

基于文本相似度的微博网络水军发现算法

杨长春, 徐小松, 叶施仁, 周 猛
(常州大学 信息科学与工程学院, 江苏 常州 213164)

摘 要: 微博中水军发表的评论内容具有重复或者相似性, 提出了基于文本相似度的微博网络水军发现算法. 评论内容可以用特征码来表示. 特征码再通过高效的 B-Tree 来索引, 使整个系统具有极高的处理效率. 根据水军发帖的重复性或者相似性很高的特点, 通过对多个相同或相似的评论内容进行统计分析找出出现次数频繁的用户, 初步定义为水军. 再对这些用户的评论内容进行分析, 发现他们的评论内容基本上都是具有重复性. 试验表明, 该方法能够准确、有效地找出水军账户.

关键词: 特征码; 索引; 文本相似度; 最长子序列

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2014)03-0082-04

A Method to Find Water Armies in Weibo Based on Text Similarity

YANG Chang-chun, XU Xiao-song, YE Shi-ren, ZHOU Meng

(School of Information Science & Engineering, Changzhou University, Changzhou 213164, China)

Abstract: The comments issued by the Water Army were repeatability and similarity. A method to find Water armies in weibo based on Text similarity was proposed. Comments are represented by signatures. By indexing the signatures on efficient B-Tree, the whole system had extreme processing efficiency. According to the repeatability and similarity characteristics of the comments issued by the Water Armies, the users which defined as Water Armies had a high number of occurrences when analyzed on the same or similar comments. And then analyzed the comments published by those users, found that those comments were almost similar. Experiments showed that this method could find the Water Armies accurately and effectively.

Key words: signature; index; text similarity; longest subsequence

1 引言

微博具有公开性、及时性等特征, 但其存在一些负面影响. 因此, 怎样辨识这些网络水军是本文所要研究的内容.

当今, 网络水军的水贴越来越多, 对网络环境的真实性产生了严重的影响, 然而对这些行为目前还没有一个有效的方法进行辨识与制止. 文献[1]中提出了对发帖进行分析、处理, 生成 Hash 值, 通过各个发帖 Hash 值之间的比较, 确定该贴是否为水贴. 该方法主要采用矩阵形式, 对大规模数据量的处理则不太现实. 文献[2]中通过相同 IP 与 ID 的统计来

得出此 IP 地址是水军的概率. 此方法只能从表面确定是否为水军, 错误率是其不能解决的一大难题. 文献[3]中采用情感分析的方法, 通过采用机器学习方法对正文信息进行情感分类, 首先判断其为正面信息或为负面信息, 经过统计后, 在特定时间内, 当某人发表的正面或者负面信息比重超过阈值时则认为其为网络推手. 此方法得到的水军更为模糊, 情感分析本来就是一个很难捉摸的难题, 情感分析的有效性、正确性还有待改进, 因此该方法的效果并不理想. 文献[4-5]主要基于特征码技术找出相同网页, 特征码再通过 B-Tree 索引可以高效的找出相同的网页.

收稿日期: 2013-04-17; **修回日期:** 2013-06-03

基金项目: 国家自然科学基金项目(61272367); 江苏省科技厅项目(BZ2010021); 江苏省研究生培养创新工程项目(20120515)

本文在借鉴已有的成果的基础上,根据水军自身的特点主要从以下三个方面进行研究:

- (1)网络水军的基本特点及描述;
- (2)求相似评论内容的方法;
- (3)水军发现.

2 网络水军的基本特点及描述

网络水军^[6],是指被网络推手召集,以在网络上发帖、回帖等方式执行传播命令,并获得相应报酬的人员群体.网络和社会上那些引起轰动的事件大多数都是由于网络水军在网络上发帖、回帖造势所引起的.而本文主要从新浪微博中评论内容这一块寻找水军.经过大量的观察与研究发现,新浪微博中水军在评论内容这一块主要运作流程如下:水军用户们得到命令,每个水军都会得到所需发出的评论,这些评论可以随处发送,就当作宣传.当然,少数水军可能会给评论内容改一两个字,但是不会大幅度的修改,修改过多会改变原来的意思,达不到雇主的要求,而且专业的水军用户每天都会接到数百例这样的任务,没有时间去作大的修改.本文找相似的评论内容要求比较高,相似度一般达到95%时才会归类为一条评论.水军的传播信息模式如图1所示,存在形式如图2所示.

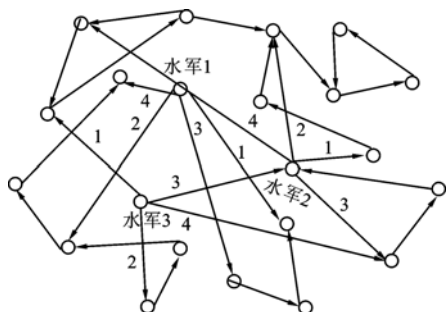


图1 水军传播信息模式

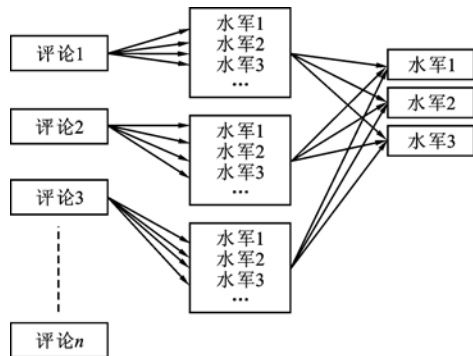


图2 水军存在形式

图1中数字表示水军们所发的评论内容,从以上两图中我们可以看到水军发表评论是随意地往四

处传播,但是他们有一个特点即他们所发的评论内容是具有重复性的,因此通过找出这些重复或者相似的评论内容,再对这些评论内容进行分析便可以很轻松地找出水军用户.

3 求相似评论内容的方法

对于大规模的评论内容而言,为了能够快速有效地得到所有的相似的评论内容,其方法步骤如图3所示.

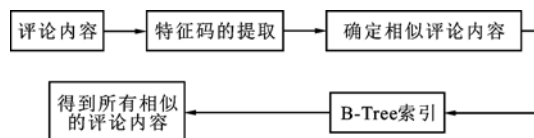


图3 步骤结构图

3.1 特征码的提取

水军发的帖子有很明显的特征即它们都是基本相同的,当然现今也有些水贴可能会添加或者是删除一些字、词以掩人耳目,而且水贴的字数也比较多.本文找出水贴主要采用类似网页去重的技术.网页去重都是基于这样一个基本思想:为每个文档计算出一组指纹,若两个文档拥有一定数量的相同指纹,则认为这两个文档的内容重叠性较高,也即二者是重复网页^[7].

网页去重技术其实也采用了检索系统^[8-9]这一思想,文献[4]中就引用了特征码这一概念,每一个网页可以找出一个特征可以跟其他不同的网页区分开,这种特征也即为网页的特征码.文献[4]中提取网页的特征码主要是取了长度为10的字符串.这样把所有的特征码检索起来便可以作为完整的检索系统.根据网页去重技术的启发,我们也可以给每条评论内容一个特征码,把所有评论内容的特征码检索起来也可以建立一个完整的检索系统.而在特征码的定义与提取中参数 L 非常重要,特征码太长会给存储、计算带来较大的开销,而如果特征码太短其区分能力便会降低.因为评论内容不超过140个字,因此本文取评论内容自身作为特征码,长度 L 为140,对于评论内容少于140个字的可以以特定的字符补位,按照多元文法(N -Gram)的定义这个特征码相当于一个140阶的文法(140-Gram),如果有6 763个汉字那么这个特征码的重复概率大概为 $\frac{1}{(6\ 763)^{140}}$,因此这个特征码在不同的评论内容中是不可能重复的.如果把长为 L 的特征码看成一个二进制数,也就是把一个评论内容映射到一些由二进

制数字构成的大小为 $(6\ 763)^{140}$ 空间里, 在这么大的空间里, 不同评论内容是不可能出现相同的特征码的. 通过这种映射就把一个复杂的计算评论内容相似度来判断评论内容是否相同的问题变为利用二进制数的比较来判别评论内容是否相同的问题.

3.2 确定评论内容相似

水军发的帖子相似度较高, 可能在原帖上添加、删除或者修改一两个字、词, 例如: “我们都知道这其实是一场阴谋”和“你们都知道这其实是一场阴谋”这两个特征码其实是一样的, 但是要按特征码完全相同才算是相同的评论内容的话, 这两个特征码就是不一样的, 因此我们提出采用最长子序列的比较方法.

两个序列 $X = \{x_1, x_2, \dots, x_n\}$, $Y = \{y_1, y_2, \dots, y_m\}$, 其中 X, Y 相当于两个特征码, x_n, y_m 相当于两个句子中的字, 最长子序列就是给定的两个序列共有的子序列中最长的一个. 最长子序列有一个阈值, 一般为 $L-1$ 或 $L-2$, 当其大于这个阈值时就认为这两个特征码是相同的. 具体阈值的大小可以根据评论内容的字数的多少而定, 为了能够精确地得到水军, 设立了这样的一个区间(count 为评论内容的字数):

(1) 当 $\text{count} < 5$ 时, 最长公共子序列为 L .

(2) 当 $5 \leq \text{count} < 15$ 时, 最长公共子序列为 $L-1$.

(3) 当 $15n \leq \text{count} < 15(n+1)$ 时, 最长公共子序列为 $L-(n+1)$, 其中 $15(n+1) < 140$, n 为正整数.

该算法经过优化后其空间复杂度为 $o(n)$, 时间复杂度为 $o(n \log n)$.

3.3 B-Tree 索引

每条评论内容都有各自的特征码, 怎样把这些特征码组织起来找出重复的评论内容是研究的重点. 运用索引的思想将所有的特征码都建立索引构成一个检索系统, 在建立索引时其实就已经确定相同的特征码. 对于大量的评论内容, 为了能够保证检索系统具有较高的检索效率, 特征码不超出内存的存储能力, 引用平衡多路查找树(B-Tree)^[10]来组织索引文件. B-Tree 最早是在一篇研究大型有序索引的文章中提出的, 它几乎是当今大型关系数据库中必须的, 因为大型数据库中的单个索引总是很容易就超过主存而 B-Tree 的辅存优化特性正好符合这样的严苛情况. 虽然一般的平衡搜索树也能维护索引, 但是在磁盘访问效能方面, B-Tree 明显更为优越. 一棵 m 阶的 B-Tree 满足:

(1) 每个结点的子结点个数不大于 m ;

(2) 根结点若不是叶子结点, 它至少有两个子结点;

(3) 除根和叶子结点外, 每个结点的子结点个数不小于 $\lceil m/2 \rceil$;

(4) 所有叶子结点都出现在同一层, 而且不带信息;

(5) 所有非终结结点都包含以下信息:

$(p_0, k_1, p_1, k_2, p_2, \dots, k_{i-1}, p_{i-1}, \dots, k_j, p_j)$

$k_i (1 \leq i \leq j)$ 是关键字, 所有关键字的值是唯一的, $p_i (1 \leq i \leq j)$ 是指向该结点的子结点的指针. 结点中的关键字是按一定规则排好序的, 若按升序排列, 有 $k_1 < k_2 < \dots < k_j$, 那么 p_0 指向一棵关键字均小于 k_1 的子树的根结点, $p_i (0 < i < j)$ 指向一棵关键字都在 k_i 和 k_{i+1} 之间的子树的根结点; p_j 指向一棵关键字均大于 k_j 的子树的根结点.

B-Tree^[11-12]的生成过程其实是特征码的插入过程, 在插入之前先进行查找, 如果要插入的特征码已与树的某个结点 Key1 相同, 说明这个 Key 特征码所代表的评论内容 Comment 与 Key1 特征码所代表的评论内容 Comment1 相同. 如果树中没有与 Key 相同的特征码则将 Key 插入到树中. 如果被插入结点的关键字的个数小于 $m-1$, 即该结点的关键字的个数不满, 就直接把给定值作为关键字插入到结点的应有位置. 如果被插入结点的关键字的个数等于 $m-1$, 即该结点的关键字的个数已满, 插入时需进行“分裂”.

“分裂”的意思就是将被插入结点分裂成两个结点, 而将结点中位于中间的关键字推到被插入结点的父结点中进行插入. 在一棵 m 阶 B-Tree 中插入新的关键字时, 如果叶子结点在第 i 层, 那么, 插入的结点就在第 $i-1$ 层. 如果被插入的结点原来关键字的个数少于 $m-1$, 那么就将新的关键字直接插入到这个结点的适当位置上; 如果被插入的结点原来的关键字个数等于 $m-1$, 那么加上新的关键字就会有 m 个. 假如这些关键字为 $k_1, k_2, \dots, k_n$ 那么, 将这个结点分裂成两个结点, 前面的结点由关键字 $k_1, k_2, \dots, k_{\lfloor n/2 \rfloor - 1}$ 组成, 后面的结点由关键字 $k_{\lfloor n/2 \rfloor + 1}, k_{\lfloor n/2 \rfloor + 2}, \dots, k_n$ 组成, 而把关键字 $k_{\lfloor n/2 \rfloor}$ 插到父结点中去. 父结点中指向被插结点的指针, 扩充为 $p, k_{\lfloor n/2 \rfloor}, q$ 的形式, 指针 p 指向分裂后的前面的结点, 而指针 q 指向分裂后的后面的结点. 父结点的插入仍然重复前面的操作. 假如父结点关键字的个数还是 $m-1$, 那么继续分裂并将中间的关键字插到祖父

结点.如果一路上关键字的个数都已满,就一路上分裂下去,最后把原来的根节点分裂成两个结点,而将中间的关键字向上推,形成新的根结点,此时,这棵 B-Tree 就长高了一层.根据以上分析,如果 m 取 100,那么如果有一百万个特征码需要插入到 B-Tree,每个插入过程需要的最大访问外存的次数不超过 3.如果要插入的特征码的个数为 n ,内存缓冲块个数为 c ,则插入每个特征码访问外存次数大概为: $\log_m n - \log_m c$,其中 $n \geq c$.因为特征码是有序排列的,因此可以通过折半查找法来定位要插入的位置,折半查找的效率为 $\theta(\log(n))$.因此使用 B-Tree 组织索引文件是有很高的效率.

4 水军发现

可以从 n 条评论内容中找出 m 组至少有 p 条评论相似的评论内容,当然 p 可以自己定义.如何从这 m 组相似内容中找出水军是解决问题的关键.由图 2 可知,一个组织下的水军们会发送上级命令发出的评论,因此不难推理出这些水军会在 $q(q \in m)$ 组相似内容下的账户里出现, q 取决于上级需要发多少条评论内容.水军一般会在相似的评论里频繁出现,因此要找出水军可以从所有所求的相似内容的用户中找出出现次数较多的用户,如果对查找水军要求严格点那么这个次数可以设定得相对高点,反之亦然.最后,将经过前面的要求后得到的用户的所有评论内容都取出来在索引库里查询,如果有 85% 以上的评论在索引库中都有评论和它相似,那么这个用户就认为是水军用户.实验证明此种水军账户的提取方法具有较高的准确率.

5 结束语

本文利用网页去重的原理,为每条评论内容产生一个特征码,引用最长子序列的方法为每个特征码产生一个二进制数,由此可以找出相似的评论内容并且使用 B-Tree 进行索引,通过设定 p 、 q 的值找出水军账户是具有可行性的,其具有较高的正确率和速率,符合实际的运用需求.不过对于字数较少的评论内容而言就比较容易容易出现误差,这也是今后需要研究的问题.

参考文献:

- [1] 胡舜良,凌宝红,胡东辉.基于文本认知 Hash 的网络水军发帖检测技术研究[J].安庆师范学院学报,2012(4):18-2.
- [2] 陈宇.“网络水军”泛滥与网络舆情检测的完善[J].新媒体,2011(15):69-70.
- [3] 李纲,甘停,寇广增.基于文本情感分类的网络推手识别[J].图书情报工作,2010(11):54-8.
- [4] 张刚,刘挺,郑实福,等.大规模网页快速去重算法[C]//中国中文信息学会二十周年学术会议.北京:清华大学,2001.
- [5] 魏丽霞,郑家恒.基于网页文本结构的网页去重算法[J].计算机应用,2007,27(11):2854-2856.
- [6] 赵敏,谭腾飞.网络水军的成因及其发展[J].新疆社科论坛,2012(3):64-66.
- [7] 周小平,黄家裕,刘连芳,等.基于网页征文主题和摘要的网页去重算法[J].广西科学院学报,2009,25(4):251-253.
- [8] David A Grossman, Ophir Frider. Information retrieval: algorithms and heuristics[J]. Boston: Kluwer Academic Publishers, 1998.
- [9] Ricardo Baeza Yates, Berthier Ribeiro Neto. Modern information retrieval[M]. New York, ACM Press, 1999.
- [10] Nakatsu N, Kambayashi Y, Yajima S. A longest common subsequence algorithm suitable for similar text strings[J]. Acta Informatica, 1982(18):171-179.
- [11] 李欣,舒风笛.最长公共子序列问题的改进快速算法[J].计算机应用研究,2000,17(2):28-30.
- [12] Comer D. The ubiquitous B-Tree[J]. Computing Surveys, 1979, 11(2):121-37.

作者简介:



杨长春 男,(1963—),教授,硕士生导师.研究方向为数据库系统、数据挖掘.

徐小松 男,(1989—),硕士研究生.研究方向为数据挖掘.
叶施仁 男,(1970—),硕士生导师.研究方向为数据挖掘.