Step-by-Step Guide for Quran and Hadith QA System

Step 1: Scrape Data

1. Select Data Sources:

   - Identify reliable websites for Quran and Hadith (e.g., Quran.com, sunnah.com).

2. Scrape Content:

   - Use Scrapy (or similar tools) to extract relevant content.

   - Store the Quran and Hadith content in structured formats (e.g., JSON, CSV).

   - Ensure data is divided into meaningful chunks (e.g., verses for Quran, individual Hadith).

Step 2: Preprocess Data

1. Tokenize Text:

   - Clean the text (remove HTML tags, special characters).

   - Tokenize into sentences or paragraphs.

2. Organize for QA:

   - Structure the data as a collection of passages with unique IDs for reference.

Step 3: Fine-Tune a Pretrained BERT Model

1. Choose a QA Model:

   - Use Hugging Face's transformers library with a pretrained BERT-based QA model.

2. Prepare Dataset:

   - Format data in SQuAD-like format (refer to the guide for JSON structure).

3. Fine-Tune:

   - Use Hugging Face's Trainer API for fine-tuning the BERT model.

Step 4: Build QA Application

1. Inference:

   - Use the fine-tuned BERT model for inference.

2. Integrate with Scraped Data:

   - Load structured Quran and Hadith data.

   - Use the model to answer user queries by selecting the most relevant passage.

Step 5: Optimize Search for Relevant Passages

1. Retrieve Context Efficiently:

   - Use an information retrieval system (e.g., Elasticsearch or faiss).

2. Example with Sentence Transformers provided.

Step 6: Deploy the System

1. Frontend:

   - Build a simple web or chatbot interface to interact with users.

2. Backend:

   - Host your model using a framework like Flask, FastAPI, or Hugging Face Spaces.

3. Example API implementation provided in FastAPI.

Optional Enhancements

1. Add Multilingual Support:

   - Fine-tune a multilingual BERT model if you want to handle Arabic and other languages.

2. Incorporate Summarization:

   - Provide summarized answers for lengthy passages using a summarization model.