智谱·AI marsoft | 安硕信息 北京交通大学 BEIJING JIAOTONG UNIVERSITY ModelScope 魔搭社区 阿里云

SPM 2023

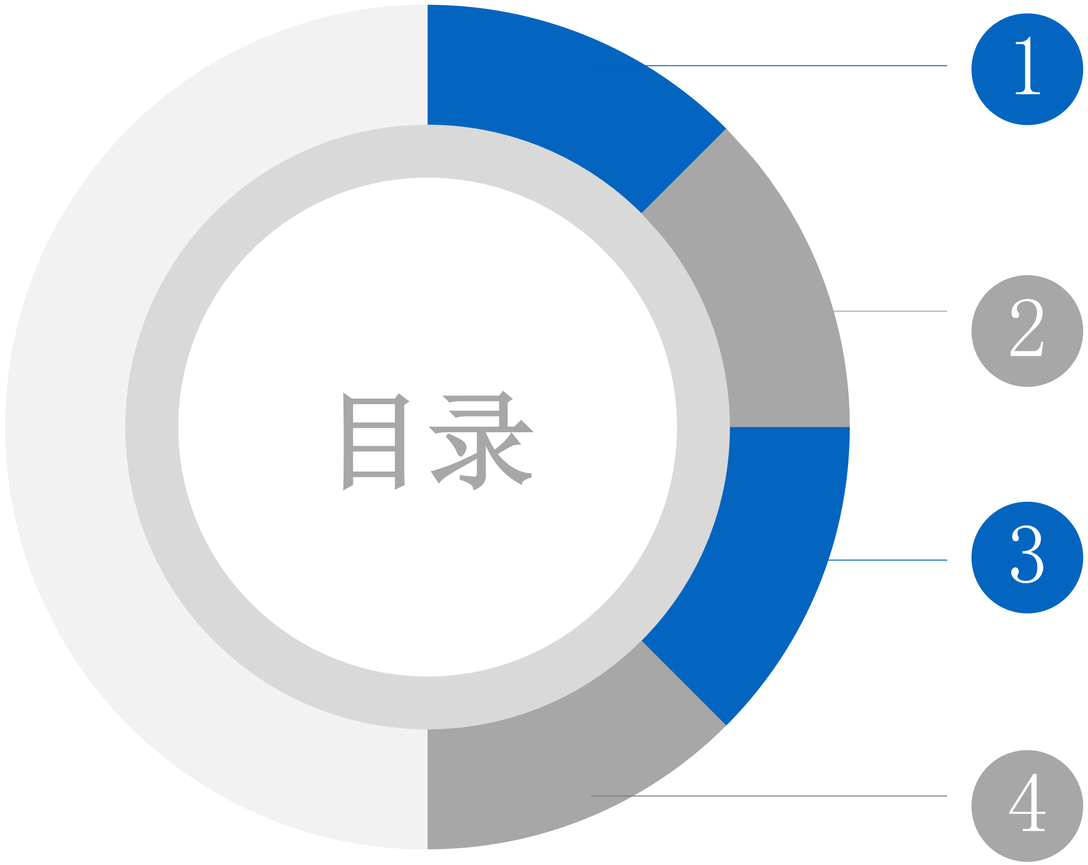
ChatGLM 金融大模型挑战赛

张昌彪（张真人）

队伍：流宝真人

项目答辩说明

赛道	创意应用赛道
赛题	SMP 2023 ChatGLM金融大模型挑战赛
项目进展程度	进展中



1

项目简介

队伍介绍、项目架构

2

方案介绍

意图识别、数据处理和配置、结果处理

3

比赛优化

意图识别不准确、数据校对、SQL处理

4

产品化总结

优缺点和调整、架构升级

项目简介 - 队伍介绍

队伍名称：流宝真人

所属公司：杭州大道一以科技有限公司（简称：道以科技）

参赛目的：学习交流为主，争取能在产品业务上有收获

成员介绍：

1. 张真人：张昌彪，道以科技CTO，负责数据处理、项目架构和开发

道以科技

项目简介 – 项目架构

流程说明

01 年报处理

1）通过PDF2TXT得到的文档，解析为块数据，提取相关表数据

2）PDF2TXT优化：第一列字段缺失、行数据粘连、分页处理等

02 数据嵌入和落库

1）文本拆分（Text Splitter），解析标题、文本内容、表内容之间的关系，初步拆分为各个段落或者表数据

2）文本分块（Text Chunks），缓存块数据，标题，大段落分块，嵌入FAISS

3）表数据落库，写入RDS

03 问题意图识别

1）问句要素提取并确认，LLM提取，向量确认

2）问句句型分类，关键字替换，向量确认

3）确认文档，如果无相关文档就给无法回答的结果

04 数据检索

1）从RDS直接查询非统计题的数据

2）通过LLM生成SQL，从RDS查询统计数据

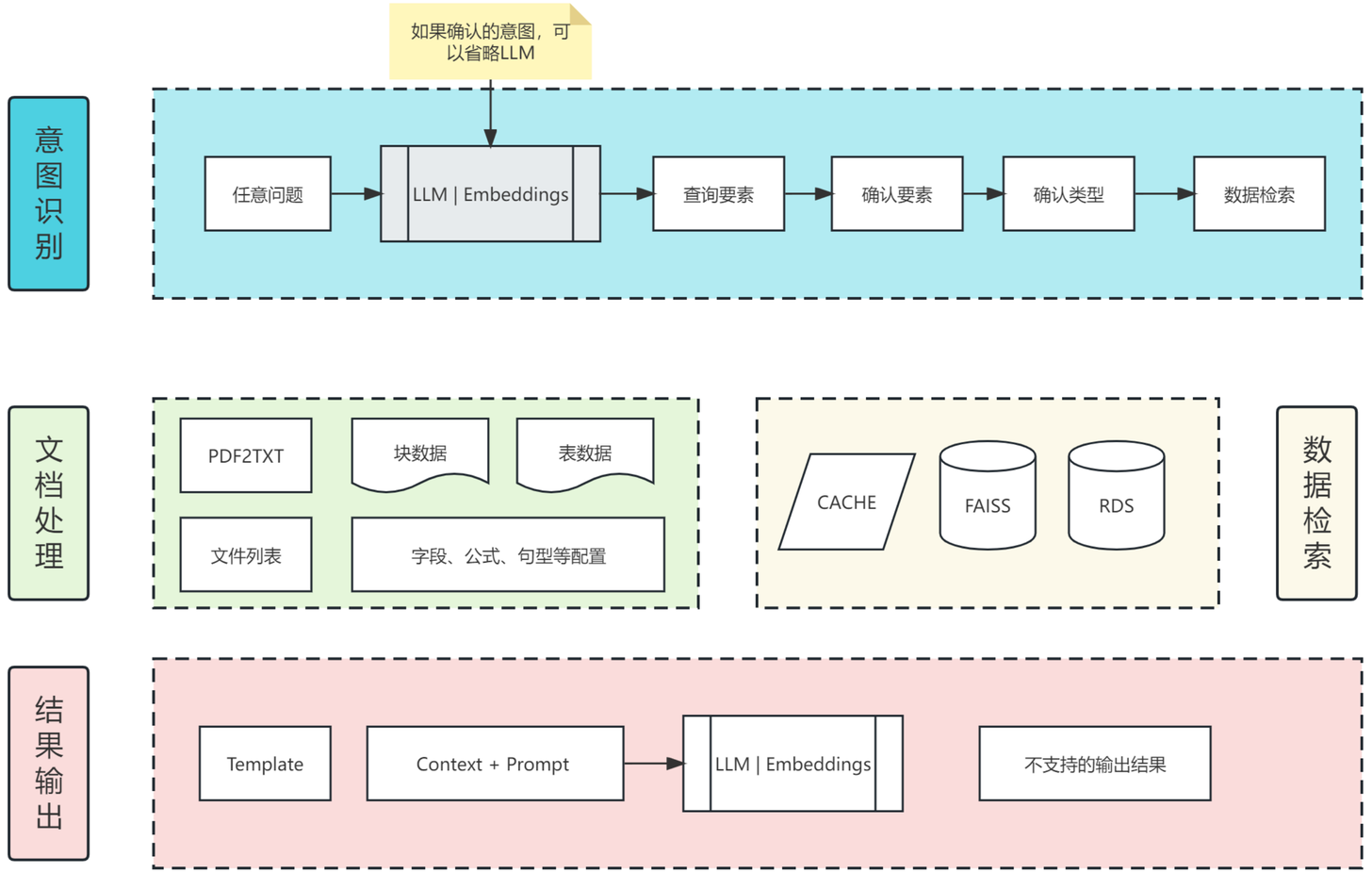
3）通过FAISS搜索上下文

05 结果输出

1）有确认结果的，通过模板输出结果

2）只有上下文的，通过组装Prompt，经LLM输出结果

3）无法回答的统一给出无法回答的结果

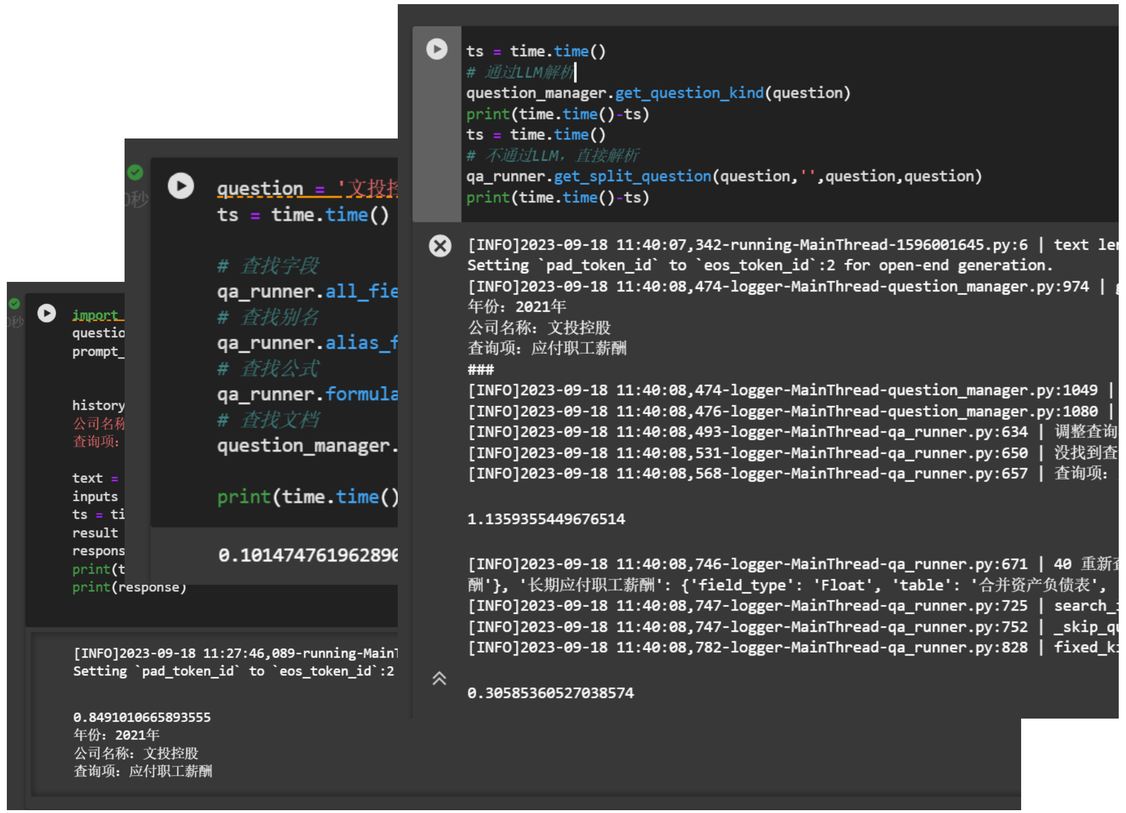


方案介绍 - 意图识别

意图识别要点：

1. 确认年报：公司名称（可以没有）、年份（可以多个）
2. 确认查询项：标准字段、不标准字段、标准公式、不标准公式、非字段
3. 确认查询范围：公司限制范围（注册地、历史注册地、办公地等）、排序、数量

对比点	通过LLM意图识别	通过向量搜索意图识别
耗时	≈1.136秒	≈0.306秒
兼容性	优化Prompt兼容各类问题	不兼容，只能针对年报
准确度	比较高	一般
扩展性	可替换LLM、API等	扩大向量库
公司名称	比较精确	一般，统计题无法确认
年份	可以解析精确范围	需要正则或算法定位（上一年）
查询项	比较精确	一般
查询范围	比较精确	无法解析



方案介绍 – 数据处理

标题处理（图）

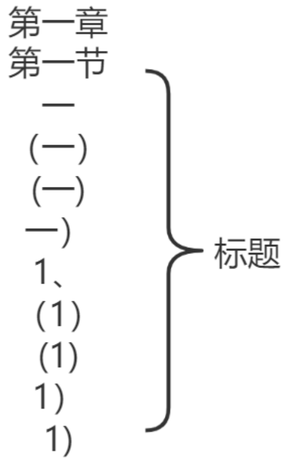
块处理（图）

向量化

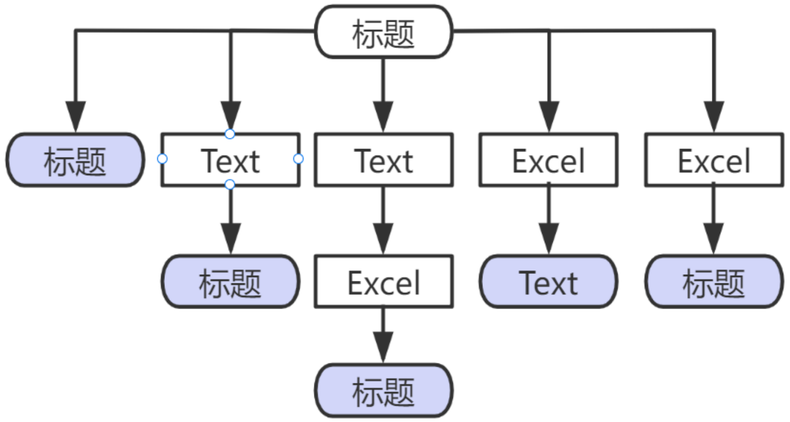
- 要素：表字段、别名、公式、公司名称等
- 上下文：标题、表数据转为文本后分块

落库RDS

- PDF文件表：文件名、公司名称、股票代码、股票名称、年份
- 公司信息表：注册地址、电子邮箱、网址、办公地、英文名称等
- 公司年报数据表：三大表
- 公司员工情况表：本科、硕士、研发、销售等人数



标题处理



块处理

方案介绍 - PDF2TXT工具优化

1. 解决处理**非封闭表格数据**使用原有lines方式获取表格缺失数据问题？ （兼容考虑第一行和第一列可能是合并单元格的情况）

   通过page.rects分析表格线条是否完整封闭，如果**非封闭的表格**，使用明确的水平线和垂直线获取表格数据；如果是**封闭的表格**使用原有的line的方式获取表格数据

2. 解决虽然是**封闭的表格**，但是使用lines的方式获取表格没有被正常提取的情况，缺失部分列的情况？

   使用原有的方式获取到表格后，对表格的宽带进行判断，如果**小于页面宽度的60%**，会基于explicit指定水平线的方式进行指定位置之下的表格进行获取，获取后判断是否替换当前的表格

3. 对**文本数据**进行优化处理（主要用于解决完全无框的数据或其他表格行数据粘连的情况）

   对现有的文本进行处理，主要处理思想为对**文本间距超过10个像素**的同行数据追加时增加空格；对于**换行数据**，判断是否为段落，符合段落条件的合并为一个一行数据

4. 对**行列数据**的进一步处理（如判断数据合并，删除空行空列，判断删除错位的空列等）
5. 对**页眉数据**的进一步优化，增加判断是否存在页眉线，如果存在页眉线以上数据作为页眉处理

方案介绍 - 数据配置

- 中文字段
- 中文字段别名
- 中文公式，eval计算
- 替换关键字的题型

```
"合并现金流量表": {
    "销售商品、提供劳务收到的现金": ["Float", 2, "元"],
    "收到的税费返还": ["Float", 2, "元"],
    "收到其他与经营活动有关的现金": ["Float", 2, "元"],
    "经营活动现金流入小计": ["Float", 2, "元"],
    "购买商品、接受劳务支付的现金": ["Float", 2, "元"],
    "支付给职工以及为职工支付的现金": ["Float", 2, "元"],
    "支付的各项税费": ["Float", 2, "元"],
    "支付其他与经营活动有关的现金": ["Float", 2, "元"],
```

字段	别名列表
资产总计	资产总额、总资产、资产总和、资产总金额、资产
流动资产合计	流动资产、流动资产总计、流动资产总金额、流动
流动负债合计	流动负债、流动负债总额、流动负债总计、流动负
非流动负债合计	非流动负债、非流动负债总额、非流动负债总计、

输出	单位	公式
三费比重	%	(销售费用+管理费用+财务费用)/营业收入
三费（销售费用、管理费用和财务	%	(销售费用+管理费用+财务费用)/营业收入
<字段1>与利润之比	<null>	<字段1>/净利润
<字段1>与利润比值	<null>	<字段1>/净利润
每股净资产	元/股	资产总计/股本
每股<字段1>	元/股	<字段1>/股本
资产负债比率	%	负债合计/资产总计
<字段1>增长率	%	(<字段1>-上年<字段1>)/上年<字段1>
<字段1>与<字段2>比值	<null>	<字段1>/<字段2>

问题描述	问题类型
<company><year>的<query_target><Float>具体是多少，结果保留2位小	1
<company><year>的<query_target><Integer>数量是多少?	1
<company><year>的<query_target><Integer>数是多少?	1
<company><year>的<query_target><Integer>比例是多少?	2
<company><year>的<query_target><Float>增长率是多少?	2
<company><year>的<query_target><Float>比率是多少?	2
<company><year><query_target><String>对比<year>是否相同?	5

向量库：

1. 建立公司名称和简称的向量库
2. 建立表字段和别名的向量库
3. 建立公式的向量库
4. 建立题型的向量库

方案介绍 - 结果处理

- 缺少年报
- 模板输出Type1和Type2
- 结合上下文的LLM输出 Type3
- 直接让LLM回答 Type4

```
我希望你是一个拥有年报数据的知识库，能够非常专业回答问题
现在用户会问年报相关的问题，我先会提供你年报内容，你通过理解用户问题和阅读部分相关的年报内容，对用户问题进行解答


年报内容
'''
{doc_context}
'''

要求如下：
保留年报内容和问题相关内容
如果年报所有内容都和问题无关联，就回复不知道

根据提供的年报内容，按要求回答以下问题
{question}   Zhang, 2023/9/4 15:33 • update
```

答案示例：

{"ID": 1,

"question": "2019年中国工商银行财务费用是多少元?",

"answer":"2019年中国工商银行财务费用是12345678.9元。"}

{"ID": 2,

"question": "工商银行2019年营业外支出和营业外收入分别是多少元?",

"answer": "工商银行2019年营业外支出为12345678.9元，营业外收入为2345678.9元。"}

{"ID":3",

"question": "中国工商银行2021年净利润增长率是多少?保留2位小数。",

"answer": "中国工商银行2020年净利润为12345678.90元，2021年净利润为22345678.90元，根据公式，净利润增长率=(净利润-上年净利润)/上年净利润，得出结果中国工商银行2021年净利润增长率81.00%。" }

比赛优化 - 意图识别不准确

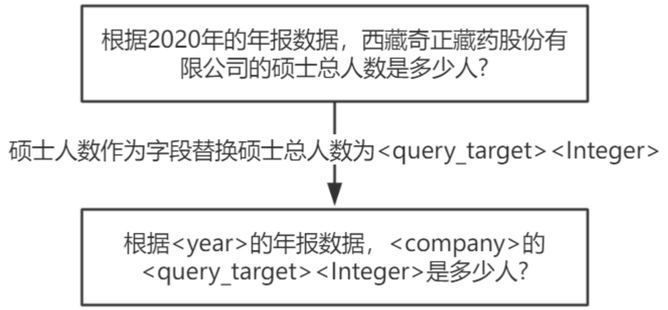
➢ 题型匹配

题型类似，查询项不一致：通过增加关键字类型<String><Float>替换后再向量相似度匹配题型

别名不准确：查询项无法完全匹配，非名词差1个字的字段，可以通过算法匹配，见右图

缺少公司名称：没有找到公司名称的题型，只要匹配到年份和查询项的都按统计题来

增加特殊题型：针对初赛和复赛题目部分提醒匹配错误，增加相关题型配置

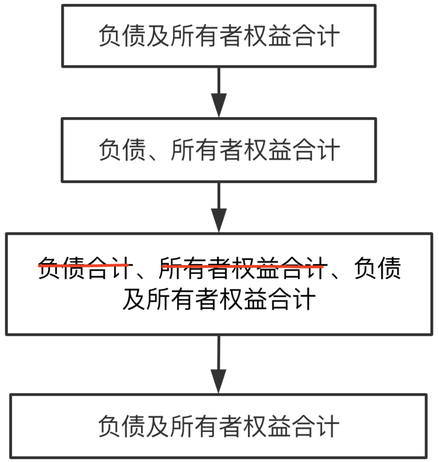


➢ 要素首次确认

公司名称：空、无、太短、非公司名称，通过~~LLM~~或向量相似度匹配

查询项：空、无、包含公司名称，通过~~LLM~~或向量相似度匹配

拆分完整字段，通过获取所有匹配字段根据长度去重，见右图



➢ 确认题型后要素二次确认

查询项：缺少或错误，通过~~LLM~~或向量相似度匹配

➢ 其他

增加别名、Few-Shot Prompt、通过标题级别搜索相关联上下文

比赛优化– 数据处理和校对

多轮数据获取

- 完整匹配三大表
- 三大表续表
- 特殊表或字段处理（股本、研发费用、非Excel表数据）
- 单文件多字段获取

数据对比工具

- 对比有关联的数据
- 对比新老数据库

比赛优化- 统计题SQL处理

- 确认年份、公司范围（注册地）
- 获取公司范围的子查询语句
- 确认统计字段
- 通过LLM获取统计SQL
- **通过分词和sqlparse校对SQL，并编写算法纠正limit、offset、order by**
- 根据limit不同模板返回结果

```python
def check_offset_limit_desc(speech_words, offset, limit, desc_or_asc):
    # 第一步看是否有最高
    cd_list = []  # 决定 limit
    od_list = []  # 决定 offset
    va_list = []  # 决定 desc
    _last_span = None
    _last_type = None

    for _words in speech_words:
        _span = _words['span']
        _type = _words['type']
        # 量词搜集
        if _type == 'CD':  # 前五 前3 前二十 前20
            cd_list.append((_span, _last_span))
        if _type == 'OD':  # 第五
            od_list.append(_span)
        if _type == 'VA':  # 最高 最低
            va_list.append((_span, _last_span))
        if _type == 'M' and _last_type == 'OD':
            od_list[-1] = od_list[-1]+_span
            va_list.append((_span, _last_span))
        _last_span = _span
        _last_type = _type
    desc_str = ''
    if cd_list:
        # 取第一个
        _num, _pre_word = cd_list[0]
        try:
            limit = int(cn2an.cn2an(_num, 'smart'))
        except:
            limit = 4
        if _pre_word in ('前', '头'):
            desc_or_asc = 'desc'
        desc_str = f'{_pre_word}{_num}'
    if od_list:
        # 取第一个
        _d_num = od_list[0]
        if _d_num[0] == '第':
            try:
                offset = int(cn2an.cn2an(_d_num[1:], 'smart')) - 1
            except:
                offset = 0
            limit = 1
            desc_str = f'{_d_num}'
    if va_list:
        _d_word, _z_word = va_list[0]  # 高,最
```

产品化总结 – 优缺点和调整

分析点	优点	缺点	调整方向
意图识别	兼容各类问题、识别准确、可替换LLM和API、通过微调优化识别	耗时高、重复处理	产品层面区分意图，通过向量匹配关键要素，无法识别的最后通过LLM获取
SQL能力	简单问句准确度高	无法处理复杂SQL	1. 问句预解析或缓存 2. 优化LLM 3. 接口化
性能	意图识别消耗少量Tokens，准确度还行，响应可以接受	整体并发性能受影响	减少LLM使用，增加缓存
灵活性	题型匹配快速调优，别名快速查询	题型调优有一定难度	输入自动完成的方式填写标准字段、公式等关键要素

产品化总结 - 架构升级

模型优化.

ChatGLM2-6b微调，Few-Shot优化.

上下文优化.

对年报段落文本预处理（压缩）

向量搜索替换LLM.

关键要素优先向量搜索

句型缓存.

支持输入问题过程，自动展示相关句型，供用户选择.

使用LangChain框架进行重构.

使用LLMChain、Prompt、Retrieval、Memory等.

接入API支持复杂SQL.

针对相关复杂报表，增加相关接口和报表名称

多种模型支持.

增加ZhipuAI的API对接、以及其他开源大模型

整合大数据.

接入大数据支持更多知识库场景


智谱·AI


marsoft

安硕信息


北京交通大學
BEIJING JIAOTONG UNIVERSITY


ModelScope
魔 搭 社 区


阿里云

Thanks