the issue, I tried the same approach, but this time using GPT-4o as the multi-modal solution. This did impressively well. All of the tables tested (15) were extracted and recreated in HTML perfectly, even going as far into detail as to color the column headers and backgrounds of tables the same as in the image. All of the tables had varying structure and size and many had long numbers in them with floating points, which once again GPT-4o extracted perfectly. Though this solution worked amazingly for getting the full context from a table inside a document, there is more to be desired since making a call to OpenAI's API for every table in every document could become costly. This is also why for the time being the software solution that is used as a demo of the system's capability is still using the default way of handling tables by Unstructured's library, though if needed this can easily be changed in the code, so long as a detection model was used for the PDF parsing.

B. Information retrieval from vector database

During the experiments section, I noticed something odd with how the retriever from Langchain worked and it should be investigated further in future work. The issue is related to the parameter k, which specifies the amount of documents to be returned from the database starting with the most similar to what is talked about in the question. Here the problem arises in a situation where k is set to a higher number such as 5 while

there is only one relevant document inside the database. In that situation alongside the relevant document, irrelevant ones would get retrieved, which I did not expect to be an issue, so long as there is still relevant information within them. In some situations, however, this makes the bot reply it is missing the relevant context. If k is changed to a lower number in those scenarios, the problem disappears. In general, during the development of this system, less emphasis was placed upon the retrieval system and mostly I left it default from what Langchain offers, since I wanted to focus more on parsing and storing information and did not expect retrieval to pose an issue. In future work upon this system, this issue should be investigated fully and mitigated. There is also a lot more potential that can be discovered here since there are different approaches to retrieving information and even reranking it in order to make sure that what is in the final context for the LLM contains only useful information for dealing with the task at hand.

C. Further improvements/additions

In addition to user experience improvements, which should be added if this software is to be used by people and relied on, the system also needs a robust and consistent pipeline for gathering new information and saving it to the already created database. In the current state of the software, such additions are possible and easily made, so long as there are new files added to the specified input folder, however in a more realistic scenario you would want the system's knowledge to be updated automatically and possibly by using multiple trusted sources of different kinds. On top of information-gathering improvements and the former mentioned issues with tables and retrieval, there is also always the possibility to add solutions for information extraction from more file extensions as well as add new and better parsing methods, models and approaches to the already existing ones. One interesting model is Doclayout Yolo, which is a fine-tuned version of Yolo trained on a synthetic dataset of PDFs to improve its performance in those tasks. I did some testing with it and it did very well. In my limited experiment with it I put it against Detectron2 and it slightly outperformed it in the task of getting bounding boxes close to where the separate sections of the PDF are. You can learn more about Doclayout Yolo from (Zhao, Kang, Wang, & He, 2024).

References

Chromdb. (n.d.). Retrieved from `https://www.trychroma.com/`

Flaskapi. (n.d.). Retrieved from `https://flask.palletsprojects.com/en/stable/api/`

Langchain. (n.d.). Retrieved from `https://www.langchain.com/`

Unstructured.io. (n.d.). Retrieved from `https://unstructured.io/developers`

Zhao, Z., Kang, H., Wang, B., & He, C. (2024). *Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception.* Retrieved from `https://arxiv.org/abs/2410.12628`

Zheng, M., Feng, X., Si, Q., She, Q., Lin, Z., Jiang, W., & Wang, W. (2024). *Multimodal table understanding.*

Question	Correct Answer	Actual Answer
What are some of the optical properties exhibited by organisms that display photonic mimicry?	Iridescence, bioluminescence, and the ability to change color rapidly.	Some of the optical properties exhibited by organisms that display photonic mimicry include iridescence, bioluminescence, and the ability to change color rapidly.
What type of organisms are of particular interest to astrozymologists?	Extremophiles.	Astrozymologists are particularly interested in extremophiles.
What is the primary goal of chronoarchaeology?	To establish precise timelines for past events.	The primary goal of chronoarchaeology is to establish precise timelines for past events.
What term describes the symbiotic relationship between ants and other organisms?	Myrmecophily.	Myrmecophily is the term that describes the symbiotic relationship between ants and other organisms.
What is the primary challenge associated with xenotransplantation?	The risk of immune rejection and the potential for zoonotic diseases.	The primary challenge associated with xenotransplantation is the risk of immune rejection due to differences in the immune systems of different species.
What is the primary focus of bioacoustics?	The study of sound production and reception in living organisms.	The primary focus of bioacoustics is the study of sound production and reception in living organisms.
What is the primary focus of astrobotany?	The study of plants in space.	The primary focus of astrobotany is the study of plants in space.
What is the primary focus of cryobiology?	The study of the effects of low temperatures on living organisms.	The primary focus of cryobiology is the study of the effects of low temperatures on living organisms.
What is the primary goal of paleoceanography?	To reconstruct past ocean conditions.	The primary goal of paleoceanography is to study the history of Earth's oceans by analyzing marine sediments, fossils, and other geological records to reconstruct past ocean conditions, such as temperature, salinity, circulation patterns, and the distribution of marine life.
What is the hypothetical focus of Lithopsychology?	Exploring the potential psychological and emotional effects of interacting with different types of rocks and minerals.	Lithopsychology is a hypothetical field that explores the potential psychological and emotional effects of interacting with different types of rocks and minerals.
What is the focus of Aeroecology?	The interactions between organisms and the atmosphere.	Aeroecology is the scientific study of the interactions between organisms and the atmosphere, focusing on how atmospheric phenomena such as wind, temperature, humidity, and air pollution affect the distribution, abundance, and behavior of living organisms.
What is the primary focus of Chronobiology?	The study of biological rhythms and how living organisms interact with cyclical time.	The primary focus of Chronobiology is the study of biological rhythms and how living organisms interact with cyclical time, including the regulation of physiological and behavioral processes such as sleep-wake cycles, hormone secretion, and body temperature.
What is the focus of Sociogenomics?	The interplay between social factors and gene expression.	Sociogenomics is an emerging field that explores the interplay between social factors and gene expression. This field investigates how social interactions, such as social status, group dynamics, and cultural influences, can affect gene activity and ultimately influence individual behavior, health, and disease susceptibility.
What is the primary focus of Astrocartography?	Identifying locations on Earth where an individual's astrological chart aligns most favorably with the local celestial energies.	The primary focus of Astrocartography is to identify locations on Earth where an individual's astrological chart aligns most favorably with the local celestial energies, aiming to pinpoint auspicious locations for personal and professional success.
What is the focus of Pneumoecology?	The ecology of air and its inhabitants.	Pneumoecology is the study of the ecology of air and its inhabitants, focusing on the complex interactions between airborne microorganisms, such as bacteria, viruses, and fungi, and their environment, including human activities, climate change, and air pollution.

TABLE I
COMPARISON OF QUESTIONS, CORRECT ANSWERS, AND ACTUAL ANSWERS IN A REALISTIC SCENARIO.