基于布局相似性的网页正文内容提取研究*

杨柳青[1,2]，李晓东[2]，耿光刚[2]
（1. 中国科学院计算机网络信息中心，北京 100190；2. 中国互联网络信息中心，北京 100190）

摘　要：合理的网页正文提取技术可以将海量互联网数据中冗余的、重复的、无用的信息去除，获取更加有实际意义和价值的数据。经过对网页的观察，发现同一网站下的网页具有在内容布局和样式结构上非常相似的特点，提出并实现了一种基于布局相似性的网页正文提取方法，即通过比对来自同一网站同一专题的网页 DOM 树中节点数据信息的相似性来实现正文提取，并对相关问题进行了尝试性的研究和实现。实验证明该方法思路简单、实用性强、普适性好，在满足较高准确率的同时，能为众多互联网内容分析应用提供支撑。

关键词：布局相似性；网页正文提取；信息检索
中图分类号：TP391.1　　**文献标志码**：A　　**文章编号**：1001-3695(2015)09-2581-06
doi:10.3969/j.issn.1001-3695.2015.09.005

Study of Web pages content extraction based on layout similarity

Yang Liuqing[1,2], Li Xiaodong[2], Geng Guanggang[2]
(1. *Computer Nework Information Center, University of Chinese Academy of Sciences, Beijing* 100190, *China*; 2. *China Internet Network Information Center, Beijing* 100190, *China*)

Abstract: Appropriate Web content extraction technique can remove the data which is redundant, repetitive and useless from massive Web pages while extracting more meaningful and more useful data. Through the observation of Web pages, this paper proposed and implemented a Web content extraction method based on the layout similarity that the pages under the same Web site showed similar in content layout and style structure. It achieves the purpose of main content extraction by comparing the similarity of the DOM node structure data from the Web pages belong to the same topic of the same sites. It also did some tentative research and implementation on some other content relevent to this content extraction method. Experiments prove that this method is simple, pratical and universal, and it can not only meet the requirement of both high accuracy but also provide support for more Internet applications of content analysis.

Key words: layout similarity; Web page content extract; information retrieval

随着互联网站点数量的不断增长以及搜索引擎技术的不断发展，互联网成为了人们获取信息的一个不可或缺的渠道。而在商业运作的因素下，为用户提供原始信息的网站，会在其包含有价值数据的网页中提供一些额外的信息，如广告数据以及对其他站点相关内容的链接（这些广告、链接数据可能是文本，也可能是图片，甚至可能是插件），甚至搜索引擎也会在用户的检索结果中加入一些广告作商业推广。这些广告、链接等数据的不断加入使得本来应该很精简的页面外观变得繁琐；各类的网页制作工具以及各种动态元素的加入也使得页面的内在结构变得复杂。网页内容与结构的日趋复杂在影响用户阅读体验的同时，也为众多 Web 内容处理应用，如搜索引擎、网络归档等技术带来了挑战。另外，移动互联网的蓬勃发展使得在移动端浏览网页成为大势所趋，而移动端所具有的屏幕小、流量受限等特点，使得网页核心内容的有效提取变得更为迫切。

网页正文提取技术作为信息抽取技术中的一个外延分支，是一项相对比较基础的工作，在很多技术应用上都有着对它的需求，如网页内容索引建立、网页聚类、网页内容聚合等，对于数据挖掘和垂直搜索等也都有着重要意义。它在完成网页内容提取和结构精简的基础上，可以有效支撑搜索引擎、网页内容归档以及移动端等的相关应用技术的发展。在这样的研究背景下，网页正文提取技术具有非常重要的应用研究意义。

1　相关工作

目前对于网页正文提取的算法主要分为四种：

a）基于启发式规则的提取算法。这类算法的主要特点是基于样本网页文件的 DOM 树分析，归纳总结出 DOM 树中正文和非正文的节点特征以制定相应的启发式规则，通过这些规则的相互配合来达到去除非正文内容、提取正文内容的目的。Gupta 等人[1]使用启发式规则来构造过滤器，以对 DOM 树中的无用节点进行过滤删除，对于广告的过滤使用的是黑名单策略。对于层出不穷的广告来说，这种方法的效率并不是特别地好。Guo 等人[2]以标点符号为特征，总结启发式规则来寻找正文片段并将其合并为正文。事实上，启发式规则的总结和适用性都存在着很大的局限性。

收稿日期：2014-06-05；**修回日期**：2014-08-01　　**基金项目**：国家自然科学基金面上项目(61375039)；国家自然科学基金青年资助项目(61005029)；中国科学院计算机网络信息中心“一三五”规划重点培育方向专项基金资助项目(CNIC_PY_1402)

作者简介：杨柳青(1990-)，男，青海人，硕士，主要研究方向为网络应用、网络安全、下一代互联网(yangliuqing@cnnic.cn)；李晓东(1976-)，男，山东人，研究员，博士，主要研究方向为互联网基础资源、网络安全与管理、互联网数据分析等；耿光刚(1980-)，男，山东人，博士，主要研究方向为模式识别、互联网信息检索.

b)基于网页模板的提取算法。这类算法可以大致分为两类,一类是从同种结构的网页集中提取出模板作为参考,以提取同种结构的其他网页中的正文内容,这种模板一般从同一网站的不同网页中提取;另外一类是从各种不同的网页中归类,并分别提取抽象层次更高、归纳性更强的通用模板。对于网页正文内容的提取主要通过模板匹配来完成[3~5]。Reis 等人[6]设计出了一种使用于树的类正则表达式 ne-pattern,以 RTDM(restricted top-down mapping,有约束的自顶向下映射)算法对样本网页进行聚类,并从聚类结果的 DOM 树中提取出 ne-pattern 作为该聚类的模板,每个网页的标题和正文部分处理 ne-pattern 的通配符部分。Vieira 等人[7]对 RTDM 算法进行拓展,使用树的最小编辑距离,记录下从一棵 DOM 树变成另外一棵 DOM 树所需要的代价最小的操作,以此来寻找两棵 DOM 树中理论上最相似的部分,实现对 DOM 树模板的检测和删除。这些方法从模板角度提供了比较新颖的思路,但是计算量非常大,在处理海量数据的时候,效率会比较低。

c)基于视觉分块的提取算法。这类算法从用户对网页的视觉感受出发,依照网页中节点的样式特点对页面分块,再从分块结果中找出正文所在的块来达到提取正文的目的,它会更多地利用 DOM 树中节点的 style 信息和 CSS 资源文件中的数据。微软亚研院提出了 VIPS 算法[8~10],该算法由于利用非常多的 CSS 信息以及多轮迭代,所以有一定的计算量;另外,他们所使用的启发式算法对于现在的一些网页的适用性也不是足够地好。黄文蓓等人[11]以 TVPS(table and vision based page segmentation,基于表格和视觉的页面分割)算法为参考,构建 DOM 树,以〈div〉〈table〉等容器标签为基准,寻找最低层容器节点的各个文本节点进行合并,计算信息量并比较最低层容器节点与其兄弟节点、父节点的信息量,从而选择出能够构成文本块的节点。该方法考虑到了 DOM 中包含文本的节点的结构性,相比于原来的算法,准确率得到了一定的提升,但算法的运算量依然比较大。

d)基于统计、机器学习的提取算法。这类算法通过对样本网页的正文分布、节点特征等信息作统计分析来建立模型规则,并通过不断学习来完善模型中的参数,以在一定程度上达到自适应性。Kitahara 等人[12]在计算单词密度分布的基础上构造内容文本密度分布,以抽取网页核心文本。但该方法实际上是一个摘要提取方法,并且单词密度分布的计算需要以网页正文的提取为基础;另外该方法并不是特别适合中文网页。Kim 等人[13]以单词密度、链接密度、HTML 标签分布以及文本块之间的距离为特征训练分类算法,以决策树的形式来判别文本块是否属于核心正文内容。但是,分类算法的训练需要人工标记样本,而样本的选择会影响算法的准确率,且算法计算量较大。Moreno 等人[14]从网页标签序列中找出对应的文本序列,在网页正文与非正文在长度和标签数量存在差异的基础上构建网页文本密度图,以统计的方法识别出网页正文部分。这种方法的准确性依赖于网页的布局风格,即网页的正文内容的分布满足一定的连续性而且 HTML 标签的使用符合规范,另外统计算法中用到的阈值难以保证对所有的网页正文分布均有效,因此该算法的适用性存在局限。

目前,也有一些开源和非开源的网页正文提取产品和项目在网络上出现。其中与网页正文提取有关的开源项目比较多,如 Decruft、BoilerPipe、cx-extractor 等;在产品上,一些“待阅”应用产品中会包含有网页正文提取的功能,如 Readability、Instapaper、Flipboard 等。这些项目与产品在效果与性能也是各有优劣。

另外,Diffbot(http://www.diffbot.com)在网页信息提取上是一款非常值得说明的产品。它使用了包括数据挖掘、机器学习、自然语言理解、人工智能和计算机视觉等诸多复杂技术,对网页内的数据以及网页的类型进行定义和分类,通过扫描 Web 页面来识别其中各个部分的数据,进而再转换为可用的数据库格式。笔者在调研这款产品时发现,在部分中文网页的内容提取上它会丢失一些关于时间和数量等的数字信息;另外该产品有时会把相关链接错误地分析为当前网页内容的分页页面链接,从而将其他网页的正文提取到当前网页正文中。总体来说,Diffbot 的准确性是在实际应用可以接受的范围内,不过,该产品的 API 作为商业用途使用并不开源。

2 基于布局相似性的网页正文提取

对不同的网页类型,其核心内容(正文内容)的定义往往因所处环境的不同而不同。主题型网页(topic)一般会包含文本、图片、视频等内容,而索引型网页(hub)一般会包含大量对其他网页的相关链接。对索引型网页作正文提取的意义并不是很大,因为真正有价值的信息位于它所包含的链接所指向的主题型网页中。对于主题型网页,不同的人对网页所关注的内容也不尽相同,有的人倾向于文本,有的人倾向于图片视频,还有的人倾向于相关内容的评论。从研究角度出发,本文只关注于主题型网页中的文本正文内容。以文本为主的主题型网页一般在新闻、博客、资讯、社交类等网站中较为常见,这也是本文重点处理的网页源对象。

2.1 网页的布局相似性

在浏览新闻、博客等主题型的网页时,会发现同一网站下同一专题(同一频道)新闻网页或者同一作者的博客网页的布局结构极其相似,这类网页在一些固定的位置上会出现相同或相似的内容。例如,新闻网页的导航、全局热点排行、广告链接、版权信息,博客网页的作者个性签名、文章标签、网站导航信息、博客导航信息、网站版权信息、广告链接等。新浪新闻频道的新闻网页布局示例如图 1 所示。

图 1 网页布局示例

在图 1 中,布局和内容上相似的数据以方框标志,从上到下分别为导航栏、注册登录按钮、头条推荐、24 小时点击排行。这样的现象与现代网站建设、网页开发模式有很大的关系。对于实时性较强的新闻网站来说,现代信息极快的增长速度使得制作静态页面的人力和时间成本升高,所以大部分的网站通常

使用动态页面。使用 HTML、CSS、JS 来制作通用的前端模板(包括视觉样式、图片以及相关的组件功能),然后利用后台脚本从数据库中读取相应的数据,与前端模板组合生成新的页面。对于博客类型的网站来说,基本也是这样的。

新浪财经频道导航信息栏 HTML 代码如下:

```
〈div data-sudaclick = "top_nav_01"〉
  〈ul class = "nav"〉
  〈li〉〈a href = "http://www.sina.com.cn/"〉首页〈/a〉〈/li〉
  …
  〈/ul〉
〈/div〉
```

从以上代码可以看到,导航信息位于 DOM 树中的一个 div 节点中:〈div data-sudaclick = "top_nav_01"〉。在新浪财经频道的其他新闻网页中,导航信息布局同样如此。在博客网站中,如博客园(www.cnblogs.com)等,同属于一个作者的博文网页也具有类似的现象。

从网页开发模板的角度来说,网页内容布局与样式结构的相似意味着网页前端代码模板的相似。在一组布局相似的网页集中,网页各个部分内容的组织遵循相同的布局规则,即在这些网页中与核心内容无关的区域中的内容对应地相同或相似。从 DOM 树的角度来说,在来自同一组布局相似的网页集的两个不同网页的 DOM 树中,会有完全相同的子树,这些完全相同的子树所代表的数据,基本上就是导航、版权、广告等与核心正文没有关系的内容,而它们就是在正文提取的过程中需要去掉的冗余噪声数据。因此,考虑到网页正文提取的实际应用场景,本文所采取的策略是:比较两个内容布局与样式结构相似度较高的网页的 DOM 树,从中删除完全相同的节点(或子树,如果这个子树中的所有节点在两个 DOM 树中的位置和数据完全相同)。理论上,在经过相同节点的删除操作后的 DOM 树中,大部分的噪声被去除,冗余信息大大减少。

2.2　算法描述

通过上文中对网页布局相似性现象的分析可以看到,布局相似性对应着 DOM 树节点的相似性。对于两个布局相似的网页,本文称其中一个为目标网页,它的正文内容可能是用户更为关心的;另外一个为参考网页,它对目标网页中的节点布局规则提供一个参照;而从提取网页正文内容的角度上来说,它们是互为目标网页和参照网页的。

本文以参与比较的节点之间的相似程度来度量该节点对的相似性,节点对按照相似程度可分为三种:

a)完全相同节点。对于目标 DOM 树中的节点〈xxx id = "id1" class = "cls1" attr = "attr1" …〉…〈/xxx〉来说,如果在它的参考 DOM 树中存在一个节点在标签名(xxx)、属性集(id = "id",class = "cls1",attr = "attr1",…)以及节点内容(包括节点中的文本内容和子孙节点)上与该节点完全对应相同,则称这样的一对节点是完全相同节点。

b)准相同节点。只在标签名、属性集上相同的一对节点(这两个节点分别来自目标 DOM 树和参考 DOM 树)称为准相同节点。

c)不相同节点。如果节点对(这两个节点分别来自目标 DOM 树和参考 DOM 树)在标签名或属性集上出现不同,则这两个节点是不相同节点。

属性集相同指的是两个属性集中包含的键值对个数相同,两个属性集中的属性名也分别相同,相同属性名对应的属性值也相同。由上所述,节点完全相同是一个递归形式,即如果两个节点完全相同,则它们的标签名和属性集对应相同,并且它们的文本内容以及子节点也对应相同。

本文首先将参与算法的两个布局相似的网页(目标网页和参考网页)的源代码利用开源 DOM 树解析工具转换成便于分析的 DOM 树结构,然后进行相关算法处理。算法的结构细节如下:

a)对这两个 DOM 树进行预处理。

(a)删除与网页核心内容无关的辅助性元素,如 script,noscript,style,link,meta。这些元素包含了页面的显示样式信息、组件功能脚本,以及对于相关 CSS 文件、JS 脚本、图片等资源的链接引用,这些内容对于网页正文来说属于无用信息,所以将它们删去。

(b)删除用于展示和人机交互等功能的组件元素,如 form,fieldset,legend,input,select,menu,optgroup,option,textarea,map,area,applet,object,param,button,label。因为本文主要面向的是主题型的网页,而这些功能型的元素一般用于动态交互,所以与核心文本内容没有太大关系,将它们删去。

(c)删除视觉上不可见的元素,主要包括具有以下样式属性的元素:[style = "display:none;"],[style = "visibility:hidden;"];等。因为这些元素的不可见,它们对于核心内容不构成影响,这些元素同样也不影响可见的广告链接的布局,所以可以将它们删除。

(d)删除被注释的代码以及其他被注释的内容(HTML 的注释由〈! -〉构成)。注释内容实际上也属于 DOM 节点的文本内容,但它们并不是读者所需要的,所以将它们删除。

(e)删除空节点。空节点主要有两种形式,即〈xxx〉〈/xxx〉和〈xxx/〉。对于前一种形式,其不包含实质性的内容,属于核心内容无关的节点;对于后一种形式,如果对于图片〈img/〉和其他样式元素〈br/〉等有所需要的话,可以选择不删除。由于本文主要处理的是主题性文本页面,更关注于文本信息,对这两类空节点都进行了删除处理。

网页 DOM 树作为网页渲染树的构建基础,包含了所有可视元素、样式数据、图片视频等相关资源的引用,功能性脚本以及一些在浏览网页时看不见的“暗链”数据。对一棵如此复杂而又庞大的 DOM 树进行分析,效率是一个必须要考虑的因素。因此,对与网页正文内容无关、影响算法效率的节点的删除,不仅能减少在 DOM 树分析上花费的时间,同时也能得到更精炼、更准确的结果。

b)将经过预处理的两棵 DOM 树作为输入,进行完全相同节点的删除操作,最后得到两个去掉完全相同节点的DOM 树。

从上文对节点按相似程度的分类可以看到,由于 DOM 树节点内容的复杂性以及节点完全相同概念中隐含的递归性,使用传统的树遍历比较法会大大降低算法的效率。事实上,对于两个完全相同的 HTML 节点来说,如果它们在网页源代码中的代码格式和代码风格相同,即网页源代码在代码行缩进、标签格式书写等规则上相同(由于网页源代码的编写相对自由,所以两个完全相同的 DOM 节点源代码格式和风格可能会因为程序员的编码风格或者所使用的开发软件的不同而不同),则它们对应的源代码的行数应该是相同的,并且在源代码中对应行的字符串的内容也应该是相同的。因此,节点的比较就可

以使用简单的方法:先将源代码转换成统一的代码风格(利用DOM树解析工具即可实现),然后以字符串的形式比较准相同节点的源代码的各行,如果完全相同,则这两个准相同节点是完全相同节点,可以将它们删除;如果出现不同,则从参与比较的目标DOM树的节点的子节点开始,逐个在参考DOM树中进行准相同节点的寻找和判断,并删除完全相同的节点,直至遍历了所有的子孙节点并且删除了所有完全相同的节点。如果处理完当前节点(包括其子孙节点),则继续处理它的兄弟节点,直至遍历了整棵树的所有节点。完全相同节点删除算法的伪代码如下:

算法 DOM树完全相同节点删除。

输入:目标网页DOM树tarDom,参考网页DOM树refDom。

```
1  StackE. push(tarDom. body( ). children( ));
2  while StackE is not empty:
3     TE < -StackE. pop( );
4     Ses < -refDom. body( ). selectSim(TE);
5     Result < -compare(TE,SEs);
6     if Result not NULL:
7        tarDom. removd(TE);
8        refDom. remove(Result);
9     else:
10       StackE. push(TE. children( ));
11 End While
```

函数说明:

body( )表示从DOM树中返回body节点;

children( )表示返回当前节点的子节点集;

selectSim(node)表示从DOM树或当前节点中返回与节点node在标签名和属性集上相等的准相同节点集;

compare(node,nodes)表示返回在节点集nodes中第一个与node节点完全相同的节点,若没有完全相同节点,则返回NULL;

remove(node)表示从DOM树或当前节点中删除节点node。

当元素栈StackE为空时,目标DOM树中的所有节点均完成了相同节点的判断和对应的删除工作,算法结束。

c)在完成了完全相同节点删除后,基于布局相似性的网页正文内容提取算法即已执行完毕。

至此,本文的算法已经基本上删除了目标网页与参考网页中完全一样的一些噪声信息,包括导航、版权、热点推荐、全局排行推荐以及相关的广告信息等,即在这两个页面DOM树中,以节点为单位,完全一样的内容已经被删除了。网页中剩下的内容都属于异质内容,包括网页正文、相关阅读、评论、评论人数、内容贡献者等信息。在这些剩余的内容中,网页正文占据了绝大部分。在不影响建立索引和内容聚合情况下,这些冗余信息可以不予处理。如果需要处理,则可以使用行块分布函数[15],即把已经去掉包含大部分无关信息的节点后的网页文档进行清除HTML标签处理,然后对剩下的纯文本内容进行每一行的字数统计,通过设置阈值来找出正文内容所在的文本行区间,以此来精确提取网页正文块。

在具体的实现中,本文使用两条简单的启发式规则来去掉可能无关的一些链接元素,即〈a〉..〈/a〉这样的元素,以达到对算法结果的一个“提纯”。事实上,也可以将这两条启发式规则与行块分布函数相结合,以求更精准的网页正文提取结果。启发式规则如下:

规则1 考察包含链接元素的节点的链接文本密度,即链接元素所包含的文本数/链接节点父节点所包含的文本总数,并设置相应的阈值;该指标的结果区间为[0,1],如果该指标大于某个阈值,则可以认为该节点(链接元素的父节点)与网页核心正文的相关性不大。因为核心内容的链接文本数不应过大,从而可以删除该节点。阈值可以根据采样来统计得到,经实验统计验证,阈值设为0.25~0.3是比较合适的。

规则2 考察包含链接元素的节点的链接分布,最简单的即如果该节点只包含链接元素,而不包含其他元素,且链接元素之间没有非空节点和实质性文本(非实质性文本主要包括空格、下划线“_”、横线“-”、方括号、原括号等特殊符号而不包括表达特定意义的字符串),则认为该节点不包含实质性内容,从而可以删除该节点。这条规则对于聚集性的链接有很好的识别性。

启发式规则应用于网页正文提取的困难之处在于,网页的编写语法相对自由宽松,相关标准也不是特别严格,元素的嵌套有时会特别复杂,甚至会有不合规则的编码(由于浏览器强大的容错性,这些错误的语法不会太大地影响用户体验),因此要想总结出较好的启发式规则需要大量的样本网页参考(样本网页的选择要多样化,否则容易出现过拟合的情况),而这也给相关工作带来了很大的不便。所以,启发式规则的使用需要在准确性和通用性两方面上进行折中。事实上,可以将这两条启发式规则与行块分布函数相结合,以求更精准的网页正文提取结果。

d)对于目标网页的正文提取操作,需要选择与目标网页在内容布局与样式外观上相似度较高的网页作为参考网页,与目标网页一起作为基于布局相似性的网页正文内容提取算法的输入。

通过观察验证,属于同一个网站、同一专题、同一频道、同一作者等的网页的内容布局与样式外观的相似度会比较高。而满足这样条件的网页,有一个最直观的规律——它们的URL相似度也非常高,这一点从网页开发与网站管理的便捷和高效性来说也是非常合理的。

Maurer等人[16]在网络钓鱼的应用背景下,使用字符串最小编辑距离,综合搜索引擎的拼写推荐,在URL的各个部分中寻找与目标URL核心词项相似的子词项,通过子词项的对比来进行URL的相似性度量。Qi等人[17]在URL相似性的计算上使用Dice系数并结合统计方法完成URL的相似度量。以上两篇文献更多地是从字符串处理角度出发,考虑到URL的语法与结构特点,在协议与域名部分相同的情况下,本文使用“相同目录层数/最大目录层数”这样一个相对简单的测度来进行目录路径和查询参数的相似性计算,即 $similarity = cl/maxcom(la,lb)$。其中,$cl$ 是从两个URL根目录开始依次对应相等的最大共同目录层数(或两个URL查询参数集中相等的键值对数目);$maxcom(la,lb)$ 是相互比较的两个URL的最大目录层数(或两个URL中查询参数集的最大键值对数),la 代表其中一个URL的目录层数或查询参数集中的键值对,lb 则代表另外一个URL的目录层数或查询参数集中的键值对。当 $similarity = 1$,两个URL目录几乎完全一样(不考虑路径参数和段参数);当 $similarity = 0$,两个URL目录完全不一样。因此,如果相似度越接近1但又不等于1,即从路径根目录开始,两个网页文件所处的目录在结构上相近,或者路径完全相同(URL

不包含文件名部分)而查询参数仅有少数不同(注意,并不是所有的 URL 都会有文件名,对于动态网页来说,更是如此),则这两个 URL 对应的页面的内容布局和样式外观相同或相似度就越大。

2.3　算法总结

从上面的算法描述可以看到,本文使用的算法利用了网页布局结构的一些特点,即在内容布局和样式外观上相似度较高的网页,它们的 DOM 树结构中会有对应相同或相似的节点。而这些相同或相似节点所包含的内容一般是这些相似网页所“共有”的内容,它们往往是与当前网页的核心内容无关的数据,也即是本文算法要删除的杂质数据。但是本文并没有对网页的样式风格、布局情况(包括布局标签类型〈div〉〈table〉,文本容器类型〈div〉〈p〉等)等信息作任何具体的假设,思路简单、计算量小、通用性较好。

3　实验效果和有效性分析

3.1　编码实现

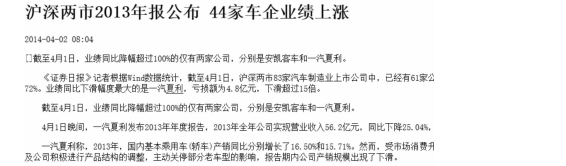
在编码实现上,由于本文的算法需要解析网页的 DOM 树结构,而网页的 HTML 源代码不尽规范,需要对一些格式不好(如标签没有完整闭合)的标签以及一些非法标签进行处理。考虑到开发的便捷性,可使用网上现有的一些开源工具包来帮助开发。可用于网页 DOM 树解析的开源工具有很多,比较常用的包括 HTMLParser、NekoHTML、JTidy、HtmlCleaner、Cobra、Beautiful Soup、jsoup 等,它们在构建 DOM 树、解析 DOM 树、操作 DOM 树等操作上提供的功能及特点不尽相同。

根据本文算法的需求,选择基于 MIT 协议发布的 jsoup(http://jsoup.org/)作为主要的 DOM 解析工具。它提供了非常便利且多样的 DOM 树解析方法,并且经由 jsoup 处理的 DOM 树节点输出为字符串时具有统一的标准代码格式,便于执行完全相同节点删除操作。另外,joup 提供的 clean 函数和 whitelist 模块可以轻易地完成 HTML 标签清除工作,便于应用行块分布函数等算法。

3.2　实验效果

用于测试的网页样本选自热门的新闻、博客网站,算法主要面向的网页类型以富文本的主题型网页为主。网页来源百度百科、新浪、搜狐、网易、腾讯、中国政府网、博客园、CSDN、凤凰网、央视网(CCTV)、联合早报等共 11 个网站,每个网站中的网页来自五个不同的专题或频道(百度百科除外),语言以中文为主。基于布局相似性的网页正文内容提取算法提取结果的效果如图 2 所示。从图 2 可以看到,该算法可以较为完整地抽取网页的正文内容(包括中文中的图片或视频的文字说明),这些即为基于布局相似性的网页正文内容提取算法的原始输出结果(没有使用行块分布函数进行提纯)。但同时,在提取结果中还残留有相关推荐阅读和一些其他杂质信息,这也是该方法本身所去除不了的信息,不过在算法描述中说过,这些杂质信息可以在算法执行的输出上加入行分布函数等统计取精算法来获得更干净的结果。

另外,该算法提取的正文即网页源代码中 HTML 标签内的文本,如果标签内的文本含有转义实体引用,则提取的结果中也会含有转义实体,可以使用 org.apache.commons.lang3 中的工具函数来完成对转义引用实体的处理。


沪深两市2013年报公布 44家车企业绩上涨

2014-04-02 08:04

□截至4月1日，业绩同比降幅超过100%的仅有两家公司，分别是安凯客车和一汽夏利。

《证券日报》记者根据Wind数据统计，截至4月1日，沪深两市83家汽车制造业上市公司中，已经有61家公72%。业绩同比下滑幅度最大的是一汽夏利，亏损跌为4.8亿元，下滑超过18倍。

截至4月1日，业绩同比降幅超过100%的仅有两家公司，分别是安凯客车和一汽夏利。

4月1日晚间，一汽夏利发布2013年年度报告，2013年全年公司实现营业收入56.2亿元，同比下降25.04%，

一汽夏利称，2013年，国内基本乘用车(轿车)产销同比分别增长了16.50%和15.71%。然而，受市场消费升及公司积极进行产品结构的调整，主动关停部分老车型的影响，报告期内公司产销规模出现了下滑。

安凯客车2013年实现归属于上市公司股东的净利润为亏损3472万元，同比下滑136.5%。公司表示，2013年同比下降13.06%，销量位居行业同类客车第六位；实现销售收入35.4亿元，同比下降7.87%。

扣除非经常性损益后，公司实现归属于上市公司股东的净利润为亏损1.29亿元，同比下滑409%。来自政府

61家公司中，业绩同比增幅超过100%的公司有9家，5家公司属于整车企业，分别是中国重汽、 比亚迪 、

中国重汽也成为目前为止暂时“登顶”行业、业绩同比增幅最大的企业。2013年公司全年累计销售汽车8 719.03%。

去年商用车行业特别是重卡行业呈现前低后高的走势，中汽协数据显示，2013年国内重卡市场全年销量

中国重汽表示，从整个行业来看，虽然市场实现复苏，销量大幅增长，但距离2010年的高峰相差甚远。

·上汽与车队发布年报 称汽车市场走出低迷 2014.03.28 ·交通年报：北京去年日均堵车115分钟 2014.02. 车企半数年报“变脸” 利润结构分化 2013.04.07

[责任编辑：chanellai]

价格： 2.88-3.80万 口碑评分： 59分 图片 资讯 配置 口碑 油耗

图 2　一篇财经新闻网页的正文提取结果

3.3　对比分析

为了对算法的准确率进行更加客观的评估,本文从上述的 11 个网站中各随机抽取 100 个网页,总数量为 1 100;再从每个网站网页样本集中随机抽出一定数量的网页的抽样集进行测试,抽样测试集容量为 100。考虑到网页类型的繁杂以及对测试网页正文标记的高成本,本文采取抽样的方法来验证算法的有效性。

由于在本文提出的算法所得到的结果中,正文的召回率接近 100%,所以主要用来衡量该算法有效性的指标是正文提取的正确率,即在抽取结果中正文内容所占的比重。对于抽样测试集中的网页正文节点内容进行人工标记,本文对于文本类网页的正文内容界定为标题、时间、来源、编辑人、文章内容,包括图片的说明以及视频的说明,但不包括图片和视频组件,也不包括文章的相关链接。对于分页显示的文章,本文只计算当前网页中的正文内容,而不考虑整篇文章的正文内容。算法的正确率定义为:在内容抽取结果中,标记正文内容所占的比重。

参与对比的算法是 Readability 正文抽取算法,在 GitHub 上有 Python 版本。为了便于正文抽取结果正确率的计算、分析和比较,笔者用 jsoup 实现了该算法的 Java 版本。抽样测试结果对比如表 1 所示。

表 1　算法平均正确率结果对比

网站名	布局相似性算法	Readability 算法
新浪	0.979 9	0.950 2
搜狐	0.839 9	0.891 2
网易	0.944 9	0.855 0
腾讯	0.892 2	0.940 3
百度	0.973 3	0.855 2
中华政府网	0.945 1	0.846 5
博客园	0.981 8	0.978 8
CSDN.NET	0.988 7	0.914 8
凤凰网	0.885 1	0.807 4
联合早报	0.981 0	0.938 0
CCTV.COM	0.833 8	0.889 4
平均正确率	0.931 4	0.897 0

从各个网站的抽样计算结果来看,本文算法与 Readability 算法的结果比较类似。在平均正确率上,本文算法优于 Readability。Readability 在正文提取上会忽略掉摘要以及层次较高的标题,例如在处理百度百科的网页时,Readability 提取的结果中不包含摘要、目录以及每一章节的大标题。而这些数据,本文的算法均能保留下来,所得到的结果也更加完整。就提取正文的具体内容来看,对于导航信息、版权、广告和全局热点等内容,两种方法均能非常有效地去除。Readability 对噪声数据过滤得相对更干净一些,它在大部分情况下,能非常有效地去

除相关链接和评论等部分的杂质信息,但在一定程度上也会降低网页正文内容的召回率,即它在正文内容的提取上并不总是提取得很完整;而本文算法在召回率上接近100%,对原网页正文内容提取得相对更加完整,但在一定程度上会引入少量杂质信息(这些信息在更严格的应用场景下,可以继续引入其他处理环节对本文算法的结果进行补充)。从内容提取与杂质信息过滤的总体效果上来说,本文算法与Readability算法在正文内容提取上表现均不错,而且本文算法效果更优一些。

另外,由于不同网站、不同专题的设计风格不尽相同,以及人们对于网页中正文的界定差异等因素的存在,本文算法的结果或多或少地都会受到一定的影响。对于正文内容为纯文本的网页,本文算法的准确率很高;对于包含过多图片、视频组件、相关链接以及正文内过短的网页,算法的准确率会因为杂质内容所占的比重较大而相对降低。另外,算法的结果受相互参照的两个页面在布局结构上的相似程度的影响也比较大。对于影响本文算法正确率的几个主要因素总结如下:

a)相互参照的页面的布局结构相似程度。这依赖于URL相似性判别等寻找参照页面的算法是否合理、鲁棒。

b)网页中的核心内容相比于无关内容的比例。如以视频、图片为主要内容的新闻和博客内容,过短的正文内容会降低提取结果的正确率指标。

c)网页的布局设计是否规范、合理。本文算法是在节点层利用布局相似性这个特性,所以在面临网页布局杂乱、有用数据与杂质数据相互混杂的情况时,会保留与正文包含于同一节点的部分杂质内容。

在一般情况下,本文算法提取的正文结果中残留的少量杂质信息并不会影响对于正文内容的阅读和分析,毕竟相比于正文内容,杂质内容所占的数据太少,可以提取的信息量也很少。因此综合来说,本文提出的基于布局相似性的网页正文内容提取算法的效果还是很不错的。

4 结束语

本文所使用的基于布局相似性的网页正文提取算法是在经过对网页结构和样式的观察以及一些相关实现后提出的,思路简单、易于实现、便于拓展。通过对他人研究成果和产出产品的调研,以及对本文算法的实验结果可以看到,由于网页结构和样式的灵活多变,基于DOM树的网页正文内容提取算法需要在保证准确率的同时,兼顾对不同网页的普适性和运算处理的高效性。

对于本文提出的方法,在今后的研究中可以考虑与统计方法相结合,提高正文提取算法结果的准确率。在URL相似性的判断计算上,本文所使用的方法比较简单,要实现更精准的判断还需要作进一步的研究,这部分内容还将有助对内容分页的网页进行完整正文的提取。另外,由AJAX动态生成网页的技术变得越来越流行,在未来的Web技术中的应用也会越来越广泛,目前出现的算法和所调研的产品对于这种网页的正文提取做得均不是很好,这也是一个值得研究的问题。针对网页的正文内容进行相关的聚合处理,同样是未来工作的一个重要组成部分,如果在聚合算法上能够对所提取的网页正文内容充分利用,那么就可以降低对于网页正文提取的准确率的要求,转而可以考虑提高正文提取的算法运算效率,以适应未来网页数量爆炸式增长的应用环境的效率需求。

参考文献:

[1] Gupta S, Kaiser G E, Grimm P, *et al*. Automating content extraction of HTML documents [J]. World Wide Web, 2005, 8(2): 179-224.

[2] Guo Yan, Tang Huifeng, Song Linhai, *et al*. ECON: an approach to extract content from Web news page[C]//Proc of the 12th International Asia-Pacific Web Conference. [S. l.]: IEEE Press, 2010: 314-320.

[3] Mane T B, Potdar G P. Template extraction from heterogeneous Web pages [J]. International Journal of Advanced Computer Research, 2012, 2(6): 197-201.

[4] Kadam V, Devale P R. A methodology for template extraction from heterogeneous Web pages [J]. Indian Journal of Compute Science and Engineering, 2012(3): 449-452.

[5] Ma Ling, Goharian N, Chowdhury A, *et al*. Extracting unstructed data from template generated Web documents[C]//Proc of the 12th International Conference on Information and Knowledge Management. New York: ACM Press, 2003: 512-515.

[6] Reis D, Golgher P, Silva A, *et al*. Automatic Web news extraction using tree edit distance[C]//Proc of the 13th International Conference on World Wide Web. New York: ACM Press, 2004: 502-511.

[7] Vieira K, SilvaI A, Pinto N, *et al*. A fast and robust method for Web page template detection and removal[C]//Proc of the 15th ACM International Conference on Information and Knowledge Management. New York: ACM Press, 2006: 258-267.

[8] Cai Deng, Yu Shipeng, Wen Jirong, *et al*. VIPS: a vision-based page segmentation algorithm, MSR-TR-3003-79 [R]. [S. l.]: Microsoft Research, 2003.

[9] Cai Deng, Yu Shipeng, Wen Jirong, *et al*. Extracting content structure for Web pages based on visual representation [J]. Web Technologies and Applications, 2003, 2642: 406-417.

[10] Mehta R, Mitra P, Karnick H. Extracting semantic structure of Web document using content and visual information[C]//Proc of the 14th Special Interest Tracks and Posters of International Conference on World Wide Web. New York: ACM Press, 2005: 928-929.

[11] 黄文蓓,杨静,顾君忠. 基于分块的网页正文信息提取算法研究[J]. 计算机应用, 2007, 27(6): 24-30.

[12] Kitahara S, Tamura K, Hatano K. Extraction of Web texts using content-density distribution [J]. Information Retrieval Technology, 2011, 7097: 273-282.

[13] Kim M, Kim Y, Song W, *et al*. Main content extraction from Web documents using text block context[J]. Database and Expert Systems Applications Lecture Notes in Computer Science, 2013, 8056: 81-93.

[14] Moreno J A, Deschacht K, Moens M. Language independent content extraction from Web pages[C]//Proc of the 9th Dutch-Belgian Information Retrieval Workshop. 2009: 50-55.

[15] 陈鑫. 基于行块分布函数的通用网页正文抽取 [EB/OL]. (2011-08-17). http://code.google.com/p/cx-extractor/.

[16] Maurer M, Hofer L. Sophisticated phishers make more spelling mistakes: using URL similarity against phishing[C]//Proc of the 4th International Symposium on Cyberspace Safety and Security. Berlin: Springer, 2012: 414-426.

[17] Qi Xiaoguang, Nie Lan, Davison B D. Measuring similarity to detect qualified links[C]//Proc of the 3rd International Workshop on Adversarial Information Retrieval on Web. New York: ACM Press, 2007: 49-56.