

基于 Simhash 算法的海量文档反作弊技术研究

徐济惠

(宁波城市学院, 浙江 宁波 315100)

摘要:以互联网重复文档反作弊需求为背景,研究了基于 Simhash 的海量文档反作弊技术。以 Simhash 算法为文档判重的核心算法作基础对该算法获取文档特征的过程进行改进,将单词意义作为衡量单词权重的一个考量因素。针对 64 位文档 Simhash 签名,提供用户维度、全文维度和黑库维度的文档判重服务,并可基于全文和段落两种粒度进行文档相似性比较。通过测试数据和分析,该技术能保证运行稳定,每个实例可存储 1 亿文档,平均请求耗时稳定在 20 ms 左右,高峰期请求耗时会增长,但一般不会超过 100 ms。

关键词:重复文本检测;Simhash;反作弊;签名计算

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2014)09-0103-05

doi:10.3969/j.issn.1673-629X.2014.09.023

Research on Huge Amounts of Documents Anti-spamming Technique Based on Simhash Algorithm

XU Ji-hui

(Ningbo City College, Ningbo 315100, China)

Abstract: On the background of the anti-spamming needs of repeated documents in Internet, research the anti-spamming technique based on the Simhash on huge amounts of documents. On the basis of taking the Simhash algorithm as core algorithm in duplicate document detection, improve the procedure of achieving document features of this algorithm. It takes the meaning of words as a consideration factor in measuring the weight of words. Aiming at the Simhash signature of a 64-bit, provide the document service of user dimension, the full dimension and black dimension, and make a similarity comparison based on the full text and paragraphs. Through test data and analysis, this technique can guarantee the stable operation, 100 million documents can be memorized in each example. The average request response time is about 20 ms. The response time will increase during the peak hour, but, in general, will not go over 100 ms.

Key words: duplicate document detection; Simhash; anti-spamming; signature calculation

0 引言

在这个信息爆炸的时代,网络上的重复文档越来越多,据统计,互联网上的重复网页约占 30% ~ 45%^[1]。对网络上的文档进行相似度判断,并根据判定结果做相应的处理,例如不予收录、删除等,成为互联网技术发展的一个重要分支^[1]。

在互联网中,大量相似文档是很常见的现象,大量重复文档不仅会降低产品质量,而且对用户不友好,如何避免大量重复或相近文档出现是一个难题。传统的 hash 算法只负责将原始内容尽量均匀随机地映射为一个签名值,原理上相当于伪随机数产生算法^[2]。传统 hash 算法产生的两个签名,如果相等,说明原始内

容在一定概率下是相等的;如果不相等,除了说明原始内容不相等外,不再提供任何信息,即使原始内容只相差一个字节,所产生的签名也很可能差别极大。而 Simhash 对相似的内容产生的签名也相近,签名值除了提供原始内容是否相等的信息外,还能额外提供不相等内容的差异程度信息。

Simhash 算法首先被 Google 应用到网页去重系统中,通过 Simhash 算法给每一个网页计算 64 位签名,并且通过海明距离来断定抓取到的网页是否与库中已存在的重复。Google 的网页去重系统主要解决镜像网站、内容复制、嵌入广告、计数改变、少量修改等重复问题^[3]。

收稿日期:2013-10-17

修回日期:2014-01-19

网络出版时间:2014-05-21

基金项目:宁波市自然科学基金资助项目(2011A610100)

作者简介:徐济惠(1964-),男,宁波人,副教授,研究方向为软件开发方法。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140525.1242.007.html>

文中研究了基于 Simhash 的海量文档反作弊技术,经过测试,程序运行稳定,对大规模数据有很高的处理效率,能满足多个实例使用服务,能处理海量文档的需求。程序仅存储文档的 Simhash 签名和基本的文档信息,且每个实例的数据可以通过存储代理存储在 4 个重划学区系统 (Redistricting system, Redis) 实例上,未来还可以进行扩展。

1 相关技术

1.1 Simhash 算法

Simhash 算法由 Google 的 Charikar 提出,是将一篇文档转化为 n 位的签名,通过比较签名的相似度来计算原文档的相似度^[4]。签名越相近,则文档越相似。因此,整个过程不会涉及到原文档文本内容的两两比较,就无需存储这些海量的文档内容,因此该算法可以推广到数以百亿的文档比较范围。另外算法简单易行,容易理解,但要达到理想的效果还需结合具体的需求处理。Simhash 算法是当前主流的近似文本检测算法^[5],Simhash 算法流程图实现的一个具体流程如图 1 所示。

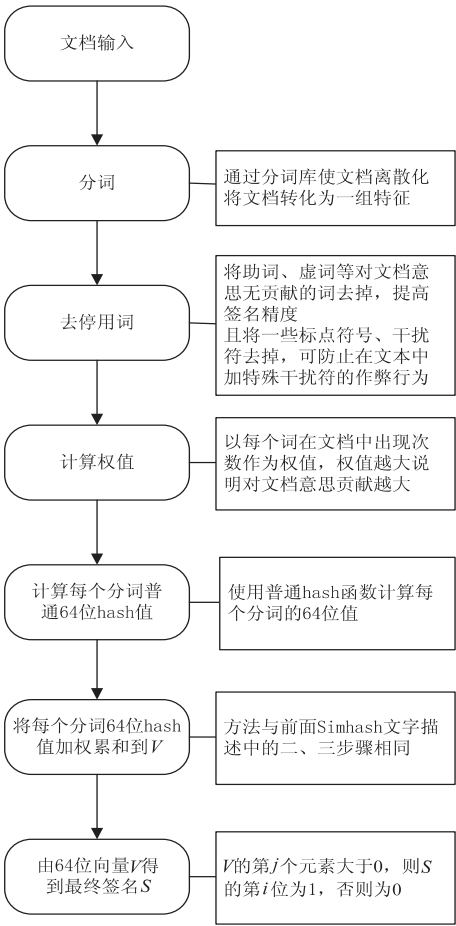


图 1 Simhash 算法具体流程图

通过以上步骤,给文档生成了 n 位签名值,将计算文档相似度的问题转化为比较两个文档签名值,通过

计算两个签名的海明距离即可实现^[6]。

文档最终以 64 位签名的形式存储,如何从两个 64 位签名判断文档是否相似,可以用海明距离来度量,下面为两篇文章的签名:

```
01101011010110101001101010110111000001
000111010111110010100101
01101011010110101001101010110111000001
000111010111110011100111
```

可见这两篇文章的海明距离为 2,海明距离越小表明两篇文章越相似。对于位数相同的两个数,海明距离为两数相异或后的结果含有二进制 1 的个数。

Simhash 算法发明人 Charikar 在论文中阐述,64 位 Simhash,海明距离在 $k=3$ 以内的文本都可以认为是近重复文本^[7],具体数值需要结合具体业务以及经验值来确定。

在实际应用中,要让签名尽量多地反映文档内容,在计算签名之前,先将文档做一些预处理是很有必要的。避免单一 Simhash 算法的使用,尽量与一些简单有效的方法综合使用,会达到更好的结果。

综上所述,Simhash 算法对待检测文档生成的签名值位数较少,通过计算签名值的海明距离来计算文档的相似度,使得该算法能够处理数以百亿的文档。另一方面,由于该算法生成签名的时候采取的是降维的思想,会丢失一定量的信息,因此在使用过程中,还需结合具体的需求,结合其他方法使用,方能达到较佳效果^[8]。

1.2 中文分词技术

Simhash 算法,以一组离散化的单词作为文档的重要特征。如何从文档中获得切分后的单词,需要用到中文分词技术。下面对中文分词技术做简要介绍。

中文分词 (Chinese Word Segmentation) 指的是将一个汉字序列,按照人脑理解的方式,切分成一个一个单独的词,最终输出中文单词、英文单词和字符串^[9]。中文分词技术,利用计算机技术,对中文序列进行自动识别,从而切分出合理的单词序列。目前,学术界对中文分词算法研究的比较多,根据其特点,可以将中文分词算法分为四大类:基于字符串匹配的分词算法、基于理解的分词算法、基于统计的分词算法和基于语义的分词算法^[10]。其中,基于字符串匹配和基于统计的分词算法在工程界应用比较成熟。

2 关键技术研究

2.1 高速检索技术设计

海明距离的计算虽然简单,但当数据达到一定规模,如 80 亿签名,如何从这 80 亿签名中找出与待比较签名海明距离小于等于 3 的所有签名,采用逐个比较

的方式是不现实的。Google 根据 f 位 Simhash 中要得到 1 到 k 位不同的签名,其中 k 较小的这个特点,提出了改进的计算方法,并提出了批量处理方法,使得效率得到很大提高^[11],下面将举例具体阐述。

将每个 64 位的签名分为 4 个部分,若两签名的海明距离小于 3,通过抽屉原理可知,则必定有一个部分相等。因此,可将 64 位的签名平分为 4 个部分,每部分 16 位,将 16 位的二进制作作为 key,将含有该 16 位 key 的签名作为 value 存储在 Redis 中。对于一个待比较的签名,均分为 4 个部分,每个部分作为 key 在 Redis 中拉取 value,再从被拉取出的 value 中计算海明距离,这种方法能大大缩小海明距离计算的范围^[12]。

但该方法最多有 $2^{16}=65\,536$ 个 key,理想情况下 80 亿签名若平均分布在 65 536 个 key 下,每个 key 下会有 $122\,070 \times 3$ (因每个 value 被存储 3 次故乘上 3) 个 value,即每次签名比较需要 $122\,070 \times 3$ 次海明距离计算,这个计算规模还是比较大的,在要求实时的需求下,难以实现。

从以上可知,查找 n 位签名之间海明距离相差 1 到 k 的签名,将 n 位签名分为 $m(m>k)$ 个区间,在至少有一个区间相同的情况下,区间分得越多,所需要存储的 key 值越少,但每个 key 下存储的 value 值越多。为达到查找过程中时间复杂度和空间复杂度的平衡需要根据实际的需求,合理地对签名值分区间后存储。下面介绍当 k 值为 3 或 4 时,64 位 Simhash 签名值比较合理的分块存储方案。

当 k 为 3 时,将 64-bit 按照 16 位划分为 4 个区间,每个区间剩余的 48-bit 再按照每个 12-bit 划分为 4 个区间,每个 16 位 key 与 12 位 key 两两组合作为新 key,因此每个签名会产生 16 个 key,每个 key 28 位,value 为含有该 28 位 key 的签名,这种情况不会遗漏任何一个相似的签名。

当 key 为 4 时,将 64 位 key 分为 11、11、11、11、10、10,任意选其中两段为 key,有 15 种组合方法,key 的位数有 20、21、22 三种,这种分段方法 key 大约有 280 w 个。假设有 10 亿篇文档,每个 key 下平均有 357 个签名,则新来一篇文档时需要 $15 \times 357=5\,355$ 个海明距离。将 k 设得更大时,每次需计算海明距离更多。

分表时需要考虑空间和时间的折中,key 位数越多,每个 key 下 value 越少,这样每次需要计算的海明距离越少。该系统中 k 值根据需求为 4,即采取的上述 key 为 4 时的分块方案,以达到空间和时间复杂度上的平衡,再辅之 Redis 高效的存储,实现签名值的高效检索^[13]。

2.2 文档特征权值计算

Simhash 算法,以文档中出现的单词作为文档的特

征,单词的频率作为每个特征的权重。单词的频率,虽然是衡量文档特征的一个重要指标,但是仅仅以频率作为权重,还是会丢失一定量的信息。

例如,对于语句,“夏天热”,切分后的结果为,单词“夏天”,频率 1,单词“热”,频率 1。虽然两个单词的频率均为 1,但这句话的核心是夏天,其代表着这句话更多的特征,因此,应该给予该词更大的权重。即从词性的角度来说,名词表征着文档更多的特征。因此,可以将词性作为衡量单词权重的一个因素。规定在词性方面,名词权重最高,动词次之,形容词再次之,其余最低,权重值如表 1 所示。

表 1 单词词性权重表

词性	权重
名词	4
动词	3
形容词	2
其他	1

另外规定在计算单词整体权重时,词频权重 $w_f=0.5$,词性权重 $w_n=0.5$,设词频为 f ,词性对应的权重为 w_{ni} , i 为 1,2,3,4,分别对应名词、动词、形容词、其他,则单词的权重计算公式为:

$$w = f * w_f + w_{ni}$$

将词性作为衡量单词权重的一个因素,能够更全面地表征文档的特征,这样所获得的 Simhash 签名值也更合理,进而提高判断文档相似的准确率^[14]。

2.3 Simhash 签名计算技术

文档反作弊技术中文档的 Simhash 签名计算是其核心过程。本节介绍 Simhash 签名计算的过程。

2.3.1 计算总体流程

Simhash 签名主要分为如下几步:

(1) 如果请求的参数直接传递的是离散化的文档特征,直接执行第三步;如果请求参数是文档内容,执行第二步;

(2) 获得离散化后的文档特征;

(3) 根据 Simhash 算法,计算文档签名。

因此,计算文档 Simhash 签名的流程如图 2 所示。

2.3.2 获取文档特征

Simhash 算法以离散化后的文档特征作为基础,计算文档的签名。提取的文档特征,越能表征原文档的内容的含义,生成的签名就越有意义。传统的 Simhash 以文档中出现的单词和单词频率作为文档特征,会丢失一部分信息,该系统将单词的词性也作为表征文档特征的一个因素,这在 2.2 节已经详细讲述。同时为了提高计算的准确率,还会对计算过程做一些基本的处理,例如文档预处理等。

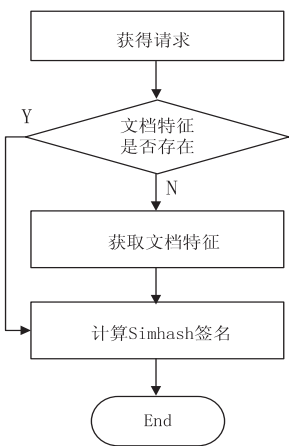


图 2 Simhash 签名计算流程图

该系统中,获取文档特征的主要步骤如下:

- (1)对文档进行预处理(可选);
- (2)对预处理后的文档进行分词;
- (3)去停用词(可选);
- (4)统计单词频率,获取单词词性;
- (5)根据 2.2 节算法所述,计算单词权重。

经过以上 5 步,就可以从给定的文档内容,得到离散化后的文档特征,为计算文档 Simhash 值提供依据。

其中的第一步,对文档进行预处理是可选的,即各个实例再发送请求的时候,可以根据自身的需求决定,是否需要文档进行预处理。预处理过程主要对文档内容按序做如下处理:

- (1)去 html 标签;
- (2)全角转半角;
- (3)英文字母大写转小写;
- (4)繁体转简体;
- (5)去空格。

对文档进行预处理的目的是,尽量剔除文档中一些无关特征或者无意义特征对文档的影响,如果某个实例使用系统服务时,认为文档预处理会丢失文档重要特征,可在请求中配置 textnorm 参数值,表明此请求不需做预处理;每个实例也可以自己对文档进行预处

理后,再向系统发送请求。

2.4 基于全文的 Simhash 判重实现

基于全文的 Simhash 判重,是指文档判重的粒度为整个文档,即根据整个文档的内容生成 Simhash 签名,然后根据计算 Simhash 签名的海明距离判断文档的相似性。该种方式,是该系统提供的文档判重的主要方式。

首先,基于文档全文计算出文档的 Simhash 值,并计算出与待检测文档海明距离为 4 以内的文档;然后,根据请求参数,决定是否需要重新设置被匹配文档的失效时间。

2.5 基于段落的 Simhash 判重实现

对文档做基于全文的 Simhash 判重,其粒度较大,很容易被作弊者绕过,如在原文前后加上一段,或中间串一段文本,都会导致海明距离变大。在计算精度要求比较高的场合,需要更细粒度的签名计算,例如基于段落的签名计算。基于段落的 Simhash 判重,与基于全文 Simhash 判重的不同点是,需要对待处理文档进行分段,然后对每段求 Simhash 签名。

3 测试结果及分析

本节首先对 Simhash 判重的准确率进行测试,为了验证 Simhash 文档相似度计算效果,抽样取了 103 对文本,计算出文本的 64 位签名,并比较每对签名的海明距离,测试结果如图 3 所示。

横坐标为文档对 id,206 篇文档,103 对;纵坐标为每对文档的海明距离。图中海明距离可分三层,第一层为 0~4,第二层为 5~19,第三层为 19 以上,通过文档内容与图 3 比较分析可得:

- (1)文本全部相同或长文本中十几个字节相异,这种文本的海明距离一般在 4 以下。
- (2)样本中有部分文本,内容差异很大,这类文本海明距离比较大。

样本中有两类文本海明距离一般处于 5~19 这

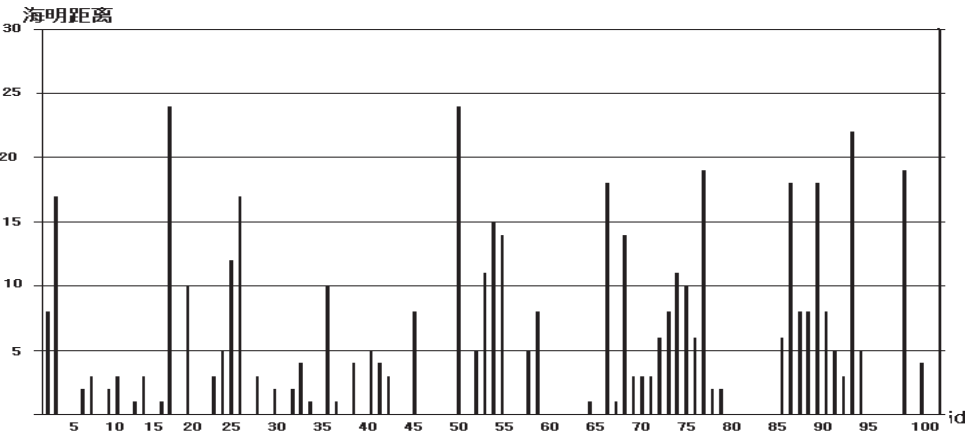


图 3 Simhash 判重统计效果图

层,一种为文本部分相同,如文本由两段组成,前一段相同,后一段不同;一种为短文本,十几个汉字,但有一两个相异。

接下来对系统的请求量与耗时的关系进行测试,维持系统中存储的数据,1 000 w 个文档不变,不断提高系统每秒的请求数,比较每个请求的耗时时间。

从图4可以看出当系统数据量一定时,请求量-耗时关系有如下变化趋势:

(1)请求频率在2 000次/s时,耗时稳定在10 ms以内,此时请求数不是系统服务的瓶颈;

(2)请求频率在6 000次/s时,耗时稳定在20 ms左右,此时请求数不是服务的瓶颈;

(3)请求频率超过6 000次/s后,耗时显著增长,说明此时请求数成为影响服务的瓶颈。

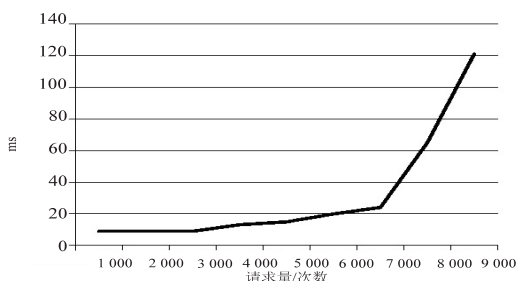


图4 请求量-耗时关系图

下面对系统文档数据量与耗时的关系进行测试,保持请求频率为4 000次/s,逐步提高文档的数据量,从100 w个文档提高到1亿文档,计算比较请求的耗时时间。测试结果如图5所示。

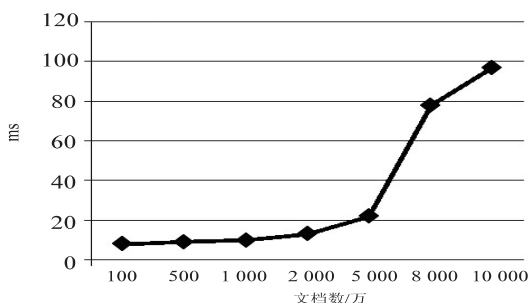


图5 数据量-耗时关系图

从图中可看出,当文档数在5 000 w以内时,请求耗时稳定在20 ms以内;超过5 000 w,文档数量大小对请求耗时影响变大,但时间均稳定在100 ms以内。

从理论上分析,文档失效时间设置得越长,其在存储系统中存储的时间就越长,存储空间变大,影响查询效率,因此对请求耗时有影响。这也是系统设置对冷热数据进行处理的原因。

4 结束语

文中结合目前在文档反作弊方面的需求,进行了基于 Simhash 的海量文档反作弊技术研究,通过改进

的 Simhash 算法可对外部请求做出实时响应。研究包括新实例注册,实例数据导入,相似文档查找;文档判重可基于用户、全文、黑库维度的判重策略;在粒度上,支持全文和段落粒度的 Simhash 判重;支持冷热数据的处理;文档反作弊技术建立在海量数据基础上,目前每个实例可以支持2亿文档的规模;另一方面,通过对冷热数据的处理策略,可以使实例的数据维持在一个比较稳定的范围内,不会因为实例本身数据的增长而过快增长。通过提供多个维度的文档判重策略,判重的时间一般稳定在10 ms以内,在请求高峰时会接近100 ms,但一般不会超过100 ms,可以说,实现了在大规模数据基础上,对文档判重的实时检测。

参考文献:

- [1] 高凯,王永成,肖君.网页去重策略[J].上海交通大学学报,2006,40(5):775-777.
- [2] Manku G S, Jain A, Sarma A D. Detecting near-duplicates for Web crawling[C]//Proceedings of the 16th international conference on World Wide Web. Banff, Alberta, Canada: [s. n.], 2007.
- [3] Zhang Zuping, Xu Xin, Long Jun, et al. Parameters correlation and optimization in text similarity measurement[J]. Journal of Chinese Computer Systems, 2011, 32(5): 983-988.
- [4] Koley A, Chowdhury A, Alspector J. Improved robustness of signature-based near-replica detection via lexicon randomization[C]//Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining. [s. l.]: [s. n.], 2010: 605-610.
- [5] Manber U. Finding similar files in a large file system[C]//Proceedings of the USENIX winter 1994 technical conference. [s. l.]: [s. n.], 2009: 1-10.
- [6] 郭双宙,梁金兰.构件库用户反馈子系统的客观反馈的设计[J].计算机技术与发展,2007,17(5):129-132.
- [7] 董博,郑庆华,宋凯磊,等.基于多 SimHash 指纹的近似文本检测[J].小型微型计算机系统,2011,32(11):2152-2157.
- [8] 刘件,魏程.中文分词算法研究[J].微计算机应用,2008,29(8):11-16.
- [9] 龙树全,赵正文,唐华.中文分词算法概述[J].电脑知识与技术:学术交流,2009,5(4):2605-2607.
- [10] 胡金栋.网页正文提取及去重技术研究[D].杭州:浙江大学,2011.
- [11] 张森.人人网广告精准投放与反作弊系统设计与实现[D].长春:吉林大学,2012.
- [12] 段飞.相似网页识别算法的研究与实现[D].北京:北京邮电大学,2011.
- [13] 肖鹏元.基于GPU并行计算的重复文本检测系统[D].杭州:浙江大学,2011.
- [14] 宋万鹏.短文本相似度计算在用户交互式问答系统中的应用[D].合肥:中国科学技术大学,2010.

基于Simhash算法的海量文档反作弊技术研究

作者: 徐济惠, XU Ji-hui
作者单位: 宁波城市学院, 浙江 宁波, 315100
刊名: 计算机技术与发展 
英文刊名: Computer Technology and Development
年, 卷(期): 2014(9)

本文链接: http://d.wanfangdata.com.cn/Periodical_wjfz201409023.aspx