DOI:10.7544/issn1000-1239.2016.20150661
53(1):165-192, 2016

知识库实体对齐技术综述

庄 严 李国良 冯建华
(清华大学计算机科学与技术系 北京 100084)
(joyear2008@163.com)

A Survey on Entity Alignment of Knowledge Base

Zhuang Yan, Li Guoliang, and Feng Jianhua
(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

Abstract Entity alignment on knowledge base has been a hot research topic in recent years. The goal is to link multiple knowledge bases effectively and create a large-scale and unified knowledge base from the top-level to enrich the knowledge base, which can be used to help machines to understand the data and build more intelligent applications. However, there are still many research challenges on data quality and scalability, especially in the background of big data. In this paper, we present a survey on the techniques and algorithms of entity alignment on knowledge base in decade, and expect to provide alternative options for further research by classifying and summarizing the existing methods. Firstly, the entity alignment problem is formally defined. Secondly, the overall architecture is summarized and the research progress is reviewed in detail from algorithms, feature matching and indexing aspects. The entity alignment algorithms are the key points to solve this problem, and can be divided into pair-wise methods and collective methods. The most commonly used collective entity alignment algorithms are discussed in detail from local and global aspects. Some important experimental and real world data sets are introduced as well. Finally, open research issues are discussed and possible future research directions are prospected.

Key words knowledge base; entity alignment; similarity propagation; probabilistic model; similarity function; blocking and indexing

摘 要 知识库的实体对齐(entity alignment)工作是近年来的研究热点问题.知识库实体对齐的目标是能够高质量链接多个现有知识库,并从顶层创建一个大规模的统一的知识库,从而帮助机器理解底层数据.然而,知识库实体对齐在数据质量、匹配效率等多个方面存在很多问题与挑战有待解决.从这些挑战出发,对十几年来的可用于知识库实体对齐的技术和算法进行综述,通过分类和总结现有技术,为进一步的研究工作提供可选方案.首先形式化定义了知识库实体对齐问题;然后对知识库的实体对齐工作进行总体概述,并从对齐算法、特征匹配技术和分区索引技术3个方面详细总结了各种可用方法和研究进展,重点从局部和全局2个角度对主流的集体对齐算法进行详细阐述,并介绍了常用的评测数据集;最后对未来重点的研究内容和发展方向进行了探讨和展望.

关键词 知识库;实体对齐;相似性传播;概率模型;相似性函数;分区索引

中图法分类号 TP311.13; TP182

收稿日期:2015-07-15;**修回日期**:2015-10-16

基金项目:国家自然科学基金优秀青年科学基金项目(61422205);国家"九七三"重点基础研究发展计划基金项目(2015CB358700)
This work was supported by the National Natural Science Foundation for Excellent Young Scholars of China (61422205) and the National Basic Research Program of China (973 Program) (2015CB358700).

自20世纪90年代蒂姆·伯纳斯·李发明万维网(World Wide Web, WWW,简写为Web)以来,随着Web技术的不断向前发展,人们经历了以文档互联为主要特征的"Web 1.0"时代和基于社交网络的以人与人互联为主要特征的"Web 2.0"时代,正在迈向基于知识互联的"Web 3.0"时代[1]. 知识互联的目标是实现人和机器都可理解的万维网,使得我们的网络更加智能化. 在这种背景下,近十几年来互联网上产生了越来越多的大规模知识库,这些知识库包含娱乐、金融、政务、出版发行和生物医学等互联网上各个方面的知识,为人们更加智能地使用Web提供了更好的方案.

然而,由于任何机构或组织都可以根据自己的需求和设计理念创建知识库,因此知识库中的数据也充满多样性和异构性,并且存在很多相互的重复或补充. 知识库的对齐问题也吸引了越来越多的研究者的目光,尤其是在信息检索、机器阅读和知识问答等领域都具有重要的应用价值. 举例来讲,当人们对某支股票感兴趣的时候,系统可以根据需求将这只股票相关的政务、经济信息知识库和公司信息知识库以及股票行情变化规律知识库对齐起来以自动获得更加全面和准确的总体情况,将综合结果呈现给用户,从而辅助用户做出正确的交易决策. 但是,由于数据质量、对齐效率等多方面因素使得这些知识库的对齐工作存在大量的困难和挑战.

知识库一般使用RDFS(resource description framework schema)或者OWL(Web ontology language)等语言描述的本体构建,这里的本体是一种采取不同的结构化形式表示的形式化的世界知识,其中定义了类别(class)、属性(property)和实例(instance)等基本元素,这些元素都可以看作是知识库中的实体(entity). 知识库的对齐的研究工作开始于"本体匹配"(ontology matching)[2-4],初期主要是针对本体类别的语义相似性进行匹配. 近几年来,随着知识库规模的扩大和实例数量的增加,不同知识库之间的实例链接的重要性日益体现,知识库中的实体对齐更加偏重于实例方面的匹配工作,这也是本文知识库对齐的主要研究内容.

实体对齐(entity alignment)也称为实体匹配(entity matching)或实体解析(entity resolution),是判断相同或不同数据集中的2个实体是否指向真实世界同一对象的过程. 数据库领域中,对象共指的消解常被称为记录链接(record linkage)、重复检测(duplicate detection)或记录匹配(record matching)[5-7];在自然语言处理和信息检索领域,常称之为共指消解(coreference resolution)[8-9],属于指代消解(anaphora resolution)中的一类工作;在语义Web领域,也称之为引用调和(reference reconciliation)或对象合并(object consolidation). 这些工作在数据清洗、数据集成和数据挖掘等方面中起着重要的作用[10].

实体对齐相关问题从数据库诞生之日起就被人们所重视,从20世纪六七十年代提出到现在,实体匹配技术也经历了一系列的发展变化. 知识库实体对齐是实体匹配发展到Web 3.0后,在不同知识库的链接过程中提出的一种问题,这个问题可以通过将经典的实体匹配技术应用到知识库领域,结合知识库的特点进行实体匹配来解决. 不同知识库的链接是实现Web 3.0提出的"知识之网"愿景的关键步骤,具有丰富的应用场景和重要的意义,特别是万维网联盟的链接开放数据项目(linking open data project, LOD)[11]的兴起和发展,为这方面的研究工作注入了更大的活力与动力,因此知识库的实体对齐技术具有重要的研究价值,是当前的热点研究方向.

本文对当前知识库实体对齐技术进行综述,对知识库实体对齐的相关算法和技术进行系统化地总结,以期应对这一领域所面临的挑战以及为进一步研究工作提供帮助.

1 问题描述

1.1 知识库实体对齐的相关概念

本文首先对论述的知识库进行定义. 知识库可以看作是对客观世界的事物及其相互关系的一种形式化描述. 当前知识库的定义有很多种,本文选择六元组的方式进行定义[12]. 这里采用RDF(resource description framework)规范定义一条事实三元组,即一条事实三元组分别由主语、谓语和宾语组成,可以简写为SPO(subject-predicate-object),则知识库的定义可以描述为

定义1[12]. 知识库. 一个知识库是一个由以下方式构成的六元组:$KB=(I,L,R,P,FR,FP)$. 其中I,L,R,P分别为1组实例、字面量、关系和属性的集合;$FR\subseteq I\times R\times I$是一个SPO三元组表示宾语为实例的关系事实;$FP\subseteq I\times P\times L$是一个SPO三元组表示宾语为字面量的属性事实.

明确知识库的定义后对知识库的实体对齐进行形式化定义. 知识库的实体可以包括知识库的任何一个元素,特别地,本文将知识库的实体对齐定义为知识库中的实例匹配,其形式化定义为[13]

$$Align_{\text{entity}}(KB_1, KB_2) = \{(e_1, e_2, con) \mid e_1 \in KB_1, e_2 \in KB_2, con \in [0,1]\},$$

其中,con 为一个刻画实体相似性大小的数值,con 越大则 2 个实体越相似.

2 个知识库实体对齐的过程如图 1 所示,可以简单描述为:给定 2 个知识库和 1 组先验对齐的数据(也可称为对齐算法的训练数据,详细内容参见 1.3 节),在可选的调节参数和一系列相关的外部资源(或称为背景数据,本文不涉及这方面的研究工作)共同控制下进行实体匹配的计算最终得到对齐结果.其中实体对齐的详细过程和相关算法是本文的主要研究内容.

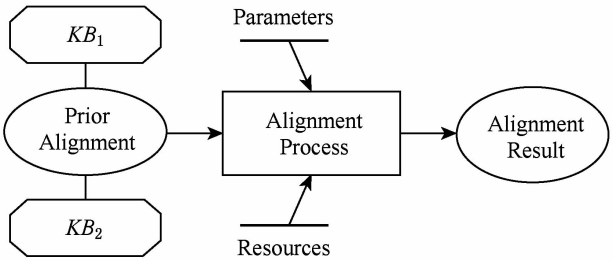


Fig. 1　Process of entity alignment of knowledge base.

图 1　知识库实体对齐过程

1.2　对齐质量和效率的评价相关概念

实体对齐效果的评价可分为质量和效率 2 个方面.

1) 质量评价主要是指对齐的准确性和全面性的评价指标.我们知道,考虑一个二分类问题,可以将实例分成正类(positive)或负类(negative),因此会出现 4 种情况:如果一个实例是正类并且也被预测成正类,即为真正类(true positive, TP);如果实例是负类被预测成正类,称之为假正类(false positive, FP);相应地,如果实例是负类被预测成负类,称之为真负类(true negative, TN);正类被预测成负类则为假负类(false negative, FN).常用的对齐质量评价指标有精度($precision$)、召回率($recall$)、F-$measure$ 以及使用图形表示的 $precision$-$recall$ 曲线等.下面对这 4 个度量指标进行简要介绍[14-16].

① 精度.精度也称为查准率,用来衡量分类结果的质量,定义为被分类器判断为正类的实例中正确分类的比例,即:

$$precision = \frac{TP}{TP + FP}.$$

② 召回率.召回率也称为查全率,用来衡量分类器发现正确匹配的能力,定义为分类器将正类判断为正类的比例,即:

$$recall = \frac{TP}{TP + FN}.$$

③ F-$measure$.F-$measure$ 也称为 f-$score$ 或 f_1-$score$,是综合考虑精度和召回率的一个评价指标,定义为精度和召回率的调和均值,即:

$$F\text{-}measure = 2 \times \left(\frac{precision \times recall}{precision + recall}\right).$$

④ $precision$-$recall$ 曲线.在指定参数条件下,在二维坐标系中将召回率作为 x 轴、精度作为 y 轴绘制的曲线.实验证明,在精度和召回率之间存在着相反的相互依赖关系:如果提高输出的精度,就会降低其召回率,反之亦然.因此图 2 中曲线右上角部分的值越高则分类器效果越好.

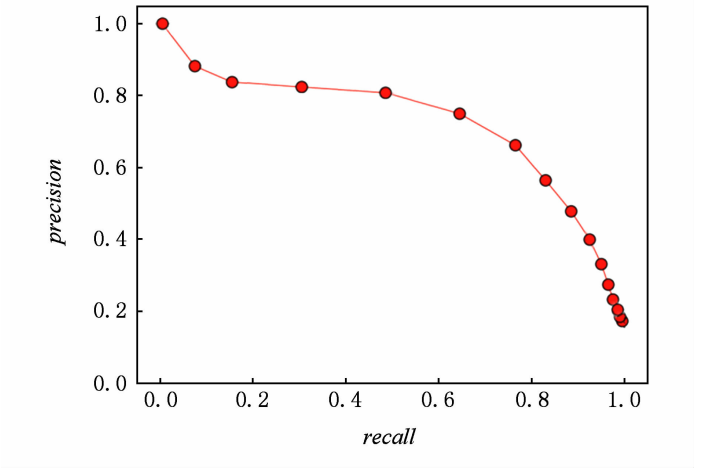


Fig. 2　$Precision$-$recall$ curve.

图 2　精度-召回率曲线

2) 效率评价主要是指对齐算法中一些分区索引技术对候选匹配对的筛选能力和准确性的度量评价标准.我们定义待匹配数据集中匹配实体对的数量为 n_M,不匹配的实体对数量为 n_N.类似地,经过某种分区索引技术得到的候选对中正确匹配的数量为 s_M,不匹配的数量为 s_N.常用的对齐效率评价指标有缩减率(reduction ratio, RR)、候选对完整性(pairs completeness, PC)、候选对质量(pairs quality, PQ)等.下面对这 3 个度量指标进行简要介绍[14].

① 缩减率.缩减率是用来衡量索引技术减少待匹配实体对数量能力的参数,定义为在不考虑候选对质量的条件下,使用索引减少的实体对与全部待匹配实体对的比值,即:

$$RR = 1 - \frac{s_M + s_N}{n_M + n_N}.$$

② 候选对完整性.对应与本节的召回率,是用来衡量索引产生的候选对的质量的度量指标之一,定义为索引产生的候选对中匹配对的数量与全部待匹配数据集中匹配实体对的数量的比值,即:

$$PC = \frac{s_M}{n_M}.$$

可以看出,这个指标的值越低,则会有越多的匹配

对被索引排除,会导致总体匹配的召回率降低.

③ 候选对质量.对应与本节的精度,是用来衡量索引产生的候选对的质量的度量指标之一,定义为索引产生的正确匹配的候选对数量与全部产生的候选对数量的比值,即:

$$PQ = \frac{s_M}{s_M + s_N}.$$

可以看出,这个指标的值越高,则该索引发现匹配实体对的能力越强,并会使总体匹配的精度有所提高.

1.3 问题与挑战

知识库实体对齐,尤其是在大数据条件下,存在许多问题和挑战,其中最突出是计算复杂度、数据质量和先验对齐数据的获取问题,都需要根据知识库实际情况设计有效算法进行解决.

1) 计算复杂度挑战.一般进行 2 个知识库实体匹配的时候,为了发现所有的匹配对,需要将一个知识库中所有实体与另一个知识库中所有实体进行比较,这将导致匹配算法的计算复杂度随着知识库的规模二次增长,在大数据条件下,其计算规模是难以接受的,而实际上可能的匹配对的数量不会超过规模较小的知识库实体数量.为了解决这个矛盾,需要设计高效的算法在保证精度和召回率的前提下尽可能减少候选对的数量,使复杂的匹配计算只在最有可能的候选对中进行.

2) 数据质量挑战.知识库对齐中的数据质量问题主要是由于不同的知识库的构建目的和构建方式不同,具体有以下 5 方面的表现:①相同实体有不同名字;②相同名字指代不同实体;③实体定义粒度不同;④相同的属性在不同知识库中具有不同的判别能力;⑤相同类别的实体在不同知识库中具有不同数量的属性.此外,由于格式、单位、大小写、空格、缩写名词、录入错误等也会给匹配过程带来很多困难.通常,这类问题可以通过数据预处理技术进行解决,但知识库对齐中的数据质量问题还需要综合考虑知识库架构方面的因素设计相应的算法来解决.

3) 先验对齐数据的获取挑战.先验对齐数据也称为训练数据,在知识库实体对齐过程中具有重要作用,无论是对匹配的准确度还是算法的收敛速度都会产生重要影响.然而,在知识库中尤其是大规模的通用知识库中这种先验数据并不容易获得.为了解决这个问题,需要针对知识库的不同情况采用不同的方法:对于具有 URI 的实体,可以直接通过 URI 来判断是否相等;对于实体名称完全一样的实体,可以确定它们相等的概率很大;对于具有 OWL:sameAs 属性的实体,可以通过属性值来判断是否相等;对于具有 OWL:Inverse Functional Property(IFP)属性的实体,可以通过其一条事实的 IFP 属性的宾语相等推导出其主语相等;如果以上条件都不满足,还可以通过对每个知识库的实体进行 IFP 属性的推导来启发式地获得近似的 IFP,进而获得先验对齐数据[17].通过这些手段可以部分地解决缺少训练数据的问题.此外主动学习及众包算法等都可以作为获得训练数据的有效手段,因此仍需将这些方法整合起来以便找出更多的高质量的先验对齐数据.

2 知识库实体对齐技术概述

高效的实体对齐算法的设计与实现是解决知识库实体对齐问题的关键手段,也是知识库实体对齐过程的主要内容.对齐算法设计的主要思路是利用多种实体匹配技术结合知识库的特点和处理方法对知识库中指向相同对象的实例进行辨析以获得对齐结果.实体对齐算法可以分为只考虑实例及其属性相似程度的成对实体对齐和在成对对齐基础上考虑不同实例之间相互关系用以计算相似度的集体实体对齐 2 类,2 类算法的配合使用是解决知识库实体对齐问题的主要内容.而对实体属性及相互关系相似程度的衡量需要用到多种类型的相似函数,称为基于相似性函数的特征匹配.相似性函数同样可以分为 2 类:1)可以用于实体匹配中属性的相似性比较,即常用的文本相似性函数;2)用于实体匹配中的实体关系比较,称为结构相似性函数.

高效的实体对齐算法需要有效解决 1.3 节提出的三大挑战,尤其是大规模知识库的匹配效率问题被认为是当前实体对齐的最大挑战之一,除了选择合适而高效的相似性函数之外,分区索引技术被广泛地应用于实体对齐过程中.可以说,分区索引技术是当今大规模知识库匹配的关键技术,通过高效的索引设计可以避免随数据库规模二次增长的计算复杂度,是解决知识库对齐效率问题的有效手段.

知识库对齐的详细流程如图 3 所示.待对齐的知识库经过数据预处理阶段进入实体对齐算法模块,算法首先对待对齐数据进行分区索引,降低计算复杂度,然后利用文本相似性函数进行成对匹配,再通过结构相似性函数或其他一些利用关系相似性的算法进行集体匹配,最后将 2 方面结果结合起来形成最终对齐结果.第 3~5 节将对实体对齐算法以及

可用于对齐算法的基于相似性函数的特征匹配技术和分区索引技术的内容进行详细阐述.

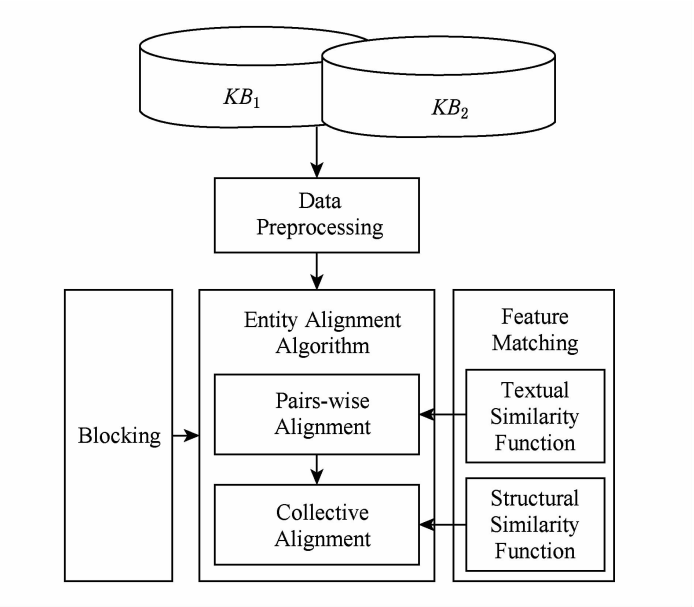


Fig. 3　Process of knowledge base alignment in detail.

图 3　知识库对齐详细流程

表 1 对本文中使用的符号进行了汇总,以方便查阅.

Table 1　Summary of Notations

表 1　符号表

Notation	Description
KB	Knowledge Base
$Align_{\text{entity}}(KB_1, KB_2)$	Alignment of Two Knowledge Bases
w_i	Weight of Component
$tokenize()$	Tokenize a String
$Attr(e)$	Attributes set of an Entity
$NB(e)$	Neighbors set of an Entity
$sim_{Attr}(e_1, e_2)$	Attribute Similarity of Two Entities
$sim_{NB}(e_1, e_2)$	Structure(neighbor) Similarity of Two Entities
$sim_{Attr}^{\text{sum}}(e_1, e_2)$	Sum of Attribute Similarity of Two Entities
$sim_{\text{name}}(e_1, e_2)$	Name Similarity of Two Entities
$sim_{\text{doc}}(e_1, e_2)$	Document Similarity of Two Entities
$sim_{\text{Jaccard}}(s_1, s_2)$	Jaccard Similarity of Two Strings
$sim_{\text{Consine}}(s_1, s_2)$	Cosine Similarity of Two Strings
$sim_{\text{Jaro}}(s_1, s_2)$	Jaro Similarity of Two Strings
$sim_{\text{JaroWinkler}}(s_1, s_2)$	JaroWinkler Similarity of Two Strings
$sim_{\text{token}}(t_i, t_j)$	Similarity of Two Tokens
$sim_{\text{ExtJaccard}}(s_1, s_2)$	ExtJaccard Similarity of Two Strings
$sim_{\text{MongeElkan}}(s_1, s_2)$	MongeElkan Similarity of Two Strings
$sim_{\text{softTFIDF}}(s_1, s_2)$	Soft TF/IDF Similarity of Two strings
$sim_{\text{CommonNB}}(e_i, e_j)$	Common Neighbor Similarity of Two Entities
$sim_{\text{JaccardCoeff}}(e_i, e_j)$	Jaccard Coefficient Similarity of Two Entities
$sim_{\text{Adar}}(e_i, e_j)$	Adar Similarity of Two Entities
$sim_{\text{KatzScore}}(e_1, e_2)$	Katz Score Similarity of Two Entities
$sim_{\text{SimRank}}(e_1, e_2)$	SimRank Similarity of Two Entities
x^*	Comparison Vector
$\boldsymbol{x}_{\text{name}}^*(e)$	Name Comparison Vector of Entity
$\boldsymbol{x}_{\text{doc}}^*(e)$	Document Comparison Vector of Entity
$\boldsymbol{x}_{Attr}^*(e)$	Attribute Comparison Vector of Entity
$\boldsymbol{x}_{NB}^*(e)$	Neighbor Comparison Vector of Entity
N_{TB}^{U}	The Number of Candidate Pairs Generated by the Traditional Blocking Which the Frequencies of the Blocking Key Values Follow Uniform Distribution
N_{TB}^{Z}	Traditional Blocking, Zipf's Law
$N_{\text{SNA}}^{\text{U}}$	Array Based Sorted Neighbor Blocking, Uniform Distribution
$N_{\text{SNA}}^{\text{Z}}$	Array Based Sorted Neighbor Blocking, Zipf's Law
$N_{\text{SNII}}^{\text{U}}$	Invert Index Based Sorted Neighbor Blocking, Uniform Distribution
$N_{\text{SNII}}^{\text{Z}}$	Invert Index Based Sorted Neighbor Blocking, Zipf's Law
$N_{\text{CCT}}^{\text{U}}$	Threshold Based Canopy Clustering Blocking, Uniform Distribution
$N_{\text{CCT}}^{\text{Z}}$	Threshold Based Canopy Clustering Blocking, Zipf's Law
$N_{\text{CCN}}^{\text{U}}$	Nearest Neighbor Based Canopy Clustering Blocking, Uniform Distribution
$N_{\text{CCN}}^{\text{Z}}$	Nearest Neighbor Based Canopy Clustering Blocking, Zipf's Law

2.1　数据预处理

数据的预处理也称作数据的标准化,是实体匹配任务的重要步骤.实体匹配中数据的质量问题包括数据的准确性、完整性、一致性、有效性、适时性和可获取性等方面问题,主要来源于数据的多源异构性、数据定义的不一致性、数据表达的多样性、数据的易变性等多个方面.数据的预处理需要综合考虑这些方面,设计出完善的解决方案,为数据的进一步处理打下良好的基础.在知识库实体对齐过程中数据的预处理同样重要,但由于知识库是预先按照一定规则建立好的数据库,数据的质量有一定的保证,很多实体对齐算法在对数据做简单的格式的整理或者停用词的处理后直接进行对齐,数据质量问题和容错处理可以在实体对齐算法过程中进行.因此,在很多情况下知识库实体对齐算法之前的数据预处理步骤的重要性并没有像在数据挖掘等数据处理任务中那样突出.随着数据清洗与整合技术的发展,数据的预处理方面产生了大量的研究成果可以借鉴到实体对齐的研究中,具体内容可以参见文献[18-19],

Christen 在专著文献[20]中也专门针对实体匹配的数据预处理问题作了介绍,本文不再赘述.

2.2 实体对齐算法

本文将在第 3 节对知识库实体对齐算法作详细介绍.知识库实体对齐算法可分为成对实体对齐和集体实体对齐 2 类.所谓成对实体对齐,即将实体对齐问题看作是根据属性相似性评分判断待匹配实体对匹配与否的分类问题.Newcombe[21] 与 Fellegi 和 Sunter[22] 给出了这种实体对齐分类方法的概率模型,用于实体对齐的大部分机器学习算法也属于成对实体对齐,这部分内容将在 3.1 节进行详细阐述.集体实体对齐又可以分为局部集体实体对齐和全局集体实体对齐.局部集体实体对齐也可称为基于简单关系的集体实体对齐,在计算实体相似性的时候将实体的关联实体属性纳入计算,即考虑待匹配实体对的邻居的属性集合,但并不将邻居节点当作平等的实体去计算结构相似度,这部分内容在 3.2 节进行概述.全局集体实体对齐基于实体对齐是相互影响的观察,通过不同匹配决策之间的相互影响调整实体之间的相似度,可以分为相似性传播和基于概率模型的集体实体对齐方法.这类方法是当前知识库实体对齐的主流算法,也是本文的重点内容,将在 3.3 节作具体阐述.

2.3 基于相似性函数的特征匹配

本文将在第 4 节对知识库实体对齐中的基于相似性函数的特征匹配作详细介绍.很多实体对齐算法需要用到相似性度量函数.一般在实际知识库实体对齐过程中 2 个实体 e_1 和 e_2 的相似性函数定义为

$sim(e_1,e_2)=(1-\alpha)sim_{Attr}(e_1,e_2)+\alpha sim_{NB}(e_1,e_2)$,

其中,$sim_{Attr}(e_1,e_2)$为对应于实体对的属性相似性函数,$sim_{NB}(e_1,e_2)$为应于实体对的结构相似性函数,$0\leqslant\alpha\leqslant1$ 为二者的调节参数.

本文将在 4.1 节对常用的文本相似性函数进行介绍.首先介绍基于 token 的相似性函数,这种方法将待匹配的实体对看作是一系列 token 的集合;其次介绍基于编辑距离的相似性函数,这种方法将待匹配的实体对作为文本字符串整体处理;最后介绍结合两者优点的混合型相似性函数.

结构相似性中的关系匹配主要是针对实体对的邻居节点的相似性,计算的基本思想是 2 个实体具有的相似节点越多,则它们越可能相似.本文 4.2 节将对常用的结构相似性函数进行介绍.

2.4 分区索引技术

本文将在第 5 节对知识库实体对齐中的分区索引技术作详细介绍.我们知道,索引是对数据库表中一列或多列的值进行排序的一种结构,使用索引可快速访问数据库表中的特定信息.在知识库对齐中建立索引是通过剪枝过滤掉知识库中不可能相似的实体对,使得相似的实体对尽量分配到一个或几个区块中成为候选对,最终的对齐处理只在这些候选对中进行,从而达到提高匹配效率的目的.

这其中的一个关键问题就是索引键值的选择问题.这里所谓索引键值就是知识库中实体集合的一个或几个属性的函数,通过这些函数值来划分待匹配实体集合,使得这些区块可以包含所有的匹配实体对,并且产生的候选对越少越好.索引键值的选择需要考虑 3 方面因素[20]:

1) 属性值的质量.因为任何作为索引键的属性值的缺失或错误都可能导致实体的错误分类,从而影响对齐的结果,因此作为索引键值的属性值要尽可能完整且正确.

2) 属性值的分布.在实体数量一定的条件下,偏斜的属性值的分布会导致部分分区中匹配对远大于其他分区,从而使匹配总数增加,而均匀分布的属性值产生的匹配对最少.因此属性值的分布要尽可能地均匀.

3) 区块数量和大小的权衡.通过索引产生相对少量较大的分区可以减少潜在匹配实体对的丢失概率,但会产生较多的候选对;而大量较小的分区虽然能减少候选对的数量,但却可能丢失更多的潜在匹配实体对.因此需要设计一种在尽量不丢失可能匹配的情况下使分区尽可能小的索引方案.

本文将可用于知识库中的分区索引技术分为 5 类:1)基本的分区索引,即按照所选择属性的索引键值直接构建分区;2)基于滑动窗口的分区索引,即所谓近邻排序索引;3)基于相似性的分区索引,包括基于 q-gram、基于后缀数组和基于 Hash 的索引;4)基于聚类的索引,包括 Canopy 聚类索引和基于 StringMap 的索引;5)动态索引,它所介绍的分区索引技术中唯一可以同时作用于属性和关系的索引技术,具体内容将在第 5 节中介绍.

3 实体对齐算法

3.1 成对实体对齐方法

3.1.1 传统概率模型的实体对齐方法

传统概率模型对齐方法是一种基于属性相似性的成对比较的方法,这种方法不考虑匹配实体间的关系,Newcombe[21] 与 Fellegi 和 Sunter[22] 通过将

基于属性相似性评分的实体匹配问题转化为分类问题(分为匹配、可能匹配和不匹配)建立了这个问题的概率模型.这种模型是实体对齐问题的重要方法,迄今为止有大量的实体对齐方面的工作建立在这种方法之上.

一种直观的实体对齐分类方法是加和所有匹配属性的相似度评分,然后设置2个相似度阈值,判断总的相似度评分 $sim_{Attr}^{\text{sum}}()$ 位于哪个相似度区间,可以形式化表示为

$$\begin{cases} sim_{Attr}^{\text{sum}}(e_1,e_2) \geqslant t_1 \Rightarrow e_1,\ e_2 \text{ 匹配};\\ t_1 \leqslant sim_{Attr}^{\text{sum}}(e_1,e_2) \leqslant t_2 \Rightarrow e_1,\ e_2 \text{ 可能匹配};\\ sim_{Attr}^{\text{sum}}(e_1,e_2) \leqslant t_2 \Rightarrow e_1,\ e_2 \text{ 不匹配}. \end{cases}$$

其中,e_1,e_2 为待匹配的实体对;t_1,t_2 为相似度阈值的下界和上界.这种简单方法的最主要问题在于没有体现不同属性对于最终相似度的影响.一个重要的解决方案是为每个匹配的属性对分配不同的权重,以体现其对对齐结果的重要性.Herzog 在 Fellegi-Sunter 模型的基础上通过建立基于概率的实体链接模型形式化地描述了这一思想[23]:定义2个待匹配的数据库(同样适用于知识库)A 和 B,e_i 和 e_j 分别为 A 和 B 中的实体,$e_i \in A$,$e_j \in B$;定义2个不相交的集合 M 和 U:

$$M = \{(e_i,e_j) \mid e_i = e_j, e_i \in A, e_j \in B\};$$
$$U = \{(e_i,e_j) \mid e_i \neq e_j, e_i \in A, e_j \in B\}.$$

定义比较向量 $\boldsymbol{x}^*$ 为待匹配实体所有匹配的属性形成的向量,比较空间 X 为所有可能的 $\boldsymbol{x}^*$ 形成的空间;定义2个条件概率的比值 R:$R = P(\boldsymbol{x}^* \in X|M)/P(\boldsymbol{x}^* \in X|U)$,则 Fellegi 与 Sunter 的匹配决策规则可表述为

$$\begin{cases} R \geqslant t_1 \Rightarrow e_i,\ e_j \text{ 匹配};\\ t_1 \leqslant R \leqslant t_2 \Rightarrow e_i,\ e_j \text{ 可能匹配};\\ R \leqslant t_2 \Rightarrow e_i,\ e_j \text{ 不匹配}. \end{cases}$$

在假设比较向量 $\boldsymbol{x}^*$ 中的属性彼此独立的条件下(实际中实体属性很可能相关,但 Herzog 的实验[23]表明这个假设仍然可以取得很好的匹配效果),属性的权重为

$$\begin{cases} m_i = P((a_i = b_i, a_i \in A, b_i \in B) \mid M);\\ u_i = P((a_i = b_i, a_i \in A, b_i \in B) \mid U). \end{cases}$$

其中,a_i 和 b_i 为待匹配实体对的第 i 个属性,m_i 为假设2个实体相同其第 i 个属性值相等的概率,u_i 为假设2个实体不相同其第 i 个属性值相等的概率.基于这2个概率值,可以计算第 i 个属性的权重 w_i 为

$$w_i = \begin{cases} \text{lb}(m_i/u_i),\ a_i = b_i;\\ \text{lb}((1-m_i)/(1-u_i)),\ a_i \neq b_i. \end{cases}$$

设比较向量 $\boldsymbol{x}^* = (x_1, x_2, \cdots, x_n)$,比较空间 $X = X_1 \times X_2 \times \cdots \times X_n$,在条件独立性的假设下,

$$\text{lb}\, R = \text{lb}\left\{ \prod_{i=1}^{n} \frac{P(x_i \in X_i \mid M)}{P(x_i \in X_i \mid U)} \right\} = \sum_{i=1}^{n} \text{lb}\left\{ \frac{P(x_i \in X_i \mid M)}{P(x_i \in X_i \mid U)} \right\} = \sum_{i=1}^{n} w_i.$$

可以看到,实体对匹配程度的大小等于各属性权重之和.因此,通过全部匹配属性的权重之和可以对实体匹配程度进行判断.

有很多工作建立在 Fellegi 与 Sunter 的研究基础之上.Porter 和 Winkler 在文献[24-25]中使用属性值的近似比较代替相等与否的二值比较,取得了较好的匹配质量;Winkler 等人在文献[26]中将待匹配属性值出现的频率代入到 m_i 和 u_i 的计算当中,他们认为出现频度较高的属性对实体对的匹配贡献越低,需要降低其权重;Winkler 还结合贝叶斯网络对属性的相关性建模,并使用最大估计算法对参数进行估计[27].在传统概率模型对齐方法中会产生2类错误:1)相同实体被分类为不相等;2)不同实体被分类为相等.通常假定产生这2类错误的代价相等,但实际上经常会有不相等的情况,文献[28]正是基于这种观察,提出一种基于代价优化的决策模型,在传统模型的基础上将不同的代价赋予不同的匹配状态,通过一个总体代价公式和贝叶斯公式产生一个代价最优化决策规则.

3.1.2　基于机器学习的实体对齐方法

基于 Fellegi-Sunter 模型的概率实体对齐方法取得了大量的研究成果,随着机器学习及统计学习的发展,很多机器学习方法也应用到实体对齐领域,并取得了巨大的进展.机器学习方法将实体对齐问题看作是二元分类问题,根据是否使用标注数据可以分为有监督学习和无监督学习2类,而主动学习属于有监督学习,通过不断交互获得更加准确的训练数据.下面对这3类基于机器学习的实体对齐方法进行介绍,由于很多基于概率模型的方法也使用了大量机器学习的技术,其相关内容在这里只作概述,详细内容参见3.3.2节的具体算法.

3.1.2.1　监督和半监督机器学习

监督机器学习分类算法需要预先标注部分实体匹配与否作为训练数据,使用训练数据的比较向量作为特征向量代入进行计算去训练分类模型,然后使用训练好的模型对未标注的数据进行分类.一个典型的监督机器学习的分类方法有3个步骤[29]:1)选择合适的分类技术和分类模型,使用训练数据对模型进行训练,并通过自动或手工的方式调节其中

的参数;2)使用同样格式的测试数据对训练出来的模型进行评估,如果评估结果达不到要求则需要调节参数或者修改模型,同时要注意测试数据的过拟合问题;3)将测试好的模型应用于实际数据进行分类.

基于监督学习的实体对齐方法可以分为2类:

1) 只通过属性比较向量判断一个实体对的匹配与否,属于成对实体对齐.这类方法的主流技术有决策树[29]、支持向量机(SVM)[30]和集成学习[31]等.决策树方法是通过训练数据迭代生成一个规则树,其内部节点为判断规则,叶子节点为规则的可能分类结果,通过这个生成的决策树就可以判断实体对是否相似.Cochinwala等人在文献[32]中使用著名的分类回归树(CART)算法、线性分析判别算法和矢量量化方法进行实体辨析;Elfeky等人在TAILOR工具包中实现了一种ID3决策树算法[33],并通过实验证明其算法的匹配效果要高于传统的概率模型方法.支持向量机方法是通过训练数据集在高维空间中产生一个分类超平面使得匹配和不匹配2类数据集的间距尽可能大,并以此分类真实数据.Bilenko等人使用SVM方法获得文本的基于向量空间模型的相似度估计,并在MARLIN系统中予以实现,实验表明相对于决策树模型SVM可以取得更好的分类效果.Christen在2阶段实体链接分类模型中使用一种迭代的SVM分类算法[34],实验表明其匹配效果远高于TAILOR中的混合算法.实体对齐中的集成学习是指为了提高对齐质量将多种基本的实体对齐系统结合起来形成一个单独的解决方法.Chen等人在文献[35]中提出了一种新的集成学习框架,使用2种监督学习将多种基础实体对齐系统和上下文特征结合起来,形成统一的聚类决策模型,实验表明相比于其他基本匹配方法这种集成学习框架可以取得更高的匹配质量.

2) 基于聚类的方法,如果只考虑属性相似性,则仍属于成对对齐的范畴.其主要思想是通过训练数据来学习如何更好地将相似的实体聚类到一起.Cohen和Richman提出了一种可扩展的自适应的实体名称匹配和聚类技术[36],通过训练样本生成一个自适应的距离函数,实验表明通过训练这种自适应技术可以有效地提高模型的准确率.类似地,McCallum和Wellner在条件随机场实体对齐模型中使用监督学习的方法训练产生距离函数度量,通过在训练集中调整权重参数去最大化特征函数和学习参数之积,使得相似的实体聚类,从而在无向图中产生正确的分区[37].Pasula等人使用半监督机器学习算法和基于图的概率关系模型进行共指关系的发现[38],这些内容属于集体实体对齐,将在基于概率模型的集体对齐方法中作详细阐述.

3.1.2.2 主动学习

监督学习一个主要的瓶颈在于很难获得足够的训练数据进行分类预测,主动学习通过与人的交互解决这个问题.基本思想是通过初始少量的训练数据集和设计高效的人机交互算法迭代式地训练分类模型,不断提升分类效果.具体过程如下:1)通过初始训练数据集训练一个分类模型;2)分类所有的比较向量;3)将难于有效分类的候选对按照一定算法选出并询问用户进行人工分类;4)将人工分类后的比较向量加入训练数据集;5)通过训练产生更好的分类模型;6)重复这个过程直到达到指定的停止标准或者最大迭代次数,获得最终的分类模型.图4为这一过程的示意图.

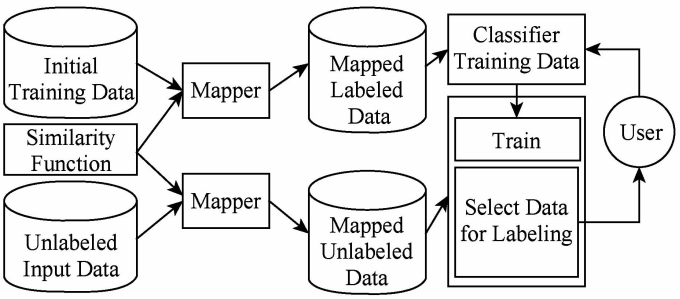


Fig. 4 Diagram of human-computer interaction entity alignment process based on active learning.

图4 主动学习人机交互实体对齐过程示意图

目前有很多基于主动学习的数据匹配研究.Sarawagi等人构建的ALIAS系统[39]基于人机交互完成实体记录链接和去重任务.系统通过3个分类树构建一种集成式的分类模型,对于那些在3个分类树模型下结果不同的比较向量则有用户手工指定,再将分类结果代入模型进行迭代以产生更好的分类结果.Tejada等人在文献[40]中基于同样的方法构建了Active Atlas系统.Arasu等人在以上研究的基础上提出了一种新颖的应用于数据匹配任务的主动学习算法[41],将索引计数与主动学习相结合以提高算法运行效率,多种分类算法都可以应用到这个模型中,由用户指定一个最小的准确率,主动学习过程则在这个准确率下获得尽可能高的召回率,同时尽可能减少需要人工分类的数据.实验表明在大型数据库匹配过程中这种方法明显优于ALIAS系统和Active Atlas系统.

3.1.2.3　无监督机器学习

在缺乏训练数据并且无法利用人工匹配的情况下,可以通过无监督的机器学习完成实体匹配任务.无监督机器学习的实体匹配的主要思想是利用聚类算法将比较向量相似的实体分配到相同的分类中,这种思想植根于 Fellegi-Sunter 概率模型分类方法.

Verykios 等人提出了一种基于聚类的"bootstrapping"技术学习匹配模型[42].其基本思想基于交互训练,利用非常少的已标记数据,采用无监督学习,根据比较向量的相似性聚类实体.所有在同一聚类中的实体对对应同一分类(匹配与否),通过少量的已标记数据的匹配情况去推理聚类中所有数据的匹配情况.Elfeky 等人利用这种思路开发了实体链接工具 TAILOR[33].Ravikumar 和 Cohen 在文献[43]中利用相似的思路提出一种无监督学习下的层次图模型框架进行实体记录的匹配,这种方法将组成比较向量的每个属性字段建模成一个表示属性值匹配与否的隐二元变量,将原有的产生式模型转换为层次化的 3 层模型,并在一定约束条件下进行推理实现实体记录的匹配.Bhattacharya 和 Getoor 在文献[44]中采用基于隐狄利克雷分配(latent dirichlet allocation, LDA)模型的产生式模型进行实体匹配的工作,这部分内容属于集体实体对齐,将在 3.3.2 节中作详细阐述.

3.2　局部集体对齐方法

知识库中实体之间的关系对于实体对齐具有重要意义,可以有效地提高匹配的准确率和召回率,关系相似性的作用在某些情况下可以超过属性相似性.基于简单关系的局部集体对齐方法,将实体本身属性(其相似度以 $sim_{Attr}()$ 表示)和与之有关联实体(这里指其邻居实体,其相似度以 $sim_{NB}()$ 表示)的属性分别分配不同的权重,并将其加权求和计算总体相似度,可形式化表达为

$$sim(e_1,e_2)=\alpha sim_{Attr}(e_1,e_2)+(1-\alpha)sim_{NB}(e_1,e_2).$$

其中:

$$sim_{Attr}(e_1,e_2)=\sum_{(a_1,a_2)\in Attr(e_1,e_2)} sim(a_1,a_2);$$

$$sim_{NB}(e_1,e_2)=\sum_{(e_1',e_2')\in NB(e_1,e_2)} sim_{Attr}(e_1',e_2').$$

一种典型的局部集体对齐算法是使用向量空间模型和余弦相似度计算大规模知识库中的实体相似性[45].算法为每个实体 e 建立 2 个向量:名称向量 $\boldsymbol{x}^*_{\text{name}}(e)$ 和虚拟文档向量 $\boldsymbol{x}^*_{\text{doc}}(e)$.名称向量简单地说来自于其标识属性;而虚拟文档向量则来自于其他属性值和其邻居节点的属性值的加权求和,可形式化定义为

$$\boldsymbol{x}^*_{\text{doc}}(e)=\boldsymbol{x}^*_{Attr}(e)+\gamma\cdot\boldsymbol{x}^*_{NB}(e),$$

其中,γ 为权重因子,$\boldsymbol{x}^*_{Attr}(e)$ 为标识属性之外的其他属性,$\boldsymbol{x}^*_{NB}(e)$ 为邻居节点的属性,其定义为

$$\boldsymbol{x}^*_{NB}(e)=\sum_{e'\in NB(e)}(\boldsymbol{x}^*_{Attr}(e')+\gamma\cdot\boldsymbol{x}^*_{NB}(e')).$$

为了评价向量中每个分量的重要性,算法使用 TF-IDF 为其赋予权重,并为每个向量建立倒排表,通过简单的过滤剪枝生成候选对,最后使用余弦相似性函数计算每个候选对的 2 个向量的相似性 $sim_{\text{name}}(e_1,e_2)$ 和 $sim_{\text{doc}}(e_1,e_2)$,并输出最终结果:

$$sim(e_1,e_2)=\{w_n\times sim^2_{\text{name}}(e_1,e_2)+(1-w_n)\times sim^2_{\text{doc}}(e_1,e_2)\}^{1/2},$$

其中,w_n 为名称向量的相似性权重.

实验证明,这种方法具有较好的召回率和较快的运行速度,可以应用到大规模数据集中,但其准确性有待进一步提高.究其根本这种方法仍属于成对的实体匹配方法,只是将实体的关系看作是实体的一类特殊属性代入计算,并没有真正地实现"collective"的方式.

3.3　全局集体对齐方法

3.3.1　基于相似性传播集体对齐方法

基于相似性传播的方法是一种真正实现了集体方式的实体对齐方法.它通过初始匹配以"bootstrapping"方式迭代地产生新的匹配[46-47].举例来讲,如果 2 个作者匹配,则与这 2 个作者具有"coauthor"关系的另外 2 个相似名字的作者会有较高的相似度,而这个相似度又会对其他作者匹配产生影响.在这种方法中,实体之间的相似度会随着算法的迭代不断变化,直到算法收敛或达到指定的停止条件.

一种典型的基于相似性传播的方法是由 Bhattacharya 等人在 2007 年提出的集合关系聚类算法[47],采用一种改进的层次凝聚算法迭代地产生匹配结果.Lacoste-Julien 等人在此基础上进一步提出了适合大规模知识库实体对齐的 SiGMa 算法[12],算法将大规模知识库实体对齐问题看作一个全局匹配评分目标函数的贪婪优化问题,这个全局函数由实体对的属性和图邻域信息加权求和构成.定义 2 个知识库的实体集合 e_1 和 e_2,使用矩阵 $\boldsymbol{y}^*$ 定义 e_1 到 e_2 的匹配 m,当 $m(i)=j$ 时 $y^*_{ij}=1$,反之 $y^*_{ij}=0$,可能的匹配空间可以表示为一组二元矩阵:

$$\boldsymbol{M}\doteq\sum_{(i,j)\in e_1\times e_2} y^*_{ij}[(1-\alpha)sim_{Attr}(i,j)+\alpha sim_{NB}(i,j)],$$

其中,$sim_{Attr}(i,j)$ 定义为实体对 (i,j) 的属性相似性;$sim_{NB}(i,j)=\sum_{(k,l)\in N_{ij}} y^*_{kl}\,w_{ij,kl}$,$N_{ij}$ 为 (i,j) 的邻居节点,二项系数 $w_{ij,kl}$ 定义为 (k,l) 匹配后 (i,j)

匹配的可能性,简单来讲,$sim_{NB}(i,j)$可以看作(i,j)的所有邻居节点数量的加权求和,则问题可以形式化表示为

$$\max_{\boldsymbol{y}^*} obj(\boldsymbol{y}^*),$$
$$\text{s.t. } \boldsymbol{y}^* \in \boldsymbol{M}, \|\boldsymbol{y}^*\|_1 \leqslant R.$$

其中,$\|\boldsymbol{y}^*\|_1 \doteq \sum_{ij} y_{ij}^*$ 表示所有匹配的数量,R 为其上界,$obj(\boldsymbol{y}^*)$为全局目标函数. 通过这种方式将原问题转化为一个运筹学二次分配问题,可以通过一种贪婪优化方法进行近似求解. 算法的总体流程为:1)初始化匹配 $\boldsymbol{y}_0^*$,在缺少 sameAs 或 URI 的情况下可以采用名称的完全匹配产生;2)在每轮迭代过程中,选择使目标函数最大化的实体对(i,j),通过设置$y_{ij}^*=1$,将(i,j)加入匹配,每次只选择匹配实体对的邻居节点做为候选对,避免二次匹配问题;3)当$\|\boldsymbol{y}^*\|_1=R$ 时算法终止.

这种方式综合利用了实体对的属性和关系,通过初始匹配在邻域图上不断迭代完成所有匹配对的发现过程,具有较好的准确性和可扩展性. 实验表明可以在关系数量有限的大型知识库中取得较好的匹配效果,但需要一定的人工参与.

3.3.2 基于概率模型的集体对齐方法

基于概率模型的集体对齐方法通过为实体匹配关系和匹配决策建立复杂的概率模型来避免这种“bootstrapping”方式的问题. 举例来讲,我们不需要知道任何 2 个作者的匹配关系,只要知道 2 个作者分别和另外 2 个作者互为“coauthor”,那么我们就认为他们在一定条件下有可能匹配. 基于概率模型的集体对齐方法一般采用统计关系学习进行计算和推理,通过集成关系或逻辑表示、概率推理、不确定性处理、机器学习和数据挖掘等方法,以获取关系数据中的似然模型. 常用的概率模型有关系贝叶斯网络模型[38,48]、LDA 分配模型[44,49]、条件随机场(conditional random field, CRF)模型[37,50]和 Markov 逻辑网(Markov logic networks, MLNs)模型[51-52]等,概率模型提供了一种标准的形式化关系建模方式,可以有效提高匹配效果,但是应用到大规模知识库上的效率问题是一个严重的瓶颈. 这里简单介绍 4 种常用概率模型在知识库实体对齐中的应用,并介绍一种匹配大规模知识库的全局概率算法[53].

3.3.2.1 基于 LDA 模型的实体对齐

LDA 模型是一种用于离散数据建模的生成式模型,能够简化大规模数据集中数据的表示,并保留对数据的相关性、相似性或聚类等分析的基本信息. 此模型一般用于文档语料库的主题发现,一般也称为主题模型. Bhattacharya 等人将 LDA 模型用于实体解析中,通过一个隐藏的群组变量获取实体之间的关系模型[44]. 以文献[44]中的作者辨析问题为例,设文档集合 D 和相应的作者集合 A,D 中作者实体引用的集合$\{a_1,a_2,\cdots,a_R\}$,作者 i 的引用为 a_i,其所写文档为 d_i,群组表示合著作者(coauthor)的集合,设 T 为不同群组个数,每个作者 a_i 关联的群组设为 z_i. 模型中每个群组表示为实体参数为 ϕ^j 的多项式分布,从群组 j 随机抽样得到实体 i 的概率 $P(a=i|z=j)=\phi_i^j$,每个文档可以表示为群组参数为 θ^d 的多项式分布,$P(z=j)=\theta_j^d$. 每个 θ^d 是从以 α 为超参数的 Dirichlet 分布中取得,每个 ϕ^j 是从以 β 为超参数的 Dirichlet 分布中取得,模型的贝叶斯网络图如图 5(a)所示.

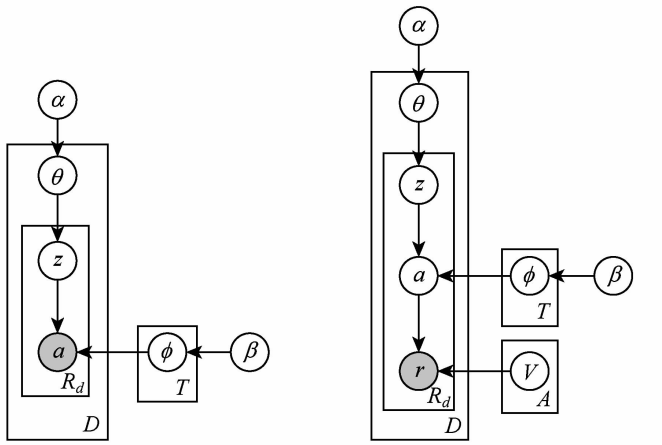


Fig. 5 Comparison of Two LDA models.

图 5 2 种 LDA 模型的比较

为了解决实体辨析问题,为每个实体引入属性 v_a,通过概率修正实体属性集合 V_a 来获得实体的引用 r. 从一个实体产生一个引用的概率为 $P(r|v_a)$,其中 r 是可观测的,实体 a 和群组 z 为隐含变量,修改后的模型如图 5(b)所示.

在给定 α,β 和 V 的条件下,文档集的所有实体引用的生成概率为

$$P(r;\alpha,\beta,V)=\prod_d P(r_d;\alpha,\beta,V)=$$
$$\prod_d \sum_{a_d} P(r_d \mid a_d;V)P(a_d;\alpha,\beta)=$$
$$\int_\varphi P(\varphi;\beta)\prod_d \sum_{a_d} P(r_d \mid a_d;V)\int_\theta P(\theta;\alpha)P(a_d;\theta,\varphi)\mathrm{d}\theta\,\mathrm{d}\varphi.$$

模型采用无监督学习的方法结合吉布斯采样进行集合实体解析. 为了提高 Dirichlet 过程的推理效率,Bhattacharya 等人在文中提出了一种聚类分块

算法,通过适时改变实体的标签,达到实体引用凝聚聚类的效果.LDA 实体解析模型可以看作是考虑实体间关系的 Dirichlet 过程混合模型的扩展,在真实世界数据集上的实验表明这种方法还有待进一步的研究和提高.

3.3.2.2　基于 CRF 模型的实体对齐

以 LDA 为代表的产生式模型具有广泛的应用途径,但其本身也具有很多不足之处:为了定义联合概率,必须列出所有可能的观察序列,实际中列出观察序列的交互特征或长距离约束是不容易实现的,并且产生式模型很难使用各种高度重叠的、非依赖的、不同粒度的特征.因此,在一些相似性计算的问题上,产生了许多使用条件模型代替产生式模型的成功案例.

McCallum 等人提出了一种基于图划分技术的 CRF 实体辨析模型[37],这个模型以观察值为条件产生实体辨析的决策.文中提出 3 种处理这种不确定性的条件模型,都属于条件训练的无向图模型,这种模型善于获取属性间的因果关系不是很明显的具有相互依赖关系的数据.设 E 为实体的集合,X 为观测随机变量的集合,Y 为每个实体唯一的整型标识符随机变量的集合,A 为实体的所有属性集合,则实体辨析问题可以表述为给定一组观察及其上下文信息 X,寻找最可能的实体标识的集合 Y 和属性集合 A,即条件概率模型的实体辨析定义为通过训练去最大化 $P(Y,A|X)$.

模型 1 定义了条件随机场模型进行实体辨析的最一般形式,将所有实体的观察值、标识值和属性都看作是无向图的顶点,边表明相连的节点具有依赖关系,模型中参数的捆绑模式定义为模板 T,每一个图中模版的实例定义为一个命中 H.根据 Hammersley-Clifford 定理,将条件概率以指数形式表示为

$$P(y,a|x)=\frac{1}{Z_x}\exp\Big(\sum_{t\in T}\sum_{h_t\in H_t}\sum_{l}\lambda_l f_l(y,a,x{:}h_t)\Big),$$

其中,势函数定义为特征函数 f 与学习参数 λ 的点积,$(y,a,x{:}h_t)$ 为一个模版命中选定的 (Y,A,X) 的子集,Z_x 为归一化函数使得 y 的概率分布之和为 1.

为了避免在推理过程中模型的结构变化,模型 2 在模型 1 的基础上去掉了依赖实体数量的部分,使用指示 2 个观察值 (X_i,X_j) 是否指向同一实体的二值随机变量 Y_{ij} 取代实体标识节点 Y_i,同时引入附加的势函数 $\lambda_{l'}\times f_{l'}(y_{ij},y_{jk},y_{ik})$ 去除不一致的决策,将不一致部分的学习参数 $\lambda_{l'}$ 置为负值使得此部分为 0 概率,公式可以形式化表示为

$$P(y,a\mid x)=\frac{1}{Z_x}\exp\Big(\sum_{i,j,l}\lambda_l f_l(x_i,x_j,y_{ij},x_i.a,x_j.a)+\sum_{i,j,k,l'}\lambda_{l'}\times f_{l'}(y_{ij},y_{jk},y_{ik})\Big).$$

当模型 2 的属性部分在实体对齐中不是必须时,我们可以将属性部分去掉以进一步简化模型,得到一个更加直接、表达能力更强的无向图模型——模型 3:

$$P(y\mid x)=\frac{1}{Z_x}\exp\Big(\sum_{i,j,l}\lambda_l f_l(x_i,x_j,y_{ij})+\sum_{i,j,k,l'}\lambda_{l'}\times f_{l'}(y_{ij},y_{jk},y_{ik})\Big).$$

由于图划分算法与某种类型的无向图模型上的推理具有等价性,因此寻找最大概率的实体解析方案可以对应于以实体为顶点、以势函数 $\sum_l \lambda_l f_l(x_i,x_j,y_{ij})$ 为边权重的图的划分,定义 $f_l(x_i,x_j,1)=-f_l(x_i,x_j,0)$.具有负权重边的图划分是一个 NP 难问题,可以采用最小分歧关联聚类算法[54]来实现.在参数学习上,可以通过最大似然来求得,即通过调整参数 λ 使得模型公式在所有训练数据实例上的积最大化,由于目标函数是凸函数,可以通过其对数似然的导数求得,可以采用投票感知器(voted perceptron, VP)形式的随机梯度上升算法进行近似计算[55].实验证明,模型 3 较以往手段可以有效地提高实体解析的准确度.

Domingos 等人在文献[37]的基础上提出了一种基于条件随机场模型的多关系的实体链接算法[50],并详细阐述了公共属性在多关系实体链接中的作用.算法将实体属性作为顶点引入到无向图中,同时引入称为信息节点的二项节点,从而产生一个更加复杂的条件概率模型.模型的推理和参数的学习采用和文献[37]相同的方法,并针对大数据量引入基于 canopy 的索引(参见 5.4 节)来提高匹配效率.Wick 等人在文献[56]中提出一种基于条件随机场的判别式层次模型,算法迭代地将实体划分为树结构,观察值作为叶子结点,树的内部节点将叶子节点的信息逐层汇总,进行摘要,从而形成一个高度精简而信息丰富的结构进行高效推理.实验证明,在大规模数据集中,这种判别式层次模型相比于其他基于 CRF 的二元匹配模型快出几个数量级,为概率模型在大规模知识库的实体对齐上的应用提供了一个有效的解决方案.

3.3.2.3　基于 Markov 逻辑网的实体对齐

Markov 逻辑网是一种结合一阶谓词逻辑和

概率图模型的复杂性和不确定性问题表示和处理方法[57-58].从概率统计的角度来看,Markov 逻辑网不仅提供了一种描述 Markov 网的有效手段,还能够灵活地在 Markov 网中融入模块化知识域;同时,从一阶谓词逻辑的角度来看,Markov 逻辑网给一阶谓词逻辑加入了不确定性处理能力,并且能够允许知识域中存在不完整性和矛盾性等问题. Markov 逻辑网在机器学习和人工智能等诸多领域都有重要应用,也可以作为知识库实体对齐的重要手段.

Singla 和 Domingos 提出了一种将 Markov 逻辑网应用到实体解析中的方法[51],将其看作是传统的 Fellegi-Sunter 概率模型的一般化形式. 他们使用一阶谓词逻辑和 Markov 随机场,VP 算法[55]形成一个综合性的实体解析解决方案. 由于 Markov 随机场和条件随机场的内在关系(CRF 是给定观察集合下 MRF(Markov random field)的分布,也就是条件分布,具体内容见 3.3.2.2 节),我们直接给出 Markov 随机场的联合分布形式:

$$P(X=x)=\frac{1}{Z}\prod_k \varphi_k(x_{\{k\}}),$$

其中,$x_{\{k\}}$ 是图中第 k 个团的状态;φ_k 是势函数;Z 是归一化因子,也称为分区函数. 将势函数用特征函数加权求和的指数形式代替,可得其线性对数模式为

$$P(X=x)=\frac{1}{Z}\exp\Big(\sum_j w_j f_j(x)\Big),$$

其中,$f_j(x)\in\{0,1\}$.

基于一阶谓词逻辑的知识库可看作是在一个可能世界的集合上建立一系列硬性规则,其基本思想是让那些硬性规则有所松弛,即当一个世界违反了其中的一条规则时,这个世界存在的可能性将降低,但并非不可能. 一个世界违反的规则越少,这个世界存在的可能性就越大. 为此,给每个规则都加上了一个特定的权重,它反映了对满足该规则的可能世界的约束力. 则 Markov 逻辑网 L 可定义为一组二元项(F_i,w_i),其中,F_i 表示一阶谓词逻辑公式,w_i 是一个实数. 这组二元项(F_i,w_i)与一组有限常量集 $C=\{c_1,c_2,\cdots,c_{|C|}\}$一起定义了一个 Markov 网 $M_{L,C}$:

1) L 中的任意闭原子(ground atom)都对应了 $M_{L,C}$中的一个二值节点. 若此闭原子为真,则对应的二值节点取值为 1;若为假,则取值为 0.

2) L 中的任意闭规则都对应着一个特征值,若此闭规则为真,则对应的特征值为 1;若为假,则特征值为 0 这个特征值的权重为二元项中该规则 F_i 对应的权重 w_i.

由此可知,$M_{L,C}$的节点由 Markov 逻辑网 L 中每个闭原子生成,边由闭原子之间的关系生成. Markov 逻辑网可看作是一个用以构建 Markov 网的模板. 由此可得一个闭 Markov 网中所蕴含的可能世界 x 的概率分布为

$$P(X=x)=\frac{1}{Z}\exp\Big(\sum_{i=1}^{F} w_i n_i(x)\Big),$$

其中,F 表示逻辑网中公式的数量,$n_i(x)$表示关于规则的取值为真的对应闭规则 F_i 的个数.

最大可能性问题是概率图模型推理中的重要内容,基本过程可表述为:给定证据变量集 x,求变量集 y 最可能处于的状态,在 Markov 逻辑网中,可以转化为求以下最大化问题:

$$\max_y P(y\mid x)=\max_y \sum_i w_i n_i(x,y),$$

其中,w_i 表示子句的权重,$n_i(x,y)$表示子句的闭子句的真值数量. 计算最大可能性问题可转换为典型的最大化加权的可满足性问题,即寻找一组变量的赋值,使得所有被满足的子句的权重之和最大. 这类问题可以高效地通过类似 MaxWalkSAT 这类加权可满足性问题解决器来计算[59]. 条件概率可以通过最小闭 Markov 网的吉布斯采样来计算. Markov 逻辑网的学习是指在 Markov 逻辑网结构确定的前提下,学习和优化模型的参数. 一般采用最大似然方法,但计算一个规则在世界中的闭规则的个数是不明智的,因此在封闭世界假设的前提下,我们采用判别训练进行参数学习.

在 Markov 逻辑网上进行实体辨析,需要定义一系列等价谓词公理,通过这些等价性公理以及其他一些公式,我们就可以在 Markov 逻辑网上进行知识库的集体实体对齐.

为了解决大规模知识库上的实体对齐的效率问题,Rastogi 等人在 Markov 逻辑网的基础上提出了一个原则性的框架来扩展任何通用的实体匹配算法[52]. 这个框架内容主要包括:1)将实体匹配器建模成一个黑盒;2)匹配器的多重实例运行在实体的有限子集上;3)通过跨实例的消息传递机制控制匹配器的交互. 实验证明,大规模数据集上这个算法无论在准确率、召回率还是运行时间上都可以取得较好的效果.

3.3.2.4 基于全局概率算法的实体对齐

基于本体的大规模知识库的实体对齐有很多新

的挑战,其中一个重要方面就是本体知识库的结构相对复杂,其对齐需要分别考虑类别、属性以及实体和它们之间的相互关系,Suchanek 等人针对这个问题提出了一种新型的基于概率的全局算法 PARIS[53],算法在不需要任何参数调节的条件下能够高效地跨本体同时对齐类别、属性、关系和实例.

Suchanek 等人对实体对齐的全局概率模型进行了如下形式化的定义:

1) 定义了逆函数性 $fun^{-1}(r)$. 逆函数性的大小表明事实三元组的同一关系中宾语相等对主语相等的决定能力,关系的逆函数值越大,如果关系中宾语相等,则主语相等的可能性也越大.

2) 结合全局逆函数性定义实例匹配的概率公式. 在知识库对齐中 2 个实例相等的逻辑规则表述为

$$\exists r,y,y':r(x,y)\wedge r(x',y')\wedge y\equiv y'\wedge fun^{-1}(r)\text{ 较高}\Rightarrow x\equiv x'.$$

转换为概率公式表达:

$$Pr_1(x\equiv x')=1-\prod_{\substack{r(x,y)\\ r(x',y')}}(1-fun^{-1}(r)\times Pr(y\equiv y')).$$

如果考虑当宾语不相等且全局函数性较高则主语相似的概率应该较低,利用同样的概率过程,可以得到:

$$Pr_2(x\equiv x')=1-\prod_{r(x,y)}(1-fun^{-1}(r)\prod_{r(x',y')}(1-Pr(y\equiv y'))).$$

因此,结合这 2 方面考虑将 2 个概率公式相乘,可以得到最终的实例匹配概率公式:

$$Pr_3(x\equiv x')=Pr_1(x\equiv x')\times Pr_2(x\equiv x').$$

3) 定义的关系匹配的概率公式. 不同于实例相等,更加广泛的关系相等的判别应该是包含关系的判别,即 $Pr(r\subseteq r')$,简言之,就是一个本体知识库中的关系 r 所对应的主语宾语对同样在另一个本体知识库中的关系 r' 中,这样的实体对所占的比例:

$$Pr(r\subseteq r')=\frac{\#x,y:r(x,y)\wedge r'(x,y)}{\#x,y:r(x,y)}.$$

分子的值需要根据知识库中已有的匹配实体对来计算,可得:

$$\#x,y:r(x,y)\wedge(\exists x',y':x\equiv x'\wedge y\equiv y'\wedge r'(x',y)).$$

可转化为概率公式:

$$\sum_{r(x,y)}(1-\prod_{r'(x',y')}(1-(Pr(x\equiv x')\times Pr(y\equiv y')))).$$

同样地可以将分母转化为在另一个知识库中有对应实体的关系 r 所对应的实体对:

$$\sum_{r(x,y)}(1-\prod_{x',y'}(1-Pr(x\equiv x')\times Pr(y\equiv y'))).$$

$Pr(r\subseteq r')$最终的表达形式为

$$\frac{\sum_{r(x,y)}(1-\prod_{r'(x',y')}(1-(Pr(x\equiv x')\times Pr(y\equiv y'))))}{\sum_{r(x,y)}(1-\prod_{x',y'}(1-Pr(x\equiv x')\times Pr(y\equiv y')))}.$$

然后,定义类别匹配的概率公式. 类别相等可以表示为 $Pr(c\subseteq c')$,即在 2 个知识库中的 2 个类别 c 和 c' 中相等的实例个数所占的比例,表述为

$$Pr(c\subseteq c')=\frac{\#c\cap c'}{\#c},$$

可以估计在 2 个知识库类别中相等实例的数量为

$$E(\#c\cap c')=\sum_{x:type(x,c)}(1-\prod_{y:type(y,d)}(1-P(x\equiv y))).$$

因此,最终的类别相等概率可表述为

$$Pr(c\subseteq c')=\frac{\sum_{x:type(x,c)}(1-\prod_{y:type(y,d)}(1-P(x\equiv y)))}{\#x:type(x,c)}.$$

4) 给出算法描述. PARIS 算法以 2 个待对齐的知识库作为输入,并假设每个知识库中不存在重复的实例,算法的输出为 2 个知识库中带有相似概率实体对的集合. 算法的主要步骤为:①根据实例匹配概率公式计算 2 个知识库中实例相似的概率;②根据关系匹配概率公式计算 2 个知识库中包含关系的概率;③迭代执行这 2 步直到算法收敛;④根据收敛后的数据和类别匹配概率公式计算类别的等价性. 为了更快执行效果,算法选择一个较小的参数 θ,并设定对于所有的关系对 r 和 r' 有 $Pr(r\subseteq r')=\theta$,从第 2 轮迭代开始,将使用计算的结果值替代 θ. 为了加速算法的收敛速度,引入了一个逐步递增的阻断因子,使得算法能够在几步后收敛.

PARIS 算法作为第 1 个在大规模知识库工作的全局式概率算法,在没有任何先验知识、调节参数和训练数据的条件下,可以同时完成实例、关系和类别对齐,并取得良好的实验效果. 但是 PARIS 不能处理结构异质性,即无法匹配一个关系和一个实例或者一个实例和一个字符串,即使他们指向的是同一个事物,PARIS 也不能匹配不同层级结构的实体,比如作为同一个人出生地的国家和这个国家的城市则无法匹配.

表 2 对 3.1～3.3 节介绍的实体对齐算法进行了分类汇总:

Table 2 Classification Summary Tables of Entity Alignment Algorithms

表 2 实体对齐算法分类汇总表

Category	Model	Content	Suitability	Advantage	Disadvantage
Pair-wise Entity Alignment Algorithm	Traditional Probabilistic Model	An attribute-similarity based pair-wised comparison method doesn't consider the relation between entities, and constructs a probabilistic model by changing the problem into a classification one.	Suitable for the entity alignment of simple domain knowledge base.	One of the earliest classic simple methods of pair-wised entity alignment, the classification idea is the fundamental of many later methods.	Simple structure; limited applicable scope; cannot utilize the characteristics and relations of knowledge bases.
	Supervised Learning and Semi-supervised Learning	Pre-tag some matched pairs as training data, and use training data to train the classification model by classification or clustering method, and then use the trained model to classify the unlabeled data.	Suitable for knowledge base entity alignment with a certain size of trained data.	Effectively classify or cluster entities with same attributes on certain conditions	Too much parameter tuning; parameter over-fitting; depend on training data; low efficiency; the result needs to be evaluated.
	Active Learning	Interact with human for obtaining the training data so as to further classify or cluster data.	Suitable for knowledge base entity alignment with a certain size which lacks of training data.	Only needs a small amount of training data or no training data; effectively classify or cluster entities with certain conditions; good human-computer interaction design can improve the alignment quality.	Too much parameter tuning; parameter over-fitting; low efficiency; complicated procedure; depend on human; the influence of human also needs to be evaluated.
	Unsupervised Learning	Use clustering algorithm to classify the entity in knowledge base.	Suitable for entity alignment with a certain size knowledge base which lacks of training data.	No training data needed; effectively classify or cluster entities with the same attributes on certain conditions.	Too much parameter tuning; parameter over-fitting; complicated procedure; low efficiency; the result needs to be evaluated.
Local Collective Alignment Algorithm	Vector Space Model(VMI for example)	Involve the attributes of related entities when calculating entity similarity.	Suitable for the entity alignment of large knowledge base with simple relations.	Simple and fast; Easy to find the effective pruning methods.	Low precision; depend on human; parameter tuning; unable to align classes and relations.
Global Collective Alignment Algorithm	Similarity Propagation (SIGMA for example)	Generate new matches by "bootstrapping".	Suitable for entity alignment of large knowledge base.	Fast; high *F-measure*; good scalability.	Depend on human; parameter tuning; unable to align classes and relations.
	LDA Model	An extension of the Dirichlet process mixture model, which obtains the relationship between the entities by a latten group variable.	Suitable for entity alignment of a certain size knowledge base with few relations.	An effective way of entity alignment.	Parameter tuning; over-fitting; low efficiency; difficult to enumerate all possible observation sequences.
	Conditional Random Fields Model	A CRF entity discrimination model which uses observation value as conditions to discriminate entities based on graph partitioning technique.	Suitable for entity alignment of a certain size of knowledge base with few relations.	An effective way of alignment; applicable for large data set; replace conditional model with the generative model to avoid the shortcomings of LDA model.	Parameter tuning; over-fitting; low efficiency; complex model.
	Markov Logic Network Model	Use the first-order predicate logic and the Markov random field, as well as the weighted SAT test and discriminative training algorithm to form a comprehensive entity resolution solution.	Suitable for entity alignment of a certain size knowledge base with few relations.	An effective way of entity alignment; applicable for large data set; able to deal with incompleteness and contradiction of domain knowledge.	Parameter tuning; over-fitting; low efficiency; complex model.
	Global Probability Algorithm (PARIS for example)	Effectively align classes, properties, relations and entities without parameter tuning.	Suitable for entity alignment of various large knowledge base.	Able to align classes, relations and entities at the same time; suitable for various knowledge base; little parameter tuning.	Low efficiency; unable to deal with structural heterogeneity; the convergence of the algorithm is to be proved.

4　基于相似性函数的特征匹配

4.1　基于文本相似性函数的特征匹配

4.1.1　基于 Token 的相似性函数

基于 token 的相似性函数使用某种函数将待匹配的文本字符串转换为一系列子串的集合,我们称这些子串为 token,称这个函数为标记化函数,记为 *tokenize*( ). 常用的基于 token 的相似性函数有 Jaccard 相似性函数[60]、余弦相似性函数[16]和基于 q-gram 的相似性函数[16].

Jaccard 系数等于 2 个集合的交与并的比值,可以用来衡量 2 个集合的相关性. Monge 和 Elkan 在文献[60]中给出了字符串相似性比较的 Jaccard 系数的形式化表示方式:

$$sim_{\text{Jaccard}}(s_1,s_2)=\frac{|tokenize(s_1)\bigcap tokenize(s_2)|}{|tokenize(s_1)\bigcup tokenize(s_2)|},$$

其中,s_1 和 s_2 为给定的 2 个待比较的字符串.

基于 Jaccard 系数的相似性函数优点在于集合相交操作是顺序无关的,因此不同 token 的先后顺序对度量结果没有影响. 但是函数具有错误敏感性,token 缺失或者录入错误等情况会对结果产生很大的影响.

余弦相似性是将 2 个文本字符串的 token 集合看作是 2 个 n 维的向量,通过计算这 2 个向量夹角的余弦值来评估这 2 个向量代表字符串的相似程度. 一般使用词频-逆文档频率(tf-idf)计算每个向量中 token 的权重 w,2 个字符串 s_1 和 s_2 对应文档的向量表示为 $\langle w_{11},w_{12},\cdots,w_{1n}\rangle$,$\langle w_{21},w_{22},\cdots,w_{2n}\rangle$,则 s_1 和 s_2 的余弦相似性可表示为

$$sim_{\text{Consine}}(s_1,s_2)=\frac{1}{W_1W_2}\sum_{t=1}^{n}w_{1t}\times w_{2t},$$

其中:

$$W_1=\sqrt{\sum_{t=1}^{n}w_{1t}^2},$$

$$W_2=\sqrt{\sum_{t=1}^{n}w_{2t}^2}.$$

余弦相似性同样具有基于 token 的相似性函数的顺序无关的优点,同时由于加入权重,可以更好地反映 token 的相似程度,但是与基于 Jaccard 系数的相似性函数一样无法有效解决错误敏感性的问题,因此提出一种基于 q-gram 的相似性函数. 这种相似性函数使用字符串的 q-gram 作为 token 进行相似性计算,由于 q-gram 产生的 token 是相互重叠的,因此可以有效地降低错误敏感性,代价是会产生更大的计算向量.

4.1.2　基于编辑距离的相似性函数

与基于 token 的相似性函数不同,基于编辑距离的相似性函数将待匹配文本字符串看作一个整体,通过将一个字符串转换成为另一个字符串所需要的编辑操作的最小代价作为衡量 2 个字符串相似性的度量,基本的编辑操作包括插入、删除、替换、交换位置等. 基于编辑距离的相似性函数可以有效地处理录入错误等错误敏感性问题. 常用的基于编辑距离的相似性函数有基于 Levenshtein 距离[61]、基于 Smith-Waterman 距离[62]、基于 affine gap 距离[63]、基于 Jaro 和 Jaro-Winkler 距离[26,64]的相似性函数.

给定 2 个字符串 s_1 和 s_2,它们之间的 Levenshtein 距离等于将 s_1 转换为 s_2 所需要的插入、删除和替换操作的最小次数. 通常 Levenshtein 距离通过动态规划来求解:算法初始化一个 $(|s_1|+1)\times(|s_2|+1)$ 的矩阵 $\boldsymbol{M}$,我们记 $\boldsymbol{M}$ 的第 i 行第 j 列的元素为 $M_{i,j}$,其中 $0\leqslant i\leqslant|s_1|$,$0\leqslant j\leqslant|s_2|$. $\boldsymbol{M}$ 的值为

$$M_{i,0}=i;$$

$$M_{0,j}=j;$$

$$M_{i,j}=\begin{cases}M_{i-1,j-1}, & s_{1,i}=s_{2,j};\\ 1+\min(M_{i-1},M_{i,j-1},M_{i-1,j-1}), & \text{otherwise.}\end{cases}$$

矩阵 $\boldsymbol{M}$ 中的值计算完成之后,$M_{|s_1|,|s_2|}$ 即为 Levenshtein 距离.

基于 Levenshtein 距离的相似性函数可以降低相似性匹配的错误敏感性,但是它为每一个字符的每一次编辑操作都赋予相同的权重(次数),并不考虑不同字符或子串的重要程度,但实际上不同位置的子串编辑操作对相似性匹配的重要性可能不同,比如一些前后缀和缩写词的处理. 基于 Smith-Waterman 距离和基于 affine gap 距离的相似性函数就是为了解决这一问题提出的 2 种算法.

对于待匹配的 2 个字符串的前后缀差异的影响,可以使用基于 Smith-Waterman 距离的相似性计算来解决,它通过寻找 2 个字符串的最长公共子序列,将最长公共子序列之外的子串作为前后缀,在计算时将最长公共子序列的编辑操作赋予较高的权重,而前后缀赋予较低的权重来降低前后缀对相似性匹配带来的影响. 对于字符串中间的缩写词可能产生的影响,可以通过基于 affine gap 距离来解决. 其基本思路是针对待匹配的 2 个字符串中一个字符串中间部分连续多个字符缺失的情况,对缺失子串整体赋予较低的权重,使得子串的缺失对相似性的影响要小于单个字符缺失累加带来的影响. 这 2 种相似性函数都可以通过动态规划的算法进行计算.

Jaro 和 Jaro-Winkler 相似性函数可以针对字符位置交换操作进行处理. 其中 Jaro 距离的主要思想是通过比较 2 个字符串的公共部分来计算相似程度,所谓"公共"这里特指 2 个字符相等并且它们在字符串中的位置距离之差 Δ 不大于较小字符串长度的一半,即 $\Delta \leqslant 0.5 \times \min(|s_1|, |s_2|)$,设 t 为公共部分发生位置交换的次数,δ 为公共字符的集合,则 Jaro 相似性函数可以定义为

$$sim_{\text{Jaro}}(s_1, s_2) = \frac{1}{3} \times \left(\frac{|\delta|}{|s_1|} + \frac{|\delta|}{|s_2|} + \frac{|\delta| - 0.5t}{|\delta|} \right).$$

基于 Jaro 距离的相似性函数可以容忍少量的拼写错误,但对于 2 个主体部分相同但前缀或者后缀不同的字符串的度量效果并不好. 这类相似性函数在人名的对齐中应用较多,在实践中人们发现人名前半部分出现错误的可能性远小于中部和尾部,因此提出一种如果字符串前半部分相同则提高相似性度量值的算法公式:对于 2 个字符串 s_1 和 s_2,以及共同前缀 τ,Jaro-Winkler 相似性函数可以表示为

$$sim_{\text{Jaro-Winkler}}(s_1, s_2) = sim_{\text{Jaro}}(s_1, s_2) + |\tau| \times f \times (1 - sim_{\text{Jaro}}(s_1, s_2)),$$

其中,f 为前缀 τ 对整体相似度影响的一个常量.

4.1.3 混合型相似性函数

基于 token 的相似性函数和基于编辑距离的相似性函数各有优缺点,在实践中为了更准确地判断相似性,经常将两者结合起来,构成混合型相似性函数,当然也会带来更高的计算复杂度. 常用的混合型相似性函数有扩展的 Jaccard 相似性函数[65-66]、Monge-Elkan 相似性函数[60]和 soft TF/IDF 相似性函数[67].

扩展的 Jaccard 相似性函数将传统的 Jaccard 方法通过 *tokenize*()函数形成的 token 的准确匹配扩展为相似性匹配,以容忍 token 的少量录入错误. 形式化上,使用 $sim_{\text{token}}(t_i, t_j)$ 作为 2 个字符串 token 的相似性度量,其中 $t_i \in tokenize(s_1)$, $t_j \in tokenize(s_2)$. token 的共享集合定义为

$$shared(s_1, s_2) = \{(t_i, t_j) \mid sim_{\text{token}}(t_i, t_j) > \theta\},$$

其中,θ 为相似性阈值. 类似地可以定义只在 s_1 不在 s_2 中出现的 token 集合为

$$Unique(s_1) = \{t_i \mid t_i \in tokenize(s_1) \wedge (t_i, t_j) \notin shared(s_1, s_2)\},$$

以及只在 s_2 不在 s_1 中出现的 token 集合为

$$Unique(s_2) = \{t_j \mid t_j \in tokenize(s_2) \wedge (t_i, t_j) \notin shared(s_1, s_2)\}.$$

定义每个 token 的权重为 w,则扩展的 Jaccard 相似性函数可以形式化定义为

$$sim_{\text{ExtJaccard}}(s_1, s_2) = \left\{ \sum_{(t_i, t_j) \in shared(s_1, s_2)} w(t_i, t_j) \right\} \Big/ \left\{ \sum_{(t_i, t_j) \in shared(s_1, s_2)} w(t_i, t_j) + \sum_{t_i \in Unique(s_1)} w(t_i) + \sum_{t_j \in Unique(s_2)} w(t_j) \right\}.$$

Monge-Elkan 相似性函数用来对由多个词构成的字符串进行相似性度量. 简单来说,它将与每一个 t_i 最相似的 t_j 的相似性度量值 $sim_{\text{token}}(t_i, t_j)$求和后标准化,可以形式化表示为

$$sim_{\text{MongeElkan}}(s_1, s_2) = \frac{1}{|tokenize(s_1)|} \sum_{i=1}^{|tokenize(s_1)|} \max_{j=1}^{|tokenize(s_2)|} sim_{\text{token}}(t_i, t_j).$$

Soft TF/IDF 相似性函数是一种在实体匹配中应用比较广泛的相似性函数. 与 Monge-Elkan 相似性函数不同的是,它只考虑相似性大于给定阈值 θ 的 token 对,并且赋予 token 不同的权重,这些与扩展的 Jaccard 相似性函数较为类似,但相对于 Jaccard 的共享相似性,soft TF/IDF 定义一个 s_1 的 token 的集合,相对于集合中的每一个 t_i,存在 t_j 使得它们之间的 token 相似性大于 θ,形式化表示为

$$close(\theta, s_1, s_2) = \{t_i \mid t_i \in tokenize(s_1) \wedge \exists t_j \in tokenize(s_2) : sim_{\text{token}}(t_i, t_j) > \theta\}.$$

对于每一个 $close(\theta, s_1, s_2)$中的 t_i,取与其相似性最大的 t_j,它们之间的度量值可表示为

$$sim_{\max}(t_i, t_j) = \max_{t_j \in tokenize(s_2)} sim_{\text{token}}(t_i, t_j),$$

则结合余弦相似性表达式(见 4.1.1)可以得到 soft TF/IDF 相似性函数的形式化表示方式:

$$sim_{\text{softTFIDF}} = \sum_{t_i \in close(\theta, s_1, s_2)} \left(\frac{tfidf_{t_i}}{W_1} \times \frac{tfidf_{t_j}}{W_2} \times sim_{\max}(t_i, t_j) \right).$$

4.2 基于结构相似性函数的特征匹配

结构相似性是实体对相似性度量的重要组成部分,常用的相似性函数包括直接计算实体对的共同邻居计数[68]、共同邻居的 Jaccard 相关系数[47]、Adar 评分[69]、Katz 评分[70]和 SimRank[71]等,下面对这 5 种函数进行简要介绍.

1) 共同邻居计数. 它是一种最简单直接的计算结构相似性的方法,通过直接计算 2 个实体相同邻居节点的个数来获得实体对的结构相似性,形式化表示为

$$sim_{\text{CommonNB}}(e_i, e_j) = \frac{1}{K} \times |NB(e_i) \cap NB(e_j)|,$$

其中,K 为一个足够大的常量能够使得所有实体对的相似性计数值都小于 1.

2) Jaccard 相关系数. 它是一种非常常用的集合相似性的度量函数. 基于 Jaccard 系数的结构相似性函数定义为实体对共同邻居集合的交集与并集的比,形式化表述为

$$sim_{\text{JaccardCoeff}}(e_i,e_j)=\frac{|NB(e_i)\cap NB(e_j)|}{|NB(e_i)\cup NB(e_j)|}.$$

3) Adar 评分. 前面 2 种方法为每个匹配的邻居赋予相同的权重,但实际中某些邻居可能对实体相似度的影响大小并不相同,Adamic 和 Adar[69] 正是基于这种观察,提出了一种类似 TF-IDF 的为关系赋予权重的思路:存在关联关系越多的实体其作为邻居在计算中所分配权重越低,结合 Jaccard 相关系数,实体相似性度量中的 Adar 评分可以定义为

$$sim_{\text{Adar}}(e_i,e_j)=\frac{\sum\limits_{NB(e_i)\cap NB(e_j)}u(e)}{\sum\limits_{NB(e_i)\cup NB(e_j)}u(e)},$$

其中,$u(e)$为实体 e 关系的唯一性程度. 可以看出,所有实体 $u(e)$相等时,

$$sim_{\text{Adar}}(e_i,e_j)=sim_{\text{JaccardCoeff}}(e_i,e_j).$$

4) Katz 评分. 有一些相似性度量考虑实体对之间关系的最短连接距离,其基本思想是如果 2 个实体之间由更多更短的关系路径所连接,则它们更相似,形式化表述为

$$sim_{\text{KatzScore}}(e_1,e_2)=\sum_{l=1}^{\infty}\beta^l\times|paths^{\langle l\rangle}(e_1,e_2)|,$$

其中,$paths^{\langle l\rangle}(e_1,e_2)$为 e_1 和 e_2 之间长度为 l 的路径之和;$\beta\in(0,1)$是一个衰减系数,β 越小则 l 越大的路径对相似度贡献就越小.

5) SimRank. 它是一种基于图的拓扑结构信息来衡量任意 2 个对象间相似程度的模型,该模型的核心思想为:如果 2 个对象被其相似的对象所引用,那么这 2 个对象也相似.

SimRank 模型定义 2 个实体的相似是基于下面的递归思想:如果指向实体结点 e_1 和 e_2 的结点相似,那么 e_1 和 e_2 也认为是相似的. 这个递归定义的初始条件是:每个结点与它自身最相似. 如果用记号 $I(e_1)$ 表示所有指向结点 e_1 的结点集合(即入邻点集合),用 $sim_{\text{SimRank}}(e_1,e_2)$表示 2 个对象间的 SimRank 相似度,则 $sim_{\text{SimRank}}(e_1,e_2)$的数学定义式为

① $sim_{\text{SimRank}}(e_1,e_2)=0$,当 $I(e_1)=\varnothing$或 $I(e_2)=\varnothing$;

② $sim_{\text{SimRank}}(e_1,e_2)=1$,当 $e_1=e_2$;

③ 在其他情况下,

$$sim_{\text{SimRank}}(e_1,e_2)=\frac{C}{|I(e_1)||I(e_2)|}\sum_{i=1}^{|I(e_1)|}\sum_{j=1}^{|I(e_2)|}sim_{\text{SimRank}}(I_i(e_1),I_j(e_2)),$$

其中,$C\in(0,1)$是一个衰减系数.

表 3 分类汇总了常用于实体对齐特征匹配的相似性函数:

Table 3　Classification Summary Tables of Similarity Function Based Feature Matching

表 3　基于相似性函数的特征匹配分类汇总表

Category	Function	Content	Suitability	Advantage	Disadvantage
Feature Matching Based on Textual Similarity	Token-Based Similarity Function	Jaccard Similarity Function	The ratio of the intersection and the union of two sets	Order independent.	Error-prone.
		Cosine Similarity Function	Tokens are seen as n-dimensional vectors. Evaluate similarity of the two vectors by calculating the cosine value of the angle between them.	Order independent; more accuracy with introducing weight measure.	Error-prone.
		q-gram Similarity Function	Use q-gram of strings as token to calculate similarity.	Order independent; reduce the error sensitivity.	Higher computational complexity.
	Edit Distance Similarity Function	Levenshtein Distance	The distance is equal to the minimum number of insertions, deletions and substitutions required to convert a string into another string.	Reduce the error sensitivity.	Don't consider the importance of different characters or sub strings.
		Smith-Waterman Distance	Find the longest common subsequence of two strings, and then use the test string as prefix or suffix, and give the longest common subsequence a higher weight, and give prefix and suffix a lower weight in calculation.	Effectively reduce the impact of different prefix and suffix.	Higher complexity; limited application scope: only take the impact of prefix and suffix into account.

Continued (Table 3)

Category	Function	Content	Suitability	Advantage	Disadvantage
Feature Matching Based on Textual Similarity	Edit Distance Similarity Function	Affine Gap Distance	For two strings to be matched, if in one string a number of consecutive characters are missing, give the missing characters lower weight.	Effectively reduce the possible impact of the abbreviations of the string.	Higher complexity; limited application scope: only take the middle part of a string into account.
		Jaro and Jaro-Winkler Distance	Calculate similarity by comparing the common part of Two strings.	Allow a few spelling errors.	Mainly used for name similarity.
	Hybrid Similarity Function	Extend Jaccard Similarity Function	Extend the traditional Jaccard method to the similarity match in order to tolerate small amount of input errors.	Relaxation of the conditions of Jaccard similarity measure can improve the recall.	Higher computational complexity.
		Monge-Elkan Similarity Function	Sum up all the similarity value of the most similar token pairs of Two strings and standardize it.	Easy implementation; be able to improve the recall rate of similarity computation.	Higher computational complexity.
		soft TF/IDF Similarity Function	Consider token pairs whose similarity values are greater than a given threshold, and use cosine similarity function with different weights.	Relaxation of the conditions of cosine similarity measure can improve the recall.	Higher computational complexity.
Feature Matching Based on Structural Similarity Function	Common Neighbors	Common Neighbors	Obtain structural similarity by directly calculate the number of common neighbors of Two entities.	Simple and straightforward.	Don't consider different weights of neighbors; parameter estimation problem.
	Jaccard Coefficient	Jaccard Coefficient	The ratio of the intersection and the union of two entities' neighbor sets	Simple.	Don't consider the different weights of neighbors.
	Adar Similarity	Adar Similarity	Give each matched pair of neighbors different weight.	Take into account the different weights of neighbors so as to obtain more accurate result.	Higher computational complexity.
	Katz Score	Katz Score	Consider the shortest distance between a pair of entities; if two entities are connected by more and more shorter path, they are more similar.	Consider all kinds of relationships among entities; effective method for structural similarity matching.	Parameter estimation problem; higher computational complexity.
	SimRank	SimRank	Use topology information to measure the similarity between two objects; two objects are similar if they are referenced by similar objects.	Consider the influence of the interaction between objects on the similarity value; a common method of structural similarity calculation.	Low efficiency and scalability on large data set.

5 分区索引技术

Christen 在文献[72]中对数据库实体匹配的索引技术进行了总结,这些数据库中的分区技术大部分也可以用于知识库的对齐过程. 本文在其基础上对用于知识库的分区索引技术进行分类和介绍. 以下符号用于各种索引计算复杂度的讨论:$n_A=|A|$ 和 $n_B=|B|$ 分别为知识库 A 和 B 对应的实体数量;b 为通过索引键值分区的块数(简单起见,假设 A 和 B 的分区块数相等). 对于索引剪枝能力的考查需要考虑索引键值的分布,由于实体的属性键值分布一般介于均匀分布和齐普夫率分布[15]之间,因此利用这 2 种分布产生的候选对数量(分别记为记为 N^U 和 N^Z)作为其上下界进行讨论.

5.1 基本的分区索引

基本的分区索引根据索引的定义直接选择实体属性作为索引键值进行构建,把具有相同索引键值的实体分配到同一区块,使得相似性匹配只在同一区块中进行. 最早是在数据库记录链接中提出[22],通常采用标准的倒排表进行实现. 相比于其他索引实现具有实现简单、速度快的优点,但索引键值的选

择对这种索引具有重要的影响,索引键值对应属性的缺失、错误或者分布的不均衡都会导致匹配的准确率和效率的下降.在不考虑索引键值缺失和错误的条件下,在知识库对齐中其产生的候选对数量的上下限分别为

$$N_{\mathrm{TB}}^{\mathrm{U}}=b\times\left(\frac{n_A}{b}\times\frac{n_B}{b}\right)=\frac{n_A\times n_B}{b},$$

$$N_{\mathrm{TB}}^{\mathrm{Z}}=\sum_{i=1}^{b}\left(\frac{1/i}{H_b}\times n_A\right)\times\left(\frac{1/i}{H_b}\times n_B\right)=\frac{n_An_B}{H_b^2}\times\sum_{i=1}^{b}\frac{1}{i^2},$$

其中,$N_{\mathrm{TB}}^{\mathrm{U}}$为索引键值服从均匀分布的索引产生候选对的数量,作为知识库对齐中产生候选对数量的下界;$N_{\mathrm{TB}}^{\mathrm{Z}}$为索引键值服从齐普夫率分布的索引产生候选对的数量,作为知识库对齐中产生候选对数量的上界;H_b为所有区块相对分布概率之和 $H_b=\sum_{i=1}^{b}1/i$.

5.2　基于滑动窗口的分区索引

基于滑动窗口的分区索引也称近邻排序索引,最早是由 Hernandez 等人在多数据库融合问题中提出[73],其核心思想是使用一个固定大小的滑动窗口在一个按照索引键值排好序的记录列表中滑动,如果窗口大小为 w,则每一次移动新进入窗口的记录与前面 $w-1$ 条记录进行匹配产生候选对,最前面一条记录则移出窗口.该文提出了一种基于排序数组的方法,在知识库实体链接过程中,将 2 个知识库的索引键值按照一定的顺序插入到一个数组中,通过将一个窗口中来自不同知识库的实体进行匹配产生候选对.这种索引方式相比于标准索引能够发现更多的匹配对,但也付出了更高的计算代价,而且如果窗口的大小不足以覆盖具有相同索引键值的所有记录,则会丢失正确的匹配对;同时这种索引方式对排序是错误敏感的,匹配记录如果因为排序的原因无法排在一起,也会丢失正确的匹配对.由于这种方式候选对的数量只与窗口大小有关,而与索引键值的分布无关,因此知识库对齐中其产生的候选对的数量为

$$N_{\mathrm{SNA}}^{\mathrm{U}}=N_{\mathrm{SNA}}^{\mathrm{Z}}=\alpha w\times\beta w+(n_A+n_B-w)\times(\alpha((w-1)\beta)+\beta((w-1)\alpha))=\frac{n_An_B}{(n_A+n_B)^2}\times(w^2+2(n_A+n_B-w)(w-1)),$$

其中,$\alpha=\frac{n_A}{n_A+n_B}$,$\beta=\frac{n_B}{n_A+n_B}$.

Christen 在另一篇技术报告[74]中提出了一种基于倒排表的近邻排序索引方法.这种方式结合了标准索引和基于数组的近邻排序索引的特点,使用标准索引的倒排表代替数组存储索引键值,同时滑动窗口产生候选对.当 $w=1$ 时,基于倒排表的近邻排序索引退化成为标准索引;当 $w>1$ 时,这种索引产生的候选对是标准索引产生候选对的一个超集,并且 w 越大,产生的候选对就越多.同样地,基于倒排表的近邻排序索引是使用更高的计算代价从更多的候选对中发现正确的匹配对,它克服了标准索引召回率低和基于数组的近邻排序索引窗口大小的问题,但同时也具有这二者的部分缺点:1)索引键值分布的不均衡会导致匹配数量的增大问题;2)索引键值错误敏感性问题.在不考虑索引键值缺失和错误的条件下,基于倒排表的近邻排序索引在知识库对齐中产生的候选对的数量的上下限分别为

$$N_{\mathrm{SNII}}^{\mathrm{U}}=w\frac{n_A}{b}\times w\frac{n_B}{b}+(b-w)\times\left(\frac{n_A}{b}\times w\frac{n_B}{b}+\frac{n_B}{b}\times(w-1)\frac{n_A}{b}\right)=\frac{n_An_B}{b^2}(w^2+(b-w)(2w-1)),$$

$$N_{\mathrm{SNII}}^{\mathrm{Z}}=\left(\frac{n_A}{H_b}\sum_{i=1}^{w}\frac{1}{i}\right)\times\left(\frac{n_B}{H_b}\sum_{i=1}^{w}\frac{1}{i}\right)+\sum_{j=2}^{b-w+1}\left(\frac{n_A}{H_b(j+w-1)}\sum_{i=j}^{j+w-1}\frac{n_B}{H_bi}+\frac{n_B}{H_b(j+w-1)}\sum_{i=j}^{j+w-1}\frac{n_A}{H_bi}\right).$$

另外一类近邻排序索引方法称为自适应的近邻排序索引[75-76],同样是结合标准索引和近邻排序索引特点改进而来.这种索引方式可分为 2 类:1)动态选择滑动窗口大小[75],使用索引键值上的“boundary pairs”来确定窗口的大小和位置进行分区,是具有相似度量值的实体尽可能分配在一起形成候选对;2)动态改变窗口移动的步长[76],而标准索引和近邻排序索引可以看作是这种“sorted blocks”方法的 2 个极端,即滑动窗口每次移动 1 则为标准索引,每次移动 $w-1$ 则为邻排序索引,通过参数来控制窗口重叠的程度.这 2 种索引方式都是“Blocking”和“Windowing”2 种方法的结合,它们对原有近邻排序索引作了改进,性能上略有提高.

5.3　基于相似性的分区索引

1) 基于 q-gram 的索引和基于后缀数组的索引

为了解决数据质量对索引带来的影响,基于

q-gram 的索引和基于后缀数组的索引方法在标准索引键值之上通过迭代的方式进行转换,生成部分列表,然后将这些列表进行处理后形成字符串列表并生成倒排索引.所不同的是基于 q-gram 的索引是生成 q-gram 列表,然后根据用户指定的阈值参数生成这些 q-gram 列表的指定长度的子列表组合,最后将这些组合转化为字符串生成倒排表形成基于 q-gram 的索引[77];而基于后缀数组的索引方法是生成后缀数组[78].这 2 种索引方法可以部分地解决错误敏感性问题,但其代价是产生大量的匹配对.文献[72]中的实验表明,这 2 种索引方式并不适合大规模知识库的对齐工作,这里不作进一步的介绍.

2) Hash 索引

Hash 索引的核心思想包含 4 个方面:①定义一个关于一项或多项属性的 Hash 函数,每个块 b_i 都有一个 Hash 值 h_i 标识;②将所有 $h(r)=h_i$ 的引用 r 都归入 b_i 中;③所有的块都互不相交;④实体解析算法仅在块内运行.其中,$h(x)$为把 x 映射成一个整数的 Hash 函数.

传统的 Hash 算法负责将原始内容尽量均匀随机地映射为一个签名值,属于伪随机数产生算法.传统 Hash 算法产生的 2 个签名,如果相等,说明原始内容在一定概率下是相等的;如果不相等,除了说明原始内容不相等外,不再提供任何信息.这种方式的 Hash 算法不能满足一些相似性的计算,因此,要设计一种对相似内容产生的签名也相近的 Hash 算法.MinHash[79]是 LSH[80]的一种,可以用来快速估算 2 个集合的相似度.MinHash 最初用于在搜索引擎中检测重复网页,也可以应用于大规模聚类问题或者用于实体对齐的索引.

$h_{\min}(S)$为集合 S 中的元素 $h(x)$后具有最小 Hash 值的元素,那么对 2 个集合 A 和 B,$h_{\min}(A)=h_{\min}(B)$成立的条件是 $A\cup B$ 中具有最小 Hash 值的元素也在 $A\cap B$ 中.假设,$h(x)$是一个良好的 Hash 函数,它具有良好的均匀性,能够把不同元素映射成不同的整数,可得 $Pr[h_{\min}(A)=h_{\min}(B)]=Jaccard(A,B)$,即集合 A 和 B 的相似度为集合 A,B 经过 Hash 后最小 Hash 值相等的概率.通过这个结论,可以根据 MinHash 来计算 2 个集合的相似度.一般有 2 种方法:

① 使用多个 Hash 函数.为了计算集合 A,B 具有最小 Hash 值的概率,我们可以选择 K 个 Hash 函数,用这 K 个 Hash 函数分别对集合 A,B 求 Hash 值,对每个集合都得到 K 个最小值,集合 A,B 的相似度为 K 个最小值中相同元素个数与总的元素个数的比值.

② 使用单个 Hash 函数.针对第 1 种方法计算复杂度高的问题可以使用单个 Hash 函数来求解.定义集合 S 中具有最小 Hash 值的 K 个元素,只需要对每个集合求一次 Hash,然后取最小的 K 个元素,集合 A,B 的相似度等于集合 A 中最小的 K 个元素与集合 B 中最小的 K 个元素的交集个数与并集个数的比值.

5.4 基于聚类的分区索引

1) Canopy 聚类索引

Canopy 聚类索引是通过高效的计算索引键值之间的距离(相似性)进行聚类,从而将索引键值对应的实体记录插入到互相重叠的聚类当中,候选对则从每个聚类的实体对中产生[36,81-82].索引键值的相似性一般使用 Jaccard 或 Cosine 相似性度量来计算.Canopy 聚类索引首先创建一个倒排表,此倒排表的键由实体属性词的集合或者词的 q-gram 的集合产生,列表的内容包含这些集合中的项所对应的实体记录的标识.倒排表建立之后,算法通过以下 5 个步骤迭代地产生互相重叠的聚类:①将知识库所有待匹配实体的标识插入到集合 P 中;②随机选择 P 中的一个标识 r_c 作为新聚类 C_i 的中心,并计算集合 P 中所有其他点与该中心的索引键值的相似性;③选择 2 个阈值 t_l 与 t_t($t_l\leqslant t_t$),将所有相似性大于 t_l 的实体标识插入到 C_i 中;④将所有相似性大于 t_t 的实体标识连带 r_c 移出 P;⑤重复执行步骤②~④,直到 $P\subset\varnothing$.从上面的算法可以看到 t_l 与 t_t 是 2 个重要的控制参数,如果 $t_l=t_t$ 则产生的聚类不会重叠,如果 $t_l=t_t=1$ 则聚类产生的索引分区与标准索引是相同的.由于聚类的大小决定于索引键值的分布、相似性函数的选择和这 2 个参数,我们无法准确地控制每个聚类的大小,也很难评估产生的候选对的数量.我们知道候选对的数量与聚类的重叠程度有关,假设每个实体记录可能插入 1~v 个聚类中,则候选对数量的上下限分别为

$$\frac{n_A n_B}{b}\leqslant N_{CCT}^{U}\leqslant\frac{n_A n_B v^2}{b},$$

$$\frac{n_A n_B}{H_b^2}\times\sum_{i=1}^{b}\frac{1}{i^2}\leqslant N_{CCT}^{Z}\leqslant\frac{n_A n_B v^2}{H_b^2}\times\sum_{i=1}^{b}\frac{1}{i^2}.$$

以上形式的聚类索引称为基于阈值的 Canopy 聚类索引,Christen 在文献[74]中提出了一种最近邻 Canopy 聚类索引方法.这种方法的思想是使用

2 个最近邻参数 n_l 和 n_t 分别替代基于阈值的参数 t_l 与 t_t,使得算法可以准确地控制每个聚类的大小,但是也会产生由于指定的聚类大小无法涵盖所有索引键值对应的实体而丢失匹配对的情况. 最近邻 Canopy 聚类索引方法可以准确地计算聚类的数量和大小,其产生候选对的数量为

$$N_{\mathrm{CCN}}^{\mathrm{U}} = N_{\mathrm{CCN}}^{\mathrm{Z}} = \frac{n_A n_l}{n_t} \times \frac{n_B n_l}{n_t} = \frac{n_A n_B n_l^2}{n_t^2}.$$

2) 基于 StringMap 的索引

基于 StringMap 的索引的核心思想是将索引键值转换为对象映射到高维空间并保留索引键值之间的相似性,然后使用与 Canopy 索引同样的方式聚类相似的实体对象[83]. 基于 StringMap 的索引建立有 2 个步骤:①迭代地产生 n 个高维空间坐标系,将原有实体分别映射到这 n 个坐标系下;②采用和 Canopy 索引类似的方法在高维空间下对索引键值相似的实体进行聚类,所有包含相似索引键值的实体将被插入到同一个区块中. 这个过程中有 3 个关键问题需要考虑:①映射到高维空间时枢轴字符串的选择,一般采用 StringMap 中的迭代的最远最优先算法;②高维空间维度的选择,实验得到的比较理想的数值为 $15 \leqslant n \leqslant 25$,如果维度选择过大则可能导致维度灾难;③存储高维对象的数据结构的选择,常用的存储结构有 R-Tree[83],KD-Tree[84],Grid[85]等.

5.5　动态分区索引

在大规模知识库的实体对齐过程中,由于缺乏属性的先验知识和训练数据,特别是在找不到可以快速计算的距离度量的情况下,无法自动地选择合适的索引键值. 针对这种情况,McNeill 等人在文献[86]的基础上提出了一种动态分区索引的方法[87],其核心思想是记录按照一定的次序(字母序)通过共享的属性值划分成区块,对于超过指定大小的区块动态选择共享属性继续分区直到所有区块大小都小于指定阈值,将所有分区数据形成候选对以进行相似性计算获得最终结果. McNeill 设计了 2 层的分区索引算法并对可能重复计算的候选对进行了筛选,同时将算法并行化运行于 Hadoop 集群上并取得了较好的实验效果.

但是,McNeill 等人的动态索引算法需要将基于属性的索引键值预先固定顺序,这个前提条件在通用知识库上并不容易实现. Lee 等人针对这个问题提出了一种改进的动态索引算法[17],不需要预先确定知识库索引键值的选择顺序,只是简单地枚举索引键值所有可能的组合,再逐步分块得到所有合适大小的索引区块. 采用这种方法的原因有 4 点:1)选择区分能力强的属性很有可能由于表示方式或者录入错误等原因导致召回率降低,我们必须最大程度降低这种风险;2)相同属性在不同知识库实例中的分辨能力不同,必须保证分区后 2 个知识库中的区块大小同时小于指定阈值;3)虽然枚举所有组合可能导致大量的区块产生,但由于每轮分区深度一般不会超过 3 层,所以所花费的计算代价并不大;4)这是最重要的一个原因,我们很难预先定义一个通用知识库的索引键值选择次序. 算法过程简要描述如下:算法首先将每个知识库看作是一个大的分区;然后根据知识库中先验对齐的类别、实例和字面量构建索引键值对,根据这些索引键值对递归地创建子分区,直到每个分区大小都小于指定阈值或者每对索引键值对都被使用后停止;最后算法将每次循环得到的分区并入候选对集合. 在分区大小指定为一个较小的数值的情况下,每个分区内实体相似性的计算代价可以看作常数时间,因此实体相似性的时间复杂度即为分区的个数,极大地减少了计算量,适合于大规模知识库的实体对齐.

表 4 分类汇总了实体对齐中的分区索引算法:

Table 4　Classification Summary Tables of Blocking Algorithms in Entity Alignment

表 4　实体对齐中的分区索引算法分类汇总表

Category	Algorithm	Content	Advantage	Disadvantage
Standard Blocking	Standard Blocking	Directly select properties as blocking key values, and assign entities with the same blocking key to the same block.	Simple and fast; suitable for simple applications	Any missing, unbalanced properties will reduce the accuracy and efficiency; unsuitable for complicated attributes.
Sliding Window Based Blocking	Based on Sorted Array	Insert blocking key value pairs from two knowledge bases into a sorted array; generate candidate pairs by matching entities from different knowledge bases in the same specified window.	Be able to find more matching pairs compared with standard index.	Higher computational cost; if the window is not large enough to cover all the records of the same key, matching pairs will be lost; sensitive to sort error.

Continued (Table 4)

Category	Algorithm	Content	Advantage	Disadvantage
Sliding Window Based Blocking	Based on Inverted Lists	Combine the features of standard blocking and the nearest neighbor blocking, and slide the window to generate candidate pairs with inverted list to store the blocking key.	Higher recall than standard index; avoid the window size problem of sorted neighborhood approach.	Higher computational cost; unbalanced distribution of blocking key will degrade performance; blocking key error-prone.
	Self-adaptive	Combine the standard index and the nearest neighbor sorted index, and dynamically select the sliding window size, or dynamically change the step size of the window moving.	Same with the inverted lists but higher performance.	Same with the inverted lists but less.
Similarity Based Blocking	Based on q-gram and Suffix Array	Use inverted index on the different string lists generated from blocking key values.	Alleviate the impact of blocking key error.	Generate too many matching pairs; unsuitable for large data set.
	Hash	Define a Hash function based on one or more properties, and each block has a Hash value. All the references that have the same Hash value are put in the same block. All the blocks are disjoint. The entity resolution algorithm is carried out within the block.	Improve the accuracy; effective data partition; reduce the computational complexity.	The choice of Hash function; algorithm efficiency needs to be further improved.
Clustering Based Blocking	Canopy Clustering	Use the distance between blocking key values to cluster so as to insert the entity within a certain range into the overlapping cluster; the candidate pairs are generated from entity pairs in each cluster.	Find more candidate pairs; reduce error sensitivity; nearest neighbor canopy clustering can precisely calculate the size and the number of cluster	High computational cost; threshold canopy clustering can lead to unbalanced distribution problem.
	StringMap Based Blocking	Convert blocking key value as object to high-dimensional space and keep the similarity, and cluster similar entity object as canopy clustering.	Find more candidate pairs; reduce error sensitivity.	High computational cost.
Dynamic Blocking	Two-Layer Blocking	Dynamically select different blocking key values shared by two knowledge bases to produce blocks until all the block sizes are less than a specified value	Easy to implement; simple but effective; suitable for large-scale knowledge base alignment.	Depend on the scope and the quantity of shared properties; a total order of properties is required beforehand.
	Improved Dynamic Blocking	Arbitrary enumerates all possible combinations of blocking key values, and iteratively blocks the instances to a proper size in a recursive way	Same as two layer blocking but no fixed order required.	Higher computational cost; depend on the scope and the quantity of shared properties.

6 常用测试数据集简介

一般用于实体对齐算法效果评测的数据集可分为2类:1)实验室中人工合成的专用评测数据集,我们称为基准测试数据集.这类数据集规模相对较小,结构和内容相对简单,不能够全面充分地测试算法的性能,但由于提供了匹配的标准结果,可以通过结果的比较从某一方面较为精确地反映算法的优劣.这类数据集主要来自本体匹配工具评测平台OAEI (ontology alignment evaluation initiative)①,具体测试数据集将在6.1节进行介绍.2)各种机构或组织根据实际需要通过各种数据采集手段从真实世界中抽取所需数据构建的知识库,将之用于对其算法评价,我们称为真实世界测试数据集.这类数据集一般规模较大、结构复杂多样、匹配难度大,可以很好地对算法的效率和可扩展性进行充分的测试,但由于很难构建标准结果,因此一般通过采样或者人机结合的方式进行结果准确性的评测.这类数据集主要来自LOD项目[11]中的知识库,常用的知识库在6.2节进行介绍.对这2类数据集的评测指标主要采用1.2节介绍的 *precision*、*recall* 和 *F-measure*,以及通过 *precision-recall* 曲线进行可视化的结果展示.

6.1 基准测试数据集

OAEI是一个评测和比较本体匹配工具的国际

① http://oaei.ontologymatching.org

比赛,从 2009 年开始增加了实例匹配的测试内容,先后提供了 6 种实例对齐的测试数据集,这些数据集可以用于知识库的实体对齐算法的评测,简要介绍如下:

1) A-R-S 测试数据集.其包括 3 个来源于文献出版领域的数据集(eprints,rexa,Sweto-DBLP),数据量从 eprints 的几百个实例到 rexa 的 1 万余个实例,再到 Sweto-DBLP 的 160 余万个实例,变化幅度很大,可以很好地测试算法的扩展性.

2) T-S-D 测试数据集.其包括 3 个涵盖多领域的根据不同结构构建的数据集(TAP,SwetoTestbed,DBpedia),这 3 个数据集的数据量相对较大,主要评测的是算法的效率性能.

3) IIMB(Islab instance matching benchmark)测试数据集.其数据来自 OKKAM 项目,包含了关于演员、运动员和企业公司的数据.根据不同的修改策略,该测试集被划分成 37 个子目录,每个测试目录含有约 300 个实例标识符及 RDF 数据,这些修改后的 RDF 文档需与一个原始的 RDF 文档进行匹配.由于提供了丰富的修改策略,这个测试集可以从不同方面较为全面地反映算法的性能,但受数据量的限制无法对算法效率和扩展性进行评测.

4) DI(data interlinking)测试数据集.它是一个规模较大的数据集,其提供的数据包主要包含了多个生物医学领域的大型知识库(dailymed,diseasome,drugbank,linkct,sider)和关于电影的大型本体知识库 LinkedMDB,部分数据集需要通过联机查询方式获取.这个评测数据集的数据来自于真实世界的知识库,数据量较大且结构较为复杂,对算法提出了较高的要求.

5) PR(persons-restaurants)测试数据集.它是较为常用的作为初始效果评测的数据集,包含了人和餐馆的信息的对齐,每个数据集包含数百个到几千个实例,实例主要是属性信息,可以通过根据某些修改策略(主要是添加或删除一些属性)对数据集作不同程度地改变来进行算法的测试.这个测试集规模小、结构简单、测试能力有限.

6) NYT(interlinking New York Times data)测试数据集.它是一个规模较大的数据集,数据来源于纽约时报的开放链接数据,内容包含人、组织和地点 3 个方面,可以用来与真实世界的知识库进行对齐进而评测算法的对齐质量.

6.2　真实世界测试数据集

LOD 项目将大量基于 RDF 知识库中的数据集链接起来.到 2014 年,LOD 项目已经采集了超过 1 000 个知识库的 600 亿条 RDF 三元组.在 LOD 项目云图中,处于核心地位的有 3 个知识库:Freebase,DBpedia,YAGO.这 3 个知识库与其他知识库存在大量的链接关系,同时它们之间也标注了很多的共指关系.可以说,这 3 个知识库是实体对齐算法的优秀评测数据集.

1) Freebase[88].它由数据库技术公司 Metaweb 创建,后被谷歌收购,成为谷歌知识图谱的重要组成部分.Freebase 中的数据来自于维基百科、IMDB、Flickr 等众多网站或数据集,由计算机和人共同维护,经过非常严格的处理过程进行增加或修改.Freebase 使用 MQL(Metaweb query language)查询语言向用户提供了数据访问接口.其类别层次结构较为简单,但事实条目数量巨大,数据质量较高,至 2014 年 Freebase 已经包含了超过 24 亿条三元组信息.谷歌公司决定于 2015 年 6 月将 Freebase 全部移入 WikiData①,并使用 WikiData 的 API 提供数据服务.

2) DBpedia[89].它由德国莱比锡大学和曼海姆大学的科研人员创建的多语言的综合型知识库.DBpedia 处于 LOD 项目的最核心地位,被广泛应用于各种科研的实际系统.其数据的主要来源是多种语言的维基百科中抽取出的结构化信息,包含了众多领域的实体信息.与 Freebase 类似,DBpedia 的类别结构并不复杂,但其实例的属性较多,且事实三元组的数量同样巨大,截止 2014 年 DBpedia 三元组的数量已经超过了 30 亿条.

3) YAGO[90-91].它是一个由德国马普所(Max Planck Institute,MPI)的科研人员构建的综合型知识库.YAGO 知识库将维基百科、WordNet 和 GeoNames 等数据源的知识整合在知识库中,特别是将维基百科的分类体系和 WordNet 的分类体系进行融合,为 YAGO 构建了一个复杂的类别层次结构体系,其类别的数量超过 36 万条,但其实体数量和事实三元组条目相对较少,目前 YAGO2 包含了 1 000 多万实体的 1.2 亿条三元组信息,最新版的 YAGO3 则提供了多语言的版本.

① https://www.wikidata.org/wiki/Wikidata:Main_Page

7 机遇与挑战

知识库的实体对齐技术来源于传统的实体匹配技术,也可以借助数据库、机器学习和自然语言处理的一些方法,并结合了当前基于语义网知识库的特点,是一个综合性的研究方向.关于知识库对齐和本体匹配的相关工作也有很多综述性的文章.文献[92]从信息检索和数据挖掘角度对基于本体的开放网络知识库的构建、知识融合、知识检索、数据挖掘和系统应用进行全面地综述,有利于我们从全局角度把握知识库实体对齐的重要作用.文献[3]对2001—2011年期间基于本体匹配的技术变迁进行了总结,并分类汇总了期间出现的各种匹配工具,虽然文献[3]综述的是本体的架构的匹配工作,但其中很多技术和思想也可以用于本文探讨的知识库实体对齐,其介绍的部分工具也可以用于实体对齐.文献[4]是另外一篇比较有影响力的讨论本体匹配的综述性文章,文中对本体匹配工作的现状和发展趋势进行了讨论,同样介绍了相关的系统和测试集,重点对本体匹配工作面临的8大挑战给予详细的阐述,对知识库的实体对齐工作有很大的启发意义.近几年随着知识库的发展,也出现了一些关于知识库实体对齐的综述性文章.文献[93]从机器学习的角度对知识库的实体链接工作相关的问题、技术和解决方案进行了汇总和分析.文献[94]则是最新的一篇从文献综述的角度对本体匹配和知识库实体对齐统一考虑的文章,该文对2003年以来这一领域相关文献的发展变化情况进行了深入的分析,对文献中涉及的系统进行了详细的分类汇总,并且对文献中关注的问题和挑战进行了总结,为这一方向的深入研究提供了详尽的资料支持.

根据文献调研[94],知识库实体对齐的研究工作兴起于2003年,并于2011—2013年达到一个高峰,目前仍处于高速发展时期,各类方法层出不穷,尤其是结合大数据的相关技术,在大数据条件下的知识库实体对齐的算法研究和系统构建成为了当下研究的热点.知识库实体对齐虽然取得了丰硕的研究成果,但仍有许多亟待解决的问题,概括起来有6方面:

1) 并行与分布式算法.大数据条件下的知识库由于数据量巨大、数据结构复杂,对实体对齐算法的效率和扩展性提出很多挑战.目前很多研究着力于使用主流的并行或分布式算法应对这些挑战[95-97],这些工作将对齐工作运行于分布式的计算平台(多核或者多处理器)或者并行的编程环境(MPI,MapReduce或者基于内存计算的Spark等),将对齐算法的不同步骤并行化,充分考虑不同模块的负载均衡和高效的消息传递机制,以期达到对齐效率和效果的大幅度提升.

2) 实时算法.目前大部分对齐的研究工作都是离线处理,在某些应用场景下可能需要实时地处理实体对齐,因此需要高效的实时算法来解决这个问题.当前已经有部分的实时实体对齐的研究工作[98-100],但大部分都是针对具体应用领域,数据结构相对简单且数据量较小.大规模知识库的实时对齐算法可以考虑将离线和在线算法更好地结合起来,还可以考虑预先建立索引及应用分区剪枝技术有效地减少候选对,并在不影响匹配质量的前提下有效地降低跨知识库查询的通信代价.

3) 众包算法.人机结合的众包算法可以有效地提高匹配质量,提供丰富的先验对齐数据,尤其在大数据条件下,设计高效的众包算法,可以以较小的代价产生巨大的效果[101-104].众包算法的设计主要是考虑数据量、对齐质量以及人工标注三者重要性的权衡,要将众包平台与对齐模型有机结合起来,同时要能够有效地判别人工标注的质量,这些方面都有待进一步研究.

4) 跨语言知识库对齐.随着LOD项目的发展,多语言知识库越来越多,建立跨语言的知识库的链接能够极大地丰富LOD的覆盖范围,提高不同知识库的互补能力.目前已经有部分跨语言知识库对齐的研究工作,取得了一定的进展[105-106],但对齐质量不高,对齐效果还有待进一步提升.

5) 测试数据集及评价.当前知识库对齐研究的一个主要问题就是缺少可供研究者测试和评价算法的统一的测试评价平台和数据集.虽然OAEI正致力于这方面的建设工作,但其测试平台数据集的全面性和评测能力并不足以覆盖当前对齐算法的多样性,很多算法仍使用自己构建的数据集进行测试.因此,迫切需要建立一个全面地适用于LOD环境的测试评价平台,能够统一地评价对齐算法的综合能力,同时需要更加全面合理的评价指标体系.

6) 实用系统的研发.现有的大部分实体对齐的研究还停留在实验室或原型系统阶段,无法在真实系统中获得广泛应用.构建一个稳定的、易用的、可扩展的、有效集成多种算法的、能够完成多种对齐任务的系统是未来研究的一个重要方向.

8 总　　结

本文在对知识库实体对齐的相关概念、算法、技术和存在问题深入研究的基础上,总结了3大类实体对齐算法以及基于相似性函数的特征匹配和分区索引2类技术,对常用的测试数据集进行了介绍,并探讨了知识库实体对齐工作的机遇与挑战.基于语义网的知识库对齐的研究工作既是近些年来一个新兴的重要课题,又有许多现有相关领域的研究成果可供借鉴,目前正处于高速发展阶段,取得了一定的成果,但仍有大量的问题亟待解决.随着知识库数量和数据规模的不断增加,将会有越来越多的算法和系统涌现出来,将不同的知识库链接起来,形成"知识之网"并推动Web不断向前发展.

参 考 文 献

[1] Sheth A, Thirunarayan K. Semantics Empowered Web 3.0: Managing Enterprise, Social, Sensor, and Cloud-Based Data and Services for Advanced Applications [M]. San Rafael, CA: Morgan and Claypool, 2013

[2] Shvaiko P, Euzenat J. Ten Challenges for Ontology Matching [C] //Proc of on the Move to Meaningful Internet Systems. Berlin: Springer, 2008: 1164-1182

[3] Bernstein P A, Madhavan J, Rahm E. Generic schema matching, ten years later [J]. Proceedings of the VLDB Endowment, 2011, 4(11): 695-701

[4] Shvaiko P, Euzenat J. Ontology matching: State of the art and future challenges [J]. IEEE Trans on Knowledge & Data Engineering, 2013, 25(1): 158-176

[5] Bleiholder J, Naumann F. Data fusion [J]. ACM Computing Surveys, 2008, 41(1):137-153

[6] Elmagarmid A K, Ipeirotis P G, Verykios V S. Duplicate record detection: A survey [J]. IEEE Trans on Knowledge & Data Engineering, 2007, 19(1): 1-16

[7] Naumann F, Bilke A, Bleiholder J, et al. Data fusion in three steps: Resolving inconsistencies at schema-, tuple-, and value-level [J]. Bulletin of the Technical Committee on Data Engineering, 2006, 29(2): 21-31

[8] Wang H F. Survey: Computational models and technologies in anaphora resolution [J]. Journal of Chinese Information Processing, 2002, 16(6): 9-17

[9] Zhao J. A survey on named entity recognition, disambiguation and cross-lingual coreference resolution [J]. Journal of Chinese Information Processing, 2009, 23(2): 4-17

[10] Hu Wei, Bai Wenyang, Qu Yuzhong. Research on resolving object coreference on the semantic Web [J]. Journal of Software, 2012, 23(7): 1729-1744 (in Chinese)
(胡伟,柏文阳,瞿裕忠. 语义Web中对象共指的消解研究[J]. 软件学报, 2012, 23(7): 1729-1744)

[11] Bizer C, Al E. Linked data-the story so far [J]. International Journal on Semantic Web & Information Systems, 2009, 5(3): 1-22

[12] Lacoste-Julien S, Palla K, Davies A, et al. SIGMa: Simple greedy matching for aligning large knowledge bases [C] // Proc of the 2013 ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2013: 572-580

[13] Li Juanzi, Tang Jie, Yi Li, et al. RiMOM: A dynamic multistrategy ontology alignment framework [J]. IEEE Trans on Knowledge & Data Engineering, 2009, 21(8): 1218-1232

[14] Christen P, Goiser K. Quality and Complexity Measures for Data Linkage and Deduplication [M]. Berlin: Springer, 2007

[15] Witten I H. Managing Gigabytes [M]. San Francisco, CA: Morgan Kaufmann, 1999

[16] Naumann F. An Introduction to Duplicate Detection [M]. San Rafael, CA: Morgan and Claypool, 2010

[17] Lee S, Hwang S. ARIA: AsymmetRy resistant instance alignment [C] //Proc of the National Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2014: 94-100

[18] Batini C, Scannapieco M. Data Quality: Concepts, Methodologies and Techniques [M]. Berlin: Springer, 2006

[19] Lee Y W, Pipino L L, Funk J D, et al. Journey to data quality [J]. Electronic Library, 2006, 25(6): 793-794

[20] Christen P. Data Matching [M]. Berlin: Springer, 2012

[21] Newcombe H B, Kennedy J M, Axford S J, et al. Automatic linkage of vital records [J]. Science, 1959, 130(3381): 954-959

[22] Fellegi I P, Sunter A B. A theory for record linkage [J]. Journal of the American Statistical Association, 1969, 64(328): 1183-1210

[23] Herzog T N, Scheuren F J, Winkler W E. Data Quality and Record Linkage Techniques [M]. Berlin: Springer, 2007

[24] Porter E H, Winkler W E, Census B O T, et al. Approximate string comparison and its effect on an advanced record linkage system, RR97/02 [R]. Washing DC: US Bureau of the Census, 1997: 190-199

[25] Winkler W E. String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage [C] //Proc of the Section on Survey Research Methods. Washington DC: ASA, 1990: 354-359

[26] Winkler W E, Thibaudeau Y. An application of the Fellegi-Sunter model of record linkage to the 1990 U. S. decennial census, RR91/09 [R]. Washington DC: US Bureau of the Census, 1991

[27] Winkler W E. Methods for record linkage and Bayesian networks, RRS2002/05 [R]. Washington DC: US Bureau of the Census, 2001

[28] Verykios V S, Moustakides G V, Elfeky M G. A Bayesian decision model for cost optimal record matching [J]. VLDB Journal, 2003, 12(1): 28-40

[29] Han J W, Kambe M. Data Mining: Concepts and Techniques [M]. San Francisco, CA: Morgan Kaufmann, 2006

[30] Vapnik V. The Nature of Statistical Learning Theory [M]. Berlin: Springer, 2000

[31] Kantardzic M. Data Mining [M]. Hoboken, NJ: John Wiley & Sons, 2011: 235-248

[32] Cochinwala M, Kurien V, Lalk G, et al. Efficient data reconciliation [J]. Information Sciences, 2001, 137(14): 1-15

[33] Elfeky M G, Verykios V S, Elmagarmid A K. TAILOR: A record linkage toolbox [C] //Proc of 2012 IEEE Int Conf on Data Engineering (ICDE 2012). Piscataway, NJ: IEEE, 2002: 17-28

[34] Christen P. Automatic training example selection for scalable unsupervised record linkage [G] //LNAI 5012: Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conf. Berlin: Springer, 2008: 511-518

[35] Chen Z, Kalashnikov D V, Mehrotra S. Exploiting context analysis for combining multiple entity resolution systems [C] //Proc of the 2009 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2009: 207-218

[36] Cohen W W, Richman J. Learning to match and cluster large high-dimensional data sets for data integration [C] //Proc of the 2002 ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2002: 475-480

[37] McCallum A, Wellner B. Conditional models of identity uncertainty with application to noun coreference [C] //Proc of Advances in Neural Information Processing Systems 17. Cambridge, MA: MIT Press, 2005: 905-912

[38] Pasula H, Marthi B, Milch B, et al. Identity uncertainty and citation matching [C] //Proc of Advances in Neural Information Processing Systems 15. Cambridge, MA: MIT Press, 2003: 1425-1432

[39] Sarawagi S, Bhamidipaty A. Interactive deduplication using active learning [C] //Proc of the 2002 ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2002: 269-278

[40] Tejada S, Knoblock C A, Minton S. Learning domain-independent string transformation weights for high accuracy object identification [C] //Proc of the 2002 ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2002: 350-359

[41] Arasu A, Götz M, Kaushik R. On active learning of record matching packages [C] //Proc of the 2010 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2010: 783-794

[42] Verykios V S, Elmagarmid A K. Automating the approximate record matching process [J]. Information Sciences, 2000, 126(1): 83-98

[43] Ravikumar P, Cohen W. A hierarchical graphical model for record linkage [C] //Proc of the 20th Conf in Uncertainty in Artificial Intelligence. Banff, Canada: AUAI, 2004: 454-461

[44] Bhattacharya I, Getoor L. A latent Dirichlet allocation model for unsupervised entity resolution [C] //Proc of the 6th SIAM Int Conf on Data Mining. Philadelphia, PA SIAM, 2006: 47-58

[45] Li Juanzi, Wang Zhichun, Zhang Xiao, et al. Large scale instance matching via multiple indexes and candidate selection [J]. Knowledge-Based Systems, 2013, 50: 112-120

[46] Dong X. Reference reconciliation in complex information spaces [C] //Proc of the 2005 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2005: 85-96

[47] Bhattacharya I, Getoor L. Collective entity resolution in relational data [J]. ACM Trans on Knowledge Discovery from Data, 2007, 1(2): 2007

[48] Tang Jie, Li Juanzi, Liang Bangyong, et al. Using Bayesian decision for ontology mapping [J]. Journal of Web Semantics, 2006, 4(4): 243-262

[49] Hall R, Sutton C, Mccallum A. Unsupervised deduplication using cross-field dependencies [C] //Proc of the 2008 ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2008: 310-317

[50] Domingos P. Multi-relational record linkage [C] //Proc of the KDD-2004 Workshop on Multi-Relational Data Mining. New York: ACM, 2004: 31-48

[51] Singla P, Domingos P. Entity resolution with Markov logic [C] //Proc of 2006 IEEE Int Conf on Data Mining (ICDM 2006). Piscataway, NJ: IEEE, 2006: 572-582

[52] Rastogi V, Dalvi N, Garofalakis M. Large-scale collective entity matching [J]. Proceedings of the VLDB Endowment, 2011, 4(4): 208-218

[53] Suchanek F M, Abiteboul S, Senellart P. PARIS: Probabilistic alignment of relations, instances, and schema [J]. Proceedings of the VLDB Endowment. 2011, 5(3): 157-168

[54] Bansal N, Blum A, Chawla S. Correlation clustering [C] // Proc of the 43rd Annual Symp on Foundations of Computer Science (FOCS 2002). Piscataway, NJ: IEEE, 2002: 238-247

[55] Collins M. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms [C] //Proc of the Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2002: 1-8

[56] Wick M, Singh S, Mccallum A. A discriminative hierarchical model for fast coreference at large scale [C] //Proc of the 50th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2012: 379-388

[57] Xu Congfu, Hao Chunliang, Su Baojun, et al. Research on Markov logic networks [J]. Journal of Software, 2011, 22(8): 1699-1713 (in Chinese)
(徐从富, 郝春亮, 苏保君, 等. 马尔可夫逻辑网络研究[J]. 软件学报, 2011, 22(8): 1699-1713)

[58] Kok S, Domingos P. Statistical predicate invention [C] // Proc of the 24th Int Conf on Machine Learning. New York: ACM, 2007: 433-440

[59] Richardson M, Domingos P. Markov logic networks [J]. Machine Learning, 2006, 62(1/2):107-136

[60] Monge A E, Elkan C P. The field matching problem: Algorithms and applications [C] //Proc of 1996 ACM SIGKDD Conf on Knowledge Discovery & Data Mining. New York: ACM, 1996: 267-270

[61] Navarro G. A guided tour to approximate string matching [J]. ACM Computing Surveys, 2001, 33(1): 31-88

[62] Smite T F, Waterman M S. Identification of common molecular subsequences [J]. Journal of Molecular Biology, 1981, 147(1): 195-197

[63] Waterman M S, Beyer T F S A. Some biological sequence metrics [J]. Advances in Mathematics, 1976, 20(3): 367-387

[64] Winkler W E. Overview of record linkage and current research directions, RRS 2006/02 [R]. Washington DC: US Bureau of the Census, 2006

[65] Ananthakrishna R, Chaudhuri S, Ganti V. Eliminating fuzzy duplicates in data warehouses [C] //Proc of the VLDB Endowment. New York: ACM, 2002: 586-597

[66] Naumann F, Weis M. DogmatiX tracks down duplicates in XML [C] //Proc of the 2005 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2005: 431-442

[67] Cohen W W, Ravikumar P, Fienberg S E. A comparison of string distance metrics for name-matching tasks [C] //Proc of the IJCAI-2003 Workshop on Information on the Web. San Francisco, CA: Morgan Kaufmann, 2003: 73-78

[68] Vazquez A. Growing networks with local rules: Preferential attachment, clustering hierarchy and degree correlations [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2003, 67(5): 369-384

[69] Lada A A, Eytan A. Friends and neighbors on the Web [J]. Social Networks, 2001, 25(3): 211-230

[70] Katz L. A new status index derived from sociometric analysis [J]. Psychometrika, 1953, 18(1): 39-43

[71] Jeh G, Widom J. SimRank: A measure of structural-context similarity [C] //Proc of the 2002 ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2002: 538-543

[72] Christen P. A survey of indexing techniques for scalable record linkage and deduplication [J]. IEEE Trans on Knowledge and Data Engineering, 2012, 24(9): 1537-1555

[73] Hernandez M, Hern'Andez M A, Stolfo S. The merge/purge problem for large databases [C] //Proc of 1995 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 1995: 127-138

[74] Christen P. Towards parameter-free blocking for scalable record linkage [R]. Canberra, Australia: Australian National University, 2007

[75] Yan S, Lee D, Kan M Y, et al. Adaptive sorted neighborhood methods for efficient record linkage [C] //Proc of Int Conf on Digital Libraries. New York: ACM, 2007: 185-194

[76] Draisbach U, Naumann F. A generalization of blocking and windowing algorithms for duplicate detection [C] //Proc of 2011 Int Conf on Data and Knowledge Engineering (ICDKE 2011). Piscataway, NJ: IEEE, 2011: 18-24

[77] Baxter R, Christen P. A comparison of fast blocking methods for record Linkage [C] //Proc of ACM SIGKDD Workshop on Data Cleaning, Record Linkage and Object Consolidation. New York: ACM, 2003: 25-27

[78] Aizawa A, Oyama K. A fast Linkage detection scheme for multi-source information integration [C] //Proc of Int Workshop Challenges in Web Information Retrieval and Integration. Piscataway, NJ: IEEE, 2005: 30-39

[79] Broder A Z. On the resemblance and containment of documents [C] //Proc of the Int Conf on Compression and Complexity of Sequences. Piscataway, NJ: IEEE, 1997: 21-29

[80] Kim H, Lee D. Harra: Fast iterative hashed record linkage for large-scale data collections [C] //Proc of the 2000 ACM SIGKDD Conf on Int Conf on Extending Database Technology. New York: ACM, 2010: 525-536

[81] Mccallum A, Nigam K, Ungar L H. Efficient clustering of high-dimensional data sets with application to reference matching [C] //Proc of the 2000 ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2000: 169-178

[82] Christen P. Development and user experiences of an open source data cleaning, deduplication and record linkage system [J]. ACM SIGKDD Explorations Newsletter, 2009, 11(11): 39-48

[83] Liang J, Chen L, Mehrotra S. Efficient record linkage in large data sets [C] //Proc of the 8th Int Conf on Database Systems for Advanced Applications. Piscataway, NJ: IEEE, 2003: 137-146

[84] Noha A. Efficient record linkage using a double embedding scheme [C] //Proc of 2009 IEEE Int Conf on Data Mining (ICDM 2009). Piscataway, NJ: IEEE, 2009: 274-281

[85] Aggarwal C C, Yu P S. The IGrid index: Reversing the dimensionality curse for similarity indexing in high dimensional space [C] //Proc of the 2000 ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2000: 119-129

[86] Borthwick A, Goldberg A, Cheung P, et al. Batch automated blocking and record matching: US, US 7899796 B1 [P]. 2011-03-01

[87] Mcneill W P, Kardes H, Borthwick A. Dynamic record blocking: Efficient linking of massive databases in MapReduce [C] //Proc of 2012 Int Workshop on Quality in DataBases. New York: ACM, 2012: 1-7

[88] Bollacker K, Cook R, Tufts P. Freebase: A shared database of structured general human knowledge [C] //Proc of the 22nd AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2007: 1962-1963

[89] Bizer C, Lehmann J, Kobilarov G, et al. DBpedia—A crystallization point for the Web of data [J]. Web Semantics Science Services & Agents on the World Wide Web, 2009, 7 (3): 154-165

[90] Suchanek F M, Kasneci G, Weikum G. YAGO: A core of semantic knowledge [C] //Proc of the 2007 Int Conf on World Wide Web. Berlin: Springer, 2007: 697-706

[91] Suchanek F M, Kasneci G, Weikum G. YAGO: A large ontology from wikipedia and wordnet [J]. Web Semantics Science Services & Agents on the World Wide Web, 2007, 6 (3): 203-217

[92] Wang Yuanzhuo, Jia Yantao, Liu Dawei, et al. Open Web knowledge aided information search and data mining [J]. Journal of Computer Research and Development, 2015, 52 (2): 456-474 (in Chinese)
(王元卓, 贾岩涛, 刘大伟, 等. 基于开放网络知识的信息检索与数据挖掘[J]. 计算机研究与发展, 2015, 52(2):456-474)

[93] Shen Wei, Wang Jianyong, Han Jiawei. Entity Linking with a knowledge base: Issues, techniques, and solutions [J]. IEEE Trans on Knowledge & Data Engineering, 2015, 27 (2): 443-460.

[94] Otero-Cerdeira L, Rodríguez-Martínez F J, Gómez-Rodríguez A. Ontology matching: A literature review [J]. Expert Systems with Applications, 2015, 42(2): 949-971

[95] Kirsten T, Kolb L, Hartung M, et al. Data partitioning for parallel entity matching [J]. Proceedings of the VLDB Endowment, 2010, 3(2): 1-8

[96] Dal Bianco G, Galante R, Heuser C A. A fast approach for parallel deduplication on multicore processors [C] //Proc of 2011 ACM Symp on Applied Computing. New York: ACM, 2011: 1027-1032

[97] Kim H, Lee D. Parallel linkage [C] //Proc of the 2007 ACM Int Conf on Information and Knowledge Management. New York: ACM, 2007: 283-292

[98] Bhattacharya I, Getoor L. Query-time entity resolution [C] //Proc of the 2006 ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2006: 529-534

[99] Christen P, Gayler R. Towards scalable real-time entity resolution using a similarity-aware inverted index approach [C] //Proc of the 7th Australasian Data Mining Conf. Sydney, NSW, Australia: Australian Computer Society, 2008: 51-60

[100] Christen P, Gayler R, Hawking D. Similarity-aware indexing for real-time entity resolution [C] //Proc of the 2009 ACM Int Conf on Information and Knowledge Management. New York: ACM, 2009: 1565-1568

[101] Wang J, Kraska T, Franklin M J, et al. CrowdER: Crowdsourcing entity resolution [J]. Proceedings of the VLDB Endowment, 2012, 5(11): 1483-1494

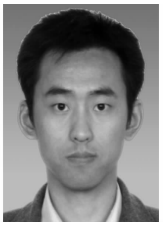
[102] Vesdapunt N, Bellare K, Dalvi N. Crowdsourcing algorithms for entity resolution [J]. Proceedings of the VLDB Endowment, 2014, 7(12): 1071-1082

[103] Demartini G, Difallah D E, Cudré-Mauroux P. Large-scale linked data integration using probabilistic reasoning and crowdsourcing [J]. VLDB Journal, 2013, 22(5): 665-687

[104] Demartini G, Difallah D E, Cudré-Mauroux P. ZenCrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking [C] //Proc of the 2012 Int Conf on World Wide Web. New York: ACM, 2012: 469-478

[105] Wang Z, Li J, Wang Z, et al. Cross-lingual knowledge linking across wiki knowledge bases [C] //Proc of the 2012 Int Conf on World Wide Web. New York: ACM, 2012: 459-468

[106] Wang Z, Li J, Tang J. Boosting cross-lingual knowledge linking via concept annotation [C] //Proc of the 23rd Int Joint Conf on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 2013: 2733-2739



Zhuang Yan, born in 1979. PhD candidate. His main research interests include knowledge base, database, and data cleaning and integration.

Li Guoliang, born in 1980. Associate professor in Tsinghua University. His main research interests include large-scale data management, knowledge base, and crowd computing.

Feng Jianhua, born in 1967. Professor in Tsinghua University. His main research interests include big data, privacy, and knowledge base.