深圳大学考试答题纸

(以论文、报告等形式考核专用)  
二○二四～二○二五学年度第二学期

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程编号 | 1501990049 | 课程名称 | | 大模型技术及开发 | | 主讲教师 | | 陈小军 | 评分 |  |
| 学 号 |  | 姓名 |  | | 专业年级 | |  | | | |
|  | | | | | | | | | | |
| 教师评语： | | | | | | | | | | |
| 项目名称（体现领域）： | |  | | | | | | | | |

摘 要：

[简单列出已经实现的功能]

评分明细：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 评价维度 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 分项分数 |  |  |  |  |  |  |  |

**（此页不打印，由老师填写后打印）**

**深圳大学课程项目报告**

|  |
| --- |
| **课程名称： 大模型技术及开发**  **项目名称：**  **学 院： 计算机与软件学院**  **专 业：**  **任课教师： 陈小军**  **报 告 人： 学号：**  **提交时间：** |

**教 务 处 制**

1. **项目要求**

面向特定领域，融合Web数据与个性化知识库，设计并构建基于大语言模型的RAG（检索增强生成）智能问答系统，实现个性化、高效的信息交互体验。具体要求如下：

**1. 项目目标**

* **核心目标：** 成功设计并实现一个具备基本功能的RAG问答系统。
  + 能够接入用户指定的Web数据源进行信息爬取。
  + 能够允许用户上传或指定个人文档，构建个人知识库。
  + 实现基于用户提问，从混合知识库（Web数据+个人知识库）中检索相关信息，并生成连贯、准确答案的功能。
* **技术目标：**
  + 掌握RAG系统的基本原理和搭建流程，包括数据预处理、向量化、索引构建、检索模块和生成模块的实现。
  + 熟悉至少一种向量数据库的使用（如FAISS, ChromaDB, Milvus等）。
  + 掌握至少一种大型语言模型（如开源的ChatGLM, Llama, Qwen或商业闭源模型）等的直接调用或API调用，及其提示工程。
  + 学习并实践Web数据爬取与清洗技术。
* **扩展目标（选做，但建议尝试）：**
  + **模型微调：** 针对特定知识领域或问答风格，对检索模型或生成模型进行微调，以提升系统性能。例如，微调Sentence Transformer模型以提高特定领域文本的检索相似度，或微调LLM以生成更符合特定格式的答案。
  + **用户界面：** 开发一个基础的Web用户界面（UI），方便用户进行提问、查看答案、管理知识库等操作。

**2. 系统核心功能模块设想**

1. **数据采集与预处理模块：**
   * **Web数据爬取子模块：** 允许用户输入URL，爬取指定网页内容。需要处理HTML解析、文本提取、数据清洗等。
   * **个人知识库导入子模块：** 支持上传常见文档格式（如.txt, .md, .pdf等），并进行文本提取和预处理（如分块/chunking）。
2. **知识库构建与向量化模块：**
   * **文本向量化子模块：** 使用预训练的语言模型（如Sentence Transformers）将处理后的文本块转换为向量表示。
   * **向量存储与索引子模块：** 将文本向量及其元数据（如来源、原始文本等）存入向量数据库，并构建高效的相似度检索索引。
3. **RAG核心引擎模块：**
   * **查询理解与向量化子模块：** 对用户输入的自然语言问题进行理解和向量化。
   * **信息检索子模块（Retriever）：** 根据问题向量，在向量数据库中执行相似度搜索，召回最相关的文本块作为上下文信息。
   * **答案生成子模块（Generator）：** 将原始问题和检索到的上下文信息组合成Prompt，输入给大型语言模型（LLM），生成最终答案。
4. **用户交互模块：**
   * **前端界面：** 提供用户输入问题、展示答案、管理知识库（如添加/删除数据源）的界面。可以使用Streamlit、Gradio、Flask/Django等框架实现。
5. **模型微调模块（选做）：**
   * **数据准备子模块：** 收集或构建用于微调的数据集（如特定领域的问答对、相关文档对等）。
   * **微调执行子模块：** 使用Hugging Face Transformers等库，对选择的预训练模型（检索模型或生成模型）进行微调。

**3. 技术选型初步建议**

* **编程语言：** Python
* **Web爬虫：** requests, BeautifulSoup4, Scrapy (可选，更强大)
* **文本处理与向量化：** LangChain, LlamaIndex, sentence-transformers, Hugging Face Transformers
* **向量数据库：** FAISS (本地，轻量级), ChromaDB (本地，易用), Milvus (更强大，可分布式)
* **大型语言模型（LLM）：**
  + 开源模型：ChatGLM系列, Qwen系列, Llama系列 (需要本地部署能力或云GPU)
  + API模型：OpenAI API, Gemini API (可能涉及费用，但部署简单)
* **用户界面：** Streamlit, Gradio, Flask + HTML/CSS/JS
* **微调（可选）：** PyTorch, TensorFlow, Hugging Face Transformers (Trainer API)

**4. 项目实施步骤建议**

**第一阶段：理论学习与环境搭建**

* + 深入学习RAG原理、向量数据库、LLM应用。
  + 搭建Python开发环境，安装所需库。
  + 完成一个最小化的RAG流程（使用少量本地文本数据）。
* **第二阶段：核心功能开发**
  + 实现Web数据爬取与个人文档导入功能。
  + 集成向量数据库，完成知识库构建与向量化流程。
  + 实现RAG核心引擎，能够基于检索到的信息生成答案。
  + 初步测试与调试。
* **第三阶段：系统优化与界面开发**
  + 优化检索效果和答案生成质量（如调整Prompt、尝试不同模型）。
  + 开发基础的用户交互界面。
  + 进行系统集成测试。
* **第四阶段：微调探索与报告撰写 (若选择微调)**
  + 研究微调方法，准备微调数据集。
  + 执行模型微调，并评估微调效果。
  + 撰写项目报告，整理代码和文档。
* **第五阶段：总结与答辩准备**
  + 完善项目文档，准备演示。

1. **考核标准（总分100分）**

1) 给出项目目标与任务分解 （10分）。

2）给出系统架构设计图及模块功能概述（10分）。

3）给出项目使用的关键技术与工具，如大模型名称及版本、向量数据库、大模型调用方式、前端的实现方式（10分）。

4) 给出系统实现的关键模块及关键代码（20分）。

5）给出系统的界面实现、5个以上的示例问答结果（20分）。

6）现场演示（30分）。

7）团队贡献系数（0.6-1）。

1. **相关时间**
2. **项目设计和项目报告由不超过5人的组队完成，且不同组之间选择的领域不同。如存在抄袭行为，考试成绩为零分。**
3. **截止时间：2024年6月30日23：59。**
4. **在截止时间之前需要提交的内容：**
   1. **提交本《项目报告》电子版以及汇报PPT；6月30日23：59前提交到blackboard系统。**
   2. **提交本《项目报告》打印版；6月27日上课时交给老师。（第一页《答题纸》单面打印，其他内容双面打印）。**
   3. **如果《项目报告》需要修改，于6月30日23：59前提交。**
   4. **逾期提交无成绩（以blackboard系统的电子版提交时间为准）**
5. **本学期最后一次课（2025年6月27日）进行课堂汇报。**

**注:第3-5页提交报告时删除。**

**报告正文**

**一、团队分工情况**

（请在此处详细描述团队成员的具体分工等）

1. **项目目标与任务分解**

（项目所要达到的目标，以及涉及到的具体开发实现任务）

1. **系统架构设计图及模块功能概述**

（项目所设计的系统架构图及包含的模块功能描述）

1. **项目使用的关键技术与工具**

（项目所使用的关键技术与工具，如大模型名称及版本、向量数据库、大模型调用方式、前端的实现方式）

1. **关键模块代码实现**

（项目涉及的关键模型的代码实现，给出关键代码截图 ）

1. **系统界面实现及示例**

（项目的系统界面实现，以及5个以上的示例结果展示）

1. **项目总结**

（此处写写你的所想、所得）