

基于课程教学数据的内容推荐和个性化智能答疑系统的开发任务书

1. 背景介绍

近年来，人工智能技术（如大语言模型、机器学习、知识图谱等）的成熟为教育场景的智能化转型提供了技术基础。通过 AI 技术，可以构建个性化的学习路径推荐系统和智能问答系统，帮助学生快速定位学习重点，解决学习中的疑难问题，同时减轻教师的教学负担。

相关研究显示，985 高校中 20% 的教师认为“教学提升手段有限”，这一比例在顶尖高校群体中尤为显著，反映出即便在资源相对丰富的院校，传统教学模式仍难以满足教学创新需求[1]。教师常受限于标准化课程设计，难以针对学生差异化需求动态调整教学内容，而海量课程数据（如学习行为记录、作业反馈、互动日志）长期处于低效利用状态。在此背景下，基于课程教学数据的内容推荐和个性化智能答疑系统成为破局关键：一方面，通过机器学习算法挖掘学生知识掌握规律，可构建个性化学习资源推荐模型，例如根据学生答题薄弱点自动关联相关微课视频或文献资料；另一方面，智能答疑系统借助大语言模型，能够实时解析学生提问语义，结合教学材料何知识图谱精准匹配解答方案，缓解教师重复答疑压力。此类技术不仅将教学数据转化为可操作的决策依据，更能通过“数据驱动-精准干预”模式，系统性提升教学手段的智能化水平，为高校破解教学提升瓶颈提供可扩展路径。

本项目旨在利用人工智能技术，结合课程教学数据，开发一套智能化工具，为在线教育平台提供实时内容推荐和智能问答服务，推动教育资源的精准化分配和教学效率的提升。

2. 欲解决问题

2.1 即时答疑需求问题：学生在学习过程中遇到问题时，往往无法及时获得解答，影响学习效率。

目标：通过大语言模型驱动的智能问答系统，提供 7×24 小时的即时答疑服务，支持多轮追问上下文理解及推理功能。

2.2 学习资源过载问题：在线教育平台通常包含大量课程资源，学生难以快速找到适合自己的学习内容。

目标：通过实时内容总结和推荐，帮助学生提纲挈领，并高效获取与当前学习进度相关的知识点。

2.3 个性化学习支持不足问题：传统在线教育平台缺乏对学生学习行为的深度分析，无法提供个性化的学习建议。

目标：基于学生的学习行为数据，构建个性化推荐模型，为学生定制学习路径。针对学生的作业情况，分析并诊断薄弱点。

2.4 知识碎片化问题：课程教学数据分散在不同模块（如课件、题库、作业等），缺乏结构化整合。

目标：构建课程知识图谱，将分散的知识点关联起来，支持推理和动态更新。

3. 推荐方案

本系统采用前后端分离的架构设计，面向学生、教师等多角色使用场景。整体系统由后端的核心数据处理与智能服务和前端的用户交互界面组成，设计方案具体如下：

(1) 系统架构设计

数据层负责采集、抓取和清洗课堂实录、课件、题库、作业、测验等多种教学数据。系统支持对课堂直播流进行录播或导入现有录像，对于非文字数据，通过 OCR 和语音转文字技术转换成文本，并经过去噪和标注实现结构化存储，兼容多种数据格式。

算法层包括三个主要模块：

内容推荐模块：基于学生的学习轨迹、作业提交和测验结果等行为数据，构建实时个性化推荐模型，为学生生成学习建议。

智能问答模块：依托本地或在线大语言模型（LLM）和增强检索生成（RAG）技术，实现 24 小时即时问答、多轮交互和上下文追问，结合课程知识图谱提高问答精准性。选用支持推理的大语言模型，如 Deepseek R1、GPT O1 等，增强回复的逻辑性和准确性。

知识图谱构建模块：整合课堂实录、课件、题库和作业等数据，通过关键词提取构建动态更新的知识点关联网络，支持知识推理和系统化学习。

服务层通过标准化 API 接口向前端、移动端客户端和其他教育平台（如在线学习系统、教务管理系统）提供数据服务。

前端层采用现代开发框架（React 或 Vue），设计响应式用户界面，适配 PC 和移动端，支持学生回看课堂、教师上传课件及课程管理，以及管理员进行权限配置，同时集成语音输入与文本输出等多模态交互功能。

(2) 关键技术实现

数据采集与清洗：制定详细的数据采集规则，从多源教学数据中提取有效信息，并利用 OCR 和语音转文字技术处理非文字数据，确保数据质量和结构化存储。

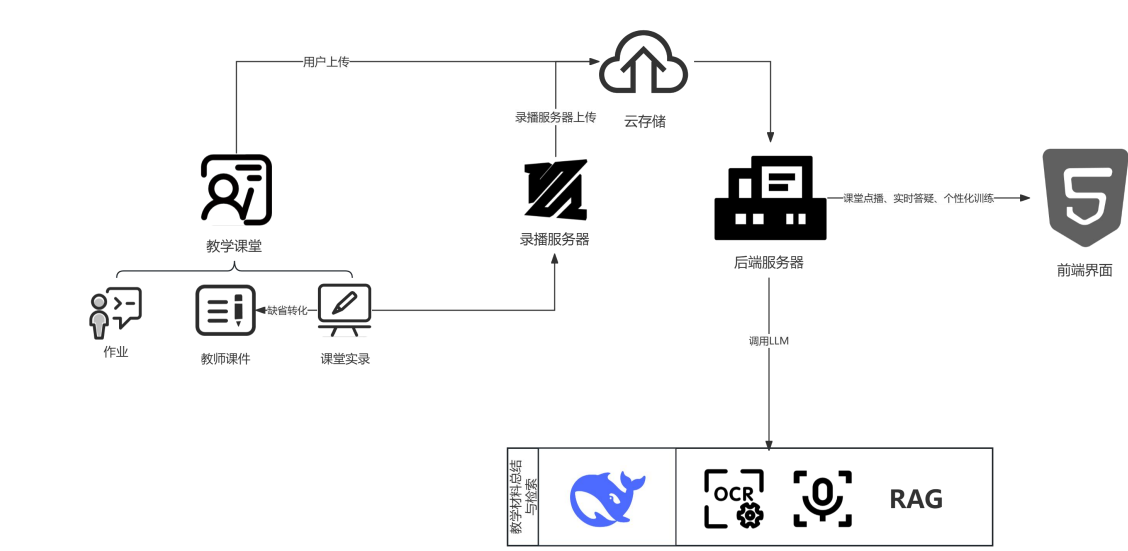
推荐系统设计：基于用户行为数据构建实时推荐模型，生成个性化学习路径和知识点推荐。

问答系统设计：构建包含丰富问答语料和知识库的系统，采用 LLM 与 RAG 技术实现语义理解、问题匹配和多轮追问。

前后端协同：通过 RESTful API 实现前后端数据交互，确保前端界面的实时响应和多模态交互支持。

- (3) 创新点
- 动态知识图谱：实时整合多源教学数据，构建知识点关联网络，实现知识推理与动态更新。
- 多模态交互支持：集成语音识别、OCR 和大语言模型技术，实现智能问答和多轮追问，提升用户交互体验。
- 轻量化设计：基于开源工具进行二次开发，降低部署和维护成本，同时确保系统的高扩展性。
- 前后端分离架构：保障系统的可维护性与灵活性，满足多角色定制化界面和跨平台交互需求。

4. 应用场景



(1) 课堂教学支持场景

- 学生端：
- 学生通过前端界面搜索教学视频，系统基于课程知识图谱和 RAG 技术推荐相关视频片段及对应时间点，支持快速定位知识点。
- 观看视频时，可结合内容大纲跳转至关键教学片段，通过 AI 答疑框提问，系统调用 LLM 与历史学习数据（作业、测验）提供精准解答。
- 利用智能总结功能，课后复习缓存的课程主旨，并通过知识图谱查看知识点关联网络，辅助系统化学习。
- 教师端：
- 教师通过管理后台导入课堂实录、课件、题库等资源，系统自动抓取并清洗数据（OCR、语音转文字），构建动态更新的知识图谱。

直播课程自动录播后存入云存储, 结合数据中心机柜进行资源管理与分发, 供学生按需调用。

(2) 自主学习与个性化推荐场景

系统分析用户学习路径(作业提交、测验结果、视频观看记录), 通过智能推荐算法推送针对性教学片段, 提升学习效率。

(3) 协作与互动场景

视频评论区提供师生互动平台, 学生可留言讨论, 教师根据反馈调整教学内容。

教师通过管理后台设置权限, 管理课程访问范围与学生行为数据, 确保教学资源的安全性与可控性。

5. 软件系统的功能描述

本系统的后端主要承担数据存储、智能推荐、问答处理及交互支持等核心功能。具体功能描述如下:

课堂数据抓取和清洗, 能导入课堂实录、课件、作业、测验等数据用于答疑; 能对课堂直播流进行录播, 或导入现有的录像; 对非文字数据进行 OCR 或语音转文字并存储, 供用户提问时 LLM 调用 RAG 使用。

智能问答系统, 通过大语言模型 (LLM) 和 RAG 支持 24 小时即时答疑, 根据用户问题推荐对应的教学片段。提供多轮对话能力, 结合课程知识图谱和用户作业、测验等数据, 提升问答的精准性与上下文理解能力, 支持语音输入及文本分析, 提高交互体验。

课堂智能总结, 通过 LLM, 利用晚间空闲时间, 分段总结教学资料的主旨大意, 并缓存在后端本地, 供问答系统和知识图谱使用。

以上三个后端功能拟在本学期内完成。

课程知识图谱构建, 整合课堂实录、课件、题库、作业等学习资源, 提取关键词并构建知识点关联网络, 支持知识点推理, 实现动态知识更新, 帮助学生系统化学习。

用户数据管理, 存储用户的学习记录、作业提交、测验结果等行为数据, 采集并分析用户的学习路径, 进行个性化推荐。

本系统的前端部分中, 提供列出多个教学视频的 ui 界面, 并提供搜索栏, 并将搜索内容传入后端; 点击搜索后需要实现页面跳转, 提示推荐的视频以及对应的时间点。

在视频观看页面, 有暂停, 快进, 时间跳转功能, 视频清晰度调节功能, 音

量调节功能，提供对应的按键、视频进度条；并且提供留言板（评论区）并记录评论内容；展示教学视频的内容总结大纲，用户能根据大纲内容跳转到对应的教学片段；知识图谱，用户能点击查看某个知识点的相关概念；以及向 AI 询问问题的 AI 答疑框，AI 能结合视频内容、历史作业和测验等资料作答。

在教师角色的管理后台，有课程创建、资料导入、学生管理、权限设置等基础功能。

拟采用 HTML5 + JavaScript (React/Vue) + CSS (Tailwind/Bootstrap) 等常用前端技术实现上述 UI 界面和功能。

6. 环境要求

(1) 开发环境硬件要求

为了确保开发过程中的高效与流畅，团队成员的开发设备应符合以下硬件配置要求：

项目	配置要求
操作系统	Windows 10/11 或 macOS 10.15+ 或 Linux (Ubuntu 20.04+)
处理器 (CPU)	Intel i5 以上 / AMD Ryzen 5 以上
内存 (RAM)	至少 16GB
硬盘空间	至少 500GB SSD 存储 (推荐使用 NVMe SSD)
显卡 (GPU)	至少 4GB 显存, 支持 CUDA (NVIDIA 推荐)
网络	至少 10Mbps 带宽, 稳定的网络连接

(2) 开发环境软件要求

为了确保开发环境的兼容性和项目的顺利进行，团队成员需安装以下软件工具。

基础开发工具

软件工具	用途
Git	代码版本控制，团队协作管理
Node.js (v16+)	JavaScript 环境，前端开发
Python (v3.8+)	后端开发与数据科学开发，算法实现与

	大模型调用
Visual Studio Code	代码编辑器，推荐使用扩展插件如 Python、Copilot 等
Postman	API 调试工具，用于开发和测试后端接口
MySQL/PostgreSQL	关系型数据库，用于数据存储与管理
Elasticsearch	用于全文检索与索引优化

(3) 系统与部署环境要求

为了确保项目的顺利部署与生产环境稳定运行，系统及部署环境需要符合以下要求。

服务器要求

配置	服务器要求
操作系统	Ubuntu 20.04+ 或 CentOS 7/8，推荐使用 Linux 系统
处理器	至少 8 核 CPU（推荐 16 核以上）
内存	至少 32GB RAM（推荐 64GB 以上）
硬盘	SSD 存储，至少 1TB（推荐使用高速 NVMe SSD）
GPU	支持 CUDA，适用于深度学习训练与推理

7. 可行性及潜在风险

技术可行性方面，系统依托 GPT 系列大语言模型和 RAG 技术，可实现高效的内容推荐与问答服务；同时，利用云存储及分布式计算架构确保系统在高并发访问条件下稳定运行。结合知识图谱技术，系统能够有效整合分散的学习资源并支持推理应用，进一步提升智能服务的水平。

条件可行性方面，现有在线教育平台积累了丰富的视频及其他数据资源，为软件开发与测试提供了坚实的数据支持。

针对潜在风险，首先，AI 生成内容准确性不足可能影响问答服务的可信度，故系统将采用 RAG 技术，通过增加知识库检索约束来降低此风险；其次，为防止用户隐私泄露，将严格实施访问权限控制，允许教师角色设置课程权限；最后，

针对系统高并发访问导致的服务器压力问题, 拟采用分布式架构以提升系统的扩展性和稳定性。

8. 承担人员

本项目团队由四人组成, 主要围绕数据处理、算法开发、后端架构、前端交互四个方向进行分工, 确保系统的完整性和高效开发。

角色	人数	主要职责	需要技能
项目负责人 (兼产品经理) 陶恒轩	1	- 统筹项目进度, 协调团队成员工作 - 需求分析与文档编写 - 设计系统架构, 规划数据流	- 计算机基础, 熟悉软件工程 - 需求分析与文档编写能力 - 具备一定的 AI、数据库知识
算法工程师- 后端 何思成	1	- 语音转文字 (ASR) 和 OCR 优化 - 负责知识图谱构建 - 设计和优化 RAG (检索增强生成) 模型	- 大语言模型 (LLM) - 机器学习/深度学习 (PyTorch/TensorFlow) - RAG 模型、知识图谱 - 语音识别 (ASR)、OCR
后端开发工程师 侯宜辰	1	- 负责系统后端开发 (API 服务) - 设计数据库结构, 优化数据存储 - 处理模型推理、数据调用	- Python/Node.js- 数据库 (MySQL/Elasticsearch/Neo4j) - Flask/FastAPI/Django
前端开发工程师 龚政源	1	- 设计用户交互界面 (UI/UX) - 负责前端页面开发 - 处理用户输入、展示 AI 问答和推荐结果	- HTML/CSS/JavaScript- React/Vue.js/前端优化

具体任务分配如下:

开发阶段	项目负责人	算法工程师	后端工程师	前端工程师
需求分	负责整理需求,			

析	编写文档			
数据处理	参与数据整理	负责 OCR、ASR、知识图谱构建	负责数据存储与调用	
RAG 模型		负责 RAG 优化, 实现智能问答	负责 API 封装	
后端架构	统筹架构设计		负责后端开发、接口设计	
前端开发	规划交互逻辑		负责 API 接口联调	负责前端界面开发
系统测试	统筹测试	参与算法测试	负责后端接口测试	参与 UI 测试
优化迭代	负责项目管理	进一步优化模型	处理 API 性能优化	进行前端交互优化

9. 参考文献

[1] 张欣慧, 徐娟.我国高校教师教学发展困境与提升路径——基于人与环境的交互作用[J].中国成人教育, 2022(05).

附: 贡献表

姓名	实际负责部分	贡献
陶恒轩	1、2、3	25%
何思成 (组长)	4、5、7、3 (修订)	30%
侯宜辰	5、6、8	25%
龚政源	4、5	20%