Deep Web 数据集成研究综述

刘 伟[1)] 孟小峰[1)] 孟卫一[2)]
[1)] (中国人民大学信息学院 北京 100872)
[2)] (纽约州立大学计算机科学系 宾汉姆顿 13902)

摘 要 随着 World Wide Web(WWW)的飞速发展,Deep Web 中蕴含了海量的可供访问的信息,并且还在迅速地增长.这些信息要通过查询接口在线访问其后端的 Web 数据库.尽管丰富的信息蕴藏在 Deep Web 中,由于 Deep Web 数据的异构性和动态性,有效地把这些信息加以利用是一件十分挑战性的工作.Deep Web 数据集成至今仍然是一个新兴的研究领域,其中包含有若干需要解决的问题.总体来看,在该领域已经开展了大量的研究工作,但各个方面发展并不均衡.文中提出了一个 Deep Web 数据集成的系统架构,依据这个系统架构对 Deep Web 数据集成领域中若干关键研究问题的现状进行了回顾总结,并对未来的研究发展方向作了较为深入的探讨分析.

关键词 World Wide Web; Deep Web; Web 数据库;查询接口; Deep Web 数据集成
中图法分类号 TP311

A Survey of Deep Web Data Integration

LIU Wei[1)] MENG Xiao-Feng[1)] MENG Wei-Yi[2)]
[1)] (*School of Information, Renmin University of China, Beijing* 100872)
[2)] (*Department of Computer Science, State University of New York, Binghamton* 13902)

Abstract As the rapid development of World Wide Web, there is tremendous information "hidened" in Deep Web, and its capacity is increasing rapidly. The information can only be accessed by the query interfaces provided by Web database. The data in Deep Web are obtained in the form of dynamic Web pages when users send a query. Due to the poor structure of Web pages and the instability and large scale of Deep Web, it is a very challenging task to integrate the abundant information automatically and use it effectively. Until now, Deep Web data integration has still been a rising research field, and there are a number of challenging issues in it. A great deal of research works is developed in this field, but it is imbalanced on the issues of this field. A framework of Deep Web data integration is proposed in this paper, and the key research works in Deep Web data integration are classified and surveyed according to this framework. At last, the deficiencies in this field are analyzed and the suggestions for future research works are put forward.

Keywords World Wide Web; Deep Web; Web database; query interface; Deep Web data integration

1 引 言

随着 World Wide Web 的飞速发展,其中蕴含了海量的信息可供我们利用.根据文献[1]最新的调查,目前整个 Web 超过了 200000TB 的信息量,而且仍在快速地增长.在 Web 领域的研究目的在于发展新的技术可以有效地从 Web 中获取有用的信息.

收稿日期:2006-05-07;最终修改稿收到日期:2007-06-17.本课题得到国家"八六三"高技术研究发展计划项目"基于 Web Service 的 Web 数据集成技术"(2002AA11304)、国家自然科学基金项目"Web 数据抽取和集成技术研究"(60273018)和国家"九七三"重点基础研究发展规划项目"语义网格的基础理论、模型和关键技术研究"子课题"基于本体的数据管理研究"(2003CB317000)资助.刘 伟,男,1976 年生,博士研究生,研究方向为 Web 数据管理与集成.E-mail: gue2@ruc.edu.cn.孟小峰,男,1964 年生,教授,博士生导师,主要研究领域为 Web 数据管理、XML 数据库、移动数据管理等.孟卫一,男,1958 年生,教授,博士生导师,主要研究领域为信息检索、元搜索引擎、Web 数据库集成等.

Web 中的信息主要通过网页的形式对外发布, 而由文本和超链接构成的网页有其独特之处: 数量惊人, 信息丰富; 由不同的个人或群体开发, 形式与内容有很大的差异; 分布在地球上 Internet 连接的每一个角落, 这就造成了 Web 数据的异质性和缺乏结构性. 正是由于这个原因, 使得自动地从中获取有价值的信息和数据变成一件十分具有挑战性的任务. 到目前为止, 为有效地利用 Web 上的信息而采用的方法涉及了广泛的领域: 数据挖掘、机器学习、自然语言处理、统计分析、数据库和信息检索等.

整个 Web 看似杂乱无章, 但如果按其所蕴涵信息的"深度"可以划分为 Surface Web 和 Deep Web 两大部分. Surface Web 是指通过超链接可以被传统搜索引擎索引到的页面的集合. Deep Web 是指 Web 中不能被传统的搜索引擎索引到的那部分内容. 广义上来说, Deep Web 的内容主要包含 4 个方面: (1) 通过填写表单形成对后台在线数据库的查询而得到的动态页面; (2) 由于缺乏被指向的超链接而没有被搜索引擎索引到的页面, 大约占整个比例的 21.3%; (3) 需要注册或其它限制才能访问的内容; (4) Web 上可访问的非网页文件, 比如图片文件、PDF 和 Word 文档等.

而在实际中应用中, 人们则更关注于 Deep Web 中的第一部分内容. 其原因不难理解. 这部分内容对结构化数据的集成更有意义, 可以采用的技术也更丰富. Deep Web 数据集成也主要是指对结构化信息的集成. 我们同时把 Web 中可访问的在线数据库称为 Web 数据库或 WDB. 这些内容只有在被查询时才会由 Web 服务器动态生成页面, 把结果返回给访问者(图 1), 因此没有超链接指向这些页面, 这是和那些可以被直接访问的静态页面的根本区别. 随着 Web 相关技术的日益成熟和 Deep Web 所蕴含信息量的快速增长, 通过对 Web 数据库的访问逐渐成为获取信息的主要手段, 而对 Deep Web 的研究也越来越受到人们的关注.

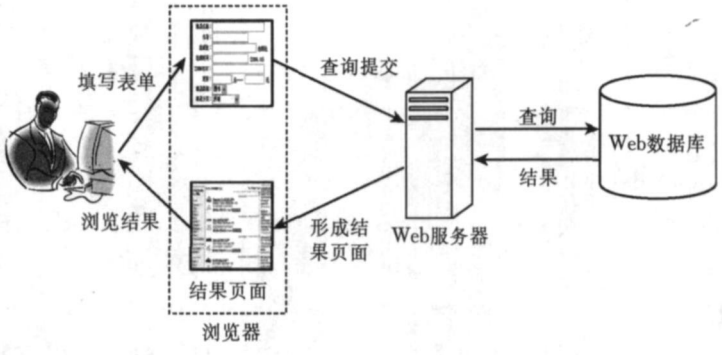

图 1 从 Web 数据库中获取数据的过程

与 Surface Web 相比, Deep Web 蕴藏了更加丰富, 更加"专业"(专注于某一领域) 的信息. 在 2000 年 7 月, Brightplanet 对 Deep Web 做了一次较为全面的宏观统计, 发布了 Deep Web 的白皮书①(在该文中 Brightplanet 对 Deep Web 的定义主要指的是 Web 数据库), 指出整个 Web 上大约有 43000~96000 个 Web 数据库, 并从宏观上对 Deep Web 做了定量的调查统计, 下面列出其中部分的调查结果:

(1) Deep Web 蕴含的信息量是 Surface Web 的 400~500 倍.

(2) 对 Deep Web 数据的访问量比 Surface Web 要高出 15%.

(3) Deep Web 蕴含的信息量比 Surface Web 的质量更高.

(4) Deep Web 的增长速度要远大于 Surface Web.

(5) 超过 50% 的 Deep Web 的内容是特定于某个域的, 即面向某个领域.

(6) 整个 Deep Web 覆盖了现实世界中的各个领域, 比如商业、教育、政府等等.

(7) Deep Web 上 95% 的信息是可以公开访问的, 即免费获取.

整个 Web 是开放的、不断变化的, 有效地评估当前整个 Deep Web 的规模, 即当前 Deep Web 上 Web 数据库的数量以及变化情况是十分重要的. UIUC 大学在 2004 年 4 月对整个 Deep Web 做了一次较为准确的估算[2], 推测整个 Web 上有 307000 个提供 Web 数据库的网站、450000 个 Web 数据库, 比 Brightplanet 在 2000 年估计的 50000 个数据库网站的数目增长了 6 倍多.

① http://www.brightplanet.com/technology/DeepWeb.asp

Deep Web 中的 Web 数据库不但数量众多, 而且覆盖了现实世界的各个领域. 一些专门的机构, 像 CompletePlanet 和 InvisibleWeb 等, 构建了 Deep Web 目录, 按现实世界的领域对 Deep Web 的内容做了分类, 主要包括商业与经济、计算机与互联网、新闻媒体、娱乐等一共十几个分类. 这只是宏观的分类, 每个分类下面还有小的分类, 比如科学可以继续分为社会科学与自然科学, 而自然科学又可分为若干学科. 在表 1[2] 中可以看出, 尽管这些网站对 Web 数据库进行了细致的分类, 但所列出的 Web 数据库仅仅只是整个 Web 数据库的很小的一个比例(即使最大的 CompletePlanet 也只有 15. 6%). 因此从宏观上对 Web 数据库按现实世界的领域分类做一个定量的分析是十分迫切而且必要的工作.

表 1　**Deep Web** 目录的覆盖率

	Web 数据库的数目	覆盖率/%
completeplanet. com	70000	15. 6
lii. org	14000	3. 1
turbo10. com	2300	0. 5
invisible-web. net	1000	0. 2

对 Deep Web 中信息的获取主要的途径是通过对网站中所提供的查询接口提交查询来获得, 图 2 是 Amazon 网站提供的查询接口. 每个查询接口支持在若干个属性上进行查询, 比如要查询某一本图书, 可以根据书名、作者、价格等. 这些属性就构成了查询接口的模式(Schema)信息. 查询接口模式的大小是指属性的数目. 查询接口顾名思义是外部访问 Web 数据库的门户, 是从 Web 数据库中获取数据的主要途径, 因此在 Web 数据库研究领域, 对查询接口的模式信息的研究占有极其重要的地位.

对 Deep Web 信息的访问是通过在查询接口上提交查询, 这和对搜索引擎的访问在某种程度上来说是相似的, 但 Deep Web 数据和搜索引擎二者之间是有着很大区别的:

(1) 搜索引擎搜索结果是网页, 而 Deep Web 中的搜索结果主要是结构化的数据.

(2) Web 数据库通常有复杂的接口, 而搜索引擎的接口较为简单, 一般是关键字搜索.

(3) 搜索引擎对结果的排序是根据搜索结果与所提交查询的相似性, Web 数据库则是根据结果中某个属性的值.

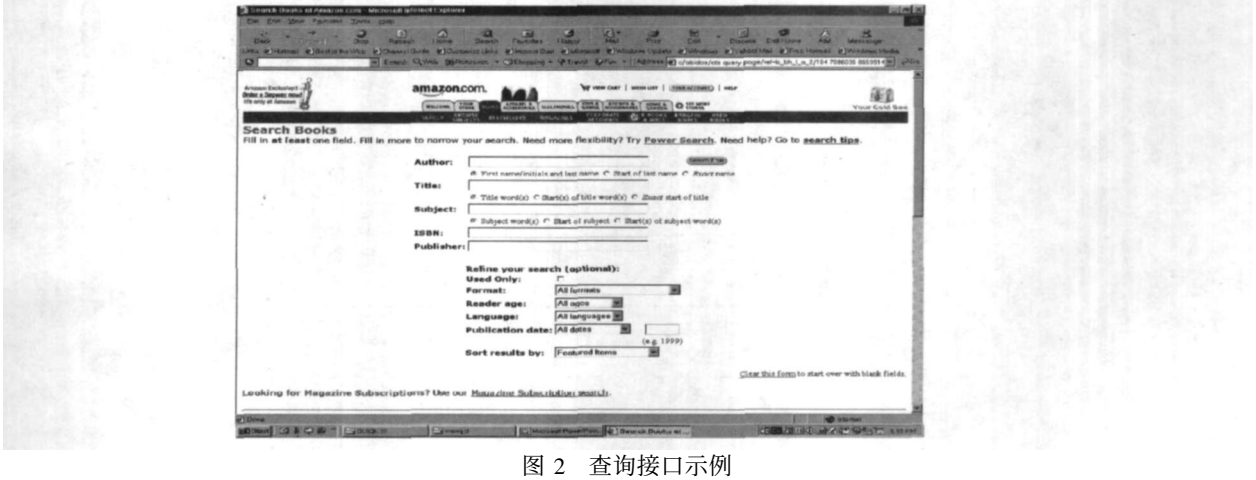

图 2　查询接口示例

2　Deep Web 数据集成框架

自从 21 世纪以来, 随着 Internet 飞速的发展和软硬件技术的日益成熟, 从 Web 中自动获取有用的信息不再只是设想, Deep Web 也受到越来越多的研究者的关注, 并且越来越多的相关研究成果发表在相关的高级别会议和期刊上. 对 Deep Web 研究的根本目的是为了能够帮助用户提供一个统一的访问途径来自动地获取利用自由分布在整个 Web 上的 Deep Web 中丰富的信息.

虽然整个 Deep Web 中几乎包含了大量我们所需要的信息, 但要想以手工的方式对其加以有效的利用在实际当中是一件非常困难的事情, 而对 Deep Web 数据的集成正是为了以尽可能自动的方式来完成对 Web 数据库中信息的有效利用. 在 Deep Web 数据集成领域存在着许多的研究问题, 已有的工作主要集中这些问题上: Web 数据库的发现、查询接口模式的抽取、Web 数据库的分类、查询接口的集成、查询的转换、查询结果的抽取、查询结果的注释等. 有些问题已经得到了较多的研究, 而有些问题还处在研究的初步阶段甚至还没有相关的报道. 为了给出一个全面的认识, 我们提出了 Deep Web 数据集成框架, 该框架共分为三个主要的模块: 查询

接口集成模块、查询处理模块和查询结果处理模块. 每个模块又分为若干子模块, 可以看作具体的研究问题并分别完成特定的功能.

集成查询接口生成模块. 为用户提供一个统一的查询接口, 使之可以同时向多个统一领域内的查询接口提交查询, 即达到同时访问属于同一领域的多个 Web 数据库的目的. 该部分共有 4 个主要的子模块: Web 数据库的发现、查询接口模式的抽取、基于领域 Web 数据库的分类和查询接口集成. Web 数据库的发现是指从 Web 中发现具有一个真正 Web 数据库的网站, 然后从中发现可访问这个 Web 数据库的查询接口. 查询接口模式的抽取是对前一步获得的查询接口中所包含的属性进行分析和抽取, 获得一个查询接口的模式信息. Web 数据库的分类是指根据已得到的查询接口的模式信息确定其对应 Web 数据库所属的领域, 即按照领域对 Web 数据库进行分类. 查询接口的集成是对属于同一个领域的查询接口进行集成, 得到一个全局的查询接口.

查询处理模块. 将用户在集成的查询接口上填写的查询转化到对各个 Web 数据库本地查询接口的查询. 该部分包含 3 个子模块: Web 数据库的选择、查询转换和查询提交. Web 数据库的选择是指为一个给定的用户查询从所有集成的 Web 数据库中选择合适的进行查询. 查询转换是指将用户在集成查询接口上提交的查询转换到 Web 数据库本地的查询. 查询提交是指自动地将转换后的查询进行提交.

查询结果处理模块. 将各个 Web 数据库返回的结果抽取并合并到一个统一的结构化的模式下. 该部分包括结果的抽取、结果的注释和结果的合并. 查询结果的抽取是指从 Web 数据库返回的结果页面中抽取出真正的查询结果. 结果的注释是指由于抽取的结果通常缺少语义, 因此要为缺少语义的数据项进行语义注释. 查询结果的合并是指把从各个 Web 数据库得到的查询结果进行有效的合并去重, 存储在一个统一的模式下.

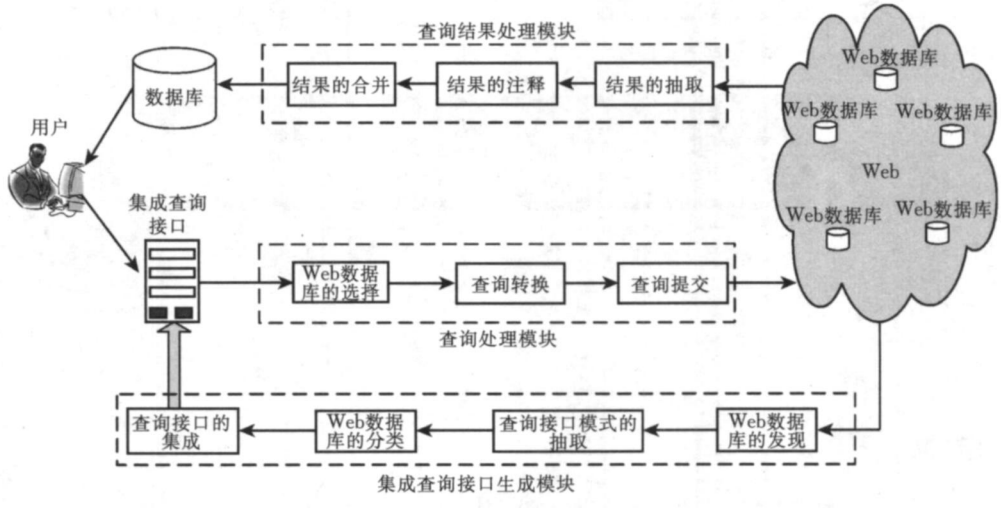

图 3 Deep Web 数据集成框架

下面我们将依据该框架对目前已有的相关工作进行简要的介绍.

3 集成查询接口的生成

为了能够同时访问多个 Web 数据库中的数据, 在 Deep Web 数据集成系统中必须要提供一个统一的访问途经. 每个 Web 数据库都提供了查询接口, 我们需要把每个 Web 数据库的查询接口进行集成并得到一个统一的接口, 该接口称为集成接口. 通过在集成接口上提交查询, 就达到了同时在多个 Web 数据库的查询接口提交查询的目的. 为了得到这个集成接口, 需要经历 4 个主要的步骤. 首先要在 Web 上发现要集成的查询接口; 其次对这些接口进行解析, 获得它们的模式信息, 即查询能力; 第三要把它们按不同的领域分类; 第四是把属于同一个领域的接口集成为一个统一的接口.

3.1 Web 数据库的发现

Web 数据库的发现是指在 Web 中发现可访问的 Web 数据库, 完成这个功能主要分为两个步骤: (1) 找到 Web 数据库所在的网站; (2) 从获得的网站中发现能够对 Web 数据库查询的查询接口. 比较全面而准确的把 Web 数据库从 Web 中搜索出来是一件非常困难而又耗时的事情, 其原因有三: 首先由

于目前 Web 中存在大约 450000 个可访问的 Web 数据库, 这些自主的、相互独立的 Web 数据库分布在整个 Web 的各个角落, 虽然对 Web 数据库做了搜集与整理, 但从表 1 中可以看到只覆盖了全部 Web 数据库的很少一部分; 其次 Web 是动态的、不断变化的, Web 数据库也是如此, 不断有新的产生和旧的消失, 即使现存的 Web 数据库内容和规模也处于不断变化之中; 第三, 查询接口在网页上都是以 Html 语言的 Form 元素所形成的表单的形式展现, 但并不是说由 Form 元素所形成的表单都是查询接口, 比如网站中用户的注册、BBS 讨论组、写发邮件, 还有搜索引擎和元搜索引擎也都是表单的表现形式, 要能够从中准确地识别出真正的 Web 数据库的查询接口.

对于第一步目前的解决途经有三种. 第一种是从completeplanet. com 和 invisible-web.net 这样的网站中获取, 虽然不能找到所有的 Web 数据库, 但这些 Web 数据库都已按领域作了分类, 对于小规模的集成仍然是一个有效的方案; 第二种是遍历 Web 中所有 IP[2], 这种方案在理论上可以把所有的 Web 数据库完整地找出来, 但目前大约有 22 亿 3 千万个有效的 IP, 逐个遍历显然代价过高, 因此只能作为一种研究统计手段, 比如估计整个 Web 上 Web 数据库的规模、Web 数据库在各个领域上比例分布等等; 第三种是利用搜索引擎进行搜索, 虽然搜索引擎不能获取 Web 数据库中的内容, 但可以用来找到 Web 数据库所在网站, 由于必须提交向搜索引擎提交查询, 因此这种方案是基于某个领域的 Web 数据库的发现, 也更加具有实际应用意义. 其关键在于如何向搜索引擎提交有效的查询, 使得含有 Web 数据库的网站尽可能多地出现在查询结果中, 并使其排名尽量靠前.

第二步是从网站中找到可以向 Web 数据库提交查询的查询接口. 由于查询接口和搜索引擎、元搜索引擎以及用户注册等都是以 Html 语言的 Form 元素表示, 因此有两个问题需要解决: 首先, 通常一个网站包含上千甚至更多的页面, 遍历所有页面找出显然代价太大; 其次需要从所有 Form 元素中将查询接口准确的区分出来. 在文献[2] 中 Chang 等通过大量的观察提出了一个巧妙的办法来来解决这个问题, 即从网站的主页开始以宽度优先遍历所形成的树, 查询接口在这棵树中的深度不会超过 5, 而且 94% 的查询接口不会超过 3, 这样搜索空间就会大大降低. 而对于第二个问题, 文献[3] 基于查询接口的特征利用 C4. 5 决策树实现了对查询接口的识别, 其中主要分为两个步骤, 首先是查询接口特征的产生; 其次是在这些可以作为判断依据的特征之上利用 C4. 5 算法得到一棵决策树, 通过这棵决策树找出真正的查询接口. 利用查询接口的特征作为判断依据是一种直观有效的解决途径, 实验结果表明: 从 Web 中随机查询的数据集准确性只达到了 87%, 还有很大提升空间. 在文献[3] 中提出了一个判断页面中是否含有查询接口的一个简单方法. 该方法共有 3 个简单的规则: 首先页面中要有 Form 标签; 其次 Form 标签中必须有 Text 输入控件; 第三, 至少出现一组关键词中的一个, 像“查询”、“搜索”等等. 这种方法在其实验中可以达到至少 93% 的准确性. 但这些方法还有一些不完善的地方, 首先它们还不能把代表 Web 数据库的查询接口与搜索引擎区的查询接口分开来, 这就需要进一步总结这二者之间可区分的特征; 另外该工作只是根据 Form 表单在页面中的源代码总结查询接口的特征, 其实还有很多的特征可以利用, 比如查询接口在页面中的视觉布局信息、所在页面的频繁词汇信息等.

3. 2　查询接口模式的抽取

查询接口的模式是一组领域相关的属性集合, 通过对其中若干属性的赋值形成一个对该查询接口所代表 Web 数据库的查询. 对查询接口模式的抽取可以获得一个查询接口的查询能力. 查询接口的模式可以被看作是建立在对应 Web 数据库上的一个视图.

对查询接口模式的抽取是指对查询接口属性的获取与分析. 对查询接口模式的抽取主要目的是为了下一步的 Web 数据库分类和查询接口集成, 其关键是把查询接口所包含的各个属性准确地抽取出来.

文献[4] 以文法分析的方式来完成对查询接口模式的抽取. 对于整个页面结构的分析已经有了较为细致的工作, 如文献[5-7] . 针对查询接口结构的分析, 该工作属于开创性的. 这种方法首先通过观察与统计提出这样一个假设: 所有查询接口都是由隐藏的文法构建而成. 为了能够准确地从一个具体的查询接口中将表示属性的各个元素组合方式识别出来, 该工作通过构建解析树对整个查询接口进行解释, 确定它们的语义角色, 并利用优先次序解决分组

方式之间存在着冲突的可能性,这样就把查询接口中的属性尽可能地发现出来了. 其 *precision* 为 80%, *recall* 为 89%, 显然还不能完全达到实际应用的程度.

完成属性抽取后,需要把查询接口形式化地表现出来以便于为下一步的工作提出模型化的解决方案. 查询接口形式化的表达方式与应用目的相关,如果是为了对 Web 数据库分类,关注的是查询接口整体的信息,即可以查询哪个域的信息;如果是为了查询接口集成,则是关注查询接口内各个属性的细节信息,即找到不同查询接口之间属性的最佳的匹配关系. 最直观的方法是将查询接口看作是一个属性的集合. 文献[8]提出了一种较为完备的形式化表达方式:首先整个查询接口表示一个三元组,包括查询接口所在网站的相关信息、属性的集合,由属性形成查询条件之间的关系,比如连接、非连接、排斥等. 属性集合是对每个属性信息的描述,每个属性表示为一个七元组,包括属性的名称、属性在查询接口中的布局位置、属性的域类型、属性的缺省值、属性的值的类型、属性值的单位. 可以看出,包含了查询接口所有有关的细节信息,可以对下一步 Web 数据库的分类和查询接口的集成提供足够的信息.

3.3 Web 数据库的分类

根据文献[2]在 2004 年的估计,整个 Web 中大约有 450000 个可访问的 Web 数据库,而且数目还在快速地增长. 为了有效地利用这些 Web 数据库中的信息,需要将其按领域进行分类. 如果手工地来完成对所有 Web 数据库的分类是个庞大而费时的工程,因此需要以尽可能自动的方式来完成对 Web 数据库的分类. 由于对 Web 数据库按领域进行分类才有实际的应用意义,因此目前所提出的分类方法也都基于领域的. 在查询接口上提交查询是获取 Web 数据库信息的主要途径,对 Web 数据库的分类实质上是对查询接口的分类.

分类方法共分为两类:指导方式和非指导方式. 文献[9]针对应用意义最广泛的电子商务的 Web 数据库提出了一种有效的分类方法. 这种方法是一种非指导的方式,主要利用了电子商务的 Web 数据库的查询接口所在页面上的可用特征信息,包括接口中出现的频繁词和商品的价格特征. 其实验结果表明,按这种分类方式进行分类, *precision* 和 *recall* 都在 90% 左右. 文献[10]完全利用查询接口的模式信息提出了一种更一般的 Web 数据库分类解决方案,属于指导方式. 他们根据统计特性认为查询接口的模式信息可以作为对 Web 数据库分类的依据. 基于这样的统计结论,他们提出通过建立概率模型来表示所有可能出现的属性在每个领域中出现的可能性. 对于一个给定的查询接口,考察其属性集合,在这个模型上计算出这个查询接口与每个领域的相似性. 前面两种方法都是基于查询接口的特征信息实现对 Web 数据库的分类,另外还提出了两种利用提交样本查询来实现分类的方法. 文献[11]从返回查询结果数量来分析一个 Web 数据库属于哪个领域,文献[12]则从分析返回文本的内容来确定一个 Web 数据库的领域. 这两个工作针对的不是结构化信息,而是文本信息,但其通过查询进行分类的思想可以为 Web 数据库的分类所借鉴.

3.4 查询接口的集成

查询接口的集成是为了给用户提供一个对属于同一个领域的 Web 数据库统一的访问途径,而对 Web 数据库的访问方式主要是通过查询接口,因此对 Web 数据库集成重要的一步就是查询接口的集成. 集成的查询接口合并了同一领域的查询接口集合中表示同一语义的属性,保留了一些查询接口中特定的属性,并尽可能地保持该领域查询接口的结构特征和属性的顺序性. 如果把各个被集成的查询接口看作 Web 数据库的一个本地视图的话,那么集成的查询接口就是建立在这些本地视图之上的全局视图.

通过属性进行分析是查询接口集成最主要的途径,至今已经有了许多的工作,如文献[11,13-16]. 这种方式主要发掘给定查询接口的模式信息和语义信息,利用这些语义信息来识别不同查询接口上属性之间的匹配关系,在这些具体的查询接口之上获得一个集成的查询接口,达到同时访问多个 Web 数据库的目的. 模式匹配与集成[17-21]是实现这一方式以及后面进行数据合并的一个关键技术,但主要是对已有技术的应用,所以不作过多叙述. 目前对查询接口的集成主要是手工的方式,这样虽然可以达到比较高的准确性,但是在大规模集成查询接口的情况下效率很难得到保证,因此需要以自动的方式来完成这个集成的过程. 对查询接口自动集成的实现方式上可分为两大类:一类属于局部方式,是基于给定的要进行集成的查询接口集合,分析属性的隐含信息,特别是语义信息,在它们之间作属性的匹配,得到一个新的全局接口;另一类属于整体方式,是基于某个确定的领域通过对这个领域范围内大量接口的处理,发现这个领域上一般的查询接口.

局部集成方式. Wise-Integrator[8] 是对电子商务进行数据集成的一个系统, 接口的集成是该系统其中的一个重要组成部分. 它是一个综合的解决方案, 首先对每个查询接口进行分析, 获取其中的属性信息. 在语义分析的过程中用到了一个很重要的工具 Wordnet①. 然后就是属性匹配, 在完成对所有查询接口的属性匹配后, 要为匹配的属性在集成的查询接口上确定它的全局名称和它的类型和取值范围, 这样就得到了一个集成的查询接口. 在实验中从正确性和完整性两个方面来衡量集成的质量, 这两项的实验结果分别为 95. 25% 和 97. 91%. 该工作总的来说实现了接口的集成, 但也存在着不足: 首先把查询接口看作是一个平的结构, 实际上查询接口具有很丰富的结构信息; 其次是只考虑了查询接口之间属性 1∶1 的映射情况, 但现实中的查询接口存在着大量的复杂映射. 针对这些不足, 文献[13] 对查询接口的集成提出了较大的扩展与改进: 首先, 把查询接口的模式看作有层次的树状结构; 其次, 通过"搭桥"的方式对查询接口的属性实现更准确地匹配聚类; 由于复杂的 1∶m 映射频繁的出现, 针对这种情况, 对复杂映射划分为 aggregate 和 is-a 两种类型; 让用户参与到集成过程中来, 对集成过程加以指导. 实验将完全自动和用户参与两种方式进行了对比, 完全自动的方式平均 *precision* 和 *recall* 分别为 88. 2% 和 91. 1%, 而通过用户者的参与两项指标分别提高了 7. 8% 和 2. 9%.

整体集成方式. 与上面通过属性分析来发现两个查询接口之间属性对匹配方式不同, 文献[22] 提出了利用统计模式匹配的方案. 这种方式认为 Deep Web 中同一域的数据源隐藏着一个共同的模式模型, 这个模型可以刻画该域的所有查询接口共同的特征. 基于这个共同的模式模型可以整体的方式匹配同一个域的所有模式. 基于这个思想, 一个一般的模式匹配框架 MGS 被提出, 包括假猜模型化、假猜生成和假猜选择. 假猜模型化这一步是定义一个模型, 这个模型是针对特定问题的. 假猜生成是对所定义模型的参数的设定. 由于会产生多个可能的模型, 假猜选择则是从这些可能的模型中选择出合适的模型. 文献[23-24] 提出了查询接口属性相关性的观点: 所有属性可分为正相关、负相关和相互独立三类. 对于如何判断两个属性是正相关、负相关还是相互独立的问题, 提出了自己判断标准 H-measure. 在确定了不同查询接口之间属性对的关系之后要从中选择最合适的匹配. 其实验结果表明对查询接口中 1∶1 的匹配可以全部准确地发现, 而对 m∶n 的匹配可以发现全部的负相关匹配, 正相关匹配在测试数据集中仅有错误的一例.

4　查询的处理

当用户在集成查询接口上填写并提交查询时, 要同时从多个 Web 数据库中获取符合该查询的结果, 并把这些异构的数据以统一的模式存储或展现, 这就是对 Deep Web 数据查询的处理. 为了能达到这个目的, 需要完成若干步骤. 首先能够为用户选择合适的 Web 数据库, 其次把查询近似等价地转化成在这些具体 Web 数据库查询接口上的查询, 然后是从返回的结果页面中抽取查询结果并添加语义注释, 最后将这些结果合并在一起. 下面对这些方面做逐一介绍.

4.1　Web 数据库的选择

当完成对一个领域的查询接口的集成后, Web 数据库集成系统的用户在集成的查询接口上提交查询, 这样从属于这个领域的 Web 数据库中获得所需的信息. 在对 Web 数据库按领域进行分类后, 每一个领域中 Web 数据库的数量仍然十分巨大. 以商业和教育这两个领域为例, 根据 CompletePlanet 的统计, 都存在上千个 Web 数据库, 由于 CompletePlanet 只是发现了整个 Deep Web 中大约 7% 的 Web 数据库, 所以在现实中还要远远大于 1000 这个数字.

当在集成的查询接口上对某一个领域进行查询时, 如果只是简单地把集成接口上的查询转换成对该领域每个 Web 数据库的查询, 并不是一个可行的方案, 原因是: 首先, 由于一个领域中存在大量的可访问 Web 数据库, 虽然在理论上来说可以获得足够丰富的查询结果, 但因访问大量的 Web 数据库, 在 Internet 上花费的代价难以承受的; 其次, 并不是每一个 Web 数据库都能够满足一个特定的查询, 显然任何一个领域的 Web 数据库不可能包含这个领域中所有的信息, 因此也不可能满足这个领域的任意查询; 第三, 一个领域中大部分的 Web 数据库之间存在着冗余的信息, 因此对一个查询而言, 访问的 Web 数据库越多, 返回信息的冗余度也会越大, 使得冗余信息的处理难度大大增加. 因此, 在 Web 数据库的选择这一步要达到的目标是如何从一个领域

① http://www.cogsci.princeton.edu

大量的 Web 数据库中选择出合适的部分, 使得在满足一个特定查询的前提下尽可能地减少所访问的 Web 数据库的数量和使得查询结果中冗余度足够小.

对一个特定的查询, 为了能够知道每个可访问的 Web 数据库对这个查询的满足程度, 即每个可访问的 Web 数据库中符合该查询的信息量, 要事先获取 Web 数据库的有用特征. 由于 Web 数据库分为结构化和非结构化两类, 结构化的 Web 数据库的特征是指其模式中各个属性上值的分布特征, 而非结构化的 Web 数据库主要是指文本数据库, 对文本数据库的查询主要使用了信息检索的技术, 因此其特征是指所存储的文档集合与域相关的关键词的相似性关系. 而对于搜索引擎的选择已有了许多较为成熟的工作, 如文献[25-27], 其中一些技术思想可以借鉴到对结构化的 Web 数据库选择的实现中. 目前对 Web 数据库特征的获取唯一途径是通过对提交查询而得到的查询结果进行分析, 非结构化的 Web 数据库主要关注一个特定查询返回结果的数量, 而结构化的 Web 数据库除了返回结果的数量外更主要是关注各个属性上值的分布特征.

由于结构化的 Web 数据库中存储的是由若干属性组成的现实世界的实体, 对结构化的 Web 数据库选择除了根据其大小是根据各个属性上特征表现, 现在主要是在数字属性(价格、日期等)上利用直方图的方法进行特征概括. 为了获得某个属性上值分布的特征, 显然获取的该属性值越多越能够得到与实际相一致的特征. 因此要以尽可能少的查询来获得尽可能多的结果并使得查询结果能够均匀的分布在整个 Web 数据库中, 这就需要设计具有代表性的查询, 既要与 Web 数据库的领域紧密相关, 又要能够近似反映出当前数据库中的信息在各个属性之上的分布. 另外, 对 Web 数据库提交一次特定的查询往往会返回较多的查询结果, 而大部分的用户并不是关注查询的全部结果, 只需要前 N 位的结果就可以满足他们的查询需求了. 因此, 在集成各个 Web 数据库的查询结果的同时, 能快速地得到最符合查询的 N 个结果是非常有应用意义的. 数据分散在各个 Web 数据库之中, 我们需要的前 N 个结果可能只是在某几个 Web 数据库的结果中. 如果可以只向这一小部分 Web 数据库提交查询, 就可以降低计算代价. 文献[28]提出了一种基于直方图的 Top-N 的选择方法. 该方法分为两步: 第一步是判断数据库与特定查询之间的相关性; 第二步是确定最适合提交查询的数据库和从返回的结果中选择最合适的记录. 算法实验表明, 作者这种计算 Top-N 查询的方法是非常有效的.

4.2 查询转换

由于 Web 数据库的自治性, 使得用户的查询受限于其给定的查询接口的表达能力, 这也造成了集成查询接口和 Web 数据库本地查询接口之间无法进行等价的查询转换, 进而造成了所返回的查询结果包含不满足用户查询的记录. 而查询转换的目的就是为了通过查询近似的转换将不满足的记录减少到最少, 即用户所应得到查询结果的最小超集.

Light-weight domain-based form assistant[29] 是一个可以处理同一领域内任意两个 Web 查询接口之间转换的框架. 它在属性、谓词和查询 3 个层次上实现查询的转换. 在属性层次上的转换实质上是模式匹配, 指将两个查询接口上表示同一语义的两个属性进行匹配. 在谓词层次上的转换是该工作的核心, 是为了解决不同查询接口上同一语义属性由于属性值的差异所带来的查询转换上的问题. 比如, 源查询接口上价格属性给出了 0~25、26~50 和 50 以上 3 个取值选择, 而目标查询接口在该属性上给出的取值是 0~30、31~60 和 60 以上. 该工作首先通过对 3 个领域 150 个 Web 数据库的查询接口进行观察统计, 从中得到谓词模板之间的映射关系, 由于任何属性都属于文本、数字或时间日期中的数据类型之一, 因此它把所有的谓词模板的映射都遵循着数据类型的约束来进行, 这样就大大提高了映射的准确性. 其实验结果表明, 该方法无论在基本测试集还是随机测试集上, 对属性 1∶1 和 m∶n 的匹配上进行查询转换都可以达到较高的准确性.

5 查询结果的处理

查询结果的处理是为了把从各个 Web 数据库返回的表现形式不同的结果在一个统一的模式下展现给用户. 目前主要的工作集中在如何从查询结果页面抽取出结构化的查询结果.

5.1 查询结果的抽取

Web 数据库返回的查询结果主要是通过 Html 语言编写的页面来展现的, 而 Html 语言的特点是在 Web 上发布的, 内容多样, 形式各异, 使得 Web 上的数据半结构甚至是无结构, 给 Web 数据库集成系统的建立造成了极大的困难. 从页面中将查询结

果抽取出来的过程是指将 Web 页面上半结构和无结构的数据通过各种技术手段抽取出来, 保存为可以自动处理的 XML 文档或关系模式, 作为下一步处理的基础.

目前普遍的 Web 数据抽取方式是编写特定的抽取程序, 主要具备两个功能: 搜寻、发现并抽取特定的数据; 以适当的格式保存数据供进一步处理, 比如 XML 和关系模式. 其中最大的挑战是如何从页面上大量的数据中完整准确地发现查询结果. 当把 Web 数据库中的信息以 Html 页面的表现形式展现时, 数据库相关模式结构信息就完全丢失了. 对页面抽取的一个主要目的就是通过把信息以结构化的格式存储来反转这个过程.

目前这个研究领域已经开展了大量的研究工作, 有了很多 Web 数据抽取的工具, 按使用的技术大致可以分为几类, 下面分别作简要介绍.

页面抽取语言. 它是指特定设计的语言, 帮助使用者实现抽取过程. 抽取是用手工的方法编写程序来实现的. 抽取过程是基于过程化的程序, 但是, 抽取结果依赖于文档的结构. 这方面主要的工作有 Minerva[30]、Web-OQL[31]. Minerva 是 Araneus[32] 系统的一个重要组成部分, 它结合了基于语法的声明方式和典型的过程化语言. Minerva 使用的语法以 EBNF 定义: 对每个文档, 定义生成式的集合; 每个生成式根据终结符和其它非终结符定义一个语法的非终结符的结构. Web-OQL 是一种陈述性的查询语言, 能够定位在 Html 页面上所选择的数据快. 为了达到这种目的, 包装器将页面解析抽象, 用获得的语法树 hypertree 来表示页面. 通过这种语言, 可以写查询, 在语法树上定位感兴趣的数据并以合适的格式输出这些数据.

基于 **DOM** 树的工具. 其依赖于 Html 页面的内在的结构特征, 在抽取之前将页面转化成 DOM 树, 以反映页面标签的层次结构, 然后自动或半自动地抽取规则在此树上应用. 主要的工作有 XWRAP[33]、RoadRunner[34-35]、Lixto[36-37]、MDR[38] 和 MDRII[39]. XWRAP 有一个组件库提供抽取规则生成的基本模块, 这个工具引导用户通过一系列的步骤, 选择每一步中正确的组件. 最后, XWRAP 输出特定源上的一个抽取规则. 在对象抽取这步中, 为 Html 页面预定义了 6 个启发式, 用户可以使用其中的启发式定位感兴趣的数据对象. 用户也可以为了使抽取结果更符合自己的要求限制或放宽每个对象的组件数目或指定数据类型. RoadRunner 其方法是进一步发掘 Html 文档内在的特征来自动产生抽取规则. 通过比较样本页面得到一个结果模式, 从这个模式可以推测出一个能够识别出样本页面中的实例. 为了准确地捕获在样本页面所有可能的结构变量, 必须提供多于两个的样本页面. 所有的抽取过程都基于这样一个算法: 比较样本页面的标签结构产生规则的表达式来处理结构之间不匹配的情况. 过程完全自动化是 RoadRunner 独一无二的特性. 它可以说是第一个完全自动的抽取工具, 具有里程碑的意义. 但它对模式的推导时间复杂性是指数量级, 因此在大量样本页面的情况下代价过高. MDR 和 MDRII 这两种抽取方法都是由美国 Illinois 大学同一研究小组提出, 其独特的地方在于能够十分准确地在 DOM tree 中完成对多记录页面的抽取. 它们的实现关键在于利用页面的嵌套结构和表现特征把查询结果从整个页面中分离出来, 并将结果中的多个记录彼此精确地划分, 其意义是把每个记录作为现实世界的实体对待, 首先从这个角度完成第一步抽取, 第二步把每一条记录从属性的角度进行分解. MDR 把标签树中节点的路径看作一个字符串, 并使用了比较字符串编辑距离的思想从数据区中发现代表数据记录的结点, 而 MDRII 则是以树的结构信息代替标签字符串, 从而达到对数据记录更准确的识别结果. 对于结果页面中记录的界定在文献[40]中早已提出, 随着对页面结构和布局的不断认识, 这种方式被重新加以发展深化.

抽取规则推导工具. 其是从给定的训练样本中产生基于分隔符的抽取规则, 更适合 Html 文档, 但需要大量的样本页面. 主要的工作有 WIEN[41]、STALKER[42]. WIEN 是归纳工具类中的先驱, 它将已经标好感兴趣数据的页面作为样本输入, 与每个样本一致的抽取规则作为输出. 这些页面有预定义的结构和特定启发式, 用来产生特定的抽取规则, 但不能处理嵌套结构和典型半结构化数据的变量. STALKER 能处理层次数据的抽取, 输入是: 以一系列包含被抽取数据的符号为形式的训练样本; 页面结构的描述, 称为 ECT. STALKER 产生一个抽取规则尽量覆盖给定的样本. 如果存在未被覆盖的样本, 它产生一个新的分离的规则. 当所有正例被覆盖后, STALKER 返回一个规则的集合. 使用 ECT, STALKER 能处理嵌套层次的对象.

基于模式的工具. 为感兴趣的对象给定一个目

标结构,尽量使页面上的数据部分符合这个结构,通过图形界面与用户交互,由用户指出页面上感兴趣的区域.由于需要和用户交互,从自动化程度上来讲属于半自动抽取工具.主要的工作是 NoDoSE[43-44]、DEByE[45] 和 SG-WRAP[46]. NoDoSE 是一个半自动化交互的工具,使用图形化的用户接口、用户层次的分解文档,划出感兴趣的区域并描述它们的语义. DEByE 是一种交互工具,把简单页面的样本对象集合作为输入,产生能够从其它类似页面抽取新对象的抽取模式. SG-WRAP 这种方法是一种预定义模式引导的数据抽取方式,通过图形化的界面把在样本页面中要抽取的数据与预定义的模式进行连接匹配,通过这种操作产生抽取规则,完成对同类页面的有效抽取.

其它方法. 抽取过程的实现还有很多方法. 有的是针对页面中特定的能够结构化表现数据的标签,如文献[47-48],显然这种方法有着很大的局限性,应用范围窄,所以这里不作过多的介绍. 值得注意的是,页面中的视觉信息越来越受到研究者们的注意,目前已经有了相当的工作利用视觉信息对页面进行分析[49-50],这里有一个重要的原因:网页被设计出来的目的是为了方便人们浏览从中获取有用的信息,而不是被计算机自动处理,因而获取页面的视觉信息可以从某种程度上模拟人类的行为对页面信息的识别. 文献[51-52]在利用视觉信息对页面分块的基础上进行了 Web 搜索和链接方面的研究,而利用视觉信息在 Web 数据库查询结果抽取方面目前是作为一种有用的辅助手段. 文献[39]在由页面形成的 DOM 树中为元素添加了在浏览器中的位置信息,并认为每个节点在视觉上占据了一个矩形的区域,而且父节点所占据的矩形区域包含子节点占据的区域,通过节点的位置和大小信息可以准确地发现在 DOM 树中不连续的数据记录,而这种情况对以往只利用页面的源码作抽取的 Wrapper 来说是无法解决的. 文献[53]是针对搜索引擎的查询结果而提出的工作,它把视觉信息和 DOM 树结构结合起来发现和分离查询结果.

从前面可以看到,到目前为止已经有了如此多的抽取工具,并按照实现技术进行了分类,如何评价抽取工具的性能,可以从下面几个角度来看待. (1) 准确性. 这是最为重要的标准,可以借用信息检索的两个主要概念准确率(Precision)和召回率(Recall)来衡量:准确率在这里指抽取到的正确结果与抽取到的全部结果的比;召回率在这里指抽取到的正确结果与要抽取页面的全部结果的比. (2) 自动化的程度. 这是另一个比较重要的标准,关系到在抽取的过程中使用者参与的程度. 这也是对 Web 抽取工具的另一个分类方式,即手工、半自动和完全自动. 目前完全自动的抽取方法已经完全取代了手工和半自动的方式成为主要的趋势. (3) 弹性和适应性. 由于 Web 页面的内容和结构经常发生变化,抽取工具要有自适应的能力,即当页面结构发生较小的变化时也能继续正常工作,这称为弹性. 一个抽取工具为某个特定领域的页面而生成,如果它也能为这个领域另一个数据源的页面工作,这称为适应性. 这对于高度动态的 Web 而言尤为重要. 使用的方便程度,提供图形化界面使抽取规则的生成更加容易. 这主要是针对半自动的方式而言. 另外大部分抽取工具都或多或少地需要调整参数,参数过多或过于复杂也会使其可用性降低.

Web 数据抽取是 Web 数据库集成系统中发展最为成熟的部分. 我们对 Web 数据抽取工具进行了分类和总结,分类的方法主要根据技术实现的角度,可以看出涉及了各种各样的方法,而且随着 Web 的发展,新的方法会不断地出现. 作为 Web 数据集成的重要一环,在这个领域还远没有达到令人满意的程度,尤其是在准确性上. 语义 Web 的提出就是为了使计算机能够对页面中的数据进行自动处理,不过在目前看来要做到全面替代传统的 Html 页面还有很长的路要走.

6 未来工作的展望

随着 Web 数据库在 Web 中不断大量的涌现,对 Web 数据库进行大规模集成的研究成为一个非常迫切的问题. 至今,人们在 Deep Web 领域已经作了大量的研究,所提出的 Deep Web 数据集成系统有文献[8,54],但它们只是属于研究性的原型系统,因此确切地说至今还没有一个真正可以作为实际应用的 Deep Web 数据集成系统. 前面,对这些工作按照 Deep Web 数据集成系统的框架进行了分类和概括总结,然而大部分工作仍然处于探索性的阶段,只有查询接口的集成和查询结果的抽取这两个方面的工作相对成熟,有些方面的工作到目前可以说是刚刚开始甚至仍然是空白. 因此要实现一个真正可用的集成系统仍然有许多的问题有待更深入的研究.

下面就 Deep Web 数据集成系统框架中仍然需要开展的工作做初步的展望.

Web 数据库的发现. 利用成熟的传统搜索引擎完成对 Web 数据库的搜索是一种行之有效的办法. 由于查询接口存在于静态的页面中, 因此可以被传统的搜索引擎抓取到. 如果能够借助搜索引擎强大的搜索能力, 那么就大大降低了搜索代价. 这种方法虽然是可行的, 但也包含了挑战性的工作. 搜索引擎的作用是搜索 Web 中的页面, 获取页面唯一途径是提交关键词查询, 而包含 Web 数据库查询接口的页面只占全部页面很小的比例, 如果提交的关键词不合理, 会导致搜索到的页面结果集中所包含的查询接口比例太小, 使得不仅每次获得的 Web 数据库数量少, 而且也会使筛选的代价过高. 因此设计合理的关键词查询是利用搜索引擎获取 Web 数据库的关键问题. 由于 Web 数据库的查询接口在页面中以 Form 表单形式表现, 但 Form 表单还可以有很多种用途, 搜索引擎和元搜索引擎的查询接口在页面中的表现形式与 Web 数据库的查询接口更加相似, 如何把 Web 数据库查询接口从中准确地发现出来至今仍未达得到很好的解决. 为了能够准确地判断一个页面中的 Form 表单元素是否是一个真正的 Web 数据库的查询接口, 有两个十分有用的方法还未加以利用: 首先, 页面中一般包含有比较丰富的语义信息, 通过这些语义信息可以用来帮助我们判断一个 Form 表单元素的用途; 其次, 通过提交试探性查询, 根据返回结果的数量来判断, 比如判断一个 Form 表单是不是一个图书信息的查询接口, 可以提交"Thinking in Java", 如果有包含该书的查询结果信息, 则说明此 Form 表单极有可能是一个图书信息的查询接口, 甚至可以进一步判断为一个计算机图书信息的查询接口.

Web 数据库的分类. 已有的工作总地说仍未把 Web 数据库的分类问题彻底解决, 其根本原因是只是利用了查询接口自身及所在页面所提供的信息, 当属性信息非常类似时就会无法区分. 另外有些领域的 Web 数据库为了方便用户的查询, 提供了极其简单的查询接口, 如音乐和图书领域, 经常只需填写关键字, 使得仅依赖查询接口的模式信息很难判断出这个接口属于哪个领域. 为了解决这些情况, 可以从两个方面考虑. 首先根据领域之间的不同特征要能够实时调整相似性判断函数里的判断标准, 并可以多阶段的执行分类过程. 其次通过在查询接口上提交与领域相关的查询, 根据返回结果进行分类, 这是直接判断一个 Web 数据库属于哪个领域的最有效方法. 以汽车、音乐和图书 3 个领域的分类为例, 如果提交"Thinking in Java" 的查询, 前两个领域的 Web 数据库将不会返回任何结果, 而图书领域的 Web 数据库则可能返回若干结果. 提交样本查询也是 Web 数据库发现的一种有效方法, 进一步说, 如果能够设计一个合适的领域相关的样本查询集合, 就可以把 Web 数据库发现和分类两个步骤合并在一起, 叫做基于领域的 Web 数据库的发现, 这样不仅保证了更高的准确性和效率, 而且更具有实际应用意义.

Web 数据库的选择. 由于 Web 数据库数量的不断增长使得 Web 数据库的选择成为一个亟待解决的问题. 为了能够降低对 Web 数据库的访问代价和获得高质量的数据, 需要在同一个领域中选取合适的 Web 数据库进行查询, 不可避免要对这些 Web 数据库进行特征概括, 通过这些特征概括来判断一个 Web 数据库是否与给定的查询相关程度. 目前已有的工作主要是针对搜索引擎和 Web 数据库中非结构化的文本数据库的, 而对于比例最大的结构化 Web 数据库而言, 现有的工作是在数字属性(如价格、日期等)和离散属性(如有限种选择的属性)上进行特征概括, 虽然对 Web 数据库的选择起到了一定的作用, 但还未从根本上解决问题. 因此下一步的研究工作要能够对非数字的不可穷举属性进行有效的特征概括, 这就要提出不同的方法来处理这类属性. 随着本体和语义领域理论的不断成熟, 可以借助于建立一个特定域的本体来对一个 Web 数据库进行特征概括, 建立一个概念的层次树结构, 最低层节点是属于父节点概念的实例集合, 这样通过实例查询可以估计每层的每个分类在一个 Web 数据库中所拥有的信息比例, 从而能够更好地刻画 Web 数据库在这个属性上的特征总结. 另外如果把 Web 数据库中所有的数据全部通过查询的方式获取出来, 虽然可以对其做最好的特征概括, 但考虑到对 Web 数据库的访问代价和 Web 数据库内容频繁的更新, 就失去了其应有的意义. 因此需要设计有效的查询获得 Web 数据库中的部分数据, 使得这些部分的数据能够具有对全部数据的代表性, 通过这种"以偏概全"的方式来降低对 Web 数据库特征概括的代价. 通过在查询接口上提交试探查询的方法不仅对 Web 数据库的选择, 而且对 Web 数据库的发

现和分类也有极其重要的意义,因此如何对一个特定的 Web 数据库构建高质量的查询自动生成器是将来迫切需要解决的问题.

对查询结果的语义注释. 为了使从页面中抽取到的数据具有使用价值,必须要为其添加语义注释,而目前在这方面的工作还在初步阶段,都是以启发式规则的方式对抽取到的数据进行语义注释[55-56],不仅准确性还未达到实际应用的标准,而且更重要的是不能对抽取到的全部数据添加语义注释. 为了能够把从各个 Web 数据库的数据有效地集成起来并加以利用,要对这两个方面做较大的改进. 对于抽取数据的自动语义添加,如果是针对一个特定的 Web 数据库,可以通过机器学习的方式预先在一组样本页面上训练形成一个自动添加语义的程序,学习出数据与对应语义之间的关系,从而能够处理新的页面. 考虑是在 Web 数据库集成系统的环境下,有的 Web 数据库的查询接口或结果中能够得到某个属性数据的语义,而有的 Web 数据库对这个属性的数据则没有语义注释,如果能够对各个 Web 数据库的模式之间建立匹配关系,利用预先建立的模式匹配关系就可以以互补的方式达到对数据语义的添加,但要保证语义的正确性,前提是要保证这种模式匹配关系的正确性,由于页面结构化程度很差,目前还很难保证在页面中模式匹配较高的正确性.

Web 数据的合并. 在 Web 数据库集成系统中,最终要把从各个 Web 数据库获得的数据合并到一个统一的模式下. 在实际中,各个 Web 数据库的数据经常存在大量的重复,因此在合并过程中必须要解决的一个重要问题是数据的去重. 由于这些数据描述的是现实世界的实体,如果能从实体的角度把重复的数据加以识别将会有效地达到去重的目的. 实体识别的问题其实普遍存在于对多个数据源集成的领域中,以前的关于对实体的识别主要还是采用手工的办法,根据特定的应用领域和使用者的要求来定制特定的规则,为了达到较高的准确性,不可避免地要求大量人工的参与,而且只能适用于特定的领域. 目前实体识别只是在关系模式和半结构化的 XML 模式上开展[57-58]. 在实体识别的问题中有两个关键的子问题:建立实体之间属性的映射关系和属性之间值的比较. 实体之间属性的映射即模式匹配,由于 Web 页面结构化程度很差,传统的模式匹配方法难以直接应用,对于大规模的集成手工的方式更加不可取,因此需要提出新的方法来解决 Web 环境下的属性匹配问题;属性之间值的比较则首先选取能够代表实体的属性,然后在这些代表性的属性上作值的比较. 由于各个 Web 数据库的异质性,要从语义角度来判断属性值对之间是否表达统一语义. 在 Web 环境下至今还没有真正意义上的实体识别的工作存在,但这又是 Web 数据库集成系统中不可缺少的一个关键环节,这将是未来最为迫切的问题之一.

Web 数据的增量维护. Web 数据库的数据经常处于频繁更新的状态,而用户总是希望能够得到当前 Web 数据库中最新的内容. 在多数据源集成的研究领域,对集成数据的增量维护是一个无法避免的问题. 同样,对 Web 集成数据增量维护问题的重要性将随着 Web 数据库集成系统的不断成熟显得日益突出. 由于 Web 数据处于快速动态更新的状态,而且 Web 页面模板也频繁地发生变化,使得增量维护变得更加复杂,需要提出新的方法来自动检测增量的 Web 数据并适应 Web 页面模板的变化.

7 结束语

随着 Web 数据库数量和其蕴含数据量飞速的增长,对 Deep Web 数据的集成越来越成为研究领域关注的问题,目前人们已经在这个方面做了大量的工作,本文对最近几年来国际上在该领域的主要研究成果进行了回顾与总结,综述了 Deep Web 数据集成系统中若干主要问题的研究现状,包括 Web 数据库的发现、Web 数据库模式的抽取、Web 数据库的分类、查询接口的集成、Web 数据库的选择、结果数据的抽取等等,并在综述的同时指出仍然存在的问题和将来可能的解决办法. 总的来说对 Deep Web 数据集成的研究仍然处于刚刚起步的阶段,离应用阶段还有很长的路要走,仍然有大量关键的问题还需要做深入细致的研究.

参 考 文 献

[1] Fetterly D, Manasse M, Najork M, Wiener J L. A large-scale study of the evolution of Web pages// Proceedings of the 12th International World Wide Web Conference. Budapest, 2003: 669-678

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

[3] Cope J, Craswell N, Hawking D. Automated discovery of search interfaces on the Web//Proceedings of the 14th Australasian Database Conference(ADC 2003). Adelaide, 2003: 181-189

[4] Zhang Z, He B, Chang K C. Understanding Web query interfaces: Best-effort parsing with hidden syntax//Proceedings of the 23rd ACM SIGMOD International Conference on Management of Data. Paris, 2004: 107-118

[5] Arasu A, Garcia-Molina H. Extracting structured data from Web pages//Proceedings of the 22nd ACM SIGMOD International Conference on Management of Data. San Diego, 2003: 337-348

[6] Crescenzi V, Mecca G, Merialdo P. RoadRunner: Towards automatic data extraction from large Web sites//Proceedings of the 27th International Conference on Very Large Data Bases. Italy, 2001: 109-118

[7] Wittenburg K, Weitzman L. Visual grammars and incremental parsing for interface languages//Proceedings of the IEEE Symposium on Visual Languages (VL). Skokie, 1990: 111-118

[8] He H, Meng W, Yu C T, Wu Z. WISE-integrator: An automatic integrator of Web search interfaces for e-commerce//Proceedings of the 29th International Conference on Very Large Data Bases. Berlin, 2003: 357-368

[9] Peng Q, Meng W, He H, Yu C T. WISE-cluster: Clustering e-commerce search engines automatically//Proceedings of the 6th ACM International Workshop on Web Information and Data Management. Washington, 2004: 104-111

[10] He B, Tao T, Chang K C. Clustering structured Web sources: A schema-based, model-differentiation approach//Proceedings of the 9th International Conference on Extending Database Technology. Heraklion, Crete, 2004: 536-546

[11] Ipeirotis P G, Gravano L, Sahami M. Probe, count, and classify: Categorizing hidden Web databases//Proceedings of the 19th ACM SIGMOD International Conference on Management of Data. Santa Barbara, 2001: 67-78

[12] Meng W, Wang W, Sun H, Yu C. Concept hierarchy based text database categorization. Knowledge and Information Systems Journal, 2002, 4(2): 132-150

[13] Wu W, Yu C T, Doan A, Meng W. An interactive clustering-based approach to integrating source query interfaces on the Deep Web//Proceedings of the 23rd ACM SIGMOD International Conference on Management of Data. Paris, 2004: 95-106

[14] He H I, Meng W, Yu C T, Wu Z. Constructing interface schemas for search interfaces of Web databases//Proceedings of the 6th International Conference on Web Information Systems Engineering. New York, 2005: 29-42

[15] He H, Meng W, Yu C T, Wu Z. Automatic integration of Web search interfaces with WISE-integrator. VLDB Journal, 2004, 13(3): 256-273

[16] Wu Z, Raghavan V, Du C, Sai K C, Meng W, He H, Yu C T. SE-LEGO: Creating metasearch engines on demand//Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Toronto, 2003: 464

[17] Li W, Clifton C. Semantic integration in heterogeneous databases using neural networks//Proceedings of the 20th International Conference on Very Large Data Bases. Santiago, 1994: 1-12

[18] Miller R J, Ioannidis E Y, Raghu R. Schema equivalence in heterogeneous systems: Bridging theory and practice. Information Systems, 1994, 19(1): 3-31

[19] Milo T, Zohar S. Using schema matching to simplify heterogeneous data translation//Proceedings of the 24th International Conference on Very Large Data Bases. New York, 1998: 122-133

[20] Gio Wiederhold. Meditation to deal with heterogeneous data sources//Proceedings of the 2nd International Conference on Interoperating Geographic Information Systems. Zurich, 1999: 1-16

[21] Doan A, Domingos P, Levy A Y. Learning source description for data integration//Proceedings of the 3rd International Workshop on the Web and Databases. Dallas, 2000: 81-86

[22] He B, Chang K C. Statistical schema matching across Web query interfaces//Proceedings of the 22nd ACM SIGMOD International Conference on Management of Data. San Diego, 2003: 217-228

[23] He B, Chang K C, Han J. Discovering complex matchings across Web query interfaces: A correlation mining approach//Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Seattle, 2004: 148-157

[24] He B, Chang K C, Han J. Mining complex matchings across Web query interfaces//Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. Paris, 2004: 3-10

[25] Leake D B, Scherle R. Towards context-based search engine selection//Proceedings of the 5th International Conference on Intelligent User Interfaces. Santa Fe, 2001: 109-112

[26] Meng W, Yu C T, Liu K. Building efficient and effective metasearch engines. ACM Computing Survey, 2002, 34(1): 48-89

[27] Yu C, Liu K, Meng W, Wu Z, Rishe N. A methodology to retrieve text documents from multiple databases. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(6): 1347-1361

[28] Yu C T, Philip G, Meng W. Distributed top-N query processing with possibly uncooperative local systems//Proceedings of the 29th International Conference on Very Large Data Bases. Berlin, 2003: 117-128

[29] Zhang Zhen, He Bin, Chang Kevin Chen-Chuan. Light-weight Domain-based form assistant: Querying Web databases on the fly// Proceedings of the 31st VLDB Conference. Trondheim, Norway, 2005: 97-108

[30] Hammer J, Hector G, Nestorov S, Yerneni R, Breunig M M, Vassalos V. Template-based wrappers in the TSIMMIS system// Proceedings of the 16th ACM SIGMOD International Conference on Management of Data. Tucson, 1997: 532-535

[31] Arocena G O, Mendelzon A O. WebOQL: Restructuring documents, databases, and Webs// Proceedings of the 14th International Conference on Data Engineering. Orlando, 1998: 24-33

[32] Mecca G, Atzeni P, Masci A, Merialdo P, Sindoni G. The Araneus Web-base management system// Proceedings of the 17th ACM SIGMOD International Conference on Management of Data. Tucson, 1998: 544-546

[33] Liu L, Pu C, Han W. XWRAP: An XML-enabled wrapper construction system for Web information sources// Proceedings of the 16th International Conference on Data Engineering. San Diego, 2000: 611-621

[34] Crescenzi V, Mecca G, Merialdo P. RoadRunner: Towards automatic data extraction from large Web sites// Proceedings of the 27th International Conference on Very Large Data Bases. Roma, 2001: 109-118

[35] Crescenzi V, Mecca G, Merialdo P. RoadRunner: Automatic data extraction from data-intensive Web sites// Proceedings of the 21th ACM SIGMOD International Conference on Management of Data. Madison, 2002: 624

[36] Baumgartner R, Ceresna M, Gottlob G, Herzog M, Zigo V. Web information acquisition with Lixto suite// Proceedings of the 19th International Conference on Data Engineering. Bangalore, 2003: 747-749

[37] Baumgartner R, Flesca S, Gottlob G. Visual Web information extraction with Lixto// Proceedings of the 27th International Conference on Very Large Data Bases. Roma, 2001: 119-128

[38] Liu B, Grossman R L, Zhai Y. Mining data records in Web pages// Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, 2003: 601-606

[39] Zhai Y, Liu B. Web data extraction based on partial tree alignment// Proceedings of the 14th International World Wide Web Conference. Chiba, 2005: 76-85

[40] Embley D W, Jiang Y S, Ng Y. Record-boundary discovery in Web documents// Proceedings of the 18th ACM SIGMOD International Conference on Management of Data. Philadelphia, 1999: 467-478

[41] Kushmerick N. Wrapper induction: Efficiency and expressiveness. Artificial Intelligence, 2000, 118(1-2): 15-68

[42] Muslea I, Minton S, Knoblock C A. Hierarchical wrapper induction for semistructured information sources. Autonomous Agents and Multi-Agent Systems, 2001, 4(1-2): 93-114

[43] Adelberg B, Denny M. Nodose version 2.0// Proceedings of the 18th ACM SIGMOD International Conference on Management of Data. Philadelphia, 1999: 559-561

[44] Adelberg B. NoDoSE—A tool for semi-automatically extracting semi-structured data from text documents// Proceedings of the 17th ACM SIGMOD International Conference on Management of Data. Washington, 1998: 283-294

[45] Laender A H F, Berthier A R, Altigran S. DEByE—Data Extraction By Example. Data Knowledge Engineering, 2002, 40(2): 121-154

[46] Meng X, Lu H, Wang H, Gu M. SG-WRAP: A schema-guided wrapper generator// Proceedings of the 18th International Conference on Data Engineering. San Jose, 2002: 331-332

[47] Cohen W W, Hurst M, Jensen L S. A flexible learning system for wrapping tables and lists in Html documents// Proceedings of the 11th International World Wide Web Conference. Budapest, 2002: 232-241

[48] Pinto D, McCallum A, Wei X, Croft W B. Table extraction using conditional random fields// Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Toronto, 2003: 235-242

[49] Cai D, Yu S, Wen J, Ma W. Extracting content structure for Web pages based on visual representation// Proceedings of the 5th Asian-Pacific Web Conference. Xi'an, 2003: 406-417

[50] Song R, Liu H, Wen J, Ma W. Learning important models for Web page blocks based on layout and content analysis. SIGKDD Explorations, 2004, 6(2): 14-23

[51] Cai D, Yu S, Wen J, Ma W. Block-based Web search// Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Sheffield, 2004: 456-463

[52] Cai D, Yu S, Wen J, Ma W. Block-level link analysis// Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Sheffield, 2004: 440-447

[53] Zhao H, Meng W, Wu Z, Raghavan V, Yu C T. Fully automatic wrapper generation for search engines// Proceedings of the 14th International World Wide Web Conference. Chiba, 2005: 66-75

[54] Chang K C, He B, Zhang Z. Toward large scale integration: Building a MetaQuerier over databases on the Web// Proceedings of the 2nd Biennial Conference on Innovative Data Systems Research. Asilomar, 2005: 44-55

[55] Arlotta L, Crescenzi V, Mecca G, Merialdo P. Automatic annotation of data extracted from large Web sites// Proceedings of the 6th International Workshop on Web and Databases. San Diego, 2003: 7-12

[56] Wang J, Lochovsky F H. Data extraction and label assignment for Web databases//Proceedings of the 12th International World Wide Web Conference. Budapest, 2003: 187-196

[57] Lim E, Srivastava J, Prabhakar S, Richardson J. Entity identification in database integration. Information Systems, 1996, 89(1): 1-38

[58] Wei W, Liu M, Li S. Merging of XML documents//Proceedings of the 23th International Conference on Conceptual Modeling-ER 2004. Shanghai, 2004: 273-285

LIU Wei, born in 1976, Ph. D. candidate. His research interests include Web data management and integration.

MENG Xiao-Feng, born in 1964, professor and Ph. D. supervisor. His research interests include Web data management, native XML database, and mobile data management.

MENG Wei-Yi, born in 1958, professor and Ph. D. supervisor. His research interests include information retrieval, metasearch engines, and Web database integration.

Background

The work is supported by the National Natural Science Foundation of China under grant No. 60273018, the National High Technology Research and Development Program (863 Program) of China under grant No. 2002AA11304, and the National Basic Research Program (973 Program) of China under grant No. 2003CB317000.

As the rapid development of Web, a large number of Web data sources are emerging. So it is more and more difficult for users to get their desired information among these Web data sources manually. The intended purpose of those projects is to provide users an automatic approach to achieve and integrate the information in Web. In recent years, more and more attentions have been given to this area, and a great number of researchers have focused on some issues in it. In the past years, the authors have researched and developed a lot of techniques in the area of Deep Web integration, and these works mainly focus on Web database clustering, Web query interface integration and Web data extraction. A lot of issues in this area still have not been addressed well, or have not been touched even. So the content of this paper mainly provides a summary for previous works and helps researchers pay attention to the interesting issues need to address.