# 一种基于统计学特征和 DOM 树的网页去噪技术

何友全[a],徐 澄[b],徐小乐[a],唐华姣[a]

(重庆交通大学 a. 信息科学与工程学院;b. 管理学院,重庆 400074)

**摘 要**:针对特定的网站或网页中抽取出用户感兴趣的信息这一问题,分析现有去噪技术的优缺点,提出了一种基于统计学特征和 DOM 树的 Web 页面去噪方法。该方法首先对原始网页进行预处理,然后分析网页的统计学特征,结合启发式的抽取规则,对网页进行去噪。实验证实该方法在较少人为干预的基础上能达到较好的抽取效果。

**关 键 词**:DOM;统计学特征;信息检索

**中图分类号**:TP393 **文献标识码**:A **文章编号**:1674 - 8425(2011)01 - 0054 - 05

# Approach of Eliminating Web Page Noise Based on Statistical Characteristics and DOM tree

HE You-quan[a], XU Cheng[b], XU Xiao-le[a], TANG Hua-jiao[a]

(a. Information Science & Engineering Department; b. Department of Management, Chongqing Jiaotong University, Chongqing 400074, China)

**Abstract**: In view of extracting the user interested information from specific websites or web pages, this paper proposes an approach of eliminating web page noise based on statistical characteristics and DOM tree after analyzing the advantages and disadvantages of existing web page noise eliminating algorithms. After pre-processing to the original pages, the approach analyzes their statistical characteristics combining with heuristic extraction rules to remove the noise in the web pages. Experiment shows that the approach achieves better retrieval results with relatively little human intervention.

**Key words**: DOM; statistical characteristics; information retrieval

随着互联网的迅速发展,互联网信息爆炸成为影响人们获取有效信息和决策的一个不容忽视的问题。而从特定的网站或网页中抽取出用户感兴趣的信息是解决信息冗余的一个有效途径[1-2],因此,Web 网页去噪和信息抽取方法研究已成为互联网信息处理中的一个热点。随着浏览

**收稿日期**:2010 - 10 - 28

**基金项目**:重庆市科技攻关项目(CSTC,2010AC6074);重庆交通大学研究生教育创新基金资助项目;重庆交通大学实验教学改革与研究基金资助项目(SYJ200922)

**作者简介**:何友全(1964—),男,重庆人,博士,教授,主要从事信息处理、数据挖掘方面的研究。

器/服务器体系结构和动态网页技术的广泛应用，越来越多的网页是根据客户端用户的请求,动态生成的具有较强格式的半结构化网页[3]。现常见的 Web 去噪技术有基于模板的、基于 DOM 树的和基于可视化信息的。

基于模板的网页分块(MPS,model-based page segmentation)方法效果较好,但它对模板依赖性很大,而模板需要人工或自动生成,因此不适合风格不同的网页。基于视觉(语义)的网页分块(VIPS,vision based page segmentation)方法目前尚不成熟,需要人工不断调整启发式规则集,对浏览器渲染核心依赖性大,且速度不够理想。基于 DOM 的网页分块(DPS,DOM—based page segmentation)一般基于预定义好的句法结构,常见的有基于 DOM 树使用 html 标记来识别块[4-5],也有一些学者除了利用标记,还综合考虑了其他一些信息进行网页信息抽取[6-7]。该技术虽简单易行,但效果不稳定,不适于复杂页面。

综合以上方法的优缺点,本文提出了一种基于页面统计学特征和 DOM 分块的改进 Web 信息抽取方法,力图在较少人为干预的基础上达到较好的抽取效果。

## 1 基于统计学特征和 DOM 树的抽取方法

### 1.1 重要概念

本文涉及到的几个基本概念[8]:

1)容器标签 container tag。用来规划网页布局的较大粒度标签,如 <body>, <ol>, <ul>, <table>, <form>,  等,而像 <tr>, <td>, ,由于粒度较小而不包括在内。

2)展现标签 display tag。网页设计者用来调整网页展现效果而加入的标签,如 <b> </br> <h1> 等。

3)内容块文本/字节 content block text/byte。将 DOM 树中以此内容块对应 DOM 节点为根的所有内容块节点的文本/字节相加后作为该内容块文本/字节。若该内容块对应节点为叶子节点,则其文本仅为本节点内部文本/字节;否则需递归加入其所有孩子节点的内容块文本/字节。

### 1.2 方法描述

#### 1.2.1 整体流程

该方法的整体流程如图 1 所示。

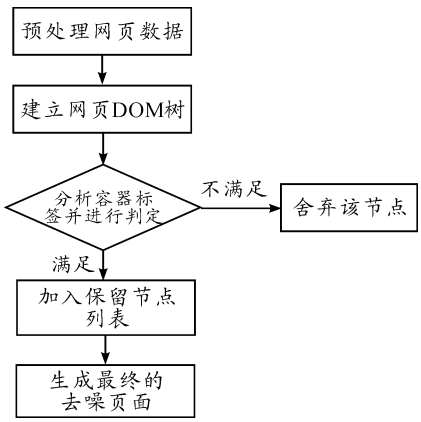


图 1 整体流程

1)页面预处理。由于爬虫抓取回来的网页包含大量用于交互、排版的页面信息,而这些信息和页面正文信息无关,保留的话会对抽取过程造成干扰,所以在建立 DOM 树之前就进行去除,采用的方法是利用正则表达式直接过滤:

```
String regEx_script = "<[\s]*? script[^>]*? >[\s\S]*? <[\s]*? \/[\s]*? script[\s]*? >";
String regEx_style = "<[\s]*? style[^>]*? >[\s\S]*? <[\s]*? \/[\s]*? style[\s]*? >";
p_script = Pattern.compile(regEx_script, Pattern.CASE_INSENSITIVE);
m_script = p_script.matcher(htmlStr);
htmlStr = m_script.replaceAll(""); // 过滤 script 标签
m_style = p_style.matcher(htmlStr);
htmlStr = m_style.replaceAll(""); // 过滤 style 标签
p_html = Pattern.compile(regEx_html,Pattern.CASE_INSENSITIVE)。
```

2)建立页面 DOM 树。HTMLParser[9]是一个

对现有的 HTML 进行分析的快速实时的解析器。它是一个开源项目,通过它可以准确高效地对 HTML 文本中的格式、数据进行处理。利用它可以很容易地对网页的内容进行分析、过滤和抓取。它的主要功能有文本信息抽取、链接提取、资源(图片、声音等)提取、链接检查和内容检验等。

本文采用 HTMLParser 分析网页所产生的 DOM 树作为待抽取的 DOM 树,部分代码如下:

```
Parser parser = newParser(c.getLink());
parser.setEncoding(c.getEncode());
for (NodeIterator e = parser.elements(); e.hasMoreNodes();)
{Node node = (Node) e.nextNode();
if (node instanceof Html)
{PageContext context = new PageContext(); context.setNumber(0);
context.setTextBuffer(new StringBuffer());
extractHtml(node, context, siteUrl);//抽取网页内容
……
```

3) 处理容器标签节点。自顶向下迭代读取 DOM 树中容器标签节点。考虑到实际页面中,正文整体所处的容器标签节点不会太深。迭代层数太多、读取层数太深会造成抽取标签粒度过细,丢失部分相关文本;而迭代层数太少又会导致粒度过粗,包含不必要的噪声信息过多。为此采用的迭代深度参数为 3 层。迭代中读取的每个容器标签节点,需要首先暂存于待处理列表,等待自底向上的递归方式得到所需的参数,再取出进行判断,选取出有用的节点予以保留。该步部分算法伪码如下:

```
Process(E)
If(E.isLeafNode)
String t = E.getText();
String b = E.getByte();
Float m = E.getModification ();
E.setAttribute({"text",t;  "byte",b;
"modification",m;  "isnoise",isnoise(E));
Else
Foreach Ei in E.childList
Process(Ei);
E.setChildAttribute(Ei);
String t = E.getText();
String b = E.getByte();
Float m = E.getModification ();
E.setAttribute({"text",t;  "byte",b;
"modification",m;  "isnoise",isnoise(E));
End If
```

4) 组合最终页面。组合所有标记为正文的节点,然后按照规定的模板输出文档并保存到硬盘上,这里仍采用标准 html 文件。

1.2.2 算法描述与实现

在分析处理和甄选容器标签节点时涉及到噪声节点的判别,即 isnoise()判别算法。在对大量网页进行统计分析时,可以发现一个共同点:节点的标签密度和节点的文本长度是 isnoise()判别算法的主要切入点,换句话说,大部分情况下,标签密度超过某阈值或文本长度未达到一定比例的节点都是噪声节点,应直接丢弃。具体步骤如下:

1) 筛选容器节点 E。

2) 对给定的容器节点,递归访问其子节点 E$i$ 获取标签信息和内容信息。

3) 调用 modify(E),由标签类型,标签属性等计算修正密度值 E_density。

4) 对于 E_density 参数满足小于阈值 $a$ 的内容块,判断文本长度是否达到给定阈值 $b$,满足标记为正文节点。

该判别算法的基本思路是考虑容器标签所包含文本长度与网页总文本长度之比、容器标签块内文本密度,并结合标签块内所含标签得出的启发性修正参数来进行判断。通常简单页面中内含文本长度最大的容器标签抽出即为正文信息,但这种方法在页面复杂,DOM 树层次多的情况下不太适合,去噪效果不理想,误判较多,此时结合容器标签内文本密度及修正参数时可以选出多个合适的容器标签,避免部分误判,达到较准确抽取正

文信息的效果。其中涉及到的 Modify(E)函数根据待判定节点 E 内所含有的页面标签返回加权值,其主要思路为 < p > < h > 等标签一般在正文节点中存在比率较高,而在噪声节点中较少存在。可以利用这种区分度来对原始计算结果进行加权,帮助区分这 2 类节点。选用的给定阈值 $a$,$b$ 由前期实验统计分析得出。

## 2 实验

以图 2 所示网页为例,其结构如图 3 所示。其中区域 1 标签为 div id = " main_header"为页面头部,主要包含 banner 网站广告和 menu 导航信息。区域 2 和 4 同属于标签 div id = "sideleft",内容包括正文和相关反馈信息,其中区域 2 为正文信息,标签为 div id = " news_main",区域 4 为反馈信息,标签为 div id = "guide"。区域 3 标签为 div id = "sideright",主要包含网站推送的链接和广告内容。区域 5 标签为 div id = "footer",主要包含网站的版权,导航内容。最终提取生成的页面仅包括网页正文信息,如图 4 所示。


图 2 实验网页

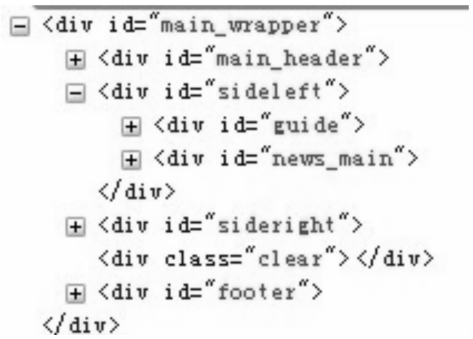


图 3 实验网页结构

图 4 最终提取生成的页面

为了检验算法的有效性,本文共采用 12 个站点的 860 个网页进行了测试。实验发现,网页噪声部分占了网页约 65% 的文本比例,因此,对 Web 网页进行分类或检索等其他应用时,如果不事先进行消除噪音,那么对于分类或检索的速度和精度将会有很大的影响。同时,本算法在 Windows XP 系统、core2(tm) duo CPU T5750 2.0GHz、2GB 内存的计算机上进行了实验。经过实验统计,对所有实验数据去噪的平均速度为 25 个/s。可见,该算法能较快速地完成网页噪音去除工作。在准确率方面,由于测试的网页较多,不可能对每个网页都进行准确率检查,因此随机抽取 100 个网页进行手工检查。现将检查结果用"优、良、中、差"4 个标准进行判断,其中:"优"代表网页主题内容正确提取,且噪音基本去除;"良"代表网页主题内容正确提取,噪音存在一部分;"中"代表网页主题内容基本能正确提取,噪音存在较多;"差"代表网页噪音基本没有消除或者主题内容没有正确提取。统计结果如表 1 所示。

表1 实验统计结果

| 检查类型 | 对应类型结果/% |
|---|---|
| 优 | 45 |
| 良 | 36 |
| 中 | 13 |
| 差 | 6 |

## 3 结束语

实验结果表明,该方法可以从主题型网页中提取出主题内容并清除噪音,速度较快,且清除噪音的准确率较高。本文的方法符合常见的网页设计风格,实验结果证明本文的方法是有效的。将其应用到搜索引擎方面,可以大大地减少索引量、提高搜索引擎的检索速度和检索的准确度;应用到分类方面,可以将 Web 网页中的主题内容提取出来,存放到文本文件中,然后就可以很方便地应用目前现有的分类器进行分类。但是,本文的方法对于 hub 型页面和论坛型页面的提取效果尚不够理想,同时算法中的阈值是在实验中得出的,其合理性还有待进一步实验和观察。因此,完善本文算法的不足、引入神经网络等机器学习方法来降低其对于人工干预的依赖,使其具有更强的通用性和适应性,将是下一步努力的方向。

## 参考文献:

[1] SODERLAND S. Learning information extraction rules for semi-structured and free text [J]. Journal of Machine Learning,1999,34(1):2332 - 2721.

[2] CHANG Chia hui,KAYED M,GI RGIS M R,et al. A survey Of Web information extraction systems [J]. IEEE Trans. on Knowledge and Data Engineering,2006,18(10):14112 - 14281.

[3] 杨少华,林海略,韩燕波. 针对模板生成网页的一种数据自动抽取方法 [J]. 软件学报,2008,19(2):2092 - 2231.

[4] Lin S H,Ho J M. Discovering Informative Content Blocks from web Documents[C]// Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data mining. [S. l. ]:[s. n. ],2002:588 - 593.

[5] Wong,W,Fu A W. Finding Structure and Characteristics of web Documents for Classification[C]// ACM SIGMOD Workshop on Researeh Issues in Data Mining and Knowledge Discovery. [S. l. ]:[s. n. ],2000:96 - 105.

[6] Embley D W,Jiang Y,Ng Y K. Record-boundary discovery in Web documents [C]//ACM SIGMOD Record. [S. l. ]:[s. n. ],1999:467 - 478.

[7] Chakrabarti S,Joshi M,Tawde V. Enhanced topic distillation using text,markup tags,and hyperlinks[C]//Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. [S. l. ]:[s. n. ],2001:208 - 216.

[8] 万乐, 左万利, 高金. 基于主题的网页噪音去除机制[J]. 计算机工程与设计,2008,29(8):2072 - 2074.

[9] Htmlparser [EB/OL]. [2010 - 03 - 09]. http://htmlparser. sourceforge. net/.

(责任编辑 刘 舸)