Retrievers in LangChain

What are Retrievers?

A **retriever** in LangChain is a component responsible for fetching *relevant documents or chunks of information* from a data source based on a user's query. Retrievers form the core of retrieval-augmented generation (RAG) pipelines, enabling the language model to access external knowledge beyond its training data. **All retrievers in LangChain are Runnables** — meaning they can be executed independently, composed with other runnables, or integrated within chains.

Overview

- Connect user queries with knowledge sources.
- Fetch only the most relevant context for the LLM.
- Support multiple backends: vector stores, APIs, or custom data sources.
- Improve the accuracy and grounding of LLM responses.

Types of Retrievers

LangChain provides several retriever types to suit different data retrieval needs.

Vector Store Retriever

Vector Store Retriever

A **Vector Store Retriever** is the most common retriever in LangChain. It searches for documents stored as vector embeddings and retrieves those most *semantically similar* to the user's query.

Mechanism:

1. User query is embedded into a vector representation.
2. The retriever computes similarity (often cosine) between the query vector and stored document vectors.
3. Top-k most similar documents are returned.

Common Vector Stores: FAISS, Pinecone, Qdrant, Milvus, Weaviate, Chroma.

Use Case Example:

- RAG system fetching knowledge base documents for answering user queries.

Wikipedia Retriever

Wikipedia Retriever

A **Wikipedia Retriever** queries the official Wikipedia API to fetch relevant content directly from Wikipedia articles.

Working:

- Converts the user's query into a Wikipedia search request.
- Retrieves article snippets or summaries most related to the query.
- Optionally filters or cleans the text for LLM input.

Applications:

- Research chatbots or educational assistants.
- Summarization or fact-checking tools using real-world data.

Contextual Compression Retriever

Contextual Compression Retriever

The **Contextual Compression Retriever** is an advanced hybrid retriever that improves retrieval efficiency and quality by *compressing* documents after retrieval.
Process:

1. Performs initial retrieval (using another retriever, e.g., vector store).
2. Applies a *compressor model* to summarize or trim irrelevant content.
3. Returns only the most query-relevant snippets.

Advantages:

- Reduces context size for the LLM.
- Retains only high-value information.

Example: Using a summarization chain as a compressor to condense Wikipedia paragraphs before passing them to the LLM.

Maximal Marginal Relevance (MMR)

Maximal Marginal Relevance (MMR)

MMR is an algorithm used to refine retrieval results by balancing two key factors:

- **Relevance:** How closely the document matches the query.
- **Diversity:** How distinct the retrieved documents are from each other.

Purpose: Reduce redundancy among retrieved results while maintaining high coverage of the topic.
Mathematical Idea:

$$\text{MMR} = \arg \max_{D_i \in R \setminus S} [\lambda \cdot \text{Sim}(D_i, Q) - (1 - \lambda) \cdot \max_{D_j \in S} \text{Sim}(D_i, D_j)]$$

where λ controls the balance between relevance and diversity.
Usage:

- Applied to vector store retrievers to return diverse and non-redundant document sets.

Multi-Query Retriever

Multi-Query Retriever

A **Multi-Query Retriever** enhances the robustness of retrieval by generating multiple query reformulations for the same user input.

Steps:

1. LLM generates several paraphrased or expanded versions of the query.
2. Each query variant is sent to the underlying retriever.
3. Results are combined and deduplicated for higher recall.

Benefits:

- Captures multiple semantic aspects of the same question.
- Especially effective when users ask vague or broad queries.

Summary

Key Takeaways

- Retrievers are crucial components that connect user queries with data sources.
- Vector Store Retrievers are the backbone of RAG systems.
- MMR and Contextual Compression improve retrieval diversity and relevance.
- Multi-Query retrievers boost recall through query reformulation.
- Wikipedia Retriever allows real-world, API-driven information access.