(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112434070 A
(43)申请公布日 2021.03.02

(21)申请号 202011466707.3

(22)申请日 2020.12.14

(71)申请人 四川长虹电器股份有限公司
地址 621000 四川省绵阳市高新区绵兴东路35号

(72)发明人 余锡娟 钟声

(74)专利代理机构 四川省成都市天策商标专利事务所 51213
代理人 赵以鹏

(51)Int.Cl.
G06F 16/2455(2019.01)

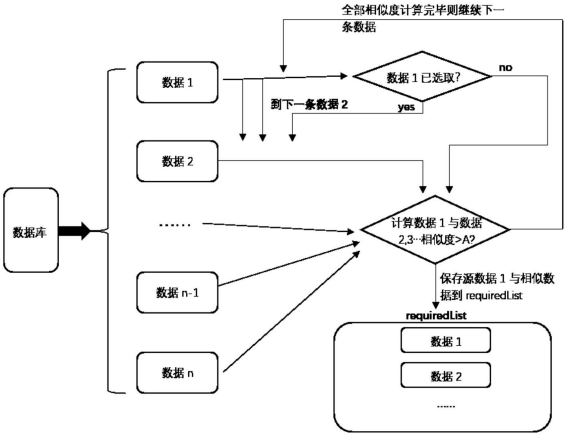
权利要求书2页 说明书5页 附图1页

(54)发明名称

一种基于相似度算法的分页查询方法

(57)摘要

本发明公开了一种基于相似度算法的分页查询方法，包括以下步骤：步骤1：查询出数据表中需要进行排序的需求数据；步骤2：新建一个空集合requiredList用于保存计算了相似度后，相似数据已相邻集中排列的数据；步骤3：根据余弦相似度算法进行排序计算；步骤4：对于dataList中除[数据1]之外的剩余需求数据进行遍历处理；步骤5：对最终所取得requiredList中的数据进行自定义分页，得到的数据即为相似数据集中展示的结果。通过相似度算法对查询结果数据进行排序及分页的方法；利用算法解决了分页查询无法将相似数据集中排列排序及查询性能降低的问题，有效提升了查询效率。



CN 112434070 A

1.一种基于相似度算法的分页查询方法,其特征在于,包括以下步骤:

步骤1:查询出数据表中需要进行排序的需求数据;

步骤2:新建一个空集合requiredList用于保存计算了相似度后,相似数据已相邻集中排列的数据;

步骤3:根据余弦相似度算法进行排序计算;

步骤4:对于dataList中除[数据1]之外的剩余需求数据进行遍历处理;

步骤5:对最终所取得requiredList中的数据进行自定义分页,得到的数据即为相似数据集中展示的结果。

2.根据权利要求1所述的一种基于相似度算法的分页查询方法,其特征在于,所述步骤1,具体包括:

步骤1.1:建立一个对象集合dataList,用于保存从数据表中查询所得的数据;

步骤1.2:通过查询条件,使用select…from…where…语句查询出对应满足查询条件的所有数据[数据1,数据2……数据n-1,数据n];

步骤1.3:将步骤1.3中查询所得到的[数据1,数据2……数据n_{-1},数据n],保存到步骤1.1中所建的对象集合dataList中。

3.根据权利要求2所述的一种基于相似度算法的分页查询方法,其特征在于,所述步骤2中所创建的空集合requiredList,其类型与步骤1.1中所建立的对象集合dataList类型一致,区别在于其中数据的排列顺序不同。

4.根据权利要求2所述的一种基于相似度算法的分页查询方法,其特征在于,所述步骤3,具体包括:

步骤3.1:从步骤1得到的dataList数据集合中的数据里选取出每条数据需要进行相似度计算的关键字,选出的关键字字段对应的值用于后续计算相似度;

步骤3.2:设置一个预定值A作为数据相似度的判断标准阈值,采用余弦相似度算法,其计算公式为:

$$\cos(\theta)=\frac{\sum_{i=1}^{k}(x_i\times y_i)}{\sqrt{\sum_{i=1}^{k}(x_i)^2}\times\sqrt{\sum_{i=1}^{k}(y_i)^2}}$$

其中,x_i指[数据m_{-1},数据m]中[数据m_{-1}]中某个字母出现的次数,y_i指[数据m_{-1},数据m]中[数据m]中此字母出现的次数,k指[数据m_{-1},数据m]中字母的个数,m为3≤m≤n的整数;预定值A的取值范围为0～1。

步骤3.3:将步骤1得到的dataList中的第一个数据[数据1]保存至步骤2创建的requiredList集合中,根据余弦相似度算法计算出此[数据1]与dataList集合中其余数据[数据2……数据n-1,数据n]的相似度a(1),a(2)……a(n-2),a(n-1),若相似度a(i)大于步骤3.2所设定的预定值A,则认为此条数据与[数据1]相似,若a<A则认为与[数据1]不相似;

步骤3.4:将步骤3.3中计算出的所有相似度大于A的数据按顺序保存到步骤2中所创建

的集合requiredList中。

5.根据权利要求2所述的一种基于相似度算法的分页查询方法,其特征在于,所述步骤4,具体包括:

步骤4.1:首先判断[数据x]是否已经保存在步骤2创建的集合requiredList中;其中,[数据x]为[数据2……数据n_{-1},数据n]中的任一一个数据;

步骤4.2:如果该数据已存在于步骤2创建的集合requiredList中,则获取下一个数据重新执行步骤4,

步骤4.3:如果数据不存在于步骤2创建的集合requiredList中,则将[数据1]替换为[数据x]后重复步骤3中的步骤3.3和步骤3.4进行相似度计算,将结果保存至集合requiredList中;

步骤4.4:获取下一个数据重新执行步骤4直到需求数据集合dataList中的最后一条数据也保存至集合requiredList中。

6.根据权利要求2所述的一种基于相似度算法的分页查询方法,其特征在于,所述步骤5,具体包括:

步骤5.1:建立一个集合pageDataList,保存分页后的数据;

步骤5.2:获取到需要展示的页码数pageNum及每页的数据量pageSize,此两条数据由前端作为分页参数传入;

步骤5.3:计算数据总数totalCount:取requiredList集合的大小作为数据总数;总页数totalPageNum:取数据总数除以每页数据量向上取整;

步骤5.4:根据步骤5.1获取的页码数及每页数据量计算出当前页需要展示的数据位于requiredList集合中的位置;其中开始位置startNum＝(pageNum-1)*pageSize;结束位置endNum＝pageNum*pageSize-1;

步骤5.5:从requiredList中取出第startNum～endNum位的数据存入步骤5.1建立的集合pageDataList中,此时获得的pageDataList集合即为相似数据集中且分页展示的数据。

一种基于相似度算法的分页查询方法

技术领域

[0001] 本发明涉及分页查询领域,更具体的说是涉及一种基于相似度算法的分页查询方法。

背景技术

[0002] 在通常的分页排序方法中,大多采用通过数据库(如mysql等)的排序方法查询,利用LIMIT限制获取数据的起始值来获取指定页的数据。但使用此种方法会存在两个问题:一是通常数据库中排序方式为按首字母排序或按数据字段编码排序,如果需要将相似数据集中排列则无法做到,如ABCDEFG和ADBCEFG在一定情况下认为是相似数据,但使用通常的方法在这两条数据之间会夹杂大量如AC……的数据;二是每一次的换页查询均会请求一次数据库,而多次进行数据库的查询操作会将时间耗费在打开连接上,导致效率低下。

发明内容

[0003] 本发明的目的在于提供一种基于相似度算法的分页查询方法,以期解决背景技术中的问题。本专利利用相似度算法,解决了分页查询无法将相似数据集中排列排序及查询性能降低的问题,有效提升了查询效率。

[0004] 为了实现上述目的,本发明采用以下技术方案:

[0005] 一种基于相似度算法的分页查询方法,包括以下步骤:

[0006] 步骤1:查询出数据表中需要进行排序的需求数据;

[0007] 步骤2:新建一个空集合requiredList用于保存计算了相似度后,相似数据已相邻集中排列的数据;

[0008] 步骤3:根据余弦相似度算法进行排序计算;

[0009] 步骤4:对于dataList中除[数据1]之外的剩余需求数据进行遍历处理;

[0010] 步骤5:对最终所取得requiredList中的数据进行自定义分页,得到的数据即为相似数据集中展示的结果。

[0011] 进一步的,所述步骤1,具体包括:

[0012] 步骤1.1:建立一个对象集合dataList,用于保存从数据表中查询所得的数据;

[0013] 步骤1.2:通过查询条件,使用select…from…where…语句查询出对应满足查询条件的所有数据[数据1,数据2……数据n-1,数据n];

[0014] 步骤1.3:将步骤1.3中查询所得到的[数据1,数据2……数据n_{-1},数据n],保存到步骤1.1中所建的对象集合dataList中。

[0015] 进一步的,所述步骤2中所创建的空集合requiredList,其类型与步骤1.1中所建立的对象集合dataList类型一致,区别在于其中数据的排列顺序不同。

[0016] 进一步的,所述步骤3,具体包括:

[0017] 步骤3.1:从步骤1得到的dataList数据集合中的数据里选取出每条数据需要进行相似度计算的关键字,选出的关键字字段对应的值用于后续计算相似度;

[0018] 步骤3.2:设置一个预定值A作为数据相似度的判断标准阈值,采用余弦相似度算法,其计算公式为:

[0019] $$\cos(\theta)=\frac{\sum_{i=1}^{k}(x_i\times y_i)}{\sqrt{\sum_{i=1}^{k}(x_i)^2}\times\sqrt{\sum_{i=1}^{k}(y_i)^2}}$$

[0020] 其中,x_i指[数据m_{-1},数据m]中[数据m_{-1}]中某个字母出现的次数,y_i指[数据m_{-1},数据m]中[数据m]中此字母出现的次数,k指[数据m_{-1},数据m]中字母的个数,m为3≤m≤n的整数;预定值A的取值范围为0～1。

[0021] 步骤3.3:将步骤1得到的dataList中的第一个数据[数据1]保存至步骤2创建的requiredList集合中,根据余弦相似度算法计算出此[数据1]与dataList集合中其余数据[数据2……数据n-1,数据n]的相似度a(1),a(2)……a(n-2),a(n-1),若相似度a(i)大于步骤3.2所设定的预定值A,则认为此条数据与[数据1]相似,若a<A则认为与[数据1]不相似;

[0022] 步骤3.4:将步骤3.3中计算出的所有相似度大于A的数据按顺序保存到步骤2中所创建的集合requiredList中。

[0023] 进一步的,所述步骤4,具体包括:

[0024] 步骤4.1:首先判断[数据x]是否已经保存在步骤2创建的集合requiredList中;其中,[数据x]为[数据2……数据n_{-1},数据n]中的任一一个数据;

[0025] 步骤4.2:如果该数据已存在于步骤2创建的集合requiredList中,则获取下一个数据重新执行步骤4,

[0026] 步骤4.3:如果数据不存在于步骤2创建的集合requiredList中,则将[数据1]替换为[数据x]后重复步骤3中的步骤3.3和步骤3.4进行相似度计算,将结果保存至集合requiredList中;

[0027] 步骤4.4:获取下一个数据重新执行步骤4直到需求数据集合dataList中的最后一条数据也保存至集合requiredList中。

[0028] 进一步的,所述步骤5,具体包括:

[0029] 步骤5.1:建立一个集合pageDataList,保存分页后的数据;

[0030] 步骤5.2:获取到需要展示的页码数pageNum及每页的数据量pageSize,此两条数据由前端作为分页参数传入;

[0031] 步骤5.3:计算数据总数totalCount:取requiredList集合的大小作为数据总数;总页数totalPageNum:取数据总数除以每页数据量向上取整;

[0032] 步骤5.4:根据步骤5.1获取的页码数及每页数据量计算出当前页需要展示的数据位于requiredList集合中的位置;其中开始位置startNum＝(pageNum-1)*pageSize;结束位置endNum＝pageNum*pageSize-1;

[0033] 步骤5.5:从requiredList中取出第startNum～endNum位的数据存入步骤5.1建立的集合pageDataList中,此时获得的pageDataList集合即为相似数据集中且分页展示的数

据。

[0034] 本发明与现有技术相比具有的有益效果是：

[0035] 通过相似度算法对查询结果数据进行排序及分页的方法;利用算法解决了分页查询无法将相似数据集中排列排序及查询性能降低的问题,有效提升了查询效率。

附图说明

[0036] 图1为本发明的一种基于相似度算法的分页查询方法的流程图；

具体实施方式

[0037] 下面结合实施例对本发明作进一步的描述,所描述的实施例仅仅是本发明一部分实施例,并不是全部的实施例。基于本发明中的实施例,本领域的普通技术人员在没有做出创造性劳动前提下所获得的其他所用实施例,都属于本发明的保护范围。

[0038] 实施例1:

[0039] 一种基于相似度算法的分页查询方法,包括以下步骤：

[0040] 步骤1:查询出数据表(如mysql或oracle等数据库)中需要进行排序的需求数据；

[0041] 步骤1.1:建立一个对象集合dataList,用于保存从数据表中查询所得的数据；

[0042] 步骤1.2:通过查询条件,使用select…from…where…语句查询出对应满足查询条件的所有数据[数据1,数据2……数据n-1,数据n]；

[0043] 步骤1.3:将步骤1.3中查询所得到的[数据1,数据2……数据n_{-1},数据n],保存到步骤1.1中所建的对象集合dataList中。

[0044] 步骤2:新建一个空集合requiredList用于保存计算了相似度后,相似数据已相邻集中排列的数据;所述步骤2中所创建的空集合requiredList,其类型与步骤1.1中所建立的对象集合dataList类型一致,区别在于其中数据的排列顺序不同。

[0045] 步骤3:根据余弦相似度算法进行排序计算；

[0046] 步骤3.1:从步骤1得到的dataList数据集合中的数据里选取出每条数据需要进行相似度计算的关键字,选出的关键字字段对应的值用于后续计算相似度；

[0047] 步骤3.2:设置一个预定值A作为数据相似度的判断标准阈值,采用余弦相似度算法,其计算公式为：

[0048] $$\cos(\theta)=\frac{\sum_{i=1}^{k}(x_i\times y_i)}{\sqrt{\sum_{i=1}^{k}(x_i)^2}\times\sqrt{\sum_{i=1}^{k}(y_i)^2}}$$

[0049] 其中,x_i指[数据m_{-1},数据m]中[数据m_{-1}]中某个字母出现的次数,y_i指[数据m_{-1},数据m]中[数据m]中此字母出现的次数,k指[数据m_{-1},数据m]中字母的个数,m为3≤m≤n的整数;预定值A的取值范围为0～1。

[0050] 步骤3.3:将步骤1得到的dataList中的第一个数据[数据1]保存至步骤2创建的

requiredList集合中,根据余弦相似度算法计算出此[数据1]与dataList集合中其余数据[数据2……数据n-1,数据n]的相似度a(1),a(2)……a(n-2),a(n-1),若相似度a(i)大于步骤3.2所设定的预定值A,则认为此条数据与[数据1]相似,若a<A则认为与[数据1]不相似;

[0051] 步骤3.4:将步骤3.3中计算出的所有相似度大于A(即与[数据1]相似)的数据按顺序保存到步骤2中所创建的集合requiredList中。

[0052] 步骤4:对于dataList中除[数据1]之外的剩余需求数据进行遍历处理;

[0053] 步骤4.1:首先判断[数据x]是否已经保存在步骤2创建的集合requiredList中;其中,[数据x]为[数据2……数据n_{-1},数据n]中的任一一个数据;

[0054] 步骤4.2:如果该数据已存在于步骤2创建的集合requiredList中,则获取下一个数据重新执行步骤4,

[0055] 步骤4.3:如果数据不存在于步骤2创建的集合requiredList中,则将[数据1]替换为[数据x]后重复步骤3中的步骤3.3和步骤3.4进行相似度计算,将结果保存至集合requiredList中;

[0056] 步骤4.4:获取下一个数据重新执行步骤4直到需求数据集合dataList中的最后一条数据也保存至集合requiredList中。

[0057] 步骤5:对最终所取得requiredList中的数据进行自定义分页,得到的数据即为相似数据集中展示的结果。

[0058] 步骤5.1:建立一个集合pageDataList,保存分页后的数据;

[0059] 步骤5.2:获取到需要展示的页码数pageNum及每页的数据量pageSize,此两条数据由前端作为分页参数传入;

[0060] 步骤5.3:计算数据总数totalCount:取requiredList集合的大小作为数据总数;总页数totalPageNum:取数据总数除以每页数据量向上取整;

[0061] 步骤5.4:根据步骤5.1获取的页码数及每页数据量计算出当前页需要展示的数据位于requiredList集合中的位置(即数据的起始位置);其中开始位置startNum=(pageNum-1)*pageSize;结束位置endNum=pageNum*pageSize-1;

[0062] 步骤5.5:从requiredList中取出第startNum～endNum位的数据存入步骤5.1建立的集合pageDataList中,此时获得的pageDataList集合即为相似数据集中且分页展示的数据。

[0063] 下面结合附图,通过具体的余弦相似度算法来详细描述本发明的具体实施例,如图1所示,具体的工作流程如下:

[0064] 步骤1:从数据库表table中按指定字段englishName对应值的首字母排序查询出全部数据,将数据存入集合dataList中。

[0065] 步骤2:创建对象集合List<Object>requiredList,用于保存计算了相似度后的数据。

[0066] 步骤3:设定一个相似度判定阈值A=0.7

[0067] 步骤4:遍历dataList,对dataList中每一个数据进行处理及判断:

[0068] 4.1:取出数据data=dataList.get(i),判断data是否存在requiredList中,若存在则取下一个数据继续判断,不存在则进行下一步计算。

[0069] 4.2:根据余弦相似度算法依次计算data的englishName字段与dataList中剩下数

据englisName字段的相似度,计算结果为a,b,c,d……

[0070] 4.3:将数据data及相似度计算结果大于A的数据依次存入requiredList集合中。

[0071] 步骤5:对处理后得到的requiredList集合进行自定义分页:

[0072] 5.1:获取前端传入的页码数pageNum及每页的数据量pageSize。

[0073] 5.2:建立一个集合pageDataList,类型与(2)中创建的requiredList一致,保存分页后的数据;

[0074] 5.3:计算数据总数totalCount:取requiredList集合的大小作为数据总数;总页数totalPageNum:取数据总数totalCount除以每页数据量pageSize向上取整

[0075] 5.4:根据5.1获取的pageNum及pageSize计算出当前页需要展示的数据位于requiredList集合中的位置(即数据的起始位置):其中开始位置startNum＝(pageNum-1)*pageSize;结束位置endNum＝pageNum*pageSize-1;

[0076] 5.5:从requiredList中取出第startNum～endNum位的数据存入5.2建立的集合pageDataList中,此时获得的pageDataList集合即为相似数据集中且分页展示的数据。

[0077] 以上所述仅为本发明较佳实例而已,本发明并不局限于此。对于本领域内的普通技术人员而言,在不脱离本发明的精神和实质的情况下,可以做出各种变型和改进,这些变型和改进也视为本发明的保护范围。

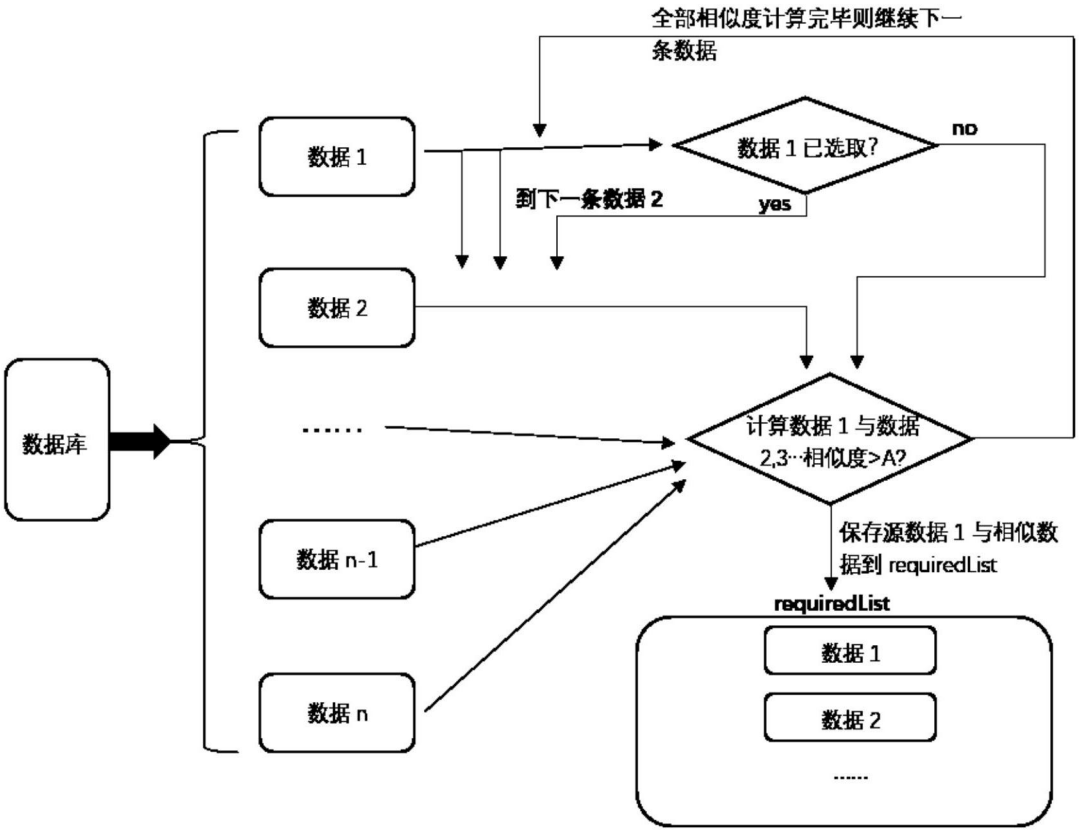

全部相似度计算完毕则继续下一条数据
数据 1
数据 1 已选取?
no
到下一条数据 2
yes
数据 2
数据库
.....
计算数据 1 与数据 2,3…相似度>A?
保存源数据 1 与相似数据到 requiredList
数据 n-1
requiredList
数据 1
数据 2
.....
数据 n

图1