

陈炫

18120929113 | chenxuanz0206@163.com | ZhuoShao26
21 years old · AI Application & Agent Developer · Open to opportunities
github.com/ZhuoS62 | 个人博客: zhuoshao.cn
厦门理工学院 (福建省重点建设高校、省 A 类一流应用型建设高校) | 软件工程 | 2027届



实习经历

福建电子口岸股份有限公司 | AI Agent 开发 | 2026.09 - 至今

AquaPilot 智慧渔业 Agent

- 背景: 面向水产养殖场景, 为养殖人员打造 AI 驱动决策辅助能力, 将分散的养殖信息转化为直观的管理建议, 辅助日常巡塘与投喂安排。
- 我的职责: 参与 Agent 核心能力与业务接口开发, 主要工作包括:
 - Skill & Tool Calling: 每个 Python Skill 封装独立能力, 通过 StructuredTool 暴露参数 Schema, Tool Catalog 按能力域组织工具; 模型通过 bind_tools 选工具, LangGraph 循环执行并回传结果; 执行层校验白名单、绑定确定的池塘 ID, 并通过轮次上限约束调用。
 - Context Engineering: 注入——将池塘信息、现场 Evidence 与工具结果组织为模型上下文, 保留来源、时间戳与质量状态; 筛选——按相关度过滤知识引用并限制检索上下文长度; 隔离——区分当前现场依据与历史回答, 避免将旧回答当作实时数据。
 - Memory: 历史消息持久化保存, 恢复会话时仅加载最近轮次; 运行态按会话 ID + 池塘 ID 隔离, 保留多轮对话上下文, 通过轮次裁剪、TTL 过期清理与容量限制控制记忆规模。
 - LLM Wiki: 构建「资料摘录 → LLM 生成 Markdown 草稿 → 人工编辑审核 → 发布 Wiki」知识整理链路, 保留适用条件、不确定性与资料冲突; 通过草稿版本与来源哈希校验拦截过期发布, 并为页面附加来源信息, 支持知识审核与追溯。

FJLabel AI 智能标注平台

- 背景: 针对图像数据集依赖人工逐框标注、重复工作量大的问题, 基于公司现有 FJLabel 标注工具引入 AI 自动化标注能力, 将原有人工标注流程升级为「AI 批量预标注 → 多模态二审 → 人工复核」, 减少人工绘框工作量, 辅助训练数据准备。
- 指标与成效: 上线后累计处理 9,000+ 张图像, 生成约 3 万个候选框; 标注精确率达 99.3%, 平均单张预标注处理耗时约 2.14 秒; 数据标注人员的处理耗时由 1–2 小时缩短至 5–10 分钟。
- Agentic Workflow: 参考样本构建 → T-Rex2 批量预标注 → 全图与候选框裁剪图组装 → 多模态 LLM 结构化二审 → 规则校验 (候选 ID / 置信度 / 删除理由) → 人工复核与结果导出; 保留低置信度候选框供人工复核, 整图全删时回退原标注, 预标注阶段支持失败重试与断点续跑

厦门游沧海科技有限公司 | AI Agent 开发 | 2026.05 - 2026.09

项目链接: xzhuo.shop | taxtool.top

Xbox 自动化充值系统 | Agentic Workflow 与人机协同

- 背景: 面向内部 Xbox 代充运营场景, 针对人工逐笔充值与对账耗时较长、登录验证与风控易导致流程中断的痛点, 构建基于 Agentic Workflow 的自动化充值系统, 实现正常订单自动处理、异常场景人工接管, 提升订单处理效率并降低运营工作量。
- 指标与成效: 上线后累计管理 874 个账号、处理 2,403 笔充值订单, 订单成功率达 94.38%, 单日处理峰值 133 笔; 单笔订单端到端耗时约 40 秒, 运营人员每日处理与对账耗时由 2–4 小时缩短至 15–30 分钟。
- 我的职责: 项目 Owner, 负责系统从 0 到 1 的架构设计与开发落地。
 - Agentic Workflow: 订单接入与分配 → 账号 / 地区 / 余额校验 → 条件路由 (复用登录态 / 重新登录) → Microsoft 会话恢复 → 卡密兑换 → 结果核验 → 订单状态与凭证回写。
 - Task Scheduling: 将充值请求交由 Dispatcher 排队调度, 以全局并发限制控制整体负载、账号级串行避免同一账号同时操作、任务去重防止重复执行; 通过全局 / 账号 / 代理三级登录冷却约束登录频率, 并在服务重启后恢复待执行队列。
 - Tool Orchestration: 基于 Playwright 将会话恢复、页面操作、卡密提交、结果解析与截图留证封装为 Tool 节点, 衔接业务校验与数据持久化, 由工作流统一编排调用。
 - Human-in-the-loop: 针对 MFA / 账号锁定 / 会话失效 / 地区不匹配 / 执行超时等异常, 保留任务上下文与截图凭证, 经人工验证或核对后复用原会话, 从中断节点继续执行; 记录处理状态与错误原因, 支撑对账与问题追溯。

财务报税提效 Skill | OCR 文档解析与报表生成

- 我的职责: 将财务报税流程 SOP 封装为可复用 Skill, 将重复性的工作简化为自动化流程, 落地「资料导入 → OCR 识别 → 规则校验 → 异常复核 → 报表导出」的标准化处理链路。
- 指标与效果: 投入使用后累计处理 431 条交易记录, 成功率达 97%; 税务人员每日耗时由平均 30 分钟缩短至 5 分钟内

福建新大陆时代科技有限公司 | Python 后端开发工程师 | 2025.02 - 2025.06

项目链接: kjzbot.com

快加账 (KJZBot) | 微信群自动记账 SaaS

- 背景: 面向外贸、跨境电商及门店商户的微信群记账场景, 针对群内账目依赖人工逐笔登记、汇总与核对的问题, 主导构建快加账 (KJZBot) 微信群自动记账 SaaS, 打通「自然语言 → 结构化账单」端到端自动转换链路, 显著降低账目整理与核对成本。
- 指标与成效: 实习结束后独立推进商业化运营, 累计服务 30+ 家持续付费客户, 接入 1,800+ 个记账群, 近 30 日生成 8 万+ 条账单; 用户每日记账与核对耗时由约 2–4 小时压缩至 5–10 分钟。
- 我的职责: 项目 Owner, 主导产品从 0 到 1 的设计与开发落地, 并且持续推进架构升级与功能迭代。
 - 1.0 (微信群自动记账): 「微信消息接入 → 指令解析 → 权限校验 → 账务处理 → 数据持久化 → 结果回复」; 实现群内记账、查询及管理。
 - 2.0 (AI 账单查询 + 消息自动化 + 多端协同 + 商业化交付):
 - AI 账单查询: 固定指令查询 → /ai 自然语言查询; 以当前群账单为查询范围, 支持通过自然语言获取账目信息。
 - 消息自动化: 「消息接入 → 记账 / 控制指令识别 → 关键词匹配 (精确 → 模糊) → 文本 / 图片回复」; 仅执行首条命中规则, 避免重复回复; 群发采用「目标选择 → 逐个发送 → 成功 / 失败统计」链路。
 - 多端协同: 本地账单管理 → 桌面端与 Web 后台协同; 按账号隔离数据, 通过后台任务双向同步分组、群组与账单, 检测两端修改冲突, 避免直接覆盖。
 - 商业化交付: 完善「账号授权 → 续费 → Windows 安装升级」交付流程; 针对微信连接异常与网络波动, 补充连接恢复、请求重试与离线授权宽限机制。

项目经历

 **ReAgent | 代码审查与安全修复 Agent Harness** | 2026.05 - 2026.09

项目链接: github.com/ZhuoS62/ReAgent

项目描述: 面向 GitHub PR 合并前的代码质量与安全治理, 主导构建基于可恢复 Agent Runtime 的多 Agent 代码审查与安全修复系统, 打通风险发现、证据复核、受控修复、回归验证与执行追踪全链路, 并通过失败案例回流、自动评测与版本门禁形成持续优化闭环。

技术栈: Multi-Agent、Agent Runtime Harness、Dynamic Skill、Redis Streams、PostgreSQL、SQLite、Docker、Prometheus。

- Agent Runtime Harness:** 统一管理任务状态、执行预算、节点重试、持久化 Checkpoint 与任务取消, 支撑中断后断点续跑, 并通过 Run Trace 完整记录执行过程。
- Multi-Agent Collaboration:** 设计 Lead / Worker / Critic 协作机制, 由 Lead 动态委派任务、评估结果并完成最终综合; 安全与正确性审查并行执行有界工具循环, Critic 独立质疑并复核代码证据, 支持最多一轮返工。
- Agent Self-Evolution:** 将误报、漏报、错误修复与执行异常沉淀为失败案例, 分别生成候选 Prompt 与 Skill 版本; 以安全性、Validation 提升及 Holdout 非退化门禁控制版本激活, 支持候选去重、版本追踪与回滚。
- Evaluation & Observability:** 围绕 Plan → Tool → Observe → Final 构建有界执行循环, 记录节点状态、Agent 消息、工具调用、重试与门禁结果, 支撑链路回放、失败归因与评测复现。

 **KJZ Support Aegnt | 可评测 RAG 与 LangGraph 故障咨询系统** | 2026.03 - 2026.05

项目链接: kjzbot.com/rag

项目概述: 面向 KJZ 微信群记账 SaaS 的售前咨询、产品答疑与基础故障排查场景, 针对产品资料分散、复杂问题召回不足、证据不足易误答三大痛点, 主导构建可评测、可拒答的 Agentic RAG 智能支持系统, 提升知识检索与问题解答的可靠性。

技术栈: LangChain、LangGraph、FastAPI、Milvus、Elasticsearch、BGE Embedding、Cross-Encoder、PEFT、ONNX Runtime、RAGAS

指标与成效: 基于 30 条评测样本完成 5 组检索配置消融实验; 冻结最优配置后, 在 10 条独立 Holdout 样本中的 7 条可回答样本上, Hit@3 / Recall@3 达 100%, MRR@5 为 0.9048, RAGAS Faithfulness 为 0.8852、Answer Relevancy 为 0.8934。

核心工作:

- Hybrid Retrieval:** 搭建 Query Rewrite / Decompose / HyDE 多策略查询扩展 → BM25 + Dense Retrieval 双路召回 → RRF 融合 → Cross-Encoder 重排的混合检索链路; 通过检索配置消融实验, 量化对比查询扩展、融合与重排策略的效果差异。
- Agentic Workflow:** 基于 LangGraph 构建 查询改写 → 检索 → 质量评估 → 条件路由 (生成 / 重试 / 拒答) 闭环; 相关度不足时改写 Query 并重试一轮, 仍不达标则拒答并提示转人工, 从机制上避免证据不足时直接生成答案。
- Embedding Optimization:** 集成基于 PEFT + PyTorch 的 LoRA / QLoRA 微调模块, 将人工标注的高频错召回 Case 纳入训练数据管理, 支持参数配置、训练控制、效果对比与模型管理; 引入 ONNX Runtime INT8 加速 Embedding 推理。
- Evaluation:** 以 Hit / Recall、MRR 衡量检索效果, 以 RAGAS Faithfulness、Answer Relevancy 评估回答质量; 分离配置选择集与独立 Holdout, 冻结配置后验证泛化表现。

获奖经历 (国家级)

第九届全国青年科普创新实验暨作品大赛	一等奖 (第一名)	2024 金砖国家职业技能大赛全国选拔赛暨国内决赛	三等奖
2023 年度全国职业技能大赛	一等奖	第八届全国互联网+创新创业大赛	铜奖
第五届中华职业教育创新创业大赛	金奖	2025 年世界职业院校技能大赛争夺赛	铜奖

专业技能

- Agent / LLM 工程:** 熟悉 Agent Runtime / Harness 架构及 RAG、Prompt、Context、Memory、Checkpoint 等核心机制, 了解 MCP、Skill; 具有工具调用、上下文管理与运行状态控制实践, 使用 Recall、MRR、RAGAS 评估检索与回答质量。
- 后端开发与前端:** 熟悉 FastAPI、Pydantic, 具有业务接口、结构化数据校验与异步任务开发经验; 能够实现并发限制、任务去重、幂等控制与异常恢复, 使用 Pytest 验证关键流程; 使用 Vue 3 完成页面开发与前后端联调。
- 数据库与基础设施:** 具有 MySQL、SQLite、Redis Streams、Milvus 项目实践, 能够结合业务需求实现数据持久化、任务流转及检索服务; 使用 Docker 完成服务容器化部署。
- 工程实践与 AI 辅助开发:** 使用 Git 管理版本, 结合源码、日志与测试定位问题; 熟练使用 Codex、Pi 等 AI Coding 工具辅助代码分析、方案设计与开发迭代。