**Lab 2 Report**

1. **项目概述**

本项目是一个基于 Gradio 框架开发的 Web 聊天界面，旨在支持多种大语言模型的切换和调用，提供流式响应、多轮对话历史管理等功能，方便用户使用不同模型进行自然语言交互。

本系统支持对 OpenAI、DeepSeek 等模型的调用，并通过统一的接口封装，实现模块化管理和扩展。前端采用 Gradio 构建，组件清晰，交互简洁直观。

1. **项目功能**

|  |  |
| --- | --- |
| **功能类别** | **描述** |
| 多模型支持 | 可以成功的调用LLM模型的API来回答用户的问题，存 在基本的输入输出的页面 |
| 流式响应 | 实现 token 级别的实时输出，提高交互体验 |
| 基础对话功能 | 可以成功的调用LLM模型的API来回答用户的问题，存 在基本的输入输出的页面 |
| 多轮对话功能 | 在同一个对话下，记录同该用户之前的对话内容，一 同输入给大模型，从而实现多轮对话功能 |
| 历史记录功能 | 可以记录用户之前同LLM的对话历史记录，并且可以 选择之前的历史记录继续聊天 |

1. **Tasks:**

**3.1不同模型在进行API调用的时候是否有区别，如果有区别，区别在什么地方?**

**3.1.1 请求结构差异**

不同平台的模型对请求参数的结构定义并不统一。例如：

OpenAI GPT-4/3.5 使用 messages 字段，包含一组带有 role 的对话信息；

Anthropic Claude 使用 prompt 或 messages，不严格区分 system/user/assistant；

DeepSeek/Moonshot 多采用与 OpenAI 类似的结构，但字段命名可能存在微调（如 stream / streaming）。

这种差异要求开发者在统一封装时对参数做动态适配，以实现 Prompt 的跨模型兼容。

**3.1.2 身份验证机制不同**

各模型提供商在身份验证方面各有要求：

OpenAI 和 DeepSeek 通常使用 Bearer Token；

Claude API 需要 x-api-key 头；

Moonshot 可能还需要额外的请求签名校验。

统一管理 API 密钥并在调用中注入对应 Header，是系统开发中的重要安全环节。

**3.1.3 响应结构格式不同**

返回的结构同样存在不一致性：

OpenAI 通常返回 response['choices'][0]['message']['content']

Claude 返回 response['completion']

DeepSeek 返回 response['data']['content']

因此，系统必须实现一套“统一的响应提取逻辑”，确保后续模块处理的是标准格式。

**3.2自己在设计用户交互页面时的思路，自己为什么要这样子设计自己的页面，页面当中的组件的作用都是什么**

页面预览:



1. 模型选择:



作用: 切换对话框中使用的LLM模型，让用户可以体会不同模型之间的差别

思路: 可以加入模型选择的按钮，让用户更加方便地切换使用的模型，在处理不同的问题时能够使用不同的模型，更能够体会不同模型之间的区别，方便了用户的使用

1. 对话管理:



作用: 可以对对话框进行管理，新建对话，或者删除已有的对话

思路: 因为用户会同时有处理不同的对话的需求，通过引入管理对话的选择组件能够更加方便用户在同一时间处理不同的对话，方便使用

1. 对话窗口布局



作用: 让用户可以直观地看到LLM对自己的输入的回复

思路: 将对话窗口直接放在右侧大窗口中，重点突出了回复窗口的内容，强调了重点

1. 输入选择



作用: 可以选择接受不同类型的消息，例如图片、文字、语音等等

思路: 用户有时候想要上传图片或者语音等等非文字内容，通过提供不同内容的选择组件可以更加直观地让用户选择自己的输入内容，更加方便使用

**3.3大模型当中存在许多可以调整的参数（例如：采样温度、核采样阈值等）这些参数对大模型生成的回答有什么影响**

采样温度用于控制输出内容的随机性。当温度较低时，模型生成的内容更具创造性，但同时可能引入更多不确定或不相关的信息。

核采样阈值用于限定在概率总和为 p 的前若干个候选词中进行随机抽取。与 Top-k 不同，Top-p 不以固定数量，而是以总概率为依据来选择词语较低Top-p 值会让输出更加集中和确定，而较高的值则提供更多样性和创造性。通过灵活调整这些参数，可以根据不同任务的需求，平衡回答的准确性和多样性。

**3.4** **不同的提示词对大模型的输出会产生不同的区别 ，在面对日常使用和专业问题的时候，你会更推荐什么风格的提示词书写方法呢？**

在大模型的应用过程中，提示词的设计风格对生成结果的质量与风格具有显著影响。具体而言，面对不同的使用场景，应采用差异化的提示策略，以提升生成内容的相关性、准确性与实用性。

首先，在日常使用场景下，用户更倾向于与大模型进行自然、灵活的人机对话，问题通常具有较强的开放性与模糊性。因此，此类提示词建议采用自然语言风格进行编写，即像与朋友交流那样表达需求。此类提示词语法松散、结构自由，强调模型的语言理解与生成能力，无需过多说明背景信息或输出格式，大模型能够凭借上下文理解并给出合理回应。自然语言风格的提示适用于写作创意、情感陪伴、日常知识问答等轻量型应用场景，其优点在于交互友好、上手门槛低、适应性强，但缺点是输出的稳定性和格式控制能力相对较弱。

相比之下，在面对专业性较强的任务时，如技术文档生成、学术报告撰写、代码分析、法律解读等，建议采用更加结构化且明确指令导向的提示词设计。结构化提示能够显著提升模型输出内容的准确性、逻辑性以及可控性，同时有助于复现性，即多次生成结果的一致性更高。这类提示的缺点是书写成本稍高，需要用户具备一定的任务抽象能力和指令表达能力。