

智辩云枢软件系统开发任务书

1. 背景介绍

辩论作为一项思辨性竞技活动，在高校教育中具有培养逻辑思维、语言表达和团队协作能力的重要作用。然而，当前辩论圈存在以下痛点：

- 立论水平差距大：新生辩手因缺乏对辩论范式（如政策辩中“需根解损”模型）的理解，难以将有深度的思考变成有深度的立论。
- 训练资源匮乏：实时陪练依赖人工匹配，缺乏即时性；复盘环节需专业评委指导，但优质资源分布不均。
- 技术工具缺失：现有辩论辅助工具局限于资料检索，缺乏AI驱动的系统性训练支持。

本系统旨在通过AI技术重构辩论训练模式，推动辩论教育普惠化发展。

2. 欲解决问题

本软件系统希望帮助辩手实现三种功能：

- 立论辅助|缩小新手与顶尖辩手的立论差距，降低辩论范式学习门槛，100%提升新手立论逻辑完整度。
- 智能复盘|提供大赛评委级评估报告，自动识别论证漏洞与战略失误，复盘耗时减少80%，准确定位关键问题。
- 对抗训练|实现7×24小时可定制化对辩模拟，支持1v1至4v4多模态赛制，提升辩手攻防效率。

3. 推荐方案

数据驱动混合结构化引导增强方案（Data+Prompt）

1. 数据预处理关键步骤

代码块

```
1 # 将历史数据转化为结构化提示模板
2 def convert_debate_data(raw_text):
3     template = """
4     [案例开始]
5     辩题: {topic}
6     持方: {stance}
```

```

7     论证结构：
8         - 核心框架：{framework}
9         - 论点层级：{argument_levels}
10        - 数据支撑：{data_sources}
11    战术特征：{tactics}
12    [案例结束]"""
13    # 使用NLP模型提取结构要素（示例）
14    return llm_extract(raw_text, template)
15
16    # 示例输出：
17    """
18    [案例开始]
19    辩题：当今中国应该扩大自主招生比例
20    持方：反方
21    论证结构：
22        - 核心框架：需根解损
23        - 论点层级：政策必要性→实施可行性→后果危害性
24        - 数据支撑：教育部2022年教育统计公报第45条
25    战术特征：归谬法反击+数据证伪
26    [案例结束]
27    """

```

2. 动态Prompt构建算法

▼ 代码块

```

1  def build_dynamic_prompt(topic, stance):
2      # 从向量数据库检索最相似案例
3      similar_cases = vector_db.search(topic, top_k=2)
4
5      # 生成元提示
6      return f"""
7      你正在参加{stance}方辩论，请严格遵循以下规则：
8
9      ## 历史最佳实践
10     {similar_cases}
11
12     ## 当前任务
13     1. 使用与上述案例相同的论证层级结构
14     2. 定义+判准+论点立论结构
15     3. 数据引用必须来自2018年后权威来源
16
17     辩题：{topic}
18     """

```

4. 应用场景

4.1 系统组成要素

AI 辩论训练系统由多个要素组成，包括**计算机软件、设备、服务和系统**等，具体如下：

(1) 软件要素

要素	功能描述
微信小程序前端	用户交互界面，提供立论、对抗、复盘等功能，支持文本/语音输入
辩论AI引擎	负责处理用户输入，生成立论建议、反驳策略，并进行智能复盘
自然语言处理（NLP）模块	用于解析辩论内容，提取关键论点、逻辑关系，判断对抗策略
知识库与案例库	存储各类辩论题目、历史案例、知识点，供 AI 提取参考
评分与反馈系统	负责对用户辩论表现进行评分，并给出优化建议
管理后台	供培训机构或管理者使用，用于监测训练效果，管理用户数据

(2) 硬件要素

设备	作用
云服务器（如阿里云、腾讯云）	运行 AI 模型、存储用户数据
用户终端设备（手机、平板）	运行微信小程序，用户进行交互
语音识别设备（可选）	若支持语音辩论，需额外配置语音输入设备

(3) 服务要素

服务类型	作用
AI 语音识别（如腾讯云 ASR）	语音输入转换为文本，支持语音辩论
AI 语音合成（如百度 TTS）	AI 生成语音输出，提高对抗真实感
AI 语言模型（DS-R1）	处理辩论逻辑，生成智能立论、反驳内容
数据库服务（如 MySQL, MongoDB）	存储用户历史训练数据、知识库
API 网关	处理前端请求，调度 AI 服务

4.2 组成要素的相互关系

1. 用户在前端（微信小程序）提交输入：

- 文字或语音（语音经 ASR 转换为文本）
 - 选择题目、启动对抗训练
2. 前端调用 API 网关，将请求发送到后端
 3. AI 语言模型（deepseek-R1）处理用户输入：
 - 解析用户观点
 - 生成立论、反驳策略
 - 提取知识库案例进行补充
 4. 系统返回响应：
 - 生成 AI 立论/反驳内容
 - 语音合成（如适用）
 - 评分及优化建议
 5. 管理后台提供数据分析：
 - 统计用户训练情况
 - 记录并优化 AI 训练策略
-

4.3 系统部署方式

为了保障系统的高效运行和可扩展性，采用**云端+终端架构**，即：

1. 云端运行 AI 模型和数据库：
 - 部署 AI 语言模型、知识库、评分系统
 - 依赖云计算资源进行高效推理
 2. 微信小程序作为用户端：
 - 仅负责前端交互
 - 通过 API 访问 AI 服务
 3. 可选本地缓存加速：
 - 部分功能（如最近训练记录）可缓存在本地，提高响应速度
-

5. 环境要求

（1）软件依赖

软件组件	作用
微信小程序框架 (WeChat Mini Program)	提供前端 UI 及用户交互
Node.js / Python	开发后端 API
AI 语言模型 (如 GPT-4o)	处理自然语言辩论任务
数据库 (MySQL / MongoDB)	存储训练数据、用户数据
云存储 (如 OSS / COS)	存储历史记录、训练数据

(2) 硬件依赖

硬件	用途
云服务器 (如腾讯云、阿里云)	运行 AI 模型, 处理推理任务
本地计算机 (开发调试)	开发、测试应用
移动设备 (手机/平板)	运行微信小程序, 用户训练

6. 可行性及潜在风险

可行性规划：

阶段	主要任务	完成时间
阶段1：前端页面搭建 & 小规模数据可行性验证	开发前端界面，设计核心交互，构建小规模数据测试 AI 模型可行性	第1个月
阶段2：立论与复盘模块开发	重点开发立论辅助模块 & 智能复盘模块，完善前端逻辑	第2个月
阶段3：立论 & 复盘优化，前端后端联调	继续优化 AI 逻辑，并开始前后端联调，提升整体稳定性	第3个月
阶段4：对抗训练模块 & 后端核心开发	重点实现 AI 对抗训练机制，确保 AI 在长时间对话中不失忆	第4个月
阶段5：系统集成 & 全面功能测试	进行系统整合、测试、性能优化，并最终上线	第5个月

🚩 阶段 1：前端页面搭建 & 小规模数据可行性验证（第1个月）

目标：

- 快速搭建前端界面，确保交互体验良好，并能支持后续 AI 功能接入。
- 构建小规模数据测试集，初步检验 AI 立论与复盘的可行性。

- **前端 & 后端初步对接**，构建 API 结构，确保后续开发顺利。

具体任务：

1. 前端搭建 (React/Vue + Tailwind/Ant Design)

- 设计 & 开发主要页面 (登录页、辩论训练页面、复盘页面)
- 先使用静态数据填充页面，后续对接 AI

2. 小规模数据集测试

- 采集 100-500 条辩论数据进行初步清理
- 训练简单 NLP 立论 & 复盘模型 (如基于 GPT 进行 Prompt Engineering)
- 运行小规模测试，验证 AI 逻辑是否可行

3. API 设计

- 定义 AI 立论、复盘 API 结构
- 设计用户交互流程，确保后端能方便对接

时间安排：

- **第1-2周**：前端页面搭建 & 小数据集清理
 - **第3-4周**：小规模 AI 测试 & API 设计
-

阶段 2：立论 & 复盘模块开发 (第2个月)

目标：

- **立论 AI**：实现 AI 辅助立论逻辑，确保辩手可获得清晰的论点建议。
- **复盘 AI**：实现 AI 解析辩论内容，自动分析论据逻辑 & 提供改进建议。
- **前端优化**：将前端与 AI 模型正式连接，并优化交互体验。

具体任务：

1. 立论辅助模块开发

- 训练 NLP 模型生成论点 (Prompt 设计、Few-shot Learning)
- 通过小样本测试，优化 AI 逻辑 & 过滤无效论点
- 接入数据库存储 AI 生成的论点

2. 复盘分析模块开发

- 设计 AI 分析辩论内容的逻辑
- 通过 NLP 模型识别漏洞，给出改进建议
- 训练数据增强，确保 AI 能应对不同类型的辩论

3. 前端优化

- **API 对接**：前端正式调用 AI 接口
- **交互优化**：确保 AI 生成内容的展示方式清晰可读

时间安排：

- **第5-6周**：开发立论 AI，调试立论逻辑
 - **第7-8周**：开发复盘 AI，优化 NLP 分析
-

阶段 3：立论 & 复盘优化，前后端联调（第3个月）

目标：

- **优化 AI 模型**，提升立论和复盘的质量，使其更贴合实际辩论需求。
- **前后端 API 调试**，保证交互流畅，减少响应延迟。
- **测试 & 修复 bug**，优化用户体验。

具体任务：

1. 优化立论 AI

- 增加案例训练，提升 AI 生成论点的质量
- 处理冗余或逻辑混乱的论点，提高可用性

2. 优化复盘 AI

- 细化 AI 识别逻辑错误的能力
- 增强 AI 对事实性论证的校验

3. 前后端联调

- 确保 AI 生成内容能够被前端正确解析 & 展示
- 调整 API 传输格式，减少请求延迟

4. UI 细节优化

- 增强交互体验，提高用户可视化效果

时间安排：

- **第9-10周**：优化 AI 立论 & 复盘逻辑
 - **第11-12周**：前后端 API 联调 & 交互优化
-

阶段 4：对抗训练模块 & 后端核心开发（第4个月）

目标：

- **开发 AI 对抗训练功能**，实现1v1对辩模拟，让 AI 能持续生成对手观点。
- **完善后端数据库逻辑**，管理用户数据、历史辩论记录等。
- **优化长时间对话记忆**，防止 AI “失忆”问题。

具体任务：

1. 对抗训练 AI 开发

- 设计 AI 生成对立观点的机制
- 训练 AI 识别用户观点，并生成针对性反驳
- 确保 AI 在多轮对话中能记住前文，避免逻辑跳脱

2. 后端开发

- 搭建数据库存储用户辩论历史 & AI 生成数据
- 实现用户管理、权限控制
- 提高 AI API 调用的稳定性

3. 测试 & 调优

- 模拟不同对话场景，优化 AI 逻辑
- 解决记忆丢失 & 逻辑混乱的问题

 时间安排：

- **第13-14周**：开发对抗训练 AI
 - **第15-16周**：后端完善 & AI 记忆优化
-

 阶段 5：系统集成 & 全面功能测试（第5个月）

目标：

- **系统整体集成**，完成所有模块的最终整合。
- **全面功能测试**，确保所有功能稳定运行。
- **上线前优化**，提高 AI 响应速度，优化界面 & 用户体验。

具体任务：

1. 整体系统联调

- 立论、复盘、对抗训练模块全部集成
- 确保系统稳定 & 交互流畅

2. 性能 & 负载优化

- 提高 AI 响应速度，优化 API 调用

- 进行压力测试，确保系统可承载高并发

3. 上线部署 & 用户反馈

- 进行 Beta 测试，收集用户反馈
- 修复剩余 bug，优化交互体验

时间安排：

- **第17-18周：**功能测试 & 性能优化
- **第19-20周：**Beta 测试 & 最终修正

潜在风险及对策：

论据真实性风险

- **问题描述：**AI在提供立论建议或复盘分析时，可能会引用不准确或者不权威的数据来源，导致辩论内容的论据不够可信，影响辩论的实际效果，特别是在面对高水平辩手时，真实性可能会受到质疑。
- **对策：**
 - **数据源筛选与验证：**建立严格的论据数据来源筛选机制，只从可信、权威的公开数据源（如政府统计、公认的学术论文、专业媒体报道等）中提取信息。
 - **实时更新与审查机制：**定期对系统中的知识库和数据进行更新，确保所引用的数据和信息始终是最新且最准确的。
 - **多源对比与交叉验证：**AI在引用论据时，可同时参考多个可靠的数据源，并进行交叉验证，减少偏差和误导。

对抗训练中的长期对话“失忆”问题

- **问题描述：**AI在进行模拟对辩时，可能会面临“记忆”丢失的问题，特别是在多轮、长时间的对辩过程中，AI难以记住早期的论点、反驳和对方的立场，这会导致辩论不连贯或失去焦点，影响辩手的训练效果。
- **对策：**
 - **对话历史管理：**采用对话管理技术，通过智能记忆模块在对话过程中实时存储所有对话信息，并在每一轮时回溯与呈现。这可以确保AI不会“失忆”，能始终根据历史上下文进行回应。
实质上就是把真实的“多轮对话”替换成n个AI能够看到前面所有对话信息的“单轮对话”

7. 承担人员

成员	所属院校	主要职责
韩柯（组长）	武汉大学计算机学院	总体架构设计，协调团队任务，负责项目进度管理，统筹技术与产品。
陈涛	武汉大学计算机学院	AI 模型集成，负责 DeepSeek API 接口调用、数据处理优化、AI 推理加速。
叶海峰	武汉大学计算机学院	前端开发，负责微信小程序 UI 设计、交互逻辑、用户体验优化。
郭垚	武汉大学计算机学院	后端开发，负责 API 服务器、数据库管理（MySQL/MongoDB）、系统部署。
彭杰	武汉大学计算机学院	知识库与数据管理，收集辩论案例、构建知识库、实现 RAG 检索增强生成。
高闯巍	武汉大学计算机学院	语音交互开发，集成 ASR（语音识别）和 TTS（语音合成），优化语音交互流程。
陈厚德	同济大学辩论队	辩论策略与测试，提供辩论场景设计，测试 AI 逻辑合理性，优化立论与驳论。