



27 届硕士 / 2002.08 / 18079143012 / m18172744982@163.com /

欢迎访问我的线上简历: <http://64.90.11.23/>

教育背景

上海第二工业大学	计算机技术（硕士）	2024.09 - 2027.6
江西理工大学	软件工程（本科）	2020.09 - 2024.6

荣誉/证书：学业奖学金二等奖、全国大学生创新创业大赛省级铜奖、国家励志奖学金、CET-4、华为认证 HCIP-HarmonyOS Application Developer（中级）、德国莱茵 TÜV 工业 4.0 智能制造工程师认证

实习经历

2026.03-2026.08 上海思创网络（专精特新） 太保数字分身销售-Agent 智能体

- 参与太保业务的数字分身外呼项目的主要开发工作，基于 LangGraph 设计多轮对话流程编排框架，支持外呼场景下复杂对话状态流转与分支处理。
- 项目通过阿里云外呼服务实现实时电话外呼，期间完成前期外呼产品调研、端到端链路联调与验证、部分链路问题修复，推动客户电话-阿里云语音文字双向转化-项目抽取数据库话术+LLM 调用的完整闭环。
- 针对项目多个核心瓶颈进行持续迭代完善，包括对话延迟、AI 坐席拟人度以及话术专业性方面。对话延迟降低超过 60%，拟人度和话术专业度已支持真实业务场景。
- 进行意图识别节点的模型微调工作，主要包括模型调研-数据清洗-模型微调及部分后训练任务。

2025.06-2025.11 上海吨吨信息技术有限公司 端边智能协同调度系统-AI 研发

- 负责协同执行链路的开发与联调，覆盖任务配置、节点分组、调度下发、结果回传以及保障异常恢复功能。
- 实现异构终端接入抽象层，统一节点注册、能力上报与心跳保活，屏蔽终端算力与网络差异，新增终端类型无需改动调度侧逻辑。
- 参与分层调度与节点切换算法设计和优化：按节点历史耗时、网络状态、任务进度打分选节点，动态迁移滞后任务；相较静态调度方案，固定时间窗内有效任务吞吐提升 30%。
- 针对强弱节点算力差异导致的等待空转，实现按算力分级的任务配额下发；配合指标采集与多组配置对比验证，100+ 高异构边缘/IoT 场景下节点资源利用率提升 40%。

项目经历

太保数字分身销售 2026.04-2026.09

项目介绍：车险续保预热需对大量存量客户完成探需、加微与预约跟进，人工坐席难以兼顾触达规模与话术一致性。本项目复刻面向车险续保预热场景的销售数字分身系统，以阶段化对话决策与结构化业务知识驱动数字分身自主规划沟通目标，在真实电话与实时语音中完成需求探询、异议处理与加微预约，更高效率的同时达到接近真人坐席的业务专业度。

技术栈：Python、MySQL、LangGraph、Vue3、阿里云 OutboundBot、LiveKit、Docker、RAGAS

- 基于 LangGraph 设计并落地分阶段多轮对话引擎，将模型意图识别与确定性流程控制解耦，通过阶段路由、异议分支与跨轮状态管理保障销售流程稳定推进，提升对话的可控性与稳定性。
- 针对对话高延迟瓶颈，构建多层性能优化机制包括规则触发意图请求、并行 LLM 执行、结合流式输出及模型微调，对话延迟降低超 60%，单轮对话平均延迟在 2s 内。
- 针对话术泛化专业度不足的问题，结合结构化知识库与提示词动态约束回复，并构建“批量对话-测试评估-定位修复-测试验证”的话术优化闭环，迭代完善后对话问题率下降 90%。
- 设计语音通道适配层，允许接入阿里云 OutboundBot、本地语音部署 LiveKit 与火山 RTC，复用同一套对话引擎打通 Agent 决策、语音交互与客户触达的端到端链路。

- 搭建面向运营的 Vue 3 可视化管理端, 基于 Vue Flow 实现对话图编排, 将知识策略、提示词与外呼任务纳入统一配置, 并支持运行监控与会话复盘, 形成配置、执行、触达和优化的运营闭环。

基于字节跳动开源 DeerFlow 的自演化机制与法律垂域增强

2026.1-2026.9

项目介绍: 在字节跳动开源 DeerFlow 基础上深度改造的 super agent harness, 独立实现受控自演化与法律垂域增强: 日常使用中自主识别能力缺口并制作新 skill / tool 增长自身能力, 涉及法律通过 GraphRAG 双引擎检索与证据校验, 输出更专业和可追溯回答, 增强原 DeerFlow 作为通用能力的短板。

技术栈: Python、LangGraph、MCP、GraphRAG、Neo4j、Milvus、LangSmith、RAGAS、Docker

Github 链接: <https://github.com/shunbao-p/Deer-flow-s>

- 设计 skill / tool 自演化链路, 支持 Agent 自主区分 skill gap 与 tool gap, 分别经创建决策-共享安装内核-制作测试-注册复用-生命周期治理组成完整链路闭环。
- 针对运行时自主创建易导致能力重复与能力库污染的问题, 设计 skill 生命周期治理: 以创建前查重与相似能力复用避免重复沉淀, 停用从 prompt 约束下沉为文件级访问拦截, 使能力边界不被绕过。
- 设计法律法规文本的入库、切分与图谱建模链路, 构建 5w+chunk 向量库与 2.3w+ 条文图谱, 并基于 Neo4j + Milvus 搭建双引擎检索架构。
- 针对条文召回不完整、召回结果问题, 设计查询路由与混合召回并支持图检索失败 fallback 回退; 引入 Cross-Encoder Rerank、证据门控与 claim 级校验, 证据不足时降级表述以保证回答专业性和可信度。
- pytest 覆盖自演化决策、安装治理与检索链路(自演化相关 100+ cases), 并基于 LangSmith 实现 tracing 与链路分析; 固定集离线回归显示优化后法规命中率由 22.9% 提升至 60%、平均时延由 72.5s 降至 8.3s, Faithfulness / Answer Correctness 超过 85% / 80%。



技能说明

技术栈

Python、LangGraph、React/Vue、FastAPI、MySQL/Neo4j/Milvus、Redis、PyTorch、Docker、Linux

Agent 智能体开发

能够基于 LangGraph 设计有状态 Agent 工作流, 解决复杂任务编排、多轮上下文管理、工具/MCP 设计与治理, 以及流程控制、状态持久化、断点恢复等长流程执行问题。

LLM 应用设计

具备多模型调用链路设计能力、结构化输出及模型调用封装, 能够围绕输出准确性、上下文预算及缓存策略、响应延迟与调用成本优化应用链路。

RAG 与知识工程

能够独立搭建 RAG/GraphRAG 系统, 覆盖数据处理、索引构建、混合召回、重排、答案校验与效果评测; 具备检索和生成问题的分析、定位与迭代能力。

模型与数据处理

具备 NLP/LLM 模型微调与评测能力, 可完成训练数据清洗、指令样本构建、SFT 及 LoRA/QLoRA 参数微调、实验评估与错误分析; 熟悉 PyTorch、HuggingFace、Pandas、NumPy。

AI 工程化与工具

熟练使用 Python、FastAPI, 具备接口设计、任务调度、状态管理等控制、日志追踪、自动化测试及容器化部署能力; 熟悉 MySQL、Neo4j、Milvus、Docker、Linux、Git。