模式匹配研究进展①

潘 超 杨良怀 龚卫华 古 辉 陈敏智

(浙江工业大学 计算机科学与技术学院 浙江 杭州 310023)

摘 要: 随着网络和信息技术的发展，各个应用领域的合作越来越密切，数据的互操作性日显重要。由于数据源数据模式的自治性、异构性，为实现数据共享，模式匹配已成为数据密集型分布式应用的一项基本任务，成为学术界近年来的一个研究热点。对模式匹配的研究现状和趋势作了简述：介绍了模式匹配的基本技术及分类，分析并比较了典型的模式匹配系统，讲述了模式匹配的发展趋势。

关键词: 模式匹配; 数据集成;本体比对;模式映射

Schema Matching Research Progress: A Brief Survey

PAN Chao, YANG Liang-Huai, GONG Wei-Hua, GU Hui, CHEN Min-Zhi (School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: With the development of networks and information technology, cooperation among various applications is becoming more prevalant, and data interoperability is becoming increasingly important. Due to the autonomy and heterogeneity of data sources, the goal to achieve data sharing and schema matching has become a fundamental task of data-intensive distributed applications, a hot research issue in recent years. This paper surveys the status quo of schema matching, the basic technologies and classifications of schema matching, analysis and comparisons of some typical schema matching systems, and issues that still need to be addressed.

Keywords: schema matching; data integration; ontology alignment; schema mapping

1 引言

随着信息技术发展，各个应用领域产生了大量的数据，各领域的高度自治性导致了数据模式的异构性。另一方面，Internet 的发展使得各应用领域的合作越来越密切，数据的互操作性日显重要。为实现异构数据源的共享，其核心是数据集成。在数据集成中，集成系统为了将在全局模式上构建的用户查询重构为针对数据源模式的查询，需要用一种机制来表示数据源模式和全局模式之间的关系，有两种典型机制：GAV(Global As View)[1]和 LAV(Local As View)[2]，GAV 将全局模式作为数据源模式的视图，是以全局模式为中心的机制。LAV 则是将数据源作为全局模式的视图，是以数据源为中心的机制。集成的途径可以有：通过合成、扩展、特化或改造已有模式来重建新模式；把多个模式合并成统一的单个模式.不论采用哪种机制，都需要在全局模式和数据源模式之间建立映射关系，发现两个模式成员之间语义上的对应关系的操作即模式匹配(Schema matching)。

1.1 模式匹配概念

模式(schema)是指具有某种结构的元素的集合，用于表示数据的组织结构。通常所说的模式有：数据库模式(关系模式、面向对象模式)、XML 模式、本体(Ontology)等。

映射(mapping)是两个模式中有特定关系的规则

① 基金项目:浙江省自然科学基金(Y1090096,Y1080102)
收稿时间:2010-02-22;收到修改稿时间:2010-03-22

集合，表示一个模式中某些特定的元素与另一个模式中某些特定的元素的对应关系。一个映射关系包含两个部分：映射的元素和元素之间的关系的描述。Euzenat[3]把一个映射元素 M 定义成一个 5 元组：

$$M=(eid, e, e', c, R),$$

其中 eid 是给定映射元素唯一标识符；e 和 e’分别是第一个和第二个模式/本体中的实体，如可以是表、XML 元素、特性、类等；c 是 e 和 e’之间对应程度的一个数学置信度；R 表示 e 和 e’之间存在的关系(如相等、泛化、不相交、相交)[4]。

模式匹配是指给定的两个模式，利用一些相关信息，找到分布在两个模式中的元素之间的某种映射关系(语义对应关系)。模式匹配将两个模式作为输入参数，其输出结果是它们之间的映射关系，即匹配结果；匹配结果中的每个元素都表示一个输入模式中的某些元素和另外一个输入模式中的某些元素存在的逻辑上的对应关系。模式匹配过程可以用一个函数 f 来表示：

f：(S, S’, A, p, r)→A’　或写为 A’ = f(S, S’, A, p, r)，其输入参数是：(1)待匹配/比对(alignment)的两个模式/本体 S、S’；(2)一个待完成的匹配/比对 A；(3)匹配/比对算法中用到的需要人为设置的参数集 p，如权重系数、阈值等；(4)需要用到的外部资源 r：A’ = f(S, S’, A, p, r)。如图 1 所示。

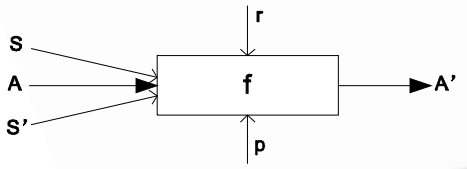

图 1　模式匹配过程的函数表示

模式匹配的关键是寻找匹配方法。理想的匹配方法是能够自动、精确、广泛适应地匹配不同的模式。然而，匹配方法难以用数学公式或者数学方法来对两个模式之间的对应关系进行准确计算，只能利用模式本身具有的结构、所蕴涵的语义以及该模式的实例数据等信息来寻找二者之间的对应关系。

模式匹配在传统的应用中是一项重要的操作，如信息集成、数据仓库、分布式查询处理等。现在模式匹配更显重要，模式匹配已几乎成为每个数据密集型分布式应用的一项基本任务。涉及的应用包括企业信息集成、电子商务、web 服务协同、基于本体的代理通信、web 目录集成以及基于模式的 P2P 数据库系统。在数据集成中用于识别模式之间的相互关系；在数据仓库中用于发现数据源模式与数据仓库模式之间的映射关系，以完成对数据源数据的抽取和转换；在电子商务中用于不同消息模式的转换；在语义网(semantic web)中用于建立不同本体概念之间的语义对应关系；在 XML 数据聚类中用于确定 XML 数据之间的语义相似性等等。此外，在语义查询处理、深网(deep web)[5,6]的查询接口集成、数据抽取、实体识别、结果合并等方面中也有模式匹配的需求。有关应用的详细信息我们将在后续小节介绍。模式匹配已经成为以上应用领域研究和开发的基础。因此，对模式匹配问题展开深入研究，有着重要意义。

近年来模式匹配成为一个研究热点，提出了一系列的算法和技术。当前模式匹配研究所用的方法涉及多个领域，包括机器学习、本体推理、数据库模式、语言学等。问题的关键在于利用语法特性、语言线索、以及结构相似性。

本文对模式匹配的研究进展和现状进行了论述，并指明了进一步研究的方向。对于模式匹配这个问题，可以从多个角度来考察。我们在引用原文时，在合适时会加以评论，以进行各种模式匹配方法的比较。因此，其中必带有某种程度的主观性。文章的组织如下：第二节介绍了模式匹配技术，包括技术分类、基本技术、匹配策略；第三节介绍模式匹配系统，并在第四节对它们进行了比较；最后一节作了总结。

2　模式匹配技术

据前述，模式匹配的任务是寻找两个模式的元素之间映射关系(语义对应关系)。由于模式匹配的复杂性，模式匹配需要使用各种技术来弥补信息的不足，如利用名字相似性、字典、公共模式结构、相交的实例数据、公共值分布、重用过去的映射结果、约束、与标准模式的相似性、常识推理。迄今，已提出了许多匹配方法和模式匹配系统。例如，Cupid[7]，Similarity Flooding(SF) [8], COMA/ COMA++[9,10], LSD[11], OntoBuilder[12], S-Match[13], SPICY[14]等。

大部分模式匹配系统采用基于规则的方法和基于机器学习的方法。基于规则的方法一般采用某种数据模型表示模式，例如模式树或者模式图，通过利用成员的名称、数据类型、结构等模式信息来指导匹配过

程，该方法通常要对模式树或者模式图进行多次遍历。基于规则的模式匹配过程主要包括三个部分：预处理，相似度计算，映射生成。预处理是用数据模型来表示模式的过程；在相似度计算中，通过计算成员之间的相似度进而计算两个模式之间的相似度；最后根据匹配算法来进行模式匹配并生成映射关系。

基于学习的实现方式采用机器学习的方法进行匹配。例如，SemInt 系统提出了一种基于神经网络的模式匹配方法；Automatch 系统提出了基于贝叶斯学习和特征选择的模式匹配方法；LSD、COMAP 和 GLUE 等设计了一种三层结构的多策略学习(multi-strategy learning)框架。

2.1 模式匹配技术分类

目前模式匹配方法很多，不同的研究人员对匹配方法的分类有不同的分法[16-20]。Rahm 与 Bernstein 在[15]中把匹配方法分为简单匹配方法和复杂匹配方法,如图 2。

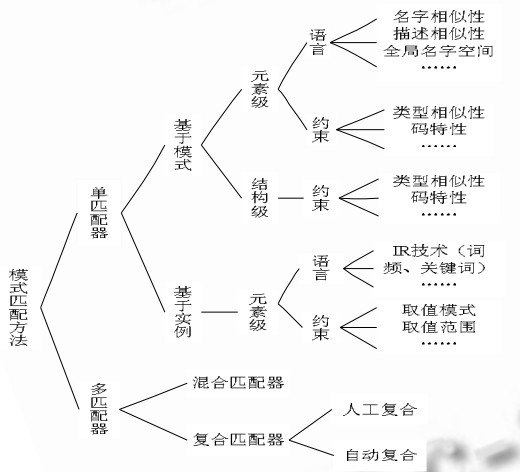

图 2 模式匹配分类

为了给单匹配器分类，Shvaiko 和 Euzenat[21] 做了两种综合的分类，如图 3 所示。类似两棵树共享他们的叶节点，叶节点代表单匹配器的类别及具体实例。该分类方法较 Rahm 与 Bernstein 的分类方法更为详细，增加了新的分类，在图中用黑体体现出来。这两种综合方法是：(1)粒度/输入解释分类：该分类方法依据匹配粒度(元素级、结构级)和用于解释输入信息的各种技术；(2)输入类型分类：依据各匹配技术所用输入类型分类。

在图 3 中，对单匹配器模式匹配方法按如下方式进行分类[21,15]：(1)模式与实例：基于模式的匹配方法仅仅考虑模式的信息，而没有利用实例数据。可利用的模式信息包括模式元素的一些属性，如元素名、描述、关系类型、约束和模式结构等。基于实例的匹配方法利用了实例级数据，利用元数据和统计数据对数据实例特征进行提取分析，这个过程中常用机器学习的方法。实例级数表明了模式元素所表示的内容和含义，在可用的模式信息非常有限的情况下，显得尤为重要，借助实例数据可以手工或自动地构造模式。(2)元素与结构匹配：元素级的匹配方法是单独对模式实体或实体的实例进行分析来计算对应关系，忽略了这些实体之间存在联系。对于第一个输入模式的每个元素，基于元素的匹配方法在第二个输入模式中确定其对应的匹配元素。在最简单的情况下，仅考虑最底层元素，也叫做原子层，如 XML 模式中的属性或关系模式中的列；元素级匹配方法也可应用于高层(非原子层)元素，包括文件记录、实体、类、关系表和 XML 元素。元素级匹配方法主要有基于字符串的方法、基于语言学(自然语言)的方法、基于约束的方法、语言资源(字典、专业领域词典等)、比对重用方法、顶层本体和领域本体方法。和元素层匹配方法不同，结构层匹配方法通过分析实体或其实例如何一起出现在一个结构中来计算他们对应关系。主要方法有基于图的方法、基于分类的方法、结构库(用于粗筛)、基于模型的方法、数据分析和统计学方法、机器学习方法等。(3)语言与约束：基于语言的匹配应用名字和文本(如单词或句子)来挖掘语意上相似的模式元素。主要方法有基于名字的匹配和描述匹配。基于名字的匹配是通过等价或相似的名字来匹配模式元素的。而描述匹配是通过对模式语言上的描述来确定模式元素之间的相似度。模式对于定义数据类型、值的取值范围、唯一性、可选性等通常都会有一些约束。基于约束的匹配方法可以有助于限制候选匹配的数量来提高匹配的精确度。(4)匹配基数：匹配基数指明实体集中的一个实体能同另一个实体集相关联的实体数目。通常有四种情况：1:1，1:N，M:1 和 M:N 匹配。(5)辅助信息：大多的匹配器不仅依赖输入模式，还依赖辅助信息，比如数据字典、已知的匹配结果和用户输入等。(6)合成的匹配技术主要有两种方式：一种是混合匹配器(Hybrid matcher)，它基于多个标准和信息源，综合了多种匹配技术来确定匹配候选；另一种是复合匹配器(Composite

Matcher)，它合并了每个匹配器独立执行的匹配结果(包括混合匹配器)。

另外还有其他一些分类方法，Ehrig[19]提出了基于二维正交分类法。横轴包含三层：第一层为数据层，实体的匹配只考虑数据值；第二层为本体层，该层进一步分为四层：语义网，描述逻辑，限制和规则；第三层为语境层(Context layer)，该层涉及一个应用情景中实体的使用。竖轴表示具体的领域知识，对应横轴上的任一层。

Doan 与 Halevy[20]把匹配技术分为基于规则的和基于学习的。基于规则的技术主要考虑模式级的信息，比如，如果实体名称(/数据类型/结构)相似或有相同数目的近邻则两个实体匹配；基于学习的方法主要考虑实例级的信息，如，比较待匹配实体所对应数据实例的值格式和分布。基于学习的方法也可以利用模式级的信息和已有的匹配结果，类似 LSD 中提出的方法。

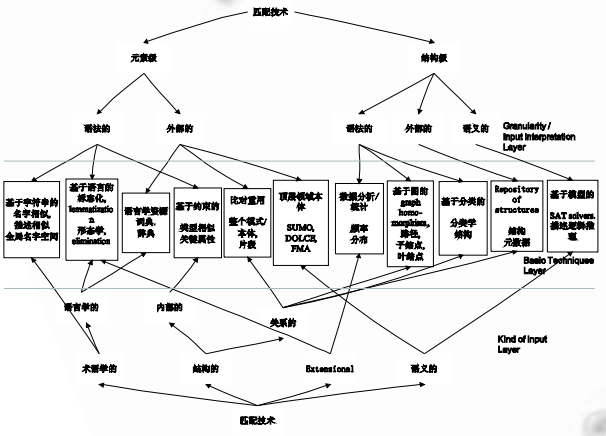

图 3　单模式匹配方法

Zanobini[18]根据代理间通信含义的认知理论把匹配方法分为三类：(1)语法的：这类代表那些仅仅使用语法匹配的匹配方法。包括基于字符串的方法(字符串的编辑距离)和图匹配技术(树的编辑距离)；(2)语用：该分类代表及依赖于数据实例比较的方法，这类方法包括自动分类器，比如贝叶斯分类器和形式概念分析法；(3)概念的：该分类代表利用概念和含义来计算比对，如利用外部词典 WordNet。

2.2 基本技术

模式匹配的目标是找出模式中实体之间的对应关系。通常，这些关系通过实体之间的相似度来发现。这里介绍计算实体之间相似度和发现实体之间关系的基本方法。

(1)基于名称的方法：通过术语字符串的比较计算名称、标签以及实体注释的相似度。比较模式实体名称及标签之间相似性的主要问题是存在同名异义、异名同义的问题。主要有两类方法来比较术语：基于字符串的方法和基于语言学知识的方法。基于字符串的方法主要利用字符串的结构，把字符串看作字母序列，常用的方法主要有：规范化(大小写、去除读音符号，如é→e，空白规范化、去除连接符、数字、标点符号)、编辑距离、路径标签序列的相似性。基于语言学知识的方法把字符串看做字符序列，从文本中提取有意义的术语以及注释。

(2)基于结构的方法：基于结构的方法主要考虑实体内部结构以及关系结构。内部结构，如名称注释、特性及数据类型等，也即实体本身的定义；关系结构即各实体之间的关系。基于内部结构的方法通常也被称为基于约束的方法。这些方法主要基于实体的内部结构，利用实体的属性集、属性范围、集的势或者多重性、属性的传递性和对称性计算实体之间的相似度。常用的有属性比较和关键字、数据类型比较、域比较、多重性和属性比较。基于内部结构的方法易于实现但不能够提供足够的信息，比如不同类型的对象具有相同数据类型的属性，它通常组合其他的一些方法一起使用。基于关系结构的方法目前主要考虑了三种类型的关系：分类关系，如 is-a 子类关系；整体—部分关系，如 part-of 关系；所有其余可能涉及的关系。

(3)外延技术：外延技术主要是基于个体实例的匹配方法。大致可以分为三类：使用共同实例集、实例识别技术、以及基于实例集的异质性统计法、基于相似度的外延比较方法等。

(4)基于语义的方法：语义方法主要特征是采用模型论语义(model-theoretic semantics)来判断结果，因此是演绎的方法。通常使用的语义方法有命题可满足性，模态可满足性技术，以及基于描述逻辑的方法。目前基于语义的方法并不多，该方法面临的一个主要挑战就是集成这些演绎方法，计算比对和发现不一致性是其中一个关键步骤。

2.3 匹配策略

为了提高匹配的质量，通常不只是采用某一种匹配技术，而是采用一定的匹配策略或者匹配策略的组

合，下面介绍常用的匹配策略。

匹配器组合：基本匹配器组合的一种自然方式是利用顺序组合来改善匹配，举个例子，可以首先使用基于分类的匹配器，然后在使用基于实体结构的匹配器或者语义匹配器；另外一种组合匹配器的方式是独立使用不同的匹配算法以并聚集其匹配结果，通常称为并行组合。举个例子，在进行匹配的时候并行采用几个匹配算法，选择它们都具有的那些对应，或选择它们最高信任度的那些对应。使用该策略的典型的匹配系统有 COMA[9,10]，AgreementMaker[22]。

相似性聚集：复合相似性关注的是相似的异构聚集。结构化对象通常涉及不同的关系，如果相关的实体之间的相似度是可计算的，则为了评估实体间的相似性，必须对得到的相似度进行汇总。

全局相似度计算：复合相似度计算是局部的，因为它只考虑邻居节点的相似性。但相似性可能涉及到整个模式，最终的相似度取决于所有实体。因此，匹配策略要考虑全局的相似性。全局相似性计算大致有两种方法：一是基于图的相似度传播，典型代表系统为相似性洪泛；另一是把相似度定义转换为一组方程，采用数值分析方法求解。

学习方法：通常利用一些实例对匹配实体进行分类，这些实例需要一些样本数据来学习，这些样本数据可以由算法本身提供，或者由用户提供。机器学习通常有两个阶段：学习(训练)阶段和分类(匹配)阶段。常用的机器学习方法有贝叶斯学习法、WHIRL学习法、神经网络、决策树和叠加学习(stacked generalisation)。

概率方法：同机器学习一样，在匹配过程中也常用概率方法。常用的基于概率方法是贝叶斯网络。

用户参与和动态组合：在设计匹配系统结构中，用户的参与是很有用的，也是很必要的。匹配过程中有三个方面用户可以参与：提供最初的比对和参数、动态的组合匹配器和给匹配器提供反馈信息以得到更好的匹配结果。

比对抽取：匹配的目的是得到实体之间合适的对应关系，通过匹配算法通常可以得到大量的对应关系，还需根据相似度进行过滤提取，如通过对相似度矩阵或已提取的比对等进行进一步的提取，这可采用阀值来截取。

由于模式匹配的复杂性，为了取得更好的匹配效果，通常采用组合匹配器的方法或者采用多种算法结合。目前越来越广泛采用组合匹配策略的方法，在匹配精确度和查全率等方面的实验结果表明该方法具有较好效果。AgreementMaker 是最近提出的一个典型的组合匹配系统，它组合了多种匹配器，包括元素级和结构级、模式级和实例级等。

3 模式匹配系统

本节介绍学术界在模式匹配方面构建的一些模式匹配原型系统[21, 15, 23, 24, 8, 7, 25, 26, 27, 28, 40]。本节将根据前面介绍的分类方法，对这些系统进行简明的介绍，主要围绕模式级匹配系统、实例级匹配系统与混合匹配系统这几方面展开。

基于模式层的典型匹配系统有 DELTA[29]、Hovy[30]、TranScm[31]、Cupid、DIKE[32]、SKAT[33]、H-Match[34]、Anchor-PROMPT[35]、COMA、XClust[36]、ToMAS[37]、Similarity Flooding(SF)、S-Match[13]等；基于实例层的典型匹配系统有 TIQS[38]、LSD、GLUE[39]、iMAP[40]、SBI[41]、Automatch[42]等；混合模式匹配系统典型代表系统有 SemInt[43]、OLA [44]、RiMOM[45] 、Clio[46]，AgreementMaker[22]等系统。除了这几方面的匹配系统，当然还存在其他方法的匹配系统。如 A.Nandi[47]提出了一种不同于传统的基于模式和基于实例的方法，其观点是在两个模式中如果通过关键字查询点击它们实例的分布相似，则认为这两个模式元素匹配。下面重点介绍适用于 XML 模式匹配的系统。

3.1 基于模式级的匹配系统

3.1.1 OntoBuilder

OntoBuilder 是万维网上的一个信息检索系统。它本身是一个本体产生和提取工具，其中包含了匹配的过程，主要应用于本体匹配，也可应用于 XML 模式匹配。

OntoBuilder 方法分为两个阶段：本体产生(训练阶段)和本体改编(改编阶段)。在训练阶段，首先从 web 站点上提取需要的信息创建本体；改编阶段包括即时匹配和对相关的本体与初始本体的交互式合并操作。在这个阶段，用户提出需要进一步浏览的 web 站点，每个 web 站点进行本体提取，产生候选本体，并被合并到实际本体中(初始本体)。在匹配过程中，采用了多种匹配器，同时在匹配中还实用信息检索技术。不匹配的结果返回给用户手动选择。

OntoBuilder 特点是：可以自动的实现本体创建和匹配过程，适用于动态网页；缺点是：只采用简单的字符串等匹配方法，匹配的精确度不是很高，在本体提取的过程中需要用户的参与，不支持复杂的匹配。

3.1.2 Cupid

Cupid 是一个基于元素级和结构级的匹配系统，属于一种通用的模式匹配器，主要应用模式是实体关系模型和 XML。Cupid 把名字匹配算法与结构化匹配算法相结合，根据这个结构化算法可以推导出元素的相似度，而元素的相似度是根据元素的成分(主要是元素名字和元素的数据类型)的相似性得出的。Cupid 强调名称和数据类型的相似度。它采用一种基于元素级匹配和结构级匹配的混合型算法。其主要思想是：如果两个元素的子元素是相似的则两元素就趋于相似，如果两个元素相似则其子元素也趋于相似。

整个算法分成三步。第一步做语言学上的元素级匹配，并通过名称、数据类型和领域进行分类。第二步，把原来的模式转化为模式树，采用树匹配算法，做自底向上的结构匹配。两元素之间的相似性取决于它们的语言相似性以及它们的叶子集的相似性。这一步计算出匹配概念对之间的语言相似系数和结构相似系数的加权平均值。在转换过程中同时考虑域约束。第三步，用这些加权平均值来选出匹配结果。Cupid 匹配基数是 1:1 和 n:1，可以做复杂匹配。

Cupid 的特点是：可以作为一个通用的模式匹配方法，可应用于多个领域和数据模型，可以进行复杂匹配；在模式中提取了比较全面的信息包括名称、数据类型、约束和结构信息，相比其它匹配方法包含了更全面的语言学匹配，偏重于匹配模式的叶子节点。可进一步改进的方面：自动调优控制参数，利用模式中的注释信息以及在系统中集成词典。

3.1.3 COMA

COMA 是德国莱比锡大学开发的一个通用自动模式匹配系统。主要应用于 XML 模式匹配。COMA 提供了一个匹配库，它由多个不同的模式级匹配器组成，支持不同的匹配算法。它所应用的匹配器主要利用模式信息，如元素和结构属性。与其他系统不同的是它可以重用以前的匹配结果，可以显著地提高匹配效率。在匹配过程的不同阶段，COMA 应用了不同的组合策略。在 COMA 的匹配处理流程中，输入模式首先通过用户模块来接受用户提供的自定义匹配和非匹配信息(可选步骤)把模式表示为有向无圈图，然后在匹配库中由三类不同的匹配器对输入模式分别独立执行匹配操作。这三类匹配器是：简单匹配器、混合匹配器、面向重用匹配器。分别利用不同的模式信息进行匹配操作，得到的匹配结果是位于[0,1]区间的相似度值.这些中间匹配结果存放在一个相似度立方体(similarity cube)中。最后系统将这些匹配结果合成后输出，得到最终的匹配结果。合成匹配结果的过程由两个子步骤组成：集成匹配器输出结果；选择匹配候选对象。

COMA 的特点：(1)COMA 可以作为一个通用的自动模式匹配方法，应用于多个领域和数据模型；同时，COMA 还可以提供用户反馈信息来不断提高匹配精确度；(2)COMA 合成了多种匹配算法；(3)该算法可以重用前面的匹配结果，提高了匹配的精确度；(4)COMA 可作为检查和比较不同匹配器和组合策略的一个评价平台。在 COMA 中，由于采用组合的匹配器和重用策略，其最佳匹配器的平均精确度可以达到 95%，平均召回率可以达到 80%，平均查全率可以达到 75%；

COMA 匹配基数是 1:1。在实际应用中，1:1 的匹配只是所有匹配当中的一部分，因此需要应用 COMA 匹配思想来开发新算法，从而提高匹配的精确度，并进一步实现对复杂匹配的挖掘。COMA 可以通过更广泛的匹配候选策略和结合实例数据以及词典等方法改进原有工作。

3.1.4 相似性洪泛法(Similarity Flooding)

相似性洪泛(SF)算法是斯坦福大学的 Melnik 等人提出的一个图匹配算法，主要应用于实体联系模型和 XML 模式匹配中。

SF 是一种基于模式结构相似度的匹配方法。它的主要思想是依据数据库模式中的列属性和数据类型进行迭代不动点计算得到匹配结果。SF 算法首先把模式转换成有向标记图，通过不动点计算对图中的这些节点进行多步迭代后，返回一个图中节点和另一个图中节点的稳定相似度，来决定两个图中的节点的匹配关系。最后通过过滤器筛选掉不正确的匹配候选对象。SF 也是一种综合使用了名称匹配和结构匹配的混合方法。其匹配基数是 1:1。

SF 算法的特点：它考虑了模式结构特征对模式匹配过程的影响，采用不动点计算和相似度传播的手段，把相似度传播的概念引入到模式匹配领域中。缺点：

SF 当列名不同，数据类型都相同时，会产生很多不确定的候选匹配对；当列名不同，数据类型也不同时，不会得到逻辑上应该匹配的候选匹配；该方法不能自动的选择，只能人为干预，并且 SF 只能发掘 1:1 的匹配。

3.1.5 XClust

XClust 中介绍了一个 XML 模式聚类方法，对于给定的模式依据相似度进行聚类。XClust 分为两个阶段：相似度计算和聚类。在 XClust 中，模式首先被转换成模式树的表示方法，为了寻找在结构和语义上相似的 XML 模式(DTD)， 提出了一种基于模式成员名称匹配、近邻匹配和叶节点上下文匹配的相似度算法，对应得到的三种相似度加权合成后得到模式间的相似度。根据模式间相似度大小实现对 DTD 模式的聚类。

XClust 的特点：XClust 是针对 XML DTD 进行匹配和聚类的，但是也可以用于 XML Schema，是自动模式匹配的一个成功延伸应用，对大规模的 XML 数据集成很有效；该方法不仅考虑了元素在语言学和结构上的匹配，还考虑了元素的上下文信息。其匹配基数是 1:1。

3.1.6 ToMAS

ToMAS 是一个可以检测和改编模式中映射的不一致性和无效性的自动工具。ToMAS 主要用于处理关系模式和 XML 模式。

ToMAS 假设匹配步骤已经执行和对应关系也已经存在，检测映射受到结构和约束变化的影响。它把两个模式和至两个模式之间的映射集作为输入，该系统主要有两个部分：首先，作为预处理步骤，分析映射，当映射不存在时转化为逻辑上有效的映射；其次，通过模式变换来维护预处理步骤的结果。ToMAS 最终得到的结果是与模式的结构和语义一致的适合的映射集。

ToMAS 的特点：ToMAS 是一个模式变化时维护映射一致性的工具，它同时考虑了源模式和目标模式得模式结构信息和语义信息的改变对映射的影响，不仅考虑原子元素的变化，还考虑了复合结构的变化，可以应用于多个领域。

3.1.7 MapOnto

MapOnto[48]是一个用于构建本体模式、关系模式或者 XML 模式之间的复杂映射的半自动系统。该系统的适应场景与 Clio 相似。当目标模式是被当作一元或二元关系表的关系模式的本体时，该系统可以说是 Clio 的扩展。MapOnto 的输入有三个参数：以本体表示语言描述的本体、关系模式或 XML 模式、简单的对应关系，比如 XML 属性和本体数据类型属性。输入的模式被转换为标签图的模式，然后寻找图之间合理的联系。系统以半自动的方法在产生复杂的映射规则，规则采用 Horn 子句(一阶逻辑子集)来表示，通过工具对逻辑规则列表进行排序，给出最合理的映射，最后查看列表并选择最好的映射。

MapOnto 的特点：为关系表和本体之间的简单对应的语义映射推理提出了启发性算法；算法依赖于数据库模式(关键码和外部关键码结构)和本体(集的势限制、is-a 结构)的信息；理论上只要是关系模式是通过标准数据库设计原则产生的，该算法可以推出所有相关的语义。可以改进之处：通过增加外部资源，如更丰富的本体信息、实际数据、语言学和语义关系等，来用于精炼算法。在实际中，为产生较全面的映射需要研究一些复杂的对应关系。

3.1.8 S-Match

S-Match 是 Giunchiglia 等人提出的一个模式语义匹配系统，主要应用于实体联系模型、本体和 XML。

S-Match 系统的输入是两个图结构模式，如分类、XML 模式或本体，输出则是图中节点间的语义对应的逻辑关系。这些对应关系被定义为五种：等价关系(=)，包含(⊇)，包含于(⊆)，不匹配(⊥)，相交(∩)。这些关系通过以下方式来决定：把本体实体表示为逻辑公式；把匹配问题归结为命题有效性问题。

语义匹配算法主要有四个步骤：对于两颗输入树中的标签，计算对应标签的概念；对于两颗输入树中的结点，计算节点对应的概念；对两棵树的所有标签对，计算标签所表示概念的语义关系；对于所有结点对，计算结点所表示概念的语义关系。

S-Match 的特点：1)S-Match 是一个混合语义匹配系统，组合了多种匹配器，包括成员级的和结构级的匹配器；2)S-Match 可作为语义匹配的一个平台，即高度模块化系统，系统中单独的模块可以根据需要进行装卸和定制；3)其核心是寻找模式之间的语义关系。其匹配基数是 1:1。

3.1.9 QMatch

QMatch[49]是一个混合模式匹配算法，主要用于 XML。QMatch 算法提出了一个基于路径的方法，依赖 XML 模式的语义和结构信息，使用 XML 固有的约

束。在 QMatch 中，一个 XML 模式树是和一组路径同义的，结点通过以期作为始点出发的路径定义，树则是起源于根结点的路径的集合。QMatch 中定义了一组分类器，对标签、属性集、路径长度和路径覆盖的匹配进行分类。QMatch 算法则定义在这些分类器之上，结合标签、属性集、路径长度和路径覆盖之间的语义和结构信息，确定两棵 XML 模式树之间的匹配关系。

QMatch 的特点：QMatch 作为一个混合匹配算法为分析和利用 XML 模式内在的语义和结构信息提供了一个独特的框架；该方法一个显著的特点就是是基于路径信息的匹配算法。虽然该算法也是基于语言学和结构的算法，但是都是组合的算法，并不是单一的，这是不同于其他方法的一个地方。

3.1.10 Spicy

Spicy 是一个验证映射质量的方法，它有三层结构，其中模式匹配模块用于给映射集成模块提供输入。映射验证模块用于检查候选映射并选择其中较好的，该系统提出了结构分析法，结构分析发法使用了电路来比较拓扑结构和树结构的信息内容来得到相似度。

模式匹配模块可以组合其它的匹配法来取得较好的匹配结果，该算法在匹配过程中不仅采用模式级的匹配方法，同时也采用了实例级的匹配方法。在进行模式级匹配时，用于比较的两个树结构是独立的，没有共同的数据结构，每个树被转换为同构的电路，通过选择描述电路的特征以及测量他们的距离来计算相似度。在实例级匹配时并不需要额外的输入，也不需要经过训练，而是基于电路。

Spicy 主要是一个验证映射质量的方法，它可以处理 XML 模式或 OWL 本体，在系统中包含模式匹配的模块，虽然在匹配的时候考虑了多对一的元素对应，但是在进行匹配的时候并没有考虑上下文关系等较复杂的关系，采用的匹配方法比较简单，但是它可以结合现存的模式匹配和映射集成系统，使系统更加完善。

3.1.11 PORSCHE

PORSCHE[50]是一个混合的自动的模式映射方法，可以进行复杂匹配。该方法可以自动发现 XML 模式的语义模式匹配，增量地为所有模式树建立一个集成模式，定义各源模式到集成模式的映射。PORSCHE 利用整体方法，把模式转换为标签树模式。该算法分为两步：首先，使用一个语言匹配器集，提取输入模式中节点标签的语义，并对语义上相似标签进行聚类；然后，使用树挖掘技术，联合相似标签聚类，计算每一个节点的上下文关系。

PORSCHE 的主要特点：可以自动、灵活地组合其它模式匹配技术或算法；在产生匹配的同时进行模式的集成；可以处理大规模的模式；采用树挖掘技术来进行模式匹配。可以改进的地方是加强系统中语言匹配部分技术。

3.1.12 本体辅助的自动化模式匹配技术

本体辅助的自动化模式匹配技术[51]在基于映射的数据交换系统框架下，提出了一种本体辅助的模式匹配方法，主要应用于数据库模式和 XML。它利用 WordNet 词汇和决策树学习相结合的方法进行属性名称匹配，构建数据类型本体计算属性数据类型的语义距离，依赖领域本体发现一对多的语义匹配关系，这 3 个过程逐步提高了匹配质量。在实际数据上的实验结果表明，该方法具有较高的精确度，可以实现一对多映射。

该方法的优点是结合了本体和决策树学习方法对属性名称进行匹配，大大提高了匹配的精确度。可以改进的地方是在进行匹配的时候考虑完整性约束、用户反馈以及已有的映射关系来帮助更好的匹配。

3.1.13 XPrüM

XPrüM[52]是基于Prüfer编码的XML模式匹配系统。XPrüM 主要包括两个部分：模式准备和模式匹配。首先，解析 XML 模式,把 XML 模式转换成模式树，对每一个模式树构建一个 Prüfer 序列，使用这些序列构建模式的表示序列。使用标签 Prüfer 序列(LPS)表示模式树的语义信息，使用数字 Prüfer 序列(NPS)表示模式树的结构信息。然后，提出一个使用 LPS 和 NPS 的新的模式匹配算法。在语言匹配器中使用 LPS 计算术语之间的相似性，包括原子元素和复杂元素中的术语；在结构匹配算法中使用 NPS，通过计算相似性得到不同模式的节点之间兼容性关系，最后通过节点兼容性得到元素之间的对应关系，完成匹配过程。

XPrüM 的特点：采用 Prüfer 序列；自动匹配；与待匹配模式的数据模型和应用领域无关。可以改进的地方是：该系统没有使用任何外部字典和本体，本体和字典的使用在一定程度上可以提高匹配精确度。

3.2 基于实例级的匹配系统

LSD 系统和它的扩展 GLUE 是华盛顿大学研制的一个自动模式匹配系统,其应用模式是 XML 模式。

LSD 使用机器学习的方法来对两个模式的成员进行匹配，是最具代表性的基于机器学习方法的自动模式匹配系统。在 LSD 中使用了多种学习器：名称学习器、内容学习器、Naïve Bayes 学习器、XML 学习器等。对模式中的每个成员，这些学习器(匹配器)通过训练产生各自的相似度预测值，然后通过一个元学习器合成这些学习结果，并根据训练中各匹配器显示的准确度来设定各匹配器的权重。由此得到的匹配结果再经过域约束检验和用户检验，产生最终的匹配结果。LSD 是一个典型的多策略基于实例的匹配系统，亦可应用于模式级的匹配：把模式描述信息(比如模式成员标记)作为学习样本进行学习而忽略模式数据实例信息，即进行模式级的匹配。

GLUE 系统用机器学习的方法来完成不同模式之间的匹配任务，其思想是多策略学习。它代表了一种自动合并不同匹配器匹配结果的组合方法，产生的是原子级的 1:1 的映射关系。除了名称匹配器之外，它还用到了几个在预处理阶段经过训练的实例级匹配器。在预处理阶段，用户先给出一些映射实例，然后用这些实例训练各学习器，发现其中特有的实例模式和匹配规则。用这些模式和规则去匹配整个模式，得到候选值的列表。虽然此方法是面向实例的，但它也能利用模式信息。此外，该系统可以扩展利用用户提供的领域约束信息以提高匹配准确性。

LSD 和 GLUE 的特点：这两种系统是实例级的匹配系统；采用机器学习的方法；基数是 1:1。缺点是不能够进行复杂匹配。

3.3 混合匹配系统

3.3.1 Clio

Clio 系统是 IBM 的 Almaden 研究中心和多伦多大学开发的一个半自动模式匹配系统，其应用模式是实体联系模型和 XML。Clio 目标是在给定的目标模式

表 1 匹配系统的比较

系统	是否手动	元素层	结构层	外部资源
OntoBuilder	手动	基于字符串、基于语言		词典
Cupid	自动	基于字符串、基于语言、数据类型、关键属性	树匹配 叶节点权重值	词典
COMA COMA++	手动	基于字符串、基于语言、数据类型	DAG/树匹配	词典、比对重用、结构
洪泛方法SF	手动	基于字符串、数据类型、关键属性	迭代不动点计算	
XClust	自动	集的势、约束条件	路径、叶节点、孩子节点、聚类	WordNet
ToMAS	自动		保持一致性、结构比较	外部比对
MapOnto	半自动		结构比较	外部比对
S-Match	自动	基于字符串、基于语言		WordNet
LSD/GLUE	自动	WHIRL	层级结构	域约束
Clio	半自动	基于字符串、基于语言、朴素贝叶斯分类	结构比较	
Xu & al.	自动	基于字符串、基于语言	决策树	WordNet、域约束
本体辅助	自动	数据类型、特性	决策树	WordNet、领域本体
QMatch	自动	约束条件	基于路径的	
Spicy	自动	属性	结构分析，集成	已有的模式匹配
PORSCHE	自动	基于语言	树挖掘数据结构、自顶向下分层结构、聚类	特定领域用户定义缩写表及同义词表
XPrüM	自动	基于语言、数据类型、属性	路径、孩子节点、叶节点、匹配细化	

和新数据源模式之间建立起半自动的匹配关系，用户有效地参与其中。Clio 由模式读取器(SR)、对应引擎(CE)、映射产生器组成(ME)。SR 读取模式并将其翻译成内部表达形式；CE 识别这些源—目标模式之间的匹配部分；ME 产生视图定义，建立源模式中的数据和目标模式中的数据映射关系。

Clio 的特点：Clio 是同时包含模式级和实例级匹配的半自动匹配系统；其匹配基数是 1:1 和 n:m，可以进行复杂的匹配。

3.3.2 Xu and Embley

Xu 与 Embley[38]提出了一个并行组合方法，用于发现图结构模式之间一一对应、一对多和多对多的对应。该方法组合了多个匹配器，采用领域本体作为辅助。该方法中用到的元素层的匹配器包括名称匹配器和值特征匹配器。名称匹配器除了进行字符串比较，还做语言学的处理，同时还利用 WordNet 来发现节点名称的同义关系，匹配规则通过决策树 C4.5 产生器获得。特征值匹配器决定两个模式元素值是否享有相似的特征值，如数值元素的均值和方差；和名称匹配器一样，匹配规则通过决策树 C4.5 产生器获得。结构层匹配器用于提出新的对应关系和确认元素层匹配器得到的对应关系,例如考虑元素级匹配器计算得到的相邻元素的相似度。

Xu 与 Embley 的特点：同时包含模式级和实例级的自动模式匹配系统；虽然该算法的精确度相对比较高，但是还有许多需要做，比如使用半自动方法构建领域本体达到期望的值；自适应参数调整等。

4　模式匹配系统比较

前面几节介绍了模式匹配背景、概念、匹配方法分类、基本技术以及常用 XML 模式匹配系统，同时介绍了各个匹配系统的特点。为了更深入的了解 XML 模式匹配的现状，下面对常用 XML 模式匹配系统做一些比较。

在表 1 中给出了各个系统的自动化程度，即是否需要用户手动处理；以及各个系统用得到的基本匹配方法以及采用的辅助资源，基本匹配方法根据前面的匹配方法分类主要分两类：基于元素层的方法和基于结构层的方法；辅助资源主要考虑是否采用词典等外部资源。

从以上模式匹配系统来看，目前大部分模式匹配系统都是基于模式级的匹配，只有很少部分是基于实例的和混合的；而且大部分系统采用的外部资源仅仅是字典，少量使用了领域本体信息；除了 Cupid、COMA 可以作为通用模式匹配方法以外，其他的匹配方法只是正对某些领域的。Bernstein 等[53]指出现有所有算法都较为脆弱，经常需要手工调整，如设置阈值、提供字典或用实例训练；即便调优之后，也不难发现不能被算法正确匹配的模式；其中多数不能胜任大规模模式匹配。

由于系统结构匹配算法的不同，匹配本身是一个比较主观的操作。另外，由于设计时的应用目标不一样，一些特征具有偏向性，因此对系统进行定量比较是不可行的。所以比较只能引用相关参考文献的描述来进行。文献[13]对 Cupid、COMA、S-match 作出了一个比较。在召回率上，COMA 是 Cupid 的 3 倍左右，而 COMA 也只是 S-match 的 0.75 倍左右；在精确度上 Cupid 约是 COMA 的 0.75 倍，而 S-match 约是 COMA 的 1.25 倍；但是 S-match 用的时间却远远多于这两者。文献[52] 给出了文献领域的比较，XPrüM 在查全率和精确度上都优于 SF。总的来说组合的匹配效果要优于单个的匹配器得到的效果；类似地，提取多样性的信息、采用重用信息和本体等信息使得匹配方法更加完善。

5　小结

综上所述，在模式匹配系统中采用的关键技术主要有信息检索技术、机器学习技术、图论技术和本体技术等。在信息检索技术中词典是必不可少的；在机器学习技术中主要采用机器学习方法对模式数据实例进行统计分析和特征抽取，得到数据实例之间的关系，在此基础上推测出模式成员之间的匹配关系以及应用机器学习方法对已知匹配属性进行学习，然后将学习结果应用到新的匹配上；图论的技术主要是在进行模式匹配时把模式成员之间的关系表达为树或图的形式，应用图论方法来计算图中节点间的语义相似度；目前本体技术主要是用本体来描述模式成员之间的关系。

已有模式匹配方法与系统有待进一步改善。大部分匹配方法都是基于模式级的匹配，只有一少部分是基于实例级的匹配。实例是对语义的深入描述，如果数据质量好，在模式级语义不明确的情况下，它能起到很好的辅助作用；很多系统只是产生 1:1 的简单匹配，而缺乏对复杂匹配的研究。在匹配技术中，大部分匹配方法在处理时都采用树结构，利用图结构的系

统比较少；已有的匹配系统缺乏对已产生映射信息以及对用户反馈信息的重用以及对本体的使用；目前模式匹配工作大部分仍以人工定义方式为主，需要寻找自动化程度高、可以应用于不同数据模型和应用领域的综合的模式匹配方法。自动化模式匹配是研究人员追求的理想和目标。

此外，也出现一些新的方法用于改善匹配系统。如，不同于传统模式匹配，整体(holistic)模式匹配[54]通过同时匹配多个模式，一次性找出全部匹配。Gottlob 等[55]提出了对模式映射规范化与优化的标准，推进了模式映射优化理论研究。Alexe 等[56]提出了一个称为 STBenchmark 的测试标准，用于对诸多模式匹配系统和方法的评估。但标准本身是否被认可，还需要进一步验证。

参考文献

1 Chawathe S, Garcia-Molina H, Hammer J, et al. The TSIMMIS Project: Integration of Heterogeneous Information Sources. Proc. of IPSJ Conference, 1994.7—18.

2 Levy AY, Rajaraman A, Ordille JJ. Querying Heterog- eneous Information Sources Using Source Descrip- tions. VLDB Conference, 1996:251—261.

3 Euzenat J. An API for ontology alignment. Proc. of the Int. Semantic Web Conference (ISWC), 2004:698—712.

4 EReddy M, Prasad BE, GReddy P. A Methodology for Integration of Heterogeneous Databases. IEEE TKDE, 1994,6(6):920—933.

5 Chang K, He B, Zhang Z. Toward large scale integration: Building a metaquerier over databases on the web. Proc. of CIDR, 2005:44—55.

6 He B, Patel M, Zhang Z, Chang K. Accessing the Deep Web. CACM, 2007,50(2):94—101.

7 Madhavan J, Bernstein PA, Rahm E. Generic Schema Matching with Cupid. VLDB Conference. 200l:49—58.

8 Melnik S, Molina-Garcia H, Rahm E. Similarity flooding: A versatile graph matching algorithm. ICDE Conference, 2002:117—128.

9 Aumüller D, Do HH, Massmann S, Rahm E. Schema and ontology matching with COMA++. SIGMOD Conference 2005:906—908

10 Do HH, Rahm E. COMA: A System for Flexible Combination of Schema Matching Approaches. VLDB Conference, 2002:610—621.

11 Doan AH, Domingos P, Levy A. Learning source descriptions for data integration. Proc. of the Workshop on the Web and Database, 2000:81—86.

12 Modica G, Gal A, Jamil H. The use of machine generated ontologies in dynamic information seeking. Proc. of CoopIS, 2001:433—448.

13 Giunchiglia F, Shvaiko P, Yatskevich M. S-Match: an algorithm and an implementation of semantic matching. Proc. of ESWS, 2004:61—75.

14 Bonifati A, Mecca G, et al. Schema mapping verification: the spicy way. EDBT Conference, 2008: 85—96.

15 Rahm E, Bernstein PA. A Survey of Approaches to Automatic Schema Matching. The VLDB Journal, 2001,10(4):334—350.

16 Do HH. Schema matching and mapping based data integration[Ph.D. Thesis]. University of Leipzig, 2006.

17 Giunchiglia F, Shvaiko P. Semantic matching. Knowledge Engineering Review, 2003,18(3):265—280.

18 Bouquet P, Serafini L, Zanobini S. Semantic coordina- tion: A new approach and an application to schema matching. ISWC, 2003:130—145.

19 Ehrig M. Ontology alignment: bridging the semantic gap. Semantic web and beyond: computing for human experience, 2007.

20 Doan AH, Halevy A. Semantic integration research in the database community: A brief survey. AI Magazine. 2005,26(1):83—94.

21 Shvaiko P, Euzenat J. A survey of schema-based matching approaches. Journal on Data Semantics, 2005:146—171.

22 Cruz IF, Antonelli FP, Stroe C. AgreementMaker: Effcient Matching for Large Real-World Schemas and Ontologies. VLDB Conference, 2009:24—28.

23 Berlin J, Motro A. Autoplex:Automated Discovery of Content for Virtual Databases. CoopIS, 2001:108—

122.

24 Doan A, Domingos P, Halevy A. Reconciling Schemas of Disparate Data Source:A Machine-Learing Approach. SIGMOD Conf., 2001:21—24.

25 Li Y, Liu D, zhang W. A Generic Algorithm for Heterogeneous Schema Matching. Interational Journal of information Technology, 2003,9(l).

26 Mitra P, Wiederhold G, Kersten M. A Graph-Oriented Model for Articulation of Ontology interde pendencies. EDBT, 2000:86—100.

27 Popa L, Hernandez MA, et al. Mapping XML and Relational Schemas with Clio. ICDE, 2002.498.

28 Lee D, Chu WW. CPI: Constraints-Preserving inlining Algorithm for Mapping XML DTD to Relational Schema. Joumal of Data & Konwledge Engineering, 2001,39:3—25.

29 Benkley S, Fandozzi J, Housmaneta1 E. Data element tool-based analysis (DELTA). The MITRE Corporation, Bedford, MA, 1995.

30 Hovy E. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. Proc. Int. Conf. on Language Resources and Evaluation (LREC), 1998:535—542.

31 Mitra P, Wiederhold G, Jannink J. Semi-automatic integration of knowledge Sources. Proc. of Fusion, 1999.

32 Palopoli L, Terracina G, Ursino D. The system DIKE: Towards the semi-automatic synthesis of cooperative information systems and data warehouses. ABDIS Conference, 2000:108—117.

33 Mitra P, Wiederhold G, Kersten M. A Graph oriented model for articulation of ontology interdependencies. Proc. EDBT 2000:86—100.

34 Castano S, Ferrara A, Montanelli S. Matching ontologiesin open networked systems: Techniques and applications. Journal on Data Semantics, 2006:25—63.

35 Noy NF, Musen MA. Anchor-prompt: Using non-local context for semantic matching. IJCAI Workshop on Ontologies and Information Sharing. 2001:63—70.

36 Lee ML, Yang LH, Hsu W. XClust: Clustering XML Schemas for Effective Integration. CIKM Conf., 2002: Schemas for Effective Integration. CIKM Conf., 2002: 292—299.

37 Velegrakis Y, Miller R, Popa L, Mylopoulos J. ToMAS: A system for adapting mappings while schemas evolve. ICDE Conference, 2004.862.

38 Xu L, Embley DW. Discovering direct and indirect matches for schema elements. DASFAA Conference, 2003:39—46 .

39 Doan AH, Madhavan J, Domingos P, Halevy A. Learning to map between ontologies on the semantic web. WWW Conference, 2002:662—673.

40 Dhamankar R, Lee Y, Doan AH, et al. iMAP: Discovering Complex Semantic Matches between Database Schemas. SIGMOD Conference, 2004:383—394.

41 Ichise R, Hamasaki M, Takeda H. Discovering relation-ships among catalogs. Proc. of Int. Conference on Discovery Science, 2004:371—379.

42 Berlin J, Motro A. Database Schema Matching Using Machine Learning With Feature Selection. CAISE 2002:452—466.

43 Doan AH, Domingos P, Halevy AY. Reconciling Schemas of Disparate Data Sources: A Machine Learning Approach. SIGMOD Conference, 2001:509—520.

44 Euzenat J, Valtchev P. Similarity-based ontology alignmentin OWL-lite. European Conference on Artificial Intelligence, 2004.333—337.

45 Tang J, Li J, Liang B, Huang X, Li Y, Wang K. Using Bayesian decision for ontology mapping. Journal of Web Semantics, 2006,4(1):243—262.

46 Miller RJ, Hernández MA, Haas LM, Yan L. The Clio Project: Managing Heterogeneity. SIGMOD Record, 2001,30(1):78—83.

47 Nandi A, Bernstein PA. HAMSTER: Using Search Clicklogs for Schema and Taxonomy Matching. VLDB Conference, 2009:24—28.

48 An Y, Borgida A, Mylopoulos J. Discovering the semantics of relational tables through mappings. Journal on Data Semantics, 2006, VII:1—32.

49 Tansalarak N, Claypool KT, Hegde V. QMatch -Using Paths to Match XML Schemas. Data & Knowledge Engineering. 2007,60(2).

50 Saleem B, Bellahsene Z, Hunt E, PORSCHE: performance oriented schema mediation. Information Systems, 2008, 33(7—8):637—657.

51 刘强,赵迪,钟华,黄涛.本体辅助的自动化模式匹配技术.软件学报, 2009,20(2):234—245.

52 Algergawy A, Schallehn E, Saake G. Improving XML schema matching performance using Prüfer sequences. Data & Knowledge Engineering, 2009,68(8): 728—747.

53 Bernstein P, Melnik S, Petropoulos M, Quix C. Industrial-strength schema matching. SIGMOD Record, 2004,33(4):38—43.

54 He B, Chang K. A Holistic Paradigm for Large Scale Schema Matching. SIGMOD RECORD, 2004, 33(4): 20—25.

55 Gottlob G, Pichler R, Savenkov V. Normalization and Optimization of Schema Mappings. VLDB Conference, 2009:1102—1113.

56 Alexe B, Tan WC, Velegrakis Y. STBenchmark: towards a benchmark for mapping systems. VLDB Conference, 2008:230—244.