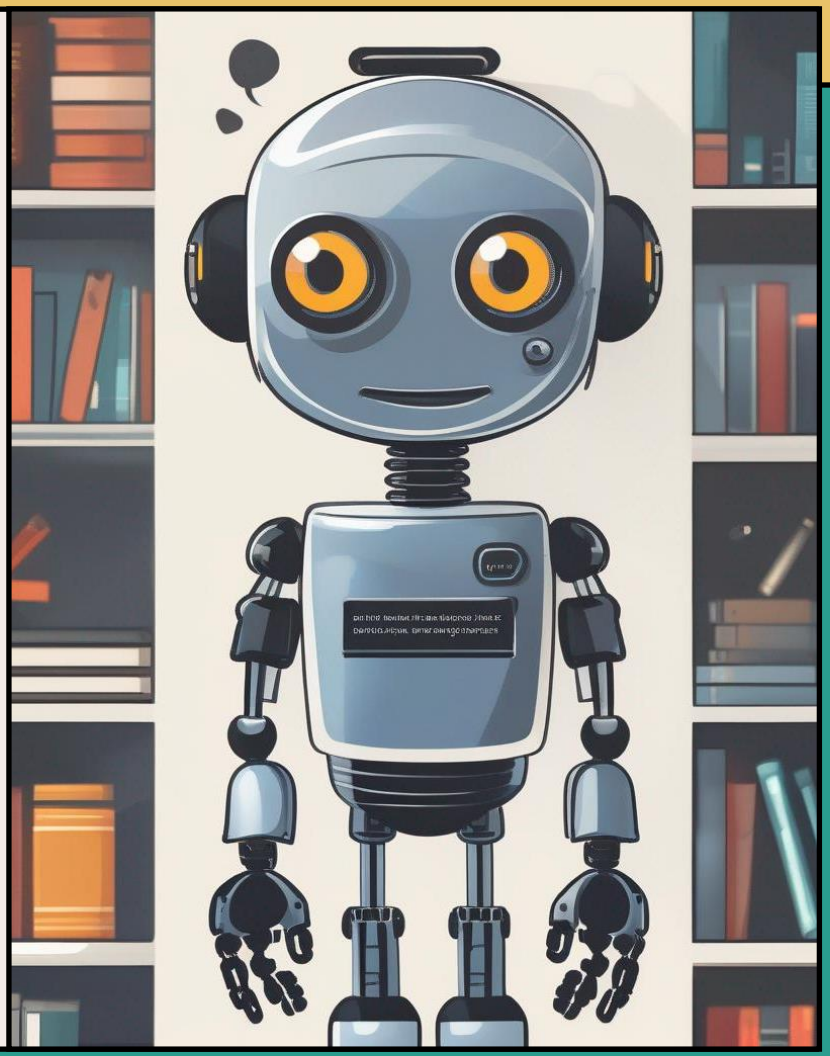
CUSTOM KNOWLEDGE BASE CHATBOT

- B N Swaminathan



Agenda

1. Abstract
2. Introduction
3. Details about the training
4. Project Description
5. Project Design

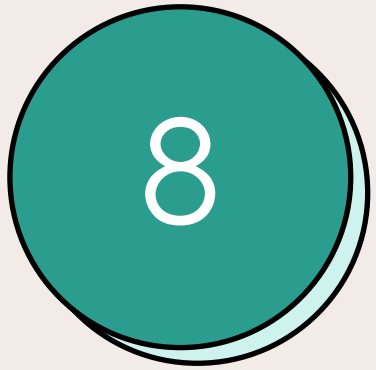
Agenda

Sample output

High level Overview

Low level Overview

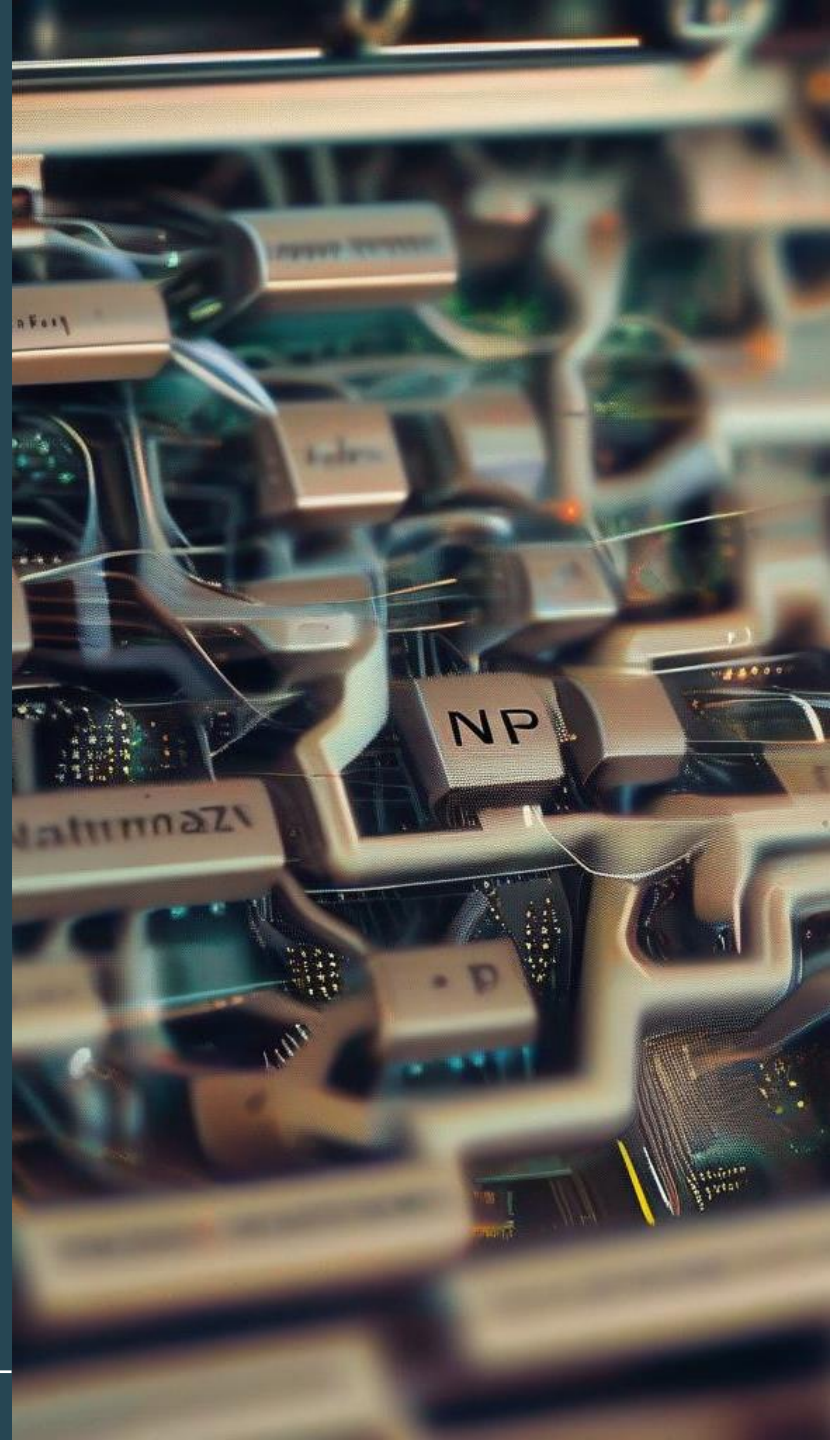
Conclusion



References

Abstract

The Custom Knowledge Base Chatbot project aims to develop an intelligent system that efficiently retrieves and provides accurate answers to user queries from a vast knowledge base. Utilizing advanced natural language processing (NLP) techniques, this chatbot combines state-of-the-art models for question answering (QA) and text generation. By integrating these technologies, the chatbot ensures precise, contextually relevant responses, enhancing user interaction and information accessibility.



Introduction

In the modern digital landscape, quick and accurate access to information is essential. Traditional search methods often fall short in providing immediate, relevant answers. The Custom Knowledge Base Chatbot addresses this gap by leveraging advanced NLP models to understand and respond to user queries naturally. This project integrates robust document retrieval mechanisms, a QA pipeline, and a generative text model to deliver a seamless user experience, capable of handling both straightforward and complex queries.

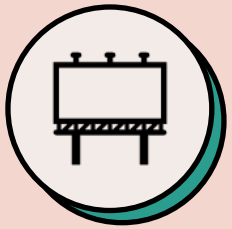
Training details

D4 Insight

Project

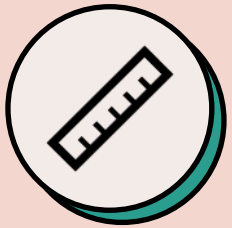
Planning

Define project objectives, scope, and success criteria.



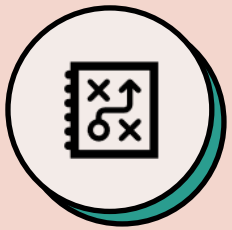
Environmental Setup

Prepare the development environment and install necessary tools.



Design

Outline system architecture and design the chatbot components.



Testing/Enhancements

Conduct tests to ensure functionality and iteratively improve the system.

Deployment

Launch the chatbot application in a production environment.

Additional Learnings

DevOps Basics

- **CI/CD**: Automating build and deploy processes.
- **Version Control**: Managing code changes with Git.

Azure Essentials

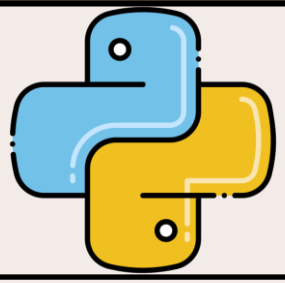
- **Azure Portal**: Managing resources and services.
- **Virtual Machines**: Running applications on Azure VMs.
- **Azure Storage**: Storing data securely in the cloud.
- **App Service:** Deploy, manage and termination of an application.

Project Description

The Custom Knowledge Base Chatbot is designed to provide users with a highly interactive and responsive tool for information retrieval. Key functionalities include:

- **Document Retrieval and Embedding**: Converts documents into embeddings to facilitate efficient searching and retrieval.
- **QA Pipeline**: Extracts accurate answers from retrieved documents using the `roberta-base-squad2` model.
- **Generative Text Pipeline**: Generates detailed responses using the `gpt-neo-2.7B` model for more complex or open-ended queries.
- **User Interface**: Built on Streamlit, providing an intuitive platform for users to interact with the chatbot.

Technology Stack

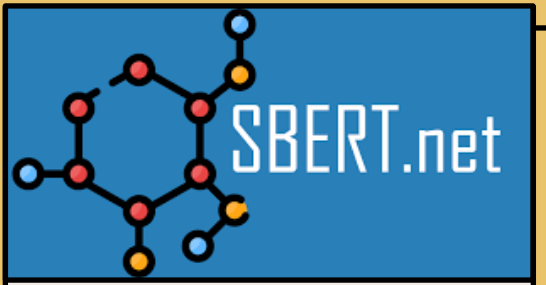


Programming language

Python

Web Framework

Streamlit



NLP Libraries

Sentence Transformers

NLP Libraries

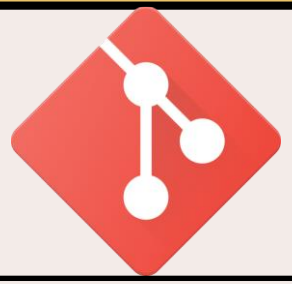
Hugging Face Transformers

Document Processing

LangChain

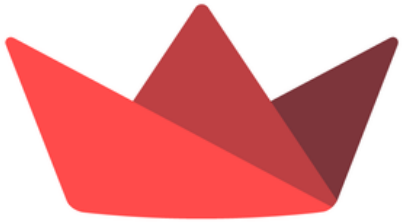
Similarity Search

FAISS



Version Control

Git



Deployment

Streamlit

custom knowledge base chatbot

Project design

Document Processing:

- Documents are ingested from various sources and converted into embeddings using the `SentenceTransformer` model.
- Embeddings are stored in a FAISS index to enable quick and relevant retrieval.

Model Pipelines:

- The QA pipeline, powered by `roberta-base-squad2`, focuses on providing precise answers.
- The generative pipeline, using `gpt-neo-2.7B`, handles more conversational and detailed responses.

Streamlit Interface:

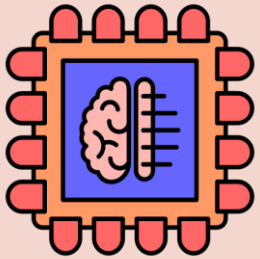
- The front-end interface is built using Streamlit, allowing users to input queries and receive responses in real time.
- The interface supports displaying retrieved documents and model-generated answers, ensuring transparency and user engagement.

User Interface
(Streamlit Frontend)

Model Initialization
(Sentence, QA, Generative)

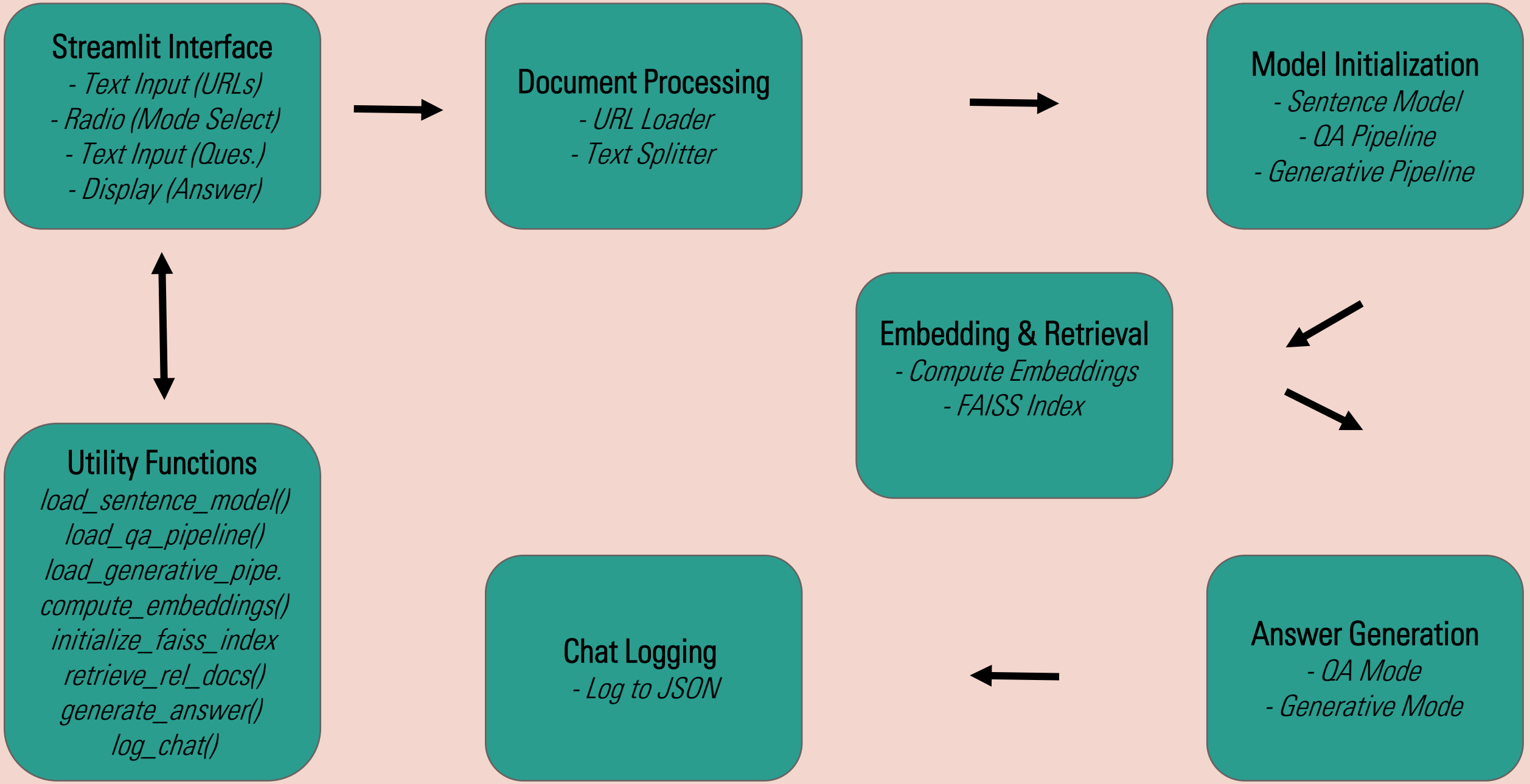
Answer Generation
(QA Mode, Generative Mode)

Document Processing
(URL Loader, Splitter)



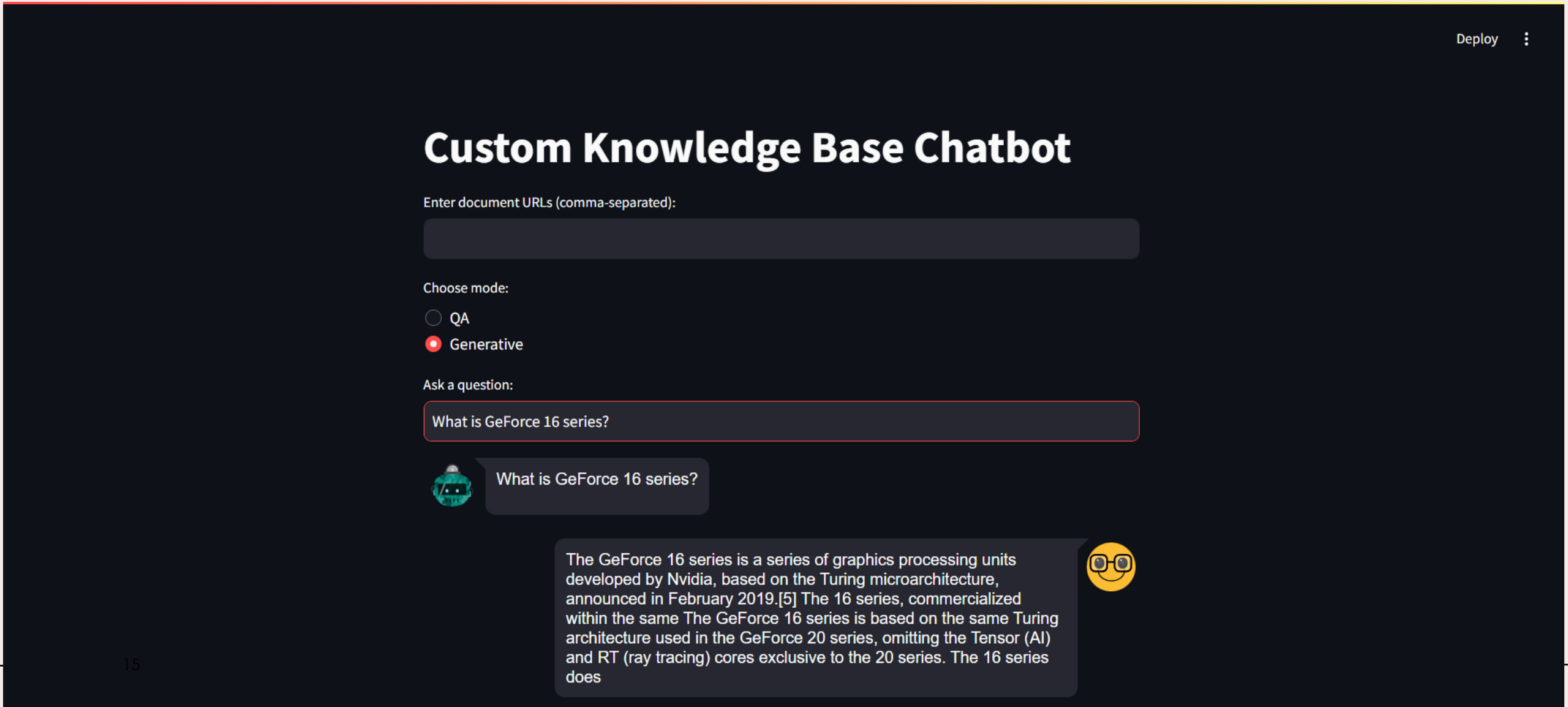
Embedding & Retrieval
(Compute Embeddings, FAISS Index)

Chat Logging
(Log to JSON)


Streamlit Interface
- Text Input (URLs)
- Radio (Mode Select)
- Text Input (Ques.)
- Display (Answer)
Document Processing
- URL Loader
- Text Splitter
Model Initialization
- Sentence Model
- QA Pipeline
- Generative Pipeline
Embedding & Retrieval
- Compute Embeddings
- FAISS Index
Utility Functions
load_sentence_model()
load_qa_pipeline()
load_generative_pipe.
compute_embeddings()
initialize_faiss_index
retrieve_rel_docs()
generate_answer()
log_chat()
Chat Logging
- Log to JSON
Answer Generation
- QA Mode
- Generative Mode

Sample Output: QA mode

Sample Output: Generative mode



Conclusion

The Custom Knowledge Base Chatbot represents a significant advancement in information retrieval and user interaction. By integrating cutting-edge NLP models for both question answering and text generation, the chatbot offers a powerful tool for accessing information quickly and accurately. This project not only enhances user experience but also sets a foundation for continuous improvement and scalability in intelligent information systems.

“

Empowering knowledge through intelligent conversations.

”

References

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. OpenAI Blog.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv preprint arXiv:1908.10084.

Johnson, J., Douze, M., & Jégou, H. (2017). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535-547.

EleutherAI. (2021). GPT-Neo. Retrieved from https://github.com/EleutherAI/gpt-neo

Thank you

B N Swaminathan

Swamibhuvanesan@gmail.com

8825803793

Github