

基于领域样本查询的 Deep Web 数据库分类

鲜学丰^{1,2}, 赵朋朋¹, 辛 洁¹, 方 巍¹, 崔志明^{1,2}

(1 苏州大学 智能信息处理及应用研究所, 江苏 苏州 215006;

2 江苏省现代企业信息化应用支撑软件工程技术研究开发中心, 江苏 苏州 215104)

摘 要: 提出了一种基于领域样本查询的方法以分类这类 Web 数据库. 通过分析领域的高级查询接口自动获取领域主属性并使用领域知识为主属性构建查询样本, 然后对查询接口提交试探查询, 根据返回结果页面的结果模式和记录内容估计 Web 数据库与领域的相关程度. 通过在多个领域的 Web 数据库上进行实验验证, 说明该方法分类只提供简单查询接口的 Web 数据库是有效的, 取得了较高的分类精确率, 召回率和 F-measure 值.

关键词: Deep Web; Web 数据库; 数据库分类; 简单查询接口

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2010)03-0020-04

Classification of Deep Web Databases Based on the Domain Sample Query

XIAN Xue-feng^{1,2}, ZHAO Peng-peng¹, XIN Jie¹, FANG Wei¹, CUI Zhi-ming^{1,2}

(1 Institute of Intelligent Information Processing and Application, Soochow University, Suzhou 215006, China;

2 Jiangsu Province Support Software Engineering R&D Center for Information
Technology Application in Enterprise, Suzhou 215104, China)

Abstract: An approach based on the domain sample query is proposed in this paper to classify the web database. it obtains domain of the main attributes by analyzing descriptive attribute labels in the advanced query interfaces, the correlations of between web database with simple query interface and domain can be estimated by result schema and records of result pages, which obtained by submitting probing queries to simple query interface. The experiments on several domains have proved that this approach is effective and can achieve high classification precision, recall and F-measure values.

Key words: deep Web; Web database; database classification; simple query interface

1 引言

Deep Web 是指内容隐藏在查询表单之后的数据源, 这些数据只能通过填写表单(Form)提交查询由后台数据库动态产生, 它们不能被传统的搜索引擎索引. 因此, Deep Web 数据库资源的有效利用将成为未来 Web 信息检索和数据库领域的重要研究之一^[1].

为了有效地利用 Web 数据库中的信息, 需要将其按领域进行分类, 这些分类方法大致分为 pre-query 和 post-query. Pre-query 基本依赖于 Deep Web

数据库表单(forms)的可视特征^[2]和表单所在页面的信息^[3]. 此外一个领域网页可能同时包含多个不同领域的 Web 数据库^[4] Post-query 是通过对 Web 数据库提交查询, 利用返回结果对数据库进行分类. 即利用提交样本查询来实现分类的方法. 文献[5]等从返回查询结果数量来分析一个 Web 数据库属于哪个领域. 但是, 目前已有的工作仍未把 Web 数据库的分类问题彻底解决, 尤其是针对只提供简单查询接口的结构化 Web 数据库的分类.

为了解决只提供简单查询接口的结构化 Web 数据库的分类问题, 文中提出一种基于领域样本查

收稿日期: 2009-01-23; 修回日期: 2009-03-28

基金项目: 国家自然科学基金项目(60673092); 江苏省重大科技支撑与自主创新项目(BE2008044); 江苏省“六大人才高峰”项目(06-E-037); 江苏省研究生创新计划项目(CX08B_099z)

询的 Web 数据库分类方法. 在 Internet 的真实数据上,进行了大量的实验,实验结果表明,该方法在分类精确度、召回率和 F-measure 测度上均有较好的表现.

2 基于领域的查询样本集合构建

基于领域样本查询分类 Web 数据库的关键是如何构建一个合适的查询样本集合. 文中所指的查询样本是关键词或关键词的组合.

经过大量观测发现同一领域的 Web 数据库有相似的属性,并且有些属性几乎在所有 Web 数据库中都出现,文中称这些属性为领域主属性.

定义 1(领域主属性) 领域主属性是指该领域 Web 数据库中出现频率较高,最能代表领域特征的一组属性的集合,记为 DA.

如果能自动获得领域主属性,然后就可以针对每个主属性构建查询样本,那么这些查询样本就是与 Web 数据库内部属性密切相关的有效查询样本,就可以使用最少的查询样本得到最多数据库内容.

2.1 领域主属性获取

领域主属性的获取对构建查询样本集合和判断 Web 数据库与领域的相关程度密切相关,Deep Web 数据库的内部属性集合对于我们是不可视的,但是根据文献[2]可知高级查询接口的标签信息是 Web 数据库的一种局部视图,并且与数据库中最重要,最丰富信息的属性密切相关. 文中利用高级查询接口的标签信息获取领域主属性.

查询接口表单即为包含在 HTML 标签 <FORM> ... </FORM> 之内的相关内容. 网页表单的内部控件可以分三大类: INPUT 控件、SELECT 控件和 TEXTAREA 控件. 因此本文抽取表单中这三类控件的标签信息. 使用这些标签信息代表该 Web 数据库的属性,然后利用图 1 所示的算法抽取领域主属性.

领域主属性抽取算法如图 1 所示,该算法为 Form 的每一个标签产生一个标签模式 L^* ,然后判断 L^* 是否与 LableSet 中存在的模式 L^i 匹配,如果匹配, L^i 被更新,即 $L_{support}^i = L_{support}^i + 1$, $L_{support}^i$ 是 F 中存在 L^i 标签的数量. 否则, L^* 作为一个新模式添加到 LableSet. 最后, LableSet 中 $L_{support}^i \geq \theta$ 的标签作为领域主属性. 算法中两个关键函数 FormLableExtraction 和 SimilarLable, FormLable - Extraction 函数的功能是从表单中抽取标签. 同一领域的高级查询接口表示同一属性所用的单词往往有较大差异(存在大量的同义词或者缩写),尤其是中文查询接口. 为了统计的准确性,必须把这些同义词

和缩写考虑为同一个标签信息. SimilarLable 函数就是判断在 LableSet 中是否存在与 L^* 相同(包括同义词和缩写)的标签.

领域主属性抽取算法

```

1. Input A set of the domain advance query interface forms  $F = \{f_1, f_2, \dots, f_N\}$ 
   Lable word number threshold  $\theta$ , a similar threshold  $\epsilon$ 
2. Output. A set of main attributes DA for the domain
3. begin
4.  $DA = \{\}$ . Lable Set =  $\{\}$ ; //Lable Set is a set of all lables of interface forms
5. foreach  $f_i \in F$  do
6.  $R = \text{Form Lable Extraction}(f_i)$ ;
7. foreach  $r \in R$  do
8. create an empty  $L^* = \{L_{Node}^*, L_{Support}^*\}$ ;
9.  $L_{Node}^* = r_1, L_{Support}^* = 1$ ;
10. foreach  $L^i \in \text{LableSet}$  do
11. if  $\text{SimilarLable}(L_{node}^*, L_{node}^i) \geq \epsilon$  then
12.  $L_{support}^i = L_{support}^i + 1$  and break;
13. end
14. if cannot align  $L^*$  with any  $L^i \in \text{LableSet}$  then  $\text{LableSet} \leftarrow L^*$ ;
15. end
16. end
17. foreach  $L^i \in \text{LableSet}$  do
18. if  $L_{support}^i \geq \theta$  then  $DA \leftarrow L_{node}^i$ ;
19. end
20. end

```

图 1 领域主属性抽取算法

2.2 构建领域查询样本集合

由 2.1 节获得了领域主属性,但是领域主属性中存在一些属性,它的出现频率很高,但是由于它太具有区分性而不利于构建查询样本集合. 例如:图书领域的价格属性.

定义 2(构建领域查询样本属性) 构建领域查询样本属性为领域主属性 DA 中可以高效构建查询样本的属性集合,记为 BA.

文中将使用领域知识为 BA 的每个属性构建一组查询样本. 主要从三个数据源构建属性的查询样本.

(1) 数据库记录

数据库记录是很好的构建查询样本的资源,因此基于特定应用领域,选择一些数据库记录作为训练集,使用记录中与 BA 属性相对应的数据项训练每个属性得到属性的查询样本. 具体方法如下所示.

步骤 1: 获取完整的数据项值关键词,即把整个数据项的值作为一个词进行考察. 这种完整的属性值有利于在查询时得到最精确的返回结果. 使用条件 1 判断数据项值 V_i 是否为查询样本.

条件 1: IF $\text{Count}(V_i) \geq \sigma_1$, then V_i 作为该属性的一个查询样本, $\text{Count}(V_i)$ 为 V_i 在训练集中出现的次数, σ_1 为指定阈值.

步骤 2: 为了获得更细粒度的查询样本, 把数据项的所有值作为一个文档处理. 文中考虑到中文文档与英文文档之间存在着较大的差异. 因此文档内容特征提取时使用的步骤与英文文档略有不同. 首先对文档进行除停用词预处理, 然后对于英文文档进行词干抽取, 将具有不同词缀的词合并成一个词, 降低文档挖掘的特征词总数; 对于中文文档进行汉字分词. 经过这两步的处理得到文档的特征词集合, 然后产生该文档的频繁词作为属性的查询样本. 查询样本抽取算法如图 2 所示. 在算法中, 4~5 行是用 stoplist 和 stemming 对 text 进行预处理, $text_i$ 是第 i 个属性对应数据库的字段的所有值组成的文本, $word_j$ 是 $text_i$ 文字中的一个单词, $tf(text_i, word_j)$ 是 $word_j$ 在文档 $text_i$ 出现的绝对频率.

```

1. Input texti as a paragraph, a stoplist file S, a frequency threshold
   ε.
2. Output A set Ki consisting of all frequency words of topic phrase.
3. begin
4. Delete stopwords from texti according to stoplist,
5. Process texti using stemming algorithm;
6. Ki = {};
7. foreach wordj ∈ texti do //wordj = word of texti which index is
   j;
8. if wordj ∉ Ki and tf(texti, wordj) > ε then
9.   Ki ← wordj,
10. end
11. end
12. end

```

图 2 查询样本抽取算法

(2) 公共分类目录

公共分类目录是领域查询样本获取的重要资源, 例如: yahoo Directory, Google Directory 和 Open Directory 等由于分类目录里的主题词是对现实世界中各领域的简单概括或者该领域包含的子部分. 我们能够从公共分类目录收集大量词汇作为这些领域的查询样本.

(3) 例子文档

为了获得更全面的查询样本, 还为 BA 中每个属性选取一组相关的训练文档, 获取这些文档的特征频繁词作为属性的查询样本.

通过以上的训练得到 BA 中属性的领域查询样本集.

定义 3 (领域查询样本集合) 领域查询样本集合为 $K = \{K_1, K_2, \dots, K_L\}$. 其中, $K_i = \{V_{i1}, V_{i2}, \dots, V_{iN}\}$ 为 BA 中第 i 个属性的查询样本集合.

3 Web 数据库与领域的相关性度量

为了估计 Web 数据库与领域的相关程度, 必须

获得 Web 数据库的内容, 使用 HTTP 和 Winsock 技术利用 K 中的查询样本填充表单, 经过若干次提交得到数据库的返回的结果页面. 以 albooks.com 的返回结果页面为例, 每条记录都由 Null、Author (by)、ISBN/LIPC、Publisher、Pub Date、Price 等属性及其对应值组成, 而这一组领域相关属性就构成了 Web 数据库的结果模式 (Result Schema). 结果模式作为 Web 数据库的一个局部视图, 比查询接口模式更具体地展现了 Web 数据库更多的属性信息.

首先对 Deep Web 结果模式作个形式化定义. 给定一个 Deep Web 结果页面, 不考虑网页上噪音数据 (广告、一般链接、其他与查询无关的数据等), 只观察查询返回的数据区域. 抽取出数据区域的属性字段名标签 $A = \{a_1, a_2, \dots, a_n\}$, 以及记录 $R = \{r_1, r_2, \dots, r_m\}$, 对每个 r_j 表示为集合 $V = \{v_{j1}, v_{j2}, \dots, v_{jn}\}$, 其中 v_{ji} 表示为记录 r_j 中属性字段名 a_i 对应的属性值. 例如表 1 中 Null、Author (by)、ISBN、Publisher 等都是属性字段名标签; Thinking in java、Bruce Eckel、0130273635、Prentice Hall 等均为字段对应的属性值. A 重组后放置模式集合中, 称为结果模式. 实际上, 结果模式抽取就是从可视的结果页面中获取后台数据库的部分模式.

表 1 部分属性字段名及其对应属性值

Null	Author (by)	ISBN/LIPC	Publisher
Thinking in java	Bruce Eckel	0130273635	Prentice Hall

由于结果模式是 Web 数据库局部模式, 因此能在较大程度上反映领域特征, 同时结果页面由若干条记录组成. 所以文中通过结果页面的结果模式和记录内容两个层次判断 Web 数据库与领域的相关程度, 以分类 Web 数据库.

设对于某一领域 D , 经过若干次查询后得到的结果页面集合 $P = \{p_1, p_2, \dots, p_n\}$, 通过文献 [6] 中的方法获得返回 Web 数据库的结果模式为 $S = \{s_1, a_2, \dots, a_n\}$, 通过文献 [7] 方法获得结果页面包含的记录集合 $R = \{r_1, r_2, \dots, r_m\}$.

3.1 结果模式与领域主属性的相似程度

设 $S = \{s_1, s_2, \dots, s_n\}$ 是结果模式, $DA = \{da_1, da_2, \dots, da_k\}$ 是领域主属性的集合. 由于结果模式比查询接口模式更具体地展现了 Web 数据库的模式信息, 所以在大多数情况下 $|S| \geq |DA|$. 结果模式与领域主属性的相似程度定义为:

定义 4 $\text{Similar}_1 = \frac{|I|}{|DA|}$, 其中 $|DA|$ 是领域

主属性数量, $I = \sum_{j=1}^k x_j$, 其中

$$x_j = \begin{cases} 1, & \text{da}_j \in S \\ 0, & \text{otherwise} \end{cases}$$

式中, da_j 表示 da_j 及其同义词。

由于领域主属性代表该领域的主要特征, 结果模式包含的领域主属性及其同义词越多说明该 Web 数据库与该领域越相关, 即 Similar_1 越大, 则 Web 数据库与领域相关程度越高。

3.2 记录与领域的相似程度

设 $R\{r_1, r_2, \dots, r_m\}$ 为查询返回的记录, 任意一条记录 r_i , $C_i = \{c_1, c_2, \dots, c_R\}$ 是 r_i 的特征词集合。

如果记录满足条件 2, 则该记录属于领域 D :

条件 2: 设定一阈值 σ_2 , 如果, $\frac{t}{|C_i|} \geq \sigma_2$, 则 r_i 属于领域 D . $|C_i|$ 为 r_i 的特征词集大小, $t = \sum_{j=1}^k x_j$, 其中 $x_j = \begin{cases} 1, & c_j \in \text{DA} \cup K \\ 0, & \text{otherwise} \end{cases}$, $\text{DA} \cup K$ 是所有领域查询样本和领域主属性及其同义词的集合。

记录与领域 D 的相似程度定义见定义 5。

定义 5 $\text{Similar}_2 = \frac{L}{|R|}$, $|R|$ 为查询返回记录总数, L 为 R 中属于领域 D 的记录数。

3.3 Web 数据库分析

经过 3.1 节和 3.2 节中的计算, Web 数据库与领域的相关程度定义见定义 6。

定义 6 设定一阈值 σ_3 , 如果 $(w_1 \text{Similar}_1 + w_2 \text{Similar}_2) \geq \sigma_3$, 则 Web 数据库属于领域 D . 其中 $(w_1 + w_2) = 1$, Similar_1 和 Similar_2 分别为结果模式与领域主属性的相似程度和记录内容与领域的相似程度。

4 实验分析

为了验证基于领域查询样本分类, 且只提供简单查询接口的 Web 数据库方法的有效性, 在 Internet 中选择了 4 个领域下的 685 个 Deep Web 数据库查询接口 Form, 涉及 Book、Job、Music 和 Movie 领域。

实验表明, 文中提出的方法对只提供简单查询接口的结构化 Deep Web 数据库进行分类达到了较好的实验效果。数据表明 F-measure 在 Book 和 Job 领域可以达到 88% 以上, 即使在 Music 领域 F-measure 最低, 但是仍然达到了 81%。

5 结束语

为了准确分类只提供简单查询接口的结构化 Web 数据库, 文中提出了一种基于领域查询样本的分类解决方法, 通过利用领域的高级查询接口获得领域主属性, 利用领域知识构建了查询样本。然后还给出了利用返回结果页面的结果模式和记录内容判断 Web 数据库与领域相关程度的方法。通过在 4 个领域的 Web 数据库上进行实验验证, 说明文中提出的方法可以有效分类只提供简单查询接口的结构化 Web 数据库。

参考文献:

- [1] Chang KCC, He B, Li CK, et al. Structured databases on the web: observations and implications [J]. SIGMOD Record, 2004, 33(3): 61-70.
- [2] 赵朋朋, 高岭, 崔志明. 基于查询接口特征的 Deep Web 数据源自动分类 [J]. 微电子学与计算机, 2006, 23(10): 47-50.
- [3] 马军, 宋玲, 韩晓晖, 等. 基于网页上下文的 Deep Web 数据库分类 [J]. 软件学报, 2008, 19(2): 267-274.
- [4] 刘伟, 孟小峰, 孟卫一. Deep Web 数据集成研究综述 [J]. 计算机学报, 2007, 30(9): 1475-1488.
- [5] Ipeirotis P G, Gravano L, Sahami M. Probe, count, and classify: categorizing hidden web databases [C]// Proc. of the 19th ACM SIGMOD International Conference on Management of Data. Santa Barbara, 2001: 67-78.
- [6] Zhong Hua, Zhao Pengpeng, Gao Ling, et al. Vision-based deep web result schema automatic extraction [J]. Computational Information Systems, 2007, 3(4): 1515-1522.
- [7] Meng X F, Lu H J, Wang H Y, et al. SG-WRAP: A schema-guided wrapper generator [C]// Proc. of ICDE. China: Beijing, 2002: 331-332.

作者简介:

鲜学丰 男, (1980-), 博士研究生, 讲师. 研究方向为 Deep Web、Web 数据挖掘、信息集成。

赵朋朋 男, (1980-), 博士, 讲师. 研究方向为 Web 数据挖掘、Deep Web、Web 信息抽取。

辛洁 女, (1982-), 博士研究生. 研究方向为 Deep Web、数据挖掘、语义 Web。

方巍 男, (1975-), 博士研究生. 研究方向为 Web 数据挖掘、语义 Web、Deep Web。

崔志明 男, (1961-), 博士生导师, 教授. 研究方向是智能信息处理、计算机网络应用与数据库应用。