# Deep Web 数据集成中基于最小超集的查询转换

姜芳艽 贾琳琳 孟小峰
(中国人民大学信息学院，北京，100872)
(jiangfj@gmail.com)

**摘 要** 近年来，随着 Web 上在线数据库的大量涌现，Deep Web 数据集成（即 web 数据库集成）成为当前信息领域的一个研究热点。查询转换是其中的核心部分，它主要负责将集成接口上的查询转换到相关 Web 数据库的接口上。由于 Web 数据库具有异构性和自治性的特点，各查询接口上的属性名，数据格式以及查询能力都不尽相同，因此相当一部分查询不能进行精确转换，那么选择何种策略进行近似查询转换是一个很具有挑战性的工作。本文对这一问题进行了深入探讨，提出了基于最小超集的近似查询转换方法。实验结果表明，该方法在 Deep Web 数据集成中可以有效地提高返回结果的准确性。

**关键词** Deep Web; Web 数据库; 近似查询转换; 最小超集

**中图法分类号** TP391

# Minimal-superset-based query translation in Deep Web data integration

Jiang Fangjiao, Jia Linlin, Meng Xiaofeng
(Information School, Renmin University of China, Beijing 100872)

**Abstract** Recently, online databases have grown rapidly, thus many researchers focus on Deep Web data integration. As a core component of this work, query translation is in charge of translating the queries from the integrated interface to related web database query interfaces. There exist differences in attribute name, data format and query capability of query interfaces due to the autonomy and heterogeneity of web databases. As a result, a considerable number of queries can not be translated precisely. So what strategy should be chosen to process query translation is a challenging task. In this paper we study the problem thoroughly and propose a method for query translation based on minimal superset. The intensive experiments on real web sites show that the proposed approach is quite promising for improving accuracy of query results.

**Key words** Deep Web; Web database; approximate query translation; minimal superset

随着World Wide Web的不断发展,网络上的在线数据库越来越多,这些在线数据库的信息不能有效地由传统的搜索引擎（Google等）通过静态链接直接得到[1]，而是需要通过在网站提供的查询页面上输入查询条件，提交给后台服务器，从底层数据库返回查询结果而得到。这些不能通过静态链接直接得到的,隐藏在Web数据库中的信息，称为Deep Web[2]。近年来的研究表明,大量重要的数据信息存在于Web数据库中。根据 2004 年 4 月的一份调查，Deep Web中大约包含了 450,000 个Web数据库以及 3,070,000 个页面，并且这些数字仍以指数级的速度继续增长。

为了有效地利用 Deep Web 中丰富的信息，越来越多的研究人员开始关注 Deep Web 数据集成的工作。而查询转换是 Deep Web 数据集成[2]的一个核心部分，它主要负责将集成接口上的查询转换到相关的 Web 数据库的查询接口上。

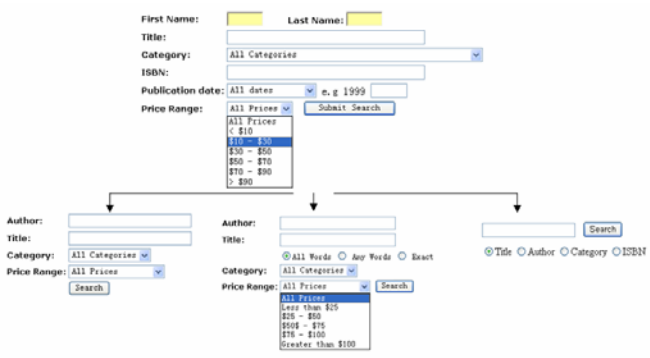


图 1 Deep Web 数据集成中查询转换的实例

收稿日期：2007-06-20
基金项目：本文得到国家自然科学基金项目(课题号：60573091)，北京市自然科学基金 (课题号：4073035)，教育部新世纪优秀人才支持计划的资助。

理想的情况是能够精确地转换查询，但由于 Web 数据库具有异构性和自治性的特点, 各查询接口上的属性名，数据格式以及查询能力都不尽相同，因此相当一部分查询不能进行精确转换，只能进行近似转换。如图 1 所示。根据我们的观察，超过 60%的查询由于 Web 数据库的自治性无法被准确地转换，因此近似查询转换是 Deep Web 数据集成中一个无法回避的重要环节。

近似转换的策略并不是单一的，但在近似转换时，一方面，由于不希望漏掉用户需要的信息, 往往选择放宽查询的条件；另一方面，如果查询条件放得太宽，会得到大量用户不需要的信息，这将加重对返回结果的过滤工作。因此寻求基于最小超集的查询转换（即包含所有结果且结果集合又是最小）将是一个非常合理的选择，如图 2 所示。基于最小超集的查询转换得到的查询结果（minimal superset）包含两部分，一部分是用户需要的信息（Correct results），另一部分是用户不需要的信息（False-positive），而且这部分信息比任何其他查询转换得到的查询结果都少。已有的相关工作主要关注于查询转换的子问题（即模式匹配）和基于静态规则的查询转换。而本文的工作，即在 Deep Web 数据集成中，用基于最小超集的思想进行查询转换，还未被涉及过。

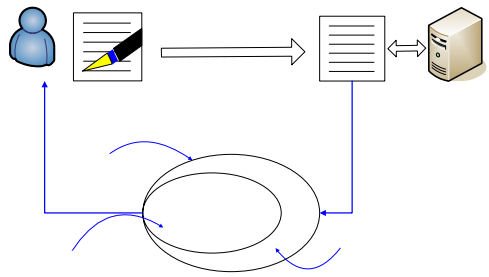

图 2 最小超集的查询转换示例

本文其余部分组织如下：第 1 节介绍查询转换的相关工作；第 2 节分析了基于最小超集查询转换中的三类问题；第 3 节针对不同问题提出如何找到基于最小超集查询转换的策略；第 4 节给出实验结果及分析；第 5 节对全文做了总结。

## 1. 相关工作

与查询转换相关的研究工作大致可以分为两类：属性的匹配和约束的映射。

属性的匹配是查询转换的必要前提，这方面的研究工作很多，Deep Web 数据集成中属性匹配的方法与传统的方法有所不同，有代表性的研究成果有[2，3，4，5，6，7]。[2]和[6]采用的是数据挖掘的方法，[2]将属性的匹配视作对属性相关性的挖掘，提出了一种新的判断相关性方法，H-measure，克服了已有的方法（例如, x 2、Lift、Confidence, Jaccard、Cosine）在处理稀疏模式数据和微小的负相关性的方面的不充分性。[4]则提出用聚类的方法发现属性之间的匹配关系。[3]和[5]分别采用整体和采样投票的统计的方法发现属性之间的匹配关系。[6，7]则通过 Web 数据库返回的结果进行属性的匹配。约束映射方面的有代表性相关工作是[8]和[9]，[8]提出在相互转换的查询接口之间用半自动的方法建立静态的匹配规则，根据这些规则进行查询转换，[9]则提出了基于数据类型的谓词映射方法。

现有的研究成果为在 Deep Web 数据集成环境中查询转换的研究打下了一定的基础，但由于 Web 数据库具有自治性和动态性等特点，大部分查询转换只能是近似转换，这方面的研究工作还处在探索阶段。

## 2 近似查询转换的问题描述

我们对 UIUC 的 ICQ 数据集[10]的 100 个查询接口的分析结果表明，超过 60%的查询只能近似转换。这些需要近似转换的约束条件可以分为三类。

例 1： 如图 1 所示，考虑一个在线购书网站的集成系统。在集成接口上用（first name，last name……）描述一本书。假设用户想要找一本“Jim Grey”编写的书。需要将查询 Q=C1^C2（其中 C1= [ First name = “Jim” ] ，C2= [ Last name = “Grey” ]）转换到一个支持 author 属性而不支持 first name 和 last name 属性的查询接口上。一种常用的策略就是分别查询“Jim”和“Grey”，然后取结果的并集。显然这样转换的结果有很多并不是用户需要的。而 Qt’ = [author = “Jim, Grey”]则是一个更好的转换。

因此查询转换并不是简单地把不同的约束条件分别转换。对于相互依赖的约束条件，必须同时考虑。

例 2：集成接口和 Web 数据库可能用不同的谓词表示相同的概念。例如，在图 1 中，Price 谓词[ price between $200 to $500 ]可能与 Web 数据库查询接口上的多个值的范围 ($151, $250), ($251, $350), ($351, $450), ($451, $550)重叠。

为了获得最小超集的查询转换，我们必须尽可能使转换后查询的搜索空间最小地覆盖原来查询的搜索空间。

例 3： 在 Web 数据库查询接口中，最主要的谓词逻辑是“AND”（即所有属性上的约束必须同时满足）。但是根据我们的观察有时接口中也会存在“EXCLUSIVE”逻辑，它表示每次只能提交一个属性上的约束条件。如图 1 所示，Web 数据库查询接口

只允许在属性 author，title，subject 和 ISBN 中选择一个属性提交约束条件（即每次只支持在 attribute，author，title，subject 和 ISBN 中的一个属性上的查询）。

我们的目标是选择一个使得查询转换后的返回结果最少的属性。

据我们统计，上述三类情况占所有近似查询 96.90%，其中，第一类情况是相互依赖的约束条件的转换问题，占到 13.95%，第二类情况是谓词的近似转换问题，占到 56.59%，第三类情况是基于能力的近似转换问题，占到 26.36%。很好地解决这些问题将有效地提高查询结果的准确性。

## 3 基于最小超集查询转换方法

**定义**（基于最小超集的查询转换）假设$Q_i$为某一领域 Web数据库的集成接口上的查询，查询$Q^*$是基于最小超集的查询转换，当且仅当同时满足以下条件：

1. $Q^*$是一个有效查询；
2. 对每一个数据库实例$D_j$，$Q^*$包含$Q_i$，即，$Q_i(D_j) \subseteq Q^*(D_j)$；
3. 不存在其他任何一个查询 Q**，使得 Q**满足条件 1 和 2，并且$Q^{**}(D_j) \subseteq Q^*(D_j)$。

针对第 2 节提到的三类情况，我们提出了三种相应的策略，使近似查询转换为最小超集的查询转换。

### 3.1 相互依赖的约束条件的转换方法

集成接口上通常包含了最重要的和最详细的属性，这些属性选自一系列的 Web 数据库查询接口并被组织成最易为用户所理解的形式，因此 M:1 的相互依赖约束条件的转换在查询转换中是非常普遍的。

我们将动态地维护一个概念层次结构以便帮助集成系统自动、正确地找到这些 M:1 条件匹配。然后根据领域知识将它们一起进行转换。这一工作在 Web 数据库查询接口变化或新加入一个 Web 数据库查询接口时显得尤为重要，因为接口集成过程中没有获得相关的属性匹配信息。

**定义（概念层次结构）**：一个概念层次结构是一棵包含了 N 个节点的有向树。树上的每一个节点描述一个概念，记 Node = (K, DT, {Si}, {Li})。它包含了以下信息：

**K** 描述这一概念的关键词

**DT** 这个概念所属的数据类型

**{Si}** 关键词 K 的若干同义、近义词

**{Li}** 指向其孩子结点的链接

假设$Attr_i$*表示Web数据库查询接口上的属性名，$Attr_j$表示概念层次结构中节点$N_j$的属性概念并且$Attr_{ju}$是节点$N_j$的关键词或近义词。如果$Attr_i$* 与 $Attr_j$的相似性超过给定的阈值，则$Attr_i$* 与$N_j$匹配。如果$N_j$是叶子节点，那么$Attr_j$ and $Attr_i$*是 1:1 的属性匹配，否则它们就是M:1 的属性匹配。见算法 1。

**算法1:**

**For** each $Attr_i$* **Do**
  **For** each node $N_j$ and not mapped **Do**
    **For** the keyword and every synonym $Attr_{ju}$ **Do**
      Compute $Sim_n(Attr_i*, Attr_{ju})$
    **End For**
    **Select** Max（Simn（$Attr_i$*, $Attr_{ju}$））
    $Sim_n(Attr_i*, Attr_j) = Max(Sim_n(Attr_i*, Attr_{ju}))$
    $Sim(Attr_i*, Attr_j) = w_t*Sim_t(Attr_i*, Attrj) + (w_n*Sim_n(Attr_i*, Attr_j) \mid w_v*Sim_v(Attr_i*, Attr_j)) + w_p*Sim_p(Attr_i*, Attr_j)$
  **End For**
  **Select** Max（Sim（$Attr_i$*，$Attr_j$））
  **If** Max ( $Sim(Attr_i*, Attr_j)$) > threshold **Then**
    $Attr_i$* is mapped to $N_j$
    insert $Attr_i$* into the correspondent synonym list of $N_j$
    **If** $N_j$ is leave node **Then**
      $Attr_j$ and $Attr_i$* is 1:1 attribute mapping
    **Else** $Attr_j$ and $Attr_i$* is m:1 attribute mapping
    **End If**
  **Else** $Attr_i$* is not mapped to any node
  **End If**
**End For**

算法中两个属性相似性：

$$Sim(Attr_i^*, Attr_i) = w_t \times (Sim_t(Attr_i^*, Attr_i)) + (w_n \times Sim_n(Attr_i^*, Attr_i) \mid w_v \times (Sim_v(Attr_i^*, Attr_i)) + w_p \times Sim_p(Sim_p(Attr_i^*, Attr_i))$$

$Sim_t$, $Sim_n$, $Sim_v$, $Sim_p$分别代表数据类型，属性名，属性值和当前节点的父结点的相似性。

算法中$Sim_n$的计算是基于字符串的编辑距离：

$$Sim_n = 1 - \frac{dist(Attr_i^*, Attr_j)}{Max((len(Attr_i^*), len(Attr_j))}$$

### 3.2 谓词的转换方法

寻找谓词转换中的最小超集就必须考虑整个搜索空间，然后选出能够包含原始的搜索空间的最小的那一个。我们用$\Omega_0$，$\Omega_i$ 和$\Omega^*$分别表示原始搜索空间、可能的搜索空间和最小超集的搜索空间。

#### 3.2.1 数字或时间类型的谓词转换方法

对于数字类型的属性，由于数字具有有序性，它的可能的搜索空间包括所有有效的组合（必须是连续的区间的组合），因此可能的搜索空间的个数为：

$$N = \sum_{i=1}^{num}(num - i + 1)$$

其中*num*是Web数据库查询接口属性范围列表中区间范围的个数，通过比较$\Omega_0$和每一个$\Omega_i$我们可以比较容易地找到$\Omega^*$。表 1 展示了寻找$\Omega^*$的过程。

显然时间类型可以参照数字类型的方法解决。

表 1 数字类型的谓词转换的实例

| | predicates | The range of Ω |
|---|---|---|
| Initial search space $\Omega_0$ | P0:Price between 10 and 35 | (10,35) |
| Possible search space $\Omega_i$ | P1:price between 0 and 20<br>P2: price between 20 and 40<br>…<br>P10: price between 180 and 200<br>P11: price between 0 and 40<br>P12: price between 20 and 60<br>…<br>P55: price between 0 and 200 | Ω1: (0,20)<br>Ω2: (20,40)<br>…<br>Ω10: (180,200)<br>Ω11: (0,40)<br>Ω12: (20,60)<br>…<br>Ω55: (0,200) |
| minimal superset search space $\Omega^*$ | P11: price between 0 and 40 | Ω*: (0,40) |

### 3.2.2 文本类型的谓词转换方法

对于文本类型，很难找到两个查询 Q1 和 Q2 的包含关系。例如，图 1 中，属性 title 的谓词为[title contains “java programming”]，那么可能的转换为[title any “java programming”]，[title all “java programming”]和[title exact “java programming”]等。那么哪一个才是最小超集的查询转换呢？与 4.1.1 中数字类型谓词转换情况所不同的是，其输入条件的情况是无限的（因为是一个文本输入框），我们无法枚举所有的情况，但我们可以用“封闭—世界”（“close-world”）假设缩减搜索空间[9]。不失一般性，我们考虑谓词[title contains “A B”]，假设 X 是非 A 且非 B 的词，那么查询转换后可能的搜索空间 Ωi 的数量为：

$$N = (C_2^1 + C_2^2) \times 3$$

其中数字 2 代表文本中的词汇个数，数字 3 是指 3 种绑定的约束条件（即，any、all、exact）。由于它的无序性，可能搜索空间的范围包含了所有的组合，而且通过绑定限定条件的语义来计算每一个可能搜索空间的大小（size of Ω）。寻找$\Omega^*$时先根据可能搜索空间的大小筛选出与$\Omega_0$大小接近的可能搜索空间，再比较它们的搜索空间的范围，找到最小覆盖的Ωi即为$\Omega^*$。表 2 展示了寻找$\Omega^*$的过程。

表 2 文本类型的谓词转换的实例

| | Predicates | The range of search space Ω | The size of Ω |
|---|---|---|---|
| Initial search space $\Omega_0$: | P0:Title contains AB | $\Omega_0$:AB,BA,ABX,AXB,XAB,XBA,BAX,BXA | P32+P22 |
| Possible search space $\Omega_i$: | P1:Title any A<br>P2:Title any B<br>P3:Title any AB<br><br>P4:Title all A<br>P5:Title all B<br>P6:Title all AB<br>P7:Title exact A<br>P8:Title exact B<br>P9:Title exact AB | Ω1:A,AB,AX,BA,XA,ABX,AXB,XBA,XAB,BAX,BXA<br>Ω2:B,BA,BX,AB,XB,BAX,BXA,XAB,XBA,ABX,AXB<br>Ω3:A,AB,AX,BA,XA,ABX,AXB,XBA,XAB,BAX,BXA<br>B,BA,BX,AB,XB,BAX,BXA,XAB,XBA,ABX,AXB<br>Ω4:A,AB,AX,BA,XA,ABX,AXB,XBA,XAB,BAX,BXA<br>Ω5:B,BA,BX,AB,XB,BAX,BXA,XAB,XBA,ABX,AXB<br>Ω6:AB,BA,ABX,AXB,XAB,XBA,BAX,BXA<br>Ω7:A<br>Ω8:B<br>Ω9:AB | P31+ P32+P22<br>P31+ P32+P22<br>2*(P31+P32+P22)<br>P31+ P32+P22<br>P31+ P32+P22<br>P32+P22<br>1<br>1<br>1 |
| minimal superset search space $\Omega^*$ | Title all AB | Ω6:AB,BA,ABX,AXB,XAB,XBA,BAX,BXA | P31+ P32+P22 |

### 3.3 基于能力的转换方法

导致不同查询能力的一个重要原因就是 Web 数据库查询接口与集成接口对查询属性组合的限制不同。例如，某个 Web 数据库只支持每次在 author，title，subject 或 ISBN 之一的属性上的查询，而集成接口上可以同时在这些属性上进行查询。我们希望选取一个最贴近原始查询的谓词而不是取多个不同的谓词，然后取它们结果的交（即 $A \cap B$），原因是这样做的效率要比选择（即，$\sigma_B A$)高。

根据以上的分析，基于能力的查询转换的挑战性在于如何选取最严格的谓词使得查询结果最少。换句话说，我们必须知道对于某个具体领域每个属性查询能力的不同。获得不同属性查询能力的过程分为四步。首先，在用户常用的属性（属性author要比ISBN常用）Ai上随机提交一个查询$Q_{Ai}$给Web数据库，数据库返回初始结果集RS($Q_{Ai}$)，这些结果将作为种子用于下一步；第二，从RS ($Q_{Ai}$)中随机选取属性Aj (j<>i)的m个不同的值，并将它们提交给web数据库查询接口，返回相应的hit number与结果集RS ($Q_{Aj}$)。我们可以计算出m个在属性Aj上的查询的平均hit number $AN_{Aj}$ (j<>i)；第三，从RS ($Q_{Aj}$)中随机选取属性Ai的m个不同的值，这里要注意属性Aj和属性Ai必须是无关的（例如author与publisher），同样我们可以得到属性Ai上的平均hit number $AN_{Ai}$；最后，将$AN_{Ai}$ (i=0 to n)按升序排列，那么$N_{Ai}$,越小的属性它的查询能力越强。

## 4 实验

为了验证本文方法的有效性，实验中我们采用了UIUC的ICQ数据集[10]，这一数据集共有 100 个样本，涉及 5 个领域，分别是机票、汽车、书籍、工作和房

地产，其中每个领域各有 20 个具有web数据库查询接口的网站样本。我们对这 100 个网站进行统计分析。表 3，表 4 分别显示对查询接口和属性进行的分析。其中N1 列、A1 列分别表示每个领域下的Web数据库查询接口数和累计属性数，N2 列、A2 列分别表示不能够精确转换的查询接口数和累计属性数，P列表示它们占总数的百分比。

表 3 不能精确转换的查询接口

| domain | N1 | N2 | P |
|---|---|---|---|
| airfare | 20 | 13 | 65% |
| auto | 20 | 5 | 25% |
| book | 20 | 18 | 90% |
| job | 20 | 9 | 45% |
| real estate | 20 | 15 | 75% |
| *total* | 100 | 60 | 60% |

表 4 不能精确转换的属性

| domain | A1 | A2 | P |
|---|---|---|---|
| airfare | 238 | 32 | 13.45% |
| auto | 92 | 7 | 7.61% |
| book | 173 | 46 | 26.59% |
| job | 111 | 13 | 11.71% |
| real estate | 119 | 31 | 26.05% |
| total | 733 | 129 | 17.60% |

总体上，在五个领域中有 60%的查询接口无法精确转换，而属性则有 17.05%无法进行精确转换。

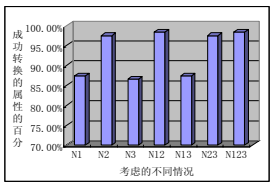


图 3 机票

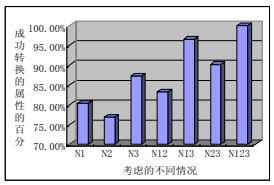


图 4 书籍

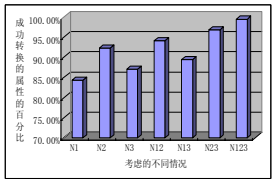


图 5 五个领域总的情况

图 3 和图 4 展示了我们所提出的方法在机票和书两个领域中实施的情况。每幅图中 N1、N2 和 N3 分别表示考虑第一、第二和第三类情况时，基于最小超集查询转换的程度。N13 、N23 和 N123 表示同时考虑第一类和第三类情况、第二类和第三类情况以及表示全面考虑三种情况后的结果。在机票领域，大部分的查询接口都提供在时间上的查询，在属性的选择列表中，它的每一项可能是整点（如 12：00am）也可能是时间范围（如 morning 或 6：00－10：00），这属于我们需要考虑的第二种情况，因此 N2、N12、N23 和 N123 的准确率较高；我们再看书籍这个领域（图 4），由于查询能力约束条件 or 出现的情况比较多，因此考虑第三种情况的 N3、N13、N23 和 N123 准确率较高。五个领域的统计结果（图 5）表明，同时考虑三类情况后，在数据集上能够保证 99.45%的属性的查询转换。可见我们的实验在这 100 个 web 数据库上达到了相当好的结果。

## 5 结论

在 Deep Web 数据集成中，由于 Web 数据库具有异构性和自治性的特点，大部分查询不能从集成接口精确转换到 web 数据库的查询接口，只能进行近似转换。我们提出了基于最小超集的查询转换方法，使查询转换得到的查询结果既包含用户需要的信息又最少地包含用户不需要的信息。实验结果表明，该方法能够在近似查询转换中有效地提高返回结果的准确性。

## 参 考 文 献

[1] K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang. Structured databases on the web: Observations and implications. SIGMOD Record, 2004, 33(3): 61~70.

[2] Bin He, Kevin Chen-Chuan Chang, Jiawei Han. Discovering Complex Matchings across Web Query Interfaces: A Correlation Mining Approach. KDD 2004, 148-157.

[3] Bin He, Kevin Chen-Chuan Chang: Making holistic schema matching robust: an ensemble approach. KDD 2005, 429-438.

[4] W. Wu, C. T. Yu, A. Doan, and W. Meng. An interactive clustering-based approach to integrating source query interfaces on the deep web. SIGMOD 2004, 95 -106.

[5] B. He and K. C.-C. Chang. Statistical schema matching across web query interfaces. SIGMOD 2003, 217-228.

[6] Jiying Wang, Ji-Rong Wen, Fred Lochovsky, Wei-Ying Ma.Instance-based Schema Matching for Web Databases by Domain-specific Query Probing. VLDB 2004, 408-419.

[7] Wensheng Wu, Clement Yu. WebIQ: Learning from the Web to Match Deep-Web Query Interfaces. ICDE 2006, 44.

[8] K. C.-C. Chang, H. Garc´a-Molina : Mind Your Vocabulary: Query Mapping Across Heterogeneous Information Sources. SIGMOD Conference 1999, 335-346.

[9] Z. Zhang, B. He, and K. C.-C. Chang. Light-weight domain-based form assistant: Querying Web Databases On the Fly, *VLDB* 2005, 97-108.

[10] http://metaquerier.cs.uiuc.edu/repository/datasets/icq/browsable.html

**姜芳芃**，女，1971 年生，博士研究生，研究方向：Web 数据管理。

**贾琳琳**，女，1984 年生，硕士研究生，研究方向：Web 数据管理。

**孟小峰**，男，1964 年生，教授（博导），研究方向：Web数据管理，XML数据库，移动数据管理。