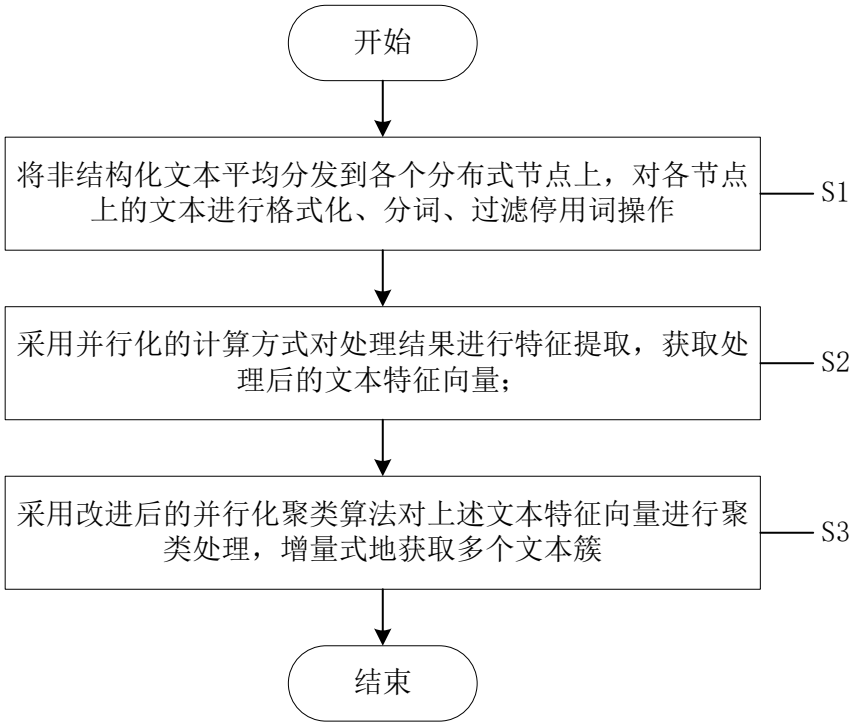
说明书摘要

本发明公开了一种并行化的文本聚类方法，其中，所述方法包括：将非结构化文本平均分发到各分布式节点上，对各节点上的文本进行预处理、分词、过滤停用词操作，采用并行化的计算方式对处理结果进行特征提取，获取处理后的文本特征向量；采用改进后的并行化聚类方法对上述文本特征向量进行聚类处理，增量式地获取多个文本簇；通过将聚类过程中的各个步骤并行化，在面对海量或高维数据时，提升了文本聚类的速度。

摘要附图


开始
将非结构化文本平均分发到各个分布式节点上，对各节点上的文本进行格式化、分词、过滤停用词操作
S1
采用并行化的计算方式对处理结果进行特征提取，获取处理后的文本特征向量；
S2
采用改进后的并行化聚类算法对上述文本特征向量进行聚类处理，增量式地获取多个文本簇
S3
结束

权利要求书

1.一种并行化的文本聚类方法，其特征在于，所述方法包括：

1.1 将非结构化文本平均分发到各分布式节点上，对各节点上的文本进行预处理、分词、过滤停用词操作；

1.2 采用并行化的计算方式对处理结果进行特征提取，获取处理后的文本特征向量；

1.3 采用改进后的并行化聚类方法对上述文本特征向量进行聚类处理，增量式地获取多个文本簇。

2.如权利要求 1 所述的方法，其特征在于，所述将非结构化文本平均分发到各分布式节点上，对各节点上的文本进行预处理、分词、过滤停用词操作，包括：

2.1 采用“key=文本编号，value=文本内容”的格式，将已有非结构化文本平均分发到各分布式节点上；

2.2 对各分布式节点上的非结构化文本进行统一格式处理，去除文本首尾非文本部分，获取纯文本部分，若为空文本则跳过；

2.3 将所述纯文本部分进行分词处理，针对词语词性，去除分词结果中的标点符号、拟声词、叹词、助词、连词、介词、副词、数词、量词；

3.如权利要求 1 所述的方法，其特征在于，所述采用并行化的计算方式对处理结果进行特征提取，获取处理后的文本特征向量，包括：

3.1 采用并行化计算方式对各分布式节点的分词结果进行处理，获取所有文本的词频向量，具体步骤包括：

为每个文本构建一个维度足够大（如2^{18}，该值可根据文本数量大小进行估计）的词频向量tf_i，向量长度为 length，求词语的 hash 值，对 length 取模，得到该词语映射到该向量上的索引；对文本中的每个词语在tf_i上其对应索引位置进行加 1 操作，将该向量转化为一个稀疏的向量（记录非零元素的索引及其值），获取所有文本的词频向量；

3.2 采用并行化计算方式对各分布式节点的词频向量进行处理，获取所有文本的逆文本频率向量，将该向量广播到各分布式节点上，具体步骤包括：

为该分布式节点构建词语出现的文本频率向量df_i，维度与tf_i一致，向量的每个元素表示词语在该节点多少个文本中出现过，遍历节点中每个文本的词频向量，循环取出词频向量中非零元素（表示该词在该文本中出现了）的位置，在df_i的对应索引位置进行加 1 操作；将各分布式节点的tf_i向量的对应成员值聚合相加得到总文本频率向量DF；进而获取所有文本的逆文本频率向量IDF，将该向量广播到各分布式节点上；

3.3 计算各分布式节点上词语的$TF-IDF$值，获取文本特征向量，具体步骤包括：

将文本对应的词频向量tf_i和逆文本频率向量IDF相乘得到每个文本的$TF-IDF$向量，按照“key=文本编号，value=TF-IDF 向量”的格式聚合所有节点

上的$TF-IDF$向量，得到总$TF-IDF$向量；

4. 如权利要求 1 所述的方法，其特征在于，所述采用改进后的并行化聚类方法对上述文本特征向量进行聚类处理，增量式地获取多个文本簇，包括：

4.1 将总$TF-IDF$向量广播到各分布式节点上，计算每个文本与总$TF-IDF$向量中该文本之前的所有文本的余弦相似度，具体步骤包括：

计算每个文本与总$TF-IDF$向量中该文本之前的所有文本的余弦相似度，从这$i-1$个余弦相似度中取出最大值$\max_{i,j}$，即第i个文本与前$i-1$个文本的余弦相似度中的最大值；

4.2 创建共享向量 Data，维度与文本数量一致，以存放聚类结果；

4.3 根据上述余弦相似度对所述文本进行改进后的并行化聚类，增量式地获取多个文本簇，具体步骤包括：

当$i=1$或$\max_{i,j}$小于设定的阈值时，为第i个文本新建一个文本簇，以“key=文本编号，group=i”的格式在 Data 索引为i的位置存放数据；

当$\max_{i,j}$大于设定的阈值时，将第i个文本与第j个文本归为同一文本簇，在 Data 获取文本j的 group 值 G，以“key=文本编号，group=G”的格式在 Data 索引为i的位置存放数据；

最后得到的 Data 向量即聚类结果，group 一致的文本被聚为同一文本簇。

一种并行化的文本聚类方法

技术领域

本文涉及计算机技术领域，尤其涉及一种并行化的文本聚类方法。

背景技术

随着信息网络技术的迅速发展和互联网的进一步普及，网络上的数据呈现几何式的增长，数据“爆炸”已成为当前网络时代的特征之一。面对如此庞大而且增长迅速的数据，高效地挖掘有用信息无论在商业、医疗还是科学研究方面，都有着非常巨大的价值。其中，大量信息都以文本形式存储，如新闻稿件、科技论文、书籍、数字图书馆、邮件、博客和网页等等。文本聚类技术可以将大量文本聚合为少数有意义的簇，从而在大量文本中导出高质量的信息，使得人们从数据中获取信息、知识和决策支持更加容易。

但是，传统串行式的文本聚类方法在处理海量或者高维数据时，聚类的速度不够快，在面对大规模数据时，受制于内存容量，往往不能有效地运行，因而传统串行式文本聚类方法已经难以满足当前实际应用的需求。

并行计算可以将大规模数据分发到多个分布式节点上并行地进行计算，最后将所有节点的计算结果归并为最终的结果，可以大大地提高聚类速度。

发明内容

本发明的目的在于克服现有技术的不足，提供一种并行化的文本聚类方法，充分利用并行计算的优点，提高文本聚类的速度，所述方法包括：

采用“key=文本编号，value=文本内容”的格式，将已有非结构化文本平均分发到各分布式节点上；

对各节点上的非结构化文本进行预处理、分词、过滤停用词操作，得到每个文本的分词结果；

基于向量空间模型对每个文本的分词结果进行特征提取，假设一个由 n 个文本构成的文本集合为$\{d_1,d_2,...,d_n\}$，第 j 个文本的词语构成为$\{t_1,t_2,...,t_m\}$，m 代表该文本中所有词语的个数，那么每一个文本表示为$d_j=\{w_{1,j},w_{2,j},...,w_{m,j}\}$，其中$w_{i,j}$为第 j 个文本中词语t_i的权重，该权重使用 TF-IDF 计算；其中词语在文本中的词频$tf_{i,j}=\frac{n_{i,j}}{\sum_k n_{k,j}}$，$n_{i,j}$表示第 j 个文本中词语t_i的个数；词语$t_{i,j}$的逆文本频率指数$idf_i=\log\frac{|D|+1}{|\{j:i\in d_j\}|+1}$，$|D|$表示所有文本数目，$|\{j:i\in d_j\}|$表示词语$t_i$所在文本个数；然后得到词语$t_i$的 TF-IDF 权值$tf\cdot idf_{i,j}=tf_{i,j}\times idf_i$；所需数据计算方法

包括：

采用并行化计算方式对各分布式节点的分词结果进行处理，具体步骤包括：为每个文本构建一个维度足够大（如2^{18}，该值可根据文本数量大小进行估计）的词频向量tf_i，向量长度为 length，求词语的 hash 值，对 length 取模，得到该词语映射到该向量上的索引；对文本中的每个词语在tf_i上其对应索引位置进行加 1 操作，将该向量转化为一个稀疏的向量（记录非零元素的索引及其值），获取所有文本的词频向量；

采用并行化计算方式对各分布式节点的词频向量进行处理，具体步骤包括：为该分布式节点构建词语出现的文本频率向量df_i，维度与tf_i一致，向量的每个元素表示词语在该节点多少个文本中出现过，遍历节点中每个文本的词频向量，循环取出词频向量中非零元素（表示该词在该文本中出现了）的位置，在df_i的对应索引位置进行加 1 操作；将各分布式节点的tf_i向量的对应成员值聚合相加得到总文本频率向量DF；设总文档数为 n，词语t_i在DF中的文本频率指数为DF_i，则词语t_i的逆文本频率指数$IDF_i = \log\frac{n+1}{DF_i+1}$，获取所有文本的逆文本频率向量$IDF$，将该向量广播到各分布式节点上；

在各分布式节点上，将文本对应的词频向量tf_i和逆文本频率向量IDF相乘得到每个文本的$TF-IDF$向量，按照“key=文本编号，value=TF-IDF 向量”的格式聚合所有节点上的$TF-IDF$向量，得到总$TF-IDF$向量，再将其广播到各节点上；现进行对各文本的聚类，方法如下：

计算每个文本与总$TF-IDF$向量中该文本之前的所有文本的余弦相似度，即$\cos\theta = \frac{d_i \cdot d_j}{|d_i| \times |d_j|}$，其中$i > j$，$d_i$表示第$i$个文本的$TF-IDF$向量，从这$i-1$个余弦相似度中取出最大值$\max_{i,j}$，即第$i$个文本与前$i-1$个文本的余弦相似度中的最大值；创建共享向量 Data，维度与文本数量一致，以存放聚类结果；

当$i=1$或$\max_{i,j}$小于设定的阈值时，为第i个文本新建一个文本簇，以“key=文本编号，group=i”的格式在 Data 索引为i的位置存放数据；

当$\max_{i,j}$大于设定的阈值时，将第i个文本与第j个文本归为同一文本簇，在 Data 获取文本j的 group 值 G，以“key=文本编号，group=G”的格式在 Data 索引为i的位置存放数据；

最后得到的 Data 向量即聚类结果，group 一致的文本被聚为同一文本簇。

附图说明

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。

图 1 是本发明实施例的一种并行化的文本聚类方法的流程示意图；

图 2 是本发明实施例的文本预处理、分词、过滤停用词操作步骤的流程示意图；

图 3 是本发明实施例的获取文本特征向量步骤的流程示意图；

图 4 是本发明实施例的改进的并行化文本聚类方法步骤的流程示意图。

具体实施方式

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它实施例，都属于本发明保护的范围。

图 1 是本发明实施例的一种并行化的文本聚类方法的流程示意图，如图 1 所示，该方法包括：

步骤 S1：将非结构化文本平均分发到各分布式节点上，对各节点上的文本进行预处理、分词、过滤停用词操作；

步骤 S2：采用并行化的计算方式对处理结果进行特征提取，获取处理后的文本特征向量；

步骤 S3：采用改进后的并行化聚类方法对上述文本特征向量进行聚类处理，增量式地获取多个文本簇；

对步骤 S1 作进一步说明：

采用“key=文本编号，value=文本内容”的格式，预先就将非结构化文本平均分发到各分布式节点，之后的大部分操作都将在各节点完成，以提升操作完成速度；数据库中文本可能存在首尾部分有冗余内容或者文本自身为空的情况，需要先进行一步预处理，再对纯文本进行分词操作，获取分词结果，其中包括词语词性，过滤掉部分词性的词语。

进一步地，图 2 是本发明实施例的文本预处理、分词、过滤停用词操作步骤的流程示意图，如图 2 所示，该步骤包括：

步骤 S11：对各分布式节点上的非结构化文本进行统一格式处理，去除文本首尾非文本部分，获取纯文本部分，若为空文本则跳过；

步骤 S12：将所述纯文本部分进行分词处理；

步骤 S13：针对词语词性，去除分词结果中的标点符号、拟声词、叹词、助词、连词、介词、副词、数词、量词。

对步骤 S2 作进一步说明：

采用并行化计算方式对各分布式节点的分词结果进行处理，获取所有文本的词频向量；采用并行化计算方式对各分布式节点的词频向量进行处理，获取所有

文本的逆文本频率向量；将文本对应的词频向量和逆文本频率向量相乘得到每个文本的$TF-IDF$向量， 即各文本的特征向量。

进一步地，图 3 是本发明实施例的获取文本特征向量步骤的流程示意图，如图 3 所示，该步骤包括：

步骤 S21：为每个文本构建一个维度足够大的词频向量，维度可根据文本数量大小进行估计，向量长度为 length；

步骤 S22：求文本中每个词语的 hash 值，对 length 取模，得到该词语映射到该向量上的索引；

步骤 S23：对文本中的每个词语在其对应索引位置进行加 1 操作；

步骤 S24：记录非零元素的索引及其值，将该向量转化为一个稀疏的向量，获取所有文本的词频向量；

步骤 S25：为各分布式节点构建词语出现的文本频率向量，维度与词频向量一致；

步骤 S26：遍历节点中每个文本的词频向量，循环取出词频向量中非零元素的位置，在文本频率向量对应索引位置进行加 1 操作；

步骤 S27：将各分布式节点的文本频率向量的对应成员值聚合相加得到总文本频率向量；

步骤 S28：对总文本频率向量进行计算得到所有文本的逆文本频率向量；

步骤 S29：将文本对应的词频向量和逆文本频率向量相乘得到每个文本的 TF-IDF 向量。

对步骤 S21 作进一步说明：

词频向量的维度应足够大，来保证步骤 S22 中词语的索引不会频繁出现冲突，如2^{18}，该值可根据文本数量进行设置。

对步骤 S23 作进一步说明：

该步骤统计了该文本中各词语出现的次数，便于接下来计算词频。

对步骤 S24 作进一步说明：

转为为稀疏向量可以降低向量维度，数据格式为“index:value”；词频是指一个词语在文本中出现的频率，词频$tf_{i,j}=\frac{n_{i,j}}{\sum_k n_{k,j}}$，其中$n_{i,j}$表示第$j$个文本中词语$\mathrm{t}_i$的个数，通过该公式计算得到所有文本的词频向量。

对步骤 S26 作进一步说明：

该步骤统计了各词语在多少个文本中出现过，便于接下来结算逆文本频率；S27 是对各分布式节点 S26 操作的统计。

对步骤 S28 作进一步说明：

词语$\mathrm{t}_{i,j}$的逆文本频率指数$idf_i=\log\frac{|D|+1}{\left|\{j:i\in d_j\}\right|+1}$，$|D|$表示所有文本数目，$\left|\{j:i\in d_j\}\right|$表示词语$\mathrm{t}_i$所在文本个数，通过该公式计算得到所有文本的逆文本频率向量。

对步骤 S29 作进一步说明：

TF-IDF 权值$tf \cdot idf_{i,j} = tf_{i,j} \times idf_i$，通过该公式计算得到所有文本的 TF-IDF 向量。

对步骤 S3 作进一步说明：

将总$TF-IDF$向量广播到各分布式节点上，计算每个文本与总$TF-IDF$向量中该文本之前的所有文本的余弦相似度；计算每个文本与总$TF-IDF$向量中该文本之前的所有文本的余弦相似度，从这$i-1$个余弦相似度中取出最大值$\max_{i,j}$，即第i个文本与前$i-1$个文本的余弦相似度中的最大值；创建共享向量Data，维度与文本数量一致，以存放聚类结果；根据上述余弦相似度对所述文本进行改进后的并行化聚类，增量式地获取多个文本簇。

进一步地，图 4 是本发明实施例的改进的并行化文本聚类方法步骤的流程示意图，如图 4 所示，该步骤包括：

（1）将总 TF-IDF 向量广播到各分布式节点上,遍历各节点每个文本；

（2）判断若该文本为总 TF-IDF 向量的首个文本则跳到步骤（7），若不是则往下执行；

（3）计算每个文本与总 TF-IDF 向量中该文本之前的所有文本的余弦相似度；

（4）从中取出最大值 max；

（5）判断是否 Max > Threshold，若是到达步骤（6），若不是则到达步骤（7）；

（6）将得到最大余弦相似度的两个文本归为同一文本簇；

（7）为该文本新建一个文本簇。

对步骤（3）作进一步说明：

余弦相似度计算公式$\cos\theta = \frac{d_i \cdot d_j}{|d_i| \times |d_j|}$，其中$i > j$，$d_i$表示第$i$个文本的$TF-IDF$向量，进行余弦相似度计算时都应按照总 TF-IDF 向量的文本排列顺序进行计算。

对步骤（5）作进一步说明：

本实施例中将 Threshold 设为 0.3 有较好的聚类结果，该阈值可根据实际情况进行设置。

在本发明实施例中，采用并行化的方式对各节点上的文本进行预处理、分词、过滤停用词、特征提取、聚类处理，增量式地获取多个文本簇；通过将聚类过程中的各个步骤并行化，在面对海量或高维数据时，提升了文本聚类的速度。

上述实施例的各步骤可以由处理器执行软件指令的方式来实现。软件指令可以由相应的软件模块组成，软件模块可以被存放于随机存取存储器 RAM、只读存储器 ROM、硬盘、光盘或者本领域熟知的任何其它形式的存储介质中。

尽管上面对本发明说明性的具体实施例进行了描述，以便于本技术领域的技术人员理解本发明，但应该清楚，本发明不限于具体实施方式的范围，对本技术领域的普通技术人员来讲，只要各种变化在所附的权利要求限定和确定的本发明的精神和范围内，这些变化是显而易见的，一切利用本发明构思的发明创造均在保护之列。

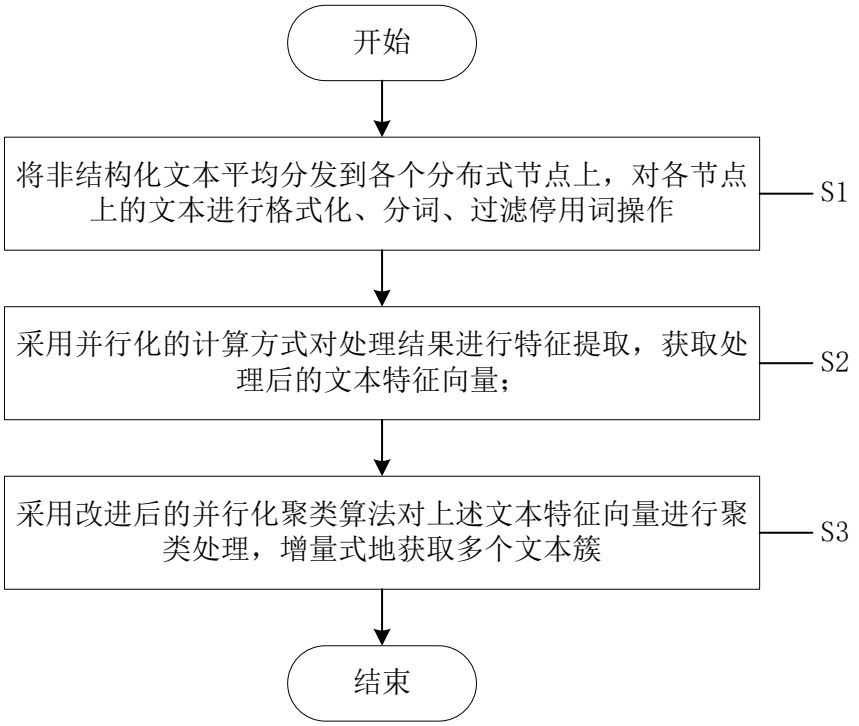

开始
将非结构化文本平均分发到各个分布式节点上，对各节点上的文本进行格式化、分词、过滤停用词操作
S1
采用并行化的计算方式对处理结果进行特征提取，获取处理后的文本特征向量；
S2
采用改进后的并行化聚类算法对上述文本特征向量进行聚类处理，增量式地获取多个文本簇
S3
结束

图 1

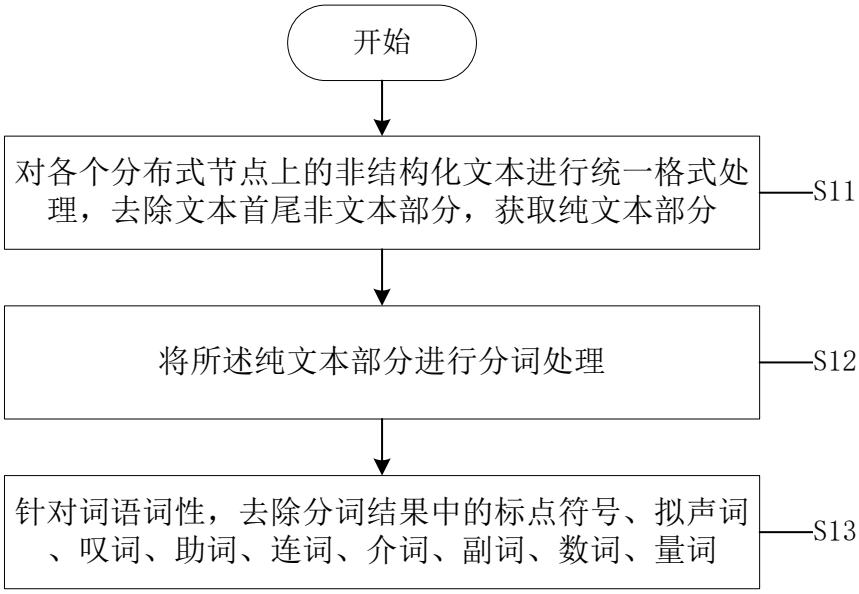

开始
对各个分布式节点上的非结构化文本进行统一格式处理，去除文本首尾非文本部分，获取纯文本部分
S11
将所述纯文本部分进行分词处理
S12
针对词语词性，去除分词结果中的标点符号、拟声词、叹词、助词、连词、介词、副词、数词、量词
S13

图 2

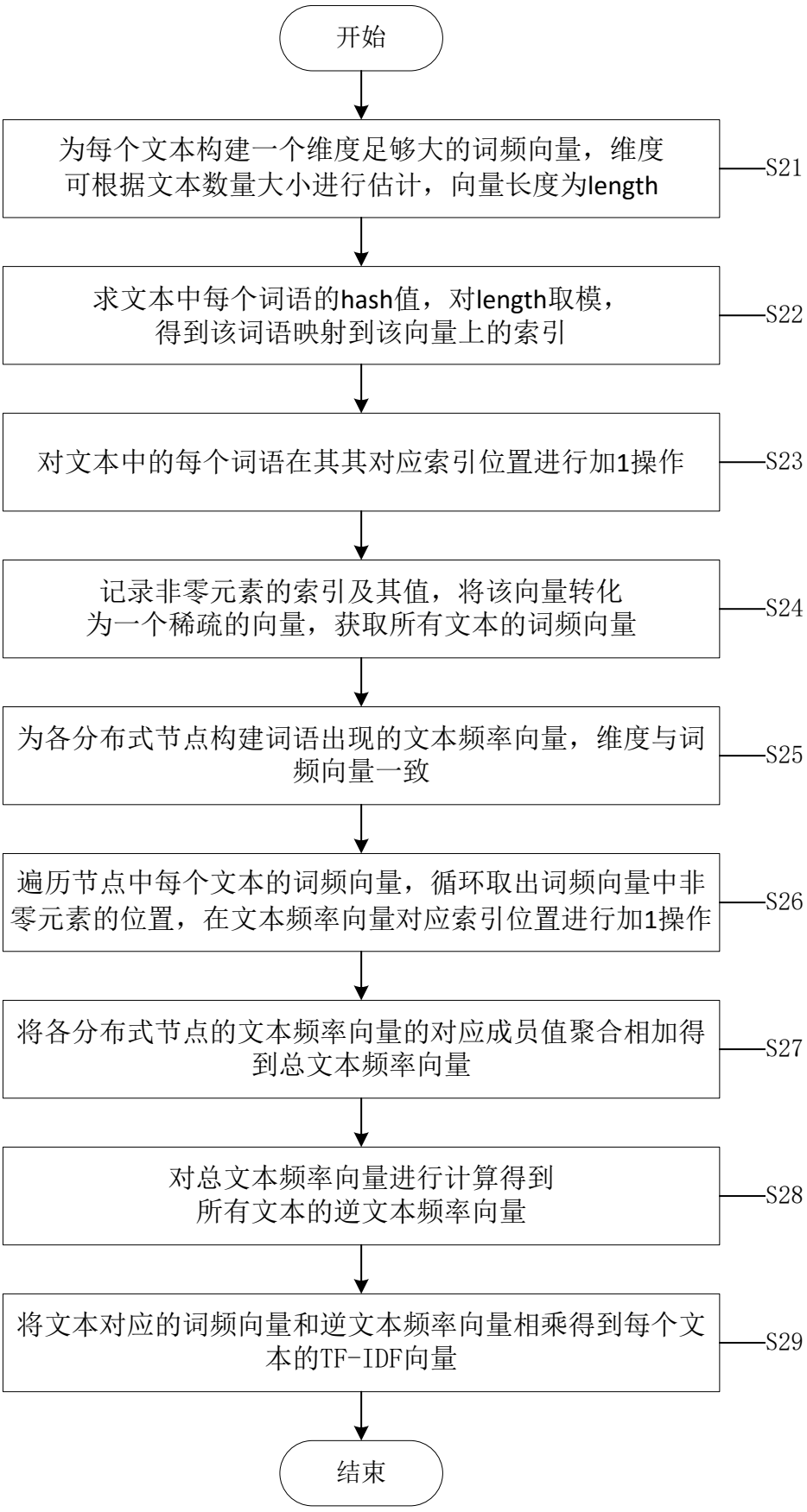

开始
为每个文本构建一个维度足够大的词频向量，维度可根据文本数量大小进行估计，向量长度为length
S21
求文本中每个词语的hash值，对length取模，得到该词语映射到该向量上的索引
S22
对文本中的每个词语在其其对应索引位置进行加1操作
S23
记录非零元素的索引及其值，将该向量转化为一个稀疏的向量，获取所有文本的词频向量
S24
为各分布式节点构建词语出现的文本频率向量，维度与词频向量一致
S25
遍历节点中每个文本的词频向量，循环取出词频向量中非零元素的位置，在文本频率向量对应索引位置进行加1操作
S26
将各分布式节点的文本频率向量的对应成员值聚合相加得到总文本频率向量
S27
对总文本频率向量进行计算得到所有文本的逆文本频率向量
S28
将文本对应的词频向量和逆文本频率向量相乘得到每个文本的TF-IDF向量
S29
结束

图 3

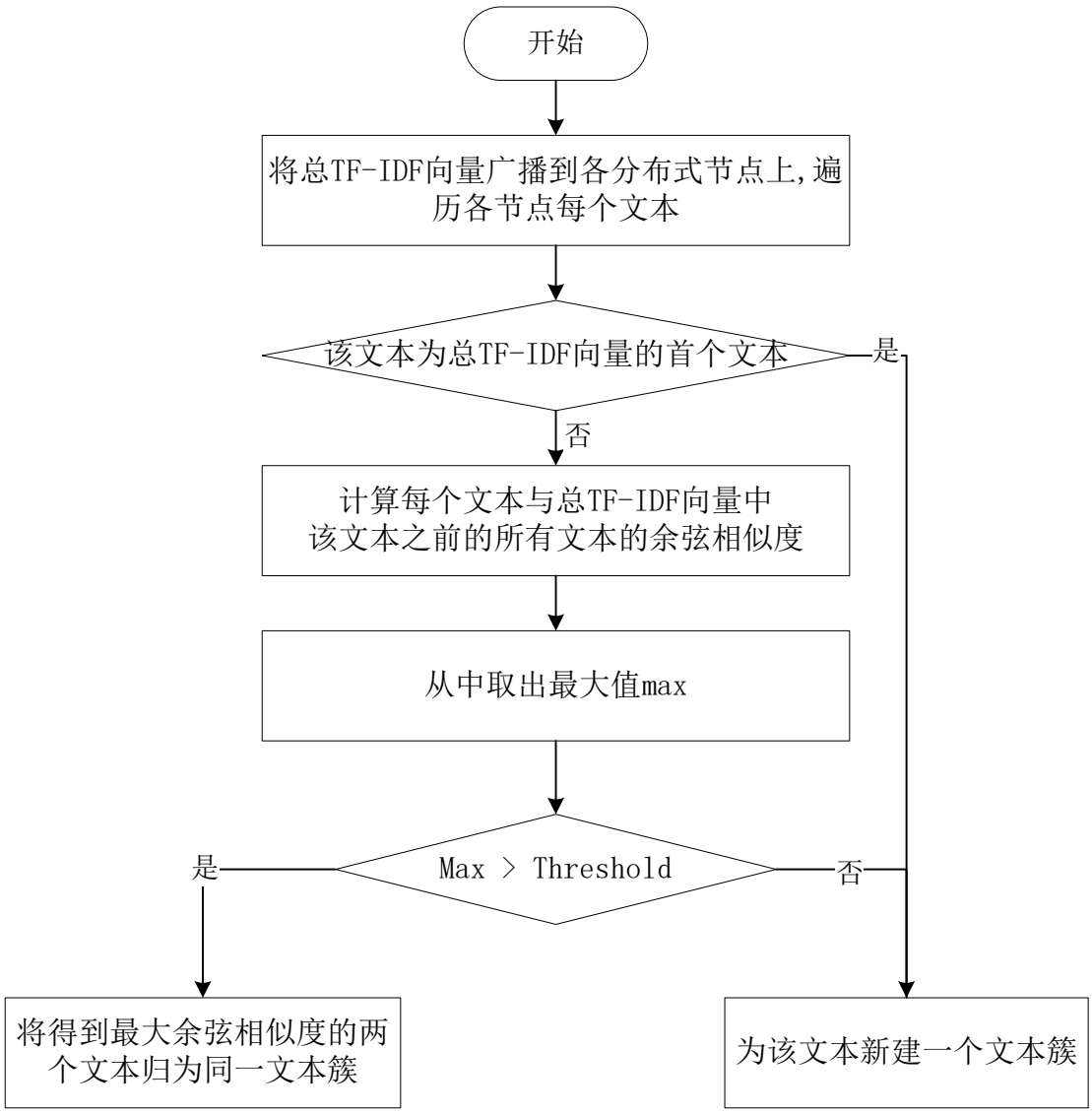

开始
将总TF-IDF向量广播到各分布式节点上,遍历各节点每个文本
该文本为总TF-IDF向量的首个文本
是
否
计算每个文本与总TF-IDF向量中该文本之前的所有文本的余弦相似度
从中取出最大值max
Max > Threshold
是
否
将得到最大余弦相似度的两个文本归为同一文本簇
为该文本新建一个文本簇

图 4