Main outcome-

We aim to design a product which helps the way the farmers gain upto-date knowledge regarding various agricultural and farming practices from the Internet. With the rise and forthcoming era of LLMs and advanced chatbots, we aim to create an interface where fellow farmers from a neighborhood can connect with each other and share their knowledge.

Our main goal is to help farmers who are starting in the agricultural practices and lack of resources to keep moving forward in their path while our product ensures they get the right knowledge through our devised advance conversational-AI framework, community forms and our interactive interface to merge it all!

Data

The data used consists of openly available PDF/Books from various government and educational sites. The text extracted from these Books and PDFs and used to fine-tune a SLM (Small Language Model) for a QnA Tasks as per of our framework.

**1. Variable/Parameter: Soil**

* **Source**: Agricultural Soil Reports/Handbooks PDF
* **Data Description**: Types and kinds of soil grown around(specific to India), factors affection it and soil management techniques.
* **Data Type**: String

**2. Variable/Parameter: Plant Diseases**

* **Source**: Common Plant Diseases Reports/Handbooks PDF
* **Data Description**: Types and kinds of diseases grown around(specific to India) agricultural fields , factors affection it and disease/pests management techniques.
* **Data Type**: String

**3. Variable/Parameter: Irrigation**

* **Source**: Agricultural Irrigation Reports/Handbooks PDF
* **Data Description**: Types and kinds of irrigation techniques(specific to India), factors affection it and water-efficient management techniques.
* **Data Type**: String

**4. Variable/Parameter: Best Agricultural Practices**

* **Source**: Best Agricultural practices for beginners in farming Reports/Handbooks PDF
* **Data Description**: Set of handy tips and techniques along with an overall outlook of agriculture and farming in our country-especially catered to the beginners.
* **Data Type**: String

Methodology-

Our conversational-AI framework consists of three major parts-

1. MoE (Mixture of Experts Fine tune LLMs)
2. Conversation Managing LLMs
3. Translation / RAG
4. MoE –

It consists of three SLMs (Phi-2 family\_ fine tuned on each of three variables/parameters stated above to get knowledge and become expert in their own domain, mainly over QnA format.

It ensures that the query sent to this part of the framework can handle complex query in the form of broken subproblems, each catered to the specific fine-tine “expert”.

1. Conversation Managing LLMs

It consists of two LLMs (Gemini family-flash and pro).

The pro is used for three tasks- Classification of queries in Simple/Complex query types through prompt-based multi-label classification(Soil, Plant Diseases or Irrigation) and another to break the complex query into the defined number of subproblems as the labels assigned to it. Finally, it takes of the translation part from the user chosen language to English for better management then to user chosen language.

The Flash is what manages and drives the conversations forward with RAG/MoE inputs.

1. RAG/Translation

We have defined RAG used in our project in two ways-

1. RAG for simple queries
2. RAG for complex queries

RAG for simple queries is when the user asks queries regarding only of the three parameters above and is easy to deal with conventional RAG-based Conversational-AI. Here, the external knowledge from our chosen number of PDFs to incorporate up-to date and appropriate knowledge into the LLM which answer the user queries in a friendly conversational manner.

RAG for complex queries comes from an aggregation for all the response from the three fine-tuned SLM, from each “expert” answering to their own subproblem passed by the managing LLMs. These acts as external inputs and are then passed onto the Managing LLMs (flash for conversation), translated and to the user.

The above two steps are done to prevent Hallucination in LLMs and provide appropriate content to the end user.

Website Features :-

The website is built using a Turborepo, which is a high-performance build system that helps manage the complexity of a monorepo. The frontend is built using Next.js, a React framework that provides server-side rendering, static site generation, and other performance optimizations.The backend is built using an Express server that handles the communication between the frontend and the ML backend. The Express server is responsible for the following tasks:

1. **Request Handling**: The Express server receives all the requests from the frontend, such as user queries, and passes them to the Redis queue.
2. **Redis Queue**: A Redis queue is used to manage the incoming requests. This helps distribute the load across multiple Node.js processes, ensuring the ML backend does not get overwhelmed.
3. **Node.js Processes**: Multiple Node.js processes are set up to consume the requests from the Redis queue. These processes then pass the requests to the ML backend for processing.
4. **ML Backend Integration**: The Node.js processes communicate with the ML backend, which is responsible for handling the various ML tasks, such as query classification, subproblem generation, and retrieval-augmented generation (RAG).
5. **Database Integration**: The website uses a single PostgreSQL database, provided by Neon DB, as the single source of truth. This database stores all the user, farmer, and retailer data, as well as the content for the website.
6. **User Management**: The Next.js frontend handles user management, including authentication, authorization, and user data storage in the PostgreSQL database.
7. **Scalability**: The use of a Redis queue and multiple Node.js processes ensures the application can scale to handle a large number of concurrent user requests without overloading the ML backend.
8. **Deployment**: The entire application, including the frontend, backend, and ML components, is packaged and deployed using Docker containers, ensuring consistent and reliable deployment across different environments.
9. **Monitoring and Logging**: The application includes comprehensive monitoring and logging solutions to track performance, errors, and user activity, enabling the team to quickly identify and resolve issues.
10. **Security**: The website incorporates industry-standard security practices, such as SSL/TLS encryption, user authentication, and input validation, to protect user data and prevent unauthorized access.

By leveraging these technical features, the website aims to provide a scalable, secure, and user-friendly platform for farmers to access agricultural knowledge and connect with their community.

Causes or Limitations-

1. MoE is an extensive task when it comes to increasing the number of experts since it won’t be feasible without heavy compute and other resources, hence should be made into production with care.
2. Translation should be done with the help of Translation APIs like the ones offered by Google or indic LLMs (like by AI4Bharat) instead of hard prompting the LLMs to do so as it may lead to unexpected error on larges ones when into production(we are yet to test this). We chose Gemini Pro because of its free tier services and appreciable translation abilities for our use-case.
3. A ML model should be more appropriate approach to train and use it for Simplex/Complex query segregation to have better control and usability. However, hard prompting worked in our case.
4. High quality dataset, especially access to quality PDFs/Books for generation of QnA pairs and RAG. Since we couldn’t find many open and big pdf on the same nor the QnA pairs we had to make use paid services like OpenAI APIs to create the pairs. We have uploaded the same for use by the community on Hugging Face handle (<https://huggingface.co/YuvrajSingh9886>)

Similar solutions-

How faithful are RAG models? Quantifying the tug-of-war

1. between RAG and LLMs’ internal prior (Research Paper)
2. Adaptive-RAG-Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity (Research Paper)
3. Adaptive Mixture of Experts in 1991 by Robert A

Decision

A total of 5 trial run were done.

The output (final translated one) is appropriate for a given, whether its simple or complex. The Pro , without custom few-shot prompts is able to correctly carry out the classification and creation of sub-problems tasks and assign correct labels, well-suited for the kinds of question any beginner in the same field is likely to have.

The translation is also appropriate 80% of the time and convey the meaning and the information. An Indic LLM specifically for the specified language conversions may do the job well.

The multi-label classification is 90% accurate and the sub-problems by breaking the main query down turns out to be accurate and concise representations/paraphrased of the main query, thus providing rich and appropriate answers to the user in the end.

Validation Strategy

The trial ran for 5 times.

Each of the trial consisted of queries with Hindi as user language and with variety of complexity, ranging of single label (Soil, Irrigation, Plant diseases) to every permutations of these as queries to our devised framework.

Each of the outputs-labels classified, sub-problems, the translated version(Hindi-English and English-Hindi) were taken and the final output was noted, based on which the above the observations under ‘Decisions’ were made.

Date –

Four days, each time one of the fine-tuning tasks were done.