ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.19, No.6, June 2008, pp.1422−1427
DOI: 10.3724/SP.J.1001.2008.01422

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 一种从 XML 数据中发现关系信息的方法*

吴扬扬[1+], 雷 庆[1], 陈锻生[1], YOKOTA Harou[2]

[1](华侨大学 计算机科学系,福建 泉州 362021)

[2](Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan)

## A Method of Discovering Relation Information from XML Data

WU Yang-Yang[1+], LEI Qing[1], CHEN Duan-Sheng[1], YOKOTA Harou[2]

[1](Department of Computer Science and Technology, Huaqiao University, Quanzhou 362021, China)

[2](Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan)

+ Corresponding author: E-mail: wuyy@hqu.edu.cn

**Wu YY, Lei Q, Chen DS, Yokota H. A method of discovering relation information from XML data. *Journal of Software*, 2008,19(6):1422−1427.** http://www.jos.org.cn/1000-9825/19/1422.htm

**Abstract**: A novel method of discovering relation information among entities buried in different nest structures of XML documents is proposed. The method is able to identify relations among different types of entities given by users, and extract relation instances and their occurrence patterns in XML documents. The solution is as follows: identify and collect XML fragments that contain all types of entity given by users at first, then calculate similarity between fragments based on semantics of their tags and their structures, and cluster fragments with a adaptively selected similarity threshold so that the fragments containing the same relation are clustered together, finally extract relation instances and patterns of their occurrences from each cluster. The experimental results show that the method can identify and extract relation information among given types of entities correctly from all kinds of XML documents with meaningful tags.

**Key words**: relation information; XML document; similarity; cluster; occurrence pattern

**摘 要**: 提出了一种发现蕴藏在不同 XML 文档嵌套结构中的关系信息及其出现模式的新方法.可根据用户兴趣,发现描述不同实体之间联系的关系信息,抽取关系实例及其在文档中的出现模式.具体解决方案是:首先识别和收集包含用户感兴趣的实体的 XML 文档片段;然后根据文档片段标签的语义和文档片段的结构计算文档片段的相似度,并采用自适应阈值方法按相似度聚类文档片段,使得包含同一种关系的文档片段聚集在同一个片段簇;最后从 XML 文档片段簇中抽取关系实例及其出现模式.实验结果表明,对于包含有意义标签的各种 XML 文档,该方法能够准确地识别和抽取出描述指定实体之间联系的各种关系信息.

**关键词**: 关系信息;XML 文档;相似度;聚类;出现模式

**中图法分类号**: TP311 **文献标识码**: A

* Supported by the Natural Science Foundation of Fujian Province of China under Grant No.A0510020 (福建省自然科学基金); the Int'l Science and Technology Cooperation Project of Fujian Province of China under Grant No.2004I014 (福建省国际科技合作项目)

Received 2006-09-22; Accepted 2007-02-05

本文探讨利用 XML 文档的标签和结构的语义信息,从 XML 文档中发现和抽取表示实体间联系的关系信息.与其他信息抽取研究不同的是:本文的目标不仅仅是抽取给定模板的数据项[1]或指定关系的元组[2,3],而是要将XML文档集中蕴藏的表示实体之间联系的各种关系发掘出来,有可能发现用户事先不知道的关系.下面对待发现和抽取的关系信息、数据源——XML 文档以及信息提取任务进行形式化的描述.

**定义 1**. 表示 $n$ 个实体型之间联系的关系模式由一个关系名 $R$ 和 $n$ 个实体类型名 $et_1,\ldots,et_n$ 组成,记为 $R(et_1,\ldots,et_n)$.$n$ 个实体型之间联系的一个实例 $r$ 由一个 $n$ 元序组$(I(et_1),\ldots,I(et_n))$表示,其中,$I(et_i)(i=1,\ldots,n)$是实体类型 $et_i$ 的一个实例.

**定义 2**. 一个 XML 文档 $d$ 可以表示为一棵带标记节点树 $d=(N,E)$,其中,$N$ 为节点集,每个节点对应文档中的一个元素或属性,节点用其元素名或属性名作标记.$E$ 是边集,$(a,b)\in E$ 当且仅当 $b$ 是 $a$ 的子元素或属性.如果一段 XML 文档 $p$ 的节点树是以 $N$ 中的某个元素 $x$ 为根的子树(由 $x$ 和 $x$ 的所有子孙节点及其边组成),则称 $p$ 为 XML 文档 $d$ 的一个片段.

这里只考虑与本文挖掘任务相关的元素、属性以及元素嵌套结构.标签名和实体类型名通常由领域专家或相关用户精心选取,通常体现了人们的共识,考虑到用词上存在差异,我们引入名称扩展向量.

**定义 3**. 设 $na$ 为标签或实体类型的名称,$Ex(na)=(na,na^1,na^2,\ldots,na^m)$称为名称 $na$ 的扩展向量,其中,$na^i(i=1,\ldots,m)$为 $na$ 的同义词、合成词或缩写形式.

**定义 4**. 设 $p$ 为一段 XML 文档片段,$T=\{t_1,\ldots,t_v\}$为 $p$ 的标签名和属性名组成的集合,$Ex(et)$为实体类型名 $et$ 的扩展向量,如果$\exists t_i\in T$,使得 $t_i$ 与 $Ex(et)$的某个元素相匹配,则称文档片段 $p$ 包含实体类型 $et$.

**定义 5**. 设 $P$ 为 XML 文档 $d$ 的所有包含实体类型 $et_1,\ldots,et_n$ 的文档片段组成的集合,$p\in P$,如果$\forall p'\in P,p'$的节点树都不是 $p$ 的节点树的子树,则称 $p$ 为文档 $d$ 的包含实体类型 $et_1,\ldots,et_n$ 的极小文档片段.

设用户感兴趣的联系所涉及的 $n$ 个实体类型为 $et_1,\ldots,et_n$,$R=\{R_1,\ldots,R_h\}$为实体类型 $et_1,\ldots,et_n$ 之间各种关系的集合,其中,$R_i=\{r_{ij}|r_{ij}=(I(et_1),\ldots,I(et_n))\}(i=1,\ldots,h)$,$D$ 为 XML 文档集合,则从 $D$ 中发现用户感兴趣的、表示实体之间联系的各种关系信息及其出现模式的任务可描述为:

从 $D$ 中找出所有包含实体类型 $et_1,\ldots,et_n$ 的极小文档片段 $P=\{p_1,\ldots,p_m\}$,将 $P$ 划分为 $P_1,\ldots,P_h$,使得每一个划分块 $P_i$ 包含一个关系 $R_i$.对每一个关系 $R_i$,建立表示该关系的关系模式 $R_i(et_1,\ldots,et_n)$,从 $P_i$ 的文档片段中识别和抽取出关系 $R_i$ 的实例及相应 XML 文档片段的结构模式.

## 1 方 法

一个 XML 文档可能由不同主题的文档段组成;包含同一种实体间关系的 XML 文档片段应该具有相似的元素和文档结构.因此,我们先提取包含相关实体类型的文档片段,然后按相似度聚类文档片段.

**算法 1**. 基于片段聚类的关系发现算法 FCRD.

输入:XML 文档集 $D$,密度 $d$;

输出:表示实体之间联系的各种关系及其出现模式.

(1) 首先由用户指定感兴趣的联系所涉及的实体类型名 $et_1,\ldots,et_n$;

(2) 利用 WordNet 和附加词典对实体类型名 $et_1,\ldots,et_n$ 进行语义扩充,得到扩展向量 $Ex(et_1),\ldots,Ex(et_n)$;

(3) 从文档集 $D$ 中自动提取出包含实体类型 $et_1,\ldots,et_n$ 的所有极小 XML 文档片段,组成集合 $P$;

(4) 调用 ATFC($P$,$d$),得到文档片段簇集 $C$; //聚类 $P$,使得每一个类对应实体 $et_1,\ldots,et_n$ 的一种关系

(5) for $C$ 的每一个文档片段簇 do

(6) {列出文档片段簇的根节点名,提示用户指定一个关系名;

(7) 创建待抽取关系的模式树,从与之匹配的文档片段中抽取关系实例及文档段结构模式}

运用上述算法,可以从 XML 文档集中发现并抽取出各种表示不同实体之间联系的关系信息.

### 1.1 XML文档片段间的相似度

根据相似性测量的信息论原理[4],对象 $a$ 和 $b$ 的相似度可用表示 $a$ 和 $b$ 相同部分的信息量在描述 $a$ 和 $b$ 的

信息总量中所占比例求得.如果能从多个角度观察 $a$ 和 $b$,则分别从每个角度测量其相似性.包含同一种实体间关系的XML文档片段通常具有相似的数据项和结构.因此,文档片段的相似度可以从元素语义和文档结构两个方面测量.基于文献[5]的相似度计算方法,本文采用分别计算文档段元素语义相似度和结构相似度,然后以两者加权和的方法计算文档段的相似度.

**定义 6**. 设 $t_1$,$t_2$ 为 XML 文档树节点标记,则 $t_1$ 和 $t_2$ 的相似度评分定义为

6: $t_1$ 和 $t_2$ 完全匹配;

5: $t_1$ 与 $Ex(t_2)$中除 $t_2$ 外的某元素完全匹配或 $t_2$ 与 $Ex(t_1)$中除 $t_1$ 外的某元素完全匹配;

4: $Ex(t_1)$和 $Ex(t_2)$中除 $t_1$ 和 $t_2$ 外的某两个元素完全匹配;

3: $t_1$ 和 $t_2$ 部分匹配;

2: $t_1$ 与 $Ex(t_2)$中除 $t_2$ 外的某元素部分匹配或 $t_2$ 与 $Ex(t_1)$中除 $t_1$ 外的某元素部分匹配;

1: $Ex(t_1)$和 $Ex(t_2)$中除 $t_1$ 和 $t_2$ 外的某两个元素部分匹配;

0: 不存在任何匹配.

如果两个节点标记的相似度评分大于0,则称这两个节点相似.

**定义 7**. 设 $p_1$,$p_2$ 为 XML 文档片段,则 $p_1$ 和 $p_2$ 的相似度为 $Sim(p_1,p_2)=\lambda_1 SemSim(p_1,p_2)+\lambda_2 StruSim(p_1,p_2)$,其中,$\lambda_1$,$\lambda_2$ 分别为语义相似度和结构相似度权值,$SemSim(p_1,p_2)$为语义相似度,计算方法为:将两段文档分别表示为矢量 $p_1=(\langle Ex(t_1^1), score_1^1\rangle,\ldots,\langle Ex(t_m^1), score_m^1\rangle)$和 $p_2=(\langle Ex(t_1^2), score_1^2\rangle,\ldots,\langle Ex(t_n^2), score_n^2\rangle)$,$Ex(t_j^i)$是文档 $i$ 的第 $j$ 个标签名的扩展向量;$score_j^i$ 是标签 $t_j^i$ 的相似度评分,取 $t_j^i$ 与另一文档片段最相似标签的评分.然后计算

$$SemSim(p_1,p_2)=\left(\sum_{i=1}^{m} score_i^1+\sum_{j=1}^{n} score_j^2\right)\Big/6(m+n).$$

$StruSim(p_1,p_2)$为结构相似度,计算方法为:对文档树节点编号,相似的节点编号相同.这样,节点树的每一条路径可用一个数字序列表示,通过挖掘同时出现在两段文档中的频繁序列找出相似路径集.然后计算

$$StruSim(p_1,p_2)=\frac{1}{N+1}\left[\left(\sum_{t=1}^{N}\frac{1}{L(R_t)}\times V(R_t)\right)+MR\right],$$

其中,$N$ 为节点较多的文档段的一级子树的数量,$R_t$ 为节点较多的文档段第 $t$ 个一级子树的根节点,$L(x)$为求树的层数函数,

$$V(R_t)=\begin{cases}F(R_t), & \text{如果}R_t\text{为叶节点}\\ F(R_t)+\dfrac{1}{N(C_t)}\displaystyle\sum_{e\in C}\frac{1}{L(e)}V(e), & \text{如果}R_t\text{为非叶节点}\end{cases},$$

$$F(R_t)=\begin{cases}1, & \text{如果}R_t\text{在相似路径中}\\ 0, & \text{否则}\end{cases}.$$

$C_t$ 为 $R_t$ 的儿子节点集合,$N(C_t)$为集合 $C_t$ 的基数,

$$MR=\begin{cases}1, & p_1\text{和}p_2\text{的根节点相似}\\ 0, & p_1\text{和}p_2\text{的根节点不相似}\end{cases}.$$

例 1:图 1 中(a)和(b)的语义相似度为 0.77,结构相似度为 0.61.取 $\lambda_1=\lambda_2=0.5$,图 1(a)和图 1(b)的相似度为 0.69.

**定理 1**. 相似性测量公式 $Sim(a,b)$具有下列性质:对任意的 XML 文挡片段 $a$ 和 $b$,

(1) 自相似度最大:$Sim(a,a)\geq S(a,b)$;

(2) 对称性:$Sim(a,b)=Sim(b,a)$;

(3) 单调性:$Sim(a,b)$随 $a$ 和 $b$ 相同部分的增加而增大,随 $a$ 和 $b$ 不同部分的减少而变小.

不难证明,公式 $Sim(a,b)$具有如上这 3 个性质.这也是相似性度量公式应具备的 3 个基本性质.

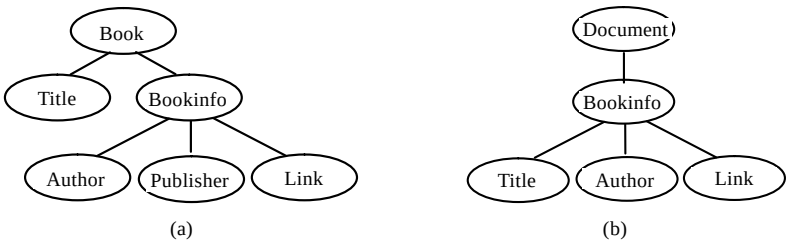


Fig.1 Two XML documents

图 1 两个 XML 文档

### 1.2 文档片段聚类

本文采用一种自适应阈值的文档片段聚类方法.在聚类过程中,根据相似度的变化情况自适应地动态选择阈值.基本思想是:从 XML 文档片段集中选择一个感兴趣的文档片段 $p$,计算 $p$ 与文档片段集中所有片段(包括 $p$)的相似度.将相似度值按递减排序形成一条非增曲线,用二阶导数过零点求相似度曲线拐点,由相邻二阶导数差最大的拐点(也是一阶导数最小)所在的相似度值作为阈值 $\varepsilon$.选取相似度超过阈值的文档片段构成一个候选簇.如果候选簇达到一定数目,即 DBSCAN 算法[6]中的密度,则构成一个文档片段簇.相似度值小于阈值的文档片段中可能包含了其他关系信息,可采用上述方法求出其他文档片段簇.

**算法 2**. 自适应阈值文档片段聚类算法 ATFC.

输入:文档片段集 $P$,密度 $d$;

输出:文档片段簇集 $C$.

(1) while $P\neq\varnothing$ do

(2) {从 $P$ 中选择一个 XML 文档片段 $p_0$;

(3) for $P$ 中的每一个 XML 文档片段 $p_i$ do 计算 $Sim(p_i,p_0)$; //计算 $p_i$ 与 $p_0$ 的相似度

(4) 将相似度值 $Sim(p_i,p_0)$按递减排序,选择拐向最大的点所对应的相似度值作为阈值$\varepsilon$;

(5) 令 $S=\{p'|Sim(p',p_0)>\varepsilon\}$,If $|S|\geq d$, then $S$ 构成一个文档片段簇存入 $C$;

(6) $P:=P-S$;}

### 1.3 抽取关系实例及其出现模式

每一个 XML 文档片段簇包含了用户感兴趣的实体之间的一种联系信息.从文档片段中抽取关系信息就是要找出与用户兴趣的关系模式匹配的所有子树,将对应的元素或属性抽取出来,组成一个个关系实例,并从匹配的文档段抽取出该关系信息的出现模式.首先创建用户挖掘请求的模式树,然后将该模式树与 XML 文档片段的节点树进行近似匹配.由于文档中标签名前后文往往决定了标签名含义,匹配时将极小 XML 文档片段根节点的父节点(如果有的话)也一同考虑.

**定义 7**. 设 $T_p=(V,E)$为用户兴趣的模式树,$T_f=(W,F)$为文档片段节点树.如果存在一个映射 $f:V\to W$,满足条件① $u=v\Leftrightarrow f(u)=f(v),u,v\in Domain(f)$;② $name(f(v))\in Ex(name(v))$;③ $u=parent(v)\Leftrightarrow f(u)=ancestor(f(v))$,则称 $T_f$存在一个 $T_p$ 的近似匹配.

其中,条件①要求 $f$ 单射;条件②要求文本树的节点名与模式树的对应节点名同名或同义,$Ex(name(v))$为节点 $v$ 标记名的扩展向量;条件③表示 $f$ 保持节点的祖先后代关系.

## 2 实 验

为检验本文提出的 FCRD 方法的可行性,我们选择发现表示“title”和“author”两者之间联系的关系信息为目标进行实验,测试其发现不同关系的能力.实验数据集 $D$ 采用 Wisconsin's XML data bank[7] bib 和 club 目录下所有文件以及 lindoc,sigrecord 目录下的部分文件,设置密度为 10.

根据指定的挖掘任务,系统从数据集 *D* 中提取出包含实体类型“title”和“author”的极小文档片段 212 个,组成文档片段集 *P*.聚类 *P* 时,系统首先选取第一个文档片段 bib_0_0.xml(结构如图 2 所示),计算 bib_0_0.xml 与 *P* 中所有文档段(包括 bib_0_0.xml)的相似度.图 3 的左图列出了片段 bib_0_0.xml 与源自目录 lindoc 和 sigrecord 的部分文档片段的相似度值,相似度都比较低,这是因为它们包含的关系不同.右图是按降序排列的片段 bib_0_0.xml 与 *P* 所有文档片段的相似度值.bib_0_0.xml 与 bib 目录下不同文档抽取出来的片段相似度均为 1.尽管这些片段与 bib_0_0.xml 不仅内容不同,结构也有差异,如 bib_0_0.xml 只有一个作者信息,有的片段却有多个作者,但由于这些片段的每一个元素名都与 bib_0_0.xml 的某个元素名相同,并且每一条路径都和 bib_0_0.xml 中的某条路径相似,所以按第 1.1 节的公式计算,其相似度为 1.

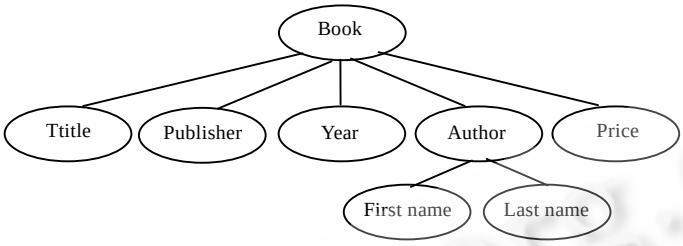


Fig.2 The tree of Bib_0_0.xml

图 2 Bib_0_0.xml 文档树

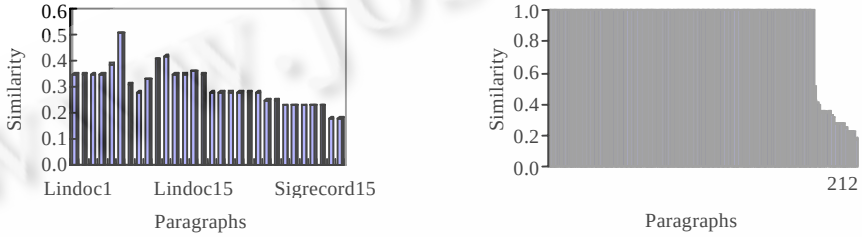


Fig.3 Similarities between Bib_0_0 and paragraphs of *P*

图 3 Bib_0_0 和 *P* 文档片段的相似度

用二阶导数过零点求相似度曲线拐点,由相邻二阶导数差最大的拐点所在的相似度值 0.6353 作为阈值.根据这个相似度阈值求得一个包含 182 个文档片段的聚簇.这个文档片段的聚簇包含了书名和作者的关系 Book(title,author),这个聚簇的文档片段全部源于 bib 目录,系统从中抽取了所有书名和作者的关系实例共 342 对,表 1 列出了其中的部分结果.这个关系的出现模式如图 2 所示.对 *P* 中第 1 个聚簇以外的其他文档片段,重复上述过程,从 *P* 中得到了另外 2 个对应不同关系的文档片段簇,这 2 个文档片段簇分别包含 linux 文档与其作者的关系 Linuxdoc(title,author)和 ACM 文章与其作者的关系 article(title,author).

**Table 1** Some results of experiment 1

**表 1** 实验 1 的部分结果

| Title | Author |
|---|---|
| Unix network programming | Richard Stevens |
| Crafting a compiler with *C* | Charles Fischer |
| Crafting a compiler with *C* | Richard LeBlanc |
| … | … |

为了进一步检验本文提出的方法,我们还从 Amazon(http://www.amazon.com)和 Barnes&Noble (http://www.barnesand noble.com)网站中选取 20 个不同的网页,根据网页数据的含义和结构,转换成不同格式的 XML 文档,加入了实验文档集 *D* 进行了实验.实验结果表明,对于标签名为 title 和 author 同义词、结构与

Wisconsin's XML data bank 差异不太大的 XML 文档,本方法也能从中将表示实体类型 title 和 author 之间关系的片段抽取出来,正确聚类,并全部挖掘出来.

## 3 总 结

本文提出了一种发现和抽取蕴藏在 XML 文档嵌套结构中实体间关系信息的方法 FCRD(fragment-clustering-based relation discovery).根据用户的兴趣,首先提取出包含相关实体的文档片段,去除与抽取任务无关的部分;然后对所选取的文档片段按语义和结构相似度进行聚类,使得包含同一关系信息的文档片段聚集在同一类;最后,从不同的文档片段聚簇抽取不同的关系信息.实验结果表明,对于包含有意义的标签的 XML 文档,只要实体类型扩展向量选择合适,本方法就能将给定文档集中所包含的关系信息全部抽取出来.由于聚类是在包含相关实体的极小文档片段集上进行的,并且采用了自适应阈值选定方法,本文所采用的聚类能够准确、有效地将包含不同关系的文档段区分开,从而准确地从目标文档段聚簇中抽取出用户指定的实体间联系的关系实例及出现模式.

FCRD 方法使用简单,用户只需给出相关的实体类型名.但在实现方面所面临的一个问题是,如何为扩展向量建立一个能够满足各种应用需要的有效的附加词典.下一步我们将继续深入探讨这个问题,进一步探讨更加精确的度量文档段之间的相似性等问题.

**References**:


[1] Chang CH, Kayed M, Girgis MR, Shaalan KF. A survey of Web information extraction systems. IEEE Trans. on Knowledge and Data Engineering, 2006,18(10):1411−1428.

[2] Brin S. Extracting patterns and relations from the world wide Web. In: Atzeni P, Mendelzon AO, Mecca G, eds. Proc. of the World Wide Web and Databases, Int'l Workshop WebDB'98. Valencia: Springer-Verlag, 1998. 172−183.

[3] Sundaresan N, Yi JH. Mining the Web for relations. Computer Networks: The Int'l Journal of Computer and Telecommunications Networking, 2000,33(6):699−711.

[4] Lin DK. An information-theoretic definition of similarity. In: Shavlik J, ed. Proc. of the 15th Int'l Conf. on Machine Learning. San Francisco: Morgan Kaufmann Publishers, 1998. 296−304.

[5] Lee JW, Lee KH, Kim W. Preparations for semantics-based XML mining. In: Cercone N, Lin TY, Wu XD, eds. Proc. of the 2001 IEEE Int'l Conf. on Data Mining. Washington: IEEE Computer Society, 2001. 345−352.

[6] Han JW, Kamber M. Data Mining Concepts and Techniques. New York: Morgan Kaufmann Publishers, 2000. 363−369.

[7] Query engine. http://www.cs.wisc.edu/niagara/data.html



吴扬扬(1957－),女,福建泉州人,教授,CCF 高级会员,主要研究领域为数据管理,数据挖掘.

陈锻生(1959－),男,教授,CCF 高级会员,主要研究领域为图像处理,模式识别.

雷庆(1980－),女,讲师,CCF 会员,主要研究领域为 Web 数据挖掘.

**YOKOTA Harou**(1957－),男,教授,主要研究领域为并行数据工程.