

基于领域模型的 Web 数据抽取与集成

李 贵, 张 淼, 李征宇, 孙 平

(沈阳建筑大学 信息与控制工程学院, 辽宁 沈阳 110168)

摘 要: Web 数据抽取与集成的目的是提供面向领域的增值服务, 结合领域数据特征, 提出 Web 数据模式与领域数据模型. 给出了基于 Web 数据模式的数据定位、抽取与基于领域数据模型的集成算法, 并结合行业领域的需求, 验证了模型和算法的有效性.

关键词: Web 数据模型; Web 数据模式; 领域数据模型; 数据抽取与集成

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2012)09-0152-05

Extraction and Integration of Web Data Based on Domain Model

LI Gui, ZHANG Miao, LI Zheng-yu, SUN Ping

(Faculty of Information & Control Engineering, Shenyang Jianzhu University, Shenyang 110168, China)

Abstract: The purpose of web data extraction and integration is to provide the value-added services. Analyzing the features of domain data, the paper proposes web data schema and domain data model. It also puts forward the data positioning and extracting on the base of web data schema and an integration algorithm according to do the main data model. The application results are given to show effectiveness of the proposed algorithm and model.

Key words: web data model; web data schema; domain data model; data extraction and integration

1 引言

通过 Web 数据抽取与集成, 人们能够获取和整合来自成千上万 Web 数据源的数据, 以提供面向领域的增值服务. 随着越来越多的公司和组织在 Web 上发布信息, 从网页中抽取和集成这种数据的能力正变得越来越重要. Web 数据抽取任务的研究始于上世纪 90 年代末, 目前针对 Web 数据抽取的研究主要采取如下三种方法^[4,6]: 手工方法、包装器归纳和自动抽取. 手工方法和包装器归纳的缺点是无法处理大量站点和网页情形. 自动抽取可能抽取大量无用数据, 需要复杂数据模式和数值的匹配, 效率不高.

而 Web 表数据信息集成的研究非常少, 大部分的研究都是针对 Web 收索界面的^[4,7].

Web 数据抽取与集成是面向领域需求的, 研究问题包括: 确定领域需求模型^[8], 依据该模型完成

Web 数据抽取和集成.

2 Web 数据模型与模式

Web 上结构化数据通常是由后台数据库中的数据记录按着一定的模板 (比如: 列表页和详情页) 展现在页面上. 结构上, 这类结构化数据通常由数据项、记录元组和记录集构成. 针对 Web 结构化数据, 本文提出一种 Web 数据模型, 作为 Web 结构化数据分析与建模的基础^[4,9].

定义 1 Web 数据模型: 为一个三元组 (Type_set, T_tree_set, T_enc_set):

Type_set 为类型集, 包括: 基类型、元组类型、集合类型、平坦元组类型、平坦集合类型、平坦关系和嵌套关系;

• 基类型 Basic_types = $\{B_1, B_2, \dots, B_k\}$, 其中, B_i 表示原子类型, 其值域 $\text{dom}(B_i)$ 是一个常量的集合.

收稿日期: 2012-02-12; **修回日期:** 2012-03-15

基金项目: 辽宁省自然科学基金项目 (20071004)

• 元组类型 $\text{Tuple_type} = [T_1, T_2, \dots, T_n]$, 其中, $T_i (1 \leq i \leq n)$ 是基类型或集合类型, $\text{dom}([T_1, T_2, \dots, T_n]) = \{[v_1, v_2, \dots, v_n] \mid v_i \in \text{dom}(T_i)\}$;

• 集合类型 $\text{Set_type} = \{T\}$, 其中, T 是一个元组类型, 其值域 $\text{dom}(\{T\})$ 是 $\text{dom}(T)$ 的幂集。

• 平坦元组类型: 对于一个元组 $[T_1, T_2, \dots, T_n]$, 其中 $T_i (1 \leq i \leq n)$ 是基类型, 那么该元组就是平坦元组类型。

• 平坦集合类型: 对于一个元组集合 $\{T\}$, T 为平坦元组类型, 那么该集合就是平坦集合类型。

• 平坦关系: 对于一个集合, 如果该集合的所有元素都是基类型, 那么该集合所表示的关系是平坦关系。

• 嵌套关系: 对于一个集合, 如果该集合的元素中有集合类型的元素(不都是基类型), 那么该集合所表示的关系是嵌套关系。

T_tree_set 为类型树, 包括: 基类型树、元组类型树和集合类型树;

• 基类型树: 一个基类型 B_i 是一颗叶子树或者一个叶节点。

• 元组类型树: 一个元组类型 $[T_1, T_2, \dots, T_n]$ 是一颗以元组节点为根的含 n 颗子树的树, 每个 T_i 对应一颗子树。

• 集合类型树: 一个集合类型 $\{T\}$ 是一颗以集合节点为根的含一颗子树的树。

T_enc_set 为类型编码规则集, 包括: 基类型编码规则、元组类型编码规则和集合类型编码规则。

• 基类型编码规则: 对于一个用 T 标注的基类型的叶子节点, 其实例 t 将被编码成:

$T_enc(T; t) = \text{STAR_TAG } t \text{ END_TAG}$, STAR_TAG 和 END_TAG 分别表示开始标志和结束标志。

• 元组类型编码规则: 对于一个用 T 标注的有 N 个属性元组节点 $[T_1, T_2, \dots, T_n]$, 这个元组类型的实例将被编码成:

$T_enc(T; [t_1, t_2, \dots, t_n]) = \text{STAR_TAG } T_enc(t_1), T_enc(t_2), \dots, T_enc(t_n) \text{ END_TAG}$, STAR_TAG 和 END_TAG 分别表示开始标志和结束标志。

• 集合类型编码规则: 对于一个用 T 标注的集合类型的节点, 这个非空的集合实例 $\{s_1, s_2, \dots, s_n\}$ 将被编码成:

$T_enc(T; \{s_1, s_2, \dots, s_n\}) = \text{STAR_TAG } T_enc(s_1), T_enc(s_2), \dots, T_enc(s_n) \text{ END_TAG}$,

STAR_TAG 和 END_TAG 分别表示开始标志和结束标志。集合元素被一个排序函数排列起来。

列表: 按照一个排序函数 < 排好序的集合实例, 一个空集合实例被编码成 STAR_TAG END_TAG 。

Web 结构化数据被建模成平坦关系和嵌套关系, 也就是可以包含集合和元组的有类型的对象。平坦关系和嵌套关系通常可以用 DOM 树来表示, 关系实例的 HTML(或 XML) 标记编码将 DOM 树的每个节点与一个基于编码规则的标注函数相关联。

后台数据库中的数据都是以半结构化的形式存在于 Web 表中, 主要包括列表类型的和详情类型的, 可以以 Web 表作为基本的数据抽取对象, 为此在 Web 数据模型的基础上定义 Web 表数据模式。

定义 2 Web 表数据模式: 为一个五元组 $(W_Table_T, W_Table_N, W_Table_DN, W_Table_R_set, W_Table_enc)$ 。

W_Table_T : Web 表类型(详情页或列表页); W_Table_N : Web 表名, 是一个基类型; W_Table_DN : Web 数据项名(可选, 对于详情页没有), 是一个元组类型 $\{B_1, \dots, B_n\}$; $W_Table_R_set$: Web 表数据记录集合, 是一个元组集合类型, 每条记录是一个元组类型; W_Table_enc 为 Web 表数据记录的 HTML 或 XML 编码规则, 是 Web 数据模型中类型编码规则集 T_enc_set 的实例。

3 领域数据模型

领域数据有以下的特点:

(1) 有限的词汇。在一个领域中, 一般只有有限的属性。(2) Web 中大量相似的 Web 表格。行业网站提供相似的服务或者销售相同的产品, 提供了大量相似的表格。(3) 附加的结构。表格上的属性往往是有意义的组织在一起的, 另外属性之间存在层次化的结构^[11]。

领域数据集成需要寻找各个表格属性之间的映射, 而寻找属性之间的映射中最基本的是寻找同义词的问题。一旦对同一个领域的一组 Web 表格完成匹配, 就可以自动的构建一个领域数据模型。领域中数据的组织和表达基于领域属性、实体和语义约束, 对这些领域的属性、实体、约束以及与领域中不同站点或页面的 WEB 表的映射关系构建领域数据模型。

定义 3 领域数据模型: 为一个三元组 $(D_Attr_set, D_Tab_set, D_mapping_set)$ 。

其中:

- D_Attr_set 为领域属性集:表示为一个三元组 $(A_i_N, A_i_Type, A_i_constr)$ 的集合 $\{(A_1_N, A_1_Type, A_1_constr) \cdots (A_n_N, A_n_Type, A_n_constr)\}$, A_i_N 为属性名, A_i_Type 为属性类型, A_i_constr 为属性约束;

- D_Tab_set 为领域实体集:表示为 $\{Entity_T_1, \cdots Entity_T_m\}$, $Entity_T_i = \{(A_{i1}_N, A_{i1}_Type, A_{i1}_constr) \cdots (A_{ij}_N, A_{ij}_Type, A_{ij}_constr)\}$, 其中 $1 \leq j \leq n$, 且任意两个实体 $Entity_T_i$ 与 $Entity_T_j$ 之间除外来关键字外不允许有重复的属性.

- $D_mapping_set$ 为领域映射集:表示为 $\{(Entity_T_i, \{W_Table_N_i\}) (A_N_i, \{W_Table_DN_{i1}, \cdots W_Table_DN_{ir}\})\}$, $1 \leq i \leq n$, $1 \leq r \leq n$, 表示领域实体名与 Web 表名和实体属性与 Web 数据项的映射关系.

4 Web 数据抽取与集成

基于上面提出的 Web 数据模式和领域数据模式, Web 数据抽取与集成的过程分为: Web 表数据区域定位、表结构数据抽取和领域数据集成三个阶段.

4.1 数据区域定位

依据 Web 表名的定位方式:

输入: URL, W_Table_N

输出: table_body

处理算法:

1. get_HTML(URL);
2. local_head=NULL;
3. while $i \leq n$, and local_head=NULL do{
4. $i=i+1$;
5. local_head=get_loca($W_Table_N_i$);}
6. table_body=get_body(local_head).

给定网址 URL, 第 1 行获取 html 源码并将其解析处理成 $\langle table \rangle$ 标签树. 第 3 至 5 行搜索源码, 来获得表头位置. 第 6 行获取 local_head 为表头的表体;

4.2 表结构数据抽取

(1) 一般列表的数据抽取

输入: 一个 table_body 的字符串编码 S, S 为编码了一个集合类型的实例, 也就是说, S 含有一个由 $mi (mi \geq 2)$ 个互不重叠的子串组成的集合 W_i , 它们编码了 mi 个元组类型的实例.

输出: $W_Table_DN_set, W_Table_R_set$

输入字符串 S 是一个 WEB 表, 它包含数据表格. 在这个 WEB 表格中, S 由元组类型 t 的 $W_Table_DN_set$ 与 $W_Table_R_1, W_Table_R_2, W_Table_R_3$ 按照某种标记编码函数 $T_enc(W_Table_DN_set), T_enc(W_Table_R_1), T_enc(W_Table_R_2), T_enc(W_Table_R_3)$ 构成.

算法:

1. data_table=get_data_table(table_body);
2. $W_Table_DN_set=get_W_Table_DN(data_table)$
3. while length(data_table) >0 do
4. $W_Table_R_set.add(get_W_Table_R(data_table))$

算法第 1 行获取数据表格. 第 2 行获取数据项名. 第 3 至 4 行获取数据项记录集.

(2) 详情表的数据抽取

输入: 一个 table_body 的字符串编码 S, S 为编码了一个元组类型的实例.

输出: $W_Table_DN_set, W_Table_R$

详情表的表名与数据项名, 数据项记录集在通常都在同一个数据表格, 可以直接进行抽取操作. 详情表是一个元组类型 t 的一个实例按照某个标记编码函数 $T_enc(W_Table_R_set)$ 组成的.

算法:

1. while length(table_body) >0 do
2. $W_Table_R_set=get_W_Table_R_set(table_body)$;
3. $W_Table_DN_set=get_W_Table_DN(W_Table_R_set)$;
4. $W_Table_R=get_W_Table_R(W_Table_R_set)$.

算法第 2 行获取数据项记录集. 第 3 行在数据项记录集中获取数据项名; 第 4 行在数据项记录集中获取数据项值;

4.3 领域数据集成

领域数据集成主要包括领域数据整合和领域模型完善两方面的问题.

领域数据整合是将已经抽取的数据项值 $W_Table_R_set$, 依据 $D_mapping_set$ 经过类型转换和一致性检查等工作之后加入到依据领域模型定义的领域数据库表里的工作.

领域模型完善是针对领域 Web 站点或页面的动态变化(包括原有 Web 表模式的改变和新 Web

表的增加),而对领域模型进行更新和增加新的领域属性、实体和约束的工作。

输入: W_Table_R_set, W_Table_DN_set, D_mapping_set

算法:

1. Connect to database;
2. While(length(W_Table_R_set)>0)
3. For i=1 to length(W_Table_R_set) do
4. add_W_Table_R_set(Entity_Ti, {A_Ni1, A_Nij})
5. For i=1 to length(W_Table_R_set) do{
6. if W_Table_N then
7. update_D_mapping_set (D_mapping_set, W_Table_N, W_Table_DN_set);
8. else
9. add_D_mapping_set (D_mapping_set, W_Table_N, W_Table_DN_set);}

算法第 1 行连接数据库. 第 3 至 4 行 W_Table_R_set 中的数据导入到数据库中. 第 5 至 9 行在导入完成之后,如果 W_Table_R_set 中还有未导入的数据,那么是否存在该表名. 如果存在,那么更新领域数据模型. 如果不存在,那么添加这个新的领域数据模型. 循环这个过程直到 W_Table_R_set 中所有的数据都被导入数据库中。

5 实验评估

为了验证基于领域模式的数据抽取的有效性,本文对全国不同的城市的房地产网站进行了抓取实验,本实验运行的环境是 3 台机器在 10MB 和 25MB 网络带宽下抓取的表现。

为了检测本文方法的效率,对沈阳房产网中的许可信息(网址: www. syfc. com. cn/work/ysxk/query_xukezheng. jsp? cur_page=1),楼盘(网址: www. syfc. com. cn/work/xjlp/new_building. jsp),楼栋(网址: www. syfc. com. cn/work/xjlp/build_list. jsp? xmmcid=18942&xmmc=居住、商业),户型(网址: www. syfc. com. cn/work/xjlp/presell_inf-o. jsp? id=4071742&xszt=可售)进行了抓取实验. 表 1 是给出了抓取的结果,其中“许可,楼盘,楼栋”是在 10MB 网速下的实验,“户型”则是在 25MB 下,“URL”为输入的网址实例,“Num”表示抓取到的记录条数,“Time(s)”表示抓取所耗费的时间,“Avg_time(s)”表示平均耗费时间. 从表中可以看出,网速的提高会很明显的改善程序的效率,同

时数据量的大小也会对实验有较大影响。

表 1 数据抽取的性能

	许可信息	楼盘	楼栋	户型
Num	3891	3312	20453	690842
Time	172	156	1687	21635
Avg_time	0.04	0.04	0.08	0.03

为了检测本文的算法的精准度和完整度,对不同的城市进行了抓取实验. 本文采用以下几个测量性能的公式:准确率(P),召回率(R)和 F-score(F), F-score 是准确率和召回率的调和平均值,从它可以看出算法的整体性能表现。

表 2 中给出了实验的结果. 其中“Number of Record”指抓取的记录数,“Records Proposed”指正确的记录数,“Correct”指抓取正确的记录数。

表 2 不同城市的抽取测试

城市	鞍山	长沙	大连	东莞	佛山
Number of Record	173313	655549	1110713	497854	387518
Records Proposed	122450	639754	933468	372024	309233
Correct	112654	530995	877459	338541	306140
Precision	0.92	0.83	0.94	0.91	0.99
Recall	0.65	0.81	0.79	0.68	0.79
F-score	0.76	0.81	0.85	0.77	0.87

从表 2 中可以看出,本文的方法在领域网站中的抽取上无论是准确率还是召回率都有很好的表现. 同时,对抓取不正确的结果分析发现有些 Web 表格的表名是以图片形式存在的,本文的方法对这种情况还缺少考虑,这将是今后的工作。

6 结束语

本文通过研究现有的 Web 数据模型,结合领域中的 Web 数据表格分析的结果,从领域实体的角度,在研究分析现有的 Web 数据抽取和集成方法的基础上,提出了基于 Web 表数据模式和领域数据模型的数据抽取和集成算法. 实验结果表明,本文提出的数据抽取和集成方法提高了领域中数据抽取与集成的效率,能够满足房地产等领域的实际应用需求. 同时如何对领域数据模型中实体和属性进行有效的扩充,以及如何有效地实现面向领域的大规模 Web 数据抽取与集成将是进一步研究的方向。

参考文献:

- [1] Cafarella, M. J.; Halevy, A.; Wang, D. Z.; Wu, E.; and Zhang, Y. 2008. WebTables: Exploring the Power of Tables on the Web [C] // Proceedings of VLDB-08. Auckland, New Zealand: ACM, 2008: 538-549.
- [2] Crestan E, Pantel P. Web-scale Knowledge extraction from semi-structured tables [C] // Proceedings of WWW-2010. Raleigh, USA: ACM, 2010: 1081-1082.
- [3] 廖涛, 刘宗田, 孙荣. 表格定位技术的研究与实现[J]. 计算机科学, 2009, 36(9): 227-230.
- [4] Liu Bing, Web Data Mining. 俞勇, 薛贵荣, 韩定一, 译 [M]. 北京: 清华大学出版社, 2009.
- [5] Chen H, Tsai S, Tsai J. Mining tables from large-scale HTML texts [C] // Proceedings of COLING-00. Saarbrücken, Germany: A CL, 2000: 166-172.
- [6] Robert G, Wilks Y. Information extraction: Beyond document retrieval [J]. Journal of Documentation, 1998, 54(1): 70-105.
- [7] Gatterbauer W, Bohunsky P, Herzog M, et al. Towards domain-independent information extraction from web tables [C] // Proceedings of WWW-07. Banff, Canada: ACM, 2007: 71-80.
- [8] 鲜学丰, 方魏, 赵朋朋, 等. 一种 Deep Web 数据源质量评估模型[J]. 微电子学与计算机, 2008, 25(10): 47-50.

作者简介:

李 贵 男, (1964—), 博士, 教授. 研究方向为现代软件工程理论与方法、Web 信息挖掘、集成与应用等职能信息处理与集成技术、Web 服务计算与 Web 智能算法.

(上接第 151 页)

作者简介:

刘 翥 男, (1985—), 博士生. 研究方向为分布式计算、计算智能、软件工程.



陈 明 男, (1949—), 教授, 博士生导师. 研究方向为分布式计算、计算智能、软件工程.