一种多策略通用模式匹配方法*)

程 伟[1] 周龙骧[2] 林河水[1] 孙玉芳[1]

(中国科学院软件研究所 北京100080)[1] (中国科学院数学与系统科学研究院 北京100080)[2]

摘 要 模式匹配问题即寻找不同模式元素之间的语义对应关系,在数据仓库、异构数据源集成及语义Web等领域都是非常重要的研究基础。目前模式匹配仍大多主要由人工来完成,因此有很大局限性。提出了一种多策略通用模式匹配架构,可以方便地兼容其他匹配策略。采用了一种基于词语间语义距离的方法来计算其语义相似度;提出了一种基于相似度传播的结构匹配算法,有效地考虑了相邻相似节点间的相互影响。实验结果表明这种匹配方法在处理模式匹配任务时能达到较高的精度。

关键词 模式匹配,相似度,XML,图

A Multistrategy Generic Schema Matching Approach

CHENG Wei[1] ZHOU Long-Xiang[2] LIN He-Shui[1] SUN Yu-Fang[1]

(Institute of Software, Chinese Academy of Sciences, Beijing 100080)[1]

(Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080)[2]

Abstract Schema matching, the problem of finding semantic correspondences between elements of two schemas, plays a key role in many applications, such as data warehouse, heterogeneous data sources integration and semantic Web. Currently, schema matching is largely performed manually by domain experts, thus a time-consuming and labor-intensive process. In this paper, we describe a multi-strategy schema matching framework, which can combine multiple matching strategies flexibly and its architecture is extensible to new matchers. We adopt an approach based on semantic distances between words to compute their semantic similarity. We propose a structural matching algorithm based on semantic similarity propagation, which consider the effect between neighboring nodes. After describe our approach, we present experiment results on several real-world domains, and show that the approach discovers semantic mappings with a high degree of accuracy.

Keywords Schema matching, Similarity, XML, Graph

1 引言

模式匹配的目标是寻找两个或多个模式的元素之间的语义上的对应关系。模式匹配问题已成为近年来数据库研究的一个热点,在数据仓库、异构数据源集成、语义Web等领域都是非常重要的研究基础,并已成为模型管理的一个重要组成部分[1]。目前模式匹配仍主要由人工(通常是系统的设计者如DBA)来完成,无疑,这是一项非常耗费人力和时间的工作。尤其当需要匹配的模式非常多的时候,由人工来完成模式匹配的工作几乎是不可能的。因此,寻求模式的(半)自动匹配方法就成为我们研究的目标。

文[6]对现有模式匹配系统进行了研究和概述:根据匹配对象的不同,可分为模式水平的匹配和实例水平的匹配、元素水平的匹配和结构水平的匹配;根据具体匹配方法的不同,可分为基于语言学特征的匹配方法和基于约束的匹配方法;根据自动匹配策略的不同,可分为基于规则的匹配和基于学习器的匹配。

近年来在模式匹配问题研究中采用的技术主要有信息检索(IR)技术、机器学习技术、图论技术等。这些技术针对不同的应用域,分别从不同的层次和角度来解决模式匹配问题,取得了较好的效果。但同时这些技术也都具有自身的局限性,在解决不同的模式匹配问题时也都有各自的缺陷。如何灵活应用这些技术来解决通用模式匹配问题,就成为本文的讨论重点。

本文提出了一种多策略通用模式匹配架构,在此架构之上可以应用多种匹配策略,有效地避免了单一匹配策略的局限性。提出了一种基于相似度传播思想的结构匹配算法,有效地考虑了相邻相似节点间的相互影响。

2 相关工作

近年来国外在这个领域已经展开了很多研究。比较著名的有LSD[2]、CUPID[3]、SEMINT[9]、SKAT[10]、DIKE[8]等。下面简要介绍几种具有代表性的模式匹配系统。

文[3]介绍了一个模式匹配原型系统CUPID。CUPID在匹配过程中,针对一些同义词、缩写、首字母缩写等,使用了同义辞典作为辅助信息源。CUPID系统使用一个Treematch结构匹配算法进行结构匹配,该算法没有考虑到相邻相似节点间的相互影响,在处理由不同语义的元素组成具有相似结构的模式匹配时效果很差,无法区分具有强结构相似性的不同语义元素。

LSD[2]系统是Washington大学研制的一个自动模式匹配系统,是目前最具代表性的基于机器学习方法的自动模式

*)本课题得到国家自然科学基金项目资助(19831020)。程 伟 博士研究生,主要研究方向为数据库、XML、中文信息处理。周龙骧 研究员,主要研究方向为数据库、XML、知识库。林河水 硕士研究生,主要研究方向为数据库、中文信息处理。孙玉芳 研究员,主要研究方向为系统软件和中文信息处理。

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,这些学习器对模式元素通过训练产生各自的相似度预测值,然后通过一个meta-learner合成学习结果,再经过域约束检验和用户检验,得到最终的匹配结果。

SKAT[10]是一个基于规则的半自动化的模式匹配原型系统。在SKAT中,用户提供初始的特定应用域的匹配和非匹配关系定义,形成表达匹配和非匹配关系的规则;然后定义派生新的匹配的方法,对根据规则派生出来的匹配关系进行接受或拒绝的识别操作。匹配方法主要采用了名称匹配和简单结构匹配。SKAT对用户定义的依赖度很高。

在这些匹配系统中,提出了各种解决方法,采用了不同的技术。这些技术针对不同的应用域,分别从不同的层次和角度来解决模式匹配问题,取得了较好的效果。我们对各种解决方法中所采用的关键技术进行了研究和总结:

1)信息检索(IR)技术。在进行模式匹配时通常应用近似匹配、基于距离(如edit distance)等匹配技术,消除了精确匹配、基于关键字的匹配等方法的局限性。这种匹配方法应用的前提是:相似的属性名称代表语义上的相似性。通常对于具有清晰规则命名的属性名称应用这种匹配方法可得到较好的匹配效果。作为这种匹配方法的一个重要组成部分,词典是必不可少的。

2)机器学习技术。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。比如在LSD系统中采用了Bayesian classifier方法,通过计算给定数据样本后验概率来计算属性间的语义相似度。

3)图论技术。主要用于结构匹配:在进行模式匹配时把模式元素表达为树图的形式,应用图论方法来计算图中节点间的语义相似度。

但同时这些技术也都具有各自的局限性,在解决不同的模式匹配问题时也都存在一定的缺陷:

·基于模式水平的匹配由于只考虑模式本身的信息,不考虑模式所包含的数据实例的信息,因此在处理一些命名规范(不易产生歧义)的模式时效果较好。反之,在处理一些命名不规则的模式时效果很差。

·基于实例水平的匹配大多使用机器学习的方法,着重考虑了对模式数据的抽取分析。但实例数据同样会产生歧义,如对Student_name和Faculty_name的数据实例分析是无法得出二者的区别的。另外,采用机器学习方法的匹配系统由于需要采集训练数据,因此匹配结果对训练数据的依赖性较高。

·采用图论技术进行匹配的方法一般应用于结构匹配,主要存在两个问题:(1)如何区分具有强结构相似性的不同语义模式元素;(2)如何应用于不同的应用域。

3 多策略模式匹配方法

在匹配原型系统中我们采用了一种多策略的模式匹配方法,从多个层次上进行模式匹配,各自的匹配结果合成后得到最终的匹配输出。匹配原型系统架构如图1所示。

从图1中我们可以知道:系统的匹配处理过程包括以下四个阶段:

1. 匹配任务模式S_1,S_2作为系统的输入模式,首先转换为有向图的形式,图中的每个节点对应于输入模式中的每个元素,有向箭头表示节点间的关系。然后输入模式通过系统的预处理模块进行匹配前预处理工作。这个阶段的工作主要包括约束检验、域检验、用户自定义条件约束等,如:系统遍历匹配空间,产生所有理论上的匹配对,然后通过域约束检验移除违背约束条件的匹配对。

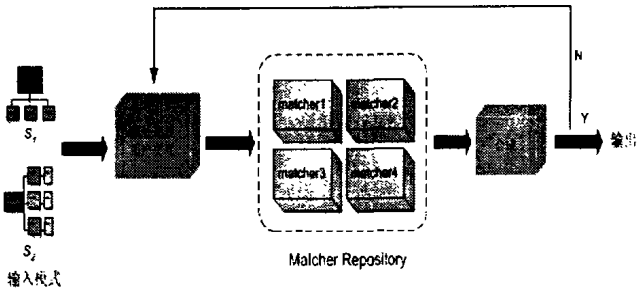

图1 模式匹配系统架构

2. 经过预处理阶段处理的输入模式接下来就由Matcher Repository中的各种类型的匹配器执行匹配操作。Matcher Repository中的匹配器主要有三种:基于语言学特征的匹配器、基于模式结构特征的结构匹配器、基于模式实例特征的实例匹配器。一方面,不同的匹配算法和匹配器可以方便地加到Matcher Repository中来;另一方面,用户也可根据自己的需要来选择使用不同的匹配策略。

3. 不同匹配器的输出结果进行加权合成,用户可以根据匹配任务的特性和需要设定各种匹配策略的权重。

4. 不同匹配器的输出结果进行合成后与用户进行交互处理,用户对匹配结果进行选择和优化。经过用户反馈处理后的匹配结果再次返回Matcher Repository进行匹配,直到输出结果达到用户满意为止。

3.1 基于语言学特征的匹配

我们采用了一种基于电子词典WordNet[7]计算模式元素间语义距离进而确定其匹配关系的匹配算法。单词被组织为词树的形式,并且任意两个节点w_1,w_2之间有且仅有一条路径R。把w_1,w_2之间的语义距离定义为路径R的步长:$d(w_1,w_2)=step\_length(w_1,w_2)$。

定义1 两个节点w_1,w_2间的相似度$sim(w_1,w_2)$可由其语义距离转换得到:

$$sim(w_1,w_2)=\begin{cases}1 & d(w_1,w_2)=0\\ \dfrac{\alpha}{d(w_1,w_2)} & d(w_1,w_2)\neq 0\end{cases} \quad (1)$$

(1)式中的α是一可调参数。

算法思想如下:

步骤1:根据节点w_1,w_2的位置和层次关系计算二者之间的语义距离;

步骤2:由定义2,将w_1,w_2之间的语义距离转换为二者之间的语义相似度。

由步骤1,2可以计算出树图中任意节点间的语义相似度。所有节点对之间的相似度经过计算后进行筛选,只有那些具有较高相似度的匹配对才被作为匹配候选对象。

对于一些特定的应用域中的某些特定的同义词如"Dept"和"Department","Pno"和"Personal Number",显然,上面的算法不能得到很好的结果。为了在这些特定情况下得到较高的匹配精度,我们构造了一个由一系列针对特定应用域的同义词表、通用缩写词表、只取首字母的缩写词表构成的同义辞典。基于这些针对特定应用域的同义辞典,名称匹配器能得到很好的匹配效果。

3.2 结构匹配算法

近年来在模式匹配领域中应用的比较流行的技术就是使用结构匹配。通常的作法是:先将待匹配模式转换为树图结构,然后对树图结构或其子结构进行匹配操作。我们提出了一种新的结构匹配算法 SM(Structural Matching Algorithm)。SM 算法基于相似度传播的思想,考虑了相似度在相邻节点间的传播影响。相似度在树图中传播的思想来源于 Similarity Flooding 算法[4],我们提出了一种新的传播机制和相似度修正更新方法,并应用于我们的原型系统中。经过实验,证明我们的算法是高效的。

该算法思想基于两个假设前提:

1)如果两个模式中的两个节点元素被相似的元素所继承或派生,即认为这两个元素相似;

2)如果两个模式中的节点元素相似,那么它们的相邻元素的相似度亦增加。

定义2 对于有向图 G 中的任一节点 v,它的所有输入邻居节点集(父节点集)为 $I(n)$,所有输出邻居节点集(子节点集)为 $O(n)$,$|I(n)|$、$|O(n)|$分别代表输入邻居节点集和输出邻居节点集中的节点数目。其中,每个输入邻居节点为 $I_i(n)$,$1\leqslant i\leqslant|I(n)|$;每个输出邻居节点为 $O_i(n)$,$1\leqslant i\leqslant|O(n)|$。

对于两个有向图 G_1,G_2,定义其节点对(a,b)之间的初始相似度值为:

$$Sim_0=\begin{cases}0/initialMap & (a\neq b)\\ 1 & (a=b)\end{cases} \qquad (2)$$

(2)式中的 *initialMap* 为通过基于语言学特征匹配得到的相似度值或用户自定义的相似度值。

节点对(a,b)的相似度 Sim_k 可由下式计算得到:

$$Sim_k=wSim_{k-1}+\frac{w_I}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|}Sim_{k-1}(I_i(a),I_j(b))+\frac{w_O}{|O(a)||O(b)|}\sum_{i=1}^{|O(a)|}\sum_{j=1}^{|O(b)|}Sim_{k-1}(O_i(a),O_j(b)) \qquad (3)$$

(3)式中 $w+w_I+w_O=1$。

由(3)式可以看出:节点对(a,b)之间的相似度由它的上一次循环的相似度和它们的所有相邻输入、输出节点对的相似度均值共同决定。当两次连续循环中的两个相似度的向量差值 $\Delta(Sim_k-Sim_{k-1})$的欧几里得距离小于某个阈值 ε 时,循环终止。这样,任意两个节点间的相似度初始值经过多次循环后就传播到整个图中。

4 实验结果及分析

我们在原型系统中实现了 SM 算法,并使用实际数据集[5]进行了实验。实验环境是一台 PIII 667MHz,256M 内存,运行 Redhat Linux9.0的 PC 机,开发工具采用 C 语言开发环境 Anjuta 1.2.1。

为了评价匹配操作的质量,我们采用了这样的评价方法:对于某个特定的匹配任务,将自动匹配的输出结果与人工匹配的结果进行比较(这里认为人工匹配的结果是正确的)。引用了在 IR 领域的常用的评价指标 *Precision*、*Recall* 及其调和均值 *F-Measure* 对匹配结果精度进行评价。*Precision* 反映了系统检测到的正确匹配数和系统检测到的全部匹配数的比率,*Recall* 反映了系统检测到的正确匹配数和实际全部正确匹配数的比率,*F-Measure* 是 *precision* 和 *recall* 的调和均值。我们定义了8个匹配任务,每个任务由两个匹配模式组成。实验结果如图2、3所示。

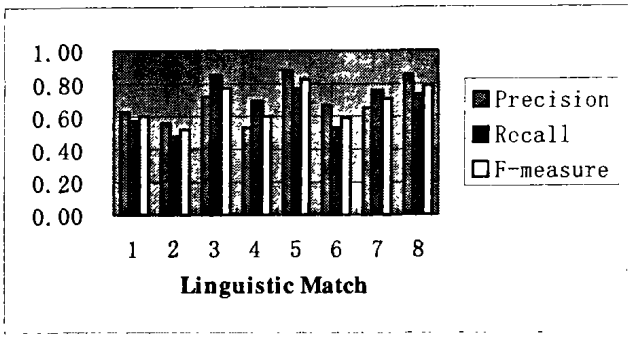

图2 基于语言学特征的匹配结果评价

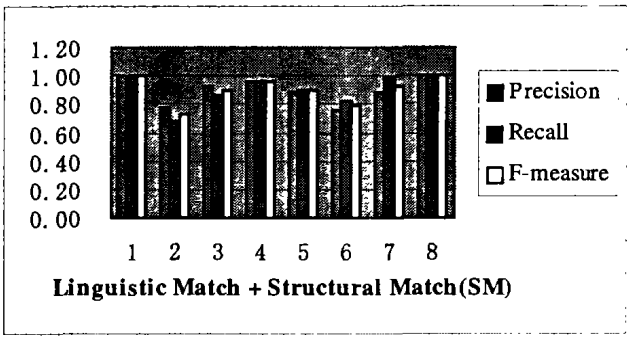

图3 语言学特征匹配+结构匹配结果评价

由实验可知,当使用基于语言学特征的匹配器时,得到的 *F-measure* 平均值为68%;当加上使用 SM 算法的结构匹配器时,*F-measure* 平均值增加到90%。实验结果表明:多策略匹配方法能显著提高匹配精度。

结论 本文针对模式匹配问题,在对现有的(半)自动模式匹配策略和关键技术进行研究和分析的基础上,提出了一种多策略匹配方法。该方法使用了基于语言学特征的匹配算法和结构匹配算法以得到模式元素之间语义上的最优匹配关系。实验结果证明:该匹配方法应用于实际应用域时是有效的。

参 考 文 献

1 Bernstein P A, Rahm E. Data warehouse scenarios for model management. In: Proc. 19th Intl. Conf. on Entity-Relationship Modeling, 2000

2 Doan A, Domingos P, Halevy A. Reconciling Schemas of Disparate Data Sources: A Machine-Learning approach. SIGMOD, 2001

3 Madhavant J, Bernstein P A, Rahm E. Generic Schema Matching with Cupid. VLDB, 2001. 49~58

4 Melnik S, Garcia-Molina H, Rahm E. Similarity flooding: A versatile graph matching algorithm. In: Proc. of Eighteenth Intl. Conf. on Data Engineering, San Jose, California, 2002

5 http://anhai.cs.uiuc.edu/archive/summary.type.html

6 Rahm E, Bernstein P A. A survey of approaches to automatic schema matching. The VLDB Journal, 2001, 10(4): 334~350

7 Fellbaum C. WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Massachussets, 1998

8 Palopoli L, Terracina G, Ursino D. The system DIKE: towards the semi-automatic synthesis of cooperative information systems and data warehouse. ADBIS-DASFAA Conf., 2000

9 Li W, Clifton. SemInt: A Tool for Identifying Attribute Correspondences in Heterogeneous Database Using Neural Network. Data & Knowledge Engineering, 2000

10 Mitra P, Wiederhold G, Jannink J. Semiautomatic integration of knowledge sources, FUSION 9