**LangChain** là một **framework mã nguồn mở** được thiết kế để đơn giản hóa việc xây dựng các ứng dụng phức tạp dựa trên các Mô hình Ngôn ngữ Lớn (LLM).

Hãy tưởng tượng bạn đang lắp ráp một mô hình phức tạp từ các khối LEGO. Thay vì phải tự mình chế tạo từng khối, LangChain cung cấp cho bạn các khối LEGO đã được làm sẵn (các module) và hướng dẫn lắp ráp (các "chain" - chuỗi). Bạn chỉ cần kết nối chúng lại với nhau để tạo ra một ứng dụng hoàn chỉnh.

Nói cách khác, LangChain đóng vai trò là **chất keo kết dính**, giúp bạn dễ dàng liên kết LLM (như GPT) với các nguồn dữ liệu bên ngoài (như file PDF, website, hoặc Vector DB) và các công cụ khác.

**Dùng LangChain để kết hợp GPT + Vector DB**

Đây chính là ứng dụng tiêu biểu và mạnh mẽ nhất của LangChain: xây dựng một hệ thống **RAG (Retrieval-Augmented Generation)**. LangChain làm cho quá trình này trở nên cực kỳ đơn giản và có cấu trúc.

Thay vì phải tự viết code cho từng bước (đọc file, phân đoạn, tạo embedding, lưu vào DB, truy vấn, tạo prompt...), LangChain cung cấp sẵn các module cho mỗi công việc:

* **Document Loaders:** Để đọc dữ liệu từ hàng loạt nguồn (PDF, TXT, web...).
* **Text Splitters:** Để chia văn bản thành các đoạn (chunks) một cách thông minh.
* **Embedding Models:** Dễ dàng tích hợp với các mô hình embedding.
* **Vector Stores:** Giao tiếp với nhiều loại Vector DB khác nhau (FAISS, Chroma, Pinecone...).
* **Retrievers:** Để thực hiện việc truy xuất thông tin từ Vector DB.
* **Chains:** Kết nối tất cả các bước trên lại với nhau thành một luồng xử lý hoàn chỉnh, ví dụ như RetrievalQA chain.

**Mini-Project: Xây dựng hệ thống Hỏi-Đáp với LangChain**

Dưới đây là một project hoàn chỉnh, minh họa cách LangChain giúp bạn xây dựng một ứng dụng RAG chỉ với vài chục dòng code.

**Giải thích mã**

Bạn có thể thấy, với LangChain, toàn bộ quy trình phức tạp được gói gọn trong các hàm rất rõ ràng:

* TextLoader(...).load(): Tải dữ liệu.
* CharacterTextSplitter(...).split\_documents(): Phân đoạn.
* FAISS.from\_documents(): Tự động tạo embedding và lưu vào FAISS.
* RetrievalQA.from\_chain\_type(): Tạo ra một chuỗi Hỏi-Đáp hoàn chỉnh.

Khi bạn gọi rag\_chain.invoke(), LangChain sẽ tự động làm tất cả các việc còn lại: lấy câu hỏi của bạn, tìm kiếm trong FAISS, lấy các đoạn liên quan, tạo prompt, gửi đến GPT và trả về kết quả.

Đây chính là sức mạnh của LangChain: **trừu tượng hóa sự phức tạp**, cho phép bạn tập trung vào logic của ứng dụng thay vì phải lo lắng về các chi tiết kỹ thuật ở tầng thấp.