基于页面标签的 Web 结构化数据抽取*)

任仲晟　薛永生
(厦门大学计算机科学系　厦门 361005)

摘　要　本文研究了从 data intensive 类型的 Web 页面中提取结构化数据的问题,提出了基于页面标签的数据抽取算法。该算法先根据标签的显示位置及其大小判断不同标签元素之间的嵌套关系,并构造简化的 HTML 树 SimHTree,有效地减少了识别数据记录的时间。在此基础上,提出子串匹配调整算法,对数据记录进行识别,标识数据项。实验表明,该算法是有效的。

关键词　Web 数据抽取,Web 挖掘,结构化数据,信息抽取

Structured Data Extraction Based on Web Page Tags

REN Zhong-Sheng　XUE Yong-Sheng
(Department of Computer Science, Xiamen University, Xiamen 361005)

Abstract　This paper studies the problem of structured data extraction from data intensive Web pages. A novel approach based on page tags is proposed to solve the problem. The proposed method identifies the nesting relationship between different page tags according to the visual display location and the size on the screen and constructs the corresponding simplified HTML tree (SimHTree for short), which reduce the time cost for the identification of data record. As the second step of the data extraction problem, substring match and adjustment algorithm is proposed, which identify the data record and data item. Experimental results show that the proposed method is effective and efficient.

Keywords　Web data extraction, Web mining, Structured data, Information extraction

1　引言

随着 Internet 的快速发展,Web 已经发展成为一个巨大的、分布式的和共享的信息资源。目前 Web 数据大都以 HTML 的形式出现。由于 HTML 缺乏对数据本身的描述,各种标签只是告诉浏览器如何显示它所描述的信息,而不包含清晰的语义信息,是一种半结构化的数据,这使得由 HTML 描述的 Web 页面只适合人类的浏览,应用程序无法直接解析并利用 Web 上的海量信息。为了增强 Web 数据的可用性,出现了 Web 信息抽取技术,它通过包装现有 Web 信息源,将网页上的信息以更为结构化的方式抽取出来,为应用程序利用 Web 中的数据提供了可能。现有的 Web 信息抽取技术不但可以直接定位到用户所需的信息,而且采用一定的方式增加了语义和模式信息,为 Web 查询提供了更为精确的方法,使 Web 信息的再利用成为可能,因此有着明显的优势和广阔的前景,是当今数据库领域的研究热点。

2　相关工作

目前关于 Web 数据抽取的工作可以大致分为以下几个类别。

1)基于语言的 Web 数据抽取,通过提供一种专门的模式说明语言(specification language),定义抽取模式。此类代表有 WICCAP[4],Lixto[11]等。

2)基于本体论(ontology)的数据抽取,通过引入领域类的本体知识以及一些启发式规则,辅助抽取过程。文[10,12]等对此进行了描述。

3)基于包装器(wrapper)学习的数据抽取,通过有监督的机器学习等方法,生成转换规则,需要人工提供学习的正例和反例。此类代表有 Stalker[13],WIEN[14]等。

以上的划分方法并不是唯一的,也可以 Web 信息抽取过程是否是自动的,划分成全自动、半自动、人工等;同时这种划分方法也不是完全的,一些 Web 信息抽取技术兼用了几种方法。文[1～3,5,6,8,15～17]等都提出了一些新颖有效的 Web 信息抽取系统。

由于 Web 界面的种类繁多,且信息抽取目的也不尽相同,不存在一种 Web 信息抽取系统,能够适应这种千变万化的应用环境。本文所讨论的 Web 信息抽取技术,针对于

1)data intensive 类型的 Web 页面;

2)同时要求这种页面上必须存在多个相似数据区域,在一些文献中亦称之为 list page。

本文之所以做出这两点假设,是有其现实意义的:

1)目前很多 Web 页面是动态生成的,通常都是通过预先定义一系列的模板(template),然后根据用户的需求,查询后台数据库,用数据库中的数据填充这些模版,然后响应用户的请求。其过程可以简单描述成图 1。因此,这种类型的页面对于数据集成等现实应用具有重要意义,抽取准确度也相对较高。

2)目前一些 Web 信息抽取技术,要求用户提供多个样本页面用于模式的查找。而事实上,大量的单个 Web 页面中就存在若干个相似的数据区域。对于这样的 list page,单个页

*)国家自然科学基金(50474033)、福建省自然科学基金(A0310008)、福建省重点科技项目(2003H043)。任仲晟　硕士研究生,主要研究方向为数据库理论与应用、数据仓库、数据挖掘等;薛永生　教授,主要研究方向为数据库理论与应用、分布式数据库、数据仓库、数据挖掘、网络技术等。

面就足够用于指导信息的抽取过程。

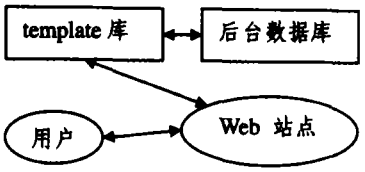

图 1 data intensive 页面生成过程示意图

本文其它部分组成如下：第 3 部分给出基于页面标签的 Web 结构化数据抽取算法；第 4 部分给出实验验证；最后为结论。

3 基于页面标签的 Web 结构化数据抽取

3.1 基于标签的 Web 数据抽取概述

本文设计的 Web 数据抽取系统，由以下 4 个主要部分构成：

1)SimHTree 树生成模块。该模块主要负责清理简化 HTML 源文件，方便后续抽取过程。

2)候选区域的确定。利用 SimHTree 树生成模块得到的信息，筛选确认含有待抽取信息的数据区域 CandidateRegion。

3)数据项的确认过程。在候选区域的基础上，通过比较不同区域之间的差别，进行匹配调整操作，最终获得待生成的数据库的表模式信息，即数据项的确定。

4)信息的提取。根据数据库的表模式信息，以及调整过的数据区域信息，完成 Web 数据的抽取，写入数据库中。

3.2 基于元素位置的 HTML 树的构造

HTML 文档中，每个元素由一个开始标签(opening tag)，结束标签(closing tag)来标志。由于 Web 浏览器(Internet Explorer 等)的容错性，对于某些错误标记的 HTML 文件，例如缺少匹配的结束标签等等，仍然能够正常显示在界面上，而这将影响正确区分 HTML 节点层次性，从而影响后续的数据抽取。同时，HTML 文档中，有相当一部分的内容，对于信息的提取没有意义，如一些脚本语言、标签属性，以及部分类型的标签等等。因此，在构造简化的 HTML 树(SimHTree)之前，有必要对其进行清理简化。

本文采取的方法，充分利用了元素在界面上的相对位置，可以准确地标记元素之间的嵌套关系。同时根据以下规则，对页面标签元素进行简化：

规则 1. 标签属性是可以删除的；

规则 2. 注释、脚本语言、命名空间、 等内容是可以删除的；

规则 3. 标签是可删除的，当且仅当其内容为空。这条规则是递归的，用于消除冗余的嵌套标签；

规则 4. select，input 等标签及其相关标记是可删除的，如与 select 相关的 option 等；

规则 5. 类型为 hidden 的标签是可删除的。

在上述规则的基础上，给出可删除集合 del 的定义。

定义 1 可删除元素集合 $del=\{x|x$ satisfies rules defined above$\}$

下面给出 SimHTree 构造算法。

```
Procedure BuildSimHTree(prePtr, HTMLfile)
  {
    for each element curElem in HTMLfile do
    {
      if (curElem in del) then
      {
        delete curElem;
        while (outerElem's content is null) do
          delete outerElem//递归删除外层空标记
        break;
      }
    getRect (curElem);
    //获取标签所占据矩形面积的位置
    if (curElem is contained in it's parent's rectangle) then
    {
    insert into tree as next child with the same parent;
      }
    else
    {
      prePtr=prePtr. parent;
    }
    prePtr=curElem;
  }
}
```

算法顺序读入 HTML 文件内容，对于可删除标签，则删除，并依次检查其外层标签，循环这一过程；否则根据占据的大小位置等信息，依次往 SimHTree 中增加节点。

考虑到实际数据往往在界面上呈现一定的分布，可以将位置作为参数，用于删除不必要的标签，达到优化的目的。例如对于左边为导航条、下部为版权信息等信息的 Web 页，位于这些区域的标签亦可以直接删除。

3.3 候选数据区域挖掘

在构造 SimHTree 树的同时，已经获得了每个标签的位置及其大小，这给挖掘候选数据区域提供了依据。

定义 2 候选数据区域 CandidateRegion，由若干个 SimHTree 中的节点构成，这些节点构成一条数据记录，并具有以下性质：

1)这些节点具有相同的父节点或者是同一代的兄弟节点；

2)这些节点在 SimHTree 中是相邻的。

直观上来看，每个候选数据区域可以理解成一条有待提取的数据记录，其位置上的相邻，体现了一条记录间不同数据项之间的相关度。同时由于某些数据项的可选性出现，使得某个相对较大的候选数据区域出现的次数不少于其子区域，于是有下面这个性质。

性质 1 相似候选数据区域的出现频度不小于各自区域内部子区域的出现频度。

根据这个性质，我们可以利用聚类的方法，对这些候选数据区域进行分组，依据出现的频度信息，得到有效的数据区域集合。

```
Procedure DataRegionMing (k)
{
    group tags into m clusters such that size difference between any two tags in the same cluster less than pre-defined threshold k and the tags share the same parents;
    sort m cluster according to the size of the cluster and the average size of regions in the cluster in decreasing order and the size of the cluster is prior to the average size of regions;
}
```

根据上述性质，m 个类的排序将呈现以下趋势：区域面积大小递减，类大小递减。

3.4 数据记录的模式生成

可以认为每个类的内部是一组记录。由于允许某些数据项的选择性出现，因此需要先生成符合类间绝大多数数据记录的记录模式 Record Model，许多文献中也将这个步骤称之为树匹配。由于我们讨论的网页是通过模版加后台数据库数据的方式生成的，因此不同记录之间的区别，以不同数据项的值的不同为主，同时允许有少量的数据项数目的差别。这些特征反映在这些记录所对应的子树上，就是这些子树的叶子节点(存储了具体的文本信息等值)的不同，或者个别标签节点的缺失。

因此，我们采取自顶向下、宽度优先的方式，进行子树的匹配。由于已经对网页根据标签的大小进行了分类，因此大大减少了需要比较的次数，从而降低了算法所需的时间。

模式生成算法可以描述成2个步骤：先是对子树(数据区域)进行层次编码，然后进行不同编码之间的匹配调整。

编码方式如下：对于非叶子节点，采取标签名称＋儿子个数的方式，对于叶子节点，填入指向数据值的指针，并采取辅助表，存储值及其指针。对于同一个层次的编码，按照相对于父节点的顺序进行，因为子树是有序的。对于缺失的部分，填充特殊字符＃。层次间的分隔符为/。编码举例如下：p2/a0b1/＃cptr，p2/a0b3/cptrg0cptr。

匹配调整算法思想如下：设置2个集合 matched 和 rematched，并初始化为空。如果进行匹配的两个根节点不同，则查阅类中其他子树的根，删除那个根节点不同的子树。如果某个非叶子节点不匹配，则考虑下面3种情况：

情形1：需要增加的节点落在匹配的节点之间(已匹配的节点之间没有其他节点)。例如，假设模式串为 p/abc，待匹配的串为 p/bdc，其中 p，a，b，c，d 等字符为标签的简写，下同。

情形2：需要增加的节点，落在已匹配节点的之外(依附于模式串的前端或者尾部)。例如，假设模式串为 p/abc，待匹配的串为 p/cd。

情形3：需要增加的节点，和已匹配节点之间的子串交错。例如，假设模式串为 p/abc，待匹配的串为 p/adc。

对于情形1，2，可以更新模式串，得到新的模式串，即上述例子新生成的两个模式串为 p/abdc 和 p/abcd。对于情况3，则将待匹配子串放入 rematched 集合，等待下轮的继续匹配。事实上，情形1，2可以合并，只需判定代插入串与模式串的相对位置而定。

下面给出匹配调整算法。

```
Procedure StrMatch (Strm,Strn)
{//算法的输入为2个串，其中 Strm 为模式串，Strn 为当前待匹配的串；
令 ptr1，ptr2 分别指向 Strm，Strn；
if ( * ptr1! = * ptr2) then
   {
    if match for the first time then
       reselect Strm;
    else delete Strn from CandidateRegion;
   }
else
{
    ptr1＋＋;
    ptr2＋＋;
    mptr1＝ptr1;
    while (mptr1! =null or ptr2! =null)
    {
      if ( * ptr1＝ * ptr2) then
      {
      addmap(ptr1,ptr2);
      ptr1＋＋;
      ptr2＋＋;
      mptr1＝ptr1;
    }
    else if (ptr1! =null)
         {ptr1＋＋;}
        else { adddiff(ptr1,ptr2);
               ptr1＝mptr1;
               ptr2＋＋;
             }
   }
switch check ()
{
case 1;
{update Strm; update Strm; update matched; matched＋＝Strn; candidateRegion-＝Strn;break;}
case 2;
{update Strm; update Strm; update matched; matched＋＝Strn; candidateRegion－＝Strn;break;}
case 3;
{candidateRegion－＝Strm;rematched＋＝Strm;}
}
}
Procedure check (map,diff)
{//算法的输入为2个集合，map 为模式串 Strm 和 Strn 之间的映射关系集合，diff 为不同字符之间的映射关系集合；
validate the position of mismatched letter according to pairs in map and diff;
return 1,2,3 respectively in accordance with the three cases discussed above;
}
```

在进行匹配的过程中，对相应的串信息进行更新。匹配调整过程结束后，根据模式串就可以得到关系数据库中的表模式，从而再根据模式，填充具体的数据实例，完成半结构化数据到结构化数据的抽取。

4 实验

在本文所述算法基础上，我们实现了一个原型系统，实验中所采用的页面数据均来自实际网站。

表1为系统抽取得到的数据(数据来自于卓越网(http://www.joyo.com/))。由于篇幅的限制，作了适当的截取和修改。其中的缺失值表示该数据项为可选的，原 Web 数据中当前记录不含有该值。

表1 抽取得到的结构化数据

乔家商学院	三石		市价	28.8
Word 排版艺术	侯捷著	img	市价	48.00
社区物业管理	潘茵编		市价	25.00
电子工业出版社纪念笔			市价	15.00
中国电子工业年鉴	本书编委		市价	368.00

表2为系统抽取不同网站以不同关键字搜索得到的不同响应页面(随机采样3个网站，查看相应的数据项识别率)。从中可以看到，系统的识别率是理想的。

表2 抽取得到的准确率统计

数据源	网页	数据项	正确	识别率
joyo.com	1	78	70	0.897
Amazon	1	80	57	0.713
Yahoo	1	30	23	0.767

同时，也与著名的 MDR 算法进行了比对。实验中发现，MDR 针对某些站点的网页，容易发生严重错误(对网页标签结构的识别问题)而挂起，本算法在容错性方面具有一定的优势。在识别率方面，两种算法都不同程度上受到了不同网站的页面模板的影响。

结论 本文研究了从 data intensive 类型的网页中提取数据的问题。和现有的一些通用的 Web 数据抽取系统不同，本文仅针对单一类型的网页，从而能够更好地利用这类网页的一些特征，提高抽取的精度和速度。实验结果表明，本文提出的算法是有效可行的。

在实验中也发现一些数据项被错误的划分，这种情形发生在标签节点有多个可匹配选择，而其叶子节点的文本值等信息不一致的情况下。在下一步的工作中，将针对这种情形进行研究，以进一步提高算法的精度。

参考文献

1 Wang Jiying, Lochovsky F H. Data-rich Section Extraction from HTML pages. In: Proceedings of the 3rd International Conference on Web Information Systems Engineering (WISE'02)

2 Estievenart F, Meurisse J R, Hainaut J L, et al., Semi-automated Extraction of Targeted Data from Web Pages. In: Proceedings of the 22nd International Conference on Data Engineering Work-

shops (ICDEW'06)
3 Chuang Shui-Lung , Hsu J Yung-jen. Tree-structured Template Generation for Web Pages. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)
4 Li Zhao, Ng Wee Keong. WICCAP: From Semi-structured Data to Structured Data. In: Proceedings of the 11th IEEE International Conference and Workshop on the Engineering of Computer-based Systems (ECBS'04)
5 陈澄,徐宏炳,王能斌. 基于标记图的 Web 数据模型. 计算机学报,1999,22(3)
6 李效东,顾毓清. 基于 DOM 的 Web 信息提取. 计算机学报,2002,25(5)
7 Mundluru D, Katukuri J R, et al. Automatically Mining Result Records from Search Engine Response Pages. In: Proceedings of the Fifth International Conference on Data Mining (ICDM'05)
8 Joshi A, Todwal S. Evolutionary Machine Learning for Web Mining. IEEE, 2003
9 鲁明羽,孙建涛,陆玉昌. 一种基于联想的网页推荐方. In: Proceedings of the 5th World Congress on Intelligent Control and Automation, Hangzhou, P R China, June 2004
10 高军,王腾蛟,杨冬青,等. 基于 Ontology 的 Web 内容二阶段半自动提取方法. 计算机学报,2004,27(3)
11 Baumgartner R, Flesca S, Gottlob G. Visual Web Information Extraction with Lixto. In: Proceedings of the 27th VLDB Conference, Roma, Italy, 2001
12 陈兰,等. 一种新的基于 Ontology 的信息抽取方法. 计算机应用研究,2004
13 Muslea I, Minton S, Knoblock C. A Hierarchical Approach to Wrapper Induction. In: Proc. Third Ann Conf Autonomous Agents, 1999
14 Kushmerick N. Wrapper Induction: Efficiency and Expressiveness. Artificial Intelligence, 2000
15 Meng Xiaofeng, Wang Haiyan, Hu Dongdong, et al. SG-WRAM Schema Guided Wrapper Maintenance: A Demonstration. In: Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
16 Meng Xiaofeng, Hu Dongdong, Li Chen. Schema-guided Wrapper Maintenance for Web-Data Extraction. WIDM'03, November 2003
17 Meng Xiaofeng, Wang Haiyan, Hu Dongdong, et al. A Supervised Visual Wrapper Generator for Web-Data Extraction. In: Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC'03)

(上接第 91 页)

在此只详述发送中的缓冲队列。

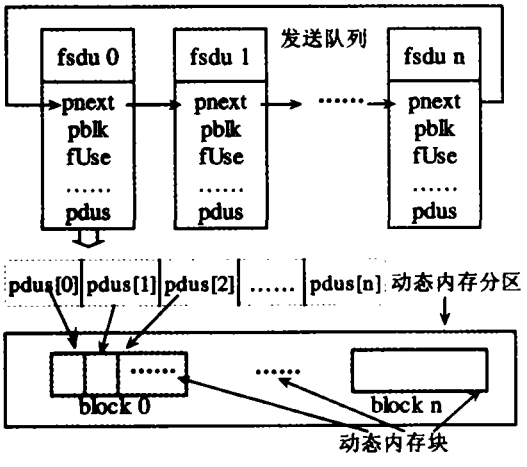

图 3 发送队列缓冲区

发送模块的队列缓冲区用于存放待发送的数据,当系统处于退避状态或各种帧间间隔时,可以继续处理来自 LLC 层的数据,将其放入缓冲队列中,等待发送。当收到上一帧传输完成的确认消息后,再将缓冲区队首的数据取出,准备发送。这样既不会对正在发送的数据产生干扰,又提高了执行效率。本方案采用单项循环串表加 3 个全局指针来实现这一功能。

基本数据类型定义如下:

```
typedef FragSdu SduQueue[Max_Frag_In_Q];
SduQueue txQ;
```

整个队列由一个 SduQueue 类型的数组构成,数组中的每个元素都是 FragSdu 结构体,每个 FragSdu 中的 pnext 域都指向下一个 FragSdu,最后一个 FragSdu 的 pnext 指向第一个 FragSdu,这样形成的队列实际上是一个单向闭合串表。

三个全局指针:

```
FragSdu * txQFreeList;/ * 指向未使用的 FragSdu * /
FragSdu * txQFirstList;/ * 指向队首的 FragSdu * /
FragSdu * txQPutList;/ * 指向将被释放的 FragSdu * /
```

关于队列的整个操作如下:

1)在收到上层模块(Prepare_MPDU)发来的消息后,将消息中的 FragSdu 提取出来,复制到 txQFreeList 所指向的 FragSdu 中,将队列中的该 FragSdu 标记为"已用"(令结构体的 fUse 域为 1),让 txQFreeList 指向队列中的下一个 FragSdu。

2)在一个 MSDU 传输完毕后,判断 txQFirstList 所指向的 FragSdu 的 fUse 域是否为 1,若为 1,则说明该 FragSdu 中是有效数据,提取出该 FragSdu,发送至下一模块。令 txQPutList 指向该 FragSdu,便于在该 FragSdu 传输完成后释放其所指向的内存块,令 txQFirstList 指向下一个 FragSdu。

3)收到传输完毕的确认消息后,释放 txQPutList 指向的 FragSdu 对应的内存块,将该 FragSdu 标为"未用"(令 fUse 域为 0)。

4 MAC 层闭合测试

目前,本方案只涉及到数据收发的软件实现,尚未和物理层及 LLC 层联合调试。我们采用 MAC 层自身闭合测试的方法来验证所做的工作,模拟了 MAC 层与物理层的接口 PHY_SAP,以及 MAC 层与 LLC 层的接口 MAC_SAP。在 MAC_SAP 的模块任务中,循环产生随机数发送到 MAC 层,并且接收来自 MAC 层的数据。在 PHY_SAP 的模块任务中,收发物理层数据传输请求及确认信息,将数据的收发联系起来。测试时,启用了 RTS/CTS 机制,产生的随机数经过 MAC 层发送端的成帧、分段、排队、计算 CRC、发送,再经过接收端的相反操作后,能成功复原,并且观察其中的每一步结果都是正确的。

结束语 系统中基本数据类型的定义和函数的实现都经过反复的测试,以适合于整个系统。在队列缓冲区的实现中,经验证最好采用 uCOS-Ⅱ系统自带的动态内存分配函数,虽然这样整个代码的可移植性有所降低,但是保证了系统在动态分配内存时,执行时间的可确定性,提高了系统的稳定性。在对连续信号的处理及整个代码结构的设计上尽量避免延时,减小系统开销,提高执行效率。

在今后的研发中,我们将系统中对速度要求比较严格的模块采用硬件来实现,即整个系统采用嵌入式芯片加逻辑电路[6]来实现,这样会使系统运行的效率更高。

参 考 文 献

1 IEEE Standards Board. 802 part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE Standard 802.11,1999
2 Labrosse J J. 邵贝贝,等译. 嵌入式实时操作系统 uC/OS-II. 北京:北京航空航天大学出版社,2003
3 宋贺强. 通信软件设计基础[M]. 北京:北京邮电大学出版社,2001
4 徐会明. 基于 IEEE802.11 协议的无线局域网 MAC 层研究与实现:[学位论文]. 武汉:华中师范大,2001
5 任哲. 嵌入式实时操作系统 uC/OS-II 原理及应用. 北京:北京航空航天大学出版社,2005
6 沈力为. IEEE802.11 无线局域网 MAC 层与物理层基带处理芯片的研究:[学位论文]. 上海:复旦大学,2004