(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 107291895 B
(45)授权公告日 2020.05.26

(21)申请号 201710476244.0

(22)申请日 2017.06.21

(65)同一申请的已公布的文献号
申请公布号 CN 107291895 A

(43)申请公布日 2017.10.24

(73)专利权人 浙江大学
地址 310058 浙江省杭州市西湖区余杭塘路866号

(72)发明人 陈珂 王伟迪 胡天磊 陈刚 伍赛 寿黎但

(74)专利代理机构 杭州求是专利事务所有限公司 33200
代理人 林超

(51)Int.Cl.
G06F 16/21(2019.01)
G06F 16/31(2019.01)
G06F 16/335(2019.01)
G06F 16/2458(2019.01)
G06F 40/284(2020.01)

(56)对比文件
US 2010179933 A1,2010.07.15,
CN 104834735 A,2015.08.12,
CN 104317838 A,2015.01.28,
CN 104866471 A,2015.08.26,
CN 103631928 A,2014.03.12,
CN 101281520 A,2008.10.08,

审查员 倪大建

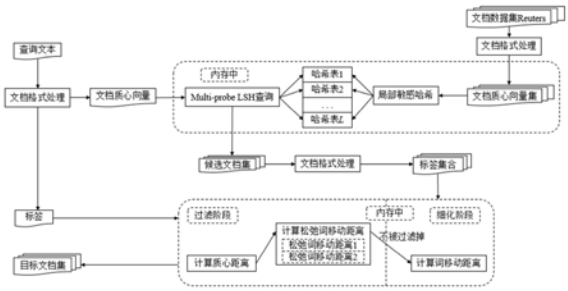
权利要求书3页 说明书8页 附图3页

(54)发明名称

一种快速的层次化文档查询方法

(57)摘要

本发明公开了一种快速的层次化文档查询方法。对文档集中的每个文档建立数据模型，对文档进行格式化处理获得文档质心向量和文档标签；生成的文档质心向量作为高维向量空间中的一个点，为每个文档集采用局部敏感哈希方法在内存中构建哈希索引结构；依据查询文本的文档质心向量，采用基于局部敏感哈希思想的查询方法在哈希索引结构中获取一个候选文档集；依据查询文本的文档标签，采用过滤-细化的层次化框架在候选文档集中获取词移动距离度量下的最近邻文档。本发明设计的层次化查询方法应用于文档分类和检索时在效率和效果上获得了良好的平衡，使得用户在进行词移动距离度量下的文档查询时能够在保证准确性的情况下快速地获取目标文档。



CN 107291895 B

1.一种快速的层次化文档查询方法,其特征在于:

所述方法的步骤如下:

1)对文档集中的每个文档建立数据模型,一篇文档的数据模型主要由单词、词向量和单词权重的三部分组成;

2)对文档进行格式化处理获得文档质心向量和文档标签;

所述步骤2)具体为:通过计算词向量对单词权重的加权均值得到文档的文档质心向量,通过对单词权重进行归一化处理并与对应的词向量一起构成文档的文档标签,将文档质心向量与文档标签存储于内存的特定数据结构中;

3)步骤2)中生成的文档质心向量作为高维向量空间中的一个点,为每个文档集采用局部敏感哈希方法在内存中构建哈希索引结构;

4)依据查询文本的文档质心向量,采用基于局部敏感哈希思想的查询方法在哈希索引结构中获取一个候选文档集;

5)依据查询文本的文档标签,采用过滤-细化框架在候选文档集中获取词移动距离度量下的k个最近邻文档,完成查询;

所述步骤5)具体是先新建空白的目标文档集并以堆数据结构形式存储,以堆数据结构形式存储的目标文档集具有按词移动距离从小到大排序的特性,然后对于候选文档集中所有候选文档,以每次一个文档采用以下方式处理,包括过滤阶段和细化阶段:

5.1)若目标文档集中的文档数量小于等于k,k表示预先设定的查询文档数值,将候选文档依次加入目标文档集;

若目标文档集中的文档数量大于k,则进行以下过滤阶段的步骤;

5.2)过滤阶段

计算查询文本与候选文档之间词移动距离的三个下界值,三个下界值分别为文档质心距离和两个松弛词移动距离,用文档质心距离、两个松弛词移动距离来判断候选文档是否剔除;

若三个下界值中的任何一个大于阈值θ,阈值θ表示查询文本与目标文档集中第k篇文档之间的词移动距离,则将候选文档从目标文档集剔除,返回步骤5.1)对下一个候选文档进行处理;

若三个下界值全部不大于阈值θ,则进入以下细化阶段的步骤;

5.3)过滤阶段

计算查询文本与目标文档集中各个文档之间的词移动距离,在目标文档集的堆数据结构中按词移动距离从小到达排序,将词移动距离最大对应的文档剔除,直到目标文档集中的文档数量保持在k个,并更新阈值θ与目标文档集的堆数据结构,再返回步骤5.1)对下一个候选文档进行处理。

2.根据权利要求1所述的一种快速的层次化文档查询方法,其特征在于:

所述步骤2)具体是将一个文档标签以一个结构体数组形式存储,将一个文档质心向量以一个数组形式存储。

3.根据权利要求1所述的一种快速的层次化文档查询方法,其特征在于:

所述的候选文档集中包含文档的数量为mk,k表示最近邻文档的数量,m表示倍数取值。

4.根据权利要求1所述的一种快速的层次化文档查询方法,其特征在于:

所述步骤3)中为每个文档集采用局部敏感哈希方法在内存中构建哈希索引结构是采用局部敏感哈希函数簇方式构建多个哈希表,具体为:

3.1)对于文档集中每个文档,采用以下公式生成多个局部敏感哈希函数:

$$h(o)=\left\lfloor\frac{a\cdot o+bW}{W}\right\rfloor$$

其中,h(o)表示局部敏感哈希函数,⌊ ⌋表示取整数下限;a表示一个随机数向量,a的维数和文档质心向量的维数相同,a每一维的随机数是基于高斯分布独立选取;o是高维向量空间中的文档质心向量,b是随机参数,是基于数值范围[0,1]的均匀分布进行选取,W表示局部敏感哈希函数宽度的实数;

3.2)将得到的所有局部敏感哈希函数复合构成一个复合函数,并作为一个哈希表,文档集中每一个文档根据复合函数映射到哈希表的哈希桶中,

3.3)重复步骤3.1)和3.2)构建获得多个哈希表。

5.根据权利要求1所述的一种快速的层次化文档查询方法,其特征在于:

所述步骤4)中查询文本的文档质心向量和所述步骤5)中查询文本的文档标签是以查询文本作为文档进行步骤1)和2),获得查询文本的文档质心向量和文档标签。

6.根据权利要求1所述的一种快速的层次化文档查询方法,其特征在于:

所述步骤4)中的基于局部敏感哈希思想的查询方法为Multi-probe LSH查询方法。

7.根据权利要求1所述的一种快速的层次化文档查询方法,其特征在于:

所述步骤5.2)中,文档质心距离和松弛词移动距离分别采用以下方式计算:

a)文档质心距离计算公式为:

$\|Xd_1-Xd_2\|$

其中,Xd_1与Xd_2分别表示查询文本和候选文档的文档质心向量;

b)第一个松弛词移动距离形式化为以下公式所示的线性优化问题,即采用以下公式计算;

$$min_{T\geq 0}\sum_{i,j=1}^{n}T_{ij}c(i,j)$$

$$s.t.\sum_{i=1}^{n}T_{ij}=w_{2j}\quad\forall j\in\{1,\dots,n\}$$

其中,$min_{T\geq 0}\sum_{i,j=1}^{n}T_{ij}c(i,j)$为松弛词移动距离;c(i,j)表示单词i与j之间的移动代价,等于Word2Vec词向量间的欧式距离,单词i属于查询文本,单词j属于文档;w_{2j}表示文档标签中归一化处理后的单词权重;T表示流动权重矩阵,T_{ij}表示单词i与单词j之间的流动权重值;n表示一个文档中单词的总数;

单词i与单词j之间的流动权重值T_{ij}采用以下公式计算:

$$T_{ij}=\begin{cases}w_{2j} & if\ i=argmin_i c(i,j)\\ 0 & otherwise\end{cases}$$

b)第二个松弛词移动距离采用以下公式计算;

$$min_{T\geq 0}\sum_{i,j=1}^{n}T_{ij}c(i,j)$$

$$s.t.\sum_{j=1}^{n}T_{ij}=w_{1i}\quad \forall i\in\{1,\dots,m\}$$

其中，$min_{T\geq 0}\sum_{i,j=1}^{n}T_{ij}c(i,j)$为松弛词移动距离；c(i,j)表示单词i与j之间的移动代价，等于Word2Vec词向量间的欧式距离，单词i属于查询文本，单词j属于文档；w_{1i}表示查询文本标签中归一化处理后的单词权重；T表示流动权重矩阵，T_{ij}表示单词i与单词j之间的流动权重值；m表示一个查询文本中单词的总数；

单词i与单词j之间的流动权重值T_{ij}采用以下公式计算：

$$T_{ij}=\begin{cases}w_{1i} & if\ j=argmin_j c(i,j)\\ 0 & otherwise\end{cases}。$$

8.根据权利要求1所述的一种快速的层次化文档查询方法，其特征在于：

所述步骤5.3)中，词移动距离采用以下方式计算：

$$min_{T\geq 0}\sum_{i,j=1}^{n}T_{ij}c(i,j)$$

$$s.t.\sum_{i=1}^{n}T_{ij}=w_{2j}\quad \forall j\in\{1,\dots,n\}$$

$$\sum_{j=1}^{n}T_{ij}=w_{1i}\quad \forall j\in\{1,\dots,m\}$$

其中，$min_{T\geq 0}\sum_{i,j=1}^{n}T_{ij}c(i,j)$为词移动距离；c(i,j)表示单词i与j之间的移动代价，等于Word2Vec词向量间的欧式距离，单词i属于查询文本，单词j属于文档；w_{1i}表示查询文本标签中归一化处理后的单词权重，w_{2j}表示文档标签中归一化处理后的单词权重；T表示流动权重矩阵作为最优解的最优矩阵，T_{ij}表示单词i与单词j之间的流动权重值；n表示一个文档中单词的总数；m表示一个查询文本中单词的总数；

单词i与单词j之间的流动权重值T_{ij}采用以下公式计算：

$$T_{ij}=\begin{cases}w_{1i} & if\ j=argmin_j c(i,j)\\ 0 & otherwise\end{cases}。$$

一种快速的层次化文档查询方法

技术领域

[0001] 本发明涉及了一种快速的层次化文档查询方法，具体涉及到了机器学习领域中的Word2Vec模型、数据库领域中的局部敏感哈希方法以及推土机距离度量下的过滤-细化框架。

背景技术

[0002] 随着信息技术的发展，人们生产、收集和存储信息的能力不断增强。其中一种主要的信息载体是文档，精确地表示两篇文档之间的相似性在文档检索、文档分类和文档聚类等方向具有广泛的应用。潜在语义分析方法是通过矩阵分解来抽取低维语义信息，主题模型是对文字中隐含主题进行建模的方法。最近，随着深度学习的发展，Word2Vec模型与Doc2Vec模型相继被提出，它们分别是在大规模语料库上学习单词向量与文档向量的表示方式，该表示在一定程度上包含了单词或文档的语义信息。为了有效地使用词向量，词移动距离被提出；遗憾的是，它在效率上存在一定的缺陷。

[0003] 数据库领域的局部敏感哈希方法是解决高维空间中的近邻查询问题的一种有效方法。局部敏感哈希的基本思想是通过局部敏感哈希函数来进行实现的，它保证了高维空间中距离较近的数据点在局部敏感哈希函数作用下相互碰撞的概率更大而距离较远的数据点相互碰撞的概率更小。对于较小的数据规模，内存中的基于局部敏感哈希思想的哈希表，可以用于在内存中快速地获取目标k近邻；对于较大的数据规模，结合局部敏感哈希思想与B^+树的特殊索引结构可以用于在磁盘中快速地获取目标k近邻。

[0004] 推土机距离是两个概率分布之间距离的一个度量。作为一种相似性度量方式，它已经被广泛应用到信息检索、数据库、多媒体、机器学习等诸多领域。由于推土机距离通常被形式化为一个线性优化问题并可以建模成二分网络的最小费用流，所以它需要一个较大的时间复杂度。推土机距离度量下的索引问题的解决方法主要是基于过滤-细化框架。在过滤阶段，通过推土机距离的下界来确定数据记录是否能够被过滤掉；在细化阶段，通过计算推土机距离来确定是否需要对目标集合进行更新。这些下界主要包括质心与投影、降维、原始-对偶空间、正态分布等等。

发明内容

[0005] 本发明的目的在于针对现有技术的不足，提供一种快速的层次化文档查询方法。

[0006] 本发明解决其技术问题采用的技术方案如下：

[0007] 本发明针对一系列的多个文档集进行处理和查询，一个文档集是多篇文档构成。

[0008] 本发明采用Word2Vec模型词向量对文档进行处理，然后采用局部敏感哈希的思想在内存中为文档集构建索引结构，并用层次化的文档查询方式从索引结构获得所对应的文档。

[0009] 所述方法的步骤如下：

[0010] 1)对文档集中的每个文档建立数据模型，一篇文档的数据模型主要由单词、词向

量和单词权重的三部分组成；

[0011] 2)对文档进行格式化处理获得文档质心向量和文档标签；

[0012] 3)步骤2)中生成的文档质心向量作为高维向量空间中的一个点，为每个文档集采用局部敏感哈希方法在内存中构建哈希索引结构；

[0013] 4)用户所输入的查询文本，依据查询文本的文档质心向量，采用基于局部敏感哈希思想的查询方法在哈希索引结构中获取一个候选文档集；

[0014] 5)依据查询文本的文档标签，采用过滤-细化框架在候选文档集中获取词移动距离度量下的k个最近邻文档，完成查询。

[0015] 所述步骤1)中，单词为文档经过预处理后剩下的有效单词，词向量为Google News Word2Vec模型的单词向量表示，单词权重为单词所对应的TF-IDF值。预处理包括依次进行的分词、去停用词和去高低频词等步骤。

[0016] 所述步骤2)具体为：通过计算词向量对单词权重的加权均值得到文档的文档质心向量，通过对单词权重进行归一化处理并与对应的词向量一起构成文档的文档标签，将文档质心向量与文档标签存储于内存的特定数据结构中。

[0017] 所述步骤2)具体是将一个文档标签以一个结构体数组形式存储，将一个文档质心向量以一个数组形式存储。具体实施所采用的词向量是300维，因此文档质心向量存储于大小为300的数组中。

[0018] 所述的候选文档集中包含文档的数量为mk，k表示最近邻文档的数量，m表示倍数取值。

[0019] 所述步骤3)中为每个文档集采用局部敏感哈希方法在内存中构建哈希索引结构是采用局部敏感哈希函数簇方式构建多个哈希表，具体为：

[0020] 3.1)对于文档集中每个文档，采用以下公式生成多个局部敏感哈希函数：

[0021] $$h(o) = \left\lfloor \frac{a \cdot o + bW}{W} \right\rfloor$$

[0022] 其中，h(o)表示局部敏感哈希函数，⌊ ⌋表示取整数下限；a表示一个随机数向量，a的维数和文档质心向量的维数相同，a每一维的随机数是基于高斯分布独立选取；o是高维向量空间中的文档质心向量，b是随机参数，是基于数值范围[0,1]的均匀分布进行选取，W表示局部敏感哈希函数宽度的实数；

[0023] 3.2)将得到的所有局部敏感哈希函数复合构成一个复合函数，并作为一个哈希表，一个哈希表对应一个复合函数，文档集中每一个文档根据复合函数映射到哈希表的哈希桶中，

[0024] 3.3)重复步骤3.1)和3.2)构建获得多个哈希表。

[0025] 所述步骤4)中查询文本的文档质心向量和所述步骤5)中查询文本的文档标签是以查询文本作为文档进行步骤1)和2)，获得查询文本的文档质心向量和文档标签。

[0026] 所述步骤4)中的基于局部敏感哈希思想的查询方法为Multi-probe LSH查询方法。

[0027] Multi-probe LSH查询方法具体是采用文献《Lv Q,Josephson W,Wang Z,et al.Multi-probe LSH:efficient indexing for high-dimensional similarity search

[C]//International Conference on Very Large Data Bases.VLDB Endowment,2007:950-961.》中的方法实施。

[0028] 所述步骤5)具体是先新建空白的目标文档集并以堆数据结构形式存储,以堆数据结构形式存储的目标文档集具有按词移动距离从小到大排序的特性,然后对于候选文档集中所有候选文档,以每次一个文档采用以下方式处理,包括过滤阶段和细化阶段:

[0029] 5.1)若目标文档集中的文档数量小于等于k,k表示预先设定的查询文档数值,将候选文档依次加入目标文档集,多篇候选文档加入后目标文档集会将该多篇候选文档按词移动距离从小到大排序来存储;

[0030] 若目标文档集中的文档数量大于k,则进行以下过滤阶段的步骤;

[0031] 5.2)过滤阶段

[0032] 计算查询文本与候选文档之间词移动距离的三个下界值,三个下界值分别为文档质心距离和两个松弛词移动距离,用文档质心距离、两个松弛词移动距离来判断候选文档是否剔除;

[0033] 若三个下界值中的任何一个大于阈值θ,阈值θ表示查询文本与目标文档集中第k篇文档之间的词移动距离,则将候选文档从目标文档集剔除,返回步骤5.1)对下一个候选文档进行处理;

[0034] 若三个下界值全部不大于阈值θ,则进入以下细化阶段的步骤;

[0035] 5.3)过滤阶段

[0036] 计算查询文本与目标文档集中各个文档之间的词移动距离,在目标文档集的堆数据结构中按词移动距离从小到达排序,将词移动距离最大对应的文档剔除,直到目标文档集中的文档数量保持在k个,并更新阈值θ与目标文档集的堆数据结构,再返回步骤5.1)对下一个候选文档进行处理。

[0037] 所述步骤5.2)中,文档质心距离和松弛词移动距离分别采用以下方式计算:

[0038] a)文档质心距离用高维向量空间中的欧式距离来表示,计算公式为:

[0039] $\|Xd_1-Xd_2\|$

[0040] 其中,Xd_1与Xd_2分别表示查询文本和候选文档的文档质心向量;

[0041] b)第一个松弛词移动距离形式化为以下公式所示的线性优化问题,即采用以下公式计算;

[0042] $$min_{T\geq 0}\sum_{i,j=1}^{n} T_{ij}c(i,j)$$

[0043] $$s.t.\sum_{i=1}^{n} T_{ij} = w_{2j} \quad \forall j \in \{1,\dots,n\}$$

[0044] 其中,$min_{T\geq 0}\sum_{i,j=1}^{n} T_{ij}c(i,j)$为松弛词移动距离;c(i,j)表示单词i与j之间的移动代价,等于Word2Vec词向量间的欧式距离,单词i属于查询文本,单词j属于文档;w_{2j}表示文档标签中归一化处理后的单词权重;T表示流动权重矩阵作为最优解的最优矩阵,T_{ij}表示单词i与单词j之间的流动权重值;n表示一个文档中单词的总数;

[0045] 单词i与单词j之间的流动权重值T_{ij}采用以下公式计算:

[0046] $$T_{ij} = \begin{cases} w_{2j} & if\ i = argmin_i c(i,j) \\ 0 & otherwise \end{cases}$$

[0047] b)第二个松弛词移动距离形式化为以下公式所示的线性优化问题,即采用以下公式计算;

[0048] $$min_{T\geq 0} \sum_{i,j=1}^{n} T_{ij} c(i,j)$$

[0049] $$s.t. \sum_{j=1}^{n} T_{ij} = w_{1i} \quad \forall i \in \{1,\dots,m\}$$

[0050] 其中,$min_{T\geq 0} \sum_{i,j=1}^{n} T_{ij} c(i,j)$为松弛词移动距离;c(i,j)表示单词i与j之间的移动代价,等于Word2Vec词向量间的欧式距离,单词i属于查询文本,单词j属于文档;w_{1i}表示查询文本标签中归一化处理后的单词权重;T表示流动权重矩阵作为最优解的最优矩阵,T_{ij}表示单词i与单词j之间的流动权重值;m表示一个查询文本中单词的总数;

[0051] 单词i与单词j之间的流动权重值T_{ij}采用以下公式计算:

[0052] $$T_{ij} = \begin{cases} w_{1i} & if\ j = argmin_j c(i,j) \\ 0 & otherwise \end{cases}$$。

[0053] 上述w下标的2代表了文档,1代表了查询文本。

[0054] 所述步骤5.3)中,词移动距离采用以下方式计算:

[0055] $$min_{T\geq 0} \sum_{i,j=1}^{n} T_{ij} c(i,j)$$

[0056] $$s.t. \sum_{i=1}^{n} T_{ij} = w_{2j} \quad \forall j \in \{1,\dots,n\}$$

[0057] $$\sum_{j=1}^{n} T_{ij} = w_{1i} \quad \forall j \in \{1,\dots,m\}$$

[0058] 其中,$min_{T\geq 0} \sum_{i,j=1}^{n} T_{ij} c(i,j)$为词移动距离;c(i,j)表示单词i与j之间的移动代价,等于Word2Vec词向量间的欧式距离,单词i属于查询文本,单词j属于文档;w_{1i}表示查询文本标签中归一化处理后的单词权重,w_{2j}表示文档标签中归一化处理后的单词权重;T表示流动权重矩阵作为最优解的最优矩阵,T_{ij}表示单词i与单词j之间的流动权重值;n表示一个文档中单词的总数;m表示一个查询文本中单词的总数;

[0059] 单词i与单词j之间的流动权重值Tij采用以下公式计算:

[0060] $$T_{ij} = \begin{cases} w_{1i} & if\ j = argmin_j c(i,j) \\ 0 & otherwise \end{cases}$$。

[0061] 词移动距离可以看作是文献《Tang Y,Leong H U,Cai Y,et al.Earth mover's distance based similarity search at scale[J].Proceedings of the Vldb Endowment,2013,7(4):313-324.》中推土机距离的一个特例,可以具体等效描述成二分网络的最小费用流问题,单词i与单词j之间的流动权重值T_{ij}可以通过文献中描述的连续最短路径算法计算。

[0062] 本发明具有的有益效果是:

[0063] 本发明采用Word2Vec模型词向量对文档进行格式化处理，并采用局部敏感哈希的思想为文档集构建索引。

[0064] 本发明层次化查询方法应用于文档分类和检索时在效率和效果上获得了良好的平衡，使得用户在进行词移动距离度量下的文档查询时能够在保证准确性的情况下快速地获取目标文档。

附图说明

[0065] 图1是本发明文档格式化处理流程图。

[0066] 图2是本发明一个文档质心向量的索引构建例子。

[0067] 图3是本发明的文档层次化查询流程图。

[0068] 图4是本发明的过滤-细化框架的实施示例。

[0069] 图5是本发明的词移动距离计算问题对应的二分网络流。

具体实施方式

[0070] 本发明的实施例及其实施过程如下：

[0071] 具体实施是以文档数据集Reuters-21578(Reuters)和示意图对本发明的技术方案作进一步说明；其中，Reuters是Newswire新闻文档集的一个分支，并可以从互联网上公开获取，它包含65个主题、8293篇文档，2347篇文档是用于测试的，5946篇训练文档用于构建哈希表。

[0072] 文档格式处理阶段

[0073] 步骤1：建立每篇文档的数据模型，一篇文档的数据模型主要由单词、词向量和单词权重的三部分组成；单词为文档经过预处理后剩下的有效单词，词向量为互联网上可公开获取的Google News Word2Vec模型词向量，单词权重为单词所对应的TF-IDF值。

[0074] 步骤2：对文档进行格式处理。如图1所示，通过计算词向量对TF-IDF值的加权均值得到文档的文档质心向量；通过对TF-IDF值进行归一化处理并与对应的词向量一起构成文档的标签；文档质心向量与标签都可以存储于特定的数据结构中。将一个文档标签以一个结构体数组形式存储，将一个文档质心向量以一个数组的形式存储。

[0075] 构建哈希索引阶段

[0076] 步骤3：步骤2中生成的文档质心向量可以表示为高维向量空间中的一个点，那么，文档质心向量间的距离可以用高维向量空间中的欧式距离来计算。基于此，局部敏感哈希方法通过局部敏感哈希函数簇来构建哈希索引，具体是：

[0077] 3.1)首先，采用以下公式生成m个局部敏感哈希函数，具体表示为$h_1,h_2,\cdots,h_m$与一个m维的局部敏感哈希函数h_{m+1}(局部敏感哈希函数的维度与文档质心向量的维度相同)，这m+1个局部敏感哈希函数中的参数W具有相同的配置数值。

[0078] $$h(o)=\left\lfloor\frac{a\cdot o+bW}{W}\right\rfloor$$

[0079] 其中，h(o)表示局部敏感哈希函数，⌊ ⌋表示取整数下限；o是高维向量空间中的文档质心向量，a表示一个随机数向量，它的维数和文档质心向量o的维数相同，a每一维的

随机数是基于高斯分布独立选取;b是随机参数,是基于数值范围[0,1]的均匀分布进行选取,W表示局部敏感哈希函数宽度的实数;

[0080] 3.2)将生成的所有局部敏感哈希函数$h_1,h_2,\cdots,h_m,h_{m+1}$复合形成一个复合哈希函数g。一个特定的文档质心向量o的具体复合方式是:

[0081] 依据前m个局部敏感哈希函数$h_1,h_2,\cdots,h_m$生成文档质心向量o的一个复合哈希键(即m维向量)$K=<h_1(o),h_2(o),\cdots,h_m(o)>$,那么文档质心向量o在复合哈希函数g作用下的哈希值可以被表示为$g(o)=h_{m+1}(K)$;其中一张哈希表对应于一个复合哈希函数,复合哈希函数值对应于哈希表中的一个哈希桶,而一个哈希桶会对应一个链表数据结构。

[0082] 3.3)重复步骤3.1)和3.2)L次,可以获得L个复合哈希函数$g_1,g_2,\cdots,g_L$;它们分别对应内存中的哈希表$table_1,table_2,\cdots,table_L$;其中每一个复合哈希函数对应一张哈希表中的函数映射。

[0083] 3.4)依据步骤3.3)中生成的每一个复合哈希函数g_i(i属于范围1～L),将文档数据集Reuters中的5946篇训练文档的每一篇文档映射到内存中哈希表$table_i$对应的哈希桶中;重复L次可以完成L张哈希表的构建。

[0084] 图2给出了一个构建哈希索引的例子,其中L=8。

[0085] 文档数据集中的每一篇文档,经过文档格式处理阶段后,可以被表示成文档质心向量o的形式。

[0086] L=8对应着8个复合哈希函数与8张哈希表,那么文档质心向量o可以被8个复合哈希函数$g_1,g_2,\cdots,g_8$映射到相应哈希表中的哈希桶上;其中$g_1(o)=2,g_2(o)=6,\cdots,g_8(o)=4$表示哈希桶的编号。

[0087] 对Reuters文档数据集中的5946篇训练文档应用相似的处理可以将每一篇文档都映射到相应哈希表中的哈希桶上。

[0088] 文档查询阶段

[0089] 步骤4:查询文本作为文档经过文档格式处理阶段的步骤可以获得查询文本的文档质心向量和文档标签。

[0090] 图3给出了文档层次化查询的流程。

[0091] 为文档集Reuters构建的哈希表存放于内存,依据文档质心向量的表示而采用Multi-probe LSH查询方法在L张哈希表中来获取候选文档集;它主要包括生成探测序列、挑选内存中最相似的哈希桶以及在哈希桶中对文档进行验证这三个部分。其中,候选文档集包含文档的数量为mk,k为最近邻文档的数量,m表示倍数取值。

[0092] 步骤5:依据查询文本的标签格式而采用过滤-细化框架在候选文档集中获取词移动距离度量下的k个最近邻文档;这里,k个最近邻文档以目标文档集形式全部存储在堆数据结构中,并维护一个阈值θ表示所输入的查询文本与最近的第k篇文档之间的词移动距离。

[0093] 先新建空白的目标文档集并以堆数据结构形式存储,以堆数据结构形式存储的目标文档集可以设置按词移动距离从小到大排序的特性,然后对于候选文档集中所有候选文档,以每次一个文档采用以下方式处理,包括过滤阶段和细化阶段:

[0094] 5.1)若目标文档集中的文档数量小于等于k,k表示预先设定的目标文档数值,将候选文档依次加入目标文档集,多篇候选文档加入后目标文档集会将该多篇候选文档按词

移动距离从小到大排序来存储；

[0095] 若目标文档集中的文档数量大于k，则进行以下过滤阶段的步骤；

[0096] 5.2)过滤阶段

[0097] 计算查询文本与候选文档之间词移动距离的三个下界值，三个下界值分别为文档质心距离和两个松弛词移动距离1与2，用文档质心距离、两个松弛词移动距离来判断候选文档是否剔除；

[0098] 若三个下界值中的任何一个大于阈值θ，阈值θ表示查询文本与目标文档集中第k篇文档之间的词移动距离，则将候选文档剔除，返回步骤5.1)对下一个候选文档进行处理；

[0099] 若三个下界值全部不大于阈值θ，则进入以下细化阶段的步骤；

[0100] 5.3)细化阶段

[0101] 计算查询文本与候选文档之间的词移动距离，将候选文档插入到目标文档集所在的堆数据结构，堆数据结构可以设置为按词移动距离从小到大排序，将词移动距离最大的文档剔除，使得目标文档集中的文档数量保持在k个，并更新阈值θ与目标文档集的堆数据结构，再返回步骤5.1)对下一个候选文档进行处理。

[0102] 图4给出一个过滤-细化框架的例子，其中m＝3、k＝3、候选文档集包含9篇文档。首先，构建一个空的目标文档集并通过堆数据结构在内存中进行存储，堆数据结构存储的文档设置为按词移动距离从小到大排序。

[0103] 目标文档集包含的文档数目为0，经过步骤5.1)，候选文档集中的前面3篇文档依次存放于目标文档集中，阈值θ_1表示查询文本与第3篇最近的文档的词移动距离，即查询文本与3号文档的词移动距离。

[0104] 在处理候选文档集中的4号文档时，由于存在一个查询文本与4号文档的词移动距离的下界松弛词移动距离或者文档质心距离大于阈值θ_1，所以4号文档在步骤5.2)过滤阶段中直接被过滤掉，阈值θ_1不变，并继续处理下一篇文档。

[0105] 在处理候选文档集中的5号文档时，由于查询文本与5号文档的词移动距离的下界松弛词移动距离和文档质心距离都不大于阈值θ_1，那么步骤5.2)过滤阶段无法将5号文档过滤掉；在步骤5.3)细化阶段中，计算查询文本与5号文档间的词移动距离，该词移动距离小于1号文档与查询文本之间的词移动距离，所以3号文档会被移除目标文档集，并更新阈值为θ_2，它为查询文本与1号文档间的词移动距离。

[0106] 对候选文档集中的6～9号文档，重复步骤5.2)与5.3)；最后目标文档集中包含2号、5号、9号这3篇文档，它们与查询文本之间的词移动距离依次增加。

[0107] 在这一阶段，本发明需要计算查询文本与文档之间的词移动距离，它作为推土机距离的一个特例，可以具体等效描述成图5所展示的二分网络的最小费用流问题，并可以通过连续最短路径算法计算得到最终结果；其中每一个word表示文档中单词所对应的Word2Vec词向量，每一个w表示文档中单词所对应的归一化TF-IDF值。

[0108] 总结

[0109] 经过文档格式处理阶段和构建哈希索引阶段后，本发明创建了L张哈希表用于索引文档集Reuters中的5946篇训练文档。

[0110] 在文档查询阶段，对于查询文本，本发明所述的层次化查询通过Multi-probe LSH查询方法计算哈希函数的方式来选取文档，而步骤5中所述的过滤-细化框架是一种线性查

询方法。这里是本发明层次化查询最重要的一个优化，Multi-probe LSH查询方法将文档集Reuters中5946篇训练文档限制在mk篇候选文档中，从而避免了步骤5中大规模的线性查询，提高了词移动距离度量下的文档查询效率。

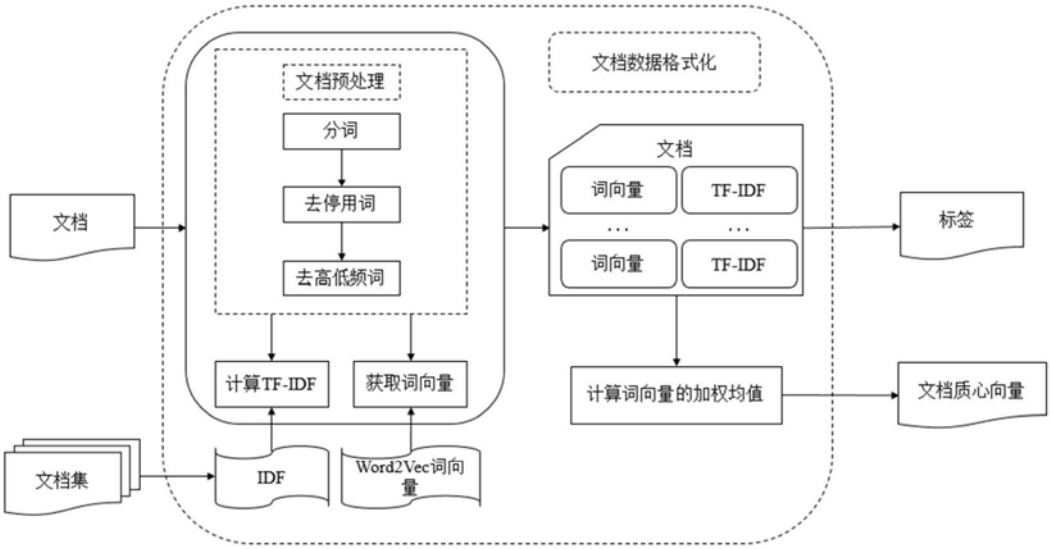

文档预处理
分词
去停用词
去高低频词
文档数据格式化
文档
词向量
TF-IDF
词向量
TF-IDF
文档
标签
计算TF-IDF
获取词向量
计算词向量的加权均值
文档质心向量
文档集
IDF
Word2Vec词向量

图1

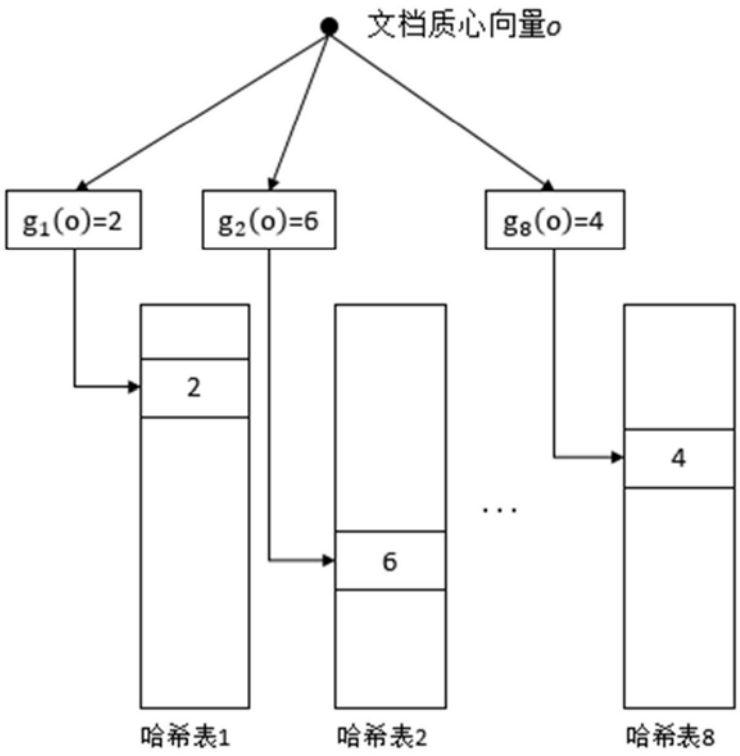

文档质心向量o
g1(o)=2
g2(o)=6
g8(o)=4
2
6
4
哈希表1
哈希表2
哈希表8

图2

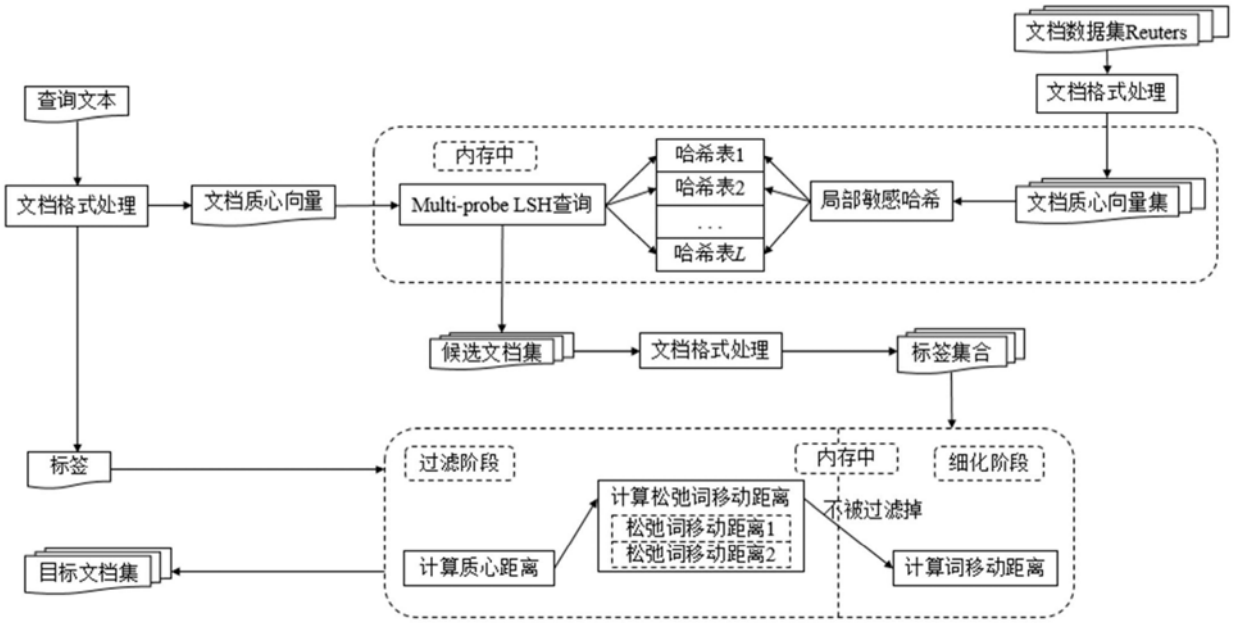

文档数据集Reuters
文档格式处理
查询文本
内存中
哈希表1
哈希表2
...
哈希表L
文档格式处理
文档质心向量
Multi-probe LSH查询
局部敏感哈希
文档质心向量集
候选文档集
文档格式处理
标签集合
标签
过滤阶段
内存中
细化阶段
计算松弛词移动距离
松弛词移动距离1
松弛词移动距离2
不被过滤掉
目标文档集
计算质心距离
计算词移动距离

图3

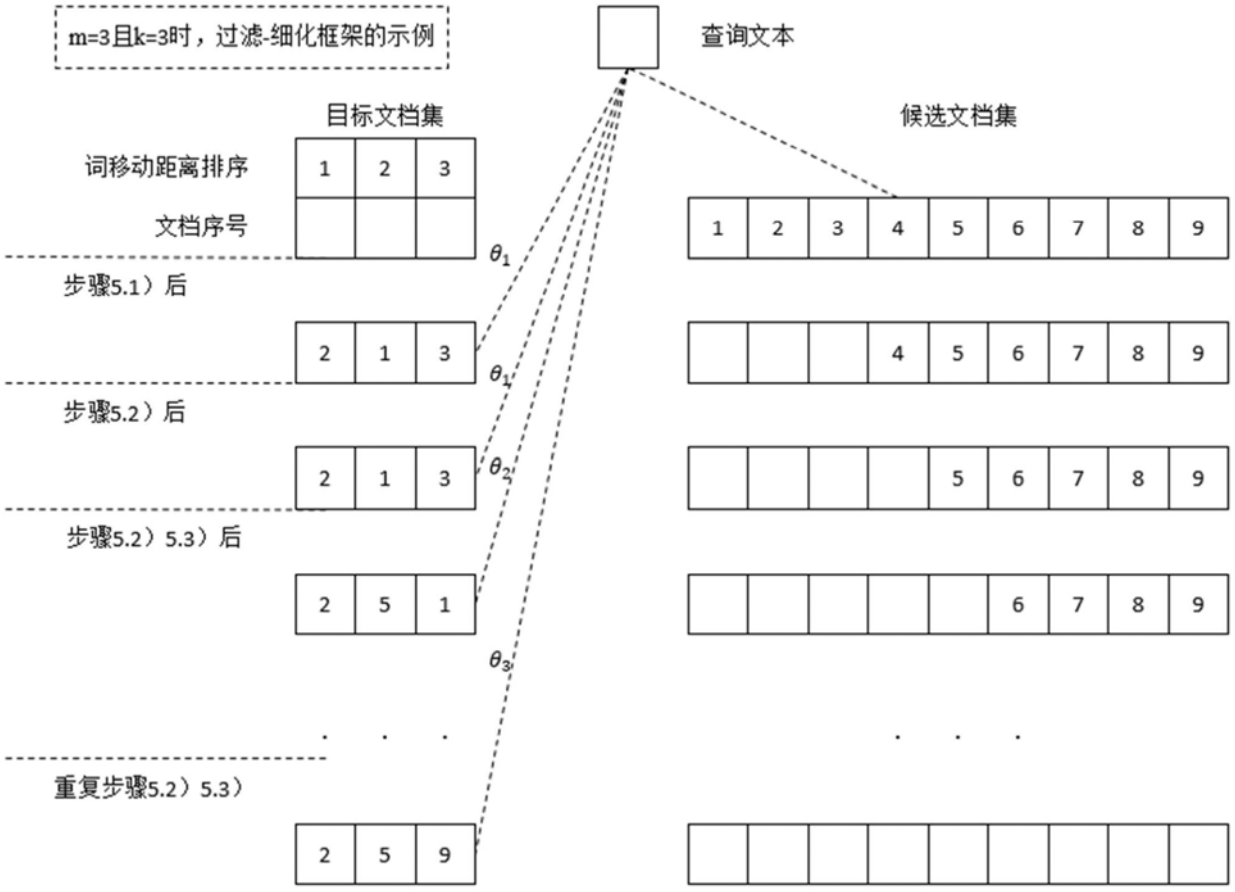

m=3且k=3时，过滤-细化框架的示例
查询文本
目标文档集
候选文档集
词移动距离排序
文档序号
步骤5.1）后
步骤5.2）后
步骤5.2）5.3）后
重复步骤5.2）5.3）

图4

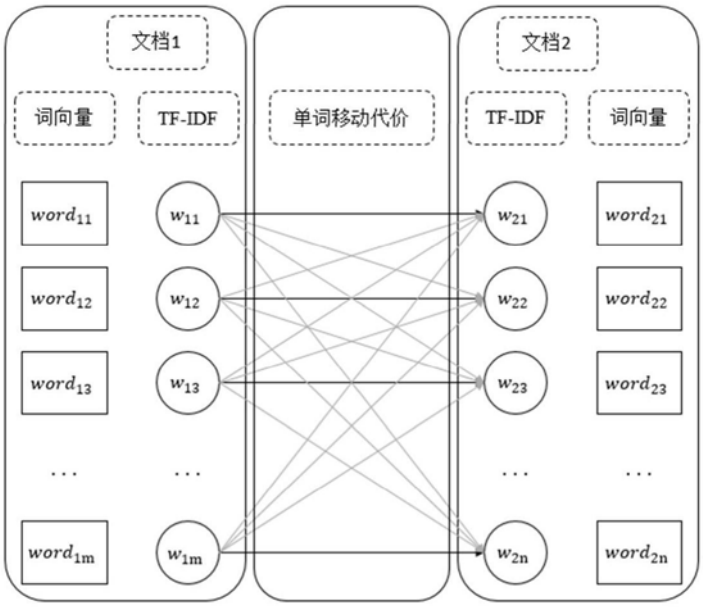

文档1
词向量
TF-IDF
单词移动代价
文档2
TF-IDF
词向量
$word_{11}$
w_{11}
w_{21}
$word_{21}$
$word_{12}$
w_{12}
w_{22}
$word_{22}$
$word_{13}$
w_{13}
w_{23}
$word_{23}$
$word_{1m}$
w_{1m}
w_{2n}
$word_{2n}$

图5