Israel-Hamas War QA System

Introduction:

The task is to create a rudimentary Question Answering (QA) system that can answer questions related to the Israel-Hamas war using a dataset containing 37,000 news articles. The dataset, which spans from October 2023 to March 2024, includes both relevant and spurious articles. The aim is to build a system that filters relevant articles, processes them, and utilizes a language model to generate accurate and relevant answers to user queries.

Methodology:

1. Environment Setup:

- Install the necessary Python packages. This includes packages for handling large datasets, embedding text, and using language models. The `requirements.txt` file is created to list all required packages for easy installation.

2. Data Preprocessing:

- The preprocessing phase involves loading the JSON dataset, cleaning the text data, and filtering articles relevant to the Israel-Hamas conflict.
- As it has web scrapped data which has noise such as punctuation, special characters, and extra whitespaces is removed. The text is also converted to lowercase to ensure uniformity.
- Then the dataset was filtered based on keywords that are related to Israel Hamas war. These keywords are the most searched on Google. By this dataset size was reduced to 36318.

3. Text Chunking:

- To manage the large amount of text, it is split into smaller chunks. These chunks are then embedded using a pre-trained model to facilitate efficient similarity searches.
- A function is implemented to split the text into chunks of size 800 words where 200 words are overlapped to ensure continuity of text.

4. Model building:

- As i have implemented two methods one is using the vector database AstraDB and the other is using a LLamaIndex-based vector store. For both methods, LLaMA2 is used.

a. Using AstraDB:

i. Embedding:

- The chunks are embedded using the `sentence-transformers/all-MiniLM-L6-v2` model from HuggingFaceEmbeddings module.
- An AstraDB vector store is created to hold the embeddings of the text chunks.

ii. Model:

- A vector store index wrapper and a language model are used to build the QA system. The system processes user queries to generate answers.
- A pre-trained LLaMA2 with a 7B parameter model is used to generate answers based on the retrieved text chunks.

b. Using LLamaIndex:

i. Embedding:

- The chunks are embedded using the `sentence-transformers/all-MiniLM-L6-v2` model from HuggingFaceEmbeddings module.
- The service context is created with the specified chunk size of 800 and the initialized language model and embedding model. This context is used to manage and configure the QA system.
- For this, I stored all relevant text into a .txt file and then it will read by using the `SimpleDirectoryReader`.
- A vector store index is created from the loaded documents using the specified service context. This index allows efficient retrieval of relevant text chunks for answering user queries.
- A query engine is created from the index to handle and process user queries. This engine uses a retriever to fetch the most relevant text chunks and a postprocessor to ensure the quality of the retrieved results.

ii. Model:

- A vector store index and a language model are used to build the QA system. The system processes user queries to generate answers by leveraging a pre-trained `LLaMA-2` model with 7 billion parameters, fine-tuned for chat applications. The

`HuggingFaceLLM` class is utilized to load and configure the language model.

5. **Relevance:**
    - Implemented with the help of cosine similarity metrics. To find the most relevant context. This context along with the question will be given to the model to generate an answer.