基于列重合度的网络表格一致性扩展

齐 飞 王 宁 张丽方 孙伟娟
(北京交通大学计算机与信息技术学院 北京 100044)

摘 要 网络表格的扩展是根据已知信息扩展与主列相关的其他属性列,以满足人们通过表格获取感兴趣信息的需求。目前的研究工作主要针对由主列和待扩展列组成的实体-属性二元表,并将主列视为其他属性列扩展的唯一依据,但该技术运用到具有多个待扩展列的网络表格时,由多个二元表拼接而成的结果表很容易出现实体不一致现象。综合考虑各属性列间以及元组行间的关系,提出一致性支持度概念,设计并实现了基于列重合度的表格一致性扩展系统CCA,其既能保证候选值的高匹配分数,又能使结果表中填值所使用的数据源表数目最小化,有效地避免了实体不一致问题。实验表明,与现有方法相比CCA系统有更高的精确度、覆盖率、一致性,以及更低的查询时间代价。

关键词 网络表格扩展,列重合度,列映射,一致性支持度
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.09.039

Consistent Web Table Augmentation Based on Column Overlapping

QI Fei WANG Ning ZHANG Li-fang SUN Wei-juan
(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract Web table augmentation refers to extend table content based on main column or other known information, which helps people to obtain information they are interested in. Current research focuses on entity-attribute binary table made of main column and extended column, where the primary column is the only basis. When it is applied to a table with multiple columns to be extended, the result table consolidated by binary tables will suffer from entity inconsistency problem. We proposed consistency support degree based on relationships between columns as well as between tuples in the table, and implemented the CCA system for table consistency augmentation based on column overlapping. Our method keeps the high matching score of candidate values using as few source tables as possible to avoid entity inconsistency. Experimental results show that the proposed CCA system has higher accuracy, coverage, consistency and lower query time cost compared with existing methods.

Keywords Web table augmentation, Overlapping degree of columns, Column mapping, Consistent support degree

1 引言

随着大数据时代的到来,网络中大量的结构化数据逐渐引起人们的关注。早在2008年,Google研究的WEBTABLES项目致力于提取并充分利用网络上大量的结构化表格[1],该项目从数以亿计的网页中抽取了141亿张HTML表格,使用统计分类技术找出了大约1.54亿张包含高质量关系型数据的网络表格[2]。国内一些学者也对网络表格的定位技术[3]、关系发现[4]等做了诸多研究。当用户想要通过表格查询来获取感兴趣的信息时,一些在线服务实现了让用户提供一组关键字或不完整的小表格作为查询条件,根据一定的相似度算法查找所有与查询表相关联的表,从相关表中提取候选值,将查询表扩展成一张信息丰富且完整的表格返回给用户,即所谓的表格扩展技术[14,15]。

目前,存在一些商业的表格搜索引擎,例如Google Tables[6-7]根据提供的关键字返回一个根据相关度排列的表格列表,用户根据需要筛选表格中的有用信息。另外,一些研究实现了表格的扩展工作,例如Rakesh Pimplikar等人设计[14]了一个结构化搜索引擎,能够实现多个关键字作为表格列属性名的组合查询。INFOGATHER系统能根据属性名和主列值进行二元关系表的属性扩展[10]。INFOGATHER+系统则着重考虑了数值型和时变型属性的扩展[11]。德国曼海姆大学数据和网络科学小组的Oliver Lehmberg等人[16-18]设计了Search Join引擎,实现了表格数据的搜索、连接和合成,用户可以指定扩展列,也可以自动扩展所有与主列相关的列。Julian Eberius等人[20]利用集合覆盖思想解决了表格查询结果的可靠性问题,用户查询后返回Top-k的结果表集合,以供用户甄选。Google公司将研究了多年的WEBTABLES项目

到稿日期:2016-08-11 返修日期:2016-12-28 本文受国家自然科学基金面上项目(61370060)资助。
齐 飞(1990-),女,硕士生,CCF会员,主要研究方向为Web数据集成、数据挖掘;王 宁(1967-),女,博士,教授,主要研究方向为Web数据集成、大数据管理与分析、数据挖掘,E-mail:nwang@bjtu.edu.cn(通信作者);张丽方(1991-),女,硕士生,主要研究方向为Web数据集成;孙伟娟(1993-),女,硕士生,主要研究方向为Web数据集成、数据挖掘。

应用于实践,实现了网络表格和融合表格的在线检索[5-9]。

当前的研究工作仍然存在一些问题,INFOGATHER 和 Search Join 在进行表格的属性扩展时忽略了各属性间及元组间的关联关系,只是单纯地考虑主列与待扩展列的二元关系,在扩展多个属性列时很容易出现实体不一致问题,导致查询结果的准确度低,给用户带来困扰。

例如,表 1 的查询表主列为"Brand",待扩展属性列为"S-size"和"OS"。表 2 和表 3 分别是介绍手机和电脑的候选表,这里只提取相关的几行作为示例。传统方法在扩展每一列时仅仅考虑属性名及主列值的匹配,经过相似度计算之后,对于扩展列"S-size",表 3 的相关分数高于表 2,候选值"13.3in"和"14.0in"被填入查询表的对应行;对于扩展列"OS",表 2 的相关分数高于表 3,候选值"iOS"和"Android 5.1"被填入查询表的"OS"列。最终的结果表如表 4 所列,出现了实体不一致问题,本应该是电脑属性的值被错误地填到了手机信息表中。

表 1　查询表

Brand	Price	Memory	S-size	OS
Apple	7288	2G		
Lenovo	3029	3G		

表 2　候选表-手机

Brand	Price	Memory	Screen Size	OS
SAMSUNG	2399	4G	5.0in	Android 6.0
Lenovo	3029	3G	6.5in	Android 5.1
Apple	7288	2G	5.5in	iOS

表 3　候选表-电脑

Brand	Price	Memory	S-size	OS
SAMSUNG	5699	8G	15.6in	Windows10
Apple	10788	8G	13.3in	MAC OS X
Lenovo	6999	8G	14.0in	Windows10

表 4　结果表

Brand	Price	Memory	S-size	OS
Apple	7288	2G	13.3in	iOS
Lenovo	3029	3G	14.0in	Android 5.1

由该实例可知,若计算相似度和填值时能够同时考虑"Price"和"Memory"两个已知列,则匹配结果将会截然不同,能够很好地避免不一致问题。因此,本文研究根据列重合度确定表格相关性,综合考虑表格各列间的关系以及数据源的可靠性问题[5,20],将一致性支持度应用于属性扩展,从而保证结果表的实体一致性。主要贡献包括以下 3 点:

(1)首次提出一致性支持度概念,综合考虑列间关系和元组间关系进行表格扩展,从而保证实体的一致性;

(2)提出了对查询表进行预处理,丰富查询条件,进而保证查询结果的可靠性;

(3)设计并实现了基于列重合度的表格一致性扩展算法,将一致性支持度应用于填值算法,保证在候选值分数高的前提下,数据源数目最少。

本文第 2 节给出表格扩展技术的相关定义和基本问题模型;第 3 节阐述基于列重合度的表格一致性扩展算法;第 4 节对实验结果进行分析和评估;最后总结全文。

2　表格一致性扩展的问题模型

2.1　数据模型和相关定义

为了更方便地介绍表格一致性扩展问题的相关定义及系统架构,列出了常用的符号及其描述(见表 5)。

表 5　符号描述

符号	描述
$T=\{t_1,t_2,t_3,\cdots\}$	数据源表集合,$t_i\in T$ 为数据源表
$TC(t)$	表格 t 中的属性列号集合($t=0,1,2,\cdots$)
$Q=\langle QC,\Omega\rangle$	查询表 Q 是属性列号集合 QC 和列权重集合 Ω 组成的二元组
key	查询表 Q 的主列号
KC	查询表 Q 中除主列外的所有已知列号的集合
EC	查询表 Q 的待扩展列号的集合
$\Omega=\{\omega_0,\omega_1,\omega_2,\cdots\}$	查询表 Q 每列的权重
$JaccardFuzzy(C_i,C_j)$	列 C_i 与 C_j 的 $JaccardFuzzy$ 相似度
σ_i	相似度阈值,$i=1,2,\cdots,7$
$kkS(u,v)$	候选表第 v 行与查询表第 u 行中主列值的相似度分数
$ccS(u,v)_i$	候选表第 v 行与查询表第 u 行中对应已知列 i 的相似度分数,$i\in KC$
$rowS(u,v)$	候选表第 v 行与查询表第 u 行的行相似度分数

进行表格的属性扩展时首先要计算表格间的相关度,本文使用 EditDistance 算法计算字符串间的相似度,设计了 NumericSim 算法计算数值型数据的相似度。由于网络表格中的数据值可能存在一些拼写错误,因此相似度计算均使用模糊匹配。

定义 1(列重合度)　假设 C_i 和 C_j 分别为表格 t_1 和 t_2 中的任意一列,C_i 和 C_j 作为列属性值的集合,二者之间的列重合度定义为根据 Jaccard 相似度思想设计的 JaccardFuzzy 算法计算得到的集合相似度,记为 $ccSim(i,j)$。

JaccardFuzzy 算法的思想为:采用集合的交集与并集之比,但在确定集合交集时采用的相似度算法为模糊相似度,即定义了一个阈值 $\sigma_1=0.85$,则当相似度大于或等于 0.85 时认为二者相似。

很显然,表 1"Brand"列与表 2"Brand"列的相似度计算为:$\{$"*Apple*","*Lenovo*"$\}\cap\{$"*SAMSUNG*","*Lenovo*","*Apple*"$\}=2$,$\{$"*Apple*","*Lenovo*"$\}\cup\{$"*SAMSUNG*","*Lenovo*","*Apple*"$\}=3$,$ccSim(0,0)\approx0.667$,同理两表相应的"Price"列和"Memory"列的相似度为分别 $ccSim(1,1)\approx0.667$,$ccSim(2,2)\approx0.667$,其他列间的相似度均为 0。表 2 的"Price"列与表 3 的"Price"列的相似度计算为:$\{$"2399","3029","7288"$\}\cap\{$"5699","10788","6999"$\}=1$,$\{$"2399","3029","7288"$\}\cup\{$"5699","10788","6999"$\}=5$。由于采用模糊相似度计算,"7288"和"6999"的相似度经数值型相似度算法计算得 $0.96>\sigma_1$,满足模糊阈值,认为二者相似,故 $ccSim(1,1)=1/5=0.2$。同理,两表相应的其他非零相似度为:$ccSim(0,0)=3/3=1$,$ccSim(2,2)=1/3=0.33$,$ccSim(3,3)=1/5=0.2$。

定义 2(列映射)　已知两个表格 t_1 和 t_2,C_i 和 C_j 分别为表格 t_1 和 t_2 中的任意一列,根据两个表格间的列重合度确定列对应关系,所有列对应关系构成的集合称为两个表格之间的列映射关系,记为 $map(t_1,t_2)$。计算公式如下:

$$map(t_1,t_2)=\begin{cases}\{\langle i,j\rangle \mid \forall i\in TC(t_1),\exists j\in TC(t_2)\}, & \text{if } \phi=1\\ \{\langle i,j\rangle \mid i\in TC(t_1), j=-1\}, & \text{if } \phi=0\end{cases} \tag{1}$$

条件 $\phi=[\![\sum_{j\in TC(t_2)}[\![C_i\approx C_j]\!]_1^0\leqslant 1, i\in TC(t_1)]\!]_1^0$，$[\![X]\!]_1^0$ 表示当条件 X 为真时取值为 1，反之为 0；$C_i\approx C_j$ 表示 C_i 和 C_j 两列满足列重合度阈值和列映射关系算法（详见 3.1 节算法 1）；映射关系 $\langle i,j\rangle$ 中，当 $j=-1$ 时，说明 t_2 中没有映射列与 t_1 中的第 i 列对应，则 $\langle i,-1\rangle$ 称为无效列映射；反之为有效列映射。式(1)的条件 ϕ 约束了有效列映射的唯一性。

根据列映射关系及定义 1，两个表格之间的列重合度均值的计算公式如下：

$$sim(t_1,t_2)=\text{avg}\sum_{\langle i,j\rangle\in map(t_1,t_2)} ccSim(i,j) \tag{2}$$

根据列重合度计算和确定列映射关系的算法得出表 2 和表 3 之间的列映射关系为 $\{\langle 0,0\rangle,\langle 1,1\rangle,\langle 2,2\rangle,\langle 3,3\rangle,\langle 4,-1\rangle\}$，其中 $\langle 4,-1\rangle$ 为无效列映射。根据式(2)可知，二者的列重合度均值为 sim(表 2，表 3)=0.346。

定义 3（扩展列支持度） 已知查询表 q 和候选表 t 的列映射关系为 $map(q,t)$，由 t 与 q 对应扩展列的有效列映射数目和 q 的扩展列数目构成的比值称为候选表 t 的扩展列支持度，表示为：

$$exCS(t)=\frac{|\{\langle i,j\rangle \mid \langle i,j\rangle\in map(q,t), i\in EC, j\neq -1\}|}{|EC|} \tag{3}$$

例如，表 1 和表 2 的列映射关系为 $\{\langle 0,0\rangle,\langle 1,1\rangle,\langle 2,2\rangle,\langle 3,3\rangle,\langle 4,4\rangle\}$，表 1 共有两个扩展列，表 2 对应扩展列的有效列映射数目为 2，因此，$exCS$(表 2)=2/2=1。

定义 4（元组行支持度） 已知 m 行的查询表 q 和 n 行的候选表 t，t 与 q 匹配的行数目与 m 的比值称为候选表 t 的元组行支持度，表示为 $rowNS(t)$。计算公式如下：

$$rowNS(t)=\frac{|\{rowS(u,v)\geqslant\sigma_2 \mid \forall u\in\{0,\cdots,m\}, \exists v\in\{0,\cdots,n\}\}|}{m} \tag{4}$$

其中，$rowS(u,v)=kkS(u,v)*\omega_{key}+\sum_{i\in KC} ccS(u,v)_i*\omega_i$，当 $rowS(u,v)\geqslant\sigma_2$ 时，则认为该行为候选行。例如，表 2 经过计算得到 $rowS(0,2)=rowS(1,1)=1.0$，故表 2 的候选行数目为 2，$rowNS$(表 2)=2/2=1。

定义 5（一致性支持度） 已知查询表 q、候选表 t 和填值候选表集合 A，候选表 t 的一致性支持度 $canTS(t)$ 由 t 的扩展列支持度、元组行支持度、t 与 A 的相似度 3 部分组成，计算公式如下：

$$canTS(t)=exCS(t)*rowNS(t)*Sim(t,A) \tag{5}$$

其中，$Sim(t,A)=avg\sum_{a\in A} sim(t,a)$ 为 t 与 A 中各个填值候选表之间列重合度均值的平均值，集合 A 初始为空集，并定义 $Sim(t,\emptyset)=1$，每次选定一个新的候选表进行填值后，将其加入集合 A 中。一致性支持度用来衡量候选表所提供的候选行值能否在最大程度上保证结果表的实体一致性。

2.2 基于列重合度的表格一致性扩展框架

基于列重合度的表格一致性扩展系统（Column-Overlap Consistency Augmentation System，CCA）用列重合度进行表格相关度的衡量，借鉴集合覆盖思想将一致性支持度应用于候选值的选择。其查询处理阶段的问题模型可以归纳为：

$$augT_q=\{R_{canTS}(C_{avgSim(q,t)}(S_q(T_{map(t_i,t_j)})))\}_{Select} \tag{6}$$

其中，$T_{map(t_i,t_j)}$ 为预处理后的数据源表集合，S 为查找种子表操作，C 为确定候选表操作，R 为锁定候选行操作，最后填值组成扩展表 $augT_q$。图 1 示出 CCA 的系统架构图，系统流程共分为 5 个步骤：

第 1 步 预处理，包括查询表预处理和数据源表预处理。如果初始查询表信息单一，则先进行预查询和预打分，确定各列权重。数据源表预处理是计算两两表格间的列重合度并确定列映射关系，根据列重合度均值构建表格间的关系。

第 2 步 查找种子表。根据查询表提供的查询条件，用表头匹配度和列重合度计算种子表与查询表的匹配分数和列映射关系。

第 3 步 确定候选表。根据预处理得到的表格间的关系图，搜索所有与种子表相关联的表格，修改查询表与候选表间的列映射值，并重新计算候选表的带权平均列重合度分数。

第 4 步 锁定候选行。判断候选表中各行是否与查询表各行匹配，若存在匹配行，则计算候选表的一致性支持度，得到候选行表集合。

第 5 步 选择最终值，即基于一致性支持度的行值填充。根据候选行表一致性支持度的高低进行排序，从高到低依次选定行表进行填充，直到结果表满足设定的覆盖率阈值 $thCov$。由于每次填充时是将整个候选行表中的值全部填入结果表，因此结果表中实际的覆盖率满足 $coverage\geqslant thCov$。

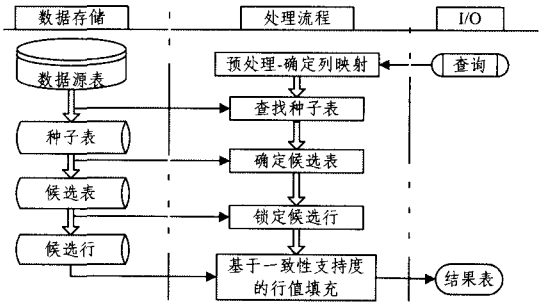


图 1 CCA 系统架构图

3 基于列重合度的表格一致性扩展

3.1 预处理并确定列映射

由于我们的重点是充分利用网络上的结构化表格，因此本系统所使用的数据源均为网络表格，考虑到用户给定的查询表可能只包括主列属性值以及各列属性名，查询条件过于单一，为了使得查询结果更加可靠，对查询表先做两步预处理。

1）预查询。根据用户提供的表格信息，在专业可靠的网站或本地数据库中进行查询，将部分列的属性值填入表格中，构成一个具有多个已知列、信息稍加丰富和完善的小表格，但仍有一些属性列无法从本地数据库中获取，这时再从网络表格库中进行查询和扩展。

2）预打分。计算预查询后构建的小表格各列属性值的相似度范围，根据每列的相似度区间对每个已知属性列打分，根

据式(7)计算得到各个已知属性列在查询表中所占的权重 ω_i,构成权重集合 Ω。

$$\omega_i=\begin{cases}\varepsilon, & \text{if } i=key\\ (1-\varepsilon)\dfrac{avgS_i}{\sum_{j\in KC}avgS_j}, & \text{if } i\in KC\\ 0, & \text{otherwise}\end{cases} \tag{7}$$

其中,$\varepsilon\in(0.5,1]$表示主列所占的权重值不小于 0.5,各列权重满足 $\sum\omega_i=1$;$avgS_i$ 为 C_i 列中各属性值间的平均相似度值,计算公式如下:

$$avgS_i=\begin{cases}\dfrac{\sum_{v_j,v_k\in C_i}ED(v_j,v_k)}{|C_i|}, & \text{if } C_i \text{ 列是字符型数据}\\ \dfrac{\sum_{v_j,v_k\in C_i}NS(v_j,v_k)}{|C_i|}, & \text{if } C_i \text{ 列是数值型数据}\end{cases} \tag{8}$$

其中,ED 和 NS 分别为字符型和数值型相似度算法。列内相似度计算方法与列间不同,列内相似度判断的是该列值是否来自同一个域,列值间的相似度越高,来自同一个域的可能性就越大,该列值的噪声可能更小,可信度也相对高,故该列在查询中所占的权重比应该更高。后文在计算源表格与查询表的整体相似度时,采用加权平均的方法(详见式(13)),各列在表格中的重要程度有所体现。经过预处理后,查询表的信息被丰富。

下面介绍数据源表的预处理,在整个系统中该部分工作至关重要,一方面需要计算两两表格间的列重合度并确定列映射关系(定义 1 和定义 2),另一方面需要根据列重合度均值构建数据源表间的关系,为后续查找种子表和候选表、锁定候选值等工作做准备。

确定列映射关系的算法思想是先计算两表格各属性列间的两两列重合度,迭代确定最匹配的列对关系,每次迭代得出一对列映射。迭代公式如下:

$$\begin{aligned}&getMap^{(1)}(x_1,y_1)=Loc(\max_{x_1\in TC(t_1),y_1\in TC(t_2)}ccSim(x_1,y_1))\\ &\quad\vdots\\ &getMap^{(n)}(x_n,y_n)=Loc(\max_{\substack{x_n\neq x_k,y_n\neq y_k\\ 1\leqslant k\leqslant n-1}}ccSim(x_n,y_n))\end{aligned} \tag{9}$$

其中,$Loc(\delta)$ 函数表示获取 δ 所在数组的行列位置得到 $\langle x,y\rangle$,迭代式(9)的算法实现如算法 1 所示。

算法 1　确定列映射关系

输入:数据源表 t_1,t_2

输出:列映射关系集合 map(t_1,t_2)

```
1. FUNCTION getPairColumnOverlap(t1,t2)
2. FOR ci∈t1 DO
3.     map(t1,t2)←⟨i,−1⟩//初始化
4.     FOR cj∈t2 DO
5.         simMatrix[][] ← ccSim(i,j)
6. num=0,num1=−1
7. While(num !=num1) DO
8.   num1=num
9.   FOR ccSim(i,j)∈simMatrix[][] DO
10.        max(x,y)=ccSim(i,j)
11.  IF max>σ3 THEN
12.        map(t1,t2)←⟨x,y⟩//定位列关系
13.        simMatrix[x][],simMatrix[][y]←0
14.        num++
15. RETURN map(t1,t2)
```

算法 1 首先遍历表格 t_1 和 t_2 的所有列,根据定义 1 计算每个列对之间的列重合度,存放于二维数组 *simMatrix*[][] 中(第 2—5 行),同时初始化列映射值为$\langle i,-1\rangle$(第 3 行)。

迭代计算每一对映射列的方法为:首先确定数组中的最大列重合度 max(第 9—10 行);若 max$>\sigma_3$,则确定列映射关系$\langle i,j\rangle$,并将数组中的第 i 行与第 j 列置 0(第 11—14 行);整个二维数组的值全为 0 时迭代终止,此时所有列映射关系确定完毕。

3.2　查找种子表和候选表

查找种子表的目的是找出与查询表直接相关的表格,查询表提供的查询条件包括主列、已知属性列和表头,先根据表头匹配与否来判断是否为潜在种子表,再用已知列重合度确定种子表与查询表的列映射关系并计算匹配分数:

$$avgSim(q,s)_s=\begin{cases}\sum_{\substack{\langle i,j\rangle\in map(q,s)\\ i\in KC\cup\{key\}}}ccSim(i,j)*\omega_i, & \text{if } q.H\approx s.H\\ 0, & \text{otherwise}\end{cases} \tag{10}$$

其中,q 为查询表,s 为种子表,H 为表头集合,$q.H\approx s.H$ 表示 q 和 s 的表头匹配,即二者表头的相似度大于 σ_4。最终确定的种子表的集合表示为:

$$S(q)=\{s\mid s\in T,avgSim(q,s)_s\geqslant\sigma_5\} \tag{11}$$

表格间列重合度的算术均值计算公式如下:

$$avgSim(s,c)=\frac{\sum_{\langle j,k\rangle\in map(s,c)}ccSim(j,k)}{|\{\langle j,k\rangle\mid\langle j,k\rangle\in map(s,c),k\neq-1\}|} \tag{12}$$

查找候选表时,首先根据 $sim(s,c)\geqslant\sigma_6$ 确定所有与种子表相关联的表格,也就是与查询表间接存在关系的表格,σ_6 为列重合度算术均值的阈值。其次,根据"查询表-种子表-候选表"三者之间的列映射关系,修改查询表与候选表之间的列映射值,并根据式(13)重新计算查询表与候选表列重合度的带权平均值。

$$avgSim(q,c)_c=\sum_{\substack{\langle i,j\rangle\in map(q,s),j\neq-1\\ \langle j,k\rangle\in map(s,c),k\neq-1}}ccSim(i,k)*\omega_i \tag{13}$$

其中,$j,k\neq-1$ 说明$\langle i,j\rangle$和$\langle j,k\rangle$存在有效列映射关系,修改后查询表与候选表间的列映射为$\langle i,k\rangle$,最后确定的候选表集合为:

$$C(q)=\{c\mid c\in T,avgSim(q,c)_c\geqslant\sigma_5\} \tag{14}$$

其中,σ_5 为查询表与种子表、查询表与候选表之间列重合度加权均值应该满足的阈值。

3.3　锁定候选行

候选表被确定后,如何锁定每个候选表中与查询表各行相匹配的元组行,同样是非常重要的工作。首先判断扩展列的列映射关系,若为有效列映射,则根据与查询表中已知属性列值的匹配与否来锁定候选行,然后根据式(5)($A=\emptyset$)计算每个候选行表的一致性支持度,并得到候选行表集合。

$$R(q)=\{c'\mid c\in C,canTS(c)>\sigma_7,c'=c\underset{row}{\cap}q\} \tag{15}$$

其中，$c\underset{nnc}{\cap}q=\{r\mid \forall r\in c,\exists r_1\in q,r=r_1\}$，$r$ 和 r_1 分别表示候选表 c 和查询表 q 中的元组行。

算法 2 锁定候选行

```
输入：查询表 q，候选表 C(q)
输出：候选行表集合 R(q)
1. Function getCandidateRow(q,C(q))
2. Row=null;RowT=null
3. FOR c∈C(q) DO
4.     Compute exCS(c)
5.     IF exCS(c)>0 THEN
6.         FOR k∈q. entitys DO
7.             canC←(key,canC)
8.     IF canC≠-1 THEN
9.             Identify (u,v) & Compute rowS(u,v)
10.            IF rowS(u,v)>σ5 THEN
11.                Row ← c. getRow(v)
12.            RowT ← Row
13.        Compute rowNS(t)
14.        IF RowT ≠ null THEN
15.            R(q) ← RowT,Compute canTS(t)
16. RETURN R(q)
```

算法 2 首先计算该候选表的有效扩展列数目，根据定义 3 得到扩展列支持度 $exCS(c)$。

若 $exCS(c)>0$，说明该表有符合条件的扩展列，则遍历查询表各行，判断该候选表与查询表主列的映射关系是否存在(第 6－7 行)，若存在，再判断该候选表中的哪一个元组行与该查询行匹配，即判断是否存在 $kkS(u,v)\geqslant\sigma_1$，若存在则确定行关系$(u,v)$并计算该行与查询表已知列的相似度 $ccS(u,v)$，得到该行的匹配分数 $rowS(r,s)$(第 8－9 行)。若 $rowS(r,s)>\sigma_2$，则将该行存入候选行表 $RowT$。

将每个候选表中的候选行全部锁定并计数，根据定义 4 计算该候选行表的元组行支持度 $rowNS(t)$，进而根据式(5)计算一致性支持度 $canTS(t)$；最终得到候选行表集合$R(q)$。

3.4 基于一致性支持度的行值填充

锁定了各个候选表中的候选行后，在填值阶段，必须选择最优的候选值写入结果表，即该表格既要与查询表有较高的相关度，又要与已选填值表有较高的整体相似度，还要保证最终填值表的个数尽可能少。

算法 3 选择行值

```
输入：查询表 q，候选行表集合 R(q)
输出：结果表 augQ
1. Function selectValues(q,R(q))
2. flag[][]=0,augQ ← q,A=∅
3. sortR(q)=BubbleSort(R(q)) //候选行表排序
4. while(coverage ⩽thCov)
5.     rT=sortR(q). get(0) //取 canTS 最大的表
6.     FOR row∈rT DO //遍历候选表各行
7.     q. rI← row. getQueryRowI()
8.     〈q. CI,row. CI〉 ← rT. mapCandidateCI()
9.       IF flag[q. rI][q. CI]=0 THEN
10.            q. Row ← setR(q. CI,Value(row. CI))
11.    augQ ← q. Row
12.    A. add(rT),sortR(q). remove(rT)
13.    FOR t∈sortR(q) DO
14.    Compute canTS(t)
15.    sortR(q)=BubbleSort(sortR(q))//排序
16. RETURN augQ
```

算法 3 首先给查询表设置标志数组 $flag[][]$，用来标记每个待扩展单元格是否已被填值(第 2 行)，然后将候选行表集合按照一致性支持度的大小进行排序(第 3 行)。

设置覆盖率阈值 $thCov$，当满足 $coverage>thCov$ 时程序终止(第 4 行)，选取 $canTS$ 值最大的候选表 rT 作为当前的填值表(第 5 行)；遍历填值表各行，获取每个候选行对应查询表的行号(第 6－7 行)，根据列映射关系确定候选行与查询行的对应列号(第 8 行)；查看标志数组，判断该单元格是否已填值，数组值若为 0 则将候选值填入查询行(第 9－10 行)，最后将该填值表提供的全部新值填入结果表 $augQ$ 中(第 11 行)。将当前的填值表 rT 加入集合 A，并将其从候选行表集合删除(第 12 行)，重新计算各个候选行表的一致性支持度，并排序(第 13－15 行)。重复以上步骤。

由于在计算一致性支持度 $canTS(t)$时，需要与所有已选填值表计算相似度，因此保证了填值表之间较高的相关度，进而保证了整个结果表的一致性。

4 实验结果与分析

4.1 实验设置

1)实验环境：硬件环境为 Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz 3.30GHz，4.00GB 内存，500GB 硬盘；软件环境为 Windows10 专业版 64 位操作系统；开发工具为 Eclipse，Java 语言。

2)数据集：实验使用的数据集是 Search Join 项目[18]提供的 web tables，其中的“English-Language Relational Web Tables 2015”包含大量的关系型表格，初始格式为 json，经过程序转化为 excel 表格，保留 pageTitle，url，context，keyColumnIndex等信息。我们随机选择 100G 左右的数据，从中筛除表格信息错乱、空值较多、主列不明确的表格后，保留数据相对规整、行列清晰、表格之间具有一定相关性的样例组作为最终对比实验所用的 4 个查询样例组——WNY，Patent-US，OBRA 和 Buffalo。

3)评估指标：精确度(precision)、覆盖率(coverage)和一致性(consistency)，评价函数如下：

$$coverage=\frac{|values\_truth\cap values\_found|}{|values\_truth|} \tag{16}$$

$$consistency=\begin{cases}1, & \text{if } |canRT\_selected|=1\\ 1-\dfrac{|canRT\_selected|}{|canRT\_found|}, & \text{otherwise}\end{cases} \tag{17}$$

$$precision=\frac{|values\_truth\cap values\_found|}{|values\_found|} \tag{18}$$

覆盖率指结果表中已扩展值中正确值的数目与全部待扩展列真实值数目的比值。一致性用填值所用的数据源数目的最小化程度来衡量，但不仅仅是源数目的最小化，经过选择行值算法后，$canRT\_selected$ 是最终被选择用来填值的候选行表，$canRT\_found$ 是与查询表有关的全部候选行表。精确度指结果表中已扩展值中正确值的数目与表格已扩展值数目的比值。

4)查询样例特征：各样例组特征如表 6 所列。

表 6　查询样例组特征

Features	WNY	PatentUS	OBRA	Borough
# Row	5	20	42	26
# Column	5	5	4	7
# exColumn	3	3	2	3

其中,# *Row* 为行数目,# *Column* 为列数目,# *exColumn* 为待扩展列数目。

5)阈值:经过多次实验,第 2 节、第 3 节中提及的各个阈值最终确定为 $\sigma_1=0.83$,$\sigma_2=0.68$,$\sigma_3=0.34$,$\sigma_4=0.85$,$\sigma_5=0.58$,$\sigma_6=0.52$,$\sigma_7=0.72$。

4.2　实验方法

针对表格的属性扩展问题,本文在查询表扩展多列的情况下,将系统实现的两种方法进行比较,分析表格扩展算法在覆盖率、一致性、精确度和运行时间上的差异。两种方法介绍如下。

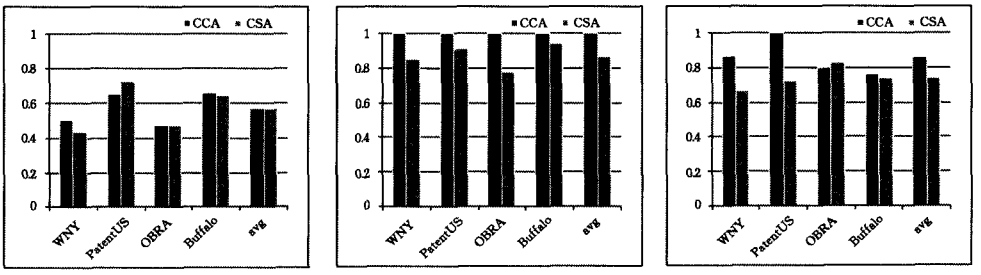
1)CCA:基于列重合度的表格一致性扩展方法。将一致性支持度应用于系统程序,同时考虑扩展列支持度和元组行支持度对候选值的影响,多个扩展列同时扩展。

2)CSA:传统的基于列重合度的表格拆分扩展方法[10-11,18,20]。无论查询表扩展单列还是多列,基本思想均是将数据源表拆分成"实体-属性"二元关系表,仅根据与查询表主列值的匹配度对各个属性列进行扩展。

两种方法都属于间接扩展方法,先根据查询表给定的条件查找种子表,再通过种子表确定与查询表间接匹配的候选表,使用候选表进行表格扩展。

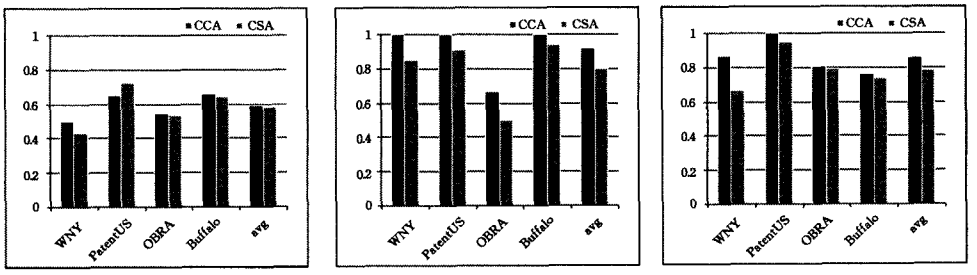
4.3　不同覆盖率阈值下的性能比较

系统对覆盖率阈值 *thCov* 的设置不同,会导致结果表中实际的覆盖率、一致性以及精确度发生变化。程序中针对每个数据集的查询,分别设置覆盖率阈值为 0.0,0.5,0.75 和 1.0,每个阈值下的覆盖率、一致性及精确度指标分别如图 2—图 5 所示。



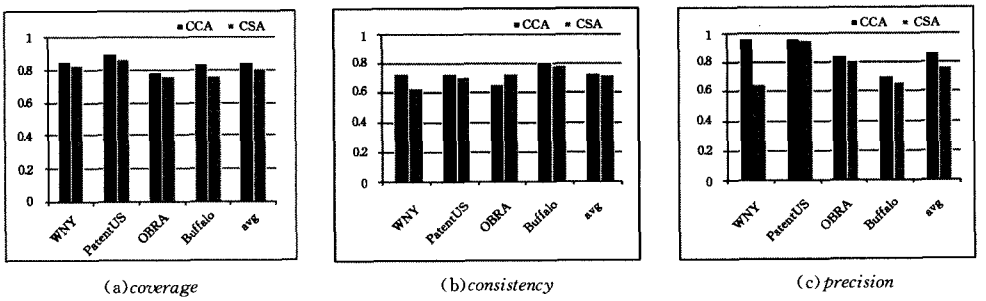
(a)*coverage*　(b)*consistency*　(c)*precision*

图 2　*thCov*=0.0 时的指标值



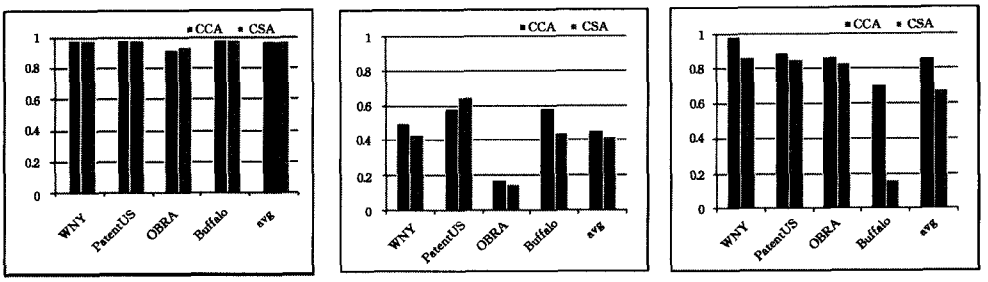
(a)*coverage*　(b)*consistency*　(c)*precision*

图 3　*thCov*=0.5 时的指标值



(a)*coverage*　(b)*consistency*　(c)*precision*

图 4　*thCov*=0.75 时的指标值



(a)*coverage*　(b)*consistency*　(c)*precision*

图 5　*thCov*=1.0 时的指标值

两种方法在各个数据集上设置不同的覆盖率阈值，得到扩展结果表不同的覆盖率、一致性和精确度。当覆盖率阈值 $thCov=0.0$ 时，各个指标如图 2 所示，覆盖率平均达到 50% 左右，一致性接近于 1。当 $thCov=0.5$ 时，图 3 显示的精确度、覆盖率和一致性与图 2 相比没有太大的变化，两种情况下的覆盖率偏低，一致性较好，精确度处于中等水平。当 $thCov=0.75$ 时，指标如图 4 所示，覆盖率提升幅度较大，一致性稍显降低，但在可接受的范围内，精确度也达到预期效果。当 $thCov=1.0$ 时，图 5 所示的指标值中虽然覆盖率几乎达到 100%，但是一致性相对较低。

将程序中的覆盖率阈值分别设置为 0.0，0.45，0.55，0.65，0.7，0.75，0.8，0.85，0.9，0.95，1.0 后，CCA 方法在所有数据集上的平均指标变化如图 6 所示。

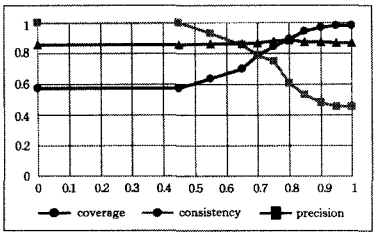


图 6 CCA 的指标均值变化

经过多次实验验证，由图 6 可知，当覆盖率阈值选取为 $thCov\approx0.75$ 时，覆盖率和一致性同时达到比较合适的值，精确度也基本达到最佳，三者的均衡保证了结果表的可靠性和一致性达到期望值。

4.4 运行时间的评估

整个系统的运行时间主要包括读取并预处理源表格、查找种子表、确定候选表及候选行、填值 4 个部分，时间主要花费在预处理和确定候选行两部分。CSA 方法中，数据源经过拆分程序后，n 元的表格根据“实体-属性”被拆成 $n-1$ 个二元表，在进行查询处理时，所进行的读写、查询、计算等次数都大量增加，故 CCA 和 CSA 在查询时间上产生了很大的差异。

由 CCA 和 CSA 查询过程中确定的中间表格数目可知，一致性扩展方法和拆分扩展方法在完成同样的表格查询时，搜索表格的数目相差较大，其查询时间的对比结果如图 7 所示。

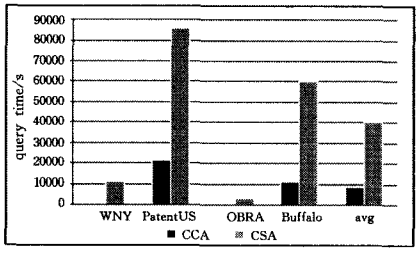


图 7 运行时间对比图

由表 6 可知，查询样例的行列数及扩展列数各不同，当查询样例的行列数及数据源表的行列数较少时，整体所用时间较少。故程序的运行时间与查询样例的大小和候选表的大小及数目接近正比关系。

结束语 本文利用列重合度确定各列间的映射关系，同时计算表格间的相似度，根据一致性支持度的大小选定填值表进行扩展。本文将列重合度、一致性支持度与表格扩展技术相结合，提出一种综合考虑列间关系和元组间关系的表格一致性扩展算法。与现有的表格扩展方法相比，CCA 提高了表格扩展的精确度和覆盖率，在填值时有效维护了实体的一致性。

但是，该方法在实际应用中还有若干问题有待解决，比如采用并行计算降低算法的时间复杂度的可行性等，都是未来需要研究的问题。

参 考 文 献

[1] CAFARELLA M J, HALEVY A, WANG D Z, et al. WebTables: exploring the power of tables on the Web[J]. Proceedings of the Vldb Endowment, 2008, 1(1): 538-549.

[2] CAFARELLA M J, HALEVY A Y, ZHANG Y, et al. Uncovering the Relational Web[C]// International Workshop on the Web and Databases, WEBDB 2008. 2008.

[3] LIAO T, LIU Z T, SUN R. Research and Implementation of Web Table Positioning Technology[J]. Computer Science, 2009, 36(9): 227-230. (in Chinese)
廖涛，刘宗田，孙荣. Web 表格定位技术的研究与实现[J]. 计算机科学，2009，36(9)：227-230.

[4] WANG N, REN H W. Detecting Snapshots for Web Tables[J]. Computer Science, 2015, 42(7): 5-11. (in Chinese)
王宁，任红伟. 网络表格间的快照关系发现[J]. 计算机科学，2015，42(7)：5-11.

[5] CAFARELLA M J, HALEVY A, KHOUSSAINOVA N. Data integration for the relational Web[J]. Proceedings of the Vldb Endowment, 2009, 2(1): 1090-1101.

[6] GONZALE H, HALEVY A Y, JENSEN C S, et al. Google fusion tables: web-centered data management and collaboration [C]// ACM SIGMOD International Conference on Management of Data(SIGMOD 2010). 2010: 1061-1066.

[7] GONZALE H, HALEVY A, JENSEN C S, et al. Google fusion tables: data management, integration and collaboration in the cloud[C]// Proceedings of the 1st ACM Symposium on Cloud Computing. ACM, 2010: 175-180.

[8] SARMA A D, FANG L, GUPTA N, et al. Finding related tables [C]// Acm Sigmod International Conference on Management of Data. ACM, 2012: 817-828.

[9] BALAKRISHNAN S, HALEVY A, HARB B, et al. Applying WebTables in Practice[C]// Biennial Conference on Innovative Data Systems Research. 2015.

[10] YAKOUT M, GANJAM K, CHAKRABARTI K, et al. InfoGather: entity augmentation and attribute discovery by holistic matching with web tables[C] // ACM SIGMOD International Conference on Management of Data. ACM, 2012: 97-108.

[11] ZHANG M, CHAKRABARTI K. InfoGather+: semantic matching and annotation of numeric and time-varying attributes in Web tables[C]// ACM SIGMOD International Conference on Management of Data. 2013: 145-156.

[12] YANG M, DING B, CHAUDURI S, et al. Finding patterns in a knowledge base using keywords to compose table answers[J]. Proceedings of the Vldb Endowment, 2014, 7(14): 1809-1820.

[13] WANG C, CHAKRABARTI K, HE Y, et al. Concept Expansion Using Web Tables[C]// Proceedings of the 24th International

Conference on World Wide Web. 2015:1198-1208.

[14] PIMPLIKAR R,SARAWAGI S. Answering table queries on the web using column keywords[J]. Proceedings of the Vldb Endowment,2012,5(10):908-919.

[15] GUPTA R,SARAWAGI S. Answering Table Augmentation Queries from Unstructured Lists on the Web[J]. Proceedings of the Vldb Endowment,2009,2(1):289-300.

[16] LEHMBERG O,RITZE D,RISTOSKI P,et al. Extending tables with data from over a million websites[C]// Semantic Web Challenge. 2014.

[17] BIZER C. Search Joins with the Web[C]//ICDT. 2014:3.

[18] LEHMBERG O,RITZE D,RISTOSKI P,et al. The Mannheim Search Join Engine[J]. Web Semantics Science Services & Agents on the World Wide Web,2015,35(P3):159-166.

[19] BRAUNSCHWEIG K,THIELE M,EBERIUS J,et al. Column-specific context extraction for web tables[C]// ACM Symposium on Applied Computing. ACM,2015:1072-1077.

[20] EBERIUS J,THIELE M,BRAUNSCHWEIG K,et al. Top-k entity augmentation using consistent set covering[C]//SSDBM. 2015:1-12.

[21] LAUTERT L R,SCHEIDT M M,DORNELES C F. Web table taxonomy and formalization[J]. ACM Sigmod Record,2013,42(3):28-33.

[22] SONG S,ZHANG A,CHEN L,et al. Enriching data imputation with extensive similarity neighbors[J]. Proceedings of the Vldb Endowment,2015,8(11):1286-1297.

(上接第 189 页)

[2] KICZALES G,HILDALE E,HUGUNIN J,et al. An overview of AspectJ[C]// European Conference on Object-oriented Programming. Springer Berlin Heidelberg,2001:327-354.

[3] SPINCZYK O,GAL A,SCHRÖDER-PREIKSCHAT W. AspectC++:an aspect-oriented extension to the C++ programming language[C]// Proceedings of the Fortieth International Conference on Tools Pacific: Objects for internet, mobile and embedded applications. Australian Computer Society, Inc., 2002:53-60.

[4] Aspect R[EB/OL]. [2016-03-01]. http://sowrceforge. net/projects/aspectr.

[5] BONÉR J. What are the key issues for commercial AOP use: how does AspectWerkz address them? [C]//Proceedings of the 3rd International Conference on Aspect-oriented Software Development. ACM,2004:5-6.

[6] HIRSCHFELD R. Aspects-oriented programming with squeak [C]// Objects, Components, Architectures, Services, and Applications for a Networked World. 2003:216-232.

[7] JBoss AOP homepage[EB/OL]. [2016-03-01]. http:// www. jboss. org/jbossaop.

[8] OSSHER H,TARR P. Hyper/J:multi-dimensional separation of concerns for Java[C]// Proceedings of the 22nd International Conference on Software Engineering. ACM,2000:734-737.

[9] JI F H,LI X,LIU Z. rCOS:A refinement calculus of object systems[J]. Theoretical Computer Science,2006,365(1),109-142.

[10] WAND M,KICZALES G,DUTCHYN C. A semantics for advice and dynamic join points in aspect-or ienting programming [J]. ACM Transactions on Programming Languages and Systems (TOPLAS),2004,26(5):890-910.

[11] JAGADEESAN R,JEFFREY A,RIELY J. A calculus of untyped aspect-oriented programs[C]// European Conference on Object-oriented Programming. Springer Berlin Heidelberg, 2003:54-73.

[12] LÄMMEL R. A semantical approach to method-call interception [C]// Proceedings of the 1st International Conference on Aspect-oriented Software Development. ACM,2002:41-55.

[13] WALKER D,ZDANCEWIC S,LIGATTI J. A theory of aspects [J]. ACM SIGPLAN Notices,2003,38(9):127-139.

[14] TUCKER D B,KRISHNAMURTHI S. Pointcuts and advice in higher-order languages[C]// Proceedings of the 2nd International Conference on Aspect-oriented Software Development. ACM,2003:158-167.

[15] MASUHARA H,KICZALES G. Modeling crosscutting in aspect-oriented mechanisms[C]//European Conference on Object-Oriented Programming. Springer Berlin Heidelberg,2003:2-28.

[16] TABAREAU N. Aspect Oriented Programming:a language for 2-categories[C]// Proceedings of the 10th International Workshop on Foundations of Aspect-oriented Languages. ACM, 2011:13-17.

[17] AVGUSTINOV P,HAJIYEV E,ONGKINGCO N,et al. Semantics of static pointcuts in AspectJ[J]. ACM Sigplan Notices, 2007,42(1):11-23.

[18] FIGUEROA I,TANTER É. A semantics for execution levels with exceptions[C]// Proceedings of the 10th International Workshop on Foundations of Aspect-oriented Languages. ACM, 2011:7-11.

[19] GANG X,BO Y,MINGYI Z. A Semantics of Poincuts in AspectJ[J]. IERI Procedia,2013,4:323-330.

[20] XIE G,ZHANG M Y,YANG B. A STATIC SEMA NTICS FOR ASPECTJ[J]. Journal of Computer Tational Information Systems,2012,8(16):6951-6962.

[21] 王砚霖,王世耆. 面向方面编程和 AspectJ[OL/EB]. [2016-03-01]. http://www. creativepioneer. com/paper/AOP_and _AspectJ. pdf.

[22] HOARE A R C,HE J. Unifying theories of program ming[M]. Englewood Cliffs:Prentice Hall,1998.

[23] MOLDEREZ T,JANSSENS D. Modular Reasoning in Aspect-oriented Languages from a Substitution Perspec tive[J]. Transactions on Aspect-oriented Software Develop ment XII. Springer Berlin Heidelberg,2015:3-59.

[24] ZHANG Q,KHEDRI R. On the weaving process of aspect-oriented product family algebra [J]. Journal of Logical and AlgebraicMethods in Programming,2016,85(1):146-172.

[25] 陆钟万. 面向计算机科学中的数理逻辑(第二版)[M]. 北京:科学出版社,2002:117-118.