第 36 卷 第 6 期 计 算 机 工 程 2010 年 3 月
Vol.36 No.6 Computer Engineering March 2010
·软件技术与数据库· 文章编号：1000—3428(2010)06—0102—03 文献标识码：A 中图分类号：TP393

基于内容相似度的网页正文提取

王 利[1]，刘宗田[1]，王燕华[2]，廖 涛[1]
(1. 上海大学计算机科学与工程学院，上海 200072；2. 上海海洋大学信息学院，上海 201306)

摘 要：提出一种将复杂的网页脚本进行简化并映射成一棵易于操作的树型结构的方法。该方法不依赖于 DOM 树，无须用 HTMLparser 包进行解析，而是利用文本相似度计算方法，通过计算树节点中文本内容与各级标题的相似度判定小块文本信息的有用性，由此进行网页清洗与正文抽取，获得网页文本信息，实验结果表明，该方法对正文抽取具有较高的通用性与准确率。

关键词：网页正文抽取；网页映射；网页清洗；文本相似度

Web Page Main Text Extraction Based on Content Similarity

WANG Li[1], LIU Zong-tian[1], WANG Yan-hua[2], LIAO Tao[1]
(1. School of Computer Science and Engineering, Shanghai University, Shanghai 200072;
2. School of Information Technology, Shanghai Fisheries University, Shanghai 201306)

【Abstract】 This paper proposes a method of simplifying complex Web page script and mapping it into tree structure easy to operate. It does not depend on DOM tree, and does not need utilize htmlparser bag to parse. By calculating text similarity, it calculates the similarity between the content of tree node and headings of different levels to determine the usefulness of the text information, cleans the Web page and extracts the content information. Experimental results show that the method has better universal property and accuracy rate in main text extraction.

【Key words】 Web page main text extraction; Web page mapping; Web page cleaning; text similarity

1 概述

随着 Internet 的飞速发展，网络上的信息呈爆炸式增长。网页已经成为 Internet 上最重要的信息资源。各种网页为人们提供了大量可供借鉴或参考的信息，成为人们日常工作和生活必不可少的一部分。然而，网页上的信息经常包含大量的噪声，如广告链接、导航条、版权信息等非网页主题信息的内容，页面所要表达的主要信息经常被隐藏在无关的内容和结构中，限制了 Web 信息的可利用性。本文主要对网页上的这些噪声进行滤除，并抽取网页正文信息，即网页清洗。它是 Web 文本分类、聚类、文本摘要等文本信息处理的基础，网页正文抽取的效果直接影响到文本信息处理的效果。

本文的方法首先抽取出 HTML 页面中的 title 及各级标题，再对网页进行标准化预处理，然后建立一种新的树型结构，HTML 中的所有正文信息都包含在这棵树的节点中。利用这种树型结构可以方便地清洗网页中的噪声、抽取出网页中的正文信息。在抽取网页正文信息时，较大的文本块根据文本的长度极易抽取出，而对于只有小文本块的节点，由于页面中的 title 及各级标题高度概括了该网页的主要内容，因此可以根据各节点内容与 title、各级标题的相似度来判定该节点的信息文本是否为有用文本，只要该小块文本与 title 或某个子标题的相似度大于设定阈值，就判定其为有用信息。

2 相关工作

虽然网页正文提取是 Web 文本挖掘中的一个重要问题，但相关研究并不多。目前对网页进行噪声过滤与信息自动抽取的方法主要有两大类：(1)针对单一页面进行处理。根据所处理页面的内容特征、可视信息等应用一些启发性规则去除页面的噪音，抽取出页面内容。这类方法对每一个待处理的网页进行同样的处理，对于抽取通过模板产生的网页集效率较低。(2)针对同一站点中页面的一般模式进行处理。这种方法是基于一个或多个网站中的页面集进行模板检测的，但局限于由同一个模板生成的网页集，直接影响清洗的自适应性。

文献[1]的研究仅限于某些特定站点，在这些站点中根据合并不同页面生成的 DOM 树来标记页面中哪些是有用信息哪些是噪声，并通过这些标记达到页面清洗的目的。文献[2]根据 HTML 标签生成树，通过分析同一网站下网页之间模板的相似性来识别数据区域。文献[3]基于 DOM 规范，提出了基于语义信息的 STU-DOM 树模型，将 HTML 文档转换为 STU-DOM 树，并对其进行基于结构的过滤和基于语义的剪枝，完成了对网页主题信息的抽取。文献[4]采用基于标记窗的方法并利用 Levenshtein Distance 公式计算标记窗中字符串与标题词之间的距离，从而判断该字符串是否为正文信息，该方法容易导致很多噪声无法滤除。

通过分析可知，现有的网页清洗方法大多基于 DOM 树并用 HTMLparser 程序包[5]对其进行解析，这种方法效率不高，而且依赖于第三方包。对此本文提出了一种简单的树型结构，在这棵树中保存了正文信息，同时消除了一些无用信息，并对各节点进行了简化，带来了操作上极大的便利。另外，在这棵树中可以通过深度搜索子节点来消除传统方法中不能处理网页正文部分被存放在多个 td 中的情况以及不能处

基金项目：国家自然科学基金资助项目(60575035, 60975033)；上海市重点学科建设基金资助项目(J50103)；上海大学研究生创新基金资助项目(SHUCX092162)

作者简介：王 利(1984－)，男，硕士研究生，主研方向：文本挖掘，事件本体；刘宗田，教授、博士生导师；王燕华，硕士研究生；廖 涛，博士研究生

收稿日期：2009-08-10 **E-mail：**wonglee07@gmail.com

理一个 td 中含有不同内容的情况，即不能处理一个 td 中存放的不仅仅是网页正文的情况。对节点中信息的可用性判别可以采用文本相似度计算方法。通过计算各节点中所含信息与网页中各级标题及大块确定文本信息的相似度来确定。实验结果表明，这种方法具有很高的准确性与通用性。

3 网页内容抽取

3.1 网页预处理

首先抽取出网页 title 及各级<h1>…<h2>…<h*n*>的标题内容，将其作为网页正文比较文本。此后抽取的网页文本信息根据与其的相似度计算结果判定该抽取的信息是否有用。

网页内容预处理步骤如下：

(1)抽取<body>与</body>中的内容，并滤除空格、脚本语言<script>…</script>、注释、网页显示风格代码<style>…</style>等无用信息。

(2)将<p>,
等换行换段符号替换成“#”，用于文本抽取之后的换行处理。

(3)HTML 标记替换。替换规则如表 1 所示。

表 1　HTML 标记替换规则

源码标记	替换后标记
<body>….</body>	<d>…</d>
<tr>…</tr>	<d>…</d>
…	<d>…</d>
<table>…</table>	<d>…</d>
	<a>

(4)滤除所有非<d>…</d>,<a>…包含的数据。

(5)对<d>与</d>进行配对处理，使每一个<d>都有一个</d>与之配对。

经过上述处理，由于标记窗口全部统一成<d></d>，因此极大地方便了操作，提高了处理效率。

3.2 树型结构建立

将 3.1 节处理后的页面信息用递归的方法映射成一棵树。树节点结构如下：

```
class Node{
public int flag; // flag=1：有子节点；flag=2：无子节点
        public boolean useful;    //true 为可用，false 为可将其滤除
         public Node parent;
         public Vector vector;    //偶位存文本，奇数存子节点}
```

设预处理后的网页代码如下：

```
<d1>
 str1
 <d2>
        str2
        <d3>
              str3
        </d3>
        str4
 </d2>
 str5
 <d4>
        str6
 </d4>
 str7
</d1>
```

其中，d1,d2 都代表 d，仅为了加以区别而加了编号；str*i* 代表文本。

通过递归建树可以得到图 1 所示的树型结构，其中，str*i* 表示属于某一节点的文本信息，如节点 d1 的文本信息包括 str1+str5+str7，节点 d2 包含的文本信息是 str2+str4，节点 d3 包含的文本信息是 str3，节点 d4 包含的文本信息是 str6；实箭头指向父节点；虚箭头指向该节点的详细内容。

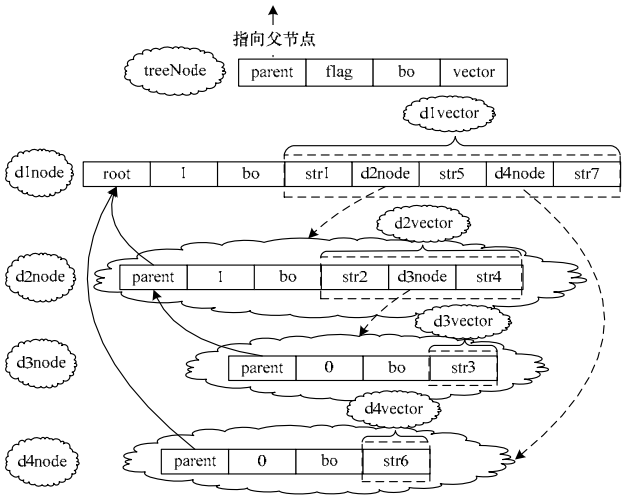

图 1　经处理网页内容所映射成的树型结构

3.3 正文抽取

建立完图 1 的树型结构，对网页内容的操作就都可以在这棵树的基础上进行了。网页中的信息为 str1,str2,…,str7，要获取这些信息，只要对该树进行一次遍历即可。现在的主要工作是判断某节点内的文本信息是否为有用信息。如果节点信息有用，则 useful 为 true，否则为 false。计算公式如下：

```
if (textsize>multi·numa && textsize>mintextsize)||α>β
        useful=true
else    useful=false
```

其中，*textsize* 代表节点所含文本字符串的长度；*numa* 代表文本中所含链接的个数；*multi* 是倍数参数；*mintextsize* 代表设定的文本最小长度阈值；$\alpha > \beta$ 表示小块文本与 title 或某一个子标题的相似度 α 大于设定阈值 β。

如果节点中的信息为大文本块并且所包含的链接数目也较少(链接数目可以根据文本中保留的<a>…来计算)，直接设定其为有用文本信息。采用的量化公式如下：

```
useful1=textsize>multi·numa && textsize>mintextsize
```

该公式表示某节点所包含的文本大于文本中所含链接数目的 *multi* 倍，并且文本长度大于设定的最小文本长度阈值。如果 useful1 为真，则 useful1=ture，该节点所含文本为有用文本。

对于节点中所含文本长度较小的小块文本，采用向量空间模型对每一个节点的文本信息进行量化。在 VSM 中，将节点文本看作是由一组词条($T1,T2,\cdots,Tn$)构成，对于每一词条 Ti，都根据其在节点文本中的重要程度赋一个加权值 Wi，并将($T1,T2,\cdots,Tn$)看成一个 n 维坐标系中的坐标轴，$W1,W2,\cdots,Wn$ 为对应的坐标值。这样由($T1,T2,\cdots,Tn$)分解得到的正交词条矢量组就构成了一个节点文本信息的向量空间。

采用 TF 方法计算各节点中文本的向量权重：

$$W_i = \frac{tf_i}{\sqrt{\sum_{j=1}^{n} tf_j}} \tag{1}$$

其中，tf_i 是第 i 个关键词在该节点文本内出现的频率；n 为该节点的文本内出现的词的个数。假设 2 个节点的文本 U,V 的相似度可用向量之间的夹角度量，相似度计算如下：

$$sim(V,U)=\cos(V,U)=\frac{\sum_{k=1}^{n}W_{vk}\cdot W_{uk}}{\sqrt{\sum_{k=1}^{n}W_{vk}^2}\cdot\sqrt{\sum_{k=1}^{n}W_{uk}^2}} \quad (2)$$

对于 3.1 节中抽取的网页中的 title 及各级标题，分别用式(1)计算其特征向量的权重，用式(2)计算各节点所包含的文本与 title 及各级标题的相似度 αi，若存在 $\alpha i>\beta$，则将文本所在节点 useful 标记为 true，否则，记为 false。将 useful 为 true 的节点的文本信息抽取出来，抽取时保持原有的顺序，网页正文即可成功地抽取出来。

4 实验与结果分析

本方法已经应用于上海统战部统战信息系统中。本文对来自新浪、腾讯、搜狐、新华网、人民网、中国汽车网等门户网站的 1 000 个 URL 地址进行了测试。将抽取结果分为：满意(对一个网页抽取的正确率达 95%以上)；有错误但错误可接受(对一个网页抽取的正确率为 85%~ 95%)；有错误且错误不可接受(对一个网页抽取的正确率低于 85%)。参数设定为 *mintextsize*=250, *multi*=25, β=0.7，本文方法与未加相似度计算的方法相比，满意度有很大的提高。实验结果如表 2、表 3 所示。

表 2 无相似度计算的实验结果

URL 总数	满意	可接受	不可接受	URL 连接异常
1 000	913(91.3%)	64(6.4%)	8(0.8%)	15(1.5%)

表 3 加入相似度计算的实验结果

URL 总数	满意	可接受	不可接受	URL 连接异常
1 000	825(82.5%)	127(12.7%)	33(3.3%)	15(1.5%)

实验中出现的抽取严重错误是指抽取的内容包含非有用信息或部分有用信息没有抽取出来，其主要由网页的各级标题抽取错误造成。网页标题与网页中大文本块的正确抽取直接关系到本系统的正文抽取效果。URL 连接异常是指网页的 URL 地址改变或网页不存在而导致的网页无法读取。从实验结果可以看出，本方法对网页信息抽取有相当高的准确度，与传统基于模板的方法相比，本方法不局限于某些特定网站的网页，具有较强的通用性，也不依赖于 DOM 树标准，不必用 HTMLparser 程序包进行解析，具有很好的独立性。

5 结束语

网页信息的正确抽取对面向 Web 的信息处理具有相当重要的意义，网页信息抽取的好坏对网页文本分类、聚类会产生直接的影响。本文将源网页转换为一种方便处理的树型结构，然后通过相似度计算判定文本信息是否为有用信息。本方法操作方便、抽取准确。之后将在此基础上进行事件本体的相关研究，如：网页文本中事件的抽取，基于事件本体的文本分类、自动文摘。

参考文献

[1] Yi Lan, Liu Bing, Li Xiaoli. Eliminating Noisy Information in Web Pages for Data Mining[C]//Proc. of the 9th Conference on Knowledge Discovery and Data Mining. [S. l.]: ACM Press, 2003.

[2] 黄健斌, 姬红兵, 孙鹤立. 网页中动态数据区域的识别与抽取[J]. 计算机工程, 2007, 33(11): 53-55.

[3] 王 琦, 唐世渭, 杨冬青, 等. 基于 DOM 的网页主题信息自动抽取[J]. 计算机研究与发展, 2004, 41(10): 1786-1791.

[4] 赵欣欣, 索红光, 刘玉树. 基于标记窗的网页正文信息抽取方法[J]. 计算机应用研究, 2007, 24(3): 144-145.

[5] 时达明, 林鸿飞, 杨志豪. 基于网页框架和规则的网页噪音去除方法[J]. 计算机工程, 2007, 33(19): 276-278.

编辑 张 帆

(上接第 101 页)

从表 1 和表 2 可以看出，随着处理器个数的增加，2 种模式的效率都呈下降趋势，其原因是消息传递和重复计算的开销随之增加。在同等条件下，主从模式的加速比大于对等模式，随着处理器个数增加，差距逐渐增大，这是由于主从模式具有更好的负载平衡能力而减少了等待时间。

为了更好地说明 2 种模式在负载平衡性上的差别，分别计算最后完成计算的处理器的墙上时间与最早完成计算的处理器的墙上时间之差，即形成最大资源浪费的处理器的等待时间，如表 3 所示。

表 3 2 种模式中处理器的最大等待时间 h

np	16	32	64	128
对等模式	4.78	4.71	4.45	2.94
主从模式	0.37	0.37	0.42	0.43

从测试结果可以看出，在主从模式中，处理器的等待时间远小于对等模式，表明主从模式能够得到更好的负载平衡，从而更有效地提高计算效率。但主从模式不可能实现负载的完美平衡，达到百分之百的计算资源使用率。

通过数据划分和分治进行并行优化是计算生物学中常用的有效方法，但对大规模数据的平均分割不一定会带来最佳的提速效果，应该通过均调各个处理器的计算时间来有效获得计算负载平衡。

5 结束语

本文对 InsPecT 软件进行并行优化，分析并对比了 2 种优化模式的效果。面对海量的质谱数据，如何继续提高计算效率和计算精度是下一步的工作重点。

参考文献

[1] Bandeira N, Tang Haixu, Bafna V, et al. Shotgun Protein Sequencing by Tandem Mass Spectra Assembly[J]. Analytical Chemistry, 2004, 76(23): 7221-7233.

[2] 胡 笳, 郭燕婷, 李艳梅. 蛋白质翻译后修饰研究进展[J]. 科学通报, 2005, 50(11): 1061-1072.

[3] Tanner S, Shu Hongjun, Frank A, et al. InsPecT: Identification of Posttranslationally Modified Peptides from Tandem Mass Spectra[J]. Analytical Chemistry, 2005, 77(14): 4626-4639.

[4] Tsur D, Tanner S, Bafna V, et al. Identification of Post-translational Modifications by Blind Search of Mass Spectra[J]. Nat Biotechnology, 2005, 23(12): 1562-1567.

[5] 张林波, 迟学斌, 莫则尧, 等. 并行计算导论[M]. 北京: 清华大学出版社, 2006.

编辑 陈 晖