This code implements a search interface for answering questions based on documents using Elasticsearch and a pre-trained model from Hugging Face.

NOTE:

Docker Setup

For Docker setup, Dockerfiles and Docker Compose have been provided for all services needed. However, note that the Torch dependency required to use SentenceTransformers might be too heavy and time-consuming to download in the base image. As a result, I did not test virtualization on my local machine. However, the provided Dockerfiles and Docker Compose configuration are expected to work.

Quick Test

For a quick test:

1. Use the run_project.sh file to install and launch the project (both backend and frontend servers).
2. Follow the steps outlined in the run_project.sh file to install and launch the project.

Key Steps:

1. **Data Preprocessing**: Documents undergo preprocessing, including stop word removal, to focus on content words.
2. **Document Indexing**: Documents are indexed into Elasticsearch. Each document includes its title, description, and cleaned paragraph (after stop word removal). The paragraph is also encoded into a dense vector using a pre-trained SentenceTransformer model.
3. **Search Functionality**:
   - **Semantic Search**: Cosine similarity between the query embedding and document embeddings in Elasticsearch is computed to find relevant documents.

- **Keyword Search**: Elasticsearch performs a keyword-based search, considering the relevance of keywords in the title, description, and paragraph fields.

4. **Result Filtering**: Search results are filtered based on a relevance score threshold. Duplicate documents are removed based on their link.

Challenges Faced:

- **Selecting the Best Pre-trained Model**: Identifying the optimal pre-trained model for semantic search involved experimentation and evaluation to ensure effective embedding generation.

- **Evaluating Stop Words Removal**: Assessing the impact of stop word removal on search accuracy required testing the model with and without stop words before embedding to determine the most effective approach.

- **Weighting Document Parts**: Assigning appropriate weights to document parts (e.g., title, description, paragraph) required experimentation to prioritize certain sections for better relevance ranking.

Advantages of Challenges Addressed:

- **Optimized Model Selection**: By selecting the best pre-trained model, the search system can achieve higher accuracy and relevance in results.

- **Improved Search Accuracy**: Evaluating stop word removal and weighting document parts enhances the system's ability to retrieve relevant documents and improve overall search accuracy.

Embedding in Semantic Search: Embedding refers to representing text as dense vectors in a continuous vector space. The SentenceTransformer model generates embeddings for both the query and documents, enabling efficient semantic search by comparing these embeddings.

This approach combines keyword-based and semantic-based search techniques to provide accurate and relevant search results while addressing key challenges to enhance search accuracy and effectiveness.