ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.2, February 2008, pp.194−208
DOI: 10.3724/SP.J.1001.2008.00194

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种基于语义及统计分析的Deep Web实体识别机制*

寇 月[1+], 申德荣[1], 李 冬[2], 聂铁铮[1]

[1](东北大学 信息科学与工程学院,辽宁 沈阳 110004)

[2](东软集团有限公司 商用软件事业部,辽宁 沈阳 110179)

A Deep Web Entity Identification Mechanism Based on Semantics and Statistical Analysis

KOU Yue[1+], SHEN De-Rong[1], LI Dong[2], NIE Tie-Zheng[1]

[1](College of Information Science and Engineering, Northeastern University, Shenyang 110004, China)

[2](Business Software Division, Neusoft Group Ltd., Shenyang 110179, China)

+ Corresponding author: Phn: +86-24-83691218, Fax: +86-24-23895654, E-mail: kouyue@ise.neu.edu.cn, http://www.neu.edu.cn

Kou Y, Shen DR, Li D, Nie TZ. A deep Web entity identification mechanism based on semantics and statistical analysis. *Journal of Software*, 2008,19(2):194−208. http://www.jos.org.cn/1000-9825/19/194.htm

Abstract: According to analyzing the traditional entity identification methods, a deep Web entity identification mechanism based on semantics and statistical analysis (SS-EIM) is presented in this paper, which includes text matching model, semantics analysis model and group statistics model. Also a three-phase gradual refining strategy is adopted, which includes text initial matching, representation relationship abstraction and group statistics analysis. Based on the text characteristics, semantic information and constraints, the identification result is revised continuously to improve the accuracy. By performing the self-adaptive knowledge maintenance strategy, the content of representation relationship knowledge database can be more complete and effective. The experiments demonstrate the feasibility and effectiveness of the key techniques of SS-EIM.

Key words: deep Web; data integration; entity identification; data deduplication; representation consolidation

摘 要: 分析了常见的实体识别方法,提出了一种基于语义及统计分析的实体识别机制(deep Web entity identification mechanism based on semantics and statistical analysis,简称 SS-EIM),能够有效解决 Deep Web 数据集成中数据纠错、消重及整合等问题.SS-EIM 主要由文本匹配模型、语义分析模型和分组统计模型组成,采用文本粗略匹配、表象关联关系获取以及分组统计分析的三段式逐步求精策略,基于文本特征、语义信息及约束规则来不断精化识别结果;根据可获取的有限的实例信息,采用静态分析、动态协调相结合的自适应知识维护策略,构建和完善表象关联知识库,以适应 Web 数据的动态性并保证表象关联知识的完备性.通过实验验证了 SS-EIM 中所采用的关键技术的可行性和有效性.

关键词: deep Web;数据集成;实体识别;数据消重;表象整合

中图法分类号: TP393 **文献标识码**: A

随着信息技术的不断发展,Web上的信息量呈爆炸性增长.按照所蕴含信息深度的不同,可以将Web划分为

* Supported by the National Natural Science Foundation of China under Grant No.60673139 (国家自然科学基金)
Received 2007-08-31; Accepted 2007-12-05

Surface Web和Deep Web两大类.统计数据表明:Deep Web蕴含的信息量及数据访问量等都远远高于Surface Web[1].因此,随着Web数据库的不断增长,能够自动获取蕴含在Deep Web中的数据资源并对其进行大规模集成显得尤为重要.然而,数据源内部及数据源之间的数据往往存在数据不一致及数据重复等问题,如果不对这些低质量的数据进行预处理而直接作为查询结果返回,将严重影响Deep Web的查询效率.

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以及数据源之间的数据不一致性.以论文检索网站ACM和DBLP为例,若要查询John Allen发表的论文信息,两个网站上的结果数据如图 1 所示.从图 1 中可以看出,由于拼写方式或定义格式的不同,无论是数据源内部还是数据源之间都具有不同表现形式但对应于同一事物的结果数据.我们将与现实世界一一对应的事物或事件称为实体,例如姓名为John Allen的某个具体的人;将实体的不同表现形式定义为实体的表象,例如可以将这个人表示成John Allen,J. Allen等.如果直接将这些表象作为结果返回,则无疑将增加用户对其进一步分类、筛选的负担.因此,为了提高结果数据的质量,需要事先通过实体识别技术将这些表象按照实体类别聚类,并将聚类结果以知识的形式存储,以指导实际查询中的资源整合.针对此例,就是要判断:DBLP网站中的表象r_2,r_3与r_4是否代表同一个人;ACM中r_1对应于DBLP中r_2,r_3与r_4的哪个人的信息.因此,通过实体识别技术为用户提供高质量的集成结果是Deep Web数据集成中必不可少的一个过程.

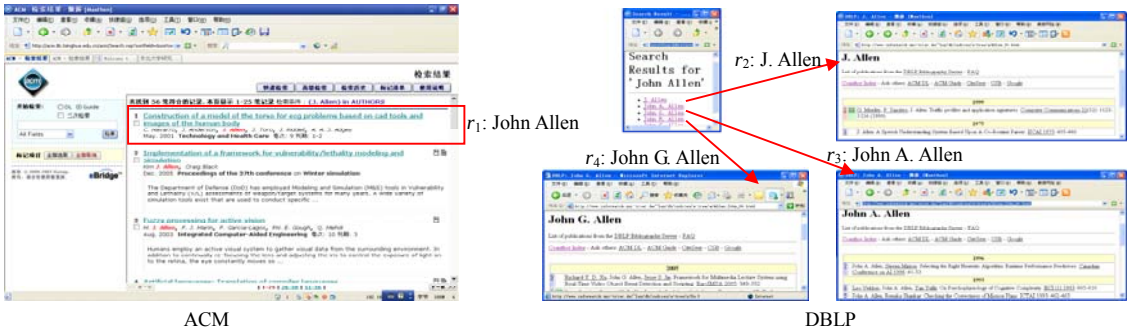

Fig.1 Demonstration of data inconsistency existing in a single data source or among multiple data sources

图 1 存在于一个数据源内部或多数据源之间的数据不一致性示例

然而,目前针对Web数据实体识别的研究与开发还处于起步阶段,大多数工作都是围绕静态环境来构建实体识别系统(如数据挖掘领域中数据清理问题的研究[2]),与这些实体识别系统相比,Deep Web实体识别的研究更具挑战性,主要体现在以下几个方面:当前大多数实体识别系统仅单一地考虑文本属性特征或上下文语义信息来衡量数据间的相似性,没有对聚类结果进行逐步求精处理,识别的准确性难以保证,因此,已有的识别策略有待改进;蕴含在Deep Web中的数据资源具有较强的动态性,如果按照传统的数据挖掘等方法将获取的所有表象关联信息保存在数据仓库中,则具有相对静态特性的数据仓库的内容不能实时地反映数据的动态变化,因此,传统的数据清理方法对于动态性强的数据资源并不适用;由于Deep Web数据量巨大,若对所有数据源进行分析识别,势必影响识别的处理效率,相反,若只分析部分信息,表象关联知识的完备性将会受到影响,因此,样本数据量的大小难以确定.

综上所述,要想有效地整合 Deep Web 上的数据就必须解决以上问题.本文将着重解决的问题是:在表象模式已知的前提下,如何基于实例信息来分析并获取表象关联知识,如何确保这些知识具有较高的完备性与准确性,以及如何保证表象关联知识库中的内容能够适应于 Deep Web 环境中数据的动态性.

1 相关工作

目前,关于实体识别技术的研究主要集中在两方面:一方面是基于实体的属性文本特征进行实体识别(feature-based similarity,简称FBS方法),侧重于研究文本相似函数的设置(包括函数定义、选取和相关阈值的确

定)[3−7]、属性权重的选取[8−10]以及相关优化措施[11]等;另一方面是基于实体上下文语义信息或特定领域知识,利用数据挖掘等方法进行实体识别,侧重于研究语义关联的表示方式及计算方法[12−17].

其中,基于属性特征进行实体识别的相关工作包括:Bilenko[3]通过采用机器学习方法SVM自适应地选取最佳文本相似度算法,以满足不同领域数据的特点及需求;Cohen[4]分析并比较了多种字符串匹配函数(如编辑距离、Token距离、Levenstein距离等),并从中选取最佳函数来衡量属性特征的相似性;Zhu等人[5]根据关系表的决定属性值划分记录集,并在每个记录集内应用动态优先队列聚类算法和合并逆序算法来检测数据库中的相似重复记录;Ling等人[6]将Web页面中的数据划分成记录块,基于记录块间的文本相似性来判断不同数据源上的记录是否重复,并通过迭代训练来确定相关参数的阈值.以上工作侧重于研究相似函数的定义、选取及阈值设置,除此之外,文本匹配中属性权重的选取及相关优化措施的采用也越发引起关注,例如,Wang等人[8]利用梯度递减算法为数据的描述属性赋予权值,并基于属性权重学习算法和聚类算法将相似的数据聚类;Chaudhuri[11]提出一种高效的重复记录模糊检测算法,通过采用特定索引、排序等优化措施有效地搜索出与当前元组最相似的K个关联元组.

基于上下文语义信息或特定领域知识进行实体识别的相关工作包括:Chen[12]将表象间的语义关联以图形化表示,并应用图分割技术对表象集进行聚类分组,每组由对应于同一实体的表象集组成;Thor[13]建立了一个灵活的表象匹配框架MOMA,通过该框架采用不同的表象匹配算法获取表象关联集,并将其合成,从而计算表象间的关联强度;Nie等人[14]利用上下文在互联网共同出现的情况,通过计算网页的URL距离来计算不同表象的互联网关联强度.

以上工作大多数是基于静态环境构建的实体识别系统,不能较好地适应 Deep Web 环境下数据资源的动态性并保证表象关联知识的完备性;另外,这些实体识别系统或者基于属性文本特征,或者基于上下文信息来识别数据间的相似性,缺乏对聚类结果进行逐步求精的过程,因此,单一地应用实体属性特征或上下文信息来进行实体识别将会影响识别结果的准确性.为此,本文探讨了一种基于语义及统计分析的 Deep Web 实体识别机制,主要贡献在于:提出了 SS-EIM(deep Web entity identification mechanism based on semantics and statistical analysis)的模型,其中包括文本匹配模型、语义分析模型和分组统计模型,能够有效解决 Deep Web 数据集成中数据纠错、消重及整合等问题;提出了文本粗略匹配、表象关联关系获取以及分组统计分析的三段式逐步求精策略,有效地提高了实体识别的准确性;基于有限的实例信息,采用静态分析、动态协调相结合的自适应知识维护策略,构建和完善表象关联知识库,能够适应 Web 数据的动态性并保证表象关联知识的完备性;通过实验验证了 SS-EIM 中所采用的关键技术的可行性和有效性,与其他实体识别策略相比,SS-EIM 在知识准确性、知识完备性等性能上具有一定的优势.

本文第 2 节介绍 SS-EIM 的模型;第 3 节介绍文本匹配模型;第 4 节提出基于上下文的语义分析模型;第 5 节介绍基于约束规则的分组统计模型;第 6 节给出三段式逐步求精算法以及性能分析;第 7 节讨论表象关联知识维护策略;第 8 节给出相关实验结果并进行分析;第 9 节总结全文.

2 SS-EIM 的模型

通过预获取的有限的实例信息可以分析和推导出实例间的关联关系并构建表象关联知识库,利用表象关联知识可以有效地指导实时查询中资源选择、消重及数据整合等操作.本节首先介绍 SS-EIM 的模型,然后针对模型中涉及的相关概念给出定义.

SS-EIM的模型如图 2 所示.其中,R是有限数据集D中的所有表象r_i构成的集合,记为$\{r_1,r_2,\ldots,r_{|R|}\}$.将$D$中与现实世界一一对应的实体集记为$E=\{e_1,e_2,\ldots,e_{|E|}\}$,$e_i$是由对应于同一实体的表象所组成的表象集.实体识别的任务就是给定一个表象集R,通过匹配、推理及分析等过程,将对应于同一实体的表象进行聚类,最终得到实体集E,$|R|\geq|E|$.实际上,E是理想情况下的实体识别结果,由于Web表现形式的多样性以及数据的复杂性很难将R准确无误地转化成E.因此,可以利用SS-EIM生成的聚类集$C=\{c_1,c_2,\ldots,c_{|C|}\}$来近似地表示实体集$E$,$|C|\approx|E|$.SS-EIM模型首先将模式已知的表象集$R$作为待识别数据,基于文本匹配模型、语义分析模型和分组统计模型对这些实例

信息进行静态分析,分别生成一系列文本相似集、语义相似集和满足约束集,并形成初始的表象关联知识;然后,基于知识维护模型以及用户的实际查询结果对当前表象关联知识库中的内容进行动态协调,使其不断被扩充和完善,以保证表象关联知识的完备性和有效性.

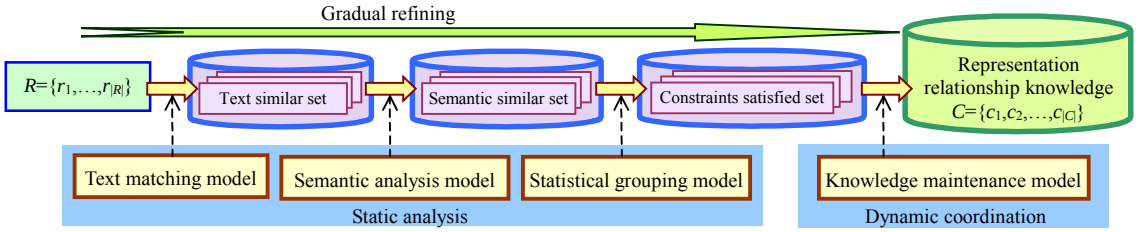

Fig.2 SS-EIM model

图 2 SS-EIM 模型

文本匹配模型通过比较属性取值的文本相似度来检测由数据格式、拼写错误等因素造成的数据不一致性.但是,文本匹配模型只是通过字面粗略地对表象进行聚类,而文本特征相似的表象很可能代表不同的实体,因此语义分析模型通过上下文语义信息来进一步提高聚类的准确性.除了分析字符串类型的属性及其语义关联以外,分组统计模型还用来对数值型属性进行统计分析以提高实体识别的准确性.为了使表象关联知识库的内容适应 Web 数据的动态性,知识维护模型通过收集实际查询过程中的动态结果数据,以此作为基准来实时检测表象关联知识库的完备性和有效性,从而对表象关联知识库中的内容不断进行调整并完善.

从SS-EIM模型中可以看出,文本相似集、语义相似集以及满足约束集的生成过程是对表象聚类集逐步求精的过程,将其分别记为S_{txt},S_{sem}和S_{con},三者的关系为$S_{txt}[r_i]\supseteq S_{sem}[r_i]\supseteq S_{con}[r_i]$,现给出具体定义.

定义 1(文本相似集$S_{txt}[r_i]$). 基于实体的属性取值、利用文本相似函数Sim计算表象间的相似度,文本相似集由文本相似度达到文本相似阈值σ_{txt}的表象以及与该表象具有异名同义或层次包含关系的表象$S_{txt_Extend}[r_i]$所组成,具体定义为

$$S_{txt}[r_i]=\left\{r_j \mid r_j\in R\wedge\exists r_k(r_k\in S_{txt}[r_i]\wedge Sim(r_j,r_k)>\sigma_{txt})\right\}\cup S_{txt\_Extend}[r_i] \tag{1}$$

定义 2(语义相似集$S_{sem}[r_i]$). 依据语义关联进一步分析每个文本相似集,语义相似集由表象间关联强度CS大于语义相似阈值σ_{sem}的表象所组成,具体定义为

$$S_{sem}[r_i]=\left\{r_j \mid r_j\in S_{txt}[r_i]\wedge\exists r_k(r_k\in S_{sem}[r_i]\wedge CS(r_j\rightarrow r_k)>\sigma_{sem})\right\} \tag{2}$$

定义 3(满足约束集$S_{con}[r_i]$). 基于约束规则检测语义相似集的聚类准确性,满足约束集由语义相似集内满足约束规则的表象所组成,具体定义为

$$S_{con}[r_i]=\left\{r_j \mid r_j\in S_{sem}[r_i]\wedge S_{con}[r_i]\text{ 满足约束规则}\right\} \tag{3}$$

其中,语义关联是指存在于表象间直接或间接的上下文联系,关联强度是对语义关联强弱的量化表示.以图 1 为例,若r_1与r_2有相同的合作者,则认为表象r_1与r_2之间存在语义关联.约束规则是指针对某领域中表象的属性聚集值所定义的约束条件.例如,若图 1 中ACM网站的r_1与DBLP网站的r_2,r_3对应同一个人,那么该作者在两个网站中被收录的文章总数应相差不多,可以将其作为约束规则来进一步检测聚类的准确性.

3 文本匹配模型

在特定的领域范围内,对应于相同实体的表象往往具有相似的属性特征.因此,本文应用文本匹配模型对实例信息进行文本粗略匹配,借鉴文本编辑距离函数[18]来衡量表象间属性级相似度,并采用多属性合成函数将多个属性级相似度进行合并,进而衡量表象级的相似程度,最终将具有相似文本特征的表象聚成一类,形成一系列文本相似集.具体步骤如下:

步骤 1. 借鉴已有的属性选择算法[8−10],利用近似函数依赖关系来量化实体描述属性a_1~a_n的重要度,并赋

予不同的权值w_1~w_n,具体定义如公式(4)所示.其中,$\hat{a}$表示a_i与其他属性的组合属性;$error(\hat{a}\rightarrow a_j)$表示为了使近似函数依赖$\hat{a}\rightarrow a_j$成立,需要从关系中移除的记录占整个关系的最小比例.$error(\hat{a}\rightarrow a_j)$越小,说明关系中满足该函数依赖的记录就越多,$a_i$的影响力就越大,因此被赋予较高的权值.

$$w_i=\sum_{j=1}^{n}\frac{1-error(\hat{a}\rightarrow a_j)}{|\hat{a}|} \tag{4}$$

步骤 2. 针对每个表象对(r_i与r_j) 中的各个属性a_k(k=1~n),分别基于文本编辑距离函数计算表象间属性级的相似度$Sim(r_i.a_k,r_j.a_k)$,具体定义如下:

$$Sim(r_i.a_k,r_j.a_k)=1-\frac{ed(r_i.a_k,r_j.a_k)}{\max\{|r_i.a_k|,|r_j.a_k|\}} \tag{5}$$

步骤 3. 基于多属性合成函数及属性权重将多个属性级的相似度进行合并,从而计算出表象级的文本相似度$Sim(r_i,r_j)$(如公式(6)所示).最终将满足文本相似阈值的所有聚类进行合并,生成一系列初始文本相似集.

$$\begin{aligned}Sim(r_i,r_j)&=Com(Sim(r_i.a_1,r_j.a_1),...,Sim(r_i.a_k,r_j.a_k),...,Sim(r_i.a_n,r_j.a_n))\\&=\sum_{k=1}^{n}w_k\times Sim(r_i.a_k,r_j.a_k)\end{aligned} \tag{6}$$

步骤 4. 针对每个初始文本相似集,基于辅助信息库,如Wordnet,将与其存在异名同义及层次包含关系的表象$S_{txt_Extend}[r_i]$扩充到其中,以保证最终生成的文本相似集具有较高的完备性和准确性.

由此可见,文本粗略匹配得到的是从属性特征上具有相似性的表象聚类集,然而,表象的外在表现形式不足以作为聚类的判定依据.例如,J. Smith既可以表示John Smith,又可以表示Jane Smith,虽然基于文本匹配,这些表象可能被聚集在同一个文本相似集内,但它们分别对应不同的实体.因此,文本匹配只是基于属性特征对表象集粗略地划分,其准确性难以保证.但是,利用文本匹配模型可以使后续的操作在较小的数据空间上进行,从而降低了实体识别的执行代价.

4 语义分析模型

语义分析模型用来对每个文本相似集的表象进行语义关联分析,将具有语义相似性的表象进行聚类,从而在属性文本匹配的基础上提高了实体识别的准确性.首先,基于表象的上下文信息将其语义关联用表象关联图来表示;然后,针对表象关联图中表象间的多条路径进行分析,并从中选取进行关联强度运算的最佳路径;最后,计算表象间的关联强度,将关联强度大于语义相似阈值的表象聚成一类而形成语义相似集.

4.1 语义关联规则

针对某一领域,对应于同一实体的表象之间往往存在着直接或间接的语义关联,利用这些关联信息可以提高实体识别的准确性.因此,需要事先挖掘出该领域表象间的语义关联规则.

设待聚类表象集为R,本文首先采取人工筛选的方式从R中确定一系列标准聚类集c,其中每个c由对应同一实体的表象组成;然后,将R与该领域内其他表象集R_k分别组合形成一系列候选关联R_R_k,借鉴数据挖掘中的Apriori算法[19,20],针对每一种候选关联R_R_k,挖掘所有标准聚类集中的频繁 2 项集$\{r_i,r_j\}$并计算其支持度$Support(r_i\Rightarrow r_j)$,$R_R_k$的支持度由这些频繁 2 项集的支持度聚集而成(如公式(7)所示);最后,选取支持度较高的候选关联以及它们的有限次迭加作为实体识别的语义关联规则.

$$Support(R\_R_k)=\sum_{r_i,r_j\in R.c\wedge\exists r_k(r_k\in R_k\wedge r_i\Rightarrow r_k\Rightarrow r_j)\wedge\{r_i,r_j\}是频繁2项集}Support(r_i\Rightarrow r_j) \tag{7}$$

以论文检索领域为例,若待聚类的表象集是作者表象集合,则该领域内的关联组合包括作者_合作者、作者_会议、作者_出版日期等候选关联.若存在某合作者表象r_k,同时与标准聚类集中的两个作者表象r_i,r_j具有合作关系$r_i\Rightarrow r_k\Rightarrow r_j$,且该现象频繁发生,则作者_合作者关联的支持度较高,因而将该关联作为实体识别的语义关联规则.以此类推,针对该领域的特征,最终确定如下语义关联规则作为构建表象关联图的依据.

- 合作者关联(作者_合作者):若两个表象的合作者集合存在交集,则认为该表象间存在合作者关联.

- 出处关联(作者_会议):若两个表象出自同一个会议或出版机构,则认为该表象间存在出处关联.
- 混合关联:表象间经过合作者关联或出处关联的有限次迭加而建立的关联.

4.2 表象关联图

基于这些预先挖掘的语义关联规则,可以检测出表象间的关联关系,将这些信息以图形化表示出来的过程就是构建表象关联图(representation relationship graph,简称 RRG)的过程.RRG 由节点集合 V 和边集合 E 组成,V 用来表示实体的不同表象,包括文本相似集内的表象以及与之具有语义关联的所有表象,如上例中的作者和会议等;E 用来表示表象间存在的语义关联,如上例中的合作者关联、出处关联等.表象关联图类似于关系数据库中的实体联系模型,不同的是,表象关联图表示的是实例(表象)间的关联信息,因此,可以将表象关联图看成是对实体联系模型的实例化表示.

针对论文检索领域某一文本相似集内的两个表象 John Allen 与 J. Allen 构建的表象关联图如图 3 所示,其中,节点分为两类,分别表示作者表象和会议表象;同样,按照表象间关联类型的不同,边也分为合作者关联与出处关联.具体来说,图 3(a)表示 John Allen 的合作者 Helen 同样也是 J. Allen 的合作者,则认为 John Allen 与 J. Allen 存在合作者关联;图 3(b)表示 John Allen 发表的某篇文章与 J. Allen 的来源于同一会议 VLDB,则认为 John Allen 与 J. Allen 存在出处关联;图 3(c)表示 John Allen 与 J. Allen 之间既存在合作者关联又存在出处关联,则认为 John Allen 与 J. Allen 之间存在混合关联.

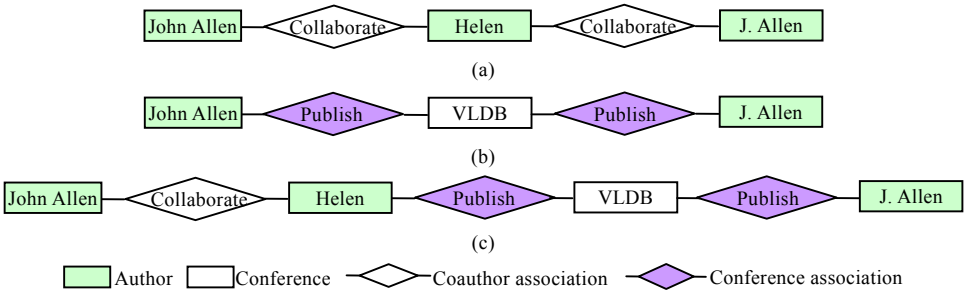

Fig.3 Demonstration of representation relationship graph

图 3 表象关联图示例

4.3 最佳路径选取

表象关联图中两个表象之间如果存在多条关联路径,如何从中选取一条最佳路径作为表象间关联强度计算的依据就成为一个有待解决的问题.通常,如果两个表象间语义关联所在的上下文环境包含的表象越少,语义关联就越有针对性,表象同属于一个实体的概率也就相对越高.为此,本文定义了路径重要度来量化表象间语义关联的独特性,并选取重要度高的路径作为最佳路径来计算关联强度.

定义 4(路径重要度). 若两个表象间某关联路径上的所有表象节点(不包含起始表象和终止表象)为$\{r_1,\ldots,r_k\}$,与节点r_i存在语义关联的表象个数为m_i,则该关联路径的路径重要度为 $\sum_{i=1}^{k}\frac{1}{m_i}$.

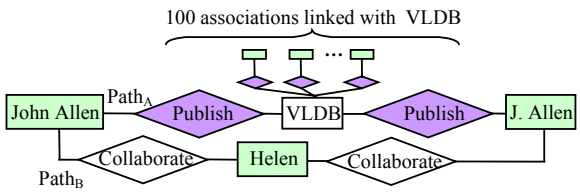

Fig.4 Selection of the best path

图 4 最佳路径的选择

如图 4 所示,表象 John Allen 与 J. Allen 之间路径 A 的重要度为 0.01,而路径 B 的重要度为 0.5.这是由于 VLDB 与很多作者之间都存在出处关联,表象覆盖率较高,因此削弱了该关联针对作者 John Allen 与 J. Allen 的表现能力;相反,Helen 只与 John Allen 和 J. Allen 具有合作者关联,该关联的针对性较强,突出表现了 John Allen 与 J. Allen 之间语义关联的独特性.因此,在本例中,优先选择表现能力较强的路径 B 作为最佳路径来计算表象 John Allen 与 J. Allen 之间的

关联强度.

4.4 关联强度的计算

关联强度是基于最佳路径对表象间语义关联的量化表示,能够在一定程度上反映出表象间潜在的语义相似性.随着关联类型的不同,其语义关联程度也有所不同,因此,需要针对不同的关联类型为其赋予不同的权值.由此可见,前文中最佳路径的选取是根据表象关联图中关联的独特性来进行的,而关联强度则是基于表象间关联的语义表达能力来进行计算的.表象r_1到r_n间关联路径的关联强度如公式(8)所示.

$$CS(r_1 \to r_2 \to ... \to r_n) = \frac{1-CS(r_1 \to r_2)+...+1-CS(r_{n-1} \to r_n)}{n-1} = 1-\frac{n_1 w_1 + ... + n_m w_m}{n-1} \tag{8}$$

$$w_i = 1-\frac{Support(R\_R_i)}{\sum_{k=1}^{m} Support(R\_R_k)} \tag{9}$$

可见,表象间的关联强度相当于关联路径上每条子路径$r_{i-1} \to r_i$关联强度的平均值.其中,n_1~n_m表示关联路径上属于各种关联类型的关联数目;w_1~w_m表示按照语义关联程度对不同的关联类型(设有m个)所赋予的权值(如公式(9)所示).需要注意的是,语义关联的表达能力越强,被赋予的权值就越小,关联强度也就越大.例如,在论文检索领域中,合作者关联比出处关联具有更强的语义表现能力,经计算,其权值分别为 0.3 和 0.7,则图 3(a)、图 3(b)和图 3(c)中表象John Allen与J. Allen间关联强度分别为 0.7,0.3 和 0.43.

5 分组统计模型

文本匹配与语义分析针对的都是字符串类型的属性,实际应用中实体的某些特征也经常用数值型属性加以描述,如报价、库存量等.对应于同一实体的不同表象在这些数值型属性上往往具有相似的聚集值,例如在不同数据源中,同一商品的平均价格基本一致、同一作者被检索的文章总数基本相同等.因此,可以基于这些数值型属性对语义相似集的表象进行统计分析,并按照属性聚集值的相似程度进一步对表象进行聚类.

5.1 约束规则

我们将存在于不同数据源中的同一聚类分组应该满足的统计规律定义为约束规则.约束规则所约束的目标是不同数据源内的聚类分组,而不同于文本匹配与语义分析以表象作为操作对象.因此,需要在聚类分组的层次上、针对不同的数据源对其进行统计分析.需要强调的是,只有当聚类分组所包含表象的个数大于 1 时,才有必要进行统计分析,因此,通常认为只包含 1 个表象的聚类分组均满足约束规则;另外,只有当数据源规模较大时,统计分析才能有效进行,因此,分组统计分析适合在表象覆盖率较大的数据源上进行.

以论文检索领域为例,假设语义相似集S_{sem}{John Allen,J. Allen,J. A. Allen,J. B. Allen,Jane Allen}在数据源DS_1和DS_2中的相关统计信息见表 1,针对作者的被收录文章数、参与会议数可以为每个聚类分组(包含的表象数大于 1)定义如下约束规则:

- 文章数目约束:对应于同一实体的表象集{r_1,…,r_n}在DS_1中被收录的文章总数应等于在DS_2中被收录的文章总数,记为$DS_1.Sum(r_1$.文章数,…,r_n.文章数)=$DS_2.Sum(r_1$.文章数,…,r_n.文章数).
- 会议次数约束:对应于同一实体的表象集{r_1,…,r_n}在DS_1中参与某会议的总次数应等于在DS_2中参与该会议的总次数,记为$DS_1.Sum(r_1$.会议数,…,r_n.会议数)=$DS_2.Sum(r_1$.会议数,…,r_n.会议数).

Table 1 Statistical information of a semantic similar set in different data sources

表 1 不同数据源中某语义相似集的统计信息

S_{sem}	Number of accepted papers		Number of attended conferences	
	DS_1	DS_2	DS_1	DS_2
John Allen	20	35	6	4
J. Allen	30	40	1	2
J. A. Allen	40	15	1	2
J. B. Allen	15	10	1	5
Jane Allen	120	50	3	11

表 1 中,S_{sem}在DS_1与DS_2中被收录的文章总数分别为 225 和 150,参与的会议总次数分别为 12 和 24,说明S_{sem}包含属于多个实体的表象而不满足约束规则,需要对其重新分组,直到各个分组均满足约束规则为止.然而,由于Web数据的复杂性,聚类分组很难完全满足预定义的约束规则,为了保证统计分析的有效性,通过设置分组阈值$\sigma_G(0\le\sigma_G\le1)$适当地放宽约束限制.例如,上例中可以将文章数目约束定义为$\mathrm{Min}\{DS_1.Sum(r_1.\text{文章数},\ldots,r_n.\text{文章数}),DS_2.Sum(r_1.\text{文章数},\ldots,r_n.\text{文章数})\}/\mathrm{Max}\{DS_1.Sum(r_1.\text{文章数},\ldots,r_n.\text{文章数}),DS_2.Sum(r_1.\text{文章数},\ldots,r_n.\text{文章数})\}\ge\sigma_G$.

5.2 基于统计分析树的表象重组

对于不满足约束规则的聚类分组,需要在语义相似集内部对其重新划分.为此,本文基于语义分析阶段生成的表象关联图构建统计分析树,并进行表象重组,最终生成满足约束规则的聚类分组.其过程如下:

步骤 1. 针对某语义相似集(包含表象个数>1)截取表象关联图 RRG 中与之相关的片断(若表象间存在多条路径,只保留最佳路径),记为 RRG′.

步骤 2. 在 RRG′中只保留该语义相似集的表象,并将表象间的关联用对应的关联强度代替,记为 RRG″.

步骤 3. 将 RRG″中所有表象组成的集合作为统计分析树的根节点,按照表象间关联强度由小到大的顺序将 RRG′划分成不连通的两部分,分别作为统计分析树的叶子节点.

步骤 4. 检测当前统计分析树中的叶子节点是否均满足约束规则,若满足,则结束操作;否则,对不满足规则的叶子节点按步骤 3 继续划分,直到所有叶子节点均满足约束规则为止.

以表 1 中的表象为例,其表象重组的过程如图 5 所示.首先,针对语义相似集{John Allen,J. Allen,J. A. Allen,J. B. Allen,Jane Allen}截取表象关联图中的相关片断 RRG′,由于图中只保留最佳路径,因此表象间不存在回路;然后,对 RRG′进行简化处理,只保留语义相似集的表象并标明关联强度;将语义相似集作为根节点构建统计分析树,优先选择 RRG″中关联强度较小的边断开,将得到的不连通的两部分作为叶子节点;由于存在叶子节点{John Allen,J. Allen,J. A. Allen,J. B. Allen}不满足约束规则,需要对其继续划分;以此类推,最终得到 3 个满足约束集:{John Allen,J. Allen,J. A. Allen},{J. B. Allen}和{Jane Allen}.

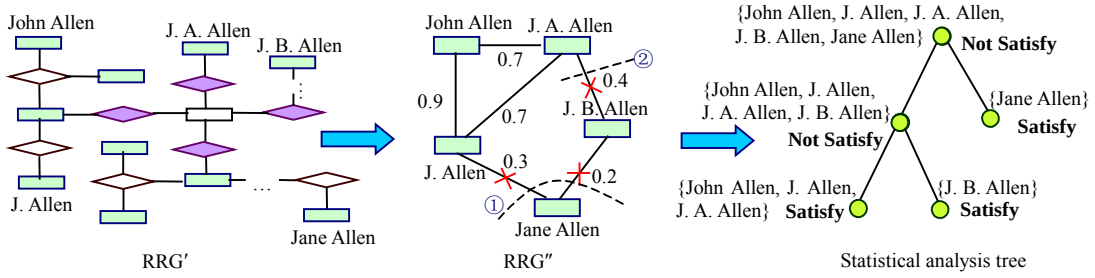

Fig.5 Transition from representation relationship graph to statistics analysis tree

图 5 从表象关联图到统计分析树的转换

6 三段式逐步求精算法及性能分析

本文基于文本匹配模型、语义分析模型以及分组统计模型,采用文本粗略匹配、表象关联关系获取以及分组统计分析的三段式逐步求精策略,结合属性文本特征、上下文语义信息及约束规则对识别结果不断进行精化,以提高识别结果的准确性.本节将介绍 SS-EIM 的三段式逐步求精算法,并对其性能进行分析.

6.1 SS-EIM的三段式逐步求精算法

基于前文定义的文本匹配模型、语义分析模型以及分组统计模型,SS-EIM 的三段式逐步求精(three-phase gradual refining,简称 TGR)算法具体描述如下.

阶段 1. 文本粗略匹配阶段.应用文本匹配模型分别计算针对属性级和表象级的文本相似度,并将表象按照

属性特征进行聚类,生成一系列文本相似集.

阶段 2. 表象关联关系获取阶段.基于语义分析模型分析文本相似集内表象所处上下文环境的语义信息,将达到一定关联强度的表象聚成一类,生成一系列语义相似集.

阶段 3. 分组统计分析阶段.基于分组统计模型进一步检测语义相似集是否满足预先定义的约束规则,利用统计分析树将不满足约束规则的表象集进行表象重组,最终生成一系列满足约束集(每个集合对应于一个实体的表象聚类集),以达到实体识别的目的,并将其作为表象关联知识存储.

由上可知,TGR 算法是对实体的表象聚类集逐步求精的过程,其目的是为了保证表象关联知识的准确性.本文重点介绍表象关联关系获取(见算法 2)和分组统计分析(见算法 3)的过程,具体识别算法如下:

算法 1. TGR 算法.

输入:$R=\{r_1,r_2,\ldots,r_{|R|}\}$,文本相似阈值$\sigma_{txt}$,语义相似阈值$\sigma_{sem}$,分组统计阈值$\sigma_G$,RRG最大搜索深度$L$.

输出:实体聚类集$C=\{c_1,c_2,\ldots,c_{|C|}\}$.

步骤:

```
1:    Vector txtSets:=R.initialTxtMatching(σ_txt);  //对R进行文本粗略匹配并将结果存入向量txtSets
2:    Vector C:=∅;  //为实体聚类集设置初始值
3:    for(int i=0;i<txtSets.size();i++)  //顺序访问 txtSets 中的每个文本相似集
4:        Vector txtSet:= txtSets.get(i);  //从向量 txtSets 中取一个文本相似集存入 txtSet
5:        Vector semSets:=getSemSets(txtSet,σ_sem,L);  //生成一系列语义相似集并存入向量semSets
6:        for(int j=0;j<semSets.size();j++)  //顺序访问 semSets 中的每个语义相似集
7:            Vector semSet:=semSets.get(j);  //从向量 semSets 中取一个语义相似集存入 semSet
8:            Vector conSets:=getConSets(semSets,σ_G);  //生成一系列满足约束集存入向量conSets
9:            C=C∪conSets;  //将当前生成的 conSets 添加到实体聚类集 C 中
10:   return C;  //将最终的实体聚类集 C 作为结果返回
```

算法 2. 语义相似集生成算法*getSemSets*(Vector *txtSet*,float σ_{sem},int L).

输入:文本相似集*txtSet*,语义相似阈值σ_{sem},RRG最大搜索深度L.

输出:一系列语义相似集 *semSets*.

步骤:

```
1:    Vector semSets:=∅;  //用来存储生成的一系列语义相似集
2:    for(int i=0;i<txtSet.size();i++)  //顺序访问文本相似集 txtSet 中的每个表象
3:        String rep:= txtSet.get(i);  //从 txtSet 中取一个表象 rep
4:        if(rep.semGroupId==0)  //如果 rep 没有被聚类到任何一个语义相似集内,则对其进行语义分析
5:            Vector oldV:={rep};  //将 rep 存入向量 oldV,作为当前的语义相似集
6:            Vector newV:=∅;  //用于与当前的语义相似集进行比较,判断其是否需要继续扩充
7:            for(int j=0;j<L;j++)  //以深度 L 遍历表象关联图
8:                newV:=oldV∪{repNew|CS(repOld→repNew)>σ_sem and repOld∈oldV};  //将所有与oldV中的表象具有一定语义关联的表象repNew扩充到当前语义相似集内
9:                if(newV==oldV)  //判断是否满足结束条件,也就是不再有新的表象被扩充到 oldV 中
10:                   oldV.setSemGroupId();  //将 oldV 作为一个语义相似集并为其设置组号标识
11:                   break;  //跳出当前循环,继续生成下一个语义相似集
12:               oldV:=newV;  //将 oldV 赋值为 newV,并对其继续扩充,直到满足结束条件
13:           semSets.add(oldV);  //将当前生成的语义相似集添加到结果集 semSets 中
14:   return semSets;  //将最终的 semSets 作为结果返回
```

算法 3. 满足约束集递归生成算法*getConSets*(Vector *semSet*,float σ_G).

输入:语义相似集*semSet*,分组统计阈值σ_G.

输出:一系列满足约束集 *conSets*.

步骤:

1: Vector *conSets*:=∅; //用来存储生成的一系列满足约束集

2: if(*semSet*.*satisfyConstraints*(σ_G)) //判断当前集合*semSet*是否满足预定义的约束规则

3: return *semSet*; //如果满足,则无须对 *semSet* 进行表象重组,直接作为满足约束集返回

4: else Vector *tempV*:=*Split*(*semSet*); //否则,将 *semSet* 划分为不相交的两部分,并存入向量 *tempV* 中

5: for(int *i*=0;*i*<2;*i*++) //顺序访问 *tempV* 中的每个表象集合(共两个)

6: Vector *tempSet*:=*tempV*.*get*(*i*); //从 *tempV* 中取一个表象集合存入 *tempSet* 中

7: *conSets*=*conSets*∪*getConSets*(*tempSet*); //针对 *tempSet* 递归调用 getConSets 方法

8: return *conSets*; //将最终的 *conSets* 作为结果返回

6.2 算法性能分析

SS-EIM 的识别时间代价可以按照处理阶段的不同划分为 3 个部分,都是将上一阶段的结果作为下一阶段的处理对象.SS-EIM 的实体识别的总体时间代价为

$$Time_{SS-EIM}=Time_{txtInitialMatching}+Time_{relAbstraction}+Time_{statAnalyse} \tag{10}$$

其中,$Time_{txtInitialMatching}$为文本粗略匹配的时间,$Time_{relAbstraction}$为表象关联关系获取所需要的时间,$Time_{statAnalyse}$为分组统计分析的时间.假设实体识别过程中产生的文本相似集、语义分析集、满足约束集的数目分别为$N_{txtSets}$,$N_{semSets}$和$N_{conSets}$,且平均每个文本相似集含有m个表象,RRG最大搜索深度为L,则基于上述算法,我们给出表象关联关系获取和分组统计分析的时间复杂度.其中,$Time_{relAbstraction}$的时间复杂度为$O(N_{txtSets}\times m\times L\times(|newV|\times\log|newV|+|oldV|\times\log|oldV|))$,由于$newV$及$oldV$包含元素的个数最大为$m$,所以$Time_{relAbstraction}$的上限为$O(N_{txtSets}\times L\times m^2\log m)$,$Time_{statAnalyse}$的时间复杂度为$O(N_{txtSets}\times N_{semSets}\times N_{conSets})$.

为了提高SS-EIM的处理效率,我们对以上算法采取了如下优化措施:首先,为了防止某些特殊情况下遍历RRG的时间过长,通过预先设置RRG最大搜索深度L,使得表象关联关系获取能够在相对有限的范围内进行,以降低$Time_{relAbstraction}$的时间复杂度;其次,由于表象关联关系获取和分组统计分析只集中在各个文本相似集内部进行,因此本文采用多线程并发处理机制,将针对不同文本相似集的语义分析及分组统计处理并行进行,从而进一步降低了$Time_{relAbstraction}$和$Time_{statAnalyse}$的时间复杂度;同时,SS-EIM将关联路径上表象间关联强度运算的中间结果存储在缓存中,从而避免了无谓的重复性计算,提高了实体识别的处理效率;另外,SS-EIM的三段式逐步求精过程为离线型服务操作,不会对用户的操作造成直接影响.

7 自适应的知识维护策略

如果抽取 Deep Web 中所有数据源上的数据进行实体识别,显然可以保证表象关联知识的完备性,但由此产生的巨额执行代价使得这种方法不可行;同时,由于 Web 环境中数据的动态特性,存储在表象关联知识库中的信息不是一成不变的,需要对其有效性及时检测.为此,本文采用静态分析、动态协调相结合的自适应知识维护策略(adaptive knowledge maintenance,简称 AKM)构建和完善表象关联知识库,以适应 Web 数据的动态性并保证表象关联知识的完备性.具体定义如下:

- 静态分析:将 Web 中一定量的静态页面中的信息作为预获取的实例信息,并进行模式抽取;基于这些实例信息进行静态分析,也就是采用前文提到的文本粗略匹配、表象关联关系获取以及分组统计分析的三段式逐步求精策略,生成初始的表象关联知识库.
- 动态协调:随着用户查询的增多,基于用户查询请求和返回结果的实例信息来动态协调表象关联知识库中的内容,对其进行不断修正和完善.

理想情况下,某个实体的所有表象应该同属于一个聚类,但实际的识别结果往往不能达到绝对的准确,本属

于同一实体的表象有可能被识别到不同的聚类中.因此,本文针对特定领域中的某类查询请求,以最终查询结果中的实例信息作为基准,引入实体熵的概念来衡量某个实体的表象集分散到多个聚类中的分散程度,从而检测表象关联知识库的有效性.假设某查询结果集中的实例信息为$e=\{r_1,r_2,\ldots,r_n\}$,表明表象r_1~r_n对应于某实体e,表象关联知识库中对应于该类查询请求的内容为$C=\{c_1,c_2,\ldots,c_m\}$,每个聚类集c_i都由一组语义上相似且满足约束规则的表象组成,则实体熵$H(e)$的定义如公式(11)所示.其中,$C'=\{c_1,\ldots,c_{m'}\}$为C中与e存在交集的聚类集合, $|c_i \cap e|$表示e中的表象被聚类到c_i中的个数,因此可以将$|c_i \cap e|/n$理解为聚类集c_i对e的覆盖率.理想情况下,e中所有表象均被聚类到C中的某个聚类集c_i中,此时$H(e)$为 0;相反,若e中表象被分散地聚类到C中的多个聚类集中,则随着分散程度的增大,$H(e)$将趋近于$\log_2 n$.因此,可以将用户的实际查询结果以及实体熵作为表象关联知识库的信息是否完备和有效的衡量标准,具体检测过程如下:

$$H(e)=\sum_{i=1}^{m'}\frac{|c_i \cap e|}{n}\log_2^{n/|c_i \cap e|} \tag{11}$$

- 完备性检测:若C'为空,则说明表象关联知识库的信息不够完备(说明目前C中至少不包含实体e的信息),因此将e中的表象集合作为c_{m+1}扩展到C中,并存入表象关联知识库;若$|C'|=1$,则说明C中只有 1 个聚类集(假设为c_1)与e存在交集,此时将c_1内原有的表象连同e中新出现的表象组合成新的聚类来取 代c_1.
- 有效性检测:若$|C'|>1$,则或者是由于静态分析阶段将本属于同一实体e的表象错误地识别成对应多个实体,或者由于Web数据的动态更新使得当前知识库中的部分内容失效.然而,小范围的表象识别错误或数据失效不会对整体的数据集成产生太大的影响,为了降低纠错的处理代价,本文通过设定阈值σ_H,只针对达到一定程度的识别错误或数据失效进行处理.具体来说,基于结果表象集e以及聚类集C'计算$H(e)$,如果$H(e)<\sigma_H$,则说明目前表象关联知识库的信息比较准确,可以保持不变;否则,需要对C'连同e中的表象重新进行实体识别.

举例说明,假设$C=\{c_1,c_2\}$,$c_1=\{r_1,\ldots,r_{10}\}$,$c_2=\{r_{11},\ldots,r_{20}\}$,$\sigma_H$为 0.8.若$e=\{r_{21},r_{22}\}$,由于$e$与$c_1,c_2$都不存在交集,则将$e$作为一个新的聚类集扩展到$C$中,以保证表象关联知识库的完备性;若$e=\{r_1,r_{21}\}$,则$C$中的$c_1$将被更新为$\{r_1,\ldots,r_{10},r_{21}\}$;若$e=\{r_1,\ldots,r_9,r_{11}\}$,则$H(e)$为 0.47 且小于$\sigma_H$,因此可以忽略$C$中少数表象的识别错误或数据失效;若$e=\{r_1,\ldots,r_5,r_{11},\ldots,r_{15}\}$,则$H(e)$为 1 且大于$\sigma_H$,此时需要对$c_1,c_2$以及$e$中的表象重新进行实体识别,并对表象关联知识库进行更新,以保证其内容的有效性.

8 实验测试

本节针对SS-EIM中提出的部分关键技术进行实验测试,主要包括对比不同数据量、不同初始分组数的实体识别时间代价,分析文本相似阈值σ_{txt}、语义相似阈值σ_{sem}、分组阈值σ_G等参数的设置对实体识别准确性的影响,以及比较不同实体识别策略的性能.

8.1 数据集

本文主要针对论文检索领域中的作者表象进行实体识别,通过获取数据的属性特征、分析数据之间的语义关联以及执行相关统计,对作者的不同表象进行识别并将论文信息按照作者进行聚类,以此来提高集成数据的质量.本文使用的数据集分别来源于论文检索网站 ACM 和 DBLP,通过向它们提交特定的查询请求来获取一定量的静态页面并进行模式抽取,最终将模式已知的 1M 条记录作为实验数据集.为了有效地进行实体识别,提交的查询请求应尽量满足以下几个条件:首先,基于这些请求所得到的查询结果集具有较大的数据量;其次,在这些结果数据中存在较多由同一个人所发表的文章信息,且该作者具有多种不同的表象;最后,查询请求相互独立,以保证结果集之间不存在交集.

按照使用阶段的不同,将数据集划分为两个类别:分析集和协调集.分析集应用于实体识别的静态分析,是文本粗略匹配、表象关联关系获取和分组统计分析的处理对象,用来构建初始的表象关联知识库;协调集用来模拟从用户实际查询中所得到的结果集,以检测表象关联知识库中的内容是否完备、准确,主要应用在实体识别的动态协调过程.从数据集中随机选取 800K 条记录作为分析集,将剩余的数据集作为协调集.

实验环境设置如下:主机采用 Dell 2.4GHz P4,内存容量为 512MB,硬盘容量为 80GB,操作系统为 Win2000.

8.2 评价指标

本文主要以执行时间、识别准确性和知识完备性作为SS-EIM的评价指标.识别准确性体现在两个方面:针对某实体的聚类分散性和针对某聚类的实体多样性.前者是指某个实体的表象集分散到多个聚类中的分散程度,可以用前文定义的实体熵来衡量;后者是指某个聚类c所包含实体类别的多样化程度,为此本文引入聚类熵$H(c)$的概念.假设某聚类表象集为$c=\{r_1,r_2,\ldots,r_n\}$,实体集为$E=\{e_1,e_2,\ldots,e_m\}$,其中每个实体e_i是由该实体的所有表象组成的集合,则聚类熵的定义如公式(12)所示.其中,$E'=\{e_1,\ldots,e_{m'}\}$为E中与c存在交集的实体集合,$|e_i \cap c|$表示聚类c中属于实体e_i的表象的数目.理想情况下,c中的表象均对应于同一个实体e,$H(c)$为 0;相反,若c中包含多个实体类别的表象,随着类别的增多,$H(c)$将趋近于$\log_2^n$.将E中所有实体的实体熵$H(e)$的平均值记为$H(E)$.同样,将C中所有聚类的聚类熵$H(c)$的平均值记为$H(C)$.$H(E)$和$H(C)$越小,表象关联知识的准确性就越高.知识完备性用来衡量表象关联知识库中的内容是否完备,我们将数据集内实际存在的实体类别数目记为m,其中通过SS-EIM正确识别的实体类别数目为n,则知识完备性(recall)为n/m.综上所述,本实验的评价指标主要包括:执行时间、知识准确性(通过平均实体熵$H(E)$和平均聚类熵$H(C)$来衡量)和知识完备性.

$$H(c)=\sum_{i=1}^{m'}\frac{|e_i \cap c|}{n}\log_2^{n/|e_i \cap c|} \tag{12}$$

8.3 执行代价

本实验首先针对不同的数据量来对比实体识别所花费的时间代价,分别从分析集中随机抽取 10K,50K,100K,500K,800K 条记录作为实验数据集;然后,基于这些数据测试实体识别所花费的时间代价,其中将 RRG 最大搜索深度设置为 3,实验结果如图 6 所示.从图中可以看出,随着数据量的增长,实体识别的时间代价不断增大,从具体的时间代价和增长速度来看,本文采用的实体识别策略是可行的.

另外,本实验针对 50K 记录测试不同初始分组数对后续处理所需要的时间代价的影响.这里的初始分组是指文本粗略匹配而产生的文本相似集,可以通过设置不同的文本相似阈值而获取到不同数目的初始分组,测试结果如图 7 所示.从图中可以看出,初始分组数越大,每个分组内的数据量就越小,分组内进行实体识别的时间代价也就越小.又由于本文采用了多线程并发处理方式,因此总体时间代价也随之减少.

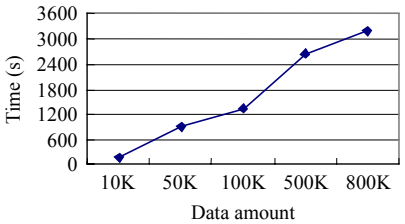

Fig.6 Time cost of entity identification in SS-EIM with different size of data set

图 6 SS-EIM 在不同数据规模下的实体识别时间代价

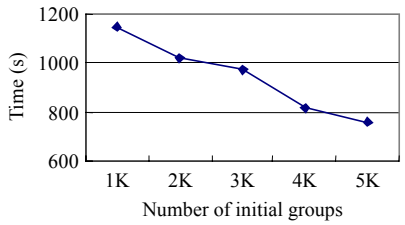

Fig.7 Time cost of entity identification in SS-EIM with different number of initial groups

图 7 SS-EIM 在不同初始分组数目下的实体识别时间代价

8.4 实体识别参数的选取

文本相似阈值σ_{txt}、语义相似阈值σ_{sem}和分组统计阈值σ_G的初始值由领域专家来设置.随着分析数据量的加大,这些参数将逐步趋于最优化配置.本实验通过综合比较$H(E)$和$H(C)$,来确定适当的阈值,实验结果如图 8 所示.随着这些阈值的增大,分组条件越发严格,从而使每个聚类包含的表象数目逐渐减少.这样,每个聚类包含的实体类别将趋向理想的单一化趋势,但同时也加重了对应于同一实体的表象分散化程度.因此,随着这些阈值的增大,$H(C)$不断降低,$H(E)$不断增加.为了综合提高SS-EIM的识别准确性,我们选取$(H(C)+H(E))/2$ 的最小值所对

应的阈值作为参数的最佳阈值.通过测试,σ_{txt},σ_{sem}和σ_G的最优配置分别为 0.6,0.6 和 0.8.

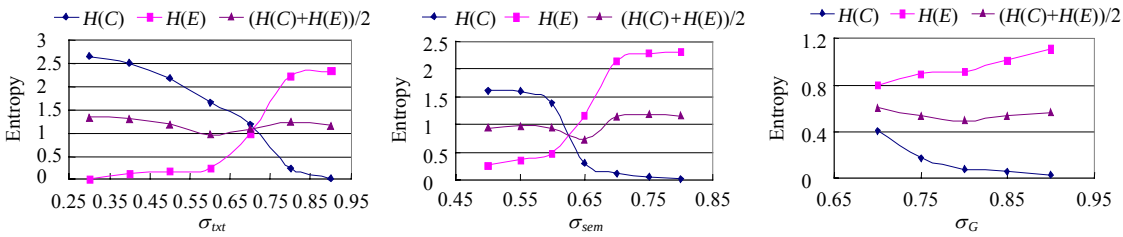

Fig.8 Selection of entity identification parameters

图 8 实体识别参数的选取

8.5 SS-EIM与FBS,Sem-EIM,Stat-EIM的性能比较

本实验将SS-EIM与相关工作中所采取的实体识别策略(基于属性文本特征的实体识别FBS[3−11]、基于语义分析的实体识别Sem-EIM[12−17])以及基于统计分析的实体识别Stat-EIM进行对比,实验结果如图 9 所示.

FBS 是单一地基于属性特征对表象进行聚类;Sem-EIM 是基于 FBS 的结果进一步作语义分析,将具有语义关联的表象聚成一类;Stat-EIM 是对 FBS 的结果基于约束规则进行统计分析.聚类数目越多,每个聚类所包含的实体类别数目就越少,因此 $H(C)$也就越小;但同时也加重了对应于同一实体的表象的分散程度,因而 $H(E)$增大.由于 SS-EIM 采用了三段式逐步求精策略,在实体识别中综合考虑了属性文本特征、表象间语义关联以及表象组的约束规则,$H(C)$减少的速度大于 $H(E)$增长的速度.因此,与其他 3 种策略相比,准确性有了较大的提高(对应的 $H(C)$与 $H(E)$的平均值最小).

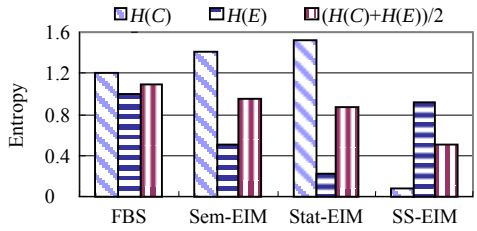

Fig.9 Performance comparison among SS-EIM, FBS, Sem-EIM and Stat-EIM

图 9 SS-EIM 与 FBS,Sem-EIM 和 Stat-EIM 的性能比较

8.6 AKM与N-AKM的性能比较

本实验将协调集内的数据通过人工进行实体识别来模拟用户实际查询中所得到的结果集,并将其应用在 SS-EIM 的动态协调过程中.然后将本文提出的静态分析与动态协调相结合的自适应知识维护策略(AKM)与只进行静态分析而没有动态协调的策略(N-AKM)进行比较,比较结果如图 10 所示.

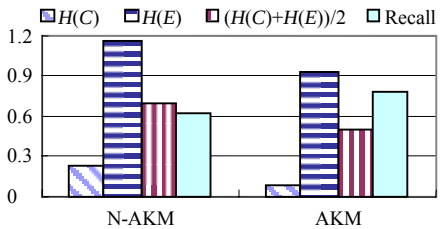

Fig.10 Performance comparison between AKM and N-AKM

图 10 AKM 与 N-AKM 的性能比较

由于 AKM 基于用户的查询结果对表象关联知识库中的内容不断扩充,与 N-AKM 相比,扩充后的表象关联知识库具有较高的完备性;另外,通过 AKM 策略将知识库中当前内容与实际结果数据进行对比,利用实体熵来衡量数据的有效性并及时修正,使得知识库的内容更能反映数据的真实状况,从而具备较高的知识准确性.

9 结 论

本文针对 Deep Web 数据集成中的实体识别问题进行了较为深入的研究,提出了一种基于语义及统计分析的实体识别机制,能够有效解决 Deep Web 数据集成中的数据消重及表象整合等问题.目前已完成的工作包括:基于文本匹配模型、语义分析模型和分组统计模型,构建了 SS-EIM 的整体模型框架;提出了文本粗略匹配、表象关联关系获取以及分组统计分析的三段式逐步求精策略,基于文本特征、语义关联及约束规则对表象进行聚类;提出了静态分析、动态协调相结合的自适应知识维护策略,利用用户实际的查询结果来不断完善和扩充表象关联知识库.通过模拟实验表明,由于 SS-EIM 综合考虑了文本特征、语义关联及约束规则,对识别结果进行不断精化,有效地提高了识别结果的准确性;通过对实体熵和聚类熵的折衷评价,以保证实体识别过程中相关参数的合理化配置;另外,在表象关联知识库构建过程中,由于采用静态分析、动态协调相结合的自适应知识维护策略对关联知识不断扩充和完善,在一定程度上保证了表象关联知识的有效性和完备性.

下一步,我们将针对语义分析与分组统计的性能改进、实体识别相关参数的合理化配置以及表象关联知识的智能化管理等方面进行深入研究.

References:

[1] Chang KCC, He B, Li CK, Patel M, Zhang Z. Structured databases on the Web: Observations and implications. SIGMOD Record, 2004,33(3):61−70.

[2] Guo ZM, Zhou AY. Research on data quality and data cleaning: A survey. Journal of Software, 2002,13(11):2076−2082 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/13/2076.pdf

[3] Bilenko M, Mooney R. Adaptive duplicate detection using learnable string similarity measures. In: Getoor L, ed. Proc. of the 9th ACM SIGKDD 2003. Washington: ACM Press, 2003. 39−48.

[4] Cohen WW, Ravikumar P, Fienberg SE. A comparison of string distance metrics for name-matching tasks. In: Kambhampati S, ed. Proc. of IJCAI-03 Workshop on Information Integration on the Web (IIWeb 2003). New York: AAAI Press, 2003. 73−78.

[5] Zhu HM, Wang NS. Improved method for detecting approximately duplicate database records. Journal of Control and Decision, 2006,21(7):805−813 (in Chinese with English abstract).

[6] Ling YY, Liu W, Wang ZY, Ai J, Meng XF. Entity identification for deep Web data integration. Journal of Computer Research and Development, 2006,43(Suppl.):46−53 (in Chinese with English abstract).

[7] Koudas N, Sarawagi S, Srivastava D. Record linkage: Similarity measures and algorithms. In: Chaudhuri S, ed. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Chicago: ACM Press, 2006. 802−803.

[8] Wang LJ, Guan SY, Wang XL, Wang XZ. Fuzzy C mean algorithm based on feature weights. Chinese Journal of Computers, 2006, 29(10):1797−1803 (in Chinese with English abstract).

[9] Das G, Hristidis V. Ordering the attributes of query results. In: Chaudhuri S, ed. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Chicago: ACM Press, 2006. 395−406.

[10] Nambiar U, Kambhampati S. Mining approximate functional dependencies and concept similarities to answer imprecise queries. In: Amer-Yahia S, ed. Proc. of the 7th Int'l Workshop on the Web and Databases (WebDB 2004). Paris: ACM Press, 2004. 73−78.

[11] Chaudhuri S, Granti V, Motwani R. Robust identification of fuzzy duplicates. In: Toyama M, ed. Proc. of the 21st Int'l Conf. on Data Engineering (ICDE 2005). Tokyo: IEEE Computer Society, 2005. 865−876.

[12] Chen ZQ, Kalashnikov DV, Mehrotra S. Exploiting relationships for object consolidation. In: Ozcan F, ed. Proc. of the 2nd Int'l ACM SIGMOD Workshop on Information Quality in Information Systems (IQIS 2005). Baltimore: ACM Press, 2005. 47−58.

[13] Thor A, Rahm E. MOMA—A mapping-based object matching system. In: Weikum G, ed. Proc. of the 3rd Biennial Conf. on Innovative Data Systems Research (CIDR 2007). Asilornar: Wisconsin, 2007. 247−258.

[14] Nie ZQ, Wen JR, Ma WY. Object-Level vertical search. In: Weikum G, ed. Proc. of the 3rd Biennial Conf. on Innovative Data Systems Research (CIDR 2007). Asilornar: Wisconsin, 2007. 235−246.

[15] Bhattacharya I, Getoor L. Iterative record linkage for cleaning and integration. In: Das G, ed. Proc. of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2004). Paris: ACM Press, 2004. 11−18.

[16] Dong X, Halevy A, Madhaven J. Reference reconciliation in complex information spaces. In: Ozcan F, ed. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Baltimore: ACM Press, 2005. 85−96.

[17] Wei M, Naumann F. DogmatiX tracks down duplicates in XML. In: Ozcan F, ed. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Baltimore: ACM Press, 2005. 431−442.

[18] Hall P, Dowling G. Approximate string matching. ACM Computing Surveys, 1980,12(4):381−402.

[19] Skikant R, Agrawal R. Mining generalized association rules. In: Dayal U, ed. Proc. of the 21st Int'l Conf. on Very Large Data Bases (VLDB 1995). Heidelberg: Springer-Verlag, 1995. 407−419.

[20] Liu B, Hsu W, Ma YM. Integrating classification and association rule mining. In: Agrawal R, ed. Proc of the 4th Int'l Conf. on Knowledge Discovery and Data Mining (KDD 1998). New York: AAAI Press, 1998. 80−86.

附中文参考文献:

[2] 郭志懋,周傲英.数据质量和数据清洗研究综述.软件学报,2002,13(11):2076−2082. http://www.jos.org.cn/1000-9825/13/2076.pdf

[5] 朱恒民,王宁生.一种改进的相似重复记录检测方法.控制与决策,2006,21(7):805−813.

[6] 凌妍妍,刘伟,王仲远,艾静,孟小峰.Deep Web 数据集成中的实体识别方法.计算机研究与发展,2006,43(增刊):46−53.

[8] 王丽娟,关守义,王晓龙,王熙照.基于属性权重的 Fuzzy C Mean 算法.计算机学报,2006,29(10):1797−1803.

寇月(1980－),女,辽宁沈阳人,博士生,助教,CCF 学生会员,主要研究领域为 Deep Web 数据管理.

李冬(1979－),男,工程师,主要研究领域为嵌入式软件环境.

申德荣(1964－),女,博士,教授,CCF 高级会员,主要研究领域为 Web 数据管理,分布式系统,数据网格.

聂铁铮(1980－),男,博士生,助教,CCF 学生会员,主要研究领域为 Web 数据集成.