Project Report: Milestone 3 (Implementation of Conversational Agent)

Comprehensive Overview

Milestone 3 focuses on the development and implementation of a Retrieval-Augmented Generation (RAG)-based conversational agent. This chatbot is designed to answer data science-related questions by leveraging a combination of semantic search, machine learning, and large language models (LLMs). The chatbot integrates the LightGBM model trained in Milestone 2 to provide predictive insights and ranks responses based on relevance and quality.

The primary goal of this milestone was to create a functional, interactive chatbot capable of:

- Retrieving relevant documents from a preprocessed dataset.
- Generating contextually accurate responses using an LLM.
- Ranking and filtering responses using a machine learning model.

Implementation Details

Technical Architecture

The chatbot is built using the following components:

1. **Data Retrieval**:
    - **FAISS (Facebook AI Similarity Search)**: Used for efficient similarity search and document retrieval.
    - **HuggingFace Embeddings**: Generates embeddings for semantic search.
2. **Response Generation**:
    - **Google Gemini LLM**: Generates conversational responses based on retrieved documents and user queries.
3. **Answer Ranking**:
    - **LightGBM Model**: Predicts the relevance and quality of responses based on features extracted from the dataset.
4. **User Interface**:
    - **Streamlit**: Provides an interactive web-based interface for user interaction.

Functional Components

1. Data Ingestion

- Preprocessed the Data Science Stack Exchange dataset using the `ingest_data.py` script.
- Created a FAISS index for efficient document retrieval.
- Stored the index in the `data/faiss_index/` directory.

2. Feature Extraction

- Extracted features from text and documents for use in the LightGBM model.
- Features include:
  - `body_length`, `title_length`, `tag_count`, `primary_tag_encoded`.
  - Metadata such as `score`, `view_count`, `comment_count`, `edit_count`.

3. Document Retrieval

- Implemented a hybrid scoring mechanism combining FAISS similarity search and LightGBM predictions.
- Filtered out low-quality documents (e.g., relevance score < 0.5).

4. Answer Ranking

- Ranked answers using the LightGBM model based on extracted features.
- Batch processing was implemented for performance optimization.

5. Conversational Workflow

- Reformulated user queries using a history-aware retriever.
- Generated responses using the Google Gemini LLM.
- Suggested related questions based on user input.

Chatbot Explanation

Workflow Processes

1. **User Input**:
   - The user enters a question via the Streamlit interface.
   - The chatbot reformulates the query if necessary, using chat history for context.
2. **Document Retrieval**:
   - The query is passed to the FAISS vector store to retrieve relevant documents.

- Retrieved documents are scored using the LightGBM model.
- Low-quality documents are filtered out.

3. **Response Generation**:
   - The Google Gemini LLM generates a response based on the top-ranked documents.
4. **Answer Ranking**:
   - Responses are ranked using the LightGBM model.
   - The chatbot provides the most relevant response to the user.
5. **Related Questions**:
   - The chatbot suggests related questions based on the user's query.

Key Algorithms and Methodologies

1. **FAISS Similarity Search**:
   - Used for efficient retrieval of semantically similar documents.
   - Embeddings are generated using HuggingFace models.
2. **LightGBM Model**:
   - Trained in Milestone 2 to predict the likelihood of a post being answered.
   - Features include post-specific, user-specific, and aggregated metrics.
3. **Google Gemini LLM**:
   - Generates human-like responses based on retrieved context.
   - Ensures responses are concise and contextually relevant.

Design Decisions

1. **Why FAISS?**
   - FAISS provides fast and scalable similarity search, making it ideal for large datasets like the Data Science Stack Exchange.
2. **Why LightGBM?**
   - LightGBM was chosen for its high performance, scalability, and ability to rank responses effectively.
3. **Why Google Gemini LLM?**

- The LLM ensures high-quality, context-aware responses, enhancing the chatbot's conversational capabilities.

4. **Why Streamlit?**
   - Streamlit offers a simple yet powerful interface for deploying interactive web applications.

How the Chatbot Works

Technical Architecture

1. **Data Flow**:
   - User input → Query reformulation → Document retrieval → Response generation → Answer ranking → User output.
2. **Components**:
   - **Frontend**: Streamlit interface for user interaction.
   - **Backend**: FAISS for retrieval, LightGBM for ranking, and Google Gemini LLM for response generation.
3. **Integration**:
   - The chatbot integrates all components seamlessly to provide a cohesive user experience.

Code References

1. **Data Ingestion**:
   - `ingest_data.py` : Preprocesses the dataset and creates the FAISS index.
2. **Chatbot Logic**:
   - `chatbot.py` : Implements the chatbot's functionality, including document retrieval, response generation, and answer ranking.
3. **Model Integration**:
   - `lgbm_model.pkl` : Pre-trained LightGBM model used for ranking and filtering.

Quality Standards

1. **Thorough Documentation**:

- All components are well-documented to ensure clarity and reproducibility.

2. **Technical Accuracy**:
   - The chatbot's functionality has been rigorously tested to ensure reliability.
3. **User-Centric Design**:
   - The chatbot is designed to provide accurate, concise, and contextually relevant responses.

Next Steps (Future Enhancements)

1. **Deployment**:
   - Deploy the chatbot for public access.
2. **Enhancements**:
   - Fine-tune the FAISS index and LLM for improved performance.
   - Incorporate additional datasets to expand the chatbot's knowledge base.
3. **User Testing**:
   - Conduct user testing to gather feedback and improve the chatbot's functionality.

Conclusion

Milestone 3 successfully implemented a functional, interactive chatbot capable of answering data science-related questions. By integrating FAISS, LightGBM, and Google Gemini LLM, the chatbot provides accurate and contextually relevant responses. This milestone lays the foundation for future enhancements and deployment, ensuring the chatbot's scalability and usability.