Web 信息抽取方法研究①

Web Information Extraction Method

韩存鸽　(武夷学院 计算机科学与工程系 福建 武夷山 354300)

燕　敏　(四川托普信息技术职业学院 计算机科学与技术系 四川 成都 611743)

摘　要： web 资源含有大量的有用信息，但由于它们欠结构化，不能为传统的数据库型查询系统所利用。如何将这些信息抽取出来，转化成结构化信息，供其它信息集成系统所利用，成为该顶域的研究热点。本文介绍了一个简单的 web 信息抽取模型，以及基于该模型研究。

关键词： 信息抽取 DOM 树 XML 数据

1　引言

随着计算机科学技术的发展，Web 已经发展成为一个全球的、巨大的、分布的、和共享的信息空间，为了应对信息爆炸带来的严重挑战，迫切需要一些自动化的工具帮助人们在海量信息源中迅速找到真正需要的信息。为了增强 Web 数据的可用性，出现了 Web 信息抽取技术。信息抽取技术最早是 G.Wiederhold 在《Mediators in the Architecture of Future Information Systems》[1]一文中提出的。它的本质是从 Web 页面所包含的无结构(HTM 格式)或半结构(XML 格式)的信息中识别用户感兴趣的信息 ,并将其转化为更为结构化为语义更为清晰的格式[2]。信息提取涉及到两个方面的因素，即(1)用户指定感兴趣的信息和待分析的文本集；(2)系统过滤文本集并以一定的格式输出匹配的信息。首先，仅仅依靠信息检索(Information Retrieval)并不能有效地实现信息提取的目标，因为信息检索只是找出满足一定检索条件的整篇文档或段落，人们仍然必须阅读所找到的每一个文档或段落才能获得所需要的信息。信息提取不仅查找信息，而且替用户理解信息，并按用户指定的方式输出信息。可以说，信息提取是"更高级的信息检索"。

2　信息提取系统如何工作

典型的信息提取系统的内部工作过程主要包括了如下几个步骤：

① 用一组信息模式(Info Patterns)描述感兴趣的信息。信息模式通常可表示为简单的一个句式，例如<公司名>“推出”<产品名>。系统可以针对某一领域的信息特征预定义好一系列的信息模式，存放在模式库中供用户选用。

② 对文本进行“适度的”(浅层、非完整的)词法、句法及语义分析，并作各种文本标引。这个过程通常包括识别特定的名词短语(人名、机构名、产品名、事件、地点等)和动词短语(事件描述、事实陈述)。这需要使用合适的词典、构词规则库等知识库的支持。

③ 使用模式匹配方法识别指定的信息(即找出信息模式的各个部分)。

④ 进行上下文关联、指代、引用等分析和推理，确定信息的最终形式。

⑤ 输出结果。出于效率的考虑，典型的信息提取系统通常包括一个预处理过程，目的在于过滤掉与提取目标不相干的文本；然后通过词法分析和标引，识别所有与提取目标相关的词汇(“关键词”识别与标引)；句法和语义分析只应用于所有包含了关键词的句子的集合，对每个句子的分析结果近似于该句子的语义框架表示，最后对这些框架进行合并、综合，便可得到所需信息的各种数据项。

3　信息抽取原理

本方法将信息抽取过程分为两个阶段：学习阶段

① 收稿时间:2008-11-17

和抽取阶段。

学习阶段：在 HTML 页面中没有模式，通过少量的 HTML 样本页面，用户根据实际的需求和选定的 HTML 样本信息具体情况定义模式信息，同时对样本页面进行适当的标记得到样本记录，系统根据样本页面和样本记录形成抽取的知识库(包含抽取信息的抽取规则和关联规则)

抽取阶段：利用样本学习阶段产生的抽取规则对 DOM(document object model 文档对象模型)树中节点集合定位那些符合抽取规则的节点。根据学习阶段定义的模式信息，建立数据库，同时根据知识库与样本页面相近的 HTML 页面进行信息抽取。将抽取出来的信息以数据库的方式存储和管理。由于选取对象关系模型，在抽取阶段可以利用对象关系数据库，这样抽取出来的信息符合用户的要求并且具有结构。

信息抽取模型如图 1 所示。信息抽取过程可描述如下：用户首先选择样本页面，在浏览样本页面的同时 HTML 的样本页面被 TIDY[3](一个能够把 html 页面转换为 xml 的功能强大的工具)转换为符合 XML 语法的文档，并被 Parser(一个对现有的 HTML 进行分析的快速实时的解析器)转换为 DOM 树，然后对树中的节点元素进行遍历和操作，通过创建概念模式 ，记录下网页反映的语义信息。有了概念模式，用户通过标记网页中相应的信息块，并将选择的信息块与概念模式中的语义项建立对应关系，学习引擎根据在用户帮助下建立的这种对应关系生成抽取规则。在抽取规则生成之后，我们就完成了信息抽取的第一阶段；这时，已经完成了样本学习，可以进行自动的信息抽取了，系统利用学习阶段产生的抽取规则，对文档进行信息抽取，并将抽取出的结果存储到分类库中，便于今后的信息集成或者查询。

无论是 HTML 文档还是 XML 文档，它们的 DOM 树结构都可以用上图表示，DOM 树主要由元素节点、属性节点、文本节点组成。元素就是 HTML 或 XML 标签，属性就是标记的属性，文本则是标签开始到对应标签结束或者从前一标签结束到下一标签开始之间的文本内容。

4　信息抽取的实现

4.1 生成关系模式

系统根据标记的信息产生各个属性响应的抽取规则，解析该文档，生成 DOM 树，然后根据映射关系生成关系模式。

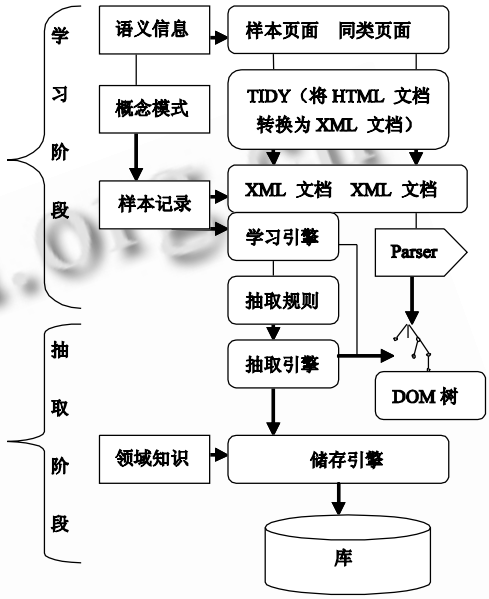

图 1　信息抽取模型图

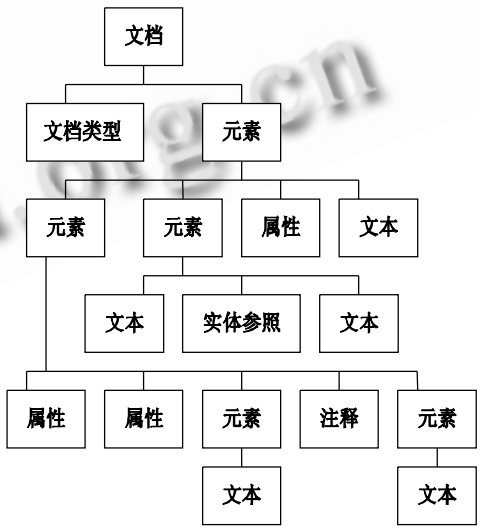

图 2　DOM 树语法结构图

具体映射规则如下：在生成 DOM 树的基础上①文档的根节点对应数据库中的一个表，并增加一个字段，字段名即为表名②如果一个元素节点的属性列表或子元素列表不为空，则该节点对应数据库中的一

个表，它的每个属性和子元素节点对应表中的各个字段 ③为文档根节点对应的表格增加一 个 ID 字段作为主键，为其它每个创建的表格增加一个 ID 字段和 PID 字段分别作为主键和外键；④文本节点不对应数据库表中的任何对象。

根据上述规则生成文档对应的关系模式的实现算法如下：

```
preOrderTraverse(Node n){
if(n 是元素节点类型){
  if(n 是文档根节点){ //处理文档根节点
    n 节点元素对应一个表;
      增加一个字段，字段名即为表名;
}
if(n 无属性)
  if(n 无子元素) return;
    取 n 节点元素，对应一个表，增加一个关键字段; //取该元素本身
    if(n 不是文档根节点) 取 n 的父节点元素，增加一个外键字段; //取父节点
    取 n 节点的属性节点，增加属性字段; //取属性节点
    取 n 节点的元素孩子节点，增加子元素字段; //取该元素的元素孩子节点
    加入一个新表;
    递归处理 n 节点的孩子节点;  }
    }
```

4.2 抽取 XML[4]数据存入数据库

抽取 XML 数据信息的过程是，解析 XML 文档，生成 DOM 树，递归遍历该 DOM 树，生成 XML 数据信息，实现算法如下：

```
preOrderTraverse(Node n,int pid){
  if(n 是元素节点类型){
    if(n 是文档根节点){ //处理文档根节点
      n 节点元素对应一个表格;
        为 n 元素绑定一个 id;
          增加外键 pid=-1，修改 pid=id;
            取 n 节点元素的文本节点内容; }
if(n 无属性)
  if(n 无子元素) return;
  取 n 节点元素，对应一个表格; //取该元素本身
    if(n 不是文档根节点){ //取父节点
    为 n 元素绑定一个 id;
    增加外键 pid，修改 pid=id;  }
    取 n 节点的属性节点内容，增加属性字段; //取属性节点
    取 n 节点的元素孩子节点的文本节点内容，增加子元素字段; //取元素孩子节点
    加入一个新表格;
    递归处理 n 节点的孩子节点;  }
      }
```

抽取算法的核心就是按照“从上到下，从左到右”的次序深度遍历 DOM 树的过程，在遍历的同时利用抽取规则对当前节点进行测试，得到符合条件的语义项暂存起来，当完成一个对象的全部语义项后进行组装，然后存入数据库。重复此过程直到遍历结束。

5 评价指标

信息抽取技术中对信息效率评价时使用的召回率(Recall Ratio)和准确率(Precision Ratio)对抽取结果进行量化。计算公式如下：

$$Precision\ Ratio = A/(A+B)*100\%$$

$$Recall\ Ratio = A/(A+C)*100\%$$

利用如上公式我们就能够较客观的抽取方法的实用性。公式中出现的字母的含义为：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的个数，C 代表未抽取出的相关对象的个数。

表 1　测试原型系统的网站

名称	网站地址	样本页面名称	测试网页数目
Amazon	http://www.amazon.com/	TOP Sellers	12
Vldb	http://www.acm.org/sigmod/dblp/db/conf/vldb	Vldb 2008	20
Web Robot	http://topic.csdn.net/t/20010429/10/108888.html	Web Robot	1

(下转第189页)

(上接第 174 页)

6 结束语

针对 Web 页中存在的大量无结构和半结构化信息，这些信息使用传统的数据库查询系统较为困难。我们设计了一种 Web 信息抽取方法，并具体描述了实现过程，将 Web 网页中的无结构和半结构化信息转换成 XML 文档，并在抽取规则中加入了语义,根据该抽取规则从 XML 文档中抽取出结构化数据并保存到中间库中。在将来的研究工作中，要进一步在抽取规则中明确语义，以及提高 Web 信息抽取过程的效率。

表 2 抽取测试效果的评价

名称	是否可以抽取	学习数目	准确率	召回率	测试网页数目
Amazon	可以	1	100%	98.3%	12
		2	100%	100%	12
Web Robot	可以	1	99%	100%	1
		2	100%	100%	1
Vldb	可以	1	100%	100%	20

参考文献

1 绍华,薛文玲,李天柱.基于 Web 的快速信息抽取技术.计算机应用, 2001,(7):24－26.

2 W3C:TidySpecification.http://www.w3.org/People/Raggett/tidy/.

3 Horstmann C S. Java2 核心技术,第 5 版.北京:机械工业出版社, 2001:40－50.

4 张清军,朱才连.基于主动学习 We 页面信息抽取.情报学报, 2004,(6):48－50.