AI Research Assistant - RAG Chatbot with Memory

…

By Sahil Faizal

Contents

- Introduction
- RAG System
    - Introduction
    - Data Flow
    - Tool Used
    - System Design
    - Key Features
    - Working Screenshots
- Agentic RAG System
    - Introduction
    - Tools Used
    - System Design
    - Key Features
    - Working Screenshots
- Deployment Architecture
- Future Scope

Introduction

Repositories likes ArXiv, Google Scholar, Semantic Scholar contains vast amounts of research papers, but finding relevant information within lengthy PDFs can be tedious. This problem paved the way for these chatbots which allows users to ask questions and receive precise answers based on research papers.

In this project, a conversational RAG chatbot has been implemented using the Cohere LLM model, which retrieves relevant information from ArXiv. Additionally, an agentic RAG chatbot has been developed using the OpenAI LLM model, which retrieves information from Semantic Scholar.

Note: Due to time constraints, I was unable to integrate both functionalities into a single application. Initially, the RAG chatbot was built, and later, the agentic version was developed for further refinement.

RAG System

Introduction

The AI Research Assistant is a Retrieval-Augmented Generation (RAG) system built using Streamlit and LangChain, and Cohere’s LLM capabilities. The architecture follows a modular design with clear separation of components.

Core Components:

- **Frontend Interface:** Streamlit-based UI with chat interface and evaluation dashboard
- **Document Retrieval System:** FAISS vector store with Cohere embeddings
- **LLM Integration:** Cohere’s command-r-plus model for response generation
- **Conversation Management:** History-aware retrieval chain with session state persistence

Data Flow

1. User inputs research keywords
2. System fetches papers from ArXiv API
3. Documents are transformed, embedded and indexed
4. User queries are processed through a history-aware retriever
5. Retrieved contexts and query are send to LLM
6. Responses with citations are presented to users

Tools Used

- LangChain (Framework)
- Streamlit (User Interface)
- FAISS (Vector Database)
- ArXiv API (Data Collection API)
- Cohere LLM (LLM Model)
- Python (Programming)
- Git (Version Control)

System Design

1) Retrieving research papers

- The ArXiv API provides an efficient way to retrieve research papers based on the keywords and create a knowledge base, using which the chatbot could generate its response for the user query

2) Storing documents in the vector database

- Since searching the full database is inefficient, the data is converted into vector embeddings using Cohere's embedding model and used FAISS to store and retrieve documents based on these embeddings

3) Storing conversational memory

- While retrieval helps answer standalone questions, an ideal chatbot shall be able to remember past conversations that the user has had with the system

System Design

4) History-aware retriever sub-chain

- A sub-chain is defined to take historical messages and the latest user question to reformulate the question if it makes reference to any information in the historical information.

5) Building the LLM chain

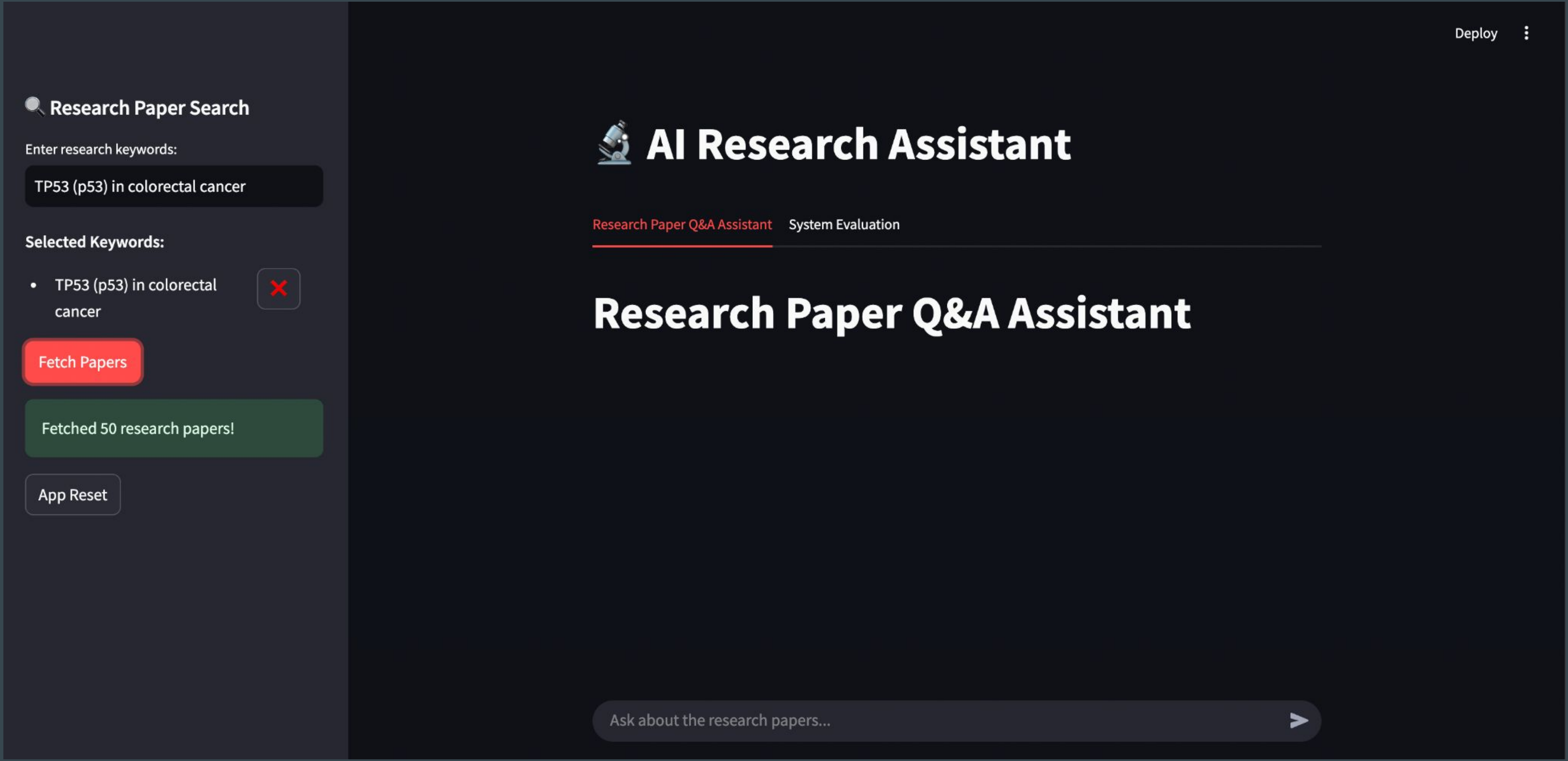
- Fetch the pre-trained LLM, create a relevant prompt and generate a second sub-chain to retrieve context along with the conversational history and user query to generate an answer. Then build the final rag chain which applies the sub-chains in sequence.

6) Stateful management of chat history

- Injecting chat history into input and updating it after invocation to make sure that chat history persists and gets automatically inserted and updated in the chain after each user-system interaction

Key Features

1) **Research Paper Retrieval and Processing**
    a) Keyword-based search of ArXiv papers
    b) Automatic metadata extraction for citations
    c) Vector embedding and indexing for semantic retrieval
2) **Intelligent Conversation**
    a) Context-aware responses with citations support
    b) Conversation history for follow-up questions
    c) Document-grounded answers with minimize hallucinations
3) **Evaluation Framework**
    a) RAGAS-based evaluation metrics (faithfulness, relevancy, precision, recall)
    b) Support for manual entry and test set uploads
    c) Visualization of evaluation results with metrics dashboard
4) **User Experience**
    a) Intuitive chat interface for paper exploration
    b) Citation management for academic integrity
    c) Session state persistence for continuous research sessions
    d) System reset functionality for new research topics



🔍 **Research Paper Search**

Enter research keywords:

TP53 (p53) in colorectal cancer

Selected Keywords:

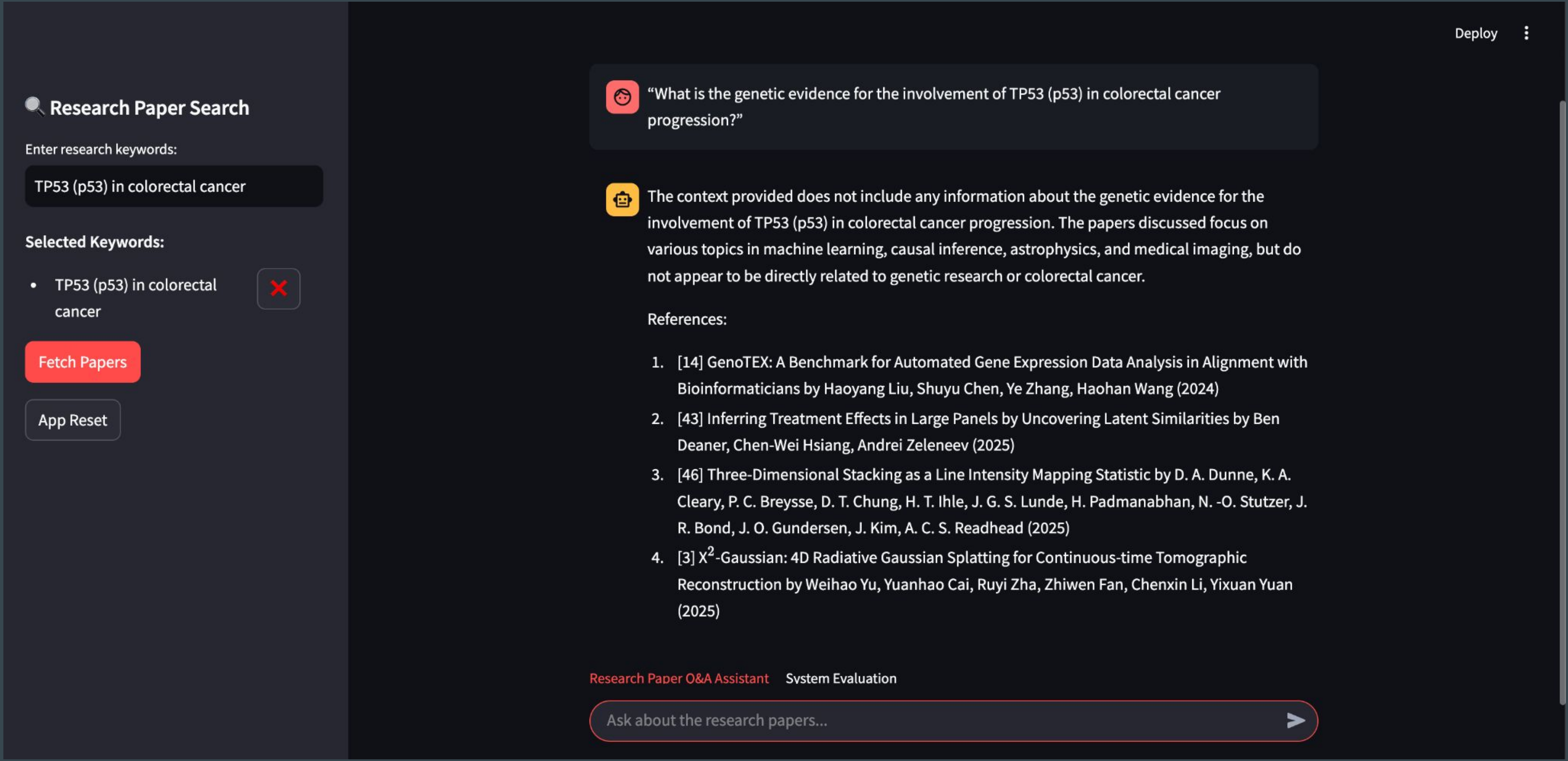
- TP53 (p53) in colorectal cancer

Fetched 50 research papers!

🔬 AI Research Assistant

Research Paper Q&A Assistant

Ask about the research papers...



Enter research keywords: TP53 (p53) in colorectal cancer

Selected Keywords:

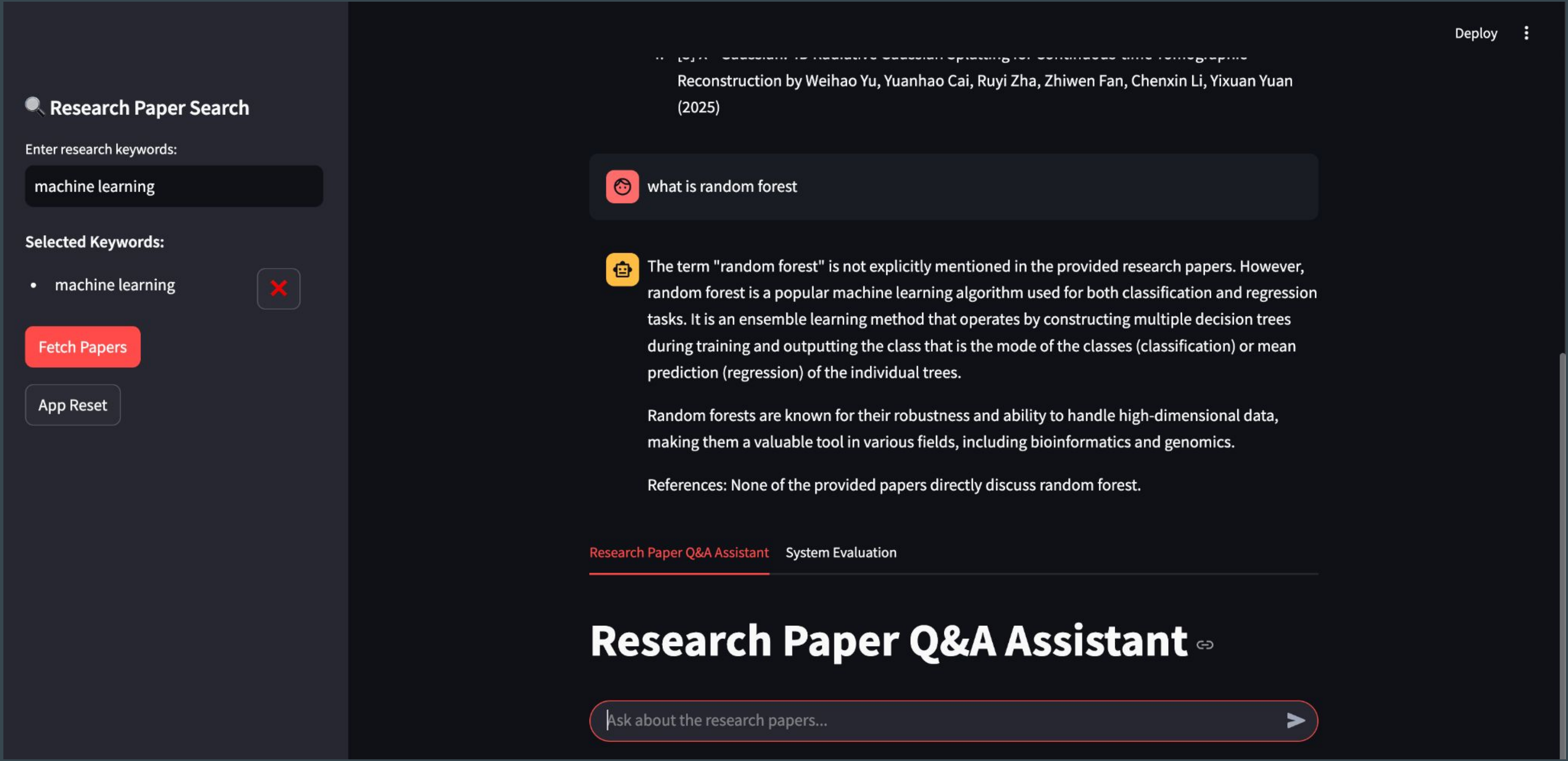
- TP53 (p53) in colorectal cancer

"What is the genetic evidence for the involvement of TP53 (p53) in colorectal cancer progression?"

The context provided does not include any information about the genetic evidence for the involvement of TP53 (p53) in colorectal cancer progression. The papers discussed focus on various topics in machine learning, causal inference, astrophysics, and medical imaging, but do not appear to be directly related to genetic research or colorectal cancer.

References:

1. [14] GenoTEX: A Benchmark for Automated Gene Expression Data Analysis in Alignment with Bioinformaticians by Haoyang Liu, Shuyu Chen, Ye Zhang, Haohan Wang (2024)
2. [43] Inferring Treatment Effects in Large Panels by Uncovering Latent Similarities by Ben Deaner, Chen-Wei Hsiang, Andrei Zeleneev (2025)
3. [46] Three-Dimensional Stacking as a Line Intensity Mapping Statistic by D. A. Dunne, K. A. Cleary, P. C. Breysse, D. T. Chung, H. T. Ihle, J. G. S. Lunde, H. Padmanabhan, N. -O. Stutzer, J. R. Bond, J. O. Gundersen, J. Kim, A. C. S. Readhead (2025)
4. [3] X^2-Gaussian: 4D Radiative Gaussian Splatting for Continuous-time Tomographic Reconstruction by Weihao Yu, Yuanhao Cai, Ruyi Zha, Zhiwen Fan, Chenxin Li, Yixuan Yuan (2025)



Enter research keywords: machine learning

- machine learning

Reconstruction by Weihao Yu, Yuanhao Cai, Ruyi Zha, Zhiwen Fan, Chenxin Li, Yixuan Yuan (2025)

what is random forest

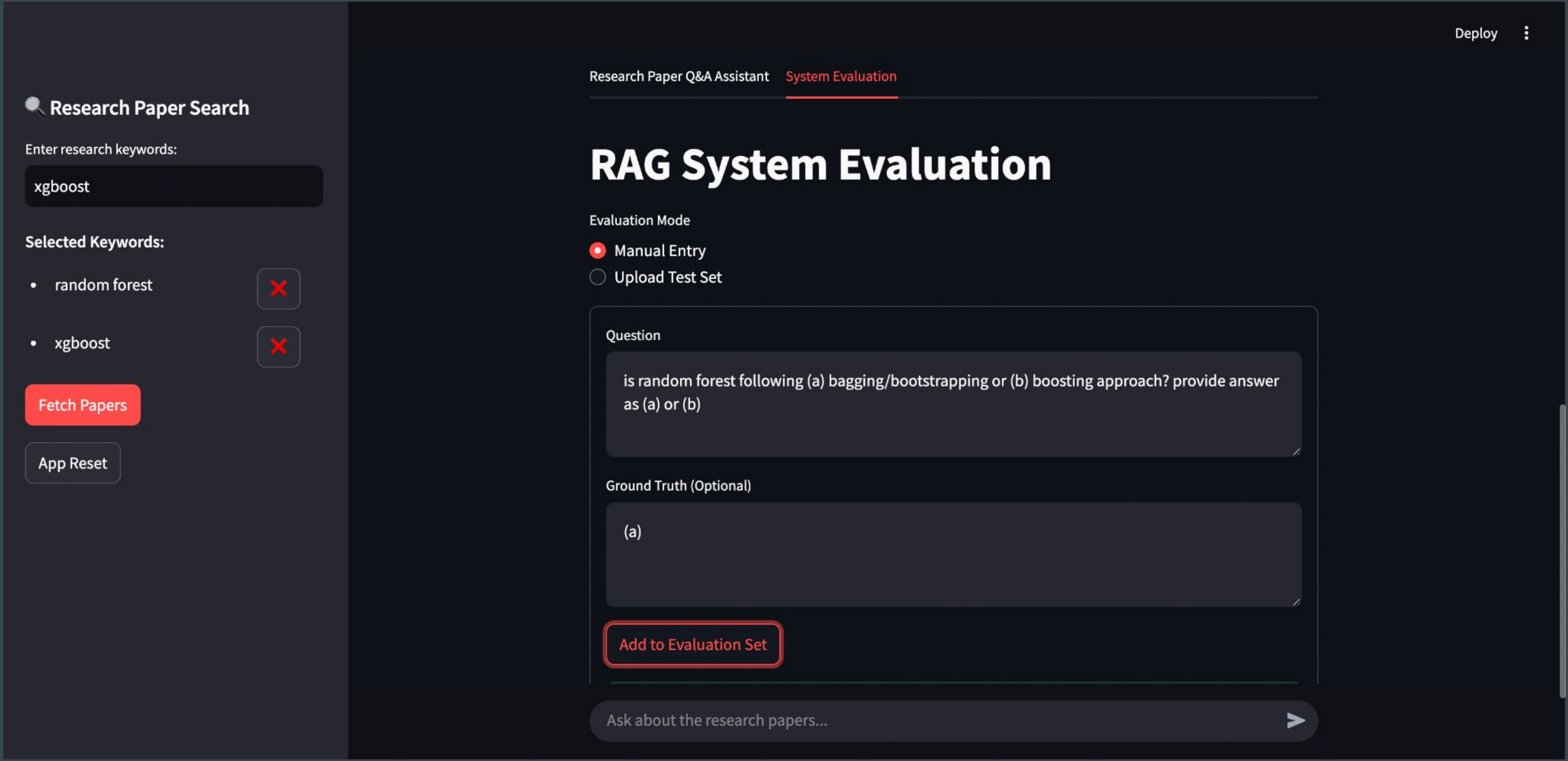
The term "random forest" is not explicitly mentioned in the provided research papers. However, random forest is a popular machine learning algorithm used for both classification and regression tasks. It is an ensemble learning method that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Random forests are known for their robustness and ability to handle high-dimensional data, making them a valuable tool in various fields, including bioinformatics and genomics.

References: None of the provided papers directly discuss random forest.

Research Paper Q&A Assistant

Ask about the research papers...



🔍 **Research Paper Search**

Enter research keywords:

xgboost

Selected Keywords:

- random forest ✕
- xgboost ✕

RAG System Evaluation

Evaluation Mode

● Manual Entry

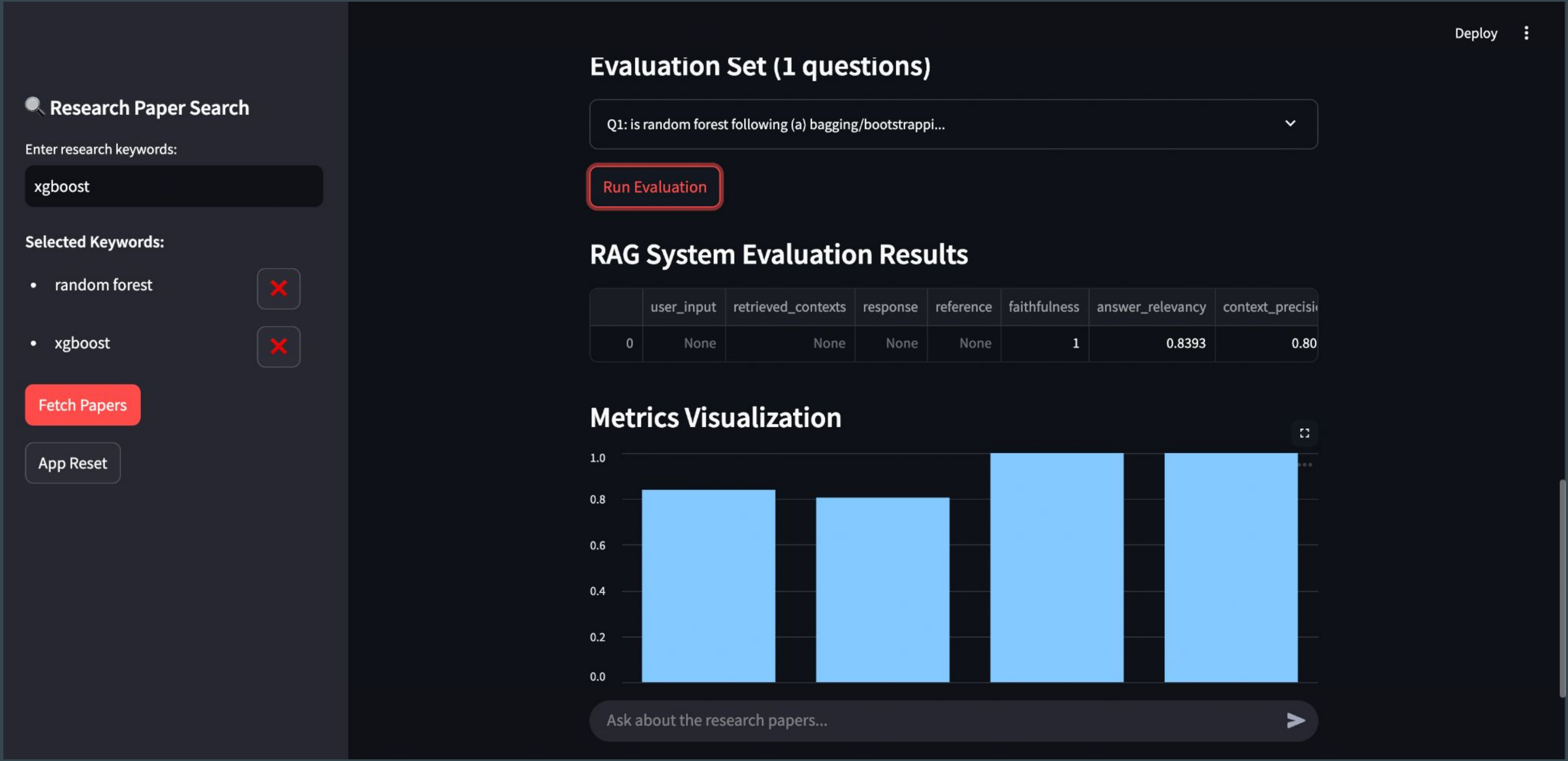
○ Upload Test Set

Question

is random forest following (a) bagging/bootstrapping or (b) boosting approach? provide answer as (a) or (b)

Ground Truth (Optional)

(a)



- random forest ✕
- xgboost ✕

Evaluation Set (1 questions)

Q1: is random forest following (a) bagging/bootstrappi...

RAG System Evaluation Results

	user_input	retrieved_contexts	response	reference	faithfulness	answer_relevancy	context_precisi
0	None	None	None	None	1	0.8393	0.80

Metrics Visualization

Agentic RAG System

Introduction

The research assistant system is designed to facilitate scholarly information retrieval and understanding through an intelligent agent architecture. The design prioritises effective paper discovery, contextual storage, and informed follow-up interactions.

Design Principles:

- **Contextual Knowledge Retrieval -**
  - Maintains conversation history to enable follow-up questions about previously retrieved papers
- **Semantic Search Integration -**
  - Leverages the Semantic Scholar API for accurate and relevant paper discovery
- **Vector-Based Storage -**
  - Employs embedding-based retrieval for efficient similarity search across research papers

Tools Used

- LangChain (Framework)
- Streamlit (User Interface)
- Chroma (Vector Database)
- Semantic Scholar API (Data Collection API)
- OpenAI (LLM Model)
- Python (Programming)
- Git (Version Control)

System Design

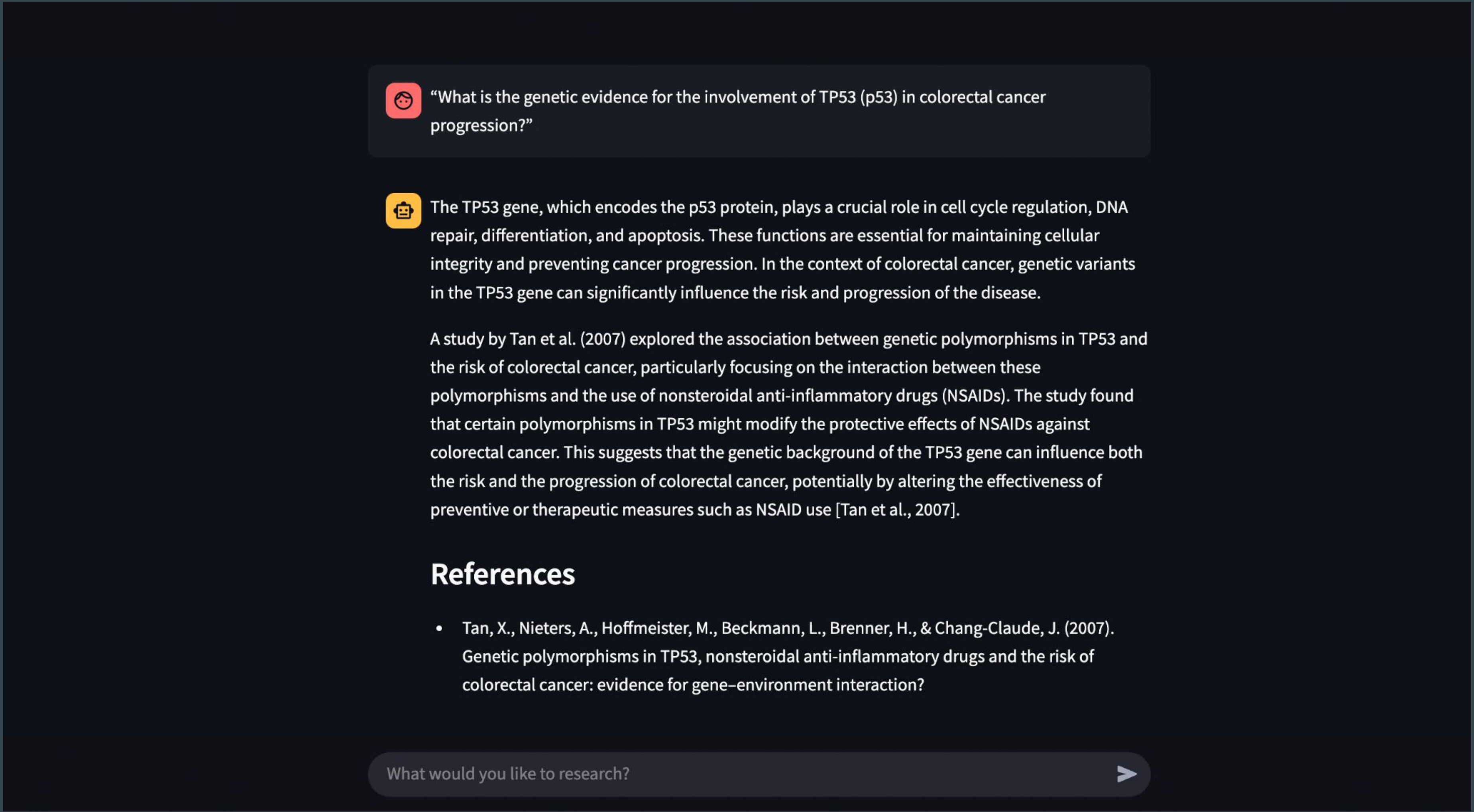
1. **Agent Layer**
   1. Implements a LangChain-based tool-calling agent with Semantic Scholar integration
   2. Utilizes OpenAIs GPT-4 for natural language understanding and response generation
   3. Maintains conversation memory for contextual awareness
2. **Vector Storage Layer**
   1. ChromaDB implementation for persistent storage of paper embeddings
   2. SentenTransformer embeddings for semantic similarity matching
   3. Document-metadata association for comprehensive paper information retrieval
3. **Interface Layer**
   1. Streamlit-based interactive chat interface
   2. Sidebar display of retrieved papers with expandable details (Optional - not implemented)
   3. Status notifications for context loading events (Optional - not implemented)

Key Features

- **Paper Retrieval and Storage**
  - Semantic Scholar Integration : Fetches relevant papers based on user queries
  - Automatic Paper Extraction : Parses API responses to extract structured paper information
  - Vector DB Storage : Embeds and stores papers for future reference and similarity search
- **Citation Management**
  - Structured Citation Format : Implements [Author, Year] citation style in responses
  - Comprehensive References Section : Automatically generates a formatted bibliography
  - Author Formatting : Handles multiple authors with appropriate “et al.” notation
- **User Interaction**
  - Contextual Follow-up Questions : Enables queries about previously retrieved papers

Key Features

- **Data Management**
  - Duplicate Prevention : Checks for existing papers before adding to the database
  - Metadata Preservation : Stores metadata including authors, year, venue, and citation count
  - Persistent Storage : Maintains paper database across sessions
- **Response Generation**
  - Citation-Enhanced Responses : Integrated paper references directly into generated text
  - Formatted Academic Output : Structures responses with proper academic conventions
  - Research Context Awareness : Tailors responses based on available paper information



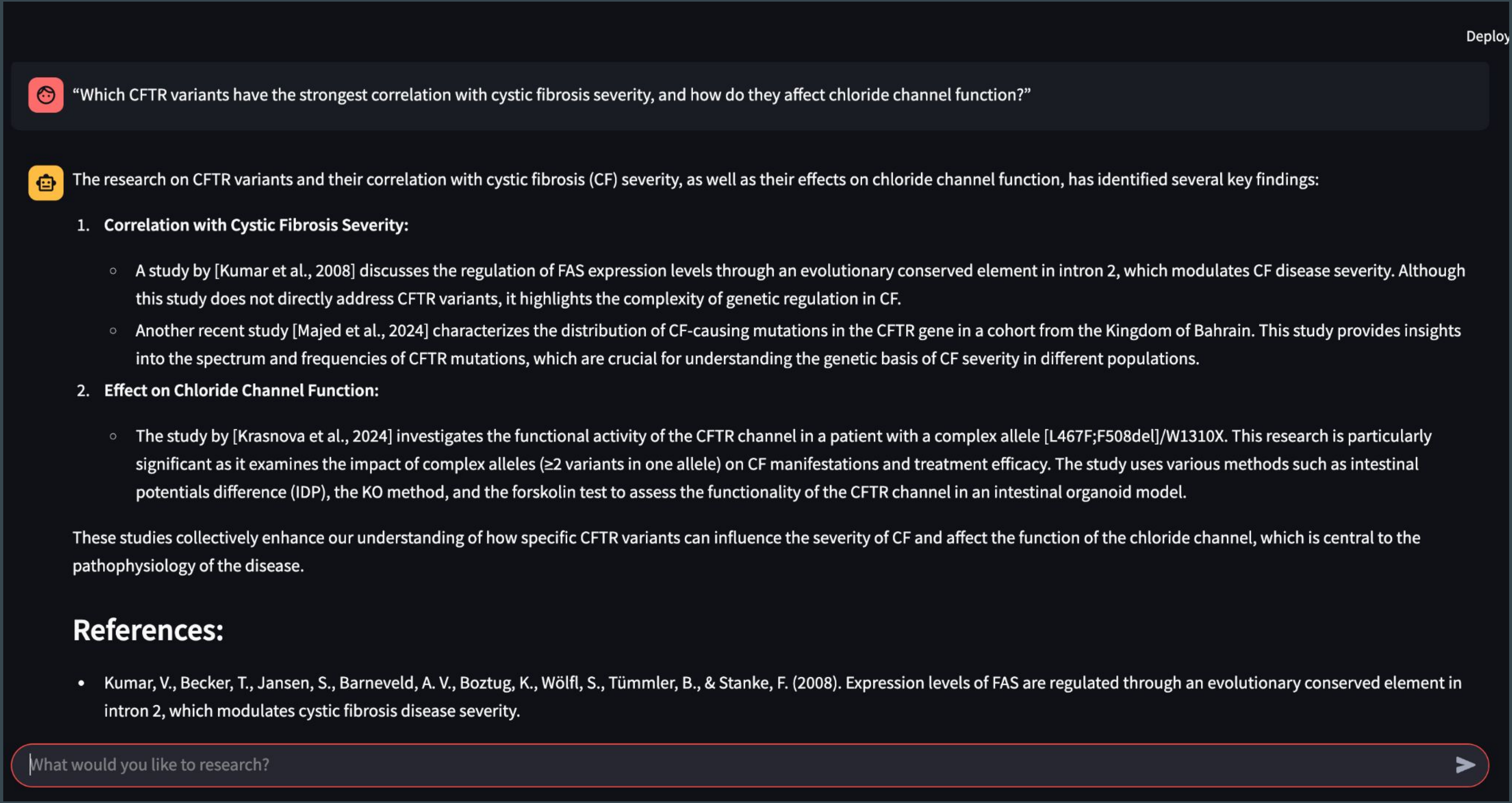
"What is the genetic evidence for the involvement of TP53 (p53) in colorectal cancer progression?"

The TP53 gene, which encodes the p53 protein, plays a crucial role in cell cycle regulation, DNA repair, differentiation, and apoptosis. These functions are essential for maintaining cellular integrity and preventing cancer progression. In the context of colorectal cancer, genetic variants in the TP53 gene can significantly influence the risk and progression of the disease.

A study by Tan et al. (2007) explored the association between genetic polymorphisms in TP53 and the risk of colorectal cancer, particularly focusing on the interaction between these polymorphisms and the use of nonsteroidal anti-inflammatory drugs (NSAIDs). The study found that certain polymorphisms in TP53 might modify the protective effects of NSAIDs against colorectal cancer. This suggests that the genetic background of the TP53 gene can influence both the risk and the progression of colorectal cancer, potentially by altering the effectiveness of preventive or therapeutic measures such as NSAID use [Tan et al., 2007].

References

- Tan, X., Nieters, A., Hoffmeister, M., Beckmann, L., Brenner, H., & Chang-Claude, J. (2007). Genetic polymorphisms in TP53, nonsteroidal anti-inflammatory drugs and the risk of colorectal cancer: evidence for gene–environment interaction?



“Which CFTR variants have the strongest correlation with cystic fibrosis severity, and how do they affect chloride channel function?”

The research on CFTR variants and their correlation with cystic fibrosis (CF) severity, as well as their effects on chloride channel function, has identified several key findings:

1. **Correlation with Cystic Fibrosis Severity:**
    - A study by [Kumar et al., 2008] discusses the regulation of FAS expression levels through an evolutionary conserved element in intron 2, which modulates CF disease severity. Although this study does not directly address CFTR variants, it highlights the complexity of genetic regulation in CF.
    - Another recent study [Majed et al., 2024] characterizes the distribution of CF-causing mutations in the CFTR gene in a cohort from the Kingdom of Bahrain. This study provides insights into the spectrum and frequencies of CFTR mutations, which are crucial for understanding the genetic basis of CF severity in different populations.
2. **Effect on Chloride Channel Function:**
    - The study by [Krasnova et al., 2024] investigates the functional activity of the CFTR channel in a patient with a complex allele [L467F;F508del]/W1310X. This research is particularly significant as it examines the impact of complex alleles (≥2 variants in one allele) on CF manifestations and treatment efficacy. The study uses various methods such as intestinal potentials difference (IDP), the KO method, and the forskolin test to assess the functionality of the CFTR channel in an intestinal organoid model.

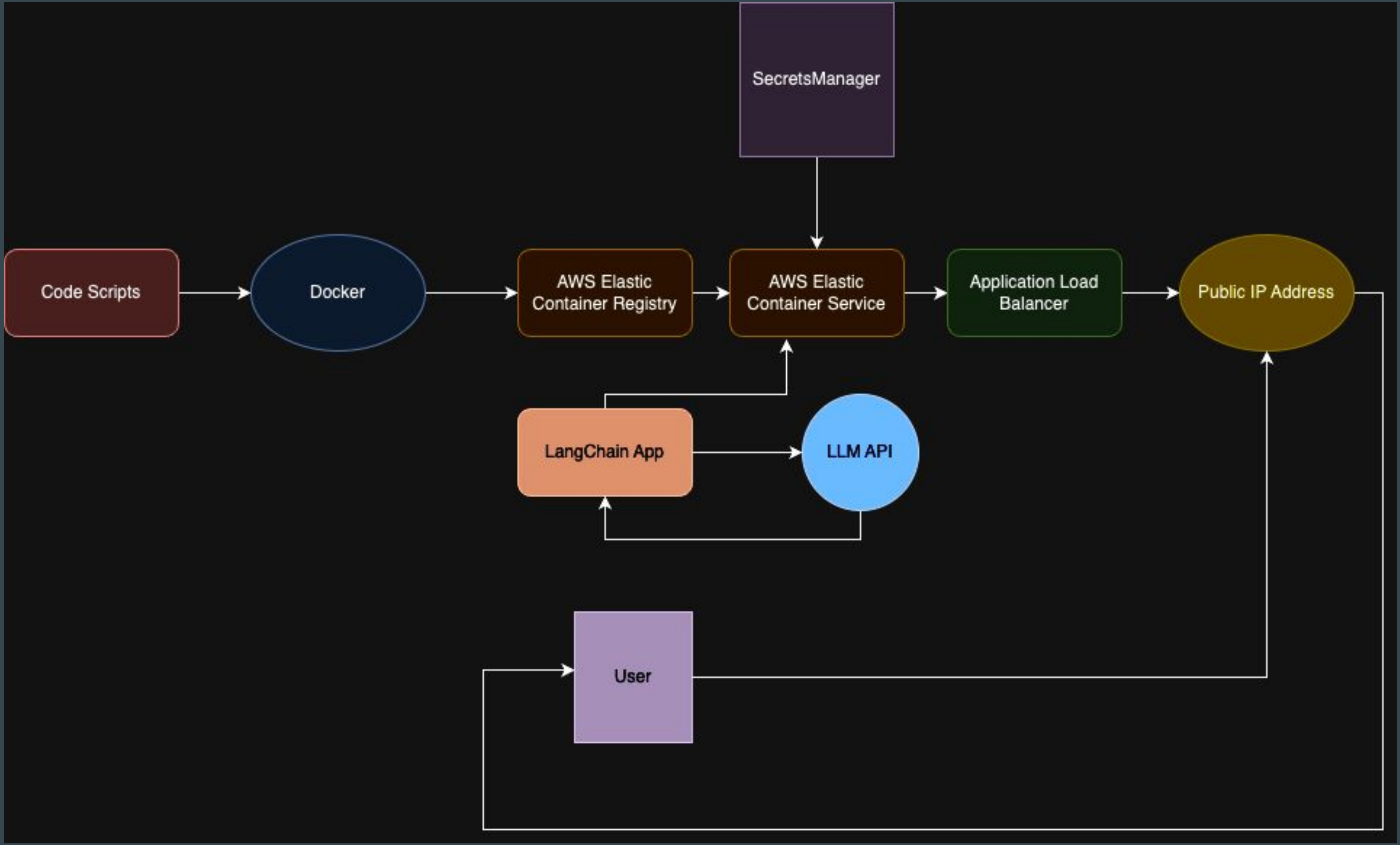
These studies collectively enhance our understanding of how specific CFTR variants can influence the severity of CF and affect the function of the chloride channel, which is central to the pathophysiology of the disease.

References:

- Kumar, V., Becker, T., Jansen, S., Barneveld, A. V., Boztug, K., Wölfl, S., Tümmler, B., & Stanke, F. (2008). Expression levels of FAS are regulated through an evolutionary conserved element in intron 2, which modulates cystic fibrosis disease severity.

Deployment Architecture

(AWS-based High Level)


SecretsManager
Code Scripts
Docker
AWS Elastic Container Registry
AWS Elastic Container Service
Application Load Balancer
Public IP Address
LangChain App
LLM API
User

Future Scope

- A hybrid retrieval system for fetching the relevant content based on user query, where both vector search and keyword-based search can be combined.
- Capture multi-modal information from images and tables.
- Multi-agent system for deeper search and reflection upon collected information.

The END, Thank you!