RAG in Depth

Advanced techniques to improve Performance

Contents

- Basic RAG
  - Preparation
  - Phase 1: Indexing
  - Phase 2: Retrieval
  - Phase 3: Generation
- Advanced techniques to improve Performance.

Basic RAG: Preparation

- Prepare .env
- Create virtualenv
- Install jupyterlab
- Open notebook
- Install required modules
- Activate LangSmith
- Load modules

Basic RAG, Phase 1: Indexing

- Load the private document.
- Split the document in small chunks of text.
- Convert the small chunks of text in numbers: Embedding.
- Load the embeddings into a vector database.

Basic RAG, Phase 2: Retrieval

- Set the retriever.
- Set the prompt.
- Set the LLM.
- Optional: pre-processing function.
- RAG chain.

Basic RAG, Phase 3: Generation

- Asking questions to our RAG application.

Query translation: intro

- Sometimes the user's question is confusing and our RAG App will not provide a good answer for it.

- We can solve this is by improving the user's question (also called "query"). This is called "query translation", and it can be done in several different ways.

Query translation: alternative techniques

- Technique #1: Convert the user's question into multiple similar questions.
  - Technique #1.1: Multi-Query technique.
  - Technique #1.2: RAG Fusion technique.

- Technique #2: Convert the user's question in (progressive or not) sub questions.

- Technique #3: Convert the user's question into step back questions.

Query translation technique #1 and #2: Similar Questions

- One way to solve the issue of a confusing user's question is to make our RAG App similar questions and ask it to produce one consolidate answer. This is called the Multi-Query technique. See coding example and the corresponding ops in LangSmith.

- When we are applying the Multi-Query technique, sometimes it will be useful to rank the similar questions we create and give them different levels of importance. This is called the RAG Fusion technique. See coding example and the corresponding ops in LangSmith.

Query translation technique #3: Sub Questions

- Another way to solve the issue of a confusing user's question is to decompose the question in several sub questions. This technique is called Decomposition.
- Sometimes we will use independent subquestions. See coding example and the corresponding ops in LangSmith.
- And sometimes we will use dependent subquestions: in order to respond the subquestion #2, our RAG App needs to know the response to subquestion #1. See coding example and the corresponding ops in LangSmith.

Query translation technique #4: Step Back Questions

- A fourth way to solve the issue of a confusing user's question is to make our RAG App step back questions and ask it to produce one consolidate answer. This is called the Step Back technique. See coding example and the corresponding ops in LangSmith.

Query translation technique #5: HyDE

- Given the user's question, create a fake document that would answer it properly.
- Then go to the vector database and find documents similar to the fake document.

Routing

Purpose: To direct the question to the appropriate data source (such as a vector database, relational database, or graph database) or the appropriate prompt or any other routing option.

Types of Routing:

- Logical Routing: Uses the LLM's knowledge of data sources to decide the best destination for a query.
- Semantic Routing: Involves embedding the question and prompts, calculating similarity, and selecting the prompt with the highest similarity for routing. For example: based on the question, use the Math prompt or the Physics prompt.

Query Structuring

- Transform the query from natural language to some query syntax.
- In the example, we transform the query from natural language to the query syntax according to the schema we have defined.