一种数据驱动的 Wrapper 自动生成与维护方法

王仲远 艾静 孟小峰

(中国人民大学信息学院 北京 100872)
(zhywang@ruc.edu.cn)

摘要 Wrapper 的生成与维护是 Deep Web 数据集成中一项非常重要的研究课题.传统的方法通常是通过对网页结构或特征的分析来推导 Wrapper,这种方法严重依赖于网站模板,在处理某些网站时可能完全失效.同时,以往研究对于 Wrapper 的维护问题关注较少.这两个问题导致无法真正实现大规模 Deep Web 数据集成.本文提出了一种新颖的数据驱动的 Wrapper 自动生成与维护方法.这种方法利用同一领域不同网站之间,以及同一网站不同版本之间的语义关系,通过数据项的匹配,来生成和维护 Wrapper.本文的方法没有模板依赖的问题,无需设置阈值.经过大量实验证明,此方法在准确性与适用性上,与原有方法相比有较大提高.

关键词 Deep Web;数据集成;Wrapper 生成;Wrapper 维护

中图法分类号 TP391

A Data Driven Approach for Automatic Wrapper Generation and Maintenance

Wang Zhongyuan, Ai Jing, Meng Xiaofeng
(School of Information, Renmin University of China, Beijing, 100872)

Abstract Wrapper generation and maintenance is a crucial research topic in Deep Web data integration. Existing methods usually induced Wrappers by structures or features analyzing of the website. However, these methods rely heavily on website templates and may be ineffective for some websites. Moreover, previous research paid less attention to wrapper maintenance. These two problems block the implement of large-scale Deep Web data integration. This paper proposes a novel method to perform this issue automatically, which is called data driven approach. This approach matches date items between source pages and target pages by the same semantic record of different websites in one domain or different templates in one site. Then it generates or maintains wrappers with these mapping data items. This approach doesn't rely on the template or set thresholds. Experimental results show that the accuracy and applicability of the method has improved greatly compared with the previous methods.

Key words Deep Web; Data Integration; Wrapper Generation; Wrapper Maintenance

随着网络与通信技术的迅速发展,Web 上的信息呈现爆炸性增长,互联网已经成为一个海量信息空间.UIUC 在 2007 年 5 月发表的一篇 Deep Web 综述[1]里估计,Deep Web 中的数据是 Surface Web 数据的 500 倍.面对如此海量的数据,主流搜索引擎却只能够覆盖到其中 32%的数据.因此,在 Deep Web 上进行数据集成是近些年来的一个研究热点.

不同于 Surface Web, Deep Web 中的数据一般是以查询结果页面以及记录的详细页面来展现.为了获取这些查询结果记录,目前通常使用 Wrapper 来进行 Deep Web 数据集成.Wrapper 是利用网站的模板,构造一组抽取规则,从而可以自动从网站上抽取出需要的信息并将其转换成结构化的数据.

收稿日期:2008-05-30
基金项目:本文得到国家自然科学基金项目(课题号:60573091); 国家 863 计划(课题号:2007AA01Z155); 教育部新世纪优秀人才支持计划; 北京市自然基金项目(课题号:4073035）的资助。

（a）当当网改版前页面 （b）当当网改版后页面

图 1 网站改版前后页面比较

目前已经有许多自动的方法来产生 Wrapper.这些方法多是考虑如何基于页面的模板特征进行分析,产生抽取规则,进而生成 Wrapper.我们将其称之为特征驱动的方法.但是这些方法都存在如下问题:第一,原有自动方法存在过多的假设以及对于模板特征的依赖,因此在处理不同模板时,查全率与查准率都有可能出现较大的波动;第二,原有自动方法多为考虑 Wrapper 的自动产生问题,而不太关注 Wrapper 的维护问题.但根据我们的观察,Deep Web 上的网站模板每隔数月就会更新一次,这是数据集成过程中不能不解决的问题.图 1 展示了网站改版前后的不同页面.通过观察可以发现,改版前后网页存在以下不同点:(1) 同一数据项的位置发生改变,例如本例中的书名的位置发生了变化;(2) 同一数据项的属性名称发生改变,例如“出版日期”变为了“出版时间”;(3) 同一数据项的值也有可能发生改变,例如网站价格的变动;(4) 数据项的数目可能会发生改变,包括增加和减少;(5) 有些数据项的值变为空值.

为了解决这些问题,本文提出了一种数据驱动的 Wrapper 自动生成与维护方法.这种方法以数据为核心,利用同一领域不同网站之间,以及同一网站不同版本之间的语义关系,在集成新的网站时,自动生成 Wrapper;在侦测到网站结构发生变化之后,自动维护 Wrapper.

本文的贡献在于:

(1)不同于以往工作使用结构或特征分析页面,进而产生 Wrapper 的方法,本文创新性地提出数据驱动的 Wrapper 导出方法.这一方法,利用同一领域不同网站,以及同一网站不同版本中相同实体之间的语义关系,进行数据项的匹配,找到相同语义的数据项,从而导出或更新抽取规则.

(2) 本文提出的方法,将 Wrapper 的生成与维护过程统一起来.Wrapper 生成与维护问题的实质是不同网站之间以及同一网站改版前后,其模板不同,但数据相似或相同.因此,以数据为驱动,能将两个问题统一起来.

(3)本文提出的数据驱动方法,无需设置参数及阀值.相对于先前方法,具有更广泛的适用性,并能达到较好的集成效果.

1 相关工作

在 Wrapper 自动生成的研究上,文献[7]提出通过分析网页 DOM 树结构,来自动导出 Wrapper;文献[8]则采取了基于视觉的方法来自动抽取数据.不过这些方法存在过多假设或参数,因此在处理结构较复杂的页面时效果会出现较大波动,甚至完全失效.文献[6]提到了这种特征驱动的方法可能带来的准确率波动,因此提出了基于集成中的上下文联系以及领域相关规则的 Wrapper 自动生成方法,称为场合感应方法.但这种方法仍没有彻底解决特征驱动方法的弊端.

在 Wrapper 的维护问题上,包括两个子问题:Wrapper 的验证与 Wrapper 的修复.文献[3]首先提出了 Wrapper 验证问题并给出了解决方法,文献[4]采用回归测试的方法检测页面的变化.这两篇文献比较好地解决了 Wrapper 验证的问题.SG-WRAM[2]是一种基于数据项特征的 Wrapper 自动维护方法.此方法利用变化后的页面上数据项保留的一些特征,进行数据项识别,自动完成 Wrapper 修复.但这种方法存在较强的假设,许多页面中,这些特征也发生了变化,从而无法使用这些特征完成 Wrapper 修复.

2 问题描述

对于数据驱动的 Wrapper 自动生成,本文假设已经存在一个或多个具有非常高的查全率与查准率的 Wrapper.我们将其称为种子点.利用种子点 Wrapper 抽取的数据项,来匹配其它网站上具有相同语义的数据

项,从而为这些网站自动生成 Wrapper,如图 2 所示.我们将这一过程定义为 Wrapper 扩散.其中的重点与难点就在于将原有的数据项与新页面上相同语义的数据项进行匹配.

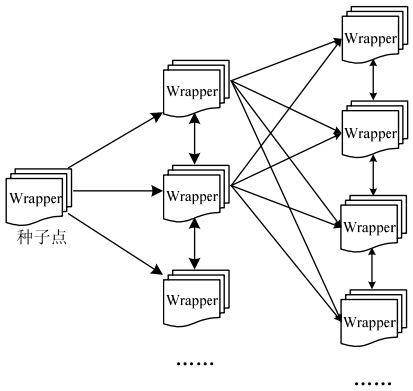


图 2 Wrapper 生成的扩散过程

对于 Wrapper 维护,先前工作基本上是着重于如何利用原有页面上保留的一些特征,来发现新页面的结构.但在实际中,我们发现这些特征并不一定能完全保留下来.不过,虽然网站的结构发生了改变,但网站上的数据,尤其是近期数据并未发生变化,如图 1 所示,这些数据只是以一种新的形式来呈现,我们可以充分利用这些数据,来进行 Wrapper 的修复,使其完全适应新的页面结构.因此,这一过程的难点也在于如何将原有的数据项与新的数据项进行匹配.

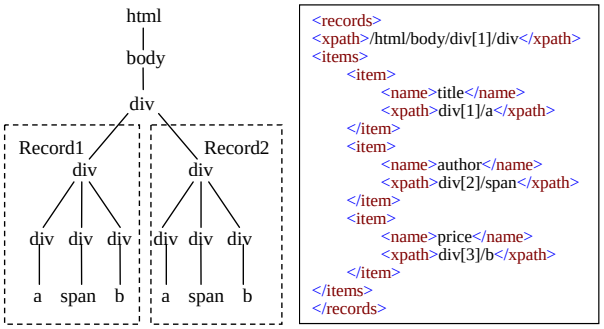
综上,数据驱动的 Wrapper 产生与维护的方法需要核心解决的问题就在于相同语义的数据块的匹配.

3 数据驱动的 Wrapper 产生与维护

3.1 基于 XPath 的 Schema-Guided 数据抽取方法

现有的基于树状结构分析或者基于视觉的抽取方法,能够处理简单的查询结果页面,但对于结构较为复杂的页面,则达不到较好的效果.而面向领域的 Schema-guided 方法[9],能够较好地描述领域特征以及网站的属性信息,仍然是较好的数据抽取方法.同时,基于 XPath 的 M-GCP 算法[10]能够进行非常精准的抽取.因此,本文考虑用 Schema-guided 与 XPath 相结合的方法来定义抽取规则,进而生成 Wrapper.

根据文献[6],可知在每个领域里,存在一个能够描述这个领域下的所有属性的模式.任意一个网站的模式,都是这个领域模式的一个子集.因此,可以利用预先定义的领域模式,来指导这个领域每个网站的属性抽取.而每个 Deep Web 网站的查询结果页面是一个 HTML 文件, 我们可以将其转换为 XHTML 文件,利用树结构及 XPath 将其中的数据抽取出来.



(a) XHTML 页面树结构 (b) Schema-Guided 抽取规则

图 3 基于 XPath 的抽取方法

XHTML 页面树结构如图 3(a)所示.在这棵树的第一个 div 结点下,包含两个孩子结点,这两个孩子结点是两条记录,分别为 Record1 与 Record2.在记录结点下面,又包括具体属性结点.这样,通过定义一组抽取规则,如图 3(b)所示,就能够利用结果页面的树结构,基于 XPath 对页面进行遍历以及抽取.

抽取出的结果,可以存储在本地数据库中,然后利用现有的成熟的数据库技术,再对其进行索引以及查询.当然,对于更复杂的页面结构,我们可以非常容易地通过扩展抽取规则,来实现数据遍历与抽取.

3.2 语义块发现

为了实现数据驱动的 Wrapper 生成与维护,最重要的步骤就是实现相同语义数据项的发现与匹配.

定义 1 源语义块:种子点 Wrapper 中定义的抽取规则所对应页面上被抽取的数据块,一般为某一条记录的属性值所对应的数据块.我们可以用如下三元组的形式表示源语义块:

$$\beta_i = (\alpha_i, \mu_i, \eta_i)$$

其中,β_i 表示该页面上的第 i 个语义块,α_i 表示该语义块的语义信息(一般为领域模式中的一个属性名),μ_i 表示该语义块的数据值(可能为文本信息或数字等),η_i 表示该语义块在网页上的 XPath 路径.

对于种子点网站或改版前网站而言,其包含的语义块是指 Wrapper 抽取规则所对应页面上的语义块.

定义 2 目标语义块:在新的待集成的网站页面或者是改版后的页面中,网页 DOM 树的每一个节点就是一个语义块.也可以使用三元组形式进行定义:

$$B_j = (A_j, V_j, P_j)$$

其中,B_j 表示该页面上的第 j 个语义块.A_j、V_j、P_j 的定义与 α_i、μ_i、η_i 相同.只是在目标语义块中,语义块的语义 A_j 为待确定项.

定义 3 匹配页面对:表示同一个实体的源页面与目标页面称为匹配页面对.用如下形式表示:

$$(SPage,TPage)_k$$

匹配页面对的发现有多种途径:针对 Wrapper 的产生,可以通过实体识别[11]的方法来得到;针对 Wrapper 的维护,可以通过保留下来的页面链接来获取匹配页面对.因此,在 Wrapper 扩散过程或者是 Wrapper 的维护过程中,通过获取到多个页面匹配对,将具有相同语义的源语义块与目标语义块匹配起来,从而完成 Wrapper 的产生或维护过程.

3.3 基于概率的语义块匹配方法

在获取一组匹配页面对并且分别在源页面以及目标页面上发现语义块后,下一步需要把源页面与目标页面上具有相同语义的语义块匹配起来.

本文方法基于以下观察:在一组匹配页面对上,

(1) 若语义块匹配,则源语义块上的数据值与目标语义块上的数据值大部分具有较高相似度值;

(2) 若语义块不匹配,源语义块上的数据值与目标语义块上的数据值大部分具有较低的相似度值.

因此,在通过一组迭代计算后,就能够去掉无用的目标语义块,筛选出匹配的语义块.

定义 4 数据相似度值: $P_k(\mu_i V_j)$表示第 k 个匹配页面对的第 i 个源语义块数据值与第 j 个目标语义块数据值的相似度值.

语义块的数据值可以分为两种类型:文本型与数值型.这两种类型的相似度值计算都已经有比较成熟的方法,例如文献[11,12]中提到的一些相似度计算方法.本文直接利用这些方法,计算数据相似度值.

计算文本型数据的相似度值可以利用 TF-IDF 余弦相似度计算方法:

$$P_k(\mu_i V_j)=\frac{\sum_{w=1}^{|D|} weight_{\mu_i}(w)\cdot weight_{V_j}(w)}{\| weight_{\mu_i} \|_2 \cdot \| weight_{V_j} \|_2} \quad (1)$$

其中,|D|为这段文本所包含的词的数量.而 $weight_{\mu i}(w)$与 $weight_{Vj}(w)$可以通过如下公式计算:

$$weight_{\sigma}(w)=\log(tf_w+1)\cdot\log(idf_w) \quad (2)$$

对于数值型数据,相似度值计算可以基于公式:

$$P_k(\mu_i V_j)=1-\frac{|\mu_i - V_j|}{\max(\mu_i, V_j)} \quad (3)$$

定义 5 语义块相似度值: $P_k(\beta_i B_j)$表示经过前 k 个匹配页面对的迭代计算后,第 i 个源语义块与第 j 个目标语义块的相似度值.

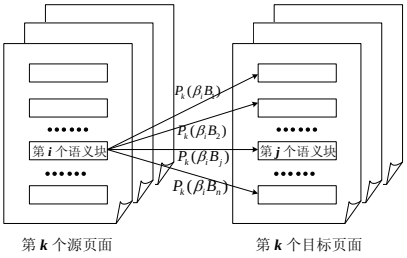


图 4 语义块匹配

如图 4 所示,假设源页面上有 m 个语义块,目标页面上有 n 个语义块,则通过 k-1 次迭代,可以得到 $P_{k-1}(\beta_i B_j)$,再利用 $P_k(\mu_i V_j)$,计算得到 $P_k(\beta_i B_j)$.

如果有 l 对匹配页面对,我们期望能够通过 l 次迭代计算后,将所有的 $P_k(\beta_i B_j)$ $(k=1,2,\cdots,l)$聚集到 0-1 概率轴的两端,从而通过聚类等方法,得到匹配语义块,避免人为设定阀值,实现全自动匹配.如图 5:

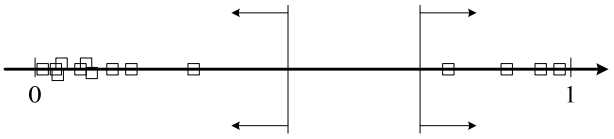


图 5 语义块相似度值在 0-1 概率轴上的分布

为此,我们考虑通过以下公式来扩大匹配语义块与不匹配语义块之间的概率差值:

$$P_k(\beta_i B_j)=\frac{P_{k-1}(\beta_i B_j)\times P_k(\mu_i V_j)}{P_{k-1}(\beta_i B_j)\times P_k(\mu_i V_j)+[1-P_{k-1}(\beta_i B_j)]\times[1-P_k(\mu_i V_j)]}$$

$$k\in(1,2,\cdots,l) \quad (4)$$

其中,公式的初值 $P_1(\beta_i B_j)$为首次计算的数据相似度值 $P_1(\mu_i V_j)$.而 $P_{k-1}(\beta_i B_j)$为通过前 k-1 个匹配页面对的计算,得到第 i 个源语义块与第 j 个目标语义块之间相似度值;$P_k(\mu_i V_j)$为第 k 个匹配页面对上第 i 个源语义块数据值与第 j 个目标语义块数据值的相似度值.通过公式(4)的迭代计算,最终得到 $P_k(\beta_i B_j)$.

3.4 使用匹配语义块进行 Wrapper 生成与维护

在完成源语义块与目标语义块的匹配后,需将源语义块的语义值 α_i 赋予匹配的目标语义块的 A_j.然后利用这些匹配语义块对完成 Wrapper 生成与维护.

对于 Wrapper 扩散过程中的新 Wrapper 的生成,可以使用 B_j 中的语义信息 A_j 与 XPath 路径 P_j,来构造 3.1 节中所提到的 Schema-Guided 的抽取规则;对于 Wrapper 的维护,可以利用 B_j 中的 XPath 路径 P_j 来替

换抽取规则中具有相同语义的抽取规则条目,这样就完成了 Wrapper 的自动修复.

4 实验

本文提出的数据驱动的方法,已经在工作信息领域进行集成实验,构建了原型系统工作通[13].经过一年的时间,集成的数据超过 200 万条.这个原型系统有力地证明了这种数据驱动的方法在适用性、容错性以及准确率上都达到了相当高的水平.

为了进一步验证本文方法有效性,我们在 4 个领域(图书、音乐、电影、计算机)中共选取了 26 个网站.在每个领域又分别选取一个网站的 Wrapper 作为种子点,以扩散方式为其他网站自动生成 Wrapper.

我们使用查全率和查准率来验证本文方法效果.

查全率:新生成或修复后的 Wrapper,正确抽取出的语义块数目占该网页上应被抽取出的语义块数目的比例;

查准率:新生成或修复后的 Wrapper,正确抽取出的语义块数目占匹配语义块数目的比例.

表 1 图书领域
(种子点:卓越网)

Deep Web 网站	#AT	#R	#RT
99 网上书城	7	6	6
china-pub 网上书店	8	7	7
当当网	17	15	15
蔚蓝网	10	9	9
新华在线	7	6	6
中国书网	11	11	11
中国图书网	17	17	16
总计	77	71	70

表 2 计算机领域
(种子点:比特网)

Deep Web 网站	#AT	#R	#RT
IT168	48	46	46
PCHOME	37	36	35
MyPrice	46	44	43
泡泡网	44	42	42
人民网 IT 频道	57	53	51
万维家电网	37	36	36
中华网科技	41	39	39
总计	310	296	292

表 3 音乐领域
(种子点:一听音乐网)

Deep Web 网站	#AT	#R	#RT
好听音乐网	7	7	7
网易娱乐资料库	12	11	10
九天音乐网	7	6	6
音乐天空	6	6	6
总计	32	30	29

表 4 电影领域
(种子点:IMDB 中文网)

Deep Web 网站	#AT	#R	#RT
环球影酷	14	13	12
中文电影资料库	9	8	8
中国影视资料馆	10	10	10
影视之狐	13	12	12
总计	46	43	42

表 5 4 个领域的 Wrapper 实验结果统计

	#AT	#R	#RT
总计	465	440	433
查全率:93.12%		查准率:98.41%	

在表 1 至表 5 中,#AT 表示该 Deep Web 网站的记录详细页面里应被抽取出的语义块数目,即有效语义块数目.#R 表示使用数据驱动的方法得到的匹配语义块数目.#RT 表示得到的匹配语义块中的有效语义块数目,即#RT 是#R 与#AT 的交集.

从实验结果可以看出,本文提出的数据驱动方法在查全率与查准率上都达到相当好的效果.而表 5 中平均 98.41%的查准率,说明了使用数据驱动方法,在 Wrapper 的生成与维护过程中,对于语义块的误判较小,得到的结果数据质量较高.并且,我们发现,本文方法在各个领域上的查全率与查准率比较平均.因此,本文方法在 Deep Web 上具有广泛的适用性.

此外,为了验证这种数据驱动方法在 Wrapper 维护方面的效果,我们还在半年内收集了卓越网与当当网改版前后的数据各 1 万条,验证数据驱动的 Wrapper 维护的效果.

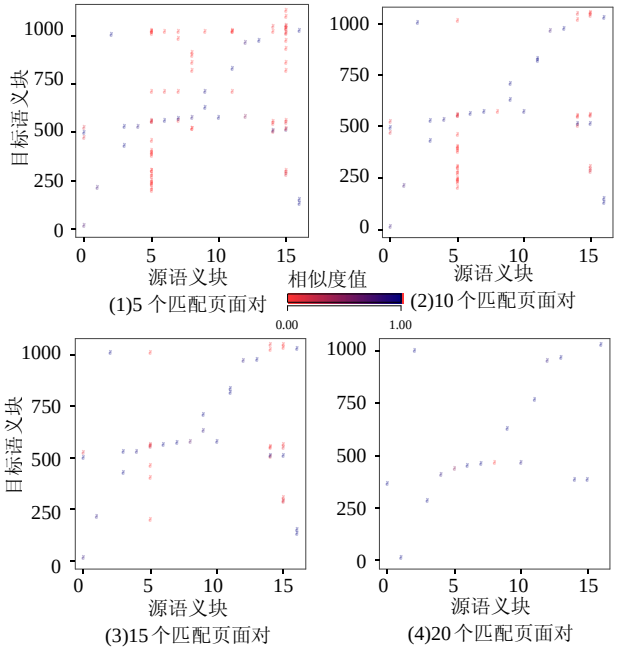


图6 Wrapper维护过程的散点图

图 6 显示了通过 5 个、10 个、15 个以及 20 个页面匹配对进行迭代计算所达到的语义块匹配效果.其中,横坐标轴为源语义块序列(β_1, β_2, …),纵坐标轴为目标语义块序列(B_1, B_2, …).源语义块与目标语义块的相似度值用不同颜色的散点来表示,其颜色值越接近红色,表示相似度值越低(当低于 10^{-4} 时,在图中就不进行标注);反之,颜色值越接近蓝色,相似度值越高.

从图 6 中可以看出,本文的方法使用较少的匹配页面对进行迭代计算,就能达到相当好的效果.在使用 20 个匹配页面对时,基本上已使相似度值趋向两极,从而完成 Wrapper 维护过程.

5 结束语

本文提出了一种数据驱动的自动的 Wrapper 生成与维护方法.这种方法利用网页上数据项的语义信息、数据值以及位置信息,构建语义块三元组,并将整

个页面看作是这种语义块的集合.然后在源页面与目标页面的语义块集合之间进行语义块的匹配,用新页面上语义块的语义信息和位置信息,生成新的 Wrapper 抽取规则.通过本文的实验,证明了该方法的有效性,并且可以达到较高的查全率与查准率.不同于传统方法的模板依赖或参数假设,本文的方法无需设置参数及阀值,因而具有更强的适用性.

参 考 文 献

[1] B. He, M. Patel, Z. Zhang, and K. C.-C. Chang. Accessing the Deep Web: A Survey. In: Communications of the ACM (CACM), May 2007, 50(5). 94-101

[2] X. Meng, D. Hu, C. Li. Schema-guided wrapper maintenance for web-data extraction. In: WIDM 2003. 1-8

[3] N. Kushmerick. Wrapper verification. In: World Wide Web Journal, 2000, 3(2). 79~94

[4] N. Kushmerick. Regression testing for wrapper maintenance. In: AAAI/IAAI 1999. 74-79

[5] B. Chidlovskii. Automatic repairing of web wrappers. In: WIDM 2001. 24-30

[6] S.-L. Chuang, K. C.-C. Chang, and C. Zhai. Context-Aware Wrapping: Synchronized Data Extraction. In: VLDB 2007. 699-710

[7] Y. Zhai, B. Liu. Web data extraction based on partial tree alignment. In: WWW, 2005. 76-85.

[8] W. liu, X. Meng, W. Meng. Vision-based Web Data Records Extraction. In: Proceedings of the 9th SIGMOD International Workshop on Web and Databases (SIGMOD-WebDB2006), June 30, 2006

[9] X. Meng, H. Lu, H.Wang, M. Gu. SG-WRAP: A Schema-Guided Wrapper Generator. In: Proceedings of the 18th International Conference on Data Engineering (ICDE), March 2002. 331-332

[10] T. Anton. XPath-wrapper induction by generalizing tree traversal patterns. In: LWA 2005. 126–133

[11] A. K. Elmagarmid, P. G. Ipeirotis, V. S. Verykios. Duplicate Record Detection: A Survey. In: The IEEE Transations on knowledge and Data Engineering (TKDE) Vol. 19 No. 1 January 2007, pp. 1-16

[12] S. Sarawagi, A. Bhamidipaty. Interactive deduplication using active learning. In: KDD. 2002

[13] Jobtong: http://www.jobtong.cn

王仲远,男,1985 年生,硕士,研究领域:Web 数据管理

艾静,女,1985 年生,硕士,研究领域:Web 数据管理

孟小峰,男,1964 年生,教授,博士生导师,主要研究方向为 Web 数据管理、XML 数据库、移动数据管理