计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(4):139−144 [doi: 10.15888/j.cnki.csa.006852]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于频繁模式的长尾文本聚类算法①

宋中山, 张广凯, 尹　帆, 帖　军

(中南民族大学 计算机科学学院, 武汉 430074)
通讯作者: 帖　军, E-mail: tiejun@mail.scuec.edu.cn

摘　要: 短文本聚类一直是信息提取领域的热门话题, 大规模的短文本数据中存在“长尾现象”,传统算法对其聚类时会面临特征纬度高, 小类别信息丢失的问题, 针对对上述问题的研究, 本文提出一种频繁项协同剪枝迭代聚类算法 (Frequent Itemsets collaborative Pruning iteration Clustering framework, FIPC). 该算法将迭代聚类框架与 K 中心点算法相结合, 运用协同剪枝策略, 实现对小类别文本聚类, 实验结果证明该聚类算法能够有效的提高小类别短文本信息聚类的精确度, 并能避免聚类中类簇重叠的问题.

关键词: 文本聚类; 长尾现象; 频繁模式; K 中心点算法

引用格式: 宋中山,张广凯,尹帆,帖军.基于频繁模式的长尾文本聚类算法.计算机系统应用,2019,28(4):139–144. http://www.c-s-a.org.cn/1003-3254/6852.html

Long Tail Text Clustering Algorithm Based on Frequent Patterns

SONG Zhong-Shan, ZHANG Guang-Kai, YIN Fan, TIE Jun

(School of Computer Science, South-Central University for Nationalities, Wuhan 430074, China)

Abstract: Short texts clustering is a popular topic in the field of information extraction. There is a “long tail phenomenon” when the scale of data is large, which causes high dimensions of features and information loss of small class. To solve these problems, this study proposes a Frequent Itemsets collaborative Pruning iteration Clustering framework (FIPC). This framework combines the iterative clustering framework with the K-mediods algorithm, using the collaborative pruning strategy to cluster text of small class. The result of experiments shows that the FIPC framework can achieve text clustering of small class with high accuracy, and avoid the problem of overlapping clusters.

Key words: text clustering; long tail phenomenon; frequent mode; K-mediods algorithm

Twitter、微博信息传递的形式为短文本.短文本最大的特点是单条短文本只有几十个字节大小, 仅包含几个到十几个词典词语, 很难准确抽取有效的语言特征[1]. 这类非规范化严重的短文本, 具有特征信息不足, 特征维度高, 数据稀疏性高等特点[2,3]. 因此使用传统聚类方法对此类短文本聚类时难度较大.

“长尾现象”普遍存在口语化的短文本集中. 大约 40% 的短文本信息集中分布在大约 20% 的空间中, 也就是“头”的部分, 称之为“热点”, 大约 60% 的信息分布在大约 80% 的空间中, 即“尾”的部分, 将所有非热点信息累加起来就会形成一个比主流信息量还要大的信息[4]. 从这些“尾”部的信息中收集到有用的信息对于政府或者一些投资商了解社会异常以及人们日常动向有很大的帮助[5,6]. 传统聚类算法, 主要通过一次筛选后得到频繁词来表征短文本, 因“长尾”部分短文本的特征词权重很小, 达不到传统算法中筛选阈值, 所以在挖掘频繁

① 基金项目: 国家科技支撑计划项目子课题 (2015BAD29B01); 农业部软科学研究课题 (D201721); 中央高校基本科研业务费专项资金 (CZY18016)
Foundation item: Sub-program of National Sci-Tech Supporting Plan (2015BAD29B01); Soft Science Research Project of Ministry of Agriculture (D201721); the Fundamental Research Funds for the Central Universities (CZY18016)
收稿时间: 2018-10-15; 修改时间: 2018-10-31, 2018-11-14; 采用时间: 2018-11-19; csa 在线出版时间: 2019-03-28

词时, 会忽略掉小类别短文本的特征词, 造成小类别短文本的特征信息不足, 在聚类结果中小类别短文本会被划分到不正确的簇里面, 导致聚类的精确度降低, 因此本文围绕提高“长尾”部分短文本聚类精确度的问题, 提出一种频繁项协同剪枝迭代聚类算法 (FIPC), 首先提取频繁词构建频繁词-文本矩阵, 接着使用 K 中心点算法对文本进行初始聚类, 然后根据协同剪枝策略对原始数据集进行剪枝得到下一次迭代聚类的文本集, 重复进行上叙过程, 直到得到对小类别短文本聚类的结果簇, 从而实现“长尾”文本的聚类. 实验结果证明, 与传统的 K-means 聚类和 FIC 算法相比, 该算法能够有效的避免类簇重叠问题, 提高了“长尾”短文本聚类精确度.

1 相关工作

目前国内外的众多学者已经对于短文本聚类方面技术有了相关研究. 基于频繁模式的文本聚类算法, 该类算法通过以频繁模式的方式, 表征文本进行聚类[7–9]. 如: 栗伟等人提出了 ACT 算法[10], 该算法将挖掘频繁项来表征文本, 以频繁项确定文本簇的中心点, 对文本进行完全聚类. 该算法所面对的数据集为疾病的病例数据, 此类数据格式规范且种类少. 但是该算法在进行日常口语化文本信息提取时, 算法效率较低. 彭敏[11]等人提出了一种基于频繁项集的短文本聚类与主题抽取 STC-TE 框架, 该框架首先对于海量短文本数据进行打分数筛选, 留下得分高类别大的文本, 结合谱聚类算法 CSA_SC 算法与主题抽取模型, 实现短文本聚类效果. 该算法在处理社交网络中的短文本信息时, 前期处理筛掉非频繁的短文本, 造成了大量的信息缺失, 对一些小的类别文本没有进行聚类, 精度不高, 效果不佳. 基于子树匹配的文本聚类算法, 该类算法利用文本生成元数据特征向量, 设置分层权重结合语义之间关系构建文本子树, 通过子树之间的相似度来进行文本匹配[12,13]. 该算法不能解决多义词的问题, 这对于日常短文本的聚类效果不佳. 基于语料库的短文本聚类算法, 该类算法在不借助外部文本信息的情况下, 引入共轭定义来表征主题词和单词结构, 提出文本虚拟生产过程, 达到解决文本稀疏性和聚类问题的目的. 如: Zheng CT, Liu C, Wong HS[14]提出的基于主题扩展的文本聚类算法, 结合特征空间及语义空间达到提高短文本聚类精度的效果, 但该算法在对于具有“长尾现象”的文本数据聚类时效果不佳. 基于主题模型的文本聚类算法, 该类算法主要结合主题模型与传统算法, 来对海量短文本进行聚类. 如: Hung PJ, Hsu PY, Cheng MS[15]的动态主题的 Web 文本聚类, 结合 EM 算法与动态主题为文本聚类特征对 Web 文本进行聚类; 张雪松和贾彩燕提出的 FIC 算法, 首先挖掘频繁词集表示文本, 构建文本网络, 运用社区划分算法对网络进行大范围划分, 最后提取主题词, 进行类簇划分[16]. 该算法在处理“长尾现象”的短文本数据时, 聚类精度不高. 针对此类问题, 彭泽映等人[17]在对实际应用中短文本信息的“长尾现象”进行分析后, 提出不完全聚类的思想, 即在聚类的过程中集中资源处理大类别的短文本, 减少资源在孤立点聚类上的浪费, 尽量减少小类别的短文本的聚类时间, 增加大类别的短文本聚类机会. 但该算法在处理社交网络口语化, 小类别文本数繁多的应用中, 容易丢失文本信息, 精确度较低.

针对以上短文本聚类面临的特征高维, 小类别信息丢失的问题, 本文提出频繁项协同剪枝迭代聚类算法 (FIPC), 通过挖掘频繁词集, 根据相关文本相似性实现文本聚类, 得到部分聚类结果, 根据协同剪枝策略, 生成主题词检索并且剔除相关短文本, 迭代进行此聚类过程, 进而实现短文本聚类. 具体来说主要有以下 2 点贡献: (1) 采用逐步降低频繁词的筛选阈值, 让权重较小的频繁词被选中来表示短文本的特征, 解决了传统聚类中小类别信息被忽略的问题, 同时避免了类簇重叠问题的出现; (2) 充分利用类簇主题词与文本之间关系, 设计协同剪枝策略, 减小迭代聚类中每一轮数据集, 减小了每轮聚类的时间消耗.

2 模型描述

2.1 频繁词挖掘模型

定义 1. 文本数据集 D 由多个文本组成 $D=\{d_1,d_2,\cdots,d_N\}$, N 表示文本集的大小总数, 以及文本切词之后得到的词集 V, $V=\{t_1,t_2,\cdots,t_n\}$, n 表示词汇表的大小.

定义 2. 特征词: 每一个文档d_N经切词之后得到词集$V=\{t_1,t_2,\cdots,t_n\}$, 词集中的每一个词, 称为特征词t_n.

定义 3. 文本集 D 中第 j 个文本的每一个频繁词对应词集 V 构成的词空间的一个维度, w_M表示每一个频繁词对应的权重, 即词空间中的每个维度的坐标.

对于特征词权重的计算方法运用 TF-IDF 算法. 其

中tf_{ij}表示在文本d_j中的特征词t_i的出现的标准化词频, 则在文本d_j中t_i标准化词频 (记做tf_{ij}) 计算如下:

$$tf_{ij}=\frac{f_{ij}}{\max\{f_{1j},f_{2j},\cdots,f_{Mj}\}} \tag{1}$$

其中, 特征词t_i的标准化逆文档频率记做 idf_i其计算公式如下:

$$idf_i=\log\frac{N}{df_i} \tag{2}$$

在短文本中不同词性重要程度不同, 代表文本的重要程度不同, 我们给每一个特征词根据词性赋予一个权值α_i.

最后文本d_j中每一个特征词的最终权重w_i为词频因子 (tf_{ij}) 与逆文档频率 (idf_i) 与词性权重α_i三者的乘积. 如式 (3) 所示:

$$w_i=tf_{ij}*idf_i*\alpha_i \tag{3}$$

由式 (3) 所示特征词的权重由三方面的因素来决定: 第一方面因素为该特征词在这个短文本中词频因子; 第二方面为该特征词在文本集里面的逆文档频率因子; 第三方面为特征词的词性因子[18].

定义 4. 频繁词集: 从词集 V 挖掘权重大于频繁词阈值Y_2的频繁词f_k组成的集合$F_i=\{f_i,f_2,\cdots,f_k\}$.

2.2 文本表示模型

向量空间模型 (Vector Space Model, VSM) 是最常用的文本表示模型[19]. VSM 模型采用特征词表征文本构建矩阵, 这对于矩阵的维度比较高, 本文运用频繁词表征文本, 选前 K 个频繁词构建频繁词-文本矩阵 L, L 为 0-1 矩阵, 其中$L[i][j]$表示矩阵 L 中文本d_j对于频繁词f_i的值, 若文本d_j中含有频繁词f_i, 则$L[i][j]=1$, 否则$L[i][j]=0$. L 的表现形式为:

$$L[i][j]=\begin{cases}1, & f_i\in d_j\\ 0, & f_i\notin d_j\end{cases}\quad(0\le i\le k) \tag{4}$$

选用前 K 个频繁词来构建矩阵 L, 而不选择所有的特征词, 这样降低了矩阵的维度, 同时也解决了文本稀疏性问题.

文本向量化后的对文本进行相似度计算, 传统的相似度计算的方法有几种, 例如余弦相似度和欧氏距离. 本文采用余弦相似度来度量:

$$\cos\theta=\frac{\vec{d_q}*\vec{d_p}}{\left|\vec{d_q}\right|\left|\vec{d_p}\right|} \tag{5}$$

其中, $\vec{d_q}$和$\vec{d_p}$分别代表文档向量化后的向量, 设定相似度余弦阈值$Q(Threshold)$, 若两者相似度余弦阈值大于$Q(Threshold)$, 则将两者归于为 1 个类簇.

对于所有文本运用 K 中心点算法对其两两进行相似度计算如式 (6) 所示, 将相似的文档来归于到一个类簇中.

$$MaxSim(d_i,d_j)=\sum_{\vec{d_i}\in d_i}\sum_{\vec{d_j}\in d_j}Sim(\vec{d_i},\vec{d_j}) \tag{6}$$

以两个向量之间相似度值$Sim(\vec{d_i},\vec{d_j})$来衡量两个文本之间的相似度值, 当$Sim(\vec{d_i},\vec{d_j})$大于相似度阈值时, 则认为两个文本相似.

2.3 协同剪枝策略

如何利用主题词与短文本之间的关系, 这在提高“长尾”文本聚类的精确度方面有重要的作用. 对于上一次聚类结果簇, 提取主题词T_{Ci}. 根据以下规则进行剪枝策略以及迭代聚类.

规则一: 对于初始样本文本集$D=\{d_1,\cdots,d_N\}$, 初始聚类后得到初始结果簇C_i, 从C_i选取频繁词作为主题词T_{Ci}, 对于初始数据集中每一个文本d_N, 若表示d_N的频繁词集中包含T_{Ci}, 则从初始样本文本集 D 当中剔除掉文本d_N, 余下文本集作为下一次聚类的输入文本.

规则二: 为了防止短文本中孤立点数据对迭代聚类次数的影响, 设置固定最小权重阈值 P, 多次迭代聚类后, 对余下文本计算特征词权重, 若所有特征词权重中最大的特征词权重值小于最小权重阈值 P, 则迭代聚类结束.

3 FIPC 算法

FIPC (Frequent Itemsets collaborative Pruning iteration Clustering framework) 频繁项协同剪枝迭代聚类算法步骤图如图 1 所示, 该算法首先对初始样本文本集 D 切词得到具有词性标注的特征词, 从其中提取频繁词, 然后构建频繁词-文本向量矩阵, 在文本向量化后利用 K 中心点算法进行聚类[20], 初始聚类结束得到部分聚类结果簇, 提取每一个簇中的主题词, 根据协同剪枝策略对初始样本文本集进行剪枝. 同时减小频繁词阈值Y_2, 再对余下数据集迭代进行第二次、第三次等多次聚类过程, 经过多次聚类之后, 若余下文本特征词满足规则二, 则迭代聚类结束, 最后得到聚类结果簇$\{C_i\}$. 算法中部分参数如表 1 所示.

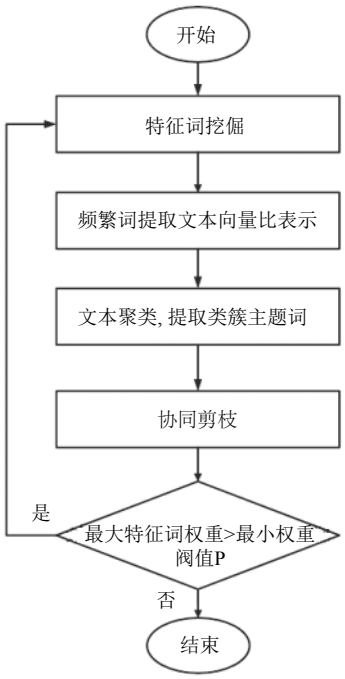

图 1 频繁项协同剪枝迭代聚类算法步骤图

表 1 部分参数含义表

参数	说明
f_i	文本集特征词t_i的词频
d_{ji}	文本d_j中的特征词t_i
f_{ji}	文本d_j中的频繁词f_i
T_{Ci}	类簇主题词
w_{ij}	文本d_j中t_i的权重
C_i	聚类文本簇

实验中采用频繁词阈值动态变化逐步减小的规律, 上一次实验选取频繁词阈值高, 代表文本可信度较高, 最后将文本分到指定簇的可信度高于其后实验可信度. 使得每一个文本能被唯一指派到唯一的类簇中, 避免了类簇重叠问题的生产.

算法描述如下:

算法 1. 频繁项协同剪枝聚类算法

输入: 最小权重阈值 P, 频繁词阈值Y_2, 文本数据集 D, 相似度余弦阈值$Q(Threshold)$.

输出: 最终聚类结果簇$\{C_i\}$.

1) 对样本文本数据集进行分词, 得到具有词性标注的特征词.

2) 依据式 (1)、(2)、(3), 计算每个特征词的权重w_i.

3) 根据频繁词阈值Y_2挖掘频繁词f_i表征文本d_N, 构建频繁词-文本表示矩阵, 文本$\vec{d_N}$表现形式为: $\vec{d_N}$={1,0,⋯}.

4) 依据式 (5) 计算表示文本向量之间的相似性.

5) 根据向量之间余弦值与相似度余弦阈值$Q(Threshold)$的大小, 将两个文本进行相似度的比较.

6) 随机选 K 个向量作为初始聚类中心点, 运用 K 中心点算法进行聚类, 得到初始类簇C_{i1}, 选取簇中频繁词作为主题词T_{Ci}.

7) 根据协同剪枝策略删除原始数据集中与主题T_{Ci}相关的文本.

8) 减小频繁词阈值Y_2, 对余下数据集进行下一次特征词权重计算, 若其中最大特征词权重值大于 P, 则跳转到步骤 1), 重复执行上序步骤 1 到 8, 且$Y_2=Y_2-\varepsilon$得到下一次聚类结果簇C_{i2}.

9) 若最大特征词权重值小于 p, 则全部算法结束.得到最终聚类结果簇$\{C_i\}=\{C_{i1},C_{i2},\cdots,C_{in}\}$.

4 实验结果与分析

4.1 数据集

其中, 文本分类语料库搜狗新闻数据集[①]包含 9 个新闻类, 共有 17 910 个文本; NLPIR 微博英文语料库[②], 包含的英文文档数为 23 万, 其中数据集的文本结构为: Id: 文章编号、article:正文、discuss: 评论数目、insertTime: 正文插入时间、origin: 来源、person_id: 所属人物的 id、time: 正文发布时间、transmit: 转发. 本文中抽取 article 中的文本短内容进行聚类, 从搜狗数据集随机选取部分数据进行实验, 如表 2 所示.

表 2 搜狗数据描述

种类名称	文档数量
IT	549
财经	472
体育	599
健康	236
教育	458
军事	375

4.2 聚类评价指标

为了测试 FIPC 算法的性能, 我们需要选择聚类评价指标, 聚类评价指标分为内部评价指标和外部评价指标. 我们采用文本聚类中常用的外部评价标准 F-measure[21], 它经常被用作衡量聚类方法的精度, 是一种平面和层次聚类结构都适用的评价标准. F-measure 综合了召回率和准确率 2 种评价标准:

$$Recall(K_i,C_j)=\frac{n_{ij}}{|k_i|} \tag{7}$$

$$precision(K_i,C_j)=\frac{n_{ij}}{|C_j|} \tag{8}$$

① http://www.sougo.com/labs/resources/list_yuliao.php

② http://www.nlpir.org/

其中, n_{ij}表示簇C_j中属于类K_j的文本数, 由召回率和准确率可得到表示簇C_j描述类K_j的能力计算:

$$F(K_i,C_j)=\frac{2*Recall(K_i,C_j)*precision(K_i,C_j)}{Recall(K_i,C_j)+precision(K_i,C_j)} \quad (9)$$

聚类的总体 F-measure 值取值范围在[0, 1], 值越大表示聚类效果越好.

4.3 实验方案设计

为了验证本文提出的文本聚类方法, 我们选用 2 个文本聚类中标准的数据集来进行实验, 对照常用的基于 K-means 文本聚类方法和 FIC 算法.

对于原始数据集首先利用 python 中的结巴分词(基于机器学习的中文自然语言文本处理的开发工具) 进行切词, 样本文本经过切词、剔除停用词以及词性标注操作之后, 得到具有词性标注的特征词t_i, 但仍然存在这大量与主题无关或者无意义的词语. 因此在本文中需要选取阈值来筛掉不相关和无意义的词语.

本文中有 3 处参数需要涉及到阈值的选取, 包括挖掘频繁词的频繁词阈值Y_2、计算余弦相似度时余弦阈值$Q(Threshold)$、实验结束时最小权重阈值 P. 本文通过手动调参, 多次试验的方式, 来获得聚类最佳效果的参数阈值, 其中在频繁词减小的过程中, 要保证每次表征文本的频繁词数不少于 5 个, 选择最后实验结束的最小权重阈值 P 时, 最少保证词频最大的频繁词所出现文本的频数不小于 2. 对于本次实验, 在英文数据集中, 选取相似度余弦阈值步长为 0.005, 由于中文数据中的停用词较多以及每个文档词数量较少, 因此采用与英文数据不同的阈值选择方式, 采用 0.01 的步长, 并且以对聚类精度影响最高的相似性余弦阈值进行实验.

4.4 实验分析

我们在不同算法上分别对搜狗数据集和 NLPIR 微博内容语料库进行试验, 其中对这 3 种算法进行 10 次实验取平均值作为最后聚类的精度结果. 对于实验初始值聚类的中心点数 K 值我们设置为 5, 实验过程中对于聚类结果不在我们初始簇类别里的, 则设为一个新的簇类. 实验中固定最佳的最小权重阈值$P=0.003$和频繁词筛选阈值$Y_2=0.020$, 如图 2 所示为相似度余弦阈值$Q(Threshold)$取不同值时 FIPC、FIC 和 K-means 算法在搜狗数据集下的 F-measure 的变化曲线图.

图 3 为 3 种算法在固定最佳的最小权重阈值$P=0.003$和频繁词筛选阈值$Y_2=0.020$之后取不同相似度余弦阈值$Q(Threshold)$时, 在 NLPIR 微博英文料库上的 F-measure 变化曲线图.

对于本次实验最后产生的类簇进行主题词提取, 我们选用提取频繁词来作为主题词. 统计该类簇中频繁词出现频率, 并且按照前 10 的频繁词来描述主题词.

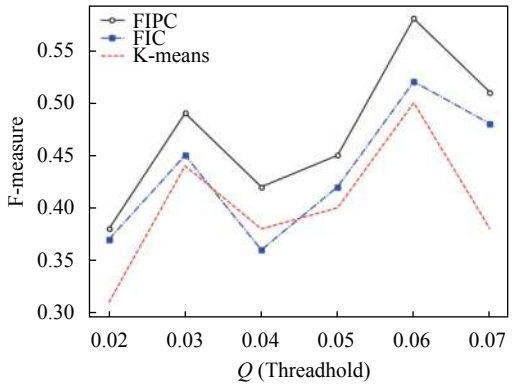

图 2 搜狗中文数据集下三种聚类算法的精度

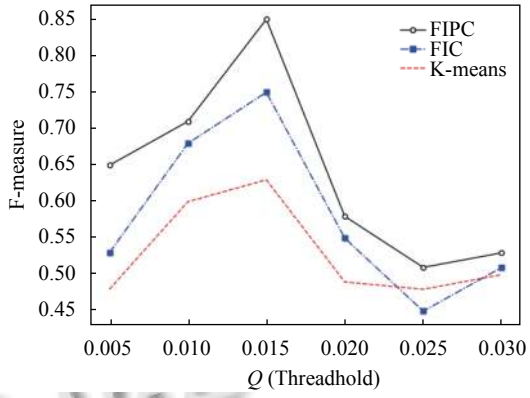

图 3 NLPIR 微博数据集下三种聚类算法的精度

如表 3 所示, 可以看出 FIPC 算法较之其他方法在精度上面有明显的优势, 原因有以下几点:

(1) 在数据处理方面, 并不只是分词和去掉停用词, 为了保留具有代表性的词汇, 采用 TF-IDF 值结合词性来筛选特征词.

(2) 运用频繁词来构建文本表示模型, 文本集中挖掘出频繁词, 有效的降低了文本表示矩阵的维度.

(3) 本文采用协同剪枝的策略, 结合主题词与文本矩阵进行协同剪枝, 缩小了数据集的大小.

经过 10 次试验取平均值作为最后聚类结果得出以下 3 个算法在 2 个不同数据集中的精确度.

由表 3 中可以看出 FIPC 算法聚类出来的 F-measure 值相较于其他 2 个算法略大, 因此体现出该算

法的聚类精确度最好.

表 3 算法 F-measure 值

算法	搜狗数据集	NLPIR 语料库
K-means	0.4152	0.4043
FIC	0.5677	0.5285
FIPC	0.6015	0.5932

5 结语

本文针对短文本“长尾现象”这一特点, 提出一种新的文本聚类算法 FIPC, 该算法基于频繁词表征文本, 解决了高稀疏性的问题, 将经典聚类算法 K 中心点应用到迭代聚类的框架中, 实现对小类别文本进行聚类, 更精确的挖掘小类别短文本的信息, 提升了聚类的准确性.

本文中采用的 K 中心点算法结合协同剪枝策略, 但是 K 中心点算法一直存在一个问题:如何选取合适的初始中心点, 本文中采用人工随机选取初始中心点的方法, 若能在选取合适的初始中心点对于实验结果的精确度能有更大的提高. 因此在下一步工作中如何有效快捷的选取合适的初始中心点来进行聚类是我们所要思考的.

参考文献

1 丁兆云, 贾焰, 周斌. 微博数据挖掘研究综述. 计算机研究与发展, 2014, 51(4): 691–706.

2 Zhao Y, Liang SS, Ren ZC, *et al*. Explainable user clustering in short text streams. Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. Pisa, Italy. 2016. 155–164.

3 Song HS, Li N, Zhang W. Application of VSM model to document structure identification. Journal of Beijing Information Science and Technology University, 2011, 26(6): 66–69, 75.

4 Dasgupta S, Ng V. Towards subjectifying text clustering. Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Geneva, Switzerland. 2010. 483–490.

5 Hinz O, Eckert J, Skiera B. Drivers of the long tail phenomenon: An empirical analysis. Journal of Management Information Systems, 2011, 27(4): 43–70. [doi: 10.2753/MIS0742-1222270402]

6 Weng JS, Lim EP, Jiang J, *et al*. TwitterRank: Finding topic-sensitive influential twitterers. Proceedings of the 3rd ACM International Conference on Web Search and Data Mining. New York, NY, USA. 2010. 261–270.

7 Marcacini RM, Corrêa GN, Rezende SO. An active learning approach to frequent itemset-based text clustering. Proceedings of the 21st International Conference on Pattern Recognition. Tsukuba, Japan. 2012. 3529–3532.

8 Zhang W, Yoshida T, Tang XJ, *et al*. Text clustering using frequent itemsets. Knowledge-Based Systems, 2010, 23(5): 379–388. [doi: 10.1016/j.knosys.2010.01.011]

9 Su ZT, Song W, Lin MS, *et al*. Web text clustering for personalized e-learning based on maximal frequent itemsets. Proceedings of 2008 International Conference on Computer Science and Software Engineering. Hubei, China. 2008. 452–455.

10 栗伟, 许洪涛, 赵大哲, 等. 一种面向医学短文本的自适应聚类方法. 东北大学学报(自然科学版), 2015, 36(1): 19–23. [doi: 10.3969/j.issn.1005-3026.2015.01.005]

11 彭敏, 黄佳佳, 朱佳晖, 等. 基于频繁项集的海量短文本聚类与主题抽取. 计算机研究与发展, 2015, 52(9): 1941–1953.

12 Singh G, Sundaram S. A subtractive clustering scheme for text-independent online writer identification. Proceedings of the 2015 13th International Conference on Document Analysis and Recognition. Tunis, Tunisia. 2015. 311–315.

13 张佩云, 陈传明, 黄波. 基于子树匹配的文本相似度算法. 模式识别与人工智能, 2014, 27(3): 226–234.

14 Zheng CT, Liu C, Wong HS. Corpus-based topic diffusion for short text clustering. Neurocomputing, 2018, 275: 2444–2458. [doi: 10.1016/j.neucom.2017.11.019]

15 Hung PJ, Hsu PY, Cheng MS, *et al*. Web text clustering with dynamic themes. In: Gong ZG, Luo XF, Chen JJ, et al, eds. Web Information Systems and Mining. Berlin Heidelberg: Springer, 2011. 122–130.

16 张雪松, 贾彩燕. 一种基于频繁词集表示的新文本聚类方法. 计算机研究与发展, 2018, 55(1): 102–112.

17 彭泽映, 俞晓明, 许洪波, 等. 大规模短文本的不完全聚类. 中文信息学报, 2011, 25(1): 54–59. [doi: 10.3969/j.issn.1003-0077.2011.01.009]

18 张群, 王红军, 王伦文. 一种结合上下文语义的短文本聚类算法. 计算机科学, 2016, 43(S2): 443–446, 450.

19 Abu-Salih B. Applying vector space model (VSM) techniques in information retrieval for Arabic language. arXiv: 1801.03627, 2018.

20 邢光林, 胡一然, 孙翀, 等. 改进的 K 中心点算法在茶叶拼配中的应用. 中南民族大学学报(自然科学版), 2017, 36(4): 126–130.

21 Linden G, Smith B, York J. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 2003, 7(1): 76–80. [doi: 10.1109/MIC.2003.1167344]