欣禹行｜Agent 开发 / LLM 应用工程实习

基本信息

- 中央民族大学（985）｜计算机科学与技术（本科）｜2023-2027
- GPA：3.6
- 英语：IELTS 7.0，CET-6
- 邮箱：huali6641@gmail.com
- 个人主页：https://seanhomepage.top
- GitHub：https://github.com/takagibit18

教育背景

中央民族大学（985）｜计算机科学与技术（本科）

核心课程：数据结构、操作系统、计算机网络、计算机组成原理、数据库系统、软件工程、自然语言处理

求职意向

Agent 开发 / LLM 应用工程实习

专业技能

- Python 工程化：使用 Python 异步编程、类型约束与结构化脚本组织实现 CLI 工具、测试用例与实验脚本。
- Agent / LLM 应用：使用 OpenAI-compatible API、结构化输出、工具调用编排与多阶段状态流实现 Agent 应用。
- RAG / 检索评测：使用 BM25、稠密向量检索、混合检索、Reranker、RAGAS 与黄金集评测方法验证召回、引用正确性与拒答策略。
- 服务化与工程化：使用 FastAPI、Docker Compose、Redis、结构化运行产物与可观测性工具实现服务化接口、缓存会话与链路追踪。

项目经历

MergeWarden｜代码审查与调试 Agent

技术栈：Python、Click、Pydantic、OpenAI-compatible API、Tool Calling、ReAct Loop、Docker、Pytest、GitHub PR Workflow

- 面向本地仓库、PR patch 与错误日志构建代码审查 / 调试 Agent，将 Review / Debug 输入组织为结构化 JSON，并输出 severity、location、evidence、suggestion、confidence 等可消费结果；项目累计约 9.5K 行 Python，覆盖 Agent 编排、工具系统、评测与测试模块。
- 设计 5 阶段 Agent 编排循环：prepare -> analyze -> execute_tools -> format -> continue/stop，用 ContextState 维护目标、约束、决策链、当前文件与错误列表；通过 submit_review / submit_debug 伪工具承接结构化终稿，避免纯文本解析不稳定。
- 优化多轮 ReAct 工具反馈链路，将工具结果以标准 assistant tool_calls + tool result 回灌，并引入最近 3 轮 raw feedback、digest 索引、prior_tool_results_summary 与 force-submit 兜底；在评测样例中将 review loop 从 4 轮无 submit、空列表 / hit_rate=0 修复到 matched 1/1、hit_rate=1.0、pass@k=1.0，且 5 次工具调用 args_digest 两两不同，闭环定位 Agent “失忆”导致的重复读文件问题。
- 构建 6 个默认工具与安全执行体系：4 个只读工具支持并发读取、目录枚举、glob / grep 检索，read_file 支持 offset/limit 按行切片以优先读取 diff 附近内容；2 个执行工具通过路径校验、命令 allowlist、shell=False、Docker argv 参数化执行和环境变量两层过滤降低代码执行风险。
- 建立可观测评测与工程化交付闭环，沉淀 11 个 golden fixture、33 份评测报告、181 条 Agent JSONL 运行日志、24 个测试文件与 166 个测试用例；基于本地 git 可追溯 25 个 merged PR，形成审查、调试、执行、评测边界清晰的 Agent 工程实践，而非简单堆叠 Planner / Executor / Critic。

ShotgunCV（面向海投的 Resume Ops 流水线）

技术栈：Python、TypeScript、CLI、OpenAI-compatible API、Run Artifacts

- 面向海投场景设计 Pipeline-first AI Resume Ops，围绕多岗位 JD 输入完成解析、简历变体生成、评分、排序与投递策略输出，支持批量求职决策。
- 采用 Pipeline-first 设计与结构化评估中间层，围绕 ScoreCard、GapMap、RankingExplanation、LLMAssessment 组织岗位匹配排序、证据绑定与风险标记。
- 采用 LLM-primary + guardrail fallback 混合决策机制，在引入 LLM Judge 结构化评估的同时，用规则护栏约束最终分数与投递建议，并在模型失败时回退到规则评分。

Vertical Support RAG Agent（垂类知识库智能客服）

技术栈：Python、FastAPI、LangChain、LangGraph、Qdrant、BM25、BGE-M3、Reranker、Redis、Docker Compose、RAGAS

- 面向 3C 数码产品售后 / 使用咨询场景实现知识库智能客服，支持 FAQ 问答、故障排查、售后政策解释、无证据拒答与转人工建议。
- 基于 LangChain + LangGraph 构建双 Agent 状态流，解耦 AnswerAgent 的对客回答职责与 PolicyGuardAgent 的权限审查职责，确保知识库检索、信息放行与最终响应生成具备清晰边界。
- 使用 Qdrant + BM25 + BGE-M3 + Reranker 实现两阶段混合检索排序，并结合 Redis 支撑会话摘要、重复问题缓存、检索结果缓存与限流；建立黄金集 + RAGAS 评测闭环，跟踪 retrieval hit rate、context precision、faithfulness、citation correctness、refusal accuracy 等指标。

自我评价

- 采用评测驱动迭代方式开发 Agent / RAG 应用，关注结构化输出、trace 复盘、最小权限边界与可复现实验资产沉淀。
- 熟悉 PR、Issue、Code Review 等协作流程并具备远程协作开发经历。
- 长期使用 Codex、Cursor 等 AI 开发工具，并持续跟踪 Agent / RAG 工程实践与行业动态。