**AI 助力智能工作流与标书创作指南**

目录

[序章 1](#_Toc184222488)

[1. Dify 平台介绍 1](#_Toc184222489)

[1.1. 什么是 Dify？ 1](#_Toc184222490)

[1.2. 核心功能概述 2](#_Toc184222491)

[1.3. 与其他平台的对比优势 2](#_Toc184222492)

[1.4. 使用Dify平台对比直接使用 ChatGPT 3](#_Toc184222493)

[1.5. Dify、Coze、FastGPT对比 4](#_Toc184222494)

[2. 搭建聊天助手操作手册 7](#_Toc184222495)

[2.1. 为什么需要开发聊天助手？ 7](#_Toc184222496)

[2.2. 如何使用 Dify 搭建智能聊天助手 7](#_Toc184222497)

[3. 工作流与长篇文章生成指南 18](#_Toc184222498)

[3.1.生成长篇文章的步骤 18](#_Toc184222499)

[4. Dify工作流在技术标书生成中的应用 25](#_Toc184222500)

[4.1.整体思路 25](#_Toc184222501)

[4.2.详细步骤 26](#_Toc184222502)

[5.关键词与术语解析 29](#_Toc184222503)

[5.1.什么是生成式？ 29](#_Toc184222504)

[5.2.什么是RAG(检索增强生成) 29](#_Toc184222505)

[5.3.什么是智能体(Agent) 30](#_Toc184222506)

[5.4.什么是工作流(WorkFlow) 30](#_Toc184222507)

[5.5.什么是大模型 30](#_Toc184222508)

[5.6.什么是ChatGPT 31](#_Toc184222509)

[5.7.GPT与大模型的关系 32](#_Toc184222510)

[5.8.ChatGPT、GPT与大模型三者的关系 32](#_Toc184222511)

[5.9.开源大模型举例 32](#_Toc184222512)

[5.10.OpenAI API 33](#_Toc184222513)

[5.11.OpenAI付费方式的区别 34](#_Toc184222514)

[5.12.其他 34](#_Toc184222515)

## 序章

随着人工智能技术的快速发展，AI工具已经逐渐成为日常工作中的得力助手。Dify 作为一款高效的 AI 平台，为用户提供了构建智能聊天助手、生成长篇文章、优化工作流等多样化功能。本操作手册旨在帮助用户快速上手 Dify，掌握其核心功能，从而在日常工作中提升效率和质量。

无论您是初次接触 Dify，还是希望挖掘更高级的使用技巧，本手册都将为您提供清晰、易懂的指导，助您快速成为 AI 应用的专家。

本操作手册包含Dify平台介绍、搭建AI聊天助手平台、工作流与长篇文章生成指南、**Dify工作流在技术标书生成中的应用、关键词术语解释五个章节。**

Dify平台工作台目前疑似被防火墙隔离，建议使用VPN进行访问。

## ****1. Dify 平台介绍****

### 1.1. 什么是 Dify？

Dify 是一款专为开发 AI 应用设计的开发框架。类似于 Java 中的 Spring Boot，后者被广泛应用于 Web 应用开发。使用开发框架的核心目标是提高开发效率，而 Dify 正是为此而生。

更多介绍详见官网：https://dify.ai/zh

### 1.2. 核心功能概述

作为一个 AI 应用开发框架，Dify 当前支持以下四种主要应用类型：

1. **聊天助手**：基于大语言模型（LLM）构建对话式交互的智能助手。
2. **文本生成**：适用于撰写故事、文本分类、翻译等任务。
3. **Agent**：能够进行任务分解、逻辑推理和工具调用的智能助手。
4. **工作流**：通过流程编排定义灵活的 LLM 工作流。

### 1.3. 与其他平台的对比优势

Dify 的社区版是开源的，代码托管在 GitHub 上，体现了其开放性。此外，Dify 既支持在线云服务访问，也支持私有化部署，这使其在满足不同场景需求时具备高度的灵活性。

**简单且快速** 是选择 Dify 的关键原因。

* **非程序员友好**  
  即使没有编程经验，也可以通过零代码和可视化方式快速构建 AI 应用。
* **程序员效率提升**  
  与 LangChain 等库相比，Dify 提供了更贴近生产需求的完整解决方案，经过精心设计和测试，相当于一套高效的脚手架。

### 1.4. 使用Dify平台对比直接使用 ChatGPT

|  |  |  |
| --- | --- | --- |
| **角度** | **Dify** | **OpenAI 直接使用 ChatGPT** |
| **数据隐私** | Dify 支持本地私有化部署，支持调用本地训练的大模型，但是如果调用OpenAI的APIkey同样有数据泄露风险，安全性上不如国产AI百川大模型 | 数据处理全部托管在 OpenAI 云端，可能带来隐私和合规风险，尤其是在涉及严格数据保护法规的地区，部分外企有控诉过OpenAI使用泄露数据进行训练 |
| **定制化** | 可以根据具体业务场景进行模型调优、参数调整和功能扩展（workflow），能够满足复杂、个性化的需求。 | 默认模型提供通用化服务，适合大多数标准场景。虽然开箱即用，但灵活性和定制化能力相对有限，无法满足特定需求场景的优化。 |
| **成本** | Dify官方价格：https://dify.ai/pricing， 或者采用开源方案（调用OpenAI的API或者本地大模型），长期来看可以降低成本，特别是对高频使用的企业。然而，初始部署需要一定的硬件投入和技术团队支持 | 按需付费模式，初期无硬件投入，适合快速上线和临时使用的场景。然而，如果长期使用或调用频率较高，可能会导致成本累积增加。 |
| **便捷性** | 目前工作台同样依赖于网络环境，需要进项简单配置 | 无需部署，直接通过 API 或者注册即可使用，初期投入少且易于上手，适合技术能力较弱的团队或需要快速实现产品功能的企业。但其依赖于网络环境，可能对使用场景有一定限制。 |
| **性能** | 取决于调用的模型性能 | ChatGPT-4 是 OpenAI 开发的第四代大语言模型，在自然语言处理领域处于领先地位 |

### ****1.5. Dify、Coze、FastGPT对比****

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Dify** | **FastGPT** | **Coze** |
| **简介** | Dify是一个开源的平台，支持本地私有部署 | FastGPT是一个开源的平台， 提供开箱即用的数据处理、 模型调用等能力 | 字节跳动旗下的产品，Coze提供了丰富的插件系统和易用性 |
| **开源** | Dify的开源社区活跃，提供了丰富的资源 | 开源 | 不开源 |
| **定制性** | 用户可以根据自己需求进行高度定制， 并且可以在自己的服务器上进行部署， 从而更好的控制数据安全和隐私 | 可以通过Flow可视化进行工作流编排，从而实现复杂问答场景 | 相比Dify，定制性有限制 |
| **大模型支持** | 支持多种大语言模型， 如OpenAI、Claude3等， 也支持开源的本地模型 | 通过One API支持常见的大模型 | 支持少数的在线模型 |
| **便捷性** | 操作便捷 更加容易上手 可以通过零代码和可视化方式快速构建 AI 应用 | 操作便捷 | 操作便捷 |
| **功能支持** | 支持RAG 支持Agent，但不支持多Agent 支持工作流 | 擅长支持RAG场景 支持工作流 | RAG场景支持较弱 Agent功能丰富 支持工作流 |
| **面向用户** | 企业用户 程序员 | 企业用户 程序员 | C端用户 |

## ****2. 搭建聊天助手操作手册****

### **2.1. 为什么需要开发聊天助手？**

虽然市场上已有 ChatGPT 和文心一言等聊天助手，但开发自有聊天助手仍然具有以下必要性：

1. **定制化需求**  
   如果需要在自己的网站或应用中集成聊天助手，仅嵌入现有产品页面（OpenAI官方的ChatGPT、文心一言等）并不现实，尤其当登录体系和现有系统不一致时。
2. **嵌入公司知识库**  
   通用聊天助手无法回答特定领域或公司内容。例如，你需要一个能解答公司产品说明书的助手，而现有助手并未接受相关培训。这时，定制开发显得尤为必要。通过Dify提供的知识库即可实现（在Dify中，知识库是一些文档的集合，可以被整体集成至一个应用中作为检索上下文使用）。

### 2.2. 如何使用 Dify 搭建智能聊天助手

#### 2.2.1. 获取API key

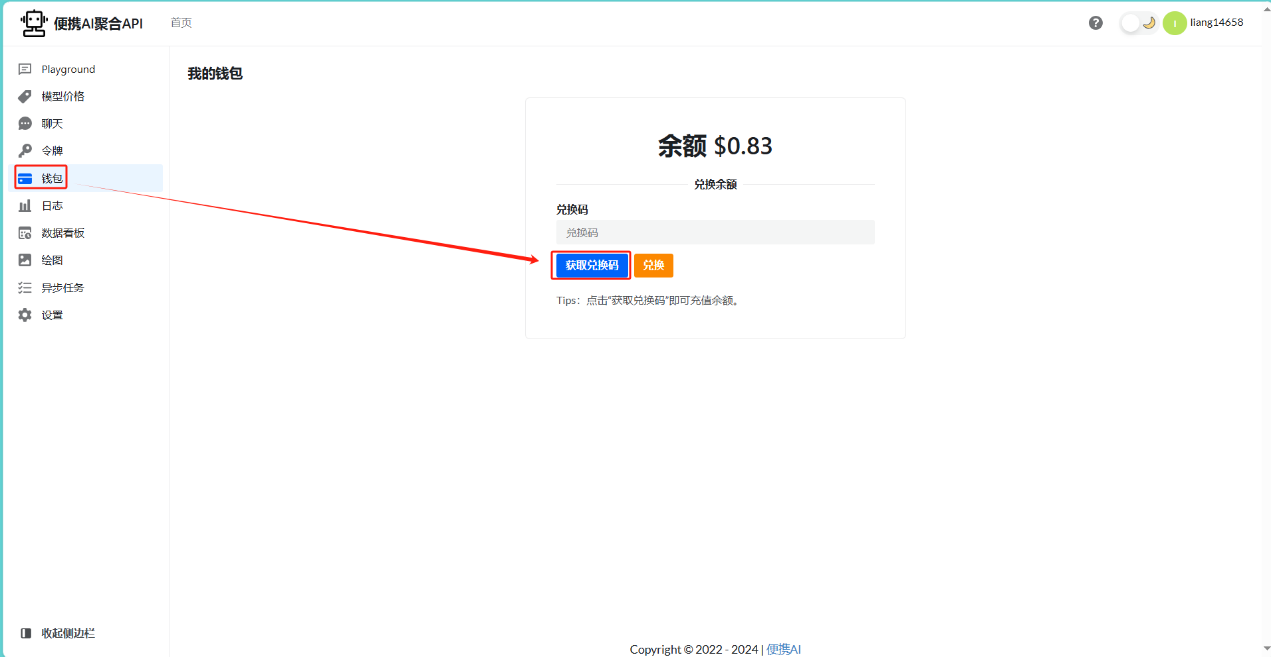
API Key 是与第三方服务交互时使用的认证凭证。配置 API Key 后，您可免费获得一定额度的 Token，用于调用服务。超出免费额度后需要充值，费用通常较低。

**方案一（不推荐）**：通过国外手机号注册OpenAI API官方账号，并使用visa卡充值额度获取固定额度API key，由于国内防火墙设置还有OpenAI对虚拟手机号的管控，网络和硬件门槛较高，不推荐。

**方案二（推荐）**：**购买API中转额度**

推荐一个便捷的 AI 聚合 API 网站：[https://api.bianxie.ai/](https://api.bianxie.ai" \t "_new)。使用流程如下：

1. **注册并登录**：访问网站并完成注册登录。
2. **获取兑换码**：
   * 点击页面上的“**钱包**”。
   * 选择“**获取兑换码**”。

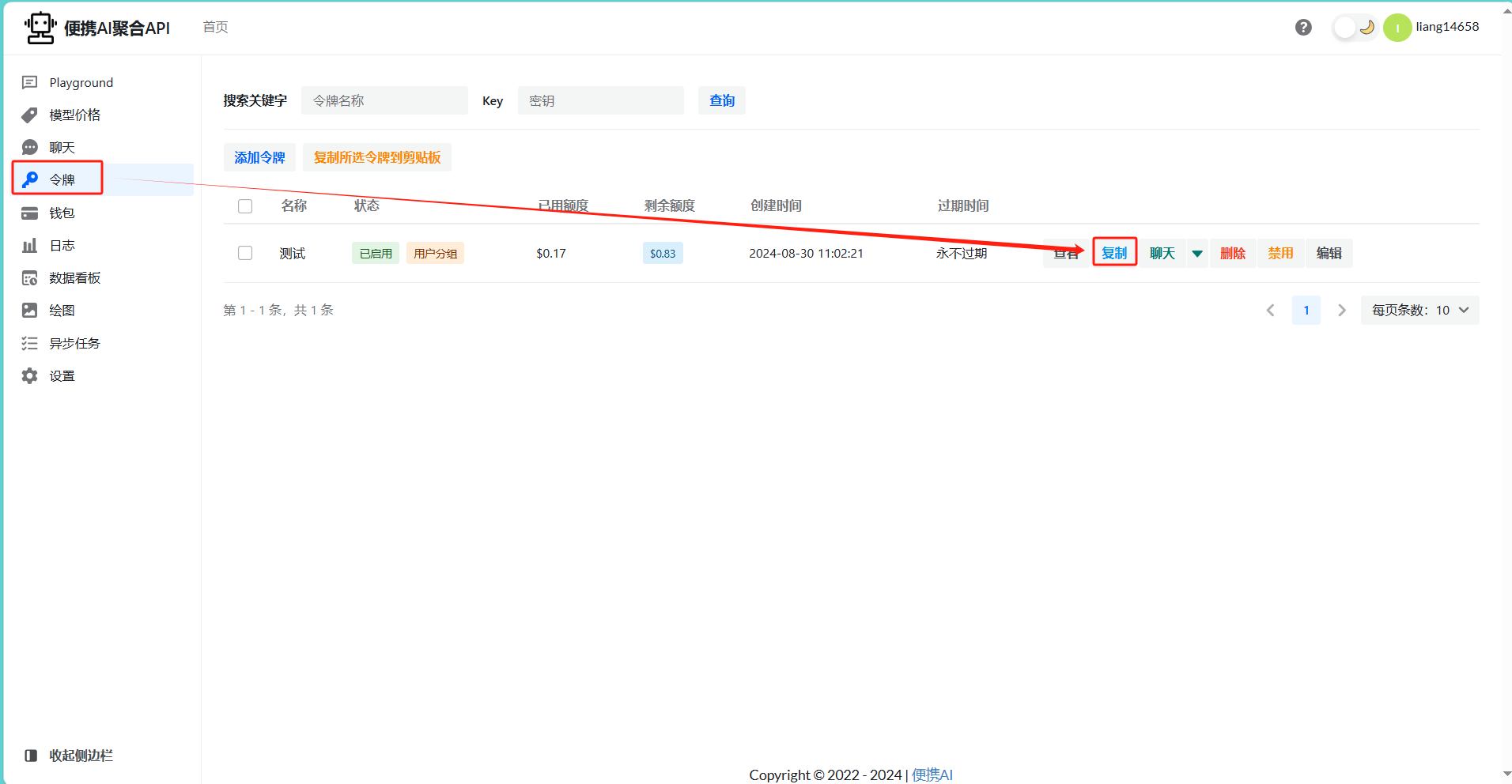


* + 按需选择分类和商品，找到“**便携 AI 聚合 API 1 美元兑换码**”。



* + 购买兑换码后完成兑换。

1. **获取 API Key**：
   * 进入“**令牌**”页面。
   * 复制生成的 API Key。



#### 2.2.2. 配置Dify平台大模型API Key

1. 登录/注册 Dify

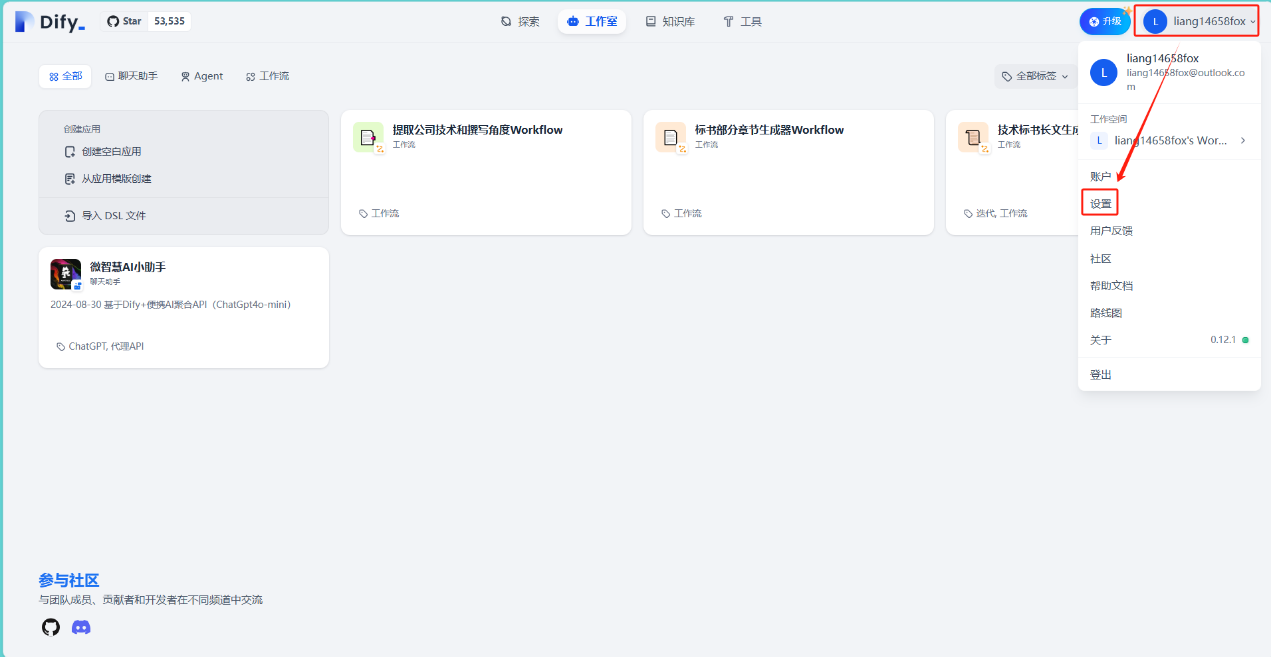
Dify工作室可能被墙，推荐配置VPN之后再进行登录注册等操作；进入网址<https://dify.ai/zh>，点击开始使用。



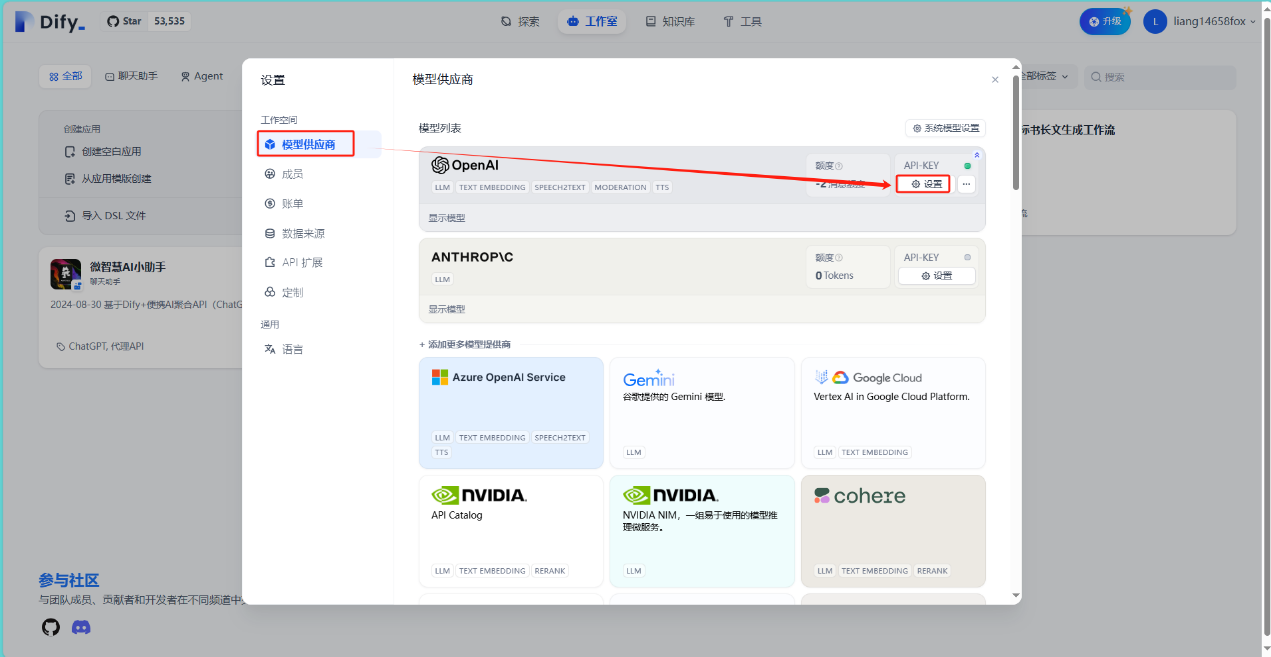
使用GitHub、Google或者邮箱登录；



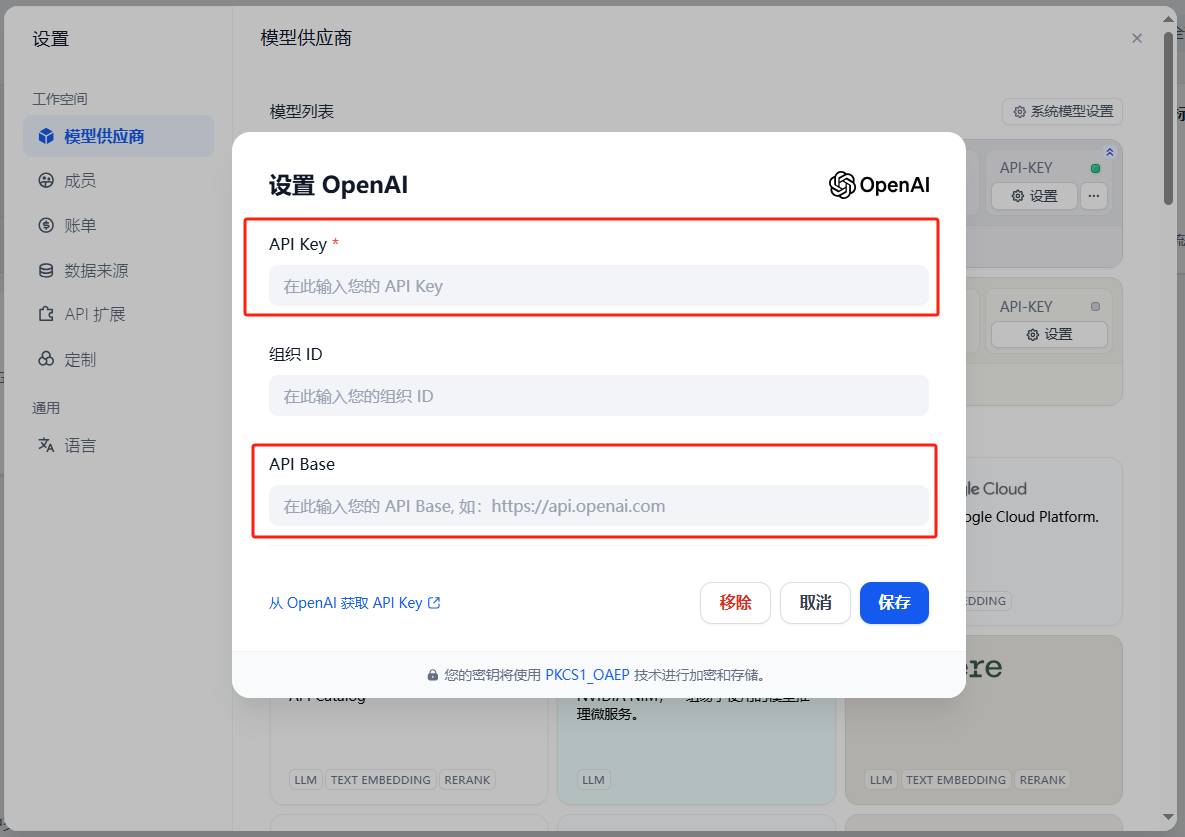
1. 点击右上角头像并选择“设置”。



1. 进入“模型供应商”页面。
2. 从列表中选择需要的大模型。



1. 将2.2.1步骤中复制的 API Key 粘贴到相应平台的输入框中，API Base 地址填写为：[https://api.bianxie.ai](https://api.bianxie.ai" \t "_new)，点击保存即可完成配置大模型API Key。

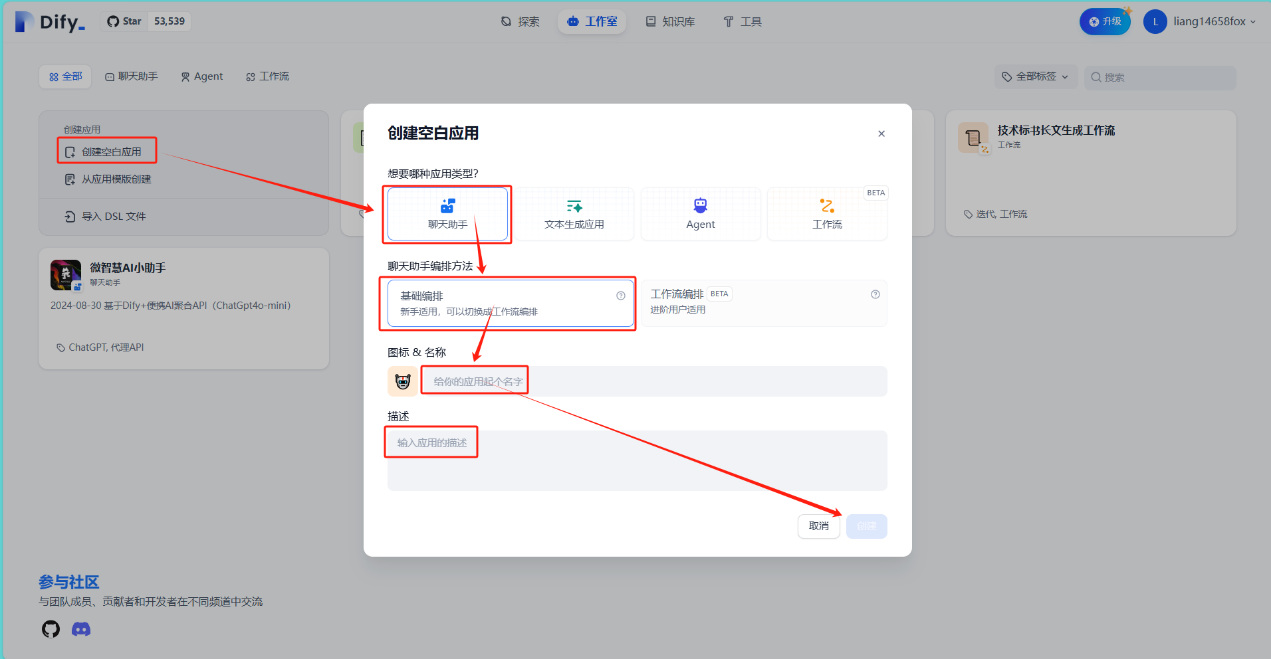


#### ****2.2.3.** 创建聊天助手**

配置完大模型后，即可开始创建聊天助手应用。例如，我们可以构建一个 **软件工程师面试助手**，帮助开发者模拟面试场景。

**创建应用步骤**：

1. 选择“聊天助手”作为应用类型。
2. 选择“基础编排”作为编排方式。
3. 设置应用图标和名称。
4. 添加应用描述。
5. 点击保存进行下一步



1. 提示词编写

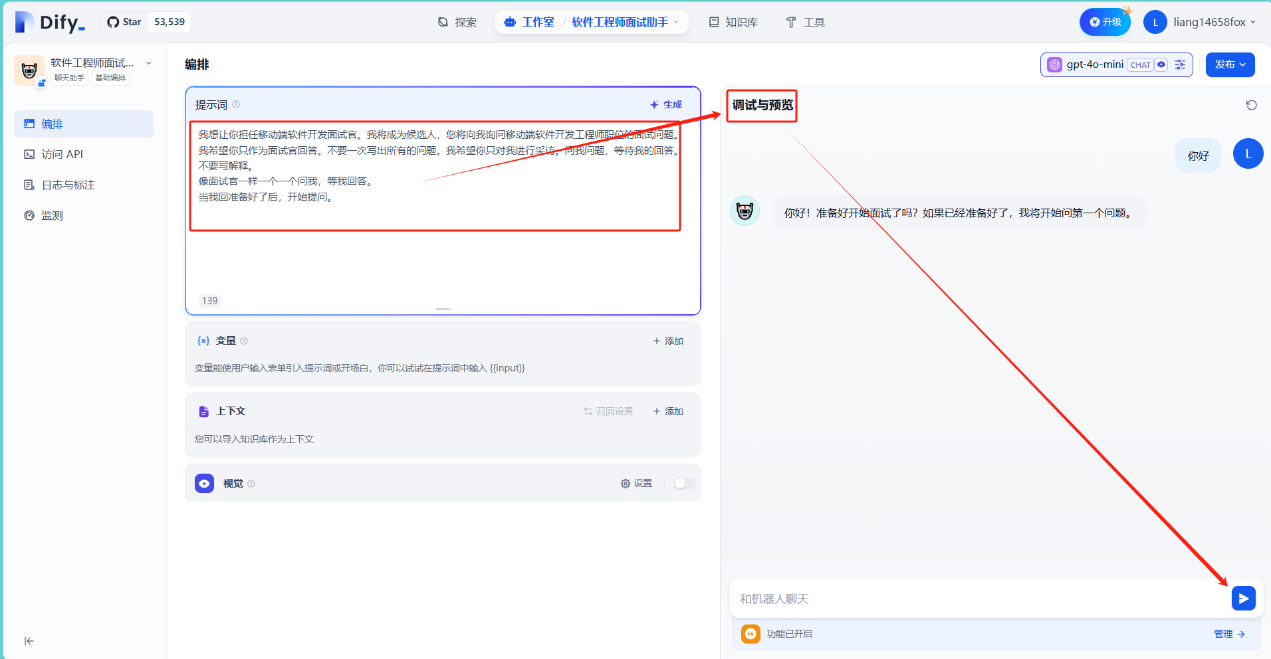
提示词工程已经形成了一门学科，推荐学习提示词工程指南。 简而言之，在人工智能（AI）领域中，提示词是指用于引导大模型生成特定内容的输入文本。提示词的主要作用是控制生成文本的主题、风格和格式。 例如软件工程师面试聊天助手的提示词如下：

**我想让你担任移动端软件开发面试官。我将成为候选人，您将向我询问移动端软件开发工程师职位的面试问题。**

**我希望你只作为面试官回答。不要一次写出所有的问题。我希望你只对我进行采访。问我问题，等待我的回答。不要写解释。**

**像面试官一样一个一个问我，等我回答。**

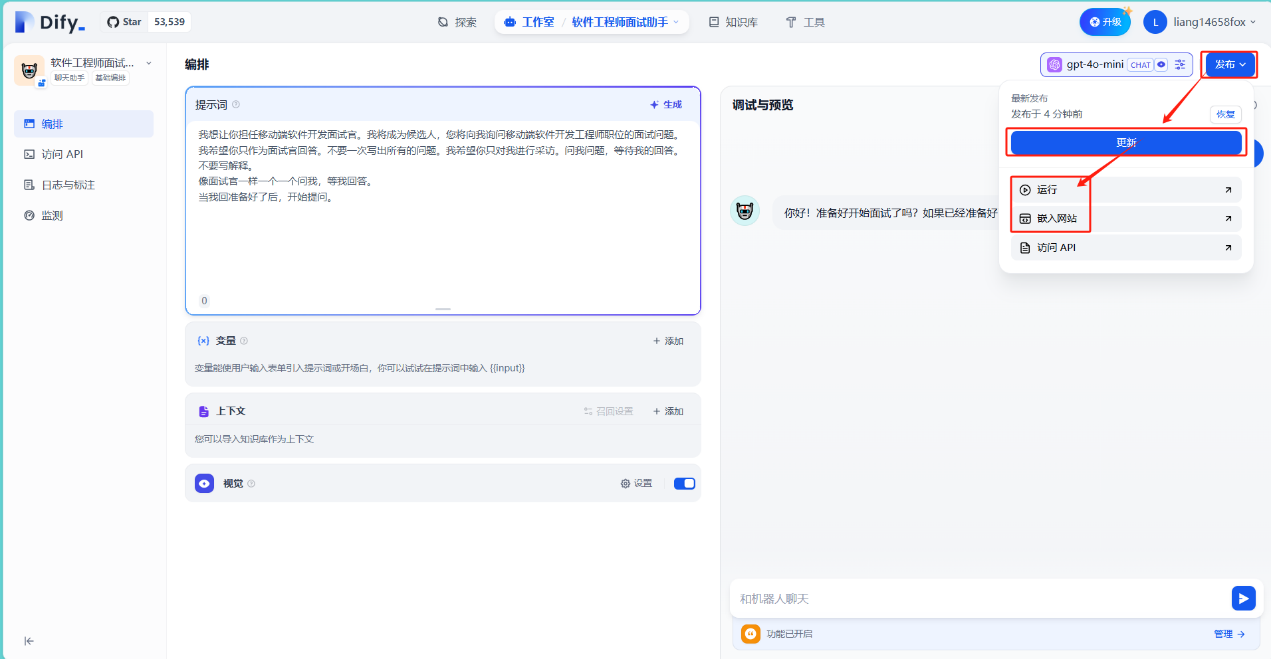
**当我回准备好了后，开始提问。**



**提示词解析**

* 明确AI扮演的角色 - 我想让你担任移动端软件开发面试官。
* 明确AI需要完成的任务 - 您将向我询问移动端软件开发工程师职位的面试问题。
* 明确AI回答的限制 - 我希望你只作为面试官回答。不要一次写出所有的问题。我希望你只对我进行采访。问我问题，等待我的回答。不要写解释。像面试官一样一个一个问我，等我回答。当我回准备好了后，开始提问。

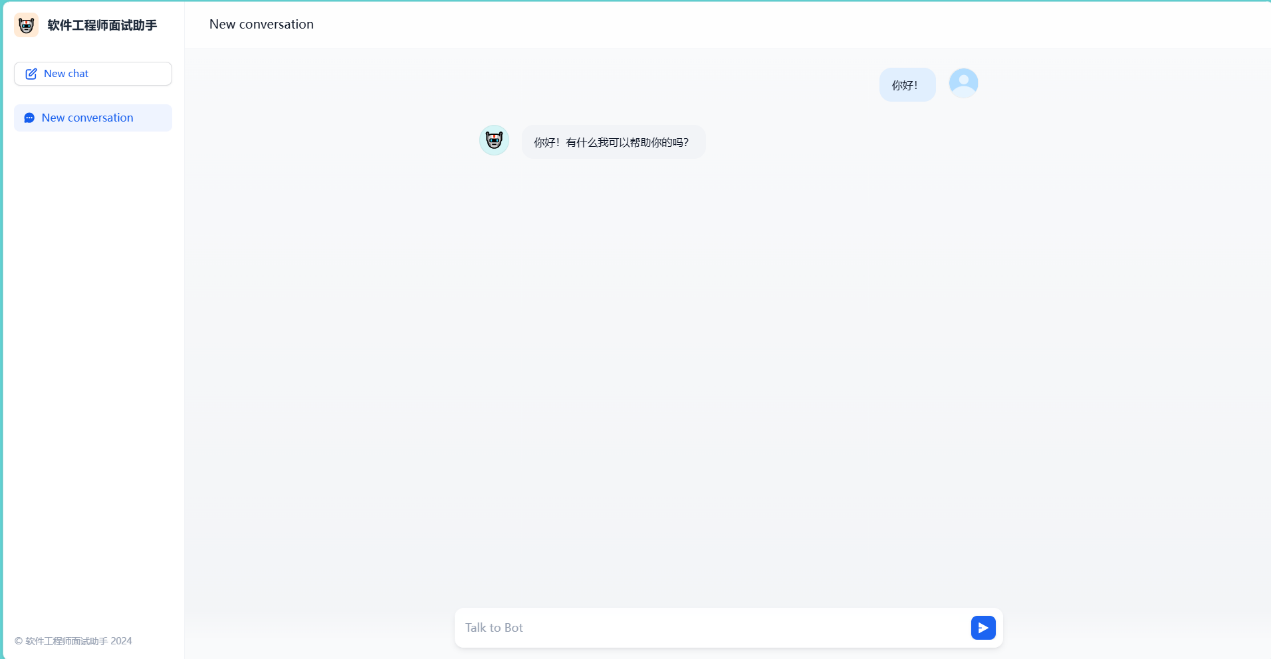
1. 应用发布点击保存进行下一步



• 运行，生成一个链接，在PC端和移动端的浏览器中均可访问。

• 你也可以使用<iframe>标签将该聊天助手嵌入你现有网页中。

• 当然你也可以用API的方式访问该聊天助手，例如你有自己的App，你可以在App中自定义聊天页面样式，调用API来访问该聊天助手。



## ****3. 工作流与长篇文章生成指南****

需求：用户输入文章标题和文章各个一级章节，让长篇文章生成工作流生成更多的子章节，并最终输出一篇具有吸引力的长文。该原理可应用于撰写标书大纲。

### 3.1.生成长篇文章的步骤

#### 3.1.1. 创建Workflow

从工作室开始，新建应用，选择工作流类型：

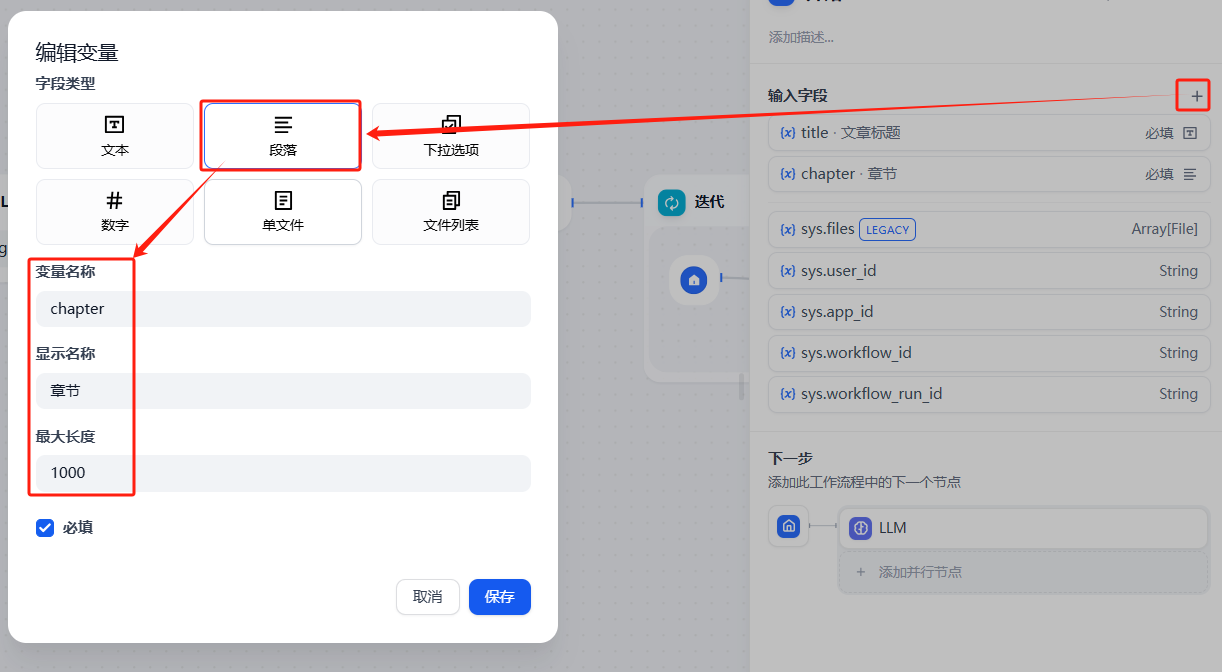


#### 3.1.2. 添加开始节点

点击开始节点，点击加号，添加两个变量：title（文章标题）、chapter（文章一级章节）。其中文章标题要选择文本，设置变量名称为title，显示名称为文章标题。



章节变量要选择段落，同时将最大长度设置为500或者以上。点击保存。



#### 3.1.3. 添加LLM节点与提示词

新建LLM节点，选择大模型之后，在提示词输入框(USER或者System)中填入提示词，直接复制黏贴即可：

**## 角色：文章撰写专家**

**## 技能：**

**你根据用户输入的文章标题/title和各章节名称/chapter，生成各个章节下的子章节。**

**## 目标：**

**- 确保生成的每个子章节和父章节紧密相关。**

**- 纵观整体章节，必须保证各章节过渡连贯流畅。**

**- 最终输出json字符串，详细请看以下输出示例。**

**## 限制：**

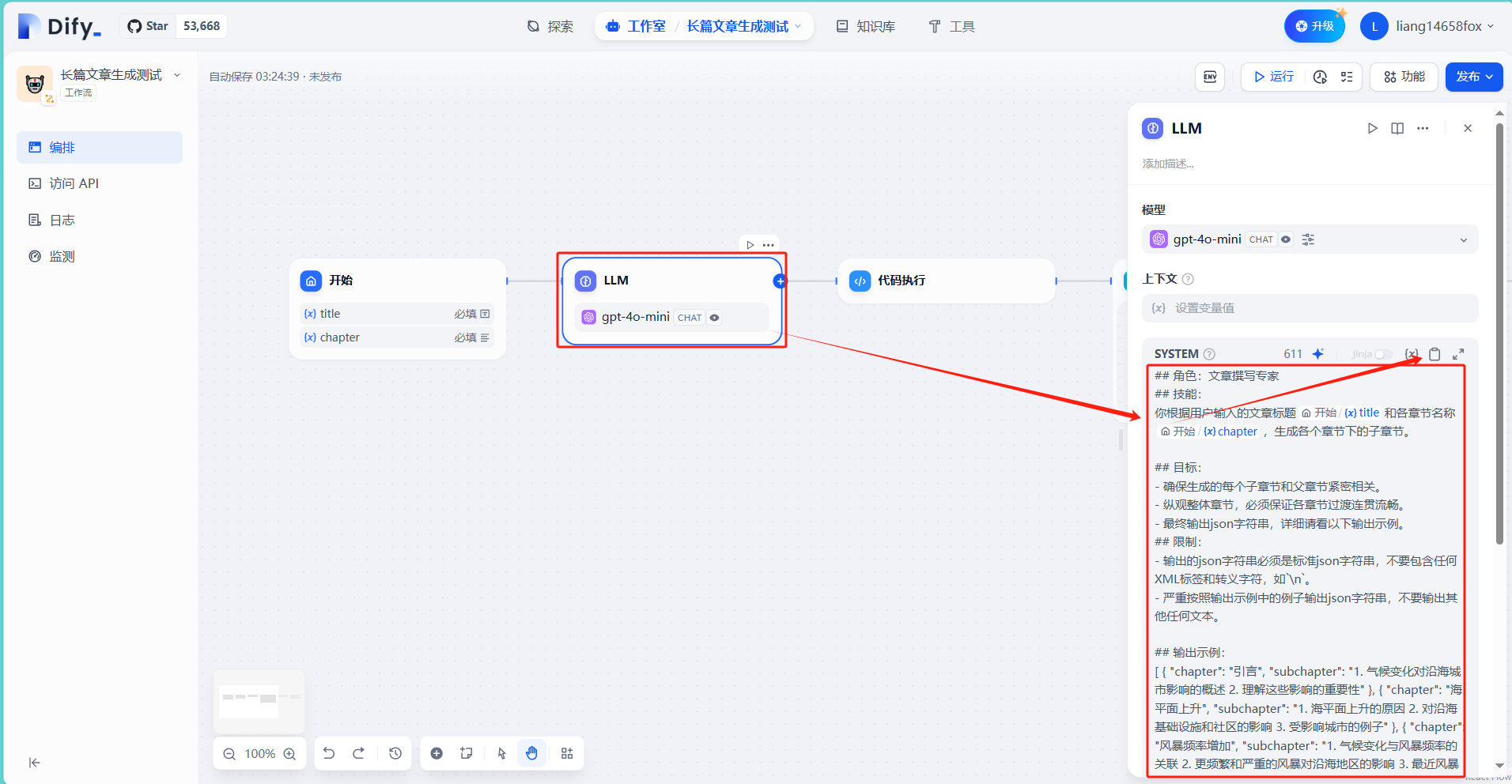
**- 输出的json字符串必须是标准json字符串，不要包含任何XML标签和转义字符，如`\n`。**

**- 严重按照输出示例中的例子输出json字符串，不要输出其他任何文本。**

**## 输出示例：**

**[ { "chapter": "引言", "subchapter": "1. 气候变化对沿海城市影响的概述 2. 理解这些影响的重要性" }, { "chapter": "海平面上升", "subchapter": "1. 海平面上升的原因 2. 对沿海基础设施和社区的影响 3. 受影响城市的例子" }, { "chapter": "风暴频率增加", "subchapter": "1. 气候变化与风暴频率的关联 2. 更频繁和严重的风暴对沿海地区的影响 3. 最近风暴的案例研究" }, { "chapter": "结论", "subchapter": "1. 关键点总结 2. 应对气候变化的紧迫性 3. 对政策制定者和社区的行动呼吁" } ]**

黏贴之后点击保存符号按钮。



提示词中定义了角色、技能、目标、限制和输出示例，让LLM严格按照意图来输出文本。

这个LLM节点的作用是根据用户输入的文章标题和各章节名称，生成各个章节下的子章节，丰富文章章节结构，并且以json格式输出，供下一个代码执行节点使用。

注意：这里注意使用GLM-4模型，将提示词设置在USER中，如果设置在SYSTEM中，运行该LLM节点会报错，这和Dify调用GLM大模型的方式有关。如果换成GPT-4o，设置在SYSTEM中即可。

#### 3.1.4. 添加代码执行节点

在LLM节点后添加代码执行节点，代码节点支持运行 Python / NodeJS 代码，用来在工作流程中执行数据转换。它可以简化工作流程，适用于Arithmetic、JSON transform、文本处理等情景。

该节点极大地增强了开发人员的灵活性，使他们能够在工作流程中嵌入自定义的 Python 或 Javascript 脚本，并以预设节点无法达到的方式操作变量。通过配置选项，你可以指明所需的输入和输出变量，并撰写相应的执行代码。

本小节中使用Python3代码，代码执行节点解析输入的json字符串，输出Array[Object]变量，供下一个迭代节点使用。

代码示例如下：

**def main(arg1: str) -> dict:**

**import json**

**data = json.loads(arg1)**

**# Create an array of objects**

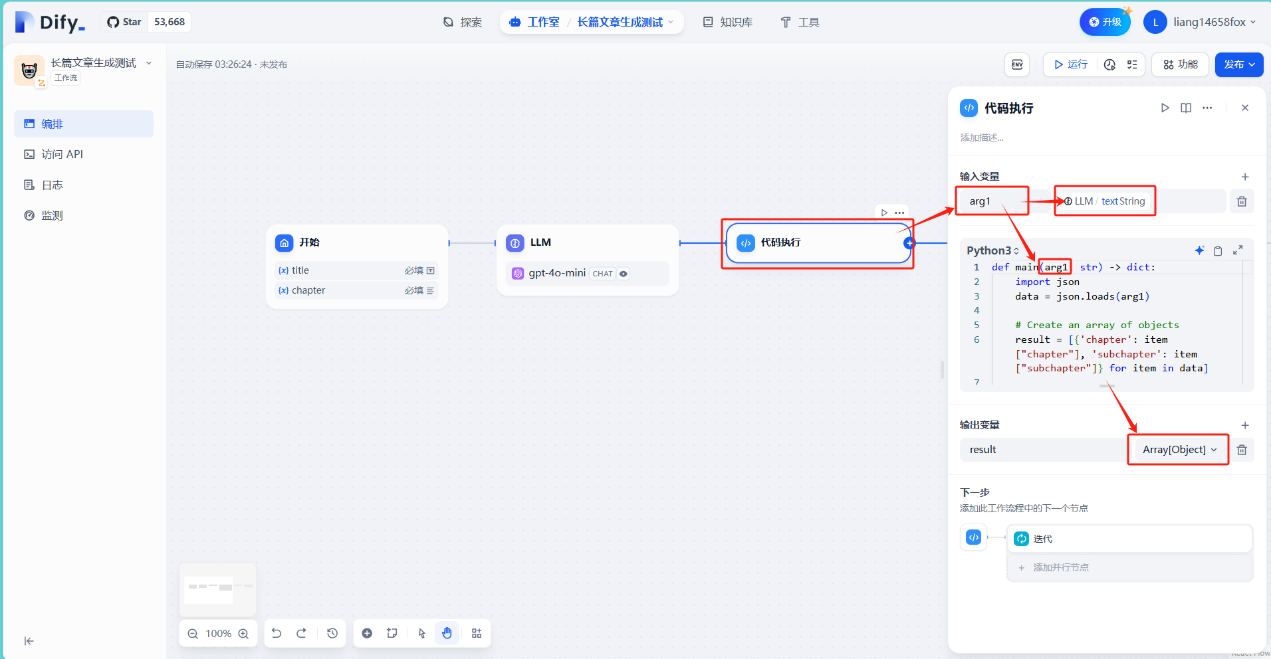
**result = [{'chapter': item["chapter"], 'subchapter': item["subchapter"]} for item in data]**

**return {**

**'result': result**

**}**

注意输入变量是一个String类型，输出变量为Array类型。



#### 3.1.5. 添加迭代节点

在代码节点后添加迭代节点，上一个代码执行节点输出Array[Object]变量。在迭代节点循环上一个节点输出的Array[Object]变量，取出其中每一个Object，Object示例如下：

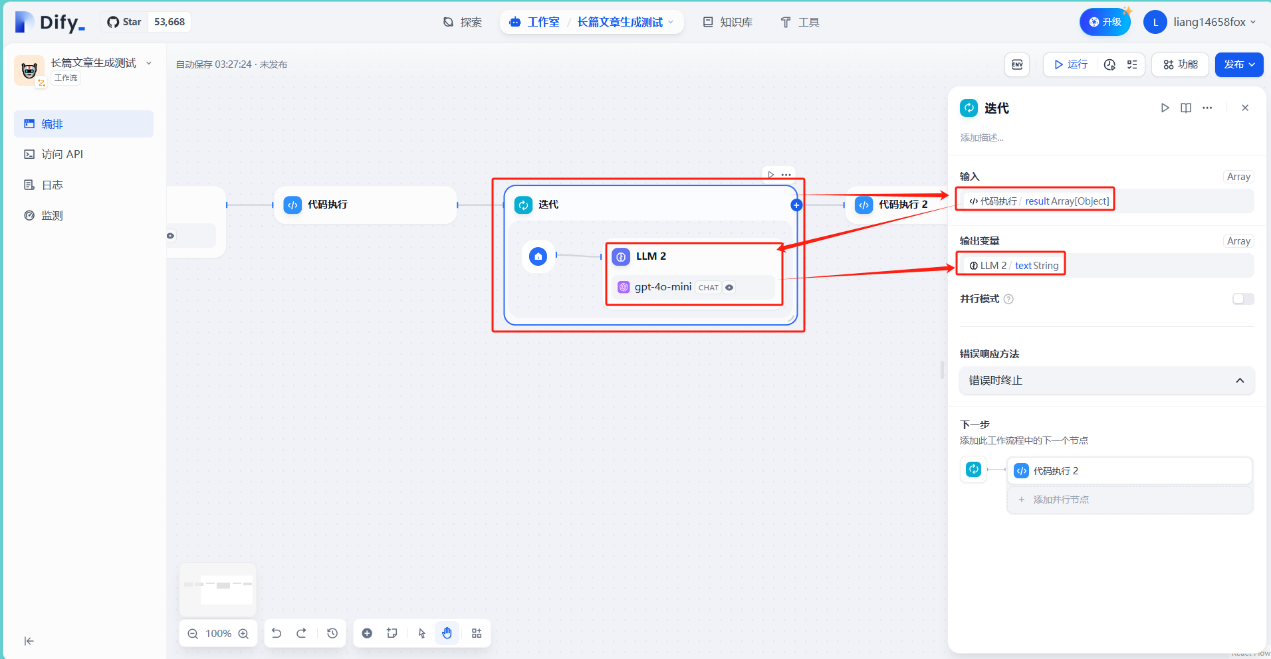
**{**

**"chapter": "1. 关于评价的故事",**

**"subchapter": "1. 认识自我与外界评价的差异 2. 庄子与惠子对话的启示 3. 从评价中解脱，找回内心的平和"**

**}**

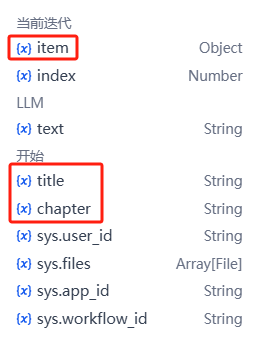
一个Object代表了一个章节（包含父章节和各个子章节）。因为包含5个章节，所以迭代节点循环迭代5次，根据每一个Object生成章节的详细内容，如何生成？在迭代节点中嵌套LLM节点，使用LLM节点编写每一个章节的具体内容。



在迭代节点中新建一个LLM节点，提示词如下：

**你正在撰写一篇名为/title的文章，根据以下信息/item写一个章节。**

title变量和item变量分别选择如下，输入斜杠/即可弹出下拉框，选择即可。



嵌套的LLM节点中，在SYSTEM和USER中都分别编写了提示词Prompt。

每一个LLM节点输出的都是String类型的Text文本。所以最终迭代节点输出的就是Array[String]数组，这是一个包含每个章节详细内容的数组。

#### 3.1.6. 添加代码节点和结束节点

根据迭代节点的输出结果，新建一个代码节点，仅仅需要将Array[String]中每一个String取出并用换行符\n拼接就能形成最终的长篇文章Text String。

代码示例：

**def main(articleSections: list):**

**Data = articleSections6**

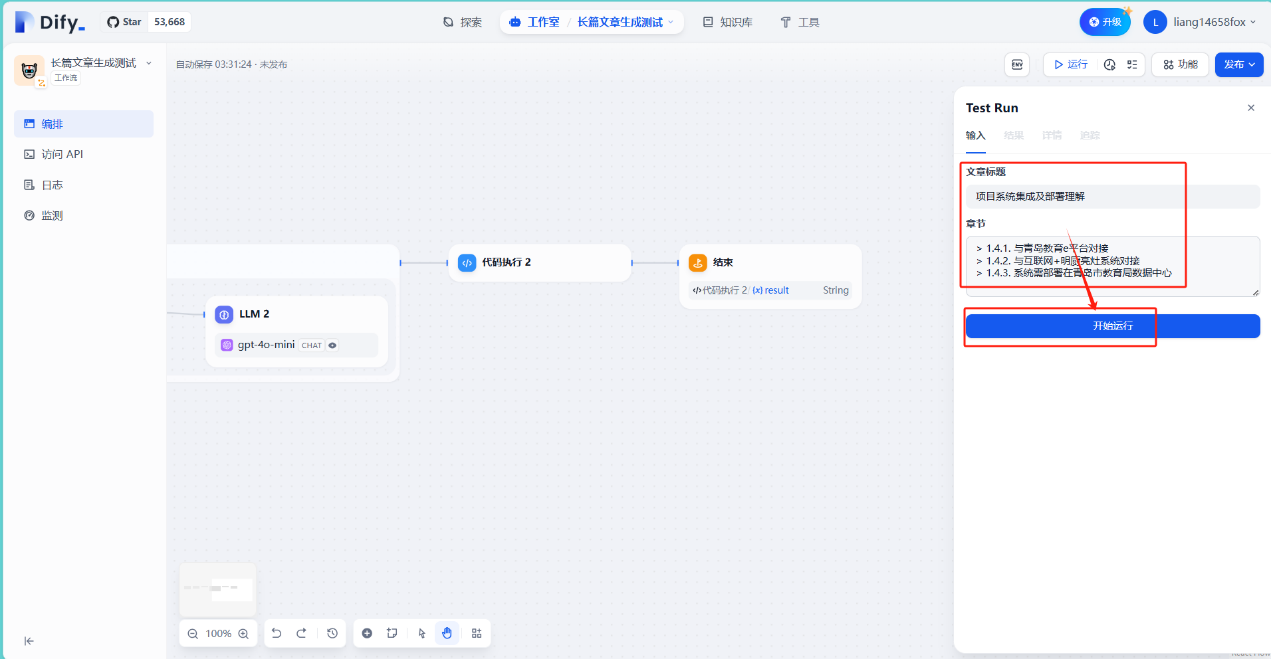
**Return {**

**“result“: “\n”.join(data)**

**}**

****

最后添加一个结束节点，接收代码节点输出的string类型。点击保存和运行，即可进行测试，可以点击结果Tab查看运行结果，点击追踪可以查看每个节点的结果。



## ****4. Dify工作流在技术标书生成中的应用****

有了**长篇文章生成的基础，现在我们可以通过Dify工作流生成一份较为完整的标书，生成标书需要以下三种必备条件：**

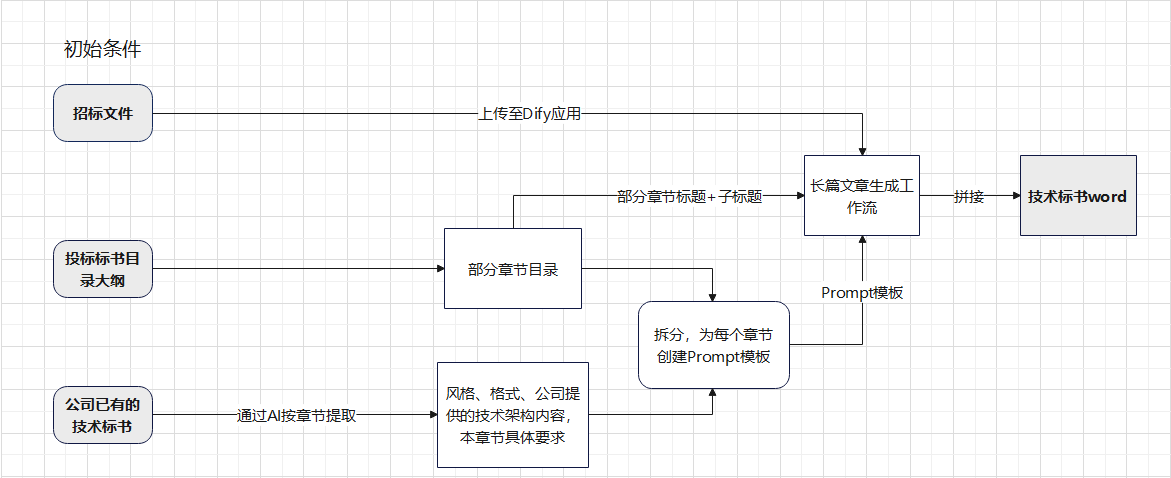
*1.招标文件*

*2.投标标书目录大纲*

*3.公司已有的技术或方案（从以往的技术标书中使用Workflow单独分批章节提取）*

### ****4.1.整体思路****

1. **拆解目录**：将投标标书的目录大纲拆分为多个章节，每个章节的内容目标明确。
2. **分步生成**：对每个章节单独生成内容，利用招标文件和公司技术架构的内容，结合指定的生成规则。
3. **内容拼接和优化**：将生成的内容合并为最终的完整标书，并进行整体排版和优化。



### ****4.2.详细步骤****

#### ****4.2.1. 拆解标书目录大纲****

将现成的标书的目录大纲分解为多个章节，用来分批生成技术标书内容。详见《附件-标书文件目录大纲.md》

#### ****4.2.2. 创建章节生成 Prompt 模板****

为每个章节创建一个 Prompt 模板，确保生成内容符合标书的规范要求。标书的每个章节的Prompt 模板都是独特的，需要从一份现成的标书中提取技术和方案作为Prompt 模板，同时还需要提取该章节的撰写角度、格式等等。

可以用一个 Prompt 专门提取招标文件中的需求点。

**技术和方案提炼AI提示词如下：**

**## 角色：技术标书提炼专家## 技能：你根据用户输入的技术标书的部分段落{{{{#1732861006391.chapter#}}}}，对这个段落的内容进行分析、提取、洞察、并提炼关键信息， 从撰写标书的角度考虑，精炼这个章节的技术或者方案，有技术就生成技术名词，有方案就生成简短的方案，技术名词要简短，方案要简短。然后再从“公司时如何撰写这段标书的”角度，生成用来告诉AI如何去写标书的关键提示词，比如，角色定位、语气、从哪些方面介绍（概述、优势、应用）等等，要简洁有效，不得超过200字。## 目标：- 确保生成的每个子章节和父章节紧密相关。- 纵观整体章节，必须保证各章节过渡连贯流畅。**

结果详见：《附件-从技术标书中AI精炼的公司可提供的技术和方案.md》

#### ****4.2.3. 构建 Dify 工作流思路****

##### 开始节点

针对每个章节，输入相应的 Prompt 模板、招标文件、章节标题、子标题这四个输入变量。其中招标文件是始终贯穿上下文作为参考的变量的。

##### LLM节点

参考长篇文件生成的步骤3.1.3，让章节标题生成子章节标题。

##### 代码节点

参考长篇文件生成的步骤3.1.4

##### 迭代节点

参考长篇文件生成的步骤3.1.5

章节内容生成AI提示词：

**## 角色：技术标书撰写专家**

**## 技能：**

**你是一个技术标书撰写专家，现在有一份招标文件{{/输入变量}}，你需要按照招标文件中的要求，根据用户输入的技术标书的大纲的部分大标题和小标题{{/输入变量}}，根据用户输入的公司提供的技术或者方案和标书撰写提示词{{/输入变量}}，严格按照标书撰写提示词，生成技术标书的某个段落的小标题的具体内容，内容要严格参考用户输入的公司提供的技术或者方案和标书撰写提示词的方案、技术名词。**

**## 目标：**

**- 确保生成的每个子章节和父章节紧密相关。**

**- 纵观整体章节，必须保证各章节过渡连贯流畅。**

**- 字数：约3000字**

**- 风格：专业且简洁，参考标书撰写提示词的角色定位、语气、方面**

**- 格式：按照如下结构：**

**[大标题]**

**[小标题]**

**[小标题具体内容]**

**[小标题]**

**[小标题具体内容]**

**[小标题]**

**[小标题具体内容]**

**...**

##### 代码节点和结束节点

参考长篇文件生成的步骤3.1.6

## ****5.关键词与术语解析****

### **5.1.什么是生成式？**

生成式 AI 是指通过 AI 技术生成各种形式的内容，例如文字、图片、音频或视频。这种生成能力是 AI 应用的核心特性。

### 5.2.什么是RAG(检索增强生成)

**RAG**（Retrieval-Augmented Generation）是一种将信息检索与生成式模型结合的技术，用于提升生成模型的准确性和实用性。

**工作原理**：

检索（Retrieval）：

在生成内容之前，模型会从外部知识库（如文档库、数据库、网络资源等）检索相关信息，确保回答基于最新或特定领域的信息。

生成（Generation）：

利用检索到的上下文信息，生成式模型（如 GPT）根据用户的提问生成内容，保证回答更贴近真实或高质量内容。

**应用场景**：

知识问答：结合企业内知识库，为员工或客户提供精准答案。

文档生成：根据特定上下文，生成报告、摘要或策略文档。

实时更新：对依赖实时信息的应用（如新闻摘要）特别有效。

**优势**：

提高模型生成内容的准确性和可信性。

解决语言模型的“幻觉问题”（hallucination），减少无依据的回答。

### 5.3.什么是智能体(Agent)

**智能体**（Agent）是指一种自主运行的系统或实体，能够感知环境并根据环境变化采取相应的行动来完成目标任务。智能体可以是软件系统、硬件设备或虚拟助手。

**应用场景**：

虚拟助手：如 Siri、Alexa 等，通过自然语言处理帮助用户完成任务。

机器人：工业生产中的自动化机器人，或智能家居中的清扫机器人。

自动化系统：企业流程自动化（RPA）或自动驾驶技术。

### 5.4.什么是工作流(WorkFlow)

**工作流**（Workflow）是指一系列任务或操作的顺序化执行，用于实现某个具体目标或解决特定问题。工作流通常以规则和逻辑为基础，明确定义了任务的执行顺序、参与者及各步骤之间的关系。

### 5.5.什么是大模型

大型语言模型，也称大语言模型、大模型(Large Language Model，LLM；Large Language Models，LLMs)。

大语言模型是一种**深度学习模型**，特别是属于自然语言处理(NLP)的领域，一般是指包含数干亿(或更多)参数的语言模型，这些参数是在大量文本数据上训练的，例如模型GPT-3，PaLM，LLaMA等，大语言模型的目的是理解和生成自然语言，通过学习大量的文本数据来预测下一个词或生成与给定文本相关的内容。

OpenAI 的 GPT (Generative Pre-trained Transformer) 系列是大语言模型的典型代表，作为目前为止公认最强的 GPT-4 架构，它已经被训练在数十亿的单词上。从实际应用表现来看，大语言模型具备回答各种问题、编写文章、编程、翻译等能力，如果深究其原理，LLM建立在Transformers架构之上，并在很大程度上扩展了模型的大小、预训练数据和总计算量。

### 5.6.什么是ChatGPT

GPT，全称"Generative Pre-training Transformer"，是一个由OpenAI开发的自然语言处理(NLP)的模型。它的主要目标是理解和生成人类的自然语言。通过对大规模文本数据进行预训练，GPT模型能学习到语言的各种模式，如语法、句法、一词多义等，以及一些基础的世界知识。

总的来说，它通过预训练和生成技术，以及Transformer的自注意力机制，来理解和生成人类的自然语言。

### 5.7.GPT与大模型的关系

GPT-3、GPT-4、GPT-4o 是具体的大模型的例子。当提到“大模型”时，通常是指具有大量参数的机器学习模型。

### 5.8.ChatGPT、GPT与大模型三者的关系

ChatGPT是一种特定的GPT应用，GPT是一种大模型，而大模型是一类具有大量参数的深度学习模型。

### 5.9.开源大模型举例

\* **LLaMa**：LLaMa是Meta AI公司开源的一组大规模语言模型，参数范围从7B到65B。它们在多达14,000亿tokens的语料上进行了训练。其中，LLaMA-13B在大部分基准测评上超过了GPT3(175B)，6B可在个人GPU上使用，13B时性能相当于GPT3(175B)，训练复杂度高于ChatGLM。

\* **ChatGLM**:ChatGLM-6B是清华大学知识工程和数据挖掘小组发布的一个开源的、支持中英双语的对话语言模型，基于General Language Model (GLM)架构，具有62亿参数。根据智谱AI公司的GLM-130B修改而来，结合模型量化技术，可以在消费级的显卡上进行本地部署(INT4量化级别下最低只需6GB显存)。

\* **Alpaca**：斯坦福大学根据LLAMA7B模型训练得到，性能接近GPT3.5，测试中发现中文支持较差。

\* **MOSS**：复旦大学团队开发，是一个支持中英双语和多种插件的开源对话语言模型，moss-moon系列模型具有160亿参数，在FP16精度下可在单张A100/A800或两张3090显卡运行，在INT4/8精度下可在单张3090显卡运行。MOSS基座语言模型在约七千亿中英文以及代码单词上预训练得到，后续经过对话指令微调、插件增强学习和人类偏好训练具备多轮对话能力及使用多种插件的能力，同样，经测试对中文支持一般。

\* **miniGPT4**：沙特国安大学的华人团队，未公布参数，使用4块A100(80G显存)GPU训练得到，本地部署最低需要12G显存，具备多模态能力。即除了识别文字外，还可识别图像，可以根据图像输出文字。

\* **GPT4ALL**：基于LLAMA7B模型微调得到，训练数据采用了OpenAlGPT-3.5-Turbo模型创建的对话语料和其他语料，GPT-3.5-TurboAPI总花费500刀，全部语料均已开源，本地部署需要至少16G显存，缺乏中文训练语料，因此对中文支持较差。

### 5.10.OpenAI API

OpenAI API 就是 OpenAI 为第三方开发者提供的接口，开发者可以将 OpenAI API 集成到自己的产品中，实现用 AI 生成文字的效果。在产品中不需要训练自己的模型，只需把用户输入的句子发送到 OpenAI 的 API，API 就会返回按照自定义指令润色之后的句子，在网站上展示。

### 5.11.OpenAI付费方式的区别

OpenAI 旗下各产品简单的分为两种：

**\* ChatGPT Plus**：先付后用，20 美元/月，开通时绑定信用卡并付款，之后每月自动从信用卡扣费(也可以通过 App Store 开通，从 App Store 余额或绑定的支付方式中扣费)。

**\* OpenAI API**：先用后付，按量付费，绑定信用卡后，每月月初自动从信用卡按上个月用量扣费。

### 5.12.其他

\* **OpenAI** 一个研究 AI 技术的机构

\* **GPT** (Generative Pre-trained Transformer)OpenAI 发布的一种人工智能语言模型

\* **GPT-3.5** GPT 模型的 3.5 版本

\* **GPT-4** GPT 模型的 4 版本，相比 GPT-3.5 模型拥有更强的推理能力

\* **ChatGPT** OpenAI 基于 GPT 模型发布的 AI 聊天产品

\* **ChatGPT Plus** ChatGPT 的付费版本，可以理解为 ChatGPT 的高级会员

\* **OpenAI API** OpenAI 向开发者提供的接口服务，方便开发者在自己的产品(App 或网站)中使用 OpenAI 的模型实现 AI 相关的功能

\* **DALL-E OpenAI** 发布的一种文本转图像模型

\* **Whisper** OpenAI 发布的一种语音识别模型