文章编号:1001 - 9081(2008)01 - 0152 - 03

一种基于 XQuery 的优化 Web 信息抽取方法

支宗良,陈少飞
(河北省烟草专卖局 烟草经济信息中心,石家庄 050051)
(teichb@ yahoo. cn)

摘 要:由于缺乏对页面特征适应性的分析,现有的典型系统难以保障抽取规则的健壮性。提出一种优化的 Web 信息抽取方法,该方法引入了相互关联的三层规则,在分析页面特征适应性的基础上,从准确率和召回率两方面出发提出了抽取规则的优化算法,并用标准 XQuery 表达复杂对象抽取规则。实验证明,该方法有效地增强了抽取规则的健壮性及可用性。

关键词:信息抽取;规则优化;XPath;XQuery
中图分类号: TP311.135.4 **文献标志码**:A

Optimized Web information extraction based on XQuery

ZHI Zong-liang, CHEN Shao-fei
(*Tobacco Economic Information Center, Hebei Province Tobacco Monopoly Administration, Shijiazhuang Hebei* 050051, *China*)

Abstract: Due to lack of the analysis of the adaptability of the Web page's characteristics, the current typical systems can hardly provide robust extraction rules. This paper proposed an optimized Web information extraction method which divided rules into three associated layers, suggested an optimized algorithm for extraction rules from the view of the precision and recall ratio through analyzing the adaptability of the page's characteristics, and expressed the complicated object rule in standard XQuery. Experiments indicate that our approach enhances the robustness and usability of the rules.

Key words: information extraction; rule optimizing; XPath; XQuery

0 引言

目前 Web 信息大多数以 HTML 形式存放且数量仍在增加,但其主要面向显示,缺乏语义信息,应用程序无法直接解析使用,造成资源浪费。为增强 Web 中数据的可用性而出现的 Web 信息抽取技术,是当今的一个研究热点。

近来涌现出了多种信息抽取工具[1],按其采用技术,主要分为基于自然语言处理、包装器推理、基于 ontology、基于 HTML 结构等方式的信息抽取。典型系统中,WHISK[2]在一定程度上应用了自然语言理解技术,但没有利用 Web 文档不同于普通文本的层次特性;XWrap[3]只适合对含有明显区域结构的网页进行抽取;W4F[4]需要用户手工书写部分抽取规则;Lixto[5]的功能与我们的原型系统比较接近,但 Elog 抽取语言实现和优化较困难。虽然已经有许多学者致力于这方面的研究,但是这些系统未对用于形成抽取规则的可用 Web 特征的适应性进行分析,生成的抽取规则缺乏健壮性。本文对这些特征的适应性进行了分析,引入了相互关联的三层规则,通过规则优化逐步获得健壮的抽取规则,并用标准的 XML 查询语言 XQuery 表达。实验证明,该方法提高了抽取规则的健壮性和可用性,可获得较高的准确率和召回率。

1 原理概述

在自主开发的原型系统 PQAgent2.0 中,信息抽取分为以下阶段:附加语义,样本学习,规则优化,产生基于 XQuery 的复杂对象抽取规则和利用 XQuery 引擎实现信息抽取。其中"附加语义"阶段,需要用户首先选择样本页面,在浏览样本页面的同时,由用户对网页创建语义模式;"样本学习"阶段,需要用户通过标记网页中相应的信息块,帮助系统建立信息块与语义模式中语义对象之间的对应关系,其他工作均由系统自动完成,工作流程见图 1。

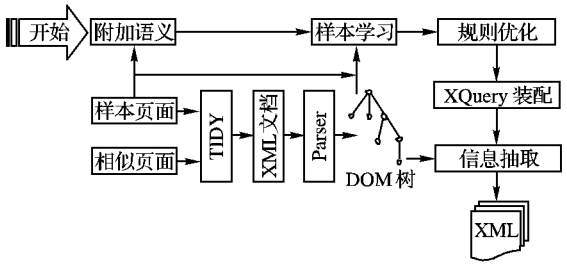


图 1 信息抽取工作流程

需要说明的是,由于 HTML 是 XML 的应用或特例,语法上与 XML 类似,只是标记固定,不含清晰的用户语义。系统内部使用 XML 表示 HTML,样本页面利用 TIDY[6]转换为 XML 文档,并被 Parser 转换为 DOM 树,系统内部所有的操作都基于 XML 的逻辑模型 DOM 树。

2 语义模型

语义模型用以表达 Web 数据潜在的语义信息。为增强输出数据格式的灵活性,本文选用"受限的 XML"(用 XML 语法表达,但与 IDL 规范兼容)作为语义模型,并用受限 DTD 表达模式,而 DTD 可直观地用树形结构表示,并称之为"模式树"。从 IDL 支持的数据类型看,模式树中可以有以下几种类型的节点,并用正则表达式定义其内容构成:

收稿日期:2007 - 08 - 10;**修回日期**:2007 - 10 - 18。

作者简介:支宗良(1971 -),男,河北衡水人,硕士,主要研究方向:数据库、数据集成、Web 数据管理; 陈少飞(1977 -),男,河北邢台人,硕士,主要研究方向:数据库、Web 信息抽取、XML。

· 原子对象:(#PCDATA)

· 集合对象:((原子对象)*)|((元组对象)*)

· 元组对象:(a_1,a_2,…,a_n)

其中,a_i:(原子对象|集合对象|元组对象)。

另外,集合对象的成分对象统称为成员对象,根节点的孩子节点是一个系统默认的"成员元组对象",称为"隐含成员元组对象"。为叙述方便,模式树中所有的节点统称为"语义对象"。

3　抽取规则的生成、优化及 XQuery 表达

本文引入了相互关联的三层规则,具体来说 Web 特征分布于规则段,由规则段组装成语义对象抽取规则,由语义对象抽取规则封装为复杂对象抽取规则,其中前两种用标准 XPath 语法表达,后一种用标准 XQuery 语法表达。

3.1　规则段及其表示

在基于页面结构的信息抽取中,常用 HTML 页面的结构特征、位置特征、语义特征、显示特征和引用特征表达语义模式和页面中数据内容间的映射,即抽取规则。本文将可用的 Web 特征分布到用 XPath 语法表达的六种规则段,且为不同类型的语义对象生成不同类型的规则段,各规则段描述如下(在后面给出相应的 BNF 定义):

· PureAttrPathExp:称为"纯属性路径表达式",形式上是完整的 XPath 表达式,对每个位置步(Location Step)做如下要求:如果含有与显示相关的属性,就在谓词中以所有该类属性名和属性值的等式作为谓词,若有多个谓词,之间的关系为"AND",否则该位置步不选用任何谓词。

· AttrOrderPathExp:称为"属性序号路径表达式",形式上是完整的 XPath 表达式,对每个位置步(节点测试为 text()的位置步除外)做如下要求:如果含有与显示相关的属性,就在谓词中以所有该类属性名和属性值的等式作为谓词,否则使用序号(在 DOM 树中,同一层的同一类节点从左到右的编号)对节点序列加以约束,若有多个谓词,之间的关系为"AND"。

· OrderPathExp:称为"序号路径表达式",形式上是完整的 XPath 表达式,对每个位置步(包括节点测试为 text()的位置步)均使用序号对节点序列加以约束。

· TxtFeaturePredicate:称为"文本特征谓词",形式上是 XPath 表达式的谓词。此谓词用来约束 DOM 树中某节点的文本值(内部节点的文本值由后代节点文本值连接构成)必须包含某个固定的文本值。若某语义对象有多个文本特征,则多个谓词之间的关系为"AND"。

· Big_ LR_BoundryPredicate:称为"大左右边界谓词",形式上是 XPath 表达式的谓词。其目的是语义模式中集合对象在 DOM 树中无与之对应的节点,而直接列出其成员对象的情况下,使用集合对象所跨越子树中最左子树根节点的所有左兄弟作为左边界,最右子树根节点的所有右兄弟作为右边界形成大左右边界谓词,以约束其成员对象。

· Small_LR_BoundryPredicate:称为"小左右边界谓词",形式上是 XPath 表达式的谓词。其含义是指仅使用某节点的左右兄弟约束该节点。本文仅对原子对象使用该谓词。其原因是在 DOM 规范中文本节点是个虚节点,多数情况下文本节点无左右兄弟,往往对网页结构不会造成很大的依赖。

前三种规则段以含序号的程度区分,且所用属性均是与显示相关的属性,后三种规则段均为谓词,和前三种规则段配合使用,不可单独使用。其中"大左右边界谓词"是将集合对象的该谓词用于其成员对象抽取规则的第一个位置步;其他两个谓词用在路径表达式的最后一个位置步。在某一个位置步若有多个谓词,之间的关系为"AND"。

各规则段的 BNF 定义如下:

```
PureAttrPathExp ::= (NodeName("[@"AttrName"="
  AttrValue"]")*) |(NodeName("[@"AttrName"="
  AttrValue"]")*"/"PureAttrPathExp) | NULL
AttrOrderPathExp ::= (NodeName("["num"]" | ("[@"AttrName
  "="AttrValue"]")*)) | NULL | (NodeName("["num"]"
  | ("[@"AttrName"="AttrValue"]")*)"/"AttrOrderPathExp)
OrderPathExp ::= (NodeName"["num"]") | (NodeName"["
  num"]""/"OrderPathExp) | NULL
TxtFeaturePredicate ::= ("[contains(string(.),"TxtFeatureValue
  ")]") +
Big_LR_BoundryPredicate ::= (("[count(../"NodeName"after.)
  ="Num"]") | ("[count(../"NodeName"before.)="Num
  "]")) +
Small_LR_BoundryPredicate ::=
  (("[count((../*after.)[1])=0]") | ("[((../"Nodename
    "after".)[1])=((../*after.)[1])]"))
  (("[count((../*before.)[1])=0]") | ("[((../"
    Nodename"before".)[1])=((../*before.)[1])]"))
```

系统为各类语义对象均生成三种路径表达式和文本特征谓词(如果有文本特征的话)。但对于成员对象,由于其个数不确定,因此在形成自身的抽取规则时,不宜使用位置特征中的序号,即不能包括"属性序号路径表达式"和"序号路径表达式"两个规则段。之所以在样本学习过程中为其生成这些规则段,是用来判别其他语义对象抽取规则的区分能力的,并不用于形成自身的抽取规则。

3.2　规则段的组合方式及适应性分析

有效的规则段组合方式中必须有且仅有一个路径表达式规则段,然后再根据某种方式在路径表达式某个特定位置步增加谓词规则段。具体来说,对于谓词规则段,将"小左右边界谓词"和"文本特征谓词"规则段(若同时具有)一起和路径表达式组合作为一种组合方式,因为二者均不会使抽取规则对网页结构的依赖程度有很大的增强;另外,对于"大左右边界谓词"规则段,是在使用"小左右边界谓词"和"文本特征谓词"规则段形成的组合的基础上再增加该规则段作为一种新的组合,其目的是增强抽取规则的区分能力,满足以上原则的规则段组合称为语义对象的抽取规则。

抽取规则中含有的 HTML 页面的结构特征越多对页面结构的依赖程度越强,当其发生改变,抽取规则失效的可能性比较大[7],对于以面向显示为目的的 HTML 页面,尽可能使用显示特征和语义特征形成抽取规则可提高其适应性,图 2 给出了各语义对象从其规则段可获得的抽取规则形式,并从抽取规则对待抽取节点的覆盖能力和区分能力两方面分析其适应性,其中覆盖能力反映召回率,区分能力反映准确率。

下面以图中的"层"为单位,分析各种抽取规则的适应性:

"底层"抽取规则　覆盖能力比较强。这类规则主要含

有HTML页的DOM路径、显示特征、语义特征。原子对象尽管多选用了“小左右边界谓词”规则段,对网页结构的依赖也是很小的。“底层”抽取规则可能区分能力不足。

“中层”抽取规则 在“底层”抽取规则的基础上增加了位置特征,区分能力增强,覆盖能力下降。其中成员对象增加了“大左右边界谓词”规则段;原子对象和集合对象将“纯属性路径表达式”换为“属性序号路径表达式”。

“高层”抽取规则 只有普通原子对象和集合对象才有。该方式通过对“中层”抽取规则进一步增加位置特征,来提高区分能力,实质上是将“属性序号路径表达式”中使用属性作谓词的位置步也换用序号。相对于“中层”抽取规则,这种抽取规则覆盖能力进一步降低,区分能力进一步增强。

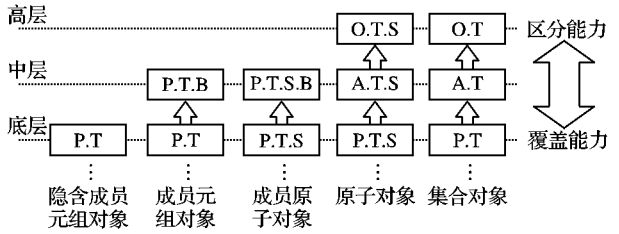


图2 各类语义对象的抽取规则及适应性

综上可知,在图2中,沿箭头方向,组合方式的覆盖能力降低,区分能力增强。

3.3 语义对象抽取规则的优化算法

优化的基本思想是:从准确率和召回率两方面出发,在保证准确率的前提下,选取具有最高召回率的抽取规则作为语义对象优化的抽取规则。

下面首先给出与规则优化算法相关的定义:

定义1 抽取规则蕴含。对于语义对象的抽取规则 R_i 和 R_j,如果用 $DP(R)$ 表示 R 的DOM路径、$CN(R)$ 表示 R 查询的起始节点(DOM树中的某个节点)、$P_k(R)$ 表示 R 的第 k 个位置步上的谓词表达式(由单个谓词或者关系为“AND”的多个谓词构成)、$L(R)$ 表示 R 位置步的个数,若 R_i 和 R_j 满足如下三个条件,则称 R_i 蕴含 R_j,并用 $R_i \Rightarrow R_j$ 表示。

1) $CN(R_i) = CN(R_j)$

2) $DP(R_i) = DP(R_j)$

3) $\forall k(P_k(R_i) \Rightarrow P_k(R_j))$

其中:$1 \leqslant k \leqslant L(R_i) = L(R_j)$。

对上述定义做如下说明:

对于条件1,由于在本系统中,抽取规则均使用相对路径。系统根据模式树中语义对象的嵌套层次,以自顶向下、自左至右的顺序,依次应用抽取规则定位出各种对象实例,即子对象的抽取规则是在父对象的抽取结果中实现定位。因此,兄弟关系语义对象的抽取规则具有相同的查询起始节点。

对于条件3,根据规则段的定义,用XPath语法表达的完整的语义对象抽取规则中,某个位置步若有谓词表达式,则只能由与显示相关的属性名和属性值的等式形成的谓词、文本特征谓词、小左右边界谓词、大左右边界谓词和序号(序号也可以认为是一种谓词,例如“para[3]”可以转换为“para[position()=3]”)五种类别的1个或多个构成(之间关系为“AND”),鉴于此可采用如下方法对条件3进行判断:用S(PredicateExpr)表示谓词表达式PredicateExpr所含谓词的集合,则 $P_k(R_i) \Rightarrow P_k(R_j)$ 当且仅当 $S(P_k(R_i)) \supseteq S(P_k(R_j))$。

根据上述定义不难发现,若 $R_i \Rightarrow R_j$,则对于样本页面 S 及其相似页面 $S1$,抽取规则 R_i 定位出的节点集是抽取规则 R_j 定位出的节点集的子集。

定义2 抽取规则重叠。用 SO_i 表示第 i 个语义对象,$S(SO_i)$ 表示 SO_i 满足规则段组合原则的规则段组合(即抽取规则)的集合,$N(S(SO_i))$ 表示该集合中抽取规则的个数,$S(SO_i)$ 中第 $k(1 \leqslant k \leqslant N(S(SO_i)))$ 个抽取规则 R_k,我们称 R_k 重叠,当且仅当对于样本页面 S 及相似页面 $S1$,$\exists SO_j(j \neq i)$,$S(SO_j)$ 中有抽取规则 $R_m(1 \leqslant m \leqslant N(S(SO_j)))$ 蕴含 $S(SO_i)$ 中的 R_k。

根据上述定义不难发现,若规则 R 重叠,则不准确。优化算法如下:

算法1 选取最优规则算法

输入:语义对象SO的规则段列表RawRuleList

输出:语义对象SO的最优规则OptimalRule

```
1)  Rule_Array = GetRule ( RawRuleList )
        //从规则段获得符合规则段组合原则的抽取规则;
2)  Order ( Rule_Array )
        //抽取规则从覆盖能力到区分能力的优先级别排序;
3)  L = GetLength ( Rule_Array )  //获得数组中规则的个数;
4)  i = 1                          //用于控制规则个数
5)  While ( Validate_ Rule_ Overlap ( Rule_Array[i] ) AND
      (i<=L))
6)  { i = i + 1
7)    if (i > L)
8)      Break }                    // 跳出 While 循环
9)  if (i<=L)
10)   return Rule_Array[i]
                  // 返回语义对象 SO 的最优规则,算法退出
11) else
12)   return       // 若 i > L 说明无可用最优规则,算法退出
```

算法描述:上述算法按从覆盖能力到区分能力的优先次序选出第一个“准确”的抽取规则作为语义对象的最优规则。函数Validate_ Rule_ Overlap(rule)用于判断规则是否重叠,若重叠返回TRUE,否则返回FALSE。所有语义对象优化完成后,将获得由各语义对象最优规则组成的最优规则列表。

3.4 基于XQuery复杂对象抽取规则的封装

系统使用XPath语法表达各语义对象的抽取规则,但XPath语句一次只能在DOM树中定位语义模式中一个语义对象的实例,而XQuery语句则可以一次定位所有语义对象的实例,所以将各语义对象的最优规则组装成一条完整的XQuery查询语句,作为复杂对象的抽取规则。

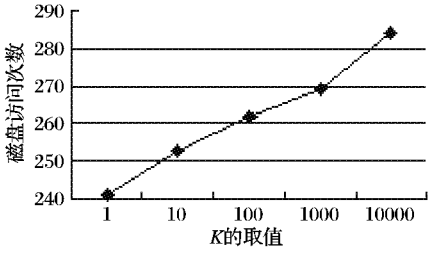
系统根据每个语义对象的最优规则按如下原则为其生成一个FLWR[8]表达式:

- 成员对象由于其个数不确定,生成一个“FR”形式的表达式(表示FOR子句和RETURN子句);
- 集合对象和原子对象生成一个“LR”形式的表达式(表示LET子句和RETURN子句);
- 无抽取规则的,仅生成一个构造算子。

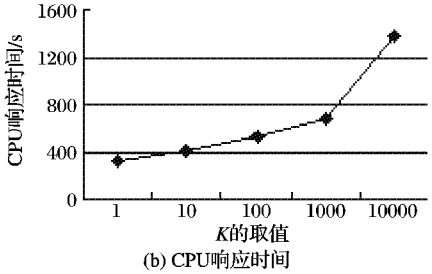
获得各语义对象的FLWR表达式之后,系统按模式树中语义对象的嵌套结构对这些表达式加以组织,形成用XQuery语法表达的复杂对象抽取规则。 (下转第158页)

空间数据集的R树高度均为3,节点大小为4 kB。

在实验中,要找出从旅馆→医院→商业街→广场→旅馆,距离最短的K个四元组,分别取K = 1,10,100,1 000,10 000进行实验。我们比较了随K值不同,CPU响应时间和磁盘访问次数,具体实验结果如图4所示。



(a) 磁盘访问次数



(b) CPU响应时间

图4 不同K值的算法性能比较

从实验结果可以看出,当K从1增加到10000时,磁盘访问次数只增加了大约18%,但CPU的响应时间增加了大约4倍。由此说明采用基于距离的平面扫描技术,可以过滤大量不满足查询条件的元组,有效减少磁盘访问次数,降低空间距离连接查询的总花费,但计算空间对象距离的CPU时间仍然很长,这部分算法还有待于进一步改进。

4 结语

多路空间距离连接查询是基于空间数据库的一种重要的查询,目前研究人员已经对两个空间数据集的距离连接查询,提出了一些算法。本文针对$n(n \geq 2)$个空间数据集的距离连接查询问题,提出了一种非增量递归算法,并采用基于距离的平面扫描技术,对候选集进行过滤,进一步降低查询处理代价。该算法可以对任意多个数据集进行距离连接查询,也可以查找距离最近的任意多个元组,有效地解决了多路空间距离连接查询的问题。实验结果表明,该算法能够有效减少磁盘访问次数和CPU的执行时间。

参考文献:

[1] CORRAL A, MANOLOPOULOS Y, THEODORIDIS Y, *et al*. Algorithms for processing K-closet-pair queries in spatial databases[J]. Data & Knowledge Engineering, 2004, 49(1): 67 - 104.

[2] PAPADOPOULOS A N, NANOPOULOS A, MANOLOPOULOS Y. Processing distance join queries with constraints[J]. Computer Journal, 2006, 49(3): 281 - 296.

[3] SHIN H, MOON B, LEE S. Adaptive and incremental processing for distance join queries [J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(6): 1561 - 1578.

[4] 肖予钦, 张巨, 陈荦, 等. 基于DJI分步实现的联机空间距离查询处理[J]. 国防科技大学学报, 2003, 25(6): 5 - 9.

[5] Census 2000 TIGER[EB/OL]. [2007 - 04 - 01]. http://www.census.gov./main/www/cen2000.html.

(上接第154页)

4 抽取测试结果

我们选择了3个网站对原型系统进行测试,并利用准确率和召回率对抽取结果进行量化(参见表1)。其中准确率 = $A/(A+B) \times 100\%$,召回率 = $A/(A+C) \times 100\%$,其中A代表抽取出的相关对象的个数,B代表抽取出的非相关对象个数,C代表未抽取出的相关对象个数,$A+C$代表相关对象总数,$A+B$代表抽取出的对象总数。

表1 抽取测试效果

名称	样本页面名称	是否可以抽取	学习次数	准确率/%	召回率/%	测试网页数目
Amazon	Top Sellers(TVs)	可以	1	100	98.3	12
			2	100	100.0	12
Web Robot	Web Robot	可以	1	99	100.0	1
			2	100	100.0	1
VLDB	VLDB 1989	可以	1	100	100.0	20

测试结论:系统在进行一次样本学习的情况下,准确率为99% ~100%,召回率为98.3% ~100%。在学习两次后,准确率和召回率都可达100%。说明该抽取方法具有样本学习次数少,准确率和召回率较高的特点。

5 结语

本文描述的Web信息抽取方法从准确率和召回率两个角度出发优化语义对象抽取规则,采用受限XML作为语义模型增强了输出格式的灵活性,利用XQuery表达复杂对象抽取规则使其具有通用性,引擎易于与基于Web的应用相结合。实验证明,该方法实用、有效。进一步研究的工作包括:提高系统自动化程度,在网页结构发生变化时自动修正现有的抽取规则等。

参考文献:

[1] LAENDER A H F, RIBEIRO-NETO B A, SILVA A S, *et al*. A brief survey of Web data extraction tools [J]. SIGMOD Record, 2002, 31(2): 84 - 93.

[2] SODERLAND S . Learning information extraction rules for semi-structured and free text [J]. Machine Learning, 1999, 34(1/3): 233 - 272.

[3] HAN W, BUTTLER D, PU C. Wrapping Web data into XML [J]. SIGMOD Record, 2001, 30(3): 33 - 39.

[4] SAHUGUET A , AZAVANT F . Building light - weight wrappers for legacy Web data-sources using W4F[C]// Proceedings of the 25th VLDB Conference. San Francisco: Morgan Kaufmann Publishers, 1999: 738 - 741.

[5] BAUMGARTNER R, CERESNA M, GOTTLOB G, *et al*. Web information acquisition with Lixto suite[C]// Proceedings of the 19th ICDE Conference. Los Alamitos: IEEE Computer Society Press, 2003: 747 - 749.

[6] W3C. Tidy [EB/OL]. [2005 - 01 - 08]. http://www.w3.org/People/Raggett/tidy/.

[7] LIU W, MENG X, MENG W. Vision-based Web data records extraction [C]// Proceedings of the 9th SIGMOD International Workshop on Web and Databases. Chicago: [s.n.], 2006.

[8] W3C. XQuery [EB/OL]. [2005 - 01 - 08]. http://www.w3.org/TR/xquery.