施奇

AI 应用开发工程师 | RAG / Agent 工程化方向

合肥 | 15375337717 | 591990368@qq.com
期望薪资：20-25K

具备企业级 RAG、Agent Runtime、Tool Calling、Text-to-SQL 与轻量微调等 AI 应用实践经验，拥有 Java 后端、Python FastAPI、分布式系统与全栈交付基础，能够将 AI 能力接入复杂业务系统，推动应用从原型验证到工程化落地。

RAG 知识库 | Agent Runtime | Tool Calling | Planner / Replanner | Python / FastAPI | Embedding / 向量检索 | Qdrant / Chroma | Hybrid Search / Rerank | LangChain | Java / Spring Boot | Redis / MQ | Elasticsearch | Vue3 / React/ 全栈交付

个人优势

- **AI 应用工程化能力：** 具备企业级 RAG、Agent Runtime、Tool Calling、Text-to-SQL 和轻量微调等 AI 应用实践经验，能够结合业务场景完成从原型验证、接口封装、权限控制到工程化落地的完整闭环。
- **企业级 RAG 系统设计能力：** 熟悉文档上传、异步解析、结构化 Chunk、Embedding、向量索引、Hybrid Search、Rerank、权限过滤、可追溯问答和链路可观测等 RAG 核心链路，具备从知识接入到问答服务化的系统设计经验。
- **Agent Runtime 与工具调用控制能力：** 理解任务型 Agent 执行链路，具备 Planner / Replanner、工具注册、Tool Calling、多步骤调度、执行状态追踪、人工确认、风险控制和审计日志等 Agent 工程化设计能力。
- **后端架构与分布式系统能力：** 具备 Java / Spring Boot / Spring Cloud、Redis、MQ、Elasticsearch、PostgreSQL 等复杂业务系统经验，能够处理异步任务、数据一致性、权限隔离、检索优化和系统稳定性问题，为 AI 能力接入业务系统提供工程基础。
- **全栈交付与项目推进能力：** 具备 Python / FastAPI、Vue3 / React、接口设计、数据处理、前后端联调和项目交付经验，能够独立推进 AI 应用从需求分析、技术选型、架构设计、Demo 验证到系统落地。

项目经历

企业级知识库 RAG 问答系统 | Python + FastAPI + Qdrant　AI应用开发（核心模块实现）　2025.05 - 至今

项目描述：面向企业内部知识问答、文档检索和知识沉淀场景，基于 Python + FastAPI 构建企业级 RAG 知识库系统，重点实现文档上传、异步解析入库、结构化解析、Chunk 切分、Embedding 向量化、Hybrid Search、Rerank、权限过滤和引用式问答等核心能力。系统通过上传与入库解耦、文档状态机、失败重试、结构化切片和检索增强，解决企业文档接入不稳定、知识检索不准确、回答不可追溯和多租户权限隔离等问题。

1. 设计独立文件上传模块，将文件上传与文档解析、切片、向量入库解耦，支持单文件、批量文件、大文件分片、断点续传、上传进度、上传取消和重复文件识别，提升复杂文档接入稳定性。
2. 实现上传入口安全校验与对象存储机制，覆盖文件大小、后缀白名单、MIME 类型、文件头、hash、用户权限、知识库状态、租户归属和容量限制等校验，原始文件存储至对象存储，数据库仅登记文件元数据。
3. 设计异步文档入库任务模块，将解析、切片、Embedding 和 Qdrant 入库从上传请求中拆分出来，通过异步任务处理，并结合文档状态机、失败重试、断点恢复和幂等控制提升入库链路可靠性。
4. 升级文档解析模块为结构化解析链路，统一输出 Markdown 与 JSON blocks，保留标题层级、段落顺序、页码、表格结构、代码块、公式、章节路径、来源位置和阅读顺序等信息，并通过 source mapping 支撑引用定位。
5. 设计结构化 Chunk 切分与 Parent-Child 机制，优先按标题和段落切分，表格、图片说明、代码块整体保留，超长内容按 token 二次切分并保留 overlap；使用 child chunk 提升召回精度，使用 parent chunk 提升上下文完整性。
6. 完善 Chunk 元数据模型，记录 chunk_id、doc_id、kb_id、tenant_id、section_path、page_start、page_end、token_count、chunk_type、parent_chunk_id 等信息，为权限过滤、引用返回、Trace 排查和索引重建提供基础。

7. 实现 Hybrid Search、Rerank 和权限过滤链路，结合向量语义召回与关键词召回进行多路检索，通过 Rerank 对候选 chunk 二次排序，并基于 tenant_id、kb_id、doc_id 和用户权限进行过滤，满足企业多租户知识隔离场景。
8. 设计文档索引重建与删除清理流程，支持文档重新入库时旧 chunk、旧 vector 和旧 metadata 清理，并在文档删除时同步清理对象存储、数据库记录和 Qdrant 向量索引，保证知识库内容与检索索引一致。
9. 实现引用式 RAG 问答与无上下文兜底策略，支持多路召回、权限过滤、Rerank、上下文组装、Prompt 构造和大模型回答生成，返回来源文档、章节路径、页码、chunk 信息和相关性分数，降低幻觉风险。

项目成果：

1. 完成从文件上传、异步解析、结构化切片、向量入库、检索增强到引用式问答的企业级 RAG 闭环。
2. 通过上传解耦、异步任务、状态机、失败重试和幂等控制，提升复杂文档入库链路稳定性。
3. 通过结构化解析、source mapping、标题路径注入和 Parent-Child Chunk，提升检索准确性、上下文完整性和回答可追溯性。
4. 已实现 Hybrid Search、Rerank 和权限过滤，增强关键词、编号、表格字段、专有名词和多租户权限隔离场景下的检索效果。

技术栈： Python / FastAPI / LangChain / OpenAI-Compatible API / sentence-transformers / Chroma / Qdrant / SQLAlchemy / SQLite / RAG / Embedding / OpenTelemetry

---

AgentWorks 通用任务型 Agent Runtime（Planner + Tool Calling + Skill 扩展） AI 2025.08 - 至今

应用开发（核心模块实现）

项目描述： 设计并实现一个通用任务型 Agent Runtime，面向代码分析、文档生成、Bug 排查、运维巡检等多类任务场景，支持用户目标理解、任务拆解、工具注册、执行状态追踪、多步骤任务调度、人工确认、审计日志和插件化 Skill 扩展。系统通过“目标理解 - 任务规划 - 工具选择 - 步骤执行 - 结果观察 - 反思修正 - 状态记录 - 结果输出”的执行链路，提升复杂任务的可执行性、可恢复性和可解释性。

1. 设计通用 Agent Runtime 架构，将用户目标、任务计划、工具调用、执行步骤、观察结果、人工确认和最终输出进行统一抽象，支持多步骤任务的状态化执行。
2. 重点实现任务拆解 Planner 模块，围绕 Intent Router、Context Collector、Tool-Aware Planner、Plan Validator、Risk Classifier、Dependency Resolver、Execution Plan 和 Replanner 构建可验证、可执行、可恢复、可审批、可解释的规划流程。
3. 设计工具注册与 Tool Calling 机制，将文件操作、代码分析、文档生成、命令执行、知识检索、运维查询等能力抽象为标准工具，支持 Agent 根据任务目标和上下文选择合适工具完成执行步骤。
4. 实现执行状态追踪能力，记录任务状态、步骤状态、工具调用参数、执行结果、失败原因和中间观察结果，为任务恢复、失败重试、问题排查和过程回放提供基础。
5. 设计人工确认与风险控制机制，对命令执行、文件修改、环境操作等高风险步骤进行风险分类和人工确认，避免 Agent 在未授权情况下直接执行危险操作。
6. 实现插件化 Skill 扩展机制，将代码分析、Bug 排查、文档生成、运维巡检等场景能力封装为可扩展 Skill，降低新增任务类型的接入成本。
7. 提供前后端基础工程结构，后端基于 FastAPI 提供 Agent 任务、工具、执行状态和审计接口，前端支持任务提交、执行过程查看、人工确认和结果展示。

项目成果：

1. 完成 AgentWorks 通用任务型 Agent Runtime 原型，打通从用户目标输入、任务拆解、工具调用、多步骤执行、状态记录到结果输出的核心闭环。
2. 构建高标准 Planner 流程，支持任务计划校验、风险分类、依赖解析和失败后的 Replanner，为复杂任务的可靠执行提供基础。
3. 通过工具注册、状态追踪、人工确认和审计日志机制，提升 Agent 执行过程的可控性、可恢复性和可解释性。

技术栈： Python / FastAPI / Agent Runtime / Planner / Tool Calling / Skill Plugin / SQLAlchemy / SQLite / React / REST API

分布式跨境电商 ERP 综合管理平台 系统架构设计 / 全栈开发 2023.03 - 2025.05

项目描述： 面向 Shopee、TikTok、速卖通等跨境平台，构建订单、商品、物流、库存、财务等一体化 ERP 系统，支撑多租户、多平台、高频 API 同步场景。

1. 基于 Spring Cloud Alibaba + Dubbo 构建微服务架构，集成 Nacos 实现配置管理与服务治理，支撑多平台订单、库存、物流等核心业务统一管理。
2. 设计多租户权限体系，实现租户逻辑隔离、动态权限控制、组织架构授权与接口级访问控制，提升系统扩展性与安全性。
3. 负责 Shopee、TikTok、速卖通等平台 API 对接，基于 MQ、Redis、限流与异步任务机制优化高频接口同步、失败重试和数据延迟问题。
4. 设计“实时消费 + 离线补偿 + 人工回放”闭环机制，结合 XXL-Job、分布式锁与状态机保障订单、库存、售后等数据最终一致性。
5. 基于 PostgreSQL 分区与 Elasticsearch 索引优化大规模业务数据查询，提升订单日志、商品数据和库存流水等场景的检索效率。
6. 基于 Flowable 实现复杂业务流程编排，支持订单审批、异常处理、数据留痕等标准化流程。

项目成果： 核心接口响应降低 40%，并发处理能力提升 3 倍以上，核心链路可用性提升至 99.99%，有效降低订单同步丢失和高延迟问题。

服务器智能运维与告警分析系统（Agent Client + RAG + Tool Calling） AI应用开发 2024.04 - 2025.05
（核心模块实现）

项目描述： 面向多服务器运维场景，基于 Agent Client、RAG 和 Tool Calling 构建智能运维与告警分析原型系统。系统在多台服务器部署轻量客户端，用于采集主机指标、查询进程状态、读取系统日志和上报服务状态；中心服务端负责告警接入、知识检索、工具调用编排、原因分析、处置建议生成、人工确认和审计记录，形成从服务器状态感知、上下文查询、智能分析到人工确认的 P0 闭环，提升多服务器环境下常见故障的排查效率和处置标准化程度。

1. 设计多服务器 Agent Client + 中心服务端架构，在每台服务器部署轻量客户端，支持主机信息查询、CPU / 内存 / 磁盘 / 网络指标采集、进程状态查询、系统日志读取和服务状态查询。
2. 设计服务器告警分析链路，围绕“告警接入 - 主机定位 - 工具查询 - 知识检索 - 原因分析 - 处置建议 - 人工确认 - 审计记录”构建 P0 闭环，辅助处理 CPU 负载过高、内存异常、磁盘空间不足、网络异常、进程异常和系统日志报错等问题。
3. 构建运维 RAG 知识库，接入服务器巡检规范、故障处理 SOP、Linux 排查手册和历史告警处置记录，为告警语义理解、原因分析和处置建议生成提供上下文依据。
4. 基于 Tool Calling 封装服务器运维查询工具，将主机指标查询、日志查询、进程查询、服务状态查询、磁盘检查、网络连通性检查等能力封装为可控工具，由中心服务端根据 host_id 调度对应 Agent Client 获取实时运行上下文。
5. 设计安全边界与人工确认机制，P0 阶段不直接执行重启服务、清理磁盘、kill 进程、修改配置等高风险操作，仅输出排查步骤、命令建议和处理建议，并记录人工确认结果，避免模型误操作影响服务器环境。
6. 建设基础可观测与审计能力，记录服务器标识、告警输入、知识检索结果、工具调用参数、客户端返回结果、模型分析结果、人工确认结果和异常信息，支持问题回放、责任追踪和后续效果优化。
7. 完成 LoRA 轻量微调验证链路，基于服务器告警与处置样本构造 Alpaca 格式训练数据，支持数据上传、格式校验、训练任务创建、日志查看、Adapter 输出管理和基础模型 / 微调模型对比测试，用于验证告警分析输出格式和处置建议质量优化效果。

项目成果：

1. 完成多服务器智能运维与告警分析 P0 原型，覆盖客户端指标采集、告警接入、工具查询、知识检索、原因分析、处置建议、人工确认和审计记录等核心流程。
2. 通过 Agent Client + RAG + Tool Calling，将服务器实时运行状态、运维知识库和大模型分析能力串联起来，提升常见服务器告警问题的排查效率和处置标准化程度。
3. 通过人工确认和审计记录机制，保证高风险操作不由模型直接执行，提升智能运维场景下的可控性和可追溯性。
4. 通过 LoRA 微调验证，对比基础模型与微调模型在告警分类、原因归纳、排查步骤完整性和处理建议格式上的输出效果。

项目预览： 点击预览

技术栈： Python / FastAPI / Agent Client / RAG / Tool Calling / Qdrant / LangChain / OpenAI-Compatible API / Transformers / PEFT / LoRA / Qwen2.5 / Redis / SQLite / Docker Compose

工作经历

安徽俏美味食品有限公司 AI应用开发

2025.05 - 至今

- 负责公司内部 AI 应用能力建设，围绕知识问答、运营数据分析和流程辅助决策等场景，引入 LLM、RAG 与 Agent 技术，推动 AI 能力在业务系统中的工程化落地。
- 设计并推进企业级 RAG 知识库能力建设，覆盖文档解析、Chunk、Embedding、向量索引、检索召回、上下文组装与带引用回答，形成内部知识库问答闭环。
- 主导Agent 工具化能力设计，将知识库检索、订单查询、库存查询、售后分析等能力抽象为可调用工具，为运营、客服和异常排查场景提供智能助手能力。
- 主导高并发数据中控平台研发，基于 MQ + Redis 构建订单、售后、库存等多源数据异步处理链路，为 AI 问答和 Agent 工具调用提供结构化数据基础。
- 建设 AI 应用相关的元数据、调用日志和可观测能力，支持 RAG 检索链路追踪、问题回放和后续效果优化。

合肥谦博贸易有限公司 架构 / 全栈开发

2023.03 - 2025.05

- 基于 Spring Cloud Alibaba + Dubbo 3.0 构建跨境电商 ERP 微服务底座，集成 Nacos 实现配置与服务治理。
- 设计 Sa-Token 多租户鉴权体系，实现租户逻辑隔离、动态权限控制与组织架构授权。
- 负责 Shopee、TikTok 等平台 API 对接与高并发适配，处理订单、商品、库存、价格、售后等核心业务数据同步。
- 设计异步任务与数据补偿机制，结合 MQ、Redis、定时任务解决第三方平台接口限流、失败重试、数据延迟和状态不一致问题。
- 参与前后端一体化开发，基于 Vue3 + Element Plus 构建商品管理、订单处理、库存管理、异常处理等业务模块，提高运营处理效率。
- 基于微服务与多服务器运维场景，探索智能运维与告警分析能力，设计 Agent Client + RAG + Tool Calling 的 P0 原型，用于服务器指标查询、日志分析、故障 SOP 检索、处置建议生成和人工确认审计。

龙图腾网科技（合肥）股份有限公司 技术部主管 / Java

2014.08 - 2022.12

- 负责知识产权业务系统架构演进，设计 MySQL + Elasticsearch + MongoDB 多存储架构，支撑商标、专利等多类型数据管理与检索。
- 优化检索与数据处理链路，基于 Elasticsearch 构建高效检索能力，提升大规模业务数据查询、筛选和聚合性能。
- 主导核心业务模块重构与性能优化，沉淀通用数据处理、批量任务、权限管理和业务流程编排能力。
- 负责技术方案设计、任务拆解、代码评审和团队协作，推动系统从传统单体应用向更高可维护性、高扩展性的架构演进。

- **早期经历：北京中科汇联科技股份有限公司、上海智青科技有限公司**

技术栈

AI 应用	LLM 应用开发、RAG 知识库、Agent、Tool Calling、Function Calling、Text-to-SQL、上下文组装、引用溯源、RAG 评估
AI 工程	Python、FastAPI、LangChain、OpenAI SDK、Ollama、Embedding、Qdrant、Elasticsearch
工程基础	Java、微服务、Dubbo、Nacos、MySQL、PostgreSQL、Redis、MongoDB、MQ、XXL-Job、Vue3、RESTful API、Docker、Linux、Git

教育经历 / 证书

安庆师范大学 本科 信息管理与信息系统 2008 - 2012 **证书：**大学英语四级