

异构信息空间中时间感知的实体集成框架

杨 丹¹ 陈 默² 申德荣²

(辽宁科技大学软件学院 鞍山 114051)¹ (东北大学计算机科学与工程学院 沈阳 110004)²

摘 要 异构信息空间中的实体和关联关系普遍具有时间信息、多种时间版本的实体数据共存,而传统的实体集成忽略了时间信息,不支持时间维度上的集成。提出一种异构信息空间中时间感知的实体集成框架 T-EI,从大量异构实体数据中聚集事实形成干净的、完整的、具有时间信息的实体概貌,进而支持时间感知的实体搜索。T-EI 利用实体及关联关系所具有的时间信息提出时间感知的实体识别算法,并通过考虑数据时效性提出时间感知的数据融合算法。在真实数据集上的实验结果表明了 T-EI 的可行性和有效性。

关键词 异构信息空间,时间感知,实体集成,实体概貌

中图分类号 TP311.13 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.02.016

Time-aware Entity Integration Framework in Heterogeneous Information Spaces

YANG Dan¹ CHEN Mo² SHEN De-rong²

(College of Software, University of Science and Technology LiaoNing, Anshan 114051, China)¹

(College of Computer Science and Engineering, Northeastern University, Shenyang 110004, China)²

Abstract In heterogeneous information spaces, entities and associations generally have time information, entities of multiple time versions coexist. While traditional entity integration (EI) ignores time information, does not support the integration on the time dimension. In this paper, a time-aware EI framework in heterogeneous information spaces T-EI was proposed, which can aggregate large collections of heterogeneous entities into a set of clean and complete entity profiles with time information to support time-aware entity search. T-EI adopts time-aware entity resolution algorithm leveraging time information of entities and associations, and adopts time-aware data fusion algorithm considering data currency. Experimental results on the real data sets demonstrate the feasibility and effectiveness of T-EI.

Keywords Heterogeneous information spaces, Time-aware, Entity integration, Entity profile

1 引言

在异构信息空间^[1]中,用户经常需要搜索各种感兴趣的、完整的、具有时间信息的实体信息。例如,在医疗健康领域,患者为了治疗某种疾病,需要了解某医院的医院等级、专家大夫、住院环境、费用等信息。而多数情况下这些实体信息分散在多个不同的数据源中,来自单个数据源的实体数据信息往往不完整,各个数据源之间的同一个实体信息有时是重叠的,即使来自同一个数据源中,现实世界的同一个实体的数据和信息通常存在于文本、HTML 网页、XML 或其他格式文件,并且遍布多个文件。这种提取实体数据、整合实体信息,然后建立特定领域的实体的过程,被称为实体集成^[2]。

异构信息空间中的实体及关联关系往往具有时间信息且随时间演化,一个实体可能演化成一个或多个不同的实体。此外,在异构信息空间中,来自一个或多个数据源的同一实体的多个不同时间版本的表象数据共存是很普遍的情况,它们

可能是某个时间的真实值,即曾经是正确的。因此,需要时间感知的实体集成。

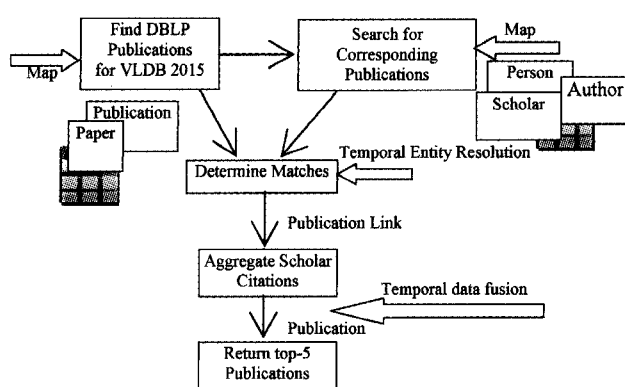


图1 时间感知的实体集成与实体搜索的例子

图1给出了一个异构信息空间中时间感知的实体集成与实体搜索的例子。从多个数据源如DBLP, Google Scholar进

到稿日期:2016-08-29 返修日期:2016-11-20 本文受国家自然科学基金项目(61402213,61402093),教育部中央高校基本科研业务费(N141604001),辽宁省自然科学基金(2015020018)资助。

杨 丹(1978—),女,博士,副教授,CCF 会员,主要研究方向为数据集成、实体搜索,E-mail: asyangdan@163.com;陈 默(1983—),女,博士,讲师,CCF 会员,主要研究方向为空间数据处理;申德荣(1964—),女,博士,博士生导师,CCF 高级会员,主要研究方向为数据集成、Web 数据管理等。

行时间感知的实体搜索:“VLDB2015 会议论文中被引用最多的 TOP5 论文”。

本文提出时间感知的实体集成框架 T-EI,该框架支持时间感知的实体识别和数据融合。经过 T-EI 集成后的实体支持时间感知的实体搜索,允许具有时间上下文的查询,例如,具有时间上下文的查询 {Xin luna Dong, Paper}@2015。

2 相关工作

实体集成主要涉及信息提取、模式匹配、实体识别与数据融合等技术。针对本文研究内容,下面将介绍 3 方面的相关工作:1)实体集成;2)实体识别;3)数据融合。

2.1 实体集成

mashups^[2]是一个很流行的从多个数据源集成 Web 数据的方法。文献[3]采用一个轻量级的编程方法进行 on-the-fly 数据集成。随着云计算的出现和发展,文献[4]提出了一个基于云计算的数据集成系统。实体集成的相关工作^[5]提出了一种实体识别和实体集成的脚本语言 HIL。为实体引入最早时间和最晚时间两个属性来支持时态聚合。

2.2 实体识别

利用实体识别技术可以对数据集成中产生的重复记录进行检测并整合,有效地消除数据源内部及数据源之间的数据不一致性。在时间记录链接方面,文献[6]提出时间衰变的概念来捕捉时间在实体的属性值演化上的作用,提出了时间记录的识别算法。文献[7]针对时间效率,提出了一种两阶段的聚类算法 SFDS 来匹配时间记录,使用了文献[6]中的方法来学习时态模型。而文献[8]为匹配时间记录提出了一个更加详细的为实体演化建模的时态模型,用于捕捉实体演化的信息。文献[9]提出转变模型来捕捉复杂的随时间演化的实体属性值的变化,并在时间记录链接时考虑数据源的新鲜度而提出数据源感知的匹配算法。

2.3 数据融合

实体识别是用于解决标识冲突,而数据融合^[10]是用于解决数据冲突。针对数据内容的冲突的解决方法大致分为两类:1)基于关系扩展的方法,通过扩展关系操作及聚合函数;2)从多个冲突值中选择真值的方法。文献[11]针对数据的多样性,将记录链接与数据融合相结合。文献[12]针对大数据量的考虑,提出在线数据融合。数据时效性^[13]研究识别数据源中实体的当前值,使用当前值(最新值)来回答查询,是影响数据质量的重要因素之一。文献[14]强调数据一致性和数据时效性,在缺失数据源准确性信息的情况下,提出一个基于数据时效性和数据一致性的冲突消解模型。现有的数据冲突消解方法多数是通过扩展关系操作及聚合函数,或利用条件函数依赖保证数据一致性,得到无冲突的结果,而没有考虑数据的时效性。

在时间感知方面,文献[15]提出为知识库中的实体生成事件和关系的时间轴方法;文献[16]关注时态数据的集成,提出偏好感知的运算符 PRAWN 和算法。尽管围绕实体集成,在模式匹配、实体识别、数据融合等方面已经取得了许多研究成果,但是由于异构信息空间中的实体具有时间信息、实体信息随时间演化、多时间版本信息共存等特性,已有的实体集成的相关工作没有解决实体集成在时间感知方面的问题。

3 时间感知的实体集成框架

本节首先介绍相关的背景知识和问题定义,然后给出时间感知的实体集成框架 T-EI。

3.1 背景知识与问题定义

定义 1(时态实体, Temporal Entity) 异构信息空间中的具有时间信息的实体,表示为实体属性(值)及时间戳的集合,即 $e = \{(attr_1, v_1, t_1), \dots, (attr_n, v_n, t_n)\}$,每个实体对应一组属性 $attr_1 - attr_n$ 、以及这些属性在 $t_1 - t_n$ 时刻的属性值 $v_1 - v_n$,其中时间戳 $t_1 - t_n$ 可以相同也可以不同。实体 e 在时刻 t_i (或时间间隔 ts) 的属性 $attr$ 的值表示为 $(e.attr)_i.value$ 或 $(e.attr)_{ts}.value$ 。

定义 2(时态实体关联关系, Temporal Entity Association) 表示为四元组 $ass = \langle R_{type}, S_{source}, T_{target}, Label \rangle$,其中 R_{type} 表示关联关系所属的关系类型; S_{source}, T_{target} 分别表示构成二元关联关系的两个端点; $Label$ 表示关联关系标签,由关联关系名 $assName$ 和时间信息 $@T$ 组成,即 $Label ::= assName @ T$; 时间信息 T 可以是一个时间点 t ,也可以是一个时间间隔 $ts = [ts_{begin}, ts_{end}]$,即 $T \in \{t, ts\}$ 。

定义 3(具有时间信息的实体概貌, Entity Profile with Temporal Information) 经过实体识别、数据融合后的正确的、完整的、具有时间信息的实体描述。实体 e 的实体概貌记作 ϵ ; 与实体概貌 ϵ 相关联的时间信息记作 $\epsilon.t$ 或 $\epsilon.ts$ 。

定义 4(时间感知的实体集成, Time-aware Entity Integration) 提取异构数据源中同一种实体类型 et 的原始实体数据和时间信息 T ,经过实体的信息集成加工,建立特定领域的、丰富的、清洁的、具有时间信息的实体概貌 ϵ 的过程,得到现实世界中同一个实体沿着时间轴的概貌。

定义 5(时间感知的实体搜索, Time-aware Entity Search) 支持具有时间上下文的关键词查询 Q 进行时态实体搜索,其中查询 $Q = \{k_1, k_2, \dots, k_n\} @ \{t, ts\}$ 由两部分组成, $\{k_1, k_2, \dots, k_n\}$ 部分为关键词(记作 Q_{text}), $@ \{t, ts\}$ 部分是可选项,表示查询 Q 的时间上下文(记作 Q_{time}), t 表示某个(单个)时间点, ts 表示某个连续的时间间隔 $ts = [ts_{begin}, ts_{end}]$ 。两部分的相关度都被满足的程度越高,实体在结果实体集中的排名越高。

例如,使用具有时间上下文的查询 {Xin Luna Dong, paper}@2015; {coauthor, Halevy}@[2007, 2009] 进行时间感知的实体搜索。

3.2 T-EI 框架概览

本文提出的异构信息空间中时间感知的实体集成框架 T-EI 如图 2 所示。

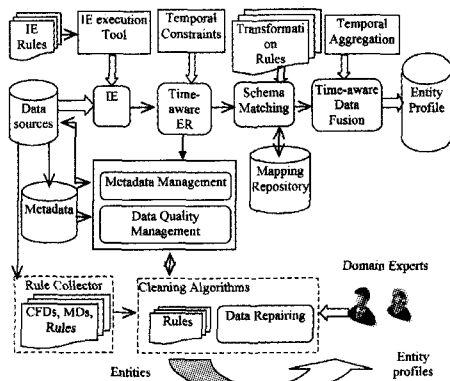


图 2 时间感知的实体集成框架 T-EI

T-EI 的目标是提供一个异构信息空间中时间感知的实体集成框架,允许用户将多个异构数据源看作透明的、单独的、清洁的数据源,进行时间感知的实体搜索,得到一个完整的、具有时间信息的实体概貌集。

T-EI 的时间感知特性主要体现在以下方面:1)时间感知的实体识别;2)基于数据时效性的时间感知的数据融合;3)支持时间感知的实体搜索。

由于篇幅所限,本文将重点介绍 T-EI 框架中所采用的时间感知的实体识别算法和时间感知的数据融合算法。

4 时间感知的实体识别

T-EI 采用时间感知的实体识别算法,利用实体的时间信息进行时间感知的 Blocking;之后在对每个 Block 中的实体表象进行聚类过程中也利用实体的时间信息,采用时间感知的相似度度量方法。

4.1 时间感知的 Blocking 算法

利用时态实体的时间信息,设计与其他属性的 Blocking 组合键,并且利用实体的时间信息过滤大的 Block,将待识别的实体划为多个时间感知的 block,记作 t-block。时间感知的 Blocking 算法如算法 1 所示。首先采用传统的 Blocking 技术将实体表象划分成多个 blocks;接着在时间轴(时间维度)上对 block 再进行分裂和过滤得到 t-block,即若 block 中表象的最大与最小时间的时间间隔 $\Delta t = \text{MAX}(r_i) - \text{MIN}(r_i)$ 大于某个阈值 k ,就分裂成多个 t-blocks。

算法 1 Time-aware Blocking Algorithm

Input: 表象集合 $R = \{r_1, r_2, \dots, r_n\}$

Output: t-blocks

1. select Blocking key;
2. blocking R according to blocking key $B = \{b_1, b_2, \dots, b_n\}$;
3. for each block b_i do
4. sorting b_i according to time info r_i, t_i ;
5. if $\Delta t = \text{Max}(r_i, t) - \text{Min}(r_i, t) > k$ then
6. splitting b_i according to time span;
7. if $|b_i| > \text{size}$ then
8. select filtering viable;
9. filtering b_i according to length filtering or count filtering;
10. end if
11. end for
12. return all t-blocks.

4.2 时间感知的相似度度量方法

两个实体表象 r_1, r_2 的相似度计算基于时态属性相似度和时态关系相似度的线性组合如式(1)所示:

$$\text{SIM}_T(r_i, r_j) = \text{SIM}_{TA}(r_i, r_j) + (1 - \alpha) \text{SIM}_{TR}(r_i, r_j) \quad (1)$$

$$\text{SIM}_{TA}(r_i, r_j) e^{-\beta|t_i - t_j|} \text{SIM}(r_1, r_2) \quad (2)$$

$$\text{SIM}_{TR}(r_i, r_j) e^{-\beta|t_i - t_j|} \text{SIM}(r_1, r_2) \quad (3)$$

其中, $\text{SIM}_{TA}(r_i, r_j)$ 为时态属性相似度; $\text{SIM}_{TR}(r_i, r_j)$ 为时态关系相似度; $e^{-\beta|t_i - t_j|}$ 是时间衰变函数, β 是衰变函数的参数(不同的实体类型可以取不同的 β 值),用来减少两个时间间隔很长的相似属性值或相似关系的相似度分值; t_i, t_j 分别是实体表象 r_i, r_j 的时间信息; α 和 $1 - \alpha$ 分别为两部分的权重系数。

4.3 时间感知的聚类算法

T-EI 基于层次聚类的时间感知的聚类算法 TC 如算法 2

所示。TC 利用实体关联关系、时间信息所带来的识别证据,在实体表象相似度计算上考虑时间演化作用的影响,在识别过程中引入语义约束和时态约束。

算法 2 Time-aware Clustering Algorithm TC

Input: 同一个 t-block 中的表象 $R = \{r_1, r_2, \dots, r_{|R|}\}$ sorted in increasing temporal order
threshold of SIM: τ

Output: a set of clusters $C = \{C_1, C_2, \dots, C_{|C|}\}$

//initialization;

1. $C_0 \leftarrow \{C_1, C_2, \dots, C_{|R|}\}$

2. $L = 0$; //L is the iterative step number

//Iterative.

3. $L \leftarrow L + 1$

4. for all two clusters $C_i \in C_{L-1}, C_j \in C_{L-1}$ do

5. compute $\text{SIM}(C_i, C_j)$

6. if $\text{SIM}(C_i, C_j) > \tau$ then

7. new cluster $C_{ij} \leftarrow \text{Merge}(C_i, C_j)$

8. Update cluster's time labels

9. $C_L \leftarrow C_{L-1} - \{C_i, C_j\} \cup \{C_{ij}\}$

10. else $\text{SIM}(C_i, C_j) \leftarrow 0$

11. end if

12. end if

13. end for

14. return C_L

算法分为初始化和迭代两阶段,在迭代阶段利用时间感知的相似度度量方法得到聚类簇集合 C 。若两个簇的相似度大于阈值 τ (实验中设置为 0.8),则将两个簇合并。

5 时间感知的数据融合

数据融合是将实体识别得到的同一实体的多个表象合成单个的、一致的、更丰富的实体表示,以解决来自不同异构数据源的不一致冲突问题,即冲突消解。例如,来自不同数据源的同一作者实体 e 不同时期的附属单位属性值分别是读博士时期($e.affiliation$)₂₀₀₉ = 〈华盛顿大学〉和博士毕业后参加工作的时期($e.affiliation$)₂₀₁₃ = 〈谷歌公司〉,两个属性值分别在不同时间(段)是正确的。

5.1 相关定义

定义 6(实体属性值时间序列, Time Sequence) 表示实体 e 在属性 $attr$ 上的属性值($v_1 - v_m$)随时间($t_1 - t_m$)的变化,记作 $TSe_{attr} = [(v_1, t_1), \dots, (v_m, t_m)]$, 并且 $e.attr(t_i) \neq e.attr(t_{i+1})$, 时序关系 $t_1 < attr t_2$ 表示 t_2 时刻的实体属性 $attr$ 的值比 t_1 时刻的值新。

定义 7(实体属性值时态约束, Temporal Constraint) 表示实体 e 的属性 $attr$ 值在时态上的约束集合,记作 $TC_{e,attr}$ 。实体 e 在属性 $attr$ 上满足时态约束记作: $e.attr.value$ staisfy $TC_{e,attr}$ 。

除了语义完整性约束之外(例如,在一个时间点或时间段,实体的某个属性值只能唯一),实体属性值还需要满足时态约束。例如,美国总统的任期为 4 年,可以连任 1 届。

5.2 数据融合与冲突消解规则

数据融合使用实体识别的结果,将提取的事实融合在一起成为更加丰富的实体,并采用时间感知的冲突消解策略为冲突的属性值找到时间感知的、最好的属性值。基于 Deci-

ding 策略^[11],对于非聚合(non aggregate)实体属性,T-EI 给出如下时间感知的数据融合与冲突消解规则。

Rule 1 时态合并规则:对于两个时间间隔 $ts_i, ts_j (i \neq j)$,如果存在 $(e.attr)_{ni}.value = (e.attr)_{nj}.value$,并且 $ts_i \supseteq ts_j$,那么 $e.ts = e.ts_i$ 。如果 $ts_i \cap ts_j \neq \emptyset$,那么 $e.ts = [ts_{begin}, ts_{jend}]$ 。对于两个时间点 $t_i, t_j (i \neq j)$,如果存在 $(e_1.attr)_i.value = (e_2.attr)_j.value$,并且 t_i 和 t_j 在时间序列上相邻,那么 $(e.attr).ts = [t_i, t_j]$ 。

Rule 2 新近(recency)规则:对于属性值缺失但不允许为空的属性,当前的属性值采用表象中时间最大时刻的属性值,即 $(e.attr)_{current}.value = (e.attr)_{\max(TQ)}.value$ 。

Rule 3 向前就近规则:某个时刻 t 属性值为空,即 $(e.attr)_t.value = \text{null}$,如果 $t_1 < t < t_2$,并且 $(e.attr)_{[t_1, t_2]}.value = \text{null}$,那么 $(e.attr)_t.value = (e.attr)_{t_1}.value$ 。

Rule 4 时效约束规则:如果 $t_i < t_j$ 并且 $(e.attr)_{t_i}.value \neq (e.attr)_{t_j}.value$,那么 $(e.attr)_{t_i}.value < (e.attr)_{t_j}.value$,即 $e.attr.value$ satisfy TSe_{attr} 。

Rule 5 保守冲突消解规则:如果 $ts_{i\begin{smallmatrix} begin \\ end \end{smallmatrix}} < ts_{j\begin{smallmatrix} begin \\ end \end{smallmatrix}}$, $ts_i \cap ts_j \neq \emptyset$,并且 $(e.attr)_{n_i}.value \neq (e.attr)_{n_j}.value$,那么 $(e.attr)_{[ts_{i\begin{smallmatrix} begin \\ end \end{smallmatrix}}, ts_{j\begin{smallmatrix} begin \\ end \end{smallmatrix}}]}.value = (e.attr)_{n_i}.value$, $(e.attr)_{[ts_{j\begin{smallmatrix} begin \\ end \end{smallmatrix}}, ts_{i\begin{smallmatrix} begin \\ end \end{smallmatrix}}]}.value = (e.attr)_{n_j}.value$ 。

Rule 6 数据源的统计多数规则:如果来自不同数据源,在相同时间点 t 或时间间隔 ts 上同一实体的同一属性的值不同,即 $(e_1.attr)_t.value \neq (e_2.attr)_t.value$, $(e_1.attr)_s.value \neq (e_2.attr)_s.value$,那么采用数据源(dss)统计中占多数的属性值,即 $(e.attr)_t.value = \text{Major}(dss).value$; $(e.attr)_s.value = \text{Major}(dss).value$ 。

对于聚合(aggregate)实体属性,T-EI 根据相同时间段内或重叠时间段内多个实体($|e_k|$ 个)的属性值进行算数和累加,即 $(e.attr)_s.value = \sum_{k=1}^{|e_k|} (e_k.attr)_t.value$ 。

5.3 时间感知的数据融合算法

时间感知的数据融合算法 TDF 如算法 3 所示。TDF 首先根据两个实体表象的时间信息分为下面 3 种情况:1)两个时间间隔存在包含关系(2—8 行);2)两个时间间隔完全不相交(9—17 行);3)两个时间间隔部分重叠(18—20 行)。每一种情况再根据同一个属性的值是否相同分为两种情况:1)相同;2)不同。对于相同的情况,将时间信息进行更新;对于不相同的情况,进行判断和冲突消解。

算法 3 Time-aware Data Fusion Algorithm TDF

Input: References of a entity $\mathcal{R} = \{r_1, r_2, \dots, r_n\}$ sorted in increasing temporal order

Output: entity profile e

1. For each $r_i \in \mathcal{R}, r_j \in \mathcal{R} j=i+1$ do
2. case $r_j.ts \supseteq r_i.ts$
3. if $(r_i.attr).value = (r_j.attr).value$ then
4. $r_{ij} \leftarrow \text{merge } r_i, r_j$;
5. update $r_{ij}.ts \leftarrow r_j.ts$;
6. return r_{ij} ;
7. else update $r_j.ts_{begin} \leftarrow r_i.ts_{begin}$;
8. return r_i, r_j ;
9. case $r_i.ts \cap r_j.ts \neq \emptyset$
10. if $(r_i.attr).value = (r_j.attr).value$ then

11. $r_{ij} \leftarrow \text{merge } r_i, r_j$;
12. update $r_{ij}.ts \leftarrow [r_i.ts_{begin}, r_j.ts_{end}]$;
13. return r_{ij} ;
14. else if satisfy TC_{attr} then
15. return r_i, r_j ;
16. else update $r_i.ts_{end} \leftarrow r_j.ts_{begin}$;
17. return r_i, r_j ;
18. case $r_i.ts \cap r_j.ts = \emptyset$
19. if satisfy TSe_{attr} then
20. return r_i, r_j ;

6 实验

6.1 实验数据

以学术文献领域的真实数据集 DBLP 和 DBLife 作为实验数据,集成数据集的统计情况如表 1 所列。使用作者实体集成和论文实体集成两种场景。实验所使用的计算机配置是 3.5GHz Intel CPU、8GB 内存和 500G 硬盘。

表 1 集成数据集的统计信息

Data Set	# authors	# papers	Time span/year
DBLP	635	9411	2000—2013
DBLife	938	6837	2000—2013

6.2 数设置

式(1)中 α 的设置如下:

$$\alpha = \frac{|Attr|}{|Attr| + |Rel|} \quad (4)$$

其中, $|Attr|$ 表示用于计算表象相似度的属性个数; $|Rel|$ 表示用于计算关系相似度的关系个数。

6.3 实验评价与分析

实验分别从时间感知的实体识别算法 TC、时间感知的数据融合算法 TDF 的性能以及 T-EI 的可伸缩性 3 个方面来评价。

时间感知的实体识别算法 TC 的性能:评价指标采用基于聚类的实体识别的正确率、召回率和 F 值,如式(5)所示。实验结果如表 2 所列。从表 2 中可以看出,在作者和论文两种实体类型上,F 值都在 0.935 以上。

$$P = \frac{|TP|}{|TP| + |FP|}$$

$$R = \frac{|TP|}{|TP| + |FN|}$$

$$F\text{-Measure} = \frac{2P \cdot R}{P + R} \quad (5)$$

其中, TP 表示表象对一个簇的判断是正确的情况; FP 表示表象对一个簇的判断是错误的情况; FN 表示表象对不为一个簇的判断是错误的情况。

表 2 时间感知的实体识别算法 TC 的性能

Entity Type	P	R	F-Measure
Author	0.93	0.94	0.935
Paper	0.96	0.95	0.955

时间感知的数据融合算法 TDF 的性能:使用 $F\text{-measure}$ (见式(6))评价数据融合的性能。其中, P 是被正确推断出的个数与推断的总数的比例; R 是被正确推断出的个数与具有冲突值的实体表象总数。时间感知的数据融合实验结果如表 3 所列。从表 3 中可以看出,在作者和论文两种实体上,时间

感知的数据融合取得了较高的 F 值,且都在 0.93 以上。

$$F\text{-measure} = \frac{2P \cdot R}{P + R} \quad (6)$$

表 3 时间感知的数据融合算法 TDF 的性能

Entity Type	P	R	F-Measure
Author	0.92	0.94	0.93
Paper	0.97	0.95	0.96

T-EI 的可伸缩性:不同数据量下的执行时间如图 3 所示。从图 3 中可以看出,在两种实体类型(作者(Author)、论文(Paper))上随着数据量的增大,执行时间缓慢增加,这说明 T-EI 具有较好的可伸缩性。

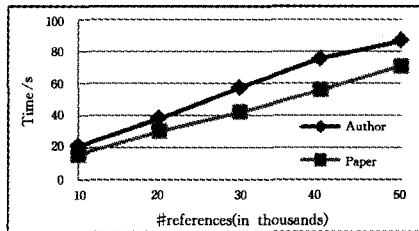


图 3 T-EI 的可伸缩性

结束语 针对异构信息空间中来自异构数据源的具有时间信息的异构实体,提出一个时间感知的实体集成框架 T-EI。在实体识别和数据融合阶段,提出时间感知的实体识别算法 TC 和时间感知的数据融合算法 TDF。在真实数据集上的实验结果表明了 T-EI 的可行性和有效性。

参考文献

- [1] IOANNOU E, NIEDEREE C, WOLFGAN N. Probabilistic entity linkage for heterogeneous information spaces [C] // Proc of CAISE, 2008.
- [2] LORENZO, HACID, PAIK, et al. Data integration in mashups [J]. SIGMOD Record, 2009, 38(1): 59-66.
- [3] EENDRULLIS S, THOR A, RAHM E. WETSUIT: an efficient mashup tool for searching and fusing web entities[J]. PVLDB, 2012, 5(12): 1970-1973.
- [4] THOR A, RAHM E. CloudFuice: A flexible cloud-based data integration System [C] // Proc of 10th Intl. Conference on Web Engineering (ICWE), 2011.
- [5] HERNADEZ M, KOUTRIKA G, KRISHNAMURTHY R. HIL: a high-level scripting language for entity integration [C] // Proc of EDBT, 2013.
- [6] LI P, DONG X. Linking temporal records [C] // Proc. of VLDB, 2011.
- [7] CHIANG Y H, DOAN A H, NAUGHTON J F. Modeling Entity evolution for temporal record matching [C] // Proc. of SIGMOD, 2014: 1175-1186.
- [8] CHIANG Y H, DOAN A H, NAUGHTON J F. Tracking entities in the dynamic world: a fast algorithm for matching temporal records [J]. PVLDB, 2014, 7(6): 469-480.
- [9] LI F R, LI M, HSU W, et al. Linking temporal records for profiling entities [C] // Proc. of SIGMOD, 2015: 593-605.
- [10] BLEIHOLDER J, NAUMANN F. Data Fusion [J]. ACM Comput. Surv., 2008, 41(1): 137-153.
- [11] DONG X, NAUMANN F. Data fusion-resolving data conflicts for integration [C] // Proc. of VLDB, 2009.
- [12] LIU X, DONG X L, Ooi B C, et al. Online data fusion [J]. PVLDB, 2011, 4(12): 932-943.
- [13] FAN W F, GEERTS F, et al. Determining the currency of data [C] // Proc. of PODS, 2011.
- [14] FAN W F, GEERTS F, TANG N, et al. Inferring data currency and consistency for conflict resolution [C] // Proc. of ICDE, 2013: 470-481.
- [15] ITHOFF T, DONG X L, MURPHY K. TimeMachine: Timeline Generation for Knowledge-Base Entities [C] // Proc. of KDD, 2015.
- [16] ALEXE B, ROTH M, WANG C T. Preference-aware Integration of Temporal Data [J]. PVLDB, 2014, 8(4): 365-376.
- [17] LU W, LOANNIDIS S, BHAGAT S, et al. Optimal Recommendations under Attraction, Aversion, and Social Influence [C] // Proceedings of the 20th International Conference on Knowledge Discovery and Data Mining, New York, USA: ACM, 2014: 657-666.
- [18] YAO W L, HE J, HUANG G Y, et al. SoRank: Incorporating Social Information into Learning to Rank Models for Recommendation [C] // Proceedings of the 23th ACM International Conference on World Wide Web, Seoul, Korea: ACM, 2014: 409-410.
- [19] DING X T, JIN X M, LI Y J, et al. Celebrity Recommendation with Collaborative Social Topic Regression [C] // Proceedings of the Twenty-third International Conference on Artificial Intelligence, Beijing, China: ACM, 2013: 2612-2618.
- [20] PURUSHOTHAM S, LIU Y, KUO C. Collaborative topic regression with social matrix factorization for recommendation systems [C] // Proceedings of the 29th ACM International Conference on Machine Learning, Edinburgh, Scotland, UK: ACM, 2012: 1255-1265.
- [21] RENDLE S, FREUDENTHALER C, GANTNER Z, et al. BPR: Bayesian personalized ranking from implicit feedback [C] // Proceedings of the 22nd International Conference on Uncertainty in Artificial Intelligence, Montreal, Canada, 2009: 452-461.
- [22] PATEREK A. Improving regularized singular value decomposition for collaborative filtering [C] // Proceedings of KDD Cup and Workshop, ACM Press, 2007: 39-42.
- [23] HU Y, KOREN Y, VOLINSKY C. Collaborative filtering for implicit feedback datasets [C] // Proceedings of the IEEE International Conference on Data Mining, Pisa, Italy: IEEE, 2008: 263-272.
- [24] PAN W K, CHEN L. GBPR: Group Preference based Bayesian Personalized Ranking for One-Class Collaborative Filtering [C] // Proceedings of the Twenty-third International Conference on Artificial Intelligence, Beijing, China: ACM, 2013: 3007-3011.
- [25] YU L, PAN R, LI Z F. Adaptive social similarities for recommender systems [C] // Proceedings of the fifth ACM Conference on Recommender Systems (RecSys), 2011: 257-260.

(上接第 92 页)

Artificial Intelligence. Barcelona, Catalonia, Spain: ACM, 2011: 2644-2649.