一种基于文本聚类的分布式索引构建方法及系统

申请号：201610154682.0
申请日：2016-03-16

申请(专利权)人　中山大学
地址　510006 广东省广州市番禺区大学城中山大学东校区教学实验中心C401
发明(设计)人　林格　邓现

主分类号　G06F17/30(2006.01)I
分类号　G06F17/30(2006.01)I
公开(公告)号　105787097A
公开(公告)日　2016-07-20
专利代理机构
代理人

(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 105787097 A
(43)申请公布日 2016.07.20

(21)申请号 201610154682.0

(22)申请日 2016.03.16

(71)申请人 中山大学
地址 510006 广东省广州市番禺区大学城中山大学东校区教学实验中心C401

(72)发明人 林格 邓现

(51)Int.Cl.
G06F 17/30(2006.01)

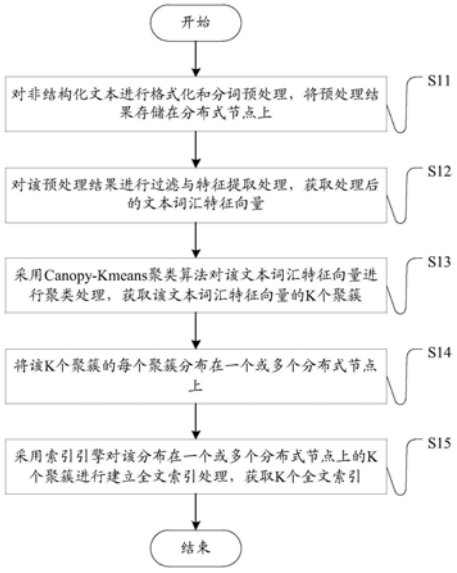
权利要求书2页 说明书9页 附图4页

(54)发明名称

一种基于文本聚类的分布式索引构建方法及系统

(57)摘要

本发明公开了一种基于文本聚类的分布式索引构建方法及系统，其中，所述方法包括：对非结构化文本进行格式化和分词预处理，将预处理结果存储在原来的分布式节点上；对所述预处理结果进行过滤与特征提取处理，获取处理后的文本词汇特征向量；采用Canopy-Kmeans聚类算法对所述文本词汇特征向量进行聚类处理，获取所述文本词汇特征向量的K个聚簇；将所述K个聚簇的每个聚簇分布在一个或多个分布式节点上；采用索引引擎对所述分布在一个或多个分布式节点上的所述K个聚簇进行建立全文索引处理，获取K个全文索引；实施本发明实施例，用于构建一种用于检索的分布式索引方式，给予用户一种快速的索引方式，提高用户的使用体验感。



CN 105787097 A

1.一种基于文本聚类的分布式索引构建方法,其特征在于,所述方法包括:

对非结构化文本进行格式化和分词预处理,将预处理结果存储在分布式节点上;

对所述预处理结果进行过滤与特征提取处理,获取处理后的文本词汇特征向量;

采用Canopy-Kmeans聚类算法对所述文本词汇特征向量进行聚类处理,获取所述文本词汇特征向量的K个聚簇;

将所述K个聚簇的每个聚簇分布在一个或多个分布式节点上;

采用索引引擎对所述分布在一个或多个分布式节点上的所述K个聚簇进行建立全文索引处理,获取K个全文索引。

2.根据权利要求1所述的分布式索引构建方法,其特征在于,所述对非结构化文本进行格式化和分词预处理,将预处理结果存储在分布式节点上,包括:

将各个分布式节点上不同格式的非结构化文本进行格式统一处理,获取格式一致的第一文本;

对所述第一文本进行分词处理,根据处理结果进行关键词提取,获取第一文本的关键词词汇;

采用“key=文本编号、value=文本词汇”的组合方式将所述关键词词汇存储在分布式节点上。

3.根据权利要求1所述的分布式索引构建方法,其特征在于,所述对所述预处理结果进行过滤与特征提取处理,获取处理后的文本特征向量,包括:

采用并行化计算方式对存储在所述分布节点的文本进行处理,获取所述文本内词汇的词频;

采用所述词频与第一阈值进行比较,保存所述词频大于第一阈值的词汇;

计算所述词汇的TF-IDF值,采用所述TF-IDF值与第二阈值相比较,保存TF-IDF值大于第二阈值的第二词汇;

根据所述第二词汇提取特征,并赋予所述第二词汇的权重,获取所述第二词汇的特征向量。

4.根据权利要求1所述的分布式索引构建方法,其特征在于,所述采用Canopy-Kmeans聚类算法对所述文本特征向量进行聚类处理,包括:

采用Canopy聚类方式对所述文本词汇特征向量进行初步聚类,获取以Canopy为中心的文本词汇特征向量初步聚簇;

根据所述文本词汇特征向量初步聚簇进行Kmeans聚类处理,获取所述文本词汇特征向量的K个聚簇。

5.根据权利要求1所述的分布式索引构建方法,其特征在于,所述采用索引引擎对所述分布在一个或多个分布式节点上的所述K个聚簇进行建立全文索引处理,包括:

采用索引引擎对每个分布节点上的聚簇进行处理,建立所述聚簇的全文索引;

对所有分布节点上聚簇的全文索引进行合并,获取K个全文索引。

6.一种基于文本聚类的分布式索引构建系统,其特征在于,所述系统包括:

预处理模块:用于对非结构化文本进行格式化和分词预处理,将预处理结果存储在分布式节点上;

过滤与特征提取模块:用于对所述预处理结果进行过滤与特征提取处理,获取处理后

的文本词汇特征向量；

聚类模块：用于采用Canopy-Kmeans聚类算法对所述文本词汇特征向量进行聚类处理，获取所述文本词汇特征向量的K个聚簇；

聚簇分布模块：用于将所述K个聚簇的每个聚簇分布在一个或多个分布式节点上；

索引构建模块：用于采用索引引擎对所述分布在一个或多个分布式节点上的所述K个聚簇进行建立全文索引处理，获取K个全文索引。

7.根据权利要求6所述的分布式索引构建系统，其特征在于，所述预处理模块，包括：

格式统一处理单元：用于将各个分布式节点上不同格式的非结构化文本进行格式统一处理，获取格式一致的第一文本；

分词处理与关键词提取单元：用于对所述第一文本进行分词处理，根据处理结果进行关键词提取，获取第一文本的关键词词汇；

存储单元：用于采用“key＝文本编号、value＝文本词汇”的组合方式将所述关键词词汇存储在分布式节点上。

8.根据权利要求6所述的分布式索引构建系统，其特征在于，所述过滤与特征提取模块包括：

并行化计算单元：用于采用并行化计算方式对存储在所述分布节点的文本进行处理，获取所述文本内词汇的词频；

第一比较单元：用于采用所述词频与第一阈值进行比较，保存所述词频大于第一阈值的词汇；

第二比较单元：用于计算所述词汇的TF-IDF值，采用所述TF-IDF值与第二阈值相比较，保存TF-IDF值大于第二阈值的第二词汇；

特征提取单元：用于根据所述第二词汇提取特征，并赋予所述第二词汇的权重，获取所述第二词汇的特征向量。

9.根据权利要求6所述的分布式索引构建系统，其特征在于，所述聚类模块包括：

第一聚类单元：用于采用Canopy聚类方式对所述文本词汇特征向量进行初步聚类，获取以Canopy为中心的文本词汇特征向量初步聚簇；

第二聚类单元：用于根据所述文本词汇特征向量初步聚簇进行Kmeans聚类处理，获取所述文本词汇特征向量的K个聚簇。

10.根据权利要求6所述的分布式索引构建系统，其特征在于，所述索引构建模块包括：

节点索引构建单元：用于采用索引引擎对每个分布节点上的聚簇进行处理，建立所述聚簇的全文索引；

索引合并单元：用于对所有分布节点上聚簇的全文索引进行合并，获取K个全文索引。

一种基于文本聚类的分布式索引构建方法及系统

技术领域

[0001] 本发明涉及检索索引构建技术领域，尤其涉及一种基于文本聚类的分布式索引构建方法及系统。

背景技术

[0002] 在传统的结构化信息管理中通常采用索引技术对信息进行检索，然而在分布式网络环境下，知识规模的增长速度非常快，索引文件的大小随着规模的增长而急剧增大，不仅无法用集中式方式存储索引，检索效率也严重被庞大索引库影响；针对这一情况有提出一种基于文档划分的索引方法，但是这种索引通过随机的方式对集合进行划分，由于各个划分的子集是等价的分布，因此在检索时仍然需要检索所有的子索引，导致检索的开销很大。

[0003] 文本聚类依据聚类假设：同一个类的对象有较高的相似度，不同的类的对象之间差别较大，是一种无监督的机器学习方法；文本聚类区别于文本分类，聚类不需要训练过程，也不需要预先对文档手工标注类别，即可将不同文本自动凝聚成不同的类别，具有一定的灵活性与较高的自动化处理能力。

[0004] 分布式技术主要包括分布式存储与并行计算两个基本功能；分布式存储提供一个透明一致的文件存取系统，而在物理上使用分布式的方式对海量的数据进行存储；并行计算将海量的输入数据分散于多个节点，由各个节点并行地进行计算，最后将所有节点的计算结果归并成最终的结果。

发明内容

[0005] 本发明的目的在于克服现有技术的不足，本发明提供了一种基于文本聚类的分布式索引构建方法及系统，用于构建一种用于检索的分布式索引方式，给予用户一种快速的索引方式，提高用户的使用体验感。

[0006] 为了解决上述问题，本发明提出了一种基于文本聚类的分布式索引构建方法，所述方法包括：

[0007] 对非结构化文本进行格式化和分词预处理，将预处理结果存储在分布式节点上；

[0008] 对所述预处理结果进行过滤与特征提取处理，获取处理后的文本词汇特征向量；

[0009] 采用Canopy-Kmeans聚类算法对所述文本词汇特征向量进行聚类处理，获取所述文本词汇特征向量的K个聚簇；

[0010] 将所述K个聚簇的每个聚簇分布在一个或多个分布式节点上；

[0011] 采用索引引擎对所述分布在一个或多个分布式节点上的所述K个聚簇进行建立全文索引处理，获取K个全文索引。

[0012] 优选地，所述对非结构化文本进行格式化和分词预处理，将预处理结果存储在分布式节点上，包括：

[0013] 将各个分布式节点上不同格式的非结构化文本进行格式统一处理，获取格式一致的第一文本；

[0014] 对所述第一文本进行分词处理，根据处理结果进行关键词提取，获取第一文本的关键词词汇；

[0015] 采用“key＝文本编号、value＝文本词汇”的组合方式将所述关键词词汇存储在分布式节点上。

[0016] 优选地，所述对所述预处理结果进行过滤与特征提取处理，获取处理后的文本特征向量，包括：

[0017] 采用并行化计算方式对存储在所述分布节点的文本进行处理，获取所述文本内词汇的词频；

[0018] 采用所述词频与第一阈值进行比较，保存所述词频大于第一阈值的词汇；

[0019] 计算所述词汇的TF-IDF值，采用所述TF-IDF值与第二阈值相比较，保存TF-IDF值大于第二阈值的第二词汇；

[0020] 根据所述第二词汇提取特征，并赋予所述第二词汇的权重，获取所述第二词汇的特征向量。

[0021] 优选地，所述采用Canopy-Kmeans聚类算法对所述文本特征向量进行聚类处理，包括：

[0022] 采用Canopy聚类方式对所述文本词汇特征向量进行初步聚类，获取以Canopy为中心的文本词汇特征向量初步聚簇；

[0023] 根据所述文本词汇特征向量初步聚簇进行Kmeans聚类处理，获取所述文本词汇特征向量的K个聚簇。

[0024] 优选地，所述采用索引引擎对所述分布在一个或多个分布式节点上的所述K个聚簇进行建立全文索引处理，包括：

[0025] 采用索引引擎对每个分布节点上的聚簇进行处理，建立所述聚簇的全文索引；

[0026] 对所有分布节点上聚簇的全文索引进行合并，获取K个全文索引。

[0027] 相应地，本发明还提供了一种基于文本聚类的分布式索引构建系统，所述系统包括：

[0028] 预处理模块：用于对非结构化文本进行格式化和分词预处理，将预处理结果存储在分布式节点上；

[0029] 过滤与特征提取模块：用于对所述预处理结果进行过滤与特征提取处理，获取处理后的文本词汇特征向量；

[0030] 聚类模块：用于采用Canopy-Kmeans聚类算法对所述文本词汇特征向量进行聚类处理，获取所述文本词汇特征向量的K个聚簇；

[0031] 聚簇分布模块：用于将所述K个聚簇的每个聚簇分布在一个或多个分布式节点上；

[0032] 索引构建模块：用于采用索引引擎对所述分布在一个或多个分布式节点上的所述K个聚簇进行建立全文索引处理，获取K个全文索引。

[0033] 优选地，所述预处理模块，包括：

[0034] 格式统一处理单元：用于将各个分布式节点上不同格式的非结构化文本进行格式统一处理，获取格式一致的第一文本；

[0035] 分词处理与关键词提取单元：用于对所述第一文本进行分词处理，根据处理结果进行关键词提取，获取第一文本的关键词词汇；

[0036] 存储单元:用于采用“key＝文本编号、value＝文本词汇”的组合方式将所述关键词词汇存储在分布式节点上。

[0037] 优选地,所述过滤与特征提取模块包括:

[0038] 并行化计算单元:用于采用并行化计算方式对存储在所述分布节点的文本进行处理,获取所述文本内词汇的词频;

[0039] 第一比较单元:用于采用所述词频与第一阈值进行比较,保存所述词频大于第一阈值的词汇;

[0040] 第二比较单元:用于计算所述词汇的TF-IDF值,采用所述TF-IDF值与第二阈值相比较,保存TF-IDF值大于第二阈值的第二词汇;

[0041] 特征提取单元:用于根据所述第二词汇提取特征,并赋予所述第二词汇的权重,获取所述第二词汇的特征向量。

[0042] 优选地,所述聚类模块包括:

[0043] 第一聚类单元:用于采用Canopy聚类方式对所述文本词汇特征向量进行初步聚类,获取以Canopy为中心的文本词汇特征向量初步聚簇;

[0044] 第二聚类单元:用于根据所述文本词汇特征向量初步聚簇进行Kmeans聚类处理,获取所述文本词汇特征向量的K个聚簇。

[0045] 优选地,所述索引构建模块包括:

[0046] 节点索引构建单元:用于采用索引引擎对每个分布节点上的聚簇进行处理,建立所述聚簇的全文索引;

[0047] 索引合并单元:用于对所有分布节点上聚簇的全文索引进行合并,获取K个全文索引。

[0048] 在本发明实施过程中,通过对文本进行格式化、分词、过滤、特征提取和聚类处理,并将处理结果建立全文索引,用于构建一种用于检索的分布式索引方式,给予用户一种快速的索引方式,提高用户的使用体验感。

附图说明

[0049] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述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的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其它的附图。

[0050] 图1是本发明实施例的基于文本聚类的分布式索引构建方法的流程示意图;

[0051] 图2是本发明实施例的预处理步骤的流程示意图;

[0052] 图3是本发明实施例的文本特征向量获取步骤的流程示意图;

[0053] 图4是本发明实施例的基于文本聚类的分布式索引构建系统的结构组成示意图;

[0054] 图5是本发明实施例的预处理模块的结构组成示意图;

[0055] 图6是本发明实施例的过滤与特征提取模块的结构组成示意图。

具体实施方式

[0056] 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

整地描述，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作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。

[0057] 图1是本发明实施例的基于文本聚类的分布式索引构建方法的流程示意图，如图1所示，该方法包括：

[0058] S11：对非结构化文本进行格式化和分词预处理，将预处理结果存储在分布式节点上；

[0059] S124：对该预处理结果进行过滤与特征提取处理，获取处理后的文本词汇特征向量；

[0060] S13：采用Canopy-Kmeans聚类算法对该文本词汇特征向量进行聚类处理，获取该文本词汇特征向量的K个聚簇；

[0061] S14：将该K个聚簇的每个聚簇分布在一个或多个分布式节点上；

[0062] S15：采用索引引擎对该分布在一个或多个分布式节点上的K个聚簇进行建立全文索引处理，获取K个全文索引。

[0063] 对S11作进一步说明：

[0064] 在数据库中的文本存在结构不一致的问题，对非结构化文本进行格式化处理得到格式统一的结构化文本，再对文本进行分词预处理，然后获取分词结果，并将该结果存储在分布式节点上。

[0065] 进一步的，图2是本发明实施例的预处理步骤的流程示意图，如图2所示，该步骤包括：

[0066] S111：将各个分布式节点上不同格式的非结构化文本进行格式统一处理，获取格式一致的第一文本；

[0067] S112：对该第一文本进行分词处理，根据处理结果进行关键词提取，获取第一文本的关键词词汇；

[0068] S113：采用“key＝文本编号、value＝文本词汇”的组合方式将所述关键词词汇存储在分布式节点上。

[0069] 对S111作进一步说明：

[0070] 将分布在分布式节点上的各类不同格式的非结构化文本进行格式统一处理，从而获取到格式统一的第一文本。

[0071] 对S112作进一步说明：

[0072] 对第一文本进行分词处理，对第一文本中分离处理的词汇进行提取，将提取出来的词汇作为关键词，从而获取第一文本的关键词词汇。

[0073] 对S113作进一步说明：

[0074] 根据各文本和该文本提取出来的关键词词汇采用采用“key＝文本编号、value＝文本词汇”的组合方式进行组合，再将组合好的key和value存储在分布式节点上。

[0075] 对S12作进一步说明：

[0076] 根据上述步骤处理的结果的文本和词汇，计算该文本内的词汇的词频，采用该词频与第一阈值相比较，保存词频大于第一阈值的词汇，然后再次计算该词汇的TF-IDF值，采用TF-IDF值与第二阈值相比较，保存TF-IDF值大于第二阈值的第二词汇，让后根据TF-IDF

值赋予剩下的第二词汇权重，提取该第二词汇的特征向量。

[0077] 进一步的，图3是本发明实施例的文本特征向量获取步骤的流程示意图，如图3所示，该步骤包括：

[0078] S121：采用并行化计算方式对存储在该分布节点的文本进行处理，获取该文本内词汇的词频；

[0079] S122：比较该词频是否大于第一阈值，若是则跳到S123，若不是，则去除该词频对应的词汇；

[0080] S123：保存所述词频大于第一阈值的词汇，计算所述词汇的TF-IDF值；

[0081] S124：比较该TF-IDF值是否大于第二阈值，若是，则跳到S125，若否，则去除该TF-IDF值对应的词汇；

[0082] S125：保存TF-IDF值大于第二阈值的第二词汇；

[0083] S126：根据第二词汇提取特征，并赋予第二词汇的权重，获取该第二词汇的特征向量。

[0084] 对S121作进一步说明：

[0085] 词频是指一个词汇在文本中出现的频率，词汇t在文本d中的频率tf(t,d)＝count(t IN d)/count(t)，即词汇出现次数与文本词汇总量的比值；通过上述的公式对文本进行处理，计算获取到文本内词汇的词频。

[0086] 对S122作进一步说明：

[0087] 在同一文本中，词汇出现的次数越多，该词汇就是越关键，而词频比较低的词汇一般不具有表示文本的能力，因此，设置第一阈值，采用词频与第一阈值进行比较，去除词频比第一阈值小的词汇，保存词频比第一阈值大的词汇；该第一阈值根据实际情况赋予，在本实施例中该第一阈值设定为0.01。

[0088] 对S123作进一步说明：

[0089] 在得到与第一阈值比较后剩下的词汇之后，计算该词汇的TF-IDF值。

[0090] 对S124作进一步说明：

[0091] 采用TF-IDF值与第二阈值进行比较，去除TF-IDF值比第二阈值小的词汇，保存TF-IDF值比第二阈值大的词汇；该第二阈值设定根据实际情况而定，在本实施例中该第二阈值设定为0.01。

[0092] 对S125作进一步说明：

[0093] 存TF-IDF值大于第二阈值的第二词汇。

[0094] 对S126作进一步说明：

[0095] TF-IDF对词汇的t在文本d中的权重w的计算公式为：

[0096] w(t,d)＝TF(t,d)*log(1/DF(t))；

[0097] 其中，DF(t)为文本频率，指某一词汇t的文本比重，词汇t的文本频率DF为DF(t)＝n(t)/n，即含有词汇t的文本数量与文本总数的比值；词汇频率TF(t,d)为词汇t在文本d中的词频。

[0098] 若某一个词汇在一个文本中的出现频率较高，而在其他文本中的出现频率较少，则可以认为这个词汇具有很好的类区别能力，合适用来表示文本，可以进一步提取特征向量。

[0099] 采用向量空间模型VSM来表示文本，对含有n个特征项的文本d(t_1,t_2,…,t_n)，每个特征项t_k被赋予TF-IDF计算得到的权重w_k，表示该特征在文本中的重要程度，即该文本可以采用特征向量d(w_1,w_2,…,w_n)表示，w_k为特征项t_k的TF-IDF权重，根据该特征项t_k的TF-IDF权重赋予对应的词汇权重。

[0100] 对S13作进一步说明：

[0101] 首先，采用Canopy聚类方式对该文本词汇特征向量进行初步聚类，获取以Canopy为中心的文本词汇特征向量初步聚簇；然后，根据该文本词汇特征向量初步聚簇进行Kmeans聚类处理，获取该文本词汇特征向量的K个聚簇。

[0102] 进一步的，Canopy聚类算法具有简单、快速和精确的特性，在处理海量的高维数时，尤其是数据量巨大的情况下，使用Canopy聚类进行初步处理，可以有效提高效率，Canopy聚类算法具体如下：

[0103] (1)将特征向量集合初始化为list，选择两个距离阈值:T1、T2。

[0104] (2)随机取list中的一个对象d作为Canopy中心，标记为c，并将d从list中删除；

[0105] (3)计算list中所有对象d_i与c的距离distance，如果distance<T1，将该对象加入Canopy c；如果distanc<T2，将该点从list中删除，也就是该对象无法作为Canopy中心；

[0106] (4)将剩下的c加入canopylist中；

[0107] (5)重复步骤2、3、4，直至list中数据为空结束，canopylist则为最后Canopy聚类结果。

[0108] 其中，考虑到由于文本词汇特征向量的高维性，因此采用余弦距离度量；

[0109] 具体的，特征向量A与特征向量B之间的余弦距离计算公式具体为：

[0110] $$Cosine\_distance(A,B) = 1 - \sum_{i=1}^{n}(a_i \times b_i) / (\sqrt{\sum_{i=1}^{n} a_i^2} \times \sqrt{\sum_{i=1}^{n} b_i^2});$$

[0111] 其中特征向量A表示为A＝(a_1,a_2,…,a_n)，特征向量B表示为B＝(b_1,b_2,…,b_n)，i＝1,2,…,n。

[0112] 再采用Kmeans聚类算法对初步聚类处理结果进行聚类处理，Kmeans聚类算法的基本思想为:以空间中k个对象做为中心进行归类，把对象空间中最靠近各个中心的对象分别归为一类，通过多次迭代的方式，将各聚类质心的值逐次计算更新，直至聚簇质心稳定不变。

[0113] 针对本发明实施例，将原来的Kmeans聚类算法进行算的修改，修改后的算法具体如下：

[0114] (1)将Canopy聚类算法的结果作为Kmeans聚类算法的输入，即Canopy聚类算法产生的Canopy中心作为Kmeans算法的初始化质心，并且各个特征向量已经分配到相应的质心中；

[0115] (2)对每个特征向量计算该特征向量到每个质心的距离，并将其分配到最近的聚类质心，其中距离计算公式仍然采用Canopy聚类算法中使用的余弦距离；

[0116] (3)对每个聚类重新计算均值得到新的聚类质心；

[0117] (4)计算所有数据对象到其对应聚类质心的方差误差值E，若E大于阈值则重复步骤2及步骤3，否则聚类结束。

[0118] 其中，E的计算公式具体为：

[0119] $$E = \frac{1}{n}\sum_{x}\left\|x - u_{k(x)}\right\|^2;$$

[0120] 其中，x为文档的文本向量；k(x)表示向量x所在的聚簇；$u_{k(x)}$表示向量x所在的聚簇的质心向量；n为文档向量数目。

[0121] 并行优化设计：同样先在每个节点上进行局部的kmeans聚类算法：对每个节点上的向量局部计算该向量到每个全局质心的距离，并将其分配到最近的全局质心得到全局聚类；对节点上的局部聚类计算均值得到局部质心以及局部方差误差值；将所有节点上的局部质心及局部误差方差值整合成全局质心以及总的误差方差值E，再根据E决定是否继续迭代或者结束聚合，最终得到K个聚类及其质心；

[0122] 全局质心计算公式为：

[0123] $v_i=(v_i[1]*m_1+\cdots+v_i[j]*m_j+\cdots+v_i[s]*m_s)/(m_1+\cdots+m_s)$

[0124] 其中，v_i为计算出的第i个聚类的全局质心向量；

[0125] $v_i[j]$为在有第j个聚类的分布式节点S上的局部质心向量，m_s为该聚类中的表示文档的向量的个数；全局方差误差值计算公式E为：$E=(E_1*n_1+\cdots+E_j*n_j+\cdots+E_t*n_t)/(n_1+\cdots+n_t)$；$E_j$为第j个节点的方差误差值；$n_j$为该节点上的向量总数；t为节点总数。

[0126] 对S14作进一步说明：

[0127] 将上述步骤中获取的K个聚簇的每个聚簇分布在一个或多个分布式节点上。

[0128] 对S15作进一步说明：

[0129] 采用索引引擎对每个分布节点上的聚簇进行处理，建立该聚簇的全文索引；对所有分布节点上聚簇的全文索引进行合并，获取K个全文索引。

[0130] 进一步的，根据具体的索引引擎，对每个分布节点上的聚簇进行全文索引建立，并且将所有节点上的相同聚类的聚簇索引进行合并，即可得到K个聚簇的全局全文索引。

[0131] 以下是本发明实施例中，用户在使用检索关键词进行检索的过程：

[0132] 对输入的查询字符串进行分词提取关键词处理，再根据索引选择算法计算出查询与子集合的相似度，选择出符合一定条件的索引。

[0133] 其中给出一种基于查询空间的索引选择算法，描述如下：

[0134] 定义系统内部的查询空间$P=\{p_1,p_2,\cdots,p_i\}$，p_i表示历史的一次查询记录；聚类索引库为$S=\{S_1,S_2,\cdots,S_j\}$；$rel(q|S_j)$表示索引库S_j与当前查询q的相关程度；

[0135] 算法步骤为：

[0136] (1)计算每个索引库与历史查询p_i的相关度$rel(p_i|S_j)$；如果S_j不在$SET(p_i)$中，则$rel(p_i|S_j)=0$；否则相关度$rel(p_i|S_j)$计算公式具体如下为：

[0137] $$rel(p_i|S_j)=\sum_{T}\frac{rel(p_i|doc)}{T};$$

[0138] 其中，$rel(p_i|doc)$是指历史查询与文档的相关度，当文档属于聚类S_j时相关度$rel(p_i|doc)=1$，否则相关度$rel(pi|doc)=0$；T是预定义值，指在评分列表中需要被考虑的前T文档数目，在本发明实施例中T设为20，即选择相关度排名在前20的文档；

[0139] (2)选择最相似的k个历史查询，采用余弦距离度量计算输入的查询q与历史查询的相似度$sim(q|p_i)$，选择相似度较高的k个查询，可根据实验测试获得最佳效果k取值；

[0140] (3)根据相似查询的相关信息计算当前查询q与索引库S_i的相关度rel(q|S_j),根据相关度rel(q|S_j)排序,选择较相关的索引库;

[0141] 当前查询q与检索库S_i的相关度rel(q|S_j)的计算公式具体为:

[0142] $$rel(q|S_j) = \sum_k rel(p_i|S_j) \times sim(q|p_i);$$

[0143] rel(p_i|s_j)表示索引库S_j与历史查询p_i的相关度;sim(q|p_i)表示当前查询q与历史查询p_i的相关度;k表示与当前查询q最相似的前k个历史查询;

[0144] (4)在处理完成查询之后,系统采集用户的反馈信息,如用户实际点击的链接等信息,最后添加此次查询至查询空间,更新查询库,从而完成一次查询。

[0145] 在符合条件的索引上进行检索,通过用全局的文档频率等信息计算得分对各个索引的检索结果进行合并以及排序,得到最终检索结果,完成对查询的检索;给出对于查询q检索结果d的评分Score(q,d)的计算依据为:

[0146] $$Score(q,d) = coord(q,d) \times queryNorm(q) \times \sum_i^q (TF(t,d) \times IDF^2(t) \times t.getBoost() \times norm(t,d));$$

[0147] 其中,t为查询q中提取的各个关键词;TF(t,d)为t在文档d中的词频,IDF(t)为逆文档频率;t.getBoost()为查询输入中对关键词设置的重要程度;norm(t,d)为建立索引时设定的文档的加权和长度因子;coord(q,d)为评分因子,文档出现查询项次数越多匹配程度越高;queryNorm(q)将查询语言归一化,使不同的查询语言直接进行比较。

[0148] 相应地,图4是本发明实施例的基于文本聚类的分布式索引构建系统的结构组成示意图,如图4所示,该系统包括:

[0149] 预处理模块11:用于对非结构化文本进行格式化和分词预处理,将预处理结果存储在分布式节点上;

[0150] 过滤与特征提取模块12:用于对该预处理结果进行过滤与特征提取处理,获取处理后的文本词汇特征向量;

[0151] 聚类模块13:用于采用Canopy-Kmeans聚类算法对该文本词汇特征向量进行聚类处理,获取该文本词汇特征向量的K个聚簇;

[0152] 聚簇分布模块14:用于将该K个聚簇的每个聚簇分布在一个或多个分布式节点上;

[0153] 索引构建模块15:用于采用索引引擎对该分布在一个或多个分布式节点上的该K个聚簇进行建立全文索引处理,获取K个全文索引。

[0154] 优选地,图5是本发明实施例的预处理模块的结构组成示意图,如图5所示,该预处理模块11,包括:

[0155] 格式统一处理单元111:用于将各个分布式节点上不同格式的非结构化文本进行格式统一处理,获取格式一致的第一文本;

[0156] 分词处理与关键词提取单元112:用于对该第一文本进行分词处理,根据处理结果进行关键词提取,获取第一文本的关键词词汇;

[0157] 存储单元113:用于采用“key=文本编号、value=文本词汇”的组合方式将该关键词词汇存储在分布式节点上。

[0158] 优选地,图6是本发明实施例的过滤与特征提取模块的结构组成示意图,如图6所

示，该过滤与特征提取模块12包括：

[0159] 并行化计算单元121：用于采用并行化计算方式对存储在该分布节点的文本进行处理，获取该文本内词汇的词频；

[0160] 第一比较单元122：用于采用该词频与第一阈值进行比较，保存该词频大于第一阈值的词汇；

[0161] 第二比较单元123：用于计算该词汇的TF-IDF值，采用该TF-IDF值与第二阈值相比较，保存TF-IDF值大于第二阈值的第二词汇；

[0162] 特征提取单元124：用于根据该第二词汇提取特征，并赋予该第二词汇的权重，获取该第二词汇的特征向量。

[0163] 优选地，该聚类模块13包括：

[0164] 第一聚类单元：用于采用Canopy聚类方式对该文本词汇特征向量进行初步聚类，获取以Canopy为中心的文本词汇特征向量初步聚簇；

[0165] 第二聚类单元：用于根据该文本词汇特征向量初步聚簇进行Kmeans聚类处理，获取该文本词汇特征向量的K个聚簇。

[0166] 优选地，该索引构建模块15包括：

[0167] 节点索引构建单元：用于采用索引引擎对每个分布节点上的聚簇进行处理，建立该聚簇的全文索引；

[0168] 索引合并单元：用于对所有分布节点上聚簇的全文索引进行合并，获取K个全文索引。

[0169] 具体地，本发明实施例的系统相关功能模块工作原理可参考方法实施例的相关描述，这里不再赘述。

[0170] 在本发明实施过程中，通过对文本进行格式化、分词、过滤、特征提取和聚类处理，并将处理结果建立全文索引，用于构建一种用于检索的分布式索引方式，给予用户一种快速的索引方式，提高用户的使用体验感。

[0171] 本领域普通技术人员可以理解上述实施例的各种方法中的全部或部分步骤是可以通过程序来指令相关的硬件来完成，该程序可以存储于一计算机可读存储介质中，存储介质可以包括：只读存储器(ROM，Read Only Memory)、随机存取存储器(RAM，Random Access Memory)、磁盘或光盘等。

[0172] 另外，以上对本发明实施例所提供的一种基于文本聚类的分布式索引构建方法及系统进行了详细介绍，本文中应用了具体个例对本发明的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本发明的方法及其核心思想；同时，对于本领域的一般技术人员，依据本发明的思想，在具体实施方式及应用范围上均会有改变之处，综上所述，本说明书内容不应理解为对本发明的限制。

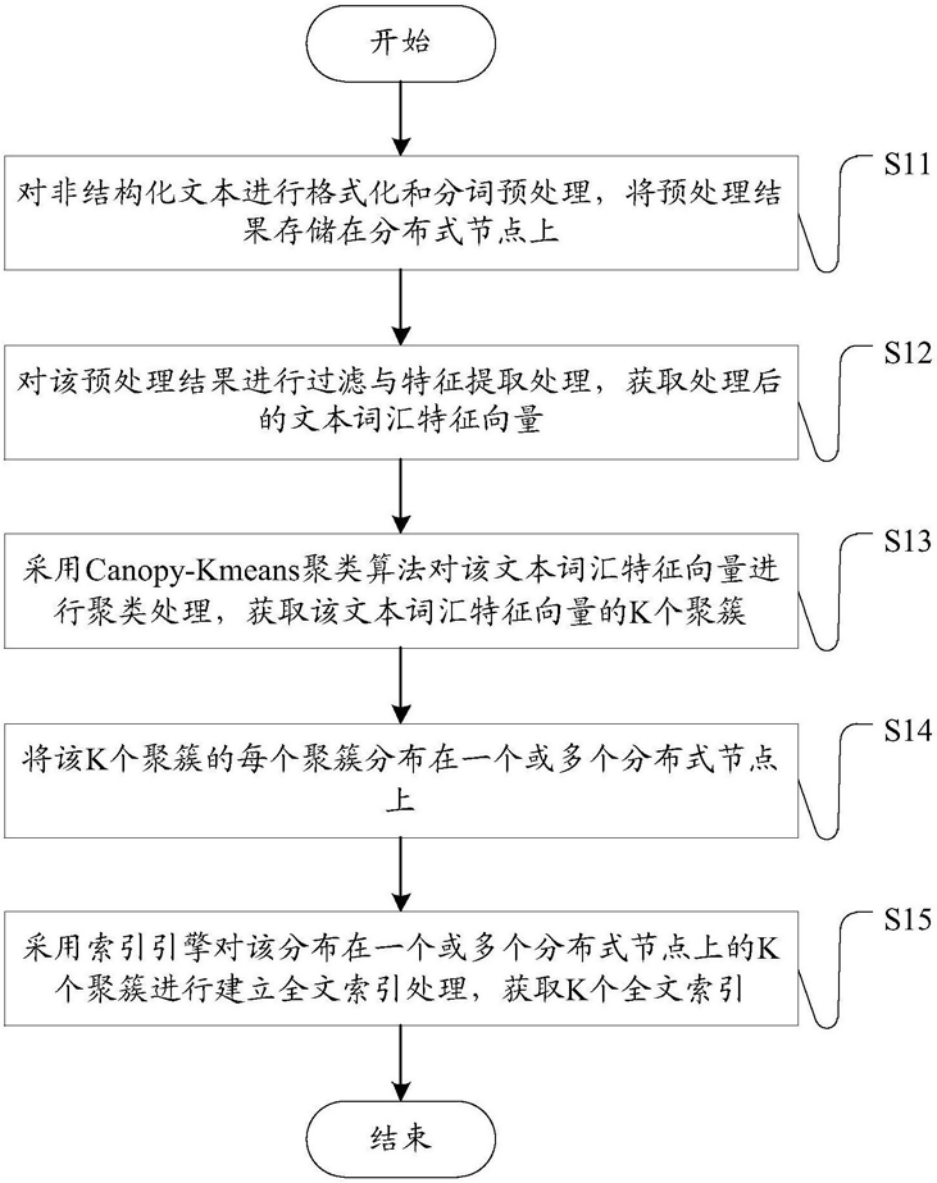

开始
对非结构化文本进行格式化和分词预处理，将预处理结果存储在分布式节点上
S11
对该预处理结果进行过滤与特征提取处理，获取处理后的文本词汇特征向量
S12
采用Canopy-Kmeans聚类算法对该文本词汇特征向量进行聚类处理，获取该文本词汇特征向量的K个聚簇
S13
将该K个聚簇的每个聚簇分布在一个或多个分布式节点上
S14
采用索引引擎对该分布在一个或多个分布式节点上的K个聚簇进行建立全文索引处理，获取K个全文索引
S15
结束

图1

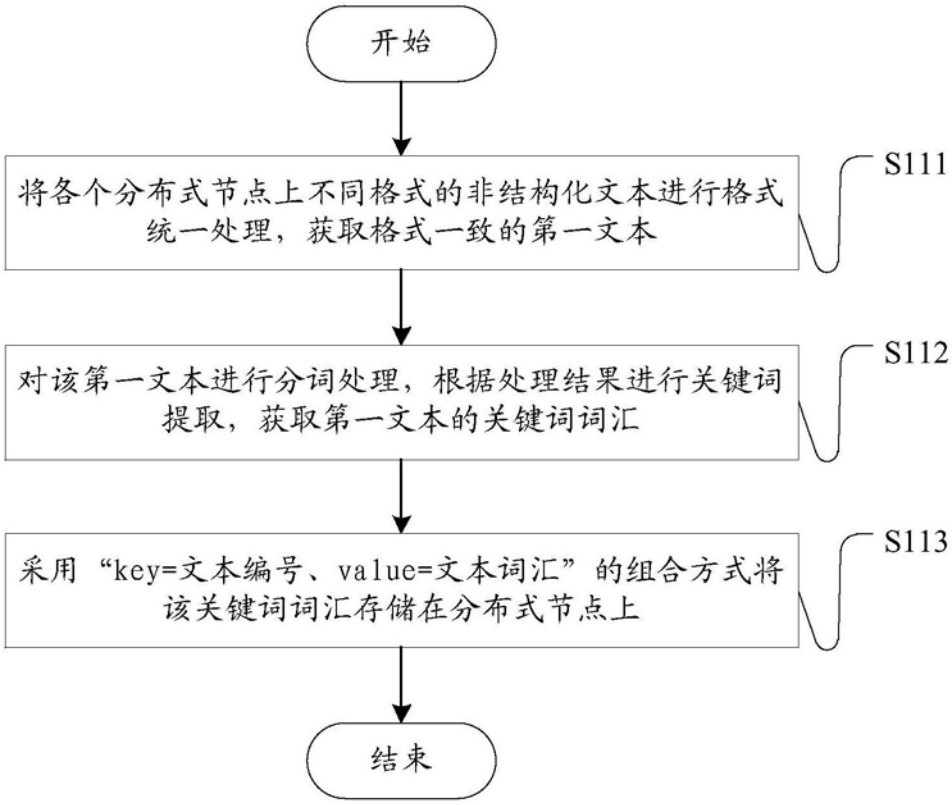

开始
将各个分布式节点上不同格式的非结构化文本进行格式统一处理，获取格式一致的第一文本
S111
对该第一文本进行分词处理，根据处理结果进行关键词提取，获取第一文本的关键词词汇
S112
采用“key=文本编号、value=文本词汇”的组合方式将该关键词词汇存储在分布式节点上
S113
结束

图2

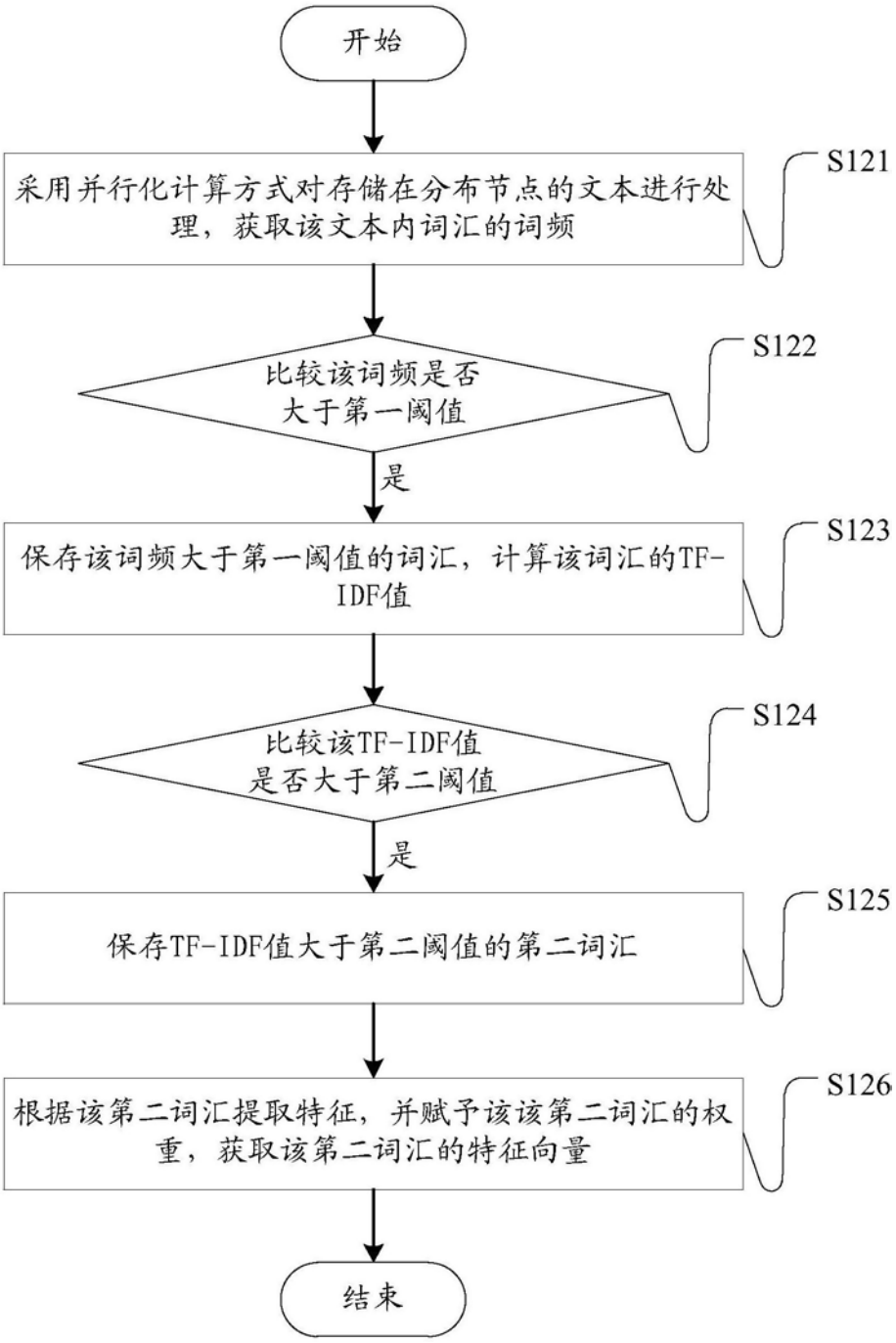

开始
采用并行化计算方式对存储在分布节点的文本进行处理，获取该文本内词汇的词频
S121
比较该词频是否大于第一阈值
S122
是
保存该词频大于第一阈值的词汇，计算该词汇的TF-IDF值
S123
比较该TF-IDF值是否大于第二阈值
S124
是
保存TF-IDF值大于第二阈值的第二词汇
S125
根据该第二词汇提取特征，并赋予该该第二词汇的权重，获取该第二词汇的特征向量
S126
结束

图3

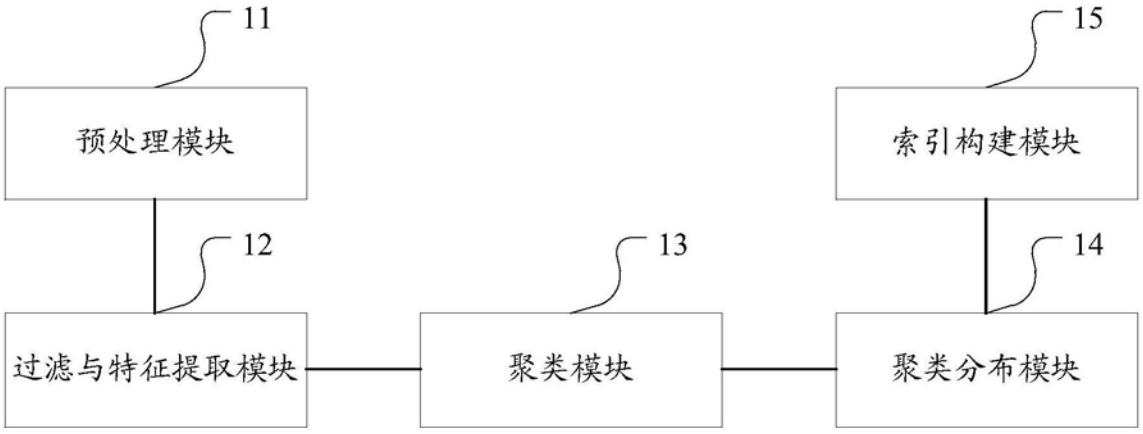

11
预处理模块
15
索引构建模块
12
过滤与特征提取模块
13
聚类模块
14
聚类分布模块

图4

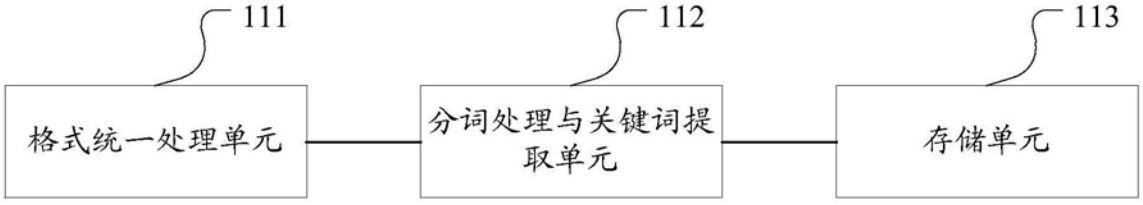

111
格式统一处理单元
112
分词处理与关键词提取单元
113
存储单元

图5

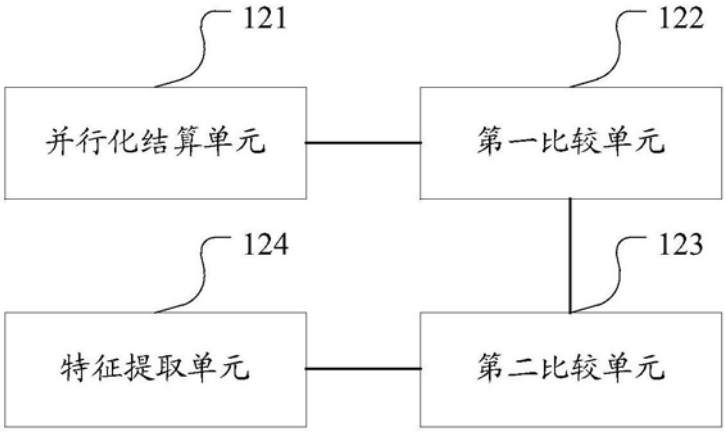

121
并行化结算单元
122
第一比较单元
124
特征提取单元
123
第二比较单元

图6