**Langchain**

LangChain is an open-source framework designed to simplify the creation of applications using large language models (LLMs). It provides a standard interface for integrating with other tools and end-to-end chains for common applications. It helps AI developers connect LLMs such as GPT-4 with external data and computation. This framework comes for both Python and JavaScript.

**Key benefits include:**

* Modular Workflow: Simplifies chaining LLMs together for reusable and efficient workflows.
* Prompt Management: Offers tools for effective prompt engineering and memory handling.
* Ease of Integration: Streamlines the process of building LLM-powered applications.

**Key Components of LangChain**

**A hexagon with text and icons

AI-generated content may be incorrect.**

**1. Chains:**Chains define sequences of actions, where each step can involve querying an LLM, manipulating data or interacting with external tools. There are two types:

* Simple Chains: A single LLM invocation.
* Multi-step Chains: Multiple LLMs or actions combined, where each step can take the output from the previous step.

**2. Prompt Management:**LangChain facilitates managing and customizing prompts passed to the LLM. Developers can use PromptTemplates to define how inputs and outputs are formatted before being passed to the model. It also simplifies tasks like handling dynamic variables and prompt engineering, making it easier to control the LLM's behavior.

**3. Agents:**Agents are autonomous systems within LangChain that take actions based on input data. They can call external APIs or query databases dynamically, making decisions based on the situation. These agents leverage LLMs for decision-making, allowing them to respond intelligently to changing input.

**4. Vector Database**: LangChain integrates with a vector database which is used to store and search high-dimensional vector representations of data. This is important for performing similarity searches, where the LLM converts a query into a vector and compares it against the vectors in the database to retrieve relevant information.

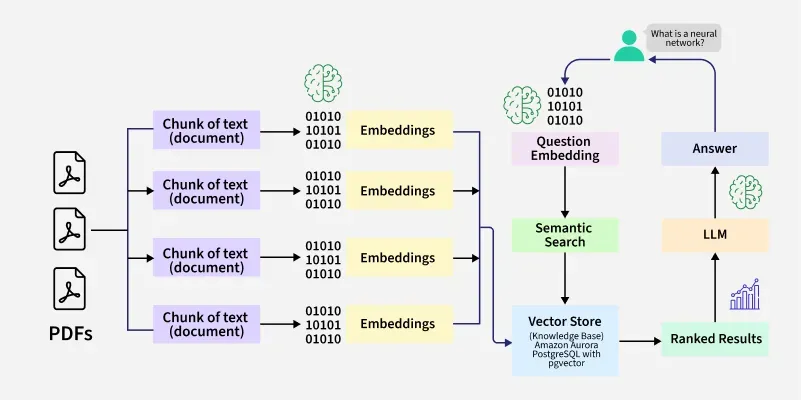
Vector database plays a key role in tasks like document retrieval, knowledge base integration or context-based search providing the model with dynamic, real-time data to enhance responses.

**5. Models**: LangChain is model-agnostic meaning it can integrate with different LLMs such as OpenAI's GPT, Hugging Face models, DeepSeek R1 and more. This flexibility allows developers to choose the best model for their use case while benefiting from LangChain’s architecture.

**6. Memory Management:**LangChain supports memory management allowing the LLM to "remember" context from previous interactions. This is especially useful for creating conversational agents that need context across multiple inputs. The memory allows the model to handle sequential conversations, keeping track of prior exchanges to ensure the system responds appropriately.

**How LangChain Works?**

LangChain follows a structured pipeline that integrates user queries, data retrieval and response generation into seamless workflow.

****

With LangChain, developers can adapt a language model flexibly to specific business contexts by designating steps required to produce the desired outcome.

**Chains**

*Chains* are the fundamental principle that holds various AI components in LangChain to provide context-aware responses. A chain is a series of automated actions from the user's query to the model's output. For example, developers can use a chain for:

* Connecting to different data sources.
* Generating unique content.
* Translating multiple languages.
* Answering user queries.

**Links**

Chains are made of *links*. Each action that developers string together to form a chained sequence is called a link. With links, developers can divide complex tasks into multiple, smaller tasks. Examples of links include:

* Formatting user input.
* Sending a query to an LLM.
* Retrieving data from cloud storage.
* Translating from one language to another.

In the LangChain framework, a link accepts input from the user and passes it to the LangChain libraries for processing. LangChain also allows link reordering to create different AI workflows.

Overview

To use LangChain, developers install the framework in Python with the following command:

*pip install langchain*

Developers then use the chain building blocks or LangChain Expression Language (LCEL) to compose chains with simple programming commands. The *chain()* function passes a link's arguments to the libraries. The*execute()* command retrieves the results. Developers can pass the current link result to the following link or return it as the final output.

Below is an example of a chatbot chain function that returns product details in multiple languages.

*chain([*

*retrieve\_data\_from\_product\_database().*

*send\_data\_to\_language\_model().*

*format\_output\_in\_a\_list().*

*translate\_output\_in\_target\_language()*

*])*

**Promptlayer**

**PromptLayer** is a comprehensive platform designed to streamline and supercharge prompt engineering for applications built on Large Language Models (LLMs) like OpenAI's GPT. It acts as a middleware between your code and LLM APIs, offering tools for **prompt management, evaluation, observability, and collaboration**.

**What Is PromptLayer?**

PromptLayer is a **prompt engineering control center** that helps developers and non-technical teams manage, test, and optimize prompts used in LLM-powered applications. It provides a visual interface and backend infrastructure to version, log, and analyze prompt interactions.



**Key Features**

**1. Prompt Management**

* Centralized **Prompt Registry** to store and organize prompts.
* Visual editor for **no-code prompt editing**.
* **Version control** with rollback and side-by-side comparisons.
* **Prompt chaining** to build complex workflows without writing code.

**2. Evaluation & Testing**

* **A/B testing** of prompt versions in production.
* **Regression testing** using historical or synthetic datasets.
* **Batch evaluations** to validate prompt updates.
* **Human and automated grading** for quality assurance.

**3. LLM Observability**

* Real-time analytics on **latency, cost, and token usage**.
* Full **request/response logging** with metadata.
* Debugging tools to trace and resolve issues quickly.

**4. Collaboration Tools**

* Shared workspaces for **domain experts and engineers**.
* Permission-based editing and approval workflows.
* Enables **non-technical stakeholders** to contribute directly.

**5. Security & Compliance**

* SOC 2-compliant infrastructure.
* API keys are never sent to PromptLayer servers—requests are logged locally.

**Use Cases**

* **Customer Support Automation**: Gorgias scaled to 5K+ monthly prompt iterations for Shopify merchants.
* **Education & Curriculum Development**: Speak launched in 11 markets by enabling non-tech teams to localize prompts.
* **Healthcare & Coaching**: ParentLab built therapeutic chatbots by converting educator expertise into prompts.
* **Debugging LLM Agents**: Ellipsis reduced debugging time by 75% using PromptLayer’s tracing tools.

A diagram of a computer

AI-generated content may be incorrect.

**Integrations with LLMs**

PromptLayer integrates with a wide range of LLM providers and frameworks:

**Supported Platforms**

* **OpenAI** (via SDK wrapper)
* **Anthropic Claude**
* **Cohere**
* **Hugging Face**
* **Google Vertex AI**
* **Amazon Bedrock**
* **LangChain**
* **LiteLLM**
* **LlamaIndex**

**Benefits**

* **Accelerates development** by decoupling prompt updates from code releases.
* **Improves prompt quality** through structured testing and analytics.
* **Empowers teams** with diverse skill sets to collaborate effectively.
* **Reduces engineering bottlenecks** and debugging time.

**How PromptLayer Works**

Data-driven prompt engineering

PromptLayer is a devtool that allows you to track, manage, and share your GPT prompt engineering. It acts as a middleware between your code and OpenAI’s python library, recording all your API requests and saving relevant metadata for easy exploration and search in the PromptLayer dashboard.

**[​](https://docs.promptlayer.com/why-promptlayer/how-it-works" \l "how-it-works)**

**How it Works**

PromptLayer works by wrapping your OpenAI API requests and logging data about them after they are sent. This is all done from your machine, your API key is never sent. This means that it does not interfere with the functionality of your existing codebase or require any changes to your application’s architecture. All you need to do is add PromptLayer as an add-on to your existing LLM application and start making requests as usual.As you make OpenAI API requests, PromptLayer records them and saves relevant metadata such as the prompt used, the response returned, and any additional parameters that were passed. This data is stored by PromptLayer and can be easily accessed via the PromptLayer dashboard

**Features**

Some of the key features of PromptLayer include:

* **API request logging:** PromptLayer records all your OpenAI API requests, allowing you to search and explore request history in the PromptLayer dashboard.
* **Metadata tracking:** Under the hood, PromptLayer logs each OpenAI request *after* the request is made, saving relevant metadata such as the prompt used, the response returned, and any additional parameters that were passed.
* **Easy integration:** PromptLayer is an add-on to your existing LLM application and requires no changes to your application’s architecture.
* **Designed for production:** PromptLayer is designed to help maintain LLMs in production and aid in the development process. It is prod-ready and will not interfere with your application’s functionality even if it fails.
* **Collaboration:** PromptLayer allows you to share your prompt engineering with others, making it easy to collaborate on projects with teammates or share your work with the wider community.

**Conclusion**

PromptLayer is a powerful devtool that makes it easy to track, manage, and share your GPT prompt engineering. With its easy integration, prod-ready design, and collaboration features, it is an essential tool for anyone working with LLMs in production or development. Sign up for an account on the PromptLayer website to get started today!