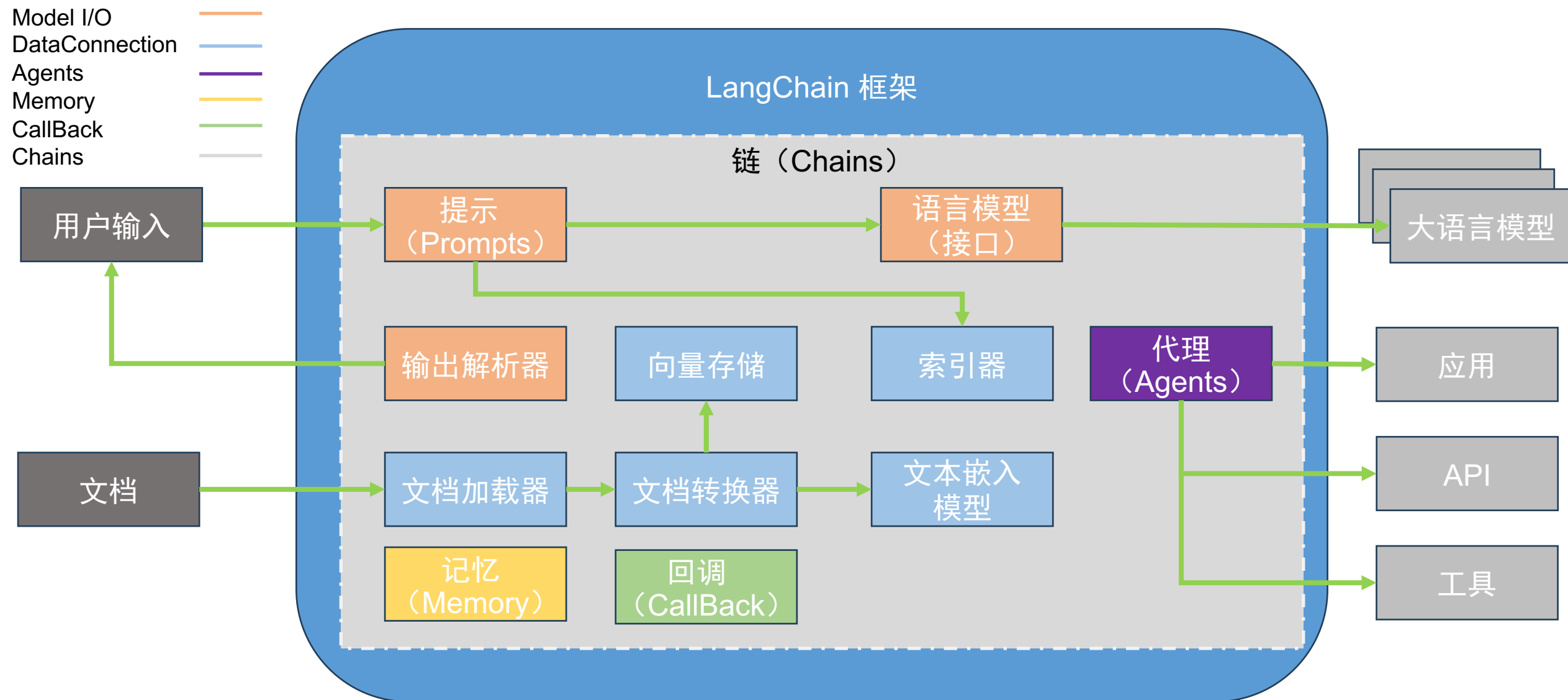


6-1 Memory 组件概述

- 回顾组件地图
- Memory 组件基本概念



LangChain 6 大组件回顾



Memory 组件概述

是什么： Memory组件主要目标是为了让AI系统能够"记住"它的对话历史以及其他相关信息，并且能够在需要的时候检索出这些信息。这样可以让AI系统在对话中提供更连贯、更个性化的回答，并且可以帮助AI系统更好地理解用户的需求。

为什么： 使用Memory的主要原因如下：

- 1.保存上下文信息：信息可以来自与历史聊天，也可以来自于实时聊天，还可以来自于外部文件。
- 2.检索信息：memory组件还能够根据给定的输入，从向量数据库中检索出最相关的信息。这个检索过程是基于语义相似度的，也就是说，被检索出的信息不一定要完全匹配输入，只需要在语义上与输入相似就可以。
- 3.跟踪对话历史：memory组件还能够跟踪对话的历史信息。这样可以让AI系统在对话中保持上下文连贯性，并且可以让AI系统根据历史信息来生成更个性化的回答。

ConversationBufferMemory

会话缓冲记忆（ConversationBufferMemory）可以将预设的聊天消息保存起来，为之后的聊天提供上下文信息。同时，也可以将实时聊天的信息保存起来，提供丝滑的聊天体验

会话缓冲窗口记忆（ConversationBufferWindowMemory）是上面一种记忆方式的升级版，因为随着聊天次数的增加，系统无法记忆所有的聊天记录，于是通过这个方法记录最近K次的聊天交互信息。

6-2 Memory 组件分类

- Conversation summary memory
- Conversation Summary Buffer Memory
- Vector store-backed memory



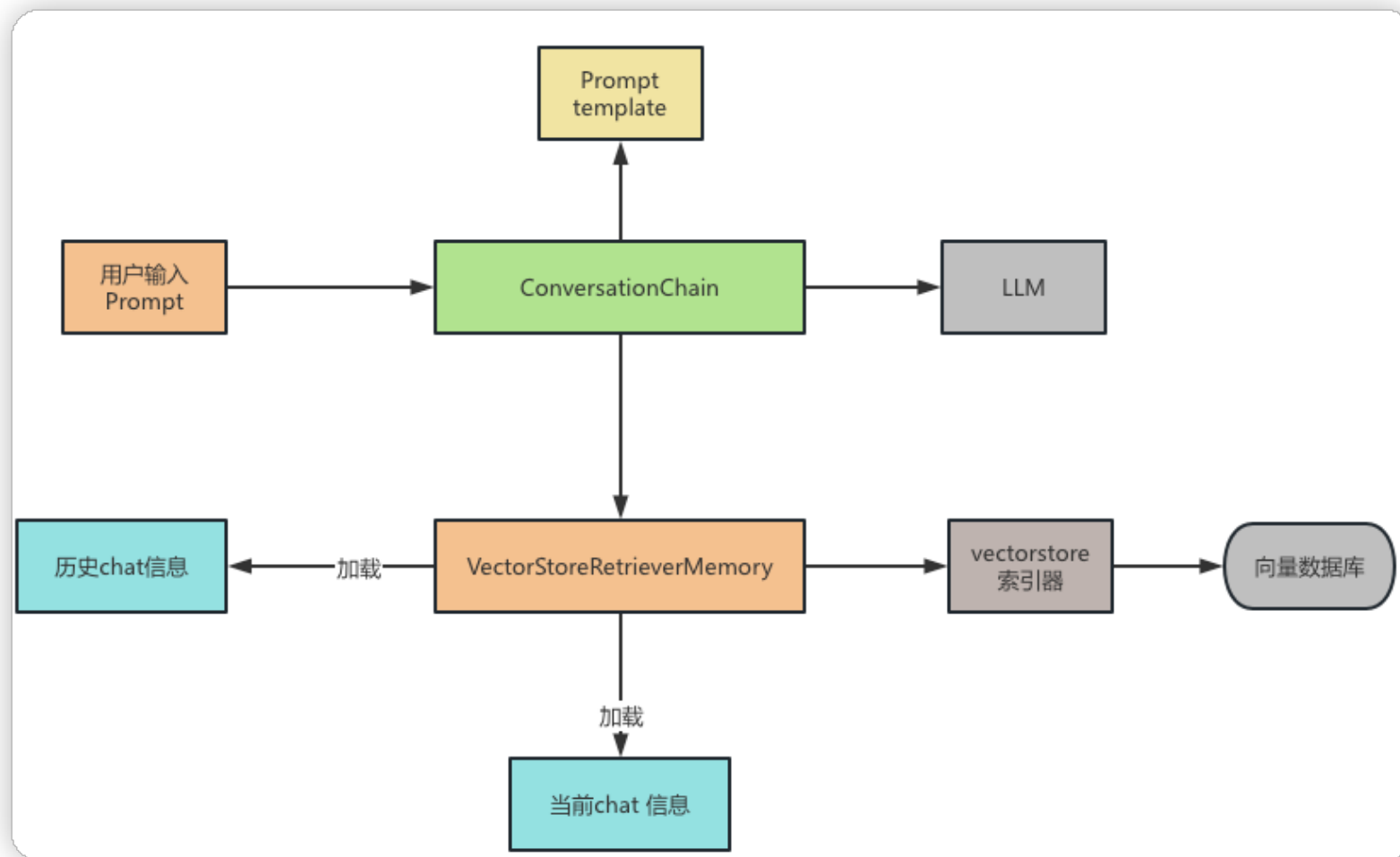
ConversationSummaryMemory

ConversationSummaryMemory在对话发生时对其进行总结，并将当前的**总结存储在内存中**。然后，可以使用这个记忆来将迄今为止的对话总结注入到提示/链中。对于较长的对话，这种记忆最有用，因为在提示中逐字保留过去的信息历史会占用太多的令牌。

Conversation Summary Buffer Memory

通过设置max_token_limit字段判断记忆上下文的长度。

Vector store-backed memory



6-3 Memory 常见用法 – 上

- 客制化对话前缀
- 多输入Memory



客制化对话前缀

对话中，可以自己定说话者的身份标签。

例如：将人类输入的标签定义为：人类；将AI 回应的标签定义为AI助手。

"AI Prefix"和"Human Prefix"都是在对话的上下文中表示每条对话的发起者身份的标签。在对话的上下文中，AI的回复通常会以"AI:"作为前缀，人类的输入通常会以"Human:"作为前缀。通过这些前缀，我们可以清晰地看到对话的流程和每条对话的发起者。

默认情况下，"AI Prefix"设置为"AI"，"Human Prefix"设置为"Human"，这些都可以被自定义为你想要的任何标签。例如，你可以将"AI Prefix"更改为"AI Assistant"，将"Human Prefix"更改为"Friend"，然后在对话中，每条由AI生成的回复都会以"AI Assistant:"作为前缀，每条由人类输入的对话都会以"Friend:"作为前缀。

多输入Memory

处理来自多个源头的Memory，例如：文档+输入。这就是两个源头，在实际应用中，对话可能会调用外部的数据源获取数据，此时Memory需要将对话内容以及外部数据源搜索到的信息一并保存。本例的数据源来自于txt 文件

主要目的是为处理多输入的链条（Chain）添加记忆功能。大多数记忆对象假设有一个单一的输入，但在这个例子中，我们处理了一个包含多个输入（相关文档和用户问题）的问题/回答链。

6-4 Memory 常见用法 – 下

- 实体记忆 (Entity Memory)
- 对话知识图谱记忆



实体记忆 (Entity Memory)

实体记忆 (Entity Memory) 在处理长期对话或者需要持续跟踪特定**实体信息**。

在一个持续的对话中，人类用户可能会提供有关某些实体（如**人名**、**地点**、**事件**等）的信息，这些信息可能会在后续的对话中被引用或查询。如果不使用实体记忆，AI可能会忘记或无法适当地处理这些在对话中提到的实体的信息，这可能会导致对话的连贯性和用户体验的降低。

例如，如果用户告诉AI，他们的狗名叫Max，那么在后续的对话中，当用户提到Max时，AI应该能够记住Max是用户的狗。如果没有实体记忆，AI可能会忘记这个信息，或者需要用户再次提供这个信息。

对话知识图谱记忆

“对话知识图谱记忆(Conversation Knowledge Graph Memory)”是一种使用知识图谱来复原记忆的方法。**记忆的关键是实体与实体之间的关系。**

首先，我们可以通过保存对话中的输入和输出来建立知识图谱。当我们询问某个实体，例如"Sam"，该方法可以返回关于该实体的历史信息，如"Sam是朋友"。

此外，这个方法还可以从新的消息中获取当前实体或知识三元组，这些都会在前面的消息的上下文中使用。例如，当我们询问"Sam的最爱颜色是什么？"，它会返回实体"Sam"。当我们说"她最喜欢的颜色是红色"时，它会返回知识三元组，表示"Sam"的"最喜欢的颜色"是"红色"。

本章课程介绍

- Memory种类

Entity memory

- Conversation buffer memory
- Conversation buffer window memory
- Conversation Summary memory
- Conversation Summary Buffer Memory
- Vector store-backed memory

- 如何为多输入链添加内存
- 如何定制对话Memory
- 会话知识图谱