基于 openEuler 的 Chatbot 文本机器人服务

项目结项报告

文 档 信 息

文件状态： [] 草稿 [yes] 正式发布 [] 正在修改	文件标识：	结项报告
	文件位置：	华中科技大学
	当前版本：	<1.0>
	作者：	杨明欣
	发布日期：	2023-9-23

文 档 更 改 记 录

版本	更改日期	更改人	更改原因	说明

目录

1 引言

随着信息技术的不断发展和应用，人工智能领域的研究和应用也进入了一个崭新的时代。在这个时代，大模型技术和 Chatbot（聊天机器人）已经成为了人工智能领域的热点和突破口之一。本文档旨在对于相关 oepkgs 的 Chatbot 文本机器人服务进行总结、分析和思考，并提出进一步的展望。

大模型，如 GPT（Generative Pre-trained Transformer）系列，已经成为自然语言处理领域的一项革命性技术。这些大型神经网络模型在海量语料库上进行预训练，可以理解和生成自然语言文本，实现了前所未有的自然语言理解和生成能力。这些模型的应用包括自动翻译、摘要生成、情感分析、对话系统等，为各行各业带来了巨大的创新和效益。

Chatbot，或称聊天机器人，是一种基于人工智能技术的应用程序，能够模拟人类对话并进行智能响应。Chatbot 的兴起源于大模型技术的发展，这使得机器能够更好地理解和生成自然语言文本。Chatbot 已经广泛应用于各个领域，如在线客服、医疗咨询、销售与营销、教育等，大大提高了用户体验，减少了人力成本，同时也推动了人机交互方式的创新。

Chatbot 的未来发展前景非常广阔。随着大模型技术的不断进步，Chatbot 将变得更加智能、个性化和逼真。它们将能够更好地适应不同的语境和用户需求，实现更加自然流畅的对话。未来的 Chatbot 还可能融合更多的多模态能力，如图像、语音和文本的混合交互，为用户提供更全面的服务。因此，openEuler 社区向大语言模型和 Chatbot 方向进行布局是非常有前瞻性的。

在本项目中，我完成了一个文本机器人服务，该服务利用大模型技术和 Chatbot 技术，能够在对话中收集关键信息，也可以针对现有知识库对于相关问题进行回答，同时对于准确度要求较高的问题，融合了对话和信息检索的技术，不断应对复杂的环境和严格的要求。通过这个项目，我为 openEuler 社区未来 Chatbot 的发展和应用贡献了一份微薄的力量，也深刻理解了大模型技术在自然语言处理中的应用潜力。

2 系统概述

基于 Oekpgs 智能问答系统采用 B/S（Browser/Server）架构，是一种基于浏览器和服务器的架构模式，采用 Gradio 实现前端，是一个端到端的智能问答解决方案。下面详细描述该系统：

系统概述：

基于 Oekpgs 智能问答系统是一个用于提供用户与 OpenEuler 操作系统相关问题的智能回答的系统。它结合了自然语言处理技术和前端界面，为用户提供了一个直观、便捷的方式来获取与 Oekpgs 相关的信息和解答问题。系统的核心功能是通过大模型技术和信息匹配检索，将用户的问题转化为系统可以理解的查询，然后直接使用大模型进行问答，或者通过大模型外挂知识库的方式进行问答，或者从基于 Oekpgs 知识库中检索相关信息并提供答案。

前端（Browser）：

Gradio 前端界面：Gradio 是一个用于构建快速、易于使用的机器学习界面的 Python 库。在基于 Oekpgs 智能问答系统中，Gradio 用于构建用户友好的前端界面，用户可以在该界面上输入问题并查看系统的回答。

用户输入界面：用户在 Gradio 前端界面上输入问题或查询，界面包括对话式文本框，用户可以在文本框中键入问题并通过点击按钮提交问题。

后端（Server）：

ChatGLM 大模型模块：当用户提交问题后，前端将问题传递给后端的 ChatGLM 大模型模块。ChatGLM 大模型模块通过推理，对于用户提出的问题进行回答。

知识库查询模块：通过外挂知识库形式，将问题与后端向量化的文本进行匹配，从向量库中进行查询。知识库可能包括基于 Oekpgs 文档、常见问题解答、操作手册等相关信息。

信息检索模块：通过查询模块对于相关问题进行模糊搜索，检索到相关信息后，将生成一个或多个可能的答案，并将它们传递回前端。

Gradio 集成：后端将生成的答案传递给 Gradio 前端界面，以便用户查看答案。

该 B/S 架构的优势包括：

跨平台性：用户只需通过浏览器访问系统，无需安装额外的客户端软件，可

以在不同的操作系统和设备上使用。

部署简便：系统部署在服务器端，用户无需进行复杂的安装和配置，只需通过浏览器访问即可使用。

维护方便：系统的更新和维护可以集中在服务器端进行，用户端无需关注，降低了维护成本和工作量。

3 架构设计

基于 Oekpgs 智能问答系统采用 B/S（Browser/Server）架构，是一种基于浏览器和服务器的架构模式，适用于 Web 应用程序的开发，下图为 flask 作为后端系统的整体框架，而我们的系统通过使用 gradio，使得 B/S 架构的书写难度大大降低。

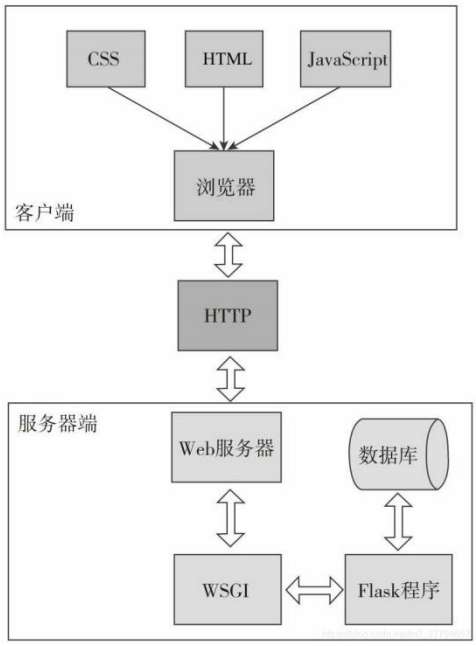

表 3-1 常见的 B/S 架构

3.1 前端架构

本部分将介绍基于 Oekpgs 智能问答系统的前端架构设计。前端架构是指系统前端部分的组织结构、技术选择和交互模式等方面的设计。

3.1.1 技术栈

Gradio 是一个用于构建快速、易于使用的机器学习界面的 Python 库，它提供了许多功能来快速搭建交互式界面。在基于 Oekpgs 智能问答系统的前端中，Gradio 是核心的前端框架。

以下是系统前端的技术栈：

Gradio：作为前端框架，Gradio 用于创建用户友好的界面，接受用户输入，向后端发送请求，以及显示后端返回的答案。

HTML/CSS：在 Gradio 界面中，使用 HTML 和 CSS 来自定义界面的外观和布局，以满足系统的需求。

JavaScript：JavaScript 与 Gradio 一起使用，以增强前端界面的交互性和功能。使用 JavaScript 执行客户端验证、动态元素生成等任务。

3.1.2 实现步骤

a. 创建 Gradio 界面：使用 Gradio 创建用户界面。定义多个文本框、按钮和其他交互元素，以便用户可以输入问题并与系统进行交互。同时，使用 HTML/CSS 自定义界面的外观。

b. 设置回调函数：在 Gradio 界面中，设置回调函数，以在用户点击提交按钮时触发。这个回调函数将获取用户输入的问题，并将其发送到后端以获取答案。

c. 与后端集成：在回调函数中，需要使用 HTTP 请求（通常是 POST 请求）将用户的问题发送到后端的 API 端点。后端将处理请求并返回答案。

d. 显示答案：一旦从后端收到答案，则将答案显示在 Gradio 界面上，以便用户查看。

3.2 后端架构

3.2.1 LLM 对话

支持直接与大模型对话，实现简单的信息交流和功能查询。

3.2.2 知识库问答模式

选择知识库名称后，即可开始问答，如有需要可以在选择知识库名称后上传文件/文件夹至知识库。

3.2.3 信息检索模式

针对一系列的问题进行问答，输入"当前可检索问题"即可查看相应所有问题，同时支持追问。

3.2.4 模型参数调整

支持对于模型参数的选择和调整。

4 前端设计

本部分详细描述 Web 程序的前端设计，包括页面布局、交互元素、样式和行为。

4.1 页面 1——LLM 对话页面

4.1.1 页面描述

支持直接与大模型进行对话，实现简单问答和功能询问等。

4.1.2 页面布局

图 4-1 LLM 对话界面展示

如图 4-1 所示，该页面主要为对话框和右侧导航栏。左侧对话框展示了系统的一些基本信息和说明情况，同时包含对话框，支持用户进行文本框的输入，对模型进行提问，同时模型会进行推理，在文本框上进行回答，类似于聊天界面。右侧导航栏显示当前模式，为“LLM 对话”，支持直接与系统助手进行对话，回答一些基本问题。

4.1.3 交互元素

文本框栏主要包括文本的输入，进行提问，同时会类似于聊天的形式给出回答。

4.2 页面 2——知识库问答界面

4.2.1 页面描述

图 4-2 知识库问答界面展示

支持新建知识库，选择知识库名称后，即可开始问答，如有需要可以在选择知识库名称后上传文件/文件夹至知识库。

4.2.2 页面布局

如图 4-2 所示，该页面主要为对话框和右侧导航栏。左侧对话框展示了系统的一些基本信息和说明情况，同时包含对话框，支持用户进行文本框的输入，对模型进行提问，同时模型会进行推理，在文本框上进行回答，类似于聊天界面。右侧导航栏显示当前模式，为“知识库问答”，可以通过新建知识库或者进行知识库的选择，实现文件上传，对于知识库的相关信息进行提问。

4.2.3 交互元素

支持新建知识库，如图 4-3 新建知识库展示所示。

图 4-3 新建知识库展示

支持文件上传，如图 4-4 上传文件界面展示所示。

图 4-4 上传文件界面展示

支持文件夹上传，如图 4-5 上传文件夹界面展示所示。

图 4-5 上传文件夹界面展示

4.3 页面 3——信息检索

4.3.1 页面描述

支持针对一系列的问题进行问答，输入"当前可检索问题"即可查看相应所有问题，同时支持追问。

4.3.2 页面布局

如**错误!未找到引用源。**所示，该页面主要为对话框和右侧导航栏。左侧对话框展示了系统的一些基本信息和说明情况，同时包含对话框，支持用户进行文本框的输入，对模型进行提问，同时模型会进行推理，在文本框上进行回答，类似于聊天界面。右侧导航栏显示当前模式，为“信息检索”，针对一系列的问题进行问答，输入"当前可检索问题"即可查看相应所有问题，同时支持追问。

4.3.3 交互元素

支持直接进行检索，如图 4-6 直接检索示意展示所示。

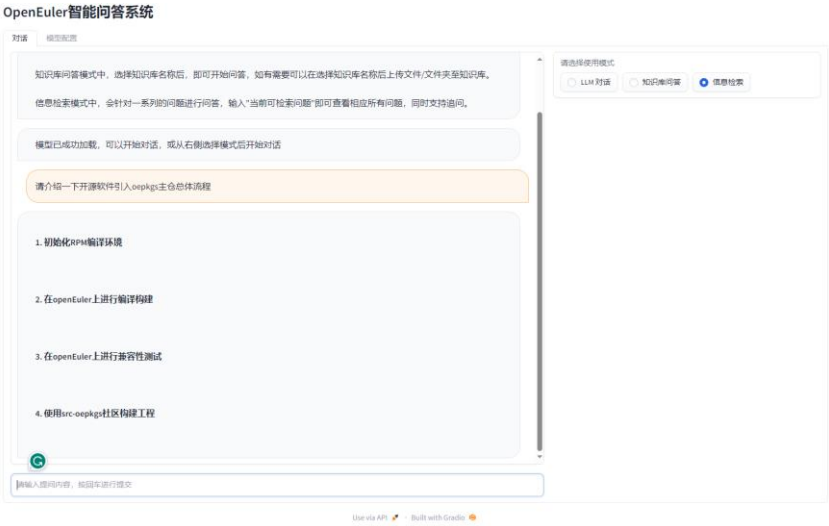

图 4-6 直接检索示意展示

支持查看可检索问题，如图 4-7 检索可询问问题示意所示。

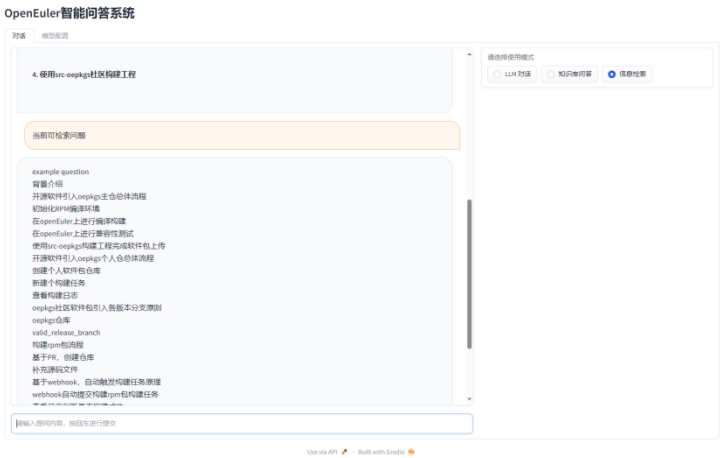

图 4-7 检索可询问问题示意

4.4 页面 4——模型配置

4.4.1 页面描述

该页面主要提供了对于模型进行选择，包括LLM模型选择，LLM对话轮数，Embedding 模型，向量匹配 top K 和重新加载模型按钮。

4.4.2 页面布局

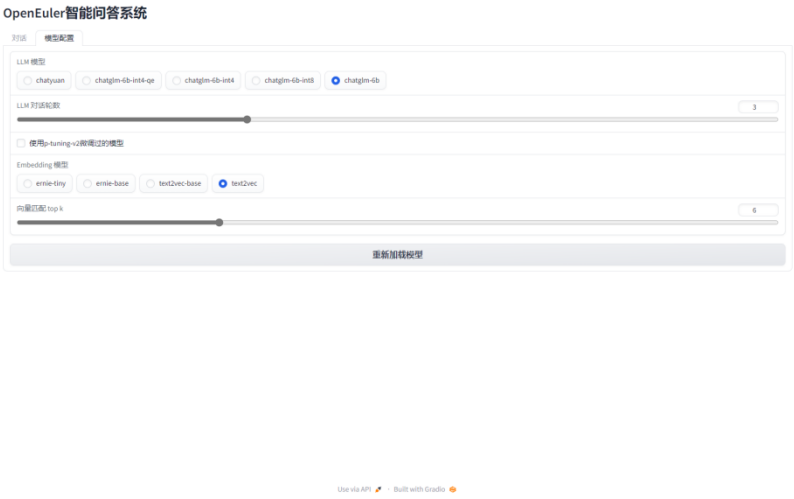

图 4-8 模型配置示意展示

如图 4-8 模型配置示意展示所示。

4.4.3 交互元素

该页面包括 LLM 模型选择，LLM 对话轮数，Embedding 模型，向量匹配 top K 和重新加载模型按钮等可以进行交互。

5 核心设计

5.1 ChatGLM 模型

本部分主要介绍对于训练基于 Oekpgs 智能问答系统的难点问题，也是关键问题，即使用大模型进行流利的问答，通过分析使用大模型进行对话的难点问题，进一步提出通过 ChatGLM 开源大模型实现对话的解决方案，进而实现对于问题的有效解决。

5.1.1 问题难点

chatbot 是这一两年最火的话题，是自然语言处理“王冠上的钻石”。其进行语言对话的难点源于自然语言处理（NLP）任务的复杂性和多样性，其中包括以下挑战：

1. 自然语言的多样性：自然语言表达方式丰富多样，人们在交流中使用不同的词汇、语法和文化背景。这使得理解和生成自然语言文本成为一个复杂的任务。
2. 上下文理解：在对话中，理解先前的对话历史对于生成连贯和有意义的回应至关重要。需要考虑上下文中的信息，以便正确解释用户的问题和需求。
3. 歧义处理：自然语言中经常存在歧义，一个问题可能有多个解释，而助手需要选择最合适的解释并提供相关的答案。
4. 语法和语义分析：正确解析语法结构和理解句子的语义是复杂的任务，尤其是在面对复杂句子和多义词汇时。
5. 个性化和上下文感知：智能助手需要能够个性化地回应不同用户的需求，并在对话中保持对用户的上下文感知，以提供更有针对性的答案。

综上所述，chatbot 面临着很多问题，包括对话上的准确度和语义的理解程度，同时对于上下文的理解和相关信息的把握存在着相当大的瓶颈，很难解决，需要更加强大的技术和资源去解决问题。

5.1.2 大模型的优势

大语言模型，如 GPT（Generative Pre-trained Transformer）系列，ChatGLM，

llama 等可以帮助解决上述难点的原因，即大模型的优势包括：

1. **语言模型的强大表示能力：**大语言模型可以在大规模语料库上进行预训练，学习到丰富的语言表示，能够更好地理解和生成自然语言文本，包括各种不同的语法和文本结构。
2. **上下文感知：**大语言模型可以捕捉上下文信息，理解先前的对话历史，并将其纳入生成回应的过程中，以提供更连贯和合理的答案。
3. **个性化：**通过在对话中使用用户的输入历史，大语言模型可以生成个性化的回应，使得对话更有针对性。
4. **自动学习语义和语法：**大语言模型可以自动学习语义和语法规则，因此能够更好地理解用户的问题，并生成符合语法和语义规则的回应。
5. **处理多样性和歧义：**大语言模型可以生成多样性的回应，帮助处理歧义问题，提供多种可能的解释和答案。

尽管大语言模型在解决这些问题上取得了显著进展，但仍然有一些挑战需要克服，例如模型的偏见、生成不准确的答案以及对话的连贯性问题。因此，在实际应用中，通常需要结合其他技术，以提高智能助手的性能和用户体验。

5.1.3 ChatGLM-6B

ChatGLM-6B 是一个开源的、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。结合模型量化技术，用户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）。ChatGLM-6B 使用了和 ChatGPT 相似的技术，针对中文问答和对话进行了优化。经过约 1T 标识符的中英双语训练，辅以监督微调、反馈自助、人类反馈强化学习等技术的加持，62 亿参数的 ChatGLM-6B 已经能生成相当符合人类偏好的回答，更多信息请参考我们的博客。欢迎通过 chatglm.cn 体验更大规模的 ChatGLM 模型。

为了方便下游开发者针对自己的应用场景定制模型，我们同时实现了基于 P-Tuning v2 的高效参数微调方法 (使用指南)，INT4 量化级别下最低只需 7GB 显存即可启动微调。

ChatGLM-6B 权重对学术研究完全开放，在填写问卷进行登记后亦允许免费商业使用。

尽管模型在训练的各个阶段都尽力确保数据的合规性和准确性，但由于 ChatGLM-6B 模型规模较小，且模型受概率随机性因素影响，无法保证输出内容的准确性，且模型易被误导。

5.1.5 小结

本部分提出了以 ChatGLM 模型为基础，搭建 chatbot 的对话模型，借助大模型的优势实现对话机器人的问答和交流。

5.2 langchain + chatglm

本部分主要探讨 chatbot 需求与实现，通过分析需求和数据现状，对比模型参数和资源进行模型分析，对于大模型的训练、预训练、微调和基于提示词的各种方案进行评估，最终提出使用 langchain + chatglm 的方案，实现基于本地知识库的 chatbot 机器人。

5.2.1 问题难点

本项目力图实现基于 Oekpgs 相关内容的智能问答系统，通过大模型的方式实现对于问题的解析和回答，对于大模型的处理方式包括训练，预训练，微调（fine-tune）以及基于提示词等形式实现对于问题的回答，然而各种方式都存在一定的难点问题。

基于大语言模型训练以实现问答任务确实面临一些挑战和难点。以下是其中一些主要的难点：

a. 数据量和质量：训练大语言模型需要大规模的文本数据来获得足够的知识和语言理解能力。这些数据必须涵盖各种主题和领域，以便模型能够回答各种类型的问题。同时，数据的质量也很重要，因为模型会学习输入数据中的错误信息和偏见。

b. 训练成本：训练大型语言模型需要大量的计算资源，包括高性能的 GPU 和 TPU。这些资源的成本很高，而且模型的训练可能需要数天甚至数周的时间。

c. 知识限制：语言模型的知识是基于其训练数据的，因此它们的知识是有限的，而且通常局限于其知识截止日期。模型可能不了解最新的信息或事件，也可能缺乏某些专业领域的知识。

d. 上下文理解：理解问题的上下文对于正确回答问题至关重要。有时，问题

可能会涉及多轮对话，需要模型保持对整个对话历史的理解。

e. 资源消耗：运行大型语言模型需要大量的计算资源，这可能不可行或不可承受的成本对于某些应用或组织来说。

总之，训练大语言模型以实现问答任务是一个复杂且具有挑战性的过程，对于我们的任务需求来说不仅是不必要的，同时也是无法实现的。

基于大语言模型的微调在某些方面确实可以用于专业领域的问答对话，但也存在一些难点和挑战，使得这个过程相对困难：

a. 数据稀缺性：在相关专业领域，可用于微调的数据相对稀缺。大语言模型通常需要大量的领域特定数据来适应该领域的术语、概念和语境。如果没有足够的数据，模型可能难以产生准确的答案。

b. 领域特定的知识：大语言模型虽然可以泛化到多个领域，但它们的知识通常是通用的，可能不涵盖某些专业领域的特定知识。这意味着模型可能会在处理特定领域的问题时产生错误的答案或缺乏深刻的理解。

c. 数据偏见：微调时使用的数据可能会包含与特定领域相关的偏见或错误信息，这可能会影响模型的性能。模型可能会学习这些偏见并在答案中反映出来。

d. 领域适应时间：微调大型语言模型需要时间和资源。在专业领域中，可能需要花费相当长的时间来创建一个高质量的领域特定数据集，并进行适当的微调和验证。

总之，虽然基于大语言模型的微调可以用于专业领域的问答对话，但需要克服许多挑战，包括数据稀缺性、领域特定知识、术语和上下文的理解、质量控制以及适应时间。解决这些问题需要投入大量的时间和资源，对于本项目要求在少样本，精准问答的项目不太适合。

最终，基于 prompt 的解决范式是最适合本项目的解决方案，同时需要考虑到 prompt 范式存在上下文段的长度限制，因此需要通过 langchain+ChatGLM 的范式，实现基于 prompt 的知识库问答。

5.2.2 langchain-ChatGLM 对话机器人

langchain-ChatGLM 是一个基于本地知识的问答机器人，使用者可以自由配置本地知识，用户问题的答案也是基于本地知识生成的。

（1）langchain

LangChain 是一个围绕大语言模型应用而开发的开源框架，可以将 LLM 模型、向量数据库、交互层 Prompt、外部知识、外部工具整合到一起，允许用户围绕大语言模型快速建 立管道和应用程序，直接与 ChatGPT 或 Huggingface 中的其他 LLM 相连，从而在一定程度上弥补模型本身的缺陷，提升用户的使用效率。

Langchain 可以很好地支持用户针对本地文档、数据进行总结、问答。在现有的模型、插件中，用户可以通过上传一篇 PDF 文档的形式令 LLM 针对这篇文档进行回答，原理类似于基于 prompt 范式实现对于信息的推理和问题的回答。

（2）chatglm-6b 模型

ChatGLM-6B 是一个基于 GLM 的生成式对话模型。它由清华大学的研究团队开发，旨在改进聊天机器人的生成质量和对话逻辑。ChatGLM-6B 采用了全新的训练方法，通过在大规模对话数据集上进行预训练，提升了模型的生成能力和对话质量。

ChatGLM-6B 具有以下几个关键优势：

1.生成质量：相较于传统的聊天 AI 模型，ChatGLM-6B 在生成质量方面表现出色。它能够生成更加自然、流畅且贴近人类的对话，提供了更好的用户体验。

2.对话逻辑：ChatGLM-6B 在对话逻辑方面的改进也是显著的。传统聊天 AI 往往会给出不连贯或无关的回应，而 ChatGLM-6B 则能更好地理解上下文，并生成有逻辑性的回复。

3.开放性：ChatGLM-6B 是一个开源项目，这意味着研究者和开发者可以自由地使用、修改和分发该模型，这有助于推动聊天 AI 领域的发展和创新。

（3）langchain+chatGLM

一种利用 langchain 思想实现的基于本地知识库的问答应用，目标期望建立一套对中文场景与开源模型支持友好、可离线运行的知识库问答解决方案。

受 GanymedeNil 的项目 document.ai 和 AlexZhangji 创建的 ChatGLM-6B Pull Request 启发，建立了全流程可使用开源模型实现的本地知识库问答应用。本项目的最新版本中通过使用 FastChat 接入 Vicuna, Alpaca, LLaMA, Koala, RWKV 等模型，依托于 langchain 框架支持通过基于 FastAPI 提供的 API 调用服务，或使用基于 Streamlit 的 WebUI 进行操作。

依托于本项目支持的开源 LLM 与 Embedding 模型，本项目可实现全部使用开源模型离线私有部署。与此同时，本项目也支持 OpenAI GPT API 的调用，并将在后续持续扩充对各类模型及模型 API 的接入。

（4）限制束搜索

限制束搜索是一种生成模型技术，用于控制生成结果的多样性和准确性。它通过维护一个候选序列集合，并根据模型生成的概率和之前的累计得分选择最佳的候选序列。限制束搜索可以应用一些约束条件，以约束生成结果的特定要求，如语法规则或模式。这样可以在生成过程中平衡结果的多样性和准确性，并获得符合要求的生成结果。

整体的模型框架如图 5-1 所示。

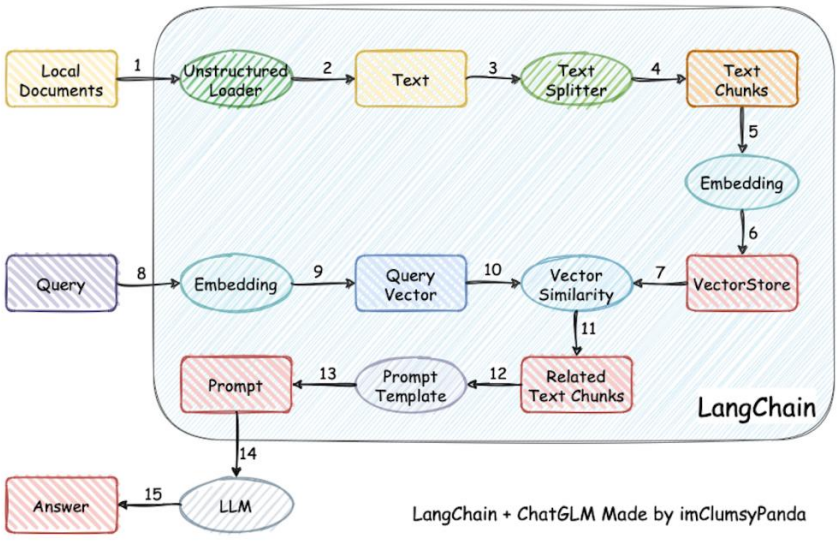

图 5-1 模型框架

5.2.3 工作流程

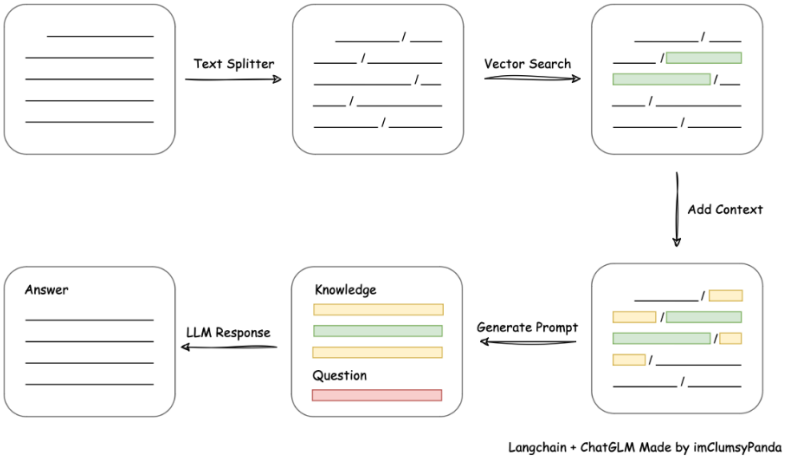

图 5-2 chatbot 工作流程

项目实现原理如图 5-2 所示，过程包括加载文件 -> 读取文本 -> 文本分割 -> 文本向量化 -> 问句向量化 -> 在文本向量中匹配出与问句向量最相似的 top k 个 -> 匹配出的文本作为上下文和问题一起添加到 prompt 中 -> 提交给 LLM 生成回答。

5.2.4 项目优化

重写文本分割器：语言模型通常受到可以传递给它们的文本数量的限制，因此将文本分割为较小的块是必要的。受限于 langchain 内部的文本分割器效果不好，对于我们的任务要求更加准确，完整的内容回答，因此需要对于重写 text splitter，帮助改善向量存储的搜索结果。

5.2.5 小结

本节对于 chatbot 需求与实现，依据模型参数和资源进行模型分析，对于大模型的训练、预训练、微调和基于提示词的各种方案进行评估，最终提出使用 langchain + chatglm 的方案，通过一定程度上的优化，实现基于本地知识库的 chatbot 机器人。

5.3 信息检索

本部分主要介绍受到模型参数量的限制，模型根据上下文回答问题的能力有限，无法实现对于问题精准，完整，具体的回答，因此需要通过更加精准的手段实现对于大语言模型的补充和改进，最终我们提出基于信息检索的方案，实现对

于相关专业领域问题的精准回答，对标业务需求，实现在保留对话效果的基础上，精准回答专业问题。

5.3.1 问题难点

基于上下文 prompt 实现的大型语言模型回答专业问题可能出现不准确和不具体的现象，这些问题与模型的限制和数据输入方式有关，具体的可能原因如下：

a. 不完整的上下文信息： 大型语言模型通常需要一个有限的上下文窗口来理解问题和生成答案。如果上下文信息不足够详细或不完整，模型可能无法正确理解问题的背景，导致不准确的答案。

b. 领域特定知识不足： 大型语言模型的知识通常是通用性的，对于某些专业领域的特定知识可能了解不深。这可能导致模型在处理专业问题时无法提供准确的、领域特定的答案。

c. 生成策略问题： 某些大型语言模型可能在生成答案时倾向于生成通用性、模棱两可或不确定的答案，而不是提供具体的、准确的信息。这可能与模型的生成策略有关。

d. 问题复杂性： 一些专业问题可能非常复杂，需要深入的推理和分析。大型语言模型可能在处理这些复杂问题时出现不准确性，因为它们通常更适用于生成通用性答案而不是解决复杂问题。

e. 模型参数问题：越少的模型参数对于语言的理解和回答能力越弱，更重要的是里理解上下文的能力也很弱，因此即使重复上下文的问题也很难。

为改善基于上下文 prompt 的模型在回答专业问题时的准确性，可以考虑以下方法：

a. 提供更详细和清晰的上下文信息，确保模型理解问题的背景。

b. 引入领域专家的反馈和审核，以改进答案的准确性。

c. 使用后处理技巧来筛选和优化模型生成的答案。

d. 确保微调数据集包含高质量、专业领域相关的内容，减少数据中的偏见和噪声。

e. 在处理复杂问题时，使用模型的答案作为参考，而不是绝对依赖，以进行进一步的验证和分析。

最终，我们选择通过基于信息检索的形式优化大模型生成，实现对于针对性

的专业问题，快速、准确、具体地检索答案，然后支持大模型进一步进行对话生成和问讯。

5.3.2 基于检索回答问题

通过基于信息检索的形式优化大型模型生成，以实现对针对性的专业问题的快速、准确、具体的答案检索，并支持大型模型进行对话生成和问讯，具体步骤如下：

a. 构建领域特定的知识库：首先，创建一个包含领域特定信息的知识库。这可以是结构化的数据库、语料库、文档集合或其他形式的知识存储。确保知识库包含有关专业领域的详细信息，包括术语、概念、事实和文档。当前项目支持自动化对于 md 文档实现问题和答案的解耦重组。

b. 实现信息检索： 开发一个信息检索模块，它能够有效地查询 question 库并返回与特定问题相关的文档或段落。可以涉及到使用信息检索技术如倒排索引、TF-IDF 权重、BM25 等来加速文档检索，当前文档数量有限，使用顺序索引和遍历检索。

c. 问题模糊搜索：对于 md 文档提取出来的问题，需要通过包括但不限于正则化，标志抽取，停用词删除等方式，对于问题提炼关键字，实现对于问题的模糊搜索，在保证准确率的情况下，尽可能实现问题的回答周全。

5.3.3 小结

本部分简要概述了在大语言模型难以实现对于专业问题的精确回答时，我们进一步通过信息检索的方式进行中和，实现对于专业问题的精确回答和准确提炼，使得回答具有可操作性。

6 模型实践及评估

6.1 模型实践简述

阶段一：MiniGPT4 阶段（2023 年 7 月 1 日至 2023 年 8 月 1 日）

MiniGPT4（微调）：准确率有限

MiniGPT4（预训练）：需要图片输入（pass）

阶段二：ChatGLM 阶段（2023 年 8 月 1 日至 2023 年 8 月 20 日）

ChatGLM（训练）：参数量过大，数据和设备不支持，最小化训练效果很差

ChatGLM（微调）：无法融入全部语义信息，对文本生成任务多种微调方式效果都很差

ChatGLM2（prompt）：文本要求长度有限

阶段三：langchain-ChatGLM 阶段（2023 年 8 月 20 日至 2023 年 9 月 20 日）

langchain-ChatGLM：效果较好，但准确度仍然有限，受限于模型参数，导致语义信息回复一致性仍有限

langchain-ChatGLM+检索：效果较好，回答专业问题时准确，但不够灵活，回答非专业问题时仍存在问题

阶段四：模型总结（2023 年 9 月 20 日至项目提交）

6.2 问题探究

6.2.1 MiniGPT4 和 ChatGPT 的区别，这两个模型算法上的优劣?

MiniGPT4 和 ChatGPT 的区别可以从几个方面来看，

从模型数据量来看，GPT3 和 GPT3.5 的数据量是惊人的，依靠 Wiki，GitHub 等，语料质量很高，MiniGPT4 能力有限；

从模型参数量来看，GPT3 的参数量是 175B，GPT3.5 的参数量未公布，但我预估肯定比这更多，GPT4 的参数量肯定会更加惊人，资源消耗量巨大。而目前官方发布的 MiniGPT4 的两大模型参数量为 7B 和 13B，毫无疑问参数量的大小直接决定了模型最终的效果，以及模型耗费的资源，虽说目前官方未开放 GPT4 的图像处理功能，但是其模型对图像的敏感程度肯定要比 MiniGPT4 高很多，同时训练的速度，包括针对图像文本生成的速度也会慢很多；

从模型架构来看，目前的 MiniGPT4 和 ChatGPT 都是基于自注意力机制和 Transformer 框架下的大语言模型。不同的是，GPT3.5 及以前的均不支持图片输入，因此其不包含多模态模型架构，但是 MiniGPT4 包含一个多模态架构，其中通过投影层将冻结视觉编码器 BLIP-2 与冻结的 Vicuna 对齐，将图片信息和文本信息统一，冻结感觉也是结合开源大模型，节省算力，提高模型效果比较好的方式。GPT4 也支持图像输入，但是目前还不确定其使用的多模态模型架构，我估计有可能和 MiniGPT4 相同，训练成本比较小；

从模型训练来看，ChatGPT 系列都是基于大量标注数据，通过大量优秀的训练范式进行预训练、微调，其中包括 prompt 范式，few-shot，PPO 强化，基于规则的奖励模型，知识图谱等，一步一步精调。MiniGPT4 也是基本的两段式训练，但是其模型中大部分参数都是冻结的，训不了。其在预训练的过程中，需要对齐图像和文本，同时进行训练，然而在对齐过程中，BLIP-2 基于对图像的理解会使得 Vicuna 生成能力降低，因此就需要更高质量的数据通过 prompt 范式进行微调。个人感觉，ChatGPT 在预训练后就可以达到很好的效果，而 MiniGPT4 在精调后，对于图像和文本之间的关系才能很好理解；

在鲁棒性、可解释性等其他方面来说，大语言模型都差不多，模型效果准确性等方面与参数量和数据量密切相关，对比也没有什么太大的意义。

总结两个模型在算法上的优劣，ChatGPT 确实在语言生成和推理方面能力很强，与其巨大的参数量、高质量数据、多种训练技术都密切相关；而 MiniGPT4 是站在开源大模型参数巨人的肩膀上，提出了将视觉编码器和 LLM 对齐的多模态训练框架，思路是非常好的。谈到算法优劣的话，MiniGPT4 和 GPT3，GPT3.5 直接对比意义不大，MiniGPT4 使用的是大模型训练好参数进行图片和文本对齐，而 GPT3，GPT3.5 是高强度训练仅支持文本的推理能力很强的大语言模型。MiniGPT4 和 GPT4 感觉也没办法比较，毕竟 GPT4 的多模态架构未知，而且其模型训练的算法肯定要优秀很多，肯定不会直接冻结参数训练，但是毕竟还没有开放使用。

所以个人感觉 MiniGPT4 是在 GPT4 开放图像理解使用之前比较好的体验多模态大模型的替代工具，也是资源受限下大规模模型训练，针对特定任务训练，对于多模态有要求的任务训练比较好的选择。

6.2.2 大模型的训练、预训练、prompt 和微调为模型带来了哪些能力?

大型模型的训练、预训练、prompt（提示）和微调各自为模型带来了一些特殊的能力和特点，以下是它们的主要特征：

预训练（Pretraining）能力：

通用语言理解和生成： 预训练使模型能够学习通用的语言理解和生成能力，从大规模文本数据中学习语法、语义、逻辑和常识。

广泛的知识： 模型通过大规模文本数据接触到多领域的信息，从而具备广泛的知识，可以处理各种主题和领域。

Prompt（提示）能力：

任务定制： Prompt 是一种用于自定义模型行为的机制。通过设计不同的 prompt，可以引导模型执行特定的任务，使其具备定制化的功能。

灵活性： Prompt 可以用于各种自然语言处理任务，包括问答、文本分类、文本生成等，因此具有广泛的应用灵活性。

微调（Fine-tuning）能力：

任务适应性： 微调允许模型在特定任务上进行优化，使其具备针对性的能力，例如，在特定领域的问答、情感分析、命名实体识别等任务上表现良好。

个性化： 微调可以使模型更好地适应用户或应用的需求，实现个性化的生成或分类。

每个阶段都为模型带来不同的特殊能力，使其适应不同类型的任务和应用场景。综合利用这些能力可以创建强大且多功能的自然语言处理系统，但也需要仔细考虑任务和数据，以确保模型在特定任务中表现出所需的性能。

7 项目分析和未来展望

7.1 项目分析

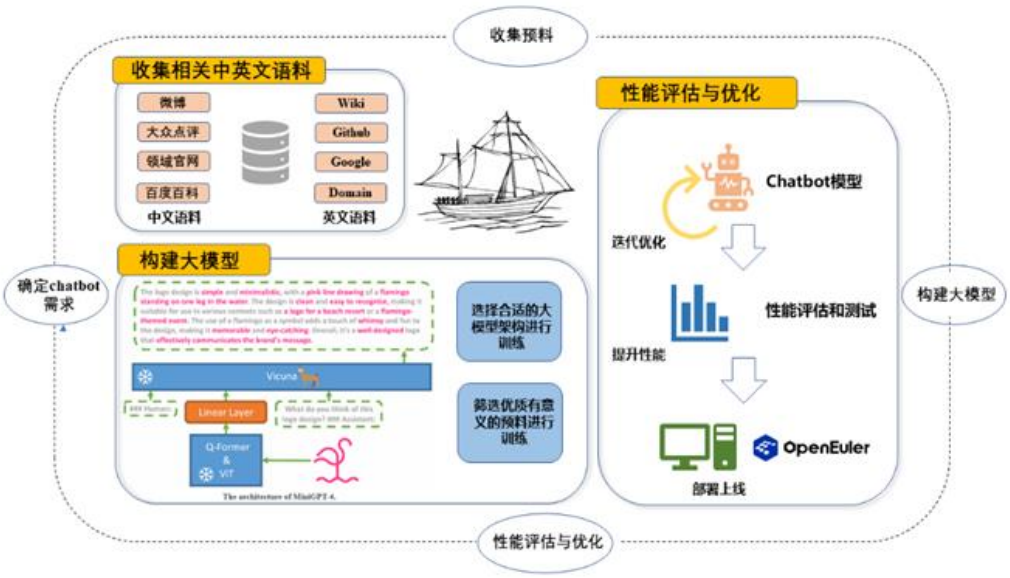

图 7-1 项目整体流程

7.1.1 收集预料

可以使用爬虫工具从网络上收集相关对话作为语料，利用现有的聊天记录或千赢平台等会话历史资料，在开源数据集中寻找大规模的训练集，然后具有针对性地进行数据标注。

当前项目使用平台提供的 md 文档，未来可以进行扩展或者使用其他格式的文档进一步提炼。

7.1.2 训练开源大模型

深入学习 GLM 相关设计理念，对于模型框架进行修改，调整超参数以充分利用计算资源提高模型性能。

根据收集到的预料对于开源大模型进行微调，利用迁移学习的思路，以提高大模型的性能。

根据特定任务，基于 prompt 提示词进行特定专业领域的学习，实现专业领域对话机器人。

将模型部署到云端，可以利用云端算力。

GLM 相关论文：

2103.10360.pdf (arxiv.org)

2210.02414.pdf (arxiv.org)

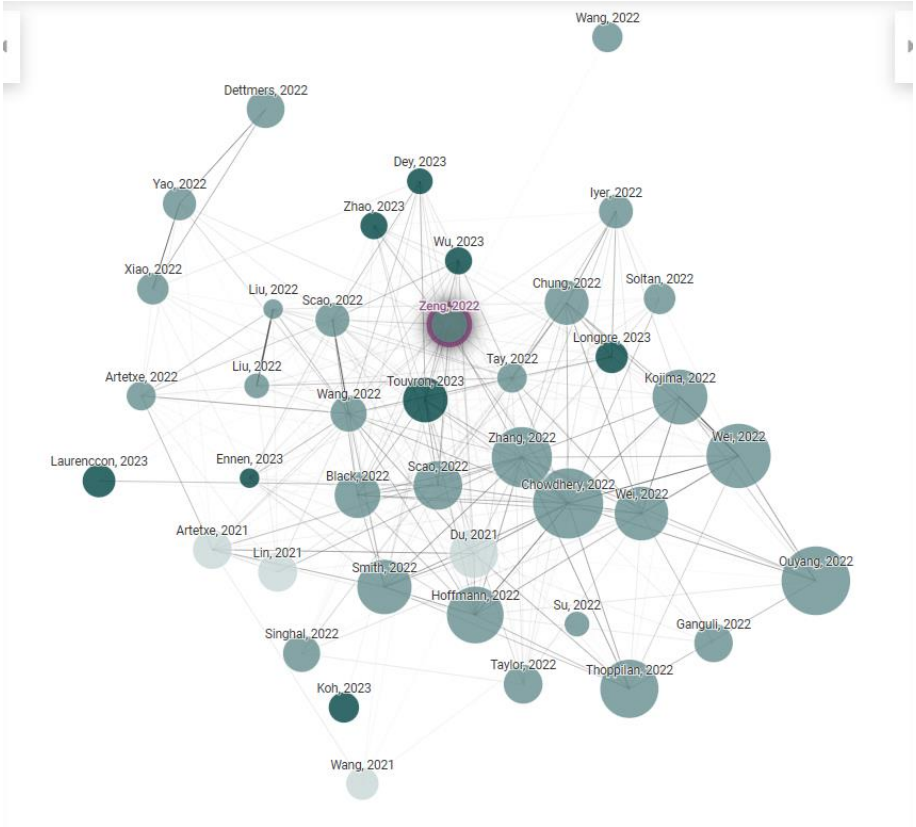

图 3.1 论文相关图谱

或者深度训练 MiniGPT4 模型，进行相关模型的深入实验。

目前已有资料包括官网地址：Minigpt-4，论文地址：2304.10592.pdf (arxiv.org)，代码地址： Vision-CAIR/MiniGPT-4: MiniGPT-4: Enhancing Vision-language Understanding with Advanced Large Language Models (github.com)

目前主要使用 ChatGLM-6B 模型，通过云端进行部署，但是 prompt 效果有限，未来可以考虑使用更大的模型或者更多的训练资源做微调。

7.1..3 完成前后端设计

进行模型的封装，通过前后端设计，使得其能够成为一个可以被使用的聊天机器人，最后可以部署到云端。

7.2 已完成的任务和解决的问题

7.2.1 已完成的任务

a. 在资源受限的条件下，完成了大语言对话模型的训练和使用；

b. 实现了后端服务器的功能，包括 ChatGLM 大模型模块、知识库查询模块和信息检索模块。这些模块协同工作，处理用户提交的问题并生成答案。

c. 构建了用户友好的前端界面。这个前端界面允许用户通过文本框输入问题，并通过按钮提交问题，以便系统进行回答。

7.2.2 需要解决的问题

在整个项目中，解决生成答案的精度问题是至关重要的，因为用户对于问答系统的期望是获得准确的、有用的答案。为了确保模型回答问题的准确程度，可以采取以下一系列措施，结合训练、预训练、微调和知识库/信息检索：

预训练大型语言模型：开始阶段，可以使用大型语言模型进行预训练。这些模型通过大规模的文本数据来学习语言理解和生成的通用能力，但可能还不具备领域特定知识。

微调模型： 针对基于 Oekpgs 智能问答系统，进行微调是关键一步。微调阶段将模型引导到特定领域和任务中。可以为模型提供基于 Oekpgs 相关的数据，包括问题和答案对、基于 Oekpgs 文档、常见问题解答等。通过微调，模型可以适应特定领域的语言和知识。

基于 prompt 的控制： 使用 prompt 作为控制机制，可以更好地引导模型生成特定类型的答案。设计具体的 prompt 格式，以引导模型回答基于 Oekpgs 相关问题，并在 prompt 中包含关键信息，以确保模型的答案具有准确性。

知识库/信息检索：引入知识库和信息检索模块是关键，特别是在处理专业问题时。将基于 Oekpgs 的文档、常见问题解答、操作手册等相关信息构建成知识库，并使用信息检索技术，以确保模型可以在知识库中查找到准确的答案。

定期更新知识库：知识库中的信息可能会发生变化，因此需要定期更新以保持其准确性和实用性。

质量控制：在整个流程中，实施质量控制措施，确保模型的输出答案是准确的、清晰的，并符合基于 Oekpgs 的标准和规范。

通过以上一系列措施，可以不断提高模型回答问题的准确度，使基于 Oekpgs 智能问答系统成为用户获取可信赖信息和解答问题的有力工具。这些措施结合了大型模型的能力和领域特定知识，以确保生成的答案具有高精度。

7.3 未来展望

a. 增加训练数据：为了提高模型的准确性，可以考虑增加用于微调的训练数据。这包括基于 Oekpgs 相关的问题和答案对，以及其他领域特定的文本数据，以便模型更好地理解基于 Oekpgs 的语境和知识

b. 生成模型的微调：改进生成模型的微调策略，使其能够更好地理解上下文和问题的复杂性。考虑使用更大的模型或更多的微调步骤

c. 考虑使用更加先进的模型，包括参数量更多的 baichuan-13B 等相关模型，可能上下文的能力会更好，对于 prompt 部分会更有帮助。

8 总结

在训练基于 Oekpgs 智能问答系统的过程中，我获得了一些宝贵的经验和感悟：在项目中，我深入了解自然语言处理领域，包括预训练模型、微调、生成模型等关键概念和技术。这些知识可以为我在 NLP 领域的进一步发展提供坚实的基础。在构建基于 Oekpgs 智能问答系统时，我深入了解基于 Oekpgs 操作系统的相关知识，以便于更好地理解和处理与该领域相关的问题。同时，用户体验和用户需求是项目成功的关键。我学会了聆听用户反馈、调整系统以满足用户需求，并不断优化模型，以达到更加高要求的需求，在项目不断要求精度的基础上不断提升自己。

总的来说，通过完成基于 Oekpgs 智能问答系统的训练过程，我不仅积累了有关 NLP 和软件工程的知识和经验，还提高了问题解决和用户导向的技能。这些收获和感悟将对我的职业发展和未来项目产生积极影响。