文章编号:1001 - 9081(2006)09 - 2024 - 04

Deep Web 查询接口选择

郑冬冬,崔志明
(苏州大学 智能化信息处理及应用研究所,江苏 苏州 215006)
(udbtuu2003@163.com)

摘 要:越来越多的信息隐藏在 Web 查询接口之后,在此情况下如何寻找与用户查询最相关的数据源接口就变得越来越重要。文中提出了一种 Deep Web 查询接口选择算法,该算法是完全依赖于查询接口特征的。给定大量异构的 Deep Web 数据源,目标是选择与用户查询最相关的查询接口集。通过对实际查询接口特征的观察,发现了查询接口上谓词间的相关性。基于此发现,设计了一种基于共同出现谓词相关度模型的数据源选择算法,用于选择与用户查询最相关的查询接口集。

关键词:谓词模型; 接口对象; 动态选择
中图分类号: TP311.132 **文献标识码**:A

Deep Web query interface selecting

ZHENG Dong-dong, CUI Zhi-ming
(*Institute of Intelligent Information Processing and Application, Soochow University, Suzhou Jiangsu* 215006, *China*)

Abstract: As Web develops, more and more data has become available under Web query interface. Therefore, how to find the data-sources that are most relevant to the user's requirements has become more and more important. This paper presented a Deep Web query interface selection arithmetic, which completely depended on the characteristics of query interface. Given numerous heterogeneous Deep Web data sources, we aimed at selecting sources most relevant to the user's requirements. By allowing the users to input an imprecise initial query, our system found appropriate sources for them. We observed the characteristics of query interface and found out the relationships between predicates. Based on this discovery, an algorithm based on co-occurrence predicate model for capturing the relevance of attributes was designed. It can be used to select the sources most relevant to the user's requirments.

Key words: predicate model; interface object; dynamic selecting

0 引言

Deep Web,Hidden Web 和 Invisible Web 均指同一个概念,它是一个与 Surface Web 相对应的概念。2001 年,Christ Sherman、Gary Price 等人对 Deep Web 的定义为:虽然通过互联网可以获取,但普通搜索引擎由于受技术限制而不能或不作索引的那些文本页、文件或其他通常是高质量、权威的信息。而 Deep Web 信息量要比 Surface Web 信息量多得多[1,2]。由于 Deep Web 页面信息多是由结构化的关系数据库产生的,因此信息质量很高。

面对海量 Deep Web 数据源,如何寻找与用户查询最相关的数据源已越来越受到人们的关注。关于 Deep Web 数据源选择的相关研究很少,其中文献[3]提出了一种基于概率方法的数据源选择,文献[4]提出了使用收缩方法选择数据源。但他们主要是针对文档数据库进行数据源选择,它需要查询提交后台数据库,因此效率比较低。Deep Web 数据源通常提供给用户两种不同的访问模式:一种是用于用户检索的查询接口模式; 另一种是用于用户浏览结果的结果模式。这里提出的 Deep Web 数据源选择算法是基于查询接口特征的,这里研究的查询接口主要是结构化的,其对应的后台数据库也多是结构化的。下面首先给出数据源的特征描述。

1 Deep Web 数据源特征表示

Deep Web 查询接口是获取后台数据库内容的唯一入口。在查询接口页面中含有的控件元素类型有:文本框(textbox)、单选按钮(radio)、复选框(checkbox)和选择列表框(selection list)。查询接口允许用户键入相应信息来查询数据库内容。Deep Web 查询接口是嵌在 HTML 页面中的,由标签名和相对应的元素来构成。

查询接口抽取[5~10]已经得到了广泛的研究。借鉴相关研究,这里将 Deep Web 查询接口抽象为一个对象 DWI(Deep Web Interface),它包括查询接口要完成的功能,如汽车导购、图书查找、产品定购等; 需要用户填写或选择具体内容的谓词属性,如图书的名称、作者、汽车价格等信息。查询接口上的按钮包括了连接后台数据库的查询方式,可以表示成对象对应的操作方法。Deep Web 查询接口通过谓词条件来访问后台数据库。谓词条件隐含了 Deep Web 查询接口之后的语义模型。

所谓谓词就是指定了查询接口上一个元素对应的标签名、内部属性名、一个或多个修饰语及其值域。它是查询接口上最小的语义单位。一个谓词由一个四元组[标签名; 内部属性名; 修饰语; 值域]构成。查询接口上所有谓词构成的

收稿日期:2006 - 03 - 24; **修订日期**:2006 - 06 - 15 **基金项目**:教育部高校博士学科点科研基金资助项目(20040285016); 江苏省高技术研究计划资助项目(BG2005019); 教育部科研重点资助项目(205059)

作者简介:郑冬冬(1980 -),男,河南焦作人,硕士研究生,主要研究方向:Web 数据挖掘、搜索引擎; 崔志明(1961 -),男,江苏苏州人,教授,博士生导师,主要研究方向:智能化信息处理、计算机网络、数据库.

集合称这个查询接口所对应的谓词模板。例如一个图书查询接口可能由作者、ISBN 号、价格等多个谓词构成。其中谓词的标签名和内部属性名包含了该谓词的语义信息;修饰语是对谓词值的限制方式,如价格小与、大于等信息属于谓词修饰语;值域包含了该谓词值所属的数据类型、可取的值列表或所包含的元素集合、查询接口上所提供的默认值。

查询接口上的每个谓词可能由一个或多个元素组成。每个元素属于四种类型中的一种,即上面提到的控件类型:文本框、单选按钮、复选框和选择列表框。每个元素也由:元素内部属性名、元素标签名、元素值域构成。如图 1 图书查询接口所示。其中,谓词"请选择文献类型"它包含有三个元素。

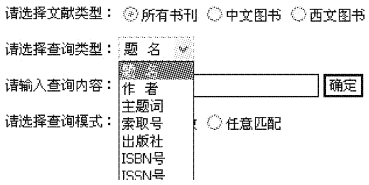

图 1　图书查询接口

P_1 = [请选择文献类型; paper; 其中之一; [文本,{E_{11}, E_{12}, E_{13}},所有书刊]]

E_{11} = [所有书刊,all_book,∅]

E_{12} = [中文图书,chinese_book, ∅]

E_{13} = [西文图书,english_book, ∅]

P_2 = [请选择查询类型; type; 其中之一; [文本,{作者,主题词 …},题名]]

P_3 = [请选择查询内容; content; 含有; [文本,∅,]∅]

P_4 = [请选择查询模式; schema; 其中之一; [文本,{E_{41},E_{42}},完全匹配]]

E_{41} = [完全匹配,whole,∅]

E_{42} = [任意匹配,any,∅]

因此查询接口对象 DWI 表示为:$DWI = (S,P,M)$。其中 S 反映了接口对象功能等的特定信息,它包含:接口对象的名字(表单标签名)和该接口站点的 URL 等基本信息。$P = \{p_1, p_2, \cdots, p_n\}$ 为接口对象所对应的谓词模板,M 为接口对象所提供的方法。建立了 DWI 对象后,用户就可以提供一个面向对象的查询来检索其所需要的信息。

2　Deep Web 数据源选择

给定一个 Deep Web 数据源接口集和一个由多个谓词构成的用户查询,目标是寻找最能满足用户查询需要的数据源子集。下面首先给出相关的形式化定义。

定义 1　Deep Web 数据源接口集(Source Interface Set):假定某领域内 Deep Web 数据源接口集为 $\{dwi_1, dwi_2, \cdots, dwi_m\}$,每个数据源接口 dwi_i 都对应一个出现在查询接口上的谓词 P_i 组成的谓词模板,谓词模板中的所有谓词的联合为 Λ,有 $\Lambda = \cup P_i$。

定义 2　数据源排序(Source Ranking):给定 Deep Web 数据源接口集和用户查询 $Q = \{a_{v1}, \cdots, a_{vk}\}$,目标是寻找与用户查询按相关度大小排列的数据源序列 $\Re = \{dwi'_1, dwi'_2, \cdots, dwi'_m\}$。

定义 3　数据源选择(Source Selecting):给定 Deep Web 数据源查询接口集和用户查询 $Q = \{a_{v1}, \cdots, a_{vk}\}$,目标是寻找与用户查询相关度值大于某一规定阈值 λ 的数据源查询接口集 $\delta = \{dwi'_1, dwi'_2, \cdots, dwi'_t\}$。

2.1　数据源选择策略

一般的数据源查询接口选择算法是基于抽取到的数据源接口对象 DWI,然后计算它与用户查询的相关度大小来选择。如对于给点目标表单接口对象 $DWI = (F,P,M)$ 和查询 $Q = (W,C,O)$ 通过计算 Q 和 DWI 对应的三个组成部分的相似度来衡量用户查询与数据源的总体相似度。因此有如下数据源相关度值计算公式:

$$Similarity(DWI,Q) = \alpha S(F,W) + \beta S(P,C) + \gamma S(M,O) + \delta S((F,P,M),(W,C,O))$$

其中 $\alpha,\beta,\gamma,\delta$ 为系统定义的常量,且有 $\alpha + \beta + \gamma + \delta = 1$。

由于 DWI 对象对应的谓词模板各不相同,而用户查询可以任意输入其中的某几个谓词项值,查询项和 DWI 对象没有很好的对应关系。因此在上述公式的最后添加一个修正项,它将表单接口对象 DWI 和查询 Q 中所有的文本一起作为总体来比较其相似度。

一般数据源选择算法利用用户查询与查询接口的相关度来进行数据源选择。然而用户查询通常是不准确的,因此一般数据源选择算法没有最大化利用查询接口特征。而我们设计的数据选择算法是基于查询接口中共同出现的谓词相关度模型,充分利用了查询接口特征。从某种意义上说它将用户查询进行了扩展,因此所选择的查询接口集更能满足用户的查询要求。共同出现谓词相关度模型是基于如下实践发现:

1)相关谓词出现在相关的数据源中,即出现在查询接口中的谓词总是与数据源所在的主题领域相关的;

2)相关数据源中包含有相关的谓词,即数据源与某主题领域的相关性通过其查询接口中的出现的谓词来体现。

基于以上发现,我们可以构造一个基于共同出现谓词的联结图。它类似 PageRank 算法中用于反映页面重要程度的页面间的联结图。使用这个图,可以发现与用户查询或与该主题相关的其他谓词。同样类似于计算页面重要程度的方法,我们可以计算查询接口中谓词与该领域的相关度值,以此判定哪些查询接口与该领域更相关。

2.2　谓词相关度模型

在 Λ 上有些谓词经常出现在该领域内许多数据源接口上,然而还有些谓词很少出现在该领域内数据源接口上。经常出现在数据源查询接口上的谓词提供了很重要的信息,即这些谓词是跟该主题领域非常相关的。为了评价一个谓词在某主题领域内的重要程度,这里提出了谓词相关性得分,它表示这个谓词与该主题相关度的大小。以下所说的相关性均指与所属主题的相关性。

我们提出一种通过判断谓词与跟它共同出现的其他谓词的相关性程度来确定该谓词与该主题的相关性程度的方法。两个谓词如果共同出现在某同一数据源中,则认为它们是共同出现的。下面给出共同出现的两个谓词相关性程度值,或称为两个谓词相关性权重计算方法。

$$w_{ij} = \frac{countDWI(a_i \wedge a_j)}{\sum_{a_k \in \Lambda} CountDWI(a_i \wedge a_k)}$$

$countDWI(a_i \wedge a_j)$ 表示查询接口集中同时有谓词 a_j 与 a_i 出现的数据源个数。$\sum_{a_k \in \Lambda} countDWI(a_i \wedge a_k)$ 表示查询接口集中有谓词 a_i 出现的所有数据源个数。w_{ij} 反映了谓词 a_i 与谓词 a_j 之间的相关性权重。从上述公式也可以看出 w_{ij} 与 w_{ji} 所代表的意义和值都是不同的。依据共同出现谓词之间的相关度模型可以推导出谓词本身与某主题领域的相关度。

用户查询中的谓词与用户兴趣非常相关,也许有些谓词没有出现在用户查询中。那些没有出现在用户查询中的谓词可能与出现在用户查询中的谓词相关性很高,因为它们经常共同出现在该领域内多数数据源接口上。任何谓词都会影响与其经常共同出现的谓词在该主题领域内的重要程度,这就导致每一个谓词都与该主题领域有一个相关度,在此称为谓词相关度。

依据谓词相关度,可以区别不同谓词在该主题领域内的重要程度。一个谓词影响每一个与该谓词共同出现在数据源接口中的其他谓词的相关性,同时它又受到与其共同出现谓词的影响。下面给出了一种计算谓词相关度值的公式:

$$P(a_j) = \sum_{a_i \in \Lambda \setminus a_j} d.\, w_{ij}.\, P(a_i)$$

这里用 $P(a_j)$ 表示谓词 a_j 的相关度值,d 为衰减因子,它类似于 PageRank 算法中的系数常量,$P(a_j)$ 值和所有与其共同出现的谓词 a_i 的相关度值 $P(a_i)$ 及它们之间的相关性权重 w_{ij} 有关。$P(a_j)$ 值计算中考虑了所有属于 Λ 的(即该主题领域内)所有谓词,即使那些没有与谓词 a_j 共同出现的谓词,但是那些谓词与 a_j 的相关性权重 w_{ij} 为0。由上可知,谓词相关性给我们描述了谓词与某主题领域相关程度的方法,它也可以用来构建某主题的统一查询接口。

2.3 数据源动态选择算法

下面描述如何使用谓词相关度来计算用户查询与某数据源的相关性得分,从而对数据源进行一个相关性排序。每个数据源查询接口对应一个谓词模板。既然谓词相关度值给出了谓词与某用户感兴趣主题相关程度,那么包含有更多具有较高相关度值的谓词所在的数据源查询接口则与用户查询就更相关。下面给出计算数据源相关性得分的公式:

$$score(dwi_i) = \sum_{a \in P_i} P(a)$$

上式很容易理解,但它是有缺陷的,如果一个数据源接口包含有很多低相关度值的谓词,即使含有很多具有较高相关度值的谓词,那么这个数据源也可能不是用户所感兴趣的数据源。因此,数据源相关性得分的计算方法可以做如下调整:那些含有相关度值较高的谓词越多的,同时含有相关度较低的谓词越少的数据源接口是与用户查询需求越符合的数据源。因此改进后的公式为:

$$score(dwi_i) = \sum_{a \in P_i} P(a) / |P_i|$$

上式中 $|P_i|$ 表示数据源接口所含谓词个数,它是用来克服数据源接口含有很多低相关度谓词因素。看上去改进后计算数据源相关性得分的公式很完美,事实上改进后的公式依然存在缺陷。因为我们计算谓词相关度的方法是基于查询接口特征的,而没有真正考虑查询接口上的语义。但上述改进至少考虑了低相关度谓词的因素。可以看出计算谓词相关度的方程是迭代方程,因此有必要设计也设计一种迭代算法来计算谓词相关度值。

下面提供一种迭代算法来计算谓词相关度估计值的方法。基本思想是一些谓词的相关度估计值在每次迭代过程中被重新计算,因此它们就触发了其他谓词的相关度值的改变。这些被重新计算相关度的谓词被放入一个队列中用于下一次迭代过程。下面举例说明谓词相关度估计值的计算方法。假设在某次迭代过程中,谓词 a_j 相关度值为 β。由谓词相关度值的计算公式可知:

$$\beta = \sum_{a_j \in \Lambda \setminus a_j} d.\, w_{ij}.\, P(a_i)$$

先假设某次迭代过程中谓词 a_1 相关度值从 α_1 变化到 α'_1,它导致谓词 a_j 相关度值从 β 变化到:

$$\beta^* = d.\, w_{1j}.\, \alpha'_1 + \sum_{a_i \in \Lambda \setminus a_j, a_1} d.\, w_{ij}.\, \alpha_i$$

上述公式推导如下:

记 δ_{1j} 为由于 $P(a_1)$ 变化而引起 $P(a_j)$ 的变化量,其中 $\delta_{1j} = d.\, w_{1j}.\, (\alpha'_1 - \alpha_1)$ 则变化后的 $P(a_j)$ 值为:

$$\begin{aligned}\beta' &= \delta_{1j} + \beta \\ &= d.\, w_{1j}.\, (\alpha'_1 - \alpha_1) + \sum_{a_i \in \Lambda \setminus a_j} d.\, w_{ij}.\, P(a_i) \\ &= d.\, w_{1j}.\, \alpha'_1 + \sum_{a_i \in \Lambda \setminus a_j, a_1} d.\, w_{ij}.\, P(a_1)\end{aligned}$$

我们看到 β^1 与 β^* 值是相同的。

当 $P(a_1)$ 变化时谓词 a_1 传递其相关度值的变化量 δ_{1j} 给谓词 a_j,然后谓词 a_j 接收此变化量来调整自身新的相关度值,这是计算谓词相关度算法的核心思想。在每次迭代过程中,算法重新调整每个具有变化量 δ 的谓词的相关度值。在迭代过程中很多与 a_j 共同出现谓词相关度可能会变化,同时它们将发送其变化量 δ 给谓词 a_j。即可能有多个谓词将自己的变化量都发送给谓词 a_j,所有被传到 a_j 的变化量 δ 都将被记录下来,以便计算新的 $P(a_j)$ 的值。

为方便记忆,对每个谓词 a_u(其原始相关度值为 α_u)设置一个相关度值变化量 Δ^u,当正在计算谓词 a_u 新的相关度值时,将其变化量 Δ^u 加上,即 $\alpha'_u \leftarrow \Delta^u + \alpha_u$。当谓词 a_u 已经调整过后其相关度值后,Δ^u 设置为0。

相应于谓词 a_u 的变化,它将 δ_{uv} 传递给每个与 a_u 共同出现的谓词 a_v,其中 $\delta_{uv} = d.\, w_{uv}.\, (\alpha'_u - \alpha_u)$。同时每个与 a_u 共同出现的谓词 a_v 合计上其自身相关度值的变化量 Δ^v,这样谓词 a_v 相关度值总的变化量为:$\Delta^v \leftarrow \Delta^v + \Delta_{uv}$。

因此,每次谓词 a_u 收到由与它共同出现的谓词产生的变化量 δ,它更新其相关度值 Δ^u。最终在某次迭代过程中,当 $P(a_u)$ 修改后,Δ^u 被用于它最近的估计值上来获取其新的相关度值。

由于数据源接口上谓词表现形式或个数各不相同,同时数据源接口在动态地不断更新,因此只要提供数据源查询接口对象 *DWI* 就可以动态计算哪些数据源与用户查询更相关。下面给出数据源动态选择算法:

```
01  ∀a_i ∈ v, Δ^i ← 1; UPDATEORADD(ζ, a_i, Δ^i)
02  While ζ 非空且没有达到上限
03  (α_u, Δ^u) ← POP(ζ)
04  α_u ← P[u]
05  α'_u ← Δ^u + α_u
06  for each a_v ∈ adjacent[a_u]
                          //a_v 是与 a_u 共同出现的谓词
07  (α_v, Δ^v) ← PEEP(ζ, v)
08  // 如果不存在则设置 Δ^v 为 0
09  δ_uv ← d. w_uv. (α'_u − α_u)
10  Δ^v ← Δ^v + δ_uv
11  UPDATEORADD(ζ, a_i, Δ^v)
12  P[u] ← α'_u
13  ℜ ← RANKDWI(P)
14  ℜ ← SourceSelkect (ℜ, λ)
15  return ℜ
```

上面数据源排序算法的输入是用户查询谓词集 v，输出是按与用户查询相关度大小选择好的数据源集 $\Re$。数据源选择过程中,只需要设定一个数据源相关性得分的一个阈值 λ,大于此阈值的选为用户提交查询的数据源接口。一旦选择了数据源查询接口集,就可以将用户查询转换到所选定的多个目标查询接口上,以产生查询,获取相应的结果页面。

2.4　实验分析

为了验证算法的有效性,本文使用 Web 目录服务(www.yahoo.com.cn)手工搜集了一个 Deep Web 查询接口集。其中包含 591 查询接口和共 373 个谓词,它覆盖了 8 个不同的主题领域。算法以每个主题查询接口集为输入,得到了与每个主题最相关的谓词列表,如表 1 所示。

表 1　各主题领域最相关谓词

主题	相关度最高的前 3 个谓词
书籍	作者,书名,ISBN 号
汽车导购	品牌,价格,系列
工作	类型,地点,薪水
宾馆	级别,位置,价格
机票	日期,起点,终点
电影	片名,类型,主演
音乐	歌名,音乐家,类型
手机导购	品牌,类型,价格

我们拿汽车导购主题中抽取的 10 个查询接口为例,给出了汽车导购主题查询接口数据源相关性排序列表,如表 2 所示。

表 2　汽车导购主题查询接口相关性列表

排序	数据源	相关度值
1	Dwi_3	0.90
2	Dwi_{10}	0.83
3	Dwi_4	0.73
4	Dwi_6	0.73
5	Dwi_7	0.73
6	Dwi_9	0.62
7	Dwi_5	0.53
8	Dwi_8	0.13
9	Dwi_2	0.10
10	Dwi_1	0.10

dwi_1 = [http://www.cbicn.com/car/index.asp; {所有分类,品牌,买车}]

dwi_2 = [http://used.chinacars.com/; {价格,新闻}]

dwi_3 = [http://auto.sina.com.cn/; {产地,系列,品牌,价格}]

dwi_4 = [http://cn.autos.yahoo.com/; {系列,品牌}]

dwi_5 = [http://www.autohm.com.cn/sltautohm/userUI/index.html; {国别,产地,用途类型,车体结构,变速方式,排量,品牌,型号,省份,地区,价格}]

dwi_6 = [http://auto.enorth.com.cn/; {产地,品牌,型号,价格}]

dwi_7 = [http://www.autohome.com.cn/; {品牌,车型,车系}]

dwi_8 = [http://auto.tfol.com/10091/10412/index.shtml; {产地,品牌,车型,价格,城市,查询方式}]

dwi_9 = [http://www.b-car.com/; {售价,产地,品牌}]

dwi_{10} = [http://www.chinaauto.net/; {品牌,车型,价格}]

3　结语

基于共同出现谓词模型还可以用于 Deep Web 数据源聚类与分类[11,12],它们都是 Deep Web 信息集成[7,8,13]的关键技术,本文提出一种基于共同出现谓词相关度模型的数据源选择算法,它用于发现与用户查询最相关的查询接口集。此方法完全基于查询接口自身特征,它没有考虑数据源后台数据库的语义特征。下一步的主要工作是设计相应系统来验证我们方法的算法,通过实验改进计算数据源相关性得分的方法。

参考文献:

[1] HE B, MITESH p, ZHANG Z, *et al.* Accessing the Deep Web: A Survey[R]. Technical report, Department of Computer Science, University of Illinois at Urbana-Champaign, 2004.

[2] CHANG KCC, HE B, LI C, *et al.*. Structured databases on the Web: Observations and implications[J]. SIGMOD Record, 2004, 33(3):61-70.

[3] LIU VZ, RICHARD JC, LUO C, *et al.* Drop: A probabilistic approach for hidden Web database selection using dynamic probing [A]. ICDE 2004[C], 2004.

[4] IPEIROTIS P, GRAVANO L. When one sample is not enough: Improving text database selection using shrinkage[A]. Proceedings of the 2004 ACM International Conference on Management of Data[C], 2004

[5] LEDDLE S, EMBLEY D, SCOTT D, *et al.* Extracting data behind Web forms[A]. Proceedings of the Workshop on Conceptual Modeling Approaches for e-Business[C]. Tampere, Finland, 2002. 38-49.

[6] ARASU A, HECTOR GARCIA-MOLINA. Extracting Structured Data From Web Pages[A]. SIGMOD 2003[C], 2003.

[7] HE H, MENG W, YU C, *et al.* Wise-Integrator: An automatic integrator of Web search interfaces for e-commerce[A]. VLDB Conference[C], 2003.

[8] HE H, MENG WY, YU C, *et al.* WISE-Integrator: A System for Extracting and Integrating Complex Web Search Interfaces of the Deep Web[A]. International Conference on Very Large Data Bases (VLDB'05)[C]. Trondheim, Norway, 2005. 1314-1317.

[9] ZHANG Z, HE B, CHANG KCC. Understanding web query interfaces: Best-effort parsing with hidden syntax[A]. SIGMOD Conference[C], 2004.

[10] CHIDLOVSKII B, BERGHOLZ A. Crawling for domain-specific hidden web resources[A]. Proceedings of 4th International Conference on Web Information Systems Engineering[C], 2003.

[11] HE B, TAO T, CHANG K. Clustering Structured Web Sources: a Schema-based, Model-Differentiation Approach[A]. International Workshop on Clustering Information over the Web[C]. Crete, Greece, 2004.

[12] PENG Q, MENG WY, HE H, *et al.* WISE-Cluster: Clustering E-Commerce Search Engines Automatically[A]. 6th ACM International Workshop on Web Information and Data Management (WIDM 2004)[C]. Washington, DC, 2004. 104-111.

[13] CHANG KCC, HE B, ZHANG Z. Toward Large Scale Integration: Building a MetaQuerier over Databases on the Web[A]. Proceedings of the Second Conference on Innovative Data Systems Research (CIDR 2005)[C]. Asilomar, California, 2005.