**📘 Project Report: Intelligent Agricultural Query Chatbot using BERTopic, FAISS & GPT**

**🧩 Introduction**

Agriculture in India plays a critical role in the economy, yet farmers often struggle to get timely and accurate information about crop management, government schemes, credit availability, and more. To address this, we propose an intelligent **Agricultural Chatbot** trained on real farmer queries and powered by **BERTopic**, **FAISS**, and **GPT (OpenAI)** to provide relevant responses in real time.

**🎯 Objectives**

* **Automate query resolution** for common farmer concerns using AI.
* **Use topic modeling** to group similar questions and extract dominant themes.
* **Integrate vector search** (FAISS) for scalable semantic retrieval.
* **Leverage OpenAI's LLM** to generate natural language answers from topic-specific contexts.

**🗂️ Dataset**

* **Source**: Kisan Call Center Queries dataset (CSV)
* **Columns**:
  + questions: Farmer queries
  + answers: Expert responses
* **Preprocessing**:
  + Handled missing values
  + Combined question + answer into a single field for embedding

**🧠 Architecture Overview**

text

CopyEdit

┌──────────────┐

│ Farmer Query │

└────┬─────────┘

│

▼

┌──────────────────────────────┐

│ SentenceTransformer Embedding│ ← all-MiniLM-L6-v2

└────┬─────────────────────────┘

│

▼

┌────────────────────┐ ┌────────────────┐

│ BERTopic Model │ │ FAISS Index │

│ (topic assignment) │ │ (topic search) │

└────┬────────────────┘ └──────┬─────────┘

│ │

└────────────┬────────────────┘

▼

┌────────────────────────────┐

│ Prompt-augmented LLM Query │ ← GPT-4o

└────────────┬───────────────┘

▼

┌──────────────┐

│ Final Answer │

└──────────────┘

**🧰 Technologies Used**

| **Tool/Library** | **Purpose** |
| --- | --- |
| **BERTopic** | Topic modeling of farmer queries |
| **FAISS** | Fast Approximate Nearest Neighbor Search |
| **Sentence-BERT** | Semantic embeddings |
| **OpenAI GPT-4o** | Natural language generation |
| **UMAP, HDBSCAN** | Dimensionality reduction & clustering |
| **Python** | Backend logic and pipeline integration |
| **Pickle/Numpy** | Model & index storage |

**🔍 Implementation Details**

**1. Data Embedding**

* Used all-MiniLM-L6-v2 via SentenceTransformer.
* Cached embeddings in .npy to avoid recomputation.

**2. BERTopic Modeling**

* Combined question + answer to better capture context.
* Used custom UMAP and HDBSCAN (with prediction\_data=True) for reusable clustering.
* Saved the model (bertopic\_model.pkl) for future runs.

**3. FAISS Indexing**

* Created a topic-level document dictionary (topic\_docs).
* Encoded each topic’s combined document and indexed via FAISS IndexIVFFlat.
* Saved faiss\_index.bin, topic\_ids.pkl, and topic\_docs.pkl.

**4. Prompt Engineering + LLM Response**

* For a new query:
  + Find the closest topic using FAISS.
  + Retrieve its context.
  + Formulate a prompt to GPT-4o for an answer.
* Ensured LLM stays within context and doesn't hallucinate.

**5. Caching & CLI**

* Added a response\_cache to avoid redundant API calls.
* CLI loop for interactive usage.

**🧪 Testing & Example**

**Query**: “How to avail Kisan Credit Card loan for sali crop?”

**Process**:

* FAISS finds topic related to credit card schemes.
* Context retrieved with actual KCC responses.
* Prompt sent to GPT-4o.

**Expected Response**:

"To avail a Kisan Credit Card (KCC) loan for Sali crop, visit your nearest bank branch with land documents, ID proof, and income details. Ensure you meet the eligibility criteria for seasonal crop loans. Contact your agriculture officer for guidance."