文章编号:1001 - 9081(2005)07 - 1713 - 03

网页查重技术在企业数据仓库中的应用

白广慧[1,2],连 浩[2],刘 悦[3],程学旗[3]
(1. 中国网通集团 研究院,北京 100036; 2. 中国科学院 研究生院,北京 100039;
3. 中国科学院 计算技术研究所,北京 100080)
(baiguanghui@rd-bta.com.cn)

摘 要:介绍了处理网页排重的三类通用方法,并介绍了在企业数据仓库系统中,通过利用相似性检索技术实现情报资料自动排重的应用。通过对测试结果的评估表明,这种基于相似性检索技术的自动排重的方法能够达到较好的效果,实现了企业情报资料智能化预处理的应用。

关键词:数据仓库;网页查重;支持向量机;向量空间模型
中图分类号: TP391.3 **文献标识码**:A

Automatic detection of online duplication documents and its application in enterprise data warehouse

BAI Guang-hui[1,2], LIAN Hao[2], LIU Yue[3], CHENG Xue-qi[3]
(1. *China Netcom Group Labs, Beijing* 100036, *China*; 2. *Graduate School, Chinese Academy of Sciences, Beijing* 100039, *China*;
3. *Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080, *China*)

Abstract: Three general methods to detect duplicate Web pages were introduced. The similarity search technique was used to detect duplicate information automatically in enterprise data warehouse. The results indicate that the similarity search method is fit for intelligent pretreatment of enterprise intelligence data.

Key words: data warehouse; online duplicate documents detection; Support Vector Machine (SVM); Vector Space Model (VSM)

0 引言

建立企业自身的数据仓库,并依据仓库中的数据提炼出企业有价值的信息情报是提升企业竞争力的一个重要环节。美国 90% 的公司均拥有自己的数据仓库系统;全球 500 强企业中,几乎所有企业更是如此。

在企业数据仓库中一个重要的信息渠道就是采集互联网中本行业的相关信息。网络机器人每天采集信息的数量非常大。其中,不乏许多转载、重复的信息,需要进行排重处理后,情报人员才能在此基础上,分析、提炼出高质量的情报信息。

对于海量信息,若单单依靠人工进行排重处理,不仅耗费宝贵的人力资源,而且时效性也不能满足实际需要。在数据仓库实施过程中,我们采用了自动排重的技术,综合运用了基于关键词过滤的规则方法和基于支持向量机的相似性检索技术。排重准确率能够达到实用化要求,并高效地节省了人力资源。

本文重点介绍三种通用的排重技术以及本系统采用的支持向量机的相似性检索技术,并对它的排重效果进行评估。

1 网页查重技术概述

网页查重技术源于复制检测技术。复制检测,就是判断一个文件的内容是否抄袭、剽窃或者复制于另外一个或多个文件。剽窃不仅仅意味着原封不动地照搬,还包括对原作的移位变换、同义词替换以及改变说法重述等方式。

在互联网中,一篇文献可能以 pdf,ps,word 等多种格式存在于多个网站上。在企业数据仓库系统中,由于实时采集互联网信息,因此也不可避免地需要应用排重技术,排除重复信息。这既节省网络资源,也节约工作人员的时间和精力。

1.1 网页查重技术相关的问题

问题 1 处理重复的文档必然会影响到精确度和效率。

1) 文档排重就会减少提供给用户的可靠信息量,即降低了结果的精确度。

2) 处理重复文档需要额外的计算量,对用户而言是不可见的,若时间耗用太多,用户会认为系统的效率太低。

问题 2 对于重复的定义并不精确。

很多站点有多个名字:如 www.fox4.com, onsale.channel9.com 和 www.realtv.com 都是指向同一个站点的;标题相同也可能内容并不完全相同。因此普遍认为:若某文档包含了和另一文档相同的语义内容,则就是重复的。

重复的四种模式如下:1) 若 2 篇文档内容和排版上毫无差别,则是 full-layout 重复; 2) 若 2 篇文档内容相同,但排版不同,则是 full-content 重复; 3) 若 2 篇文档有部分重要内容相同,且排版相同,则称为 partial-layout 重复; 4) 若 2 篇文档有部分重要内容相同,但排版不同,则称为 partial-content 重复。

排重的处理步骤如下:

第一步:从输入的文档中提取出适当的特征;

第二步:和以前输入的文档的特征进行比较和判断。

收稿日期:2004 - 12 - 24;**修订日期**:2005 - 03 - 11

作者简介:白广慧(1976 -),女(回),山东德州人,工程师,主要研究方向:计算机及应用、网络安全; 连浩(1980 -),女,湖北武汉人,硕士研究生,主要研究方向:中文处理、网络安全; 刘悦(1971 -),女,山东泰安人,博士,主要研究方向:P2P 网络、Web 搜索引擎; 程学旗(1971 -),男,安徽安庆人,研究员,主要研究方向:网络与信息安全、大规模内容计算、P2P 网络、信息网格.

1.2 DSC 和 DSC-SS 方法

早先提出的完全用于查重的两种算法是:DSC 算法(digital syntactic clustering)和 DSC-SS 算法(DSC's super shingle)。

这两种算法使用较为广泛,但效率很低。还有一个明显的缺点是:在处理较小文档的时候效果很差。

下面简单描述一下算法的工作过程:

1) shingling 技术

首先把文档转换成相同的文本格式,把 w 个连续的单词称为一个 shingle,然后选取一定量的 w 长的 shingle 构成子集合;比较 2 篇文档之间重叠的 shingle 的数目,就可计算出 2 篇文档的重叠度。由于比较的是子文档,这样可减少系统运行时间,但每篇文档却可能产生不止一个副本的匹配,返回这种结果的话,用户就需要自己处理大量的副本,降低了系统的可用性。为了改善这种状况,可以应用一些优化算法减少比较的次数:一种是减少 shingle 的个数,但会降低精确度,甚至会判断错误。

DSC 的改进算法 DSC-SS 使用的方法是:super shingles。就是将几个 shingle 合起来形成一个 super shingle,要比较的就是唯一的一个 super shingle,而不是大量的 shingle。这使运行时间和效果都有提高。但是也使 DSC-SS 处理较小的文档效果很差。

2) Similarity measure calculations

这步类似于文本分类,使用相似度计算对可能重复的文档进行分类。每篇文档都要和其他所有文档进行一次比较。这一步看上去似乎不可行,因为每篇文档和其他所有文档比较一次,运行时间是 $O(d^2)$,d 是文档数,实际上在上一步就把完全没有重复的文档过滤掉了,所以实际运行时间和处理的数据有关,较难估计。

该算法的目标是:能够处理大量的小文档,在处理的时候每篇文档只被分入一类中,要么是非重复的,要么只存在于一类副本中。

1.3 I-Match 方法

I-Match 是在 DSC 和 DSC-SS 基础上开发的一个新系统。I-Match 不依赖于严格的语法分析,而是使用集合统计的方法来识别哪些连续的字符串被选为比较的基石。每个串的 idf 被定义为 $tx = \log(N/n)$。N 是集合中的文档数目,n 是文档中包含的某串的数目,数值较高的即重复次数少被舍去。

输入一篇文档,过滤出一些关键串,并且计算出这篇文档的 Hash 值,Hash 值相同的文档就是重复的。使用 SHA1 作为 Hash 函数,因为它的速度很快而且适用于任何长度。

I-Match 方法算法伪代码如下:

```
Get document;
Parse document into a token stream, removing format tags;
Using term thresholds (idf), retain only significant tokens;
Insert relevant tokens into unicode ascending ordered tree of unique tokens;
Loop through token tree and add each unique token to the SHA1 digest; Upon completion of token tree loop, a (doc id, SHA1 Digest) tuple is defined;
The tuple (doc id, SHA1 Digest) is inserted into the storage data structure based on SHA1 Digest key;
If there is a collision of digest values then the documents are similar
```

算法分析:

SHA-1 生成一个 20 字节的 Hash 值,并且使用一个安全的冲突消解算法,使得不同的标志串生成相同的 Hash 值的概率低于 $P(2^{-160})$。

把 <docid, hashvalue> 元组插入树结构的时间复杂度是 $O(d \log d)$,其他的如检索数据结构(Hash 表)需要 $O(d)$。对副本的识别是在将数据插入 Hash 数组或是树结构中进行的,任何 Hash 值的冲突就表示检测到一个副本。最坏的情况下时间复杂度是 $O(d \log d)$,集合中所有的文档都是重复的,最好的时候是 $O(d)$。

1.4 对文档集合进行比较

DSC,I-Match 等方法是基于一篇篇文档进行比较的。现今的网络中为了实现更迅速的本地读取服务,更高的可用性,大量的文档集在各个服务器之间互相复制。一般而言,这些文档集合由于使用频度比较高,所以在上百个站点上都有镜像,如 LDP collection 包含 25M 的数据,上千页文档,全世界有 180 个服务器上有它的镜像存在。

要识别备份或是镜像的一个主要的困难是:网络中很多备份不是严格相同的,导致这种情况的原因有:不同的镜像更新率不同,镜像的覆盖范围不同,还有格式不同。

如果发现 2 个站点之间有大量的相似文档时,就可以将这 2 个站点进行比较。此时只要比较这 2 个站点内的超链接的情况就可以了。

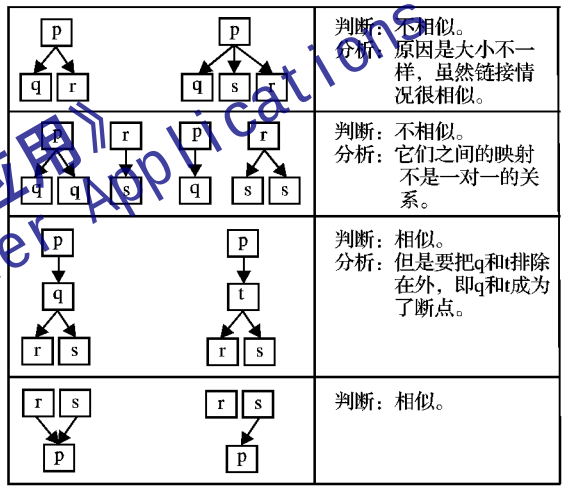

图 1 链接的相似度分析

相似的定义如下:

对 2 个集合 C_1, C_2,有 $p \in C_1$,定义 $P_1(p)$ 是 C_1 中所有存在指向 p 的超链接的网页集合。同样 $P_2(M(p))$ 是 C_2 中所有指向 $M(p)$ 的网页的集合。若存在 $p_1 \in P_1(p)$ 并且 $p_2 \in P_2(M(p))$ 且有 p_1 和 p_2 相似,则可以说 C_1 和 C_2 相似。(在 $P_1(p)$ 和 $P_2(M(p))$ 都不为空的情况下)。

我们把一组大小相同的网页集合称为一个群。相似群定义如下:

若一个群 $R = \{C_1, C_2, \cdots, C_n\}$,其中的集合两两相似,那么就称这个群是相似的。这样,我们的目标就是找出一组网页中的相似群。

相似群的查找算法如下:

1) 寻找最小群(trivial cluster)

最小群是由相似的网页组成的。即它所包含的 collection 大小为 1,且每两个 collection 中的网页都是相似的。

例如:图 2 中两个网页的标记都是 t,它们是相似的,它们就组成一个最小群。

2) 合并最小群

合并最小群的约束条件:

a) 找出所有的最小群之后,再合并这些最小群,就可以找到一个群了。

b) 设有最小群 $R_i = \{p_1, p_2, \cdots, p_n\}$ 和 $R_j = \{q_1, q_2, \cdots, q_m\}$,我们定义:$s_{i,j}$ 为 R_i 中有链接指向 R_j 中某页的网页的数目,$d_{i,j}$ 为 R_j 中有链接指向 R_i 中某页的网页的数目,$|R_i|$ 是 R_i 中网页的数目,$|R_j|$ 是 R_j 中网页的数目。

c) 合并的条件是:$|R_i| = s_{i,j} = d_{i,j} = |R_j|$。

图2 中的最小群 a 和最小群 b,它们都含有 3 个 collection,而且每个标记为 a 的网页都有一个超链接指向 b。所以这两个最小群满足合并条件,可以合并形成一个含有 3 个 collection 的群。同样道理可以把最小群 c 和最小群 d 也合并进去。合并结果为相似群。

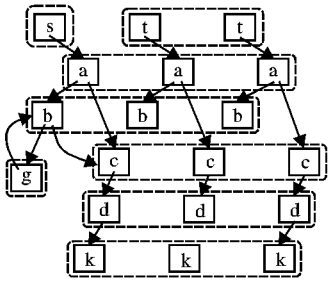

图2　最小群示例

1.5　各类方法比较

DSC, I-Match 等方法将网页作为单独的个体,所有的网页存放在一个大的集合里,两两进行比较,以提高搜索结果的可用度。

网页集合比较方法将网页作为集合,查找备份或是镜像,若是备份或镜像,则只用查找其中的一个即可,节约了搜索时间。

总的来说:若要面向大众化,使用第一种方法会比较好一点。如像"球赛"等大众化的话题,不可能会有服务器备份这种内容,就算有,它们的更新率很高,可能过几天内容就已经完全不同了。第二种方法只有在查找较专业内容时比较好。

2　本系统使用的方法

考虑本系统的定位,采集本专业相关的信息资料,因此,在参考网页及集合排重方法的基础上,本系统使用了相似性检索技术进行自动排重。相似性检索是指对于给定样本文献,在文献数据集合中查找出与之内容相似的文献的技术。

相似性检索技术需要在文献数字化表示(空间向量模型 VSM)的基础上,通过计算文献之间的相似程度(向量之间的距离)给出文献之间的相关度指标。

本算法主要基于特征词提取和倒排索引技术:

1) 对样本库中的每篇文档进行自动分词和提取特征词;

2) 对样本库中的文档按特征词建立倒排索引库,建立索引的相关属性,包括词频、位置以及文本长度等;

3) 在进行相似性检索时,对输入的检索文档分词并提取特征词,然后按特征词在倒排索引库中查找到与之相关的文档,获得词频、位置及文本长度等属性;

4) 根据每篇文档中包含特征词的多少、位置、词频、文档的长度等信息来计算样本库中文档与待检索文档的相关度,相关度超过一定阈值的文档即可作为相关文档处理,并给出相关系数;

5) 系统为了提高排重的效果,还增加了同义词库,从语义上扩大相重信息的含义。

3　本系统的排重测试及评价

利用采集器从互联网上下载网页测试排重的效果。首先利用没有加入排重功能的采集器从互联网上下载 10 多万篇网页。分别利用前 500 篇、1 000 篇网页作为样本库,然后采集器加入排重模块,重新下载十多万篇网页,测试结果如表 1。

表 1　测试结果

样本组成	召回率(R)	准确率(P)
前 500 篇作为样本库	80.0%	94.5%
前 1 000 篇作为样本库	82.0%	90.5%

召回率(Recall) =
算法发现的重复网页数/网页总数 × 100%
准确率(Precision) =
正确重复的网页数/算法发现的重复网页数 × 100%

由于测试的网页较多,无法一一判断哪些是重复的网页,我们将算法发现的重复网页分为 10 份,在这 10 份中随即挑选 100 个网页,人工判断这 100 个网页的准确率,用这 10 个准确率的平均值作为算法的准确率。结果表明,采用此算法自动排重的平均准确度可以达到 90% 以上,可以满足企业的实用化要求。

参考文献:

[1] CHO J, SHIVAKUMAR N, GARCIA-MOLINA H. CA 94305, Finding replicated web collections[R]. Department of Computer Science Stanford, 1999.

[2] 鲍军鹏,沈钧毅,刘晓东,等. 自然语言文档复制检测研究综述[J]. 软件学报,2003,14(10).

[3] CHOWDHURY A, FRIEDER O, GROSSMAN D, *et al*. Collection Statistics for Fast Duplicate Document Detection[J]. ACM Transactions on Information System, 2002, 20(2):171 - 191.

[4] LOPRESTI DP. Models and Algorithms for Duplicate Document Detection Bell Labs[A]. Proceedings of the Fifth International Conference on Document Analysis and Recognition[C], 1999.

[5] CAMPBELL DM, CHEN WR, SMITH DM. Copy Detection Systems for Digital Documents [A]. Advances in Digital Libraries 2000 (ADL 2000)[C], 2000.

(上接第 1691 页)

因此,在未来的应用中,可以很好地满足穿越 NAT,在小区/企业网内部使多媒体通信变成现实。但是,由于所有报文都必须经过 TURN Server 转发,包的延迟和丢包的可能性仍然成为通信的瓶颈,这是要进一步解决的问题。

参考文献:

[1] ROSENBERG J, SCHULZRINNE H, CAMARILLO G, *et al*. RFC3261, SIP: Session Initiation Protocol[S], 2002.

[2] HANDLEY M, JACOBSON V. RFC2327, SDP: Session Description Protocol[S], 1998.

[3] SCHULZRINNE H, CASNER S, FREDERICK R, *et al*. RFC3550, RTP: A Transport Protocol for Real-Time Applications[S], 2003.

[4] ROSENBERG J, MAHY R, HUITEMA C. Traversal Using Relay NAT (TURN)[EB/OL]. draft-rosenberg-midcom-turn-02, 2003 - 10.

[5] FRANKSJ, HALLAM-BAKERP, HOSTETLERJ, *et al*. RFC2617, HTTP Authentication: Basic and Digest Access Authentication[S], 1999.

[6] HUITEMA C. RFC3605, Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)[S], 2003.

[7] ROSENBERG J, WEINBERGER J, MAHY R. RFC3489, STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)[S], 2003..