



(12) 发明专利申请

(10) 申请公布号 CN 117725078 A

(43) 申请公布日 2024. 03. 19

(21) 申请号 202311619928.3

(22) 申请日 2023.11.30

(71) 申请人 浙江大学计算机创新技术研究院
地址 311200 浙江省杭州市萧山区利一路
188号天人大厦33楼

(72) 发明人 查良瑜 苏常保 黄清仪 杨赛赛
袁静

(74) 专利代理机构 杭州求是专利事务有限公
司 33200
专利代理师 林超

(51) Int. Cl.
G06F 16/242 (2019.01)
G06F 16/25 (2019.01)

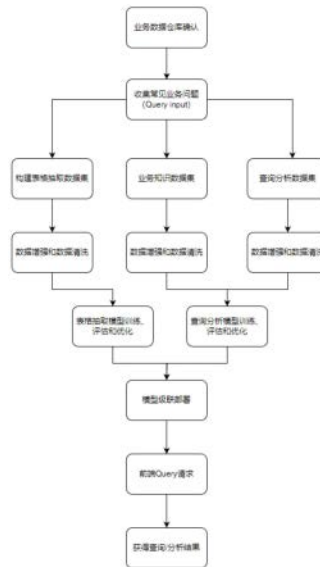
权利要求书1页 说明书8页 附图3页

(54) 发明名称

一种基于自然语言的多表格数据查询和预测方法

(57) 摘要

本发明公开了一种基于自然语言的多表格数据查询和预测方法。基于已有的业务数据库和具体的业务场景,收集常见的业务问题;在此基础上分别构建表格抽取数据集、业务知识数据集和查询分析数据集;选择表格抽取模型和查询分析指令生成模型,采用全参数微调的方式进行训练;在生产环境中部署模型,并根据实际功能需求差异,适配开发对应的指令翻译、矫正和执行模块;最后,通过WEB前端页面对模型发送数据查询、分析请求。本发明能够实现使用自然语言在多表格、多字段的复杂业务数仓环境中进行精确的数据提取查询、可视化分析和数据预测,让业务人员能够跨越SQL、Python等结构化数据提取分析语言的使用门槛,进行交互式的数据查询和分析。



1. 一种基于自然语言的多表格数据查询和预测方法,其特征在于,包括以下步骤:

步骤一:根据实际业务需求确认对接的数据仓库、数据表格,接着获取不同数据仓库之间的基本区分信息以及不同数据表格之间的基本区分信息,并记为数据来源信息;

步骤二:从当前对接的数据仓库中收集用户问题和请求,从而形成种子问题以及业务名词解释;

步骤三:基于当前对接的数据仓库、数据表格、数据来源信息、种子问题和业务名词解释,分别构建表格抽取数据集、业务知识数据集和查询分析数据集;

步骤四:分别对表格抽取数据集、业务知识数据集、查询分析数据集进行数据增强和数据清洗操作后,分别获得预处理后的表格抽取数据集、业务知识数据集、查询分析数据集;

步骤五:使用预处理后的表格抽取数据集训练表格抽取模型,获得训练好的表格抽取模型;以及使用业务知识数据集和查询分析数据集训练查询分析指令生成模型,获得训练好的查询分析指令生成模型;

步骤六:将训练好的表格抽取模型和查询分析指令生成模型级联部署后,获得多表格查询指令生成模型;将发送的用户请求文本输入到多表格查询指令生成模型中,获得最终的查询-分析结果。

2. 根据权利要求1所述的一种基于自然语言的多表格数据查询和预测方法,其特征在于,所述查询分析数据集包括在用户问题以及对应涉及或可能涉及的表格定义语句的基础上,输出查询分析对应领域限定语言的数据。

3. 根据权利要求1所述的一种基于自然语言的多表格数据查询和预测方法,其特征在于,所述步骤五中的表格选取模型,其模型基座采用RoBERTa-Chinese-WWM,模型训练时采用双塔式结构。

4. 根据权利要求1所述的一种基于自然语言的多表格数据查询和预测方法,其特征在于,所述步骤五中的查询分析指令生成模型,其基底模型为大语言生成模型WizardLM,采用全参数微调的方法对模型进行微调训练。

5. 根据权利要求1所述的一种基于自然语言的多表格数据查询和预测方法,其特征在于,所述步骤六中,对训练好的表格抽取模型的输出前处理后再与训练好的表格抽取模型的输入一起作为训练好的查询分析指令生成模型的输入。

6. 一种计算机设备,包括存储器和处理器,所述存储器存储有计算机程序,其特征在于,所述处理器执行所述计算机程序时实现权利要求1至5中任一项所述方法的步骤。

7. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1至5中任一项所述的方法的步骤。

一种基于自然语言的多表格数据查询和预测方法

技术领域

[0001] 本发明涉及了大数据和人工智能领域中的一种低代码自动化的数据查询分析方法,具体涉及了一种基于自然语言的多表格数据查询、可视化和预测方法。

背景技术

[0002] 数据驱动时代,数据查询和预测分析在日常生活中的普遍性和重要性不可忽视。无论是金融、营销、医疗还是教育等多个领域,数据分析成为决策支持的核心工具,深入影响着专业人士的工作方式和业务运营的效率。然而随着公司、业务规模的扩张,数据分析流程日益复杂化,另外不同部门和层级对于数据需求、数据归属、数据分析语言掌握程度存在差异,导致公司体系内数据查询分析过程冗长,效率低下。

[0003] 为解决以上问题,近来开展了大量低代码商务智能BI应用设计相关的研究工作,如陶宏等人研究了Power BI软件在国家采集药品使用数据分析和应用管理方面的使用(2022),朱晓伟等人对勘察设计企业BI软件的应用进行了尝试(2023)。另外随着生成式大语言模型(LLM)技术的发展,通过自然语言生成SQL语句(NL2SQL)来处理表格数据进行数据分析成为可能,在最新的spider-sql竞赛榜单中,阿里巴巴团队(Gao and Wang, 2023)使用DAIL-SQL+GPT-4方案,通过知识注入、多路生成召回等方式,在3-5张表格场景下获得了86.6%准确率的成绩。

[0004] 然而,对于传统低代码BI工具来说存在软件学习时间成本较高、无法跨越数据口径障碍的问题,在数据分析查询过程中,依然避免不了跨部门、层级、人员的沟通协调,效率提升有限。另外,目前对于基于生成式大语言模型LLM的生成SQL的技术(NL2SQL)在公司实际业务数分使用,则存在:1)多表格数据场景下,表格数量太多造成输入的上下文长度(context-length)过长,容易发生内容遗忘,影响生成结果的准确率;2)直接生成SQL语句的可控性较差,一旦出现生成错误,结果矫正困难;3)受限于SQL语句的功能,NL2SQL强于查询,但对于可视化和预测分析等功能,支持有限,难以满足实际使用的需求。

发明内容

[0005] 本发明的目的在于针对目前数据智能分析查询领域存在的上述不足,本发明设计了一套基于纯自然语言对话交互,采用多个模型级联来完成表格抽取、合并,并最终实现查询、分析、预测展示等功能的方法。另外针对生成式大语言模型(LLM)生成SQL语句可控性差,分析功能覆盖不完善的问题,专门设计一套能够覆盖大部分SQL查询需求,同时能够进行可视化和预测分析的领域限定语言(Domain-specific language,DSL)。

[0006] 为了解决上述现有技术的不足之处,本发明提供了以下技术方案:

[0007] 一、一种基于自然语言的多表格数据查询和预测方法

[0008] 步骤一:根据实际业务需求确认对接的数据仓库、数据表格,接着获取不同数据仓库之间的基本区分信息以及不同数据表格之间的基本区分信息,并记为数据来源信息;

[0009] 步骤二:从当前对接的数据仓库中收集用户问题和请求,从而形成种子问题以及

业务名词解释;

[0010] 步骤三:基于当前对接的数据仓库、数据表格、数据来源信息、种子问题和业务名词解释,分别构建表格抽取数据集、业务知识数据集和查询分析数据集;

[0011] 步骤四:分别对表格抽取数据集、业务知识数据集、查询分析数据集进行数据增强和数据清洗操作后,分别获得预处理后的表格抽取数据集、业务知识数据集、查询分析数据集;

[0012] 步骤五:使用预处理后的表格抽取数据集训练表格抽取模型,获得训练好的表格抽取模型;以及使用业务知识数据集和查询分析数据集训练查询分析指令生成模型,获得训练好的查询分析指令生成模型;

[0013] 步骤六:将训练好的表格抽取模型和查询分析指令生成模型级联部署后,获得多表格查询指令生成模型;将发送的用户请求文本输入到多表格查询指令生成模型中,获得最终的查询-分析结果。

[0014] 所述查询分析数据集包括在用户问题以及对应涉及或可能涉及的表格定义语句的基础上,输出查询分析对应领域限定语言的数据。

[0015] 所述步骤五中的表格选取模型,其模型基座采用RoBERTa-Chinese-WWM,模型训练时采用双塔式结构。

[0016] 所述步骤五中的查询分析指令生成模型,其基底模型为大语言生成模型 WizardLM,采用全参数微调的方法对模型进行微调训练。

[0017] 所述步骤六中,对训练好的表格抽取模型的输出前处理后再与训练好的表格抽取模型的输入一起作为训练好的查询分析指令生成模型的输入。

[0018] 二、一种计算机设备

[0019] 所述计算机设备包括存储器和处理器,所述存储器存储有计算机程序,其特征在于,所述处理器执行所述计算机程序时实现所述方法的步骤。

[0020] 三、一种计算机可读存储介质

[0021] 计算机可读存储介质上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现所述的方法的步骤。

[0022] 本发明的有益效果是:

[0023] 本发明方法设计了一套以自然语言作为输入,中间过程完全排除其他人工协助、干预,使用者无需任何Python/SQL等常用数据分析工具语言基础知识,直接获得数据查询、分析结果的流程框架。能够极大地降低业务数据分析、查询的技术门槛,减少中间跨部门、人员口径确认等沟通成本,提升数据分析、查询效率。

[0024] 本发明采用表格选取模型和查询分析指令生成模型级联的模式,通过选取模型精确选表,极大减少了查询分析指令生成模型的输入上下文长度,使得自然语言驱动的数据分析技术(如NL2SQL)能够更好地适配真实的多表格、多字段生产环境,同时提升指令(代码)生成的准确度。

[0025] 本发明针对数据查询语言SQL无法进行可视化分析、模型训练、预测等数据分析领域常用功能,以及直接生成SQL可控性较差、生成错误时矫正困难的问题,设计了一套能够覆盖SQL查询、可视化以及训练、预测功能的领域限定语言(Domain-specific Language)。这套DSL指令集,采用key:value对模式输出,生成后能快速矫正且易读易懂。并能够通过输

出多步指令列表的方式,完成复杂多步查询任务。另外需要强调和补充的是,针对对接的不同生产环境,还能灵活增加、定义该套DSL指令集内容,集成生产环境已有工具、组件。

附图说明

[0026] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明中记载的一些实施例,对于本领域普通技术人员来讲,还可以根据这些附图获得其他的附图。

[0027] 图1为本发明实施例提供的使用自然语言进行数据分析查询框架的搭建流程;

[0028] 图2为本发明实施例提供的从前端发送查询请求获得最终查询、分析结果的中间过程;

[0029] 图3为本发明实施例提供的在某基金指标管理数据库基础上完成模型搭建部署后,使用自然语言直接进行数据查询分析请求的样例。

[0030] 图4为本发明实施例提供的的数据查询分析请求样例(请求为:最近一周不同类型基金的平均涨跌幅是多少),对应的查询分析结果。

具体实施方式

[0031] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整的描述。显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0032] 如图1所示,本发明实施例属于大数据分析和人工智能领域,实施例以某证券公司基金智能查询项目为例,其基于自然语言进行数据查询分析的系统搭建过程基于以下几部分工作:

[0033] 步骤一:根据实际业务需求确认对接的数据仓库、数据表格,接着获取不同数据仓库之间的基本区分信息以及不同数据表格之间的基本区分信息,并记为数据来源信息;不同数据仓库之间以及不同数据表格之间的基本区分信息包括数据子仓库、数据表格描述、数据字段描述、表格外键约束等关键信息;

[0034] 数据子仓库,是指在数据仓库中,不同业务类型对应的数据子集。以某公司运营指标数据仓库为例,其可能包含的子仓库为:公司人事指标仓库、公司财务指标仓库、公司项目投资指标仓库等;

[0035] 数据表格描述、数据字段描述、表格外键约束等信息,可从数据仓库对应的数据定义语言Data Definition Language中获取,是区分不同数据仓库、数据表格的基本信息。

[0036] 在实施例下,需要注意的是,对接业务数据仓库的原始条件如表格、字段分层以及描述(comment)、外键约束完整性会对模型选表、查询分析指令生成的准确性造成较大影响。因此,在搭建该框架之前,需要对数据仓库进行必要的治理工作。

[0037] 步骤二:从当前对接的数据仓库中收集用户常用的问题和请求,从而形成基础的种子问题以及常用的业务名词解释;

[0038] 种子问题,指的是由使用方提供的,在特定业务场景上最常使用的问题。种子问题,在下述数据增强过程中,会拓展为多个语义近似的不同问题,用以强化模型对具体问

题、语义的泛化理解能力。

[0039] 在实施例下场场景对应数据仓库为例,总共涉及500余张表格,单张表格字段数量从5个到30个不等,对应单张表格对应收集5-10个种子问题为佳。这5-10个种子问题,应保证能覆盖表格中大部分常用字段,且种子问题涉及业务和字段内容应保持较大差异,最终收集约3000个种子问题。

[0040] 基于当前对接的数据仓库、数据表格、数据来源信息、种子问题和业务名词解释,分别构建表格抽取数据集、业务知识数据集和查询分析数据集;

[0041] 步骤三中,表格抽取数据集,指的是为了训练深度学习模型区分不同问题与不同表格以及表格字段映射关系能力的数据集,每条数据由“input”,“right_label”,和“false_label”标记的一个三元组构成,具体如下所示:

```
{
  "input": "{用户 query: 我想知道上周的退料情况和主要的退料原因。}",
  "right_label": "{正确表格对应的数据定义语言}"
  "false_label": "{从其他错误表格中随机抽样表格对应的数据定义语言}"
}
```

业务知识数据集,是指对具体业务涉及到的一些专有名词的文本型解释,包含了可能涉及的计算方法,业务阐释等。数据形式如下所示:

```
[0042] {
  "input": "同比增长率是什么?",
  "output": "同比增长率比较了在同一时间段(通常是按年、季度或月)的两个相继的时期某个经济或财务指标的表现。它表示了某个指标的相对变化,通常以百分比表示。
  计算同比增长率的公式如下:
  同比增长率=(本期值 - 同期基准值) / 同期基准值×100%"
}
```

[0043] 查询分析数据集包括在用户问题以及对应涉及或可能涉及的表格定义语句的基础上,输出查询分析对应领域限定语言的数据。数据形式如下所示:

```
{
  "input": "You are a Domain-specific language generator, you should generate suitable content to analysis data according to the input date information: {数据定义语言}, and the human input query: {用户 Query 如: 我想知道上周的退料情况和主要的退料原因。}",
  "output": "{特定的领域限定语言组合}"
}。
```

[0045] 领域限定语言(Domain-specific Language,DSL),在该数据查询分析场景下,覆盖了数据查询语言SQL的常用关键字组合、可视化分析以及训练预测分析等功能。单个领域限定语言(DSL)指令由“input”,“output”,“command”和“command_args”构成的四元组组

成。每个单独的DSL指令可以看做一行如下形式的程序语言:output=command(input,**command_args)。其中command可以看做一个function_api,command_args则是这个function_api的输入参数,input和output则分别是这个function_api的输入和输出数据。任意复杂查询、分析任务,都可以拆解为有限个DSL指令的组合,根据每个DSL语句中input/output的名称,可以获取DSL指令间的执行顺序或连接关系,最终可以翻译为由有限程序语言定义的一个简单程序。

[0046] 受限于篇幅,不便于将该场景下完整的领域限定语言指令集全部展示。对于查询功能,以SQL分组聚合查询Select Agg()GroupBy关键字对应的领域限定语言为例,对应指令输出如下所示:

```
{
    "input": ["df_ori"],
    "output": ["df_groupby_agg_0"],
    "command": "GroupbyAgg",
    "command_args": {
[0047]     "by": ["City"],
        "agg_args": {
            "Age": ["mean"]
        }
    }
}
```

[0048] 该段指令的具体含义是:从原始名为df_ori的表格中,以“City”列为分组列,对不同“City”类型的“Age”列求平均值,将结果保存为“df_groupby_agg_0”的表格。

[0049] 对于可视化功能,以饼图绘制为例,对应的领域限定语言DSL指令输出如下所示:

```
[0050] {
```

```
        "input":["df_ori"],
        "output":[],
        "command":"Area",
        "command_args":{
[0051]         "x":["xticks"],
        "y":["signups", "sales","visits"],
        "row_values":[]
        },
    }
```

[0052] 该指令的实际含义是：从原始名为“df_ori”的数据表中，抽取“signups”、“sales”、“visits”三列的值绘制饼图。

[0053] 对于训练预测分析，以模型训练为例，对应的领域限定语言DSL指令如下所示：

```
{
    "input": ["df_ori"],
    "output": [],
    "command": "Training",
[0054]    "command_args":{
        "tasktype": "classification",
        "target": ["Survived"]
    }
}
```

[0055] 该指令的实际含义是：在原始数据表df_ori的基础上，以“survived”列为预测目标列，训练一个分类(classification)模型。

[0056] 上述领域限定语言的举例都只涉及单步简单查询，对于更复杂的多步查询分析，查询、分析指令生成模型能够生成一套由多个特定领域限定语言(DSL)指令构成的列表集合：[DSL1,DSL2,⋯,DSL_N]。

[0057] 步骤四:分别对表格抽取数据集、业务知识数据集、查询分析数据集进行数据增强和数据清洗操作后,分别获得预处理后的表格抽取数据集、业务知识数据集、查询分析数据集;

[0058] 数据增强,在步骤二种子问题的基础上,进行近义词替换、近似语句生成、关键条件(如问题中涉及的日期、数值)替换等操作,将一条原始数据拓展为多条近似数据的操作。

[0059] 数据清洗,指的是使用一套固定的规则过滤、稀有字段剔除、人工筛选的流程,对标定、生成的原始数据进行筛选和优化。

[0060] 在实施例下,经过增强、清洗后的数据样本与原始种子问题对应的数据样本的比例大约在10:1左右,即每一条原始种子问题(Query)将分别对应10条表格抽取数据和10条查询分析数据。

[0061] 实践表明,增强和清洗后的数据,能够显著提升表格选取模型以及查询分析指令生成模型对于场景问题的理解、泛化能力。

[0062] 步骤五:使用预处理后的表格抽取数据集训练表格抽取模型,获得训练好的表格抽取模型;使用业务知识数据集和查询分析数据集训练查询分析指令生成模型,获得训练好的查询分析指令生成模型;具体实施中,还使用提前预留的测试/验证数据,分别对表格抽取模型和查询分析指令生成模型进行效果评估。根据评估结果是否能满足生产需求,通过增补数据、调整模型参数等方式,对模型进行迭代调优,直至满足生产需求为止。

[0063] 步骤五中的表格选取模型,其模型基座采用RoBERTa-Chinese-WWM,模型训练时采用双塔式结构。每次将[“query”,“right_label”,“false_label”]的三元组样本拆拆分为[“query”,“right_label”],[“query”,“false_label”]两条样本,分别输入到两个基座模型中,对两个模型的输出进行池化(Pooling),计算对比损失(Constrasive Loss)。

[0064] 对比损失(Constrasive Loss)的定义如下式:

$$[0065] \quad L(W, (Y, X_1, X_2)) = \frac{1}{N} \sum_{n=1}^N Y D_W^2 + (1 - Y) \max(m - D_W, 0)^2$$

[0066] 式中: D_W 代表两对样本([“query”,“right_label”],[“query”,“false_label”])的欧式距离;Y表示两个样本是否匹配的标签,对于[“query”,“right_label”]样本,Y=1,对于[“query”,“false_label”],Y=0;m为设定的超参数阈值;N为样本数量;max()表示求最大值函数, $\max(m - D_W, 0)^2$ 表示的是当两个样本的欧式距离超过阈值m时,loss为0,该表达式的内涵是如果两个不相似的样本特征距离很远,其对比损失应该很低。

[0067] 步骤五中的查询分析指令生成模型,其基底模型为大语言生成模型WizardLM,以查询、分析数据集和业务知识数据集和输入训练语料,采用全参数微调的方法对模型进行微调训练。

[0068] 全参数微调(Supervised fine-tuning)指的是训练过程会对大模型所有模块、中间层涉及的参数进行调整,采用该训练方法的原因是,在查询分析场景中,对输出指令的准确度要求较高。实践经验表明,相比模型的低秩适配(LORA)等微调方式,全参数微调后的模型,指令生成的精确度最高。

[0069] 在实施例下,表格选取模型基座采用Roberta-chinese-ext-large,查询、分析指令生成模型基座采用WizardLM 13B base。

[0070] 表格选取模型选表采用[“Query”,“Table schema DDL”]文本对向量余弦相似度

Top10召回,评估过程中要求Top10召回正确率大于99%。

[0071] 召回正确率的评估方式是:相似度排名前10对应表格集合包含正确表格的概率大于99%。

[0072] 查询分析模型推理输入形式为[“Table schema DDL+Query”],输出内容为对应的领域限定语言(DSL)指令集列表[DSL1,DSL2,⋯,DSL_N]。评估过程要求在验证集上,模型生成指令集和标注指令集的结果匹配度超过90%。

[0073] 指令集的结果匹配度计算方法是:

[0074] 1.计算生成指令集DSL_p和标注指令集DSL_{true}完全匹配的数量及比例。

[0075] 2.对于无法完全匹配的DSL_p和DSL_{true},分别将DSL_p和DSL_{true}输入指令集解释执行器,获得最终分析结果,如果两对分析结果相同,则认为DSL_p与DSL_{true}能够完成配对。

[0076] 3.将1)和2)中能够匹配的样本对应的正确率求和,最终达到90%以上。

[0077] 步骤六:如图2所示,将训练好的表格抽取模型和查询分析指令生成模型级联部署后,获得多表格查询指令生成模型;将用户从产品前端页面发送的用户请求文本输入到多表格查询指令生成模型中,逐级进行选表、查询分析指令生成、查询分析指令解释执行等步骤,由指令集解释和执行器执行模型生成的指令集,获得最终的查询-分析结果。

[0078] 步骤六中,对训练好的表格抽取模型的输出前处理后再与训练好的表格抽取模型的输入(即用户请求文本)一起作为训练好的查询分析指令生成模型的输入。

[0079] 级联部署,是指将表格选取模型和查询分析指令生成模型串联部署。每次接受到前端发送的请求时,先将用户请求文本输入表格选取模型,表格选取模型会根据用户请求query与各表格对应数据定义语言DSL的相似度,输出召回5-10张表格对应的数据定义语言。随后对表格抽取模型的输出进行前处理后再次与用户请求文本结合后,共同输入查询分析指令生成模型。

[0080] 采用这种模型级联部署形式的原因是,在一般业务数据仓库中,表格数量、表格字段数量较多,直接将所有原始表格、表格字段的定义语句输入查询分析指令生成模型,很容易超出模型的上下文长度限制。即便使用了诸如ROPE等技术增强长文本输入能力,仍容易发生长文本遗忘现象,对于生成内容的精准度影响很大。使用表格选取模型,先提前剔除与问题不相关的表格,只选取5-10张表格及相关字段作为输入,能够极大减轻查询分析指令生成模型的上下文长度负担,提高指令输出精准度。

[0081] 在实施例下,指令集解释和执行器对应的代码采用python编写,不同指令(DSL)产生的中间结果表格,保存为.csv格式文件,持久化保存在特定的存储单元内。

[0082] 图3和图4分别是该实施例下,从前端发送请求完成组件执行和数据分析,最终得到查询分析结果的样例图。

[0083] 最后所应说明的是,以上实施例和阐述仅用以说明本发明的技术方案而非进行限制。本领域的普通技术人员应当理解,可以对本发明的技术方案进行修改或者等同替换,不脱离本发明技术方案公开的精神和范围的,其均应涵盖在本发明的权利要求保护范围之内。

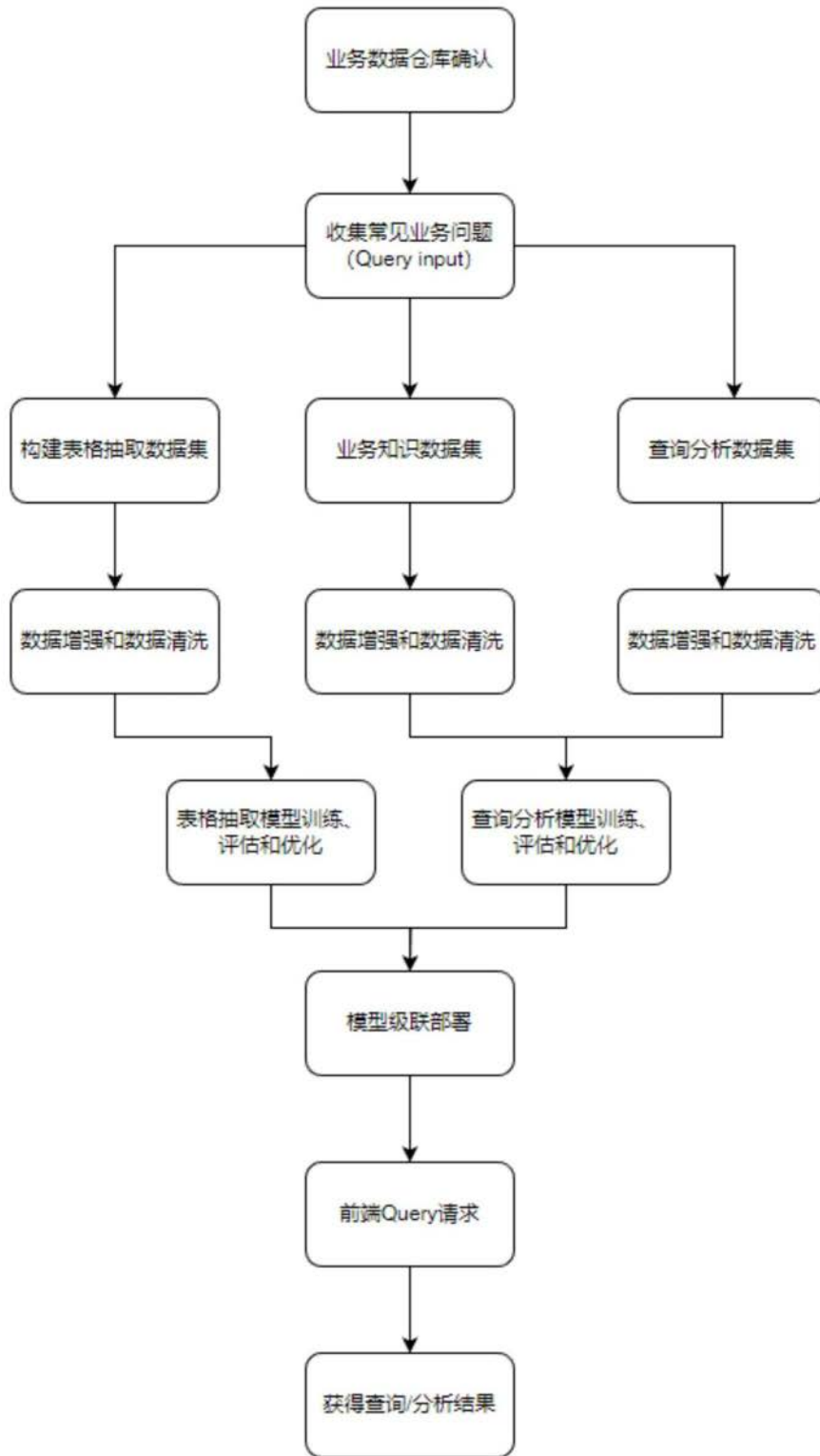


图1

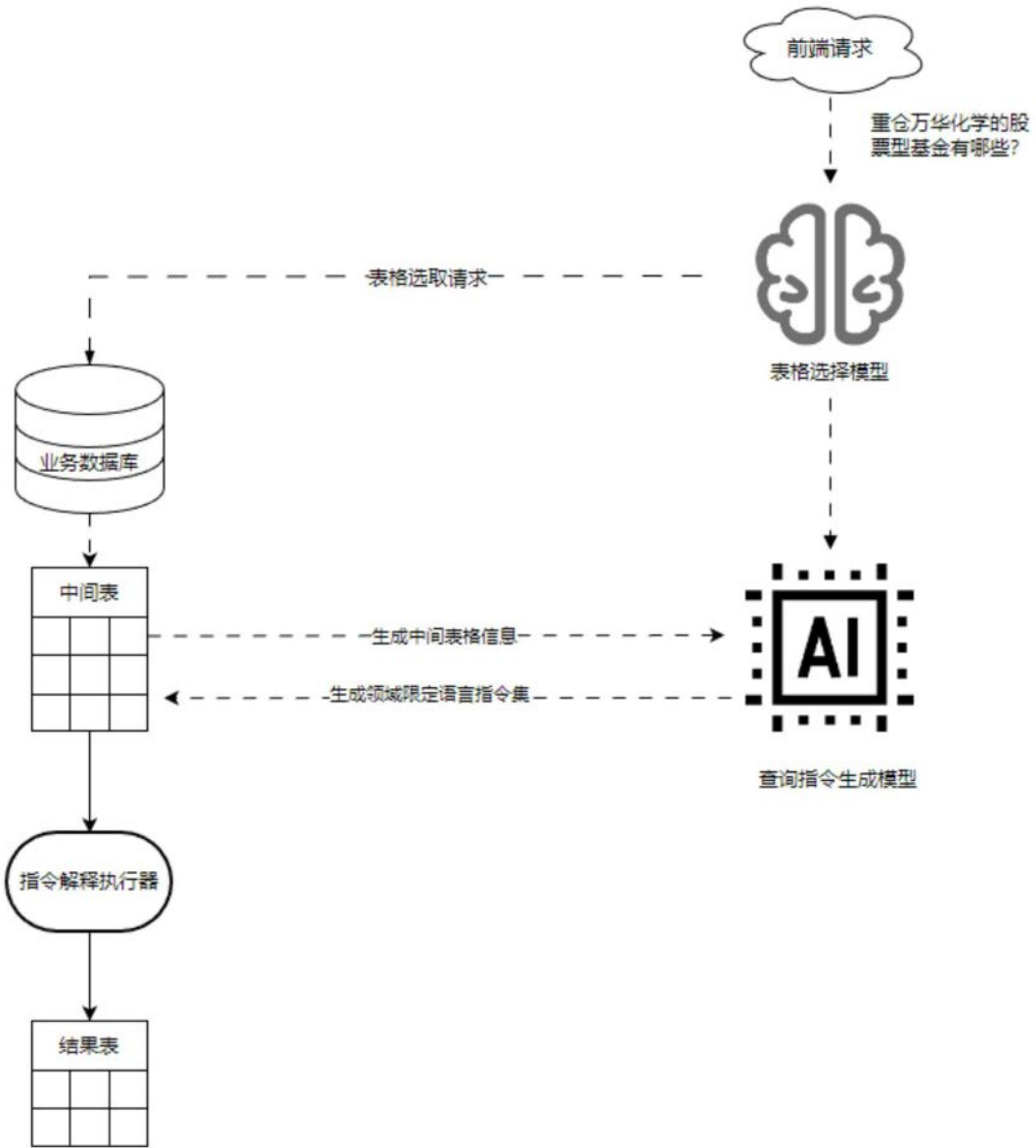
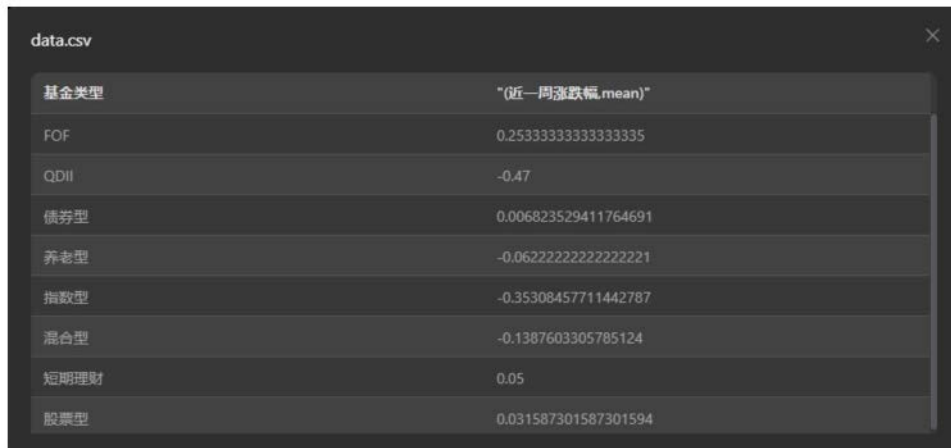


图2



图3



The image shows a screenshot of a file named 'data.csv' with a close button in the top right corner. The file contains a table with two columns: '基金类型' (Fund Type) and a numerical value. The values represent the mean of the weekly percentage change in the past week, as indicated by the header '(近一周涨跌幅,mean)'. The data is as follows:

基金类型	(近一周涨跌幅,mean)
FOF	0.25333333333333335
QDII	-0.47
债券型	0.006823529411764691
养老型	-0.06222222222222221
指数型	-0.35308457711442787
混合型	-0.1387603305785124
短期理财	0.05
股票型	0.031587301587301594

图4