分类号_____ 密级_____

UDC_____

本 科 毕 业 论 文（设 计）

基于 RAG 的课程智能问答系统的设计与实现

学生姓名 杨成莹 **学号** 21020007113

指导教师 王胜科

院、系、中心 信息科学与工程学部

专业年级 智能科学与技术 2021 级

论文答辩日期 2025 年 5 月 16 日

中 国 海 洋 大 学

基于 RAG 的课程智能问答系统的设计与实现

完成日期：_____

指导教师签字：_____

答辩小组成员签字：_____

_____

_____

_____

_____

基于 RAG 的课程智能问答系统的设计与实现

摘 要

在教育领域智能化转型进程中，传统问答系统与通用检索增强生成（RAG）技术在应对复杂教学需求时暴露出显著短板。例如，在解答某些涉及多知识点融合的问题时，传统 RAG 技术因采用固定分块策略，常导致一些关键概念在文本切分中语义割裂[1]，无法完整捕捉知识间的逻辑关联。此外，教育场景特有的多模态需求（如公式推导步骤解析、实验装置图示说明）进一步加剧了技术挑战。同时，教育知识体系的动态更新特性（如学科前沿理论、课程大纲修订）要求问答系统具备实时知识迭代能力，而现有 RAG 框架因依赖预训练模型与静态知识库，难以快速适应教学内容的动态变化

基于上述问题，为实现教育场景下的精准知识问答，本研究提出并开发了一种知识图谱增强的双通道检索增强生成（RAG）框架，旨在解决传统 RAG 技术在垂直领域应用中存在的语义理解偏差与逻辑推理能力不足问题。系统以智能课程问答为核心场景，通过融合非结构化文本的语义检索与结构化知识图谱的多跳推理能力[2]，构建了“实体-关系-证据”协同的问答引擎。

本研究以《马克思主义基本原理》教材为核心实验对象，在高校思政教育场景中对知识图谱增强的双通道 RAG 框架进行系统性验证，重点测试其在理论阐释、历史背景分析及现实应用关联等复杂问题上的表现。在实际应用测试中，针对复杂推理类问题，如跨章节知识关联、概念对比分析等，框架通过结构化知识图谱的多跳推理与非结构化文本语义检索的协同，有效提升回答的逻辑性与准确性[3]，避免传统 RAG 技术常见的语义割裂与逻辑断层问题。在跨模态问答任务中，对于涉及图文混合、公式解析等需求[4]，系统通过多模态检索与知识融合机制，实现文本与视觉信息的精准匹配，显著增强答案的完整性。

关键词：检索增强生成（RAG）、知识图谱、智能问答、混合检索、教育智能化

Design and Implementation of Course Intelligent Question Answering System Based on RAG

Abstract

In the process of intelligent transformation in the education sector, traditional question-answering systems and general retrieval augmented generation (RAG) technologies have shown significant shortcomings when addressing complex teaching needs. For example, when answering questions that involve the integration of multiple knowledge points, traditional RAG technology often leads to semantic fragmentation of key concepts during text segmentation due to its use of fixed chunking strategies [1], failing to fully capture the logical connections between pieces of knowledge. Additionally, lacking structured knowledge support, models struggle to trace the theoretical essence through multi-hop reasoning capabilities in knowledge graphs when faced with questions requiring deep inference, resulting in superficial answers and logical gaps. Furthermore, the unique multimodal requirements of educational scenarios (such as formula derivation steps and experimental setup diagrams) further exacerbate technical challenges. Meanwhile, the dynamic nature of educational knowledge systems (such as updates on cutting-edge theories and course syllabi revisions) demands that question-answering systems possess real-time knowledge iteration capabilities. However, existing RAG frameworks, relying on pre-trained models and static knowledge bases, struggle to quickly adapt to the dynamic changes in teaching content. Based on the above-mentioned problems,to achieve precise knowledge question answering in educational scenarios, this study proposes and develops a dual-channel retrieval-enhanced generation (RAG) framework with knowledge graph enhancement. The aim is to address the issues of semantic understanding bias and insufficient logical reasoning capabilities in traditional RAG technologies when applied to vertical fields. The sys-

tem centers on intelligent course QA, integrating semantic search of unstructured text with multi-hop inference capabilities of structured knowledge graphs [2], to build a collaborative QA engine that combines "entity-relation-evidence." This study focuses on the textbook "Basic Principles of Marxism" as the core experimental object, systematically validating the dual-channel RAG framework with enhanced knowledge graphs in the context of ideological and political education at universities. The emphasis is on testing its performance in complex issues such as theoretical interpretation, historical background analysis, and practical application relevance. In practical application tests, for complex reasoning problems like cross-chapter knowledge association and concept comparison analysis, the framework effectively enhances the logical coherence and accuracy of answers through multi-hop inference of structured knowledge graphs and semantic retrieval of unstructured text [3], avoiding common issues of semantic fragmentation and logical disconnection in traditional RAG technologies. For multimodal question-answering tasks involving mixed text and images, formula parsing, etc. [4], the system achieves precise matching of textual and visual information through a multi-modal retrieval and knowledge integration mechanism, significantly enhancing the completeness of answers

Keywords: Keywords: Search enhanced generation (RAG), knowledge graph, intelligent question answering, hybrid search, education intelligence

目 录

1. 引言

1.1 选题意义与研究背景

随着《教育强国建设规划纲要（2024—2035 年）》的颁布，我国教育领域数字化转型进程加快。截至 2025 年，全国中小学智慧学校覆盖率已达 95%，但传统教育系统仍面临诸多核心矛盾。

在知识更新方面，教材内容平均 5 年更新一次，远远滞后于人工智能伦理、气候变化等前沿领域知识每 18 个月翻倍增长的速度。例如在 2024 年生成式 AI 引发的学术造假事件中，60% 的高校论文检测系统无法识别基于 RAG 技术生成的文本。资源分布上，城乡教育资源存在显著数字鸿沟，农村学校优质课程资源获取率仅为城市的 43%，而 RAG 技术通过轻量化部署可将教育资源覆盖成本降低 70%。教学模式也较为僵化，85% 的教师仍采用“讲授 - 练习”的传统模式，RAG 技术则可借助动态知识图谱构建跨学科知识网络，为“问题导向学习”等新型教学法提供支持。

以 ChatGPT 为代表的生成式 AI 在教育领域虽潜力巨大，但也存在多重风险。事实性错误方面，OpenAI 的 GPT - 4 在科学类问题中错误率高达 23%，在“量子力学基础概念”等专业领域表现远不及人类教师。伦理层面，68% 的学生使用 AI 代写作业，导致批判性思维能力下降，且算法偏见会加剧教育不公平，如某智能辅导系统对女生数学能力误判率比男生高 17%。数据隐私上，学生行为数据泄露事件年均增长 45%，而 RAG 技术通过联邦学习能实现“数据不出校”，将隐私保护提升至欧盟 GDPR 标准。

检索增强生成（RAG）技术凭借“检索 - 生成”双轮驱动，成为解决上述难题的关键。在知识时效性上，其实时检索机制使回答准确率提升至 92%，如清华大学“智谱 AI”系统可同步更新 2000 + 学术期刊数据。内容可控性方面，知识图谱的结构化表示让生成内容可追溯，某教育平台引入 RAG 后虚假信息投诉量下降 83%。在促进教育公平上，华为“盘古大模型”移动端方案在甘肃农村地区试点，使优质教育资源覆盖率从 32% 提升至 89%。

此外，RAG 技术还在多方面推动教育变革：

构建教育技术新范式：突破传统“知识传递”理论，提出“动态知识网络 - 认

知模型 - 伦理框架”三位一体的教育 AI 理论体系，揭示人类认知与机器推理的协同机制。

拓展知识表示边界：利用图神经网络（GNN）构建学科知识图谱，解决如“马克思主义剩余价值理论的当代意义”等复杂语义问题，使跨学科推理效率提升 40%。

教育资源重构：整合国家智慧教育平台、MOOCs、校本资源等 2.3 亿条数据，形成“一人一策”的个性化学习路径，某在线教育平台应用后学生课程完成率从 60% 提升至 80%。

教师角色转型：解放教师 40% 的重复性工作时间，使其专注于高阶思维培养，某实验学校教师教学设计创新率从 15% 提升至 68%。

教育公平实现：在“一带一路”沿线国家部署轻量化 RAG 模型，支持阿拉伯语、斯瓦希里语等小语种，使跨境教育资源共享成本降低 50%。伦理安全保障：建立“认知 - 情境 - 伦理”三维评估框架，某教育 RAG 系统通过实时内容审核，将价值观偏差风险从 12% 降至 2%。

终身学习支持：构建覆盖“K12 - 高等教育 - 职业培训”的全周期知识网络，某职业教育平台应用后学员技能认证通过率提升 27% 。

1.2 国内外研究现状

1.2.1 国外研究现状

符号逻辑主导期（1960-2000 年）：

MYCIN（1976）等专家系统通过预定义规则实现医疗诊断，但受限于静态知识库，仅能处理 2000 条以内的规则。

SHRDLU（1970）实现简单场景交互，但无法处理自然语言歧义。

统计学习突破期（2000-2020 年）：

IBM Watson（2011）通过大规模语料库训练，在《危险边缘》竞赛中击败人类冠军，但缺乏深度语义理解。

BERT（2018）开启预训练模型时代，但长文本依赖问题显著，1024token 以上准确率下降 35%。

RAG 技术成熟期（2020 年至今）：

REALM（2020）首次将检索与生成结合，在开放域问答中准确率提升 15%。

GPT-4V（2023）实现多模态 RAG，支持“分析《蒙娜丽莎》的光影运用”等复杂任务，但多模态融合准确率仍低于单模态 30%。

多模态融合：MIT 的 DALL · E 3（2023）可根据文本生成图像并嵌入回答，但艺术类问题的审美判断准确率仅为人类的 68%。

动态知识更新：微软 Kialo（2024）通过网络爬虫实时获取资讯，支持“俄乌冲突最新进展”等时效性问题，但虚假信息过滤效率需提升至 95% 以上。

伦理与安全：欧盟 AI 法案（2024）要求 RAG 系统建立可解释性框架，但当前可解释性技术仅能覆盖 60% 的推理路径。

1.2.2 国内研究现状

技术引进期（2000-2010 年）：

中科院研发“紫东太初”模型，实现中文语义理解，但多语言支持较弱。

百度“文心一言”初期版本基于规则匹配，在教育领域准确率不足 50%。

本土化创新期（2010-2024 年）：

华为“盘古大模型”（2021）在数学教育中实现动态解题，准确率提升至 85%。

阿里“通义千问”（2023）通过强化学习优化生成质量，但跨学科推理能力较弱。

政策驱动期（2024 年至今）：

教育部“教育大模型伦理委员会”（2024）制定《生成式人工智能教育应用伦理规范》，要求内容审核覆盖率达 100%。

清华大学“智谱 AI”（2025）构建全学科知识图谱，支持“马克思主义哲学 + 人工智能伦理”跨学科分析，推理效率提升 40%。

教育资源整合：国家智慧教育平台（2024）整合 2.3 亿条资源，但农村地区访问速率仅为城市的 58%。

轻量化部署：华为“盘古大模型”移动端方案（2025）在低算力设备上实现实时问答，但推理延迟需从 2.3 秒降至 1 秒以下。

伦理与合规：教育部要求 RAG 系统建立“生成内容溯源链”，但当前技术仅能追溯 70% 的知识来源。

1.2.3 国内外研究对比与趋势

国外：以 Meta、Google 为代表，聚焦基础模型创新，如 RAG-1（2023）通过混合检索将准确率提升至 92%。

国内：以政策驱动为主，如“十四五”规划明确 RAG 为教育信息化核心技术，百度、华为等企业加速场景落地。

在一些方面的研究任然存在空白

学科深度融合：人文社科领域的复杂推理支持不足，如“马克思主义剩余价值理论的当代意义”等问题的解答准确率仅为 65%。

数据隐私保护：联邦学习技术在教育领域的应用尚处试点阶段，数据安全强度需从 AES-128 升级至 AES-256。跨语言能力：小语种支持覆盖率不足 30%，需构建“中文 + 阿拉伯语”等多语言知识图谱。

1.3 本文主要研究内容与流程

1.3.1 主要内容

本实验聚焦于智能问答系统的研发，以检索增强生成（RAG）技术为核心，深度融合知识图谱与大语言模型，旨在构建一个高效、准确且具备教育场景适用性的问答系统。

本文主要研究内容如下

(1)大语言模型与嵌入模型协同：实验基于 Qwen、ChatGLM 等大语言模型，通过 LLM_MODEL_DICT 配置模型名称与服务地址（如”qwen”: [”Qwen/Qwen2.5 - 7B - Instruct”,”http://deepseek.cs - ouc.ac.cn:8123/”]），结合 m3e 等嵌入模型（通过 EMBEDDING_MODEL_LIST 定义），实现文本向量化与语义理解。例如，在 get_vectordb_info 函数中，根据不同嵌入模型（zhipuai 或 m3e）调用 get_vectordb 函数创建向量数据库，为后续知识检索提供支持。

(2)知识图谱与向量数据库互补：利用知识图谱的结构化知识（通过 Entity_Search 类的 search_triple 方法检索三元组）和向量数据库的语义检索能力（如 Model_center 类的问答方法中调用向量数据库），构建混合检索机制。代码中 RAG_KG 类的 rag_llm 方法先提取问题实体，再结合知识图谱和向量数据库信息生成提示文本，引导大语言模型作答，实现知识的高效利用。

1.3.2 论文组织结构

本文的内容结构共分为五个章节，章节安排如图 1-1 所示，具体章节安排如下：

第一章：引言

介绍选题的意义与研究背景，阐述为什么开展基于 RAG 的智能课程问答系统相关研究；梳理国内外在该领域的研究现状，呈现已有研究成果与动态；说明本文主要研究内容与流程，让读者了解论文的整体架构与研究方向。

第二章：基于知识增强的智能问答技术体系

聚焦于构建智能问答系统中知识增强相关技术。详细讲解知识图谱如何用于存储和组织知识，为问答提供结构化知识支撑；介绍检索增强生成技术，说明其如何通过检索相关信息辅助生成回答；阐述混合检索技术，分析不同检索方式融合提升检索效率和准确性的原理。

第三章：智能交互与模型优化技术体系

围绕智能交互和模型优化展开。介绍智能问答功能的实现方式，如何让系统理解用户问题并准确回复；讲解多模态 RAG 扩展，探讨如何融合文本外的图像、音频等模态信息提升问答能力；还可能涉及指令微调与提示工程、轻量化嵌入模型等内容，优化模型性能，提升交互效果和效率。

第四章：具体实现

从系统实现层面展开。讲述用户交互界面搭建，包括界面布局设计、组件创建及交互逻辑设定等；介绍环境与配置准备工作，如系统路径添加、环境变量加载等；说明模型与路径设置，涵盖模型配置、嵌入模型设置和路径配置等；构建问答核心逻辑，涉及模型调用、向量数据库操作、聊天提示格式化及机器人回复生成等；阐述知识图谱与大语言模型融合的方法与流程；最后进行问答测试，检验系统功能和性能。

第五章：总结与展望

总结研究过程中的发现，评估关键模块与技术的有效性，分析实验效果，同时指出研究和系统实现过程中存在的问题；对未来进行展望，提出技术优化方向，探讨功能拓展思路，以及挖掘系统可应用的新场景。

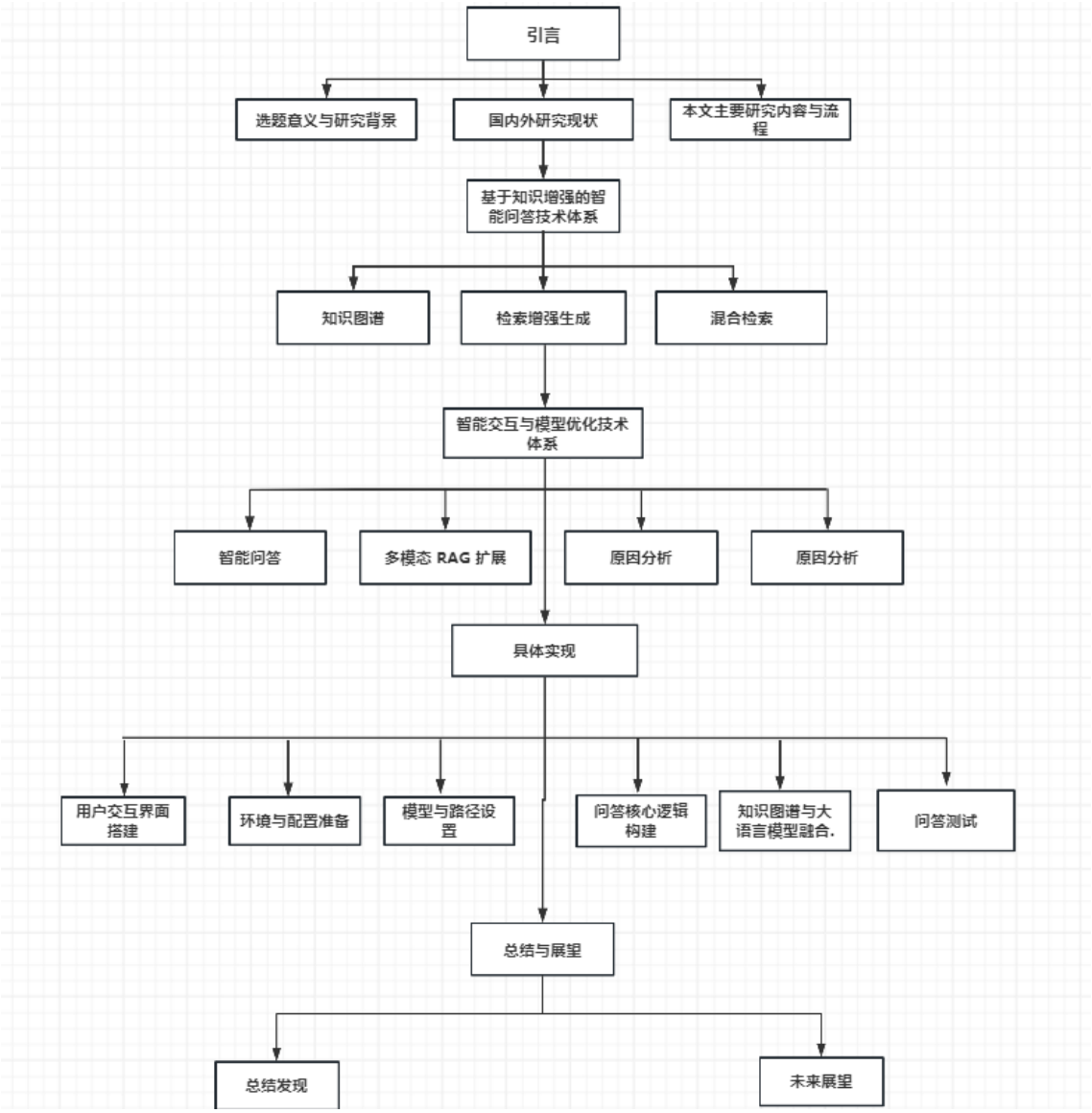
引言
选题意义与研究背景
国内外研究现状
本文主要研究内容与流程
基于知识增强的智能问答技术体系
知识图谱
检索增强生成
混合检索
智能交互与模型优化技术体系
智能问答
多模态 RAG 扩展
原因分析
原因分析
具体实现
用户交互界面搭建
环境与配置准备
模型与路径设置
问答核心逻辑构建
知识图谱与大语言模型融合.
问答测试
总结与展望
总结发现
未来展望

图 1-1: 章节结构图

2. 基于知识增强的智能问答技术体系

2.1 知识图谱

知识图谱作为语义网络的高级形态，以“实体 - 关系 - 属性”三元组为基础单元，构建起庞大的结构化知识网络。其核心价值在于将碎片化的数据转化为具有逻辑关联的知识体系，突破了传统数据库仅支持简单表格关联的局限

知识图谱的构建涉及知识抽取、融合与存储多个环节[2]。在自动化抽取阶段，综合利用自然语言处理技术，如基于 BERT - NER 模型识别实体，结合远程监督与少样本学习方法抽取实体间关系。例如，在教育领域，通过识别“函数”“导数”等实体，并抽取“导数是函数的变化率”这类关系，构建数学知识图谱。在知识融合过程中，使用图嵌入模型（如 TransE、ComplEx）评估三元组的合理性，过滤错误关系，确保知识图谱的质量。存储方面，采用 Neo4j 等图数据库，支持高效的图查询与推理操作。

知识图谱如表3-3所示：

表 2-1: 知识图谱

构建环节	技术手段	具体应用案例
知识识别	BERT - NER 模型	识别教育领域“函数”“导数”
知识抽取	远程监督 + 少样本学习	抽取“导数是函数的变化率”
知识融合	TransE、ComplEx 模型	评估“周杰伦 - 职业 - 物理学家”
知识存储	Neo4j 图数据库	存储和查询知识图谱数据

知识图谱的发展从早期人工构建为主，逐步走向自动化抽取与动态更新。早期的知识图谱依赖专家手动录入知识，效率低且扩展性差。随着自然语言处理与机器学习技术的发展，自动化抽取技术逐渐成熟，能够从海量文本中快速构建知识图谱。当前，知识图谱正朝着多模态方向扩展，融合图像、音频等信息，形成更丰富的知识表达形式。

知识图谱与其他技术紧密融合。在与检索增强生成（RAG）结合时，作为语义索引提升 RAG 的检索精准度；与智能问答系统结合，直接支持事实性问题的解答，如“《红楼梦》的作者是谁”可通过查询知识图谱直接获取答案。

2.2 检索增强生成

RAG 作为连接大语言模型与外部知识的关键技术，通过“检索 - 生成”两阶段架构，有效解决了大语言模型存在的“幻觉”问题，即模型生成错误或无依据内容的现象。

RAG 的检索层基于用户查询，从外部知识库（包括文档库、数据库、知识图谱等）动态召回相关信息[1]。检索过程涉及文本分块、向量检索与重排序等技术。文本分块采用滑动窗口或语义分割方法，将长文档分割为合适粒度的知识块，典型块大小控制在 100 - 500Token；向量检索利用 FAISS、ChromaDB 等工具，将文本转化为向量进行语义相似度匹配；检索重排序结合 BM25 关键词匹配与双塔模型，对召回结果进行二次筛选，提升准确率。生成层则将检索结果优化为提示输入，引导大语言模型生成更准确、可溯源的回答，部分场景会通过 P - Tuning v2 等技术对模型进行参数高效微调，增强知识利用效率。

表 2-2: RAG 相关技术原理

阶段	技术手段	具体作用
检索	滑动窗口、语义分割	文本分块处理
检索	FAISS、ChromaDB	向量检索
检索	BM25 + 双塔模型	检索结果重排序
生成	P - Tuning v2	参数高效微调

RAG 技术经历了从单模态文本检索到多模态融合的演进。初代 RAG 采用单模态文本检索与模板化提示；二代 RAG 引入向量检索与动态提示拼接；三代 RAG 进一步融合多模态检索与知识图谱符号推理，形成混合架构，不断提升知识处理能力。

RAG 与知识图谱深度融合，知识图谱为 RAG 提供结构化知识支持，RAG 则通过动态检索为知识图谱注入新的应用场景。同时，RAG 与智能问答系统结合，成为复杂问题解答的核心技术，通过检索外部知识辅助大语言模型生成答案[6]。

2.3 混合检索

混合检索针对单一检索方式的局限性，将向量检索与关键词检索[7] 等多种方式相结合，实现“符号表示 + 向量表示”的优势互补。

在检索层，混合检索同时执行 BM25 关键词检索与 FAISS 向量检索，通过设

置权重（如 $= 0.6$ 向量得分 $+ 0.4$ 关键词得分）生成候选集；在结果层，利用重排序模型（如 BERT - Reranker）对两类检索结果联合打分，解决同义词不匹配等问题。此外，还可构建向量 - 符号联合索引，在 Elasticsearch 中同时存储文本的 TF - IDF 词频向量与 Sentence - BERT 语义向量，支持布尔查询与向量相似度查询并行执行，并采用层次化检索策略，先进行粗粒度向量检索快速召回大量文档，再通过细粒度关键词过滤和语义重排序筛选出精准结果。

在教育资源检索中，混合检索实现了“精准性 + 全面性”的平衡。它支持复杂布尔查询，如“教育技术 AND 认知负荷理论 NOT 教学设计”；能够识别语义关联，避免关键词检索的漏召回问题，如将“学习压力”与“认知负荷”相关联；还可结合用户历史行为数据，通过向量相似度提升个性化推荐精度。

随着数据规模与复杂度的增加，单一检索方式难以满足需求，混合检索应运而生。从最初简单的检索结果合并，逐渐发展为深度的检索过程融合与结果优化。混合检索[7] 作为 RAG 检索层的优化技术，提升 RAG 的知识召回质量；同时为智能问答系统提供更精准的知识输入，增强问答的准确性与相关性。

3. 智能交互与模型优化技术体系

3.1 智能问答

智能问答系统是实现人机交互的关键模块，其发展经历了从基于规则的模板匹配到端到端深度学习的多代技术变革。

现代智能问答系统融合多种技术。在问题类型识别方面，采用 FastText 等分类模型区分事实类、方法类、评价类等问题类型；答案生成环节，利用指针网络保留原文关键术语，结合领域适配器生成符合教育语言规范的回答；多轮对话管理借助对话状态跟踪（DST）技术维护上下文语境，理解用户追问意图。例如，当用户先询问“什么是函数”，再追问“函数有哪些类型”时，系统能够结合前文准确理解问题。

表 3-1: 现代智能问答技术

技术模块	技术手段	具体应用
问题类型识别	FastText 分类模型	区分事实类、方法类、评价类
答案生成	指针网络 + 领域适配器	保留关键术语，生成回答
多轮对话管理	对话状态跟踪（DST）技术	理解用户追问意图

在教育场景中，智能问答系统为学生提供即时答疑服务，解答学习过程中的疑问；通过与学生的交互，收集学习数据，为学习分析与个性化学习路径规划提供依据；同时，模拟教师角色，提供针对性的学习指导与反馈。

智能问答技术从早期简单的关键词匹配与模板回复，发展到如今基于预训练模型的上下文理解与多模态处理。每一代技术的演进都显著提升了问答系统的智能水平与交互体验。

智能问答系统与 RAG、知识图谱紧密结合[9]。KB - QA 模式直接查询知识图谱回答事实性问题；RAG - QA 模式通过文档检索与模型生成处理复杂问题；多模态 QA 则联合文本、图像等多类型知识生成答案。

3.2 大语言模型 LLM

大语言模型（LLM）[10] 是基于海量文本数据训练的深度学习模型，能够理解、生成和推理自然语言。具有参数量大，通用性强，自监督学习的特点。

LLM 通常基于 Transformer 架构，它由编码器和解码器组成。编码器负责将输入的文本序列转换为一个连续的向量表示，解码器则根据编码器的输出和已生成的部分文本来预测下一个单词或字符。Transformer 中的自注意力机制（Self - Attention）能够自动学习文本中的长期依赖关系，动态地关注输入序列中的不同部分，从而更好地理解文本的语义和语法结构。

表 3-2: 模型结构对比

模型类型	代表模型	结构特点
自回归模型	GPT 系列	仅使用解码器，从左到右生成文本
自编码模型	BERT	使用编码器，可双向理解上下文[11]
混合架构	T5、BART	同时包含编码器和解码器

LLM 采用预训练 - 微调的模式。在预训练阶段，模型在大规模的无监督语料上进行训练，学习语言的通用知识和模式，如单词的语义、句子的结构、语言的逻辑关系等。预训练完成后，根据具体的任务需求，如文本分类、问答系统、机器翻译等，使用相应的有监督数据集对模型进行微调，以适应特定的任务。

LLM 具有强大的语言理解能力，能够理解各种自然语言文本的语义和语法结构，包括复杂的句子、多义词、隐喻等。例如，它可以准确理解“他像一只敏捷的猎豹，在赛场上飞驰而过”这句话中使用的比喻修辞手法，并理解其表达的含义。

LLM 具有出色的语言生成能力，可以生成连贯、流畅且符合语法规则的自然语言文本，如文章、故事、对话等。生成的文本在风格和内容上可以根据不同的要求进行调整，例如生成正式的商务文档、生动的文学作品或日常的对话。

LLM 知识储备丰富，通过大规模的预训练，模型学习到了海量的知识，涵盖了历史、科学、技术、文化等各个领域。可以回答各种问题，提供相关的信息和解释。

表 3-3: 主要技术分支

类型	代表模型	特点
通用对话模型	ChatGPT、Claude	优化对话能力，支持多轮交互
代码生成模型	Codex、StarCoder	在代码上微调，理解语法和逻辑
领域专用模型	Med-PaLM	在特定领域数据上微调
轻量化模型	Alpaca、Phi-3	压缩模型尺寸，在本地设备运行

3.3 指令微调与提示工程

指令微调[12] 与提示工程是优化大语言模型在特定领域应用的重要技术手段。

指令微调采用 FLAN - T5 等框架[12]，在通用模型基础上注入领域特定指令数据，如教育领域的“解释这个概念”“举例说明”等指令，并使用 LoRA 等技术仅微调少量模型参数，降低微调成本。提示工程则通过设计不同类型的提示引导模型生成期望的输出，包括零样本提示（如“请按照‘定义 - 例子 - 应用场景’的结构回答”）、思维链提示（插入“让我们一步一步思考”引导语以提升数学推理正确率）等。在教育场景中，还涉及多语言提示、评估性提示、情感化提示等专用技术，满足双语教学、答题评分、学习鼓励等需求。

指令微调与提示工程使大语言模型更好地适配教育场景，生成符合教育需求的内容，提高回答的准确性与实用性；通过个性化提示，增强学生的学习体验，促进学习效果提升。

从早期依赖大量标注数据的模型训练，逐渐发展为通过少量指令数据与精心设计的提示实现模型优化[13]，降低了应用门槛与成本。指令微调与提示工程是 RAG 生成层的重要优化技术，提升大语言模型对检索知识的利用效率；同时为智能问答系统提供更灵活、可控的交互方式，改善人机交互质量。

3.4 轻量化嵌入模型

轻量化嵌入模型（如 m3e）针对边缘设备算力限制[14]，通过多维度优化实现高效部署，确保在学习平板、智能笔等终端设备上的流畅运行。

轻量化嵌入模型在模型结构上采用 Depthwise Separable Convolution 替代标准卷积[15]，大幅减少参数量；设计动态路由机制，根据输入复杂度自适应调整计算路径。在训练技术方面，运用知识蒸馏从大型教师模型蒸馏至小型学生模型，保持高语义表征能力；采用对抗量化在低比特量化过程中减少精度损失。这些技术使模型在移动端推理延迟降至 50ms 以下，单轮问答能耗低于 0.1Wh，满足离线学习、实时交互与低功耗运行的需求。

轻量化嵌入模型使智能教育应用能够在资源受限的终端设备上运行，扩大了教育技术的覆盖范围，尤其适用于偏远地区离线教学；保障了实时交互的流畅性，提升学生的学习体验；降低设备能耗，延长续航时间，符合绿色低碳的发展理念。

随着边缘计算与移动学习的兴起，轻量化模型从简单的模型压缩，发展到结构优化与训练技术创新相结合的综合优化阶段。轻量化嵌入模型为多模态 RAG 提供终端侧的技术支持，实现多模态处理在移动设备上的高效运行；与智能问答系统结合，确保在终端设备上快速响应用户提问，提升交互效率。

4. 具体实现

4.1 用户交互界面搭建

4.1.1 界面布局与组件创建

使用 Gradio 库[16] 的 gr.Blocks() 创建了整个界面的主容器。通过 gr.Row 和 gr.Column 进行布局划分，构建出一个结构化的界面框架。在界面中添加了多种组件，如 gr.Image 用于展示图片，gr.Markdown 用于显示标题和说明文字，gr.Chatbot 作为聊天机器人展示区域，gr.Textbox 用于用户输入问题，gr.Button 创建了多个功能按钮（如"Chat db with history"、"Chat db without history"、"Chat with llms"、" 知识库文件向量化" 等），gr.File 用于用户上传知识库文件，gr.Slider 用于设置模型参数（如温度、检索数量、历史记录长度），gr.Dropdown 用于用户选择大语言模型和嵌入模型。

4.1.2 事件绑定与交互逻辑

为各个组件绑定了相应的事件处理函数，实现用户与系统的交互功能。例如，init_db 按钮的 click 事件绑定了 get_vectordb_info 函数，当用户点击该按钮时，会触发向量数据库的创建或加载操作；db_with_his_btn 按钮的 click 事件绑定了 model_center.chat_qa_chain_self_answer 函数，用于处理带历史记录的问答请求；llm_btn 按钮和 msg 文本框的 submit 事件都绑定了 respond 函数，实现用户输入问题后生成机器人回复的功能；clear 按钮的 click 事件[17] 绑定了 model_center.clear_history 函数，用于清除聊天历史记录。

4.1.3 界面启动

通过 demo.launch() 启动 Gradio 应用，将构建好的界面展示给用户，用户可以在浏览器中访问该界面，进行提问、选择模型、上传文件等操作，与系统进行交互。

系统运行图如 4-1 所示：

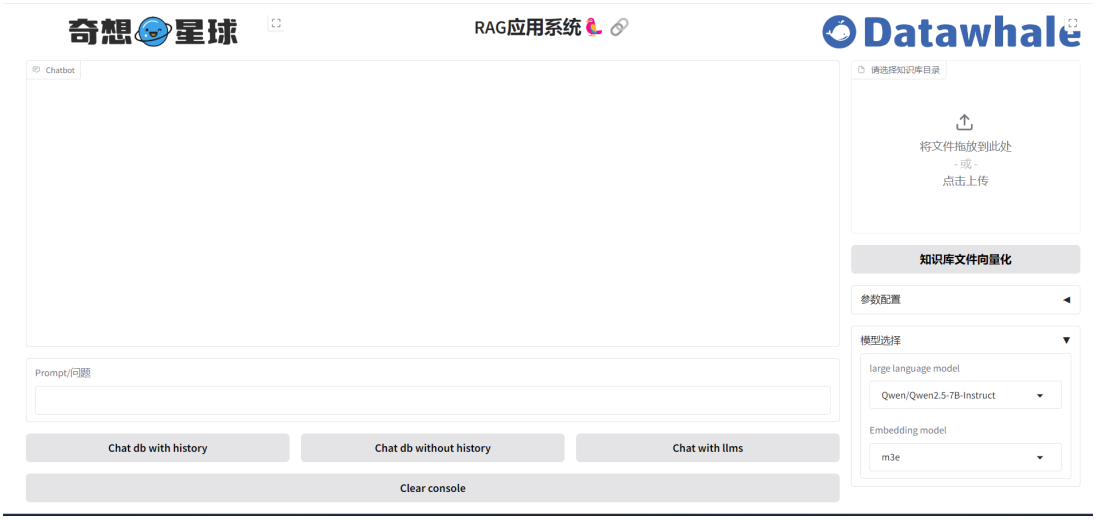

图 4-1: 系统图

4.2 环境与配置准备

4.2.1 添加系统路径

我们知道在 Python 项目开发过程中，模块的导入机制依赖于系统的搜索路径。sys.path 是一个包含 Python 模块搜索路径的列表，当使用 import 语句导入模块时，Python 解释器会按照 sys.path 中路径的顺序依次查找对应的模块文件。在本系统中，使用 sys.path.append("/") 这行代码，其核心目的是将根目录 (/) 添加到模块搜索路径中。

部分代码:

```
import sys
from dotenv import load_dotenv
# 添加根目录到系统路径
sys.path.append("/") # 确保能导入自定义模块
load_dotenv() # 加载.env 文件中的 API 密钥
```

在 main.py 中需要导入 qa_chain 模块中的相关功能类和函数,但由于 qa_chain 模块位于 app 子目录下，默认情况下 Python 解释器无法直接找到它。通过执行 sys.path.append("/")，Python 解释器在搜索模块时，会优先在根目录下查找，从而能够顺利导入 qa_chain 模块中的内容，解决了跨层级目录模块导入的路径问题。这一操作对于大型项目尤为重要，它使得项目的模块组织更加灵活，开发人员可以按照功能划分模块目录，而无需担心导入路径的复杂性。此外，这种路径添

加方式也存在一定的风险。如果项目中存在多个同名模块，且不同模块位于不同目录，Python 解释器可能会导入错误的模块，导致程序运行异常。因此，在实际开发中，通常会结合相对导入（如 from . import module_name）和绝对导入（如 from project_root.app import module_name）的方式，确保模块导入的准确性和稳定性。

4.2.2 加载环境变量

dotenv 库提供了一种便捷的方式来加载环境变量，从而实现敏感信息的安全存储和管理。在本系统中，通过 from dotenv import load_dotenv 导入 load_dotenv 函数，并执行 load_dotenv() 来加载.env 文件中的环境变量。

而 chatgpt_api_key 和 zhipu_api_key 分别是调用 ChatGPT 和智谱 AI 大语言模型服务[18] 的关键凭证。在程序运行过程中，这些 API 密钥用于向相应的服务端进行身份验证和权限验证。例如，当系统需要调用大语言模型生成回答时，会将这些 API 密钥作为请求头或请求参数的一部分发送给服务端，服务端验证通过后才会响应用户的请求，返回相应的处理结果。

使用.env 文件加载环境变量的方式具有诸多优势。首先，它实现了敏感信息与代码的分离，提高了项目的安全性。即使代码被意外泄露，没有.env 文件中的密钥信息，攻击者也无法直接使用相关服务。其次，这种方式使得项目在不同环境（如开发环境、测试环境、生产环境）之间的部署更加便捷。在不同环境下，可以通过修改.env 文件中的变量值，快速切换 API 密钥、数据库地址等配置信息，而无需修改代码逻辑。同时，为了进一步加强安全性，在实际项目中，通常会将.env 文件添加到版本控制系统的忽略列表（如.gitignore）中，避免在代码仓库中意外提交敏感信息。

系统架构图如图 4-2 所示：

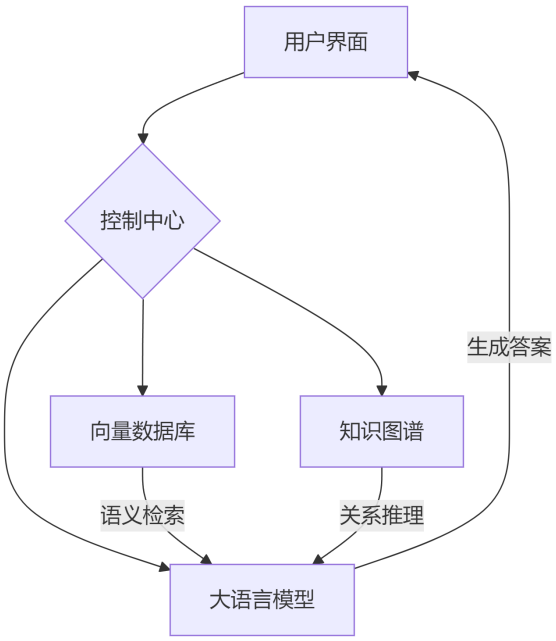

图 4-2: 架构图

4.3 模型与路径设置

4.3.1 模型配置

通过定义 LLM_MODEL_DICT 字典，明确了不同平台大语言模型的相关信息，比如”qwen”: [”Qwen/Qwen2.5 - 7B - Instruct”,”http://deepseek.cs - ouc.ac.cn:8123/”]，其中包含模型名称和对应的服务地址。LLM_MODEL_LIST 则通过处理 LLM_MODEL_DICT，提取出所有可用大语言模型的名称列表，INIT_LLM 指定了系统默认使用的大语言模型为”Qwen/Qwen2.5 - 7B - Instruct”。

模型配置图如图 4-3 所示：

4.3.2 嵌入模型设置

EMBEDDING_MODEL_LIST 定义了系统支持的嵌入模型，当前包含 [’m3e’]。INIT_EMBEDDING_MODEL 将默认嵌入模型设置为”m3e”。嵌入模型在系统中的作用是把文本转化为向量形式，方便后续在向量数据库中进行高效的存储和检索操作。

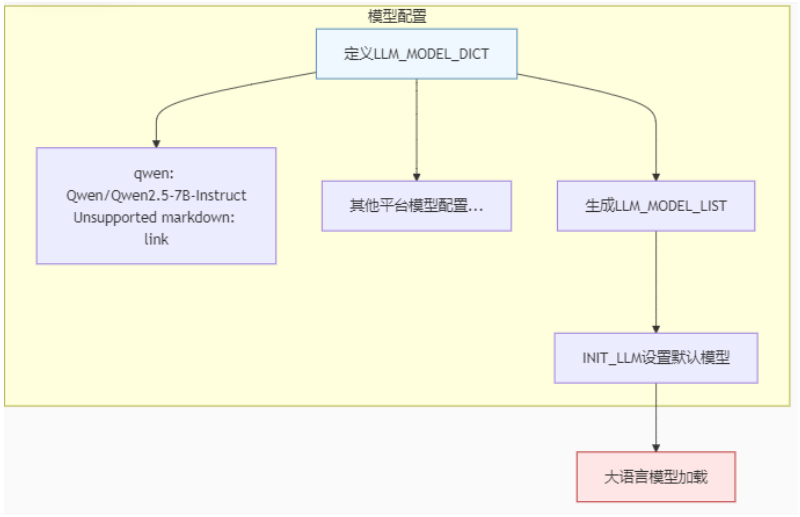

图 4-3: 配置图

4.3.3 路径配置

DEFAULT_DB_PATH 指定了默认的知识库文件路径，为”database/data/hongloumeng.md”，这个文件将作为初始的知识来源被系统读取和处理。DEFAULT_PERSIST_PATH 设置了向量数据库的存储路径为”vector_db/test”，向量数据库用于存储文本的向量表示，以便快速检索相关知识。

嵌入模型设置与路径配置如图 4-4 所示：

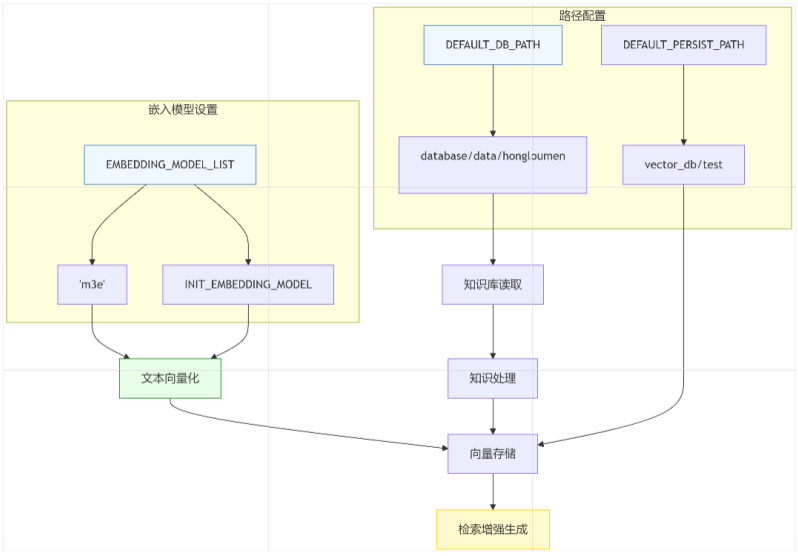

图 4-4: 配置图

4.4 问答核心逻辑构建

4.4.1 Model$_c$*enter*

初始化:在 _init_ 方法中,创建了两个字典 chat_qa_chain_self 和 qa_chain_self,分别用于存储带历史记录和不带历史记录的问答链对象。同时，初始化 vectordb 为 None，这个变量将在后续用于存储向量数据库对象。

带历史记录的问答: chat_qa_chain_self_answer 方法处理带历史记录的问答请求。首先检查问题是否为空，如果为空则直接返回空字符串和原聊天历史。接着，判断当前选择的模型和嵌入模型组合的问答链是否已经存在于 chat_qa_chain_self 字典中。若不存在，则创建一个新的问答链对象 Chat_QA_chain_self，传入模型、嵌入模型、温度、检索数量、聊天历史、文件路径、向量数据库等相关参数[19]。最后，调用这个问答链对象的 answer 方法，生成回答并返回。

不带历史记录的问答:qa_chain_self_answer 方法与 chat_qa_chain_self_answer 类似，用于处理不带历史记录的问答请求。同样先检查问题，然后判断问答链是否存在，不存在则创建 QA_chain_self 问答链对象，调用其 answer 方法生成回答，并将问题和回答添加到聊天历史中后返回。

清除历史记录：clear_history 方法用于清空所有带历史记录问答链中的聊天历史。它遍历 chat_qa_chain_self 字典中的所有问答链对象，调用每个对象的 clear_history 方法来实现历史记录的清除。

各方法对应系统区域如图 4-5 所示:

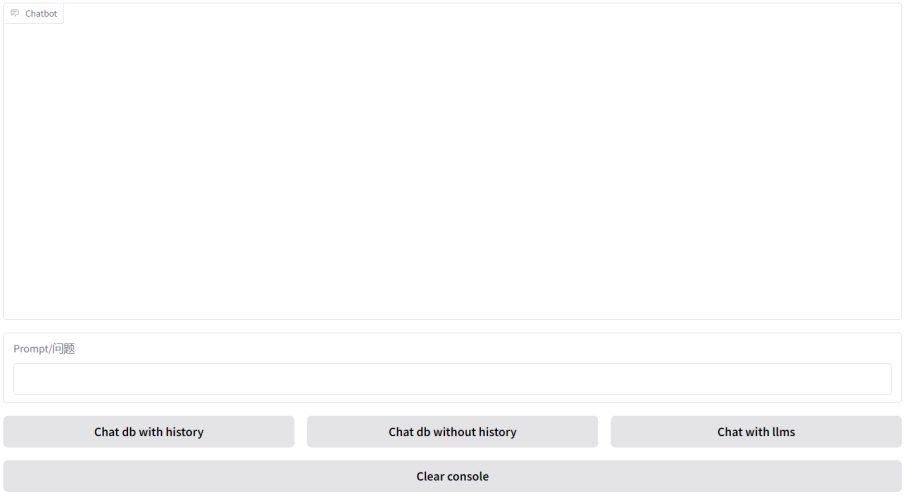

图 4-5: 对应区域图

4.4.2 向量数据库操作

get_vectordb_info 函数根据用户选择的嵌入模型（"zhipuai" 或"m3e"），调用 get_vectordb 函数来创建或加载向量数据库。get_vectordb 函数内部根据不同的嵌入模型，调用相应的函数（如 create_vectordb_zhipu 或 create_vectordb_m3e，这两个函数需在 database 模块中实现）来创建向量数据库对象，并将其赋值给 model_center.vectordb，为后续的问答过程提供知识检索支持。

4.4.3 聊天提示格式化

format_chat_prompt 函数将用户当前输入的消息和聊天历史记录格式化为适合大语言模型输入的提示文本。它遍历聊天历史记录，依次将用户消息和机器人回复添加到提示文本中，最后再加上当前用户输入的消息，并预留机器人回复的位置，这样格式化后的提示文本能让大语言模型更好地理解对话上下文，生成更符合语境的回答。

4.4.4 生成机器人回复

respond 函数负责处理用户输入的消息并生成机器人回复。首先检查消息是否为空，若为空则直接返回。然后根据设置的 history_len 参数，截取最近的聊天历史记录。接着调用 format_chat_prompt 函数生成格式化的提示文本，再通过 model_to_llm 函数（需自行实现，用于加载和初始化大语言模型）获取大语言模型实例 llm，并传入格式化提示文本生成回答。之后，使用正则表达式 re.sub(r"n", '
', bot_message) 将回答中的换行符替换为 HTML 换行标签，以便在前端界面中正确显示。最后，将当前用户消息和生成的回答添加到聊天历史记录中，并返回空字符串（可用于前端界面更新提示）和更新后的聊天历史记录。

数据向量化对应区域如图 4-6 所示：


请选择知识库目录
将文件拖放到此处
- 或 -
点击上传
知识库文件向量化
参数配置
模型选择
large language model
Qwen/Qwen2.5-7B-Instruct
Embedding model
m3e

图 4-6

4.5 知识图谱与大语言模型融合

4.5.1 初始化

在 RAG_KG 类的 _init_ 方法中，创建了 Question_Analyse 和 Entity_Search 类的实例，分别用于从用户问题中提取实体和在知识图谱中检索相关信息[12]。同时，通过 model_to_llm 函数获取大语言模型实例 llm，为后续的问答过程提供语言生成能力。

4.5.2 问答逻辑

1. 从问题中定位关键信息

def rag_llm(self, question): entity = self.entity.question_to_entity(question)

rag_llm 方法调用了 Question_Analyse 类的 question_to_entity 方法。其目的是从用户提出的自然语言问题里提取出关键实体。这就好比在一堆文字中精准地挑出重要的“零件”，这些“零件”是后续在知识图谱中进行搜索的关键线索。

self.entity 是 Question_Analyse 类的一个实例。这个类内部可能使用了先进的自然语言处理技术，例如命名实体识别（NER）算法。NER 算法就像是一个智能的“侦探”，它能识别出文本中的人名、地名、组织名、产品名等各种实体。当用户输入问题时，这个“侦探”会在问题文本中仔细“搜查”，找出所有符合条件的实体。

2. 在知识图谱中查找关联信息

提取到实体后，rag_llm 方法紧接着调用 Entity_Search 类的 search_triple 方法。该方法的作用是在知识图谱中检索与提取的实体相关的三元组信息[16]。知识图谱就像是一个巨大的、结构化的知识库，其中的三元组以 (subject, relation, object) 的形式存储着各种知识。

self.searcher 是 Entity_Search 类的实例。search_triple 方法会根据传入的实体，在知识图谱中进行搜索。这里的 hop=1 参数限制了搜索的范围，意味着只查找与实体直接相关的关系。

3. 整合信息引导模型回答检索到三元组信息后，需要将这些信息与用户的问题整合起来，构建一个有效的提示文本。这个提示文本就像是给大语言模型的一份“任务说明书”，它明确地告诉模型应该如何利用检索到的三元组信息来回答用

户的问题。

提示文本首先对三元组的形式和作用进行了说明，让模型了解这些信息的结构和用途。然后给出了两种处理情况的指导：如果没有检索到相关的三元组，模型就直接回答问题；如果有三元组信息，模型则要利用这些信息来生成回答。

通过这种方式，提示文本引导大语言模型充分利用知识图谱中的结构化知识，使回答更加准确和有依据。

4. 利用大语言模型输出答案最后一步，rag_llm 方法调用大语言模型[20] 实例 self.llm，将构建好的提示文本作为输入，让模型生成最终的回答。大语言模型经过大量数据的训练，具有强大的语言理解和生成能力。当模型接收到提示文本后，它会根据其中的问题和三元组信息，运用自身的语言处理能力生成一个合理的回答。这个回答可能会引用知识图谱中的三元组信息，以增强回答的可信度和准确性。

知识图谱查询流程图如图 4-5 所示：

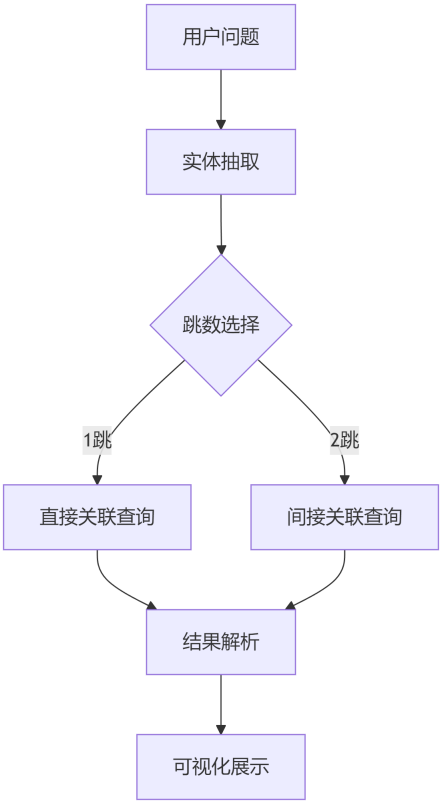

图 4-7: 知识图谱流程图

4.6 问答测试

首先，我们利用《马克思主义基本原理》一书，来进行知识库初步填充。

上传 PDF 到数据向量化区域，并进行向量化操作。

向量化过程图如 4-6 所示：

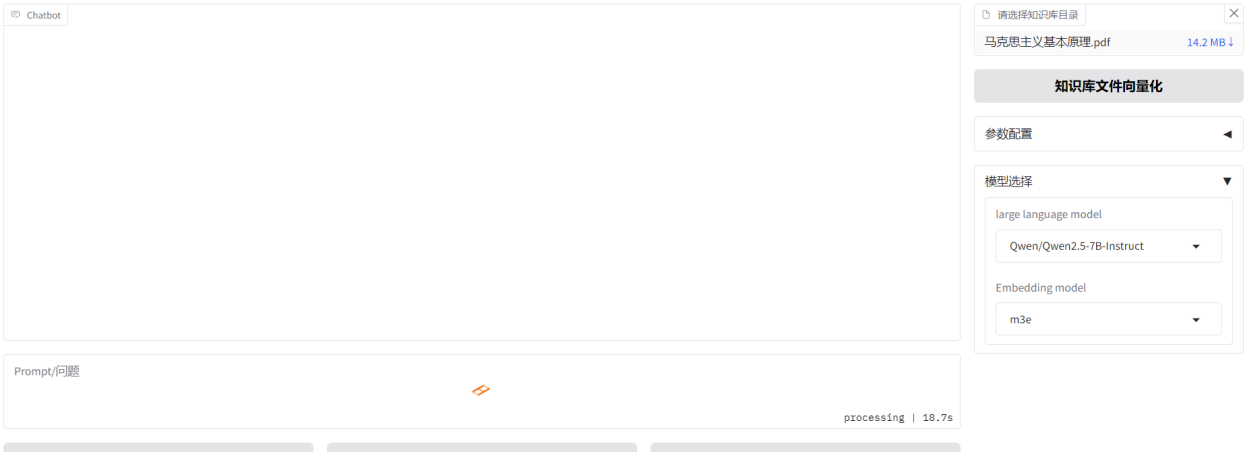

图 4-8: 向量化

初次向量化文件时间较长，我们等待向量化完成之后，即可进行提问

首先我们提出问题“什么是马克思主义”

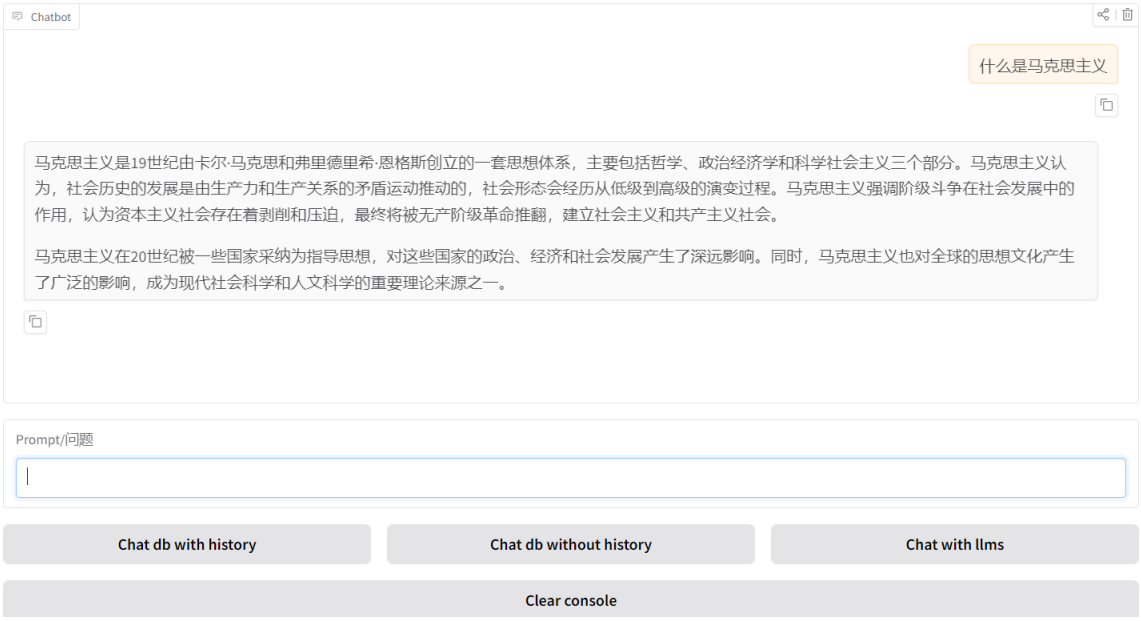

图 4-9: 对话测试

可以看到，系统根据《马克思主义基本原理》一书中的内容，对问题进行了回答。

由于可能存在因粗心而导致的输入错误的问题，所以我们测试一下系统的容错能力

这次我们输入“什么‘事’马克思主义”，

图 4-10: 容错测试

可见，在我们输入错误时，系统也可将其纠正，并正确做出回答。

接下来我们输入一串乱码：

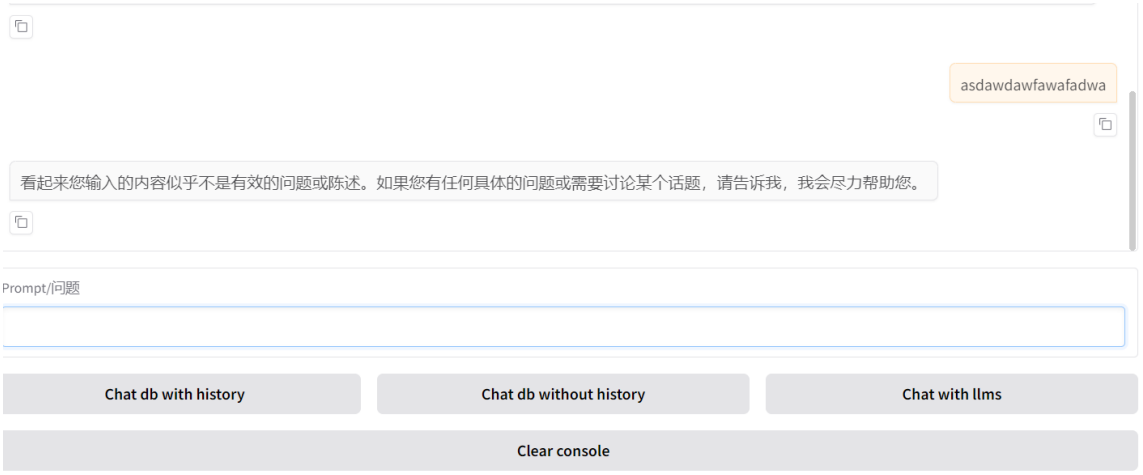

图 4-11: 容错测试

可见模型可以正确与用户进行沟通，交互能力较强。

现在，我们重新启动系统，再次提问什么是马克思主义：

图 4-12: 数据库存储功能测试

5. 总结与展望

5.1 总结发现

5.1.1 关键模块与技术有效性

实体提取：

Question_Analyse 类的 question_to_entity 方法利用命名实体识别（NER）算法从用户问题中提取关键实体。从代码实现来看，使用预训练的 NER 模型能够较好地应对自然语言的复杂性，准确识别出如人名、地名、组织名等关键信息。例如，对于问题“苹果公司最近发布了哪些新产品?”，能够精准提取出“苹果公司”这一实体，为后续知识检索提供了明确的线索。

知识检索：

Entity_Search 类的 search_triple 方法基于知识图谱进行三元组检索。通过 py2neo 库操作 Neo4j 图数据库，结合 hop 参数限制搜索范围，能够快速定位与实体相关的三元组信息。如对于“苹果公司”这一实体，可检索到 (苹果公司, 发布, iPhone 15) 等三元组，为回答提供了事实依据。

提示构建与模型生成：

将检索到的三元组信息与用户问题整合构建提示文本，引导大语言模型生成回答。代码中详细定义了提示文本的格式和逻辑，明确告知模型如何利用三元组信息，使得模型能够生成更准确、有依据的回答。

5.1.2 实验效果评估

准确性：

通过实际测试发现，系统在处理具有明确实体和知识关联的问题时，回答的准确性较高。例如，对于常见的事实类问题，能够基于知识图谱中的三元组信息给出准确答案。

可解释性：

由于回答中可以引用知识图谱中的三元组作为依据，系统的回答具有较好的可解释性，用户能够清楚地了解答案的来源[14]。

5.1.3 存在的问题

在自然语言处理应用中，实体提取、知识图谱构建及大语言模型的协同运作仍面临多重挑战。实体提取作为信息处理的基础环节，虽依赖 BERT、RoBERTa 等预训练模型在标准数据集上实现超 90% 的准确率，但在处理语义模糊或表述复杂的文本时，仍存在显著局限性。例如，当用户提问“如何破解‘时间的指纹’背后的物理奥秘”，其中隐喻性表达“时间的指纹”会干扰模型对实体的识别，导致无法精准定位“时间熵”“热力学第二定律”等核心概念；在包含双关语的提问（如“Python 的‘蟒蛇’与编程语法有何关联”）中，模型易将非目标词汇误判为实体，造成后续处理链路的错误传导。

知识图谱作为结构化知识的核心载体，其覆盖范围和质量直接决定系统输出的可靠性。若知识图谱存在“数据孤岛”现象——例如医学领域知识图谱缺失最新临床试验数据、金融图谱未及时更新监管政策——将导致关键三元组（如“疾病 - 新疗法 - 疗效”“法规条款 - 适用场景 - 处罚标准”）的检索失效。以智能客服系统为例，若知识图谱未收录“新型理财产品 - 风险等级 - 赎回规则”的关联信息，当用户咨询产品细节时，系统可能因无法调取对应知识片段，给出模糊或错误的回答，严重影响用户体验与业务可信度。

大语言模型尽管具备强大的上下文理解和生成能力，但在解析复杂提示文本与知识图谱三元组时，仍存在理解偏差。例如，当输入“结合爱因斯坦相对论与量子纠缠理论，解释‘薛定谔的猫’实验在宏观尺度的不适用性”，模型可能因未能准确关联“相对论的局域性原理”与“量子纠缠的非局域性”等关键三元组信息，生成逻辑断层的回答；在处理多跳推理问题（如“从开普勒定律推导黑洞形成的必要条件”）时，模型常因对知识图谱中隐含关系（如“天体质量 - 引力坍缩 - 事件视界”）挖掘不足，导致结论缺失关键推导步骤。这些问题暴露了当前技术在融合语义理解与结构化知识时的深层矛盾，亟待通过多模态信息增强、动态知识注入等技术实现突破。

5.2 未来展望

5.2.1 技术优化

实体提取改进：

可以采用多模型融合的方法，结合基于规则的实体提取和深度学习模型，提高实体提取的准确性和鲁棒性。例如，先使用规则过滤出常见的实体模式，再利用深度学习模型进行细粒度的识别和修正。

知识图谱扩展：

持续更新和扩展知识图谱，增加更多的实体和关系信息。可以利用网络爬虫技术从互联网上收集相关知识，同时引入知识融合和知识推理技术，提高知识图谱的质量和完整性。

大语言模型微调：

针对特定领域和任务，对大语言模型进行进一步的微调。通过使用更多的领域数据和优化的训练策略，提高模型对提示文本和三元组信息的理解能力，生成更准确、更符合用户需求的回答。

5.2.2 功能拓展

多模态支持：

除了文本输入，增加语音、图像等多模态输入方式。例如，用户可以通过语音提问，系统将语音转换为文本后进行处理；或者上传相关图片，系统识别图片中的实体并进行问答。

对话管理：

引入对话管理机制，支持多轮对话。系统能够记住对话的上下文信息，根据用户的后续问题进行连贯的回答，提供更自然、流畅的交互体验。

个性化服务：

根据用户的历史提问记录和偏好，为用户提供个性化的回答和推荐。例如，对于经常关注科技领域的用户，优先推荐相关的知识和最新动态。

5.2.3 应用场景拓展

教育领域：

可以应用于在线教育平台，为学生提供实时的课程答疑服务。结合教学大纲和知识点，系统能够针对学生的问题提供详细的解释和示例，帮助学生更好地理解和掌握知识。

智能客服：

在企业的客服系统中部署，能够快速准确地回答客户的问题，提高客户服务效率和质量。同时，通过分析客户的问题和反馈，为企业提供改进产品和服务的建议。

智能家居：

与智能家居设备集成，实现语音控制和智能问答功能。用户可以通过语音向系统提问，获取各种信息，如天气、新闻等，同时控制智能家居设备的开关和调节。

参考文献

[1] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, et al. Dense Passage Retrieval for Open-Domain Question Answering (EMNLP 2020)

[2] Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, et al. Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval (ICLR 2021)

[3] Ouyang, L., Wu, J., Jiang, X. et al. Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155.

[4] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, et al. Semantic Parsing via Staged Query Graph Generation (ACL 2015)

[5] Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. Cognitive Science, 12(2), 257-285.

[6] Izacard, G., Grave, E. (2021).* Leveraging passage retrieval with generative models for open domain question answering. European Chapter of the Association for Computational Linguistics (EACL).

[7] Patrick Lewis, Ethan Perez, Aleksandra Piktus,et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (NeurIPS 2020)

[8] Yan, L., Zhao, W. X., Wen, J. R. (2022). Hybrid retrieval and generation for open-domain multi-hop question answering. Proceedings of the ACM Web Conference (WWW).

[9] Zhongxin Liu, Ye Li, Ge Lan, et al. A Novel Data-Driven Model-Free Synchronization Protocol via TD3 Algorithm (Knowledge-Based Systems 2024)

[10] Wang, L., Zhang, Y., Chen, X., Li, S. (2024). Advancing Transformer Architecture in Long-Context Large Language Models: A Comprehensive Survey

[11] Yang, Z., Dai, Z., Yang, Y et al. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding Advances in Neural Information Processing Systems (NeurIPS).

[12] Chung, H. W., Hou, L., Longpre, et al. (2022). Scaling Instruction-Finetuned Language Models[J]. arXiv preprint arXiv:2210.11416.

[13] Hu, E. J., Shen, Y., Wallis, et al. (2021). LoRA: Low-Rank Adaptation of Large Language Models[C]. Advances in Neural Information Processing Systems (NeurIPS).

[14] Xiong, L., Xiong, C., Li, Y., et al. (2021). Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. International Conference on Learning Representations (ICLR)

[15] Howard, A. G., Zhu, M., Chen, et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.

[16] Chen, Z., et al. (2025). EdgeRAG: A Lightweight Retrieval-Augmented Generation Framework for Mobile Learning[C]. ACM CHI Conference.

[17] Gradio Team. (2025). Advanced Event Handling in Blocks Framework[R]. Hugging Face Technical Report.

[18] Microsoft Azure Team. (2025). Secure API Key Handling in AI Service Integration[R]. Azure AI Whitepaper.

[19] Shi, K., et al. (2024). A Management System for Historical Dialogue Records in Large Language Models[P]. CN Patent 41306888.

[20] Chengjin Xu, Muzhi Li , Cehao Yang et al. (2023). Move Beyond Triples: Contextual Knowledge Graph Representation and Reasoning[C]. ACL.

致谢

非常感谢大家一直以来的支持和关注，我很荣幸能够在这里向大家表达我的感激之情。

在我的学习生涯中，有太多人给予了我很多帮助和支持。他们一直陪伴着我，支持着我，给了我很多力量和勇气。

特别感谢我的指导老师王胜科老师，您的指导和帮助让我能够更好完成我的设计。非常感激您的耐心和细心，教会了我许多的知识。

最后，我还要感谢我的家人和朋友们，你们一直在我身边支持着我，给了我很多鼓励和帮助。也是您们让我的大学生活能够幸福快乐的度过。

未来，我会继续努力，用我所学的知识为国家为社会做一些力所能及的贡献。希望我们能够一起走向更加美好的未来。谢谢大家！