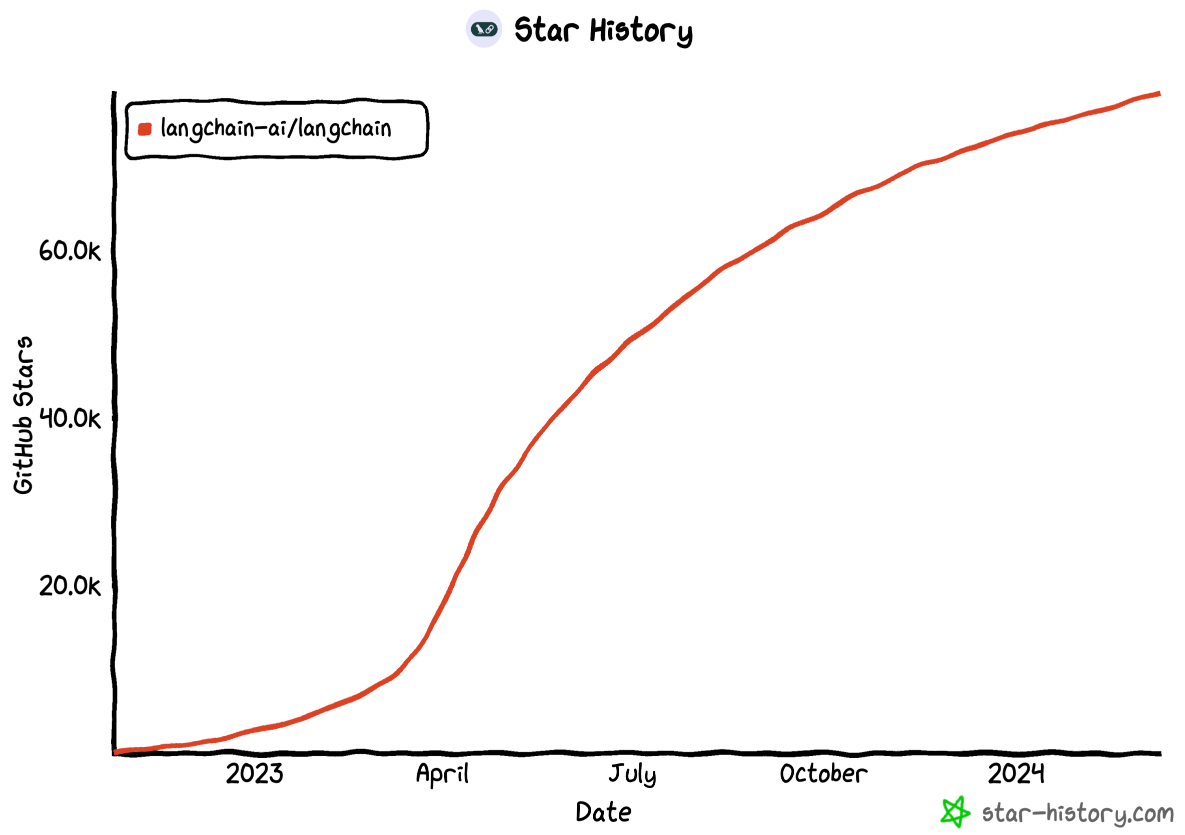
1933 words | 5 minutes

LangChain

1. What is LangChain

The huge success of ChatGPT has inspired more and more developers to develop applications based on large language models using the API or private models provided by OpenAI. Although the call of large language models is relatively simple, creating a complete application still requires a lot of custom development work, including API integration, interaction logic, data storage, and so on.

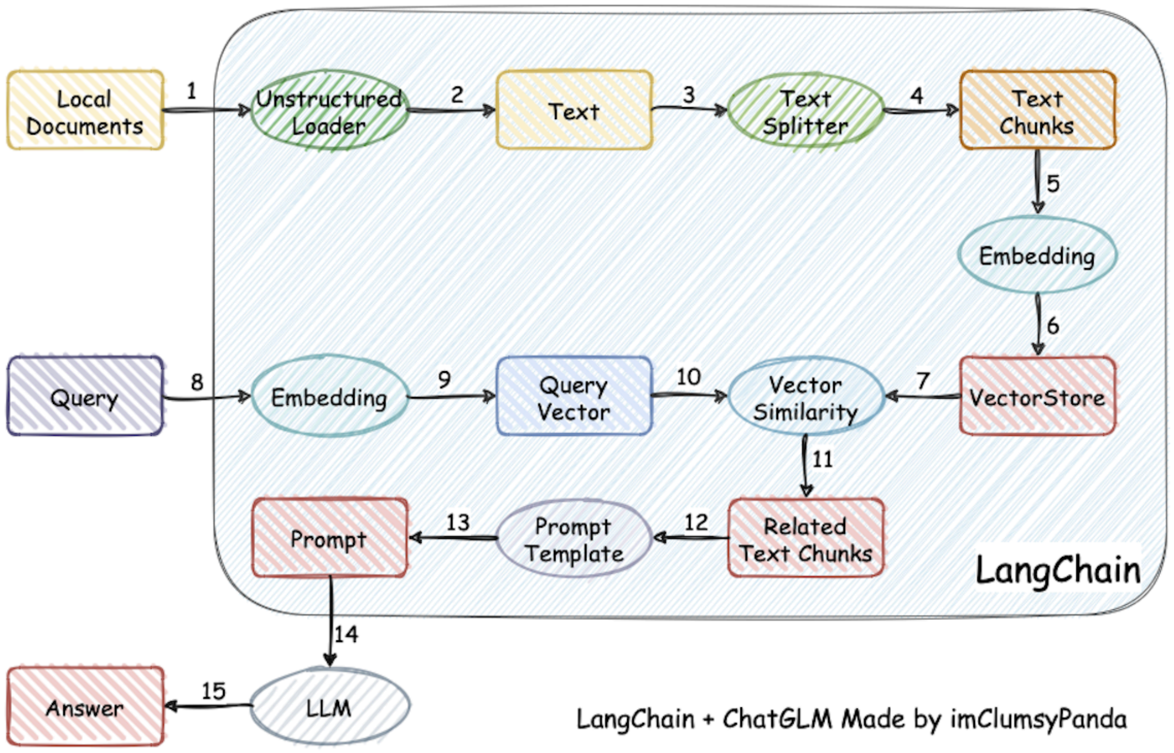
To solve this problem, many institutions and individuals have launched a number of open source projects since 2022 to **help developers quickly build end-to-end applications or workflows based on large language models** . One of the most popular projects is the LangChain framework.



The LangChain framework is an open source tool that leverages the power of large language models to develop a variety of downstream applications. Its goal is to provide a common interface for a variety of large language model applications, thereby simplifying the

application development process . Specifically, the LangChain framework enables data awareness and environmental interaction, that is, it enables language models to connect with other data sources and allows language models to interact with their environment.

Using the LangChain framework, we can easily build a RAG application as shown below (**Image source**). In the figure below, 每个椭圆形代表了 `LangChain` 的一个模块 , such as a data collection module or a preprocessing module. 每个矩形代表了一个数据状态 , such as raw data or preprocessed data. The arrows indicate the direction of data flow, from one module to another. At each step, LangChain can provide corresponding solutions to help us handle various tasks.



2. Core Components of LangChain

As a large language model development framework, LangChain can integrate LLM models (dialogue models, embedding models, etc.), vector databases, interactive layer prompts, external knowledge, and external agent tools, and can freely build LLM applications. LangChain mainly consists of the following 6 core components:

- **Model I/O** : Interface for interacting with language models
- **Data connection** : An interface for interacting with application-specific data.
- **Chains** : Combine components to achieve end-to-end applications. For example, we will build it 检索问答链 to complete search question and answer later.
- **Memory** : used to persist application state between multiple runs of the chain;

- **Agents** : Extend the reasoning capabilities of the model. Used for complex application call sequences;
- **Callbacks** : Extend the reasoning capabilities of the model. Used for complex application call sequences;

During the development process, we can flexibly combine according to our own needs.

3. Stable version of LangChain

In the rapid development of LLM technology, LangChain, as an evolving innovation platform, continues to push the boundaries of technology. `2024 年 1 月 9 日` LangChain officially released its stable version **v0.1.0** . This milestone update brings comprehensive and powerful functional support to developers. It covers key components such as model input and output processing, data connection, chain operation, memory mechanism, proxy service, and callback processing, providing a solid foundation for the development and deployment of LLM applications. At the same time, LangChain's continuous optimization and functional iteration will bring more innovative features and performance improvements in the future.

- **Compatibility and support** : LangChain v0.1.0 takes into account `Python 和 JavaScript` the support for while maintaining backward compatibility, ensuring that developers can seamlessly transition during the upgrade process and enjoy a more secure and stable development experience.

- **Architecture Improvement** : By effectively separating the core component langchain-core from the partner package, LangChain's architecture design has become more organized and stable, laying a solid foundation for future systematic expansion and security improvement.

- **Observability** : LangChain provides industry-leading debugging and observation capabilities through deep integration with LangSmith. This enables developers to have a clear understanding of each operation and its input and output in the LLM application, greatly simplifying the debugging and troubleshooting process.

- **Wide range of integrations** : LangChain has nearly **700** integrations, covering multiple technical areas from LLM to vector storage, tools, and agents, greatly reducing the complexity of building LLM applications on various technology stacks.

- **Composability** : With `LangChain 表达式语言（LCEL）` , developers can easily build and customize chains and take full advantage of the data orchestration framework, including advanced features such as batch processing, parallel operations, and alternatives.

- **Streaming processing** : LangChain has deeply optimized streaming processing to ensure that all chains created using LCEL can support streaming processing, including data streaming transmission in intermediate steps, thereby providing users with a smoother experience.
- **Output parsing** : LangChain provides a series of powerful output parsing tools to ensure that LLM can return information in a structured format, which is crucial for LLM to execute specific action plans.
- **Retrieval capabilities** : LangChain introduces advanced retrieval technologies suitable for production environments, including text segmentation, retrieval mechanisms, and indexing pipelines, allowing developers to easily combine private data with the capabilities of LLM.
- **Tool Usage and Agents** : LangChain provides a rich collection of agents and tools, and provides a simple way to define tools. It supports agent workloads, including allowing LLM to call functions or tools, and how to efficiently make multiple calls and inferences, greatly improving development efficiency and application performance.

4. LangChain Ecosystem

- **LangChain Community** : Focusing on third-party integration, it greatly enriches the LangChain ecosystem, making it easier for developers to build complex and powerful applications, while also promoting community cooperation and sharing.
- **LangChain Core** : The core library and core component of the LangChain framework, which provides basic abstractions and the LangChain Expression Language (LCEL), and provides infrastructure and tools for building, running, and interacting with LLM applications, providing a solid foundation for the development of LangChain applications. The document processing, formatting prompt, output parsing, etc. that we will use later all come from this library.
- **LangChain CLI** : A command line tool that enables developers to interact with the LangChain framework through the terminal to perform tasks such as project initialization, testing, and deployment. It improves development efficiency and allows developers to manage the entire application lifecycle through simple commands.
- **LangServe** : A deployment service for deploying LangChain applications to the cloud, providing a scalable, highly available hosting solution with monitoring and logging capabilities. It simplifies the deployment process so that developers can focus on application development without having to worry about the underlying infrastructure and operations.
- **LangSmith** : A developer platform that focuses on the development, debugging, and testing of LangChain applications, providing a visual interface and performance analysis tools

designed to help developers improve the quality of their applications and ensure that they meet the expected performance and stability standards before deployment.

> **In this chapter, we briefly introduced the development framework LangChain. In the next chapter, we will introduce the overall process of developing LLM applications.**

Previous chapter

2. Introduction to Retrieval Enhancement Generation RAG

Next Chapter

4. Overall process of developing LLM application