

基于树结构的 Web 信息抽取

周 登,戴玉刚,付 涛

(西北民族大学 中国民族信息技术研究院,甘肃 兰州 730030)

摘 要:信息提取就是从大量的数据中检索出有用的信息,但一般的 Web 信息提取技术都是基于对 Web 上 HTML 文档的分析。文中提出了一种先将 HTML 转化为 XML 形式,再提取信息的方法。XML 是用于描述在 Internet 网上用于数据交换的数据文档的格式的一种语言标准,它将结构、内容和表现分离。数据可被 XML 唯一标识,从而有利于用户对数据的组织和检索。这种方法能够达到较高的正确率,同时随着文档的增大,方法也能够保证线性的时间复杂度。

关键词:Web;信息提取;XML 数据文档;树结构

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2009)09-0038-04

Extracting Web Data Using Tree Structure

ZHOU Deng, DAI Yu-gang, FU Tao

(Department of China Minorities Information Technology Institute,
Northwest University for Nationalities, Lanzhou 730030, China)

Abstract: Information retrieval is the technique that searches the useful information from abundant data. But the common way of web information retrieval is based on the analysis of HTML documents on the web. Provides a kind of method that transform HTML into XML first, and then get information. XML is a language standard, which is used to describe the format of data documents in data exchange. It makes the structure, content and representation separated. The data can be identified uniquely by XML. And then, it helps people organize and search the data. The proposed approach can effectively extract desired data with high accuracies and with linear complexity.

Key words: Web; information retrieval; XML data document; tree structure

0 引 言

随着 Internet 和 WWW 的迅速发展,任何利用 Web 数据进行生产或者研究的项目必然先遇到 Web 数据抽取的问题。页面中包含了大量与主题信息无关的导航条、广告信息、版权信息以及调查问卷等内容。这些内容不仅增加了人们浏览网页的时间,而且给基于网页主题信息抽取的研究工作带来困难。通过抽取主题信息不仅能够减少用户一半的浏览时间,而且用网页主题信息代替源网页,可以提高网页自动分类以及自动摘要等应用的性能。用于表达 Web 页面信息的 HTML 语言存在着与生俱来的缺点^[1]。HTML 的“标记”只是告诉浏览器软件如何显示所定义的信息,却不包含任何语义。因此 HTML 网页很难提供操纵和使用其数据,其中的数据信息很难被一般的应用程

序直接使用。

1 网页信息结构化

1.1 HTML 网页在信息抽取时的缺点

HTML 是通过大量的标记来定义文档内容以什么样的形式呈现在人们面前,可以说它是一种“显示描述”语言,它仅仅描述了 Web 浏览器应该如何如何在页面上布置文字、图形等,并没有对网页上最重要的东西——信息的本身含义进行描述。这些通过 HTML 表现出来的文字、图形内容很容易被人理解,而要计算机去理解这些标记内的文字的含义,就很困难了。

1.2 解析 HTML 成标签树

解析 HTML 成标签树结构以后,不但可以很容易取得想要的元素,同时也很容易将 HTML 转换成对应的 XML 文件。算法的关键如下:

1) HTML 中每个 Tag 都是将作为树中的一个节点存在的,每个 tag 都属于树中的某一层。

2) 辅助数据结构: 栈(stack)、List、HashTable。其中 HashTable[i] 是一个 List,用于临时存储第 i 层子 Tag。

收稿日期:2008-12-25;修回日期:2009-03-22

基金项目:国家科技计划资助项目(2005DIB6J174)

作者简介:周 登(1983-),男,湖北荆州人,硕士研究生,研究方向为网络数据库和多媒体技术;戴玉刚,硕士生导师,教授,研究方向为网络数据库和多媒体信息处理。

3)顺序扫描 HTML 文本,当遇到“<A~Z>”这样的标志,表示可能是一个 Tag,调用 GetTag()函数对此段代码进行解析,解析出 Tag 名,Tag 属性等等。如果返回值不为空,那么将返回值入栈。并且记录此 Tag 的开始位置。

4)遇到</A~Z>这样的标志,表示可能是某个 Tag 的结束。解析出此结束标志的 Tag 名。如果在栈中找到与此结束标志名同名的元素(此元素属于栈中第 iLevel 层),那么表示找到匹配的 Tag。则 Tag 出栈,将 HashTable[iLevel+1]到 HashTable[maxLevel]中的所有元素取出作为此 Tag 的子节点,放入第 HashTable[iLevel]中。并记录 Tag 的结束位置。

5)对于<Tag> </Tag>之间的字符串,将其作为特殊的 HTML Text Tag 处理。出栈和入栈操作与普通 Tag 类似。当栈为空的时候表示最后一次出栈的 Tag 给根节点。

伪代码如下:

```
public void Parse()
{
    char ch = GetCurrentChar();
    while (! Eof())
    {
        if (ch == '<')
        {
            ch = MoveNext();
            if ((ch >= 'A') && (ch <= 'Z') || (ch == '!'))
            {
                iBeginPos = Index;
                HtmlTag tag = GetTag();
                if (tag != null)
                {
                    if (m_CurrentText.Length > 0)
                    {
                        Stack.Push(new HtmlTextTag(m_CurrentText));
                    }
                    tag.BeginPos = iBeginPos;
                    Stack.Push(tag);
                }
            }
            ch = GetCurrentChar();
            if (ch == '/')
            {
                tagName = GetTagName();
                if (FindInStack(tagName))
                {
                    PopTag(tagName);
                }
            }
        }
    }
}
```

```

    }
    else
    {
        m_CurrentText.Append(GetCurrentChar());
    }
    ch = MoveNext();
}
if (Stack.Count > 0)
{
    HtmlTag tag = null;
    while (Stack.Count > 0)
    {
        tag = Stack.Pop();
        PopTag(tag);
    }
    if (tag != null)
    {
        m_listRoot.Add(tag);
    }
}
.....
private void PopTag(HtmlTag tag)
{
    int iLevel = Stack.Count;
    for (int i = iLevel + 1; i < m_iMaxLevel; i++)
    {
        for (j = 0; j < HashTable[i].Count; j++)
        {
            tag.Children.Add(HashTable[i][j]);
        }
    }
    private HtmlTag GetTag()
    {
        if (“如果发现是 <! - 开头的元素”)
        {
            SkipComment();
        }
        HtmlTag tag = new HtmlTag();
        tag.Name = GetTagName();
        tag.Attribute = GetTagAttribute();
        return tag;
    }
}
```

1.3 网页信息抽取与 XML

XML(Extensible Markup Language)和 HTML 一样,同样来源于 SGML,但 XML 是一种能定义其他语言的语言。W3C 对 XML 作了如下描述:XML 描述了一类被称为 XML 文档的数据对象,并部分描述了处理它们的计算机程序的行为^[2]。XML 最初设计的目的是弥补 HTML 的不足,以强大的扩展性满足网络信

息发布的需要,后来逐渐用于网络数据的转换和描述。它将结构、内容和表现分离。数据可被 XML 唯一标识,从而有利于用户对数据的组织和检索。在高效可扩充方面 XML 支持复用文档片断,使用者可以发明和使用自己的标签,也可以与他人共享,可延伸性大。在 XML 中,可定义一组无限量的标准,可以有效地进行 XML 文件的扩充。XML 具有卓越的性能,它具有四大特点:优良的数据存储格式、可扩展性、高度结构化以及方便的网络传输。因为 XML 能针对特定用户的应用定义自己的标记,这就使 XML 能够在网页信息抽取时起到很大的作用。一个具有正规格式的 XML 文档的形式是一个简单的层次树,每个都有且仅有一个根节点,称为文档实体,或文档根^[3]。这个节点总是包含子元素树,子元素树的根被称为文档元素。这个元素是这个树中其它所有元素的父元素,而且它可能不包含在其它任何元素当中。

2 信息抽取

组件对象与一般意义上的对象既相似也有区别,一般意义上的对象是一种把数据和操纵数据的方法封装在一起的数据类型的实例,而组件对象使用接口 (Interface) 而不是方法来描述自己并提供服务。所谓接口,其精确定义是“基于对象的一组语义上相关的功能”,实际上是一个纯虚类,真正实现接口的是接口对象 (Interface object)。DOM 为我们提供的访问 XML 文档信息的媒介是一种分层对象模型,而这个层次的结构,则是一棵根据 XML 文档生成的节点树。一个 XML 分析器,在对 XML 文档进行分析后,不管这个文档有多简单或者多复杂,其中的信息都会被转化成一棵对象节点树。在这棵节点树中,有一个根节点,即 Document 节点,所有其他的节点都是根节点的后代节点。节点树生成之后,就可以通过 DOM 接口访问、修改、添加、删除、创建树中的节点和内容。对于 XML 应用开发来说,DOM 就是一个对象化的 XML 数据接口,最基本的 XML 开发通常都要使用它^[4]。简单地说,DOM 就是一组对象的集合,通过操纵这些对象,就能操纵 XML 和 HTML 数据。DOM 的核心概念是 Node(节点)。XML 的每种结构如元素和属性都用它来表达。

下面是一个 XML 文档片段:

```
<? xml version="1.0" encoding="gb-2312" ? >
<addressbook>
  <person sex="male">
    <name>张三</name>
    <email>zhs@xml.net.cn</email>
  </person>
  <person sex="male">
    <name>李四</name>
    <email>ls@xml.net.cn</email>
  </person>
</addressbook>
```

DOM 树的顶级节点是 Document。对于一个 XML 文档是唯一的,代表 XML 文档。它继承自 Node。Document Element 是 Document 的顶级节点。Element 对象代表 XML 元素,它从 Node 对象继承过来。树模型如图 1 所示。

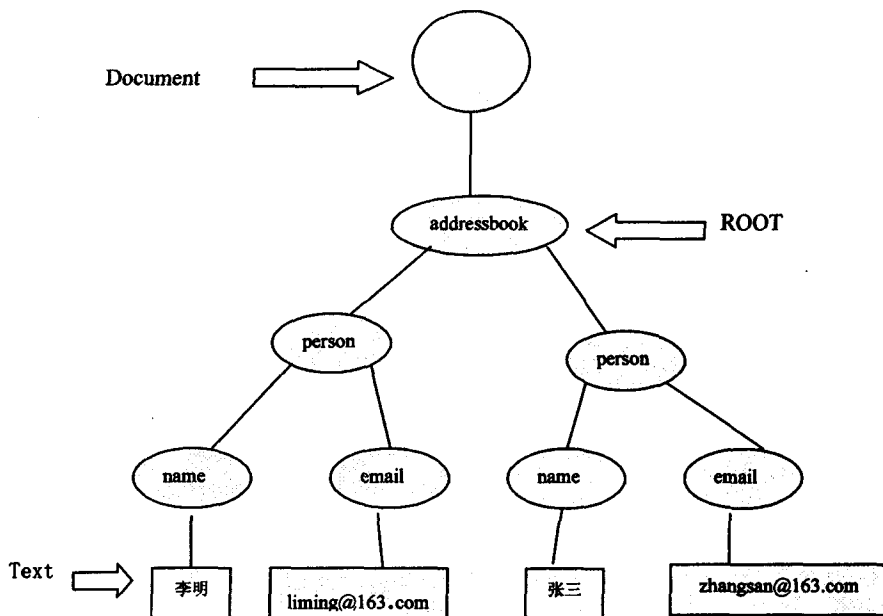


图 1 树模型

在这棵文档对象树中,文档中所有的内容都是用节点来表示的。一个节点又可以包含其他节点,节点本身还可能包含一些信息,例如节点的名字、节点值、节点类型等。文档中的根实际上也是一个元素,之所以要把它单独列出来,是因为在 XML 文档中,所有其他元素都是根元素的后代元素,而且根元素是唯一的,具有其他元素所不具有的某些特征。事实上,DOM 中还包含注释、处理指令、文档类型、实体、实体引用、命名空间、事件、样式单等多种对象模型^[5]。文档对象模型利用对象来把文档模型化,这些模型不仅描述了文档的结构,还定义了模型中对象的行为。换句话说,在上面给出的例子里,图中的节点不是数据结构,而是对

象,对象中包含方法和属性。在 DOM 中,对象模型要实现:用来表示、操作文档的接口、接口的行为和属性、接口之间的关系以及互操作^[6,7]。可以说 DOM 是一种在浏览器内编程的模型,同时也是 XML 的主要接口,它与语言 and 平台无关,它是基于树的 API,它把所有数据以父子的节点层次结构装入内存构成一棵树,这些节点的类型可以是元素、文本、属性、注释或其它,可以读取、创建、删除和编辑 XML 数据。DOM 的“一切都是节点(everything-is-a-node)”。XPath 描述在可扩展标记语言(XML)文件中使用基于文件逻辑结构或层次路径的地址语法定位和处理项目的方法。如果每个表达式必须理解典型 XML 标记及其在文件中的顺序,XPath 将使得编程表达式的书写变得容易。XPath 也允许程序员在抽象的更高层次处理文件。可扩展样式表语言转换(XSLT)和 XPointer(SML 指针语言)都使用 XPath 并将其作为自身的一部分。XPath 使用 XML 信息集(Infoset)定义的信息抽取,可用于 XML 之外的其它文本。XPath 使用的语法相似于寻找特定地理位置所用的方向信息集。XPath 与更早的语言之间的主要差异在于 XPath 指定一条路径而不是指向字符、单词或其他元素的集合或序列。XPath 使用概念节点(路径地址的起点)的概念,这一概念表达了 XML 信息集定义的逻辑关系。使用归纳法可以生成用于 Web 信息提取的 XPath 规则。这些规则的信息提取能力依赖于页面结构,而对于已经规范化的 XML 信息易如反掌,几乎可以直接 following-sibling 轴的 XPath 规则提取信息。

3 信息抽取技术的评价标准

信息抽取技术的评测起先采用经典的信息检索(IR)评价指标,即召回率(Recall)和查准率(Precision),但稍稍改变了其定义。经修订后的评价指标可以反映 IE 可能产生的过度概括现象(Over-generation),即数据在输入中不存在,但却可能被系统错误地产生出来(Produced),就 IE 而言,召回率可粗略地被看成是测量被正确抽取的信息的比(fraction),而查准率用来测量抽出的信息中有多少是正确的。计算公式如下:

$P = \frac{\text{抽出的正确信息点数}}{\text{所有抽出的信息点数}}$

$R = \frac{\text{抽出的正确信息点数}}{\text{所有正确的信息点数}}$

两者取值在 0 和 1 之间,通常存在反比的关系,即

P 增大会导致 R 减小,反之亦然。评价一个系统时,应同时考虑 P 和 R ,但同时要比较两个数值,毕竟不能做到一目了然。许多人提出合并两个值的办法。其中包括 F 值评价方法:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

其中 β 是一个预设值,决定对 P 侧重还是对 R 侧重。通常设定为 1。这样用 F 一个数值就可看出系统的好坏^[3]。

4 结束语

文中介绍了一个基于 HTML 的网页信息抽取解决方法。最终目的是将隐藏在 HTML 网页中的关键信息抽取出来并表达为结构化的、扩展性很强的 XML 文档。以 DOM 树的形式表现出来,由于大部分网站是由一个后端的数据库和一些 HTML 的模板所驱动,网页上信息的表示结构都相同或非常相似,因此,利用基于记录结构相似的启发知识具有较好的通用性^[3]。此外,还利用相同结构中相同位置的信息作为词法模式学习样本,获得的词法模式不仅可以帮助更加精细地抽取所需信息,还可以有助于适用于不同表示风格的网页甚至是非结构化文本。

参考文献:

- [1] 王 茹,宋瀚涛,陆玉昌.网页数据自动抽取系统[J].计算机工程与应用,2004(19):135-137.
- [2] 赵金仿,赵 艳,缪建明.网页信息抽取及其自动文本分类的实现[J].计算机技术与发展,2008,18(10):38-39.
- [3] Laender A H F, Ribeiro-Neto B A, Da Silva A S, et al. A Brief Survey of Web Data Extraction Tools [J]. SIGMOD Record, 2002, 31(2): 84-93.
- [4] 李效东,顾毓清.基于 DOM 的 Web 信息提取[J].计算机学报,2002,25(5):526-533.
- [5] 卢 睿.基于 XML 的 Web 信息抽取研究[D].大连:大连海事大学,2005.
- [6] 仲 华,崔志明.基于 XML 的信息抽取和多层向量空间技术研究[J].计算机技术与发展,2007,17(7):49-52.
- [7] Wessman A, Liddle S W, Embley D W. A generalized framework for an ontology-based data-extraction system[C]//The 4th International Conference on Information Systems Technology and its Applications. Palmerston North, New Zealand:[s. n.], 2005:239-253.