doi:10.3772/j.issn.1002-0470.2015.10-11.007

基于网页结构的网站检测研究①

李大辉② 何清刚 王佰玲③ 邹新一

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 为了准确检测出仿冒网站,提出了一种基于网页结构的页面相似度计算方法。该方法首先将网页分块并进行合理的筛选,其次通过初步比对确定相似节点群,最后将网页数据量化并计算出网页是否相似。试验表明,该方法可以有效地检测出网页相似情况,对于仿冒网站的镜像尤其明显,误报率及漏报率均不超过10%。

关键词 网页结构,节点筛选,网页比对

0 引 言

近年来随着镜像技术的出现,网站访问变得更加快捷,但是也随之出现了类似仿冒网站的镜像网站,可能会带来像网页钓鱼那样的网络安全问题,而且仿冒网站逐渐向着自动化、智能化、多样化方向发展,这就给仿冒网站的准确检测带来了极大困难。针对这一问题,研究人员加强了网站检测研究。

Jin 等人为了识别广告网站中是否含有敏感内容,设计了一种先进行敏感内容分类、后进行迭代训练数据的方法对是否含有敏感词的公告网站进行分类[1],试验表明该方法准确率很高,但是由于对敏感词内容的准确度要求很高,所以该方法需要做大量的准备工作。Peng 等人采用对网页内容进行分层的方法实现了对网页的分类[2]。Park 等人同样基于网页的内容提出了一种新的方法,他们将网页的小部分数据进行标记,利用训练获得标记值并通过解析未标记的数据最终判定相似度[3],这种方法要求对标记的数据具有很强的代表性。Lin 等人对含有大量文本信息的网页做了大量的分析,并提出了一种可扩展的网页内容识别系统[4]。Guo 等人在内容识别的基础上增加了贝叶斯方法并使用虚拟文件技术训练文本[5],该方法能够准确地对网页进行分类但是时间较长。Li 等人利用非网页文本的知识库来识别网页[6],此方法仅对部分网站适用。朱毅华等人利用网页文档对象模型(document object model, DOM)树中的子树进行矩阵运算[7],通过计算网页向量空间判定网页内容相似度来确定网页相似度[8]。通过上述研究发现,仿冒网站的镜像与原网站具有一定的相似性,在网页结构上体现出了位置相似性及面积相似性。由于基于网页结构特点分析网页相似性的研究仅限于树形结构特点并且准确率较低,本文提出一种基于网页结构的页面相似度计算方法,该方法运用网页结构对特定网站进行检测,网页结构的检测主要以网页 DOM 树为依据,并根据面积大小在不改变树形结构的情况下对树节点进行筛选和合并得出 VTree,通过 VTree 的相似节点个数以及碰撞面积比较两网站的相似性,将网页数据量化并计算出网页是否相似。

1 基于网页结构的网页相似性检测技术

通过网页标签结构可以生成二维树形结构,通常称之为 DOM 树。在整个网页的 DOM 树结构中有布局信息、文本、图片信息以及脚本语言等构成的

① 国家自然科学基金(61170262,61371177)资助项目。
② 男,1990 年生,硕士;研究方向:网络安全;E-mail: 904070319@qq.com
③ E-mail: wbl@hit.edu.cn
(收稿日期:2015-04-21)

其他信息,将 DOM 树中的节点进行合理的筛选合并把关键属性从中提取出来,对结构相似的网页比对分块内容,以确定待检测网站是否为特定网站,具体设计如图 1 所示。

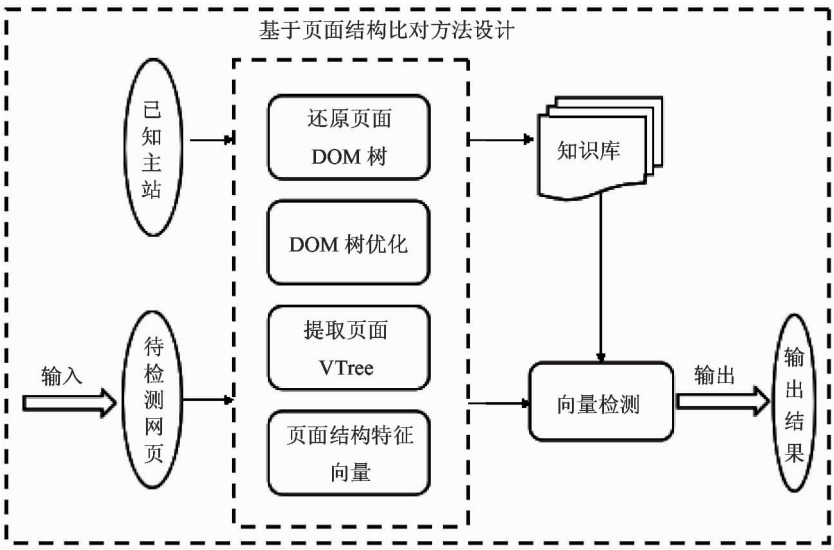

图 1　基于页面结构比对方法

通过对网页进行渲染还原出被保护页面的 DOM 树,根据 DOM 树中的节点特性对树中节点进行筛选。筛选完毕后对生成的 VTree 进行进一步的解析,并提取网页结构特征向量,将其作为依据存入相应的知识库中。进行比对检测时,使用同样的方法将待检测网站的特征向量提取出来,并与知识库中已有的特征向量进行比对,得出相应的可疑网站列表。

1.1　网页分块算法

一个页面 VTree 的节点是 DOM 树节点的子集,与 DOM 树不同,VTree 中的每个节点都对应页面实际显示中的某一部分。其中位置坐标为一个四元组 *Position*(*top*, *left*, *heigh*, *width*),其中 *top* 和 *left* 分别指节点显示区域矩形左上角定点与页面上边沿和左边沿的像素距离,*heigh* 指高度,*width* 指宽度。通过四元组可以确定节点的大小以及显示区域,并且可根据 *heigh* 和 *width* 过滤掉 DOM 树中面积为零的节点。

设 father 为节点的父节点,child 为节点的子节点,*min _ num* 为最小阈值,*pp _ num* 为正序遍历的筛选阈值,*rt _ num* 为倒序遍历的筛选阈值。网页分块算法如下所示:

01

算法输入:页面 DOM 树根节点。
算法输出:筛选后的树根节点。
算法伪代码:

```
01 def node _ selection(List DTree)
02     while node in DTree then
03         if heigh * width < min _ num then
04             delete node
05         end if
06         node = node + 1
07     while node in DTree then
08         if heigh * width > pp _ num then
09             node-> father-> child + = node-> child
10             delete node
11         end if
12         node = node + 1
13     while node in DTree then
14         if heigh * width < rt _ num then
15             if node has child then
16                 add _ child _ attribute(node, node-
                   > child)
17                 delete node-> child
18             end if
19         end if
20         node = node-1
21     while node in DTree then
22         if node has child then
```

```
23                  if all node->child(heigh, width) is
                    same & Location adjacent then
24                      if last node-child(top,left) in page
                        then
25                          add _ child _ attribute(node ,
                            node->child);
26                          delete node->child
27                      else
28                          delete node-. child except first
29                      end if
30                  end if
31              end if
32              node = node + 1
33          return DTree
```

2 到 6 行是删除面积小于最小面积阈值的节点的过程,7 到 12 行将 DOM 树正序遍历,若节点面积比阈值大,并完全包含其所有子节点,则删除该节点。13 到 20 行对 DOM 树倒序遍历,若节点面积比阈值小,并完全包含其所有子节点,则删除其所有子节点。21 到 32 行将 DOM 树正序遍历,若节点为父节点且具有大小相同、位置相邻的子节点则计算子节点是否有超出页面外的情况,若有超出情况则为滚动条,只保留一个子节点,删除其他子节点;若均在页面中则为列表节点,把子节点合并至父节点中并删除子节点。

1.2 网页 Vtree 比对算法

对网页 DOM 树节点进行一系列的筛选并得出相应 VTree,由筛选方法得知 VTree 中的节点具有视觉面积适中、节点个数较少、树形结构保留较好的特点。根据以上的特点总结了一种根据视觉面积碰撞配合 VTree 结构的网页相似度比对算法。该算法通过节点快速扫描后得到页面间的相似节点并根据总节点、相似节点、相似节点个数和总面积进行最终的网页相似度判定。设 $node\_num$ 为 VTree 的总节点个数,$node\_area$ 为总面积,f_node_num 为 VTree 的扫描总节点个数,f_node_area 为扫描总面积,s_node_num 为 VTree 的相似总节点个数,s_node_area 为相似总面积,遍历节点占比:$p\_node\_p = f\_node\_num / node\_num$,遍历面积占比:$t\_area\_p = s\_node\_area / node\_area$,相似节点占比:$s\_node\_p = s\_node\_num / f\_node\_num$,相似面积占比:$s\_area\_p = s\_node\_area / f\_node\_area$。

计算相似判断阈值,当两个网页相似节点占比和相似面积占比大于阈值时,则判断相似。为了提高判断的准确性,通过对结构相似的不同网页进行分析,发现遍历占比和相似节点阈值成反比例关系,即当遍历的节点越多、面积越大时,相似判断阈值越小。

图 2 和图 3 所示为已知相似情况的网站集中通过计算得出 p_node_p、s_node_p 值与 t_area_p、s_area_p 值所画出的结果分布图,由图可以得出以下结论:

(1) 当两个网页相似时相似节点较多,由于遍历匹配失败时才递增,所以递增次数较少,遍历的节点较多,相似网页的遍历节点占比普遍比较大。但由于排版不同等原因,两个相似节点的位置不同,导致检测中无法确定其是否为相似节点,所以检测到的相似节点较少,相似节点占比普遍较低。

(2) 圆形点集中于横纵坐标均较低的区域,说明当遍历时发生较多递增,且遍历的节点中相似节点较少。绝大多数情况下,说明这两个网页不相似。

(3) 在遍历节点占比为 0 ~ 8 这个区间时,由于遍历时发生多次递进,所以两个网页相似节点很少,大多数情况下为 0,网页也均为不相似。

(4) 当两个网页遍历节点占比为 100 或接近 100 时,基本为相似网页。因为遍历时匹配基本全成功,即所有对应节点的长宽基本相同,相似节点占比不高或许因为节点的移位等原因,所以几乎均为相似网页。

通过测试发现,相似网页测试结果多集中在第一象限的右上部,不相似网页测试结果多集中在第一象限的左下部,所以当该曲线为反比例函数在第一象限的曲线时,可以较好的将相似网页和不相似网页的测试结果分开,从而满足计算的简便性的同时尽可能降低误差。

由图 3 可看出,面积比对时,相似网页和不相似网页的测试结果分布随机性强。这就导致相似网页的相似面积占比较分散,随机性强。但是总的趋势依然是相似网页测试结果多集中在第一象限的右上部,不相似网页测试结果多集中在第一象限的左下部,所以曲线依然满足反比例函数。

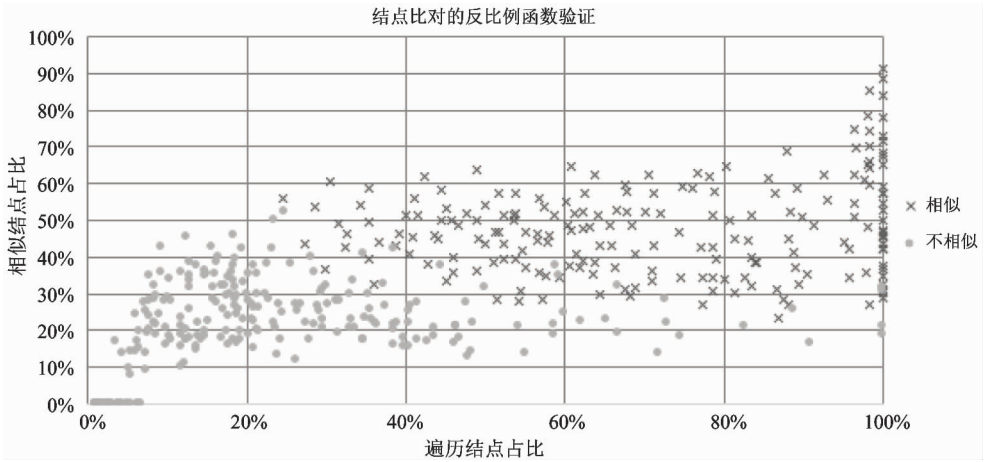

图 2　p _ node _ p、s _ node _ p 关系分布图

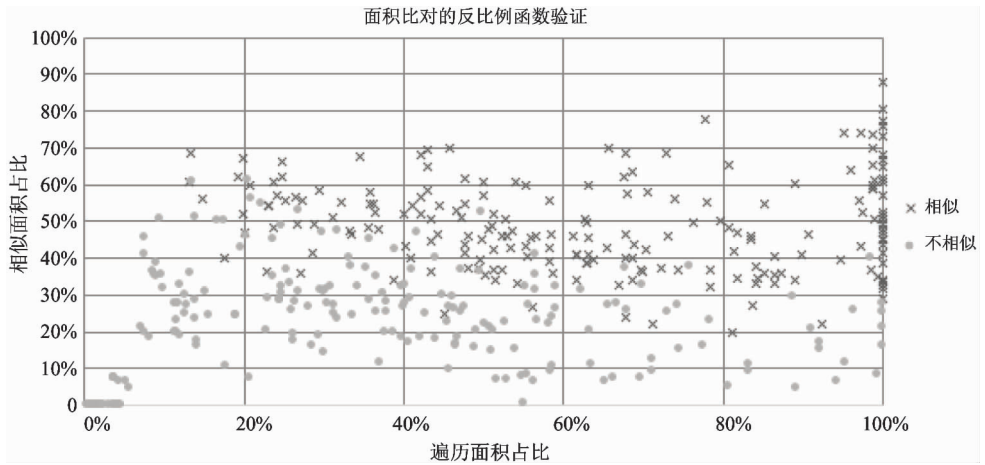

图 3　t _ area _ p、s _ area _ p 关系分布图

通过线性回归分析可以粗略得到一种反比例函数(形如 $y = k/x$(k 为常数,$k \neq 0, x \neq 0$) 的函数),利用反比例函数,根据预先设定的系数 k 和 b,计算相似判断阈值。

设

相似节点数判断阈值为 $node\_num = k_1/p\_node\_p + b_1$

相似节点面积判断阈值为 $area\_num = k_2/t\_area\_p + b_2$

分别计算相似节点占比与相似节点数判断阈值的差和相似面积占比与相似节点面积判断阈值的差。当两个差值之和大于等于 0 时,判断两个网页结构相似,否则判断不相似。

设 n 为扫描阶段节点比对连续失败次数,node 与 node'分别代表两个 VTree 上的节点,l_num、t_num、a_num 分别代表左坐标、上坐标以及面积的差阈值,f _ list 为扫描到的节点,s _ list 为扫描到的并且相似的节点;VTree'中的 f _ list'、s _ list'与 VTree 定义相同,页面 VTree 比对算法实现如下:

算法输入:页面 VTree 根节点。
算法输出:相似度判定值。
算法伪代码:

```
01 def VTree _ compare(List VTree, List VTree')
02     int n = 0
03     while node in VTree & node' in VTree' then
04         if |left - left'| < l _ num & |top - top'| < t _
           num & |heigh * width-heigh' * width'| < a _
           num then
05             save node in f _ list and node in s _ list
06             save node' in f _ list' and node' in s _
               list'
07             node = node + 1
08             node' = node' + 1
09             n = 0
```

```
10          else
11              node = node + (n + 1)^2
12              save node in f_list
13          end if
14      if (s_node_p - node_num) + (s_area_p - area_num) > 0 then
15          return 1
16      else
17          return 0
18      end if
```

2 测试结果

2.1 网页分块方法测试

本测试集由 600 个随机选取的 URL(统一资源定位符,亦称网页地址)作为测试集,记录测试集在不同面积阈值情况下的平均选择节点占比。图 4 示出了分块阈值和视觉块节点占页面总节点数目比率。

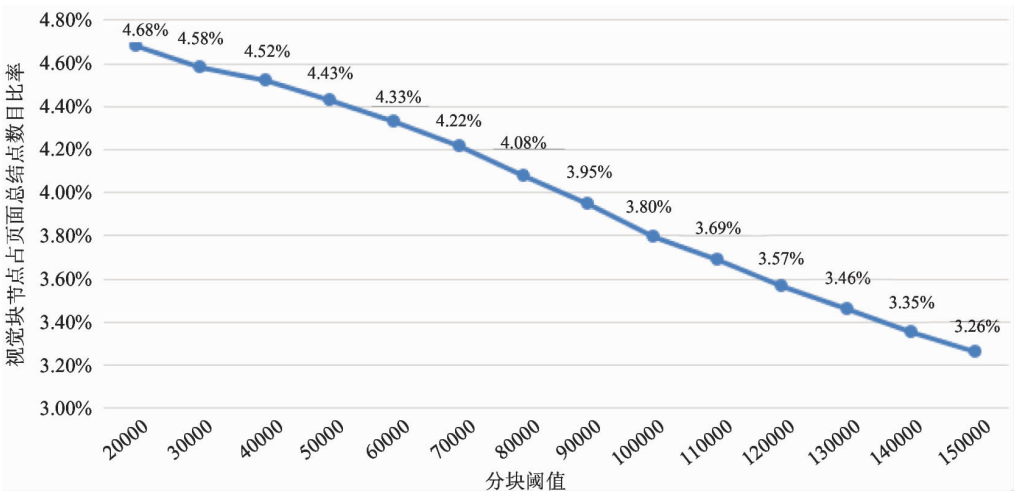

图 4 分块阈值和视觉块节点占页面总节点数目比率

由图 4 可知,在 30000 到 50000 之间时下降较为平缓,当超过 50000 时下降幅度较大,所以本文以 50000 为阈值进行页面分块筛选。

2.2 反比例函数 k、b 最优取值测试结果

由于整个结构比对过程包括节点比对和面积比对两部分,两部分比对的反比例函数参数不同,为了选择最优参数来达到最佳比对效果,同时为了避免相互之间的干扰,本文分别对其测试。图 5 和图 6 分别示出了反比例函数不同参数下节点比对和面积比对的情况。

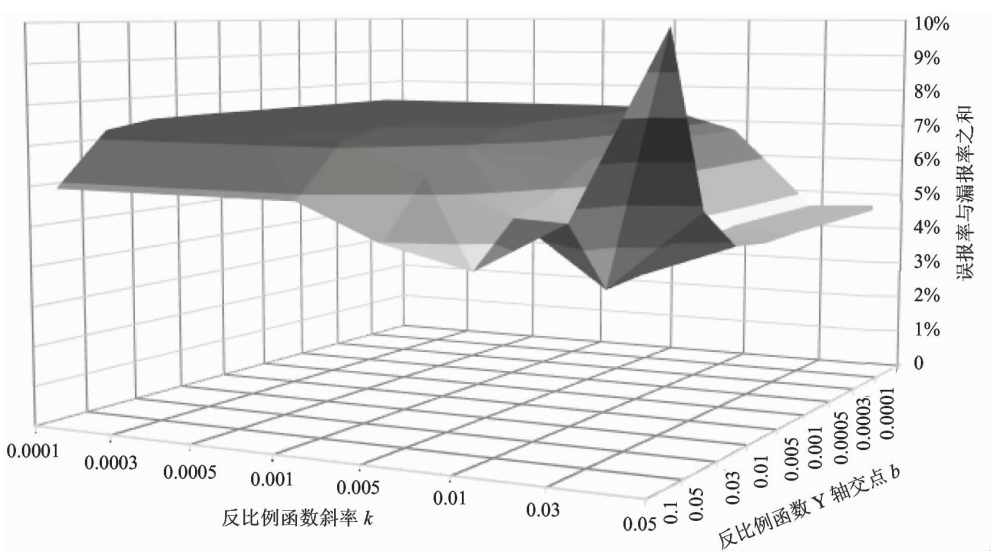

图 5 反比例函数不同参数下节点比对情况

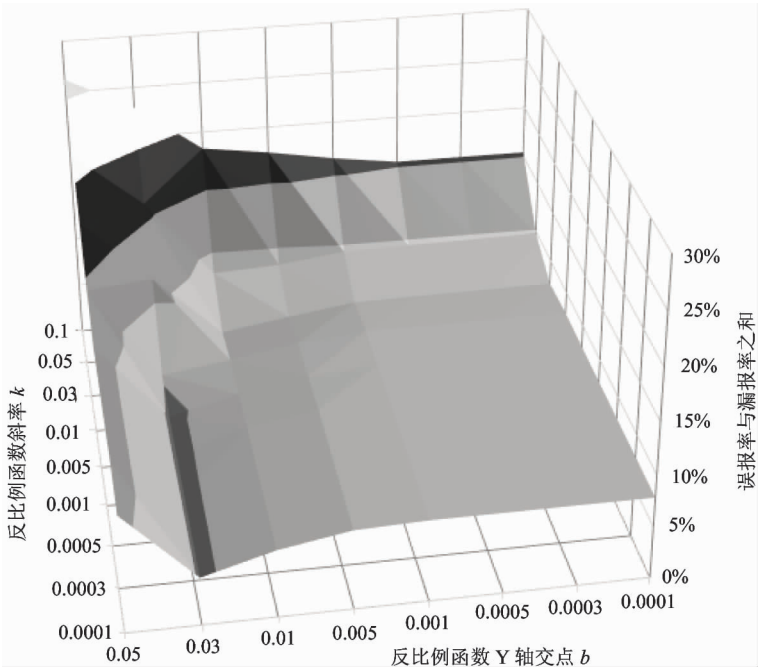

图 6　反比例函数不同参数下面积比对情况

如图 5、图 6 所示,测试时反比例函数的 k,b 的选取范围为0.0001,0.0003,0.0005,0.001,0.005,0.01,0.03,0.05,0.1,0.15,0.2,0.25,随着 k 和 b 的增大,误报率逐渐减小,漏报率逐渐增大。Z 坐标为漏报率与误报率的和,通过图可以看出最低点情况。

总误报率为 610 个可信网站相互比对得到,总漏报率为所有镜像网站、两组论坛网站、两组新闻网站的漏报率取平均值后得到,最终将误报率和漏报率做和得到总结果,结果中最小值对应的 k,b 即为所求最优解,节点比对和面积比对分别测试,并分别得到最优解。由图中可以看出,当 $k_1 = 0.03$、$b_1 = 0.05$ 时漏报率与误报率和最低;当 $k_2 = 0.03$, $b_2 = 0.001$时漏报率与误报率和是最低的。

2.3　总体测试结果

根据反比例函数参数测试中得到的两组最优 k,b,同时用节点比对和面积比对进行测试,测试结果如图 7 所示。

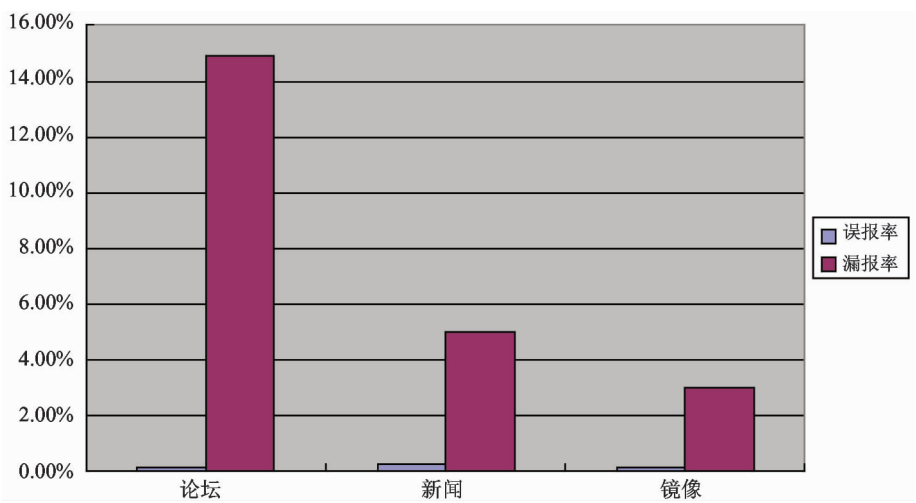

图 7　漏报率统计

论坛类网站测试:主要是对论坛类的网站识别做测试,测试集为 817 个 URL,其中有 47 个论坛类网站主站,经过比对检测出一个非论坛类网站,得出误报率为 0.13%。有 7 个论坛类网站没有通过检测,求得漏报率为 14.89%。由于论坛类网站页面中的楼数,小版块分布会有差异,从而漏报率略高。

新闻类网站测试:该测试主要针对知名新闻类网站进行测试,测试集 760 个正常网页中,有 20 个新闻类网站主站,将这 760 个网站与知名的新闻类网站(新浪、搜狐、腾讯等)进行比对,检测结果有两个为非新闻网站,误报率为 0.26%。有 1 个新闻类网站没有通过检测,求得漏报率为 5%。由此可以看出本方法对于新闻类网站有一定的检测效果。

镜像类网站测试:测试集为 32 个已知镜像网站以及 1110 个正常网站,镜像网站的主站并不在这 1110 个网站中,有 1 例比对成功,求得误报率为 0.09%。所有镜像网站有 1 例未通过检测,漏报率为 3.12%。由此看出,在现阶段的测试中,本方法对于镜像网站的测试效果明显。

3 结 论

本研究从网页视觉块的筛选和网页结构的比对两个方面,运用网页结构对网站进行检测。系统通过正序遍历、倒序遍历、节点筛选将 DOM 树转化为更加简单、特点更加鲜明的 VTree,筛选后的 VTree 具有结构代表性并且筛选节点总数占总节点比重少。根据 VTree 进行相似节点选取与计算。检测表明,本文方法对论坛类、新闻类以及镜像类网站检测准确率高。

参考文献

[1] Jin X, Li Y, Mah T, et al. Sensitive webpage classification for content advertising. In: Proceedings of the 1st International Workshop on Data Mining and Audience Intelligence for Advertising, San Jose, USA, 2007. 28-33

[2] Peng X G, Ming Z, Wang H T. Text learning and hierarchical feature selection in webpage classification. *Advanced Data Mining and Applications*, 2008, 5139: 452-459

[3] Park S B, Zhang B T. Automatic webpage classification enhanced by unlabeled data. *Intelligent Data Engineering and Automated Learning*, 2003, 2690: 821-825

[4] Lin L, Zhou L Z. Leveraging webpage classification for data object recognition. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Fremont, USA, 2007. 667-670

[5] Guo M X, Wu Y Y. Improving technology of webpage classification based on hyperlinks structure information. *Journal of Quanzhou Normal University*, 2008, 26(4). doi: 10.3969/j.issn.1009-8224.2008.04.006

[6] Li J S, Xue W M, Dong N P. Application of the Naïve Bayesian method with user current usage and hierarchy from website in Chinese webpage classification. In: Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China, 2007. 1364-1367

[7] 朱毅华, 张超群, 曾通等. 基于子树相似度计算的网页评论提取算法研究. 现代图书情报技术, 2013, (11): 52-56

[8] 何忠秀, 王霜, 安礼成. 基于向量空间的网页内容相似度计算方法研究. 计算机与现代化, 2010, (09): 53-55

Web site testing research based on webpage structure

Li Dahui, He Qinggang, Wang Bailing, Zou Xinyi

(Department of Computer Science & Technology, Harbin Institute of Technology, Harbin 150001)

Abstract

To detect spoofing websites accurately and effectively, a page similarity computing method based on webpage structure is proposed. The method firstly handles the webpages by segmentation and performs reasonable filtering, then, determines similar-node groups by preliminary comparison, and finally, quantifies the webpage data and calculates whether the webpages are similar or not. The proposed method was tested by experiment, and the results showed that it could detect the similarity of webpages accurately and effectively, especially to the mirror images of spoofing websites (the false positive rate and the false negative rate were both no more than 10%).

Key words: webpage structure, node selection, webpage comparison