实体解析技术综述与展望

朱 灿 曹 健
(上海交通大学电子信息与电气工程学院计算机系 上海 200240)

摘 要 实体解析是数据清理、数据集成、数据挖掘等技术中关键的一步,是数据质量的保障。介绍了实体解析含义、背景起源以及算法基础。列举并解释了实体解析发展过程中的经典算法,包括成对实体解析、集合实体解析、大数据的实体解析、复杂数据上的实体解析等,以及它们的特点和局限性,分享了在新的应用环境下衍生出来的针对不同需求的新的实体解析算法。最后展望了实体解析领域当前的研究热点以及发展方向。

关键词 实体解析,记录链接,集合数据,复杂数据,大数据
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.3.002

Summary and Prospect on Entity Resolution

ZHU Can CAO Jian
(Department of Computer Science, School of Electric Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract Entity Resolution(ER) is a key step in data cleaning, data integration, data mining and the insurance of data quality. This paper listed and explained some classic algorithms in the development of entity resolution, including pair-wise entity resolution, collective entity resolution, entity resolution on big data, and entity resolution on complex data et al. We also introduced the characteristics and limitation of these algorithms and shared some state-of-the-art algorithms derived from new application environment according to different requirements. Finally, the research hotspots and the development direction of this field were discussed.

Keywords Entity resolution, Record linkage, Collective data, Complex data, Big data

1 引言

在信息时代,数据的重要性毋庸置疑。以数据为中心的系统也得到了广泛应用,然而这些信息并非总是正确无误的,可能存在各种错误,比如重复、不一致、不正确、不完整等。据调查,全球财富1000强公司中有超过25%的关键数据存在不正确或不准确的问题[1]。

不同的数据提供方对同一个事物即实体(Entity)可能会有不同的描述(这里的描述包括数据格式、表示方法等),每一个对实体的描述称为该实体的一个引用。实体解析,是指从一个"引用集合"中解析并映射到现实世界中的"实体"过程。这在数据清理、数据集成、数据挖掘等以数据为中心的记录中都起着至关重要的作用,是数据质量的重要保障。实体解析(Entity Resolution)又被称为记录链接(Record Linkage)、对象识别(Object Identification)、个体识别(Individual Identification)、重复检测(Duplicate Detection)等。

1946年,Helbert L. Dunn在《American Journal of Public Health》发表了名为《Record Linkage》的文章,其后,Ho-ward Borden, Ivan Fellegi以及Alan Sunter等人为其奠定了理论基础。经过几十年的发展,实体解析技术已经被广泛应用在国民医疗系统、人口普查、多媒体数据库整合、银行信贷系统等领域。

根据输入类型的不同,实体解析可分为单实体解析和多实体解析。单实体解析,即指所有集合中的记录所对应的实体都是同一类型,如社会网络;多实体解析,指集合中的记录可能对应多种类型的实体,如商业销售系统中的商品、零售商。

根据待处理数据集的复杂程度又可将其分为简单数据实体解析和复杂数据实体解析。简单数据实体解析指的是数据结构简单,比如一般关系数据库中的元组;而典型的复杂数据实体解析则有XML数据和图数据。

2 实体解析算法基础

·数据预处理

用于实体解析的数据可能来自多个数据库,而这些数据库通常不会具有统一的数据模式和表现方法,当然更不可能有统一的标识符。因此需要对数据进行预处理,这个过程通常又被称为标准化(Normalization),包括统一格式、单位、大小写、删除空格、以标准格式对缩写名词进行扩写等,这是一个繁琐但是至关重要的环节。对数据的标准化处理可以基于

到稿日期:2014-04-17 返修日期:2014-07-16 本文受国家自然科学基金(61272438),上海市科委项目(12511502704,14511107702)资助。
朱 灿 男,硕士生,主要研究方向为网络数据分析与挖掘、网络计算,E-mail:0627wshhg@gmail.com;曹 健 男,教授,博士生导师,主要研究方向为网络计算、服务计算、数据智能分析。

规则，也可以基于隐马尔可夫模型进行[2]。

· 相似度函数

相似度（距离）函数用来度量记录对属性值之间的一致程度，相似度越高，距离越近，属性值就越一致。相似度（距离）函数种类繁多，应该根据实际情况进行选取。常用的有：基于字符串的度量函数，包括编辑距离、Smith-Waterman 算法、Jaro-Winkler 算法、TF-IDF 算法等；基于集合的度量函数，包括 Jaccard 距离、Dice 等；基于向量的度量函数，包括欧氏距离、Cosine 相似度等。

3 实体解析技术分类及比较

为了更好地说明问题，本文对实体解析中常用概念做如下定义：(1)数据源集合 $D\{d_1,d_2,\cdots,d_{|D|}\}$；(2)实体集合 $E\{e_1,e_2,\cdots,e_{|E|}\}$；(3)记录集合 $R\{r_1,r_2,\cdots,r_{|R|}\}$[；(4)属性集合 $A\{a_1,a_2,\cdots,a_{|A|}\}$；(5)属性值集合 $V\{v_1,v_2,\cdots,v_{|V|}\}$，且满足 $\forall v_p\in V,\exists r_j\in R,a_q\in A,\text{s.t. } r_j.a_q=v_p$。

根据以上定义，实体解析算法的输入是实体 E 的记录集合 R，R 的属性集合和属性集集合分别为 A 和 V；输出是经过解析的记录集合 $R'\{r_1,r_2,\cdots,r_{|E|}\}$。

3.1 经典实体解析算法

3.1.1 成对实体解析

最基本的实体解析问题有基于单个数据库和基于两个无重复记录数据库两种类型。前者可以看成是重复记录检测，而后者则可以看作是对记录对集合 $P\{p(r_1,r_2)\mid r_1\in d_1,r_2\in d_2\}$ 中的每一对记录用函数 $fun(r_1,r_2)$ 进行匹配，fun 函数首先计算各项属性之间的相似度（距离），然后根据其各属性相似度做出判断。判断的方法主要有以下几种：基于规则，通常需要比较深厚的领域知识，且不容易构造，同时调整起来也比较困难；基于权重，为各属性分配权重，计算各对属性相似度的加权和，根据事先设定的阈值来决定是否匹配，各属性权值可以由领域内专家分配，也可以通过机器学习获得；基于机器学习，包括决策树[3]、支持向量机[4,5]、嵌套分类器[6]、条件随机场[7]等，机器学习方法的主要缺点在于能高度反映目标数据集特点的训练集非常难以构建。

3.1.2 集合实体解析

· 相似度传播

传统的实体解析算法通常基于记录的属性，即使考虑了记录之间的关系，也是将这些关系看作是“特殊的属性”。基于属性的算法都是“静态”的，因为任意一对记录 $p(r_1,r_2)$ 匹配值是固定的，不随时间变化，也不受其他记录对的影响，只取决于 r_1 和 r_2 本身。在 Fellegi 和 Alan Sunter 的概率模型中[8]，所有记录对都是独立同分布的（Independent Identically Distributed，IID）。然而，在现实世界中，仅仅考虑属性是远远不够的，同时，记录“所有记录对”都独立同分布这样的假设也不符合实际情况。考虑下面的例子[9]，两篇文章的作者均为“W Wang”和“A Ansari”：

$p_1(paper_1.W\ Wang, paper_2.W\ Wang)$

$p_2(paper_1.AAnsari, paper_2.AAnsari)$

在文章 1 和文章 2 中，“W Wang”和“A Ansari”都有合作关系，假如已经确定了 p_2 匹配，显然这对 p_1 的匹配有正面的影响（Positive Evidence）。所以，两个记录对之间并不是独立的，而是相互影响的。

记录集合可以很自然地被转化为“图”，每个记录都是图中的一个叶节点，每一个合作关系都可以用一条超边表示[11]。“图”上的实体解析问题可以看作是叶节点的聚类问题。

a)构建初始的簇集合 $C\{c_i\}$，在这个阶段，需要一些已经合并过的簇来激活整个聚类过程。这些“激活簇”所提供的“证据”在之后的迭代中通过记录间的关系逐步传播，最终达到稳定状态，所以这类算法又叫“相似性传播”算法。激活簇的构建可以参考文献[9]。

b)在初始的 C 上进行迭代，聚类的依据仍然是基于“相似度”，只不过这里的相似度并不是记录之间的，而是簇之间的，记为 $\text{Sim}(c_i,c_j)$。可以作为簇间相似度的度量参数有很多，常用的有以下几种[12]。

公共邻居：

$$CommonNbrScore(c_i,c_j)=\frac{1}{K}\times|Nbr(c_i)\cap Nbr(c_j)| \quad (1)$$

其中，K 为确保函数值小于 1 的足够大的正常数，$Nbr(c_i)$ 为簇 c_i 的邻居。但是考虑下面这种情况，有 3 个簇 c_1，c_2，c_3，其中 c_1，c_2 有一个邻居，c_3 有 10 个邻居，且 c_1 与 c_2，c_1 与 c_3 都只有一个公共邻居。若根据式(1)，c_1 与 c_2 的相似度和 c_1 与 c_3 的相似度是相同的，但根据直观的感受，c_1 与 c_2 相似度应大于 c_1 与 c_3 的相似度，因为 c_1 与 c_2 的邻居是完全相同的。所以当考虑到邻居集合的大小时，就有了 Jaccard 系数。

Jaccard 系数：

$$Jaccardcoeff(c_i,c_j)=\frac{|Nbr(c_i)\cap Nbr(c_j)|}{|Nbr(c_i)\cup Nbr(c_j)|} \quad (2)$$

Jaccard 系数考虑的是两个簇的公共邻居在它们所有邻居中所占的比例。Jaccard 系数也并不够完善，在计算 Jaccard 系数时，是基于这样的假设：每个邻居对两个簇之间相似度的贡献都是一样的，但是如 Adamic 和 Adar 所提出的“逆文档频率”（Inverse Document Frequency，IDF）所言，一个被两个文档共享的词出现的频率越高，它就越不重要，对相似度的贡献也就越低，这在实体解析问题中也具有同样的意义。

Adar 系数：

$$AdarCoeff(c_i,c_j)=\frac{\sum_{c\in Nbr(c_i)\cap Nbr(c_j)}u(c)}{\sum_{c\in Nbr(c_i)\cup Nbr(c_j)}u(c)} \quad (3)$$

其中，$u(c)$ 表示 c 的单值性（uniqueness），类似“逆文档频率”可以定义为：

$$u(c)=\frac{1}{\log(|Nbr(c)|)} \quad (4)$$

c)每当有新的簇产生，更新与之关联的簇（有超边连接）的相似度。

d)重复步骤 b)，c)，直到不再产生新的簇为止。

· 马尔科夫逻辑网

一个马尔科夫逻辑网 L 由一组二元对 (F_i,w_i) 表示，其中 F_i 是一阶逻辑规则，w_i 是一个实数。L 与一组有限常量集合 $C=\{c_1,c_2,\cdots,c_n\}$ 一起定义了一个马尔科夫网 $M_{L,C}$[10]：

a)L 中的任意闭原子（ground atom）都对应了 $M_{L,C}$ 中的一个二值节点。若此闭原子为真，则对应的二值节点取值为 1；若为假，则取值为 0。

b)L 中的任意闭规则（ground formula）都对应着一个特征值，若此闭规则为真，则对应的特征值为 1；若为假，则特征

值为0，并且这个特征值 F_i 的权重为二元项中该规则对应的权重 w_i。

$M_{L,C}$ 中的节点由 L 中的闭原子生成，而边则由这些原子的关系生成。当两个闭原子在同一个闭规则中时，这两个闭原子所对应的节点就存在一条边。$M_{L,C}$ 中一个可能世界发生的概率是其包含的真闭规则的权重之和，这些权重可以通过梯度下降等方法学习得到[15]。

记录的属性都可以用一系列谓词来表示，如 *hasAuthor*(*paper*,*author*)，*hasTitle*(*paper*,*title*)。这样，记录集合就可以转化为马尔科夫网。

另外，文献[36]提出基于直觉的关联强度模型，并根据两个记录的关联强度进行匹配；而文献[37]在文献[36]的基础上，提出了结合受监督学习的自适应关联强度模型，该基础模型可以通过使用不同领域的训练集进行训练，以得到对该领域有更好适应性的关联强度模型；文献[38]通过借助实体的事务日志，分析实体的行为模式，来解决实体解析的问题。

3.2 大数据上的实体解析

目前的实体解析算法都是基于成对比较的，这使得算法难以应用在大数据上。想像一下，在1000个城市中分别选取了1000家公司，并在其上进行实体解析。假如不做任何预处理，则需要进行 $(1000*1000)^2=10^{12}$ 次比较，假设每次比较耗时1微秒，那么整个解析过程需要花费11天。然而，如果在进行解析之前稍微进行一些预处理，比如只在同一个城市内的公司之间进行比较，因为根据常识，不在同一个城市的两个公司不太可能是同一个公司，那么比较次数将减少到 10^9 次，仅耗时16分钟，这就大大提高了效率。像这样，根据某种知识或规则对数据进行预处理，将它们分成规模更小的数据块(Block)，并在这些块里进行实体解析，以提高算法效率的方法，统称为分块技术(Block Technique)。

· 基于hash函数的分块

该方法的核心思想是：

(1)定义一个关于一项或多项属性的hash函数，每个块 b_i 都有一个hash值 h_i 标识；

(2)将所有 $hash(r)=b_i$ 的引用 r 都归入 b_i 中；

(3)所有的块都互不相交；

(4)实体解析算法仅在块内运行。

hash函数的种类多种多样，比如城市名、邮编等。但是对于一个复杂的问题来说，仅用单个规则是不够的，通常需要将规则合并，如城市名+手机号码后4位。minHash算法[16]就是基于多种hash键值来进行分块的算法。

(1)假设 $F\{f_i\}$ 是定义在记录集合 X 上的一系列hash函数，$F(x)$ 是 $\{f_i\}$ 作用于记录 x 而得到的一组键值向量 $K_x\{k_{x,i}\}$；

(2)假设 π 是(1-|F|)的一个随机排序，π 的第一个值为 m，则 $minHash(x)=k_{x,m}$。

根据minHash的定义，不难得出，对于两条记录 x 和 y，它们的minHash值相同的概率为：

$$P(minHash(x)=minHash(y))=\frac{|K_x\cap K_y|}{|K_x\cup K_y|} \tag{5}$$

这正是两个记录 x 和 y 的键值集合 K_x 和 K_y 的Jaccard系数。所以 x 和 y 的Jaccard系数越大，它们被分配到同一个块中的概率就越大。

· 基于相似度与距离的分块[17,18]

Canopy算法[18]：将记录集合中的每一条记录都映射成空间中的点，然后根据空间中各点的位置，将聚集的点划到一个块中：

(1)首先设置两个阈值 T_1 和 T_2，且 $T_1>T_2$，并设计好距离函数 $distance(x,y)$ 表示任意两个记录之间的距离；

(2)设记录集合为 R，任取 $r\in R$，新建一个块 B_i，将所有与 r 距离小于 T_1 的记录都加入 B_i；

(3)删除所有与 r 距离小于 T_2 的记录；

(4)重复步骤(2)和(3)直到 R 为空。

除此之外，文献[30,31]提出token分块算法，并根据块的性耗比[31]对块进行排序，剔除低性耗比的块，以保证算法的质量和效率；文献[32]提出的后缀数组算法，根据分块键值的特定长度后缀进行分块；文献[33]将每条记录的键值映射到多维的欧氏空间中，然后根据键值之间的距离来确定相似的记录对；文献[34,35]根据键值特定长度的字串来进行分块；文献[42]借鉴数据挖掘中的频繁项集(Frequent Itemsets)，定义基于最大频繁项集(Maximal Frequent Itemsets)的分块算法，该算法可以很大程度上减弱在设定分块键值时对特定领域专业意见的依赖。

3.3 复杂数据上的实体解析[28]

前文所讨论的实体解析算法都是基于简单数据的。然而真正的互联网上往往有更复杂、更庞大的数据且还在日益增大，这些数据很有可能来自多个数据源，典型的有facebook、twitter等社交网站，其社会网络数据有大量不同的用户维护，这些数据中不可避免地会出现大量的不一致数据，也可能会以各种结构和形式被存储。

XML由于其灵活性和可扩展性，被广泛应用在各种互联网应用和商务应用中。但是来自不同数据源的数据可能具有不同的模式，当然也可能存在同一实体的不同表示，在这样的数据上做出分析和决策自然不可靠。所以在XML数据上的实体解析很有必要和价值。当前，XML数据上的实体解析的应用包括了Web与Peer-to-Peer环境下描述同一实体数据的发现[19]、Web上交换与发布的重复数据的发现[20,21]、Web上多数据源集成[22]等。

由于XML含有大量的结构信息，因此最常用的方法还是基于结构相似度或距离的匹配方法；又由于XML可以转化为树结构，因此用于描述树相似性的算法可以很自然地移植到XML。最早的用于衡量数的相似性的算法是树的编辑距离算法[23]，其定义与字符串的编辑距离相似——树A转化为树B所需要增加、删除或者修改的最少节点数即为树A与树B的编辑距离。之后，又出现了基于贝叶斯网络的XML文档相似性算法[24]，其基本思想是：

(1)文档之间的相似性是它们的根节点之间的相似性；

(2)两个节点之间的相似性由它们的后代节点相似性对应的条件概率决定；

(3)两个叶节点的相似性是它们内容的相似性。

除上述方法外，文献[25]提出了基于路径集合相似性的算法。文献[19,26]提出了结合树的结构相似性与内容相似性的算法，其中文献[19]将结点名称相似性、路径相似性和节点所有后代的内容相似性的平均值作为XML文档的相似性；文献[26]将公共叶节点作为内容相似性，叶节点的平均路径相似性作为结构相似性，并将这两者的乘积作为XML文

档的相似性。

3.4 动态实体解析

传统的实体解析算法都应用于静态数据库，然而现实中有很多数据是动态的，不管是实体的属性还是实体间的关系都随着时间演化。一个简单的例子，作者在发表文章时使用的署名就可能会随着时间演化，比如一位名为 Xin Dong 的作者在 2006 年之前发表文章的署名是“Xin Dong”，而在 2007 年开始使用署名“Xin Luna Dong”。如果忽略了这类数据的特点，就很有可能造成不一致的结果。从表 1 的例子中可以清晰地看到时态实体解析的特点与面临的问题。

表 1

实体 ID	表象 ID	名字	附属单位	邮箱地址	合作者	时间/年
e_1	a_1	Xin Dong	R. Polytechnic Institue	null	Michael J. wozny	1991
e_2	a_2	Xin Dong	Univ of Washington	lunadong@cs. washington. edu	Halevy	2004
	a_3	Xin Dong	null	lunadong@cs. washington. edu	Halevy	2005
	a_4	Xin Dong	Univ of Washington	lunadong@cs. washington. edu	Halevy	2007
	a_5	Xin Luna Dong	Univ of Washington	null	Halevy	2007
	a_6	Xin Luna Dong	Univ of Washington	lunadong@cs. washington. edu	Halevy	2009
	a_7	Xin Luna Dong	AT&T Labs	lunadong@research. att. com	Naumann	2010
e_3	a_8	Dong Xin	Univ of Illinois	null	Wah	2007
	a_9	Dong Xin	Microsoft	dongxin@microsoft. com	Yeye He	2010

从表中不难看出，作者“Xin Dong”所属单位（2004－2009 年一直在“华盛顿大学”工作，而在 2010 年加入“AT&T 实验室”）以及署名（从“Xin Dong”变为“Xin Luna Dong”）的变化。还可以看出当 Xin Dong 的所属单位发生变化时，其合作者也发生了变化，这在日常生活当中是很正常的，然而传统的考虑引用间的关系信息的算法则很可能将这两个引用划分到不同的实体中。这个例子充分说明，不管是实体的属性还是实体间的关系都是随着时间演化的。

在考虑时间属性的实体解析问题时，输入是具有时间信息的实体 $E\{e_1,e_2,\cdots e_{|E|}\}$ 的记录集合 $R\{r_1,r_2,\cdots,r_{|R|}\}$，其中 $r_i=\{value_1,value_2,\cdots,value_{|A|},t\}$，$t$ 为记录的时间戳，$value_j$ 为 r_i 在时刻 t 的第 j 项属性 $attribute_j$ 的值，$|A|$ 为 r_i 的属性个数；输出是将记录集合划分成一个或多个子集合（簇），即 $C\{c_1\{r_{1,1},r_{1,2},\cdots,r_{1,m}\},c_2\{r_{2,1},r_{2,2},\cdots,r_{2,n}\},\cdots,c_{|E|}\{r_{|E|,1},r_{|E|,2},\cdots,r_{|E|,k}\}\}$。T-Clustering 算法的簇相似度公式如下：

$$\mathrm{SIM}(c_i,c_j)=\alpha\times\mathrm{Sim}_{TA}(c_i,c_j)+\beta\times\mathrm{Sim}_{TR}(c_i,c_j)+\gamma\times\mathrm{Sim}_G(c_i,c_j) \tag{6}$$

其中，Sim_{TA}、Sim_{TR}、Sim_G 分别表示两个簇基于时间的属性相似度、基于时间的关系相似度以及群相似度，α、β、γ 分别为其权重，且 $\alpha+\beta+\gamma=1$。其中 Sim_G 可用两个簇中记录在不同关系上的公共邻居率的平均值表示。公式如下：

$$\mathrm{Sim}_G(c_i,c_j)=\frac{\sum_{rel\in REL}\frac{1}{|c_i|\times|c_j|}\sum_{r_p\in c_i}\sum_{r_q\in c_j}CNR_{rel}(r_p,r_q)}{|REL|} \tag{7}$$

$$CNR_{rel}(r_p,r_q)=\frac{|r_p.Neighbor\cap r_q.Neighbor|}{|r_p.Neighbor\cup r_q.Neighbor|} \tag{8}$$

其中，REL 表示所有引用间的关系的集合，$CNR_{rel}(r_p,r_q)$ 表示引用 r_p,r_q 在关系 rel 上的公共邻居率。后文主要介绍 Sim_{TA} 和 Sim_{TR} 的设计与公式，对于 Sim_G 不再详述。

为了解决随时间变化的属性（记为 TVA，相应的不随时间变化的属性记为 TIA）对实体解析的影响，在计算属性相似度时引入了时间调节器（记为 tta）[29]：当属性相似度大于一定阈值（记为 θ_{high}）时，记为 $tta_=(tva_i,\Delta t)$，表示不同实体的属性 tva 的值在时间间隔 Δt 上保持不变的概率，目的是减少属性 tva 在 Δt 内不随时间变化的相似度奖励；当属性相似度小于一定阈值（记为 θ_{low}）时，记为 $tta_{\neq}(tva_i,\Delta t)$，表示不同实体的属性 tva 的值在时间间隔 Δt 上发生变化的概率，目的是减少属性 tva 在 Δt 内随时间发生变化的相似度惩罚。引入 tta 后，两个记录之间的属性相似度的公式表示如下：

$$\begin{aligned}\mathrm{Sim}_{TA}(r_1,r_2)&=\omega\times\frac{\sum_{i=1}^{|TVA|}(1-tta(tva_i,\Delta t))\times\mathrm{Sim}_A(r_1.tva_i,r_2.tva_i)}{\sum_{i=1}^{|TVA|}(1-tta(tva_i,\Delta t))}\\&+(1-\omega)\times\sum_{j=1}^{|TIA|}w_j\times\mathrm{Sim}_A(r_1.tia_j,r_2.tia_j)\end{aligned} \tag{9}$$

其中，ω、$1-\omega$ 分别表示 tva 相似度和 tia 相似度的权重，w_j 表示各 tia 的权重且和为 1，Sim_A 表示不考虑时间信息时的相似度。如前文所述，根据 Sim_A 的不同大小 $tta(tva_i,\Delta t)$ 将取不同的表达式，即：

$$tta(tva_i,\Delta t)=\begin{cases}tta_=(tva_i,\Delta t), & \mathrm{Sim}_A>\theta_{high}\\ tta_{\neq}(tva_i,\Delta t), & \mathrm{Sim}_A<\theta_{low}\end{cases} \tag{10}$$

类似于时间调节器对于属性相似度的作用，演化一致性系数用于调整记录间的关系相似度（记为 $rec_{\neq}$）。当记录间的属性相似度小于阈值（θ_{low}）时，潜在的属性变化可能导致潜在的关系变化，演化一致性系数的引入可以减少同一个实体的关系在 Δt 内发生变化的相似度惩罚，记为 $rec_{\neq}(tvr_i,\Delta t)$[29]。根据定义，在应用演化一致性系数之前，首先要明确，哪些属性的改变可能导致哪些关系的变化，也就是说，需要找到一些如下形式的潜在演化对集合 $EP\{ep(tva,rel)\}$，并且保存在时间知识库中。所以 $rec_{\neq}(tvr_i,\Delta t)$ 表示在时间间隔 Δt 上，当属性 tva 发生变化（即 $\mathrm{Sim}_A(r_1.tva,r_2.tva)<\theta_{low}$，并且存在潜在演化对 $ep(tva,rel)\in EP$）时，关系 tvr 也发生变化的概率。引入两个记录间的关系，相似度公式改写如下：

$$\begin{aligned}\mathrm{Sim}_{TR}(r_1,r_2)=&\\ \lambda\times&\frac{\sum_{i=1}^{|TVR|}(1-rec_{\neq}(tvr_i,\Delta t))\times\mathrm{Sim}_R(r_1.tvr_i,r_2.tvr_i)}{\sum_{i=1}^{|TVR|}(1-rec_{\neq}(tvr_i,\Delta t))}\\&+(1-\lambda)\times\sum_{j=1}^{|TIR|}s_j\times\mathrm{Sim}_R(r_1.tvr_j,r_2.tir_j)\end{aligned} \tag{11}$$

其中，λ、$1-\lambda$ 分别表示 tvr 相似度和 tir 相似度的权重，s_j 表示各 tir 的权重且和为 1，Sim_R 表示不考虑时间信息时的关系相似度。tta 和 rec 的值可以通过在标注的数据集上进行机器学习得到（如 time decay[27] 的学习方法）。

动态实体解析还包括针对数据流和查询的实体解析算

法，如文献[39]提出的算法可随时间变化，通过已得到的解析结果来减少冗余作业；文献[40]实现的相似度可感动态反向索引算法，旨在解决记录查询流的实时实体解析问题，其主要思想是预先算好各块内各对属性的相似度，将其载入内存，并维护相应的反向索引来提高响应速度；文献[41]通过分析查询的语义信息，引入“退化”(vestigiality)概念，剔除对查询结果不会造成影响的无必要比较，以提高查询效率。

结束语 实体解析技术发展到现在，已经提出了许多针对各类问题的方法，也有相当一部分已经被实际应用，但仍有些问题尚待解决：

(1)尽管实体解析的算法已不少，但目前仍缺少有效的评价方法。准确率和召回率是目前最常用的评价方法，准确率可以通过对识别结果进行人工检测得到，但是召回率却需要与整个数据集的真实结果进行比较，当数据集较大时这样的人力耗费是不可承受的，需要有新的有效的衡量方法或指标来解决这个问题。

(2)可用于测试的大规模真实数据集的缺失也是目前亟待解决的一个问题。现在用于研究和测试实体解析算法的数据很多是人为合成的，这样的数据虽然易于扩展和分析，但是并不能完全代替真实的数据集，因为它们不能完整体现现实世界数据的复杂性。要想进一步提升算法与现实的契合度，一个优质的大规模数据集是必不可少的。

(3)实体解析技术已经被运用在一些实时应用中，这些应用通常不要求解析的结果十分准确，但要求解析过程必须在规定时间内完成。因此，实时实体解析需要在有限的时间内最大化解析的效果。比如，在反恐应用中，各种数据的分析都需要实时完成，虽然效果没有在完整数据库上的好，但是快速处理可以大大提高抓捕嫌疑人的可能。

(4)实体解析可以从多个数据源中找到相同的实体，也就是说，从不同的数据源中可以提取出特定用户的不同方面的信息，从而得到该用户的全面信息，这无疑会给用户的隐私和信息安全带来隐患。目前，隐私保持的实体解析算法研究并不太多，但无疑是实体解析必须面对的问题。常用的做法是预先将数据交给可信赖的第三方，由其对敏感信息进行剥离，并限制暴露给请求数据者能得到的可识别数据项[43]。这种方法不可避免地降低了解析的准确性，同时依然无法摆脱对“人”的依赖。如何更好地在保证安全和隐私的情况下进行实体解析也是当前研究的一个热门课题。

参 考 文 献

[1] Redman T C. The impact of poor data quality on the typical enterprise[J]. Communication of ACM, 1998, 41(2): 79-82

[2] Tejada S, Knoblock C A, Minton S. Learning object identification rules for information integration[J]. Information Systems Journal, 2001, (08): 607-633

[3] Cochinwala M, et al. Efficient data reconciliation[J]. Information Sciences, 2001, 137(1-4): 1-5

[4] Bilenko M, Mooney R. Adaptive Duplicate Detection Using Learnable String Similarity Measures[C]// KDD 2003. 2003: 39-48

[5] Christen P. Automatic record linkage using seeded nearest neighbour and support vector machine classification[C]// KDD 2008. 2008: 151-159

[6] Chen Z, et al. Exploiting context analysis for combining multiple entity resolution systems[C]// SIGMOD 2009. 2009: 207-218

[7] Sarawagi R. Answering Table Gupta & S. Augmentation Queries from Unstructured Lists on the Web[J]. PVLDB, 2009, 2(1): 289-300

[8] Fellegi I, Sunter A. A Theory for Record Linkage [J]. JASA 1969, 64(328): 1183-1210

[9] Bhattacharya I, Getoor L. Collective Entity Resolution in Relational Data[C]// TKDD 2007. 2007

[10] Richardson M, Domingos P. Markov logic networks [J]. MachineLearning, 2006, 62(1/2): 107-136

[11] Dong X, et al. Reference Reconciliation in Complex Information Spaces[C]// SIGMOD 2005. 2005

[12] Liben-Nowell, Kleinberg. The Link-Prediction Problem for Social Networks[J]. Journal of the American Society for Information Science and Technology, 2007, 58(7): 1019-1031

[13] Bhattacharya I, Getoor L. A Latent Dirichlet Model for Unsupervised Entity Resolution[C]// SDM 2007. 2007

[14] Broecheler M, Getoor L. Probabilistic Similarity Logic [C]// UAI 2010. 2010

[15] Singla P, Domingos P. Entity Resolution with Markov Logic [C]// ICDM 2006. 2006: 572-582

[16] Broder A, et al. Min-Wise Independent Permutations[J]. Journal of Computer and System Science, 2010, 60(3): 630-659

[17] Hernandez M, Stolfo S. The merge/purge problem for large databases[C]// SIGMOD 1995. 1995: 127-138

[18] McCallum A, et al. Efficient clustering of high-dimensional data sets with application to reference matching[C]// KDD 2000. 2000

[19] Kade A M, Heuser C A. Matching XML documents in highly dynamic applications[C]// Proceedings of the 2008 ACM Symposium on Document Engineering. SaoPaulo, Brazil, 2008: 191-198

[20] Puhlmann S, Weis M, Naumann F. XML duplicate detection using sorted neighborhoods[C]// Proceedings of the 10th International Conference on Extending Database Technology. Munich, Germany, 2006: 773-791

[21] Weis M, Naumann F, Dogmati X. tracks down duplicates in XML[C]// Proceedings of the ACM SIGMOD International Conferenceon Management of Data. Baltimore, Maryland, USA, 2005: 431-442

[22] Carvalho J C P, Silva A S. Finding similar identities among objects from multiple websources[C]// Proceedings of the 5th ACM CIKM International Workshop on Web Information and DataManagement. NewOrleans, Louisiana, USA, 2003: 90-93

[23] Tai K C. The tree-to-tree correction problem [J]. Journal of ACM, 1979, 26(3): 422-433

[24] Leito L, Calado P, Weis M. Structure based inference of xml similarity for fuzzy duplicate detection[C]// Proceedings of the 16th ACM Conferenceon Information and Knowledge Management. Lisbon, Portugal, 2007: 293-302

[25] Joshi S, Agrawal N, Krishnapuram R, et al. A bag of paths model for measuring. structural similarity in Web documents[C]// Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and DataMining. Washington, DC, USA, 2003: 577-582

(下转第 18 页)

机器人编程语言进行对比分析，并以此对 RPL 语言的语法和机制进行改进和完善；(2)本文是将单个机器人当作一个 Agent来考虑，未能充分利用 Agent 的社会性等特性，后续工作将尝试用多 Agent 技术来考虑单个机器人的控制软件开发问题以及考虑多机器人系统的软件开发问题；(3)本文基于反应式 Agent 来考虑机器人的编程问题，但反应式 Agent 模型仅仅是通过对事件处理机制即反应式规则来支持个体Agent 的自主决策，因而其自主决策能力非常有限。考虑采用Agent 的其他软件模型(如慎思型、混合型等)来支持机器人的编程以提高机器人自主决策的能力将是下一步研究的重点之一。

参 考 文 献

[1] Ziafati P, Dastani M, Meyer J J, et al. Agent Programming Languages Requirements for Programming Autonomous Robots [M]// Programming Multi-Agent Systems. Springer Berlin Heidelberg, 2013: 35-53

[2] 戴齐，姚先启. 机器人程序设计语言[J]. 机器人，1997，19(5)：390-400

[3] Jennings N R, Sycara K, Wooldridge M. A roadmap of agent research and development[J]. Autonomous agents and multi-agent systems, 1998, 1(1): 7-38

[4] Le T G, Fedosov D, Hermant O, et al. Programming Robots with Events [M] // Embedded Systems: Design, Analysis and Verification. Springer Berlin Heidelberg, 2013: 14-25

[5] Auyeung T. Robot programming in "C" [OL]. 2006-02-15 [2014-07-01]. http://wild-puppy.drtak.org/teaches/ARC/cisp299_bot/b-ook/book.pdf

[6] Kang S C, Chang W T, Gu K Y, et al. Robot Development Using Microsoft Robotics Developer Studio[M]. CRC Press, 2011

[7] Preston S. The definitive guide to building Java robots [M]. Apress, 2006

[8] Jayaram K R, Eugster P. Context-oriented programming with Event-Java[C] // International Workshop on Context-Oriented Programming. ACM, 2009: 9

[9] Baillie J C. Urbi: Towards a universal robotic low-level programming language[C] // Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05). 2005: 820-825

[10] Chinello F, Scheggi S, Morbidi F, et al. Kuka control toolbox [J]. Robotics & Automation Magazine, IEEE, 2011, 18(4): 69-79

[11] Logic Design Inc. Robologix[OL]. 2014-06-28[2014-07-01]. http://www.robologix.com/programming_robologix.php

[12] Cohen N H, Kalleberg K T. EventScript: an event-processing language based on regular expressions with actions[J]. ACM Sigplan Notices, 2008, 43(7): 111-120

[13] Holzer A, Ziarek L, Jayaram K R, et al. Putting events in context: aspects for event-based distributed programming[C] // Proceedings of the tenth international conference on Aspect-oriented software development. ACM, 2011: 241-252

[14] 张连新，高洪明，张广军，等. 混合式弧焊机器人编程语言[J]. 焊接学报，2006，27(7)：105-108

[15] 毛新军. 面向 Agent 软件工程：现状，挑战与展望[J]. 计算机科学，2011，38(1)：1-7

[16] Aldebaran Robotics. Nao software documentation[OL]. [2014-07-02]. https://community.aldebaran.com/doc/

(上接第 12 页)

[26] Viyanon W, Madria S K. A system for detecting xml similarity in content and structure using relational database[C] // Proceedings of the 18th ACM Conference on Information and Knowledge Management. HongKong, China, 2009: 1197-1206

[27] Li Pei, Dong X L, Maurino A, et al. Linking temporal records [C] // Proceedings of the 37th International Conferenceon Very Large Data Bases (VLDB 11). Seattle, Washington, USA, 2011

[28] 王宏志，樊文飞. 复杂数据上的实体识别技术研究[J]. 计算机学，2011，38(10)：1843-1852

[29] 杨丹，申德荣，于戈，等. 数据空间中时间为中心的集合实体识别策略[J]. 计算机科学与探索，2012，39(11)：1673-9418

[30] Papadakis G, Ioannou E, Palpanas T, et al. A Blocking Framework for Entity Resolution in Highly Heterogeneous Information Spaces[J]. IEEE Trans. Knowl. Data Eng. (TKDE), 2013, 25(12): 2665-2682

[31] Papadakis G, Ioannou E, Niederée C, et al. Efficient entity resolution for large heterogeneous information spaces[C] // WSDM 2011. 2011: 535-544

[32] de Vries T, Ke H, Chawla S, et al. Robust Record Linkage Blocking Using Suffix Arrays[C] // Proc. 18th ACM Conf. Information and Knowledge Management (CIKM). 2009: 305-314

[33] Jin L, Li C, Mehrotra S. Efficient Record Linkage in Large Data Sets[C] // Proc. Eighth Int'l Conf. Database Systems for Advanced Applications (DASFAA). 2003

[34] Baxter R, Christen P, Churches T. A Comparison of Fast Blocking Methods for Record Linkage[C] // Proc. Workshop Data Cleaning, Record Linkage and Object Consolidation at SIGKDD. 2003: 25-27

[35] Gravano L, Ipeirotis P, Jagadish H, et al. Approximate String Joins in a Database (Almost) for Free[C] // Proc. 27th Int'l Conf. Very Large Data Bases (VLDB). 2001: 491-500

[36] Kalashnikov D V, Mehrotra S. Domain-independent data cleaning via analysis of entityrelationship graph[J]. ACM Trans. Datab. Syst., 2006, 31(2): 716-767

[37] Nuray-Turan R, Kalashnikov D V, Mehrotra S. Adaptive connection strength models for relationship-based entity resolution [J]. Journal of Data and Information Quality, 2013, 4(2)

[38] Yakout M, Elmagarmid A K, Elmelegy H, et al. Behavior based record linkage[C] // Proceedings of VLDB. 2010

[39] Whang S E, Garcia-Molina H. Entity Resolution with Evolving Rules[J]. Proceeding of the VLDB Endowment, 2013 (1/2): 1326-1337

[40] Ramadan B, Christen P, Liang Hui-zhi, et al. Dynamic Similarity-Aware Inverted Indexing for Real-Time Entity Resolution[C] // Trends and Aplplications in Knowledge Discovery and Data Mining. Gold Coast Australia, Volume 7867, 2013: 47-58

[41] Altwaijry H, Kalashnikov D V, Mehrotra S. Query-Driven Approach to Entity Resolution [J]. PVLDB, 2013, 6(14): 1846-1857

[42] Kenig B, Gal A. MFIBlocks: An effective blocking algorithm for entity resolution[J]. Inf. Syst. (IS), 2013, 38(6): 908-926

[43] 王颖颖，黄杜英，许多顶. 向量空间中基于隐私保护的记录链接协议[J]. 现代电子技术，2009，32(14)：138-141