

大模型应用开发框架 LangChain (中)

彭靖田

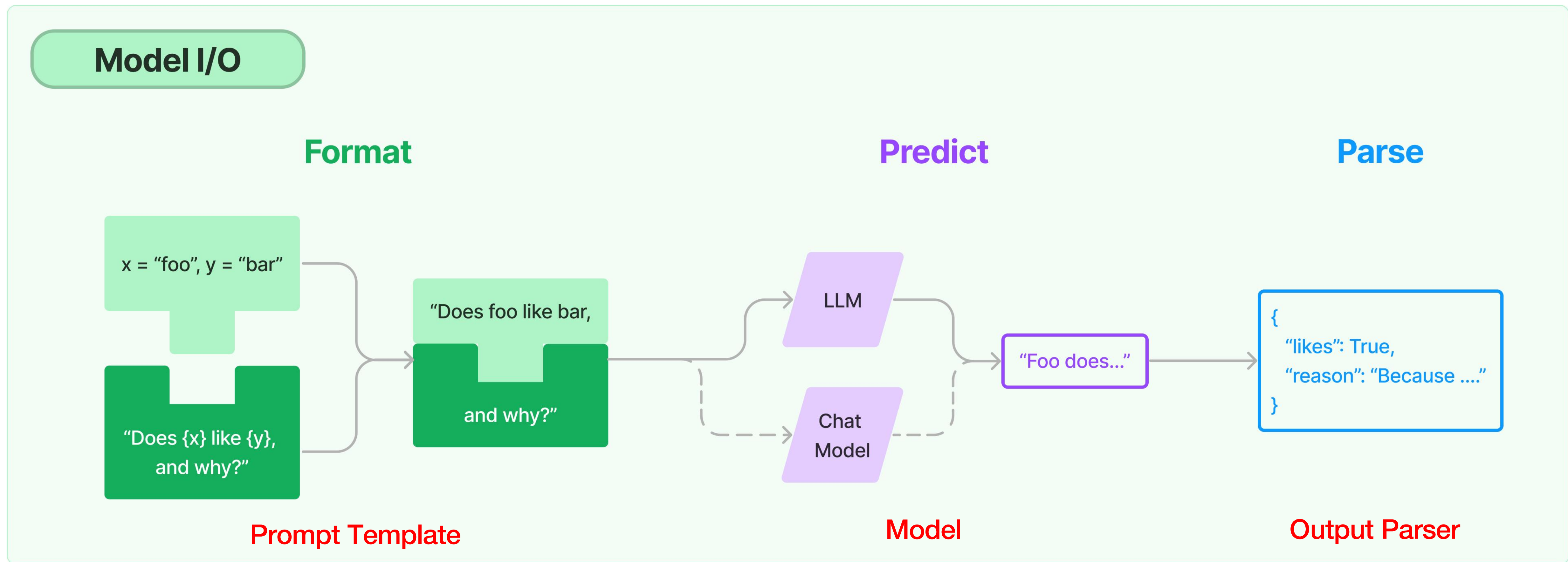
Google Developer Experts
LangChain Developer

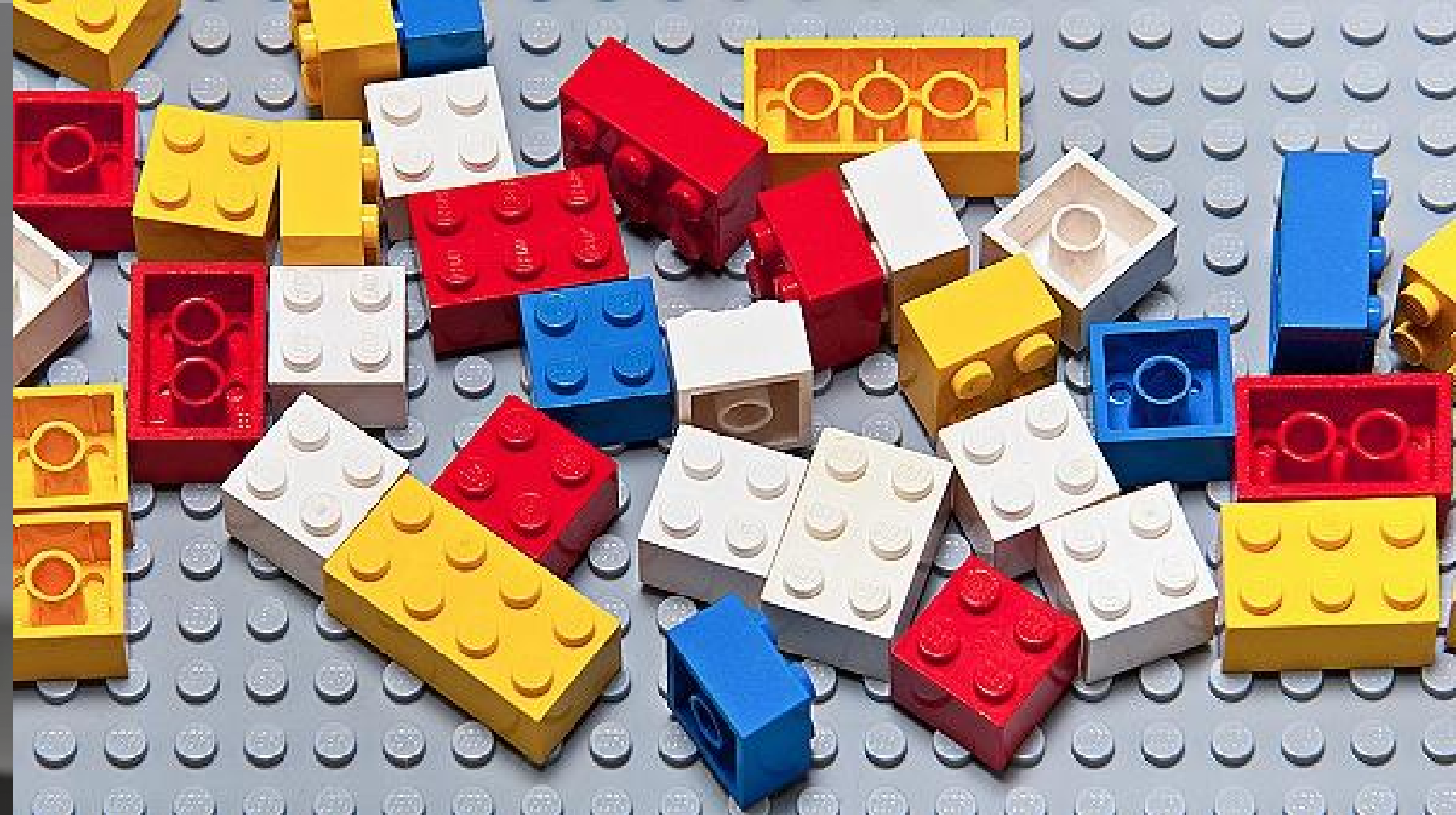


目录

- 大模型应用的最佳实践 **Chains**
 - 上手你的第一个链: **LLM Chain**
 - 串联式编排调用链: **Sequential Chain**
 - 处理超长文本的转换链: **Transform Chain**
 - 实现条件判断的路由链: **Router Chain**
- 赋予应用记忆的能力: **Memory**
 - **Memory System** 与 **Chain** 的关系
 - 记忆基类 **BaseMemory** 与 **BaseChatMessageMemory**
 - 服务聊天对话的记忆系统
 - **ConversationBufferMemory**
 - **ConversationBufferWindowMemory**
 - **ConversationSummaryBufferMemory**

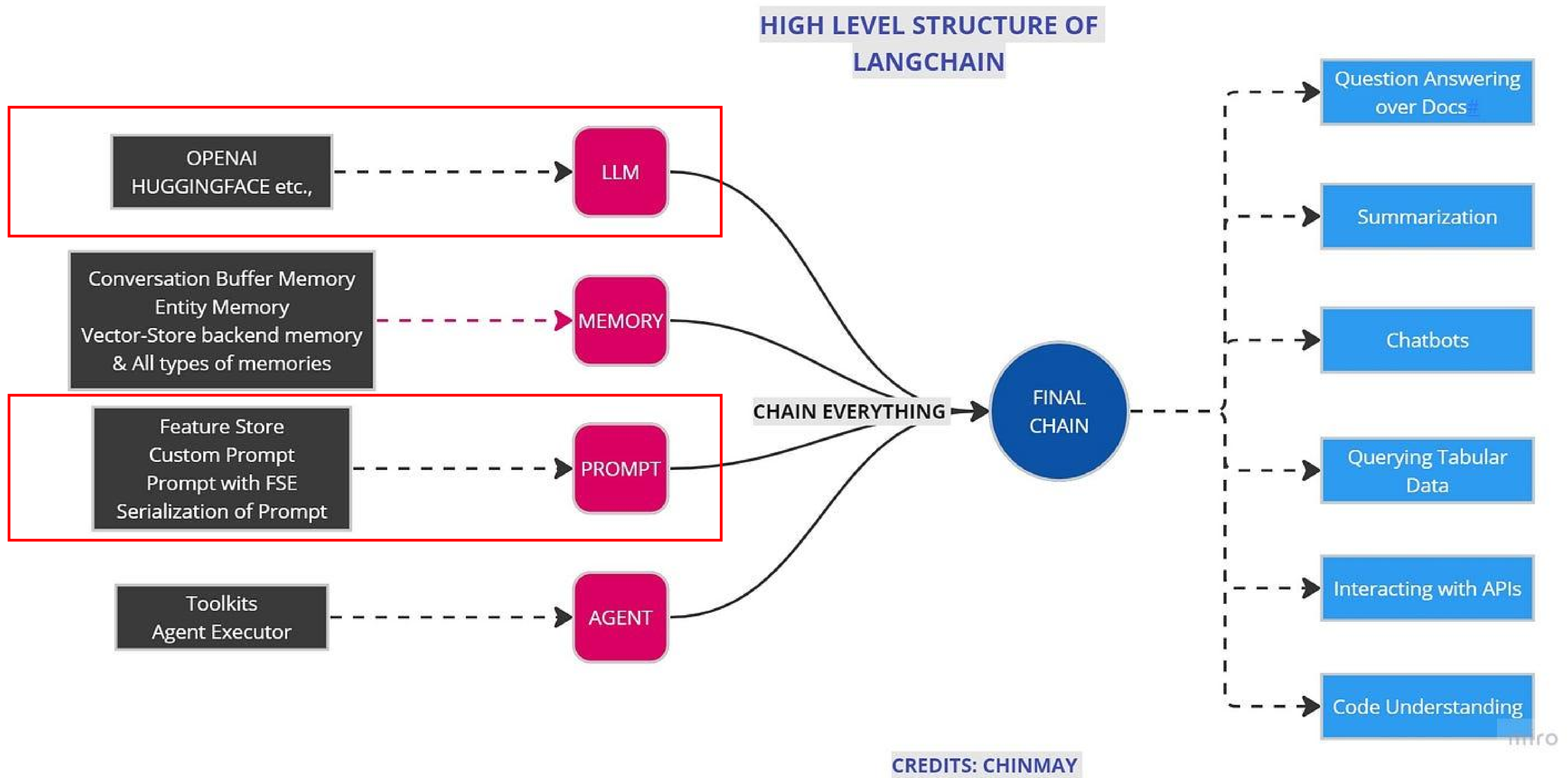
Chains



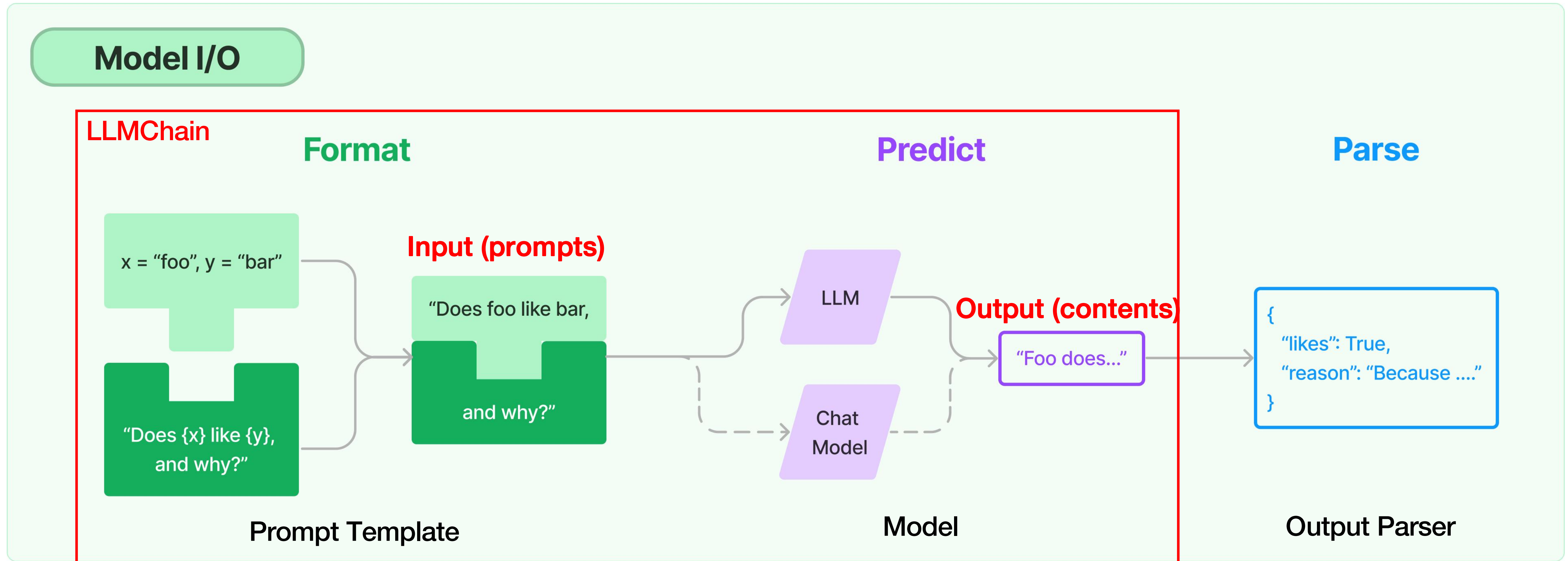


他山之石：乐高式创新

Chains: 大模型应用的最佳实践



LLMChain: 整合语言模型和提示模板的最简单链



LLMChain 公司取名

```
[18]: from langchain.llms import OpenAI
      from langchain.prompts import PromptTemplate

      llm = OpenAI(temperature=0.9, max_tokens=500)
```

```
[19]: prompt = PromptTemplate(
      input_variables=["product"],
      template="给制造{product}的有限公司取10个好名字, 并给出完整的公司名称",
      )
```

```
[20]: from langchain.chains import LLMChain

      chain = LLMChain(llm=llm, prompt=prompt)
      print(chain.run({
          'product': "性能卓越的GPU"
      })))
```

1. Hyperion High-End GPU Company
2. CyberGPU Ingenuity Technologies
3. NeoTronix Advanced Graphics Inc.
4. FusionWorks Graphics Systems
5. CryptoCore Creative Computing
6. Levai Visual Solutions
7. Quanta Vision Technologies
8. Catalyst Technologies International
9. ZenCore Graphic Innovations
10. ViZoX Precision Graphics LLC

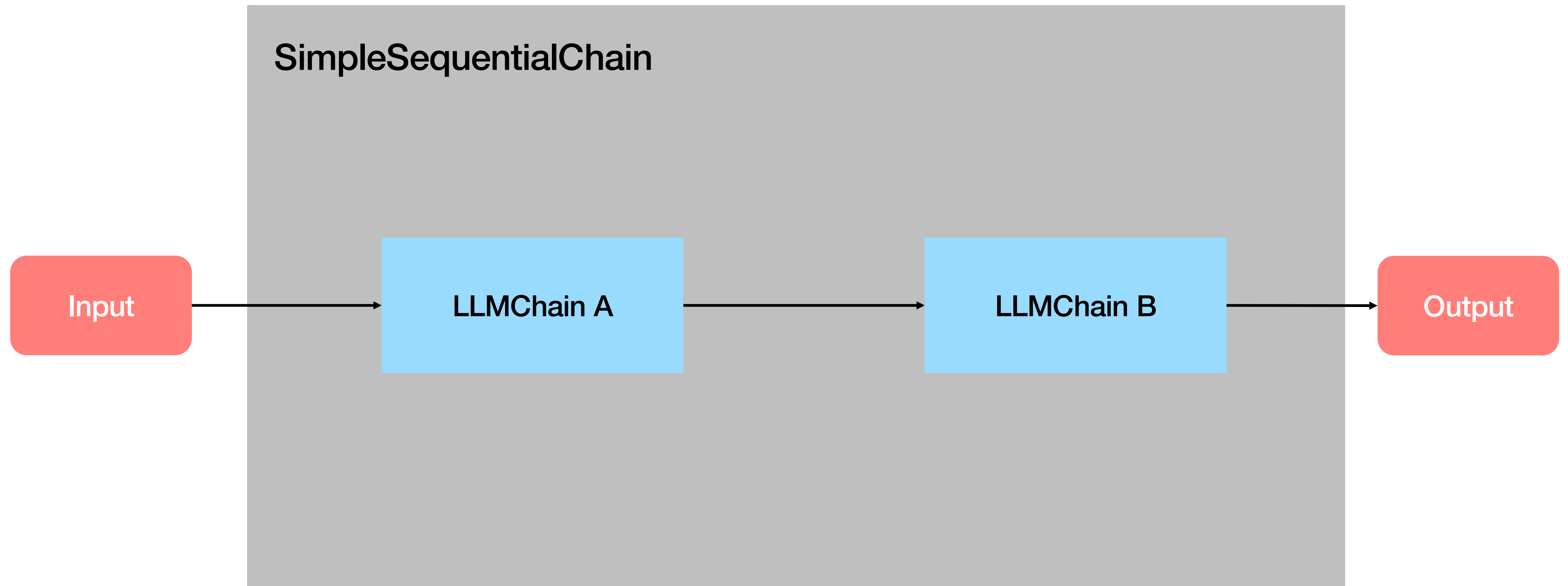
SequentialChain: 串联式调用语言模型链

串联式调用语言模型（将一个调用的输出作为另一个调用的输入）。

顺序链（**Sequential Chain**）允许用户连接多个链并将它们组合成执行特定场景的流水线（**Pipeline**）。有两种类型的顺序链：

- **SimpleSequentialChain**: 最简单形式的顺序链，每个步骤都具有单一输入/输出，并且一个步骤的输出是下一个步骤的输入。
- **SequentialChain**: 更通用形式的顺序链，允许多个输入/输出。

SimpleSequentialChain: 单输入/输出串联调用



LLMChain 实战: synopsis_chain 根据剧目的标题撰写简介

synopsis_chain

input_variables

title = “ 三体人不是无法战胜的”

prompts

你是一位剧作家。根据戏剧的标题，
你的任务是为该标题写一个简介。

标题: {title}
剧作家: 以下是对上述戏剧的简介:

Prompt Template

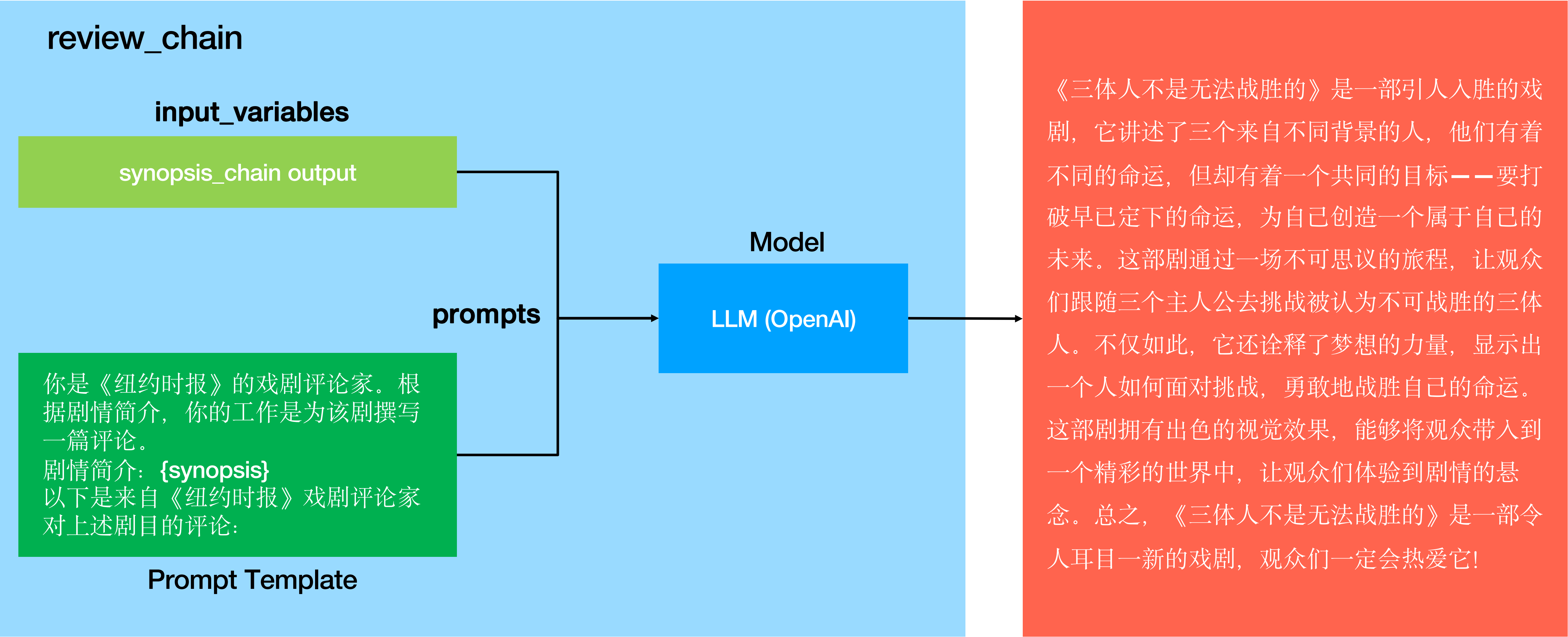
Model

LLM (OpenAI)

Output (synopsis)

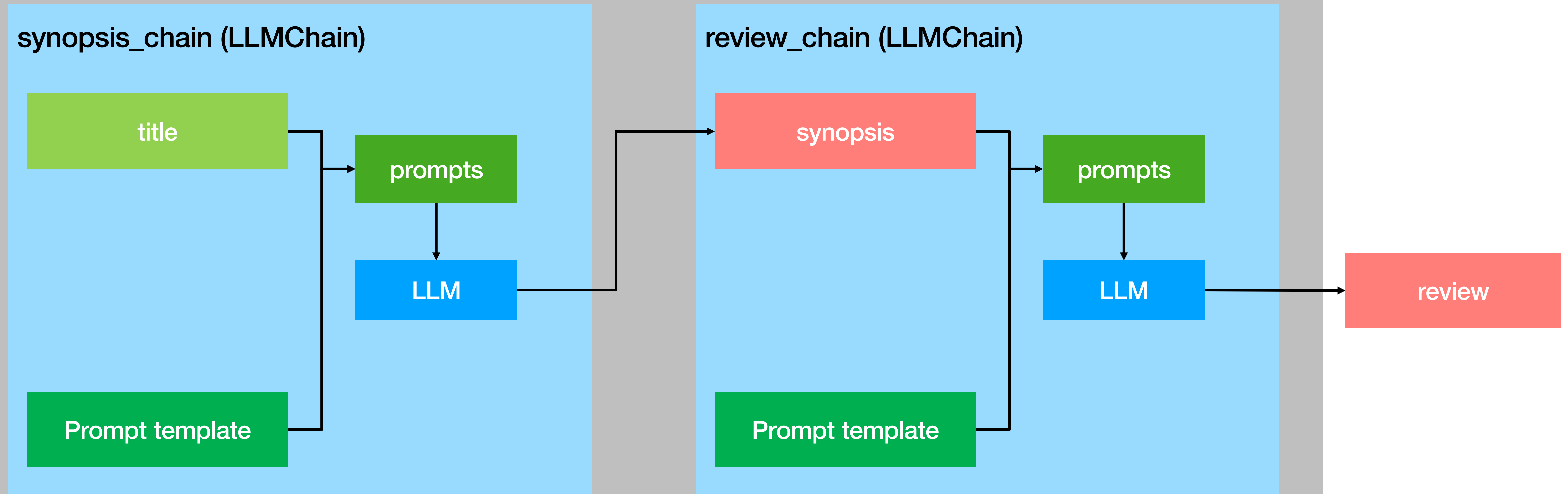
《三体人不是无法战胜的》讲述了一个古老而又充满悬念的故事，讲述了三个被当作宿命的人之间的奋斗。它们是不同的背景也有不同命运的人，但他们共同的目标是要打破早已定下的命运，为自己创造一个属于自己的未来。他们踏上了一场不可思议的旅程，去挑战被认为不可战胜的三体人。他们在这场旅程中面临着各种挑战，但最终他们能否战胜三体人，改变命运，实现自己的梦想？让我们一起见证这场精彩的故事发展！

Output (review)



SequentialChain: 语言模型串联式调用链

SimpleSequentialChain



SimpleSequentialChain 戏剧评论

```
[52]: # 这是一个SimpleSequentialChain, 按顺序运行这两个链

from langchain.chains import SimpleSequentialChain
overall_chain = SimpleSequentialChain(chains=[synopsis_chain, review_chain], verbose=True)

[54]: review = overall_chain.run("三体人不是无法战胜的")
```

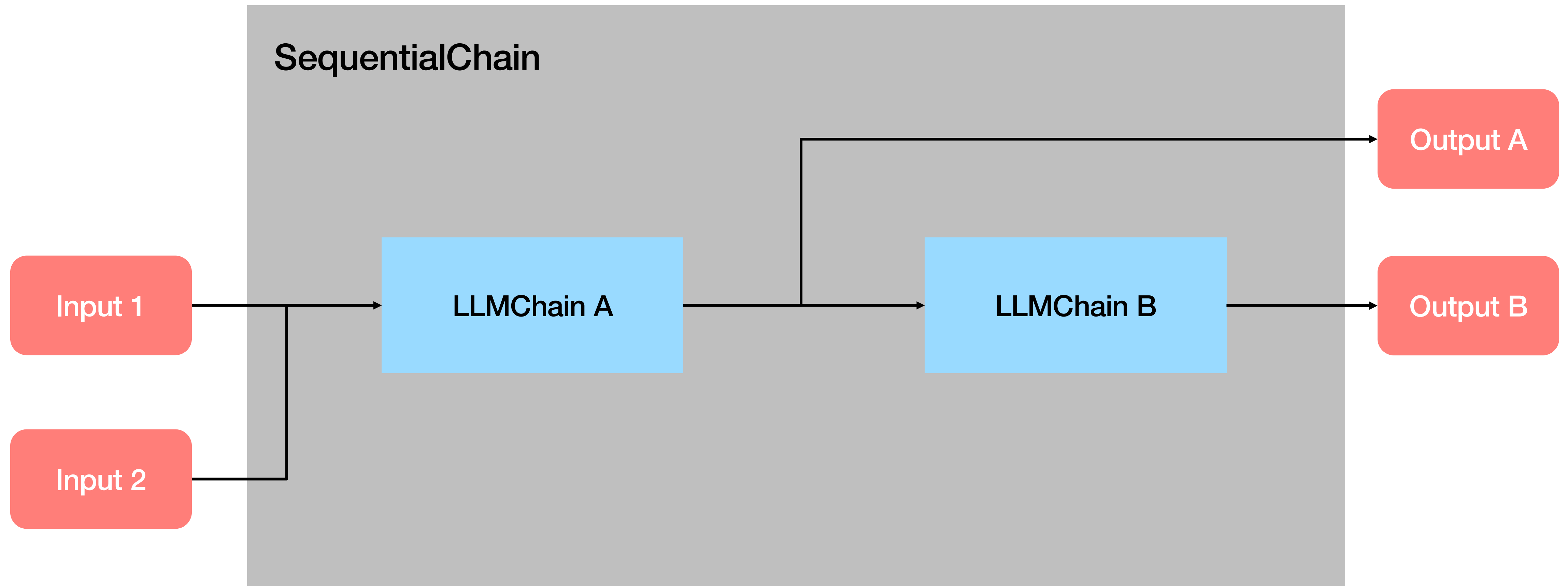
> Entering new SimpleSequentialChain chain...

《三体人不是无法战胜的》讲述了一个古老而又充满悬念的故事，讲述了三个被当作宿命的人之间的奋斗。它们是不同的背景也有不同命运的人，但他们共同的目标是要打破早已定下的命运，为自己创造一个属于自己的未来。他们踏上了一场不可思议的旅程，去挑战被认为不可战胜的三体人。他们在这场旅程中面临着各种挑战，但最终他们能否战胜三体人，改变命运，实现自己的梦想？让我们一起见证这场精彩的故事发展！

《三体人不是无法战胜的》是一部引人入胜的戏剧，它讲述了三个来自不同背景的人，他们有着不同的命运，但却有着一个共同的目标——要打破早已定下的命运，为自己创造一个属于自己的未来。这部剧通过一场不可思议的旅程，让观众们跟随三个主人公去挑战被认为不可战胜的三体人。不仅如此，它还诠释了梦想的力量，显示出一个人如何面对挑战，勇敢地战胜自己的命运。这部剧拥有出色的视觉效果，能够将观众带入到一个精彩的世界中，让观众们体验到剧情的悬念。总之，《三体人不是无法战胜的》是一部令人耳目一新的戏剧，观众们一定会热爱它！

> Finished chain.

SequentialChain: 多输入/输出串联调用



Sequential Chain 戏剧评论

```
[67]: from langchain.chains import SequentialChain
```

```
m_overall_chain = SequentialChain(  
    chains=[synopsis_chain, review_chain],  
    input_variables=["era", "title"],  
    # Here we return multiple variables  
    output_variables=["synopsis", "review"],  
    verbose=True)
```

```
[68]: m_overall_chain({"title": "三体人不是无法战胜的", "era": "二十一世纪的新中国"})
```



> Entering new SequentialChain chain...

> Entering new LLMChain chain...

Prompt after formatting:

你是一位剧作家。根据戏剧的标题和设定的时代，你的任务是为该标题写一个简介。

标题：三体人不是无法战胜的

时代：二十一世纪的新中国

剧作家：以下是对上述戏剧的简介：

> Finished chain.

> Entering new LLMChain chain...

Prompt after formatting:

你是《纽约时报》的戏剧评论家。根据该剧的剧情简介，你需要撰写一篇关于该剧的评论。

剧情简介：

《三体人不是无法战胜的》是一部关于新中国二十一世纪的戏剧。故事描述一个普通家庭中的三体人，他们受到传统的社会和政治压力，必须解决他们自己的问题以及社会上出现的问题。他们不得不面对社会的歧视与偏见，以及政治压迫和控制，却又一直坚持自己的价值观和信念

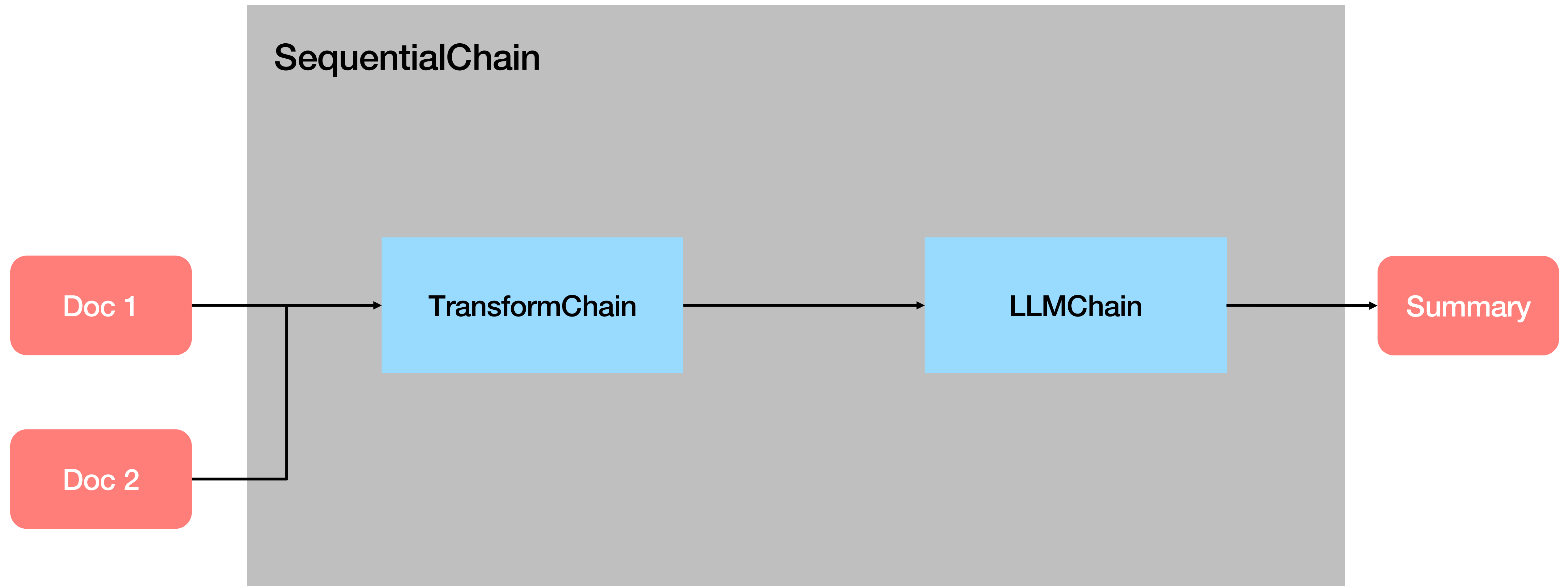
来自《纽约时报》戏剧评论家对上述剧目的评价：

> Finished chain.

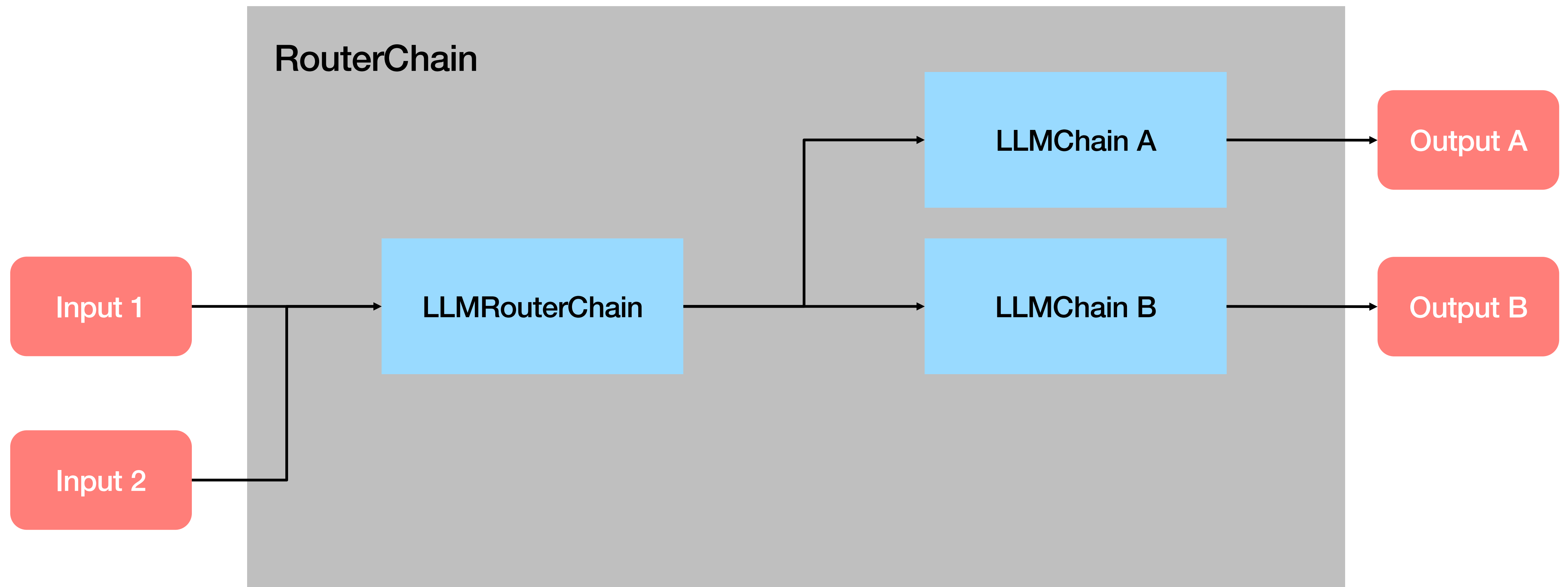
> Finished chain.

```
[68]: {'title': '三体人不是无法战胜的',  
      'era': '二十一世纪的新中国',  
      'synopsis': '\n\n《三体人不是无法战胜的》是一部关于新中国二十一世纪的戏剧。故事描述一个普通家庭中的三体人，他们受到传统的社会和政治压力，必须解决他们自己的问题以及社会上出现的问题。他们不得不面对社会的歧视与偏见，以及政治压迫和控制，却又一直坚持自己的价值观和信念',  
      'review': '\n\n《三体人不是无法战胜的》是一部非常有趣且有意义的戏剧。它既有趣又充满活力，讲述的是一群三体人生活在新中国二十一世纪的社会中所遇到的困难。他们尝试在传统的社会和政治压力下依然坚持自己的价值观和信念。这部剧让我们看到了社会歧视和偏见，以及政治压迫和'}
```

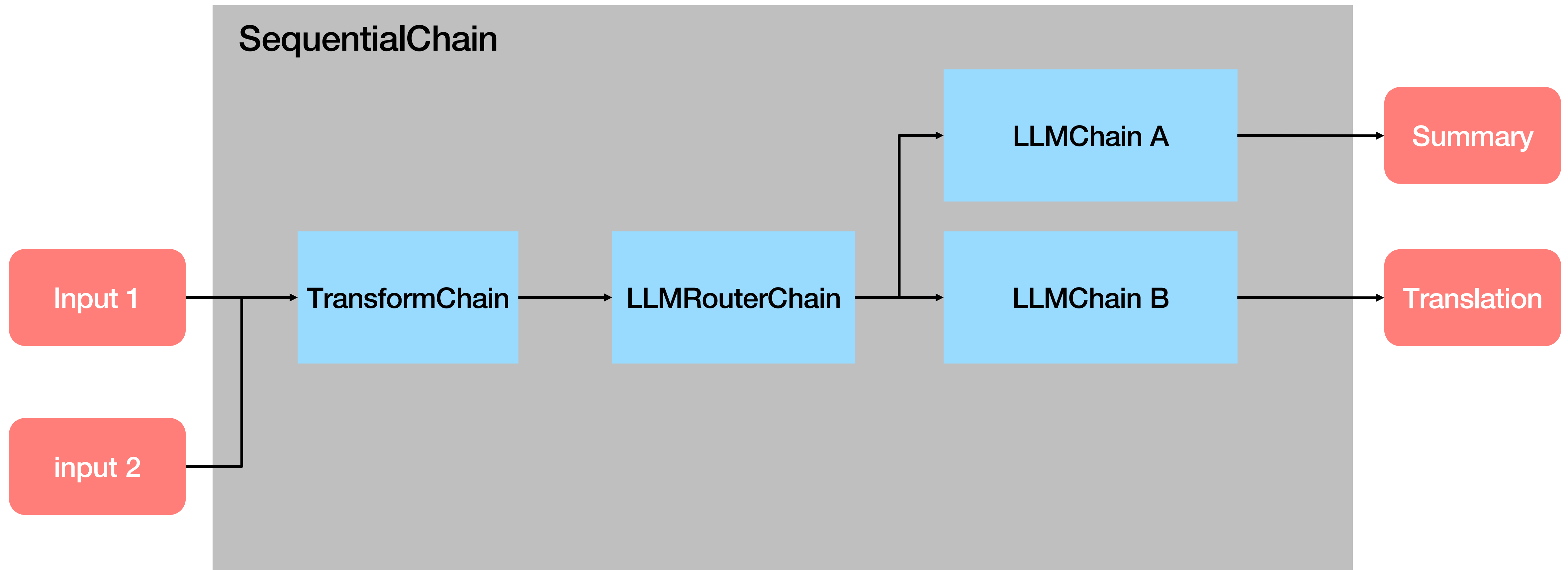

TransformChain: 处理超长文本的转换链



RouterChain: 实现条件判断的大模型调用



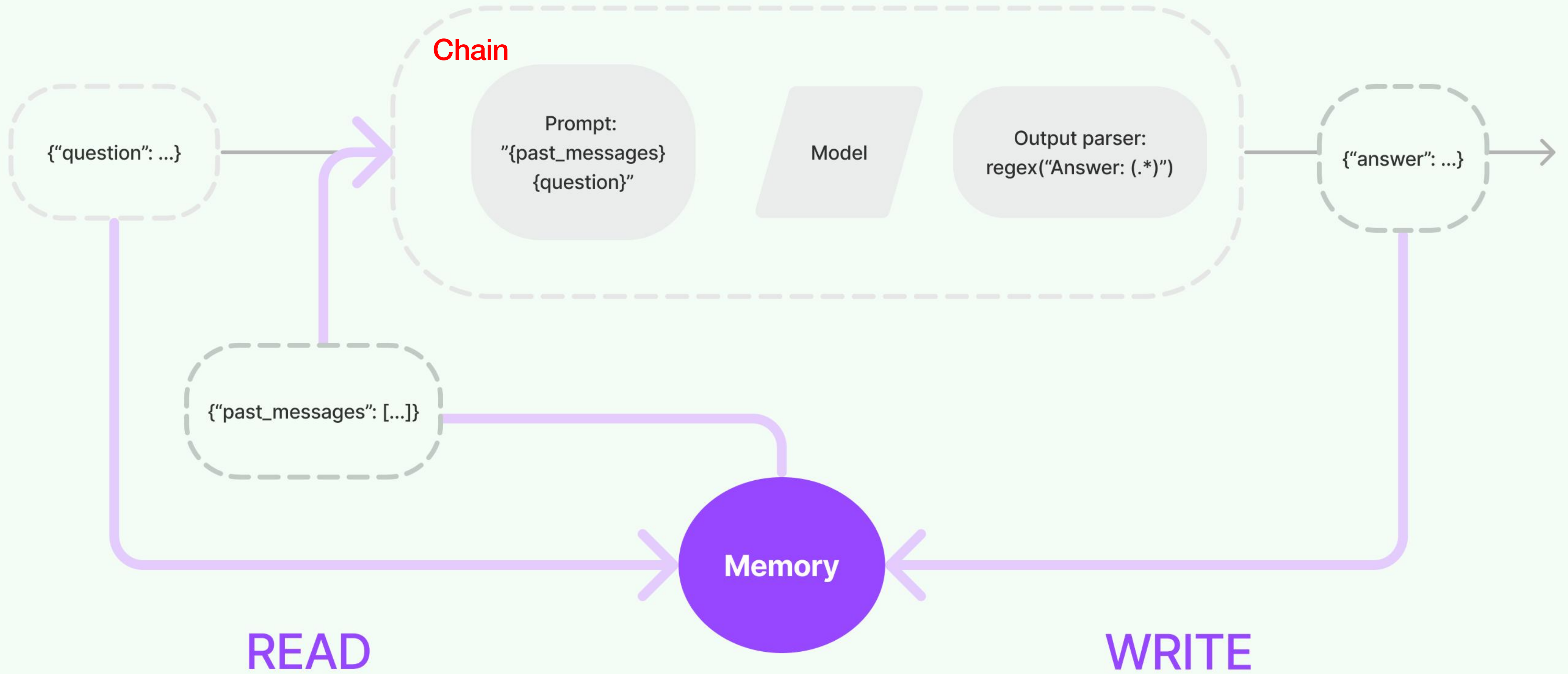
[Optional] Homework: 实现文本摘要和翻译聊天助手



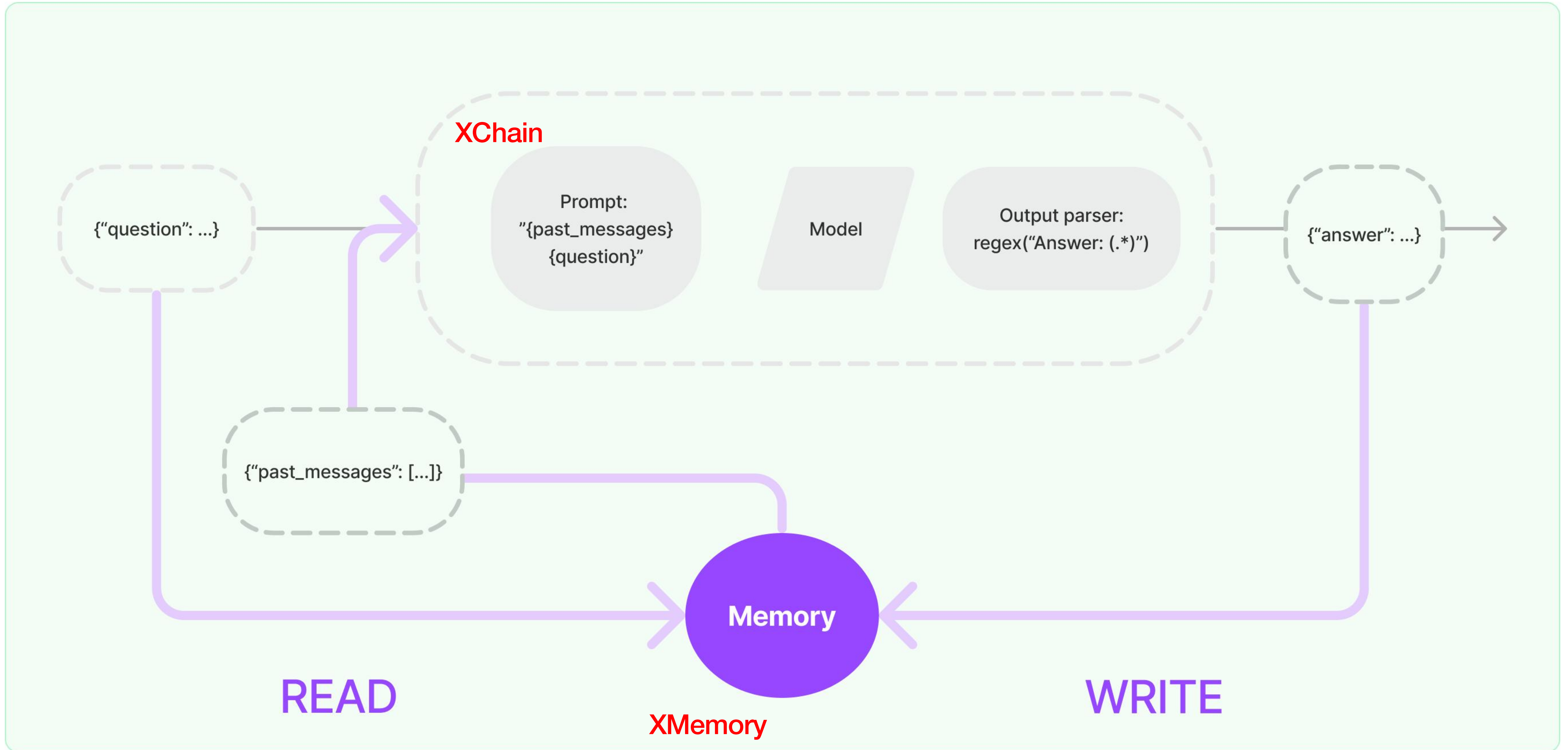
DEMO

Memory

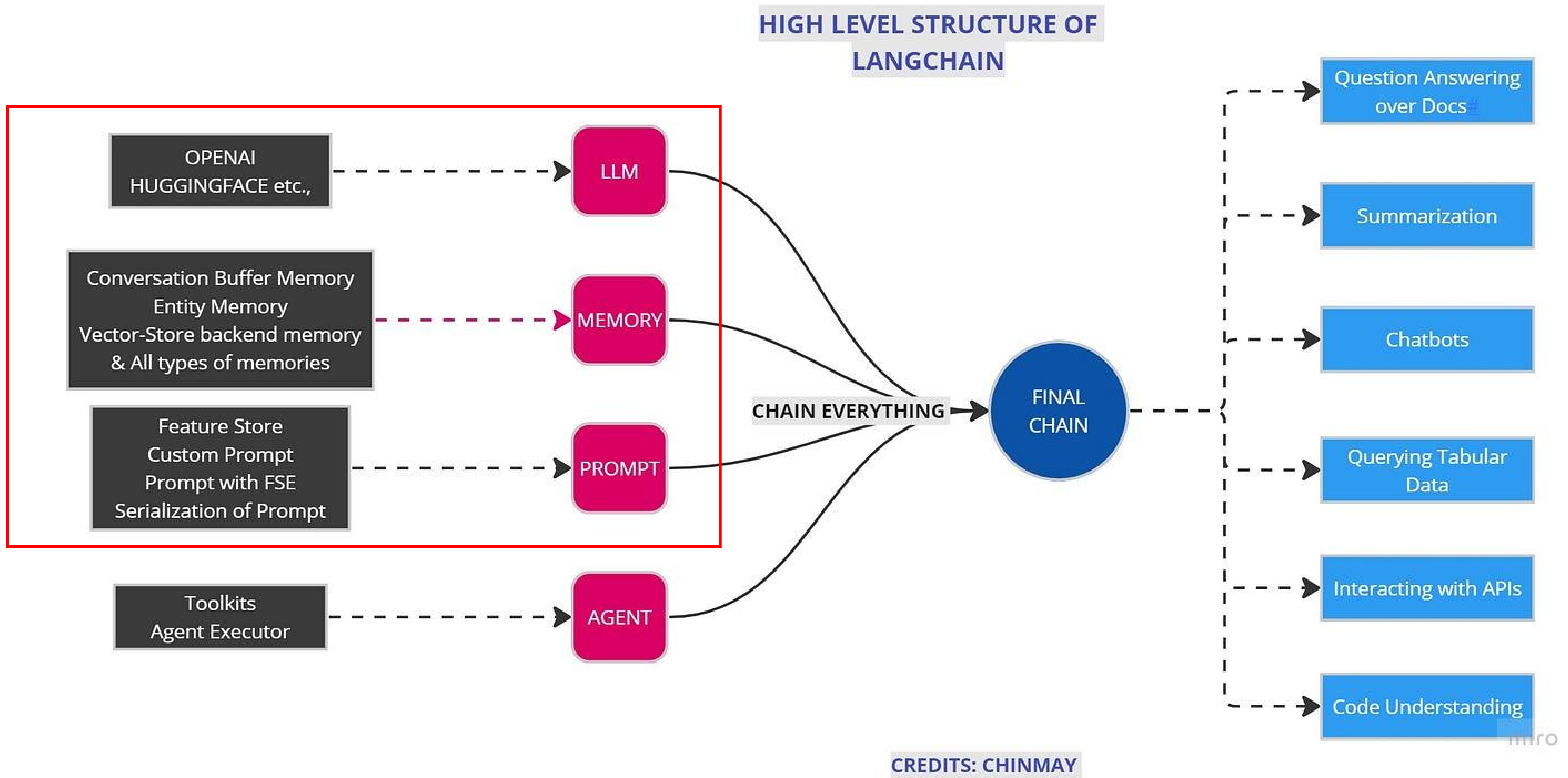
Memory System



Memory System: 高度定制化



Chains + Memory: 有记忆能力的大模型应用



DEMO

课程项目：GitHub openai-quickstart

openai-quickstartPublic

UnpinUnwatch10Fork145Starred306

main1 branch0 tagsGo to fileAdd fileCode

DjangoPeng Merge branch 'main' of github.com:DjangoPeng/openai-quickstart int...6bd7e132 minutes ago57 commits

chatgpt-plugins	add initial version of chatgpt plugins	2 minutes ago
docs	make layout more readable	last week
openai-translator	Merge branch 'main' of github.com:DjangoPeng/openai-quickstart in...	2 minutes ago
openai_api	Remove fine_food_reviews_with_embeddings_1k_demo.csv	last week
.gitignore	update embedding demo	2 weeks ago
LICENSE	Initial commit	2 weeks ago
README-CN.md	fix: correct file path of pdfplumber.ipynb on readme	yesterday
README.md	fix: correct file path of pdfplumber.ipynb on readme	yesterday

README.md

OpenAI Quickstart

English | 中文

This project is designed as a one-stop learning resource for anyone interested in large language models and their application in Artificial Intelligence Governance and Control (AIGC) scenarios. By providing theoretical foundations, development basics, and hands-on examples, this project offers comprehensive guidance on these cutting-edge topics.

Features

About

A comprehensive guide to understanding and implementing large language models with hands-on examples using LangChain for AIGC applications.

- Readme
- Apache-2.0 license
- Activity
- 306 stars
- 10 watching
- 145 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Contributors8

<https://github.com/DjangoPeng/openai-quickstart>