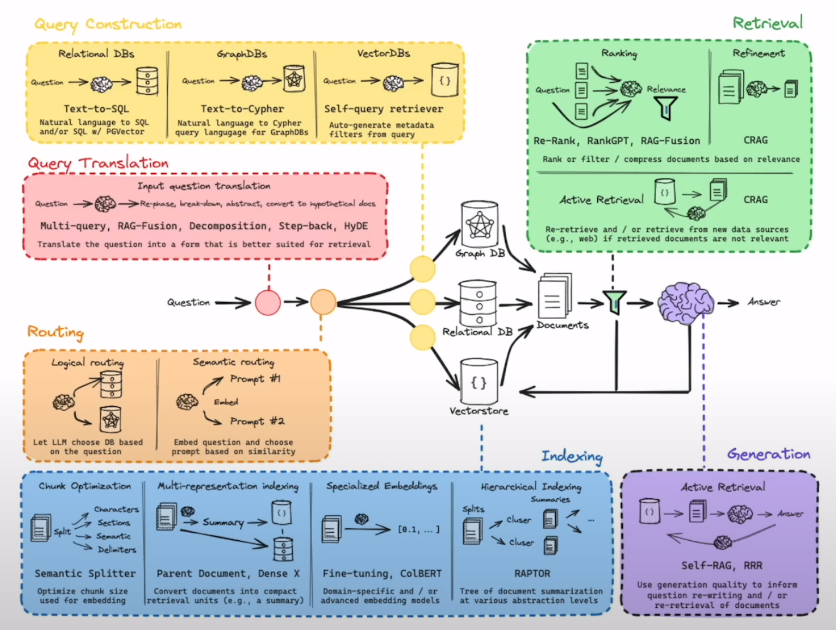
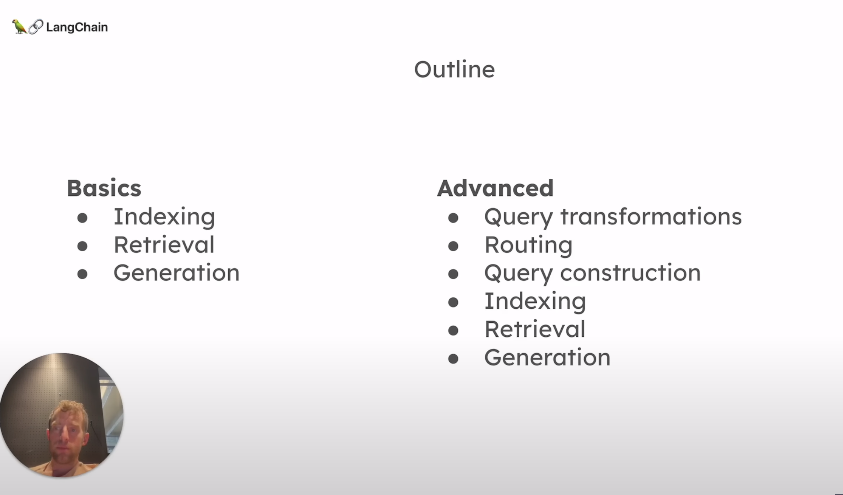
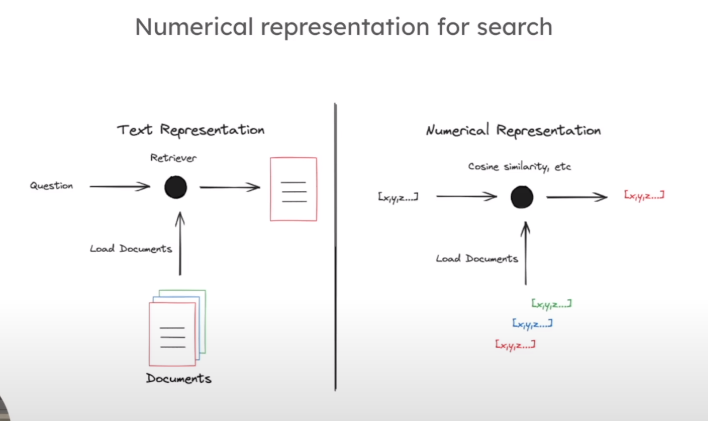
**RAG**

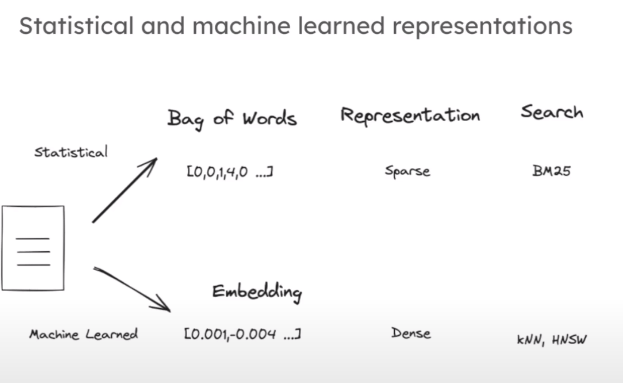
LLMs are trained on a large but fixed corpus of data, limiting their ability to reason about private or recent information. Fine-tuning is one way to mitigate this, but is often [not well-suited for facutal recall](https://www.anyscale.com/blog/fine-tuning-is-for-form-not-facts) and [can be costly](https://www.glean.com/blog/how-to-build-an-ai-assistant-for-the-enterprise). Retrieval augmented generation (RAG) has emerged as a popular and powerful mechanism to expand an LLM's knowledge base, using documents retrieved from an external data source to ground the LLM generation via in-context learning.

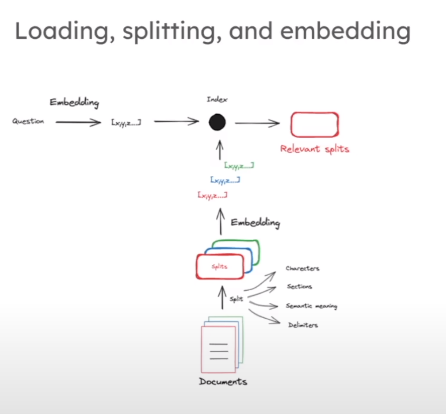




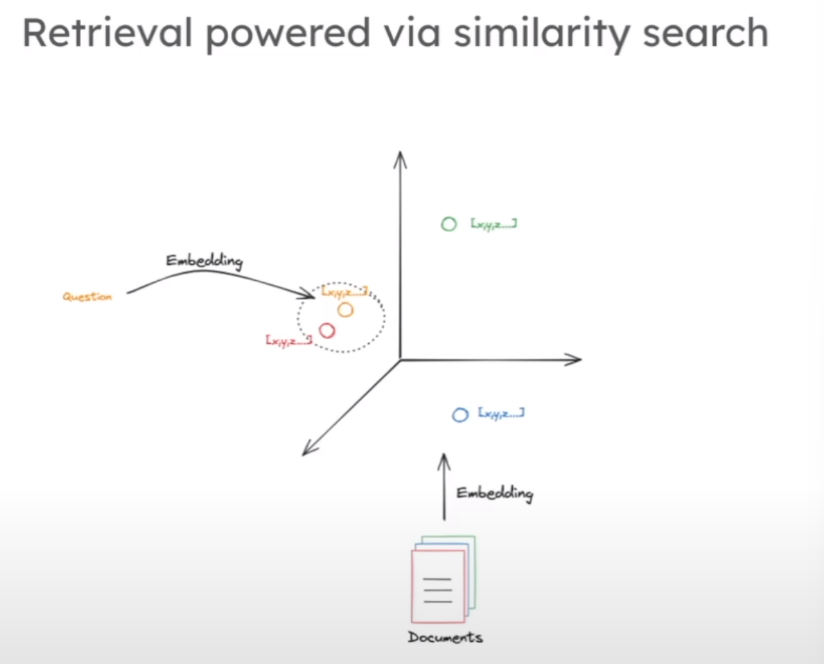
# Indexing:

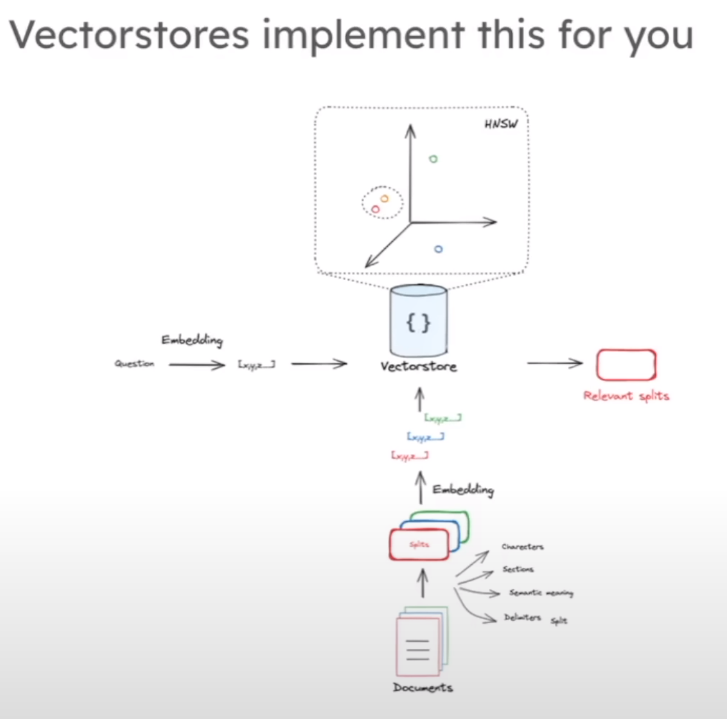


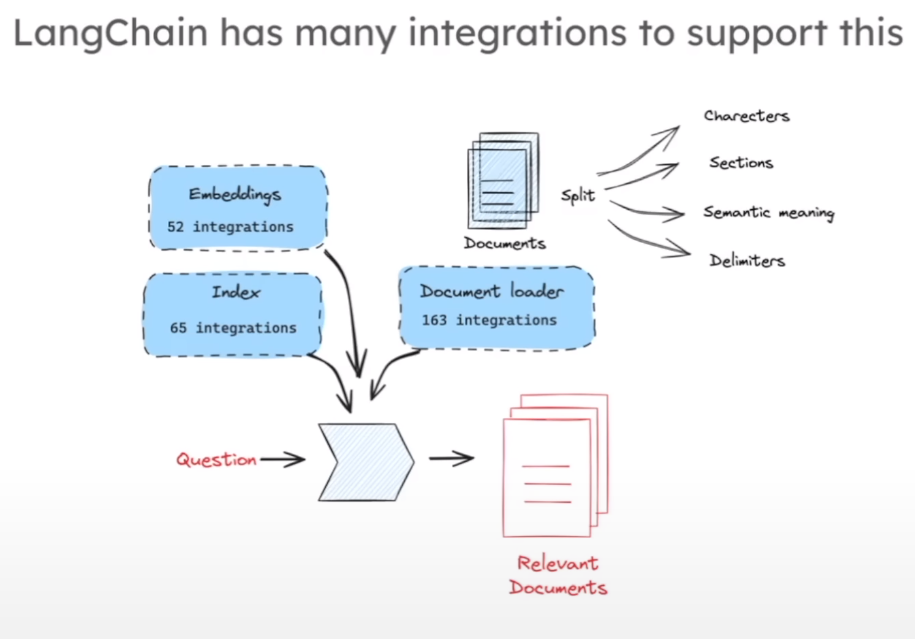




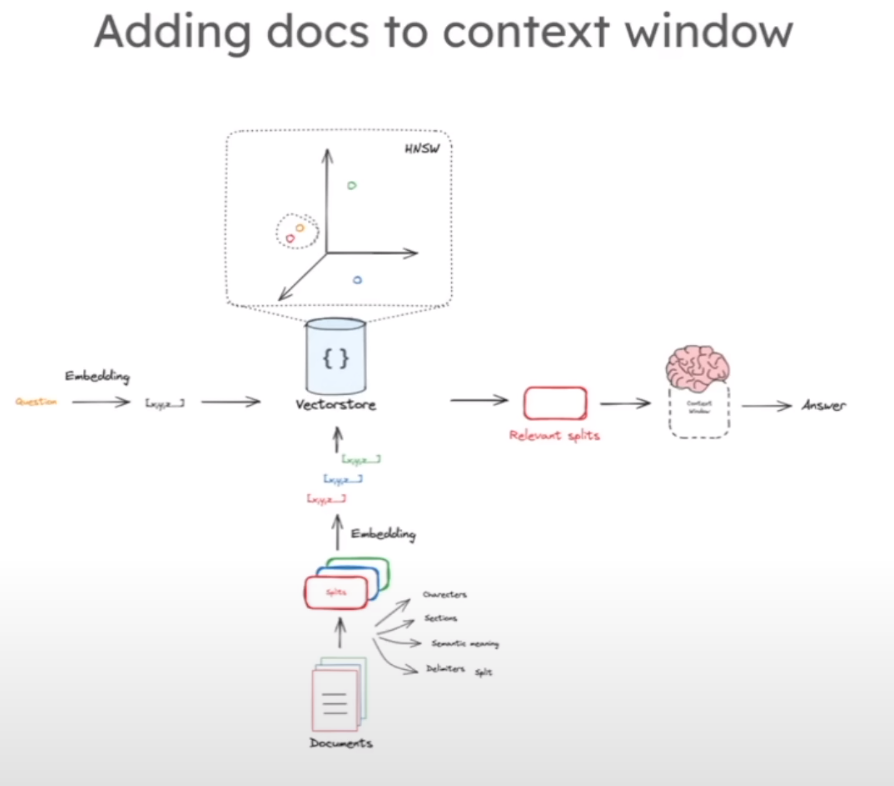
# Retrieval:

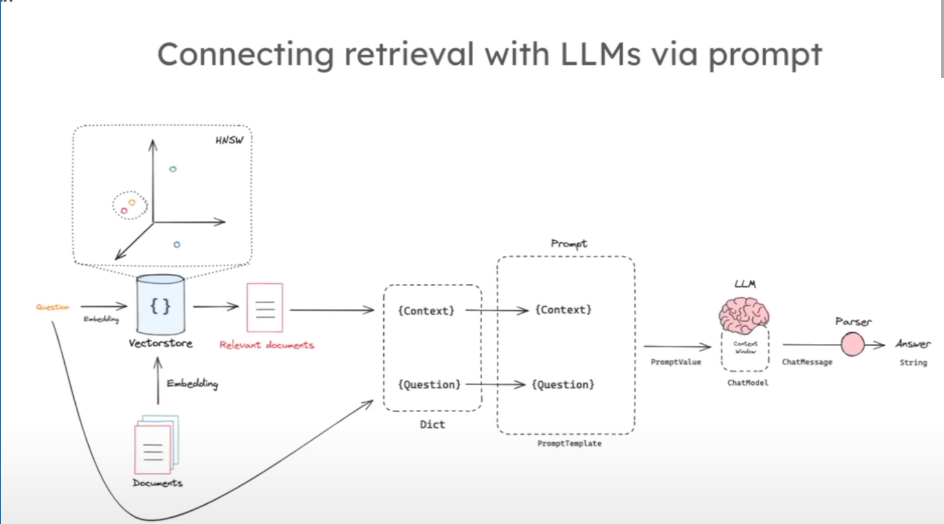






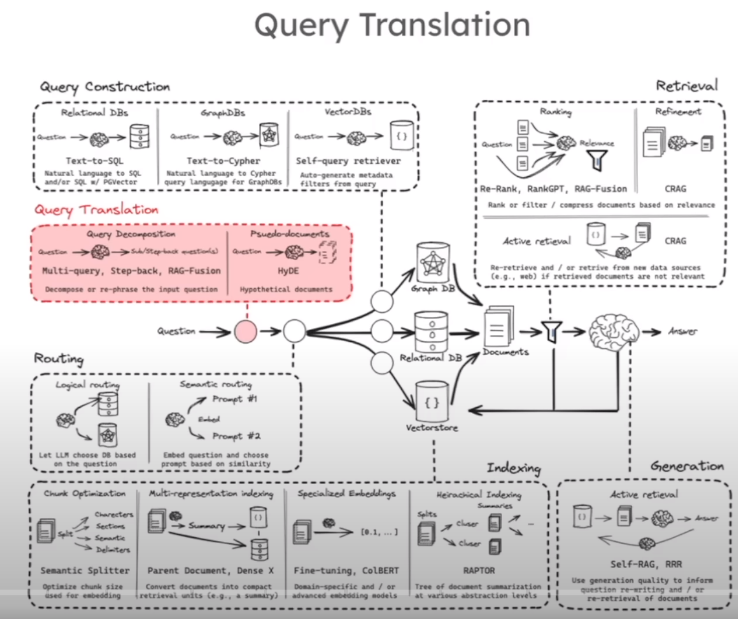
# Generation:

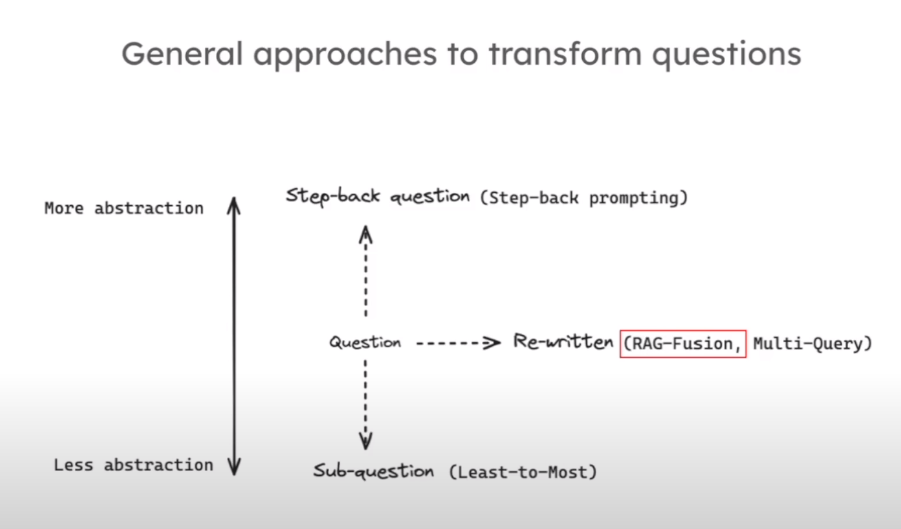




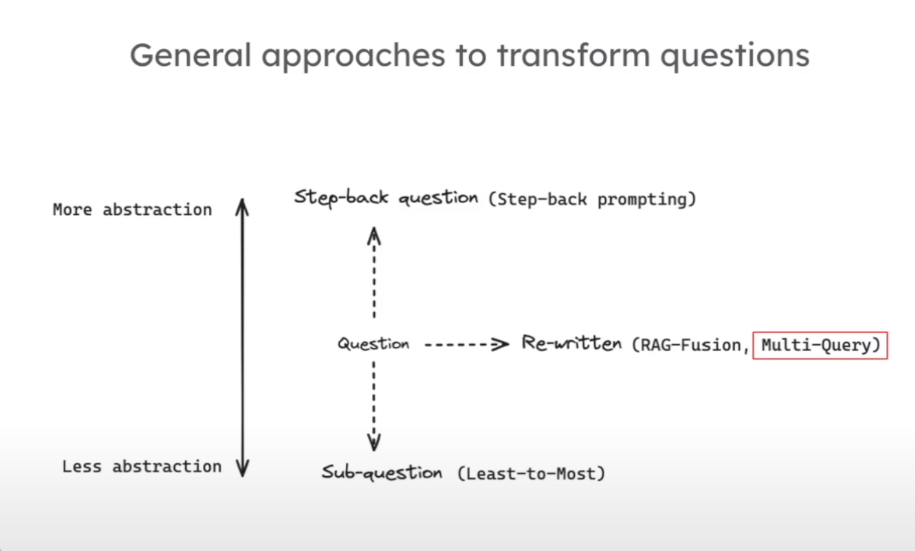
# Query Translation:

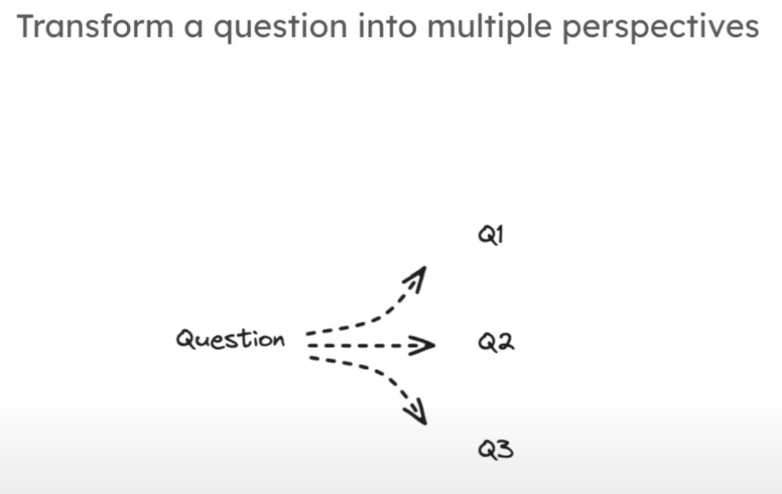


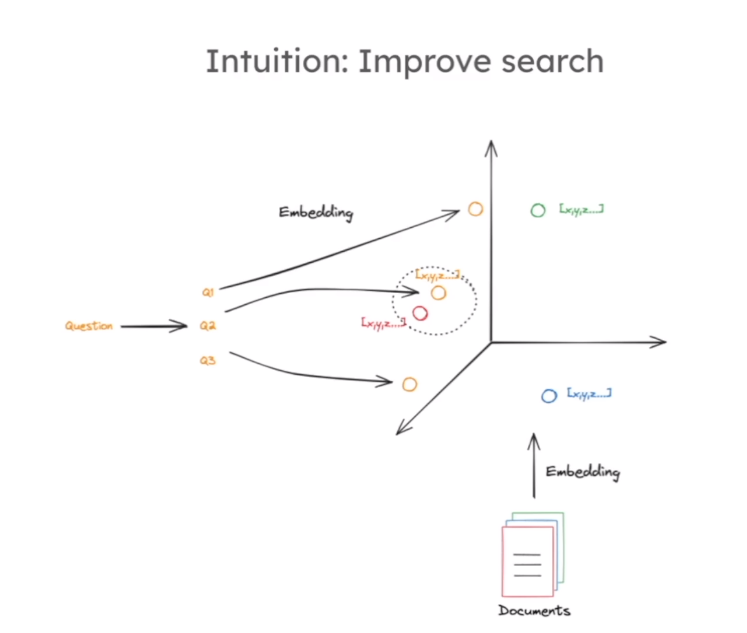




## 1) Multi Query approach:

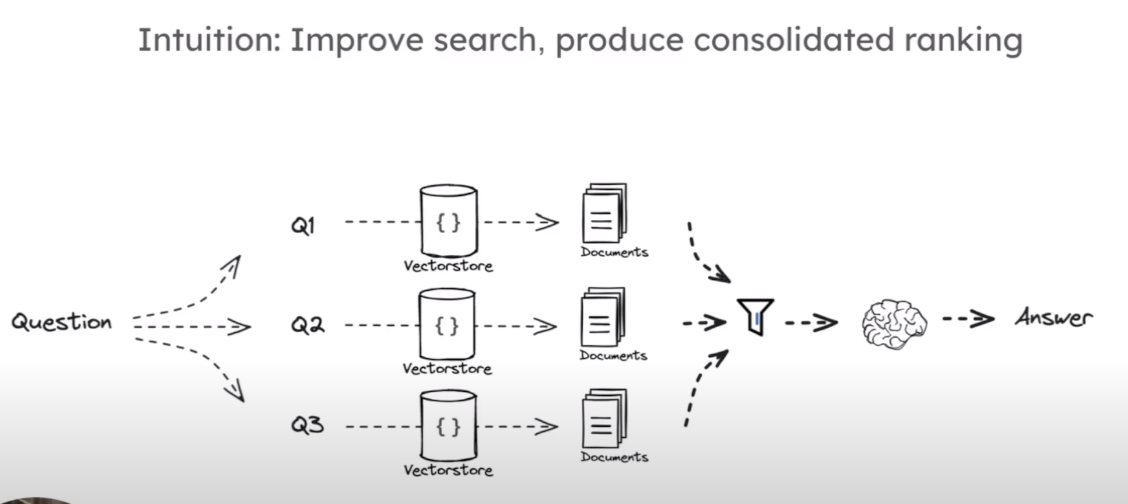




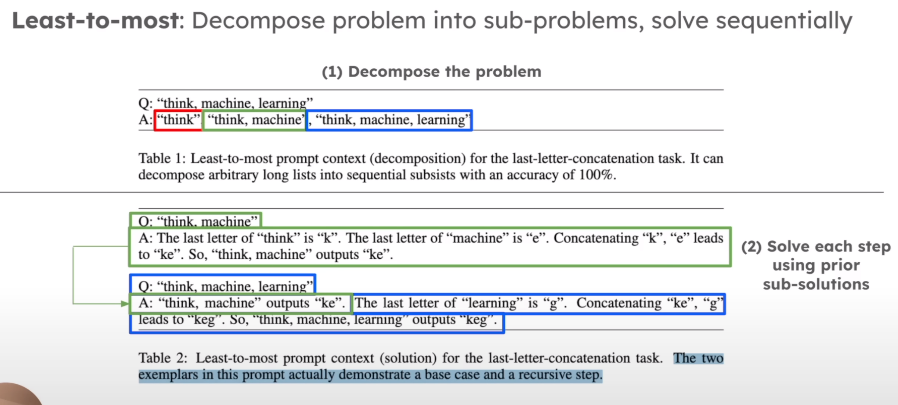


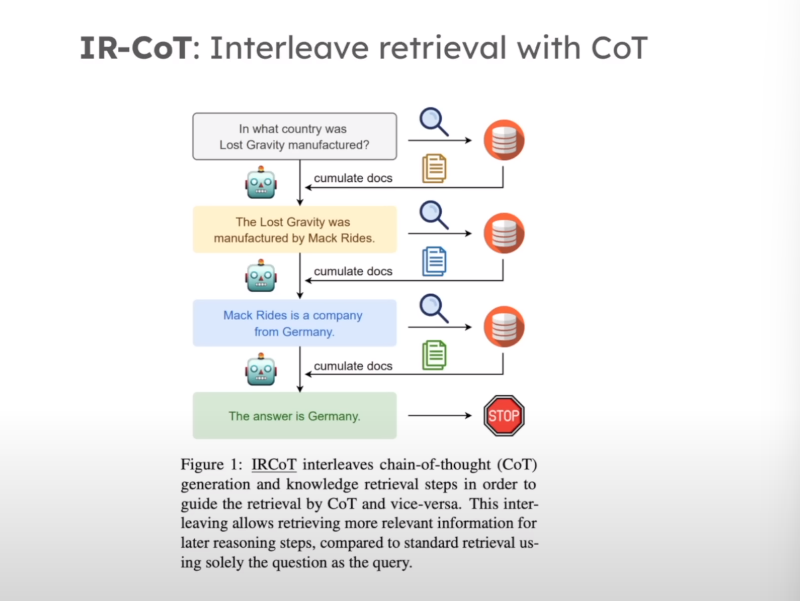
## 2) RAG Fusion:

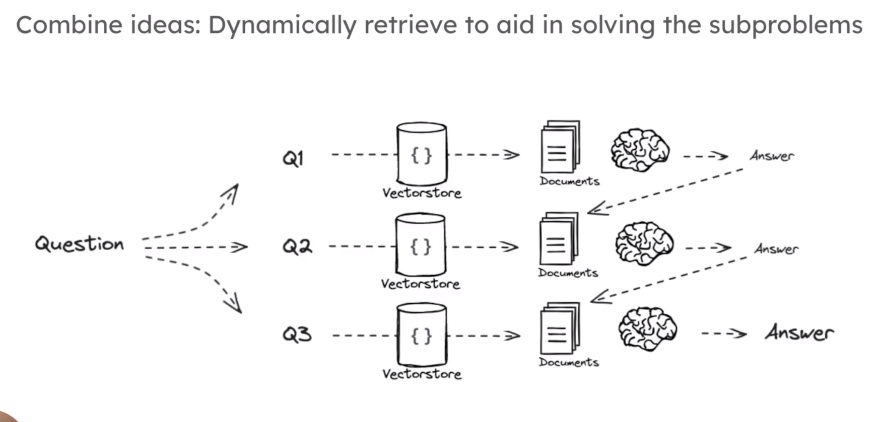
Similar to multi query but we rank different retrievals



## 3) Decomposition:





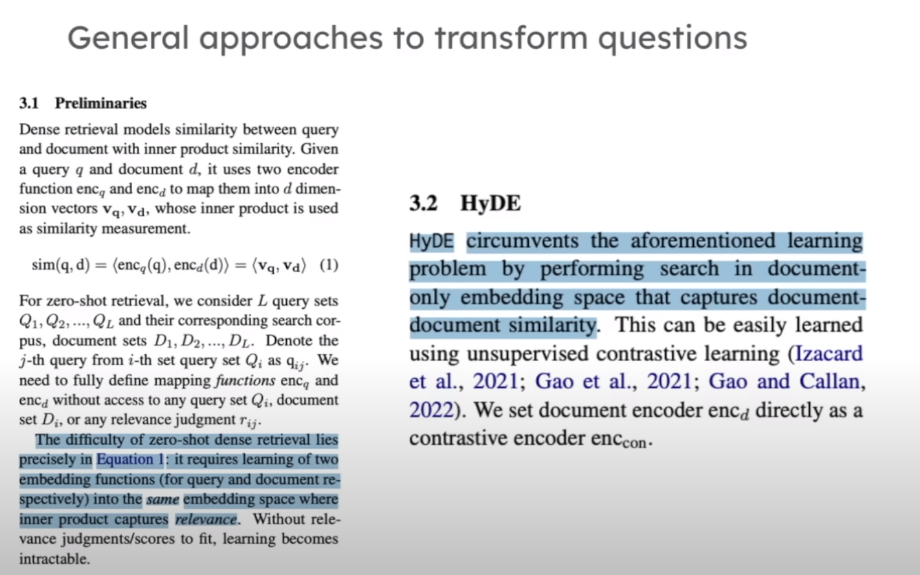


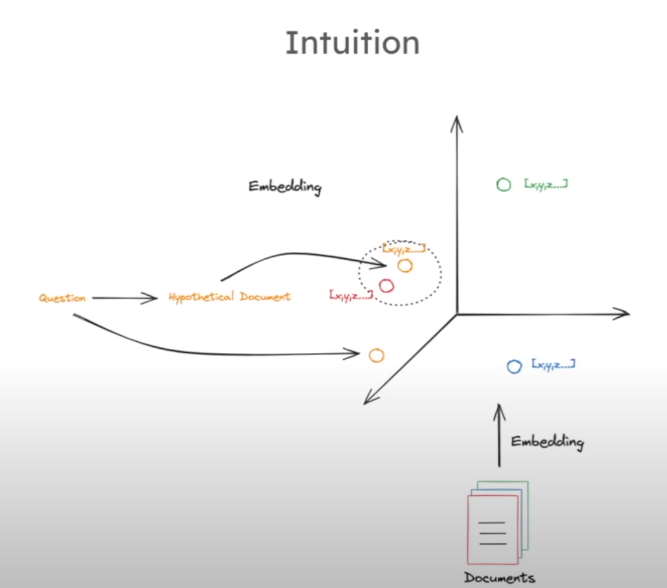
## 4) Step-Back prompting:





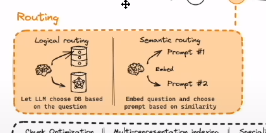
## 5) HyDE:

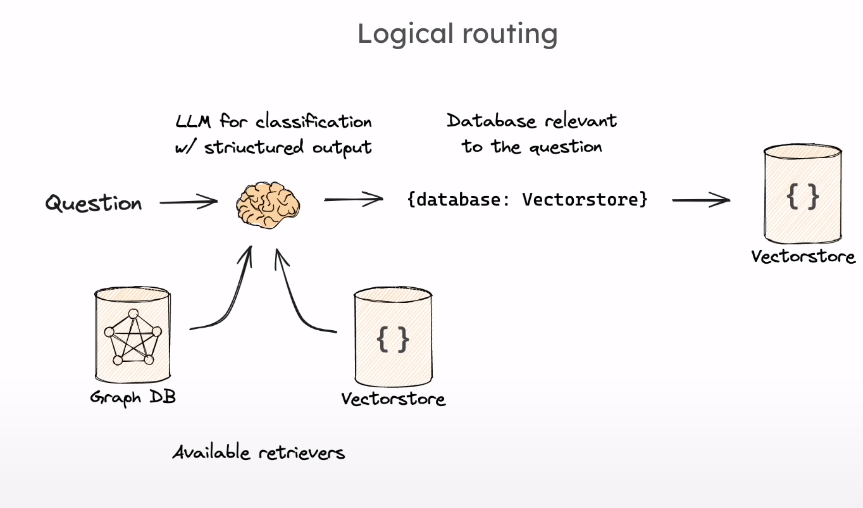


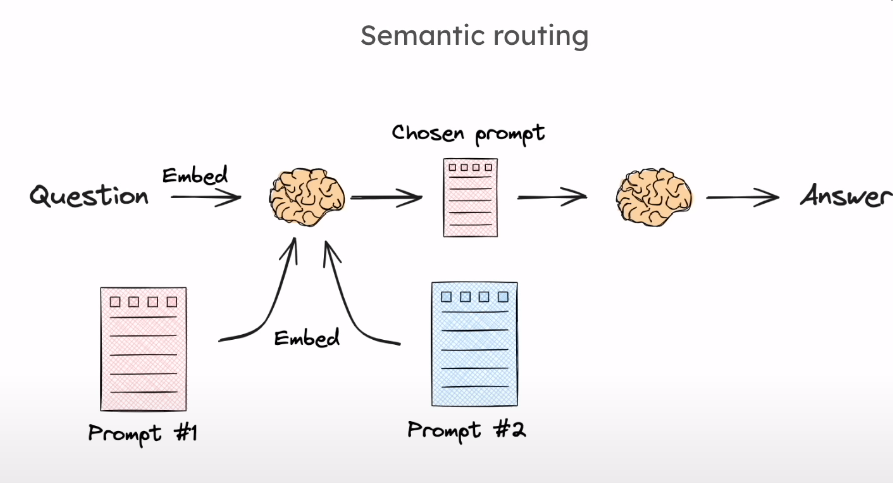


# Routing:

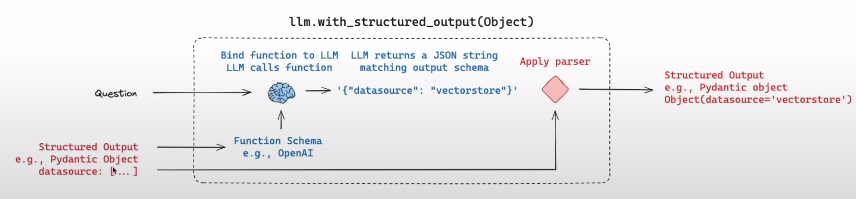
Routing the llm to the correct sources like GraphDB or RelationalDB or VectorstoreDB



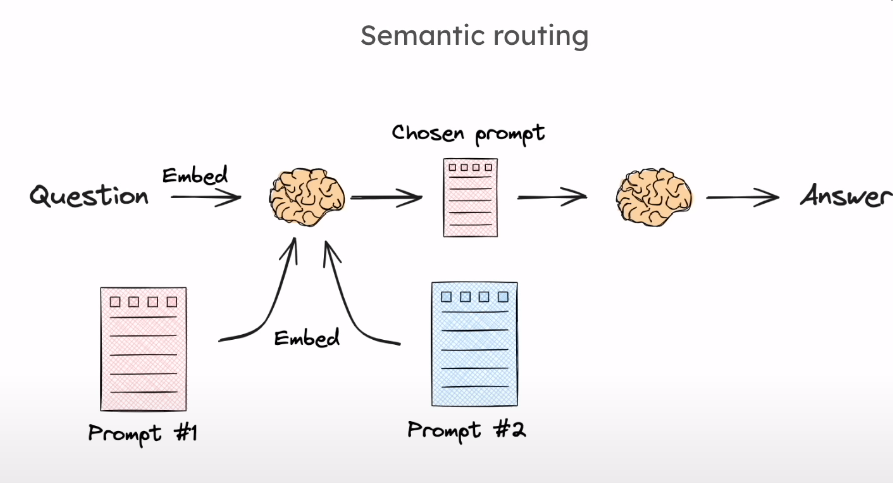




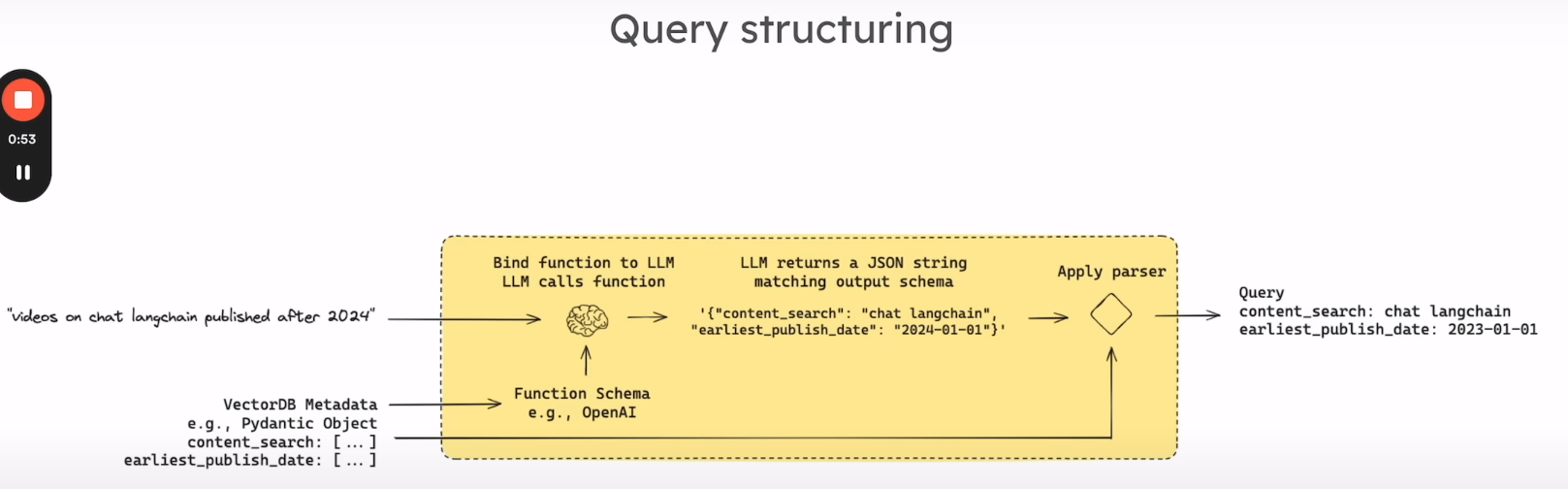
## 1. Logical Routing:



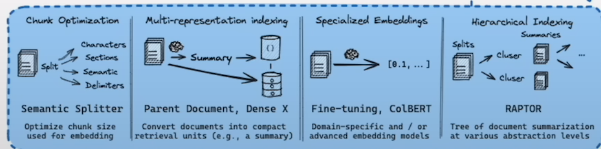
## 2. Semantic Routing:

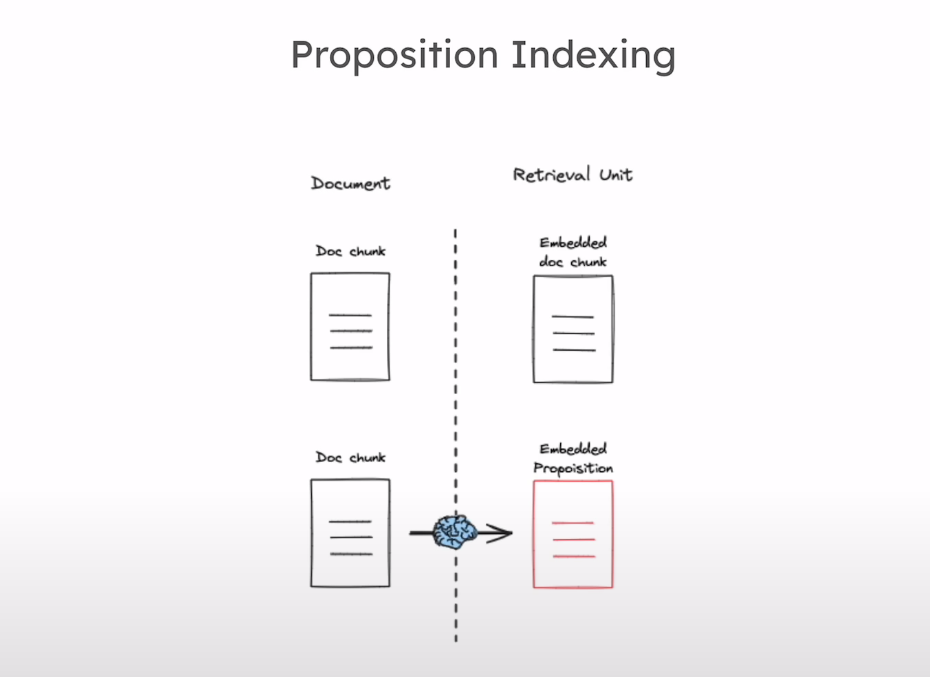


# Query Construction:

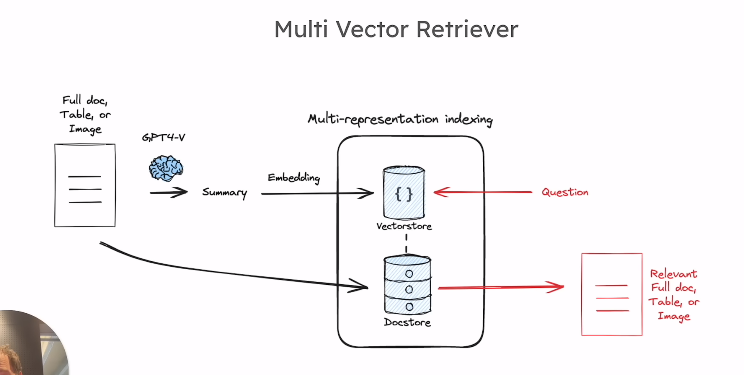


# Indexing:



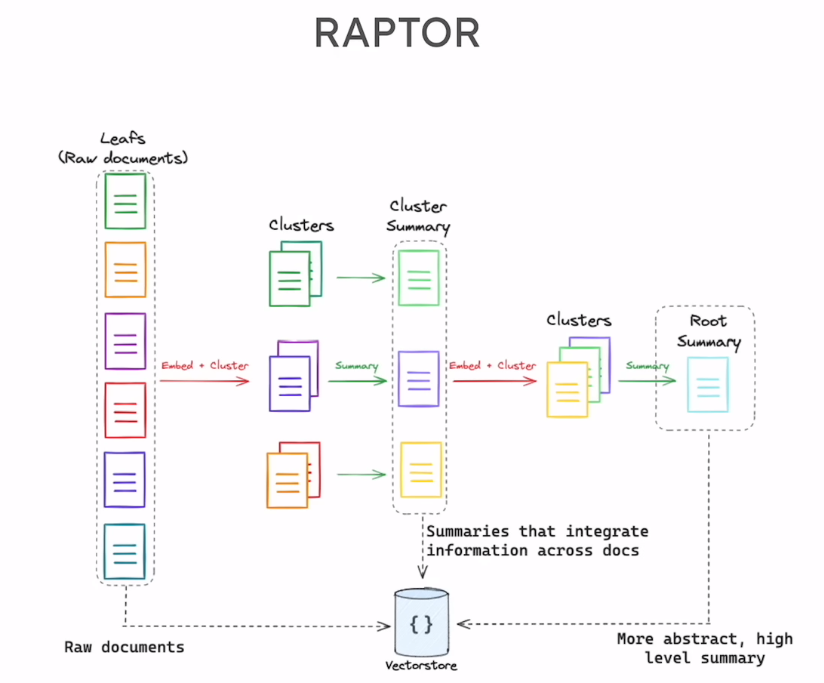


Change the split chunk is some way to make it crisper for retrieval using LLM



## RAPTOR:

For hierarchical indexing



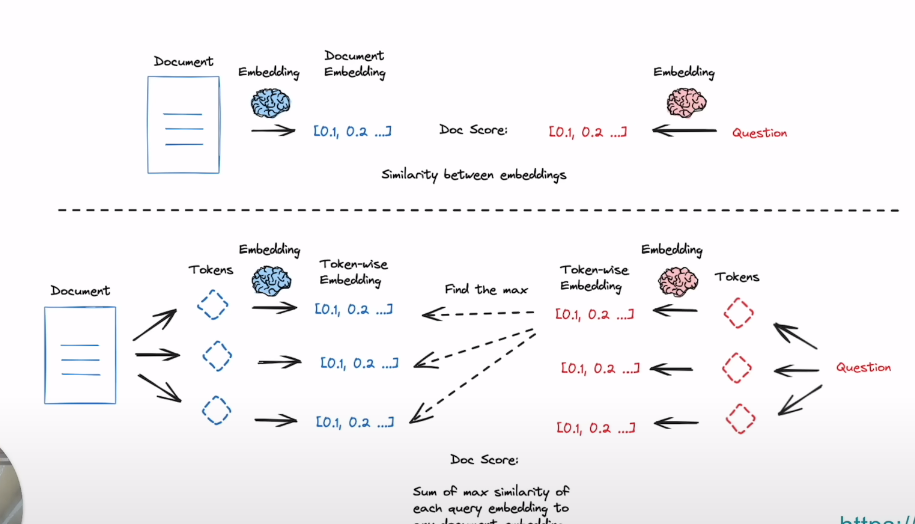
## ColBERT:

RAGatouille makes it as simple to use ColBERT.

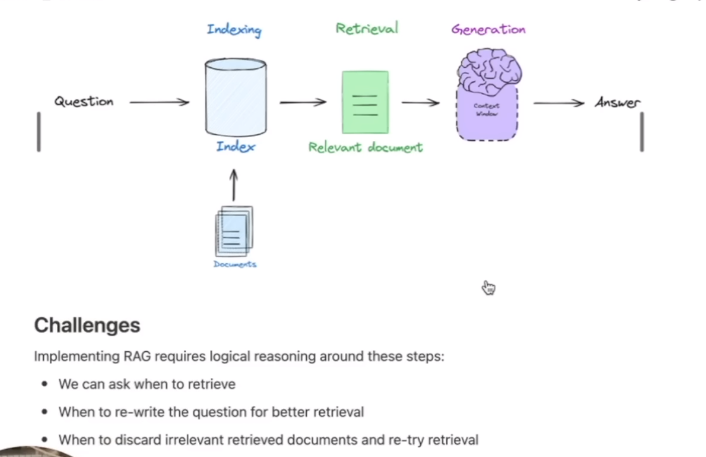
ColBERT generates a contextually influenced vector for each token in the passages.

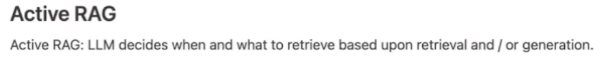
ColBERT similarly generates vectors for each token in the query.

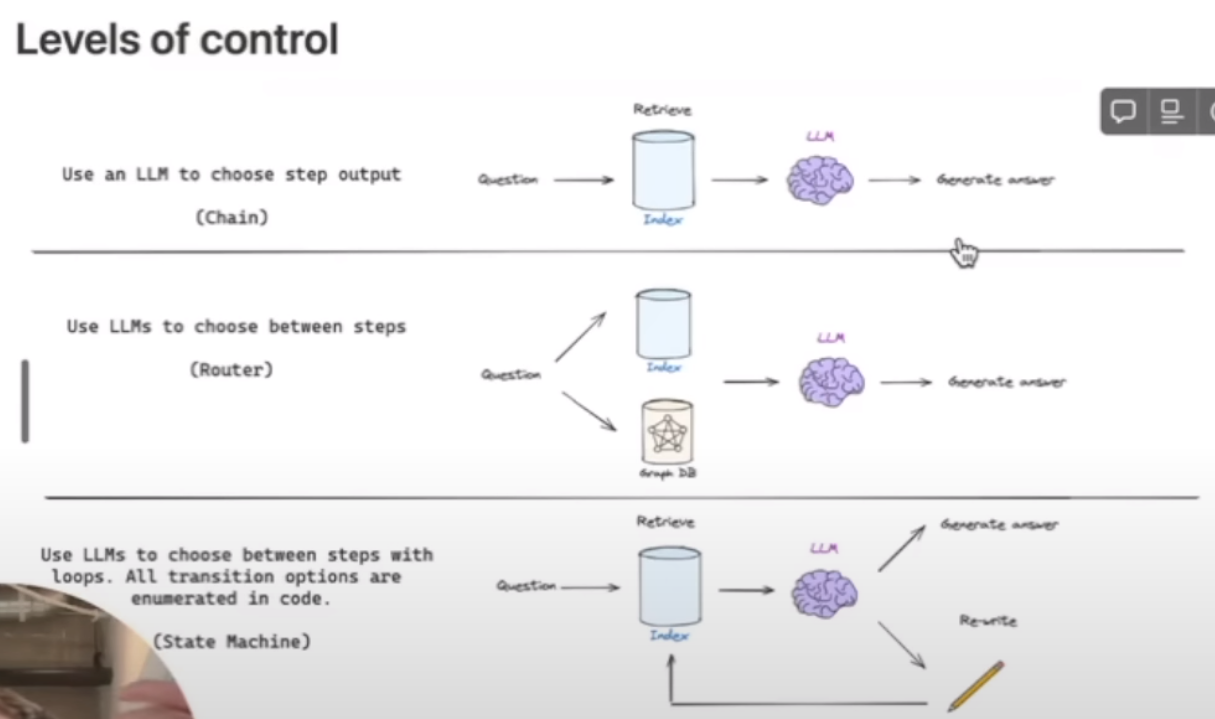
Then, the score of each document is the sum of the maximum similarity of each query embedding to any of the document embeddings:

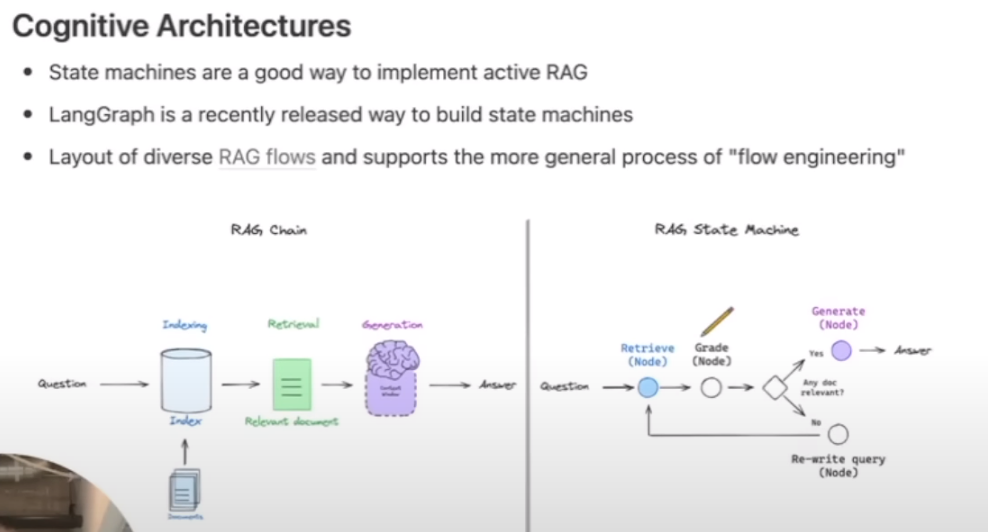


# Active RAG:









## CRAG:

1) load the documents.

2) We grade them based on relevance.

3) if at least one doc exceeds the threshold of relevance we generate.

4) if no doc exceeds the threshold, web search and pass this data as the context.

