山西大学学报(自然科学版)39(3):386－391,2016
Journal of Shanxi University(Nat. Sci. Ed.)

DOI:10.13451/j.cnki.shanxi.univ(nat.sci.).2016.03.007

融合结构和内容特征提取多类型网页文本要素

王宇龙[1,2],赖华[1,2]*,余正涛[1,2],洪旭东[1,2],刘书龙[1,2]

(1.昆明理工大学 信息工程与自动化学院,云南 昆明 650500;
2.昆明理工大学 智能信息处理重点实验室,云南 昆明 650500)

摘　要:针对网页设计结构与文本内容上的关联特点,提出了融合结构和内容特征的多类型网页文本要素提取方法。依据网页头部标题元素与网页体内容上的联系提取网页标题;提取网页正文区域的网页结构和内容上的多个特征分类网页DOM节点,定义节点的扩展、整合规则获得正文候选块,引入密度值和影响因子从各候选块中甄别正文块;利用发布时间与标题、正文之间的位置关系,通过正则表达式实现发布时间的提取。对国内新闻网站、博客、论坛及贴吧进行抽取试验,结果表明该方法具有较好的效果。

关键词:多类型网页;网页要素自动提取;结构特征;内容特征

中图分类号:TP391　　**文献标志码**:A　　**文章编号**:0253-2395(2016)03-0386-06

Extracting Textual Elements of Multi-types Webpages by Fusing Content and Structure Features of the Webpage

WANG Yulong[1,2],LAI Hua[1,2]*,YU Zhengtao[1,2],HONG Xudong[1,2],LIU Shulong[1,2]

(1. *School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming* 650500, *China*;
2. *Key Laboratory of Intelligent Information Processing, Kunming University of Science and Technology, Kunming* 650500, *China*)

Abstract: In order to effectively use the webpage related features in the design structure and text content, we proposed a way about extraction textual elements of multi-types webpages by fusing structure and content of the webpage. According to the contact between the title elements of webpage header and content of body in a webpaqe, we can extract the title. Moreover, extracting the multiple feature about the webpage structure and content of the text area classified DOM node, the extension and integration rules of nodes are defined to get the candidate blocks of the text, the introduction of density value and impact factor for the candidate blocks in order to distinguish the text block. Using the position relation between the release time and the title and the text, we can extract the release time by the regular expressions. On the domestic news websites, blogs, forums and the post bar to extract test, the experimental results show that the proposed method has a good effect in extracting webpage text elements.

Key words: multi-types webpage; automatic extraction of webpage text elements; structural features of webpage; content features of webpage

收稿日期:2016-03-02;**修回日期**:2016-03-29
基金项目:国家自然科学基金(61175068;61472168);云南省自然科学基金重点项目(2013FA030)
作者简介:王宇龙(1991－),硕士研究生,主研领域:文本挖掘。
* **通信作者**:赖华(LAI Hua),E-mail:lhkg666@163.com

0 引言

随着互联网技术的迅猛发展,大量有价值的信息分布在各式各样的网页上。然而,由于种种原因很多的噪音被植入网页中,这给下一步大规模的文本分析任务带来了极大的困难,所以,从包含大量垃圾信息的网页中提取网页的标题、正文、发布时间三要素就显得非常迫切和重要了。当前,对正文提取的研究比较多,现有的网页正文提取方法主要分为基于网页结构和基于统计理论两类方法[1]。基于网页结构抽取正文的方法[2-3,5-6],其主要思想是利用一些手段将分散的网页结构有序化,再依据网页结构中某些部分的相似性,通过定制模板或者设定规则将噪音内容与正文内容区分开。比如杨柳青等人[4]发现同一网站的同一专题的网页在布局和样式方面具有相似的特点,提出并实现了一种基于布局相似性的网页正文提取方法。欧建文等人[7]提出通过机器学习算法生成网页集的结构模板,依据模板的提取规则对网页正文进行提取。耿焕同等人[8]借助网页解析器对网页结构进行视觉上的预处理,转换成规范的 Tag 视觉树,再利用相似子树启发式选择函数等,找出正文的抽取路径。不过面对日益复杂多变的网页结构,再加上大量不规范的页面在互联网上普遍存在等问题,对应用此方法提取网页正文造成很大的挑战。基于统计理论抽取正文的方法[9-10],其主要思想是通过统计网页中各个 HTML 标签所包含的信息量,根据经验设定正文阈值,最后确定网页正文的位置。秦成磊等人[11]依据 HTML 标签的每对“>”和“<”之间的文本信息,对其长度进行统计并按照匹配顺序进行排序,设定文本长度的最优阈值,划定文本行号区间,最后利用公共子序列进行优化并完成正文提取。刘利等人[12]利用多个正文文本特征,将这些特征转化为统计信息值,根据多次实验的结果设定阈值确定正文范围。此类方法提取网页正文的关键点在于设置合理的阈值,阈值确定是否正确对正文区域的判定产生直接的影响,在实际应用中有一定的局限性。

通过观察大量网页的结构,设计者会将标题内容和其他的一些无关信息(比如网站名称)共同作为网页头部中标题元素的内容,而在网页体中,存在某几个文本节点的内容与头部中标题标题元素的内容对应关系,利用这些文本节点的标签和字符占比等特征,制定过滤规则剔除标题元素中无关信息是提取网页标题的核心。确定网页正文区域与标题有很大的关联。通过分析正文在网页布局和文本内容上的特点,发现标题的实词会多次出现在正文内容中、标题的位置在网页正文之上、正文区域的文本分布比较密集,并且正文区域的 HTML 标签也呈现出一些规律等特征,但是通过这些特征只能确定部分正文内容并且还会夹杂很多噪音信息。通过分析网页中正文区域的结构,一般正文区域会被一种特定的容器标签所包含,但是有时候噪音内容也会被这种标签所包含,为了区分正文和噪音,通过比较两者,发现正文区域相比噪音区域一般会有较少超链接、显示样式和标签数量,并且正文区域的标题实词出现总数和标点符号数量也比噪音区域多,利用这些特征就可判断出正文区域。然后,根据国内网站展示网页内容的习惯,在网页中,发布时间的位置在标题和正文之间,利用这种位置关系特点即可提取发布时间。这样,就可以自动抽取出网页文本三要素。

1 HTML 网页预处理

由于网页中存在大量的噪音,所以在对网页解析前,先进行清理:(1)删除与网页文本要素无关的内容,包括:〈script〉、〈link〉、〈img〉、〈style〉等标签及内容;(2)删除注释标签及内容。然后,选取第三方 Jar 包 Jsoup.jar,以<body>标签为根节点构建网页 DOM 树。最后,遍历 DOM 树并且对每一个标签节点(只标注成对出现的标签)按照逐一递增的顺序进行唯一性标识。比如:原本是,〈div〉〈/div〉,经过标识可能是〈div[3]〉〈/div[3]〉。

2 融合内容和结构特征的多类型网页文本要素的抽取

2.1 网页标题要素的发现

先将网页头部中,标题元素的标签内容作为候选网页标题保留下来,称之为 WebHeadTitle。比如:〈title〉交通部:让专车推动传统出租车改革|出租车|专车_新浪财经_新浪网〈/title〉,〈title〉广州日报—2015年11月8日—A1:头版版—习近平同马英九会面〈/title〉。其中“交通部:让专车推动传统出租车改革”和“习近平同马英九会面”分别是两个网页的标题。利用非中文字符(比如上面的“|”、“_”和“—”字符)将

WebHeadTitle 分割成 headCollection={Str1,Str2,…}文本集合,并且 hcSize=Str1. length+Str2. length+…(hcSize 为文本集合中元素的总字符长度,Str1. length 为 Str1 字符长度),同时在网页体中,会有 N(0<N<集合元素个数)个文本节点的字符串与 headCollection 中 P(P<N)个元素相同,不过在这 P 个元素中,仅仅只有一个是网页标题。具体算法如下所述。

获得 headCollection 之后,这时有两种情况,一种是 headCollection 中只有一个元素;另一种情况是 headCollection 有多于一个元素。对于第一种情况,遍历 DOM 树,选出字符长度等于 hcSize 的文本节点,然后将这些文本节点的字符与 Str1 比较,全部相同的即为网页标题。对于第二种情况,先排除文本长度大于 hcSize 的文本节点,余下的文本节点组成集合 nodeCollection={Node1,Node2,…}。然后将 totalStr 和 Node1,Node2,…进行字符预处理,在整个抽取标题的过程中共有两次处理。

处理 1:开始抽取之前,去掉两者文本中所有的空格、大小写字母、数字、百分号、小数点、右括号和加号等常用的标点符号。

处理 2:在计算两者的文本长度之比之前,仅去掉原文本中的空格。算法主要步骤总结如下:

```
算法:网页标题抽取算法。
输入:{Str1,Str2,…}和{Node1,Node2,…}集合
输出:网页标题
步骤:
1: 执行处理 1
2: for head=Str1,Str2,… do
3:   for node=Node1,Node2,… do
4:     if(compareNE(head, node)) then continue;
5:     else { if(tag(node,'a')) then remove(head) ;break;
6:       if(contain(node,next(Str)) then join(Str);}
7: for filterHead=headCollection do
8:   执行处理 2
9:   maxSort (filterHead/hcSize) then return filterHead;
```

其中,compareNE(head, node)用于判断 head 与 node 字符不一致,tag(node,'a')判断节点的标签是否为'a',remove(head)表示从 headcollection 中删除当前元素,next(Str)表示集合中下一个元素,contain(node,next(Str))表示 next(Str)字符串是否是当前 node 字符串的一部分,join(Str)连接这几个 Str 字符串并从原集合中删除它们,然后向集合中插入新连接的 Str, maxSort(filterHead/hcSize)将这两者的比值排序并取出最大值对应的字符串。

2.2 多类型网页正文要素的提取

多类型网页按照正文部分的是否在一个特定的容器标签(此标签的判别后面有论述)中分为正文连续型网页和正文非连续型网页。本文提出的网页正文要素提取方法的主要内容是:先将网页正文要素提取看作一个二分类问题,即正文文本节点与非正文文本节点的分类问题,利用标题节点与其余节点的上下位置关系、标题实词在文本节点中出现的次数等,结合正文区域在网页内容和结构上的特征,再利用支持向量机,选择径向基(RBF)核函数(实际中使用 LIBSVM,其余参数默认)训练模型获得正文候选节点,然后在这些候选节点中选出一个最有可能属于正文部分的节点,即参考节点(referenceNode),再根据特定的容器标签,扩展参考节点成参考块(referenceBlock),同时扩展余下的正文候选节点,形成多个正文候选块,再利用是否与参考块有包含或者交集关系,对正文候选块进行整合,为了去除噪音,最后计算筛选出来的候选块(包括参考块)中超链接文本密度、标签属性密度,引入标题影响因子、标点符号和标签数量影响因子,将这些值加权求和,取值最小的候选块为正文区域。

实际操作中首先将 DOM 树的各个节点特征化和标准化,即转化成向量表示,构成的向量集作为 SVM 的训练数据,利用得出的模型标注待测 DOM 树节点是否属于正文。根据实验效果确定如表 1 所示的正文多个特征。

表 1 多特征描述

Table 1 Description of multi-features

特征	特征描述
文本长度(TL)	节点的字符总长度
标点符号数量(PN)	节点的标点符号数量
两节点的相对距离(ND)	两个字符长度大于零的节点,它们间隔字符长度为零的节点数量,如〈Li〉字〈/Li〉〈p〉〈a〉〈/p〉,a 节点字符长度为 2,p 为 0,Li 为 1,则 Li 与 a 节点相对距离为 1
标题实词在节点的文本中出现次数(TDTN)	先将标题分词,提取名词、动词、成语、机构团体、时间和简称略语等实词组成集合,再统计集合元素在本节点中出现次数
节点中的文本单句长度(TAL)	节点的字符长度/标点符号数量
节点的标签特性(TP)	标签为“a”,则此特征值记为“A”,若是“div”、“table”、“p”、“tbody”,则记为“D”,其余全为“S”
节点与标题的位置关系(NTP)	位置在标题之上的节点为“U”,在标题之下的记为“D”,节点的内容是标题的记为“B”

统计 ND 这个特征也是为了解决 SVM 模型输出的向量与 DOM 树节点的对应问题,某向量在 DOM 树中对应节点的位置,是此向量与之前全部向量的本特征值之和,这样下一步的工作就又建立在 DOM 树之上了。

通过模型标注出节点的类别:N 或 Y,N 表示非正文,Y 表示正文,将标注后的文档记作 DOC。然后要注意将各块中标签为〈a〉的子孙标签也改为〈a〉,统计链接文本数量时就不会产生遗漏。

referenceNode 的选取方法是:①收集并统计 DOC 中被标记为 Y 的节点。如果统计个数为零则结束抽取,如果个数为 1,此节点即为 referenceNode,如果节点个数大于 1,则将这些节点按照 TDTN 特征值降序排列;②取出排列好的前两个节点,若第一个节点与第二个节点的 TDTN 特征值之比大于 3,将第一个节点设为 referenceNode,反之③利用公式

$$F = L \times 0.3 + M \times 0.7 \tag{1}$$

(其中,L 为文本长度,M 为标题实词出现次数)计算两个节点的得分 F_1 和 F_2,得分最高的为 referenceNode,得分相同的,则取 PN 值最大的为 referenceNode,如果 PN 值都相同,则选择第一个节点作为 referenceNode。

特定容器标签的选取和正文候选节点扩展方法:

① 如果是正文连续型网页,特定容器标签为〈div[?]〉(问号表示唯一标识的数字)标签且存在标签属性 class 或者 id;也可以为〈table[?]〉标签,对于此标签不关心有没有属性。然后以 referenceNode 为基础,不断向上遍历父节点、祖父节点等,直到找出特定容器标签的节点停止,这个节点及其子孙节点中包含的所有内容即为 referenceBlock;② 如果是正文非连续型网页,必须约束条件,〈div[?]〉的标签属性仅存在 class 或者不能仅仅只有“id”或者“style”属性,〈table[?]〉标签的规则同上;或者仅存在〈li[?]〉标签,然后以 referenceNode,为基础,不断向上遍历父节点、祖父节点等,直到找出特定容器标签的节点停止。

正文连续型网页候选块整合方法:① 为了提高抽取效率,先去掉所有被 referenceBlock 包含的正文候选节点,再按照上述 referenceNode 扩展方法扩展余下的候选节点,形成多个正文候选块(candidateBlocks);② 如果某个 candidateBlocks 包含 referenceBlock 或者某些 candidateBlocks 与 referenceBlock 有交集,则舍去这些 candidateBlocks,保留其余的 candidateBlocks。

正文连续型网页确定正文区域的方法:在 referenceBlock 和保留的 candidateBlocks 计算各块中的密度和因子值。

$$\mathrm{AD}=\mathrm{AL}/\mathrm{SumL}\ , \tag{2}$$

其中,AL 为该块中超链接字符的长度,SumL 该块的总字符长度,AD 为超链接文本密度。

$$\mathrm{TD}=\mathrm{PNum}/\mathrm{SumL}\ , \tag{3}$$

其中,PNum 是标签中属性的个数,SumL 同上,TD 为标签属性密度。

以下分别是标题(TIF)、标点符号(PIF)和标签数量(TNIP)影响因子:

$$\mathrm{TIF}=\begin{cases} 0.7 & \mathrm{TDTN\_block}=0 \\ 1.0/\mathrm{TDTN\_block} & \mathrm{TDTN\_block}>0 \end{cases}, \tag{4}$$

其中,TDTN_block 为该块的标题实词出现次数。而 PIF 和 TNIP 计算方法与 TIF 相同,但当计算 TNIP 时将值 0.7 设定为 0.5。

最后,利用公式5将密度和影响因子加权求和,取其最小值对应的文本块为正文区域。

$$sort=1.4*AD+0.8*(TD+TNIF)+1.3*(TIF+PIF) \quad (5)$$

正文非连续型网页的正文内容确定方法:主要是利用正文非连续型网页的正文块一般会有相似的网页结构。首先,去掉referenceBlock对应特定容器标签的"id"属性,同时与去掉"id"属性的其他特定容器标签相比较,按顺序将相同的特定容器标签收集起来(包括子孙标签及其内容),如{pDiv1,pDiv2,…},然后在这些特定容器标签搜索其子孙标签"li"、"tbody",按顺序将这些标签及其内容收集起来,如{pDiv1 $ child1, pDiv1 $ child2,…, pDiv2 $ child1, pDiv2 $ child2,…}(pDiv1 $ child1表示第一个特定容器标签中满足上述搜索规则的第一个子孙标签及其内容),按这样的顺序获取集合元素的文本内容,就是正文非连续网页的正文内容。

2.3 提取发布时间要素

根据国内网页的展示风格,一般网页发布时间的位置都是在网页标题和网页正文之间,所以在前面已经获得标题信息和正文区域信息的前提下,仅需将两者之间的文本节点与表示时间的正则表达式:20[0-9]{2}[\\s\\S][0-1]?[0-9]{1}[\\s\\S][0-3]?[0-9]{1}([\\s\\S]{1,2}[0-2]\\d[\\s\\S][0-5]\\d([\\s\\S][0-5]\\d)?)*匹配即可实现发布时间要素的提取。

3 实验与结果分析

为了验证本文方法的可行性,实验中选取了12个网站,其中正文连续型网站8个包括四大门户新闻网站(新浪、腾讯、网易、搜狐)、云南网、东方财富网,两大博客包括新浪博客、凤凰博报。4个正文非连续型网站包括天涯论坛、百度贴吧、大众点评和中关村手机。

对于两种不同类型的网站,有不同的评价标准,但都是通过人工观察网页确定标题、发布时间和正文内容,然后与本文方法提取的结果对比。

将以下情况视为正确抓取:

(1)标题内容没有多字、缺字、误抓和没抓现象。

(2)发表时间没有多字、缺字、误抓和没抓现象。

(3)对于正文连续型网站,正文抽取与人工观察的结果内容、顺序完全一致,或者是正文抽取结果仅仅只包含不超过一句话的长度的非正文内容。对于正文非连续型网站,凡是发帖人发的信息都认为是正文内容。

正文连续型网站每个网站随机选取了400个网页,对于正文非连续型网站每个网站随机选取200个网页,实验结果如下表2所示。评价抽取的结果用提取数据的准确率(Z)表示,计算公式如下,分别为抽取标题、发布时间和正文的准确率。

表2 网页文本要素提取实验结果

Table 2 Experimental result of extraction textual elements of webpages

网站	网页总数(C)	正确抓取标题数(C_t)	标题准确率(Z_t)	正确抓取时间数(C_m)	时间准确率(Z_m)	正确抓取正文数(C_e)	正文准确率(Z_e)
腾讯网	400	372	93%	385	96.25%	393	98.25%
新浪网	400	364	91%	392	98 %	392	98%
网易	400	372	93%	380	95%	395	98.75%
搜狐	400	366	92.5%	395	98.75%	396	99%
云南网	400	362	90.75%	374	93.5%	381	95.25%
东方财富网	400	358	89.5%	374	93.5%	377	94.25%
新浪博客	400	380	95%	394	98.5%	394	98.5%
凤凰博报	400	383	95.65%	395	98.75%	395	98.75%
百度贴吧	200	195	97.5%	—	—	187	93.5%
天涯论坛	200	194	97%	—	—	188	94%
大众点评	200	188	94%	—	—	194	97%
中关村手机	200	187	93.5%	—	—	199	99%

$$Z(t)=\frac{C(t)}{C}\times 100\%\ ; \tag{6}$$

$$Z(m)=\frac{C(m)}{C}\times 100\%\ ; \tag{7}$$

$$Z(e)=\frac{C(e)}{C}\times 100\%\ . \tag{8}$$

其中,C 表示实验的网页总数,$C(t)$表示正确抓取标题的网页个数,$C(m)$表示正确抓取发布时间的网页个数,$C(e)$表示正确抓取网页正文的网页个数。

为比较正文的提取效果,选取基于统计理论抽取正文的方法进行对比[10],对比结果见表 3。

表 3 不同方法提取网页正文的实验对比结果

Table 3 Comparison of experimental result of different methods for extracting webpages text

网站 \ 方法	对比方法	本文方法
腾讯新闻	90.5%	98.25%
搜狐新闻	91.75%	99%
新浪博客	93.5%	98.5%
云南网	89%	95.25%
天涯论坛	91%	94%

从表 2、表 3 的实验数据可以看出,本文提出的方法抽取标题、发布时间和正文内容等要素有着较高的准确率,特别是正文中存在图片,文本内容少的网页也能较准确地抽取出来。根据网页结构和内容选取的多特征和有针对性规则的制定,对抽取结果有很大的帮助,且提取方法可以达到实用效果。

4 结语

依据网页的结构和内容特点,提出了融合网页结构与内容的特征,自动提取标题、时间及正文三要素,实验的结果也证明了方法的有效性。进一步工作将考虑深挖网页文本三要素与网页中其他部分在内容与结构方面的关联,特别是寻找效果更好的正文候选块的分块方法,这些研究内容都有继续改进的空间。

参考文献:

[1] 陈钊,张冬梅. Web 信息抽取技术综述[J]. 计算机应用研究,2010,(12):4402-4405. DOI:10.3969/j.issn.1001-3695.2010.12.001.

[2] 黄健斌,姬红兵,孙鹤立. Web 网页中动态数据区域的识别与抽取[J]. 计算机工程,2007,(11):53-58.

[3] Kohlschutter C,Fankhauser P,Nejdl W. Boilerplate Detection Using Shallow Text Features[C]. Proc of the 3th ACM International Conference on Web Search and Data Mining. New York,USA,2010:441-450.

[4] 杨柳青,李晓东,耿光刚. 基于布局相似性的网页正文内容提取研究[J]. 计算机应用研究,2015,(9):2581-2586. DOI:10.3969/j.issn.1001-3695.2015.09.005.

[5] Cunhe Li,Juan Dong,Juntang Chen. Extraction of Informative Blocks from Web Pages Based on VIPS[J]. *Journal of Computational Information System*,2010:271-277.

[6] 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.

[7] 欧健文,董守斌,蔡斌. 模板化网页主题信息的提取方法[J]. 清华大学学报(自然科学版),2005,**45**(S1):1743-1747.

[8] 耿焕同,宋庆席,何宏强. 一种基于视觉分块 Web 信息抽取方法研究[J]. 情报理论与实践,2009,(3):106-109.

[9] 朱德泽,李淼,张建,等. 基于文本密度的 WEB 正文抽取[J]. 模式识别与人工智能,2013,(7):668-672.

[10] Pasternak J,Roth D. Extracting Article Text from the Web with Maximum Subsequence Segmentation[C]. Proc of the 18th Interna-tional Conference on World Wide Web. Madrid,Spain,2009:971-980.

[11] 秦成磊,魏晓,杨阳. 一种基于统计的复杂页面正文提取方法[J]. 计算机应用与软件,2015,(7):90-93. DOI:10.3969/j.issn.1000-386x.2015.07.021.

[12] 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,(7):47-49. DOI:10.3969/j.issn.1000-386x.2014.07.013.