基于上下文相似度矩阵的Single-Pass短文本聚类

黄建一 李建江 王 铮 方明哲
(北京科技大学计算机与通信工程学院 北京 100083)

摘 要 在线社交网络已经成为人们信息交流的重要渠道和载体,形成了与现实世界交互影响的虚拟社会。众多的网络事件通过社交网络进行快速传播,可以在短时间内成为舆论热点,而负面事件会对国家安全和社会稳定造成冲击,从而引发一系列的社会问题。因此,挖掘社交网络中蕴含的热点信息,无论是从舆论监督方面还是舆情预警方面都具有重要的意义。文本聚类是挖掘热点信息的一种重要方法,然而,使用传统长文本聚类算法处理海量短文本时准确率将变低,复杂度急剧增长,从而导致耗时过长;现有的短文本聚类算法的准确率偏低、耗时过长。文中基于文本关键词,提出了结合上下文和相似度矩阵的关联模型,从而判断当前文本与上一文本的关联性。此外,根据该关联模型对文本关键词权重进行调整,以进一步降低噪声。最后,在Hadoop平台上实现了分布式的短文本聚类算法。与K-MEANS,SP-NN,SP-WC算法的比较实验验证了所提算法在话题挖掘速度、准确率和召回率等方面都具有更好的效果。

关键词 在线社交网络,短文本序列,文本聚类,分布式处理

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.04.008

Single-Pass Short Text Clustering Based on Context Similarity Matrix

HUANG Jian-yi LI Jian-jiang WANG Zheng FANG Ming-zhe
(School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China)

Abstract Online social network has become an important channel and carrier, and it has formed a virtual society interacting with the real world. Numerous network events rapidly spread through social networks, and they can become hot spots in a short period of time. However, the negative events vibrate national security and social stability, and may cause a series of social problems. Therefore, mining hotspot information contained in social networks is of great significance both in public opinion supervision and public opinion early warning. Text clustering is an important method for mining hotspot information. However, when the traditional long text clustering algorithms process massive short texts, their accuracy rate will become lower and the complexity will increase sharply, which will lead to long time-consuming. The existing short text clustering algorithms also have low accuracy and takes too much time. Based on the keywords of text, this paper presented an association model combining context and similarity matrix to determine the relevance between the current text and the previous text. In addition, the text keyword weights were modified according to the association model to further reduce the noise. Finally, a distributed short text clustering algorithm on Hadoop platform was implemented. Through the experiments, it is verified that the proposed algorithm has better results and performance compared with K-MEANS, SP-NN and SP-WC algorithms in terms of the speed of mining topics, the accuracy and the recall rate.

Keywords Online social network, Short text sequence, Text clustering, Distributed processing

1 引言

在移动互联网时代,随着人们生活节奏的不断加快,简洁、快速地用短语进行交流互动更符合人们的生活。例如人们经常使用几个简单的关键词通过搜索引擎查询信息;使用微信、QQ等聊天交友。挖掘短文本信息可以获取相关的商业信息、人际关系信息、热点新闻、趋势信息等内容,以及对历史事件进行相关分析总结。为了使系统能够自动挖掘短文本

到稿日期:2018-08-27 返修日期:2018-12-17 本文受国家重点研发计划资助项目(2017YFB0803302),中央基本业务费(06116104)资助。
黄建一(1989-),男,博士生,主要研究方向为社交网络分析与人工智能,E-mail:huangjianyi_ustb@163.com;**李建江**(1971-),男,博士,副教授,主要研究方向为并行计算、云计算、大数据,E-mail:lijianjiang@ustb.edu.cn(通信作者);**王 铮**(1986-),男,讲师,主要研究方向为社交网络分析与人工智能;**方明哲**(1989-),男,博士生,主要研究方向为在线社交网络信息溯源分析。

信息，现有的重要技术手段之一是通过聚类将相似文本组织在一起，从而形成一个话题。

各类社交网络应用产生了海量的短文本，具有持续性、无限性和动态性等特点[1]。文本序列中包含大量嘈杂冗余的信息；文本内容来源多样，而且其内容和受关注程度处于动态变化之中[2]，话题中心也随着时间不断变化[3]；相邻文本间的上下文相关性非常强；信息时效性要求越来越高。这些因素使得通过及时、准确地检测和追踪话题来获取相关信息的难度越来越大。

传统的长文本聚类分析一般是基于文本的特征向量进行关联性判定，如典型的基于划分的聚类分析、基于层次的聚类分析以及基于密度的聚类分析[4-6]。基于划分的聚类算法(Partition-based Methods)的总体思想是将一个具有 N 条记录的文本集最终划分为 K($K \ll N$)个分组，每一个分组表示一个类别，聚类结果的评定标准为组内的记录关联性越近越好，不同组间的记录关联性越远越好，并且每一条记录必须属于唯一一个分组，每一个分组中至少包涵一个记录[7-9]。典型的 K-MEANS 算法[10]、K-MEDOIDS 算法[11]以及 CLARANS 算法[12]都是基于划分思想的聚类算法。此类算法的优点是：简单、高效；缺点是不能很好地处理噪音及孤立点[13]，而且聚类效果对于初始值的选取有很大的依赖性，导致其聚类消耗的时间很不稳定，而目前并没有很好的方法选择合适的初始值，在处理大数据时很难估计处理文本所需的时间，无法满足时效性要求，同时数据集较大时结果容易陷入局部最优。基于层次的聚类算法(Hierarchical Methods)是按层次对给定的待处理文本集进行关联性判定，如此循环，逐层判定，直至达到设定的退出条件为止[14-16]。层次聚类算法按照处理方式的不同，可以分为“自底向上”的层次聚类和“自顶向下”的层次聚类两种不同的聚类方式。以“自底向上”的层次聚类为例，初始时将每一个元素当成一个单独的类，之后对这些类之间进行关联分析，把达到一定关联阈值的类合成一个，之后再对新得到的类进行同样的判定，直至满足算法设定的退出条件为止[17-18]。基于层次思想聚类的算法有很多，典型的有 BIRCH 算法[19]、CURE 算法[20]以及 CHAMELEON 算法[21]等。相比基于划分的聚类算法，层次聚类更加灵活，但其计算量更大，时间复杂度更高，适用于较小的数据集，且在特征很少的情况下判断类簇相似度时存在较大的误差，致使后续判断受到影响。基于密度的聚类算法(Density-based Methods)并不是根据元素与类之间的距离进行相似性判定，而是根据元素分布的密度进行划分，因此不存在基于距离判定文本相似性所造成的过多依赖文本分布形状的不足。其判定依据是如果某个区域中点的密度大于设定的阈值，则将该点归属到与之相邻的聚类中[21-22]。常见的 DBSCAN 算法[23]、OPTICS 算法[24]以及 DENCLUE 算法[25]都属于基于密度的聚类算法。DBSCAN 算法的一个不足之处在于其对设定的扫描半径及最小包含点数极其敏感，这两个参数的细微变动都会对聚类效果造成很大的影响。但遗憾的是，这两个参数的选择并没有固定的规律可循，只能根据经验进行人为设定。

这些方法都是采用词频-逆向文件频率(Term Frequency-Inverse Document Frequency，TF-IDF)计算关键词的权值，根据文本间的特征向量计算两文本之间的距离或者对两文本之间的紧密程度进行判定[26-27]。这些方法对长文本能取得不错的聚类效果，因为长文本一般有很多关键词，即使挖掘的特征向量存在一定的误差，对聚类效果也不会存在很大的影响。社交网络短文本具有“精简”的特点[28]，导致文本向量的稀疏程度大；同时，通过 TF-IDF 计算的关键词的权值非常小，几乎为 0，在计算文本之间的相似度时会存在很大的误差，而且浪费了不少的 CPU 处理时间及内存资源。当处理的话题种类较多以及文本数据量较大时，若根据每个话题下所有文本来计算话题中心，每次进行相似度判断都要对所有话题中心进行比较，则随着数据规模增长，计算量呈指数级增长，导致算法性能急速下降，耗时迅速增加。因此，这种全局范围内的文本向量表示法不适合社交网络的短文本聚类算法。

近年来，专门针对短文本聚类的算法层出不穷，但大多都是对传统聚类算法进行改进[29-30]，如 WR-KMEANS[31]等，其在一定程度上提高了短文本聚类的效果，但准确率依旧较低，且计算量较大，耗时过长，不能很好地解决海量数据处理的问题。Shen 等[32]提出了 5 种 Single-Pass 聚类算法：SPB，SP-WC，SP-NN，SP-WNN，SP-LF。Single-Pass 聚类算法是数据流聚类的经典方法，可以实时或者离线处理数据集，有效应对文本内容的动态变化，并且算法的可解释性强。其缺点是对文本输入的顺序相对敏感，但对类似新闻信息这类按时序严格组织的文本的聚类结果的影响不大；对于每一个新文本，都要计算其与已有文本的相似度，随着数据规模的增长，计算复杂度也急剧增长。其他的热门方法如词共现聚类的方法[33-34]在论文数据集上的效果很好，但是该方法对词的选择非常敏感，而社交网络存在各种不规范用语，导致其在处理社交网络短文本时会出现不确定性问题。因此，针对社交网络中的短文本聚类问题，迫切需要寻找新的方法。

基于上述问题，本文结合社交网络短文本的特点，提出了基于上下文相似度矩阵的 Single-Pass 聚类方法(Single Pass-Context Similarity Matrix，SP-CSM)。该算法首先从短文本的关键词入手，利用上下文信息计算相邻文本的关键词相似度矩阵，来判断当前文本与上一文本的关联性，避免了短文本关键词较少带来的稀疏性问题。每个新文本不需要与已有的所有文本进行相似度比较，只需要计算与已有话题的最后一个文本构成的文本集合间的关联度，避免了数据规模不断增长而导致算法复杂度急剧增长的问题。通过对已有文本数据的更新机制，保证了算法性能的稳定、高效和准确。然后针对关联文本，对关键词权重进行调整，突出重要关键词，抑制次要关键词；同时，设置阈值来调节文本在每个关键词上的偏移程度，检测出与主题无关的信息，过滤掉大量嘈杂冗余的信息，从而提高算法性能和准确率。实验结果表明，所提算法与

K-MEANS,SP-WC,SP-NN 等算法相比,无论是从话题挖掘速度,还是从准确率和召回率方面,都具有更好的效果。

2 问题描述与模型

2.1 基本定义

文本序列是一组顺序、大量、快速、连续到达的数据序列,可被视为一个随时间延续而无限增长的动态数据集合。本文首先对文本序列进行形式化描述,在此基础上,定义滑动时间窗口及其内的文本集合。

定义 1 在时序文本数据中,在时间上连续的文本数据项及其时间窗口构成一个文本序列 $TS=\{x_1,x_2,\cdots,x_k,\cdots\}$,其中 x_k 是第 k 个到达的文本。

定义 2 如图 1 所示,t 表示时间,p 表示时间间隔,时间序列按$(t,t+p]$分割,滑动时间窗口(Sliding Time Window)定义为$(t,t+p]$上的文本集合:

$$SW_t=TS_t^{t+p} \tag{1}$$

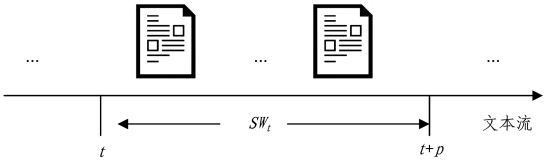

图 1 滑动时间窗口

Fig. 1 Sliding time window

假设 SW_t 的长度为 p,文本流的速度为 v,则 SW_t 的文本总量 S_t 可由式(2)计算:

$$S_t=v\times p \tag{2}$$

当 $p=1$(单位时间),$v=1$(一条文本)时,$S_t=1$,表示单位时间间隔内只有一条文本进入时间窗口,也就是经典的 Single-Pass 过程。如果 p 为一个常数,则 SW_t 是一个稳定滑动的时间窗口;如果 p 是可变的,则 SW_t 是一个动态的窗口,具有更好的灵活性,复杂性也更高。

2.2 关键词抽取

社交网络短文本的关键词数量很少,每一个关键词对描述该文本相关事件信息都有很大的贡献值。如果抽取的关键词不够准确,就会存在很大的误差,从而对聚类效果产生很大的影响。因此,抽取的关键词要尽可能反映出该文本所包含的相关信息,本文将从文本中提取相关的人物、地点及相关事件信息作为文本的关键词。

一个典型的微博文本一般包含微博作者的昵称、@其他人、微博的具体内容(事件的相关信息和涉及的人物)。分词系统不能很好地处理社交网络中的昵称,因此本文以“@”和其后面的“空格”或者“@”和其后面的“:”为分界线(标识符)来抽取文本的人物信息。地名抽取直接根据分词系统提供的词性即可完成。由于挖掘的事件关键词在动词之后,因此本文以句为单位,将每句中词性为动词性质的词作为开始记录,之后将其后一定距离内的名词性质的词作为事件关键词。具体伪代码如算法 1 所示。

算法 1 关键词抽取算法

输入:文本 x_k 分词后的词序列 list

输出:$W_k=\{w_1^k,w_2^k,\cdots,w_M^k\}$

```
1. for word in list
2.   if word 词性为 nr/nr2/nrj/nrf/nz
3.     添加 word 到 Wk 中;//word 是真实人名
4.   else if word 在@后面
5.     添加 word 到 Wk 中;//word 是昵称
6.   else if word 词性为 ns/nsf
7.     添加 word 到 Wk 中;//word 为地名
8.   else if word 在动词后,word 为名词,且 length<=3
9.     添加 word 到 Wk 中;//word 为事件关键词
10.  end if
11. end for
12. return Wk;
```

2.3 关联模型

为了克服现有方法在处理社交网络短文本时出现的稀疏性问题,本文根据同一话题序列文本在内容上具有非常强的关联性这一特点,利用时序最近邻的两个文本构建相似度矩阵来建立局部的关联模型,以此判断文本之间的关联性。

假设当前处理文本为 $x_k=(u_k,W_k,t)$,$W_k=\{w_1^k:m_1^k,w_2^k:m_2^k,\cdots,w_M^k:m_M^k\}$,其中 w_i^k 为 x_k 的第 i 个关键词,m_i^k 为 w_i^k 的频次;L 为所有话题序列中最后一个文本构成的集合,并按时间降序排序;$x_{k-1}=(u_{k-1},W_{k-1},t)$,$W_{k-1}=\{w_1^{k-1}:m_1^{k-1},w_2^{k-1}:m_2^{k-1},\cdots,w_N^{k-1}:m_N^{k-1}\}$,且 $x_{k-1}\in L$。word2Vec[35-37] 表示每个关键词用词向量(50 维)表示,w_i^k 的词向量为 vec_i^k,w_j^{k-1} 的词向量 vec_j^{k-1}。用余弦距离计算相似度,获得文本 x_k 和 x_{k-1} 的相似度矩阵 $\boldsymbol{C}=(c_{ij})_{M\times N}$,$w_{ij}$ 可由式(3)计算:

$$c_{ij}=vec_i^k\cdot vec_j^{k-1} \tag{3}$$

根据相似度矩阵 $\boldsymbol{C}$ 可知文本 x_k 的每个关键词在时序最近邻文本 x_{k-1} 的每个关键词上的偏移情况。为了量化关键词的偏移情况,将相似度矩阵 $\boldsymbol{C}$ 中元素高于阈值的置为 1,低于阈值的置为 0,得到关键词偏移矩阵 $\boldsymbol{D}=(d_{ij})_{M\times N}$,$d_{ij}$ 可由式(4)计算。如果某行都为 0,即表示该关键词和上一文本所有关键词的相似度都较小,可以判断为噪声。x_k 的权重向量与关联话题文本权重向量相乘,保留主要关键词,去除噪声,并重新分配主要关键词的权重,从而增强主要关键词,降低噪声的影响。

$$d_{ij}=\begin{cases}1, & \text{if } c_{ij}=\max(\boldsymbol{C}[i][*])\geqslant\lambda\\ 0, & \text{if } c_{ij}\leqslant\max(\boldsymbol{C}[i][*])<\lambda\end{cases} \tag{4}$$

其中,$\boldsymbol{C}[i][*]$表示相似度矩阵 $\boldsymbol{C}$ 的第 i 行;阈值 $\lambda\in(0,1]$用于衡量文本在每个关键词上的偏移程度。

为了判断新文本与已有话题的关联性,文本 x_k 和 x_{k-1} 的关联度 sim_k 可由式(5)计算:

$$sim_k=n_k/M \tag{5}$$

其中,n_k 为矩阵 $\boldsymbol{D}$ 中的非零行数。

聚类流程如图 2 所示。如果一个文本中有至少一半的关键词和另一文本的关键词的相似度超过阈值,则两个文本相

关联，否则这两个文本的关联度不大。因此，$sim_k \geqslant 0.5$，即文本 x_k 和 x_{k-1} 相关联，用文本 x_k 替换 L 中的文本 x_{k-1}，并且将 x_k 加入到文本 x_{k-1} 所属的话题文本序列中。这里主要考虑：1)时序因素，选择时序最近且相似度大于阈值的话题；2)降低计算量，提升算法性能。若 $sim_k < 0.5$，则文本 x_k 和文本 x_{k-1} 没有关联，根据时序最近邻原则，x_{k-1} 依次遍历 L 直至遍历完 L 中的所有文本。若文本 x_k 和 L 中的所有文本均没有关联，则将文本 x_k 作为一个新的候选话题序列初始文本。具体伪代码如算法 2 所示。

算法 2　聚类算法

输入：文本序列 TS={x_1，x_2，…，x_k，…}，L

输出：话题及其文本序列

```
1. for x_k in TS
2.    sort(L);//将 L 按时间降序排序
3.    flag=0;//记录遍历 L 后的结果
4.    for x_{k-1} in L
5.       通过 word2Vec 将 x_{k-1}和 x_k的关键词转化为词向量;
6.       计算 x_k和 x_{k-1}的相似度矩阵 C;
7.       计算关键词偏移矩阵 D;
8.       计算矩阵 D 中非零行数 n_k;
9.       sim_k=n_k/M;
10.      if sim>=0.5
11.         flag=1;
12.         将 x_k加入到文本 x_{k-1}所属的话题文本序列中;
13.         用 x_k替换掉 L 中的文本 x_{k-1};
14.         break;
15.      end if
16.   end for
17.   if flag=0
18.      将 x_k加入到 L 中;
19.      将 x_k作为新的话题序列初始文本;
20.   end if
21. end for
22. return 话题及其文本序列;
```

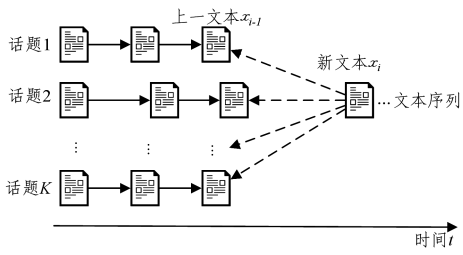

图 2　聚类流程

Fig. 2　Clustering process

海量的社交网络短文本中，话题种类和数量众多，每个话题中的文本规模分布不均，存在大量的孤立点和噪声点。为了消除话题序列集合 TP 中的孤立点以及防止文本集合 L 的规模无限增长，保证算法稳定、高效、准确，需要定时更新 TP 和 L。给定任意话题序列 $TP_k \subset TP$，TP_k 为话题 z_k 起始至今所有关联文本构成的集合，$lifet_k$ 为话题 z_k 起始至今的时间间隔，l_k 为 TP_k 的文本总数。本文对 434 个话题进行测试，结果如图 3 所示，横坐标为 x 时间(小时)内话题文本数达到 10 所需的时间，纵坐标为 x 时间(小时)内文本数达到 10 的话题数量。可以看出，在 1 h 内，大部分话题的文本数都能达到 10。本文更关注热点话题，虽然会遗漏话题初始到爆发前极少的零散文本，但话题开始爆发后的文本均可收集分析；另外，相比话题初始阶段，本文更加关注话题爆发后的舆情。因此，如果 $lifet_k > 1$ h 且 $l_k < 10$，则删除 TP 中的 TP_k，同时删除 L 中的 x_k，x_k 为 TP_k 中时序上最新的文本。时间 t 为文本 x_k 的到达时间，保存 L 在 $t-G\times p$ 时间前的副本 L'，G 为正整数。设定 $p=1$ min，$G=5$，用来控制时间间隔，即每 5 min 检测一次话题是否有新文本到来。如果 $L \cap L'$ 不为空，则表示 L 中存在某些文本持续未更新，即部分话题没有新文本到来，更新 $L=L-L\cap L'$，并将 $L\cap L'$ 中的文本所在话题移出 TP，将其保存在外部存储中。具体伪代码如算法 3 所示。

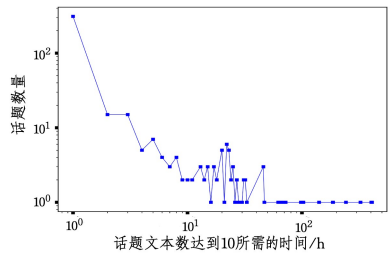

图 3　在 X 时间内话题的文本数达到 10 的话题分布情况

Fig. 3　Distribution of topics whose text number reach 10 in X hours

算法 3　更新算法

输入：话题序列集合 TP，TP_k，文本集合 L，L′，话题 z_k

输出：更新后的话题序列集合 TP，文本集合 L

```
1. for TP_k in TP
2.    if lifet_k>1h and l_k<10 //长时间文本过少
3.       delete TP_k in TP;TP_k//为噪声或孤立点
4.       delete x_k in L;//x_k∈TP_k 且 x_k∈L
5.    else:L∩L′//存在话题持续没有新文本到来
6.       L=L-L∩L′;//更新 L
7.       for x_k in L∩L′
8.          保存 TP_k 到外部文件 z_k.txt;
9.          delete TP_k in TP;//更新 TP
10.      end for
11.   end if
12. end for
13. return TP and L;
```

2.4　降噪处理

任何与数据上下文和最终输出无关的文本都可被判作噪声。文本是非结构化数据，存在各种各样的噪声。噪声会增加算法耗时、内存占用，更严重的是导致算法的准确率降低。因此，消除噪声并尽可能减少噪声带来的不利影响具有非常重要的意义。

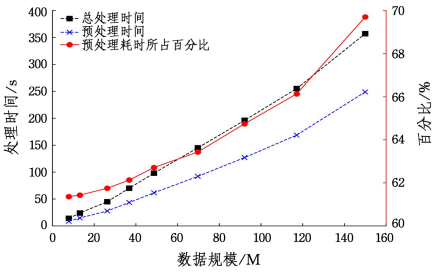

图 4 单机模式下预处理时间测试

Fig. 4 Pretreatment time test in stand-alone mode

本文首先在预处理中通过整理的停用词表(1 900 个)过滤掉停用词及无关字符,另外通过 2.3 节提出的关联模型对噪声进一步过滤。

x_k 的初始权重向量为 $\alpha_k=[\alpha_1{}^k,\alpha_2{}^k,\cdots,\alpha_M{}^k]$,$w_i^k$ 的初始权重 α_i^k 可由式(6)计算:

$$\alpha_i^k=\frac{m_i^k}{\sum_{i=1}^{M}m_i^k},\text{且 }\alpha_i^k\in[0,1],\sum_{i=1}^{M}\alpha_i^k=1 \tag{6}$$

x^{k-1} 的权重向量为 $\beta_{k-1}=[\beta_1^{k-1},\beta_2^{k-1},\cdots,\beta_N^{k-1}]$,$\boldsymbol{D}^{\mathrm{T}}[*][i]$ 表示矩阵 $\boldsymbol{D}$ 转置后的第 i 列,文本 x_k 的关键词 w_i^k 的权重 β_i^k 可由式(7)计算:

$$\beta_i^k=\frac{\alpha_i^k\times[\beta_1^{k-1},\beta_2^{k-1},\cdots,\beta_N^{k-1}]\times\boldsymbol{D}^{\mathrm{T}}[*][i]}{\sum_{j=1}^{M}\alpha_j^k\times[\beta_1^{k-1},\beta_2^{k-1},\cdots,\beta_N^{k-1}]\times\boldsymbol{D}^{\mathrm{T}}[*][i]} \tag{7}$$

通过文本 x_{k-1} 的关键词权重调整文本 x_k 的关键词权重,目的在于突出重要关键词,抑制次要关键词。此外,如果 all($\boldsymbol{D}[i][*]$)=0,其中 $\boldsymbol{D}[i][*]$ 为矩阵 $\boldsymbol{D}$ 的第 i 行,则表示文本 x_k 的关键词 w_i^k 为该文本内的噪声,删除该关键词,并重新归一化文本 x_k 的所有关键词权重。

2.5 聚类算法的 MapReduce 实现

社交网络中的短文本具有更新快、数量大的特点,如果采用单机模式对其进行处理,在数据量不是很大的情况下尚能满足需求;但是若数据量过大,单机模式下受到 CPU 处理能力以及内存大小的限制,则无法满足数据处理要求。本文经过对单机模式下的聚类程序测试发现,程序 60%~70%的运行时间消耗在分词、去除停用词、特征抽取等预处理上,而具体的聚类运算的时间只是占小部分,如图 4 所示。由于文本预处理任务相互之间没有依赖关系,可以同时分别进行预处理。因此,如果将预处理部署在分布式平台上,可在很大程度上提高程序的处理性能。本文采用 Hadoop 分布式平台对海量短文本进行预处理,将文本数据分配到各个分布式节点上进行预处理,将 Map 输出的 key 值设为相同的值,各节点的处理结果通过一个 Reduce 函数合并在一起,从而获得所有的 Map 输出,即所有文本的关键词,之后再对这些文本进行聚类分析。MapReduce 的实现方式如下:

1)Jobtrack 负责任务的分配和均衡调度,按照时间间隔 p 将文本流分割成 N 份 SW_t 数据,并分发任务数据到各分节点(Task Tracker)。

2)Map 函数得到〈行号,行内容〉的键值对,对行内容进行文本关键词提取,得到文本的关键词,输出〈"文本关键词",文本 ID_该条文本关键词〉的键值对。

3)Combine 过程对 key 相同的键值对进行合并操作,由于 Map 的输出 key 均为"文本关键词",因此会将所有的 Map 输出合并成一个形如〈"文本关键词",〈文本 1 关键词,文本 2 关键词,…〉〉的键值对,得到 N 个〈key,list〈value〉〉的键值对序列。

4)Reduce 过程中一个 reduce 函数获取了所有的 Map 输出,合并成文本序列,文本 $x_k=(u_k,W_k,t)$,$W_k=\{w_1^k:m_1^k,w_2^k:m_2^k,\cdots,w_M^k:m_M^k\}$,$TS=\{x_1,x_2,\cdots,x_k,\cdots\}$。之后进行聚类分析和热点挖掘。

3 实验结果与分析

本文实验主要是对单机实现的 SP-CSM 算法以及 Map-Reduce 模式下的 SP-CSM 算法从运行时间、加速比、准确率、召回率等方面进行相关性能的综合评测。实验环境为 11 个节点的 Hadoop 分布式处理平台,处理器为 16 * Intel(R) Xeon(R) CPU E7320 @ 2.13 GHz,服务器内存为 16 GB 内存、1 GB 缓存以及 2 GB 交换区,操作系统为 64 位 Linux 操作系统(Red Hat 4.1.2-42)。

实验 1 采用准确率、召回率和 F1 值来评估算法结果的精度。通过已标注好的 6 000 条新浪微博(2013 年 6 月的数据,共包含 10 个话题类别)对单机模式下的 SP-CSM 算法进行评测,算法的准确率和召回率分别取各个话题类别下的准确率和召回率的均值,结果如表 1 所列。

表 1 算法的准确率和召回率

Table 1 Precision and Recall of different algorithms

(单位:%)

算法	准确率	召回率	F1 值
SP-CSM	**84.39**	**85.73**	**85.05**
SP-WC	76.34	75.56	75.95
SP-NN	80.13	79.76	79.94
K-MEANS	68.94	70.71	69.81
LDA	82.87	82.68	82.77

表 1 中的测试结果显示,SP-CSM 在准确率、召回率以及 F1 上均有着相对较高的指标,说明 SP-CSM 算法在判定话题类别上具有不错的性能。

实验 2 测试 SP-CSM 算法在单机模式下和 MapReduce 模式下的处理性能。在单机模式和 MapReduce 模式下(6 个分布式节点)对 SP-CSM 算法与 K-MEANS 算法处理不同规模数据的耗时进行对比实验,结果如图 5 所示。

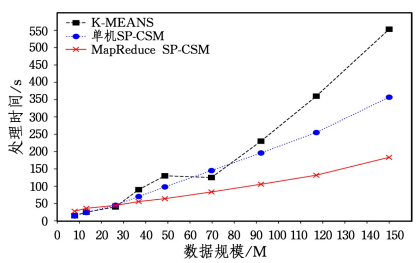

图 5 不同算法的耗时对比

Fig. 5 Processing time of different methods

可以看出，在数据量不大的情况下，SP-CSM 算法在单机模式下的耗时比 MapReduce 模式下的耗时更低，这是因为 Hadoop 平台本身是为处理大数据而设计的，在数据量很小的情况下并不能发挥它的优势，其反而因网络传输及任务调度的消耗增加了耗时。但是随着数据量的增加，SP-CSM 算法的耗时近似线性增长，而 K-MEANS 算法的耗时呈不稳定增长状态，这主要是因为 K-MEANS 算法的性能与初始点的选取有很大的关系。在数据量很大的情况下，K-MEANS 算法的耗时急剧增长；相比 K-MEANS 算法，SP-CSM 算法的耗时明显降低，尤其是 MapReduce 模式下的 SP-CSM 算法的耗时大幅度降低。这说明 K-MEANS 算法不能很好地解决大数据聚类的问题，而 SP-CSM 算法的耗时随着数据量的增长在一定范围内几乎呈线性增长，说明 SP-CSM 算法具有更好的稳定性及可扩展性。在数据量很大的情况下，Hadoop 平台发挥了其优势，对大数据有较高的吞吐率。

实验 3　测试 MapReduce SP-CSM 算法的可扩展性。加速比是同一个任务在单处理器系统和并行处理器系统中运行所消耗的时间的比率，用于衡量并行系统或程序并行化的性能和效果。对 SP-CSM 算法在 MapReduce 模式下使用不同节点数时处理 150 M 文本集（大约 40 万条微博）的耗时和加速比进行实验，结果如图 6 所示。可以看出，MapReduce 模式下的 SP-CSM 算法的耗时随着节点数的增加而减小，加速比随节点数的增加而增加，说明 MapReduce 模式下的 SP-CSM 算法具有不错的可扩展性。

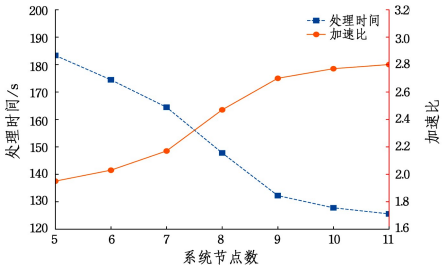

图 6　在 Hadoop 平台上的可扩展性测试

Fig. 6　Scalability performance test on Hadoop

结束语　本文提出了针对社交网络短文本聚类的 SP-CSM 算法，能够检测和追踪话题。通过相邻文本的关键词相似度矩阵判断当前文本与上一文本的关联性，解决了稀疏性问题。将话题检测与追踪技术融合在一套系统中，通过分析算法的瓶颈，将 SP-CSM 算法 MapReduce 并行化，很大程度上提高了程序运行的效率，解决了聚类算法随着数据量增长而导致性能下降的问题。通过与 K-MEANS，SP-WC，SP-NN 等算法的比较实验可知，SP-CSM 在运行时间、准确率、召回率及可扩展性等方面都具有更好的效果。本文对 SP-CSM 只实现了局部并行化，虽然很大程度上提高了程序的性能，但是聚类部分依旧为单机处理，没有很好地利用 Hadoop 各节点的处理能力，SP-CSM 的性能有待进一步提升。因此下一步工作将对 SP-CSM 算法进行改进和优化，并对社交网络短文本序列进行演化预测分析。

参 考 文 献

[1] NGUYEN H L，WOON Y K，NG W K. A survey on data stream clustering and classification[J]. Knowledge and Information Systems，2015，45(3)：535-569.

[2] HUANG J，PENG M，WANG H，et al. A probabilistic method for emerging topic tracking in microblog stream[J]. World Wide Web，2017，20(2)：325-350.

[3] XIE W，ZHU F，JIANG J，et al. TopicSketch：real-time bursty topic detection from Twitter[C]// International Conference on Data Mining. 2013：837-846.

[4] LI X H，HE T N，RAN H Y，et al. A novel graph partitioning criterion based short text clustering method[C]// International Conference on Intelligent Computing. Springer，Cham，2016：338-348.

[5] BEIL F，ESTER M，XU X. Frequent term-based text clustering[C]// Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM，2002：436-442.

[6] SALLOUM S A，AL-EMRAN M，MONEM A A，et al. A survey of text mining in social media：facebook and twitter perspectives[J]. Adv. Sci. Technol. Eng. Syst. J，2017，2(1)：127-133.

[7] ALI A，QADIR J，RASOOL R U，et al. Big data for development：applications and techniques[J]. arXiv：Computers and Society，2016，1(2)：1-24.

[8] HUANG J，PENG M，WANG H，et al. A probabilistic method for emerging topic tracking in Microblog stream[J]. World Wide Web，2017，20(2)：325-350.

[9] ALLAN J，CARBONELL J，DODDINGTON G，et al. Topic detection and tracking pilot study：final report[C]// Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop. 1998：194-218.

[10] WAYNE C L. Topic detection &-tracking (TDT)[C]. Workshop held at the University of Maryland on. 1997.

[11] CAPO M，PEREZ A，LOZANO J A，et al. An efficient approximation to the K-means clustering for massive data[J]. Knowledge Based Systems，2017，117：56-69.

[12] ARORA P，VARSHNEY S. Analysis of K-Means and K-Medoids algorithm for big data[J]. Procedia Computer Science，2016，78：507-512.

[13] NG R T，HAN J. CLARANS：A method for clustering objects for spatial data mining[J]. IEEE Transactions on Knowledge and Data Engineering，2002，14(5)：1003-1016.

[14] ABUALIGAH L M，KHADER A T. Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering[J]. The Journal of Supercomputing，2017，73(11)：4773-4795.

[15] PINTO，DAVID，et al. A Self-Enriching Methodology for Clustering Narrow Domain Short Texts[J]. The Computer Journal，2011，54(7)：1148-1165.

[16] PINTO D，BENEDÍ J M，ROSSO P. Clustering narrow-domain

short texts by using the Kullback-Leiblerdistance[M]. Computational Linguistics and Intelligent Text Processing. Springer Berlin Heidelberg,2007:611-622.

[17] HU X,SUN N,ZHANG C,et al. Exploiting internal and external semantics for the clustering of short texts using world knowledge[C]// Proceedings of the 18th ACM Conference on Information and Knowledge Management. ACM,2009:919-928.

[18] THOMAS R E,KHAN S S. Co-Clustering with side information for text mining[C]// International Conference on Data Mining. 2016:105-108.

[19] BHANUSE S S,KAMBLE S D,KAKDE S,et al. text mining using metadata for generation of side information[J]. Procedia Computer Science,2016,78:807-814.

[20] HAHSLER M,BOLAOS M. Clustering data streams based on shared density between micro-clusters[J]. IEEE Transactions on Knowledge and Data Engineering,2016,28(6):1449-1461.

[21] STEINBACH M,KARYPIS G,KUMAR V. A comparison of document clustering techniques[C]// KDD Workshop on Text Mining. 2000:525-526.

[22] KARYPIS G,HAN E H,KUMAR V. Chameleon:Hierarchical clustering using dynamic modeling[J]. Computer,1999,32(8):68-75.

[23] SCHUBERT E,SANDER J,ESTER M,et al. DBSCAN revisited:why and how you should (still) use DBSCAN[J]. ACM Transactions on Database Systems (TODS),2017,42(3):19.

[24] GAO T,LI A,MENG F,et al. Research on data stream clustering based on FCM algorithm[J]. Procedia Computer Science,2017,122:595-602.

[25] REHIOUI H,IDRISSI A,ABOUREZQ M,et al. DENCLUE-IM:a new approach for big data clustering[J]. Procedia Computer Science,2016,83:560-567.

[26] SPARCK J K. A statistical interpretation of term specificity and its application in retrieval[J]. Journal of Documentation,1972,28(1):11-21.

[27] CONG Y,CHAN Y,RAGAN M A. A novel alignment-free method for detection of lateral genetic transfer based on TF-IDF [J]. Scientific Reports,2016,6(1):30308.

[28] GUO L,VARGO C J,PAN Z,et al. Big social data analytics in journalism and mass communication:Comparing dictionary-based text analysis and unsupervised topic modeling[J]. Journalism & Mass Communication Quarterly,2016,93(2):332-359.

[29] ALLAHYARI M,POURIYEH S,ASSEFI M,et al. A brief survey of text mining:classification,clustering and extraction techniques[J]. arXiv preprint arXiv:1707.02919,2017.

[30] XU J,XU B,WANG P,et al. Self-taught convolutional neural networks for short text clustering[J]. Neural Networks,2017,88:22-31.

[31] LI X H,HE T N,RAN H Y,et al. A novel graph partitioning criterion based short text clustering method[C]// International Conference on Intelligent Computing. Springer,Cham,2016:338-348.

[32] SHEN D,YANG Q,SUN J T,et al. Thread detection in dynamic text message streams[C]// Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM,2006:35-42.

[33] KENTER T,DE RIJKE M. Short text similarity with word embeddings[C]// Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM,2015:1411-1420.

[34] AKIMUSHKIN C,AMANCIO D R,OLIVEIRA J O N. Text authorship identified using the dynamics of word co-occurrence networks[J]. PloS one,2017,12(1):e0170527.

[35] MIKOLOV T,SUTSKEVER I,CHEN K,et al. Distributed representations of words and phrases and their compositionality[C]// Advances in Neural Information Processing Systems. 2013:3111-3119.

[36] BOJANOWSKI P,GRAVE E,JOULIN A,et al. Enriching word vectors with subword information[J]. Transactions of the Association for Computational Linguistics,2017,5(1):135-146.

[37] LI C,WANG H,ZHANG Z,et al. Topic Modeling for Short Texts with Auxiliary Word Embeddings[C]// International Acm sigir Conference on Research and Development in Information Retrieval. 2016:165-174.