# 基于多特征融合的网页正文信息抽取

刘 利 戴 齐 尹红风 贾 真 胡万亭
(西南交通大学信息科学与技术学院,思维与智慧研究所 四川 成都 610031)

**摘 要** 当今主流网页分为单正文体网页和多正文体网页。这些网页的正文信息都具有多个正文特征。想要准确定位正文信息所在位置,可以从其所具有的多个特征和网页设计者的设计习惯着手。鉴于此,融合这些特征提出一种基于多特征融合的网页正文信息抽取方法。实验结果表明,该方法对单正文体网页和多正文体网页的正文抽取具有较高的准确率和通用性,很好地适应了风格多样的网页。

**关键词** 单正文体 多正文体 多特征 信息抽取

**中图分类号** TP391 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2014.07.013

# EXTRACTING WEBPAGES TEXT INFORMATION BASED ON HETEROGENEOUS FEATURES FUSION

Liu Li Dai Qi Yin Hongfeng Jia Zhen Hu Wanting
(*Institute of Noetics and Wisdom*, *School of Information Science and Technology*, *Southwest Jiaotong University*, *Chengdu 610031*, *Sichuan*, *China*)

**Abstract** Nowadays, the mainstream webpages are divided into single text body webpages and multiple text body webpages. These webpages text information all have the heterogeneous text features. In order to accurately position the location of text information, one can commence from the heterogeneous features they possess and the design habits of the web designers. Therefore, in this paper, we fuse these features and present a webpages text information extraction method which is based on heterogeneous features fusion. Experimental results show that the method has higher accuracy rate and universality for extracting the texts from single test body webpages and multiple text body webpages, fits well the webpages with a variety of styles.

**Keywords** Single text body Multiple text body Heterogeneous features Information extraction

## 0 引 言

随着互联网技术的快速发展和信息的日益膨胀,不仅当今的互联网充斥着大量的垃圾信息,而且 Web 网页已经不再像以往的网页内容简洁、风格简单。网页里面加入了很多元素比如显示样式,脚本和大量的广告等等噪声信息。如何从众多的垃圾信息中找到有用信息? 如何在网页中准确并完整地找到主题信息所在的位置? 成为当今研究的热点课题。

Web 网页正文信息提取领域,已经有大量的研究工作和许多比较成熟的方法,要求抽取的网页数据源来自于同一网站或者网页结构相似的主要有基于网页模板的方法[1-3],基于 DOM 树结构及其它延伸的方法[4,5],很多研究者还把这两种方法相结合进行信息抽取,比如 RoadRunner 系统[6];抽取的网页数据源不局限于同一网站的主要有基于视觉特征的方法[7,8],基于统计理论的方法[9,10]等。在实际应用中可根据面向数据源的不同选取不同的方法,并且很多方法都取得了不错的实验结果。

基于统计理论的方法中,用网页正文特征来确定正文信息位置的方法主要有:Song 等人[11]利用正文信息常见的三个特征(即:标点符号,非超链文本和超链文本),将这些特征转化为统计信息值,以此确定正文信息的位置;周等人[12]延续了 Song 的方法并在后续处理过程中做了改进,提出 SCF 方法进一步提高了抽取的效果,更好地适应了风格多样的网页;李连霞等人[13]总结了网页的多个特征,利用统计概率的方法确定正文信息的位置。

在实际应用中现今的基于统计理论的方法有其局限性,随着网页风格的多样化,抽取准确率有所降低,通用性不强。本文旨在开发一个面向实际应用的、针对不同类型网页的正文信息提取及其结构化的系统,该系统是"基于人机共建智慧平台的语义智能搜索引擎"项目中的一个子系统(辅助扩展搜索引擎的后台知识库,以及前台用户搜索时及时反馈结构化的网页正文信息提高用户体验),该系统尽可能适用于不同风格类型网页和任意网站。抽取结果的高准确率以及通用性是设计网页正文信息抽取算法的难点。作者以百度百科、互动百科以及各知名导航网站里面的网址为基础不断往外延伸共爬取了五亿多个 URL,下载了三千多万的网页,以此为实验数据源研究具有较高准确率、通用性较强的网页正文信息抽取方法。为了满足网页风格的多样性和算法本身的通用性,提出了一个基于多特征的

收稿日期:2012-12-04。国家自然科学基金项目(61152001,61170111);中国科学院自动化研究所复杂系统管理与控制重点实验室开放课题(20110102)。**刘利**,硕士,主研领域:数据挖掘,机器学习,人工智能。**戴齐**,副教授。**尹红风**,教授。**贾真**,讲师。**胡万亭**,硕士。

网页信息抽取方法，即 WIEHF（Webpages Information Extraction based on Heterogeneous Features）方法。

## 1 WIEHF 方法

### 1.1 基本概念

现今主流网站网页的设计思想都是按照块进行设计的，把某个模块的内容放到一个块中，而 WIEHF 方法的思想也按照网页设计的习惯，先对网页进行分块，然后再对每个块进行正文识别，算出最可能包含正文信息的块并提取出正文信息。在网页设计中常用的能进行分块的标签有 < div > < table > < tbody > 标签（后面称为容器标签）等。以下给出容器标签等相关概念的定义。

**定义 1** 标签里面能嵌套其他标签和正文信息的标签称为容器标签。

**定义 2** 单正文体网页是指网页正文信息集中在一个容器标签中。

**定义 3** 多正文体网页是指网页正文信息分布在多个容器标签中。

### 1.2 多特征的描述

对大量网页的分析发现，网页主要分为单正文体网页和多正文体网页。

单正文体网页的正文特征主要是文本集中在一个容器标签中，里面包含很多标点符号，文本里面具有对标题描述的语言，而且较多数其他容器标签而言更靠近标题标签。

多正文体网页的正文特征主要是文本分布在多个容器标签，而且根据网页设计的视觉习惯，这些容器标签的显示风格极有可能是一样的，里面可能也包含很多标点符号，文本里面也具有对标题的少量描述语言，靠近标题标签。

综上对不同类型网页正文特征的分析和描述，网页正文所具有的多个特征包括：正文文本数量、正文标点符号、正文超链接文本和非超链接文本的关系、正文对标题的描述性语言、正文离标题的远近以及正文信息显示的样式和位置。

融合这些网页正文特征而提出的 WIEHF 方法是先将 html 文档转换成 DOM 树，然后计算每个容器标签的正文支持度；同时伴随着计算过程中和计算后的一些处理，详见 1.3 节中算法步骤具体描述。

本文通过 Jsoup. jar 第三方 jar 包是实现对网页 DOM 树的构建，该 jar 包实现的功能是先对网页的缺省标签进行修复，然后解析 html 文档以 html 标签为根节点遍历所有的标签建立 DOM 树，如图 1 所示。

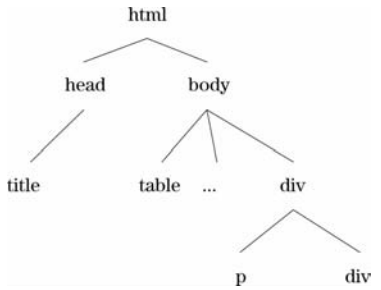


图 1 网页 DOM 树

在遍历 DOM 树时，通过对各个容器标签进行唯一性的标注来实现每个容器标签位置路径的唯一性，比如通过对图 1 的 DOM 树标注以后得到结果如图 2 所示。

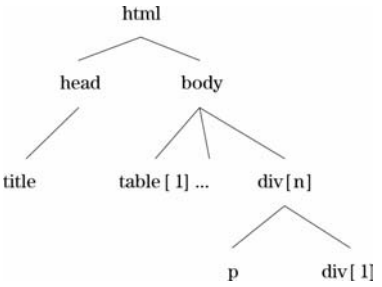


图 2 标注后的 DOM 树

我们可以得到每个容器标签惟一的路径分别为 html/body/table[1]、…、html/body/div[n]、html/body/div[n]/div[1]，以及 title 标签的路径 html/head/title。

计算每个容器标签的正文支持度（*SD*），它的计算思想是将正文特征分为三类，第一类是正文离标题的远近，借此算出距离支持度（*DSD*）；第二类是正文对标题的描述性语言，借此算出标题支持度（*TSD*）；第三类是正文文本数量、正文标点符号、正文超链接文本和非超链接文本的关系，借此算出一般支持度（*PSD*）。然后利用这些支持度算出总的正文支持度即可确定最有可能是正文的容器标签，计算方法如下：

$$SD = DSD \times (TSD + PSD) \tag{1}$$

其中，*SD* 为正文支持度，*DSD* 为距离支持度，*TSD* 为标题支持度，*PSD* 为一般支持度。

*DSD* 的计算思想是：距离标题标签越近的容器标签就越有可能支持它是包含正文的容器标签。具体做法是将所有容器标签和 title 标签映射到一维坐标系中，在计算距离支持度时假设 title 为坐标系上的原点，容器标签可用式(2)进行映射和转换成坐标系上的点，即可算出距离支持度。

$$DSD = 1/\left(\sum_{i\in n}^{n} rd_i \times q^{i-1}\right) \quad n \geqslant 1 \tag{2}$$

其中，$rd_i$ 为容器标签路径上的序号，$q$ 是大于 0 的一个整型常数，在实验过程中发现 $q = 10$ 比较合理，比如：如图 1 的某个容器标签路径 html/body/div[1]/div[1]，$rd_1 = 1$，$rd_2 = 1$，则 $DSD = 1/(1 \times 10^0 + 1 \times 10^{-1}) = 1/1.1$。

*TSD* 的计算思想是：容器标签中包含标题的实体词越多就越有可能支持它是包含正文的容器标签，具体做法是先将标题分词（本文采用的分词系统是西南交通大学耶宝智慧中文分词、词性标注和实体标注一体化系统 http://www. yebol. com. cn），提取出里面的实体词，然后统计在所有容器标签中总的出现次数，选取出现次数最多的两个词（FirstWord 和 SecondWord），用式(3)求得标题支持度。

$$TSD = \alpha \times FW + \beta \times SW \tag{3}$$

其中，*FW* 为 FirstWord 在容器标签中出现的次数，*SW* 为 SecondWord 在容器标签中出现的次数，$\alpha$ 和 $\beta$ 是两个常数，并且 $\alpha < \beta$，为了平衡算出的正文支持度和考虑到这两个词的重要性程度，实验中它们的取值是 $\alpha = 0.5$，$\beta = 1$。

*PSD* 的设计思想是：每个网页里面都具有普遍特征，即标点符号超、链接文本、非超链接文本。通过下面的计算方法，建立起它们之间的关系并算出其对所在容器标签的支持度，具体公式如下：

$$PSD = FP \times (NC/HC) \tag{4}$$

$$FP = \begin{cases} 0 \leqslant p < 3 & FP = 0.001 \\ 3 \leqslant p < 6 & FP = 0.1 \\ p \geqslant 6 & FP = 0.5 \end{cases} \tag{5}$$

其中,*NC* 为非链接文本的字数,*HC* 为链接的文本的字数,*FP* 为标点符号支持度,*p* 为标点符号个数。标点符号越少是正文的可能性就越小,所以就减小其对自己是正文的信息支持。

由于在 1.3 节中会涉及到路径距离的计算,所以在此先介绍它的相关概念和计算方法。

**定义 4**　路径距离是指容器标签之间距离远近的一种描述。

$$juli(i,j) = (len(i) - pre(i,j)) + (len(j) - pre(i,j)) - 1 \quad (6)$$

其中,$juli(i,j)$是容器标签 $i$ 和容器标签 $j$ 之间的路径距离,$len(i)$是根节点到容器标签 $i$ 所经过的节点个数,$len(j)$是根节点到容器标签 $j$ 所经过的节点个数,$pre(i,j)$是到容器标签 $i$ 和到容器标签 $j$ 所经历的相同节点个数。比如:容器标签 $i$ 的路径为 html/body/div[1]/div[2]/div[2]和容器标签 j 的路径为 html/body/div[1]/div[2]/div[3],此时 $len(i)=5$,说明从根节点到标签 $i$ 时要经过 5 个节点才能到;同样可算 $len(j)=5$,$pre(i,j)=4$,最后可算出 $juli(i,j)=1$。又比如:若有标签路径 html/body/div[1]/div[1]/div[1]/div[0]和 html/body/div[1]/div[1]/div[2]/div[1]的路径距离为 3。

### 1.3　算法步骤

WIEHF 方法的具体处理步骤描述如下:

输入:某个网页的源代码

输出:抽取信息的结果集

步骤:

1) 清除 script、meta、style 等噪声标签,保存 title 标题标签和子标题标签。

2) 遍历网页 DOM 树,将容器标签依次抽取出并以 key-value 的形式保存,其中 key 为标签的路径,value 为容器标签包含的内容。以容器标签为单位,用式(4)计算 *PSD*。

3) 对 title 标签和各级子标题标签进行分词,统计出现次数最多的两个词(FirstWord 和 SecondWord)。用式(3)计算出每个容器标签的 *TSD* 值。

4) 再次遍历容器标签集合,用式(2)计算每个容器标签的 *DSD*,并同时用式(1)计算出每个容器标签的 *SD*。

5) 利用式(6)计算容器标签之间的路径距离。当 $juli=1$ 时,比较这两个容器标签的 class 属性,若没有则比较他们的 style 属性及其他属性,如果属性相同,则对它们的内容进行合并,并将它们的 *SD* 值进行相加。

6) 选取前七个 *SD* 值的容器标签,选取最大 *SD* 值的容器标签,其文本长度在前七个容器标签文本长度的总长度中占的比例,与 0.5 相比,若大于等于 0.5 则设路径距离阈值 $JULI=2$,若小于 0.5 则设路径距离阈值 $JULI=4$。

7) 计算前七个容器标签到最大 *SD* 值的容器标签的路径距离 *juli*,如果 $juli \leqslant JULI$ 则将此容器标签加入到最后要返回的集合 lastset 中。

8) 对 lastset 里面的元素进行遍历,若 lastset 里面的元素只有一个,则直接返回,若不是则将对里面的元素挨个进行判断。判断方法如下:如果元素的内容里出现了 3 个及以上词库(版权信息的词库)里面词的并且无标点符号,则将对该元素直接舍弃。

## 2　实验验证和结果分析

实验选取 5 个单正文体类型的网站和 5 个多正文体类型的网站,分别是:sina、sohu、tom、pharmnet、chinayy 和搜狐人物频道、百度知道、百度贴吧、智联招聘、ubuntu。在这些网站中分别随机选取了 200 个网页,实验结果见表 1 所示。对于结果用提取正文信息的准确率($P$)和完整率($R$)进行评价,它们计算公式如下:

$$P = \frac{C2}{C1} \times 100\% \quad (7)$$

$$R = \frac{C3}{C2} \times 100\% \quad (8)$$

其中,$C1$ 表示实验的网页总数,$C2$ 表示正确提取正文信息的网页个数,$C3$ 表示完整提取正文信息的网页个数。准确率是以网页总数为前提,完整率是以正确提取正文信息的网页个数为前提。

**表 1　实验结果**

| 网页来源 | 网页总数 | 正确提取个数 | 正确率 | 完整提取个数 | 完整率 |
|---|---|---|---|---|---|
| Sohurenwu | 200 | 195 | 97.5% | 191 | 97.95% |
| Pharmnet | 200 | 194 | 97% | 178 | 91.75% |
| Sohu | 200 | 194 | 97% | 188 | 96.9% |
| Chinayy | 200 | 196 | 98% | 194 | 98.98% |
| Baidutieba | 200 | 193 | 96.5% | 184 | 95.33% |
| Sina | 200 | 194 | 97% | 190 | 97.94% |
| Tom | 200 | 198 | 99% | 172 | 86.87% |
| Zhilianzhaopin | 200 | 180 | 90% | 175 | 97.22% |
| Ubuntu | 200 | 176 | 88% | 172 | 97.73% |
| Baiduzhidao | 200 | 191 | 95.5% | 175 | 91.62% |

由于 Song 等人和周等人也是利用网页正文特征确定正文信息的位置,和文本的方法属同类,但在处理方式上有所不同。所以表 2 中加入他们的实验结果进行对比。

**表 2　实验对比结果**

| 方法 / 网页来源 | SONG | SCF | WIEHF |
|---|---|---|---|
| Sohurenwu | -- | -- | 97.5% |
| Pharmnet | 89% | 93% | 97% |
| Sohu | 96% | 95% | 97% |
| Chinayy | 95% | 95% | 98% |
| Tom | 96% | 93% | 99% |
| Sina | 95% | 94% | 97% |
| Baiduzhidao | -- | 94% | 95.5% |
| Ubuntu | -- | 88% | 88% |
| Zhilianzhaopin | -- | -- | 90% |
| Baidutieba | -- | -- | 96.5% |

从表 2 中可以看出,在单正文体网页和多正文体网页的抽取准确率上较以往的同类方法要高。

(下转第 77 页)

从以上实验结果的分析比较,将 JUnit 测试框架迁移到 Hadoop 平台上的 HadoopUnit 测试框架是一个很好的解决方案。首先随着回归测试包变得越来越大且不能交互时,基于 Hadoop 平台的 HadoopUnit 的优势就越明显。从图 4 可以看出,当测试用例数据量小时,并行运算的总时间反而大于单机执行的时间,是由于启动 Hadoop 集群和运行作业时需要传输中间文件和生成最终文件需要耗费一定的时间。但随着测试用例数量的增多,基于 Hadoop 的分布式平台将测试用例分割后分派给多个节点并行处理,使并行处理的总时间小于单机执行的时间,且随着测试用例集的增加,两者执行效率的差距也越来越大。

## 4　结　语

由于软件规模和复杂程度的增加,软件测试在执行时间和耗费资源上面临着巨大挑战。针对目前仅停留在单机平台上的软件测试耗费大量的资源问题,本文在对云计算的 Hadoop 集群框架研究的基础上,把 JUnit 框架迁移到 Hadoop 平台上,提出了一种基于云计算环境下的分布式测试框架 HadoopUnit。该框架利用 Hadoop 的并行处理技术 MapReduce 重构了 JUnit 测试框架,实验结果表明,由于测试用例的分布式计算,获取了更快的执行速度。基于云计算的软件测试提高了软件测试的效率,对软件测试领域有一定的指导意义。

## 参　考　文　献


[1] Gamma E, Beck K. JUnit[OL]. http://www.junit.org.

[2] 戴建国,赵庆展,郭理,等. 持续集成在项目开发中的应用研究[J]. 计算机工程与设计,2009,30(10):2573-2576.

[3] 隋智泉. 一种改进的单元测试 JUnit 框架[J]. 电脑知识与技术,2007,(08):478-479.

[4] 王晓卓. JUnit 框架的改进及其应用研究[D]. 大连:大连海事大学计算机学院,2008.

[5] 胡元甲,洪玫,黄梅,等. 分布式测试平台任务管理子系统的设计与实现[J]. 计算机工程与设计,2011,32(3):958-962.

[6] PoweredBy-HadoopWiki[EB/OL]. [2009-11-17]. http://wiki.apache.org/hadoop/PoweredBy.

[7] HDFS Architecture[EB/OL]. [2008-12-10]. http://hadoop/apache.org/core/docs/ current/hdfs_design.html.

[8] Dean J. Ghemaw at S. MapReduce:Simplified Data Processing on Large Clusters[J]. Communications of the ACM,2005,51(1):107-113.

[9] Map/reduce tutorial[EB/OL][2009-11-17]. http://hadoop/apache/org/common/docs/current/mapred_tutorial.pdf.

[10] 51testing 软件测试网[EB/OL]. [2012-01-15]. http://www.51testing.com/?287554.


(上接第 49 页)

图 3 为对应表 2 的直观条形图。

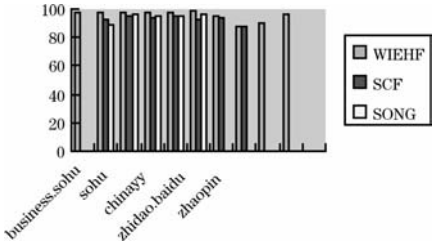


图 3　实验结果对比

在图 3 中,可以直观发现本文方法的较高准确率。更多更完善的网页正文特征因素加入到网页信息抽取方法中,对抽取效果有很大帮助,对单正文体类型的网页和多正文体类型的网页的抽取结果都有较高的正确率。

## 3　结　语

由于本文项目组研究课题的关系,该方法是建立在较大规模训练集上的,实验结果和实际应用相差不大,可以满足一定的科学研究和实际应用。对于多正文体的网页的抽取效果较单正文体稍差,分析其原因在于网页结构中标签嵌套很深或者是它们显示样式差别大并且距离相差很远,对于网页正文特征还有很多,尤其是多正文体还具有其自身的独特性,所以在后期中还将深入研究,算法有继续改进的空间。

## 参　考　文　献


[1] 刘辉,陈静玉,徐学洲. 基于模板流程配置的 Web 信息抽取[J]. 计算机工程,2008(20):55-57.

[2] 冀高峰,汤庸,道炜,等. 基于 XML 的自动学习 Web 信息抽取[J]. 计算机科学,2008(03):87-90.

[3] 郑长松,傅彦,佘莉. 基于模板的 Web 信息自动提取方法[J]. 计算机应用研究,2009(2):570-572.

[4] Wang Jiying, Lochovsky F H. Data-rich section extraction from HTML pages[C]//Proc of the 3rd International Conference on Web Informations Systems Engineering. Washington DC: IEEE Computer Society, 2002:2313-2322.

[5] 刘亚东,彭舰,张达平. 基于智能的网页信息抽取系统的研究与设计[J]. 四川大学学报:自然科学版,2009,46(4):957-962.

[6] Crescenzi V, Mecca G. RoadRunner: towards automatic data extraction from large Web sites[C]//Proc of the 27th VLDB Conference. San Francisco: Morgan Kaufmann Publishers,2001:109-118.

[7] Deng Cai, Shipeng Yu, Jirong Wen, et al. VIPS: a vision-based page segmentation algorithm[R]. Microsoft Technical Report. MSR-TR-2003-79, November 2003.

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

[9] Gupta S, Kaiser G. DOM-based content extraction of HTML documents[C]//Proc of the 12th World Wide Web Conference. New York: ACM Presss,2003: 207-214.

[10] 孙承杰,关毅. 基于统计的网页正文信息抽取方法的研究[J]. 中文信息学报,2004(5):17-22.

[11] Song Mingqiu, Wu Xintao. Content extraction from Web pages based on Chinese punctuation number[C]//Proc of International Conference on Wireless Communications, Networking and Mobile Computing, Sempt 2007.

[12] 周佳颖,朱珍民,高晓芳. 基于统计与正文特征的中文网页正文抽取研究[J]. 中文信息学报,2009(5):80-85.

[13] 李连霞,马军,陈竹敏. 基于多特征的网页内容提取研究[C]//第三届和谐人机环境联合学术会议,2007.