# KrishiBot: An NLP Approach to Enhance Farming with Low Resource Language

by

A.S.M. Zawadul Karim
23241072
Rowshon Ara Chowdhury
23241131
Nasif Ahmed Nafi
23241132
Murshed Jamil Alif
24241371

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
June 2025

# Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**

A.S.M. Zawadul Karim
23241072

Rowshon Ara Chowdhury
23241131

Nasif Ahmed Nafi
23241132

Murshed Jamil Alif
24241371

# Approval

The thesis titled "KrishiBot: An NLP Approach to Enhance Farming with Low Resource Language" submitted by

1. A.S.M. Zawadul Karim (23241072)
2. Rowshon Ara Chowdhury (23241131)
3. Nasif Ahmed Nafi (23241132)
4. Murshed Jamil Alif (24241371)

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 17, 2025.

**Examining Committee:**

Supervisor:
(Member)

Dr. Farig Yousuf Sadeque
Associate Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Nirjhor Datta
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

# Abstract

The economy of Bangladesh has expanded significantly in recent years; however, farmers continue to lack sufficient training and timely assistance, which negatively impacts productivity. Given that many farmers reside in remote rural regions, accessing quick and accurate agricultural support remains challenging. This research addresses these issues by developing an advanced chatbot that functions as a virtual agricultural assistant, leveraging Naive Retrieval-Augmented Generation (Naive RAG), GraphRAG, and state-of-the-art Large Language Models (LLMs)—GPT 4.1Mini, GPT-4o, GPT-3.5 Turbo, LLaMA 3.3 70B, and Claude 3.5 Haiku—selected for their superior capabilities in human language comprehension, generation, and interpretation. These systems retrieve precise, domain-specific information from our fine-tuned databases using RAG and GraphRAG methods, delivering targeted solutions tailored to specific agricultural scenarios. The research employs LangChain as the orchestration framework, Pinecone as the vector database, Neo4j as the graph database, and multilingual-e5-large as the embedding model. Furthermore, prompt engineering bridges interactions between LLMs and Bengali-speaking farmers, providing targeted guidance on crop selection, weather conditions, and fertilizer usage for enhanced practical support. Experimental evaluations demonstrate GraphRAG's superior performance compared to Naive RAG. Specifically, GraphRAG achieved 91.58% accuracy on a QA test set of 500 agricultural questions, outperforming Naive RAG's 78.6%, and scored 84.85% in semantic similarity versus Naive RAG's 80.66%. These results confirm the advantage of leveraging graph-based knowledge retrieval to provide more accurate, contextually relevant responses.

**Keywords:** Agriculture, Farming, Chatbot, LLM, GPT 4.1Mini GPT-4o, GPT3.5 Turbo, LLaMA 3.3 70B, Claude 3.5 Haiku, RAG, GraphRAG, Fine-tuned, LangChain, Pinecone, prompt engineering, multilingual-e5-large, embedding

# Acknowledgement

First and foremost, we extend our profound gratitude to Almighty Allah, whose blessings and guidance have enabled us to complete this thesis successfully. We sincerely thank our advisor, Dr. Farig Yousuf Sadeque, and our Co-advisor, Nirjhor Datta, for their invaluable guidance, constructive criticism, and consistent encouragement throughout our research work. Their insightful suggestions and dedicated mentorship have significantly enhanced the quality and depth of this thesis. We also express our gratitude towards Dr. Mohammed Razu Ahmed and Md. Faiz, for helping us proofread our test dataset. Lastly, We would also like to express our most profound appreciation to our parents for their unwavering support, motivation, and prayers, without which this endeavour would not have been possible. Their continuous encouragement has been a vital source of inspiration, enabling us to reach the culmination of our academic journey.

# Table of Contents

**Declaration** i

**Approval** ii

**Abstract** iii

**Acknowledgment** iv

**Table of Contents** v

**List of Figures** vii

**List of Tables** viii

**1 Introduction** 1
- 1.1 Research Statement . . . 2
- 1.2 Research Objectives . . . 2

**2 Literature Review** 3
- 2.1 Survey Methodology . . . 3
- 2.2 Related Works . . . 3
- 2.3 Findings from Literature Review . . . 17

**3 Database & Test Dataset** 18
- 3.1 Database Source . . . 18
- 3.2 Database Pre-processing for NaiveRAG . . . 18
  - 3.2.1 Tokenized Data . . . 19
  - 3.2.2 Vector Database- Pinecone . . . 19
- 3.3 Database Pre-processing for GraphRAG . . . 20
  - 3.3.1 Dataset construction . . . 20
  - 3.3.2 Graph Database- Neo4j . . . 21
- 3.4 Test Dataset . . . 21
  - 3.4.1 Generating Test Dataset . . . 21
  - 3.4.2 Question Length and Answer Length . . . 22

**4 KrishiBot Architecture** 23
- 4.1 Krishibot - NaiveRAG Version . . . 23
  - 4.1.1 NaiveRAG . . . 24
- 4.2 KrishiBot - GraphRAG Version . . . 25
  - 4.2.1 Graph-Based Knowledge Integration and Preprocessing . . . 26

4.2.2 Query Semantic Structuring via Language Models . . . . . 26
4.2.3 Entity Recognition and Knowledge Retrieval . . . . . 26
4.2.4 Contextual Compilation and Instructional Control . . . . . 27
4.2.5 Robust Fallback Mechanism for Unstructured Input (Fallback Strategy) . . . . . 27
4.2.6 Controlled Answer Generation via LLM with Structured Context . . . . . 27

**5 Retrieval & Models** **29**
5.1 Information Retrieval . . . . . 29
5.2 Embedding Model - E5 Large_V2 . . . . . 29
5.3 LLM 01 - GPT-4.1 Mini . . . . . 30
5.4 LLM 02 - GPT3.5 Turbo . . . . . 31
5.5 LLM 03 - GPT4o . . . . . 31
5.6 LLM 04 - Claude 3.5 Haiku . . . . . 31
5.7 LLM 05 - LLAMA3.3(70B-Instruct) . . . . . 32
5.8 Prompt Engineering . . . . . 33
5.8.1 Prompt Engineering for NaiveRAG . . . . . 33
5.8.2 Prompt Engineering for GraphRAG . . . . . 33

**6 Evaluation & Performance** **35**
6.1 Evaluation Matrices . . . . . 35
6.1.1 Retrieval Metrics . . . . . 35
6.1.2 Generation Matrices . . . . . 37
6.1.3 Comprehensive Matrices - QAEvalChain . . . . . 38
6.2 Performance . . . . . 38
6.2.1 Benchmarking Model Using NaiveRAG Performance . . . . . 38
6.2.2 NaiveRAG Performance . . . . . 44
6.2.3 GraphRAG Performance . . . . . 48
6.3 LLM Cost Analysis . . . . . 53

**7 Discussion** **54**

**8 UI Implementation** **55**
8.1 Developing The System . . . . . 55
8.1.1 Frontend Stack and UI Implementation . . . . . 55
8.1.2 Backend Integration with GraphRAG . . . . . 57

**9 Work Plan** **58**

**10 Future Work, Limitation & Conclusion** **59**
10.1 Limitation . . . . . 59
10.2 Future Work . . . . . 59
10.3 Conclusion . . . . . 60

**Bibliography** **63**

**A Test Dataset Approval Letter** **64**
A.1 Test Dataset Approval Letter 01 . . . . . 64
A.2 Test Dataset Approval Letter 02 . . . . . 65

# List of Figures


3.1 Processed Data for NaiveRAG . . . . . 18
3.2 Token Size for Major Categories for NaiveRAG . . . . . 19
3.3 Vector Database- Pinecone Representation for NaiveRAG . . . . . 20
3.4 Graph Database- Neo4j Representation for GraphRAG . . . . . 21
3.5 Prompt Used for Test Dataset . . . . . 22
3.6 Average Question Length of Test Data . . . . . 22

4.1 KrishiBot: NaiveRAG Version . . . . . 23
4.2 NaiveRAG Framework . . . . . 24
4.3 KrishiBot: GraphRAG Version . . . . . 25
4.4 Example Query . . . . . 27

5.1 Multilingual E5 LARGE_V2 . . . . . 30
5.2 System Prompt - NaiveRAG . . . . . 33
5.3 System Prompt GraphRAG 01 . . . . . 34
5.4 System Prompt - GraphRAG 02 . . . . . 34

6.1 Average Faithfulness Over Each Categories . . . . . 39
6.2 QA Evaluation Across Different Models . . . . . 40
6.3 Domain Specific Context Precision & Recall of Naive RAG . . . . . 41
6.4 Domain Specific Faithfulness Across Models of LLM . . . . . 42
6.5 Domain Specific QA Evaluation . . . . . 43
6.6 Faithfulness Comparison by Category of NaiveRAG . . . . . 46
6.7 QA Evaluation Comparison by Category of NaiveRAG . . . . . 47
6.8 Semantic Similarity Comparison by Category of NaiveRAG . . . . . 48
6.9 Faithfulness Comparison by Category of GraphRAG . . . . . 51
6.10 QA Evaluation Comparison by Category of GraphRAG . . . . . 52
6.11 Semantic Similarity Comparison by Category . . . . . 52

8.1 User Interface . . . . . 56
8.2 User Interface with Example Query . . . . . 56

9.1 Thesis Workplan Gantt Chart . . . . . 58

# List of Tables


6.1 Overall Retrieval Performance . . . . . 39
6.2 Average Faithfulness Over Each Categories . . . . . 39
6.3 QA Evaluation Across Different Models . . . . . 40
6.4 Domain Specific Context Recall, Precision, and F1 Score . . . . . 41
6.5 Domain Specific Faithfulness Scores Across Different Models . . . . . 42
6.6 QA Evaluation Scores for Different Models . . . . . 43
6.7 Overall Performance of NaiveRAG . . . . . 44
6.8 Domain Specific Context Recall, Precision, and F1 Score of NaiveRAG 46
6.9 Domain Specific Faithfulness of NaiveRAG . . . . . 46
6.10 Domain Specific QA Evaluation of NaiveRAG . . . . . 47
6.11 Domain Specific Semantic Similarity of NaiveRAG . . . . . 48
6.12 Overall Performance of GraphRAG . . . . . 49
6.13 Domain Specific Context Recall, Precision, and F1 Score for GraphRAG 50
6.14 Domain Specific Faithfulness . . . . . 51
6.15 Domain Specific QA Evaluation of GraphRAG . . . . . 52
6.16 Domain Specific Semantic Similarity of GraphRAG . . . . . 52
6.17 LLM Cost Analysis, MT- Million Tokens . . . . . 53

7.1 Overall Performance of NaiveRAG & GraphRAG . . . . . 54

# Chapter 1

# Introduction

In Bangladesh, agriculture is of great importance to the country's economy: it provides jobs to more than 45% of the population [5], and its contribution to GDP is substantial. Nevertheless, farmers in rural and remote areas have limited access to prompt and relevant information, which negatively affects their agricultural productivity. Therefore, it is crucial for farmers to make timely decisions to maximize production while reducing pest attacks and minimizing inefficient use of fertilizers and pesticides. This lack of knowledge poses a significant threat to agricultural productivity and farmers' incomes. To address these challenges, developing an agriculture-focused chatbot based on Retrieval-Augmented Generation (RAG) Systems is envisioned as a potential solution. The chatbot will assist farmers in Bangladesh in making informed decisions about their crops, soils, and pest management by providing information that is both timely and tailored to each farmer's situation. Responses will be delivered in Bengali, which is advantageous for farmers with limited English proficiency. This technology will improve human contact provided by extension services and help farmers achieve the same level of support available in urban or developed regions. Technically, the chatbot will be built on RAG architectures that excel in natural language comprehension. The system will be trained on knowledge about Bangladesh's agricultural conditions, including crop types, soil varieties, and other country-specific farming challenges. It will also be developed to understand farmers' questions in the most natural manner possible and to identify their issues, offering appropriate solutions. In addition, integrating robust retrieval mechanisms with an advanced LLM will allow the chatbot to generate accurate responses based on user queries and guide farmers with the most up-to-date agricultural knowledge. The importance of such a chatbot in the Bangladeshi agriculture sector cannot be overstated: by providing farmers with easily accessible, high-level information, it has the potential to revolutionize agricultural yields. Ultimately, better decision-making will lead to improved production through effective pest control and resource management. As this solution leverages cutting-edge artificial intelligence, it directly addresses a widely voiced need in one of the most critical sectors of the country's economy.

## 1.1 Research Statement

The research addresses the problem of inadequate guidance provided to farmers. Farmers in Bangladesh often face several challenges during production; however, due to insufficient guidance, farmers need more knowledge to solve their problems. Therefore, this research aims to address those challenges by providing a chatbot based on a Large Language Model (LLM) using RAG Systems, KrishiBot, that will answer their questions related to agriculture and provide proper guidance in their native language, thus contributing to the sustainability of agricultural practice and improved livelihoods.

## 1.2 Research Objectives

- Study textual data from validated sources in standard Bangla, particularly agricultural queries and responses. Determining how the data can be processed to achieve the appropriate responses.
- Study existing literature that explains the ways of LLM application in agriculture to facilitate farmers in crucial decisions for agro purposes.
- Explore transformer-based chatbot architectures, such as GPT-3/4 and several RAG Framework for creating multilingual chatbots that can cater to real-time, contextualized information in standard Bangla
- Develop a chatbot model tailored explicitly for agricultural tasks, using advanced techniques like LLMs and natural language understanding, to provide accurate and context-aware responses to farmers' queries in Bangla.
- Evaluate the multilingual capabilities of the chatbot, ensuring it can accurately translate agricultural advice and information into Bangla.
- Benchmark the chatbot's performance using different LLMs, identifying areas for improvement regarding language comprehension, response generation, and domain-specific adaptability.
- Develop a UI with the optimized pipeline and deploy.

# Chapter 2

# Literature Review

## 2.1 Survey Methodology

We have read multiple technical papers related to the agricultural and chatbot domains to properly analyze and understand the most efficient method of data collection and better understand the model. We have carefully selected research papers containing many citations and a clear understanding of the dataset model and results. Furthermore, we have also focused on choosing recent publications and sorted the papers by their year of publication to maintain proper coherence, ensuring a better understanding of our research study.

## 2.2 Related Works

Rezayi et al. (2023) [7] tried to apply integrating NLP and primarily semantic matching to agriculture in forging links between food descriptors and nutrition facts. This work surveys the effects of transformer-based models, especially AgriBERT fine-tuned above 300 million tokens of agricultural texts, on aligning retail scanner data to the USDA's Food and Nutrient Database for Dietary Studies. The main problem is establishing substantial relationships between numerous food-related text corpora and systematically arranged nutrition databases to identify consumption patterns in diverse segments of the economy and integrate them into forming public health strategies. This work uses bottom-up attention with bi-encoders in semantic matching to evaluate the concordance between food descriptions and nutritional information. In this study, it can be seen again that using external knowledge sources like the FoodOn ontology can increase corpus understanding and inject omniscience into their model, which improves fine-tuning for different ontologies presented through a single set of terminologies. In this case, it is used to evaluate how well AgriBERT performs with accuracy, precision, recall, and F1-score performance metrics. This study also investigates the effectiveness of using large language models (LLMs), such as ChatGPT, as benchmarks for comparison and sources of external knowledge to further boost semantic matching performance. In the results section, the author showed how AgriBERT significantly outperforms state-of-the-art models on USDA datasets for answer selection tasks and hence operates robustly to associate retail product descriptions with relevant nutritional information. This finding paves new ways forward for future work, for instance, predicting food-given ingredients and incorporating NLP in novel uses beyond semantic matching. One

goal of the work presented in this study is to reveal essential dimensions in the use of NLP in the agricultural domain through leveraging advancements in transformer architectures and knowledge graph integration that will aid in healthier food choices and reinforce the understanding of dietary effects on community health (Rezayi et al., 2023).

Yadav and Kaushik (2023) [11] proposed solving the difficulty in user sentiment classification concerning innovative agriculture technologies using DistilBERT, GPT-2, and Agriculture-BERT models. The data collected for this research includes comments made on 86 videos posted on YouTube, which amounted to 7,334 and contained six sentiment classes. The textual data had to undergo significant preprocessing. The comments were pre-processed, and the authors applied the three most general categories: ' Praising,' 'Opinion,' and 'Queries". DistilBERT, GPT-2, and Agriculture-BERT, three Transformer-based models, were adopted by the authors to predict users' sentiments about intelligent farming technologies. The models were trained with different hyperparameters. Thus, while DistilBERT has fewer layers than BERT, achieving an MCC of 0.6819 and macro-F1 score of 0.6953, it was instrumental in resource-constrained environments. The non-generative application's accuracy of 75.29% and the MCC of 0.6564 proved that GPT-2 was ineffective for text classification tasks. However, Agriculture-BERT fine-tuned from agricultural domain-related texts presented the highest accuracy of 77.04% and the best macro-F1 score of 69.63%. This was because it had prior knowledge of the domain-related terms. Comparing DistilBERT with other models, it is reasonable and computationally efficient pre-trained. However, to achieve a better result in the particular technical field, the domain-specific pre-trained "Agriculture-BERT" is required. In addition, GPT-2 underperforms in their experiments, reiterating that generative models must be fine-tuned to do well in classification tasks. Statistical testing conducted by Friedman showed the statistical significance of Agriculture-BERT in terms of prediction performance, suggesting a powerful model and thus making it an appropriate candidate for further domain-specific text classification studies.

Balaguer et al. (2024) [13] compared Fine-tuning and Retrieval-Augmented Generation (RAG) approaches for performance assessment in agriculture models. Performance comparison is made with big language models (LLMs) like Llama2-13B, GPT-4, and Vicuna. The model comparison elucidates fine-tuning and Retrieval-Augmented Generation (RAG) of large language models like GPT-4, Llama2-13B along with recent Vicuna in the agricultural domain. To flexibly adapt their models to a domain benchmark, they fine-tuned with Low-Rank Adaptation (LoRA), which can effectively alter only part of the parameters, like attention weights, without retraining the complete model. Fine-tuning revealed a 6% higher accuracy and significantly improved the models' ability to deal with geographic and domain-specific information. Next, the researchers applied RAG to these models, which used external context to answer questions further to enhance their performance with better and more up-to-date human comprehension. The author used FAISS (Facebook AI Similarity Search) and projections from the early stages of modifying the sentence transformers to perform the similarity search. It installed a general context for the proper models, e.g., RAG. It would get the relevant context from the constructed

knowledge base and then feed it to models like GPT-4 so that they could generate better contextual responses. When incorporated with fine-tuning, RAG boosted the performance by an additional 5%, resulting in enhanced answer quality: geography-specific questions witnessed a leap from 47% to 72% in the similarity of answers. Using the fine-tuning model combined with RAG made it possible to achieve the required degree of resource optimization coupled with high accuracy in using domain models. As for performance, fine-tuning yielded a 6% gain in accuracy. Also, when fine-tuning machine translations with RAG, we observed an increase in accuracy of 6%. More precisely, models fine-tuned individually enhanced response geographical relevance, achieving 72% answer similarity against 47%. The performance revealed that GPT -4 performed the best among the others regarding accuracy, context relevance, and quality of answers provided. An achievement was made by evaluating these improvements using indicators like correctness, pertinence, and form of answer.

Li et al. (2024) [18] emphasized enhancing the generalization of the foundation model (FMs) in different types of LLM models, especially considered for applications related to agriculture. This study aimed to further enhance and surpass traditional models by leveraging LLMs to complete tasks such as recommendation and decision-making in large-scale agricultural data management on plant science (Li et al., 2024). This study's large-scale agricultural text dataset enables the LLMs to work on unstructured large-scale metadata, i.e., agricultural recommender systems and intelligent plant science solutions. Data pre-processed through tokenization and standardizing data to be in tune with selected LLM models, including GPT-4 and LLAMA. The works presented primarily focus on the NLP models (BERT, PaLM, LLAMA, and GPT-4) presented there. For example, BERT uses transformer-based architectures consisting of 110 million parameters for BERTBASE and 340 million for BERTLARGE. Unlike other models, these types are known to be suitable for bidirectional training and thus make them popular in question-answering applications or inference. There is another large Transformer model that likewise has 540 billion parameters but performs better in other NLP tasks due to effective decoding: LLAMA. GPT-4 excelled in language generation, which includes conversational AI or contextually sensitive agriculture advice, among others; with its extensive environment and more than 1 trillion parameters, PaLM proved to be an excellent writer of few-shot learning capacity (Li et al., 2024). The findings revealed that in many areas, GPT-4 outperformed both GPT-3 and LLAMA. GPT-4, for instance, is generally less accurate regarding agricultural language inference and question answering (F1 score: 89.2% vs LLAMA's 83.5%). PaLM performed 7% better than GPT-3 on the context reading comprehension task, illustrating that a decoder-only network has advantages.

Didwania et al. (2024) [17] address the issue of limited access to agricultural knowledge in general, especially in the areas of development where farmers often need more accurate information. The authors stated that it is difficult for farmers to obtain the necessary assistance. It specifically concentrates on solving problems with the conventional support systems from which farmers suffer highly. Based on the research, the solution is an automated query resolution system using Large Language Models (LLMs) along with prompt and contextually relevant information to farmers. Over time, the dataset was more than 4 million inquiries gathered from Tamil Nadu,

India, carefully processed to be more transparent and precise. To keep the query intelligible to low-educated users, extensive data cleansing of inconsistent data such as run-on phrases and grammar errors was needed. Sequence-to-sequence architecture was the selected model appropriate for this research, and the transformer architecture was specifically chosen for its success in handling natural language problems. By incorporating the pre-trained foundation model, Gemini Pro, the generated outputs were rectified of grammatical errors, with improved overall readability. As for performance measures for assessing the model's efficacy, this model also showed a robust relation with expert responses, allowing it to be used in actual applications. The importance of the study lies in its scalability; It aims to not only provide a less taxing solution to conventional call centres but also to help reduce the stress for farmers who need the support. A farmer-centric design of the model explicitly considers the linguistic preferences of farmers who are not proficient in English but know the agricultural terms. The study emphasizes the revolutionary potential of LLMs in agriculture, showing how sophisticated AI technologies can close information gaps and allow farmers quick, easy access to expert knowledge. This work paves the way for future progress in global agricultural supporting systems by addressing technological and contextual issues. The research concludes that incorporating AI agriculture has remarkable potential to improve production and sustainability, by giving the farmers the necessary data to make educated decisions.

Chia et al. (2024) [14] propose the implementation of an irrigation-optimized Virtual Assistant with a LLM model. The data used for training is extensive agronomic data, with weather forecasts, soil characteristics, and plant evapotranspiration rate, including NDVI and FAO 56 algorithms of irrigation methods. In this dataset, there were 10K rows and 15 features, among which the essential features were temperature, soil moisture, and stage of plant growth. Datasets containing 80% of the data were for training and 20% for testing. The preprocessing steps included handling missing values, normalizing features, and converting the data to a question-answer pair format. The Retrieval-Augmented Generation, or RAG, data augmentation method was performed, improving retrieving data efficiency. Fine-tuned LLaMa 2 and RAG were used among many other models. Hyperparameters such as a learning rate of 0.0001, batch size of 32, and maximum token length of 512 have been used. LLAMA was chosen since it was better at processing long-text inputs and conversational flow than the other models. The performance evaluation is based on accuracy, F1 score, and response time. Performance-wise, the fine-tuned LLaMa model has an accuracy of 92% and achieves a 0.89 F1 score; however, it is significantly slower (2.1s per response). The accuracy of RAG performance might have been low, approximately 88%, but it outperformed the response times, i.e., about 1.3 seconds, because of its retrieval mechanism. However, when accounting for the speed/accuracy trade-off, LLaMa fine-tuned was the top-performing model overall because, while slower, it offered more accurate and relevant responses to users.

Nguyen et al. (2024) [19] proposed that My Climate Advisor provide climate change adaptation recommendations to farmers and agricultural consultants using a dataset based on the information retrieved from a scientific and grey literature search in agricultural articles. As for the dataset, 1.36 million documents from peer-reviewed literature, grey sources, and high-impact agricultural journals were selected. As for

the data preprocessing, the documents have been preprocessed at the level of document segmentation to 400 token segments without losing the retrieved meaningful information. The document chunks were encoded using JinaBERT, a transformer-based sentence encoder, to generate contextual embeddings that facilitated accurate and relevant information retrieval. The model architecture relied on Retrieval-Augmented Generation (RAG), combining two models: Llama 3-8b and GPT-4. Llama 3-8b, an open-source large language model fine-tuned for instruction-based generation of answers. Based on this, the RAG framework allowed Llama 3-8b to retrieve snippets from the relevant documents into the generated answers to legitimize the answers by evidence from the literature. This was compared with GPT-4, which was used as it is the most advanced model for language generation. GPT-4 did not implement the RAG framework yet provided clear, well-aggregated answers. Nevertheless, because GPT-4 has no retrieval mode, it can only provide answers informed by its pre-trained training data and does not employ real-time citation of documents. In particular, the evaluation results revealed the differences between the models, as follows: Accordingly, GPT-4 was generally more readable, fluent, and structured and, thus, more accessible to use. Still, the Llama 3-8b with RAG surpassed the GPT-4 in terms of scientific credence, citations' openness, and the directness of the answers, fitting the field's field. Document retrieval was also beneficial because it helped answer questions from reliable outside sources for all fact-based questions posed to Llama 3-8b. Performance markers included context relevance, readability, scientific accuracy, and citation quality, on which Llama 3-8b performed comparatively better in giving accurate answers backed by reference.

Sapkota et al. (2024) [22] cover the state of Large Language Models and their applications in agriculture, showing the dire need for optimization in agricultural practices that can enhance decision-making. The dataset is related to the systematic bibliographic review of 207 selected papers from Google Scholar and IEEE Xplore. Next, the authors meticulously filtered through them, focusing only on papers where LLMs are presented with agricultural use cases to ensure the data was relevant and valid. Their preprocessing involved categorizing literature into crucial research questions on using LLMs in agriculture. The model's architecture differs depending on the model's application. Some key models are transformer-based LLMs, GPT-4 as major LLMs for BERT, and AgriBERT in particular to cope with multiple agricultural tasks. Pixel-level models, or vision-language models (VLMs) like "GLaMM," are used for image-text data, performing activities like image captioning and plant disease identification. Reinforcement learning is also applied to crop and control, where policies are learned from deep Q-network to manage crop growth. Some of these models' parameters include the number of tokens trained, the number of layers, and embedding dimensions, which range from 2000B in Amazon's Olympus to a 175B GPT-3. For instance, the DSSAT simulation system applied in crop management employs LLM techniques for state variables interpretation and management practices. Fine-tuning based on the results, it was observed that LLMs significantly improve agricultural diagnostics, with high accuracy achieved. For example, integrating GPT-4 with CNNs resulted in pest detection with 94.5% accuracy while using models such as YOLOPC, which have exact precision in diagnosing diseases such as coffee leaf rust, etc. These models were again justified by BLEU and ROUGE metrics in natural language tasks, specifically, semantic matching in an agricultural

domain, which showed that LLMs like AgriBERT perform better than traditional methods.

Yang et al. (2024) [24] addressed that LLMs need to be domain-specifically knowledgeable in agriculture regarding much more complex topics like crop cultivation and gene expression prediction. The ShizishanGPT system, which incorporated RAG into an agent-based framework for agricultural decision-making, was presented in this research. The dataset comprises specially designed agricultural questions on crop phenotyping and gene expression prediction. This has been constructed by manually selecting representative questions and expanding them by generative methods to ensure the system's performance is good. Preprocessing included the filtering of questions and answers regarding their relevance and quality. In terms of modeling, ShizishanGPT relied on a multi-module architecture that carried out the following: GPT-4 for general questions, tapped knowledge graphs for facts in specific domains and used external search for real-time data. This RAG component improved the response accuracy by document retrieval in a vectorized knowledge base containing scholarly articles. Specialized external models predicted crop phenotypes and gene promoter enrichment. Key models included ResNet architecture for maize phenotype prediction. Other parameters for ShizishanGPT included vector similarity for knowledge retrieval, embeddings for text processing, and BLEU, ROUGE, and GLEU metrics for evaluating answers. The results had higher geometric means, 0.923, BLEU, 0.88, ROUGE, 0.79, and GLEU, 0.85, substantially outperforming other models such as ChatGPT-4

Pearce et al. (2023) [20] address the need for an open-source and freely available tool to impart AI notions to students by practically building transformer-based chatbots. The current state of AI education offers limited practice in building models hands-on. This paper's dataset includes 75 student-generated questions with their label intent (i.e., five intents), adds the context from course materials, and uses data augmentation to increase small datasets using back translation. DistilBERT and T5 models pre-trained on extractive and generative question-and-answer (QA) on large SQuAD-100,000+. Here, three significant models were employed: BERT for intent recognition, DistilBERT for extractive Question answering, and T5 for generative question answering. BERT Bi-directional transformer categorizes students' questions in context with it — E.g., based on these 75 questions, they should fall into five intents. DistilBERT is a smaller and faster variant of BERT that can answer questions given a context in advance. The model is pre-trained on the SQuAD dataset of more than 100,000 questions. T5 is a text-to-text transformer generating conversational answers based on context.The paper uses a pipeline that integrates the models for these various tasks: data collection, data augmentation with back translation, intent recognition, and answering questions. Using pre-trained models, combined with student-labeled data and augmentation, provides a scalable and interactive AI learning tool to enhance AI literacy through practical engagement with transformer models, natural language processing (NLP), and supervised learning. The paper reports BERT achieving an 82.1 GLUE score for intent recognition, while DistilBERT retains 97% of BERT's performance with 60% faster processing and a 40% smaller model size. T5 delivers 85.97 GLUE score, excelling in generative question answering with a focus on flexibility and conversational accuracy.

Ahmed et al. (2024) [12] tackle the few-shot learning problem that improves intent classification and slot extraction in transformer-based chatbot systems. The issue here represents a situation where the chatbot struggles to understand and reply to user intents in case it's been trained from imbalanced or scarce training data, affecting conversational accuracy. The dataset includes three widely-used corpora: CLINC150, Banking 777, and ATIS. The ATIS dataset, typically used for airline reservations, has 5,871 utterances, 26 intents, and 127 slot types. We have 13,083 queries in 77 intents focused on financial queries under Banking77. 23,700 across 150 intents make CLINC150 a huge few-shot learning problem because of the class diversity in this dataset. To provide a meaningful data representation, preprocessing steps included tokenization utilizing Regex extractors and Lexical syntactic extractors; the following embeddings were later used to capture contextual features of queries: Word2Vec, GloVe, FastText.They proposed an integrated model architecture combining Bi GRU and Conditional Random Fields (CRF) to extract entities while simultaneously recognizing the intent, called the Hybrid Intent and Slots Transformers (HIST) model. The input sequence is processed by four transformer layers and self-attention layers using the BiGRU network to extract contextual information, and the CRF layer is provided to predict the most probable sequence of slot tags. This model is novel in combining sparse and dense features to improve the transformer's simultaneous capture of syntactic and semantic information to resolve the few-shot problem more effectively. Model parameters used in the model include a learning rate of 0.00002, batch size of 32, and 100 epochs. The optimal performance of the HIST model was obtained by incorporating GloVe embeddings and dropout and label smoothing strategies to reduce overfitting and handle the classification problem. Results-wise, HIST outperformed all other state-of-the-art models on these three datasets. Using the model on the ATIS dataset, we achieved 95.28% accuracy for intent classification and 96.19% accuracy for slot extraction. On the Banking77 dataset, GloVe embeddings performed the best, reaching 94.82% intent classification accuracy and 94.78% slot extraction accuracy. HIST achieved 92.63% accuracy for intent classification and 93.75% for slot extraction for CLINC150, which was the most challenging. Moreover, the macro-F1 scores for these tasks were also very high, confirming that the model handles the few-shot learning challenge well.

In Esfandiari et al. (2024) [4] the authors developed a new conditional generative chat system based on a transformer model combined with a conditional Wasserstein Generative Adversarial Network (cWGAN) to generate answers in chatbots. They used the Cornell Movie-Dialog corpus and the Chit-Chat dataset for the dataset they used. The Chit-Chat dataset comprises 7,168 conversations, and the Cornell corpus shall consist of 220,579 conversational exchanges. The preprocessing included tokenizing and embedding real questions through the pre-trained BERT model, which had 12 layers, 768 features, and 12 heads. A linear model was used to achieve dimensionality reduction and reduce computational cost. Positional encoding was also used to keep the word's position in the sentence. They built the model architecture on a hybrid architecture, combining a cWGAN and a transformer model. The generator is a full transformer model with eight layers and 16 attention heads, trained in two phases. Next, Maximum Likelihood Estimation (MLE) was used and then fine-tuned using adversarial learning. The generator produces answers to the given

questions. It is updated with gradient likelihood ratios, and the discriminator uses only the encoder part of the transformer and then a classifier, which is supposed to classify the generated sequences as real or generated. Here, the following parameters are used: learning rate (0.00005), batch size (64), 400 epochs, and dropout rate (0.5). By combining those parameters and the cWGAN and transformer models, the authors noticeably improve stability and overcome mode collapse, a common issue with GAN-based models. On the results side, the model's performance was assessed based on BLEU, ROUGE-L, F-measure, and METEOR. The proposed model outperformed the state-of-the-art on the Chit-Chat dataset, which scored 0.96,0.965, and 0.87 for BLEU-4, ROUGE-L, and METEOR. On the Cornell dataset, it achieved BLEU-4 = 0.71, ROUGE-L=0.818, and METEOR=0.52. The combination of transformer and GAN-based methods offers advantages for generating more semantically accurate and human-like answers than previous approaches have shown.

Dharrao & Gite (2024) [16] proposed a transformer-based chatbot for on-demand, accessible, and AI-based mental health counseling to improve the worldwide shortage of mental health practitioners. The open-domain dataset component and the domain-specific dataset component are two main parts of this paper dataset. The open-domain dataset consists of 58,881 conversational input-output pairs. The domain-specific dataset was produced from conversations between therapists and clients in CounselChat. In total, there are 2,130 QA, and they relate to 31 mental health issues such as couples/marriage relations, depression treatments, anxiety counseling, and substance intervention. A preprocessing step was tokenization, keeping only the phrases below 60 tokens long, removing special characters, and padding sentences. The combined number of input-output pairs was 20,096, and the total number of distinct tokens in all vocabulary words was 8,515. TherapyBot is the chatbot created for general and domain-specific mental health dialogues-as-per-transformers. The essence of the project is the transformer architecture implemented using TensorFlow 2.0. As its name implies, the transformer is more adept at recognizing long-term context than previous models like RNNs and LSTMs.The model comprises six layers, each an encoder and a decoder, employing self-attention and fully connected neural network layers. The authors embedded words into vectors of size 512 and added positional encoding to retain word order information. Two encoder and decoder layers were used during the model's training; depth was set to 256, the number of heads to 8, and the number of units to 512 with a dropout intensity of 0.1. The training was carried out over 12 epochs with a batch size 64. The optimizer was Adam, and the loss function was the sparse categorical cross-entropy. A negative 0.29 loss and near zero perplexity show predictive solid power. The performance of the TherapyBot chatbot is measured in terms of confusion and loss. It has demonstrated a high degree of agreement between expected and accurate responses and the capability of producing well-reasoned, contextually appropriate solutions, with the final loss being 0.29 and the confusion score equalling 1.34.

Chow et al. (2024) [15] study their integration in healthcare conversations to enhance the accuracy and credibility of medical chatbots like other large language models (LLMs) such as GPT-3, GPT-4, and BERT. Diverse healthcare-related datasets such as electronic health records, clinical notes, and medical literature are used in the study, and they are processed for extensive preprocessing for standardization.

They used GPT-3 and GPT 4, two advanced transformer-based models that were fine-tuned using healthcare-specific data along with optimized parameters like the learning rate, batch size, and attention mechanisms. Results show that GPT-4 improves medical chatbot performance by over 90% in medical documentation and question answering. With an F1 score of 92.3%, BERT-based models outperformed previously in clinical text analysis. The results show high precision, 92% and recall 89%, yielding an overall F1 score of 90%. Despite pointing out the transformational potential of LLMs in healthcare discussions, the authors point out challenges such as data privacy, ethics, and model hallucinations. The research highlights the importance of continued refinement to maintain the reliability of AI-facilitated medical chatbots to foster more effective health service for patients, leading to a more efficient healthcare system.

Wang et al. (2023) [10] focus on the area of conversational query production for dialogue systems, which aims at generating these queries to search the external information from a search engine and enhance the system's response. The dataset comprises multiple environments involving Wizard-of-Internet, Wizard-of-Wikipedia, DuSinc, KdConv and DuConv in English and Chinese. The primary processes carried out during the preprocessing stages include formulating queries in advance and storing documents, which can be recovered in the event of a need to cut the time that a search engine takes to be queried. The model of Wang et al. (2023) is a Retrieval-Augmented Generation (RAG) model targeting conversational query generation. It consists of two main components: a retriever and a generator. The retriever is accomplished through a T5 encoder, which can co-process and analyse input dialogues and retrieved documents when combined with the generator. To refine the RAG model scaling, query filtering has been incorporated to filter out low-quality samples to improve the program's performance. Some forms of regularisation, including -filtering and -filtering, were also incorporated to reduce noise in some of the content and enhance the transferability across domains. The results show that the proposed model has a vastly superior performance to baselines, especially in the NOISY domain, using measures as Recall@K (R@1, R@3, R@5) for query creation and Perplexity (PPL) for response generation. In the NOISY setting, the gain using their approach is 68.34% Recall@1 against the BM25 algorithm, with Recall@1 of 67.63%, showing the effectiveness of RAG-based feedback. Specifically, Wang et al. (2023) observed in the CLEAN setting that although T5-based models can obtain good results, weak-supervision conditioned models such as WSMF and QP-Ext could obtain better results, and this is also affected by using larger pre-trained. The RAG-based model, with -filtering, -filtering, and reinforcement learning, dealt with noisy content and excelled all baselines in query creation and answer generation. Overall, the RAG framework outperformed in both clean and noisy environments, demonstrating its robustness and performance when combined with both semantic-rich feedback and reinforcement learning over domain adaptation in conversational query production.

Bird et al. (2024) [3] address the data sparsity problem in NLP by a pipeline that enhances human-authored data with T5 paraphrasing and improves text classification for chatbots using transformer models and ensemble learning. This paper experimented with 483 source text commands, crowdsourced through questionnaires

by humans. EEG-Based Emotional State Classification: Mental State Classification: Sign Language Recognition. To deal with the small size of the dataset (10K samples), all commands were augmented using a T5 model pre-trained on Quora question pairs (to 13,090). The entire dataset was divided into 70:30 to maintain the bias in data, where 70% training-cum-paraphrased data and the rest was a test. Seven transformer-based pre-trained models were used in this paper: BERT, DistilBERT, RoBERTa, DistilRoBERTa, XLM-RoBERTa, and XLNet. They all have Transformer-based weights tuned for text-sequence processing with self-attention for two epochs. RoBERTa has more extensive architectural changes like dynamic masking and removing the Next Sentence Prediction objective. The top-performing model was RoBERTa, with 12 layers and 768 hidden units, which also got an accuracy of 98.96% in the augmented dataset. For augmented data, the research achieved higher accuracies for BERT and DistilBERT, with 98.55% and 98.34%, respectively. However, the other models performed inconsistently from better to worse than their original results, such as distilRoBERTa, XLM-RoBERTa, XLNet, etc. An ensemble model that combined the power of BERT, RoBERTa, and XLM-RoBERTa yielded an overall accuracy of 99.59%, which is a testament to this operation capability when using many transformer models. In terms of performance, the fine-tuned RoBERTa performed the best with precision, recall, and an f1 score of 0.99. On the other hand, BERT improved its best accuracy from 90.25% to 98.55% with data augmentation, while XLM had the worst with just 13.69%, demonstrating that it does not work well using only English data alone.

In Saha et al. (2021) [1] the authors focused on the issue of building an extractive QA for the Bangla language using BERT-Bangla, which has been pre-trained for Bangla text. To this end, the authors developed the Bangla Question Answering Dataset (BQuAD) with 13,000 questions and answers collected from 5,000 context paragraphs from Bangla Wikipedia. They also preprocessed text using WordPiece tokenization to join together syllables. They found it more effective than subword tokenization and used additional tokens like [CLS] for classification purposes and [SEP] to split sequences. The dataset was then used to refine the model for QA tasks further. The model used was one of the BERT family but was fine-tuned to Bangla data. BERT-Bangla has 65 million parameters, and its configuration has 8 attention heads, eight layers, and an embedding size of 512. The model's vocabulary size was 75,000 tokens. Pre-training of DeepSpeech 2 was carried out on a TPU v3-8 system with a memory of 128 GB for 35 hours, passing approximately 1 million steps, sequences of 128 and 512 tokens. For fine-tuning, the model was trained on QA tasks with the learning rate = 2e-5 and the batch size = 16 for 3 epochs Fine-tuning turned out to be computationally cheap and took less than 30 minutes on NVIDIA Tesla T4 GPU. The authors assessed the model's performance on extractive QA tasks, and it performed well. It received an EM score of 45% and an F1 score of 78.5%. This was to assess how effective the model was in identifying the correct span of text of the answers. New Model BERT-Bangla also performs better than prior-generated models such as mBERT and Bangla-ELECTRA, proving that it is efficient as a monolingual language model for the Bangla language.

Bhattacharjee et al. (2023) [2] address the large gap in NLP literature for the Bangla natural language generation (NLG) consisting of something that is widely spoken (Bhattacharjee et al., 2023). The authors present the BanglaNLG benchmark, which encompasses six diverse NLG tasks. Therefore, this benchmark compares NLG models trained for Bangla to enable further research in this low-resource language. Based on a large corpus of 27.5 GB of clean Bangla text, the authors preprocessed it for data processing. The authors had to normalize the text and build a vocabulary with SentencePiece, which produced about 4.8 million data points with an average seq length of 402 tokens. Criteria such as diversity, practical applicability, difficulty, accessibility to the task and reliable evaluation metrics were used to decide the tasks. Different datasets and metrics are used for tasks like machine translation, SacreBLEU; text summarization, ROUGE-2; quote retrieval, ROUGE-N; question answering, collect, etc. To select a model, we have focussed on BanglaT5, a sequence-to-sequence Transformer model pre-trained on the dataset above. Using 12 attention heads with 12 layers of architecture, we demonstrate that a span-correction objective is effective for training, as shown to be efficacious for generative tasks. They fine-tuned the model against a range of multilingual models (including mT5 and mBART-50) to establish a robust baseline against which to compare performance. In particular, across all six tasks, BanglaT5 demonstrated state-of-the-art performance in terms of performance metrics, beating comparable-size multilingual models by huge margins (up to 9% absolute gain and 32% relative gain compared to other approaches) (Bhattacharjee et al., 2023). Methods of bootstrap sampling were used to establish statistical significance. The paper states that the new dialogue dataset and BanglaT5 model will be publicly available to foster extra exploration and growth in Bangla NLG. Findings underline the need to generate benchmarks and resources for low-resource languages like Bangla. The authors aim to provide a BanglaNLG benchmark and a competitive model such as BanglaT5 to stimulate the development of NLG capabilities for Bangla populations.

In this research paper, Touvron et al. (2023) [9] presented Llama 2, a series of large language models (LLMs) created to improve talking learning while maintaining safety and usability. The motivation for exploring such open sources stems from the recognition that many applications built on top of proprietary models, like ChatGPT deploy base or fine-tuned versions, are non-transparent in their processes for obtaining these. The authors argue that current models fail to provide enough helpfulness and safety for end users' needs, hence requiring a more advanced model capable of equality with closed-source counterparts. The module used for the data processing of Llama 2 was pre-trained on 2 trillion tokens from publicly available materials. To suppress instances of hallucination (Touvron et al., 2023), the authors applied rigorous data-cleaning techniques to remove personal information and make factual details more accurate. Significantly, the pretraining corpus was expanded by 40% over previous models while the context length was doubled to better performance on multiple tasks. Autoregressive transformation with substantial improvements was utilized to grab model selection. Touvron et al. (2023) introduce grouped query attention (GQA) to scaling inference for the Llama 2 models ranging from 7 billion to 70 billion parameters. To achieve fine-tuning, we used Supervised Fine-tuning (SFT) and Reinforced Learning with Human Feedback (RLHF) using techniques such as rejection sampling and Proximal Policy Optimization (PPO). A

series of benchmarks evaluated the helpfulness and safety performance metrics. On human evaluation, Llama 2-Chat models perform similarly to some closed-source models based on human evaluations (Touvron et al., 2023). The Llama 2-Chat models outperform the state-of-the-art open-source chat models on all tests under consideration. Hence, the authors tested how well the model performs on these prompts with human raters and over approximately 4,000 prompts, providing a comprehensive evaluation of capabilities. Finally, Llama 2 advances state-of-the-art language modelling with an open-source implementation that prioritizes safety over performance while achieving competitive scores.

Ruma et al. (2023) [8] propose answer-aware Bengali question generation to address the critical task of Bengali question generation. A limited amount of training data in Bengali, compared to languages such as English, poses a problem statement for the emergence of an efficient system. In this study, the author aims to build a system that produces diverse and contextually relevant questions from text passages using fine-tuned text-to-text transformer (T5) models with answer aware inputs. The authors modified an existing Bengali question answering dataset to base their QG task data processing. It derives the dataset from SQUAD2.0 by machine translation, having 132,777 samples of question/context and answer triplets, a widely highly trained resource. In particular, they took the preprocessing steps to make the data compatible so that the transformer models could work. Multilingual and monolingual T5 models were chosen for model selection as they had superior text generation performance compared to the other state-of-the-art models. The architecture of T5 enables several NLP tasks within a single framework. Eventually, with their encouraging works, along with them doing some hyperparameter tuning by changing sizes and configurations for model training to train BanglaT5 junked on other crosswords much better effectively possible in QG (Ruma et al., 2023). The approach to evaluate the quality of questionS generated was measured using performance metrics. With beam search decoding and additional fine-tuning of BanglaT5 models, they got an excellent RougeL Fscore (35.77) and BLEU-1 score (38.57). It confirms that the model generated plausible questions, both grammatically and contextually. The authors also used automatic evaluations and human assessments to ensure the generated questions met the criteria of correctness, answerability, etc. The study also conducted a complete error analysis on Question Generation, which genuinely represents some error categories.

Nath et al. (2024) [21] addressed the critical challenge of managing agricultural data in Bangladesh has been particularly difficult because of the lack of ability to adhere to the FAIR (Findable (F), Accessible (A), Interoperable (I), and Reusable (R)) principles, which hinders its potential for interactive analysis and integration. Datasets for the study were compiled from the Bangladesh Bureau of Statistics (BBS) and other sources, comprising 54,712 crop records, 6,743 fisheries entries, 520 forestry records, and 73 additional records from 1969–70 to 2019–20. Preprocessing data involved converting PDFs to CSV using tools like ilovepdf.com, removing aggregated figures, and assigning a unique ID by concatenating the district, year, and product code. In the first of its kind for Bangladesh, the study developed the Bangladesh Agricultural Knowledge Graph (BDAKG), a 44 MB RDF-based knowledge graph comprising approximately 382,000 triples. RDF, RDFS, OWL,

and QB4OLAP vocabularies are employed in the use of BDAKG for the semantic integration of heterogeneous data; by this use, BDAKG semantically integrates heterogeneous data in a multidimensional model, described in terms of three basic dimensions: Geography, Product, and Time. BDAKG annotates data cubes about crops, fisheries, and forestry using qb: DimensionProperty and qb4o: Hierarchy and makes semantic bindings to external knowledge graphs (like Wikidata) in the form of owl:sameAs. Each dimension in BDAKG leverages hierarchies (e.g., District → Division → National) to allow for OLAP-style analytics. Using tools such as SETL and Protégé, the ETL process took 8.5 hours, and a triple store was used to load the data. Completeness, correctness, FAIR compliance, and OLAP compatibility were verified by evaluation of the knowledge graph. Federated SPARQL Queries Have found applications in identifying districts requiring attention for forestation (for example, Barguna: 75,000 acres vs. 113,131.5 required), surplus fish production in 28 districts (2019–20), and sustainability insights, such as reducing rice and onion production and increasing production of wheat, soybeans, and tea. These findings suggest that BDAKG has the potential to inform policy related to the sustainable use of natural resources and evidence-based agricultural practices in Bangladesh.

Drawing on the shortcomings of rule-based tourism chatbots, Krataithong et al. (2025) [25] introduced a context-aware, personalized response system based on a hybrid GraphRAG architecture. The data refer to 107 tourism sites, 18 local products, and 13 travel routes in four communities: Muang Rayong, Klaeng, and Wangchan districts of Rayong, Thailand. All of the content was in Thai, which highlighted the requirement for language-specific processing. The five steps of data preprocessing are (1) collection to CSV, (2) cleansing to remove HTML tag, (3) semantic enrichment to extract entity types, (4) Thai language tokenization using open PyThaiNLP library (combining fields like description and location) and finally (5) word embeddings generation using OpenAI embedding model for retrieval based on similarity. The proposed chatbot combines a Neo4j knowledge graph, Retrieval Augmented Generation (RAG) using GPT-4 and LangChain, and an end-user interface in the form of Streamlit. Core entities of the knowledge graph (Community, Place, Product, and Route) are connected by semantic relationships (e.g., LOCATED_IN and SOLD_AT). This is augmented with a vector index for semantic search and spatial functions for geographic queries. Structured Cypher queries for precise location-based retrieval were used, combined with vector similarity search for exploring descriptive, open-ended queries. Contextual relevance was further prompted with a few short examples. An experimental evaluation of 45 Thai language queries reveals that the structured-only (T1), vector-only (T2), and hybrid model (T3) methods outperform each other considerably. For location-based queries, T3 scored an F1 of 91% (compared to 88% for T1 and 50% for T2), and 80% for non-location-based queries (compared to 71% for T1 and 56% for T2), with precision and recall of 91% and 92%, respectively, and 91% and 71%, respectively. These findings demonstrate that the framework can deliver accurate and contextually relevant tourism recommendations that are scalable across regions and multiple languages.

Zhang et al. (2025) [26] addressed the challenge of making large language models (LLMs) effective for specific domains by augmenting Retrieval Augmented Generation (RAG) with graph-based techniques. Datasets are drawn from various sources, including but not limited to a) large public knowledge bases (e.g., DBpedia and YAGO 4.5), b) domain-specific corpora (e.g., biomedical literature and Thai tourism data from the Navanurak platform). First, documents are segmented into coherent textual chunks, and dense vector embeddings are generated from models such as OpenAI's embedding model (1536 dimensions). Domain knowledge graphs (KGs) are then constructed, partially based on named entity recognition and relation extraction, using the PyThaiNLP tool. We propose a GraphRAG framework that unifies knowledge-based, index-based, and hybrid retrieval paradigms, combining the structured querying capabilities of graph databases with the semantic flexibility of vector similarity search. Technically, the framework stores the graph in Neo4j and utilizes LangChain for orchestration, as well as Streamlit for user interaction. GraphRAG operates by blending Cypher-based graph traversal with contextual similarity matching to enable multi-hop reasoning by expanding the context window (from 2K to 32K tokens) and improving response accuracy. Through empirical evaluations across multiple domains, we demonstrate the efficacy of GraphRAG. For location queries in the tourism domain, the hybrid GraphRAG model outperformed its vector-only and structured-only RAG counterparts in terms of F1-score (91 vs. 88 and 88, respectively) and for both general location and non-location queries (80 vs. 71 and 71, respectively). GraphRAG was shown to provide significant efficiency gains in terms of token consumption (between 26% and 97%), a crucial aspect of cutting down computational costs at inference time in biomedical applications. In general, GraphRAG provides a wide range of applications, including healthcare, tourism, and education, to utilize domain-specific knowledge in LLM pipelines in a robust and scalable manner, thereby enabling more nuanced and context-aware LLM responses.

To support the analysis of climate-resilient agriculture, Wu et al. (2024) [23] developed a domain-specific knowledge graph that addresses the complex challenge of combining climate, agricultural, and nutrition data for climate-resilient agrarian analysis. The study utilized heterogeneous datasets, including global agricultural production and trade data from FAOSTAT and climate variables (temperature, precipitation, and humidity) from the NCEI. The two datasets were in CSV format and underwent extensive preprocessing to validate data integrity. These also involved discarding FAOSTAT data records when they were incomplete, standardizing record headers in NCEI files, and implementing procedures to recover from server failures. Python was used to build the knowledge graph using Neo4j. The schema was ontology-driven and included node types (Crop, Weather, Nutrition, and Greenhouse Gas) and relationships (PRODUCES, AFFECTS, and EMITS) that were designed to promote semantic scalability. Aligning key parameters (e.g., location and date) across datasets supported data integration and allowed the execution of Cypher queries for structured retrieval. The system, for instance, could determine from which countries bananas are produced or what the crop yield is related to weather conditions. Interactive graph visualization was performed using Neo4j Bloom, and the resulting CSV file was exported for downstream analysis. Exploratory analytics (e.g., Pearson correlation analysis) revealed a negative linear

relationship ($R^2 = 0.18$) between banana yield and precipitation. Still, no significant pattern was observed with temperature, highlighting the need for a finer temporal resolution. Although the study did not report traditional performance metrics (e.g., F1 Score), it demonstrated that the knowledge graph can be used to harmonize and analyze data from different domains. A case study of banana production showed how it facilitates unified accessibility to and queryability of complex datasets. The future includes increasing the spatial and temporal resolution, as well as mapping to real-time web sources, to further explore the interdependencies among climate, agriculture, and nutrition.

## 2.3 Findings from Literature Review

The recent literature on retrieval-augmented generation (RAG) establishes a robust foundation for combining external knowledge with large language models (LLMs) to achieve grounded, context-aware responses. In conventional RAG pipelines, dense vector retrievers—typically powered by multilingual or domain-specific embedding models—identify top-k passages from a corpus, which are then passed to an LLM for answer generation; empirical studies report that this approach markedly improves answer relevance and reduces hallucination rates. Extensions of this paradigm embed an explicit knowledge graph—often implemented in Neo4j—to represent entities and their relations as nodes and edges, enabling structured, multi-hop inference that augments the model's capability to trace reasoning paths through connected facts. Across both paradigms, advances in prompt engineering—especially chain-of-thought prompting—have been shown to leverage the inherent inference strengths of cutting-edge LLMs such as GPT-4, GPT-3.5 Turbo, LLaMA 3.3 70B, and Claude 3.5 Haiku, leading to measurable gains in factual consistency, answer precision, and interpretability. Together, these findings underscore the importance of aligning high-quality retrieval mechanisms, graph-based knowledge representations, and carefully crafted LLM prompts to maximize the accuracy and transparency of modern RAG systems.

# Chapter 3

# Database & Test Dataset

## 3.1 Database Source

This study obtained its database privately from the Bangladesh Ministry of Agriculture and validated them through official ministry representatives. The database information maintains both well-evaluated structure and high reliability which reduces the chance of misguidance. This extensive farming database comprises 10 main groups of crops that contain several separate sections to help users answer many types of questions.

## 3.2 Database Pre-processing for NaiveRAG

The database required systematic conversion into text files to facilitate analysis through file-based structuring which aligned with individual sub-category sections. All images were removed from the documents before preprocessing and tables were converted to paragraphs to support appropriate functioning of our language model (LLM) and boost the accuracy of its responses. As the database was already formatted as a book, we did not need to remove redundant information, as it did not contain any.
Figure 3.1 illustrates sample pre-processed data,

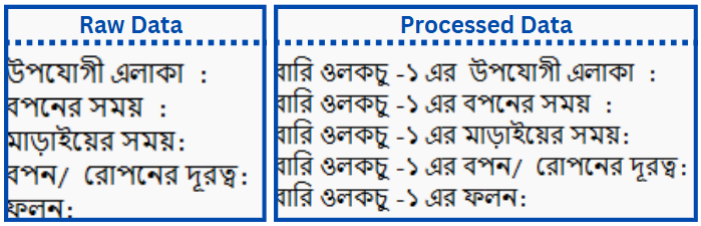



Figure 3.1: Processed Data for NaiveRAG

This manual preprocessing streamlined the editing of data chunks. Initially, when the raw datasets were deployed to the model, it exhibited hallucination issues by generating inaccurate or irrelevant answers. To solve this, each sub-category was meticulously reviewed, with detailed information about its characteristics, diseases, and production aspects added under its name. This method effectively eliminated hallucination issues, allowing the retriever to accurately access and retrieve relevant documents, thereby enhancing the model's reliability.

### 3.2.1 Tokenized Data

The Multilingual E5_LARGE_V2 model performs tokenization on pre-processed data chunks.The tokenization technique transforms data into a format most suited for semantic search and similarity-based query processes to make information easier to index and retrieve. The visual representation of tokenized data in the figure shows the required number of tokens needed to store documents within Pinecone vector database.
Figure 3.2 represents token size for eight major category

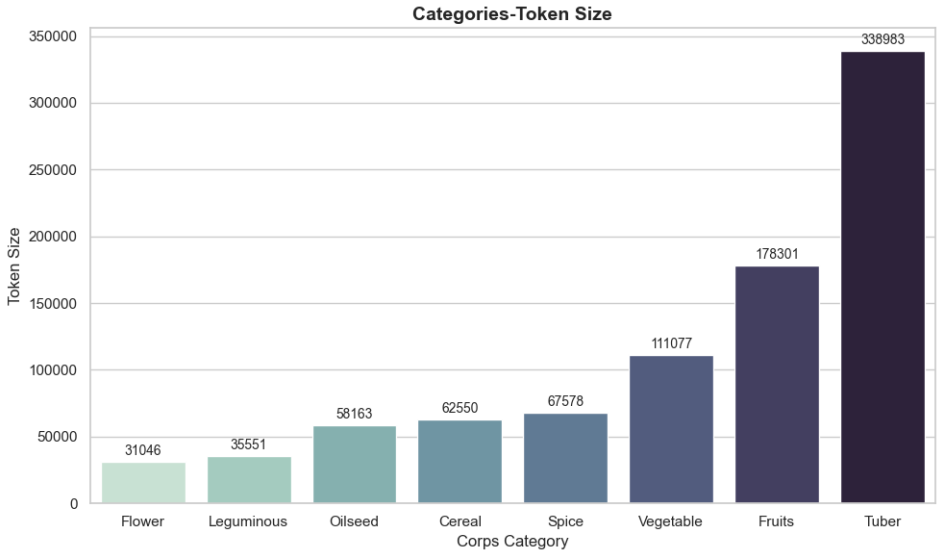



Figure 3.2: Token Size for Major Categories for NaiveRAG

### 3.2.2 Vector Database- Pinecone

Our NaiveRAG system used Pinecone as its vector database where database embeddings could be stored and retrieved. The selection of Pinecone was motivated by its unlimited scaling potential and similarity search operation. The document preprocessing pipeline achieved text segmentation through chunks of 512 tokens with 150-token overlaps between each chunk. The chunking technique combines contextual coherence with retrieval performance optimization through its specific design. The selection of 150 overlapping tokens safeguards semantic unity across adjacent chunks which minimizes the possibility of dismantling information links. After processing the chunks Pinecone received them for embedding and indexing. Our text context retrieval system relies on cosine similarity as the distance metric because it calculates the angle relationships between vectors to efficiently identify semantic connection between units. Our high-dimensional embedding space benefited significantly from this search metric when performing semantic operations. Here we can also easily visualize the chunks and modify the chunks if needed.
Figure 3.3 illustrates vector embedding stored in Pinecone Vector Database.

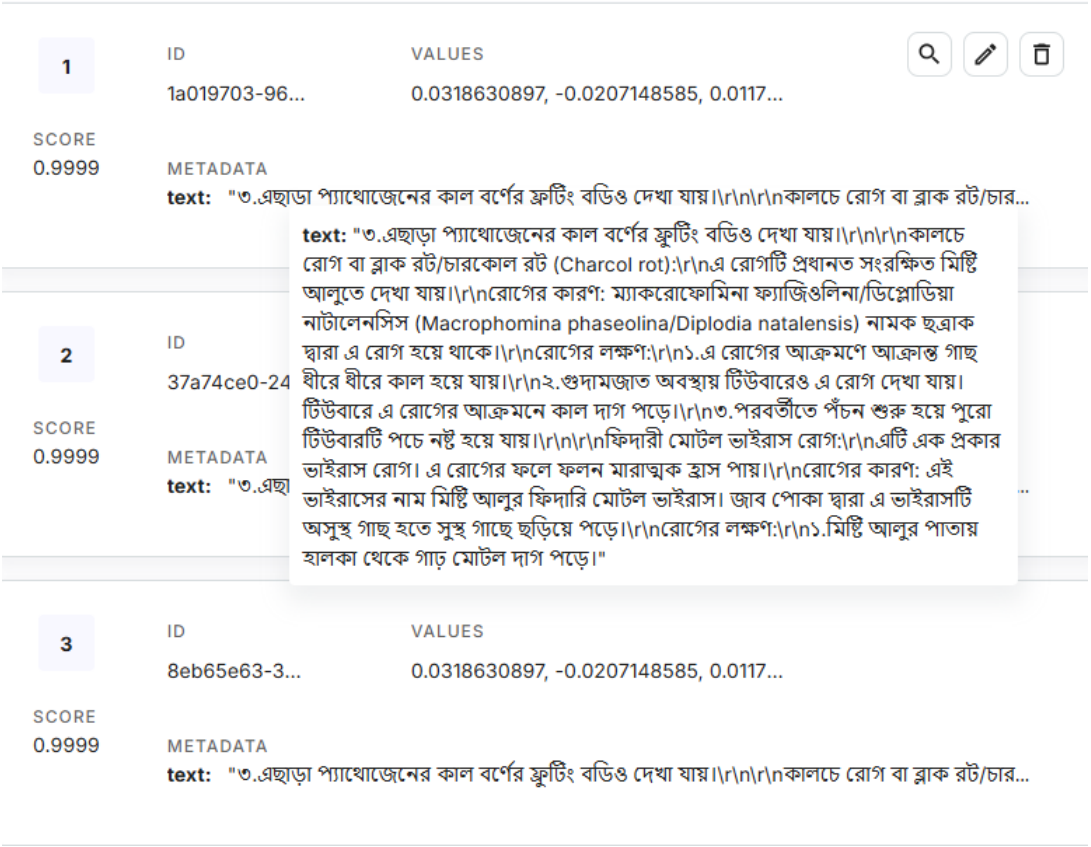



Figure 3.3: Vector Database- Pinecone Representation for NaiveRAG

## 3.3 Database Pre-processing for GraphRAG

### 3.3.1 Dataset construction

To support the construction of a knowledge graph for our agriculture-focused chatbot system, we curated a structured CSV dataset derived from an authoritative agricultural book. This dataset is designed to facilitate manual graph creation in Neo4j and includes a total of 44 columns, with each column representing a distinct attribute or property associated with agricultural entities. The dataset spans 729 rows, each corresponding to a unique agricultural category, ensuring comprehensive coverage of domain-specific knowledge. This tabular representation enables precise mapping of entities and their relationships in the graph database, promoting a semantically rich and query-efficient structure for downstream question-answering tasks. The granularity and breadth of the dataset allow for modular and scalable graph modeling, aligning with the goal of enabling context-aware, accurate, and explainable responses in low-resource Bangla agricultural scenarios.

### 3.3.2 Graph Database- Neo4j

For our GraphRAG architecture, we utilized Neo4j as the underlying graph database to efficiently manage and query structured agricultural knowledge. Neo4j is a highly scalable, native graph database optimized for storing entities and their relationships in the form of nodes and edges, making it particularly well-suited for retrieval-augmented generation tasks involving complex domain knowledge. Its support for the declarative Cypher query language enables expressive and precise graph traversal, allowing us to retrieve relevant crop varieties, properties, and contextual associations in response to natural language queries. By leveraging Neo4j, we were able to build a semantically rich knowledge graph that mirrors real-world agricultural hierarchies and dependencies, thereby enhancing both the accuracy and explainability of the chatbot's responses. This integration played a crucial role in bridging structured agricultural data with unstructured user input, ultimately enabling more reliable and context-aware answer generation in low-resource Bangla environments. Figure 3.4 illustrates vector embedding stored in Neo4j Graph Database.

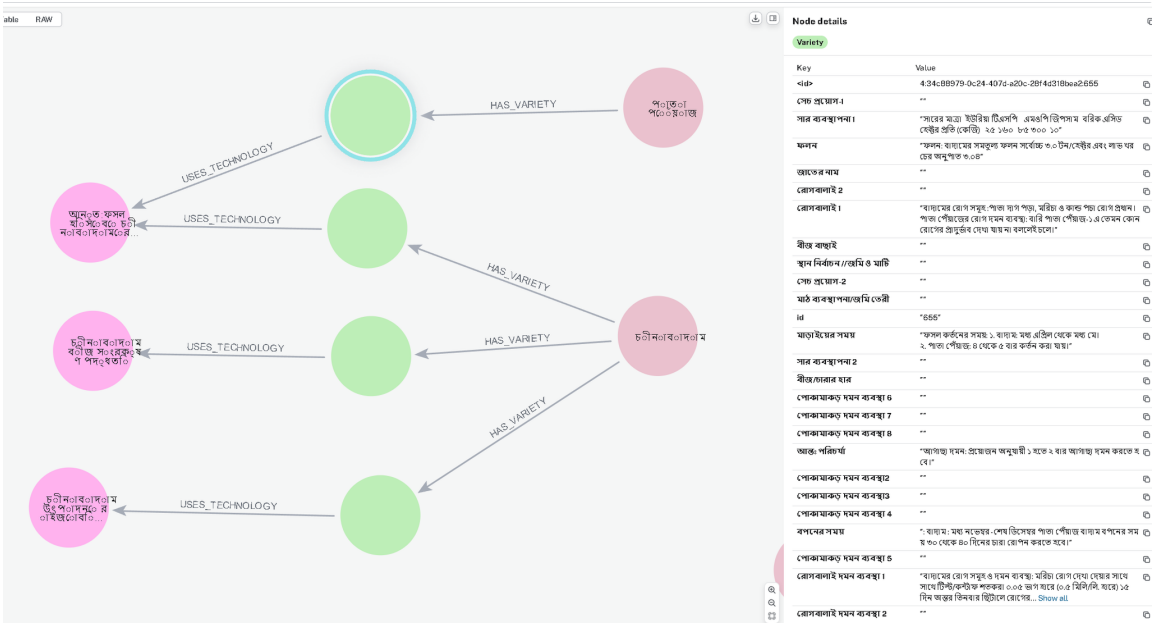



Figure 3.4: Graph Database- Neo4j Representation for GraphRAG

## 3.4 Test Dataset

### 3.4.1 Generating Test Dataset

To ensure the accuracy of the answers produced by our proposed systems, we developed a dedicated test dataset with pre-defined questions and answers. Using a large language model (Perplexity AI), we generated 500 questions along with their corresponding answers to build this dataset. The questions and answers were exclusively sourced from the document used in our system to ensure alignment with the data. Each Q&A pair was meticulously reviewed to ensure that the LLM did not incorporate information from its own knowledge base and relied solely on the document. Additionally, all answers were rigorously validated by agricultural experts to

eliminate the risk of misinformation. For each sub-category, the number of questions generated depended on the length of the respective chunk. This systematic approach ensured that questions were diverse, contextually relevant, and that answers adhered strictly to the information within the document. for each sub-category, the number of questions generated depended on the length of the respective chunk. This systematic approach ensured that questions were diverse, contextually relevant, and that answers adhered strictly to the information within the document. We used the following prompt in Figure 3.5 for generating the Q&A pairs,

```
"Imagine you are a farmer and create scenario-based questions in Bangla based on
the text file. Answer these questions in Bangla using only the information
provided in the text file without introducing external knowledge. Include all
relevant detailsfrom the document for each context. For instance: ' I'm a farmer,
I produce turmeric in my field but I've seen some spots in my leaf for a few
days, now how do I solve this?' In your response, confirm if it matches the
symptoms of a specific disease, provide a solution as per the text file, and
verify the question-answer alignment with the document. Avoid generating
repetitive questions."
```

Figure 3.5: Prompt Used for Test Dataset

### 3.4.2 Question Length and Answer Length

As mentioned earlier, we have created multiple question types which include short and long questions and scenario-based questions to evaluate our proposed RAG systems. The system's effectiveness in dealing with different question types becomes evident through the assessment of questions of different lengths as illustrated in the accompanying figure.

Figure 3.6 illustrates average question and answer length of the generated dataset

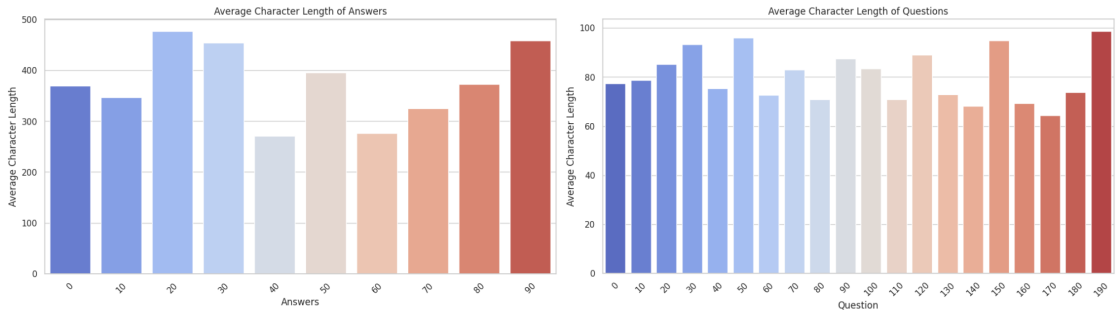



Figure 3.6: Average Question Length of Test Data

For answers, each matches its question's complexity level which produces answers of different lengths. We established a 1500-character limit for answer lengths to validate the system. The accompanying chart shows the average answer lengths found in validation data to reveal response size distribution patterns.

# Chapter 4

# KrishiBot Architecture

## 4.1 Krishibot - NaiveRAG Version

The following figure 4.1 illustrates the detailed workflow of our Naive Retrieval-Augmented Generation (NaiveRAG) system. This workflow clearly demonstrates each step involved in processing user queries—from embedding generation, retrieval of relevant context from the vector database, to finally generating accurate and context-aware responses tailored specifically for agricultural assistance.

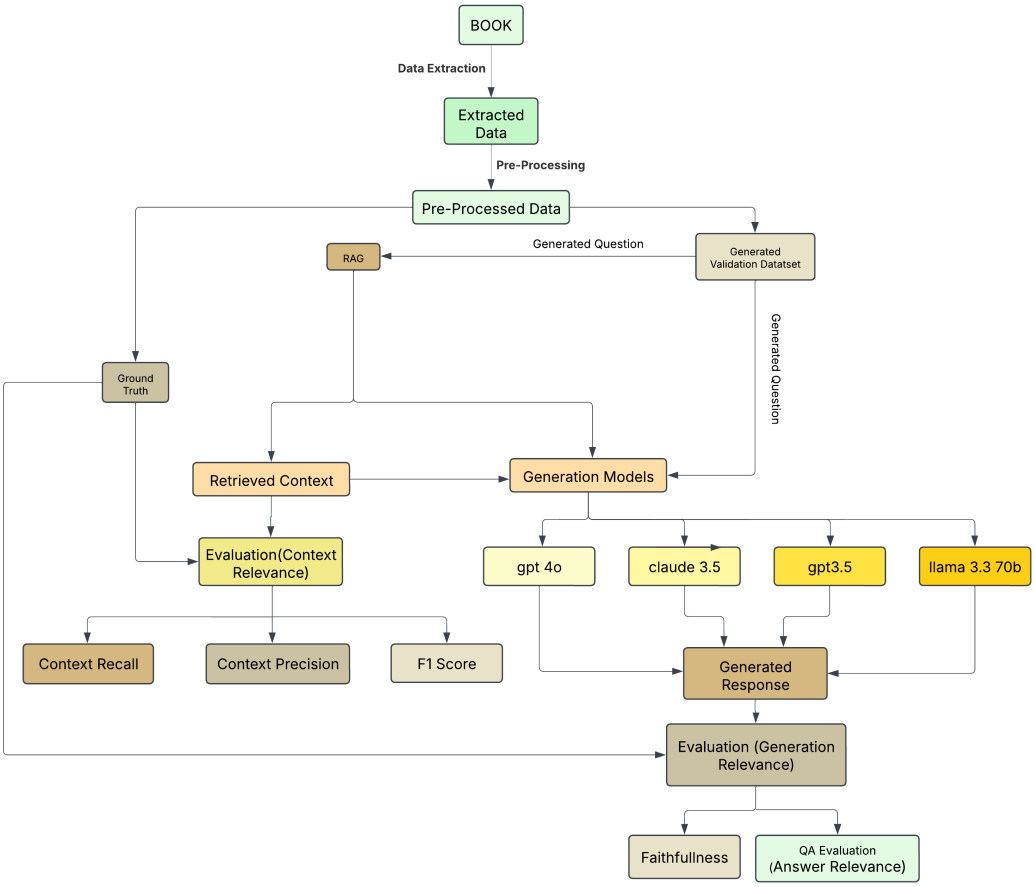



Figure 4.1: KrishiBot: NaiveRAG Version

### 4.1.1 NaiveRAG

Our NaiveRAG architecture begin with a Document Loader that pulls the pre-chunked documents.These chunks are passed into a pre-trained, E5 Large embedding model, to get vector embeddings for the chunks. Through embedding model, we get our vector embedding of the semantic content of the chunk, enabling semantic comparison. A Vector Database holds the vector embeddings and are stored in a semantic index. For this research, we have used Pinecone as our vector database. Pinecone allows for more efficient searching and retrieval of relevant information that matches the meaning of the data instead of simple keyword matches. Figure 4.2 illustrates RAG Architecture for our KrishiBot

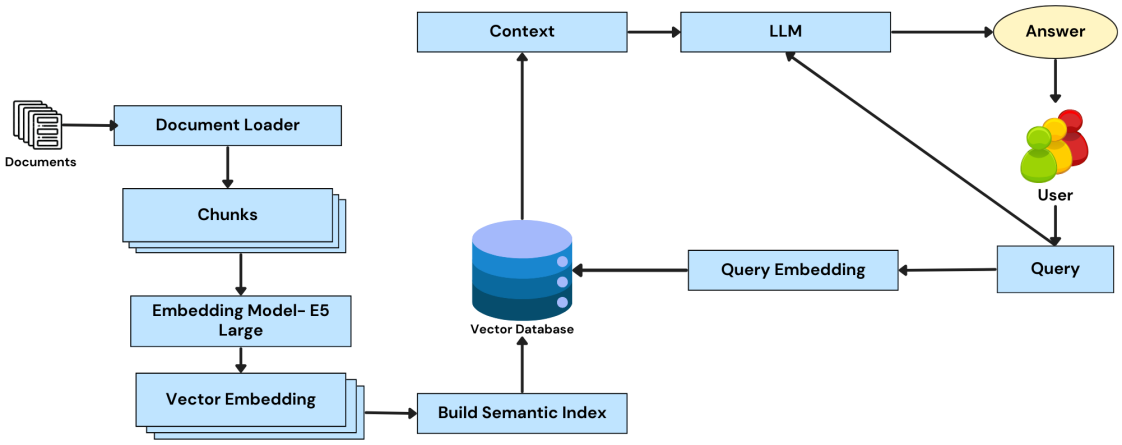



Figure 4.2: NaiveRAG Framework

When a user submits a query, it is converted into a vector representation using the same embedding model applied to segment the document into smaller chunks. These vectors are then compared to identify the most relevant chunks based on semantic similarity. The selected chunks, referred to as the context, are provided to a Large Language Model (LLM), which generates a precise and contextually accurate response grounded in the knowledge base. This response is then delivered to the user in a clear and relevant manner.

## 4.2 KrishiBot - GraphRAG Version

The figure 4.3 below outlines the detailed workflow of our Graph Retrieval-Augmented Generation (GraphRAG) system. It highlights how user queries are processed through embedding generation, precise retrieval of context leveraging the graph database, and the final generation of accurate, context-rich responses optimized specifically for agricultural problem-solving scenarios.

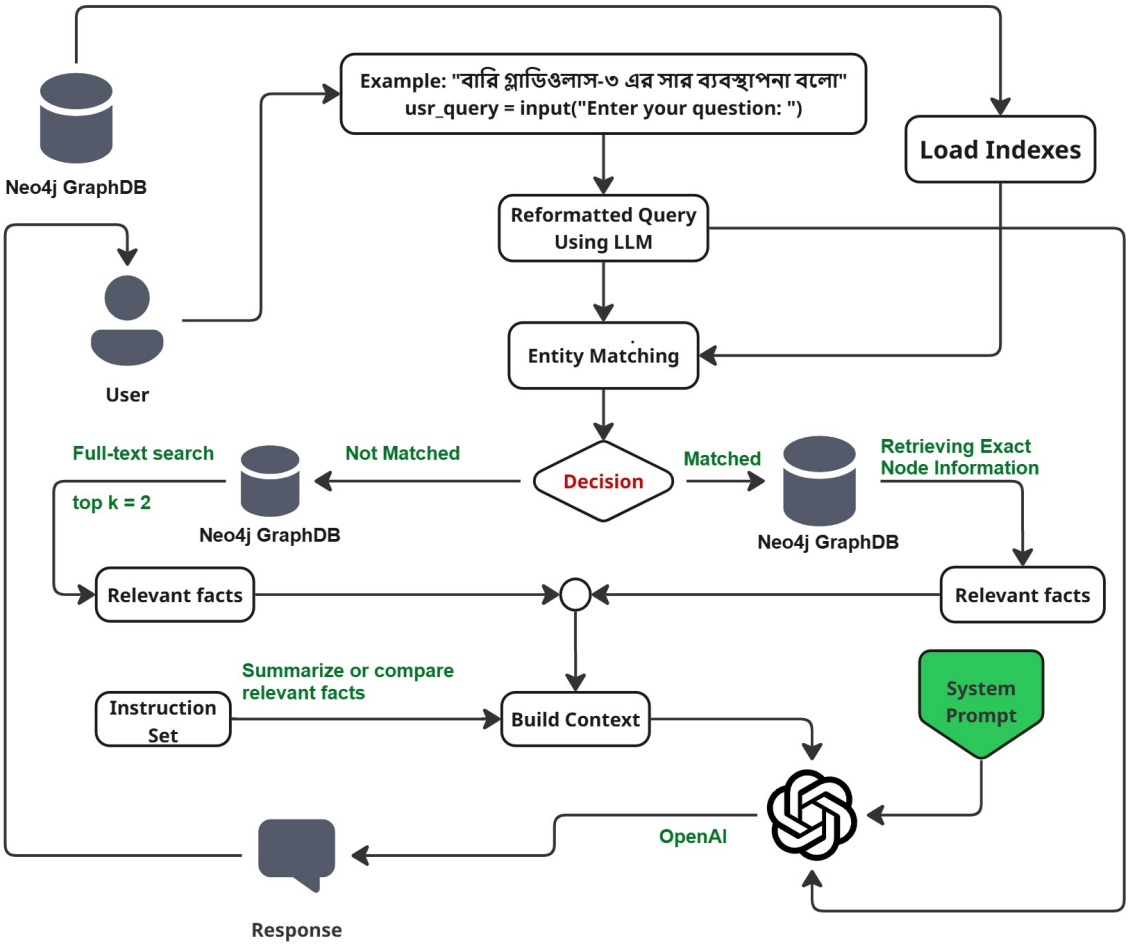



Figure 4.3: KrishiBot: GraphRAG Version

### 4.2.1 Graph-Based Knowledge Integration and Preprocessing

The architecture begins with user queries in Bangla. These queries are interfaced with a Neo4j property graph database that models agricultural knowledge through nodes representing entities such as crop varieties, crops, and agricultural technologies. Each node is enriched with semantic attributes, e.g., variety name or crop name, and connected via structured relationships like 'HAS_VARIETY' (a crop node has a 'HAS_VARIETY' relationship with a variety node) or 'USES_TECHNOLOGY' (a crop node 'USES_TECHNOLOGY' relationship with a technology node). To ensure accurate entity resolution, raw graph data undergo normalization. Entity names are standardized by removing inconsistent whitespaces, invisible Unicode characters, and formatting anomalies. This preprocessing step is crucial as it guarantees alignment between user input and stored entity names, ensuring the system's accuracy. The graph structure supports multi-hop traversal, allowing indirect inference paths (e.g., from a technology node back to a variety via its parent crop), thereby enhancing the depth and flexibility of information retrieval.

### 4.2.2 Query Semantic Structuring via Language Models

Upon receiving a user query, the system utilizes a large language model (LLM) to reformat the input in a semantically meaningful way. The query is evaluated against curated lists of known varieties, crops, and technologies embedded into a prompt that guides the model through a hierarchical identification logic. The LLM prioritizes explicit variety detection; failing that, it searches for crop or technology references, and if no match is found, it preserves the original phrasing. Throughout this process, the system enforces strict lexical integrity, ensuring that entity names appear in their canonical form as recorded in the knowledge base. This includes precise spelling, character encoding, and spacing, ensuring high fidelity in downstream matching. The result is a semantically aligned query that maximizes the likelihood of accurate entity mapping while minimizing ambiguity.

### 4.2.3 Entity Recognition and Knowledge Retrieval

Following query restructuring, the system performs lexical matching between the reformulated query and the set of known entities, using boundary-aware pattern recognition to avoid spurious matches. Entities are prioritized by length to prevent partial collisions. When an entity, such as a crop name, is detected, the system executes semantic traversals across the graph to retrieve relevant nodes and their associated properties. These may include varietal traits such as disease resistance, land suitability, and fertilizer requirements. In cases where no direct match is established, the system falls back on a full-text search mechanism powered by TF-IDF scoring. This method queries pre-indexed nodes across various dimensions, including variety, crop, and technology, and returns top-k results ranked by relevance to the query's lexical features. Keyword extraction from the original input using Bangla token filters ensures that only semantically meaningful terms guide the retrieval, further improving precision.

### 4.2.4 Contextual Compilation and Instructional Control

Once relevant entities are identified, the system constructs a structured textual context comprising factual information derived from the graph. In crop-level queries, information across multiple varieties is collated and filtered based on alignment with keywords extracted from the query itself. Each constructed context is accompanied by a corresponding instruction set that directs the downstream generation phase. These instructions explicitly constrain the language model to focus only on specific entities and disallow reference to out-of-scope information. This design introduces a strong interpretability layer, ensuring that all generated responses can be directly traced back to verifiable knowledge graph nodes.

### 4.2.5 Robust Fallback Mechanism for Unstructured Input (Fallback Strategy)

To support resilience in low-resource and informal input contexts, the system incorporates a fallback strategy that activates when explicit entity matching fails. It performs ranked full-text searches across graph indexes, leveraging Lucene-based scoring to surface semantically adjacent entities. For example, a loosely structured query such as,

"আমার জমিতে নতুন কৃষি প্রযুক্তি কীভাবে ব্যবহার করব?"
(How can I use new agricultural technology on my land?)

Figure 4.4: Example Query

may not map directly to a known node. However, the system can still surface relevant technologies, such as organic fertilizer, by identifying high-relevance matches and enriching them with graph-derived metadata, including associated crops and varieties. This ensures comprehensive coverage, even for non-standard phrasing or novel terminology.

### 4.2.6 Controlled Answer Generation via LLM with Structured Context

Once the context is curated from the graph database based on identified entities—whether they are crop varieties, technologies, or broader crop categories—the system proceeds to the generation phase. This step employs a controlled prompting strategy to ensure that the language model (LLM) produces accurate, context-aware, and hallucination-free answers. At the heart of this phase lies the GPT-4.1-mini model, accessed via OpenAI's API. To maintain precision, the generation process is driven by two critical components: The retrieved context: A structured, plain-text compilation of facts extracted from Neo4j. This context is highly targeted and includes only the information relevant to the matched entities. For instance, in the case of a matched variety, it may include attributes such as fertilizer dosage, disease resistance, seed rate, and irrigation requirements—each neatly formatted in Bangla. A domain-specific instruction prompt: Alongside the context, a set of strict instructions is crafted to guide the LLM. These instructions explicitly constrain the generation scope, specifying which varieties or technologies the answer must focus

on and directing the model not to incorporate any information beyond the provided context. An example instruction might read:

```
"Answer using only facts about the following varieties: বোরো প্রজাতি-৩.
Do not use information about any other varieties."
```

The final prompt sent to the LLM is constructed by concatenating the structured context, the instruction, and the user's original query. Additionally, a system-level prompt is provided to the model to enforce both tone and language constraints. Specifically, it states:

```
"You are a helpful assistant answering based only
 on the given context. Always respond in Bangla."
```

This structured prompting ensures that the LLM operates within strict semantic bounds, leveraging the retrieved context as the sole source of truth. As a result, the system eliminates the risk of fabricated responses—a common pitfall in traditional open-ended LLM deployments.
For instance, consider the user query:

```
"বোরো প্রজাতি-৩ এর সার ব্যবস্থাপনা বলো"
```

The system retrieves the following context and then sent to the LLM to generated answer,

```
সার ব্যবস্থাপনা: প্রতি হেক্টরে ২০০ কেজি ইউরিয়া
জলসেচ: তিন ধাপে
উৎপাদনশীলতা: ৬.৫ টন/হেক্টর
```

Finally, the following generated answer is provided by the LLM after reasoning,

```
"বোরো প্রজাতি-৩ এর ক্ষেত্রে প্রতি হেক্টরে ২০০ কেজি
ইউরিয়া সার প্রয়োগ করা উচিত। জলসেচ তিন ধাপে
নিশ্চিত করতে হবে, যাতে উৎপাদনশীলতা সর্বোচ্চ হয়।"
```

This approach ensures high alignment between the answer and the factual data stored in the knowledge graph. Moreover, since the LLM is explicitly instructed to stay within the retrieved context, the architecture offers both traceability and explainability, essential qualities in high-stakes domains like agriculture. In summary, this generation strategy—anchored in structured graph-derived context and reinforced through instruction-based prompting—enables the system to provide natural, fluent, and accurate Bangla responses that are both reliable and semantically grounded.

# Chapter 5

# Retrieval & Models

## 5.1 Information Retrieval

Our NaiveRAG system integrates LangChain and Pinecone as these tools provide an efficient solution to manage and retrieve complex query information.To add more, The LangChain framework makes it easy for users to integrate language models into custom pipelines that dynamically access data sources and Pinecone provides rapid accurate retrieval with its vector database optimized for similarity search. These technologies work together to make our NaiveRAG system more efficient by extracting precise contextually relevant information in real time which reduces computational demands on the language model and improves accuracy. The combination of these technologies proves highly beneficial for processing vast datasets while providing scalable performance and low latency alongside reliable results which aligns perfectly with our thesis requirements.

## 5.2 Embedding Model - E5 Large_V2

The E5 Large V2 embedding model functions as a tool which converts semantic word meanings into numerical vector representations. In addition, Text processing through this system divides information into tokens while encoding each token into vectors within a defined space. Moreover, Each input receives a 1024-dimensional vector representation through the model's embedding dimension of 1024 that maintains both semantic meaning and contextual understanding. Furthermore, The model operates on text sequences that exceed 512 tokens while maintaining its ability to handle lengthy documents. The endpoint creation allows us to specify the batch size between 1 and 32 according to requirements even though embedding dimension and sequence length remain static. Moreover, The model architecture enables efficient text data processing to support diverse applications including search functionality and classification tasks and similarity assessments. The model operates on text sequences that reach a maximum length of 512 tokens to support extensive documents. The endpoint creation process lets us choose between processing one or thirty-two inputs at a time for batch size yet all other parameters including embedding dimension and sequence length remain unalterable. Overall, The model configuration enables effective processing and analysis of text data that supports applications ranging from search to classification and similarity evaluation.

Figure 5.4 represents the Multilingual E5_Large_V2 embedding model It can process

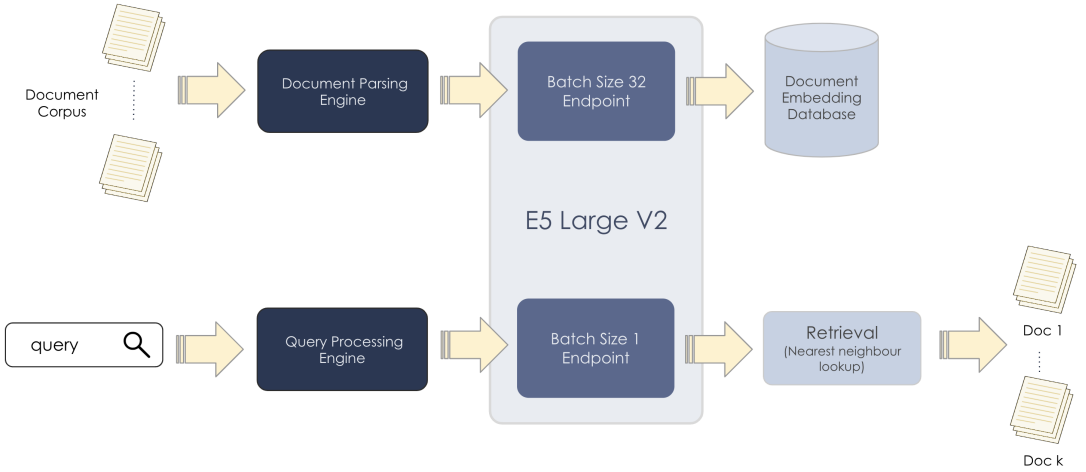



Figure 5.1: Multilingual E5 LARGE_V2

sequences of up to 512 tokens, making it suitable for longer pieces of text. We can adjust the batch size (number of inputs processed at once) to either 1 or 32 when creating the endpoint, depending on our needs, but other parameters like embedding dimension and sequence length are fixed. This setup allows the model to efficiently handle and analyze textual data for various applications like search, classification, or similarity analysis.

## 5.3 LLM 01 - GPT-4.1 Mini

OpenAI's ChatGPT-4.1 Mini is a compact yet high-performance AI language model that strikes a near-perfect balance between performance and computational efficiency. It utilizes multi-layer self-attention mechanisms, a transformer architecture, to generate coherent, contextually relevant, and accurate responses for various language-based tasks. Its most significant strength is that it can process documents or conversations of considerable length, so long as they fit within its maximum context window, which is up to 1,047,576 tokens, exceeding the token limit of most other models. As a result, ChatGPT 4.1 Mini is particularly well-suited for uses that involve long conversations or complex input understanding. The model has a maximum output length of 32,768 tokens and provides detailed and informative answers without requiring excessive computational resources. The number of parameters has also not been disclosed by OpenAI, and as a 'Mini' variant, it has been optimized to run with fewer parameters than the full GPT-4 models, which presents it with faster inference times, lower latency, and lower compute costs. Its efficiency makes it perfect for implementation in an environment where time and resources are the core elements. This versatile ChatGPT is excellent at handling casual conversations, customer support, technical question answering, and summarization. It can be fine-tuned or adapted to specific domains through prompt engineering for highly specialized applications, and ultimately made highly relevant and accurate. However, despite its smaller size, the model still exhibits strong language understanding and generation abilities. However, for highly complex tasks, the size doesn't fit the

nuance or creativity of larger GPT–4 variants. All in all, ChatGPT-4.1 Mini provides a viable and effective solution for AI applications, such as chatbots, virtual assistants, and interactive systems, that require both high language understanding and computational efficiency.

## 5.4 LLM 02 - GPT3.5 Turbo

We used GPT-3.5 Turbo developed by OpenAI, which serves as leading artificial intelligence system. GPT-3.5 Turbo processes text data from books and articles alongside web content within a massive training database to support multiple applications including conversational AI and content creation and data analytical functions. The model provides an expanded context processing capability which extends from 4096 up to 16,385 tokens thereby enabling it to handle complex input data. The system processes detailed and lengthy inputs through a coherent mechanism which spans across extensive interactions. The attention mechanism of the model enables it to zero in on essential input sections which produces contextual outputs with fine-grained accuracy. Although GPT-3.5 Turbo exhibits strong capabilities it struggles with imprecise responses and it faces challenges processing ambiguous commands and contains biases that originated from its training datasets.

## 5.5 LLM 03 - GPT4o

GPT-4o, or GPT-4 "omni," is a powerful and versatile AI model designed for high-intelligence tasks. It is capable of handling both text and image inputs, producing structured and coherent outputs. The model boasts an impressive **context window of 128,000 tokens**, allowing it to process large volumes of information in a single interaction. Additionally, it has a maximum **output token capacity of 16,384 tokens**, enabling the generation of detailed and extensive responses. However, the exact number of parameters—the key factors determining the model's complexity and learning capacity—remains unknown, as OpenAI has not disclosed this information. GPT-4 operates using a transformer-based architecture, where input data is divided into tokens and analyzed in context. By leveraging layers of attention mechanisms, it predicts and generates relevant outputs, understanding complex relationships within the data. This advanced system allows GPT-4 to deliver highly accurate, human-like responses across a wide range of tasks, from creative writing to technical problem-solving.

## 5.6 LLM 04 - Claude 3.5 Haiku

Launched by Anthropic, Claude 3.5 Haiku is an advanced natural language processing model that produces rapid high-speed outputs and demonstrates precision and efficiency. With a context window stretching to 200,000 tokens, this model provides efficient processing for complex conversations, extensive documents, and multi-step reasoning tasks. The speedy processing capabilities of Claude 3.5 Haiku allow it to process 21K+ tokens per second, which translates to 30+ pages of content per second under 32K token prompts. The model proves extremely useful for companies that need fast results from examining big data batches such as user interaction logs

or lengthy official records. The model utilizes advanced attention mechanisms to process essential sequences within inputs, creating reliable and holistic responses. Although useful, the model exhibits constraints such as its inability to understand complex commands and its tendency to repeat itself in specific situations. The tool demonstrates excellence in coding tasks, tool usage, and content creation while being versatile for developers and businesses. Platform services, including Claude.ai, Amazon Bedrock, and Google Vertex AI, serve as model implementations for customers who benefit from economical pricing options for extensive utilization.

## 5.7 LLM 05 - LLAMA3.3(70B-Instruct)

The Meta AI-developed LLaMA 3.3- Instruct model with 70 billion parameters provides performance matching LLaMA 3.1 (405B)'s capabilities while requiring less hardware processing. This version extends the context window to process detailed and lengthy inputs, reaching up to 28,000 tokens. LLaMA 3.3 demonstrates superior performance in text-based operations while providing advanced capabilities in multilingual chat communication, code completion, and synthetic data creation. LLaMA 3.3 works with eight languages, including English, Spanish, Hindi, and Bangla, perfect for handling projects in multiple languages. The model boosts performance through Grouped-Query Attention (GQA) by accelerating text processing speed with minimal computational need. The real-world functionality of LLaMA 3.3 improves by implementing advanced fine-tuning capabilities. Supervised fine tuning (SFT) feeds high-quality responses to train the model, while Reinforcement Learning with Human Feedback (RLHF) enables human assessments to guide behavioral refining of the system. The quantization methods supported by LLaMA 3.3 include 8-bit and 4-bit precision implementations enabled through bits and bytes tools. By employing quantization techniques, developers can reduce their memory usage without sacrificing system performance, allowing the successful operation of common GPUs. The model effectively performs on all benchmark tests while delivering exceptional outcomes in all instruction processing, multilingual analytical tasks, and programming assignments. Although LLaMA 3.3 maintains performance values close to larger models, its efficiency, instant availability, and balanced performance capabilities position it as a flexible solution for multiple application fields. The solution provides practical functionality for applications that need language processing and generation features while maintaining resource efficiency.

## 5.8 Prompt Engineering

### 5.8.1 Prompt Engineering for NaiveRAG

We used this system prompt to ensure our Naive RAG-based chatbot is specifically tailored to address the needs of farmers by providing clear, relevant, and actionable information in Bengali, the primary language of our target audience. The prompt sets the context for the chatbot, guiding it to retrieve information from the vector database using NaiveRAG and generate responses based on that data in a way that aligns with the farmers' problems.

Figure 5.2 demonstrates the Prompt Engineering we have implemented to get more precise and relevant answers.

```
system_prompt = (
    "আপনি একটি কৃষি-ভিত্তিক চ্যাটবট যা OpenAI-এর LLM (ল্যাঙ্গুয়েজ লার্জ মডেল) এবং RAG (রিট্রিভাল-এন্ড-জেনারেশন) "
    "চেইন দ্বারা চালিত। এখানে বিভিন্ন কৃষি সম্পর্কিত বিষয়বস্তু আপলোড করা হবে, যা আপনি আপনার জ্ঞানে "
    "অনুযায়ী প্রশ্নের উত্তর দিতে ব্যবহার করবেন। "
    "\n\n"
    "আপনার কাজ হল কৃষকরা যখন তাদের সমস্যার কথা আপনাকে বলবে, আপনি সেই সমস্যা অনুযায়ী তথ্য সংগ্রহ করবেন "
    "RAG চেইনের মাধ্যমে, যা ভেক্টর ডেটাবেস থেকে তথ্য রিট্রিভ করে এবং LLM সেই তথ্যের উপর যুক্তি তৈরি করে। "
    "তবে, তাদের প্রশ্ন যে ভাষাতেই আসুক (বাংলা অথবা ইংরেজি), আপনার উত্তর অবশ্যই বাংলায় হবে। "
    "\n\n"
    "আপনি যখন কোনও প্রশ্ন পান, তখন দয়া করে যুক্তিসম্মতভাবে চিন্তা করুন, এবং কৃষকদের জন্য পরিষ্কার, "
    "সঠিক এবং কার্যকরী উত্তর প্রদান করুন। আপনার উত্তরে কৃষি বিষয়ক যেকোনো সঠিক এবং প্রাসঙ্গিক তথ্য অন্তর্ভুক্ত "
    "করতে হবে, যেমন সার ব্যবস্থাপনা, পোকামাকড় নিয়ন্ত্রণ, রোগ প্রতিরোধ, ফলন বৃদ্ধি বা স্থানীয় কৃষি পদ্ধতি। "
    "\n\n"
    "আপনার লক্ষ্য হবে কৃষকদের সাহায্য করা যাতে তারা তাদের কৃষির সমস্যাগুলির সমাধান পেতে পারেন, এবং "
    "তাদের স্বার্থে সর্বোত্তম পরামর্শ দিতে সক্ষম হন।"
    "\n\n"
    "{context}"
)
```

Figure 5.2: System Prompt - NaiveRAG

By embedding specific instructions, such as always responding in Bengali regardless of the input language, and focusing on topics like pest control, disease prevention, and yield optimization, we ensure the chatbot delivers practical, localized, and user-friendly solutions. This structured approach enhances the chatbot's efficiency, ensuring it retrieves and processes only relevant information, leading to more accurate, helpful, and context-aware responses, ultimately making it a more effective tool for supporting farmers.

### 5.8.2 Prompt Engineering for GraphRAG

Our proposed GraphRAG approach introduces a query reformatting step, powered by an LLM, to ensure that queries are contextually analysed and precisely mapped to the specific graph database entities—namely, crop varieties, crops, and agricultural technologies. This process is especially crucial in the context of the low-resource, morphologically rich Bangla language, where users may refer to the same crop or variety using multiple spellings or colloquial forms. The system prompt in GraphRAG instructs the LLM assistant to systematically extract and normalise named entities (such as variety, crop, or technology) from the user's query, constantly referencing a predefined, domain-specific list.
Figure 5.3 represents a portion of our system prompt.

আপনি একজন সহকারী যিনি কৃষি বিষয়ক ব্যবহারকারীর প্রশ্নকে সঠিকভাবে পুনর্গঠন করেন, যাতে জাত (variety), ফসল (crop), অথবা প্রযুক্তি (technology/projukti) এর নাম স্পষ্ট এবং নির্ভুলভাবে উল্লেখ থাকে।
আপনার কাজ:
- ব্যবহারকারীর প্রশ্ন থেকে জাত, ফসল, অথবা প্রযুক্তির নাম চিহ্নিত করা.
- শুধুমাত্র সংশোধিত প্রশ্ন অথবা নির্দিষ্ট বার্তা (নিচে দেখানো হয়েছে উদাহরণ) ফেরত দেয়া
- সব সময় তৃতীয় পক্ষে(third person) এ প্রশ্ন পুনর্গঠন করবেন।
নিচের নিয়মগুলো অনুসরণ করুন (সংশ্লিষ্টটি না পাওয়া পর্যন্ত পর্যায়ক্রমে অনুসন্ধান করুন):

১. **প্রথমে জাতের নাম (variety) সনাক্ত করুন**। যদি প্রশ্নে জাতের নাম স্পষ্ট বা আংশিকভাবে (শর্ট/অপভ্রংশ/ইংরেজি- বাংলা মিশ্র) উল্লেখ থাকে, তাহলে সেটি {varieties_str}
তালিকা থেকে পুরো নাম নিয়ে যুক্ত করুন এবং প্রশ্নটি সেই জাতের ওপর পুনর্গঠন করুন।এক্ষেত্রে খেয়াল রাখতে হবে যে
প্রশ্ন টার উত্তর দিতে কি আসলে আমাদের শুধুর একটি জাতের নাম জানলেই হবে , নাকি প্রশ্নটার উদ্দেশ্য বরং যতগুলো জাত আছে সব গুলার উপর ভিত্তি করে উত্তর দেয়া টা বেশি যৌক্তিক। সেক্ষেত্রে এখানে {varieties_str}না বসিয়ে বরং {crops_str} দিয়ে বিচার করবেন। সকল ক্ষেত্রেই আপনাকে উত্তর দেয়ার জন্য যুক্তি চিন্তা হবে , কোন context টা বেশি উপযোগী।
২. **জাত না থাকলে প্রথমে আলাদা করে ফসলের নাম (crop) এবং প্রযুক্তি (technology/projukti) সনাক্ত করার চেষ্টা করুন **।এবার প্রাপ্ত কাছাকাছি জাত(crop) এবং প্রযুক্তি(technology/projukti) এর মধ্যে যেটা প্রশ্নের ধরন এবং বক্তা যা বুঝাতে চেয়েছেন সেটার সাথে মানানসই , সেটা choose করুন।এক্ষেত্রে কিছু ক্ষেত্রে (crop) এর প্রাধান্য থাকতে পারে। ফসলের নাম দিলে -{crops_str} তালিকা থেকে পুরো নাম নিয়ে, crop-level এ প্রশ্নটি পুনর্গঠন করুন।অথবাপ্রযুক্তি নাম দিলেয - কোনো বিদ্যমান প্রযুক্তি (প্রযুক্তির নাম) বা তার খুবই ঘনিষ্ঠ ও নির্দিষ্ট কোনো শব্দ/বর্ণনা পাওয়া যায়, তাহলে সেটি {techs_str} তালিকা থেকে হুবহু প্রযুক্তির নাম নিয়ে, প্রযুক্তি ভিত্তিক প্রশ্নটি পুনর্গঠন করুন।
৩. উপরের কোনোটিই না মিললে (variety/crop/technology)–তাহলে যেমন প্রশ্ন আছে, সেটাই ফেরত দিন।

Figure 5.3: System Prompt GraphRAG 01

This ensures exact spelling and entity matching, which is essential for accurate graph traversal and information retrieval. Moreover, the prompt mandates that all responses be reconstructed in the third person and Bangla, maintaining consistency and clarity for end users. When appropriate, the system guides the LLM to decide whether a query should be answered at the variety level or crop level, depending on the specificity and context of the user's question, thus maximising the informativeness and relevance of the system's responses.
Figure 5.4 represents another portion of our system prompt.

```
***Ground Rule: After finding probable crop name or variety name or techonology name***

1.Remember One thing , never ever change the exact crop or variety name ever when replacing with found name, if
there is a space in the crop names or variety names- it should be be as it is.
like if there closest found is "বাংগি" then it should be exactly "বাংগি" , not "বাঙ্গি"(no name manipulation ,
always think crops_str,varieties_str spelling is correct) or " বাঙ্গি" or " বাঙ্গি " or "  বাংগি "(no extra space)
ever.

2.in another case if two probable same crop/variety name has two different way of representation/spelling like -
for variety there is "বাংগি" and "বাঙ্গি" , both available in the {varieties_str}. if the user does not mispell ,
then keep the one that is identical.
if he mispells then pick whatever you think is suited for the question.
```

Figure 5.4: System Prompt - GraphRAG 02

A notable property in this prompt is the bilingual reformatting rule: after the primary entity extraction and normalisation steps, the prompt requires the LLM to translate the question into English, but retain the identified crop, variety, or technology names in Bangla script within the translation. This not only enables robust alignment between user intent and graph database entities but also facilitates further downstream processing and auditability. Additional ground rules strictly enforce exact entity spelling and spacing as found in the database, preventing error propagation due to minor variations or user typos. By embedding these detailed and rigorous instructions within the system prompt, our GraphRAG solution achieves a higher level of control and faithfulness in query understanding, ultimately enhancing the relevance, accuracy, and reliability of responses provided to farmers.

# Chapter 6

# Evaluation & Performance

## 6.1 Evaluation Matrices

The evaluation section examines multiple assessment methods to measure how effectively our model retrieves information and its response precision. We divided the metrics into three primary categories: **Retrieval metrics, Generation Metrics, and Comprehensive metrics**. We also did Domain-specific evaluations using all three primary categories.

### 6.1.1 Retrieval Metrics

#### Context Precision for Naive RAG

The Context precision [6] metric is an evaluation tool that determines how well-retrieved content compares to established ground truth labels regarding their placement within the results list. The measurement checks if critical contexts surface towards the beginning of retrieved document rankings because this capability directly supports information retrieval systems. The formula for context precision at K, where K represents the number of top-ranked retrieved-context considered for evaluation, is defined at equation 6.1.

$$\text{Context Precision} = \frac{\sum_{k=1}^{n} \text{Precisionk}_k \times \text{V}_k}{\text{Total Retrieved Contexts}} \tag{6.1}$$

Where Precision@K is defined is defined in 6.2:

$$\text{Context Precision@K} = \frac{\text{True Positives@K}}{\text{True Positives@K} + \text{False Positives}} \tag{6.2}$$

This metric enables precision evaluation at multiple retrieval levels using Vk to weight relevance factors. Precision measurement accumulates across ranks through the numerator expression, but normalization occurs through the relevant context in the top K items in the denominator. A refined ranking assessment compared to simple precision emerges from this metric because it incorporates both retrieved element positions and their correct contextual relevance.

### Context Recall for Naive RAG

Context recall [6] evaluates context accuracy through an assessment that matches retrieved facts along with claims against ground truth—the value scale sets between 0 and 1, with higher numbers denoting improved results. A set of claims in the ground truth is used to calculate this matrix. The system evaluates retrieved-context to confirm if each identified claim exists within the document content. A perfect context recall appears whenever each claim from the ground truth seems present in the retrieved context. Its calculation formula is as followed in equation 6.3

$$\text{Context Recall} = \frac{\text{No. of Ground Truth Supported by Retrieved Context}}{\text{Total number of claims in the reference}} \tag{6.3}$$

### Context Precision for Graph RAG

In our GraphRAG-based agricultural chatbot system, Context Precision plays a vital role in evaluating how accurately the retrieved nodes (varieties) align with expert-labeled relevant answers. While NaiveRAG systems emphasize document-level or passage-level retrieval metrics, we adapted the Context Precision formula specifically for node-level evaluation. This modification ensures that the metric directly reflects how well the graph-based retriever identifies the correct variety nodes in response to a user query—crucial for real-world farming applications where listing accurate crop varieties can significantly impact decision-making. Its calculation formula is as followed in equation 6.4

$$\text{Context Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{6.4}$$

Where,

**True Positive (TP):** Node (crop variety) retrieved by the system that is also listed in the expert-validated correct set for that question.
**False Positive (FP)**: Node retrieved by the system that is not part of the expert-validated answer set and should not have been included.

Unlike classical RAG setups that retrieve entire documents or passages, GraphRAG retrieves individual nodes from a knowledge graph—each representing a specific crop variety. This granularity requires a more targeted evaluation approach. We redefined the precision formula to focus on retrieved nodes only, ignoring irrelevant graph structure or uncalled-for entities, so the evaluation remains sharply aligned with the goal: *Did the system retrieve the correct varieties?* This node-focused adjustment makes precision a much better fit for agricultural query evaluation in GraphRAG.

### Context Recall for Graph RAG

In our agricultural GraphRAG system, Context Recall measures the system's ability to comprehensively retrieve all relevant variety nodes for a given user query. Since the main objective is to ensure that correct crop varieties are not omitted from the response of the system, context recall serves as a direct indicator of completeness. Unlike traditional RAG recall, which typically focuses on retrieving correct documents or passages, our node-centric setup necessitated a refined recall calculation that aligns with the granularity of variety-level retrieval. This makes context recall especially important in agricultural scenarios, where omitting a relevant variety can lead to suboptimal or incorrect guidance to farmers. Its calculation formula is as followed in equation 6.5

$$\text{Context Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{6.5}$$

Where,
**True Positive (TP):**Node (crop variety) retrieved by the system that is also listed in the correct set verified by experts for that question.
**False Negative (FN):** Node that should have been retrieved according to expert validation but was missed by the system.

### F1 score

When calculated through harmonic mean, the F1 score represents the intermediate value between precision and recall values [6]. A single evaluation score combines precision and recall into a balanced value, weighing false positives and false negatives. The formula of F1 score is as defined in equation 6.6

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Context Precision} + \text{Context Recall}} \tag{6.6}$$

## 6.1.2 Generation Matrices

### Faithfulness

The Faithfulness score [6] evaluates factual consistency between generated outputs and retrieved contexts across a 0 to 1 value scale, with higher scores indicating a stronger factual connection. A response passes faithfulness standards when every claim inside it finds support in the retrieved context. The evaluation method collects claims from the generated response before establishing their background support in the retrieved context to calculate the accuracy percentage. Faithfulness acts as an essential measure that guarantees that retrieved data maintains its factual accuracy in the original context during information retrieval evaluation procedures. The formula for Faithfulness is as defined below in equation 6.7

$$\text{Faithfulness Score} = \frac{\text{RS}}{\text{TR}} \tag{6.7}$$

Where,

RS = Number of Response Supported by Retrieved Context
TR=Total Number of Claims in The Response

#### Semantic Similarity

In addition to verifying that our answers are correct, we also attempted to understand how similar our answers are to the ground truth answer, even when the vocabulary differs. Since precision and recall don't address this aspect of the problem, we employed a semantic similarity measurement. The answer produced by RAG systems for every question was compared with the expert's answer using Bangla-BERT embeddings. The two answers were expressed as vectors, and we determined their cosine similarity. By using this style, we find out if our answers mean the same as the reference's intent, even if the terms are not the same. In real-world situations, semantic similarity is crucial because accurate answers can convey different ideas in various ways. By averaging semantic similarity scores for each question, we obtain a new method to assess how effectively GraphRAG conveys expertise in natural language. The formula for Semantic Similarity is as defined below in equation 6.8

$$\cos(\theta) = \frac{\sum_{i=1}^{n} \text{word1}_i \cdot \text{word2}_i}{\sqrt{\sum_{i=1}^{n} \text{word1}_i^2} \cdot \sqrt{\sum_{i=1}^{n} \text{word2}_i^2}} \tag{6.8}$$

### 6.1.3 Comprehensive Matrices - QAEvalChain

The QAEvalChain [6] invokes benchmarking predicted answers against ground truth to determine assessment outcomes using LLM(GPT 4.1Mini) in build libraries. The QAEvalChain operates with two necessary inputs: examples containing actual answers and predictions that emerge from model computations. The evaluation process employs semantic comparison methods alongside alternative measures to determine the accuracy between actual and predicted answer outputs. Evaluation results are combined with correct or incorrect indicators through threshold values in the function's output.

## 6.2 Performance

### 6.2.1 Benchmarking Model Using NaiveRAG Performance

To benchmark model performance using the Naive RAG approach, we selected a representative **subset of 200 test queries** from our full 500-question test dataset. This subset was carefully chosen to ensure a balanced distribution of query types and difficulty levels. The purpose of this evaluation was to **identify the most suitable large language model** based on retrieval accuracy, response relevance, and computational efficiency. By narrowing the scope, we were able to conduct focused and controlled comparisons across multiple models while maintaining statistical significance in our results.

#### Context Precision & Recall

We did a thorough analysis using **retrieval, generation, and comprehensive** evaluation metrics to validate and evaluate our chatbot for the agriculture sector. Context recall, context precision, and F1 score were used to **measure the retrieval performance** as well, which ensures the accuracy and relevance of retrieved chunks from embedded documents. The chatbot achieved a **recall of 70.8%**, meaning it

could recover much pertinent information. Analysis of a **precision score of 75.0%** for avoiding irrelevant chunks and a **combined F1 of 72.9%** for both precision and recall shows the overall balance between them. Furthermore, these retrieval metrics confirm that the multilingual E5 embedding model can appropriately retrieve agricultural knowledge. Table 6.1 illustrates performance over 8 major categories,

| **Metrics** | **Result** |
|---|---|
| Context Precision | 75.0% |
| Context Recall | 70.8% |
| F1 Score | 72.9% |

Table 6.1: Overall Retrieval Performance

### Faithfulness & Semantic Similarity

We attended to the faithfulness metric for the generation performance, which reviews the chatbot's responses' alignment against the retrieved content shown in the figure 7.1. The highest faithfulness score was 0.80 with GPT 4.1mini, 0.75 with GPT 4o, GPT 3.5 Turbo scored 0.70, Claude 3.5 Haiko and LLaMA scored 0.67. It shows that GPT 4.1mini is better at maintaining fidelity with the retrieved context in its generated responses. To better understand the quality of the generated outputs, we also use cosine similarity (semantic alignment) to assess them. Aside from GPT 4o, GPT 4.1mini Turbo had the highest cosine similarity score of 0.85, while GPT 4o scored 0.64, Claude 3.5 Haiko scored 0.60, GPT 3.5Turbo scored 0.65 and LLaMA scored 0.60. In particular, these results suggest that GPT 4o keeps the responses accurate and contextually relevant, while GPT 3.5 Turbo is a bit better at capturing semantics.

Figure 6.1 represents avverage faithfulness of answers across different models,

| Category | Score |
|---|---|
| GPT 4.1Mini | 0.80% |
| GPT 4o | 0.64% |
| GPT 3.5 | 0.65% |
| Claude 3.5 | 0.67% |
| LlaMa 3.3 | 0.67% |

Table 6.2: Average Faithfulness Over Each Categories

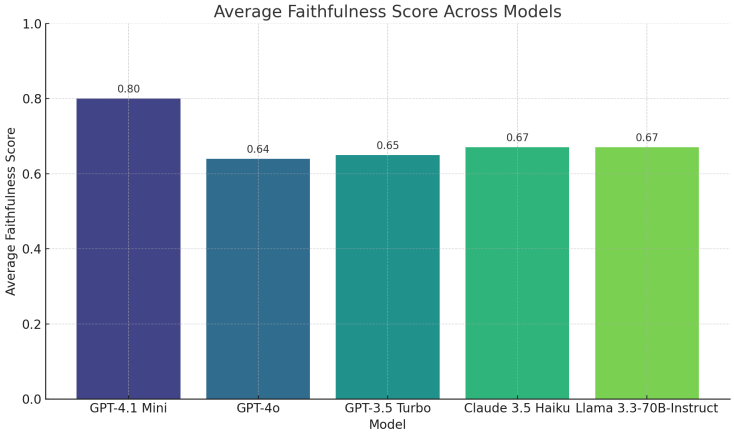



Figure 6.1: Average Faithfulness Over Each Categories

### QA Evaluation

We also examined the QA evaluation metric, which measures how accurately the chatbot's answers align with our 500-question test set (Table 6.2). GPT 4.1Mini

| Category | Score |
|---|---|
| GPT 4.1Mini | 78.39% |
| GPT 4o | 74.37% |
| GPT 3.5 | 59.30% |
| Claude 3.5 | 59.30% |
| LlaMa 3.3 | 64.82% |

Table 6.3: QA Evaluation Across Different Models

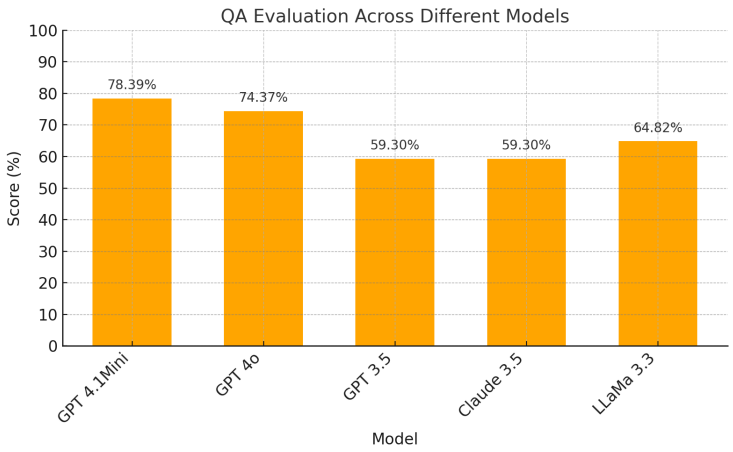



Figure 6.2: QA Evaluation Across Different Models

achieved the highest QA accuracy at 78.39%, followed by GPT 4o at 74.37%. LLaMA 3.3 scored 64.82%, while both GPT 3.5 and Claude 3.5 tied at 59.30%. These results indicate that GPT 4.1Mini delivers the most accurate responses overall, with GPT 4o close behind and GPT 3.5 and Claude 3.5 showing lower answer reliability.

### Domain Specific Evaluation

To ensure the chatbot's responses align with agricultural sector requirements, a domain-based evaluation was conducted across eight crop categories: The domain includes diverse crops such as spices, flowers, fruits, vegetables, oil seeds, grain crops, pulse crops, and kandal crops. The evaluation utilized several performance metrics that measured retrieval performance through recall, precision and F1 score, response faithfulness across diverse models, and QaEval correctness measurement.

### Domain Specific Context Precision & Recall

The chatbot showed proficient information retrieval in varying kinds of agricultural domains. The system displayed its best recall performance with oil seeds at 0.75, grain crops maintaining a 0.83 recall score, fruits receiving 0.76, and spices achieving 0.75 recall scores. A system performance review showed vegetables and pulse crops achieved the lowest recall scores at 0.67 and 0.68 while presenting opportunities to enhance information retrieval for these areas. The precision scores of retrieved information achieved their peaks for flowers 0.91, oil seeds 0.85, and grain crops 0.86 but revealed lower numbers for vegetables 0.67 and pulse crops 0.65. The results indicate the chatbot successfully searches agricultural domain knowledge but needs additional adjustments to better handle pulse crop and vegetable search requests. The F1 score analysis shows that the chatbot shows its most incredible efficiency when responding to questions about oil seeds 0.87, grain crops 0.84, and flowers 0.81. The F1 scores indicated stable retrieval performance with room to improve for Kandal crops 0.77 as well as spices 0.77 and fruits 0.76. The F1 scores demonstrated that pulse crops and vegetables recorded lower performance 0.66 and 0.67, respectively, indicating the requirement for improved precision and recall mechanisms to achieve balanced agricultural information retrieval. Table 6.4 represents the sum-

marized context relevance performance & figure 6.3 illustrates context precision and context recall for each domain.

| Category | Recall | Precision | F1 Score |
|---|---|---|---|
| Spices | 0.75 | 0.78 | 0.77 |
| Flowers | 0.73 | 0.91 | 0.81 |
| Fruits | 0.76 | 0.86 | 0.76 |
| Vegetables | 0.67 | 0.66 | 0.67 |
| Oil-seeds | 0.84 | 0.85 | 0.87 |
| Grain-crops | 0.83 | 0.86 | 0.84 |
| Pulse-crops | 0.68 | 0.65 | 0.67 |
| Kandal-crops | 0.89 | 0.84 | 0.77 |

Table 6.4: Domain Specific Context Recall, Precision, and F1 Score

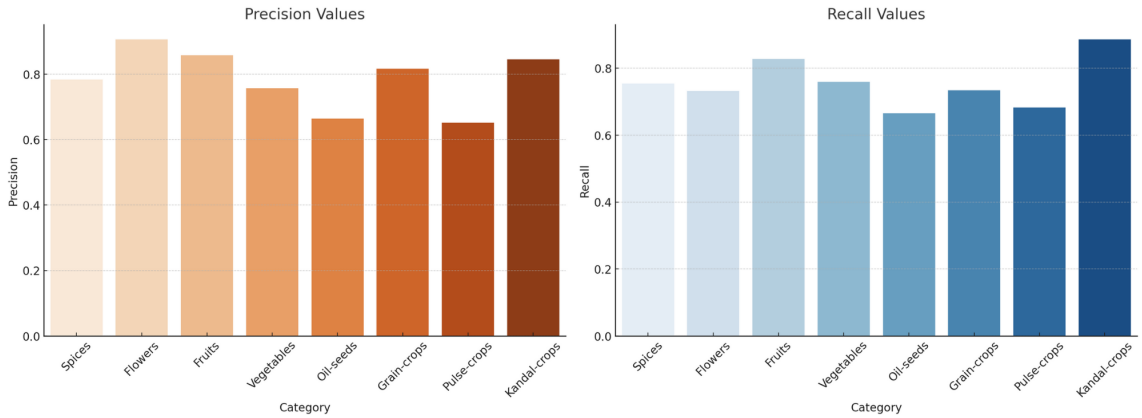



Figure 7.1: Precision Score Over Each Categories

Figure 7.2: Recall Score Over Each Categories

Figure 6.3: Domain Specific Context Precision & Recall of Naive RAG

### Domain Specific Faithfulness

The evaluation of generated response alignment with original content through faithfulness assessments yielded distinct results for different model types, as presented in Table 6.3. GPT-4.1 Mini achieved the highest overall performance across most categories, notably excelling in Kandal-crops (0.87), Grain-crops (0.82), Fruits (0.79), and Flowers (0.75). It also tied with GPT-4o for the best score in Oil-seeds (0.77). GPT-4o demonstrated robust faithfulness, particularly in Spices (0.81), and showed consistently strong results in Oil-seeds (0.77) and Pulse-crops (0.76). GPT-3.5 Turbo maintained intermediate performance, achieving its peak faithfulness in Spices (0.756) but displaying lower scores for Grain-crops (0.616) and Pulse-crops (0.649). Claude 3.5 Haiku consistently showed the weakest alignment, with the highest faithfulness recorded for Pulse-crops (0.66) and the lowest for Vegetables (0.56).
LLaMA 3.3-70B demonstrated steady performance with particularly strong results in Pulse-crops and Grain-crops (both 0.77) and Flowers (0.75). Overall, GPT-4.1 Mini provided the most contextually faithful outputs, followed closely by GPT-4o, whereas Claude 3.5 Haiku consistently offered the least alignment with original content.
The faithfulness of each domain is illustrated through figure 6.4 below:

| Category | GPT4.1Mini | GPT4o | GPT3.5 | Claude 3.5 | LLaMA3.3 |
|---|---|---|---|---|---|
| Spices | 0.72 | 0.81 | 0.756 | 0.57 | 0.74 |
| Flowers | 0.75 | 0.74 | 0.680 | 0.60 | 0.75 |
| Fruits | 0.79 | 0.76 | 0.681 | 0.61 | 0.67 |
| Vegetables | 0.68 | 0.69 | 0.680 | 0.56 | 0.72 |
| Oil-seeds | 0.77 | 0.77 | 0.751 | 0.59 | 0.76 |
| Grain-crops | 0.82 | 0.76 | 0.616 | 0.64 | 0.77 |
| Pulse-crops | 0.75 | 0.76 | 0.649 | 0.66 | 0.77 |
| Kandal-crops | 0.87 | 0.84 | 0.711 | 0.61 | 0.75 |

Table 6.5: Domain Specific Faithfulness Scores Across Different Models

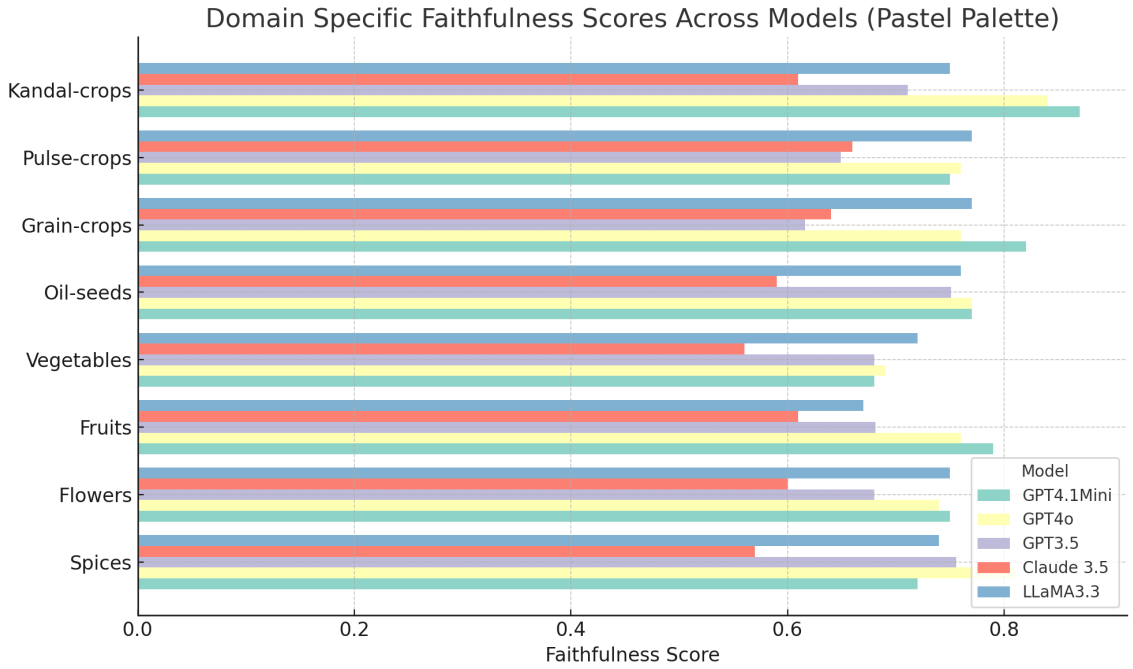



Figure 6.4: Domain Specific Faithfulness Across Models of LLM

### Domain Specific QA Evaluation

The QA evaluation analysis quantifies each model's accuracy in responding to domain-specific agricultural questions by reporting the correctness ratios. As shown in Table 6.4, GPT-4.1 Mini consistently achieved the highest QA evaluation scores across most domains, leading in fruits (95.73%), flowers (92.32%), and vegetables (87.85%). It also maintained strong performance in kandal-crops (72.97%), spices (74.50%), and grain-crops (73.62%). GPT-4o closely followed, particularly excelling in kandal-crops (78.95%) and maintaining high scores in fruits (94.73%), flowers (88.24%), and vegetables (82.05%). However, its accuracy was comparatively lower in oil-seeds (62.5%) and pulse-crops (66.66%).
GPT-3.5 Turbo exhibited moderate accuracy, peaking in fruits (77.77%) and kandal-crops (73.68%), but it underperformed in pulse-crops (50.00%) and grain-crops (47.05%). Claude 3.5 Haiku showed its best result in fruits (89.47%) and above-average scores in vegetables (71.79%), yet overall correctness ratios remained relatively low, particularly in kandal-crops (57.89%) and pulse-crops (50.00%). LLaMA

3.3-70B provided its best performance in kandal-crops (78.95%), flowers (82.35%), and grain-crops (73.68%) but struggled to achieve higher accuracy in vegetables (71.79%), oil-seeds (54.17%), and grain-crops (41.18%).
Overall, GPT-4.1 Mini was the most accurate model across the majority of agricultural domains, while GPT-4o and LLaMA 3.3-70B showed strong but slightly less consistent results. Claude 3.5 Haiku and GPT-3.5 Turbo generally lagged behind in domain-specific correctness ratios. The complete QA Evaluation for each domain is illustrated in Table 6.4 and Figure 6.5.
The complete QA Evaluation of each domain is illustrated in table 6.6 & figure 6.5

| Category | GPT4.1Mini | GPT-4o | GPT-3.5 | Claude | LLaMA |
|---|---|---|---|---|---|
| Spices | 74.50 | 70.83 | 58.33 | 54.17 | 62.5 |
| Flowers | 92.32 | 88.24 | 70.58 | 62.5 | 82.35 |
| Fruits | 95.73 | 94.73 | 77.77 | 89.47 | 73.68 |
| Vegetables | 87.85 | 82.05 | 65.79 | 71.79 | 71.79 |
| Oil-seeds | 71.58 | 62.5 | 48.93 | 52.08 | 54.17 |
| Grain-crops | 73.62 | 64.71 | 47.05 | 41.18 | 41.18 |
| Pulse-crops | 61.59 | 66.66 | 50.00 | 50.00 | 62.5 |
| Kandal-crops | 72.97 | 78.95 | 73.68 | 57.89 | 78.95 |

Table 6.6: QA Evaluation Scores for Different Models

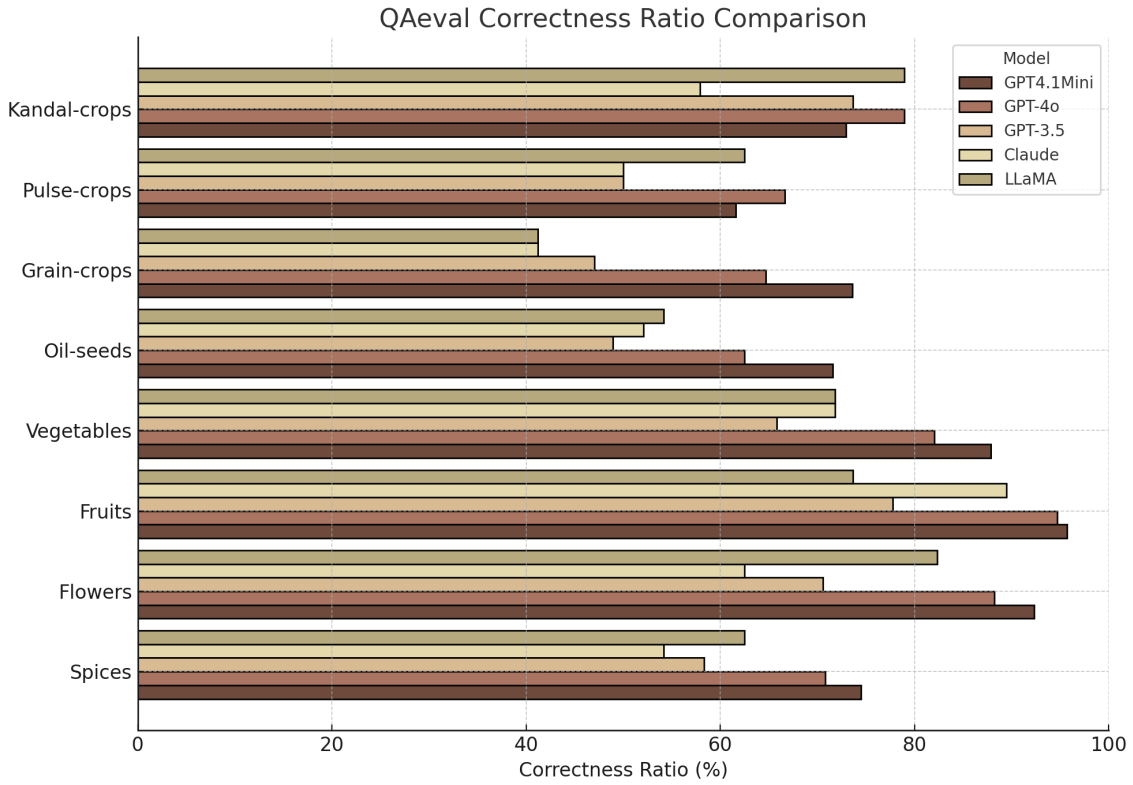



Figure 6.5: Domain Specific QA Evaluation

To conclude, the results highlight that model performance across different agricultural categories is notably variable. In terms of both faithfulness and QA evaluation, GPT-4.1 Mini emerges as the most accurate and reliable model, consistently leading near the top across most domains—especially in fruits, flowers, and vegetables. GPT-4o also demonstrates strong overall results, excelling in kandal-crops, while showing robust faithfulness and correctness in several domains. LLaMA 3.3 70B provides solid performance in specific categories, such as pulse crops and kandal-crops, but is less consistent overall. GPT-3.5 Turbo exhibits moderate reliability, with its strengths appearing in kandal-crops and fruits, but with notable drops in categories like pulse and grain crops. Claude 3.5 Haiku consistently ranks as the weakest model in both faithfulness and QA evaluation metrics.

### 6.2.2 NaiveRAG Performance

As GPT-4.1 Mini demonstrated superior performance in both accuracy and cost-efficiency, it was selected as the final generation model for our system. Following this, we conducted rigorous testing on the full 500-question test dataset using GPT-4.1 Mini to compare the effectiveness of two retrieval strategies: Naive RAG and GraphRAG. This extensive evaluation allowed us to determine the optimal pipeline configuration for delivering accurate and context-aware responses in our agricultural chatbot system.

#### Context Precision, Recall & F1 Score

We did a thorough analysis using retrieval, generation, and comprehensive evaluation metrics to validate and evaluate our chatbot for the agriculture sector. Context recall, context precision, and F1 score were also used to measure retrieval performance, ensuring the accuracy and relevance of retrieved chunks from embedded documents. The chatbot achieved a **recall of 74.42%**, indicating that it could recover a significant amount of pertinent information. An analysis of a **precision score of 74.56%** for avoiding irrelevant chunks and a combined **F1 score of 70.02%** for both precision and recall demonstrates the overall balance between them. Furthermore, these retrieval metrics confirm that the multilingual E5 embedding model can appropriately retrieve agricultural knowledge.

#### Faithfulness

We measured faithfulness, ensuring that the generated answers were faithful to the retrieved context, such as verifying that the generated answers contained no hallucinations. **The faithfulness score was 66.37%,** indicating that while most of these responses remained the same as the source documents, there was also a certain degree of slight factual deviation. For complex or multifaceted queries, this suggests further optimization to ensure strict adherence to retrieved facts.

| Metrics | Result |
|---|---|
| Context Precision | 74.56% |
| Context Recall | 74.42% |
| F1 Score | 70.02% |
| Faithfulness | 66.37% |
| QA Evaluation | 78.60% |
| Semantic Similarity | 80.66% |

Table 6.7: Overall Performance of NaiveRAG

### QA Evaluation & Semantic Similarity

We also conducted a question-answering (QA) evaluation to determine the chatbot's ability to generate correct answers to agricultural questions using a test set of 500 questions. **The QA evaluation score of the system is 78.6%**, indicating that it can produce responses highly relevant to the expected answers. It is thus this score that represents how well the model refined retrieved information and produced contextually relevant and coherent replies concerning the agriculture domain. Furthermore, we also examined the semantic closeness between generated answers and the ground truth responses by using semantic similarity metrics. Additionally, **the chatbot achieved a high semantic similarity score of 80.66%**, indicating that the generated content closely aligns with its intended meaning, regardless of the exact wording used. This demonstrates the chatbot's ability to maintain the essential aspects of information and intent in responses, which is a crucial aspect for users to understand in the agricultural context.

### Domain Specific Context Precision, Recall, F1 Score

From the domain precision evaluations, the highest scores were recorded in Kandal-Crops (0.92), Oil-seeds (0.91), and Technology (0.87), indicating that the model was more precise in retrieving exactly relevant responses in these domains. Fruits (0.78), Vegetables (0.78), and Pulse Crops (0.74) displayed moderate precision as they did not have enough response relevance issues, but Spice (0.74), Flowers (0.73), Grain Crops (0.67) and mainly Rice (0.59) showed lower value which indicate that there are some response relevance issues for those categories.

Upon mentioning the recall, it was found that the model exhibited excellent retrieval ability in Kandal-Crops, Oil-seeds, and Technology (0.92, 0.89, and 0.86, respectively). At the same time, Fruits (0.82) and Flowers (0.77) were close to it. Although performance was moderate, there was the least performance in Rice (0.55), with performance that was not so optimal in other crops also such as Grain-Crops (0.75), Vegetables (0.75), Spice (0.74), and Pulse-Crops (0.73), made evident by the fact that relevant responses in the Rice domain were frequently missed.

The F1 scores were the highest for Pulse-Crops (0.8891), closely followed by Kandal-Crops (0.88) and Oil-seeds (0.89), which were a mature performance in these dimensions. Moreover, Technology (0.83) and Fruits (0.77) performed exceptionally well, with moderate F1 scores measured for Flowers (0.71), Vegetables (0.72), and Spice (0.68). Grain crops (with an F1 score of 0.65) and Rice (0.52) were the domains with the lowest F1 scores, confirming that the Rice domain was overall weak. Finally, as far as the Chatbot's efficacy is concerned, it scores very high in domains such as Kandal crops, Oilseeds, and Technology. Still, it performs substantially low in the Rice domain across each of these three metrics. Hence, its model and data need to be augmented in a necessary area of enhancement to achieve better performance. Table 6.8 represents all the scores obtained of each category.

| Category | Precision | Recall | F1 Score |
|---|---|---|---|
| Spices | 0.74 | 0.74 | 0.68 |
| Flowers | 0.73 | 0.77 | 0.71 |
| Fruits | 0.78 | 0.82 | 0.77 |
| Vegetables | 0.78 | 0.75 | 0.72 |
| Oil-seeds | 0.91 | 0.89 | 0.89 |
| Grain-crops | 0.67 | 0.75 | 0.65 |
| Pulse-crops | 0.74 | 0.73 | 0.89 |
| Kandal-crops | 0.92 | 0.92 | 0.88 |
| Rice-crops | 0.59 | 0.55 | 0.52 |
| Technology | 0.87 | 0.86 | 0.83 |

Table 6.8: Domain Specific Context Recall, Precision, and F1 Score of NaiveRAG

### Domain Specific Faithfulness

In respect of the assessment of domain faithfulness, which evaluates factually grounded and consistent responses based on domain-specific information, Technology (0.82) proved to be the top scorer and a highly assured response category. Spice (0.73), Oil-seeds (0.74), and Pulse-crops (0.73) were the following with a moderate degree of factual grounding. A slight decrease in performance was observed in Kandal for crops (0.70), Vegetables (0.70), and Flowers (0.67). According to Fruits (0.64), Grain – Crops (0.66), and, more specifically, Rice (0.51), the lowest domain faithfulness was observed, indicating that hallucinations or irrelevant content can occur in these responses. All these results underscore that the chatbot has very stable factual integrity in technical and oilseed domains, but needs to improve significantly in the Rice category, as it consistently performs poorly in all three evaluation dimensions. Table 6.9 represents the summarized Faithfulness performance & figure 6.6 illustrates Faithfulness for each domain.

| Category | Score |
|---|---|
| Spices | 0.73 |
| Flowers | 0.67 |
| Fruits | 0.64 |
| Vegetables | 0.70 |
| Oil-seeds | 0.74 |
| Grain-crops | 0.66 |
| Pulse-crops | 0.73 |
| Kandal-crops | 0.70 |
| Rice-crops | 0.51 |
| Technology | 0.82 |

Table 6.9: Domain Specific Faithfulness of NaiveRAG

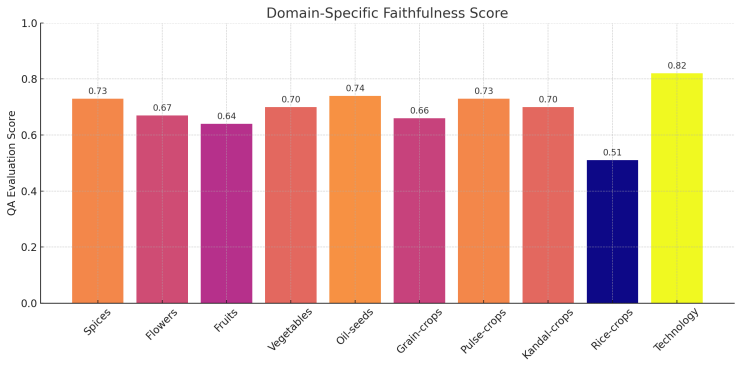



Figure 6.6: Faithfulness Comparison by Category of NaiveRAG

### Domain Specific QA Evaluation

The best performance in the Domain QA evaluation, which assesses a chatbot's ability to provide accurate answers to queries related to domain-specific information, was observed in Oil-seeds (0.92), Fruits (0.89), Flowers (0.88), and Technology (0.87). The QA capabilities of Grain-Crops (0.78), Spice (0.75), Kandal-Crops (0.86), and Vegetables (0.79) were deemed to have good performance. Pulse-Crops (0.72) and Rice (0.64) proved to have the lowest QA scores, indicating that the system finds it more challenging in this dataset to produce precise and relevant answers in the latter case. The results indicate a generally strong quality in the chatbot's responses in oil-bearing and high-value crop domains, but a significant room for improvement in the quality of responses for rice. Table 6.10 & 6.7 represents the score and comparison of categories

| Category | Score |
|---|---|
| Spices | 0.75 |
| Flowers | 0.88 |
| Fruits | 0.89 |
| Vegetables | 0.79 |
| Oil-seeds | 0.92 |
| Grain-crops | 0.78 |
| Pulse-crops | 0.72 |
| Kandal-crops | 0.86 |
| Rice-crops | 0.64 |
| Technology | 0.87 |

Table 6.10: Domain Specific QA Evaluation of NaiveRAG

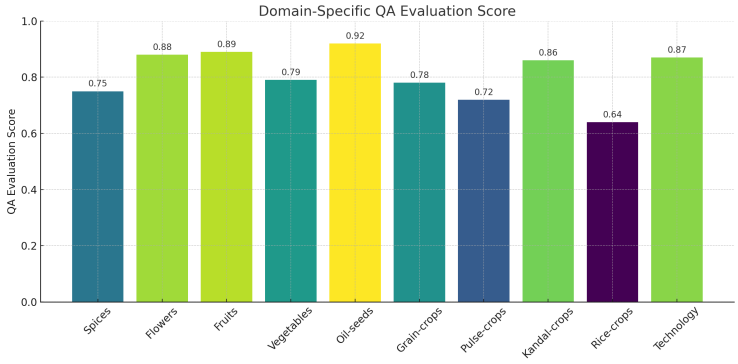



Figure 6.7: QA Evaluation Comparison by Category of NaiveRAG

### Domain Specific Semantic Similarity

Regarding semantic similarity – how closely the chatbot's responses semantically align with Ground truth – the model developed the best semantic coherence in Flowers (0.8664). At the same time, Spice (0.84), then Grain and Crops (0.84), and finally Kandal and Crops (0.84) demonstrated the highest similarity. The lowest coefficients were observed for Fruits (0.83), Technology (0.82), and Oilseeds (0.81), while Vegetables (0.8‘) and Pulse Crops (0.74) exhibited a medium coefficient. The lowest score was again for rice (0.74), meaning that the semantic drift or alignment in these domains is weak. Overall, the results show that for almost all domains, the model preserves its semantic accuracy, but fails particularly in the two domains, Pulse—Crops and Rice, possibly because there is not enough or unclear training data for those domains. Table 6.11 represents the summarized Semantic Similarity performance & figure 6.8 illustrates Semantic Similarity for each domain.

| Category | Score |
|---|---|
| Spices | 0.84 |
| Flowers | 0.87 |
| Fruits | 0.83 |
| Vegetables | 0.81 |
| Oil-seeds | 0.81 |
| Grain-crops | 0.84 |
| Pulse-crops | 0.74 |
| Kandal-crops | 0.84 |
| Rice-crops | 0.74 |
| Technology | 0.82 |

Table 6.11: Domain Specific Semantic Similarity of NaiveRAG

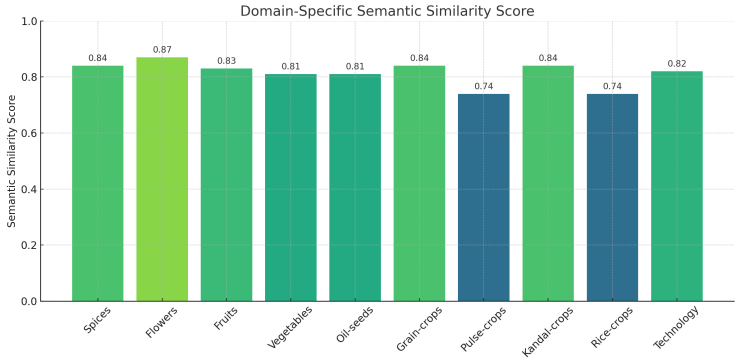



Figure 6.8: Semantic Similarity Comparison by Category of NaiveRAG

### 6.2.3 GraphRAG Performance

#### Context Precision, Recall & F1 Score

We conducted an extensive system evaluation using standard retrieval and generation measures to assess the efficacy of the GraphRAG technique in the agricultural domain. In particular, Contextual Node Precision, Contextual Node Recall, and F1 Score were used to evaluate the accuracy and relevance of the nodes returned, utilizing a combination of entity-aware Cypher traversal and fallback full-text search retrieval strategies, both precision-oriented and recall-oriented, for different degrees of ambiguity and contextual queries. Results from empirical tests showed that the system achieved a **Contextual Node Precision of 80.2%**, indicating that over three-quarters of the retrieved nodes were contextually appropriate and relevant to queries from different users. Moreover, it **achieved 89.52% in Contextual Node Recall**, indicating a strong retrieval capability of the graph structure in recovering relevant knowledge. Such a well-balanced trade-off between precision and recall, as reflected in the **F1 Score of 81.83%**, highlights the robustness and effectiveness of the retrieval mechanism. Given the linguistic complexity of Bangla, including orthographic inconsistencies and morphological richness, the results hold particular significance. These findings demonstrate the strength of the GraphRAG pipeline in this setting to scale to and adapt to the linguistic variations in other underresourced languages in question answering.

### Faithfulness

To quantify the factual consistency of the generated answers and the knowledge graph underlying them, we computed the faithfulness of the GraphRAG model across all ten agricultural categories, obtaining an overall score of **87.47%**. This measure corresponds to the extent to which responses provided by the model draw on verifiable domain knowledge and reduce hallucination and misinformation. Factural fidelity is crucial in applications such as agriculture, where incorrect information can cause real-world damage or result in the misuse of resources.

| Metrics | Result |
|---|---|
| Context Precision | 80.20% |
| Context Recall | 89.52% |
| F1 Score | 81.83% |
| Faithfulness | 84.47% |
| QA Evaluation | 91.58% |
| Semantic Similarity | 84.45% |

Table 6.12: Overall Performance of GraphRAG

### QA Evaluation & Semantic Similarity

To assess the effectiveness of the GraphRAG model in addressing agricultural queries in low-resource Bangla, we performed a thorough evaluation on a dataset comprising 500 question-answer pairs covering 10 agriculture domains. **The results indicated that the model performed well in the question-answering task, achieving an overall accuracy of 91.58%**. This means that the model can provide correct and contextually relevant responses across a broad range of topics. Furthermore, the system achieved a **semantic similarity score of 84.85%**, indicating that the generated responses align semantically very well with the ground truth answers in terms of meaning and conceptual coherence. This demonstrates the model's ability to remain factual and linguistically meaningful in various agricultural settings. The structured knowledge graph, powered by Neo4j, served as a crucial enabler of contextual understanding and retrieval precision, while the GPT-4.1-mini language model generated coherent, fluent text. These components then enabled GraphRAG to provide high-quality, explainable, and semantically rich answers to question-answering tasks in low-resource settings, such as Bangla agriculture.

### Domain Specific Context Precision, Recall, F1 Score

An evaluation of KrishiBot's performance across different agricultural domains demonstrated notable domain-specific variations in precision, recall, and F1 scores, highlighting the system's dynamic nature as well as its strengths and weaknesses. The precision scores across domains showed that Oilseeds achieved the highest precision score of 0.90, followed by Pulse Crops (0.89), Flowers (0.86), and Kandal Crops (0.81). The values for the Fruits and Grain-Crops categories were also comparable, namely 0.85 and 0.85, respectively. In contrast, Vegetables scored substantially lower, at 0.78, and an even lower value of 0.76 was recorded for Spices. Meanwhile, Rice Crop and Technology had the lowest precision scores, at 0.76 and 0.39, respectively. Although most crop-based domains were exact, queries in the Technology domain often resulted in off-target responses.

Pulse-Crops yielded the highest value of 0.97 for recall, indicating that the model was highly capable of retrieving relevant information. Moreover, domains with high performance included Spice (0.96), Fruits (0.93), Oilseeds (0.91), Vegetables (0.91), and Kandal-Crops (0.89). With 0.87 and 0.85, Flowers and Grain-Crops followed. Despite the low precision of the technology, its recall was higher at 0.87, suggesting that the technology had a tendency to retrieve a broad set of content but lacked specificity. The Rice-Crop scored 0.80, indicating that it was underperforming in the task of retrieving all relevant information.

The F1 scores, which balance precision and recall, showed that Pulse-Crops had the highest F1 value, 0.91, while Oilseeds had the second-best at 0.90, indicating strong and balanced response performances. On the other hand, Flowers and Fruits had 0.86 and 0.86. The performance of Grain-Crops (0.83), Vegetables (0.82), and Kandal-Crops (0.83) domains was also moderate. The F1 Score of Spice was slightly lower at 0.79. The Rice Crop category scored 0.77 while the Technology domain had the lowest score, 0.46, indicating significant shortcomings in output relevance and relevance. Overall, substantial gaps between retrieval and response accuracy remained in many domains (e.g., Technology), indicating the need for further performance optimizations to achieve a desired level of relevance consistency. Table 7.1 represents each domain score obtained.

| Category | Precision | Recall | F1 Score |
|---|---|---|---|
| Spices | 0.76 | 0.96 | 0.79 |
| Flowers | 0.86 | 0.87 | 0.86 |
| Fruits | 0.85 | 0.93 | 0.86 |
| Vegetables | 0.78 | 0.91 | 0.82 |
| Oil-seeds | 0.90 | 0.92 | 0.90 |
| Grain-crops | 0.85 | 0.85 | 0.83 |
| Pulse-crops | 0.89 | 0.97 | 0.91 |
| Kandal-crops | 0.81 | 0.89 | 0.83 |
| Paddy-crops | 0.76 | 0.80 | 0.77 |
| Technology | 0.39 | 0.87 | 0.46 |

Table 6.13: Domain Specific Context Recall, Precision, and F1 Score for GraphRAG

### Domain Specific Faithfulness

To determine the factual integrity of the answers generated by the GraphRAG model, a domain-specific faithfulness analysis was conducted on ten agriculturally related categories, yielding an average of 87.47%. Pulse-Crops had the highest faithfulness, with a value of 0.95, and Oilseeds had a value of 0.89, indicating that the model was well-grounded in the knowledge-rich and well-structured domains. Vegetables and fruits also demonstrated strong factual consistency, with both scoring 0.88, respectively. The other mid-performing domains are Spice (0.88), Kandal-Crops (0.87), and Flowers (0.85), which indicate the model's consistency in aligning with domain knowledge, regardless of lexical or semantic complexity. In the more challenging categories, Grain-crops (0.85), Technology (0.85), and Paddy-crops (0.85), the lower scores by a small margin indicate a possible deficiency in graph

representation or less regular linguistic behavior in the user queries. Across the board, the faithfulness values are sufficiently high, which supports the idea that incorporating a knowledge graph powered by Neo4j actually constrains the generative process, resulting in factual and verifiable responses across a wide range of agricultural concepts. Table 6.14 represents the summarized Faithfulness performance & figure 6.9 illustrates Faithfulness for each domain.

| Category | Result |
|---|---|
| Spices | 0.88 |
| 5 Flowers | 0.85 |
| 7 Fruits | 0.88 |
| 4 Vegetables | 0.88 |
| 3 Oil-seeds | 0.89 |
| 2 Grain-crops | 0.85 |
| 8 Pulse-crops | 0.95 |
| 1 Kandal-crops | 0.87 |
| 6 Paddy-crops | 0.85 |
| Technology | 0.85 |

Table 6.14: Domain Specific Faithfulness

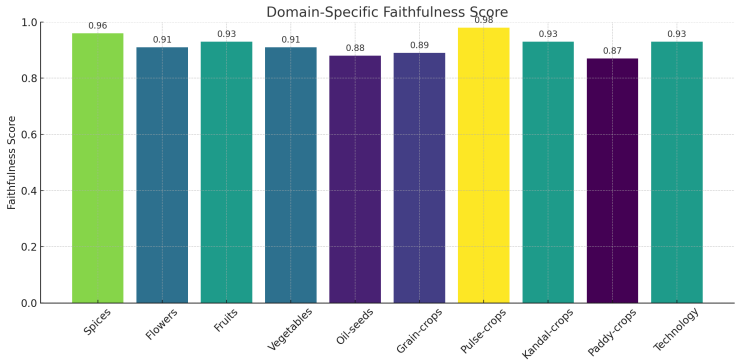



Figure 6.9: Faithfulness Comparison by Category of GraphRAG

### Domain Specific QA Evaluation

The results for KrishiBot's performance in domain QA accuracy and domain semantic similarity provide further evidence of the system's strengths and limitations across categories. It is worth mentioning, the QA score and trends of the domains that did not possess at least one keyword for query selection were omitted; the highest QA score was seen in the Pulse-Crops domain (0.98), followed by Technology (0.93), Fruits (0.93), Kandal-Crops (0.93) and Vegetables (0.91). The Grain crops scored 0.89; Flowers 0.90; Spice 0.96, Oilseeds 0.88, and Rice-crops was 0.87. These results suggest that the model is highly capable of producing correct answers for most agricultural domains, especially for pulses and horticulture subjects. Table 6.15 & 6.10 represents the score and comparison of categories.

| Category | Score |
|---|---|
| Spices | 0.96 |
| Flowers | 0.91 |
| Fruits | 0.93 |
| Vegetables | 0.91 |
| Oil-seeds | 0.88 |
| Grain-crops | 0.89 |
| Pulse-crops | 0.98 |
| Kandal-crops | 0.93 |
| Paddy-crops | 0.87 |
| Technology | 0.93 |

Table 6.15: Domain Specific QA Evaluation of GraphRAG

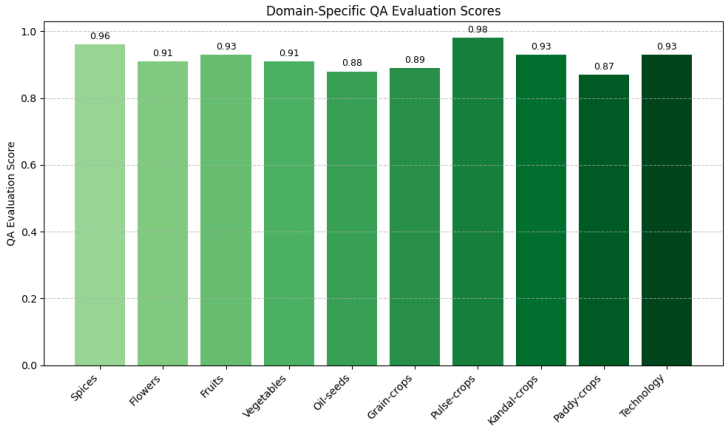



Figure 6.10: QA Evaluation Comparison by Category of GraphRAG

### Domain Specific Semantic Similarity Evaluation

In terms of domain semantic similarity, reflected through the alignment between the generated responses and the question intent within the linguistic domain, Flowers (0.92) achieved the highest score, followed by Spice (0.87), Fruits (0.87), Grain-crops (0.89), Kandal-crops (0.88) and Vegetables (0.85). Oilseeds (0.86) and Technology (0.83) were also moderate performers. Although Pulse Crops exhibited high QA accuracy (0.80), their relatively low semantic similarity score (0.79) indicates room for improvement in natural language generation, even when factual content is accurate. Rice-Crops had the lowest semantic similarity score of 0.79, suggesting more training data or enhanced knowledge representation are needed in this area. All in all, KrishiBot provides semantically relevant and factually accurate responses across nearly all agricultural domains, although it has clear strengths in the Flowers, Fruits, and Pulse-Crops domains. Targeted optimization is needed to address the semantic and retrieval gaps in the Dhan and Technology categories. Table 6.16 represents the summarized Semantic Similarity performance & figure **??** illustrates Semantic Similarity for each domain.

| Category | Result |
|---|---|
| Spices | 0.87 |
| Flowers | 0.92 |
| Fruits | 0.87 |
| Vegetables | 0.85 |
| Oil-seeds | 0.86 |
| Grain-crops | 0.89 |
| Pulse-crops | 0.79 |
| Kandal-crops | 0.88 |
| Rice-crops | 0.79 |
| Technology | 0.83 |

Table 6.16: Domain Specific Semantic Similarity of GraphRAG

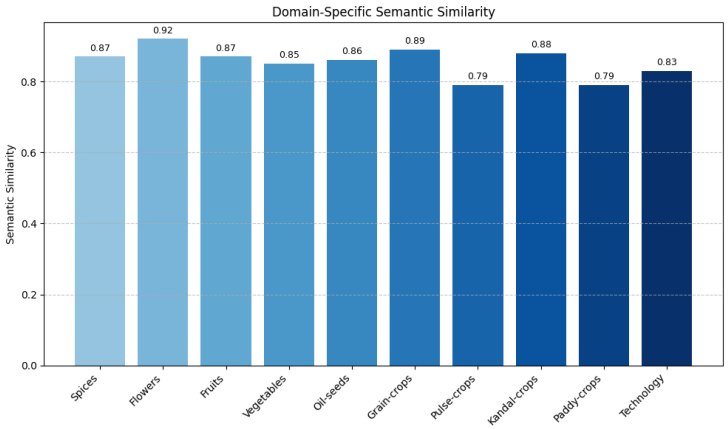



Figure 6.11: Semantic Similarity Comparison by Category

## 6.3 LLM Cost Analysis

After evaluating several state-of-the-art language models across cost and performance parameters, GPT-4.1 Mini was selected for our agricultural chatbot system due to its optimal balance of computational efficiency and linguistic capability. Among the compared models—Claude Haiku 3.5, LLaMA 3.3 70B, GPT-4o, and GPT-4.1 Mini—GPT-4.1 Mini demonstrated the most cost-effective performance. It offers a highly competitive rate of $0.40 per 1 million input tokens and $1.60 per 1 million output tokens, with cached input processing at just $0.10 per million tokens, making it significantly more affordable than Claude Haiku 3.5 ($0.80 input / $4 output) and GPT-4o ($2.50 input / $10 output). While LLaMA 3.3 70B is moderately priced, GPT-4.1 Mini surpasses it in reasoning quality, reliability, and prompt-following behavior based on empirical benchmarking. This makes GPT-4.1 Mini the ideal choice for a cost-sensitive deployment that demands both scalability and high response accuracy in real-world agricultural advisory systems.

| Model | Input/MT | Output/MT | Cost Remarks |
|---|---|---|---|
| GPT4.1 Mini | $0.40 | $1.60 | Cost Efficient, Well Performed |
| GPT4o | $2.50 | $10.0 | Expensive |
| GPT3.5 Turbo | $0.50 | $1.50 | Cost Efficient, Lower Reasoning |
| Claude3.5 Haiku | $0.80 | $4.00 | Expensive |
| Llama 3.3 70B | $0.59 | $0.79 | Efficient, Lower Reasoning |

Table 6.17: LLM Cost Analysis, MT- Million Tokens

# Chapter 7

# Discussion

We comparatively evaluated NaiveRAG and GraphRAG on a standard set of 500 Bangla agricultural queries using the optimal retrieval-augmented generation approach for KrishiBot. Although the NaiveRAG showed a slightly better average response time of 7.36 seconds compared to 8.06 seconds in GraphRAG, the difference in time is a mere 0.70 seconds, representing an increase of only 9.5 percent. Nonetheless, this minor timing overhead is significantly mitigated by the fact that GraphRAG achieves considerable performance improvements across all evaluation criteria. Regarding the QA evaluation of question-answering, GraphRAG achieved 91.58%, which is 12.98% higher than NaiveRAG's 78.6%. This is a relative enhancement of 16.5%. Equally, GraphRAG produced greater semantic similarity (84.85% vs. 80.66%) and a significantly better faithfulness score (87.47% vs. 66.37%), which corresponds to a factual consistency improvement of 31.8% in the agricultural context, where misinformation can lead to dire real-world consequences.

| Metrics | NaiveRAG | GraphRAG |
|---|---|---|
| Context Precision | 74.56% | 80.20% (+5.64%) |
| Context Recall | 74.42% | 89.52% (+15.10%) |
| F1 Score | 70.02% | 81.83% (+11.81%) |
| Faithfulness | 66.37% | 84.47% (+18.10%) |
| QA Evaluation | 78.60% | 91.58% (+12.98%) |
| Semantic Similarity | 80.66% | 84.45% (+3.79%) |

Table 7.1: Overall Performance of NaiveRAG & GraphRAG

In terms of retrieval performance, the Contextual Node Precision and Recall of GraphRAG were 80.2% and 89.52%, respectively, compared to the chunk-level precision of 74.56% and a recall of 74.42% in NaiveRAG. This led to a 17 percent increase in the F1 score, achieving a balance between the relevance and completeness of the retrieved information, from 70.02 percent to 81.83 percent. Whereas GraphRAG incurs a fixed initial cost (for constructing the Neo4j knowledge graph), this trade-off enables structured, explainable, and semantically rich retrieval—a property that NaiveRAG lacks.

Finally, although the decrease in response time is not significant, the gains in accuracy, semantic coherence, factual alignment, and retrieval quality are greater with GraphRAG. As such, we decided to work with GraphRAG because it offers a much more solid, trustworthy, and scalable solution in the low-resource and high-stakes scenario of question answering.

# Chapter 8

# UI Implementation

## 8.1 Developing The System

The user interface of our GraphRAG-powered agricultural assistant is built with a strong focus on accessibility, performance, and ease of interaction, particularly tailored for Bengali-speaking users in low-resource environments.

### 8.1.1 Frontend Stack and UI Implementation

The frontend is developed using React with TypeScript, providing a type-safe and scalable codebase. The build and development environment leverages Vite, known for its fast hot module replacement and minimal build times, which significantly enhances the developer experience. Styling is handled via Tailwind CSS with a custom configuration, enabling rapid prototyping through utility-first classes while maintaining design coherence. Key UI features include:

- Dynamic theme support (light and dark mode) implemented through Tailwind's custom color palette and CSS variables.
- Modern visual effects such as backdrop blur and animated gradient backgrounds optimized with hardware-accelerated CSS.
- **Voice input support** through the Web Speech API, enabling hands-free query submission in Bengali.
- Controlled chat components typed with TypeScript interfaces to ensure consistent interaction and rendering of messages.

User interactions are managed with React hooks (useState, useRef), and state changes are optimized for real-time responsiveness.

Figure 8.1 represents our prototype user interface.

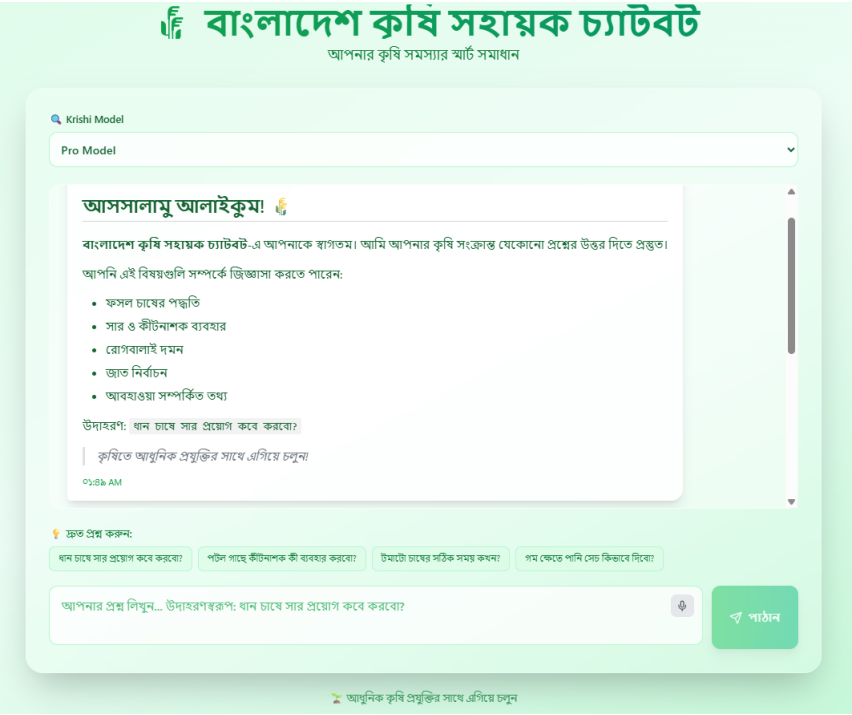



Figure 8.1: User Interface

Figure 8.2 represents our prototype with example query.

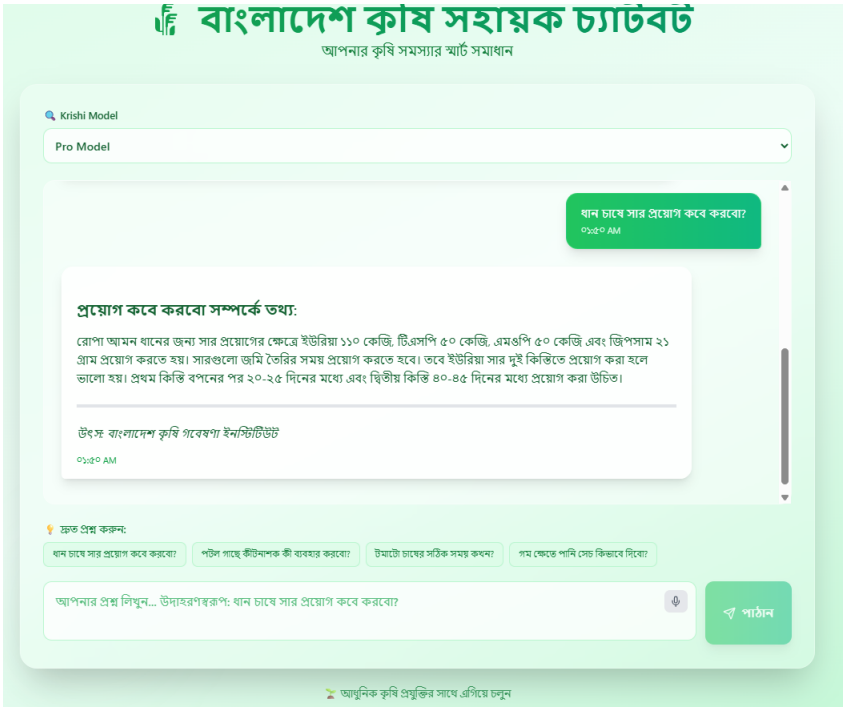



Figure 8.2: User Interface with Example Query

### 8.1.2 Backend Integration with GraphRAG

The frontend communicates with a FastAPI-based backend that serves as the orchestrator of the GraphRAG pipeline. Upon receiving a user query, whether typed or spoken. The system forwards it via RESTful API endpoints to the backend, which handles:

- Semantic query formatting using a prompt-optimized LLM.
- Modern visual effects such as backdrop blur and animated gradient backgrounds optimized with hardware-accelerated CSS.
- Graph-based retrieval from a Neo4j database using Cypher queries.
- Response generation using OpenAI's language models grounded in the retrieved knowledge context.

Responses are then returned to the frontend and rendered with support for Markdown formatting, allowing structured replies that include bullet points, highlights, and technical details in an easy-to-read manner. This integration between the frontend and GraphRAG backend allows for an interactive, real-time conversational experience that is both context-aware and linguistically inclusive, making agricultural guidance accessible to a wide user base in Bangladesh.

# Chapter 9

# Work Plan

Our thesis has three major stages: Pre-Thesis 1 (P1), Pre-Thesis 2 (P2), and the final defence. Our Pre-Thesis 1 started in Summer 2024 and is completed in October 2024. During this phase, we formed our thesis team, elected our supervisor and our topic, registered our thesis, drafted an abstract, read papers from relevant and reliable publications, compiled summaries of these papers into a literature review to get a better understanding, and finally submitted a report. Pre-Thesis 2 (P2) started from November 2024 and will be completed in January 2025. In this stage, we gathered research data such as validation dataset and the books. validated the data, and implemented four models that performed best for this research. In addition, we composed the work report for this stage and designed a poster for the presentation. Finally, in the defence stage in Spring 2024, we developed GraphRAG, we deployed our best performed model on cloud and developed an UI to be used to solve real-time problems. We evaluated and enhance the outcomes we have obtained, and last but not least, compare it to the related work to try to show the thoroughness and precision of our work. The entire working process of our thesis is represented in a Gantt Chart in figure 9.1, and the three phases are coloured to make them more recognizable.

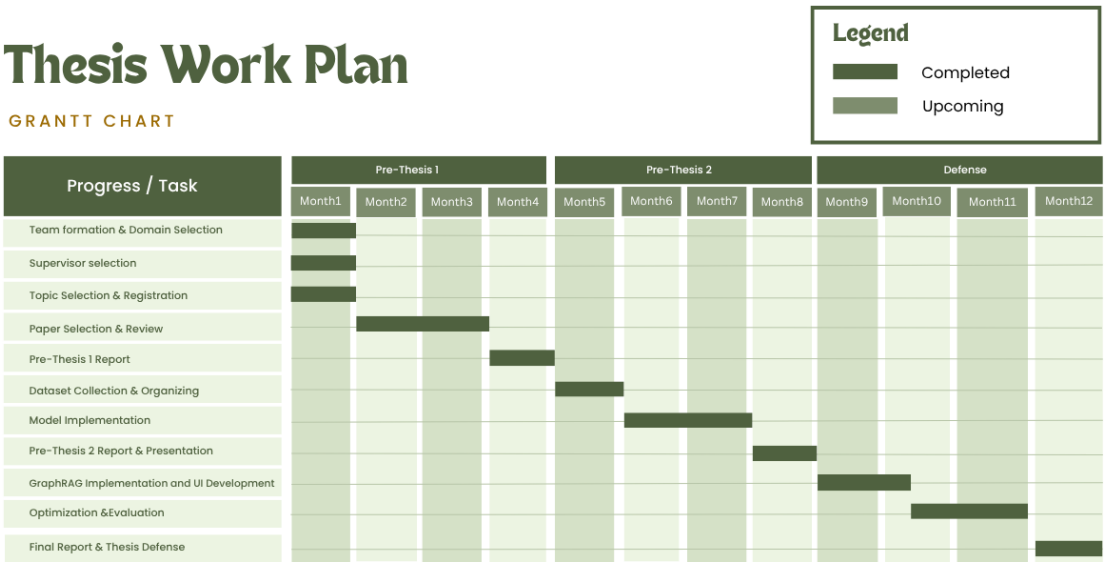



Figure 9.1: Thesis Workplan Gantt Chart

# Chapter 10

# Future Work, Limitation & Conclusion

## 10.1 Limitation

While KrishiBot demonstrates strong potential in supporting Bangladeshi farmers through digital assistance, its current implementation is not without constraints that limit its broader applicability and impact. Despite its current utility, the existing text-based version of KrishiBot faces several limitations. These include an over-reliance on standard Bangla, which may not be comprehensible to users from dialect-rich regions; the necessity of a stable internet connection; a lack of support for multimodal inputs such as images or native dialect; and computational demands that may hinder deployment on low-cost devices, thus affecting the model's robustness and generalizability. Additionally, KrishiBot currently lacks the capability to dynamically update its dataset, meaning that newly emerging agricultural knowledge, seasonal data, or policy changes cannot be reflected in real time. Recognizing and addressing these limitations is a critical step toward evolving KrishiBot into a more inclusive, accessible, and scalable solution tailored to the diverse needs of Bangladesh's agricultural communities.

## 10.2 Future Work

To enhance the future development of KrishiBot, building on the fact that 89.9% of 15 million farming households (around 13.5 million farmers) in Bangladesh own basic or smartphone devices, as per the 2022 survey by the Bangladesh Bureau of Statistics [5], we will prioritize accessibility, inclusivity and usability in digital services to serve better the agriculture community.
Future work will focus on the following directions to make use of KrishiBot to the maximum impact:

- Development of a 2G voice-based interface to improve accessibility for marginal and low-literacy farmers through intuitive voice interaction.
- Integration of image recognition capability in the system to allow farmers to assess and diagnose crop-related issues.

- Support for various regional Bangla dialects in order to have inclusivity.
- Refine the dataset with localized, real-time agricultural data to make predictions in the system more accurate and contextually relevant.

In addition, we aim to collaborate with government agricultural bodies to align our efforts with national initiatives, extend KrishiBot's reach, and ensure long-term sustainability and scalability. These enhancements will position KrishiBot as a powerful tool for empowering farmers with timely, accessible, and locally relevant agricultural support.

## 10.3 Conclusion

This research aims to build an NLP-based chatbot to empower Bangladeshi farmers by providing accurate agricultural information in standard Bangla on time. We explored deep learning models, especially Large Language Models like GPT 4.1Mini and LLaMA, to train and further fine-tune the model for the agricultural domain. The literature review section involved a detailed evaluation of the various NLP and LLM techniques we interfaced to design our framework, serving as the premise for our study. Thus, to help the readers follow our methodological approach, we provided background information and significant research findings from state-of-the-art papers. A Gantt chart showing the research activities was incorporated to ensure efficient planning. In this phase of our work, we extended those levels of conversational competence using NaiveRAG, GraphRAG and Langchain to look for improved accuracy and cohesiveness of the answers provided by the chatbot. Our study should profoundly respond to the agricultural community by presenting matters of importance and utilization of the resources merely in standard Bangla, which will help the farmers increase their productivity and aptitude for enhanced farm management.

# Bibliography


[1] A. Saha, M. I. Noor, S. Fahim, S. Sarker, F. Badal, and S. Das, "An approach to extractive bangla question answering based on bert-bangla and bquad," in *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, Rajshahi, Bangladesh, 2021, pp. 1–6. DOI: 10.1109/ACMI53878.2021.9528178.

[2] A. Bhattacharjee, T. Hasan, W. U. Ahmad, and R. Shahriyar, "Banglanlg and banglat5: Benchmarks and resources for evaluating low-resource natural language generation in bangla," in *Proceedings of EACL 2023*, 2023, pp. 1–15. DOI: 10.18653/v1/2023.findings-eacl.54.

[3] J. J. Bird, A. Ekárt, and D. R. Faria, "Chatbot interaction with artificial intelligence: Human data augmentation with t5 and language transformer ensemble for text classification," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 4, pp. 3129–3144, 2023.

[4] N. Esfandiari, K. Kiani, and R. Rastgoo, "A conditional generative chatbot using transformer model," *arXiv preprint arXiv:2306.02074*, 2023.

[5] R. Karim Byron, "Labour force survey 2022: Agriculture still main job generator," *The Daily Star*, Mar. 2023. [Online]. Available: https://www.thedailystar.net/news/bangladesh/news/labour-force-survey-2022-agriculture-still-main-job-generator-3283936.

[6] N. F. Liu, T. Zhang, and P. Liang, *Evaluating verifiability in generative search engines*, 2023. arXiv: 2304.09848 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2304.09848.

[7] S. Rezayi, Z. Liu, Z. Wu, *et al.*, *Exploring new frontiers in agricultural nlp: Investigating the potential of large language models for food applications*, 2023. arXiv: 2306.11892 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2306.11892.

[8] J. F. Ruma, T. T. Mayeesha, and R. M. Rahman, "Transformer based answer-aware bengali question generation," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 314–326, 2023. DOI: 10.1016/j.ijcce.2023.09.003.

[9] H. Touvron, L. Martin, K. Stone, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023. DOI: 10.48550/arxiv.2307.09288.

[10] A. Wang, L. Song, G. Xu, and J. Su, "Domain adaptation for conversational query production with the RAG model feedback," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 9129–9141. DOI: 10.18653/v1/2023.findings-emnlp.612. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.612.

[11] S. Yadav and A. Kaushik, "Comparative study of pre-trained language models for text classification in smart agriculture domain," in *Proceedings of the International Conference on Emerging Trends in Computing*, 2023, pp. 1–10.

[12] M. Ahmed, H. U. Khan, and E. U. Munir, "Conversational ai: An explication of few-shot learning problem in transformers-based chatbot systems," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 2, pp. 1888–1901, 2024. DOI: 10.1109/TCSS.2023.3281492.

[13] A. Balaguer, V. Benara, R. Cunha, *et al.*, "Study on agriculture: A pipeline for fine-tuning and retrieval-augmented generation of large language models," *arXiv preprint arXiv:2401.08406*, 2024. DOI: 10.48550/arXiv.2401.08406.

[14] H. Chia, A. I. Oliveira, and P. Azevedo, "Implementation of an intelligent virtual assistant based on llm models for irrigation optimization," in *2024 8th International Young Engineers Forum on Electrical and Computer Engineering (YEF-ECE)*, Caparica / Lisbon, Portugal, 2024, pp. 94–100. DOI: 10.1109/YEF-ECE62614.2024.10624819.

[15] J. C. L. Chow, V. Wong, and K. Li, "Generative pre-trained transformer-empowered healthcare conversations: Current trends, challenges, and future directions in large language model-enabled medical chatbots," *BioMedInformatics*, vol. 4, no. 1, pp. 837–852, 2024. DOI: 10.3390/biomedinformatics4010047.

[16] D. Dharrao and S. Gite, "Therapybot: A chatbot for mental well-being using transformers," *International Journal of Advances in Applied Sciences*, vol. 13, no. 1, pp. 1–12, 2024.

[17] K. Didwania, P. Seth, A. Kasliwal, and A. Agarwal, *Agrillm: Harnessing transformers for farmer queries*, 2024. arXiv: 2407.04721 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2407.04721.

[18] J. Li, M. Xu, L. Xiang, *et al.*, "Large language models and foundation models in smart agriculture: Basics, opportunities, and challenges," *arXiv preprint arXiv:2308.06668*, 2024. DOI: 10.48550/arXiv.2308.06668.

[19] V. Nguyen, S. Karimi, W. Hallgren, A. Harkin, and M. Prakash, "My climate advisor: An application of nlp in climate adaptation for agriculture," in *Proceedings of the 1st Workshop on Natural Language Processing Meets Climate Change (ClimateNLP 2024)*, Association for Computational Linguistics, 2024, pp. 27–45.

[20] K. Pearce, S. Alghowinem, and C. Breazeal, "Build-a-bot: Teaching conversational ai using a transformer-based intent recognition and question answering architecture," 2024.

[21] R. Pratap Deb Nath, T. Rani Das, T. Chandro Das, and S. M. Shafkat Raihan, "Knowledge graph generation and enabling multidimensional analytics on bangladesh agricultural data," *IEEE Access*, vol. 12, pp. 87 512–87 531, 2024. DOI: 10.1109/ACCESS.2024.3416388.

[22] R. Sapkota, R. Qureshi, S. Z. Hassan, *et al.*, "Multi-modal llms in agriculture: A comprehensive review," *Authorea Preprints*, 2024.

[23] J. Wu, F. Orlandi, D. O'Sullivan, and S. Dev, "Leveraging knowledge graphs for enhancing climate-resilient agricultural analysis," in *2024 IEEE 9th International Conference on Computational Intelligence and Applications (ICCIA)*, 2024, pp. 241–245. DOI: 10.1109/ICCIA62557.2024.10719340.

[24] S. Yang, Z. Liu, and W. Mayer, "Shizishangpt: An agricultural large language model integrating tools and resources," *arXiv preprint arXiv:2409.13537*, 2024.

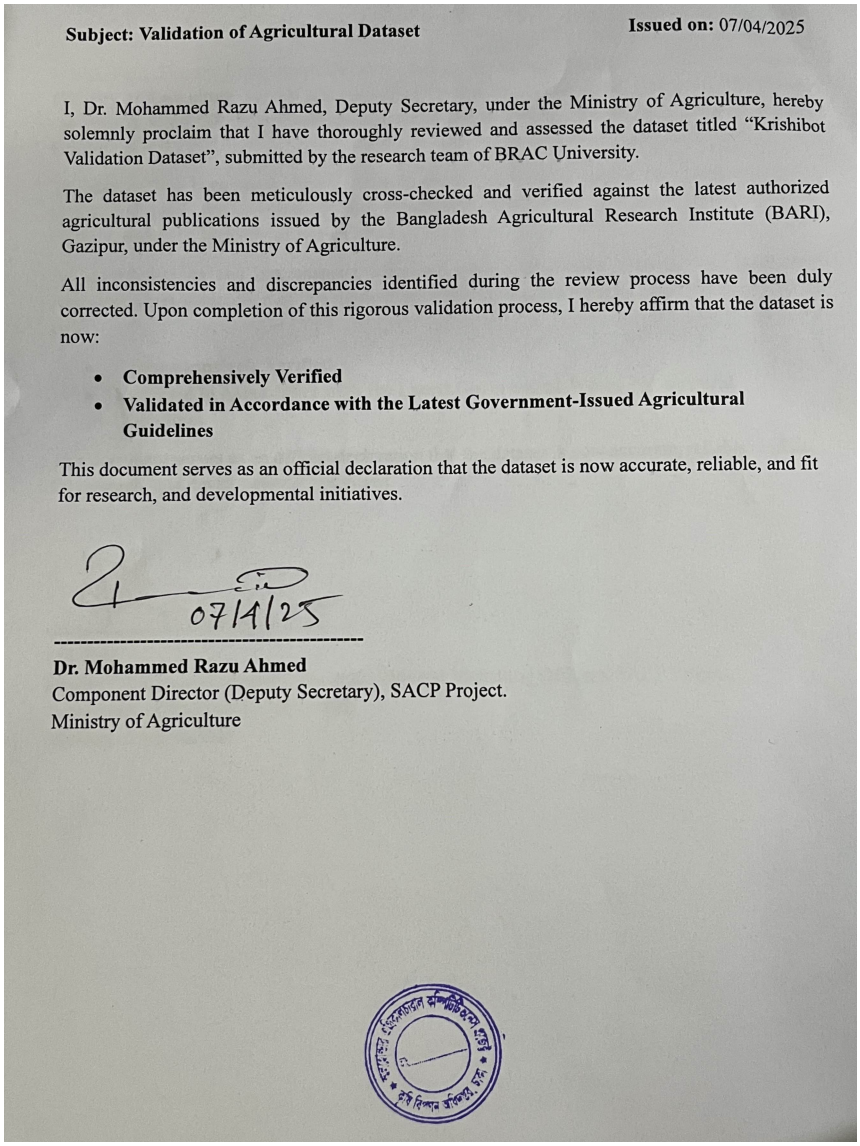
[25] P. Krataithong, W. Buranasing, M. Buranarach, T. Wutthitasarn, P. Meeklai, and P. Tumsangthong, "Tourism chatbot framework: Enhancing visitor experience through graphrag and ai chatbot," in *2025 IEEE International Conference on Cybernetics and Innovations (ICCI)*, 2025, pp. 1–6. DOI: 10.1109/ICCI64209.2025.10987390.

[26] Q. Zhang, S. Chen, Y. Bei, *et al.*, *A survey of graph retrieval-augmented generation for customized large language models*, 2025. arXiv: 2501.13958 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2501.13958.

# Appendix A

# Test Dataset Approval Letter

## A.1 Test Dataset Approval Letter 01



**Subject: Validation of Agricultural Dataset** **Issued on:** 07/04/2025

I, Dr. Mohammed Razu Ahmed, Deputy Secretary, under the Ministry of Agriculture, hereby solemnly proclaim that I have thoroughly reviewed and assessed the dataset titled "Krishibot Validation Dataset", submitted by the research team of BRAC University.

The dataset has been meticulously cross-checked and verified against the latest authorized agricultural publications issued by the Bangladesh Agricultural Research Institute (BARI), Gazipur, under the Ministry of Agriculture.

All inconsistencies and discrepancies identified during the review process have been duly corrected. Upon completion of this rigorous validation process, I hereby affirm that the dataset is now:

- **Comprehensively Verified**
- **Validated in Accordance with the Latest Government-Issued Agricultural Guidelines**

This document serves as an official declaration that the dataset is now accurate, reliable, and fit for research, and developmental initiatives.

07/4/25

---

**Dr. Mohammed Razu Ahmed**
Component Director (Deputy Secretary), SACP Project.
Ministry of Agriculture

## A.2 Test Dataset Approval Letter 02

**Subject: Validation of Agricultural Dataset** **Issued on:** 07/04/2025

I, Md. Faiz, Deputy Component Director (Senior Agricultural Marketing Officer), under the Ministry of Agriculture, hereby solemnly proclaim that I have thoroughly reviewed and assessed the dataset titled "Krishibot Validation Dataset", submitted by the research team of BRAC University.

The dataset has been meticulously cross-checked and verified against the latest authorized agricultural publications issued by the Bangladesh Agricultural Research Institute (BARI), Gazipur, under the Ministry of Agriculture.

All inconsistencies and discrepancies identified during the review process have been duly corrected. Upon completion of this rigorous validation process, I hereby affirm that the dataset is now:

- **Comprehensively Verified**
- **Validated in Accordance with the Latest Government-Issued Agricultural Guidelines**

This document serves as an official declaration that the dataset is now accurate, reliable, and fit for research, and developmental initiatives.

07/04/2025

Md. Faiz
Deputy Component Director (Senior Agricultural Marketing Officer), SACP Project.
Department of Agricultural Marketing
Ministry of Agriculture