DOI:10.11721/cqnuj20150619

基于模式匹配的网页净化方法*

曾 蒸[1,2], 马 燕[2]
(1. 重庆师范大学 传媒学院/新媒体学院;
2. 重庆师范大学 计算机与信息科学学院, 重庆 401331)

摘要:新闻网页主要由大量文字描述构成,相比网页其他区域的噪音内容,其主题内容含有大段连贯的文字。根据这一特点提出一种基于模式匹配的网页净化方法,即在网页源代码中匹配最长文字字符串,从而准确定位主题内容源代码在网页源代码中位置,实现网页净化。本方法可去除来自不同网站网页的噪音内容,无需事先训练数据集来生成模板,不需要生成网页 DOM 树。对同构、异构和不符合 XML 规范的网页净化,试验证明效果理想且性能稳定。

关键词:网页噪音;网页净化;信息提取
中图分类号:TP391 **文献标志码**:A **文章编号**:1672-6693(2015)06-0103-07

网络资源大部分都是以网页形式存在。搜索引擎、Web 挖掘、知识发现等各种以网页作为数据源的 Web 应用越来越多,这些 Web 应用关注的主要对象是网页中的主题内容。但是网页中还有很多与主题无关的内容,如导航栏、广告连接、版权信息等,称之为噪音内容。噪音内容给基于主题内容的 Web 应用系统带来干扰,是妨碍应用和研究的脏数据。在 Web 应用之前消除页面的噪音内容,只保留主题内容,可有效提高应用和研究的效率。由此,形成了一个研究领域 Noisy Information Elimination,国内学者称其为"HTML 页面净化[1]"或者"Web 页面区域划分和主题区域搜索[2]"。过去的研究很多是基于单一网站的同构网页平台下实现[3],这些研究对于不同子集的网页采用不同的模板,模板是一个子集通过一组页面训练后生成的 DOM 树,这个树去掉了包含噪音内容的节点,仅保留主题内容对应的那些节点,把该子集中其他网页应用于这个模板,从而实现网页净化,这种方法复杂且一个子集只能对应一个模板,应用效率不高[4-6]。另外一些研究不需要事先生成模板,但仍然需要把网页转为 XML 可以解析的 HTML DOM,通过对 DOM 树的分析来找到主题内容区域从而实现页面净化,其实现效率仍然低下[1,7-11]。本文提出一种在网页源代码中双向快速收缩的网页噪音去除方法,无须模板,而且不需要对网页标签进行分析来转为 HTML DOM,对不符合 XML 规范的网页一样可以达到很好的净化效果,净化速度快,精度高。

1 相关工作

网页可以分成 3 种类别:hub 网页,称为骨架链接页面,其他页面的导入接口和分类器,一般作为入口页面;主题网页,文本信息,构成网站的主要信息;图片网页,基本以图片为主。本文选择主题网页来做净化。

根据功能的不同,文献[1]把 HTML tag 分为两类:一类是容器标签,它把一个网页分割,以不同的视觉内容展现到访问者面前,例如<body>,<table>,<tr>,<td>, 等;还有一类是描述标签,是内容的一个组成部分。比如<img>表示一副图片。

描述标签可以进一步划分为以下两类:

1) 格式标签。这类标签对一些文字、表格、图像等进行规范处理,用来改变这些内容元素的样式。常用的标签有<style>,<script>,<font>, ,<p>,以及一些特殊标签,主要由 Office 文档转换而来,如<o:p>,<u1:p>,<st1:csdate>等。其中<font>是改变其包含文字的大小、颜色、字体等样式。

* **收稿日期**:2015-06-17 **网络出版时间**:2015-9-28 12:02
资助项目:重庆市教育委员会高等教育教学改革项目(No. 143031)
作者简介:曾蒸,女,工程师,实验师,研究方向为计算机应用、新媒体技术,E-mail:akk310@qq.com
网络出版地址:http://www.cnki.net/kcms/detail/50.1165.n.20150928.1202.018.html

2) 元素标签。这类标签是从视觉上在网页上能实实在在看到的元素,有的独为一体,独自表示一个视觉单元元素,有的需要和其他元素配合使用。常用的标签有<a>,<object>,<img>等。其中<a>和一段文字配合表示一个超级链接,<img>表示一副图片。

容器标签把网页分成多个区域,是网页的骨架,而主题内容就位于其中某个区域中。如果要获取主题内容,只需要删除这段主题内容前后的其他数据,最后剩下的就是主题内容。下面通过一种方法来实现噪音内容的删除,并最终保留主题内容。

2 基于模式匹配的网页净化方法

通过对网页进行大量研究发现:去掉格式标签后的网页,其页面布局和内容没有任何变化,唯一的变化只是从颜色和字体等有所变化,对后续的 Web 应用没有任何影响;另外网页源代码中的各项注释对网页内容、版面和后续 Web 应用也不起任何作用,在对网页净化之前也可以去掉这些注释。

去掉格式标签后的网页的主题内容区域和其他噪音区域所包含的 HTML 源代码有如下区别:主题内容区域的源代码文字描述较多且分布集中,其中有许多大段连续的文字;而噪音区域的源代码文字相对较少且分散。

由此给出如下定义:

定义 1 去掉格式标签后的网页源代码中,最长的不包含标签的字符串必然位于主题内容的源代码中。

本净化方法分为 3 步:

1) 提取网页<body></body>容器标签内的源代码,删除源代码中所有格式标签和注释,得到一个没有格式标签的网页,我们称之为无风格网页 (Non-Style Page, NSP),其源代码为无风格网页源代码 (Source Code of Non-Style Page, SCNSP)。

2) 依据模式匹配原理,在 SCNSP 中遍历所有不含标签的文字子串,找到其中最长文字子串在 SCNSP 中位置。根据定义 1,其中最长的文字子串必然位于主题内容的源代码内。

3) 提取最长子串在 SCNSP 中的起始和结束位置,向前扩展到最近的起始容器标签,向后扩展到最近的结束容器标签。提取扩展后的字符串中的所有不成对的容器标签,比如单独的<table>, 或者</form>,在 SCNSP 中再次前后扩展,直到所有不成对的容器标签都找到对应的起始和结束容器标签。这时候的字符串就是主题内容源代码。

2.1 去除源代码格式标签

通过对网页源代码进行研究,有以下发现:

1) 视觉上看起来相对集中一段文字,其在网页源代码中有可能被很多格式标签分割,文字分布不均匀且分散,特别是 Office 文档转换来的网页,这种现象更为严重。图 1 是一个 Word 文档转换的网页和它的源代码,图中上半部分是网页从视觉上看起来的效果,文字相对集中且文字量大;图中下半部分是网页源代码,其中文字变得分散。

2) 网页源代码中的格式标签对于各种 Web 应用而言没有什么用处,例如 Web 数据挖掘中只关心网页中的文字和各种元素标签构成的元素,搜索引擎只关心文字和图像。

3) <body>标签以外的内容对后续 Web 应用没多大用处。因此,出于网页净化效果的考虑,完全可以事先就去掉网页的格式标签和注释,并只保留<body>标签中的内容,生成一个由容器标签、元素标签、文字和其它对象构成的网页。本文后面网页净化方法的描述都基于这种去掉格式标签后的网页实现。通过下面的正则表达式可以提取<body>中源代码,去掉其中的格式标签和注释:

算法 1 网页格式标签去除算法

输入:网页的源代码 SCP

输出:去掉格式标签和注释的网页源代码

1) // 正则表达式提取<body>并去掉其中的格式标签和注释。

2) SCP = Regex.Replace(SCP, @"[\s\S]*<body.*?>", "", RegexOptions.IgnoreCase);

3) SCP = Regex.Replace(SCP, @"</body>[\s\S]*", "", RegexOptions.IgnoreCase);

4) SCP = Regex.Replace(SCP, @"<script.*?>.*?</script>", "", RegexOptions.IgnoreCase | RegexOptions.Singleline);

5) SCP = Regex.Replace(SCP, @"<style.*?>.*?</style>", "", RegexOptions.IgnoreCase | RegexOptions.Singleline);

6) SCP = Regex.Replace(SCP, @"<'?(?:|FONT|SPAN|P|BR|CENTER|O…*|U1…*|ST1…*)[^>]*>", "", RegexOptions.IgnoreCase);

7) SCP = Regex.Replace(SCP, @"<![^>]*?>", "", RegexOptions.IgnoreCase);

8) return SCP;

算法描述:前面两行代码分别去掉<body>标签之前和之后的 HTML 源代码,第 3、4 行去掉<script>和<style>包含的 HTML 源代码,第 5 行去掉其他格式标签,但是格式标签包含的文字保留下来,第 6 行去掉 HTML 源代码中的注释,最后返回去掉格式标签、注释和仅保留<body>之后的 HTML 源代码。

2.2 基于模式匹配的源代码最长文字子串搜索算法

网页的主题内容区域所包含的文字相比其他区域较多,而且相对集中。要找到主题内容就必须把网页划分为多个区域,比对各个区域中文字的数量,数量多的文字密度就大。在网页的视觉感受上,很容易从语义上判断出哪个区域的文字最多且最集中,去掉格式标签后的源代码也从一定程度上反映了这个特点。视觉上文字集中部分,在去掉格式标签的源代码中文字相对集中。基于此,本文从无风格网页源代码 SCNSP 着手进行分析,首先分析文字和各种 HTML 标签之间的关系。文字存在于网页标签之中,文字被这些标签隔开,从而形成分散的文字区域。主题内容区域中的描述标签则较少,里面的文字一般都按一定数量的方式聚集在一起,形成一个大的文字区域。

图 1　网页及其源代码

图 2 是一个主题文字网页的 SCNSP,由于原始 SCNSP 太长,为了能在一幅图中显示出来,对该网页进行了大量精简。图 2 中虚线多边形框起来的各个区域就是无标签文字子串。其中最长的无标签文字子串出现在主题内容的源代码中。从图中可以看出,所有文字子串都是以字符“>”开始,字符“<”结束,也就是说每个字符对(“>”,“<”)包含一段文字子串。对于字符对(“>”,“<”)中没有文字的,可以假设为其包含的文字子串长度为零。字符“>”和字符“<”在 SCNSP 中的数量相等。

图 2　提取 SCNSP 中的无标签文字子串

算法 2　计算某个特殊字符在 SCNSP 中的位置 iPos(C)

输入:无风格网页源代码 S,和特殊字符 C

输出:某个特殊字符在 SCNSP 中的位置,整形数组 iPosArray

1) ArrayList iPosArray = new ArrayList();　//数组用于存储位置数据

2) int index = 0;　//初始化位置为 0

3) foreach(Char ch in S){　//遍历 SCNSP 中所有字符

4) if (ch == C){　//当某个字符为特殊字符时,加入数组中

5) iPosArray)Add(index);

6) }

7) index++;

8) }

9) return iPosArray; //返回结果

算法2描述:算法中第3行的S代表一个网页的SCNSP,第4行的C代表特殊字符。通过计算返回特殊字符C在SCNSP中出现的所有位置$iPos(C)$。$iPos(C)$为整形一维数组。

算法3 计算文字子串在SCNSP中起始和结束位置以及子串长度 sPos(iPos('>'), iPos('<'))

输入:特殊字符">"和"<"在SCNSP中位置,算法2返回结果 iPos('>')和 iPos('<')

输出:所有文字子串 sPosArray

1) ArrayList sPosArray = new ArrayList();

2) Array[] sPosA = new Array[3]; //三元数组用来存储某对标签('>','<')的位置和包含文字长度

3) ArrayList iPos1 = iPos('>');

4) ArrayList iPos2 = iPos('<');

5) int j = iPos1)Count;

6) for (int i = 0; i < j-1; i++){ //遍历,完成 sPosA 的计算并加入 sPosArray 中

7) sPosA[0] = Convert. ToInt32(iPos1[i]);

8) sPosA[1] = Convert. ToInt32(iPos2[i + 1]);

9) sPosA[2] = Convert. ToInt32(iPos2[i + 1]) -Convert. ToInt32(iPos1[i]-1);

10) sPosArray. Add(sPosA);

11) }

12) return sPosArray; //返回结果

算法3描述:算法2中计算生成的特殊字符">"和"<"在SCNSP中的位置作为输入参数,计算SCNSP中所有无标签文字子串的起始和结束位置以及子串长度。通过算法2的计算得到的特殊字符">"的输出$iPos('>')=\{X1,X2,X3,\cdots,Xn\}$,特殊字符"<"的输出$iPos('<')=\{Y1,Y2,Y3,\cdots,Yn\}$。因为SCNSP中所有标签以为"<"开始,">"结尾,且成对出现,所以必然有以下规律:$Y1<X1<Y2<X2<Y3<X3<\cdots<Yn<Xn$。那么第一个(">","<")通过算法3的计算得到的$sPosA=\{X1, Y2, Y2-X1-1\}$,所有(">","<")对通过算法3得到的$sPosArray=\{\{X1, Y2, Y2-X1-1\}, \{X2, Y3, Y3-X2-1\},\cdots, \{Xn-1, Yn, Yn-Xn-1-1\}\}$。

算法4 计算最长文字子串在SCNSP中的起始结束位置和长度 ssPosA

输入:算法3的返回结果 sPos

输出: 最长文字子串的三元数组 ssPosArray(起始位置,结束位置,子串长度)

1) Array ssPosArray = { 0, 0, 0 }; // 初始化一个文字子串 ssPosArray

2) foreach (Array sposa in sPos){ //遍历所有文字子串

3) if (sposa[2] > ssPosArray [2]){ //当某个子串的长度大于 ssPosArray 时,该子串为之前所有子串中最长的

4) ssPosArray = sposa;

5) }

6) }

7) return ssPosArray; //返回结果

算法4描述:经过算法3计算得到的所有文字子串的数组作为输入参数,刚开始初始化一个子串 ssPosArray 作为最长子串,遍历 sPos,当某个子串的长度大于 ssPosArray 时,把该子串赋值给 ssPosArray,如此循环计算,最后得到的 ssPosArray 为所有子串中最长的。

通过算法4得到的最长文字子串只是位于主题内容源代码中的一段,还不是主题内容源代码的全部。要获取所有主题内容源代码,还要经过下面的扩展计算才能得到。

2.3 扩展最长文字子串,得到主题内容源代码

这时候得到的最长文字子串是一段网页源代码,其并不能表示成一个完整的网页,这段源代码可能只是网页主题内容源代码的一部分。这时候需要丰富这段源代码的内容。

所有容器标签都是由两部分构成,一个是起始容器标签,一个是结束容器标签。如＜div＞和＜table＞就是起始容器标签,其对应的结束容器表示是＜/div＞和＜/table＞。一对起始和结束标签之间的内容就是这个容器标签包含的内容。图 3 左上图所示一个网页源代码的最长文字子串,其内容全部由文字和标点构成,不包含 HTML 标签。找到这段文字子串在 SCNSP 的位置,向前搜索最近的一个起始容器标签,向后搜索最近的一个结束容器标签,这两个标签并不一定是一对容器标签,有可能分属于两对容器标签,扩展后的主题内容源代码如图 3 右上图所示。提取扩展后主题内容源代码中不成对的容器标签,把这些标签分为起始容器标签和结束容器标签两类。在 SCNSP 中向后搜索所有起始标签对应的结束标签,向前搜索所有结束标签对应的起始标签,从而前后扩展主题内容源代码,最后得到的就是需要的主题内容源代码(Source code of topic content, SCTC)。SCTC 表示的网页就是净化后的页面。

图 3　两次扩展后的主题内容源代码 SCTC

3 实验评估

通过 C＃语言实现算法,从国内的几大网上新闻提供商获取网页。由于算法中不需要使用者事先定义任何参数,所以本方法使用起来非常方便。主题文字网页来自 Sohu 等多家大型新闻提供商的官方网站。网页规模 15 000 篇文档,为了体现无模板网页净化方法的优势,把这 15 000 文档打乱混合后进行处理。最后按照网站对净化后的网页分类进行评估,得到各个网站净化效果。净化效果从下面两项数据来评估:

1) 各个网站正确净化网页百分比。这部分网页被完全净化并保留了所有需要的主题内容。

2) 各个网站基本净化网页百分比。这部分网页净化效果不是很理想,但主题内容基本被保留下来,可能有部分净化后网页的主题内容不完整,或者含有少量噪音内容。

结果如表 1 所示。

表 1　净化实验结果

来源网站	网页数/个	正确净化率/%	基本净化率/%
新浪新闻中心	3 000	89	3
网易新闻	3 000	90	0
搜狐新闻	3 000	87	1
太平洋网	3 000	79	4
CSDN	3 000	79	8

从实验结果上看,基本分为下面 3 种情况:当网页主题内容的文字描述太少,会导致本方法把主题内容去掉,而误把超链接区的内容作为主题内容;当网页的主题内容的文字描述较多,且 SCNSP 中最长文字子串不是位于主题内容源代码中的某个表格中,本方法的成功率高;当网页主题内容内包含表格数据,且网页的 SCNSP 中最长文字子串位于表格中时,净化的网页会只保留表格数据而删除表格以外的数据,从而导致网页的主题内容不完整。因为主题文字网页是以文字描述为主,主题内容中包含许多大段连续文字,所以实验结果的第一种情况出现不多;因为主题内容中的表格包含的都是一些短小的文字描述,很少有大段文字,所以第三种情况也很少出现。大量实验结果都集中在第二种情况,所以本方法在主题文字网页净化方面的成功率是很高的。

4 结语

本文提出了一种网页净化算法,使用该算法的方法利用主题内容的文字集中且字数远远大于噪音区域文字

字数的特点来实现网页净化。实验结果证明净化效果良好,且因为不受模板限制,其应用范围广泛,可用于混合网站网页页面净化。和其他基于 DOM 标签数的网页净化技术相比,本方法计算时间成本上远远优于它们。对不符合 XML 规范的网页实现净化效果也是本方法的重要亮点。但是因为本方法基于主题内容文字字数最多的假设,所以对一些主题内容区域文字描述较少的网页,其净化效果不理想,这需要进一步完善算法,使其具有更强的适用性。

参考文献:

[1] 张志刚,陈静,李晓明. 一种 HTML 网页净化方法[J]. 情报学报,2004,23(4):387-393.
Zhang Z G,Chen J,Li X M. An approach to reduce noise in HTML pages[J]. Journal of The China Society for Scientific and Technical Information,2004,23(4):387-393.

[2] 胡飞. 基于标记树的 Web 页面区域划分和搜索方法[J]. 计算机科学,2005,8:182-185.
Hu F. How to get the main part of Web pages[J]. Computer Science,2005,8:182-185.

[3] 毛先领,何靖,闫宏飞. 网页去噪音:研究综述[J]. 计算机研究与发展,2010,47(12):2025-2036.
Mao X L,He J,Yan H F. A survey of Web page cleaning research[J]. Journal of Computer Research and Development,2010,47(12):2025-2036.

[4] Gibson D,Punera K,Tomkins A. The volume and evolution of Web page templates[C]//Proc of the 14th int conf on World Wide Web. New York:ACM,2005:830-839.

[5] Yi L, Liu B,Li X. Eliminating noisy information in Web pages for data mining[C]//Proc of the 9th ACM SIGKDD int conf on knowledge discovery and data mining. New York:ACM,2003:296-305.

[6] Yi L,Liu B. Web page cleaning for Web mining through feature weighting[C]//Proc of the 18th int joint conf on artificial intelligence (IJCAI-03). San Francisco: Morgan Kaufmann,2003:43-50.

[7] Cai D,Yu S,Wen J R,et al. Extracting content structure for Web pages based on visual representation[C]//Web technologies and applications:5th Asia-Pacific Web conf. Berlin:Springer,2003:406-417.

[8] Song R,Liu H,Wen J R,et al. Learning block importance models for Web pages[C]//Proc of the 13th int conf on World Wide Web. New York:ACM,2004:211-220.

[9] Cai D,Yu S,Wen J R,et al. VIPS:A vision based page segmentation algorithm, MSR-TR-2003-79[R/OL]//(2003-11)[2009-02-01]. http://research. microsoft. com/apps/pubs/default. aspx? id=70027.

[10] Yu S,Cai D,Wen J R,et al. Improving pseudo-relevance feedback in Web information retrieval using Web page segmentation[C]. Proc of the 12th World Wide Web conf. New York:ACM,2003.

[11] 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.
Wan L,Zuo W L,Gao J. Web pages noise removal based on focused topic[J]. Computer Engineering and Design,2008,29(8):2072-2074.

Eliminating Noisy Information in Web Pages Based on Pattern Matching

ZENG Zheng[1,2], MA Yan[2]

(1. Media College/New Media College, Chongqing Normal University;
2. College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China)

Abstract: A piece of news web page has a lot of words. Compared with the noisy content, the topic content contains larger sections of coherent text. According to this feature, we propose an approach to purify pages based on pattern matching. This searches a page's source code for the location of the longest text string. The longest text string belongs to the topic content. That means we get the location of the topic content. It is a simple and fast approach for isomorphic pages, non-isomorphic pages and pages not meeting XML specification. The test proves it is satisfactory and stable.

Key words: web page noise; web page purification; information extraction

(责任编辑 游中胜)