基于文本及符号密度的网页正文提取方法

洪鸿辉,丁世涛,黄 傲,郭致远
(武汉邮电科学研究院 湖北 武汉 430000)

摘要: 大多数的网站的网页除了主要的内容,还包含导航栏,广告,版权等无关信息。这些额外的内容亦被称为噪声,通常与主题无关。由于这些噪声会妨碍搜索引擎对Web数据的挖掘性能,所以需要过滤噪声。在本文中,我们提出基于网页文本密度与符号密度对网页进行正文内容提取,这是一种快速,准确通用的网页提取算法,而且还可以保留原始结构。通过与现有的一些算法对比,可以体现该算法的精确度,同时该算法可以较好的支持大数据量网页正文提取操作。

关键词: 文本密度; 算法; 噪音; 正文提取
中图分类号: TP391 **文献标识码:** A **文章编号:** 1674-6236(2019)08-0133-05

Text extraction method based on text and symbol density

DOI:10.14022/j.cnki.dzsjgc.2019.08.029

HONG Hong-hui, DING Shi-tao, HUANG Ao, GUO Zhi-yuan
(*Wuhan Research Institute of Posts and Telecommunications*, *Wuhan* 430000, *China*)

Abstract: Most web pages contain not only the main content, but also navigation bar, advertising, copyright and other irrelevant information. These extra contents are also referred to as noise, usually irrelevant to the topic. Since these noises will hamper the performance of search engine for Web data mining, noise removal is needed. In this paper, we propose a fast, accurate and general web content extraction algorithm based on text density and symbol density, which can preserve the original structure. Compared with some existing algorithms, the algorithm can reflect the accuracy of the algorithm, and the algorithm can better support the large amount of data Web page text extraction operation.

Key words: text density; algorithm; noise; text extract

自互联网问世以来,经过多年的发展,互联网站点的数量在不断的增长,互联网上的信息也在不断的增加,然而,由于商业因素的问题,这些网站在为我们提供有价值的信息的同时,还会包含其他信息,例如广告或其他网站的链接。链接可能是图片,文字。这些相对于正文内容无用的信息会降低我们的阅读效率,而且这些无用的文字可能会被搜索引擎作为索引关键词,不仅降低了搜索的效率还影响了用户的体验。

很多互联网公司也发现了这一问题,所以现在越来越多的网页都会支持RSS。若一个网页支持*RSS*,我们就可以很轻易的提取网页的正文内容,但大多数网页还是不支持*RSS*,所以关于正文提取这一方面的研究工作一直没有停止。网页的类型有很多种,比如新闻网站,博客网站,论坛等。新闻类网站的正文提取一直是研究的主要方向,新闻类的文章通常要提取正文内容,标题,时间,作者等。文章通常要提取正文内容,标题,时间,作者等。一方面,网页正文提取结果的好坏会影响着文本聚类,去重,语义指纹等结果。另一方面,网页正文提取在大数据时代也是一项不可或缺的环节。

1 相关工作

1.1 VIPS

2003年,微软公司亚洲研究所提出了一种网页进行视觉分块[1]算法—VIPS[2]算法。该算法的思想是模仿人类看网页的动作,基于网页视觉内容结构信息结合Dom树对网页进行处理。简单的说就是把页面切割不同大小的块,在每一块中又根据块网页的内容和CSS的样式渲染成的视觉特征把其分成小块,最后建立一棵树[3]。

但是,VIPS必须完全渲染一个页面才能对其进

收稿日期: 2018-07-20 **稿件编号:** 201807113
作者简介: 洪鸿辉(1992—),男,广东揭阳人,硕士研究生。研究方向:大数据处理。

行分析。这就导致VIPS算法占用的内存资源以及CPU运算资源较多。由于该算法在提取一个网页时消耗的资源过多，所以这种网页提取方法在面对海量网页处理时并不适用。

1.2 基于块分布网页正文提取

该方法由哈尔滨工业大学的陈鑫提出。该方法大致过程如下：

1）将网页中的html标签全部去掉，再去掉空白行和空白部分，得到文本。

2）将文本的行按照一定的数量分成一个一个的文本块。

3）对这些块进行分析，找出骤升和骤降的块，最后分析取出骤升和骤降块之间的内容。

这个算法代码不到100行，可以在O(n)的时间复杂度内提取出网页正文，但是该方法的准确率在95%左右，而且无法保留原有的html标签。这对于一些特殊场景并不适用。

1.3 Readability

网页提取中，应用最广泛的就是Readability，该算法需要解析DOM树，因此时间复杂度和空间复杂度较高。

在使用过程中，发现该算法有很多种语言实现，虽然使用方便，但是提取的网页正文的时间比较长。

1.4 基于网页模板的抽取算法

总的来说，基于模板的网页抽取算法是通过移除所有输入的网页中相同的部分。通过URL判断出所输入的网页是否有相同的结构。页面中相同的部分为非正文，页面间相差较大的是正文[4-5]。

例如，在很多网页中，导航栏，页眉，页脚等都是一样的，这些就不是正文。这是一种较为精确的方法，但这种方法需要对每个网站进行建模，所以也是很繁重的一种算法。而且每次网站改版以后，原有的模板就不能用了，必须重新生成一次模板。所以如果不是有针对的对某些网站进行爬取的话，这种方法并不是很好[6]。

2 网页正文提取设计

2.1 系统设计

系统功能：本系统输入网页文件或者URL即可提取出网页正文，系统框架如图1所示。

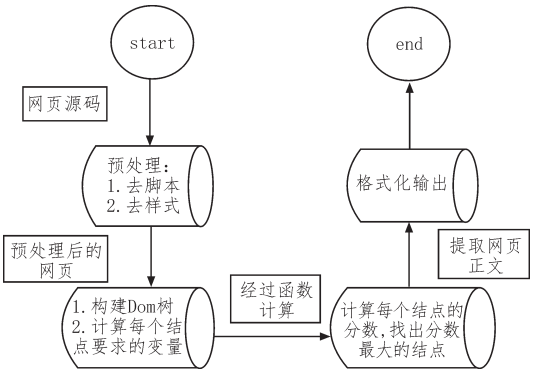

图1 系统框架

正文提取的内容结构如图2所示，首先是网址，然后是标题，接着是时间，内容，最后是网页源码。通过文本提取后，我们可以得到正文的主要内容以及逻辑结构，保留的网页源码是为了保留其完整的文本结构样式。

```
网址：http://tech.ifeng.com/a/20180902/45148684_0.shtml
标题：阿里病逝员工妻子：起诉自如已获立案 9月27日开庭
时间：2018年09月02日 10:40:22
内容：原标题：阿里病逝员工妻子：起诉自如已获立案 9月27日开庭 阿里病逝员工妻子：起诉自如已获法院立案，9月27日开庭“阿里员工得白血病身故，生前租住自如房被指甲醛超标”一事，再次令公众舆论聚焦自如房屋的空气质量问题。 刚入职阿里不久的王先生租自如的房子2个多月后，便在首都医科大学附属北京朝阳医院被确诊为急性髓系白血病并过世。可资对比的是，王先生半年前入职阿里时的体检报告并无明显异常。
网页源码：
<div id="main_content" class="js_selection_area">
 <!--mainContent begin-->
 <p>原标题：阿里病逝员工妻子：起诉自如已获立案 9月27日开庭</p>
 <p><strong>阿里病逝员工妻子：起诉自如已获法院立案，9月27日开庭</strong></p>
 <p>“阿里员工得白血病身故，生前租住自如房被指甲醛超标”一事，再次令公众舆论聚焦自
```

图2 输出结果

2.2 基于文本密度提取

早期有一些算法比如MSS算法，程序把html网页转换为序列，然后打分，比如一个标签给负分，一个内容文本给正分。通过打分，我们可以获取一个序列，然后我们取出token序列中分数最大的子序列[7]。但是在实验中发现其提取正文的准确度并不是很好。但是这给我们提供了一个思路。

根据笔者的工作经验以及对大量不同的网站进行等结构分析可以发现，即使不同的人编写不同的网页，但不同的网页时有一定的共性[8]：

1）正文内容一般在网页的<body>标签内的<p>标签中；

2）<p>标签一般都为 、<td>等标签的子标签，并不独立存在。

同时，在分析的过程中也发现了一些问题：

1）各个网站对标签的class，id等属性名称命名不统一。

2）网页正文内容中可能有超链接，其可能是文字也可能是图片，标签<p>中不一定就是内容，还可能还会其他标签。

3) 标签之间可能还会有JS脚本或者是其他的html标签。

4) 标签的内容不全都是或不一定是正文内容等。网站的版权信息一般也是 标签的内容，但这部分内容属于噪声。

通过文献[9-10]。，我们知道可以用每行的文本密度来判断这一行属不属于正文。我们把网页解析成Dom树，然后判断每个节点属于正文内容的可能性是多少。

每个网页都可以被解析成一颗Dom树，所有的标签都是节点，而文字和图片等都是叶子节点。

例如以下是html代码示例：

```
1. <div class="box">
2.  <h1 class="title">article</h1>
3.  <div id="content">
4.    <p> hello world</p>
5.  </div>
6.  <div class="foot">
7.    <a> pre</a>
8.  <a> next </a>
9.  </div>
10. </div>
```

该html代码解析成Dom树如图3所示，从树中我们可以很清楚地看出各个节点之间的关系以及各个层次之间的联系。

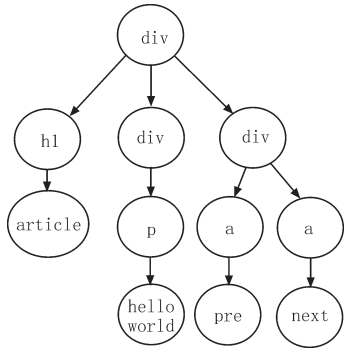

图3　Dom树

根据以前编写网页的经验来看，一个网页中，正文标题通常会用<h*></h>(*:1-6)标签包含，而正文通常会使用 标签包含。有可能在 标签中还会包含<a>链接或者标签等，但是这些都没关系，我们只需要找到包含正文内容的 既可以，无论 标签内会包含什么标签，我们都视为正文内容[11]。

可以通过Jsoup提供的标准库中的Parse函数把本地的网页文件解析，返回Document对象，即Dom树。通过该Dom树，我们可以很轻松的得出以下变量。而且我们还可以的计算出每个点字符串子树和标签数的比率。

定义一：如果i为Dom树的一个结点，那么该节点的文本密度TD_i为：

$$TD_i = \frac{T_i - LT_i}{TG_i - LTG_i} \tag{1}$$

T_i：结点i的字符串字数。

LT_i：结点i的带链接的字符串字数。

TG_i：结点i的标签数。

LTG_i：结点i带链接的标签数。

核心程序：

```
Algorithm 1 computeInfo (node)
hashMap<Node, NodeInfo> map;
computeInfo(node){
      if(node is Element){
      NodeInfonodeInfo;
      for(childNode in node){
nodeInfo = computeInfo(childNode);
nodeInfo.T+=childNode.T;
nodeInfo.LT+=childNode.LT;
nodeInfo,TG+=childNode.TG;
      nodeInfo.LTG+=childNode.LTG;
     }
if(node is a){
      nodeInfo.LTG_i ++;
nodeInfo.txt+=node.txtLength;
}else if(node is p){
      nodeInfo.PNum++;
}
map.put(node,nodeInfo);
}
else if(node is TextNode){
nodeInfo.T=node.txtLen-node.LT;
}
}
```

TD_i是衡量一个网页的每个结点文本密度，如果一个结点的纯文本字数比带链接的文本字数明显多很多的时候，根据公式(1)，我们可以判定该结点的文本密度就会很大，从而就可以很容易判断该节点

是不是正文的一部分。

在计算文本密度之前，应对网页进行预处理。应该先移除JavaScript脚本，CSS样式，IFrame等。因为这些信息对构建Dom树并没有什么帮助，而且还可能破坏我们的结果，根据W3C的标准，我们可以得知，网页的内容主体在body标签中，所以我们取body标签作为根节点构建Dom树。构建的Dom树与文本密度关系如图4所示。

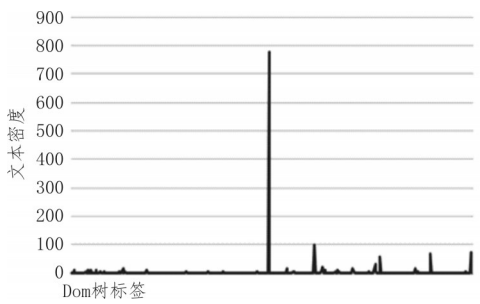

图4　Dom树每个节点的文本密度

TD_i 作为一个网页中结点的衡量文本密度的值，如果某 TD_i 的值很大的话，就意味着这个结点的无格式无链接的文字一定比有格式有链接的文字要多，那么证明这个结点属于正文内容的可能性。这让我们可以很清楚的判断哪个结点是对于我们有用的正文内容。

但是如果只是简单根据这个文本密度就可以得出文本内容位于Dom树的那个结点中，那么网页正文提取这个问题也不会一直困扰着大家。因为繁多的网页采用的布局各不相同，所以如果想要一个算法可以通用提取不同的网页，我们需要考虑的因素还有很多，于是我们建立了一个数学模型，该公式为

$$\text{sc}\,ore=\log(SD)*ND_i*\log 10(PNum_i+2) \quad (2)$$

SD：节点文本密度的标准差。 ND_i ：节点i的文本密度。 $PNum_i$ ：节点i的p标签数；

我们根据这个评分对一个网页进行提取结果如图5所示。

从图4中我们可以看出两个Dom树标签文本密度相差最小的是700，而经过公式（2）过滤后我们可以得到从图5中看出Dom树标签分数相差最小的是1500。

然后我们对网页尝试进行提取，把提取的正文和我们人为提取的网页进行比对，看提取的效果如何。从Marek M等人的文献[12]中可以得知有开源的数据库用于评判正文提取的效果，比如CLEANEVAL。

CLEANEVAL是一个关于清理任意网页的共享

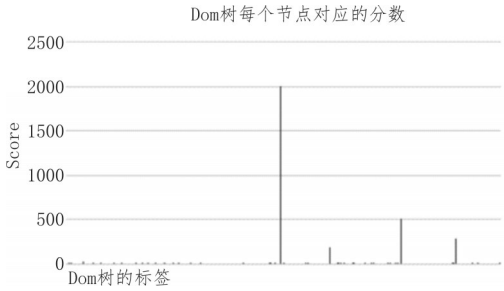

图5　Dom树每个节点对应的分数

任务和竞争性评估，其目标是准备网页数据作为语料库，用于语言和语言技术的研究和开发。该数据包含源网页以及标准的经过提取的正文内容。

本来是用它提供的一个用Perl语言写的软件用来评估本文的算法提取的正文和标准库，但是在测试中，发现其耗时太长，改为多线程执行以后也依旧不理想。所以本文摒弃了该软件，采用了查准率R和查全率P来评估算法的可行性。

$$R=\frac{LCS(a,b).length}{a.length},P=\frac{LCS(a,b).length}{b.length} \quad (3)$$

a:提取的正文内容

b:标准的正文内容。

LCS（Longest Common Subsequence）：最长公共子序列[13]。

除了CLEANEVAL这个数据源，我们使用爬虫技术下载了凤凰网，参考消息，新浪新闻网页。每个网站下载1 500个网页，同时，对每个网页使用正则匹配把人眼看到的正文内容保存下来，然后同样使用LCS对本文的正文提取算法进行判断。网页正文提取结果如表1所示。

表1　网页提取效果

网站	P	R	Score
cleanEval-Eng	92.68%	75.21 %	71.23%
cleanEval-Zh	80.50%	68.29%	60.09%
凤凰新闻	96.71%	97.56%	94.44%
参考消息	98.12%	99.68%	98.66%
新浪新闻	97.68%	98.56%	98.14%

2.3　基于文本密度与符号密度提取

根据表1可以得知这样的模型在提取文本的时候已经有足够好的表现，但希望还能进一步提高正文提取的正确率。在研究了很多网页以后发现正文中基本上都会有标点符号[14]，而网页链接，广告信息由于文字少通常是没有标点符号的，假设 SbD_i 为一段文字的符号密度。

定义二：

$$SbD_i = \frac{T_i - LT_i}{Sb_i + 1} \quad (4)$$

Sb_i:符号数量。

符号密度为文字数量与符号数量的比值，根据我们的经验，通常正文的 SbD_i 会比非正文要大。非正文可能没有符号，而且由于非正文通常会比较少字，可能就是一些导航的信息之类的，所以，在相同字数下它的 SbD_i 相对正文来说就会比较小。于是我们修改 Score 函数(2)，经过多组数据测试修正为函数(5)。使用函数(5)作为核心函数提取网页正文内容得到表格 2。

$$score = \log(SD) * ND_i * \log 10(PNum_i + 2) * \log(SbD_i) \quad (5)$$

如表 2 所示，对于凤凰网等国内的新闻网站，添加了标点符号因素后，其提取效果比原来的更加好，可以接近 99%。这样已经达到预期标准，但是还剩下 1%不知道在哪里出了问题。所以我们随机抽取了几篇提取出来的正文对相应的网页进行匹配，查看效果如何。

实验结果发现通过算法提取的网页正文内容与标准的正文内容完全一样，只是空格的数量有所不一样，但是正文的文字所差无几，这也就解释了为什么还有 1%的误差。所以该算法对于国内的新闻网站的正文提取还是非常优秀的。

表 2　文本与符号密度的网页提取

网站	P	R	Score
cleanEval-Eng	93.88%	77.43 %	73.11%
cleanEval-Zh	81.62%	69.18%	62.16%
凤凰网新闻	97.51%	98.18%	95.76%
参考消息	98.80%	99.88%	98.68%

3　结束语

针对新闻网站，在原有的根据文本密度提取网页正文内容的算法的思想基础上进行改进，得到了超过高的 P 值，从而实现了对新闻网站精准快速提取正文内容的目的。

参考文献:

[1] Song R. Learning block importance models for web pages[C]// International Conference on World Wide Web. ACM, 2004:203-211.

[2] 吕芳. 基于视觉特征的钓鱼网页相似性计算技术研究[D].哈尔滨:哈尔滨工业大学,2015.

[3] 沈怡涛. 基于视觉特征和文本结构分析的中文网页自动摘要技术研究[D].上海:华东师范大学,2014.

[4] Bar-Yossef Z, Rajagopalan S. Template detection via data mining and its applications[C]// International Conference on World Wide Web. ACM, 2002:580-591.

[5] 杨一柳. 基于模板的网页信息抽取技术研究[J].渤海大学学报(自然科学版), 2013(3):320-322.

[6] 顾韵华,高原,高宝,等.基于模板和领域本体的Deep Web信息抽取研究[J].计算机工程与设计,2014,35(1):327-332.

[7] Zachariasova M, Kamencay P, Hudec R, et al. A novel imaging approach of web documents based on semantic inclusion of textual and non-textual information[J]. AasriProcedia,2014,9(D6):31-36.

[8] 张奇,郝志峰,温雯,等.基于互信息度量的Web信息抽取[J].计算机应用与软件,2013,30(12):15-18.

[9] Sun F, Song D, Liao L. DOM based content extraction via text density[C]// International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2011:245-254.

[10]吴共庆,刘鹏程,胡骏,等.基于块密度加权标签路径特征的Web新闻在线抽取[J].中国科学:信息科学,2017,47(8):1078-1094.

[11]刘利,戴齐,尹红风,等.基于多特征融合的网页正文信息抽取[J].计算机应用与软件,2014,31(7):47-49,77.

[12]Marek M, Pecina P, Spousta M. Web Page Cleaning with Conditional Random Fields[C]// Web As A Corpus Workshop, Incorporating Cleaneval. 2007.

[13]王永新,王秋芬,梁道雷.一种高效LCS算法[J].南阳理工学院学报,2013(6):67-70.

[14]ErdinçUzun, HayriVolkanAgun, TarıkYerlikaya. A hybrid approach for extracting informative content from web pages[J]. Information Processing and Management,2013,49(4):928-944.

[15]朱泽德,李淼,张健,等.基于文本密度模型的Web正文抽取[J].模式识别与人工智能,2013(7):61-66.

[16]张乃洲,曹薇,李石君.一种基于节点密度分割和标签传播的Web页面挖掘方法[J].计算机学报,2015,38(2):349-364.