

侯宇

📞 18321736710 · ✉ hy572801400@163.com

🎓 教育经历

- 硕士，上海大学，信号与信息处理，2013.9—2016.4
- 学士，天津职业技术师范大学，电子信息工程，2009.9—2013.7

🔧 专业技能

- 技术全面,基础扎实,具备多年AI行业经验;具备丰富的系统架构设计经验;有一定的团队管理经验
- 熟悉C++、Python, 对常用中间件缓存、存储、消息队列有一定研究, 阅读过相关源代码
- 熟悉常见的机器学习及深度学习方法在NLP及语音中的应用, 擅长NLP、对话系统及向量检索的算法架构设计及系统调优
- 熟悉AI全流程生命周期, 熟悉在CPU/GPU等硬件平台上的训练推理加速的常见方法.

🏢 工作经历

- **循环智能，算法平台，技术负责人，2021.04~至今**
 - 业务侧负责组建技术团队组建，日常管理规模10人左右. 建立标准化的开发测试流程和规范，组织测试自动化
 - 技术侧负责对话机器人及AI平台(数据标注、自动化训练、模型仓库及部署、推理加速、资源调度)架构设计及开发管理工作
- **追一科技，业务架构组，高级技术经理，2018.08~2021.04**
 - 产品整体架构升级与优化，架构设计与核心代码编写, 丰富完善公司产品.
 - KA客户需求调研，产品交付评估及统筹, 制定并完善标准产品交付流程.
 - 负责伏羲NLP平台架构设计及核心代码开发
- **华为，麒麟软件，软件工程师，2016.04~2018.07**
 - 负责Tee及BootLoader侧安全启动、生物特征安全、硬件安全、加解密方案设计与研发
 - 负责深度学习在安全业务上的应用预研

🏠 项目经历

- **AI平台 * 循环智能**
 - 项目介绍: 打造可复用、持续演进的AI平台能力，提升AI生产效率、降低生产成本
 - 项目职责: 负责人
 - 主要成绩:
 - AI平台的产品功能设计及分析、技术路线设计及架构设计
 - 从0-1构建模型(语音、NLP)自动化训练，模型管理及部署自动化.
 - 成功落地NLP大模型自动化训练及线上推理加速，性能提高约15%, 资源要求降低30%.
 - 打通了模型从数据到训练、部署推理的全流程，使得模型生产和上线对人力依赖降低50%.
- **伏羲NLP平台 * 追一科技**
 - 项目介绍: 以NLP为基础能力，打造NLP中台能力
 - 项目职责: 技术负责人

- 主要成绩:
 - 伏羲NLP平台产品功能需求分析设计、技术选型及技术架构抽象化;非功能需求统筹规划
 - 伏羲NLP平台的在线部分、离线部分、算法部分的架构设计
 - 其他架构设计及实现: 文本召回、模型训练、模型推理、A/B Test、语义理解、聚类分析、数据一致性、服务发现、API网关、权限等方案设计
 - 平台上线后, 模型生产成本降低50%, 模型更新速率提高100%.
- **智能客服 * 追一科技**
 - 项目介绍: 公司营收拳头产品, 数据闭环并不断迭代的智能客服
 - 担任角色: 产品架构师
 - 主要成绩:
 - 针对多客户的不同接入需求、流控需求进行抽象, 形成了支持Restful、Web Service等多种方式的通用接入层解决方案
 - 通用接入层架构设计与实现、提供可供二次开发的核心框架
 - 离线数据发布、离线模型训练高可用方案设计及核心代码实现
 - web后端数据库优化、对话系统服务端整体性能优化, 性能提升4倍
 - 各行业客户平均人力成本降低30%, 效率提升30%+
- **向量检索后台 * 追一科技**
 - 项目介绍: 提供通用向量化召回技术解决方案
 - 担任角色: 技术负责人
 - 主要成绩:
 - 负责高性能、高可用向量检索架构设计, 核心代码实现
 - 支持并发增删改查, 单机K级别的查询与写入功能
 - 其他架构设计: 向量索引发布、索引一致性等方案设计
- **恶意应用识别---华为**
 - 项目介绍: 利用深度学习技术和终端NPU构建恶意应用识别能力
 - 项目职责: 项目负责人
 - 项目业绩:
 - 利用sklearn中的SVM算法训练baseline的模型,探究方案可行性
 - 利用TextCNN构建文本分类模型,进行模型训练与优化,验证集检出率90%,误报率为8%;测试集检出率85%,误报率为20%
 - 利用终端NPU能力,开发端侧demo模型,展示深度学习成果
- **人脸性能优化项目**
 - 项目描述: 当前人脸解锁整体性能时间约为500ms,不符合性能需求,需提高人脸性能
 - 项目职责: 项目负责人
 - 项目业绩:
 - 人脸镜像校验流程优化方案设计,代码开发及测试
 - 人脸镜像校验方案优化前,Tee侧加载校验时间约为125ms
 - 采取时间换空间优化方案后,Tee侧加载时间约为25ms,优化时间约为100ms
 - 方案缺陷:预留人脸镜像大小的内存空间;启动时间增加约为70~80ms.