第 44 卷 第 2 期 计 算 机 工 程 2018 年 2 月
Vol. 44 No. 2 Computer Engineering February 2018

·先进计算与数据处理· 文章编号:1000-3428(2018)02-0017-07 文献标志码:A 中图分类号:TP391

# 一种高效的文本区间热词查询算法

赵志洲,路 畅,何震瀛,王晓阳

(复旦大学 计算机科学技术学院,上海 200433)

**摘 要:** 文本区间热词查询是根据用户指定的查询时间范围,从文本数据中提取热词。现有的热词提取算法主要面向挖掘任务,时间复杂度较高,难以直接应用于热词的在线查询处理。为此,提出一种文本区间热词的在线查询处理算法。利用数据划分和范围查询技术,在准确率和空间复杂度不变的条件下降低提取热词的时间复杂度。实验结果表明,与现有的面向挖掘算法相比,该算法在 CNN、BBC 和 NYT 3 个数据集涉及的整个时间范围上的运行时间分别减少59.7% 、65.1% 和 75.5% ,有效提高热词在线查询的效率。

**关键词:** 热词;时间复杂度;在线查询;数据划分;范围查询

**中文引用格式:** 赵志洲,路 畅,何震瀛,等. 一种高效的文本区间热词查询算法[J]. 计算机工程,2018,44(2):17-23,30.

**英文引用格式:** ZHAO Zhizhou,LU Chang,HE Zhenying,et al. An Efficient Text Interval Hot Word Query Algorithm[J]. Computer Engineering,2018,44(2):17-23,30.

# An Efficient Text Interval Hot Word Query Algorithm

ZHAO Zhizhou,LU Chang,HE Zhenying,WANG Xiaoyang

(School of Computer Science,Fudan University,Shanghai 200433,China)

**【Abstract】** Text interval hot word query is based on user-specified query time range,it extracts from the text data hot words. Existing hot words extraction algorithm is generally oriented to mining tasks,which has a high time complexity and is difficult to be directly applied to an online query processing of hot words. Therefore,an online query processing algorithm for text interval hot words is proposed. Using data partitioning and range search technology,the time complexity of extracting hot words is reduced with the same accuracy and space complexity. Experimental results show that compared with the existing mining-oriented algorithms,the running time of the algorithm is reduced by 59.7% ,65.1% and 75.5% respectively over the entire time range covered by the three CNN,BBC and NYT datasets,which effectively improves the hot words online query efficiency.

**【Key words】** hot word;time complexity;online query;data partition;range query

**DOI:** 10.3969/j.issn.1000-3428.2018.02.003

## 0 概述

互联网高速发展使人们获取海量信息变为可能,但随之带来一些新问题,如“如何从海量 Web 文本信息中提取用户感兴趣的热门话题”。

为有效进行话题检测和跟踪(Topic Detection and Tracking,TDT),研究者开展了大量研究工作[1-2],其中从文本数据中提取热词成为当前研究的热点问题之一[3-5]。文献[6]提出 TF-IDF(Term Frequency-Inverse Document Frequency)用于词权重计算,TF-IDF 综合考虑词频和反文档频率,弱化频繁出现在多个文档中的词的重要性。文献[7]提出 TF-PDF(TF-Proportional Document Frequency)方法,TF-PDF 综合考虑词频和文档频率,将更高的权重赋予出现在多个文档中的词。文献[8](下文称 Chen 算法)在 TF-PDF 方法的基础上,考虑词频随时间的波动情况,并重新定义词权重的计算方法。上述方法能够有效提取与话题相关的词,即满足算法的有效性。文献[9]基于热度矩阵对微博热点话题进行检测。但是这些算法的一个特点是:面向挖掘任务,时间复杂度较高,因此,难以直接应用于热词在线查询问题。

为此,本文对文本数据的区间热词在线查询问题展开研究。热词的在线查询处理方法需要同时满足 2 个特性:有效提取与话题相关的词,即在线查询的有效性;快速获得查询时间范围内的热词,即在线查询

**基金项目:** 国家自然科学基金(61370080);上海市科技创新行动计划项目(16DZ1100200)。

**作者简介:** 赵志洲(1992—),男,硕士研究生,主研方向为数据库技术、数据分析;路 畅,硕士研究生;何震瀛,副教授、博士;王晓阳,教授、博士。

**收稿日期:** 2017-02-24 **修回日期:** 2017-03-24 **E-mail:** zhizhou0019@gmail.com

的时效性。因此,设计同时满足有效性和时效性的热词在线查询方法依然是一个具有挑战性的问题。针对上述方法时效性不足的缺点,本文提出一种文本数据区间热词在线查询处理算法(EHWE),该算法可以在已划分的数据上进行快速区间查询处理。

## 1 相关工作

本文常用到的符号如表 1 所示。

**表 1 符号说明**

| 符号 | 说明 | 符号 | 说明 |
|---|---|---|---|
| $D$ | 整个数据集 | $T$ | $D$ 中的不同词个数 |
| $R$ | $D$ 的时间范围 | $s$ | 单位时间间隔 |
| $t$ | 某一个词 | $k$ | 指定的热词个数 |
| $S$ | $R$ 中 $s$ 的数目 | $q$ | $D$ 上的某一次查询 |
| $\alpha$ | 指定的词频率阈值 | $T_q$ | $R_q$ 内的不同词个数 |
| $R_q$ | $q$ 中指定的时间范围 | $S_q$ | $R_q$ 中 $s$ 的数目 |
| $TFPDF_t$ | $t$ 的广泛性属性值 | $Var_t$ | $t$ 的时事性属性值 |

### 1.1 热词提取

热词提取广泛应用于热门话题分析。热门话题是指在一段时间内,受到人们广泛关注并讨论的话题。在新闻报道中,话题的热度取决于 2 个因素[7]:在一篇报道中热词出现的频率以及包含这些热词的新闻报道的数量。一般而言,话题不会在所有时间中保持它的热度,它有一个从产生、兴起、成熟和衰亡的过程[10]。国内外学者在 TDT 的基础上提出了很多热门话题发现的方法,其中基于热词的提取并聚类的方法成为一种主流的研究方法[11-14]。

热词是指在一段时间内频繁出现的词。它用来描述热门话题,并且会随着热门话题的生命周期而不断变化。针对热词的提取,Chen 算法综合考虑词频和词表述话题的能力。实验结果表明,与 TF-IDF[5] 和 TF-PDF[6] 相比,Chen 算法能够有效地过滤与话题无关的词,并赋予与话题相关的名词短语或实体名词更高的权重,满足算法的有效性。Chen 算法中一个词的权重被定义如下:

$$weight_t = TFPDF_t + Var_t \times \left(1 + \frac{FO_t - VO_t}{T}\right) \quad (1)$$

其中,$TFPDF_t$ 表示词 $t$ 的广泛性,$Var_t \times \left(1 + \frac{FO_t - VO_t}{T}\right)$ 表示词 $t$ 的时事性,$FO_t$ 和 $VO_t$ 分别表示词 $t$ 的 $TFPDF_t$ 值和 $Var_t$ 值在所有词中的排名,用来调节词对时事性的偏差。热词的广泛性与词在数据集中出现的频率有关,频率越高意味着词越具有广泛性,计算公式如下:

$$TFPDF_t = F_t \times \exp\left(\frac{n_t}{N}\right) \quad (2)$$

其中,$t$ 是一个词,$F_t$ 表示词 $t$ 在数据集中出现的频率,$n_t$ 表示包含词 $t$ 的文档个数,$N$ 表示在给定数据集中文档的总数。热词的时事性用来刻画词的使用频率随着时间的变化情况。一个词在不同时间段上的使用频率差异越大,那么这个词越具有时事性,计算公式如下:

$$Var_t = \sqrt{\frac{1}{S}\sum_s (lifesupport_{t,s} - \overline{lifesupport})^2} \quad (3)$$

其中,$t$ 是一个词,$s$ 是一个单位时间间隔,$S$ 是在给定的时间范围 $R$ 内,单位时间间隔的个数,$lifesupport_{t,s}$ 是词 $t$ 在单位时间间隔 $s$ 上的生命值,$\overline{lifesupport}$ 是词 $t$ 所有单位时间间隔的生命值平均值。词 $t$ 在单位时间间隔 $s$ 上的生命值与 $s$ 和 $t$ 的卡方值有关,表示词 $t$ 和单位时间间隔 $s$ 的相关程度。

Chen 算法在提取热词时分为 3 个步骤:

**步骤 1** 计算所有词的词频,并获得跟踪列表;

**步骤 2** 对于跟踪列表中的每个词,计算词权重;

**步骤 3** 根据权重将词排序,选取权重最大的 $k$ 个词作为热词。

Chen 算法能够有效地提取与话题相关的词,但时间复杂度较高,在实际应用中,由于数据集的庞大以及需要多次查询的缘故,该算法的效率低下,因此难以直接用来进行文本区间热词的在线查询处理。

### 1.2 具有词频约束的区间查询

给定一个长度为 $n$ 的数组 $A$,考虑以下问题:对于任意的下标 $i,j(1 \leqslant i \leqslant j \leqslant n)$,给定一个阈值 $\alpha$,查询并返回子数组 $A[i:j]$ 中出现次数超过 $\alpha \times (j - i + 1)$ 的元素。针对具有词频约束的区间查询问题,研究人员展开了大量的研究工作[15-17]。文献[18]研究在子数组 $A[i:j]$ 中,查询任意一个元素出现的次数是否为 $k$ 的问题,并证明了当 $k$ 是大于 1 的固定值时,查询的时间复杂度是 $O\left(\frac{\lg n}{\lg \lg n}\right)$;当 $k$ 需要在查询中指定时,时间复杂度的下限是 $\Omega\left(\frac{\lg n}{\lg \lg n}\right)$。文献[19]研究了在几何设置中类似的问题,指出需要 $O\left(\frac{n}{\alpha}\right)$ 的存储空间,时间复杂度是 $O\frac{(\lg \lg n)^2}{\alpha}$。文献[20]给出在子数组 $A[i:j]$ 中满足词频大于 $\alpha$ 的词的个数的上限,并通过构建逻辑二叉树进行预处理,使得查询的时间代价是 $O\left(\frac{1}{\alpha}\right)$,空间代价是 $O\left(n \lg\left(\frac{1}{\alpha} + 1\right)\right)$。

## 2 文本区间热词的查询处理方法

### 2.1 问题描述

文本区间热词的在线查询问题描述如下:

对于数据集 $D$ 上的任意查询 $q, q = \{R_q, k\}$,返回区间 $R_q$ 上的 $k$ 个热词,其中 $R_q = [m, n]$,$m$ 和 $n$

表示查询时间范围的起止时间;$k$ 表示指定提取的热词个数。例如查询 $q=\{20160101,20160331,1000\}$,返回 2016 年 1 月 1 日—2016 年 3 月 31 日之间按热度递减排序的前 1 000 个词。原始数据集中的每篇文档都有一个时间属性,以原始数据集中出现的最小的时间间隔作为单位时间间隔,统计每个单位时间间隔内不同词出现的次数,并构建数据集 $D$。它的形式化描述为 $D=\{s,t,c\}$,其中,$s$ 表示时间间隔,$t$ 表示词,$c$ 表示相应时间间隔 $s$ 中词 $t$ 出现的次数。数据集 $D$ 上的数据结构如表 2 所示。

**表 2　数据集 $D$ 的数据结构**

| 时间间隔 | 词 | 计数 |
|---|---|---|
| 2016-01-01 | a | 5 |
| | b | 6 |
| | c | 7 |
| 2016-01-02 | b | 3 |
| | c | 5 |

若文本数据中时间属性值的下限和上限分别是 $l$ 和 $u$,则 $l\leqslant m\leqslant n\leqslant u$。

## 2.2　EHWE 算法

### 2.2.1　数据结构

通过对 Chen 算法进行分析发现,对于一个给定的查询 $q$,在获得跟踪列表时,需要计算 $q$ 中所有词在 $R_q$ 中的词频;在计算一个词在 $R_q$ 中的词频和 *TFPDF* 值时,需要遍历这个词在查询时间范围内的每个单位时间间隔的计数。为减少获得跟踪列表时需要计算的词数目以及优化词的计数结构,本节分别利用数据划分和范围查询的思想,对原始数据进行预处理并建立数据结构,使得获得跟踪列表和对查询 $q$ 中词计数的时间复杂度分别为 $O(1)$。

1)数据划分

本文利用数据划分的思想,将查询 $q$ 与一个四元组 $U$ 相关联。在获得跟踪列表时,只需要判断 $U$ 对应的候选词列表 $C$ 中的词是否满足在 $R_q$ 中的词频大于 $\alpha$ 的条件。因为 $C$ 的长度要远小于 $R_q$ 中不同词的个数 $T_q$,所以达到了优化目标。

在数据集中,一个时间间隔内可以有任意整数个不同词,每个词可以出现任意整数个数。首先为全部单位时间间隔建立索引,索引范围为[1:$S$];随后在每个单位时间间隔内,为每个不同词依次构建索引,最大索引下标是 $S\times T$,故索引范围为[1:$ST$],所以建立索引结构后,数据集 $D$ 中包括:单位时间间隔 $s_i$ 及其索引 $i$,$s_i$ 中出现的词 $t_{ij}$,词 $t_{ij}$ 的索引 $j$,词出现的次数 $n_{ij}$。假设词 $t_{ij}$ 在 $s_i$ 上不出现时,它的计数为 0。

表 3 是对数据集 $D$ 建立的索引结构的举例,和表 2 对应。

**表 3　对数据集 $D$ 建立的索引结构**

| 时间间隔 | 时间间隔索引 | 词索引 | 词 | 计数 |
|---|---|---|---|---|
| 2016-01-01 | 1 | 1 | a | 5 |
| | | 2 | b | 6 |
| | | 3 | c | 7 |
| 2016-01-02 | 2 | 4 | b | 3 |
| | | 5 | c | 5 |

令 $R_q=[m:n]$,在计算查询 $q$ 中满足词频大于 $\alpha$ 的词时,需要根据 $m$ 和 $n$ 找到词索引的范围[$j_m$,$j_n$],并计算该范围内不同词的词频。算法 1 是对数据集划分算法的形式化描述:

**算法 1**　数据集的划分算法
**输入**　数据集 $D$
**输出**　候选词列表 $C$
1.　构建概念上的二叉树;
2.　FOR 树中的每层 l:
3.　　构建每层的全部四元组 U;
4.　FOR 全部层的 U:
5.　　计算候选词列表;

基于建立的索引结构,算法 1 对数据集构建概念上的完全二叉树,其中根节点是整数数组[1:$S$],每个子节点都是父节点的子数组,且叶子节点的长度为 1,对一个节点[$a$:$b$],它的左子节点是[$a$:$b$]的左半部分,即 $\left[a:\left\lfloor\frac{a+b}{2}\right\rfloor\right]$,右子节点是[$a$:$b$]的右半部份,即 $\left[\left\lfloor\frac{a+b}{2}\right\rfloor:b\right]$。为方便说明,假设 $S$ 是 2 的幂,则这棵树一共有 $\lg S+1$ 层,每层 $l$ 有 $2^l$ 个节点,每个节点的长度是 $\frac{S}{2^l}$。需要注意的是,这棵二叉树是用来概念上划分与查询范围相关的四元组,不需要实际存储。

对于这棵二叉树的每一层 $l$,令 $2\leqslant l\leqslant\lg S$,每隔 2 个子节点将连续 4 个子节点划分为一个四元组 $U$,则第 $l$ 层有 $2^{l-1}-1$ 个四元组。设 $U_p$ 是第 $l$ 层划分的第 $p$ 个四元组,范围是[$a$:$b$],长度是 $\frac{4S}{2^l}$,则 $1\leqslant p\leqslant 2^{l-1}-1$,且有:

$$a=\frac{2(p-1)S}{2^l}+1$$
$$b=\frac{2(p-1)S}{2^l}+\frac{4S}{2^l}\qquad(4)$$

例如,当 $R_q=[1:16]$、$l=3$ 时,一共有 8 个节点,每个节点的长度为 2;可以划分为 3 个四元组,每个四元组的长度是 8,它的划分规则如图 1 所示。

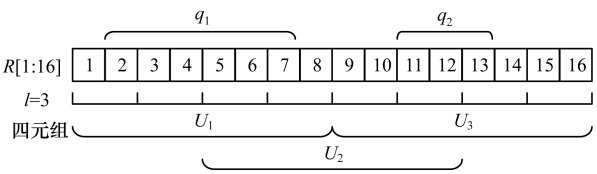



**图 1　$S=16$、$l=3$ 时数据的划分规则**

通过对二叉树每一层划分四元组,可以发现对于在整个时间范围上的任意查询 $q$,能够找到一个特定的层 $l$ 和四元组 $U_p$,使得查询 $q$ 至少包含这个四元组中的一个节点,至多包含 2 个非兄弟节点,即查询 $q$ 是和 $U_p$ 相关的。例如图 1 中的查询 $q_1$ 和 $q_2$,$q_1$ 对应于 $l=3$,$p=1$;$q_2$ 对应于 $l=3$,$p=3$。

假设四元组 $U_p$ 对应的时间间隔中一共有 $\sigma$ 个不同的词($\sigma<T$),通过遍历这些词,可以计算每个词的词频。其中 $U_p$ 对应的 4 个子节点中包含的词数目依次为 $c_1$、$c_2$、$c_3$、$c_4$。令 $c=\min(c_1,c_2,c_3,c_4)$,若词 $t$ 在 $U_p$ 中的出现次数大于 $\alpha\times c$,那么将 $t$ 加入到 $U_p$ 对应的候选词列表 $C_p$ 中,通过遍历这个四元组内的所有词,可以获得这个四元组对应的候选词列表。算法 2 需遍历每层的所有词,并且存储每层上所有四元组所对应的候选词列表,因此,算法 2 的时间代价是 $O(ST\lg S)$,空间代价是 $O(ST)$。

综上,对于任意的查询 $q$,在获取满足条件的词列表时,只需要遍历这个查询 $q$ 对应的四元组 $U_p$ 中的候选词列表 $C_p$,如果词满足在 $R_q$ 中的词频大于 $\alpha$,则将这个词加入跟踪列表中。需要注意的是,若 $c_1=c_2=c_3=c_4$,即每个单位时间间隔内出现的词总数相等,那么四元组 $U_p$ 对应的候选词列表 $C_p$ 的长度上限为 $\frac{4}{\alpha}-1$;如果每个单位时间间隔内出现的词总数是随机的,那么在极端情况下,不妨设 $c_1=1$,$c_2=c_3=c_4=T$,则 $C_p$ 的长度为 $T_q$,即没有减少候选列表的词总数。为了使在极端情况下依然达到优化目标,需要将词总数很少的连续时间间隔合并后,重新构建完全二叉树并划分四元组,以保证重新划分的时间间隔内词总数大致相等,或者词总数在整个时间范围上的方差小于一个经验值。

2)词计数结构

令 $R_q=[m:n]$,$U_p$ 为与其相关的四元组,$C_p$ 是 $U_p$ 对应的候选词列表。为获得跟踪列表,需要遍历 $C_p$ 中每个词,并计算它们在 $R_q$ 中的词频,如果满足条件,则将其加入到跟踪列表中。在计算一个词的词频时,Chen 算法需要遍历词在 $R_q$ 内每个单位时间间隔内的计数,这个过程的时间复杂度是 $O(ST)$。显然,算法 1 虽然减少了获得跟踪列表时需要计算的词数目,但整体的时间复杂度并没有降低,仍为 $O(ST)$。为优化计算 $C_p$ 中每个词在 $R_q$ 中的词频的时间复杂度,算法 2 优化了词计数的数据结构,使得计算一个词 $t$ 在 $R_q$ 中的词频的时间复杂度为 $O(1)$。这一过程的形式化描述如下:

**算法 2** 查询 $q$ 中词计数优化算法

**输入** 数据集 $D$

**输出** 每个单位时间间隔中词总数的数组 $total$;每个词在每个单位时间间隔中的数量的数组 $t\_c$

```
1.  FOR 每个不同的词 t:
2.      t_c[t][1] = t 在第一个单位时间间隔中的数量;
3.      FOR s FROM 2 TO S:
4.          t_c[t][s] = t_c[t][s-1] + s 中 t 的数量;
5.  total[1] = 第一个单位时间间隔中所有词的数量
6.  FOR s FROM 2 TO S:
7.      total[s] = total[s-1] + s 中所有词的数量
```

由算法 2 可知,当 $R_q=[m:n]$ 时,词 $t$ 的词频 $freq_{t,m,n}=count_{t,m,n}/Total_{m,n}$,计算 $freq_{t,m,n}$ 的时间为 $O(1)$,其中,$count_{t,m,n}=t\_c[t][n]-t\_c[t][m-1]$、$Total_{m,n}=total[n]-total[m-1]$。

因为 $C_p$ 的长度上限为 $O\left(\frac{1}{\alpha}\right)$,且计算一个词在 $R_q$ 中的词频的时间复杂度为 $O(1)$,所以获得跟踪列表的时间复杂度为 $O\left(\frac{1}{\alpha}\right)$,且空间复杂度保持不变,为 $O(ST)$。由于 $\alpha$ 是一个 0~1 范围内的固定值,因此获得跟踪列表的时间复杂度是 $O(1)$。

同理,在计算一个词的 *TFPDF* 值时,可以用算法 2 中的数据结构来优化词频和文档频率的计算过程,使得计算的时间复杂度为 $O(1)$。

### 2.2.2 EHWE 算法描述

基于以上的数据结构,对于任意一个查询 $R_q=[m:n]$,EHWE 算法的形式化描述如下:

**算法 3** EHWE 算法

**输入** $C$,$total$,$t\_c$,查询 $q$

**输出** 热词和权重

```
1.  根据查询 q 指定的参数 m、n
2.      获得候选词列表 C_p
3.      计算 m、n 范围内词总数Total_{m,n}
4.  FOR C_p 中的每个词 t
5.      计算词 t 在中出现的次数Count_{t,m,n}
6.      IF Count_{t,m,n} > α × Total_{m,n} THEN
7.          将词 t 加入到跟踪列表
8.  FOR 跟踪列表的每个词 t
9.      计算 t 的权重 w_t(根据式(1));
10. 根据权重排序选择前 k 个词
```

对一个查询 $R_q=[m:n]$,它的长度是 $m-n+1$,如果对应的层级是 $l$,四元组是 $p$,那么有:

$$\lg\frac{S_q}{m-n+1}<l<\lg\frac{S_q}{m-n+1}+1 \tag{5}$$

$$\frac{n-4}{S_q}\times 2^{l-1}\leqslant q\leqslant\frac{m-1}{S_q}\times 2^{l-1} \tag{6}$$

其中,$l$ 和 $q$ 是整数。如果令 $z=\left\lceil\lg\frac{S_q}{m-n+1}\right\rceil$,若 $(m-1)2^z \bmod S_q=0$ 或 $(n-1)2^z \bmod S_q=0$,且有 $\left\lceil\frac{(m-1)2^z}{S_q}\right\rceil\neq\left\lfloor\frac{(n-1)2^z}{S_q}\right\rfloor$,那么 $l=z$;否则 $l=z+1$。

## 2.3 复杂度分析

为获取跟踪列表,Chen 算法需要遍历数据集涉及的查询时间范围上的所有词,它的时间复杂度为

$O(ST)$。经过筛选后,跟踪列表中词数目的上限和 $\alpha$ 有关,为 $\left(\frac{1}{\alpha}-1\right)$。根据式(1),计算这些词的权重时需要分别计算它们的广泛性和时事性所对应的 *TFPDF* 值和 *Var* 值。根据式(2),计算词 $t$ 的 $TFPDF_t$ 时需要将词 $t$ 在 $S$ 个时间间隔上出现的次数和包含的文档个数累加,时间复杂度为 $O(S)$。根据式(3),计算词 $t$ 的 $Var_t$ 时,需要计算该词在 $S$ 个单位时间间隔的生命值,并求出方差。计算一个词在一个时间间隔上的生命值的时间复杂度是 $O(1)$,所以获得一个词在 $S$ 个时间间隔上的生命值所需要的时间是 $O(S)$。因为最多有 $\left(\frac{1}{\alpha}-1\right)$ 个词需要计算权重,所以步骤 2 的时间复杂度是 $O\left(\frac{1}{\alpha}\times S\right)$。获得每个词权重后,排序并且取最大的 $k$ 个词需要 $O\left(\frac{1}{\alpha}\lg\left(\frac{1}{\alpha}\right)\right)$ 的时间。综上,Chen 算法提取热词的时间复杂度为 $O(ST)+O\left(\frac{1}{\alpha}S\right)+O\left(\frac{1}{\alpha}\lg\left(\frac{1}{\alpha}\right)\right)$。对于给定的常数 $\alpha\in(0,1)$,$\frac{1}{\alpha}$ 也是一个常数,所以,计算给定时间范围 $R$ 内的词权重的时间复杂度为 $O(ST)$。

由算法 3 可知,在 EHWE 算法中,计算词权重并提取热词时主要分为 4 个步骤:

**步骤 1**　根据 $m$、$n$ 的值,返回与查询 $q$ 相关的候选词列表;

**步骤 2**　判断候选列表中的词是否满足频率大于 $\alpha$ 的条件,若满足,则将其加入到跟踪列表;

**步骤 3**　对于跟踪列表中的每个词,计算词的权重;

**步骤 4**　根据权重,将词降序排序,并返回前 $k$ 个作为热词。

与 Chen 算法相比,EHWE 算法利用数据划分,范围查询的计数对数据集 $D$ 进行预处理,并通过步骤 1 和步骤 2 来获得跟踪列表,这 2 个步骤的时间复杂度是 $O(1)$。在步骤 3 中,需要计算跟踪列表中每个词的 *TFPDF* 和 *Var* 值,它的时间复杂度分别是 $O(1)$ 和 $O\left(\frac{1}{\alpha}\times S\right)$。步骤 4 的时间复杂度不变,为 $O\left(\frac{1}{\alpha}\lg\left(\frac{1}{\alpha}\right)\right)$,$\alpha$ 是 0 ~ 1 的一个常数,所以 EHWE 算法的时间复杂度是 $O(S)$。

通过对 2 种算法的空间复杂度分析可以发现,Chen 算法需要存储整个数据集 $D$,空间复杂度为 $O(ST)$;EHWE 算法需要存储所有四元组对应的候选词列表 $C$ 以及 $C$ 中每个词的计数数组,它的空间复杂度为 $O(ST)$。所以,Chen 算法、EHWE 算法的空间复杂度保持不变。

## 3　实验与结果分析

### 3.1　数据源与数据分析

#### 3.1.1　数据源

本节对 Chen 算法和本文提出的 EHWE 算法进行实验比较。实验环境为:Intel Xeon(R) CPU E5-2650v3 @ 2.30 GHz × 40,128 GHz 内存和 256 GB磁盘,操作系统为 Ubuntu Kylin16.04,程序设计语言为 Java,使用 JDK1.8。实验数据来源于美国有线电视新闻网(CNN)、英国广播公司(BBC)和纽约时报(NYT),数据集统计信息如表 4 所示。

**表 4　数据统计信息**

| 数据来源 | 日期范围 | 文档数 | 不同词数 |
|---|---|---|---|
| CNN | 2016-01-01—2016-06-23 | 8 086 | 82 789 |
| BBC | 2011-01-01—2016-06-01 | 174 144 | 252 446 |
| NYT | 2010-01-01—2015-12-31 | 307 984 | 730 485 |

数据集 $D$ 中最小的时间单位是 d,所以本文实验中查询的单位时间间隔为 d。在实验之前,本文使用 Stanford CoreNLP 对原始的文本集进行预处理,包括去除停止词、分词、词形还原。

#### 3.1.2　数据分析

在 EHWE 算法中,为使任意的查询都能找到与之相关的四元组,需要对整个数据集涉及的时间间隔构建完全二叉树,并且保证每个叶子节点实际大小(包含的词总数)近似相等。如果不满足这个条件,则会使得四元组对应的候选词数目大于上限($4/\alpha-1$)。在极端情况下,若每个四元组中都有一个节点的大小为 1,那么候选词的数目等于查询 $q$ 中词的总数。为使候选词列表中词的数目小于 $q$ 中词的数目,需要将包含词数目非常少的连续单位时间间隔合并,并重新构建完全二叉树。

在本文实验中,数据集的单位时间间隔是 d,所以,统计了 3 个数据集中每天的新闻报道中词的总数,如图 2 所示为 BBC 数据集在 2014 年每天新闻报道中的出现的词总数,可以发现词总数在一条水平基线上波动,说明在 2014 年英国广播公司每天的新闻报道中用词数量是稳定的。

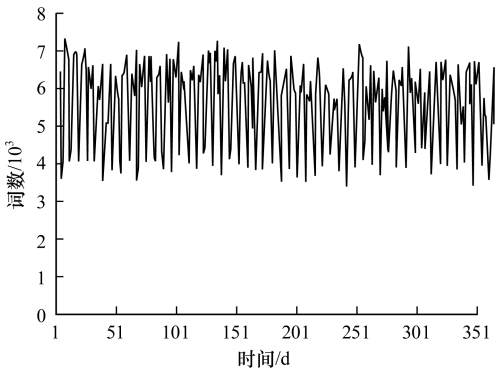



**图 2　BBC 2014 年每天的新闻报道中词总数**

表 5 中统计了 3 个文本集在整个时间戳上每天出现词数目的最大值、最小值以及对数据 0-1 标准化后的标准差,3 个数据集的标准差均小于 0.2,说明这 3 个新闻来源每天的新闻报道数量整体上是稳定的,不需要进行合并操作,可以直接在整个时间间隔构建完全二叉树。

**表 5 数据集分析**

| 数据源 | 最大值 | 最小值 | 标准差 |
|---|---|---|---|
| CNN | 7 083 | 100 | 0.192 0 |
| BBC | 11 906 | 7 | 0.124 3 |
| NYT | 22 935 | 59 | 0.148 9 |

### 3.2 Chen 算法和 EHWE 算法的运行时间比较

为比较 Chen 算法和 EHWE 算法运行时间,本节在上述 3 个数据集上分别进行实验,实验结果如图 3 所示。

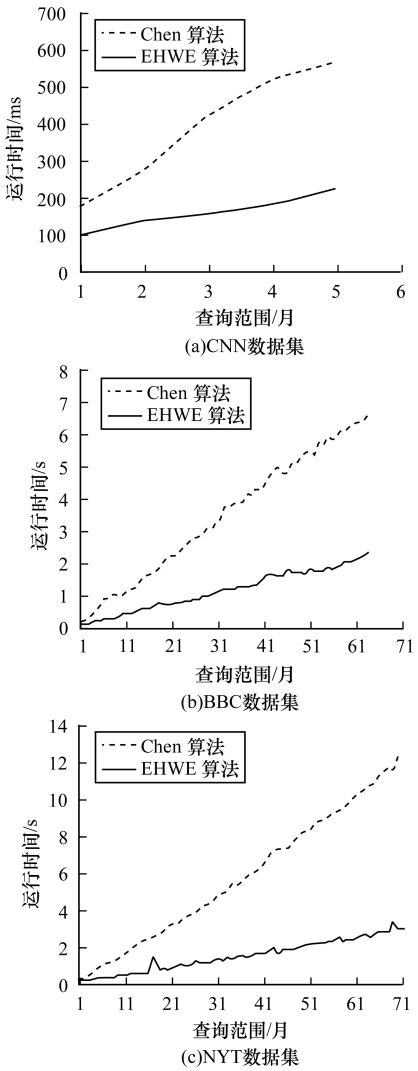



**图 3 运行时间随查询时间长度的变化**

在实验中,频率阈值 $\alpha$ 为 0.000 015。从图 3 可以看出,在 CNN、BBC 和 NYT 3 个数据集上,Chen 算法和 EHWE 算法的运行时间会随着查询时间的增长而增长,与 Chen 算法相比,EHWE 算法在 3 个数据集涉及的整个时间范围上的运行时间分别减少 59.7%、65.1% 和75.5%。与 Chen 算法相比,EHWE 算法减少的时间主要来源于获得跟踪列表并计算词的 *TFPDF* 值时运行时间。本文统计了 NYT 数据集上 2 种算法在获得跟踪列表并计算词的 *TFPDF* 值过程的时间消耗,如图 4 所示。

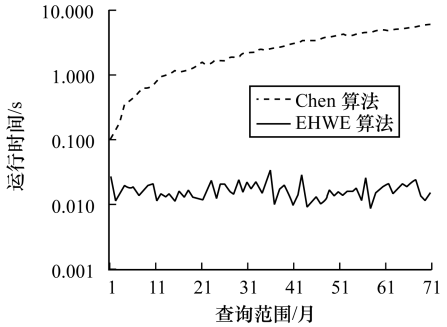



**图 4 跟踪列表及词 *TFPDF* 值的时间比较**

从图 4 可以看出,EHWE 算法的这部分运行时间接近为常数,而 Chen 算法在这部分的运行时间随着查询时间的增长而增长。这是因为 EHWE 算法这部分的时间复杂度为 $O(1)$,而 Chen 算法的这部分时间复杂度是 $O(ST)$,并且 $T$ 和 $S$ 相关。

如图 5 所示,通过统计纽约时报在 2014 年随着时间按照月份依次递增时,在数据集中出现不同词的个数,可以发现 $T$ 和 $S$ 是线性相关的,所以在获得跟踪列表并计算词的 *TFPDF* 值时,Chen 算法的时间消耗会呈现如图 4 所示的指数增长。

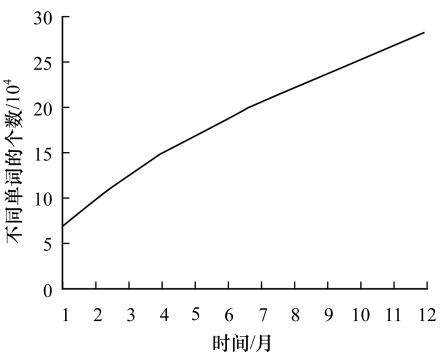
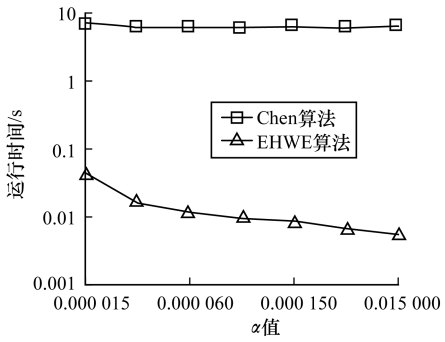


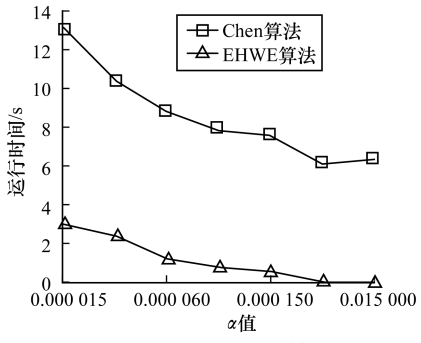

**图 5 纽约时报 2014 年前 *n* 月份中出现的词总数**

### 3.3 参数 $\alpha$ 的分析

$\alpha$ 是用户事先设定的频率阈值,表示一个词能成为热词的最低频率。$\alpha$ 值越大,表示满足条件的词个数理论上就会越少。本文实验比较 $\alpha$ 值不同时对 Chen 算法和 EHWE 算法的影响,如图 6 所示。




(a)计算词*TFPDF*值的时间




(b)提取热词的时间

**图 6　2 种算法运行时间随 $\alpha$ 变化的影响**

图 6(a)和图 6(b)分别表示在给定不同的$\alpha$ 值时,Chen 算法和 EHWE 算法关于获得跟踪列表并计算词 *TFPDF* 值(以下称 A 过程)以及整个提取热词(以下称 B 过程)过程中时间的消耗。实验的数据集是 NYT,查询的时间范围是 2010 年 1 月—2015 年 12 月。从图 6(a)可以发现,当 $\alpha$ 值增大时,2 种算法在 A 过程的时间消耗并没有显著变化,说明 A 过程的运行时间消耗与 $\alpha$ 的大小无关,这是因为它们的时间复杂度分别是 $O(ST)$ 和 $O(1)$,并且 $S$ 和 $T$ 与查询的时间范围有关。由图 6(b)可以发现,随着 $\alpha$ 的增加,2 种算法在 B 过程的运行时间消耗会降低,并且趋于稳定。这是因为随着 $\alpha$ 的增加,跟踪列表中词数目会减小,所以计算词权重的时间消耗会减少。在极端情况下,当跟踪列表中的词数目为 0 时,2 种算法在 B 过程的时间消耗会近似等于 A 过程的时间消耗,并且 Chen 算法的时间消耗大于 EHWE 算法。综上所述,当 $\alpha$ 值不断增大时,EHWE 算法在数据集上的运行时间仍小于 Chen 算法的运行时间。

本文主要贡献包括:

1)提出文本区间热词的在线查询处理问题,与面向挖掘的热词提取问题相比,更加关注在线查询的 2 个特性:有效性和时效性;

2)设计了 EHWE 算法,该算法能够在保证计算结果准确率的前提下,降低了提取热词的时间复杂度;

3)理论分析和实际数据集上的实验结果都表明 EHWE 算法优于 Chen 算法。

## 4　结束语

本文对文本区间热词在线查询问题进行研究。在 Chen 算法基础上,通过数据划分和范围查询,对原始数据进行预处理,并提出 EHWE 算法,在保证准确率和空间复杂度不变的条件下,降低了提取热词的时间复杂度。实验结果表明,EHWE 算法在 CNN、BBC、NYT3 个文本集上的时间代价要小于 Chen 算法。本文 EHWE 算法中尚未考虑对计算词 *Var* 值的优化,这将是下一步的主要工作。另外,基于热词的文本聚类算法以及对聚类的优化也需要进一步研究。

### 参考文献

[1] FISCUS J G,DODDINGTON G R. Topic Detection and Tracking Evaluation Overview[M]. New York: USA: Kluwer Academic Publishers,2002.

[2] 洪　宇,张　宇,刘　挺,等. 话题检测与跟踪的评测及研究综述[J]. 中文信息学报,2007,21(6):71-87.

[3] ALLAN J, PAPKA R, LAVRENKO V. On-line New Event Detection and Tracking[C]//Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York,USA:ACM Press,1998:37-45.

[4] 周亚东,孙钦东,管晓宏,等. 流量内容词语相关度的网络热点话题提取[J]. 西安交通大学学报,2007,41(10):1142-1145.

[5] HISAMITSU T,NIWA Y. A Measure of Term Representativeness Based on the Number of Co-occurring Salient Words[C]//Proceedings of the 19th International Conference on Computational Linguistics-Volume 1. Stroudsburg, USA:Association for Computational Linguistics,2002:1-7.

[6] SALTON G,YANG C S. On the Specification of Term Values in Automatic Indexing[J]. Journal of Documentation,1973,29(4):351-372.

[7] BUN K K,ISHIZUKA M. Topic Extraction from News Archive Using TF * PDF Algorithm[C]//Proceedings of International Conference on Web Information Systems Engineering. Washington D. C. , USA: IEEE Computer Society,2002:73-82.

[8] CHEN K Y,LUESUKPRASERT L,CHOU S T. Hot Topic Extraction Based on Timeline Analysis and Multi-dimensional Sentence Modeling[J]. IEEE Transactions on Knowledge & Data Engineering,2007,19(8):1016-1025.

[9] 聂文汇,曾　承,贾大文. 基于热度矩阵的微博热点话题发现[J]. 计算机工程,2017,43(2):57-62.

[10] CHEN C C, CHEN Y T, SUN Y, et al. Life Cycle Modeling of News Events Using Aging Theory[C]// Proceedings of European Conference on Machine Learning. Berlin,German:Springer,2003:47-59.

(下转第 30 页)

## 参考文献


[1] WIDMER G, KUBAT M. Learning in the Presence of Concept Drift and Hidden Contexts[J]. Machine Learning, 1996, 23(1): 69-101.

[2] WANG Haixun, FAN Wei, YU P S, et al. Mining Concept-drifting Data Streams Using Ensemble Classifiers[C]// Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2003: 789-802.

[3] GAMA J, MEDAS P, CASTILLO G, et al. Learning with Drift Detection[M]//BAZZAN A L C, LABIDI S. Advances in Artificial Intelligence. Berlin, Germany: Springer-Verlag, 2004: 286-295.

[4] GAMA J, CASTILLO G. Learning with Local Drift Detection[M]//LI Xue, ZAÏANE O R, LI Zhanhuai. Advanced Data Mining and Applications. Berlin, Germany: Springer-Verlag, 2006: 42-55.

[5] BAENA-GARC M, CAMPO-ÁVILA J D, FIDALGO R, et al. Early Drift Detection Method[C]//Proceedings of the 4th International Workshop on Knowledge Discovery from Data Streams. Pittsburgh, USA: [s. n.], 2006: 1-10.

[6] LI Peipei, WU Xindong, HU Xuegang, et al. A Random Decision Tree Ensemble for Mining Concept Drifts from Noisy Data Streams[J]. Applied Artificial Intelligence, 2010, 24(7): 680-710.

[7] 朱 群, 张玉红, 胡学钢, 等. 一种基于双层窗口的概念漂移数据流分类算法[J]. 自动化学报, 2011, 37(9): 1077-1084.

[8] ZHU Qun, HU Xuegang, ZHANG Yuhong, et al. A Double-window-based Classification Algorithm for Concept Drifting Data Streams[C]//Proceedings of IEEE International Conference on Granular Computing. Washington D. C., USA: IEEE Press, 2010: 639-644.

[9] 桂 林, 张玉红, 胡学钢. 一种基于混合集成方法的数据流概念漂移检测方法[J]. 计算机科学, 2012, 39(1): 152-155.

[10] LI Peipei, WU Xindong, HU Xuegang. Mining Recurring Concept Drifts with Limited Labeled Streaming Data[J]. ACM Transactions on Intelligent Systems and Technology, 2012, 3(2): 241-252.

[11] 孙 雪, 李昆仑, 韩 蕾, 等. 基于特征项分布的信息熵及特征动态加权概念漂移检测模型[J]. 电子学报, 2015, 43(7): 1356-1361.

[12] 郭躬德, 李 南, 陈黎飞. 一种基于混合模型的数据流概念漂移检测算法[J]. 计算机研究与发展, 2014, 51(4): 731-742.

[13] BLEI D M, NG A Y, JORDAN M I. Latent Dirichlet Allocation[J]. Journal of Machine Learning Research, 2003, 3: 993-1022.

[14] HOFFMAN M D, BLEI D M, BACH F R. Online Learning for Latent Dirichlet Allocation[M]//LAFFERTY J D, WILLIAMS C K I, SHAWE-TAYLOR J. Advances in Neural Information Processing Systems. [S. l.]: Neural Information Processing Systems Foundation, Inc., 2010: 856-864.

[15] LI Peipei, WU Xindong, HU Xuegang, et al. Learning Concept-drifting Data Streams with Random Ensemble Decision Trees[J]. Neurocomputing, 2015, 166(C): 68-83.

[16] FRIASBLANCO I, CAMPOAVILA J D, RAMOSJIMENEZ G, et al. Online and Non-parametric Drift Detection Methods Based on Hoeffding's Bounds[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(3): 810-823.

[17] HABABOU M, CHENG A Y, FALK R. Variable Selection in the Credit Card Industry[C]//Proceedings of NESUG'06. Philadelphia, USA: SAS Institute Inc., 2006: 1-5.


编辑 金胡考

(上接第 23 页)


[11] 王 林, 藏冠中. 基于复杂网络社区结构的论坛热点主题发现[J]. 计算机工程, 2008, 34(11): 214-216.

[12] 高 妮, 周明全, 耿国华, 等. 基于文本挖掘的话题发现技术[J]. 计算机工程, 2009, 35(19): 36-38.

[13] KUMARAN G, ALLAN J. Text Classification and Named Entities for New Event Detection[C]//Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM Press, 2004: 297-304.

[14] 黄承慧, 印 鉴, 侯 昉. 一种结合词项语义信息和 TF-IDF 方法的文本相似度量方法[J]. 计算机学报, 2011, 34(5): 856-864.

[15] YAO A C. Space-time Tradeoff for Answering Range Queries[C]//Proceedings of the 14th Annual ACM Symposium on Theory of Computing. New York, USA: ACM Press, 1982: 128-136.

[16] DUROCHER S, SHAR R, SKALA M, et al. Linear-space Data Structures for Range Frequency Queries on Arrays and Trees[J]. Algorithmica, 2016, 74(1): 344-366.

[17] CHAN T M, DUROCHER S, SKALA M, et al. Linear-space Data Structures for Range Minority Query in Arrays[J]. Algorithmica, 2015, 72(4): 901-913.

[18] GREVE M, JORGENSEN A G, LARSEN K D, et al. Cell Probe Lower Bounds and Approximations for Range Mode[M]. Berlin, Germany: Springer, 2010.

[19] KARPINSKI M, NEKRICH Y. Searching for Frequent Colors in Rectangles[C]//Proceedings of the 20th Annual Canadian Conference on Computational Geometry. Vancouver, Canada: [s. n.], 2008: 11-19.

[20] DUROCHER S, HE M, MUNRO J I, et al. Range Majority in Constant Time and Linear Space[C]// Proceedings of International Colloquim Conference on Automata, Languages and Programming. Berlin, Germany: Springer, 2011: 244-255.


编辑 索书志