基于最大频繁等价类的 Web 信息自动抽取*)

陈华昌[1,2]　薛永生[1]　任仲晟[1]　张东站[1]
(厦门大学计算机科学系　厦门 361005)[1]　(福建生态工程职业技术学校　福州 350008)[2]

摘　要　在定义模板的基础上,提出了页面创建模型。该模型描述了如何使用模板将来自于后台数据库的值编码生成页面。基于这个模型,设计了一个基于最大频繁等价类的抽取算法 EBMFEC,通过分析给定的数据导向型页面的终端符号的出现情况,找出最大频繁等价类,并推导出用于生成页面的未知模板。然后使用推导出的模板,从输入页面中提取出相关信息。在大量实际 HTML 页面上的实验证明,EBMFEC 在大部分情况下都可以从给定页面中推导出模板,并正确抽取出数据信息。

关键词　等价类,信息抽取,模式,模板

Automatic Web Information Extraction Based on Maximal and Frenquent Equivalence Classes

CHEN Hua-Chang[1,2]　XUE Yong-Sheng[1]　REN Zhong-Sheng[1]　ZGANG Dong-Zhan[1]
(Department of Computer Science, Xiamen University, Xiamen 361005)[1]
(Fujian Vocational Technical School of Ecological Engineering, Fuzhou 350008)[2]

Abstract　A novel approach based on MFEC(Maximal and Frenquent Equivalence Classes) is proposed to solve the problem of automatically extracting data from data-intensive Web pages. A template is defined and a model of page creation is proposed to describe how values are encoded into pages using the defined template. We present an algorithm, EBMFEC that takes, as input, a set of template-generated pages, analyzes the page tokens of given pages to discover MFEC, deduces the unknown template used to generate the pages and extracts, as output, the values encoded in the pages. Experiments on a large number of HTML pages indicate that our algorithm correctly extracts data in most cases and the results are also provided.

Keywords　Equivalence classes, Information extraction, Schema, Template

1　引言

Web 是当今人们获取信息的主要渠道之一。许多网站都包含大量具有结构化数据的页面,这些页面一般都是通过后台关系数据库结合网页模板动态生成的。这类页面称为数据导向型页面[1,2]。

从 Web 页面中抽取结构化数据是非常有用的,因为抽取得到的结构化数据可以进行各种复杂的查询。当集成的数据出现在不同站点上时,结构化数据抽取问题也是信息集成系统的一个非常重要的子问题[3,4]。因此,近年来,在数据库和人工智能领域都有不少关于从 Web 页面抽取数据问题的研究[5,6]。

已有的手工以及半自动化的 Web 信息抽取的方法存在以下不足之处[7]:首先,大量的手工操作对使用者提出了很高的要求。使用者必须很清楚所选取的例子对于最终生成的抽取规则的影响,同时,为了提高信息抽取的准确率,必须很恰当地选择全面而且正确的例子。其次,用于 Web 信息抽取的包装器的维护问题。因为信息的抽取与 HTML 文档的格式密切相关。如果页面的设计者更改了原有文档的格式,那么包装器就可能因为无法抽取信息而失效。

本文研究了在没有任何手工输入(例如:手动生成规则和训练集合)情况下对给定页面集实现自动信息抽取的问题,整个过程不需要用户的任何参与。

文章其余部分组织如下:第 2 节提出了一个页面结构模型并描述了数据抽取问题;第 3 节定义了最大频繁等价类;第 4 节具体描述算法 EBMFEC;第 5 节通过实验对算法进行验证;最后是总结。

2　页面结构模型及抽取描述

本节中提出了一个页面结构模型,描述数据导向型页面是如何使用模板对数据进行编码生成的。最后,形式化描述了本文要解决的数据抽取问题。

2.1　页面结构模型

图 1 展示了页面结构模型的工作原理:使用一个模板 T 可以将一个来自后台数据库的值 v 编码为一个页面,并把产生的页面标记为 $\omega(T, v)$。

来自后台数据库的值就是本文算法要抽取的结构化数据。它是类型对应的数据值的集合。用 token 表示一个终端符号串,即一个单词或者是 HTML 标签。用 null 表示特殊的常量——空值,如果某个类型的取值范围包含 null,则称它

*)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)。陈华昌　硕士研究生,讲师,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;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;张东站　博士,讲师,主要研究方向为数据库理论与应用、数据仓库、数据挖掘等。

是可空的。类型可以递归定义如下[8,9]:

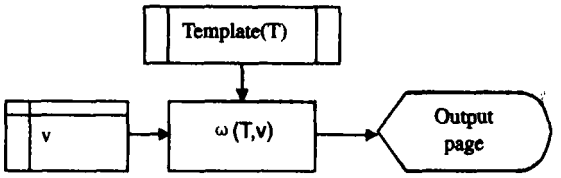

图 1　页面结构模型

(1) 基本类型: 记作 U, 是一个 token 串。

(2) 如果 $T_1, \cdots, T_n$ 是类型, 并且其中最少有一个是不可空的, 则有序列表 $[T_1, \cdots, T_n]$ 是一个元组类型, 其取值范围 $dom([T_1, \cdots, T_n]) = \{[a_1, \cdots, a_n] \mid a_i \in dom(T_i)\}$。

(3) 如果 T 是一个类型, 则 $\langle T \rangle$ 是一个列表类型, $dom(\langle T \rangle)$ 是 $dom(T)$ 中各元素的有穷列表。

(4) 如果 T 是一个类型, 则 $(T)?$ 是一个可选类型, $dom((T)?) = dom(T) \cup \{null\}$。

(5) 如果 T_1, T_2 是一个类型, 则 $(T_1 \mid T_2)$ 是一个或类型, $dom((T_1 \mid T_2)) = dom(T_1) \cup dom(T_2)$。

在下面的描述中, 术语"类型构造器"表示用于构造某种类型的函数,"值"表示一个类型的实例。若干个类型的组合称为模式。

例 2.1　考虑图 2 中的例子, 不难得到页面中包含的结构化数据如图 3 所示, 页面中显示了一个作者的信息, 包含名字、email、图书列表。其中, 图书列表包含书名和出版信息、描述, 出版信息又包含版本细节和出版时间。因此, 被编码到页面中的数据的模式为: $S1 = \{U, (U)?, \langle [U, \langle [U, U]_{\Psi 4} 13 \rangle_{\Psi 3} 12 U]_{\Psi 2} 12 \rangle_{\Psi 1} 11]_{\Psi} 11$。模式 S_1 有三个元组构造器 Ψ_{t1}、Ψ_{t2} 和 Ψ_{t3}, 另外有两个列表构造器 Ψ_{l1} 和 Ψ_{l2}。值:

$v_1 = \langle n, m, \langle [b_1, \langle [e_1, t_1], [e_2, t_2] \rangle, c_1], [b_2, \langle [e_3, t_3] \rangle, c_2] \rangle \rangle$ 是模式 S_1 的一个实例, 其中 n 为作者名字, m 为 email 等。模式和值可以等价地看成树形结构。图 4 所示为 S_1 和值 v_1 的树形结构表示。模式树的一棵子树也是一个模式, 被称为原模式的子模式。类似可以定义一个值的子值。

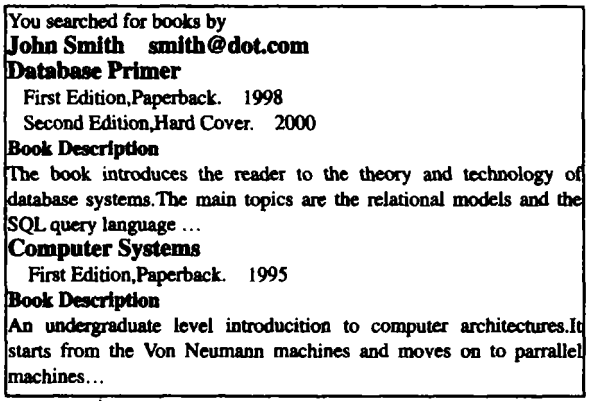
You searched for books by
John Smith　smith@dot.com
Database Primer
First Edition,Paperback.　1998
Second Edition,Hard Cover.　2000
Book Description
The book introduces the reader to the theory and technology of database systems.The main topics are the relational models and the SQL query language …
Computer Systems
First Edition,Paperback.　1995
Book Description
An undergraduate level introducition to computer architectures.It starts from the Von Neumann machines and moves on to parrallel machines…

图 2　例子页面

A	B	C	D	E	F
John Smith	smith@ dot.com	Database Primer	First Edition, Paperback	1998	The book introduces the reader to The theory and technology….
			Second Edition. Hard Cover	2000	[TRUNCATED]
		Computer Systems	First Editon, PaperBack	1995	An undergraduate level introducition to…. TRUNCATED]

图 3　抽取得到的数据

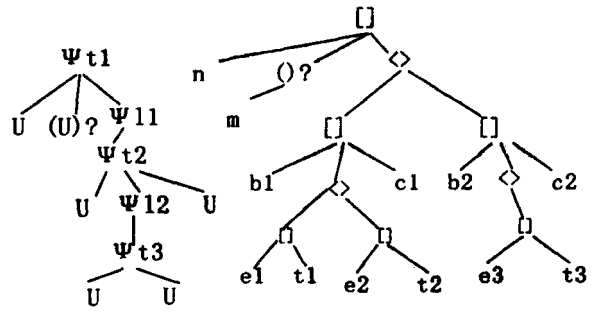

图 4　模式和实例图

定义 2.1 (模板)　模式 S 中类型构造器 Ψ 到有序字符串集合 $\Gamma(\Psi)$ 之间映射关系称为模板。

给定一个模板 $T(S)$, S 的一个实例 v 的编码 $\omega(T, v)$ 可通过对 v 的子值的编码递归定义如下:

(1) 如果 v 是基本类型 U, 则 $\omega(T, v)$ 是 v 自身。

(2) 如果 v 是一个元组 $[v_1, \cdots, v_n]\ \Psi_t$, 则 $\omega(T, v)$ 是字符串 $C_{\Psi 1}\omega(T, v_1) C_{\Psi 2}\omega(T, v_2) \cdots C_{\Psi h}\omega(T, v_n) C_{\Psi(n+1)}$。其中, v 是以 S 的类型构造器 Ψ_t 为根的子模式的一个实例, $\Gamma(\Psi_t) = \langle C_{\Psi 1} C_{\Psi 2} \cdots C_{\Psi k} C_{\Psi(n+1)} \rangle$。

(3) 如果 v 是一个列表 $\langle u_1, \cdots, u_m \rangle\ \Psi_l$, 则 $\omega(T, v)$ 是一个字符串 $\omega(T, u_1) M \omega(T, u_2) M \cdots M \omega(T, u_m)$。其中, v 是一个以 S 中的类型构造器 Ψ_l 为根的子模式的实例, $\Gamma(\Psi_l) = M$。

以上定义中, C_{Ψ} 和 C_{Ψ} 与对应的 Ψ 相关联。当一个字符串与一个类型构造器相关联, 则字符串中每个 token 都与该类型构造器相关联。

例 2.2　例 2.1 中模式 S_1 的模板 $T_1(S_1)$ 可以表示为以下几个映射关系, $\Gamma(\Psi_{t1}) = \{A, B, C, D\}$, $\Gamma(\Psi_{l1}) = M$, $\Gamma(\Psi_{t2}) = \{E, F, G, H\}$, $\Gamma(\Psi_{l2}) = N$, $\Gamma(\Psi_{t3}) = \{J, K, L\}$, 其中 A-N 都是一个字符串。通过模板 $T_1(S_1)$ 可以把值 v_1 编码为一个页面, 编码结果 $\omega(T1, v_1)$ 为 AnBmCEb1FJe1Kt1LNJe2Kt2LGc1HMEb2FJe3Kt3LGc2HD。

2.2　抽取及模板描述

在上面定义的页面结构模型基础上, 自动抽取数据的问题可以描述如下:

抽取: 给定 n 个用相同模板 T 分别由值 $\{v_1, v_2, \cdots, v_n\}$ 生成的 HTML 页面 $\{p_1, p_2, \cdots, p_n\}$, $p_i = \omega(T, v_i)$, 仅从这些页面推导出模板 T 和值 $\{v_1, v_2, \cdots, v_n\}$。

假设存在页面集合 $p_v = \{p_{v1}, p_{v2}, p_{v3}, p_{v4}\}$, 并且 p_v 中每个页面包含了书的作者名字、email(可选) 和图书信息, 每本图书信息包含书名、出版信息和描述(如图 5 所示) 。这些页面分别由图 5 的值和图 6 中模板创建。这些值的模式是 S_v

$= [B, (B)?, \langle [B, B, B]_{\Psi 2} \rangle_{\Psi 1}]_{\Psi 1}$。则对于输入页面集合 p_v 的抽取的正确结果是模板 T_v 和值 $\{Xv_1, Xv_2, Xv_3, Xv_4\}$。

Xv_1	John Smith	smith@dot.com	< [Computer Systems, …] [Database Primer, …]]>
Xv_2	Paul Jones	null	< [XML at Work, …] >
Xv_3	Tom Break	break@ibm.com	< [C# Programming, …] >
Xv_4	Brown Dean	null	< [Data Mining, …] [C++ Primer, …]]>

图 5　正确的值

```
[
<HTML>1
<BODY>2<B>3*</B>4
<A>5 (*) ? </A>6
<UL>7
<[
<LI>8<I>9<B>10Title:11</B>12*</I>13
< BR/ >14<B>15Edition16</B>17*
<BR/>18<B>19Book20
Description21</B>22*
    </LI>23
]>
</UL>24
<BODY>25</HTML>26
```

图 6　正确的模式

3　最大频繁等价类

在以下的论述中, 假定存在一个页面集合 $P = \{p_1, p_2, \cdots, p_n\}$, 其中 $p_i = \omega(T(S), v_i)(1 \leqslant i \leqslant n)$。模式 S 包含类型构造器 $\{\Psi_1, \cdots, \Psi_k\}$。算法以 P 作为输入, 但是并不了解 T, S 和 $\{v_1, \cdots, v_n\}$。

一个 token 在模板、值或页面中的一次出现分别称为一个 T-token、V-token 或 P-token。从模型可知, 每一个 P-token 都是由 T-token 或者 V-token 创建的。当两个页面符号由同一个模板符号创建, 则说它们具有相同的角色。因此, p 中两个 P-token 的角色相同当且仅当它们具有相同的 T-token。

定义 3.1 (出现路径)　某个 p-token 的出现路径是指它在页面的语法分析树(如图 7 所示) 中从根结点到该 token 所经过的路径。

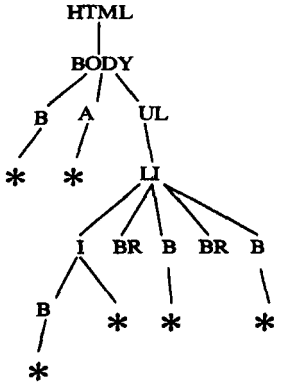

图 7　HTML 语法分析树

定义 3.2 (出现向量)　某个 token 的出现向量是指它在 P 的每个 p_i 中出现的次数组成的向量 $\langle f_1, \cdots, f_n \rangle$。向量中不为 0 的分量的个数称为 token 的支持度。

定义 3.3 (等价类)　具有相同出现向量的最大 tokens 集合, 称为等价类。

如果一个等价类起源于某个类型构造器, 则说它是有效的; 否则, 它是无效的。在 P 中, 所有等价类构成了所有 tokens 集合的一个划分。

定义 3.4 (有序等价类)　如果某个等价类 ε 是有序的, 则该等价类的 tokens 可以排序为 $\langle t1, \cdots, t_m \rangle$ 使得对每一个页面 $p_i(1 \leqslant i \leqslant n)$ 都满足:

$\forall h \in [1, q], \exists j, k \in [1, m], j < k, q$ 为 ε 在 P_i 出现次数, $Cur_h(t_j) < Cur_h(t_k) < Cur_{h+1}(t_j) < Cur_{h+1}(t_k)$, Cur 为出现位置。

令 $\varepsilon = \langle t_1, \cdots, t_m \rangle$ 是一个有序等价类, 则 $Cur_h(\varepsilon)$ 为区间 $[Cur_h(t_l), \cdots, Cur_h(t_m)]$, $\forall h \in [1, q]$, q 为 ε 在 P_i 出现次数。称 $Cur_h(\varepsilon)$ 为 ε 在 p_i 中的一次出现跨度, 它被分隔为 $m-1$ 个位置, 记作 $pos(1), \cdots, pos(m-1)$, $Pos_h(k)$ 包含区间 $(Cur_h(t_k), Cur_h(t_{k+1}))$ 中的全部内容。

定义 3.5 (等价类嵌套)　两个等价类 ε_i 和 ε_j 如果满足下面条件之一, 则它们是嵌套的。

(1) $\forall h \in [1, q]$, q 为 ε 在 P_i 出现次数, $Cur_h(\varepsilon_i) \cap Cur_k(\varepsilon_i) = \Phi$;

(2) $\forall h \in [1, q], \exists k \in [1, w], \exists p \in [1, m)$, q 为 ε_j 在 P_i 出现次数, ω 为 ε_i 在 P_i 出现次数 $Cur_h(\varepsilon_j) \subset pos(p), pos(p) \in Cur_k(\varepsilon_i)$。

如果等价类集合 $\{\varepsilon_1, \cdots, \varepsilon_n\}$ 中每一对等价类都是嵌套的, 则说该等价类集合是嵌套的。

定义 3.6 (大而频繁等价类)　如果一个等价类中所有 token 的支持度都大于等于用户设定的支持度阈值 MinSupport, 则称该等价类为频繁等价类; 如果一个频繁等价类中 token 的数量大于等于用户设定的数量阈值 MinSize 时, 称其为大而频繁等价类(Large and Frenquent Equivalence classes), 记为 LFEC。

如果一个 LFEC 的所有 tokens 在每个页面只出现一次, 称该 LFEC 为根 LFEC。

定义 3.7 (最大频繁等价类)　如果一个 LFEC 中包含了它的出现跨度内的最大数量的 tokens, 则它是最大频繁等价类 MFEC(Maximal and Frenquent Equivalence classes)。

通过以上的定义可以得到下面几个结论:

结论 1　与模式中同一类型构造器相关联且具有唯一角色的 tokens 出现在相同的等价类中。

证明: 不妨假设这些 tokens 不属于相同的等价类, 则由前面等价类的定义可知, 这些 tokens 具有不同的出现向量, 从而不可能与同一类型构造器相关联, 因此结论成立。

结论 2　大而频繁等价类 LFEC 总是有效的。

证明: 对大而频繁等价类分成两种情况讨论:

a) 若 LFEC 为空, 则该 LFEC 为有效的;

b) 若 LFEC 不为空, 则至少存在一个等价类, 其规模大于 MinSize, 且其中 tokens 的支持度均大于 MinSupport。由于所有的 p-tokens 均由 T-token 得到, 而 T-token 来自类型构造器, 因此该 LFEC 是有效的。

结论 3　有效等价类是有序的并且两个有效等价类必定是嵌套的。

证明: 首先证明有效等价类是有序的。

由有效等价类的定义可知,有效等价类来自某个类型构造器,从而保证其中的 tokens 的出现顺序符合有序等价类定义中规定的条件。

其次证明两个有效等价类是嵌套的。

有效等价类来自类型构造器,而类型构造器是不重叠的,因此有效等价类一定满足前面关于等价类嵌套定义中的两个条件,从而两个有效等价类是嵌套的。

由此结论得证。

结论 4　设 ε 是源于类型构造器 Ψ_i 的一个有效等价类。Token t 在 ε 的任意出现跨度之外的一次出现对应的角色,不同于在 ε 的某次出现跨度内部的一次出现的角色。另外,Token t 在 ε 某次出现的 $Pos(j)$ 内的角色,不同于在 $Pos(k)$ 内的角色。

证明:由角色的定义可知,同一角色在 HTML 树中对应着相同的路径,因此 Token t 的出现必定落在 ε 的一个跨度当中。类似可得,Token t 在 ε 某次出现的 $Pos(j)$ 内的角色,不同于在 $Pos(k)$ 内的角色,从而结论得证。

4　抽取算法

基于最大频繁等价类的定义和结论,可以得出 Web 数据抽取算法 EBMFEC(Extraction Based on Maximal and Frenrquent Equivalence Classes)。算法分析给定页面的 tokens,产生 MFEC,然后推导出页面模板并利用模板进行数据的抽取。

算法通过以下三个步骤完成自动信息抽取:

(1) 分析 HTML 语法树中 tokens 的出现情况,对出现在 P 中的 tokens 进行分类,生成 MFEC。

输入:$P=\{p_1, p_2, \cdots, p_n\}$、最小支持度 MinSupport,最小数量 MinSize。

输出:$MFEC=\{m_1, m_2, \cdots, m_n\}$,$m_i$ 是对应于某个类型构造器 Ψ_i 的最大频繁等价类。

过程:

Step1. 把页面集合 P 中的每个 HTML 页面看成一棵 HTML 分析树,从根结点出发递归地进行深度遍历,记录下每个 token 的名称、所在页面、出现路径、在文档中位置标记以及该 token 同路径时的出现次数。

Setp2. 计算大而频繁等价类 LFEC(Large and Frenquent Equivalence Classes)

①通过对 token 名称和出现路径分组求出同路径的 tokens 在每一个页面中的出现次数;

②从①得到的结果中,筛选出所有支持度大于等于用户给定最小支持度阈值 MinSupport 的 tokens;

③扫描②中得到的结果,得到频繁等价向量集合;

④通过频繁等价向量,分组得到频繁等价类;

⑤逐个处理每个频繁等价类,对 token 个数大于等于 MinSize 的频繁等价类按 token 在页面中出现位置对其 tokens 进行排序,然后将其加入到 LFEC 集合。

Step3. 利用算法 HANDINV(handle invald LFEC) 对大而频繁等价类 LFEC 进行有效性处理,得到有效的大而频繁等价类。

根据结论 3,如果一个等价类不是有序的,则它是无效的;如果两个等价类不是嵌套的,则其中最少有一个是无效的。算法 HANDINV 将在这个结论指导下对等价类的有效性进行处理。

无效等价类产生的原因主要有三种:①两个或多个类型构造器实例化时相互联系;②频繁出现的元组。例如一个子模式在不同页面中重复实例化为相同的元组;③与不同类型构造器相关联的 tokens 相互重叠。

前两种原因产生的无效等价类是有效等价类的组合,它们有明显可辨识的特征,即近似排序或近似嵌套。算法 HANDINV 将这种无效等价类拆分成多个有效等价类。由第③种和其它原因产生的无效等价类无法被划分成有效等价类,HANDINV 算法将删除这种无效等价类。把由前两种原因产生的无效等价类称为 TypeA,其它无效等价类称为 TypeB。

定义 4.1 (近似排序)　如果某个等价类的终端符号可以被划分成不重叠的有序等价类,则该等价类近似排序。

定义 4.2 (近似嵌套)　任意 $t\in\varepsilon_i$,存在某个固定位置 p,使得 t 的每一次出现要么在 ε_i 的任何出现跨度之外要么在 ε_j 的某个出现的 $Pos(p)$ 内,则等价类 ε_i 和 ε_j 近似嵌套。

HANDINV 算法描述:

输入:一个等价类集合(可能包含无序或不嵌套等价类)

输出:一个嵌套的有序等价类集合

过程:

①检查每个等价类是否有序或近似有序。删除不是有序和近似有序的等价类。如果是近似有序,则将其划分成不重叠的有序等价类。

②检查保留下来的等价类两两之间的嵌套情况。如果 ε_i 和 ε_j 既不嵌套又不近似嵌套,则其中至少有一个是 TypeA。通过贪心法找出不嵌套的等价类,将其删除。也就是说,可以不检查有效的和 TypeA 等价类。

③迭代处理所有等价类,找出每对非嵌套但近似嵌套的等价类,并处理。如果 ε_i 和 ε_j 近似嵌套,则其中至少有一个是无效的。

1)如果 ε_i 有效,ε_j 是 TypeA 等价类,则 ε_j 的每一次出现都与 ε_i 的一次出现重叠,反之不成立。此时,HANDINV 算法划分 ε_j,将 ε_i 某次出现的 $Pos(p)$ 的每个连续的 ε_j 的 tokens 集合,划分为同一个等价类。

2)如果存在 ε_i 某次出现不与 ε_j 重叠且 ε_j 的某次出现不与 ε_i 重叠,则 ε_i 和 ε_j 都不是有效的。此时,将出现在 ε_i(或 ε_j)的 $Pos(p)$ 中的 ε_j(或 ε_i)的连续 tokens 集合划分为一个等价类。

Step4. 区分每个 token 同路径多次出现的不同角色,扩充 LFEC 的尺寸,最终得到 MFEC。从根 LFEC 开始迭代处理每个 LFEC:

1)找出当前 LFEC 跨度内的所有同路径多次出现的 tokens,根据结论(4) 进行角色区分。

2)根据位置标记所有 tokens,产生带标记的 tokens;同时记录下原始 token 与带标记的 token 之间的对应关系。

3)判断经过标记的符号所在的等价类,将其加入。

把经过标记的 token 表示为 d_i,最大频繁等价类 $MFEC=\langle d_1, d_2, \cdots, d_n\rangle$。

(2) 利用 MFEC 推导出抽取数据用的模板 T 和模式 S。

用下面的模板代数描述模板的递归构造过程:(a) 如果 $T_1(S_1), T_2(S_2), \cdots, T_m(S_m)$ 是模板,并且 $C1, C2, \cdots, Cm+1$ 是字符串,$T(S)=[T_1, T_2, \cdots, T_m]_{\langle C_1, C_2, \cdots, C_{m+1}\rangle}$ 表示一个模板,其中 $S=[S_1, S_2, \cdots, S_n]_\Psi$,$T$ 由如下映射定义:$\Gamma(\Psi)=\langle C1, C2, \cdots, Cm+1\rangle$,且 $\Gamma(\Psi_k)=\Gamma_i(\Psi_k)$,所有 Ψ_k 属于 S_i $(1\leqslant i\leqslant m)$。(b) $T_i(S_i)$ 是一个模板,且 M 是一个字符串,则 $T(S)=\langle T_i(S_i)\rangle_M=$ 表示一个模板,其中 $S=\langle S_i\rangle_\Psi$,$T$ 由如下映射定义:$\Gamma(\Psi)=\langle M\rangle$ 且 $\Gamma(\Psi_k)=\Gamma_i(\Psi_k)$,所有 Ψ_k 属于 S_i

$(1 \leqslant i \leqslant m)$。(c) $T_i(S_i)$ (对于模式(S_i)?)和$(T_i(S_i) \mid T_j(S_j))$ (对于模式$(S_i \mid S_j)$)定义类似;(d) T_u 表示基本类型 U 的原子模板。

根据上面的模板代数,模板递归构造过程的算法 ConstT 可以实现如下。

```
输入: Set{ε1, ε2, …, εm}, 根 MFEC
输出: 对应于根 MFEC 的模板 T(S), S 是对应于根 MFEC 的模式, S=[S1, S2, …, Sn]ψ
过程: 令 εi=〈d1, d2, …, dn〉, ti 表示与 di 对应的token; pi(表示 εi 中非空位置, 定义 q+1 个字符串〈Ci1, Ci2, …, Ci(q+1)〉, 其中 Ci1= t1…tp1, Cij= t(p(j-1)+1)…tpj 和 C(q+1)= tpq…tn。
εi= 根 MFEC, k= 1;
While k<= n-1
{
  求 PosStr(εi, k); //通过判断 dk, dk+1是否总连续
  If PosStr(εi, k)空{k++; continue;}//进入下一次循环
  If PosStr(εi, k)内存在等价类ε
  { If(PosStr(εi, k)不存在对应的已有模式)
      ConstT(Set, εj); //求 Sk
    Sk= Tεj}
  Else
    Sk= U;
  Cij= t(p(j-1)+1)…tpj;
  K++;}
Sεi=[S1, S2, …, Sq]ψ://组合 S1, S2, …, Sq 得到模式S//并通过等价向量判断类型
Tεi=〈Tεp1, …, Tεsq〉〈Ci1, Ci2, …, Ci(q+1)〉;
```

(3) 抽取数据,得到最终的值$\{v_1, \cdots, v_n\}$。在本文中忽略通过模板抽取数据的过程,因为这部分内容相对容易实现。

5 实验

在本文所述算法基础上,我们实现了一个原型系统,本文的实验全在该系统上进行。在实验中所用的页面(见表 1 第 1 列) 全部来自于实际的网站。

为了比较的目的,用 S_m 表示手工确定的数据模式, S_e 表示通过系统抽取得到的模式。A_m 表示 S_m 中的数据项, A_e 表示 S_e 中的数据项。为方便评估,考虑 S_m 中的每个属性 A_m,并将评估结果分为三种类型: 完全正确# c, 部分正确# p, 不正确# i。# c 表示 S_e 中所有 A_e 与 S_m 中的 A_m 完全相同;# p 表示 S_e 中 A_e 与 S_m 中的 A_m 部分相同;# i 表示 S_e 中所有 A_e 与 S_m 中的 A_m 完全不相同。n 表示某个数据源用于测试的页面数, a 表示 S_m 中属性的个数。

在实验中成功实现了 Web 页面信息自动抽取。与其它抽取算法比较,它可以不用任何人工输入的信息。所得到的结果如表 1 所示。

表 1 实验结果

Data Source	n	a	# c	# p	# i
aAmazon(Book)	20	5	5	0	0
Yahoo Shopping	10	10	8	2	0
eBay(auction)	20	9	9	0	0
Bigbook	50	8	6	2	0
Webcrawler	10	10	10	0	0
Buy.com(prod)	10	12	9	2	0
LA Weekly	10	7	6	1	0
Amazon(Cars)	21	13	13	0	0
Exite	10	11	11	0	0
Openfind	5	6	4	2	0
Total	166	90	81	9	0

实验的目的主要是检验抽取算法的有效性。表 1 概括了实验的结果,它说明算法 EBMFEC 用于抽取数据是非常有效的。其中完全正确的有 56%,部分正确的是 44%。按全部属性平均统计,抽取的正确率大概为 98.9%。

结论 本文提出了一种从数据导向型页面中自动抽取数据的算法 EBMFEC。先给出了基于模板的页面模型,在这个模型之上,分析给定页面的 tokens,产生有效的最大频繁等价类 MFEC,然后推导出页面的模板并利用模板进行数据的抽取。在实际页面上的实验已经证明该方法可以达到较高的正确率。实验中发现算法使用的假设存在不成立的情况,但只是少量属性存在这种问题。在以后的工作中,将进一步对数据导向型页面的检索和查询支持以及对从 Web 中抽取得到的数据进行数据模式挖掘开展研究。

(下转第 202 页)

(上接第 141 页)

3.2 改进方法 2

【结论】: 考察目标模式中的常量字 sConstWord。如果 sConstWord 不在事实库、规则库中出现,那么,目标模式推理失败。

【证明】: 目标模式的模式推理,最终归结为目标模式(及其派生的子模式) 与事实库的匹配。目标模式中的常量字 sConstWord 不在事实库中出现,所以,目标模式匹配事实库失败。

如果目标模式匹配规则库失败,那么目标模式推理失败。

否则,不妨令目标模式匹配规则 R 的右部。因为常量字 sConstWord 不在规则库中出现,所以,常量字 sConstWord 代换规则 R 的一个变量。对于一条规则来说,规则右部的变量,肯定在规则左部出现。所以,在派生出来的子目标中,至少有一个子目标 SGi 含有常量字 sConstWord。

子目标 SGi 匹配事实库失败。即使子目标 SGi 匹配规则库成功,派生出来的子目标 SGii, 也会含有常量字 sConstWord。

总之,在目标模式的模式推理过程中,无法最终归结为目标模式(及其派生的子目标) 与事实库的匹配。所以,目标模式推理失败。

结论和下一步的工作 本文首先介绍了相关定义,然后根据现有模式推理的特点,提出了一种基于自然语言的模式推理算法,并提出了改进方法。实验结果表明,本算法可以有效地解决基于自然语言的模式推理问题。下一步的工作,将致力于减小算法的时间复杂性,进一步提高算法的效率。

参考文献

1 王树西, 白硕, 姜吉发. 模式合一的"斩首"算法及其应用. 计算机工程, 2004, 21: 22~24

2 王树西, 白硕, 等. 模式合一的"斩首"算法. 见: 中国人工智能学会第 10 届全国学术年会论文集(上), 广东, 2003. 528~532

3 白硕. 大规模内容计算. 见: 语言计算与基于内容的文本处理. 北京: 清华大学出版社, 2003. 13~15

4 王树西, 白硕. 事实库、规则库的一体化全文索引算法. 计算机科学, 2006, 33(4): 174~176

5 Post E L. A variant of a recursively unsolvable problem. Bull of the Am Math Soc, 1946, 52

6 Ehrenfeucht A, Karhumaki J, Rozenberg G. The (generalized) post correspondece problem with lists consisting of two words is decidable. Theoret Comput Sci, 1982, 21(2)

7 Matiyasevich Y, Senizergues G. Decision problems for semi-Thue systems with a few rules. In: Proceedings, 11th Anual IEEE Symposium on Logic in Computer Science, 1996

8 Halava V, Harju T, Hirvensalo M. Binary (Generalized) Post Correspondence Problem: [Technical Report No. 357]. TUCS, August 2000

9 Simmons R E. Natural Language Question-Answering Systems: 1969. Communications of the ACM, 1970, 13(1): 15~30

比较苛刻。此外,计算方法本身还存在两点缺陷,首先是局部点关联趋向,即由关联系数大的点决定整体关联度,其次是关联系数的平均值生成关联度,容易造成信息损失。在实际应用中采用的改进方法,如趋势关联度、型关联度、绝对关联度和灰色关联熵等。然而,这些方法通常只是针对其中的某一个缺陷进行调整,仍然有不满意的地方。灰色斜率熵关联度计算方法在上述基础上作出了改进。根据曲线间相似程度,灰色关联度的基本思想如下:

设 $X(t)$ 为母函数,$Y_i(t)\ (i=1,2,\cdots,m)$ 为子函数,则称

$$\xi(t)=\frac{1+0.5*\left|\frac{\Delta x(t)}{\Delta t}\right|}{1+0.5*\left|\frac{\Delta x(t)}{\Delta t}\right|+\left|\frac{\Delta x(t)}{\Delta t}-\frac{\Delta y_i(t)}{\Delta t}\right|}$$

为函数 $X(t)$ 与 $Y_i(t)$ 在 t 时刻的斜率关联系数。其中

$\frac{\Delta x(t)}{\Delta t}$ 为母函数 $X(t)$ 在 t 到 $t+\Delta t$ 的斜率;

$\frac{\Delta y_i(t)}{\Delta t}$ 为子函数 $Y_i(t)$ 在 t 到 $t+\Delta t$ 的斜率;

且 $\Delta x(t)=x(t+\Delta t)-x(t)$、$\Delta y_i(t)=y_i(t+\Delta t)-y_i(t)$,由于在事务拓扑空间中,所有的序列都为 1 时距的离散序列,故有 $\Delta t=1$。根据上述定义,有灰色斜率关联系数,设 $X_0=\{\langle x_0(k),\oplus\rangle \mid k=1,2,\cdots,n\}$ 是 Ω 中关于时间 $\oplus$ 的主属性序列,$X_i=\{\langle x_i(k),\oplus\rangle \mid k=1,2,\cdots,n\},(i=1,2,\cdots,m)$ 是非主属性序列,其中,时间 $\oplus$ 为属性集中需考察某一给定的时间区间,序列中 n 的取值由 $\oplus$ 确定。

令 $\gamma(\langle x_0(k),\oplus\rangle,\langle x_i(k),\oplus\rangle)=\frac{1+0.5*a(k)}{1+b(k)+0.5*a(k)}$

其中,$a(k)=|\langle x_0(k+1),\oplus\rangle-\langle x_0(k),\oplus\rangle|$,$b(k)=|(\langle x_0(k+1),\oplus\rangle-\langle x_0(k),\oplus\rangle)-(\langle x_i(k+1),\oplus\rangle-\langle x_i(k),\oplus\rangle)|$,且 $k=1,2,\cdots,n-1$,则称 $\gamma(\langle x_0(k),\oplus\rangle,\langle x_i(k),\oplus\rangle)$ 为 X_0 与 X_i 在 k 点的灰色斜率关联系数。

根据 X_i 相对于 X_0 在时间 $\oplus$ 的灰色斜率熵关联度,可以获得相应的灰关联规则。由该条规则,便能得知在时间 $\oplus$ 时不同非主属性对主属性的影响程度,管理人员根据这种影响程度的排序,可以根据实际需要适时地调整控制参数或相关条件,以适应商务网站的实际运作。在采用典型灰关联度计算方法时,对时间区间进行改变,可以获得具有时间属性的灰关联规则集;这些关联规则在不同时间区间的变化,从定性的角度说明了各影响因素的发展变化历史,有利于商务网站控制。

而引入灰色斜率关联熵的概念后,可以构造一个时间窗口,根据商务网站的特性,采用一个时间跨度。利用该时间窗口的滑动,便可以获得同一个因素在不同状态时熵的变化。利用这些熵,构造一条熵值曲线,由熵的涨落便能得知系统的稳定程度,以及与外界的协调性。或者构造一个系统进化的模型,可以表示为,$\Delta S=E_n(X_i)_{k+1}-E_n(X_i)_k$,显然 ΔS 的取值只有 3 种可能:$\Delta S<0$,$\Delta S>0$ 或者 $\Delta S=0$,ΔS 的含义是系统与外界进行能量、信息、物质等的交换引起的熵变。这种熵变,说明了系统状态的变化,为商务网站的分析提供了新的思考方法。

这种时间窗口的滑动引起的熵变和灰关联规则的改变,即为灰关联规则的增量更新。在灰色系统理论中,根据解的非唯一性原理,可能存在有不同的灰关联度。无论是采用典型的灰关联度计算方法,还是利用改进的灰关联度计算方法,在实际的应用中,需要结合商务网站的实际需求,综合比较各种算法,以获得满意的效果。

3 联机分析处理

商务网站联机分析处理工具,采用多维数据库,利用灰关联规则进行数据挖掘,可以发现隐藏在商务数据间的相互关系,它能发现商务数据库中所分析对象之间的关系,把分析所需的数据从数据仓库中抽取出来,物理地组织成多维数据库。联机分析处理可以对商务网站的联机数据访问和分析,通过对信息进行快速、稳定、一致和交互式的存取,对数据进行多层次、多阶段的分析处理,以获得高度归纳的分析结果。电子商务是数据挖掘的重要应用领域,从 WallMart 到 Amazon.com,都有着成功的应用,如销售、顾客、产品、时间和地区的多维分析,网站购物的推荐服务等等。在银行和金融业中,信用欺诈的建模与预测、风险评估、收益分析、客户关系优化以及股票价格、商品价格和金融危机的预测等方面,有着较好的应用。

商务网站联机分析处理是一种自上而下、不断深入的分析工具,在用户提出问题或假设之后,它负责提取出关于此问题的详细信息,并以直观的方式呈现给用户。在进行联机分析处理分析时,从用户角度和查询性能考虑,可以采用多视图模式,即除了使用数据报表格式向用户表现查询结果以外,还采用曲线图、饼图、柱状图等多种格式显示。

参 考 文 献

1 黄斐. 电子商务网站学习网站建设. 计算机应用, 2002(7)
2 黄斐. 电子商务网站数据处理. 微机发展, 2002(3)
3 刘业翔. 铝电解控制中灰关联规则挖掘算法的应用. 中国有色金属学报, 2004(3)
4 邓聚龙. 灰理论基础. 武汉: 华中科技大学出版社, 2002
5 黄斐. 基于网络环境的项目协调管理. 计算机科学, 2004(10)
6 黄斐. 基于 MS Project 的多课程管理系统. 成都: 西南交大出版社, 2004
7 黄斐. MS Project 2002 项目管理与应用. 科学出版社, 2004

(上接第 173 页)

参 考 文 献

1 Haas L M, Kossmann D, Wimmers E L, et al. Optimizing queries across diverse data sources. In: Proc of the 23th VLDB Conf. Athens, 1997. 276~285
2 Levy A, Rajaraman A, Ordille J J. Querying heterogeneous information sources using source descriptions. In: Proc. of the 22th VLDB Conf. Bombay, 1996. 251~262
3 Kushmerik N. Wrapper induction: Efficiency and expressiveness. Journal of Artificial Intelligence, 2000, 118(1-2): 15~68
4 Soderland S. Learning information extraction rules for semi-structured and free text. Journal of Machine learning, 1999, 34(1-3): 233~272
5 Embley D W, Campbell D M. A conceptual-modeling approach to extracting data from the web. In: Proc. of the 17th Intl. Conf on Conceptual Modeling. Singapore, 1998. 78~91
6 Chang C, Lui S. IEPAD: Information extraction based on pattern discovery. In: Proc of 10th WWW Conf. Hong Kong, 2001. 681~688
7 Crescenzi V, Mecca G, Merialdo P. ROADRUNNER: Towards automatic data extraction from large web sites. In: Proc of the 27th VLDB Conf. Roma, 2001. 109~118
8 Crescenzi V, Mecca G. Automatic Information Extraction from Large Websites. Journal of the ACM, 2004, 51(5): 731~779
9 Sarawagi S. Automation in InformationExtraction and Data Integration (tutorial). VLDB, 2002
10 Myllymaki J. Effective Web data extraction with standard XML technologies. In: Proc of 10th WWW Conf. Hong Kong, 2001. 689~696