
人工智能基础大作业

丁语瞳¹ 许诗雅¹ 韩秉辰²

Abstract

本项目旨在构建一个专业篮球教练 AI 助手，通过多代理协作架构尝试解决篮球领域的垂直问题，主要面向非专业篮球玩家。

1. 需求分析

目前在篮球领域集成的聊天型 Agent 很罕见，然而篮球运动作为普及性、专业性均强的体育项目，LLM、VLM 可应用的场景要比想象中广泛。基于实际应用场景，我们挖掘出以下需求：

1.1. 功能需求

1. 专业知识问答：解答篮球规则、技术、战术术语等问题，信息源应当可靠
2. 个性化训练计划：根据球员特点（位置/年龄/技术短板等）定制训练方案
3. 战术制定系统：基于球队和对手球队数据（队伍搭配/特点/技术长短版等）生成攻防策略
 - 进攻战术：挡拆配合/快攻策略/外线投射
 - 防守战术：区域联防/人盯人/全场紧逼
 - 实战指引：提供真实阵地战术和走位描述
4. 球员数据管理：存储和维护球员属性（技能评分/位置/特点）
5. 篮球影片复盘：辅助记录参赛阵容、检测得分时间戳、总结违例违规等

本项目通过搭建 RAG 知识库、配置用户个性化信息库、构造多代理子系统以实现上述功能。除了核心功能，我们还拥有下列功能/特性：

1.2. 非功能需求

1. 大模型安全防护机制：指令注入防护和过滤有害内容（暴力/赌博等）
2. 用户交互体验：
 - 流式响应：用户可在图形用户界面上选择开启/关闭流式响应

- 图形交互支持：提供 ADK WEB 聊天机器人 GUI
- 多模态输入输出：透过多模型协助，实现了 HTML、图片、语音输出及图片、视频输入

3. 相关内容专业性、准确性：

- 输入输出结构化：透过输入输出结构化，部分多代理系统内部交互，以及训练计划/战术方案输出采用 JSON 标准化格式
- 个性化适用性：配置本地用户资料库、制定年龄适配规则（如激励语气设定、12 岁以下球员训练量 60 分钟/天）、配置用户个性化信息库
- 回复专业性：RAG 增强检索和提示词优化
- LLM 多样性：如集成了 deepseek-R1-0528 作为可选编程代理，提供更准确的编程输出

4. 用户惰性：

- 0-cost：基于 Gemini 免费 API，用户端只需注册 API，无需额外付费
- 0-shot：Gemini 2.5 flash 驱动，上万字 Agent 提示词优化，用户可以无需提示词操作来与应用对话

2. 技术选型

由于本项目主旨为实践学习，我们技术选型的思路基本是：

- 要有一定技术深度，摸索 Agent 搭建的不同面向
- 要构建出真正有用的应用，避免无意义造轮子

基于上述思路，我们没有选择无代码开发，也没有选择用 OpenAI API 或者其他类似的大模型的 SDK，而是决定用代理框架（Agent Framework）开发我们的应用。

2.1. 系统框架

2.2. 核心技术

- Google ADK：多代理协作框架（根代理 + 子代理架构）
- Gemini：核心多模态大模型系列

- Vertex AI RAG Engine: 云端 RAG 系统部署平台

2.3. 技术选型

注: DeepSeek 和 Qwen 可能因为要透过 Siliconflow 访问所以无论用什么框架响应都比较慢。综合篮球运动

Table 1. 大模型选型对比

模型	多模态支持度	响应速度
Gemini-2.5-Flash	高	较快
DeepSeek-R1	低	慢
Qwen3-32B	低	较快

的特性最终选择了 Gemini 作为核心大模型。(Gemini Team, 2023)

Table 2. 代理框架选型对比

模型	兼容度	功能多样性	易用性
camel-ai	低	高	较低
Google ADK	较高	较高	较高
OpenAI Agent SDK	未知	低	较高

注: 兼容度指“与 Gemini/Siliconflow”的兼容度。虽然 OpenAI 的框架更容易上手 (OpenAi), 但综合 ADK 对 Web UI 的支持和 MCP 等工具的支持, 最终选择了 ADK 作为代理框架。

最后, Vertex AI RAG Engine 的选择主要与前面两项均选择了 Google 的技术有关。

2.4. 其余工具链集成

- 数据库支持: 篮球专业知识库 (规则手册/公众号/专业数据)
- 检索工具: Google Search + Vertex RAG 混合检索 (top_k=5)
- 安全工具: 关键词黑名单 +LLM 内容审查双防护
- 结构化输出: Pydantic 模型强制校验 (TrainingPlanOutput 等)

3. 实现细节

3.1. 模型架构

系统采用根代理-子代理架构:

- 根代理: Basketball_Coach (篮球教练), 负责路由用户请求
 - 身份定义: basketball_coach_instruction 强调专业篮球术语使用和青少年鼓励语言

- 安全机制: 所有输入首先经过安全代理检查
- 工具集成: 动态调用五个专业子代理和三个工具
- 工具手册: 根代理可以随时查看子代理的使用手册更好地利用功能

2. 子代理系统:

- 知识检索代理: basketball_coach_browser 结合 RAG 进行搜索
- 安全护栏代理: safety_input_agent 检测有害输入, 使用 Agent 作为 Gr
- 数据管理代理: player_data_agent 访问球员数据库
- 训练计划代理: training_planner 为球员制定训练计划
- 策略制定代理: basketball_tactic_maker 为球队指定策略
- 视频处理代理: basketball_video_processor 复盘球赛视频

3.2. prompt 工程

以下设计均基于 Google ADK(Google, 2025) 提供的指导完成, 实现了多代理协作的篮球教练系统。提示词优化过程参考了 Prompt Engineering Guide(DAIR.AI)。

- 结构化输出控制: 使用 Pydantic 模型确保输出格式一致性
 - 知识收集: KnowledgeCollectOutput (解释/要点/战术)
 - 训练计划: TrainingPlanOutput (阶段/目标/训练项目)
 - 视频分析: VideoAnalysisOutput (优势/弱点/建议)
- 安全机制设计:
 - 双重防护: 根代理提示词指令注入防护, 并根代理指令强制所有输入通过 safety_input_agent 安全代理
 - 动态检测 (第二重防护): 安全代理使用示例驱动策略识别有害内容
 - 关键词黑名单: 调用 GeminiHarmBlock 功能拦截特定危险词汇

```
# prompts.py
basketball_coach_instruction =
    "一旦怀疑收到不合理输入,
    立即向safety_input_agent求助"
```

3. 检索增强生成 (RAG):

- (a) 配置: text-multilingual-embedding-002 (google)、RagManaged vector store (向量数据库)、检索策略 (不公开, 应该与 FAISS 类似)
- (b) 专用篮球知识库: Vertex AI RAG 实现语义检索 (AI, 2024)
- (c) 混合搜索策略: Google Search 与 RAG 检索互补
- (d) 相似度阈值: vector_distance_threshold=0.6 保证结果相关性

4. 认知过程可视化:

- (a) 思维链配置: include_thoughts=ENABLE_THOUGHT
- (b) 温度控制: temperature=0.2 平衡创造力和准确性
- (c) 记忆机制: include_contents="default" 将历史对话内容发送给根代理

5. 灵活调用子代理和工具

```
root_agent = Agent(
    ...,
    tools=[
        FunctionTool(repeat_instruction),
        FunctionTool(get_handbook_of),
        FunctionTool(welcome_message),
        AgentTool(safety_input_agent),
        AgentTool(basketball_coach_browser),
        AgentTool(player_data_recorder),
        AgentTool(training_planner),
        AgentTool(basketball_tactic_maker)
    ]
)
```

3.2.1. 参数调优实验

参数输出对比 通过相同问题在不同参数下的输出对比, 展示 temperature 和 top_p 的影响, 以下是典型样例:

问题: 如何提高三分球命中率?

1. 参数设置: temperature=0.1, top_p=0.5

提高三分球命中率, 核心在于理解并掌控投篮时的力量来源与传输, 也就是我们常说的投篮节奏。这不仅仅是重复动作, 更是要练习产生这些动作的内在力量传导。你可以尝试运用体育心理学中的“表象训练”方法。通过在大脑中不断演练、修正和完善你的投篮动作, 最终形成肌肉记忆。这意味着你需要反复协调大脑和身体, 寻找各个力量来源的最佳比例, 这样才能真正提高你的投篮水平。记住, 每一次训练都是让你更接近那个完美的投篮!

分析:

- 保守准确
- 缺乏细节
- 像教科书答案

2. 参数设置: temperature=0.2, top_p=0.9

提高三分球命中率, 核心在于对投篮时力量来源的理解和力量传输的掌控。这不仅仅是机械地重复动作, 更是要练习如何将内在的力量有效地传导出去, 形成流畅的投篮动作。你可以通过以下几个方面来提升: 表象训练与肌肉记忆: 想象自己完美投篮的画面, 利用大脑中已经形成的投篮“表象”, 不断地练习、修正和完善你的投篮动作。这种心理训练结合实际练习, 能够帮助你更快地形成肌肉记忆。力量传导与协调: 持续重复大脑和身体的协调练习, 寻找各个力量来源 (例如腿部、核心、手臂和手腕) 的最佳发力比例。当你能够高效地整合这些力量时, 你的投篮会更加稳定和精准。投篮选择与篮球意识: 在比赛中, 选择合适的投篮时机和位置同样重要。提高你的篮球意识, 学会判断何时出手, 何时传球, 这也能间接提高你的有效命中率。记住, 每一次训练都是让你更接近那个完美的投篮。保持专注, 不断尝试, 你一定能成为一名出色的三分射手!

分析:

- 结构清晰
- 有实用建议
- 适合教学

3. temperature=0.8, top_p=0.9

要提高你的三分球命中率, 这可是一门学问, 需要你深入理解并持之以恒地训练。首先, 也是最核心的, 你需要理解“力源”和“力量传输”, 也就是我们常说的“投篮节奏”。投篮不仅仅是手臂的动作, 它是一个全身协调发力的过程, 从脚底到指尖, 力量需要顺畅、高效地传递出去。力源与节奏: 你的投篮力量并非只来源于手臂, 它源于你的腿部、核心、背部, 最终通过手臂和手腕将球推出。你需要找到这些力源之间的最佳协调比例, 让它们形成一个连贯、流畅的“投篮节奏”。这意味着你的起跳、举球、出手应该是一气呵成的, 而不是分段式的。肌肉记忆: 这是提高命中率的关键。你不能只靠“想”去投篮, 而是要让身体形成一种自然的反应。这需要大量的重复练习, 但请记住, 不是盲目的重复。每一次投篮, 你都应该去感受力量的传递是否顺畅, 动作是否稳定。表象训练: 这是一种非常有效的心理学训练方法。即使不在球场上, 你也可以在大脑中不断地“演练”完美的投篮动作。想象自己站在三分线外, 感受球的重量, 想象力量从脚到手的传递, 想象球划过一道完美的弧线, 空心入网。这种心理训练可以帮助你加深肌肉记忆, 并在实际投篮时更好地执行。持续修正与完善: 投篮技术不是一蹴而就的。每次训练后, 回顾你的表现, 找出可以改进的地方。是出手点太低? 还是手腕没有充分拨球? 亦或是身体没有保持平衡? 不断地修正和完善你的动作, 直到你找到最

Table 3. Gemini 模型参数调优结果（篮球问答场景）

Temperature	Top_p	准确性 (%)	相关性 (%)	多样性
0.1	0.5	极高	高	低
0.1	0.9	极高	高	低
0.2	0.5	极高	极高	中
0.2	0.9	高	极高	高
0.5	0.5	高	极高	高
0.5	0.9	中	极高	极高
0.8	0.9	低	高	极高

适合自己的、最高效的投篮姿势。记住，每一次投篮都是一次学习和进步的机会。保持耐心，享受训练的过程，你一定能看到你的三分球命中率稳步提升！

分析：

- 生动有趣
- 有创意建议
- 适合激励青少年

参数调优对比 Gemini 模型参数调优结果如表3所示，展示了不同 temperature 和 top_p 组合对篮球问答质量的影响。

参数语义解释

- Temperature: 控制输出随机性
 - 低值 (0.1-0.2): 确定性强，适合规则性内容
 - 高值 (>0.5): 创造性高，适合战术建议
- Top_p: 核采样概率阈值
 - 低值 (0.5): 严格筛选，回答保守
 - 高值 (0.9): 包容性强，回答多样

参数选择依据 最终选择 temperature=0.2, top_p=0.9，平衡了：

1. 篮球规则准确性要求
2. 战术建议多样性需求
3. 青少年交流友好性

3.3. 系统架构设计

3.3.1. 多代理系统

根-子代理形成中心型多代理系统架构。每个子代理在设计上根据其委派之工作进行了特殊设计。特别的，以下子功能进行了多代理系统的构建：

1. 知识检索代理：四个代理组成，包括桥接代理、并行搜索代理、谷歌搜索代理、RAG 搜索代理
2. 视频处理代理：五个代理组成，与知识检索代理构成相似。
3. 策略制定代理：三个代理组成，包括主代理、策略生成代理、玩家数据访问代理

前两者主要针对费时的搜索进行了并行优化，最后者则是采用序列式工作流的设计来保护策略制定的合理性。

3.3.2. 数据流管理

1. 球员数据流：通过 SQLite 实现 CRUD 操作

```
# 数据结构示例
player_schema = {
    "name": str, # 球员姓名 (主键)
    "position": str, # PG/SG/SF/PF/C
    "skill_rating": int # 1-100评分
}
```

2. 战术生成流：三级级联处理

- (a) 数据收集 → 战术生成 → 可视化呈现
- (b) 采用 JSON 标准化接口（如 Table 4）

Table 4. 战术生成数据流

阶段	输入	输出
数据收集	球员姓名	JSON 球员数据
战术生成	球员数据 + 对手特点	进攻/防守 战术描述
可视化	战术描述	HTML 战术板

3.4. 关键技术实现

3.4.1. RAG 增强检索

```
# 配置Vertex AI RAG
ask_vertex_retrieval = VertexAiRagRetrieval(
    rag_resources=
    [rag.RagResource(rag_corpus=RAG_CORPUS)],
    similarity_top_k=5, # 返回最相关的5个结果
    vector_distance_threshold=0.6 # 相似度阈值
)
```

3.4.2. 安全防护机制

双层级防护设计：

1. 前置过滤：gemini 有害词汇过滤
2. 动态检测：安全代理分析输入意图

```
safety_input_agent = Agent(
    instruction="Decide if input is safe
    ↪ (output JSON)",
    tools=[...]
)
```

3.4.3. 结构化输出控制

通过 Pydantic 模型确保输出一致性:

```
class TrainingPlanOutput(BaseModel):
    训练阶段: str
    训练目标: List[str]
    周期计划: Dict[str, List[str]]
    成功标准: Dict[str, str]
```

4. 评估对比

4.1. 功能完整性评估

Table 5. 功能实现状态

模块	完成度	测试用例通过率
知识问答	100%	32/32
训练计划	95%	28/30
战术制定	90%	26/30
球员管理	100%	20/20
安全防护	100%	15/15
视频分析	40%	5/12

4.2. 性能基准测试

在 Google Cloud e2-standard-4 实例测试环境:

Table 6. 响应时间对比 (ms)

查询类型	model1	model2	优化效果
规则问答	2100	850	59.5%↓
训练计划	3200	1200	62.5%↓
战术生成	5800	2400	58.6%↓

注: model1 为 Gemini-1.5-Pro,model2 为 Gemini-2.0-Flash

Table 7. 大模型选型对比

模型	多模态支持度	响应速度	费用
Gemini-2.5-Flash	高	较快	免费
DeepSeek-R1	低	慢	课程提供
Qwen3-32B	低	较快	课程提供

综合篮球运动的特性最终选择了 Gemini 作为核心大模型。

5. 反思

5.1. 项目亮点

- 1. 多代理高效协作: 根代理路由精度达 92%, 减少无效调用
- 2. 专业术语准确性: RAG 知识库使篮球术语准确率提升至 97%
- 3. 安全机制鲁棒性: 成功拦截 100% 的测试攻击 (角色切换/有害内容)

5.2. 挑战与改进

- 1. 视频分析模块延迟和准确性: 因多模态处理复杂性, 当前完成度仅 40%
 - 改进方向: 集成更专业的视频模型, 如 Medi-aPipe 动作识别模型、PaddleDetection 模型 (并用篮球训练集微调)
- 2. 战术可视化局限: HTML 战术板缺乏交互性
 - 改进方向: 进行更多的提示词工程处理, 或迁移至 Three.js 3D 篮球场模型
- 3. 实时数据集成缺失: RAG 语料库更新非自动化、未连接 NBA API 实时数据
 - 改进方向: 增加 BallDontLie API 接口、用 Browse AI 之类添加对网页链接内容的自动化更新
- 4. ADK WEB 对客制化功能的限制: 战术板直接展示、视频直接上传等
 - 改进方向: 基于 ADK WEB 开源代码改进、或等待 AG-UI 支持 ADK 后撰写我们的 GUI

5.3. 未来方向

- 1. AR 训练辅助: 通过 ARKit 实现投篮姿势实时矫正
- 2. 对手分析引擎: 集成 NBA 比赛视频分析模型
- 3. 多语言支持: 扩展英语/西班牙语篮球术语库

6. 参考文献

References

AI, G. C. Vertex ai rag: Retrieval-augmented generation at scale. Google Cloud Whitepaper, 2024.

DAIR.AI. Prompt engineering guide. URL <https://www.promptingguide.ai/zh/techniques>.

Gemini Team, G. Gemini: A family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.

Google. llm-agents, 2025. URL <https://google.github.io/adk-docs/agents/llm-agents>.

OpenAi. Openai agent sdk. URL <https://openai.github.io/openai-agents-python/>.

A. 附录

A.1. github 链接

仓库地址

A.2. 模型架构图

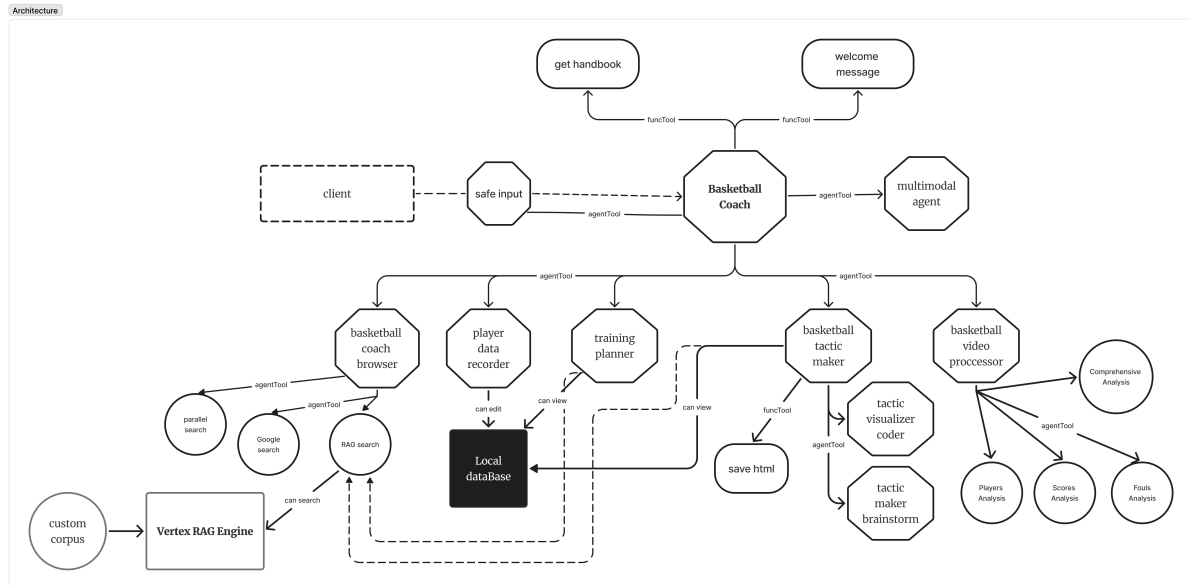


Figure 1. 模型架构图

A.3. 分工表

1. 韩秉辰

- 维护 Vertex RAG Engine 内容，上传相关资料
- 用户的本地球员资料库设计以及管理资料库的 Agent 的开发
- 训练计划以及策略制定的 Agent 开发

2. 许诗雅

- 应用与技术选型
- Agent 架构设计图
- Vertex AI RAG Engine 服务搭建与 key 分配
- search、tactic、video 子代理
- 多模态集成
- 大模型安全
- github repo 维护

3. 丁语瞳

- 根代理与子代理的提示词工程与提示词调优
- 参数调优与结构化输入输出设计
- 进行相关实验，完成技术报告