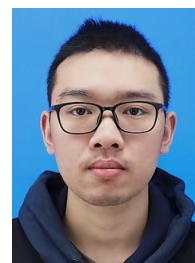


王锡宽

手机：182-1556-3618 · 邮箱：51275902053@stu.ecnu.edu.cn

现居城市：成都 · 实习时长：6 个月以上（随时入职）



教育背景

华东师范大学 985, 软件工程, 学术型硕士 2024.09 - 2027.06

• Rank: 15/60 GPA: 3.44/4.0

华东师范大学 985, 软件工程, 本科 2020.09 - 2024.06

• Rank: 36/184 GPA: 3.51/4.0 均分: 87.75/100 4次校级奖学金 CET4: 507 CET6: 523

科研经历

STAR：基于捷径学习测试的大模型公平性增强框架 2025.09 - 2026.01

研究背景: 深度模型在训练中往往依赖“捷径特征” (shortcut), 即利用敏感属性或与任务弱相关的表面规律进行预测, 这会导致模型学习到虚假相关性, 对于敏感属性的错误学习进而导致群体间不公平。本项目提出 STAR 框架, 通过捷径强度建模与自适应重加权机制减少模型对捷径的依赖。

研究内容:

1) 构建捷径学习测试模块, 支持两种设置:

• **已知敏感属性:** 使用多任务学习的方法, 在敏感属性分支使用梯度缩放因子 g 控制其编码强度, 用以评估模型对敏感信息的依赖;

• **未知敏感属性:** 在模型各中间层加入辅助分类器, 该分类器用于预测主分类器是否能够成功分类, 通过层间预测差异计算并汇总得到捷径强度 w ;

同时引入 **VAE 稳定器** 以约束潜在表示, 减少检测过程中梯度缩放导致的表示漂移带来的噪声, 使捷径强度 w 更稳定可靠。

2) 设计 **自适应实例加权**: 依据捷径强度 w 动态调整训练样本权重, 使模型在训练时自动弱化对捷径特征的依赖, 提升不同群体间预测一致性。

成果: 在 Sentiment、Dutch 任务上公平性与性能的权衡指标 UF-score 分别提升 68% 与 47%, 显著优于 DFL、INLP 等 baseline; 在无敏感属性信息时, 本框架得到的模型的 UF-score 同样优于 BLIND、MABR 等无敏感属性 baseline。模型准确率保持稳定 (约 0.82), Independence / Separation / Sufficiency 等公平性指标均有改善。引入 VAE 后潜在空间偏差显著降低 (如 TPR_rms 自 0.152 降至 0.133), 验证了捷径检测与重加权框架的稳定性。

项目经历

智能家居场景下的电商客服 Agent 系统 2026.03 - 2026.05

项目技术: AgenticRAG Multi-Agent GraphRAG LangChain/LangGraph ReAct

针对基于 FAQ 的传统客服系统长尾问题命中率极低、人工介入成本高的问题和常规 RAG 多跳推理能力差、幻觉严重以及结构化/非结构化数据割裂的痛点设计并实现了基于 GraphRAG 的多智能体电商客服系统。

多智能体架构设计: 基于 LangGraph 搭建核心状态机, 解耦意图识别、规划器与执行器。引入 Map-Reduce 机制, 针对复杂子任务实现图数据库与文本检索的 Send 并发调用, 将单条复杂询问的平均全链路延迟由 4.8s 压降至 2.1s。

高精度意图识别与路由: 采用 Prompt-template + DeepSeek v3 构建意图路由节点。通过上线日志进行难负样本挖掘, 结合 Few-Shot 机制进行 Prompt 迭代, 将 JSON 格式化输出的意图分类 F1 分数由 0.82 稳步提升至 0.93。

异构数据双轨检索引擎: 设计安全的 Text2Cypher 流水线。提炼 100+ 高频预定义 Cypher 模板进行极速响应; 对于动态查询, 构建包含 EXPLAIN 语法校验与 LIMIT 1 实体探针的防御链, 阻断大模型生成的幻觉数据库执行。接入 GraphRAG 处理复杂售后政策与长尾问题, 通过动态局部/全局社区检索解决跨域多跳推理难题。

兜底与护栏机制: 在任务分解层与工具执行层双向引入 Fallback 逻辑, 系统幻觉率压降至 0.8%, Cypher 查询逻辑正确率稳定在 88%。

相关技能

- 熟悉 MySQL 存储引擎、索引、事务和隔离, 了解 MySQL 主从复制、读写分离、分库分表的使用场景。
- 深入理解并紧跟大模型前沿生态。熟练运用 LangChain / LangGraph 框架主导多智能体协同系统的全链路开发; 深刻理解 RAG 架构演进与 MCP 工具链底层协议; 深入掌握 ReAct、Plan-and-execute 等前沿 Agent 规划与推理范式的原理及工程实现。
- 常用开发语言为 C++、Python, 熟悉数据结构、并发编程及内存管理机制, 具备快速学习和适应新编程语言的能力。