第 34 卷 第 5 期 计 算 机 工 程 2008 年 3 月
Vol.34 No.5 Computer Engineering March 2008
·开发研究与设计技术· 文章编号：1000—3428(2008)05—0274—03 文献标识码：A 中图分类号：TP391

互联网商品信息抽取技术

于鲁波[1]，陈 超[2]
(1. 中国科学技术大学电子工程与信息科学系，合肥 230027；2. 多媒体计算与通信教育部微软重点实验室，合肥 230026)

摘 要：针对网页信息抽取中格式多样化的问题，提出一种基于路径统计聚类的信息抽取算法。该算法充分利用电子商务网站网页的特点，给出网页统计信息的一般数学表达式，在此基础上，采用基于统计聚类的思想，分割信息块，实现抽取信息。通过对实际电子商务网站网页信息的抽取，证明算法的有效性，分割正确率达 92.27%，信息抽取正确率达 98.24%。

关键词：网页分割；网页信息抽取；包装器；路径聚类

WWW Merchandise Information Extraction

YU Lu-bo[1], CHEN Chao[2]
(1. Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027;
2. MOE-Microsoft Key Laboratory of Multimedia Computing and Communication, Hefei 230026)

【Abstract】 In response to format diversity problem in the webpage information extraction, this paper proposes a new information extraction method based on XPATH clustering. The method utilizes the character of e-commerce website and gives a general mathematic formula. Based on it, this paper uses the thought of webpage statistical information clustering, segments the information block, and realizes the information extraction. This paper proves the validity of the algorithm through the practical website information extraction, achieves good results. Segmentation accuracy is 92.27%, and information extraction accuracy gets 98.24%.

【Key words】 Web page segmentation; Web page information extraction; wrapper; XPATH clustering

电子商务网站具有按照某种目录形式排列产品信息的共同特点。这些信息基本都是通过后台数据库产生，即所谓的隐藏网页(hidden Web)[1]。因此，这些网站的页面版式非常规则整齐。信息抽取中将网页中的待抽信息从HTML格式转化为XML或数据库格式的软件叫Wrapper。Wrapper利用一系列的规则(手写[2]或者通过机器学习自动获得[3])，来进行模式匹配抽取信息。文献[4]按照抽取任务的不同将Wrapper划分为 3 种类别：记录级(record-level)，网页级(page-level)，网站级(site-level)。本文的抽取工作同时涉及记录级和网页级。商品网站一般有着较为统一的结构：在主页上商品信息一般按照相关类别，按照信息条排列商品信息。依据这个特点采用网页分割的方法，将记录(商品信息条)从网页中分割出来，这个过程对应于抽取任务的记录级。同时可以发现，对于每一信息条都有一个对应于商品信息网页的链接，这个网页采用网页级抽取。

1 相关工作

文献[3]采用机器学习的方法进行网页表格的识别与分割。首先将每一个表格中的信息实例(待抽取信息)按照结构约束、坐标约束，将一个个信息实例分配到各个抽取单元中，各个实例满足连续性(空间上两两相连的实例)条件和唯一性条件(一个实例仅仅可以赋给一个抽取单元)，最后通过求解一组线性规划方程，得到一组最优解，分割网页。

文献[5]采用网页内容分析方法进行网页分割。利用大量的规则建立了一棵自顶向下生成的标记树，将相似度较大的节点聚类，如果类内各个节点的聚合度小于阈值，则继续向下聚类。

文献[6]以统计方法为基础，提出了一种层次聚类，来帮助确定信息条的位置。与文献[5]不同的是文献[6]没有采用规则，而是定义了一系列的统计参数，例如：2 个节点的相似度，类内的方差，聚类层次等。将学习样本表示成 DOM 树，通过统计相邻节点的相似度，得到一个相似度矩阵，然后采用经典的聚类算法，对节点按照相似度的大小依次聚类。

2 算法实现

互联网网页商品信息抽取系统实现框图如图 1 所示。

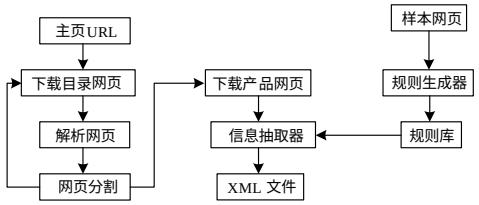

图 1 系统结构框架

本文将网页分为 2 类：目录网页(taxonomy page)，产品网页(detail page)。

2.1 相关概念

目录网页具有很明显的块状信息。例如：商品信息条，下一页链接，导航条。对于目录网页采用网页分割的方式进行信息提取。网页分割算法是基于启发式规则的，算法分为 2 步：(1)XPATH 聚类；(2)对聚类的 XPATH 进行分割。本文

基金项目：多媒体计算与教育部-微软重点实验室开放基金资助项目(06120809)
作者简介：于鲁波(1981－)，男，硕士研究生，主研方向：信息抽取；陈 超，工程师、硕士
收稿日期：2007-04-06 **E-mail：**yulubo@mail.ustc.edu.cn

约定，DOM 树的叶节点按照其在原始 HTML 文件出现的先后顺序编号。

XPATH 聚类：将具有最大相似度的叶节点聚类。节点最大相似度，即 2 个节点 XPATH 完全相同。本文用向量 $\boldsymbol{X}_i=[x_{i,1},x_{i,2},\cdots,x_{i,n}]$ 表示第 i 个 XPATH 的聚类。其中，$x_{i,j}$ 表示第 i 个 XPATH 聚类中的第 j 个叶节点。

节点间距：1 个 XPATH 聚类中 2 个节点编号之间的间隔。

$$\Delta Span_{i,j,k}=\left|x_{i,j}-x_{i,k}\right| \tag{1}$$

表示第 i 个 XPATH 聚类的第 j 与 k 节点之间的编号间隔。

平均周期：一个 XPATH 聚类中相邻节点间距的均值。

$$\Delta T_i=\frac{1}{n-1}\sum_{j=1}^{n-1}\Delta Span_{i,j,j+1} \tag{2}$$

间距方差：考察一个聚类中各个节点离散程度的量。

$$\sigma^2(\Delta T_i)_i=\frac{1}{n-1}\sum_{j=1}^{n-1}(\Delta Span_{i,j,j+1}-\Delta T_i)^2 \tag{3}$$

分割点：将一个聚类中的不连续点称为分割点。为了反映分割点的具体位置定义了一个量 θ，它是前后 2 个间隔之间的比值，如果聚类中出现分割点，那么其 θ 值会变化很快。

$$\theta=\frac{\Delta Span_{i,(j+1),j}}{\Delta Span_{i,(j+1),j}}=\frac{x_{i,(j+2)}-x_{i,(j+1)}}{x_{i,(j+1)}-x_{i,j}} \tag{4}$$

为了增强分割鲁棒性将 θ 设定一个阈值范围。实验证明 $\theta\in[0.85,2]$ 可以得到较好的分割效果。

算法采用的启发式规则：

(1)如果 $\theta\notin[0.85,2]$，将向量 $\boldsymbol{X}_i$ 在分割点处分割开。

(2)如果 1 个向量的平均周期 $\Delta T>\mathrm{Pre}Span$，并且没有进行分割，节点数目大于预定义值，则认为已经到达产品记录边界。

(3)对于产品记录，直接将相应聚类中的节点加入边界。

2.2 XPATH 聚类算法

XPATH 聚类算法首先将 1 个目录页面表示为 DOM 树结构，采用深度优先的遍历策略，提取 DOM 树中的每一个叶节点。对于每次遍历的叶节点，通过比较其 XPATH，将其序号添加到具有最大相似度的 XPATH 聚类中。具体算法如下：

```
Input : DOM Tree
Output: XpathCluster
   Cluster(DOM Tree)
{   XpathCluster= ∅ ;
For each xpath of   leaf node
{ If(XpathCluster.xpath.Find(xpath))
{   XpathCluster.xpath.Insert(node)}
Else
{ XpathCluster.Insert(xpath);
          XpathCluster.xpath.Insert(node);
}endif
} endfor
Return XpathCluster;
}
```

2.3 网页分割算法

网页分割算法依据 2.1 节的启发式规则和 2.2 节中的 XPATH 聚类。由于聚类过程中，可能将非产品信息聚类到产品信息，因此首先分析其方差。若一个聚类中的方差很大，利用式(4)定位到分割点，将产品信息条与噪声分割开。另外还利用了产品信息条的聚类平均周期、产品信息条数目等统计信息帮助定位分割信息条。当第 1 个满足全部启发式规则和统计信息的聚类出现时，就可以认为已经找到了产品信息列表，完成分割任务。分割算法如下：

```
Input :XpathCluster;//Xapth 聚类
Output :SegBoundary;//分割边界
WebPageSeg
{ SegBoundary = ∅ ;
   Count=0;
While( Count!= XpathCluster.size() )
{     If(XpathCluster.at(c).var() is within threshold )
  {   If(XpathCluster.at(c).size() >MAXSIZE && ΔT > PreSpan )
{ SegBoundary.insert(each node within XpathCluster.at(c))
Break;}
Else Count++;
}endif
 }Else{//利用启发式规则(1)进行分割
   Detect segment point use(4)
   Sort(new cluser);
   Count++;
}end if
}end while
Return SegBoundary;
}
```

2.4 产品网页抽取

对于产品网页，采用基于DOM的实例抽取[7]。首先用户提供一张学习网页，通过GUI让用户输入待抽取的信息实例，生成一个实例文件，读取并分析实例文件，产生规则。网页信息规则抽取器如图 2 所示。

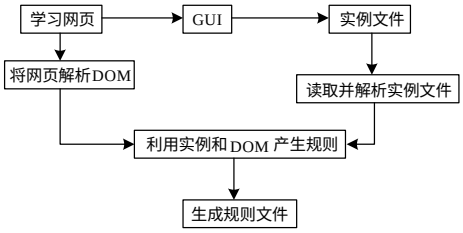

图 2　规则生成器结构

规则产生算法主要采用匹配的方法，将 DOM 树的叶节点与信息实例进行匹配。匹配成功则返回信息实例的路径作为规则。

```
Input :  学习实例网页，实例文件
Output :  规则文件。
Step1:将学习网页表示成 DOM 树
Step2:读取生成的实例文件
Step3: Curleafnode =DOM root;
for each instance
while(Curleafnode.find(instance) is false)
{ Curleafnode=NextLeafNode;
}end while
//输出信息实例的 XPATH 作为规则
      outXpath(Curleafnode);
      Curleafnode=NextLeafNode;
End for
```

信息抽取算法是 1 个覆盖算法，为了能够处理缺失属性值，当 1 个规则在 DOM 中没有相应的路径，就将该抽取信息的属性值设为空。

```
Input:  产品网页(detail page)，规则文件
Output: XML 形式的抽取文件
ExtraInfo(page , rule)
{   ParseToDom(page);
```

```
  For each rule
      If(page.find(rule))
      Attr=node.text;
Else Attr=NULL;
End if
End for
Output XML document;
}
```

3 实验结果

3.1 评价度量

对于目录网页的信息条分割，采用分割正确率来衡量网页分割算法的性能。

$$S_i = \frac{|M_i|}{|I_i|} \tag{5}$$

其中，S_i 代表第 i 个网站的分割正确率；M_i, I_i 分别代表第 i 个网站分割正确的信息条数和全部信息条数。

3.2 实验结果及其分析

从互联网上找到 9 个网站，对产品信息条、信息抽取正确率、召回率进行测试，测试结果如表 1 所示。

表 1 算法测试结果

网站	记录条数	分割正确率/(%)	召回率/(%)	正确率/(%)
www.walmart.com	16	100.00	99.00	100.00
www.valve168.com	20	96.34	94.71	96.10
www.smartbargains.com	24	82.10	93.50	95.44
www.walgreens.com	10	80.50	94.73	97.30
www.pvindex.com.cn	10	100.00	100.00	100.00
www..taobao.com	40	95.40	99.00	100.00
www.ye-ya.com	20	100.00	96.40	100.00
www.chinapipe.net	6	100.00	100.00	100.00
www.homevisions.com	20	76.10	93.90	95.32
合计	166	92.27	96.80	98.24

可以看出分割算法总体效果还是很好的，平均正确率在 92.27%。但也有个别网站的分割正确率很低，例如 www.homevisions.com。分析了一下，主要原因在于这些网站的产品列表的聚类方差非常大，算法会将起伏较大的产品列表分割开，造成了分割正确率的下降。但是大部分网页的产品信息条排列还是很整齐的，这可以从表中的统计数据看出，有 4 个网站的分割正确率为 100%。

另一方面，产品具体信息的抽取无论在正确率还是召回率方面都非常高，平均正确率 98.24%，平均召回率 96.80%。这与在引言中提出的假设非常一致。因为产品的信息都是来自后台数据库，然后填充统一的页面模板。

图 3 是本文示例中对产品信息抽取得到的最终 XML 文件。从最后的结果看，本文的算法确实可以处理缺失属性。例如 productprice 和 productprovide 的属性值就是空值。

```
<?xml version ="1.0" encoding ="gb2312"?>
<?xml  - stylesheet type ="text /css" href="cs.css"?>
<Product>
<productname >膨胀阀</productname>
<productspecifion >A/B-TV、C/D-TV、ETV 系列等 </productspecifion >
<productprice></productprice >
<productwrap >按客户要求</productwrap>
<productprovide></productprovide >
<producingarea >上海 </producingarea >
<productproducer> 上海奉申制冷控制器有限公司 </productproducer>
<productdescriprion > 吸取了美国 ALCO 公司充注技术，可提供各种制冷剂及其各种的需求，并能和美国 ALCO 公司的热力膨胀阀互换。</productdescriprion >
<companywebsite >http ://www.fengshen -sh.com</companywebsite ></Product>
```

图 3 产品页面抽取结果

4 结束语

本文介绍了一种用于电子商务网站的网页分割算法，并且成功地用于信息提取，取得了较好的效果。但另一方面，算法对于规则度较小的网页的分割正确率还有些偏低，下一步主要工作就是要在保持较高的信息抽取正确率的前提下，增强分割算法的鲁棒性。

参考文献

[1] Raghavan S, Garcia-Molina H. Crawling the Hidden Web[EB/OL]. (2000-12-08). http://dbpubs.stanford.edu:8090/pub/2000-36.

[2] Hammer J, McHugh J, Garcia-Molina H. Semistructured Data: the TSIMMIS Experience[C]//Proc. of the 1st East-European Symp. Advances in Databases and Information Systems. St. Petersburg, Russia: [s. n.], 1997: 1-8.

[3] Lerman K, Getoor L, Minton S, et al. Using the Structure of Web Sites for Automatic Segmentation of Tables[C]//Proc. of the 2004 ACM SIGMOD International Conference on Management of Data. Paris, France: [s. n.], 2004: 119-130.

[4] Sarawagi S. Automation in Information Extraction and Integration [EB/OL]. (2002-08-02). http://www.cse.ust.hk/vldb2002/program-info/tutorial-slides/T1sarawagi.pdf.

[5] Cai Deng, Yu Shipen, Wen Jirong, et al. VIPS: a Visionbased Page Segmentation Algorithm[R]. Microsoft, Technical Report: MSR-TR-2003-79, 2003.

[6] Papadakis N K, Skoutas D, Raftopoulos K, et al. STAVIES: a System for Information Extraction from Unknown Web Data Sources Through Automatic Web Wrapper Generation Using Clustering Techniques[J]. IEEE Computer Society, 2005, 17(12): 1638-1652.

[7] 于 琨, 蔡 智. 基于路径学习的信息自动抽取方法[J]. 小型微型计算机系统, 2003, 24(14): 2147-2149.

（上接第 273 页）

主流系统中必须实现的。为了能够支持这 2 个系统特性，本设计在系统监控任务主循环流程中，增加主从切换、外围板安装、外围板卸载等 3 种消息，增加 2 个状态：外围板空，系统板处于从板。通过增加这些消息和状态，并跟上述心跳状态机结合，并密切配合，形成了一套完整的系统运行机制。

4 结束语

本文研究实现了 cPCI 平台下基于共享内存的多机通信机制，该机制很好地支持了 cPCI 架构的高可用、热插拔等功能，实现方案简单高效，支持多用户并可扩展。经过在国家部委预研项目多个硬件平台上验证和使用，形成了一套机制完善、问题诊断方便、简单实用的完整解决方案。

参考文献

[1] Intel Corporation. 21555 Non-Transparent PCI-to-PCI Bridge User Manual[Z]. 2001.

[2] 徐 恪, 吴建平, 喻中超, 等. 一种基于总线的多处理器共享内存机制[J]. 小型微型计算机系统, 2003, 24(3): 321-326.

[3] Wind River System Inc.. Vxworks Network Programmer's Guide[Z]. 1999.

[4] 邹志强. 基于 cPCI 总线的共享内存技术的实现[J]. 计算机工程, 2006, 32(16): 232-234.