

张学强

3年经验

13966093597 | 13966093597@163.com



教育背景

安徽医科大学 - 本科

2018-09 ~ 2022-06

工作经验

eBay - 算法开发工程师

2024-03 ~ 至今

- 参与 Hub GPT 的项目的文本分块、增强语义、微调模型等核心步骤的开发。
- 为 Chat2SQL 项目的处理流程提出合理化的解决思路，并负责实施项目中 SQL Agent 的 Json2SQL 生成方案的落地。
- Hub GPT 和 Chat2SQL 项目的运维和更新。
- 对原有图数据库的更新，进行算法方面的升级。

eBay - 大数据开发工程师

2022-08 ~ 2024-03

- 负责 PG2.0 PE2.0 Trust 项目的数据ETL和数仓建设。
- 设计并实施 OGG task check，解决数据漂移问题和数据质量的筛查。
- 数仓的日常的维护和管理。
- sparkSQL的优化等。

技能特长

AI应用开发与建设：

- 主导完成从数据预处理、数据分词、数据入库、数据召回重排等RAG项目全流程开发。
- 参与ChatBI项目工程的重要部分（数据提取、生成SQL）。
- 成功将自然语言转化为精确的BI分析或SQL查询，让业务人员通过对话的方式直接获取数据洞察。
- 通过实际的框架和业务流程创建多维的评估体系、确保模型性能与业务需求对齐。

大数据数仓：

- 精通Spark分布式计算引擎，具备TB级数据的处理、性能调优及复杂ETL流程开发经验。
- 熟练掌握Hadoop生态（HDFS、MapReduce、Hive、HBase）。
- 熟悉数仓建模理论。
- 有企业级数仓规划与建设的经验。

项目经验

Hub GPT

项目背景：

企业内部的知识分散在各个系统各个格式和各个位置中，希望有一个东西可以打破这种知识孤岛。公共的大模型虽然强大，但不能回答涉及企业内部的机密、流程等。直接将数据传给外部的服务商存在巨大的数据泄露的风险。在某些岗位，员工需要频繁的查询大量的SOP来回答问题，重复劳动高。

技术工具：

Markerpdf、Hagging Face、Sentance transformer、Milvus、Rerank

架构流程：

- 预训练数据和用户Query增强。
- 对不同的文本类别进行分词、去重。

- 对embedding模型和rerank模型做微调。
- 文本块embedding成向量后入Milvus数据库。
- 用户Query+召回数据片段通过rerank做重排。
- 大模型通过提示词做问题总结输出。

效果：

1. 打破知识孤岛，建立一个统一的、智能化的知识入口，让员工能通过自然语言提问，快速获得精准答案。
2. 保证公司数据安全的同时将模型回答内容限制在公司内部知识范围内，避免幻觉。
3. 降低员工的培训成本，快速的召回相关的知识，发生相关的事件时可以快速定位到相关的SOP，快速的处理解决。
4. 召回的问题会来源于最新的知识库，避免了因员工查找的版本不同导致问题的错误处理。

Chat2SQL

项目背景：

基于公司现有的复杂的架构，数据部门无法满足业务部门的快速准确的查询数据的需求。尤其是对于跨域问题的数据查询，通常会涉及到多个部门，但各个部门人员的沟通成本和协作成本很高。导致数仓开发人员会花费大量的时间用在给用户沟通和返回数据结果上。减少开发人员开发其他指标的时间。而业务人员也会因为数仓取数时间长的问题导致任务进程缓慢。

技术工具：

LangGraph、RAG、MySQL、SQLGlott、Hugging Face

架构流程：

1. 基于LLM来识别用户的Query的询问环境、实体类型、各种过滤条件等。把相应的输出放置在json中，并对缺失的必选项进行多轮对话补充。
2. 通过对询问环境和实体的拼接的方式，到RAG中召回相应的表和列字段。并对多个召回进行条件匹配选择。
3. 把补充完整的json送入给大模型生成SQL，转换SQL类型。
4. 提取SQL执行代价，并对代价进行评估。
5. 执行指定的SQL，并返回给平台，通过前端页面来更改输出类型。

效果：

1. 参与并设计了一套自然语言转化为SQL的自动化流程，解决了业务部门取数慢、数据团队沟通成本高的问题。
2. 采用多轮对话补全的缺失的查询条件，并利用RAG从知识库中精准的召回相关的表和字段，将SQL生成准确率提升至94%。
3. 开发SQL执行代价评估模块，有效的拦截了高风险的查询，保障数据库的稳定性。
4. 使业务人员自助获取数据的时间从数天缩短至5分钟内，将数据团队的人工辅助查询降低到2%，显著提升分析效率。

NuGraph(引入GPLinker提升效率和精度)

项目描述：

针对eBay海量商品信息存在大量非结构化、半结构化文本，导致商品属性不全、搜索相关性差等核心业务挑战，需要构建一套自动化的商品知识图谱生产管线。通过设计与训练先进的信息抽取模型，从商品标题、描述、规格等文本中精准抽取关键三元组知识，以解决传统方案处理复杂文本效果差、人工标注成本高、知识更新滞后的瓶颈。

技术架构：

GPLinker、GlobalPointer、Neo4j、Transformers、PyTorch、FastAPI

设计方案：

- 为突破传统流水线方案（Pipeline NER + RE）的性能天花板，主导将核心模型升级为GPLinker。利用其联合抽取实体与关系的优势，高效解决电商文本中出现的实体嵌套问题。
- 针对项目冷启动阶段高质量标注数据稀缺的困境，创新的采用了“GlobalPointer模型预标注 + 类目运营专家审核”的人机协同模式。
- 完成GPLinker模型全流程训练、微调及性能优化，并针对电商场景建立一套覆盖实体、属性、关系的精细化评测体系。
- 为确保注入Neo4j知识图谱的数据质量，在自动化入库流程中设计并集成了“置信度阈值筛选 + 关键类目人工抽检”的双重校验机制。

效果：

- 模型抽取F1值达到92%，关系抽取F1值达到88%。

- 每日可稳定处理千万级商品文本，成功将商品知识的更新周期从“周”级缩短至小时级。
- 为下游精准搜索、个性化推荐、智能客服等多个下游业务提供了持续、高质量的结构化知识源泉，有效提升用户体验和商业化效率。

PG2.0 PE2.0 Trust

项目背景：

payment graph 和 payment executor域从1.0升级到2.0。更换抽表的策略，把原有的一些字段放置到新的字段中，减少一些非必要表的抽取。可以加速落入数仓的速度。payment graph 来做支付路由的判断，payment executor 来做支付路由的执行。trust是做卖家的等级评分，和卖家各个指标的分析。

技术工具：

spark

架构流程：

- 把数据导入ods后，然后把ods的数据经过清洗过滤，并对加密的json字段解密后，对大的json逐层拆解，提取需要的字段保留聚合。
- pg 和 pe 是支付流程的判断路由的步骤。后续会和 fast 支付记账的数据进行数据比对。
- trust 域是先做了买家坏的体验的一张宽表，再根据这张宽表来做的各个维度的指标分析。

效果：

1. pg 和 pe 2.0 成功开发并平稳的运行，完成关键环节的对账。
2. trust 开发成功后，给商家提供了可视化的商家评级图，让商家对自己所在的等级有大致的认识，有利于更好的服务买家。

OGG check task

项目背景：

因为，业务数据库同步到从表的过程中有延时，而且从表可能也不是抽取的节点，可能需要从从表的从表来拉数据。这样导致数据链路过长，会出现数据漂移问题。导致数仓拉取数据的时间不能保证数据完全的同步到从表而导致数据缺失的问题。也可以预防上游任务出问题delay后，导致的抽数不足的现象。

架构流程：

- 这里采用Oracle自带的DAG和OGG来对数据同步是否delay做预测，并对数据的数据量进行统计。把统计的数值存储起来。
- 对主表和从表做数据量的对比判断，并存储起来。
- 定时调度任务，并统计数仓中各个表的数据。
- 对比历史中未同步好的表，是否已经同步。并对同步好的数据和未同步的数据进行标注。
- 未同步好的数据表格，发送邮件或电话进行提示。

效果：

- 可以在上游数仓同步好数据后再进行抽取任务。解决数据漂移问题。
- 每日对比数仓中的数据，可以及时的为数仓中的数据缺失问题进行及时的探查。
- 对一些未能及时修补的表，进行相关数据的留存，并在每日邮件中查看修补情况。

自我评价

1. 从大数据在组内流畅的转到大模型应用开发，学习能力强，爱思考问题。
2. 面对Chat2SQL中多表字段召回问题，创新性的提出了用单表存储表与表之间多跳的关系，对面对未知的问题有自己的思考。
3. 提倡使用先转通用AST树再转回指定SQL的形式，减少模型对单一SQL的专项训练，善于沟通，对于自己所探索的新的想法会及时的和组内人员交流。