**Data Pre-processing Pipeline**

A diagram of a computer process

AI-generated content may be incorrect.

*GitHub Reference: Dify-KB-Pipeline*

Proposed data pre-processor pipeline, as it covers both local information and general information.

* Localize Information (Individual data)
  + Row data
* Generalize Information (Metadata)
  + Summary
  + Overview
  + Relationships of data
  + Distributions of data
  + Metadata
  + Statistics of data
  + Trend of Data

**How to Retrieve Related Chunk Data that was not Retrieved by Traditional RAG Method?**

The traditional RAG Method uses a knowledgebase that has been chunked into multiple parts. Each chunk has its own vectors values based on the embedding model that has been used. When a user entered a prompt, the prompt is then compared to the vector values that represent the closest semantic similarity to the chunks of knowledgebase.

A top K (K means the number of tops that has exceeded the acceptance threshold) result chunks are retrieved, depending on the number K, the top chunk/s will then be used as the context for LLM to answer the user query.

In reality, these chunks are not really the context that the LLM really needs to answer the user query, the retrieved chunks are semantically similar to the user query, but not contextually that in which the LLM really needs.

Semantic similarity is useful when the user query requires only localized information. However, when the query requires information about data relationships, data count, data distribution, etc. the localized information isn’t enough and increasing the top k result is inefficient as it practically just feeding the LLM the entire file itself.

To entertain the user queries that require more than localized information, there are several methods that can be used:

1. ***Dedicated Knowledgebase for generalized information***
   1. A high-level data that houses information about the knowledgebase.
   2. Information such as:
      1. Data Overview
      2. Data Summary
      3. Data Relationship
      4. Data Distribution
      5. Data Count
      6. Data Statistics
      7. Data Trend

is included to entertain user’s query that requires more information than localize data.

* 1. This approach still runs on a RAG pipeline that retrieves data from high-level knowledgebase.

1. ***Query Routing***
   1. Classifies the user queries on what data it wanted to extract (Analytical/Metadata). And instead of running on a RAG pipeline, it can take a direct approach and take the entire high-level knowledgebase a context and let the LLM generate an appropriate answer to the query.

**How to Handle Compounded Queries (Complex Query)?**

It is a type of query that contains two or more queries in a single query. These queries may contain a requirement to retrieve data from multiple localized information, generalized information, and/or a combination of both.

To entertain these kinds of queries, there are several methods that can be used:

1. Query Decomposition
2. Query Expansion
3. Query Decomposition and Synthesis