What is RAG?

RAG Principle - RAG, or Retrieval Augmented Generation, is a technique that combines the capabilities of a pre-trained large language model with an external data source. RAG operates in two primary phases: retrieval and generation. Initially, it identifies and retrieves pertinent information from an external knowledge base. Subsequently, it synthesizes this data with the LLM's internal knowledge to generate a coherent and accurate response.

Use RAG to Improve LLMs - Generic responses & Lack of specific information (local data training), Hallucinations.

Flow steps -

1) Data collection

Gathering data from sources and passing it to the system.

2) Data chunking

Data chunking is the process of breaking your data down into smaller, more manageable pieces. This way, each chunk of data is focused on a specific topic. When a piece of information is retrieved from the source dataset, it is more likely to be directly applicable to the user’s query, since we avoid including irrelevant information from entire documents.

3) Document embeddings

Now that the source data has been broken down into smaller parts, it needs to be converted into a vector representation. This involves transforming text data into embeddings, which are numeric representations that capture the semantic meaning behind text.

4) User queries

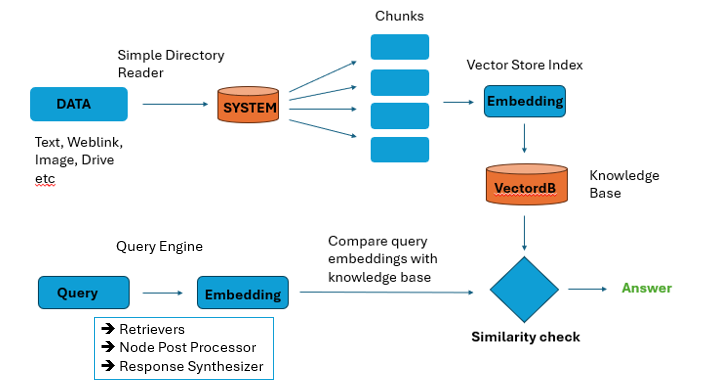
When a user query enters the system, it must also be converted into an embedding or vector representation. Once the query is converted into an embedding, the system compares the query embedding with the document embeddings. It identifies and retrieves chunks whose embeddings are most similar to the query embedding, using measures such as cosine similarity and Euclidean distance.

These chunks are considered to be the most relevant to the user’s query.

5) Generating responses with an LLM

The retrieved text chunks, along with the initial user query, are fed into a language model. The algorithm will use this information to generate a coherent response to the user’s questions through a chat interface.

Diagram of the flow –



Query Engine –

Retrievers are used in query engines to retrieve the most similar responses with respect to your queries.

Node Post Processor is a component responsible for processing the retrieved nodes from the retrieval phase. like filtering out irrelevant or low-confidence nodes based on criteria such as relevance score or confidence level and performs text cleaning on your response, it removes irrelevant characters from response.

Response Synthesizer, also known as the answer synthesizer or answer generator, is responsible for synthesizing the final answer or response based on the processed and refined nodes from the node postprocessor.