计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20150252
52(8):1794-1805,2015

大数据环境下的电子商务商品实体同一性识别

胡亚慧[1,2] 李石君[1] 余 伟[1] 杨 莎[1,3] 甘 琳[1] 王 凯[1] 方其庆[2]
[1](武汉大学计算机学院 武汉 430079)
[2](空军预警学院 武汉 430019)
[3](汉口学院计算机科学与技术学院 武汉 430212)
(hyh5800@163.com)

Recognizing the Same Commodity Entities in Big Data

Hu Yahui[1,2], Li Shijun[1], Yu Wei[1], Yang Sha[1,3], Gan Lin[1], Wang Kai[1], and Fang Qiqing[2]
[1](*Computer School, Wuhan University, Wuhan* 430079)
[2](*Air Force Early Warning Academy, Wuhan* 430019)
[3](*School of Computer Science and Technology, Hankou University, Wuhan* 430212)

Abstract The recent blossom of big data and e-commerce has revolutionized our life by providing everyone with the ease and fun never before. How to identify the same commodity entities from these multi-source heterogeneous, fragmented, various and inconsistent e-commerce data for better business intelligence raises a very valuable and challenging topic. In this light, we analyze the characteristics of Web big data and collect the crawled original commodity information data from the different e-commerce platforms, which are the multi-source heterogeneous and mass scales of data. Then, we build an index model based on commodity's attributes and values, and construct a global model map to record the commodity's attribute and value, and form the unified model and high efficient commodity information for the next step. And we measure the similarity of the commodity's identity on the multilayer hierarchical probabilistic model, including identifying the possible candidate commodity set, similarity filtering the candidate commodity set and similarity filtering based on the special items of candidate commodities set. Finally, we output the same commodity set in the inverted index list. We also evaluate our method on the datasets collected from Chinese three main-stream B2C e-commerce platforms with Hadoop framework. Experimental results show the accuracy and effectiveness of our method.

Key words Web big data; e-commerce; hierarchical probabilistic model; commodity; Hadoop

摘 要 怎样从多源异构的、自治独立的、多样化的、不一致的电子商务数据中找出同一商品实体是当前面临的主要挑战.通过分析不同平台的数据特征,首先建立基于商品属性/值的索引模型,构造商品属性-值的全局模式图并进行模式集成,形成模式统一、质量高效的商品信息数据;而后基于层次概率模型对商品的同一性进行多层相似度量;最终完成商品实体识别,并归一化输出满足同一性的商品集和关联属性并进行排序.基于Hadoop平台对3个B2C电子商务数据源中的商品进行了实验,并与传统方法和产品进行了比较,实验结果证明了本框架的可行性、精确性和高效性.

收稿日期:2015-03-26;**修回日期**:2015-05-26
基金项目:国家自然科学基金项目(61272109);中央高校基本科研业务费专项资金项目(2042014KF0057);湖北省自然科学基金项目(2014CFB289);空军预警学院青年创新基金项目(2013ZDJC0101)
通信作者:李石君(shjli@whu.edu.cn)

关键词 Web 大数据;电子商务;层次概率模型;商品;Hadoop

中图法分类号 TP391

随着电子商务的不断发展,电子商务大数据为产业界和学术界带来了宝贵机遇和严峻挑战[1].到2013年12月底,国内B2C,C2C与其他电商模式的平台数达29 303家.这意味着电子商务在为人们的生活和工作带来便利的同时,更期待人们能够全面深入地发现和挖掘更有价值的信息和知识.

从各种电子商务数据源中自动识别出描述同一商品实体的所有网页,是进行数据集成和数据分析的基础.但是多源异构的电子商务数据具有数量巨大的商品种类、千差万别的模态、参差不齐的数据质量、杂乱多样的网站结构等特点,且缺乏统一的模式定义规范和理论模型,极大影响了电子商务大数据的分析和应用.为此,本文基于Map-Reduce架构提出了一种基于数据索引、数据集成、实体识别和数据排序的混合框架,在大数据环境下实现了通过机器学习进行实体识别,输出满足同一性的数据集和关联属性,并进行排序.该研究能有效地在电子商务大数据环境下的数据融合、数据搜索和个性化推荐等方面进行应用.

1 相关工作和科学问题

1.1 国内外研究现状

大数据中的商品实体同一性研究是数据实体识别的一个典型分支. Herndndez等人[2]、Arasu等人[3]利用经验知识给出解决实体同一性问题的准则,有效地解决了数据库中存在描述同一个现实世界实体的多数据元组,但不适用于求解多个数据源中的同一指代问题.Fan等人[4]首次形式化地提出了实体同一性描述规则,通过给出的规则推理问题,使得实体同一性描述规则不再是松散的集合,而是可以相互推理、相互配合,有效地提高了描述实体同一性的能力.另外,Chaudhuri等人[5]通过聚簇的方法、Chen等人[6]用机器学习的方法对多个属性相似性测量以判定实体同一性,这些方法在复杂的Web环境下精确度不够.Singla等人[7]利用基于Markov链的方法提出描述实体同一性的相似性测度.Augsten等人[8]通过滑动窗口描述XML节点之间的相似性,从而实现XML数据上的实体同一性判定,并取得了良好的效果,但不适用于Web大数据环境下的实体同一性识别.王立等人[9]设计了一种将数据集成、数据清理和商品归一化相结合的混合框架,通过相似度矩阵实现了商品之间的同一性度量,但当商品种类多样和数量巨大的情况下,该方法的精度和性能会出现波动.李建中等人[10-13]对大数据的可用性、XML实体抽取方法以及大数据中的实体识别进行了透彻全面的分析.韩家炜等人[14]对异构Web文本中的命名实体,提出了统计模型SHINE,采用迭代方法自动学习元路径的权重分布,这种基于统计的方式进行不一致的发现具有良好的效率.这些方法在分布式、非结构的Web大数据环境下,都具有极大的不确定性.

本团队之前的研究工作[15]是从站点结构、特征数据和知识规则3个方面对多源异构的数据进行抽取并建立Web数据模型.本文是将之前的研究工作具体到大数据环境下对电子商务中商品实体的同一性识别展开深入的研究.

1.2 主要科学问题

由于电子商品应用的特殊性和Web大数据的复杂性,使得在电子商务大数据中进行实体同一性研究充满了挑战,主要表现在:

1) Web大数据的挑战.当前的电子商务具有Web大数据的4V典型特征:

① 电子商务数据量巨大(volume),这是电子商务的重要特征.目前各类不同的电子商务平台超过了100家,主流电子商务平台中的商品种类数量平均在200万件以上,如果采用传统的遍历查找方法来对商品进行实体识别,时间复杂度为$O(m^n)$,其中m表示单个电子商务平台中的商品数量,$m>2\times10^6$,n表示电子商务平台的数量,$n>100$,因此采用传统方法进行分析是不可行的.

② 电子商务数据多样性(variety).商品种类繁多,不同商品的属性结构、描述方法以及不同平台的页面布局、商品组织方式、数据模态都不尽相同,如果将所有的平台和商品在一个统一的模型下进行分析研究是一个具有挑战性的问题.

③ 商品信息的增长具有高速性(velocity).每天都会有超过1 000个新商品被增加,商品价格的变化效率也很高,新出生的商品交易记录和用户评论记录更是庞大.因而需要能有效地对新增信息和更新数据进行同步.

④ 电子商务依托互联网进行信息更新,数据具

有一定的虚拟性(virtual),即信息可能没有经过严格的审查和校对,使得数据质量参差不齐,经常会出现缺失、矛盾、错误、不严谨的数据项,这为商品实体的挖掘带来了困难.

2) 标题的不确定性.由于各个电子商务数据源的规则不同,各个商家的行为习惯也各有差异,使得利用标题进行实体识别具有很大的困难,主要表现:

① 同一款商品在不同的商家进行销售时,因为习惯不同,所描述的标题文本上区别很大.例如同样是"罗技 G400S 鼠标",在不同的商家中标题分别为"顺丰送礼罗技最新款游戏有线鼠标 G400 升级版 G400S 正品"和"Logitech 罗技台式机笔记本鼠标 G400S 二代 USB 鼠标",这 2 个标题的文本相似度很低.

② 不同商品的标题描述,可能具有很大的相似性.例如对于手机商品"iPhone5"和"iPhone5S",他们的商品名称分别是"苹果(Apple)iPhone5 16 GB 版"和"苹果(Apple)iPhone5S 16 GB 版",这 2 个标题的文本相似度很大,实际上是不同的商品.

③ 很多商家为了增加曝光率和搜索命中率,在标题中增加了很多与商品无关的热门词汇.例如某销售"iPad4"的商品,标题为"国行 Apple/苹果 iPad4 16 GB WIFI 平板电脑 iPad2/3/4 超 iPad air5",产生大量的噪音,为标题分析带来了困难.

3) 商品属性与值的复杂性.任何一款商品的信息本质上由多个〈属性,值〉的数据对组成,例如商品的型号、材质、大小、颜色、编号等属性,其中标题、价格和详细介绍可以看作为特殊的专有属性单独进行分析,其他的作为普通属性统一进行研究,研究的主要难点在于:

① 商品的种类多样.不同的商品具有不同的属性结构,将所有的商品属性置于一个模型难度较大.

② 某属性相同意义的值描述在不同的数据源或商品中表现不同.例如同样是〈"颜色","白"〉的属性,在不同的商品页面中描述为〈"颜色","白色"〉和〈"颜色","亚光白"〉,这在传统数据关系模式下是属于不同的数据项.

③ 属性名表达不规范.同样是描述产品型号的属性,在不同的数据源中分别表述为"产品型号"、"商品型号"等.

④ 很多商品的部分属性的数据值缺失、错误甚至相互矛盾.

4) 短文本的语义分析.在对商品的属性、值的文本描述进行语义分析时,由于传统的语义分析方法在处理短文本时效果不佳且同一类型不同型号商品的描述相似性很高,确定是否是同一商品是很困难的.

基于上述原因,直接对电子商务大数据中的商品实体进行同一性识别难度较大.为解决这些问题,本文的主要贡献有:

1) 设计了基于海量商品实体的索引方法;

2) 提出了复杂商品模式属性/值的规范化处理方法以实现有效的数据集成;

3) 设计了完整的商品实体同一识别的理论方法和算法实现过程;

4) 通过真实数据的实验验证了本文的方法,并且对数据集合的大小具有良好的可扩展性.

2 问题的定义与描述

定义 1. 商品实体.商品实体是指电子商务大数据中描述现实存在的拥有具体特征的商品数据抽象,用符号 E 表示.例如商品 iPhone6 是一个独立的商品实体,该实体在不同的数据源中有不同的表示形式和具体存在形式,即具体的商品对象.

定义 2. 商品对象.商品对象是指在电子商务大数据中每一个商品记录数据所表示的对象.通常该对象是通过一个页面或一组页面进行描述的,因此也称为商品对象页面,用符号 W 表示.商品对象是唯一存在的具体数据,对于任意一个电子商务数据源 S_i,该数据源中包含了大量的商品网页,这些网页以不同的分类、不同的表现形式描述同一个商品信息.对于任意一个商品页面 W_j,包含了该商品网页的结构信息(所在栏目、页面布局)和内容信息,通过对页面进行结构分析、信息抽取和语义挖掘,得到商品的对象数据模型:

$$W_j=(C,E,B), \tag{1}$$

其中,C 表示该商品的所在栏目和结构信息,是对网站、栏目、子栏目和网页页面的抽象.将网站描述成一棵 5 层非空树,网站数据源是根节点,栏目及其各级子栏目是中间节点,网页页面是叶子节点.根据 SEO 的优化原则,每个网页最多离网站首页 4 次点击就能到达,所以将网站描述成 1 棵 5 层非空树,而且叶子节点的深度最大值为 5,则网页 W 的栏目结构信息可表示为 $C=\{c_1,c_2,c_3,c_4,c_5\}$,$c_1$ 为根节点,c_5 为最小节点,每个节点由该节点的基本信息和特征信息组成,电子商务数据源 S_i 的所有网页的 C 组成了网站 S_i 的栏目空间.

E 表示该商品对象所描述的商品实体,B 表示该页面所包含的特征属性数据项集合,在本文中将采用基于属性/值键值对的树模型来描述商品对象特征,后面再作详细介绍.

因此本文所要研究的问题是通过高效的算法实现自动从多个电子商务数据源 $S_1,S_2,\cdots,S_n$ 中找出所有描述同一实体 E 的所有商品对象 $W_1,W_2,\cdots,W_m$.

定义 3. 基于属性/值的树模型. Web 数据的结构和特征具有多样性,很难通过关系模式来定义所有的数据和实体的属性. 为了统一地表示所有的电子商务数据,本文提出一种基于键值型和树形结构结合的电子商务特征数据表示方法,其主要思想是将获取的所有数据进行初步处理后转换成具有层次关系的键值数据元,该模型包括键值型数据单元和数据单元关系关联. 一个对象数据由多个具有层次关系的元数据组成,每个元数据被封装为四元组:

$$Node=\langle E_i,P_i,key,value\rangle, \tag{2}$$

其中,E_i 为该元数据所描述的实体(非实体的则为空);P_i 标识了该元数据的父级对象;key 描述了该对象的属性名称(键名);$value$ 为可变长度数据(值)、存储数据值. 该模型表示模型自身不解析数据内容、不识别任何数据结构,这使其可处理所有数据类型,对于电子商务中任何的 Web 数据格式和数据类型都可以转换成该模型,使其具有高可扩展性.

对于任一商品,包含有许多描述该商品不同特征的属性/值,把属性/值看作树的节点,数据源看作根节点,构建不同数据来源的商品实体. 即对于任意一个商品 W,其特征项 $W.B$ 是由多个键值对节点 $Node$ 以一定方式组成的. 如图 1 所示,来自某个数据源的商品实体 iPhone,有若干个对象 iPhone5,iPhone6 等,商品对象 iPhone6 有名称、型号、内存、CPU 等若干属性,对于 CPU 属性又有具体的值.

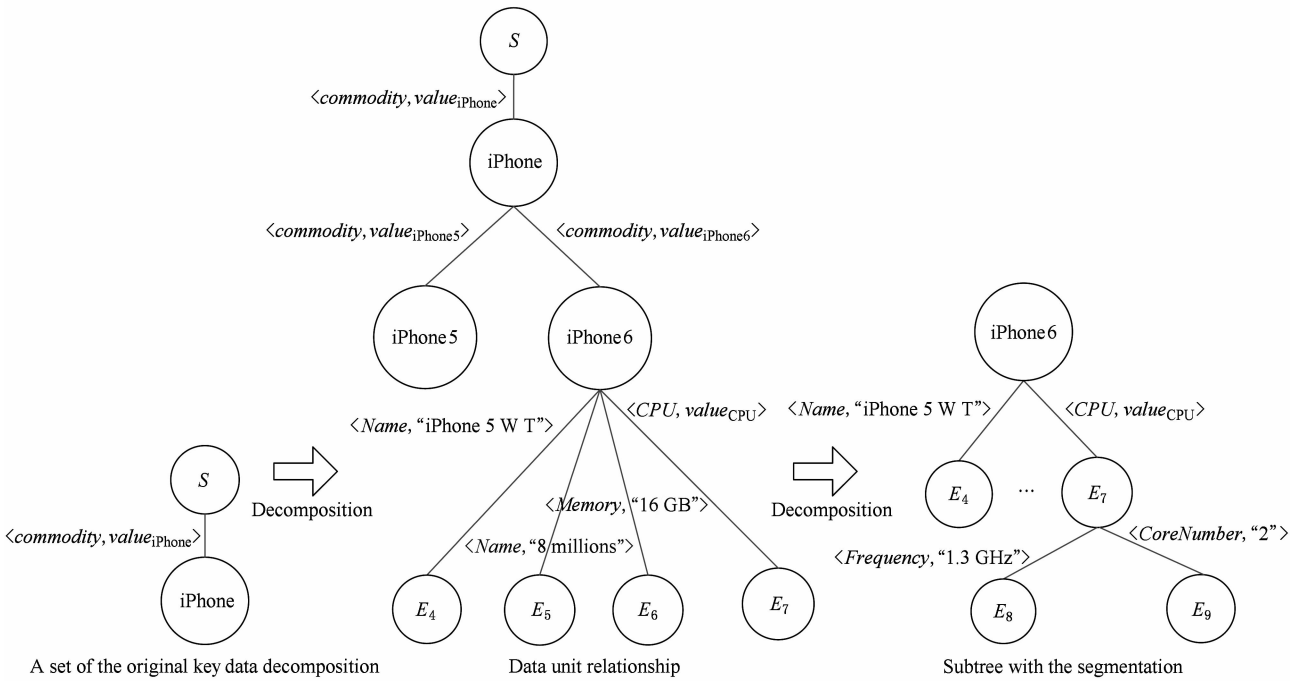


Fig. 1 Samples of attribute/value tree model.

图 1 基于属性/值的树模型示例

3 大数据中商品实体的索引

为降低多个数据源之间商品两两比较带来的 NP 难问题,本文将对商品每个属性的属性值构造倒排索引表,从而对同一实体的商品进行有效查询.

建立倒排索引集合 R,每一条倒排索引记录可以表示为 $R_i=\{A,V,Z\}$,其中 A 表示该条记录所记录的属性名称,V 表示其取值,Z 表示任意子树中包含有满足 $key=A$ 且 $value=V$ 的节点 $Node$ 的所有商品页面的集合,即 $Z=\{W_{i_1},W_{i_2}\cdots\}$. 对每个数据源中的每个商品页面进行遍历,提取该商品页面中的每一个特征项键值对 $Node$ 并构造成层次树 $W.B$,并对每个数据项进行检查. 如果节点 $Node$ 的 $\langle key,value\rangle$ 在 R 中存在,则在该记录的 Z 中增加该商品页面;如果节点 $Node$ 的 $\langle key,value\rangle$ 在 R 中不存在,则在 R 中增加一条记录 $\{key,value,\{W_i\}\}$;最后形成了所有数据源的所有商品的属性/值的倒

排索引表集合 R,这样完成了对所有商品的初步聚类,整个算法遍历一遍每一个数据源上的每一个商品页面,如算法 1 所示:

算法 1. 建立所有商品数据的属性/值倒排索引算法.

输入:每个页面的特征属性键值树 $W.B$;

输出:数据项$\langle A,V\rangle$的倒排索引表集合 R.

1) Map 阶段

输入:$key=W$,$value=$该商品的数据项集合;

步骤:

① $Map(key, value)$

② for each $item$ in $value$

③ $\langle A_{i_1},V_j\rangle, W_i \leftarrow Extraction(item)$;

④ end for;

⑤ $collect(\langle A_{i_1},V_j\rangle, W_i)$.

2) Reduce 阶段

输入:$key=\langle A_i,V_j\rangle$,$value=W_i$;

步骤:

① $Reduce(key, value)$

② for each $data$ in $Datalist$

③ if $key=data.key$

④ $data.value\ Push\ (data.value, value)$;

⑤ else

⑥ $Datalist \leftarrow DAdd(Datalist, key, value)$;

⑦ end if;

⑧ end for;

⑨ $collect(key, data.value)$.

/ * 按照数据项的属性/值进行归并 * /

算法 1 的 Map 阶段,按照属性和值进行映射,Reduce 阶段中将相同的属性和值进行归并以得到每个属性/值的倒排索引表.算法 1 对所有商品的属性/值都建立倒排,并将输出结果存储在 HDFS 中以用于后续处理.

4 属性/值的规范化处理

在实际的电子商务平台中,有许多表述不同但相互等价的属性节点和值节点,例如属性“显示芯片”和“显卡类型”是相互等价的,值“第 3 代智能英特尔酷睿 i5 处理器”和“Intel Core i5 3 代系列”是相互等价的.因此属性/值的规范化处理是合并那些相互等价的属性和值,从而有利于更准确地进行数据项比较和实体识别.

根据倒排索引集合 R 中的所有属性/值的记录,建立全局模式图 $G=\langle M,N,H\rangle$,其中 M 表示由所有属性组合成的点集,N 表示由所有值组成的点集,H 为连接属性和值的点集的带权边集合.对于任意属性 $A\in M$ 和值 $V\in N$,特征项$\langle A,V\rangle$在 R 中存在的商品集合 Z 的商品数为 k,则在 H 中存在一条权重为 k 的边$\langle A,V\rangle$,其权重记为 $\omega\langle A,V\rangle$.

定义 4. 等价值集合.等价值集合中的所有元素都是相互等价的值,这些值描述了相同或相似的意义.等价值集合是对相互等价值的合并,一般用累计权重最大的值元素作为代表节点.等价值集合表示为 $\bar{V}=\{V_1,V_2,\cdots\}$,其中的每个元素所表示的值是相互等价的.由于电子商务平台的复杂性,等价值集合可以扩展为所有元素的相似性大于某个给定的阈值 μ,即对于等价值集合 $\bar{V}=\{V_1,V_2,\cdots\}$, $\forall V_i\in\bar{V}$, $\forall V_j\in\bar{V}$:

$$Sim_{\text{value}}(V_i,V_j)\geqslant\mu_1. \tag{3}$$

构造等价值集合的本质是建立基于值文本语义相似度的聚类,采用 Brown 等人[16]提出的方法进行 2 个值 V_i 和 V_j 的语义相似性分析:

$$Sim_{\text{value}}(V_i,V_j)=\frac{L_i+L_j}{2L_iL_j}\times(\delta\times(1-\lambda\times(1-S_0))+\sum_{i\in X,j\in Y}\gamma_{ij}\times(1-\lambda\times(1-S_1))), \tag{4}$$

其中,L_i 和 L_j 分别表示 2 个值的概念数;λ 代表的是词序相似性对语义相似性的重要程度,它的值一般在 0.5 以下;X 和 Y 分别表示 2 个值的相似概念序列;γ 表示类似概念对的概念相似性,从相似性矩阵中获得.式(4)分为 2 个部分,$\frac{L_i+L_j}{2L_iL_j}\times(\delta\times(1-\lambda\times(1-S_0))$表示 2 个短文本长度因素,$\sum_{i\in X,j\in Y}\gamma_{ij}\times(1-\lambda\times(1-S_1)))$ 表示概念相似性和词序相似性的影响.如果 2 个短文本完全相同,则相似性为 1,具体构造等价值集合的方法如算法 2 所示:

算法 2. 对商品数据项中的值进行等价合并.

输入:原始的属性/值全局模式图 $G=\langle M,N,H\rangle$;

输出:进行值等价化处理的模式图 $G=\langle M,N',H'\rangle$.

1) Map 阶段

输入:$key=\langle A_i,V_i\rangle$,$value=H_i$;

步骤:

① $Map(key,value)$

② for each $data$ in $Datalist$

③ if $key.V_i = data.key$
④ $data.value \leftarrow Push(\langle key.A_i, H_i\rangle)$;
⑤ else
⑥ $Datalist \leftarrow DAdd(Datalist, key.V_i, \langle key.A_i, H_i\rangle)$;
⑦ end if;
⑧ end for;
⑨ $collect(data.key, data.value)$.

2) Reduce 阶段

输入:$\langle key = V_j\rangle$, $value = (\langle A_{j_1'}, H_{j_1'}\rangle, \langle A_{j_2'}, H_{j_2'}\rangle, \cdots)$;

步骤:

① $Reduce(key, value)$
② for each $value$ in $valueSet$
③ if $Sim_{value}(key, value) > \mu_1$ in each $value$
④ $\{value.key \leftarrow Push(key)$;
⑤ $value.value \leftarrow Update(key, value);\}$
⑥ else
⑦ $valueSet \leftarrow EAdd(valueSet, key, value)$;
⑧ end if;
⑨ end for;
⑩ $collect(value.key, value.value)$.

算法 2 的 Map 阶段将具有相同值的模式关系进行合并,并建立以值为键的索引;Reduce 阶段中,对输入的值 V_{j_1} 与之前每个等价值集合进行比较,如果该值与某集 V' 中的每一个值元素满足 $Sim_{value}(V_{j_1}, V_k') > \mu_1$,则将 V_{j_1} 加入到等价值集合 V' 中,并更新所有与 V_{j_1} 关联的属性点和关系边,形成新的对应关系($\langle A_{j_1'}, V', H_{j_1'}'\rangle, \langle A_{j_2'}, V', H_{j_2'}'\rangle, \cdots$). 如果输入的值 V_{j_1} 找不到合适的等价值集合,则将该值单独作为一个等价值集合. 最终实现所有值的等价合并.

定义 5. 等价属性集合. 等价属性集合中的所有元素描述商品的同一个特性,并且相互等价. 即等价属性集合 $\bar{A} = \{A_1, A_2 \cdots\}$ 中,$\forall A_i \in \bar{A}$, $\forall A_j \in \bar{A}$:

$$Sim_{attr}(A_i, A_j) \geqslant \mu_2. \tag{5}$$

但是由于属性的文本描述较短且缺乏上下文参考,采用文本语义相似度进行合并的效果有限. 例如“CPU 核数”和“CPU 个数”的相似度很高,但他们所描述的并不是相同的属性;又例如“屏幕尺寸”和“显示器大小”的相似度很低,但他们实际上是相同的属性. 因此在对属性进行等价合并时,不仅度量属性名称的语义相似性,还分析 2 个属性取值的值域正交情况,这是因为具有相同意义属性的取值范围具有很大的交集. 因此将属性节点之间的语义同一性定位为

$$Sim_{attr}(A_i, A_j) = \frac{1}{2} \times Sim_{str}(A_i, A_j) + \frac{1}{2} \times Sim_{range}(A_i, A_j), \tag{6}$$

其中,Sim_{str} 表示文本上的相似度,与式(4)计算方法相同;$Sim_{range}(A_i, A_j)$ 表示取值范围的契合度,$Sim_{range}(A_i, A_j) = \dfrac{Q(A_i) \bigcap Q(A_j)}{Q(A_i) \bigcup Q(A_j)}$,$Q$ 表示该属性的值域. 具体构造过程如算法 3 所示:

算法 3. 对商品数据项中的属性进行等价合并.

输入:原始值等价化处理的模式图 $G = \langle M, N', H'\rangle$;

输出:属性等价化处理的模式图 $G = \langle M', N', H''\rangle$.

1) Map 阶段

输入:$key = \langle A_i, V_i'\rangle$, $value = H_i'$;

步骤:

① $Map(key, value)$
② for each $data$ in $Datalist$
③ if $key.A_i = data.key$
④ $data.value \leftarrow Push(\langle key.V_i', H_i'\rangle)$;
⑤ else
⑥ $Datalist \leftarrow DAdd(Datalist, key.A_i, \langle key.V_i', H_i'\rangle)$;
⑦ end if;
⑧ end for;
⑨ $collect(data.key, data.value)$.

2) Reduce 阶段

输入:$\langle key = A_j\rangle$, $value = (\langle V_{j_1'}', H_{j_1'}\rangle, \langle V_{j_2'}', H_{j_2'}\rangle, \cdots)$;

步骤:

① $Reduce(key, value)$
② for each $value$ in $valueSet$
③ if $Sim_{attr}(A_{j_1}, A_k') > \mu_2$ in each $value$
④ $value.key \leftarrow Push(key)$;
⑤ $value.value \leftarrow Update(key, value)$;
⑥ else
⑦ $valueSet \leftarrow EAdd(valueSet, key, value)$;
⑧ end if;
⑨ end for;
⑩ collect(value.key, value.value).

算法 3 的 Map 阶段,将具有相同值的模式关系进行合并,并建立以属性为键的索引;Reduce 阶段中,对输入的属性 A_{j_1} 与之前每个的等价属性集合进行比较,如果该属性与某集合 A'_k 中的每一个属性元素满足 $Sim_{\text{attr}}(A_{j_1},A'_k)>\mu_2$,则将 A_{j_1} 加入到等价属性集合 A'_k 中并更新所有与 A_{j_1} 关联的值点和关系边,形成新的对应关系($\langle A'_{j'_1},V',H''_{j'_1}\rangle$,$\langle A'_{j'_2},V',H''_{j'_2}\rangle$,…).如果输入的属性 A_{j_1} 找不到合适的等价属性集合,则将该属性单独作为一个等价属性集合.最终实现所有属性的等价合并.

5 基于层次概率模型的实体同一性识别

电子商务商品的同一性识别的目的,是对每个商品找到其他同样商品描述的页面.在实际应用中,很难有 2 个页面的描述完全相同,因此将 2 个商品是同一实体的条件等价为:$Sim(W.B,W_i.B)>\sigma$.其中 Sim 为计算 2 个商品相似度的函数,下面对 Sim 进行详细分析说明.

商品信息的核心是由多个属性/值的数据项组成的集合,通过比较 2 个商品页面的数据项集合来度量他们的相似性和同一性.考虑到商品中存在一些特殊属性,不同属性对实体识别的影响不同,因此对属性进行分类和设定权值,商品的特征属性 $W.B$ 可以将每个特征键值对节点扩展为

$$Node_i=\langle E_i,P_i,key,value,T,P,\omega\rangle, \quad (7)$$

其中,$Node_i$ 表示该商品第 i 个节点,按先序遍历的遍历方式进行编号;前 4 个参数与式(2)中的相同;T 表示数据项类型,主要包括一般数据项和特殊数据项;P 表示第 i 个数据项的值的可信度;ω 表示第 i 个数据项的属性在实体识别中的权重,权重的取值范围为[0,1].

由于商品的数据量很大,对每个商品进行遍历比较效率较低,因此本文建立了一个 3 层的层次概率树.其每一层次的叶子左节点表示满足概率判定的候选商品集,每一层次的叶子右节点表示不满足条件的终节点.这样大大降低判定次数,提高了算法效率.对给定一个商品 W_a,找出其同一商品实体页面 3 个层次的每层判定方法:

1) 层次 1.找出可能为同一实体的候选商品集合.

商品 W_a 一共有 k 个普通数据项,本文将满足与这 k 个普通数据项中存在超过 φ_k 个相同的数据项的商品作为候选商品集.其中 φ 为候选阈值,$\varphi\in(0,1)$.

2) 层次 2.对候选商品集合进行相似性筛选.

候选商品 W_i 的数据项集合与商品 W_a 可能存在 4 种情况:

① 存在有数据项满足完全相等的条件,即满足:

$$W_i.B.Node_1.A=W_a.B.Node_1.A,$$
$$W_i.B.Node_1.V=W_a.B.Node_1.V.$$

② 候选商品中存在属性不在商品 W_a 中,即满足:

$$W_i.B.Node_2.A\notin W_a.B.Node.A.$$

③ 商品 W_a 中存在属性不在候选商品中,即满足:

$$W_a.B.Node_3.A\notin W_i.B.Node.A.$$

满足情况 2 和情况 3 的数据项称之为缺项.

④ 候选商品和商品 W_a 中存在相同的属性,但取值不一样:

$$W_i.B.Node_4.A=W_a.B.Node_4.A,$$
$$W_i.B.Node_4.V\neq W_a.B.Node_4.V.$$

满足情况 4 的数据项称之为矛盾项.

因此商品 W_a 和商品 W_i 的数据项相似度可以度量为

$$Sim_{\text{item}}(W_a.B,W_i.B)=$$
$$\left\{\sum_{i\in A_1}Node_i.\omega-\sum_{i\in A_4}Node_i.\omega\right\}\Big/$$
$$\left\{\sum_{i\in A_1}Node_i.\omega+\sum_{i\in A_2}Node_i.\omega+\sum_{i\in A_3}Node_i.\omega+\sum_{i\in A_4}Node_i.\omega\right\}, \quad (8)$$

其中,A_1,A_2,A_3,A_4 分别表示为情况①~④中的属性集;$Node_i.\omega$ 为对应属性的权重;$-\sum_{i\in A_4}Node_i.\omega$ 是表示因为数据明显矛盾而进行的相似度惩罚值.

最后将不满足 $Sim_{\text{item}}(W_a.B,W_i.B)\geqslant\tau$ 的商品移出候选商品集.

3) 层次 3.基于特殊数据项候选商品集合进行相似性筛选.

除了商品的一般属性外,还有一些例如标题、价格、图片、详情介绍、用户评论等特殊数据项.其中标题和价格是最具有代表性的,本文主要基于这 2 个特殊数据项进行 2 次筛选.

① 标题是最直接反映商品内容的属性,也是最直接体现不同商品区别的属性,理应具有最大的权重,但是由于不同的商家习惯且很多商家在标题中添加一些与本商品无关的热门关键词,导致同一商

品的标题相似度可能很低,也有可能导致不同商品的标题相似度很高,严重影响了实验结果.为了克服此问题,本文使用实体分词工具对标题进行关键名词提取,得到标题的关键词向量 $\boldsymbol{K}=(k_1,k_2,\cdots,k_n)$;然后将每个关键词和该商品的整个数据项集合中的所有值进行相关性度量,得到该关键词与商品的相关度,标准化后将其作为标题关键词向量的权重,得到权重向量 $\boldsymbol{\omega}_{\boldsymbol{K}}=(\omega_{k_1},\omega_{k_2},\cdots,\omega_{k_n})$,其中 $\sum_{i=1}^{n}\omega_{k_i}=1$;最后计算 2 个商品标题的相似性.

对于商品 W_a 和 W_b,其标题的关键词向量分别为 $\boldsymbol{K}_a=(k_{a_1},k_{a_2},\cdots,k_{a_n})$,$\boldsymbol{K}_b=(k_{b_1},k_{b_2},\cdots,k_{b_m})$,其权重向量分别为 $\boldsymbol{\omega}_{\boldsymbol{K}_a}=(\omega_{k_{a_1}},\omega_{k_{a_2}},\cdots,\omega_{k_{a_n}})$ 和 $\boldsymbol{\omega}_{\boldsymbol{K}_b}=(\omega_{k_{b_1}},\omega_{k_{b_2}},\cdots,\omega_{k_{b_m}})$,对 2 个商品的关键词向量进行合并为 $\boldsymbol{K}_{ab}=\boldsymbol{K}_a\cup\boldsymbol{K}_b=(k_{a'_1},k_{a'_2},\cdots,k_{a'_n},k_{b'_1},\cdots,k_{b'_m})$,其中 $(k_{a'_1},k_{a'_2},\cdots,k_{a'_m})$ 为商品 W_a 标题的向量,$(k_{b'_1},k_{b'_2},\cdots,k_{b'_m})$ 为商品 W_b 标题向量包含但不在 W_a 中的关键词组成的向量,即为 $\boldsymbol{K}_b-\boldsymbol{K}_a$.

因此商品 W_a 和 W_b 的权重向量可扩展为 $a_n+b'_m$ 维的向量,若该元素不在其原向量中,则权重为 0.因此 2 个商品标题的 Tanimoto 相似度可以表示为

$$Sim_{\text{title}}(W_a,W_b)=\frac{\boldsymbol{\omega}_{\boldsymbol{K}_a}\cdot\boldsymbol{\omega}_{\boldsymbol{K}_b}}{\|\boldsymbol{\omega}_{\boldsymbol{K}_a}\|^2+\|\boldsymbol{\omega}_{\boldsymbol{K}_b}\|^2-\boldsymbol{\omega}_{\boldsymbol{K}_a}\cdot\boldsymbol{\omega}_{\boldsymbol{K}_b}}. \tag{9}$$

对不满足 $Sim_{\text{title}}(W_a,W_b)\geqslant\xi$ 的商品进行过滤,其中 ξ 表示商品标题同一性的阈值.

② 价格是区别商品同一性的重要特征,尤其是同名商品和附属商品这类相似度较高但确实完全不同的商品.2 个商品价格之间的距离定义为

$$Sim_{\text{price}}(W_a,W_b)=\sqrt{1-\frac{|W_a.price-W_b.price|}{\max(W_a.price,W_b.price)}}. \tag{10}$$

对不满足 $Sim_{\text{price}}(W_a,W_b)\geqslant\rho$ 的商品进行过滤,其中 ρ 表示商品价格同一性的阈值.

最终度量 2 个商品页面 W_a 和 W_b 属于同一商品的可信度为

$$Sim(W_a.B,W_b.B)=\{Sim_{\text{item}}(W_a.B,W_b.B)+Sim_{\text{title}}(W_a,W_b)+Sim_{\text{price}}(W_a,W_b)\}/3. \tag{11}$$

对任意给定的商品进行同一性识别查询算法如算法 4 所示:

算法 4. 商品实体同一性识别查询算法.

输入:某商品 W_a;

输出:所有与商品 W_a 为同一实体的商品集合.

1) Map 阶段

输入:$key=W_a$,$value=W_a.B$;

步骤:

① $Map(key,value)$

② for each $item$ in $value$

③ $\langle A_{a_1},V_{a_1}\rangle,W_a\leftarrow Extraction(item)$;

④ end for;

⑤ $collect(\langle A_{a_1},V_{a_1}\rangle,W_a)$.

2) Reduce 阶段

输入:$key=\langle W_a\rangle$,$value=\langle A_{a_1},V_{a_1}\rangle$;

步骤:

① $Reduce(key,value)$

② $Z'\leftarrow\varnothing$;

③ for each $data$ in $avList(W_a)$

④ $Z\leftarrow Search(data)$;

⑤ end for;

⑥ $Z'\leftarrow Filter(\varphi_k,Z_1,Z_2,\cdots)$;

⑦ for each $Commodity$ in Z'

⑧ if $Sim_{\text{itemv}}(W_a.B,Commodity)<\tau$

⑨ $Z'\leftarrow Remove(Z',Commodity)$;

⑩ end if;

⑪ if $Sim_{\text{title}}(W_a,Commodity)<\xi$

⑫ $Z'\leftarrow Remove(Z',Commodity)$;

⑬ end if;

⑭ if $Sim_{\text{price}}(W_a,Commodity)<\rho$

⑮ $Z'\leftarrow Remove(Z',Commodity)$;

⑯ end if;

⑰ $\{Stafcom\leftarrow Commodity$;

⑱ $Similarity\leftarrow Sim(W_a.B,W_i.B);\}$

⑲ end for;

⑳ $collect(Stafcom,Similarity)$.

算法 4 的 Map 阶段,表示对查询结果的所有数据项进行属性/值分解;Reduce 阶段中,对 W_a 的每一个数据项在倒排索引表中查找具有相同数据项的商品集合 Z,找出 W_a 的所有 k 个数据项的 k 个商品集合 $Z_1,Z_2,\cdots,Z_k$ 中出现了超过 φ_k 次的商品,筛选后置于候选集合 Z' 中;计算 Z' 中每个商品与 W_a 的数据项相似度,对不满足 $Sim_{\text{item}}(W_a.B,W_i.B)\geqslant\tau$ 的商品 W_i 移出候选集合 Z';计算 Z' 中每个商品与 W_a 的标题相似度,对不满足 $Sim_{\text{title}}(W_a,W_b)\geqslant\xi$ 的商品 W_b 移出候选集合 Z';计算 Z' 中每个商品与 W_a 的价格距离,对不满足 $Sim_{\text{price}}(W_a,W_b)\geqslant\rho$ 的商

品 W_b 移出候选集合 Z';最终 Z' 中的商品为与 W_a 描述同一商品实体的页面,作为最终输出结果,并计算该商品 W_i 与 W_a 的相似度 $Sim(W_a.B, W_i.B)$ 作为 $value$ 输出,该值用于度量 2 个商品的同一性,在输出时根据该值进行排序.

6 实验结果及分析

为了有效地检验本文提出方法针对大规模电子商务数据并行计算的精确性和效率,进行了详细的实验.实验的数据集来源于对京东、易迅和 1 号店这 3 个中国主流的综合 B2C 电子商务平台的实时数据采集.实验采用了 Map-Reduce 框架的开源实现.

6.1 实验环境及数据集

本实验利用虚拟化技术,通过开源的虚拟化软件 Oracle Virtualbox 将高性能 NF8560M2 服务器虚拟出 30 个主机节点,并以此为底层的分布式硬件环境.每个节点虚拟出一颗 XENO E7-4807 的 CPU 和 4 GB 的内存,主机采用的是 Windows Server 2008 R2 的操作系统,节点采用的是 Ubuntu 12 操作系统的 Hadoop 0.20.2 平台.为了更充分地体现算法的效果,实验采用真实的数据集,通过自定义爬虫采集 3 个中国主流的综合 B2C 电子商务平台的商品信息,并对部分抽样数据加入了人工标注.截至到 2014 年 5 月,整个采集的商品信息包括 10 大类 38 个二级分类 938 781 件商品,不同平台不同大类商品数量如表 1 所示:

Table 1 Distribution of the Number of Various Classes of Dataset

表 1 数据集各分类的分布数量

Type	JD	YIXUN	YHD
Mobile	12 298	11 692	10 509
Digital	43 910	66 521	45 393
Computer	13 014	18 741	9 473
Appliance	96 518	98 379	64 852
Houseware	20 435	33 452	38 280
Jewelry	78 536	62 143	74 997
Cosmetics	23 736	20 541	13 988
Sports	5 260	5 214	2 448
Food	3 923	5 784	6 211
Daily Use	16 843	24 649	11 041
Total	314 473	347 116	277 192

6.2 评价标准

为了判断本文算法和实验的效果,从算法精确性和效率 2 个方面进行检验.

1) 精确性

本文采用平均准确率 P、平均召回率 R、平均综合评价指标 F_1 作为识别结果精确性的标准.设候选商品集为 $Z=\{W_1, W_2, \cdots, W_n\}$,第 k 个商品识别到的同一性商品为 $W'_k=\{W'_{k_1}, W'_{k_2}, \cdots, W'_{k_m}\}$,其中 k_m 表示第 k 个商品识别到的同一商品数,根据实际数据分析其中有 k_{m1} 个商品识别错误,真实的同一性商品数为 k_{m2},则整个算法的平均准确率:

$$P = \sum_{k=1}^{n} \frac{k_m - k_{m1}}{k_m} \Big/ n. \tag{12}$$

整个算法的平均召回率:

$$R = \sum_{k=1}^{n} \frac{k_m - k_{m1}}{k_{m2}} \Big/ n. \tag{13}$$

整个算法的平均综合评价指标:

$$F_1 = \frac{2 \times P \times R}{P + R}. \tag{14}$$

2) 效率

本文采用单位数据计算时间(running time per 100 thousand data, RT)和计算时间随数据量增长而增加的速度(increment speed, IS)来度量算法的效率.单位数据计算时间:

$$RT = 100\,000 \times \frac{T_1 + T_2 + T_3 + T_4}{Datasize}, \tag{15}$$

其中,T_1, T_2, T_3, T_4 分别表示算法 1、算法 2、算法 3、算法 4 的时间;$Datasize$ 表示数据集的大小.计算数据增长速度 IS 为不同大小数据集上 RT 的线性逼近斜率.

6.3 实验参数取值

本实验涉及到的参数较多,包括值同一性阈值 μ_1、属性同一性阈值 μ_2、数据项候选集阈值 φ、数据项同一性阈值 τ、标题同一性阈值 ξ、价格同一性阈值 ρ 这 6 个参数.

本实验采用 2 种人工标注的方式进行训练.1)抽验选择数据集,将数据集内商品的所有同一性商品页面进行人工提取,构造精确的数据集进行初始训练;2)根据初始训练参数进行识别,对结果进行抽验和标注,根据标注结果进行参数调整,求解满足最优的 ROC 指标的参数值.训练得到的各阈值参数如表 2 所示:

Table 2 Threshold Parameters

表 2 本文所采用的阈值参数

Parameter	Value
μ_1	0.82
μ_2	0.93
φ	0.61
τ	0.49
ξ	0.71
ρ	0.79

6.4 实验结果分析

本文的实验是对整个数据集中的所有商品进行同一性实体识别和聚类,为了有效地对本文的方法和实验进行验证,构造了另外 3 组实验. 第 1 组实验在同样的硬件环境下采用传统的多线程并发,软件环境采用 Java+Oracle 11g 进行计算,称之为基于 Oracle 的商品识别方法;第 2 组实验利用奇虎 360 科技有限公司出品的 360 购物搜索进行结果比较;第 3 组实验利用网易有道信息技术有限公司出品的惠惠购物助手进行结果比较.

第 1 组实验是用来对比针对大数据的信息处理中,采用 Map-Reduce 和采用传统多线程并发的性能区别. 表 3 反映了本实验和第 1 组实验在算法各个阶段的耗时情况:

Table 3 Time Consuming Based on Map-Reduce and Oracle

表 3 基于 Map-Reduce 和 Oracle 各阶段耗时 s

Stage	Map-Reduce	Oracle
Index	1 681	4 662
Normalization	2 416	5 123
Identification	9 237	32 136
Total	13 334	41 921

表 3 的数据表明,本文基于 Map-Reduce 的商品同一性识别算法在各个阶段的运行效率都要优于传统方法.

为了度量 2 种方法在数据量增长后的性能,从数据集中抽样了 10 万、20 万、30 万、40 万、50 万、60 万、70 万、80 万、90 万的子数据集进行实验,并与整个数据集(93.8 万)进行比较,图 2 中的每个点表示在该数据集下不同方法的计算时间. 结果表明基于传统多线程的方法在关系数据库 Oracle 中,数据量较少时的性能要优于本文基于 Map-Reduce 的方法;但是随着数据量的增长,基于 Map-Reduce 体现了其强大的优势.

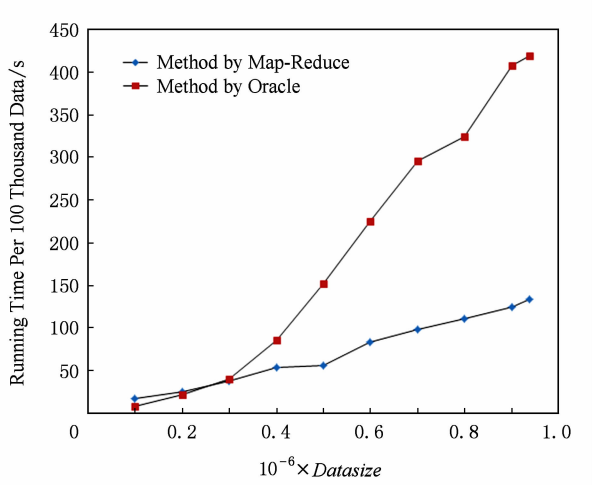


Fig. 2 Time cost comparison of different data by Map-Reduce and Oracle.

图 2 基于 Map-Reduce 和 Oracle 不同数据耗时比较

这是因为数据集较小时,Map-Reduce 框架的并行性能无法得到充分发挥,只有在数据量大的情况下,本算法有效地调用了各个进程执行并行任务,因此运行效率得到了明显的改善. 表 4 表明本文采用的 Map-Reduce 算法在运行效率的各个方面都优于传统方法:

Table 4 Performance Comparison Between Map-Reduce and Oracle

表 4 基于 Map-Reduce 和基于 Oracle 的性能比较 s

Performance	Map-Reduce	Oracle
RT	1 420	4 465
IS	142.6	534.7

第 2 组和第 3 组分别使用奇虎 360 科技有限公司出品的 360 购物搜索和网易有道信息技术有限公司出品的惠惠购物助手进行实验. 实验方法是针对随机的一组商品页面,在这 2 个平台上分别获取同一商品,然后对其结果进行验证. 比较的结果如表 5 所示:

Table 5 Accuracy Comparison of our Algorithm with Map-Reduce and Oracle

表 5 本文算法和第 2,3 组实验精确性比较

Accuracy	Our Algorithm	360 Shopping Search	Huihui Shopping Search
P	0.85	0.82	0.52
R	0.89	0.71	0.91
F_1	0.81	0.76	0.66

实验结果表明,本文所提出的方法在准确率、召回率和综合评价指标方面都优于360购物搜索,召回率略低于惠惠购物助手;但准确率和综合评价指标远高于惠惠购物助手.经过分析,这是因为惠惠购物助手获取的实体同一性商品列表中有许多类似但不相同的商品,因此召回率得以提高但准确率急剧下降.

从3组实验可以得出,本文算法以及在Map-Reduce环境下的实验,具有较高的精确性和运行效率,且算法的效率与数据集的大小影响不大,即便在更庞大的数据集和更复杂的数据环境中本文算法都有良好的适用性.

7 结束语

当前异构多源的电子商务具有典型的大数据特征,由于数据量庞大、数据来源复杂、数据模态千差万别、质量参差不齐、网站结构杂乱多样且缺乏统一的模式定义规范和理论模型,使得传统的学习和识别方法不适用于电子商务大数据的数据集成和同一性识别,因此基于Map-Reduce架构提出了一种基于数据索引、数据集成、实体识别和数据排序的混合框架.该框架首先建立基于商品属性/值的索引模型,构造商品属性/值的全局模式图并进行模式集成,形成模式统一、质量高的商品信息数据;而后基于层次概率模型对商品的同一性进行多层相似度量;最终完成商品实体识别,输出满足同一性的商品集和关联属性并进行排序.本框架在Hadoop平台上对3个B2C电子商务数据源中的商品进行了实验,并与传统方法和其他产品进行了比较.实验结果证明,本框架具有较高的精确度以及良好的高效性和可扩展性.然而电子商务大数据环境还有更多的复杂性,各种不同的数据模态、各位故意的数据误差以及属性/值之间的交叉性和相容性等,如何在更为复杂的应用环境下更高效地解决问题,将是下一步的研究工作.

参 考 文 献

[1] Meng Xiaofeng, Li Yong, Zhu Jianhua. Social computing in the era of big data: Opportunities and challenges [J]. Journal of Computer Research and Development, 2013, 50(12): 2483-2491 (in Chinese)
(孟小峰, 李勇, 祝建华. 社会计算: 大数据时代的机遇与挑战[J]. 计算机研究与发展, 2013, 50(12): 2483-2491)

[2] Herndndez M A, Stolfo S J. The merge/purge problem for large databases [C] //Proc of the 1995 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 1995: 127-138

[3] Arasu A, Kaushik R. A grammar-based entity representation framework for data cleaning [C] //Proc of the ACM SIGMOD Int Conf on Management of Data (SIGMOD 2009). New York: ACM, 2009: 233-244

[4] Fan Wenfei, Jia Xibei, Li Jianzhong, et al. Reasoning about record matching rules [C] //Proc of the 35th Int Conf on Very Large Data Bases. Trondheim, Norway: VLDB Endowment, 2009: 407-418

[5] Chaudhuri S, Ganti V, Motwani R. Robust identification of fuzzy duplicates [C] //Proc of the 21st Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2005: 865-876

[6] Chen Z, Kalashnikov D V, Mehrotra S. Adaptive graphical approach to entity resolution [C] //Proc of the 7th ACM IEEE-CS Joint Conf on Digital Libraries. New York: ACM, 2007: 204-213

[7] Singla P, Domingos P. Entity resolution with Markov logic [C] //Proc of the 6th IEEE Int Conf on Data Mining. Piscataway, NJ: IEEE, 2006: 572-582

[8] Augsten N, Bohlen M, DyresonC, et al. Approximate joins for data-centric XML [C] //Proc of the 24th Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2008: 814-823

[9] Wang Li, Zhang Rong, Sha Chaofeng, et al. A product normalization method for e-commerce [J]. Chinese Journal of Computers, 2014, 37(2): 312-325 (in Chinese)
(王立, 张蓉, 沙朝锋, 等. 电子商务商品归一化方法研究[J]. 计算机学报, 2014, 37(2): 312-325)

[10] Li Jianzhong, Liu Xianmin. An important aspect of big data: Data usability [J]. Journal of Computer Research and Development, 2013, 50(6): 1147-1162 (in Chinese)
(李建中, 刘显敏. 大数据的一个重要方面: 数据可用性[J]. 计算机研究与发展, 2013, 50(6): 1147-1162)

[11] Liu Xianmin, Li Jianzhong. Key-based method for extracting entities from XML data [J]. Journal of Computer Research and Development, 2014, 51(1): 64-75 (in Chinese)
(刘显敏, 李建中. 基于键规则的XML实体抽取方法[J]. 计算机研究与发展, 2014, 51(1): 64-75)

[12] Huo Ran, Wang Hongzhi, Zhu Rong, et al. Map-Reduce based entity identification in big data [J]. Journal of Computer Research and Development, 2013, 50(Suppl Ⅱ): 170-179 (in Chinese)
(霍然, 王宏志, 朱镕, 等. 基于Map-Reduce大数据实体识别算法[J]. 计算机研究与发展, 2013, 50(增刊Ⅱ): 170-179)

[13] Li Lingli, Wang Hongzhi, Gao Hong, et al. EIF: A framework of effective entity identification [C] //Proc of the 11th Int Conf on Web-Age Information Management. Berlin: Springer, 2010: 717-728

[14] Wei Shen, Han Jiawei, Wang Jianyong. A probabilistic model for linking named entities in Web text with heterogeneous information networks [C] //Proc of Int Conf on Management of Data(SIGMOD 2014). New York: ACM, 2014: 1199-1210

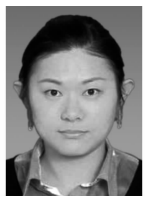
[15] Yu Wei, Li Shijun, Yang Sha, et al. Automatically discovering of inconsistency among cross-source data based on Web big data. Journal of Computer Research and Development, 2015, 52(2): 295-308 (in Chinese)
(余伟, 李石君, 杨莎, 等. Web大数据环境下的不一致跨源数据发现[J]. 计算机研究与发展, 2015, 52(2): 295-308)

[16] Brown P F, Della Pietra S A, Della Pietra V J, et al. Word sense disambiguation using statistical methods [C] //Proc of the 29th Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 1991: 264-270

Hu Yahui, born in 1980. PhD candidate in Wuhan University and lecturer in Air Force Early Warning Academy. Her research interests include data mining and location based service.

Li Shijun, born in 1964. Professor and PhD supervisor in the Computer School of Wuhan University. Member of China Computer Federation. His main research interest include data ming.

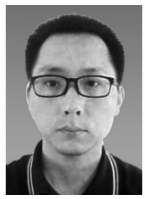
Yu Wei, born in 1987. PhD. Lecturer in the Computer School of Wuhan University. Member of China Computer Federation. His research interests include Web data ming(yuwei@whu.edu.cn).



Yang Sha, born in 1980. PhD candidate in Wuhan University and lecturer in Hankou University. Student member of China Computer Federation. Her research interests include data quality evaluation (chloe4doc@163.com).

Gan Lin, born in 1988. PhD candidate in Wuhan University. Her research interests include data mining and big data(85906478@qq.com).

Wang Kai, born in 1988. PhD candidate in Wuhan University His research interests include urban computing and data mining (kevin.w@whu.edu.cn).



Fang Qiqing, born in 1979. PhD. Lecturer in Air Force Early Warning Academy. His main research interests include data mining (fangqiqing@yeah.net).