第35卷 第22期 计 算 机 工 程 2009年11月
Vol.35 No.22 Computer Engineering November 2009
·软件技术与数据库· 文章编号：1000—3428(2009)22—0059—03 文献标识码：A 中图分类号：TP311.12

基于数据区域发现的信息抽取规则生成方法

曲著伟[1,2]，李敏强[1]
(1. 天津大学管理学院，天津 300072；2. 浙江财经学院信息学院，杭州 310018)

摘 要：提出一种自动检测网页中数据记录结构特点并生成Web信息抽取规则的方法，以网页DOM树为基础，自动发现和分离Web数据区域所对应的DOM子树，将其分解为数据记录子树集合，综合数据记录子树的结构特点生成抽取规则。实验结果显示，该方法具有较高的抽取准确率和查全率。

关键词：信息抽取；抽取规则生成；Web数据区域；树匹配

Information Extraction Rule Generation Method Based on Data Area Discovery

QU Zhu-wei[1,2], LI Min-qiang[1]
(1. School of Management, Tianjin University, Tianjin 300072;
2. Information School, Zhejiang University of Finance & Economics, Hangzhou 310018)

【Abstract】This paper proposes an automatic method for detecting the structure characteristic of Web data records and generating Web information extraction rules. Based on Web DOM tree, Web data area is identified from Web DOM tree automatically and segmented into data records, and extraction rules are generated by synthesizing the structure of Web data records. Experimental result shows that the method gains high accuracy in terms of recall and precision.

【Key words】information extraction; extraction rule generation; Web data area; tree matching

1 概述

Web信息抽取一般可分为规则生成和信息抽出2个阶段，如何高效发现页面中的数据记录结构模式并得到普适的抽取规则，是该领域的一个研究热点[1]。

根据抽取规则生成方式可将Web信息抽取分为3类：(1)人工的规则生成方法，针对数据抽取网页，以编程的方式生成抽取规则，开发和维护很困难。(2)机器学习方法[1-3]，需要一定数量的网页作为训练集，利用机器学习算法生成抽取规则，自动化程度较高，但对网页中特定内容进行抽取仍需专门人员进行数据结构注释。(3)交互式信息抽取[4]，提高了用户交互性及抽取灵活性，但面对大量信息抽取工作时，规则生成过程单调重复，效率较低。

为了降低抽取规则生成所需工作量，同时保留必要灵活性，本文提出了一种自动化的信息抽取规则生成方法：以网页DOM树为基础，自动获得仅包含Web数据记录的数据区域子树，然后从中分离出单一Web数据记录树结构，并据此生成抽取规则。

2 数据网页及数据区域树

信息抽取的处理对象为一类特殊网页，若干结构类似的Web数据记录构成了网页的主体，一般称之为数据网页[3]。称来自于同一数据源、由相同或相似HTML模板驱动生成的网页为同类别网页[5]。同类别数据网页具有相似的网页结构和外观。根据内容是否与用户查询相关，可将数据网页分为2个区域：数据区域和非数据区域，前者包含Web数据记录集合；后者则显示网站导航等用户查询无关信息。设计规范的数据网页一般具有如下特点[2,5]：

(1)不同区域间存在明显分界，数据区域显示为一个连续的区域，且Web数据记录在其中连续显示。

(2)任意2个同类别页面数据区域的代码结构类似，但所含数据(记录)内容基本不同。

(3)任意2个同类别页面非数据区域的代码结构和数据内容基本相同。

本文以DOM树[1,4]为处理基础，从中识别数据区域对应的DOM子树，进而分离出Web数据记录子树结构。

定义1 数据区域树是指数据网页DOM树T中包含所有查询相关Web数据记录的最小树结构，记为T_d；T中除数据区域树外的所有子树统称为非数据区域树。

由数据网页的特点可得如下结论：

(1)数据区域树与网页数据区域完整对应，数据区域树对应DOM树中的1棵子树，且其一级子树中存在结构类似的多棵子树[5]。

(2)设T_1和T_2是同类别网页，T_{d1}，T_{d2}和T_{n1}，T_{n2}分别是2个页面的数据区域树和非数据区域树，则T_{n1}和T_{n2}的树结构及叶节点内容基本相同，而T_{d1}和T_{d2}具有基本相同的树结构和基本不同的叶节点内容。

图1展示了一个数据页面DOM树的结构，Web数据区域树对应一个以*table*为根节点的子树，*table*节点的每棵子树对应一条Web数据记录。

作者简介：曲著伟(1980－)，男，讲师、博士研究生，主研方向：信息检索；李敏强，教授、博士生导师

收稿日期：2009-02-27 **E-mail：**quzw21@163.com

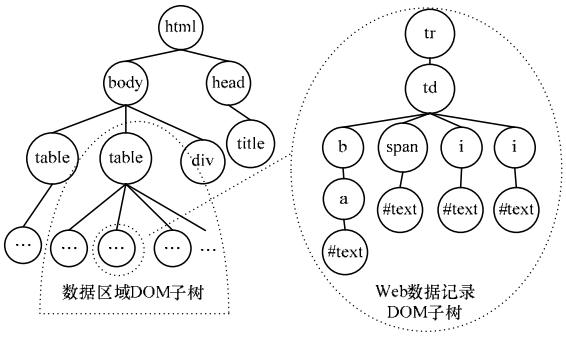

图 1　网页数据区域树及数据记录

3　数据区域树的获取

同类别页面之间数据区域树结构的相似性和非数据区域树结构、内容的相同性是本方法的根本依据。通过比较 2 个同类别页面的 DOM 树结构异同点，识别 DOM 树中的数据区域树与非数据区域树，进而分离出数据区域树。

3.1　TOP-DOWN 树匹配

TD(TOP-DOWN)树匹配是进行树对比的基本算法。TD 树匹配以 TD 树映射为基础[4]，严格 TD 树映射 M 为树之间一系列节点对的集合，有如下性质：(1)若节点对$(t_1(i_1), t_2(j_1))$, $(t_1(i_2), t_2(j_2))\in M$，必有：$t_1(i_1)$是 $t_1(i_2)$的左节点，当且仅当 $t_2(j_1)$是 $t_2(j_2)$的左节点；$t_1(i_1)$是 $t_1(i_2)$的祖先节点，当且仅当 $t_2(j_1)$是 $t_2(j_2)$的祖先节点。(2)若$(t_1(i),t_2(j))\in M$，必有$(p_1(i), p_2(j))\in M$，其中，$t_1(i)$和 $t_2(j)$分别表示树 T_1 和 T_2 的任意节点；$p(i)$表示 $t(i)$的父节点。

网页 DOM 树是节点标识有序树，图 2 展示了节点标识有序树 T_1 和 T_2 的 TD 映射。本文使用简单 TD 树匹配算法[4]计算树映射 M 及 TD 匹配值。简单地说，DOM 树之间 TD 映射的节点对数即为 TD 匹配值，匹配值与 2 棵树的匹配度呈正比，而不同的节点映射定义会导致不同的匹配值及匹配度。

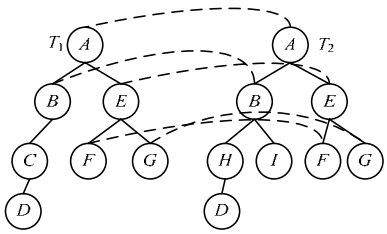

图 2　树 T_1 和 T_2 的 TD 映射

定义 2　树 T_i 和 T_j 的 TD 匹配度 θ 为 2 棵树的 TD 匹配值与 2 棵树平均节点数的比值。即

$$\theta\left[T_i,T_j\right]=\frac{2\times Match[T_i,T_j]}{size[T_i]+size[T_j]} \tag{1}$$

其中，$Match[T_i,T_j]$表示 T_i 和 T_j 的 TD 匹配值，且节点之间匹配与否仅取决于其类型；$size[T_i]$表示树 T_i 的节点数目。

定义 3　树 T_i 和 T_j 的叶节点匹配度 θ_L 为 2 棵树的叶节点匹配值与 2 棵树平均叶节点数的比值。即

$$\theta_L\left[T_i,T_j\right]=\frac{2\times MatchLeaf[T_i,T_j]}{size_L[T_i]+size_L[T_j]} \tag{2}$$

其中，$size_L[T_i]$是 T_i 的叶节点数目。叶节点匹配值 $MatchLeaf[T_i, T_j]$为 2 棵树 TD 匹配的叶节点数目，计算公式为

$$MatchLeaf[T_i,T_j] = Match[T_i,T_j] - Match_L[T_i,T_j] \tag{3}$$

相对于 $Match[T_i, T_j]$而言，$Match_L[T_i, T_j]$对叶节点的匹配与否要求较为严格，计算过程中需同时比较节点的类型、值，以此区分不同 Web 数据记录子树中同类不同值的叶节点。

定义 4　若树 T_i 和 T_j 的 TD 匹配度 θ 大于阈值 λ_1，称 T_i 和 T_j 为同构树；若同构树 T_i 和 T_j 的叶节点匹配度 θ_L 小于阈值 λ_2，称 T_i 和 T_j 为同构数据树。其中，λ_1 和 λ_2 为实验中可调节的参数。

3.2　数据区域树获取

给定同类别数据页面 DOM 树 T_1 和 T_2，设 T_{d1} 和 T_{d2} 分别是 T_1 和 T_2 中的数据区域树，用以识别数据区域树的启发式规则如下：

规则 1　T_{d1} 和 T_{d2} 的树结构相同或相似，两者的 TD 匹配度较高。

规则 2　T_{d2} 的一级子树中必存在 T_{d1} 任意一级子树的同构数据树，反之亦然。

规则 3　$T_{di}(i=1,2)$的一级子树中均衡分布若干同构数据树对。

从给定 DOM 树中获取数据区域树 T_d 的基本步骤如下：

(1)选择 T_1 为 T_d 的生成树，T_2 为结构对比树；T_1 的第 2 层节点序列为 $t_1(1),t_1(2),\cdots$，对应子树序列 $\Psi_{T1}=(T_1(1), T_1(2),\cdots)$。$T_2$ 的第 2 层节点序列为 $t_2(1), t_2(2),\cdots$，对应子树序列 $\Psi_{T2}=(T_2(1), T_2(2),\cdots)$。

(2)计算 T_1 和 T_2 的 TD 匹配映射 M，对于任意 $(t_1(i),t_2(j))\in M$，计算 $\theta[T_1(i),T_2(j)]$和 $\theta_L[T_1(i),T_2(j)]$。

(3)若 $T_1(i)$中包含 T_d 或 Web 数据记录子树，则 Ψ_{T2} 中必存在 $T_1(i)$的同构数据树，即必有$(t_1(i),t_2(j))\in M$，且 $\theta[T_1(i),T_2(j)]>\lambda_1$, $\theta_L[T_1(i),T_2(j)]<\lambda_2$。从序列 Ψ_{T1} 中删除所有不满足上述条件的成员树。

(4)若 Ψ_{T1} 的剩余成员中存在 $T_1(k)$，使得$(t_1(k), t_2(k'))\in M$ 且 $size_L[T_1(k)]/size_L[T_1]$ 大于设定阈值 δ，则令 $T_1=T_1(k)$, $T_2=T_2(k')$，转(2)，否则，转(5)。

(5)若 Ψ_{T1} 的剩余成员中存在同构数据树对，转(6)，否则，从 Ψ_{T1} 中选择在 Ψ_{T2} 中存在同构数据树且叶节点数目最多的成员树，记为 $T_1(k)$，其同构数据树为 $T_2(k')$，令 $T_1=T_1(k)$, $T_2=T_2(k')$，转(2)。

(6)记 Ψ_{T1} 的剩余成员为$(T_1'(1), T_1'(2),\cdots)$，从中检测并删除在本序列中无同构数据树的任意成员树 $T_1'(i)$。处理结束的 T_1 为数据区域树 T_d。

算法通过逐层摒除不满足设定条件的子树得到数据区域树 T_d，等同于寻找一条从生成树的根节点到 T_d 的节点路径，该路径在设计规范的数据网页中是唯一的，例如图 1 中从根节点到 T_d 的路径为：/html/body/table[2]。

4　数据记录结构获取及抽取规则生成

为了从数据区域树中分离出 Web 数据记录结构，提出如下树垂分的概念：

定义 5　树垂分 T_p 是源树 T 的根节点和其部分子孙节点组成的满足下述性质的树结构：(1)设 T_p 的任意 2 个节点 $t_p(i)$和 $t_p(j)$分别来源于 T 中的节点 $t(i)$和 $t(j)$，$t_p(i)$是 $t_p(j)$的父节点，当且仅当 $t(i)$是 $t(j)$的父节点，$t_p(i)$是 $t_p(j)$的左相邻兄弟节点，当且仅当 $t(i)$是 $t(j)$的左相邻兄弟节点。(2)T_p 的根节点对应于 T 的根节点，T_p 的叶节点在 T 中也是叶节点。(3)若 T 中的任意 2 个兄弟节点出现在 T_p 中，则 2 个节点的所有子孙节点也以不变的相对位置出现在 T_p 中。

定义 6　称树垂分 T_{p1} 和 T_{p2} 为相邻树垂分，若满足如下性质：(1)T_{p1} 和 T_{p2} 来自相同源树；2 棵树叶节点集的交集为空。(2)合并 2 棵树的同源节点后可得到一棵树 T_p'，其先序

遍历节点序列为源树的先序遍历序列的子集。

对于相邻树垂分 T_{p1} 和 T_{p2}，必存在整数 $K(K\geq 1)$，使得：(1)T_{p1} 和 T_{p2} 的第 $i(i=1,2,\cdots,K)$层节点序列都只有 1 个节点，且这 2 个节点是同源节点。(2)T_{p1} 和 T_{p2} 的第 $K+1$ 层节点序列对应源树中的相邻兄弟节点序列。

数据区域包含若干连续且结构类似的数据记录，可将数据区域树 T_d 切分为多个结构类似的相邻树垂分，使之与 Web 数据记录相对应，如图 3 所示。

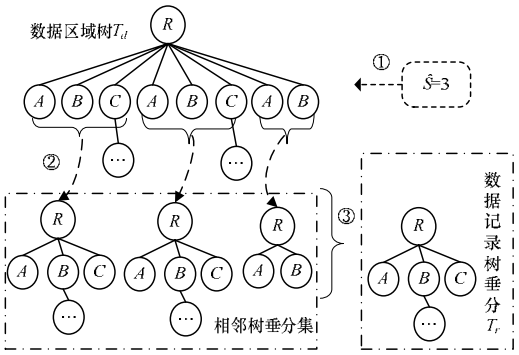

图 3　Web 数据区域树的切分过程

设 T_d 的根节点为 R，一级子树序列为 $\Psi_d=(T(1), T(2),\cdots,T(n))$，将 T_d 划分为多个相邻树垂分的关键步骤是将 Ψ_d 切分为若干相似的子树序列，每个子树序列结合 R 即得到所需树垂分。综合这些树垂分的一般性结构得到初步树垂分 T_r，若 T_r 可继续被切分，则采用相同的办法将 T_r 继续切分为多个树垂分。数据区域树切分步骤如下：

(1)确定对 Ψ_d 序列划分的步长值。给定步长值 S=1, 2,$\cdots$,$\lfloor n/2 \rfloor$，自前而后将 Ψ_d 划分为多个长度为 S 的子树序列 $(\Psi_1, \Psi_2,\cdots)$，计算任意相邻子树序列的匹配度，匹配度平均值为 $\bar{\theta}(S)$；记 $\bar{\theta}(\hat{S})$ 为所有 $\bar{\theta}(S)$ (S=1,2,$\cdots$, $\lfloor n/2 \rfloor$)的最大值，$\hat{S}$ 即为最佳序列划分步长，若 $\hat{S}$ 有多个可选值，则取其中最小者。相邻序列 Ψ_i 和 Ψ_j 的匹配度计算方法如下：

将 Ψ_i 和 Ψ_j 分别结合 R 生成 2 棵树垂分，计算两者的 TD 映射 M，得到 Ψ_i 和 Ψ_j 之间相互匹配的子树对集合，求得每一子树对的 TD 匹配度并利用式(4)计算 Ψ_i 和 Ψ_j 的匹配度：

$$\theta\left[\psi_i,\psi_j\right]=\frac{2\times\sum_{k=1}^{m}\eta_k}{Count(\psi_i)+Count(\psi_j)}\times\left(1-\rho\times\frac{S-1}{n}\right)$$
$$\eta_k=\begin{cases}0 & \theta_k-\lambda_1<0\\ 1 & \theta_k-\lambda_1>0\end{cases} \tag{4}$$

其中，$Count(\Psi_i)$为子树序列 Ψ_i 的成员数目；m 为 Ψ_i 和 Ψ_j 之间相互匹配的子树对数；θ_k 为第 k 对匹配子树的 TD 匹配度；λ_1 为判断 2 棵树是否为同构树的阈值；0-1 变量 η 用于确保 Ψ_i 中任意成员对于判断 Web 数据记录结构的同等重要性；实验中发现，随着 S 取值的增大，$\bar{\theta}(S)$ 呈现周期性变化规律，参数 $\rho(0\leq\rho\leq 1)$用于避免这种情况的发生。

(2)得到初步树垂分。计算 $T(1)$与 Ψ_d 中任一成员 $T(i)$的 TD 匹配度，得到 $T(1)$的同构子树集，以这些子树所处位置为分界点，将 Ψ_d 划分为若干子序列$(\Psi_1',\Psi_2',\cdots)$，若 $\sum_{j=1}^{k}\psi_j' \leq\hat{S}$ 且 $\sum_{j=1}^{k+1}\psi_j' >\hat{S}$，则合并$(\Psi_1', \Psi_2',\cdots, \Psi_k')$，从 Ψ_{k+1}'开始按相同规则继续合并子序列，直至所有 Ψ_j'被合并，将新得到的每个序列结合 R 生成树垂分。选择节点数最多的树垂分(记为 T_r)继续处理，也可将所有树垂分进行树比对合并[5]，得到具有综合特征的树结构。图 3 展示了 Web 数据区域树切分的一般过程。

若 T_r 的第 2 层节点数为 1 且二级子树序列又可分为多个结构类似的子序列，则令 T_r 的一级子树代替 T_r，重复前述步骤对 T_r 进行垂直切分，直至不可再分。最终得到表示 Web 数据记录一般性结构的树垂分，结合数据语义定义信息即得到信息抽取规则[4]。

5　信息抽取实验及结果

实验数据是从常用搜索引擎、电子商务、学术搜索等 10 个网站[2,4-5]上搜集到的 Web 数据页面集。实验分 2 个阶段：(1)从每个页面集中随机选取 2 个页面作为规则生成页面，利用本文方法自动生成信息抽取规则。(2)利用抽取规则对同类别网页集进行数据抽取，该过程应用了此前介绍的 Web 数据抽取方法[4]。表 1 展示了实验过程的基本参数及结果。

表 1　实验数据集及结果

网页来源	每页记录数	每条记录属性数	记录有无属性缺失	抽取网页数	是否获取规则	查全率/(%)
AltaVista News	10	4	无	30	是	100.0
Microsoft	10	3	无	30	是	100.0
Excite Web	20	5	无	16	是	100.0
Baidu Web	10	3	无	30	是	100.0
Google News	10	8	有	20	是	100.0
Amazon Books	25	7	有	32	是	100.0
Buy.com Books*	11	6	有	16	是	91.6
Taobao	40	7	有	20	是	100.0
eBay	50	6	有	20	是	100.0
CiteSeer	20	4	有	24	是	98.7
平均值	20.6	5.3	-	23.8	-	99.0

同构数据树在规则生成过程中起着非常重要的作用，参数 λ_1 取值过高或 λ_2 取值过低易导致数据区域树的发现困难，反之则易导致所得数据区域树中含有过多噪声数据，经过反复实验，发现 λ_1=0.7，λ_2=0.5 是较为普适的取值。实验中通过参数控制保证抽取准确率为 100%[4]，然后测得信息抽取查全率，高于 99%的平均查全率证明了本方法的有效性。

6　结束语

本文提出一种自动化的信息抽取规则生成方法，其具有如下特点：规则生成过程基本无需人工参与且充分利用 DOM 树结构，效率较高；专门针对网页数据记录区域生成抽取规则，无冗余规则产生。但本方法难以处理网页非数据区域中的数据信息，这是需进一步关注的重点。

参考文献

[1] Kim Yeonjung. Web Information Extraction by HTML Tree Edit Distance Matching[C]//Proc. of International Conf. on Convergence Information Technology. Busan, Republic of Korea: [s. n.], 2007.

[2] Chang Chia-Hui, Kuo Shih-Chien. Automatic Information Extraction from Semi-structured Web Pages by Pattern Discovery[J]. Decision Support Systems, 2003, 35(1): 129-147.

[3] Crescenzi V, Mecca G, Roadrunner M P. Towards Automatic Data Extraction from Large Web Sites[C]//Proc. of the 27th International Conf. on VLDB. Roma, Italy: [s. n.], 2001: 109-118.

[4] 张慧颖, 曲著伟. 基于子树匹配的交互式 Web 数据抽取方法[J]. 计算机工程, 2006, 32(9): 78-80.

[5] Zhai Yanhong, Liu Bing. Web Data Extraction Based on Partial Tree Alignment[C]//Proc. of the 14th Int'l World Wide Web Conference. [S. l.]: IEEE Press, 2005: 76-85.

编辑　张　帆