Agricultural Chatbot: Improving Context-Specific Query Resolution with LLMs, RASA, and RAG Systems

Madhav Mishra[1][0000-0002-1249-0372], Shyama Wilson[2], Neeraj Goel[3][0000-0003-3824-9437], and Athula Ginige[4][0000-0002-7445-588X]

[1] Indian Institute of Technology, Ropar, India
[2] Western Sydney University, Australia

Abstract. Natural Language Understanding (NLU) capabilities of Large Language Models (LLMs) have improved with recent developments. These models have excellent general knowledge after being trained on large and varied datasets covering a variety of domains. They can answer a wide range of questions with its huge knowledge. However, because of their limited contextual grounding in domain-specific details, such as in Agriculture where farmers need precise, reliable and actionable advices related to crop diseases, pest control or fertilization techniques etc. LLMs often fail to provide accurate and trustworthy answers. To overcome this problem, the authors introduced an agriculture chatbot system that helps farmers in getting accurate, realtime advice related to agriculture practices. Our system integrates three essential technologies: Retrieval-Augmented Generation (RAG) for precise information retrieval, Large Language Models (LLMs) for natural language response generation, and RASA for intent recognition and dialogue management, is designed, assembled, and evaluated in this paper. By giving farmers an interactive assistant that can respond to questions regarding cultivation, the system meets the urgent need for easily accessible agricultural knowledge. Advanced features of the chatbot involve understanding complex agricultural queries, requesting clarification when information is lacking, obtaining necessary information from a knowledge base, and producing natural language responses. We evaluated our system on some different agricultural situations and showed it to be effective in enhancing response accuracy, user interaction, and contextual understanding. The findings highlight the promise of the integration of modern NLP technologies to facilitate informed decision-making in agriculture, helping farmers in growing more with easy-to-use and smart digital support.

Keywords: RASA. RAG. LLM.

1 Introduction

Agriculture is one of the most important industries in the world, ensuring food security and means of living for billions of individuals [17]. However, today the farmers face more challenges than ever before like climate, unpredictable weather, soil degradation, pest and disease outbreaks, resource scarcity, and a growing knowledge gap among the farmers [1]. All these challenges are threatening the crop yields and food security. To overcome these challenges, farmers need access to timely, accurate and context-relevant agricultural information for a better crop management [6]. As digital technologies are extending into rural areas, there is a growing opportunity to utilize the artificial intelligence for bridging the agricultural information gap.

Farming communities are diverse and geographically dispersed, it is difficult to reach all farmers, particularly those in geographically isolated areas, using traditional methods (such as agriculture experts and some traditional print materials) [21] of agricultural knowledge transmission and also these traditional methods are slow, expensive and not easy to scale. Even when the experts are available, they may not reach remote villages in time to help the farmers. There are some more resources for the farmers for resolving their queries such as LLMs, Internet and Broadcasting TV shows, but all these resources have their limitations general purpose LLMs has not solved the problem because generally these models do not have domain grounding and context awareness, which leads to inaccurate, incomplete or irrelevant responses to the farmers and information available on the internet may not be reliable and related to the actual concern of the farmer. Furthermore, these methods may not offer the real-time, tailored solutions that farmers often require when encountering immediate agricultural problems.

To address these challenges there are several conventional solutions such as expert-driven helplines, mobile helpline services, and LLMs for general questioning answering. To deliver pre-programmed agriculture information or frequently asked questions, chatbots are used that are rely on rule-based system or structured database to guide the farmer [20]. All these solutions brought some progress but they generally depend on static content and do not support dynamic or ambiguous queries.

Natural language processing (NLP) models and artificial intelligence are progressing rapidly in recent years, allowing conversational systems to be more advanced than ever before. General purpose LLMs are very much capable in generating fluent language responses but often hallucinate or provide unreliable responses when they encounter domain-specific queries. The specialized technical information, and the varying level of ambiguity of a user's query such as incomplete query, incorrect query, ambiguous query and query not related to domain or query which is not clear to farmer himself/herself. A farmer can make an open ended query such as "How to protect my crop? or a subject like "How to get rid of aphids on potato plants?" The system needs to deal with such cases effectively. Most of the chatbots do not maintain the conversational context that is important when farmer provides information with vague terminology, which limits the trust and applicability of these solutions in real world of agriculture.

This paper introduces a contextual agriculture chatbot that addresses these issues by combining three crucial technologies:
RASA Framework: An open-source conversational AI tool that recognizes the intent of the query, extract entities, and do dialogue management [5]. RASA helps the chatbot grasp user questions, pick out important details (such as crops, pests, diseases), and keep conversations flowing, including asking for more information when needed. Retrieval-Augmented Generation (RAG) Pipeline: A mixed method that brings together information lookup and AI text creation. The RAG pipeline finds relevant farming information from a knowledge base using vector matching search and gives this background to the language model making sure answers are based on correct information [14]. Large Language Model (LLM) Integration: Uses the natural language writing abilities of big language models to create clear fitting responses based on the found information [3]. The intuition is that putting these technologies together creates a system that can understand farming questions with different levels of detail, handle complex, vague queries, ask clarification queries, find relevant information, and write natural language answers that are both correct and useful to farmers.
Our experiments show that the proposed solution improves the contextual relevance compared to traditional chatbots or LLMs. The proposed chatbot is able to deal with amnbigous or incomplete queries and by asking clarification questions it maitains the conversation flow. By providing quick reliable guidance to the farmers, this system can help in increasing crop yields, reduce losses from diseases and pests, and also save time and money by avoiding dependencies on different different resources to resolve the query. Our work shows how modern NLP technologies can be used effectively to solve the challenges in agriculture.

2 Related Work

This section discusses previous research that is applicable to our agriculture chatbot, including existing agricultural chatbots, application of conversational frameworks such as RASA, RAG methods, Knowledge Graph (KG) incorporation, and applications of Large Language Models in domain-specific conversation.

2.1 Chatbots: From Rule-Based to LLM-Augmented Systems

Early agricultural chatbot systems mainly used rule-based methods to provide informations to the farmers. These include bots based on pre-determined templates for common questions like irrigation schedules or fertilizer applications or pest control techniques[4, 22] . Following research included machine learning algorithms like sequence-to-sequence (seq2seq) models, Term Frequency–Inverse Document Frequency (TF-IDF), and K-Nearest Neighbors (KNN) to give more relevant responses [2]. But these approaches have some limitations. TF-IDF and KNN depend on text similarity and do not have semantic understanding, while seq2seq models require large domain data and result in incorrect or hallucinated responses. [13] Proposed a method by combining Large Language Model GPT-4 into a Knowledge Graph. The model translates user questions into Cypher queries through prompt engineering and runs them against a domain specific Knowledge Graph. In comparison to general LLMs this system[13] generates more accurate responses, but it requires a carefully constructed knowledge graph which again requires a lot of manual work and expert knowledge to construct and maintain it.

2.2 Machine Learning and Deep Learning for Agricultural QA

There are systems[2] which have used ML methods such as TF-IDF and KNN for user query matching to static QA pairs. These methods are simple to apply but do not work when the queries contain synonyms, grammatical variations, or ambiguous context. Seq2seq models and CNN-based classifiers performed better than keywords matching retrieval but these models cannot deal with various question structures, out-of-vocabulary words or new unseen inputs and do not provide information about how a decision is reached. Some systems [12] also used transfer learning and pre-trained BERT models for better generalization but these systems also produced hallucinated response when user asked domain-related questions, which highlighting the need for external knowledge integration.

2.3 LLMs for Domain-Specific Dialogue

LLMs such as GPT-3.5 and GPT-4 perform well in understanding language but hallucinate when they are not contextually grounded with facts. It is resource expensive to fine-tune them for particular areas like agriculture or healthcare and They can also begin to provide wrong information or be difficult to explain in the sense of how they reached their conclusions (lack of explainability) [7, 19]. In ChatQA 2 [23], LLaMA 3.0 enhances long-context processing with 128K token windows, improving RAG performance. But, this method is expensive and not probably practical in resource-constrained settings. Moreover, its general purpose nature leads to give lower accuracy responses for domain specific areas such as agriculture.

On the other hand, our system uses the LLM as a controlled generation mechanism producing responses based only on retrieved, structured facts from the RAG pipeline. This prevents hallucination while ensuring natural language fluency. This guarantees more resource effective utilization and delivers correct, context-specific answers without the requirement for big context windows.

2.4 Conversational AI Frameworks in Agriculture

There are several conversational platforms such as DialogFlow and IBM Waston[15] which are used to develop conversational systems but because of their poor ability to handle domain specific terms (80-83% accuracy), very limited customization and poor performanace in complex queries, these frameworks face challenges in agriculture sector. Compared to this, RASA's open source platform has high flexibility, improved agricultural terminology recognition (89% accuracy)[24], and efficient contextual dialogue management.

3 Base Chatbot Model

The base model's flow Fig.1 begins when a farmer submits a query through the User Interface . This question is then forwarded to a Large Language Model (LLM) , which processes the query and generates a Cypher query which is used to fetch relevant information from the Knowledge Base . The response retrieved from the Knowledge Base is sent back to the LLM, which then generates a human readable answer and passes it back to the User.

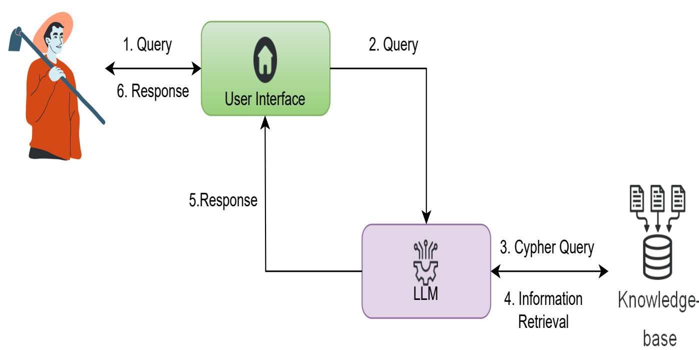

Fig. 1. Base model

3.1 Challenges in Base Model

The basic model has several limitations. First, when the user's query is incomplete or ambiguous it fails to request additional details (that is required for generating response), which leads the system in generating the inaccurate or irrelevant answers or sometimes no response. Second, its scope is strictly dependent to the Knowledge Base, meaning it cannot incorporate external or dynamic sources of information. Third, it struggles to scale effectively as the Knowledge Base grows, leading to performance and maintenance challenges.

3.2 Proposed Methodology

Starting from the ideas discussed in the introduction, this study presents an advanced agricultural chatbot designed to deliver context-specific information to the farmers. This chatbot facilitates rapid access to agricultural knowledge by integrating the RASA framework and RAG pipeline Fig.2. The RASA framework is the core component of the chatbot's which consists of conversational abilities, managing dialogue flow, accurately identifying the user's intent, and extracting key entities like crop_name, pest_name, or diseases mentioned in the query. When a farmer asks a question in natural language, RASA processes the input to understand the need, engaging in clarifying dialogue if the initial query is ambiguous or requires more detail. After collecting the missing information, the RAG pipeline is activated. This component searches an agricultural knowledge base which consists of numeric embeddings, using vector matching techniques to find the most relevant information related to the farmer's query. The retrieved information, rich in relevant context, is then passed to the integrated Large Language Model. Then LLM uses its natural language response generation capabilities by taking the retrieved information obtained by the RAG process and then a clear and easily understandable response is generated, which is then delivered back to the farmer.

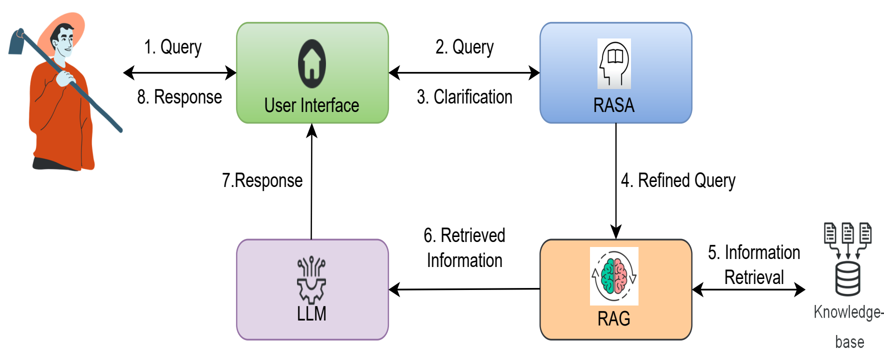

Fig. 2. Basic Flow of the ChatBot

Gathering missing information using RASA Framework The biggest challenge is interpreting the intent of the query so that the system can deal with vague or incomplete user queries from farmers such as "How to protect my crop?" with no identifier of which crop or what problem, and the system having no way to ask intelligent follow-up questions to narrow down the search. The RASA framework, which is used to manage dialogue, natural language understanding, and custom actions, is the foundation for conversational capabilities of chatbots.
RASA NLU, component is required for understanding the user's queries and extracting the information required for continuing the conversation. It performs three tasks mainly: confidence scoring, entity extraction, and intent classifica-

tion. The purpose of user, such as asking about disease management, pest control, or fertilizer recommendations, is determined with the help intent classification. Names of crops, pests, and diseases are among the important details that entity extraction [8] extracts from the user query. Confidence scoring assists in dealing with uncertainty and enables the system to respond properly when ambiguity is encountered by intent confidence value. Farmers may ask question in any way, to precisely recognize the various ways, the NLU module is trained on 1340 number of sample agricultural queries that are annotated with intents and entities.
RASA Dialogue Manager is in charge of managing the conversation flow based on the NLU module outputs. It stores the conversation state, handles information gathering via forms, and decides the next best action based on pre-defined policies. The dialogue manager tracks filled and unfilled slots to make sure that all required information is collected before deciding. Form management enables the bot to collect some context data (e.g., crop name or pest name) required in order to generate correct responses. By incorporating rule-based and machine learning approaches, the dialogue manager follows pre-established conversation structures (stories) but dynamically adapts to each user's context, ensuring a flexible and personalized interaction experience. RASA Custom Actions are used to add the capabilities of the framework to deal with domain-specific needs by acting as a middle component between the RASA core engine and outside modules such as the RAG pipeline [25]. The actions are created to fetch and display specific agricultural information. For example, ActionRetrieveProtectionInfo retrieves information on crop protection measures, while ActionRetrievePestControlInfo gets pest control strategies for particular crop-pest combinations. ActionRetrieveDiseaseControlInfo and ActionRetrieveFertilizerInfo are also similarly offering advice on controlling diseases and using fertilizers, respectively. They are automatically invoked based on state tracking and intent identification by the dialogue manager so that they can be easily integrated with the RAG pipeline to retrieve structured information and employ an LLM for generating natural language responses in an easy-to-use format.

Solving Knowledge model limitations using Retrieval-Augmented Generation (RAG) RAG overcomes the constraints of being limited to a static Knowledge Base and scaling problems. Rather than depending mainly on an internal static database, RAG fetches relevant data from a large external knowledge sources, e.g., documents, web data, or dynamic repositories, during the time of answering a query. This allows the system to provide more precise and accurate answers. Additionally, because RAG separates the storage and retrieval of knowledge from the language generation process, it processes big and expanding databases much more effectively. This makes the system better scalable, simpler to support. As shown in Fig.3, the RAG pipeline allows conversion of user requests into effective responses by initially retrieving content relevant to the request and subsequent generation of a meaningful natural language response based on a language model [11]. The architecture greatly enhances the accuracy and relevance of responses, especially for domain-specific settings such as agriculture.

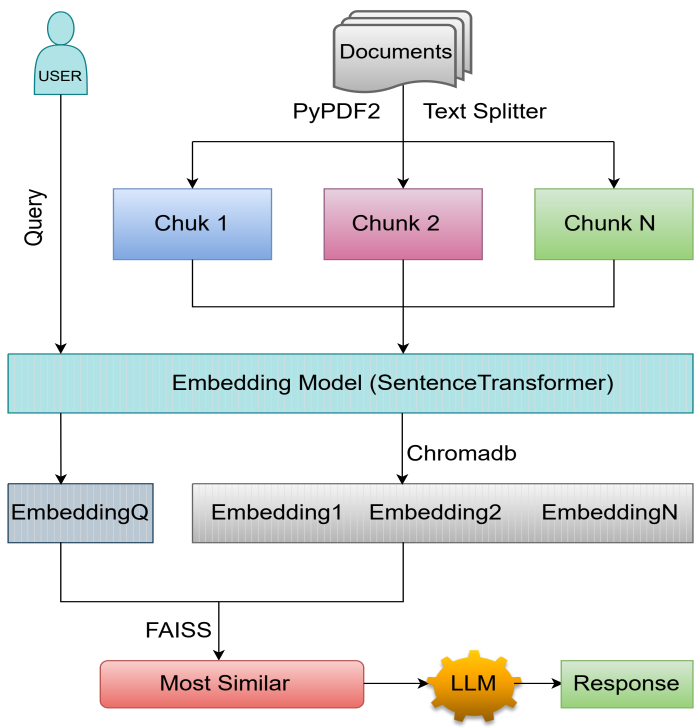

Fig. 3. RAG Pipeline

Document Retrieval component is the base of the RAG system to retrieve relevant information to the query. It uses the vector similarity search algorithm, Facebook AI Similarity Search(FAISS) [16] to compare the processed user queries with an indexed document dataset [9]. This is done by initially transforming user queries and documents into vector embeddings with the help of pre-trained sentence-transformers, enabling the system to identify semantic information over traditional keyword matches [18]. These are the following documents whose embeddings are used to create the knowledge base: Rice Cultivation Handbook of the National Agriculture Research Organization, Farmer's Handbook on Basic Agriculture by Desai Fruits & Vegetables Pvt. Ltd and Handbook of Agriculture by Indian Council of Agriculture Research. These embeddings are indexed with FAISS (Facebook AI Similarity Search), a fast similarity search tool in high-dimensional space. During a query, the system executes a Top-K retrieval, and the most similar K documents are returned based on similarity scores. This method guarantees that the information returned is both semantically similar and context-aware, performing much better than classical keyword-based search methods in agricultural environments [10].

4 System Architecture

The architecture of the agriculture chatbot system is described in detail in this section, along with how the three main technologies: LLM, RASA, and RAG are combined to produce a unified and efficient agricultural assistant. The system processes user queries, retrieves relevant information, and produces natural language responses using a modular architecture that combines several components. Fig.4 illustrates the system's high-level architecture.

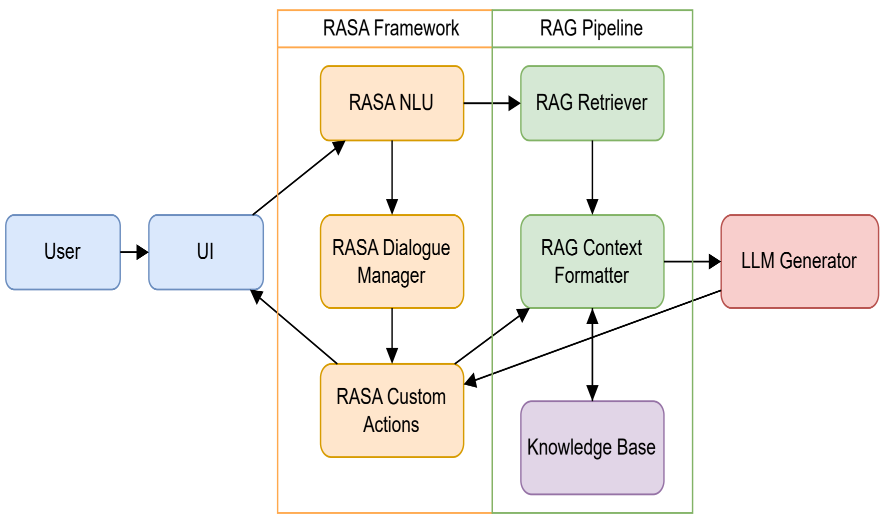

Fig. 4. System Architecture

4.1 Conversation Flow

The complete conversation flow in the agriculture chatbot system is illustrated in Fig.5, showing how the various components interact to process user queries and generate responses back to the user. The conversation flow includes the following steps:
User Query: The user submits a query through the UI.

NLU Processing: Processing of the query for intent and entity identification.
Slot Filling Check: It verifies whether the required slots have been filled.
Clarification (if necessary): In case of missing information, the system creates an explanation question.
RAG Retrieval: When all required information is known to respond to the question, relevant information is retrieved from the knowledge base.
Context Formatting: Information retrieved is framed for the LLM.
Response Generation: The LLM produces a natural language response based on the context.

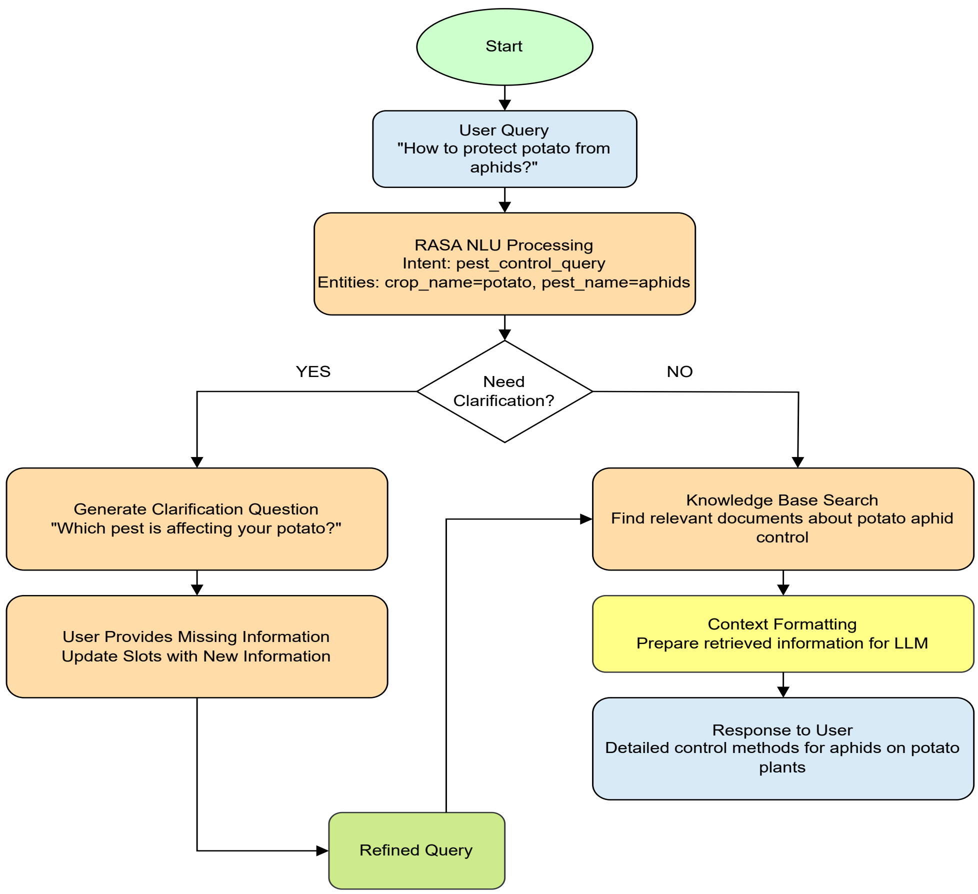

Fig. 5. Conversation Flow of the System

4.2 Intent and Slot Detection using RASA

Understanding what a user wants and the key details in their message is crucial for any helpful chatbot. In our agriculture chatbot, we use the RASA framework to handle this task, which involves two main parts: identifying the user's query intent and collect important pieces of information called entities using slot filling mechanism .

Dataset and Training Data Preparation To train RASA effectively, we needed relevant examples of questions farmers might ask. We downloaded a dataset obtained from the Kisan Call Center (KCC), which contains 40,000 number of real-world queries from farmers from Punjab and Uttar Pradesh. However, this raw dataset included many types of queries, not all of which were suitable for our chatbot focused on providing specific agricultural advice. Therefore, we performed several data processing steps to refine the dataset and focus it on our target domain.
The next stage involves initial pre-processing, which includes identifying and addressing issues like duplicate entries, incomplete sentences, and empty columns. Following this, data filtering is performed to refine the dataset by removing irrelevant queries related to weather, government schemes, animals, flowers, farming fairs, and contact information, along with any remaining duplicates. In the final step, the prepared data is fed into the RASA NLU module to train the natural language understanding system for chatbot development. By filtering out these irrelevant queries, we created a cleaner, more focused dataset. We took experts' reviews on this cleaner version of data and then annotated each query to its corresponding intent type and slots. This processed dataset primarily contained 1340 number of farmers' queries and their corresponding intents and slots. We then used this refined data to create structured training examples for RASA and then feeded into RASA NLU for training.

RASA Training Process After preparing the training dataset, we used it to train the RASA Natural Language Understanding (NLU) model. In order for RASA to recognize different user intents and extract the relevant entities (slots), it must be fed these examples during the training process. RASA's NLU pipeline, defined in the configuration file, specifies the components used for this learning process :
Tokenizer splits the user's message into single words or sub-words (tokens). Featurizers transforms these tokens into numerical forms (features) that can be interpreted by machine learning models. Intent Classifier a machine learning model (DIET - Dual Intent and Entity Transformer) that uses the features to predict the most probable intent of the user query given the patterns derived from the training data. For example this query, "How do I stop aphids on my potatoes?" the intent is classified as the ask_pest_control intent. Entity extractor from the query this module (and the intent classifier in DIET) identifies and extracts specific information pieces (entities or slots) . In the given example,

"potatoes" is identified as the crop entity and "aphids" as the pest entity. The training step tunes these modules so that they can predict intents and extract entities correctly for unseen user messages.

Intent Detection Mechanism The chatbot uses the trained RASA NLU pipeline when it answers a new user query, like "What fertilizer is best for rice?", the query goes through the trained RASA NLU pipeline. The tokenizer splits it down, the embedding module create numerical representations, and finally, the intent classifier analyzes these features. On the basis of its training on examples like "fertilizer for potato", "recommend fertilizer for rice", etc., the classifier predicts the intent. In this case, it would likely predict the ask_fertilizer_info intent with high confidence.

Slot Extraction Process Concurrently, the entity extractor module operates to tag significant words or phrases in the query as slots. For the query "What fertilizer is suitable for potato?", the entity extractor, which has been trained on such similar examples wherein crop names have been tagged, would tag "potato" and mark it with the crop entity type. These extracted entities (slots) are then saved by RASA's dialogue manager and can be utilized later in the conversation, for example, to ask the knowledge base specifically about potato or to verify information with the user.

5 Evaluation and Results

Our system's effectiveness is evaluated by performing the independent evaluations of its two central components: RASA and RAG. The modular evaluation method enabled us to test each component individually, which gives a better insight into their individual contribution to the functionality of the whole system. RASA is tested for its intent classification and entity recognition performance, whereas RAG is tested on the basis of how well it can retrieve useful information from the knowledge base in response to user queries.

This modular test strategy enabled us to test each component in isolation, giving a better insight into their individual contribution to the system functionality as a whole. RASA was tested mainly on its intent classification and entity recognition performance, whereas RAG was evaluated on how well it can retrieve content from the knowledge base when a user asks any query.

5.1 Testing of RASA

To analyze the performance of the trained RASA model, we performed testing on two of its most important features: intent classification and entity recognition. For training the RASA NLU module, we utilized a total of 35 unique intents. For testing, we created a dataset of 140 user queries, both abstract and complete types queries.

Intent Classification Accuracy is evaluated by doing a comparison between predicted intents and ground truth labels for each query in the test dataset. The model achieved the an accuracy of 88.97% in intent classification, which is indicating its strong ability in interpreting the user's intention.

Entity Recognition Accuracy We also checked the accuracy of the RASA module to recognize relevant entities in the queries. For this purpose, we annotated the test set manually and matched the entities that were extracted with the ground truth. The model achieved an entity identification accuracy of 96.67%, which shows a very high degree of precision in extracting the entities from the user query.

5.2 Testing of RAG

We tested the performance of our Retrieval Augmented Generation (RAG) module by doing a comparison between its output and manually curated set of question-and-answer pairs. This RAG Testing assures that the system can retrieve relevant information and create correct, natural-language answers. For example:
Query: Which upland rice variety has an aromatic grain, and what are its maturity period and yield?
Ground Truth Answer: NERICA 1 has an aromatic grain. Its maturity period is 105–115 days, and its yield is 3–4 tons per hectare.

Ground Truth Creation For creating a reliable benchmark and for preparing the ground truth used in testing RAG modules, we selected 50 sample questions from agricultural handbook documents and determined correct answers ourselves. We crafted every question carefully with a focus on key details, and the correct answers we determined are used as the reference to ground truth.

RAG System Response Using RAG system then we generated response for all the curated set of questions by retrieving the relevant information from the vector database.
For the above example the system generated : The upland rice variety with an aromatic grain is NERICA 1. Its maturity period is 105-115 days, and the yield ranges from 3 to 4 tons per hectare.

Evaluation Method The responses generated by system were compared with the ground truth answers using the following criteria:
Content Accuracy: Does the generated response include the correct and specific information that has asked in the query such as variety name, fertilizer name, pest name, and yield values mentioned in the response?
Completeness: Does the response generated by system cover all components of the question without leaving any aspect?

Fluency: Is the response grammatically correct and easy to understand?
Other embedding matching methods Fig.6 such as BERTScore, Cosine similarity, Euclidean Distance, Jaccard similarity and BLUE score are used to compare the generated response with ground truth.
All answers were evaluated by agricultural domain specialists in terms of content

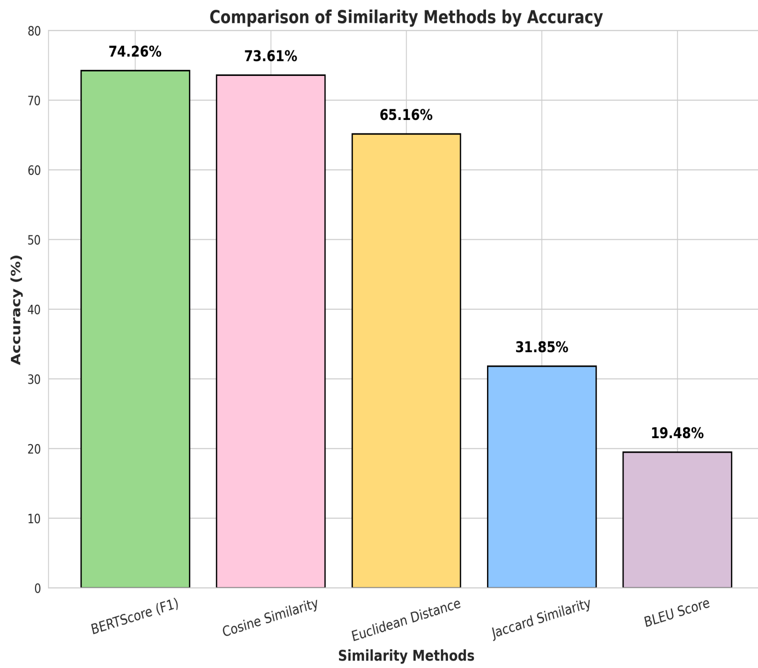

Fig. 6. Similarity Comparison

correctness, completeness, and fluency. A total of 50 ground truth question-answer pairs, the RAG system produced correct outputs in 49 instances.

6 Conclusions and Future Work

This research proposed an agriculture chatbot that efficiently integrates RASA, RAG, and LLM technologies to provide precise and relevant information to the farmers which helps to resolve their queries. The major contribution lies in the development of a novel architectural framework that uses RASA for managing dialogue and clarifying incomplete queries, RAG for domain-specific knowledge retrieval, and LLM for natural language generation. The system has an intelligent clarification facility that can recognize incomplete or ambiguous user input, improving the accuracy of responses. The current system has some limitations at present, It supports English conversations only, lacks long-term memory for dialogue continuity, and does not support multimodal inputs like image recognition. The knowledge base also does not get updated automatically it requires manual efforts for updating its knowledge. Future work will focus on Enhancing the knowledge base to cover a wider varieties of crops, pests and diseases will enhance the system's comprehensiveness. By including the image-based input will enable users to upload images for visual inspection so that problems can be identified more accurately. Automating the updating of the knowledge base will keep the system up-to-date with new agricultural knowledge. Lastly, implementing the chatbot in actual farm settings and performing user studies will be able to evaluate its efficiency and will identify more future enhancements.

References

1. Alam, A.: Soil degradation: A challenge to sustainable agriculture. International Journal of Scientific Research in Agricultural Sciences **1**, 50–55 (08 2014). https://doi.org/10.12983/ijsras-2014-p0050-0055
2. Biswas, R., Goel, N.: Intelligent chatbot assistant in agriculture domain. In: Saini, M.K., Goel, N., Shekhawat, H.S., Mauri, J.L., Singh, D. (eds.) Agriculture-Centric Computation. pp. 180–194. Springer Nature Switzerland, Cham (2023)
3. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners (2020), https://arxiv.org/abs/2005.14165
4. Chathurya, C., Sachdeva, D., Arora, M.: Agriculture chatbot(agribot)using natural language processing. In: 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT). pp. 1–5 (2023). https://doi.org/10.1109/ICCCNT56998.2023.10306512
5. Darapaneni, N., Raj, S., V, R., Sivaraman, V., Mohan, S., Paduri, A.: Lstm-rasa based agri farm assistant for farmers (04 2022). https://doi.org/10.48550/arXiv.2204.09717
6. De Clercq, D., Nehring, E., Mayne, H., Mahdi, A.: Large language models can help boost food production, but be mindful of their risks. Frontiers in Artificial Intelligence **Volume 7 - 2024** (2024). https://doi.org/10.3389/frai.2024.1326153, https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2024.1326153
7. Gekhman, Z., Yona, G., Aharoni, R., Eyal, M., Feder, A., Reichart, R., Herzig, J.: Does fine-tuning LLMs on new knowledge encourage hallucinations? In: Al-Onaizan, Y., Bansal, M., Chen, Y.N. (eds.) Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 7765–7784. Association for Computational Linguistics, Miami, Florida, USA (Nov 2024). https://doi.org/10.18653/v1/2024.emnlp-main.444, https://aclanthology.org/2024.emnlp-main.444/
8. Henderson, M., Casanueva, I., Mrkšić, N., Su, P.H., Wen, T.H., Vulić, I.: Convert: Efficient and accurate conversational representations from transformers (2020), https://arxiv.org/abs/1911.03688

9. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. IEEE Transactions on Big Data **7**(3), 535–547 (2021). https://doi.org/10.1109/TBDATA.2019.2921572
10. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6769–6781. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.550, https://aclanthology.org/2020.emnlp-main.550/
11. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., tau Yih, W.: Dense passage retrieval for open-domain question answering (2020), https://arxiv.org/abs/2004.04906
12. Kpodo, J., Kordjamshidi, P., Nejadhashemi, A.P.: Agxqa: A benchmark for advanced agricultural extension question answering. Computers and Electronics in Agriculture **225**, 109349 (2024). https://doi.org/https://doi.org/10.1016/j.compag.2024.109349, https://www.sciencedirect.com/science/article/pii/S0168169924007403
13. Kumar, M., Wilson, S., Goel, N., Ginige, A.: Krishiai: Architecture for harnessing capabilities of llms for delivery of accurate cultivation information to farmers. In: Saini, M.K., Goel, N., Miguez, M., Singh, D. (eds.) Agricultural-Centric Computation. pp. 68–79. Springer Nature Switzerland, Cham (2025)
14. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive nlp tasks (2021), https://arxiv.org/abs/2005.11401
15. Liu, X., Eshghi, A., Swietojanski, P., Rieser, V.: Benchmarking natural language understanding services for building conversational agents (2019), https://arxiv.org/abs/1903.05566
16. Mescheder, L., Nowozin, S., Geiger, A.: The numerics of gans (2018), https://arxiv.org/abs/1705.10461
17. Pretty, J.: Agricultural sustainability: Concepts, principles and evidence. Philosophical transactions of the Royal Society of London. Series B, Biological sciences **363**, 447–65 (07 2007). https://doi.org/10.1098/rstb.2007.2163
18. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1410, https://aclanthology.org/D19-1410/
19. S, S.K., Khan, A.K.M.A., Banday, I.A., Gada, M., Shanbhag, V.V.: Overcoming llm challenges using rag-driven precision in coffee leaf disease remediation (2024), https://arxiv.org/abs/2405.01310
20. Samuel, D., Skarga-Bandurova, I., Sikolia, D., Awais, M.: Agrollm: Connecting farmers and agricultural practices through large language models for enhanced knowledge transfer and practical application (02 2025). https://doi.org/10.48550/arXiv.2503.04788
21. Sennuga, O.: Title: Use of ICT among Smallholder Farmers and Extension Workers and its Relevance to Sustainable Agricultural Practices in Nigeria) MSc in

Agriculture and Development Thesis Title: Application of Information and Communication Technologies (ICT) among smallholder farmers in Nigeria (11 2022)

22. Suebsombut, P., Sureephong, P., Sekhari, A., Chernbumroong, S., Bouras, A.: Chatbot application to support smart agriculture in thailand. In: 2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT NCON). pp. 364–367 (2022). https://doi.org/10.1109/ECTIDAMTNCON53731.2022.9720318
23. Xu, P., Ping, W., Wu, X., Xu, C., Liu, Z., Shoeybi, M., Catanzaro, B.: Chatqa 2: Bridging the gap to proprietary llms in long context and rag capabilities (2025), https://arxiv.org/abs/2407.14482
24. Yanamala, A.K.Y., Suryadevara, S.: Adaptive middleware framework for context-aware pervasive computing environments. International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence **13**(1), 35–57 (2022)
25. Youn, J., Tagkopoulos, I.: Kglm: Integrating knowledge graph structure in language models for link prediction (2023), https://arxiv.org/abs/2211.02744