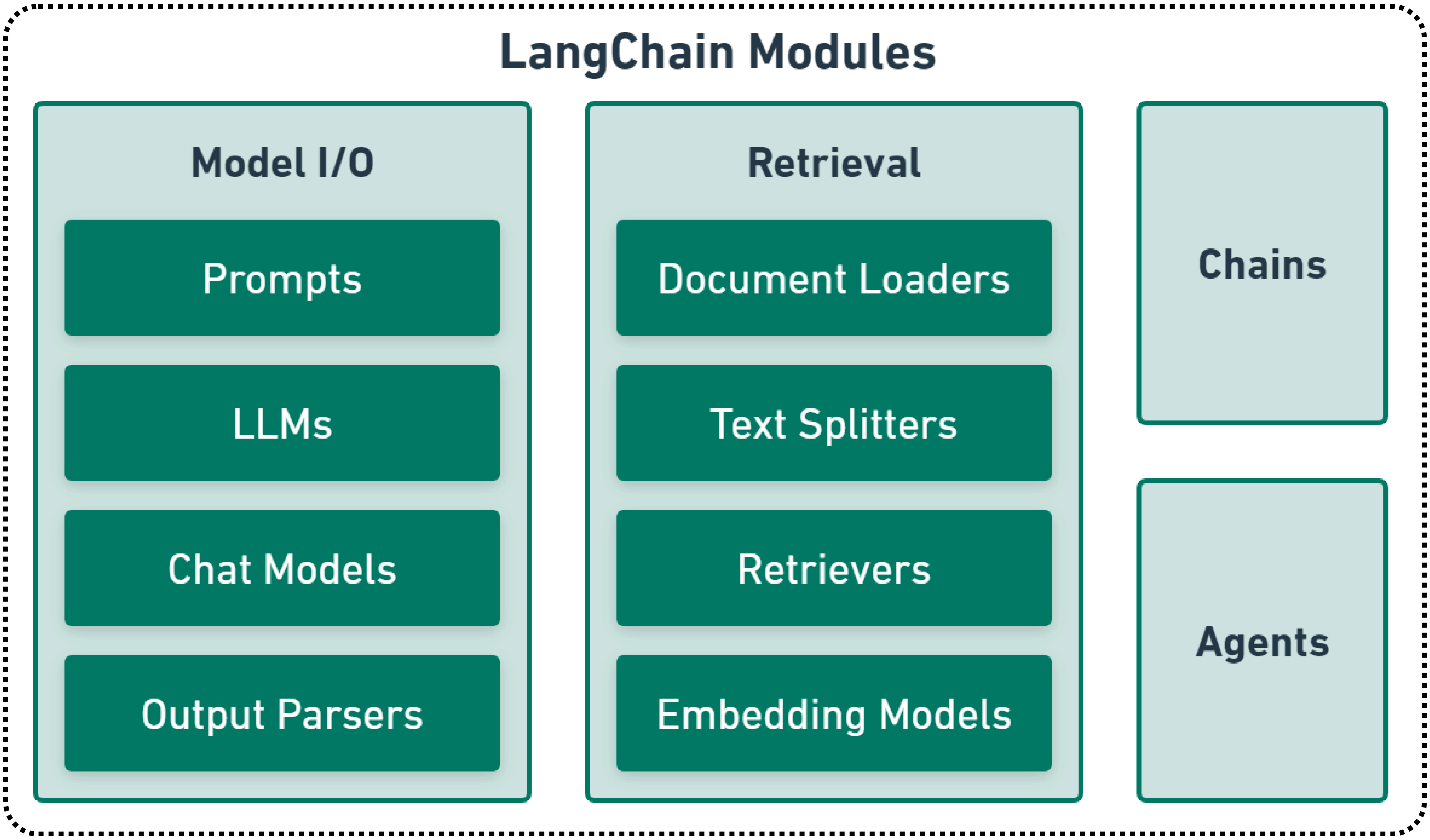
Analytics Vidhya

Deep Dive into Langchain



LangChain Modules

Model I/O

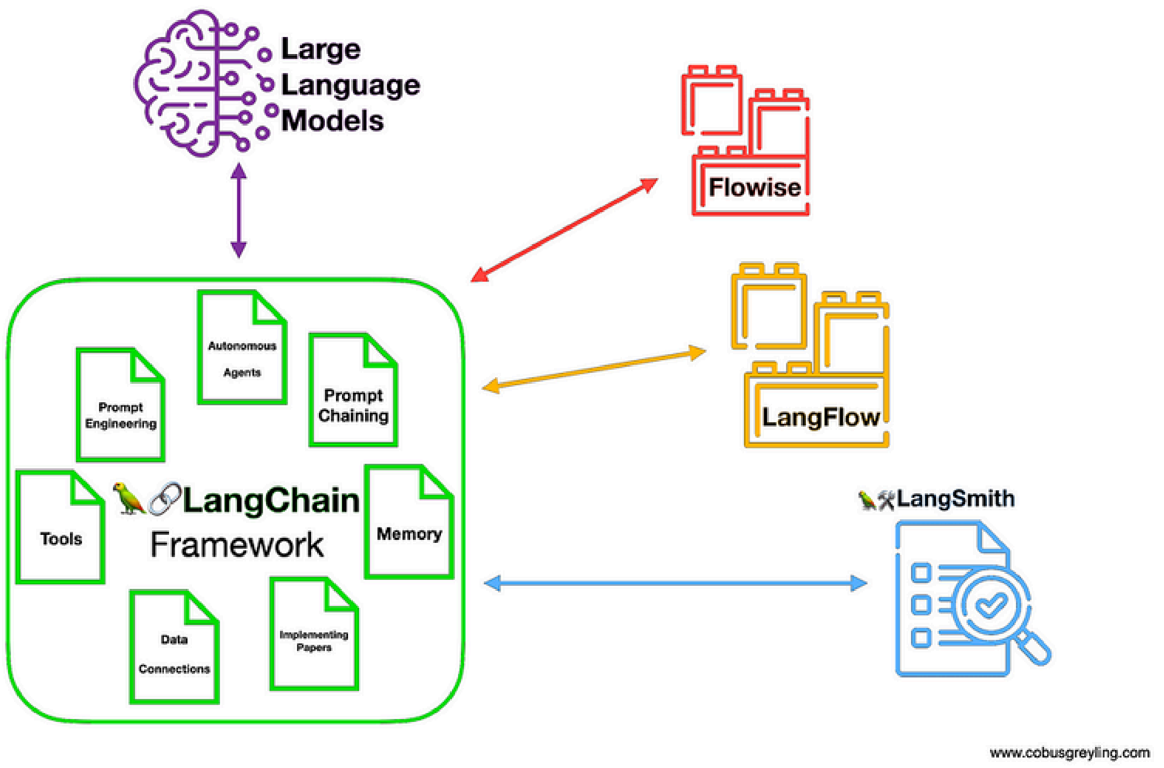
- Prompts
- LLMs
- Chat Models
- Output Parsers

Retrieval

- Document Loaders
- Text Splitters
- Retrievers
- Embedding Models

Chains

Agents



What is LangChain?

- LangChain is an open-source orchestration framework for building applications using large language models (LLMs). Available in both Python and JavaScript-based libraries, LangChain provides a centralized development environment and set of tools to simplify the process of creating LLM-driven applications like chatbots and virtual agents.

- Serving as a generic interface for integrating with various LLMs, LangChain's modular design allows developers and data scientists to dynamically compare different prompts and even different foundation models with minimal need to rewrite code.

- This flexibility also enables building programs that utilize multiple LLMs together, such as one model for interpreting user queries and another for generating responses.

- The way I see it, LangChain is filling an important gap – providing a common framework to build upon, so that innovators and creators don't have to reinvent the wheel every time they want to leverage the power of LLM.

How Does LangChain Work?

The core idea behind LangChain is to provide a modular, flexible framework for building applications that utilize large language models (LLMs). At the heart of LangChain are a few key concepts:

LLMs

- At the core of LangChain is the ability to seamlessly integrate with a variety of large language models (LLMs) from different providers, such as OpenAI, Anthropic, and Google. LangChain provides a standardized interface to interact with these powerful AI models, abstracting away the complexities of working with each vendor's unique APIs and input/output formats.

Chains

- LangChain's Chains are the basic building blocks for creating complex workflows and processing pipelines. A Chain is a sequence of operations that can be performed on the outputs of an LLM. For example, you might have a Chain that first uses an LLM to extract key information from user input, then passes that to another LLM to generate a relevant response. Chaining multiple LLM-powered steps together enables developers to tackle increasingly sophisticated natural language tasks.

Agents

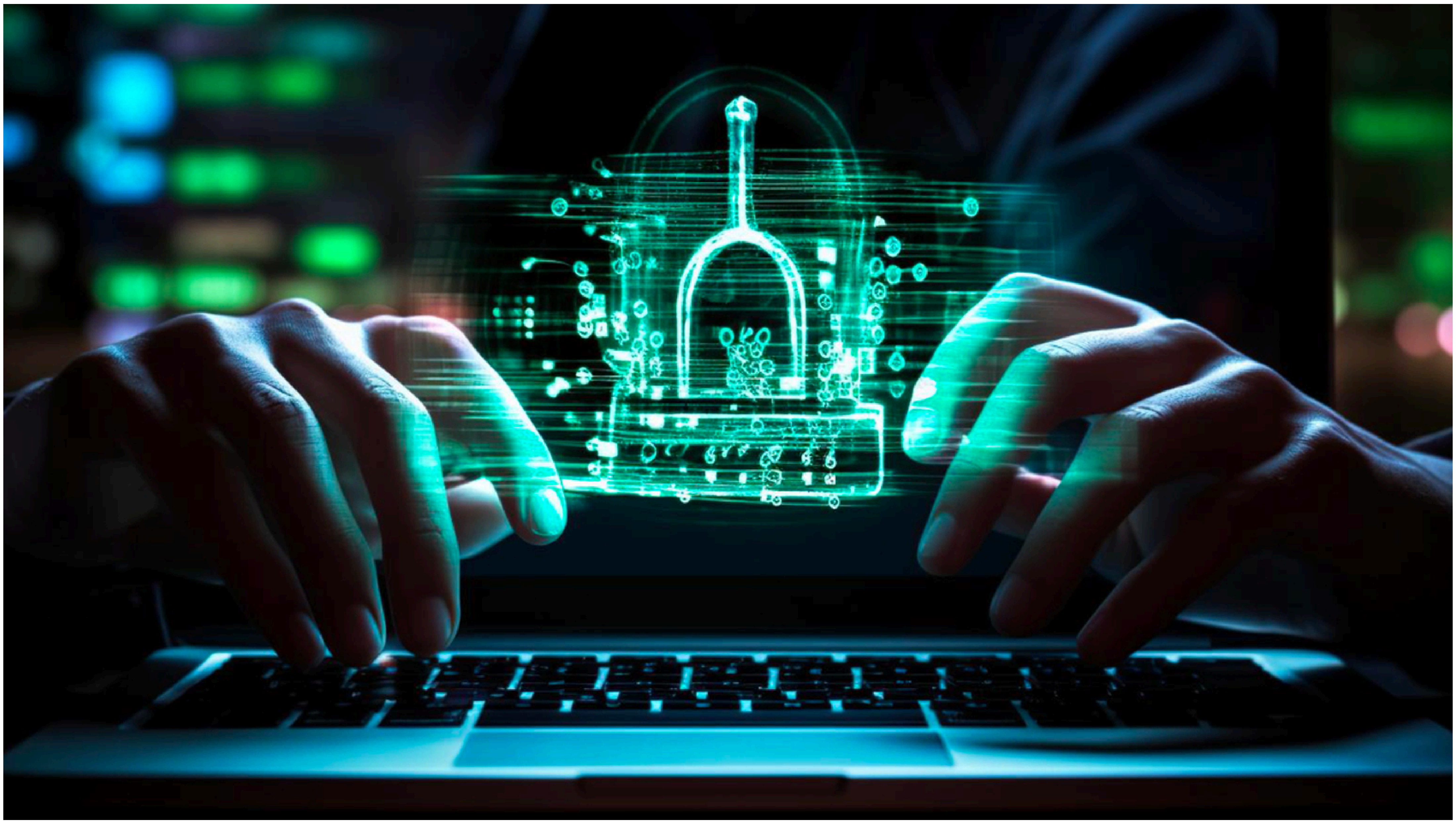
Building on the Chains concept, LangChain introduces higher-level abstractions called Agents. Agents are self-contained units that can leverage Chains and other LangChain components to autonomously solve complex, goal-driven tasks. Agents encapsulate the logic for interacting with LLMs, managing state and memory, and coordinating multi-step workflows. This allows developers to create intelligent, LLM-powered “actors” that can engage in more natural, contextual conversations and complete intricate assignments.

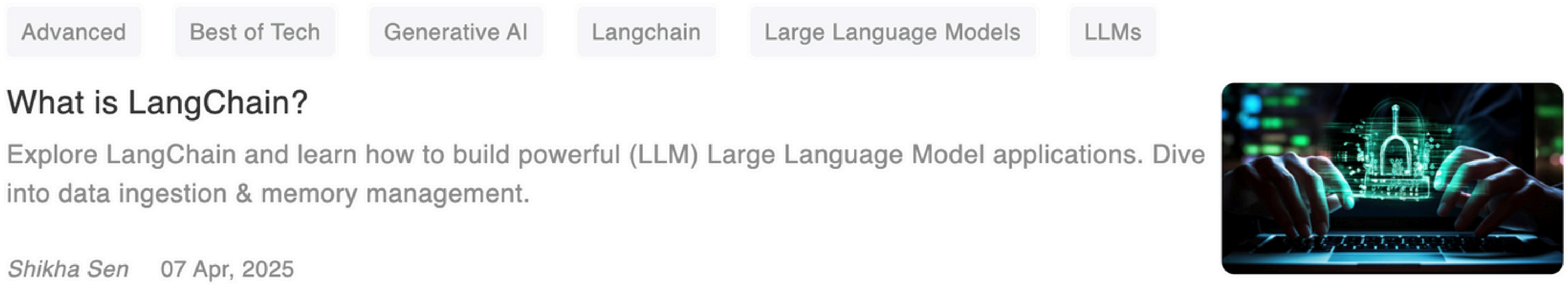
Memory

A crucial capability provided by LangChain is its memory management system. This allows LLMs to store and retrieve relevant information during the course of a multi-step workflow, enabling context preservation and statefulness across executions. The memory component is essential for building conversational applications and other LLM-powered experiences that require an understanding of previous interactions and intermediate results.

By combining these core elements – LLM integrations, Chains, Agents, and Memory – LangChain gives developers a comprehensive toolkit for building sophisticated applications driven by large language models. The framework’s modular, flexible design empowers creators to experiment, iterate, and scale their LLM-powered solutions more efficiently.

For more information, kindly visit this article





Advanced · Best of Tech · Generative AI · Langchain · Large Language Models · LLMs

What is LangChain?

Explore LangChain and learn how to build powerful (LLM) Large Language Model applications. Dive into data ingestion & memory management.

Shikha Sen 07 Apr, 2025