doi：10.13413/j.cnki.jdxblxb.2023062

基于节点语义相似度的本体映射方法

何 杰，王佳蓉，王恒恒
（宁夏大学 地理科学与规划学院，银川 750021）

摘要：针对本体映射特别是大尺度的异构本体映射由于语义异质性导致的映射精度和效率较低的问题，提出一种基于节点语义相似度的本体映射方法. 首先，研究基于网络的本体解析和表示、本体自动分块、相似子本体快速识别、基于节点语义的子本体映射等关键技术；其次，以本体对齐评估倡议评估数据集中会议本体集进行实验，结果表明，该方法在性能上优于传统映射方法，在精度上高于基于片段的映射方法.

关键词：语义相似度；本体映射；本体分块；本体对齐估计倡议；精度；效率
中图分类号：TP208 **文献标志码**：A **文章编号**：1671-5489(2024)02-0399-11

Ontology Mapping Method Based on Node Semantic Similarity

HE Jie, WANG Jiarong, WANG Hengheng
(*School of Geography and Planning, Ningxia University, Yinchuan* 750021, *China*)

Abstract: Aiming at the problem of low mapping accuracy and efficiency caused by semantic heterogeneity in ontology mapping, especially in large-scale heterogeneous ontology mapping, we proposed an ontology mapping method based on node semantic similarity (NSS). Firstly, we studied key technologies such as web-based ontology parsing and representation, automatic ontology partitioning, rapid recognition of similar sub ontologies, and node semantic based sub ontology mapping. Secondly, the experiments were conducted on the conference ontology set in the ontology alignment evaluation initiative (OAEI) evaluation datasets. The results show that the proposed method outperforms traditional mapping methods in performance and has higher accuracy than fragment based mapping methods.

Keywords: semantic similarity; ontology mapping; ontology partition; ontology alignment evaluation initiative; precision; efficiency

0 引 言

本体(ontology)是指对共享概念模型的明确形式化规范说明[1]. 本体在解决实现知识共享与重用等方面的语义异构问题时具有重要作用，已被广泛应用于语义网、数据检索、异构数据集成与融合及工农业、旅游、生物、地理信息等领域中[2-7]. 通过建立关联本体元素(概念、关系、实例)间的语义映射关系，即本体映射实现异构本体间的信息共享和集成. 目前，关于本体映射(或称本体匹配、本体配准)的研究已取得了许多成果[8-11]. 由于本体应用范围和本体规模的增大，导致不同领域、甚至同一

收稿日期：2023-02-23.

第一作者简介：何 杰(1978—)，男，汉族，博士，副教授，从事传感网与网络地理信息系统的研究，E-mail：459195435@qq.com.

基金项目：国家自然科学基金(批准号：42061062)和宁夏自然科学基金(批准号：2022AAC03054).

领域的不同专家构建的本体在语法、结构、语义上存在差异，这些差异导致了本体映射精度和效率的下降，阻碍了本体间信息共享. 文献[12-14]对大规模本体映射系统、解决方法和存在问题进行了详细阐述，为基于分治思想的模块化方法[15-20]在大规模本体映射中的应用提供了依据. 模块化本体映射技术逐渐成为解决大尺度本体映射问题的主流技术，当前围绕这一技术的研究主要包括：1) 通过在本体映射中引入并行运算技术提高映射性能和质量. 如 Zhang 等[21]提出了一种基于 MapReduce 框架和虚拟文档技术的大规模本体匹配方法，通过三阶段 MapReduce 的运用使得匹配性能、精度和召回率提升，不足之处是该方法未考虑到本体结构，算法的健壮性有待提高. 2) 为提高分块的合理性，把不同聚类算法应用到本体分块. 如 Algergawy 等[22]提出了一种可扩展的基于结构聚类的本体映射方法，将大型匹配问题分解为小型匹配问题，利用一种有效的轻量级语言学方法确定每两组聚类之间的相似聚类，以获得一组小的匹配任务，使匹配性能和质量得以提升. 该方法虽然考虑到本体结构，但聚类算法复杂，聚类结果对映射结果影响较大. 蒋猛等[23]提出了一种基于模块化和局部置信度的多策略自适应大规模本体映射算法，对本体内部进行聚类和模块化，基于信息检索策略发现相似子本体，利用映射策略计算子本体相似度，通过启发式贪心策略提取映射结果并基于映射规则矫正结果，不足之处是映射结果对映射规则的依赖性强. 叶霞等[24]提出了一种基于主成分分析和 K-Modes 蚁群聚类的本体映射方法，实现对本体概念映射关系的批量计算，同时提高了映射的准确性，不足之处是算法对样本的适应性有待提高. 3) 把场势函数和动态分块技术引入到本体分块. 如仲茜等[25]提出了一种基于数据场的大规模本体映射算法，根据数据场势函数计算周围本体元素对当前元素的影响不断修正相关度,确定相关子本体，对子本体进行更有效的映射提高映射质量和效率，不足之处是算法中的某些参数及阈值依靠用户反馈的方式进行选择和设置，用户对数据的熟悉程度会对映射效果产生一定的影响. 吕青等[26]提出了一种动态分块的大规模本体匹配方法，该方法保留了每个分块的语义完整性，通过紧凑进化算法优化子匹配任务的阈值和再分配的块标志位，用精英解参与的概率向量更新方式对算法进行改进，提高了匹配质量，但优化算法的性能有待提升，对匹配框架的适应性和可移植性需进一步验证.

此外，为提高本体映射精度，概念语义和实例也应用于本体映射中. 如文献[27]提出了一种改进的语义相似度计算模型，通过计算概念间的语义距离提高映射精度，不足之处是未考虑背景知识，且信息资源标注效率也较低. 文献[28-29]通过利用本体数据实例提高查全率和查准率，但实例相似度计算方法通用性有待验证，由于实例的复杂性导致计算性能降低. 此外，机器学习、层次分析法、多策略本体映射等方法也被用来提高本体映射质量和效率[30-39].

上述本体映射方法从不同方面提高了本体映射的效率和质量，但也存在一定问题，主要包括：1) 大规模本体分块方法通用性较低，不同规模和领域本体分块粒度不易控制，合理性不易验证；2) 本体映射算法虽然提高了映射质量，但算法复杂度普遍较高，适应性较弱，且有的算法用户参与度高，在一定程度影响了映射效率；3) 对本体模式元素包含的语义信息利用不够. 已有的大多数映射算法未考虑本体元素语义信息，因此，本文提出一种基于节点语义相似度的本体映射方法(ontology mapping method based on node semantic similarity, NSS). 首先对待映射的源和目标本体表示图通过一种简单、通用的图(树)分割的方式进行动态分块，然后基于一种轻量级图匹配算法快速发现最相似的子本体块对，最后基于节点语义相似度算法计算子本体块实体间的映射关系，并以本体对齐评估倡议(ontology alignment evaluation initiative)评估数据集本体间的映射实验验证该方法的有效性. 本文方法的主要创新点为：1) 研究利用一种基于树分裂的本体分割算法，解决了大规模本体分块方法通用性较低的问题，算法简单易实现；2) 基于提出的轻量级相似子本体识别算法提高本体映射效率；3) 相似子本体映射算法中综合考虑本体元素节点的语义信息解决同名异义、同义异名问题，有效提高了本体映射精度.

1 系统体系结构

本文提出的 NSS 映射方法目标是尽可能提高本体分块方法的通用性，同时改进本体映射的效率，

并在映射计算中充分利用现有的本体元素概念语义信息和各种辅助知识(如知识库)，图 1 为本文本体映射体系结构. 由图 1 可见，整个系统由本体解析、本体分块、相似子本体识别、子本体映射、相似值组合等 5 个主要部件及相似子本体管理、映射器管理、映射选择器等 3 个辅助部件和 1 个知识库组成. 本体解析器实现把输入的各种格式(XML，OWL，RDF 等)本体模式文件进行解析并生成本体模式的内部表示(图或树)；本体分块部件则通过一种自动分割算法把本体图/树分割成子图/树；相似子本体识别部件通过轻量级的图匹配算法快速发现源和目的本体子图/树中最相似的本体子图/树对，这些相似子图/树对会暂时保存在相似子本体管理模块，相似子本体管理模块可根据子本体映射结果对子本体图/树对进行更新，选择不同的子本体图/树对到映射执行部件进行更精细的匹配；子本体映射部件利用提出的基于节点语义相似度的映射算法计算每对子本体图/树中不同节点(实体)间的语义相似度值，节点语义相似度算法组合了多种映射器，这些映射器由映射管理模块管理；映射管理模块可以为映射执行部件根据不同的映射任务选择不同的映射器或映射器组合，还可以根据映射器评估结果对映射器进行更新；相似值组合部件则通过某种组合方法(如最大值、平均值)计算各不同节点对的最终相似值；映射选择器则根据给定的阈值、给定的参考映射及知识库中的专家知识选择最佳映射并输出.

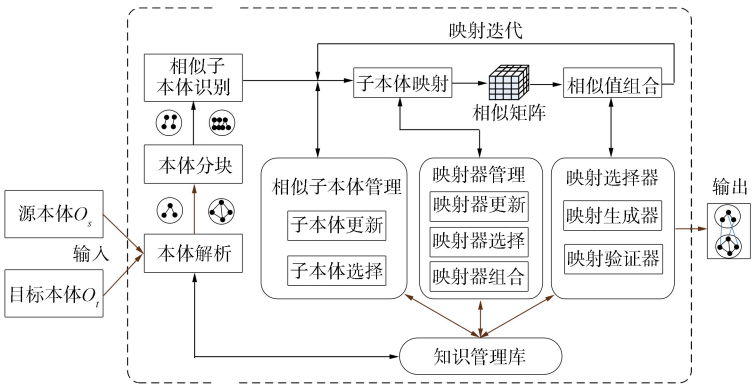


图 1　本体映射体系结构

Fig. 1　Architecture of ontology mapping

2　系统实现

下面介绍 NSS 系统实现的几个关键技术，包括基于网络的本体动态解析、基于树分裂法的本体分块、基于节点概念和结构相似值的相似子本体识别及基于字符相似度和标签概念相似度的子本体映射技术.

2.1　本体解析

网络本体描述语言(OWL)允许一个模式分布在几个文档和名称空间，并通过提供不同定向把在一个文档中定义的部分导入到一个新文档. 为确保本体解析过程中关联文档信息收集的完整性，保证关联文档间关系的正确性，同时为解析后本体模式表示的方便性和正确性，提出一种基于网络的本体动态解析方法，即在本体模式文件解析过程中，首先对分布的相关文档进行解析，并把导入或交叉引用部分导入到一个单独的文档中然后再进行处理. 解析过程为：首先构建本体解析器，建立与本体资源(url)的连接，解析本体资源生成对应的本体对象，然后解析并加载本体所有类；如果概念分层不唯一，则加载本体对应的对象属性和数据属性；加载本体数据类型对象和类公理对象；如果概念分层不唯一，则加载属性公理；对本体进行渲染，如果标注内容不为空，则首先渲染标注内容，然后渲染本体类、数据类型、对象属性、数据属性、标注属性，最后渲染类公理及属性公理，解析过程结束. 解析结束后本体资源在内部统一表示成本体模式图或树的形式.

算法 1　本体模式解析.

输入：本体模式文件(url)；

输出：本体模式表示图 G；

ParseAndRenderOntology(url)

步骤1) parser=getOWLRDFParser( );//构建本体解析器

步骤2) connection=getOWLConnection( );//获取连接对象

步骤3) parser.setConnection(connection);//建立连接

步骤4) owlOntology=parser.parseOntology(url);//解析本体资源文件

步骤5) loadClass(owlOntology);//加载所有本体类

步骤6) if(!conceptHiearchyOnly);//如果概念分层不唯一

步骤7) loadObjectProperty(owlOntology);//加载本体所有对象属性

步骤8) loadDataProperty(owlOntology);//加载本体所有数据属性对象

步骤9) loadDataType(owlOntology);//加载本体所有数据类型对象

步骤10) loadClassAxiom(owlOntology);//加载本体所有类公理对象

步骤11) if(!conceptHiearchyOnly);//如果概念分层不唯一

步骤12) loadPropertyAxiom(owlOntology);//加载本体所有属性公理对象

步骤13) if(!owlOntology.getAnnotations( ));//如果本体标注不为空

步骤14) renderAnnotationContent( );//渲染标注内容

步骤15) renderClass(owlOntology);//渲染本体类

步骤16) renderDataType(owlOntology);//渲染本体数据类型

步骤17) renderObjectProperty(owlOntology);//渲染本体对象属性

步骤18) renderDataProperty(owlOntology);//渲染本体数据属性

步骤19) renderAnnotationProperty(owlOntology);//渲染本体标注属性

步骤20) renderClassAxiom(owlOntology);//渲染本体类公理

步骤21) renderPropertyAxiom(owlOntology).//渲染本体属性公理,算法结束

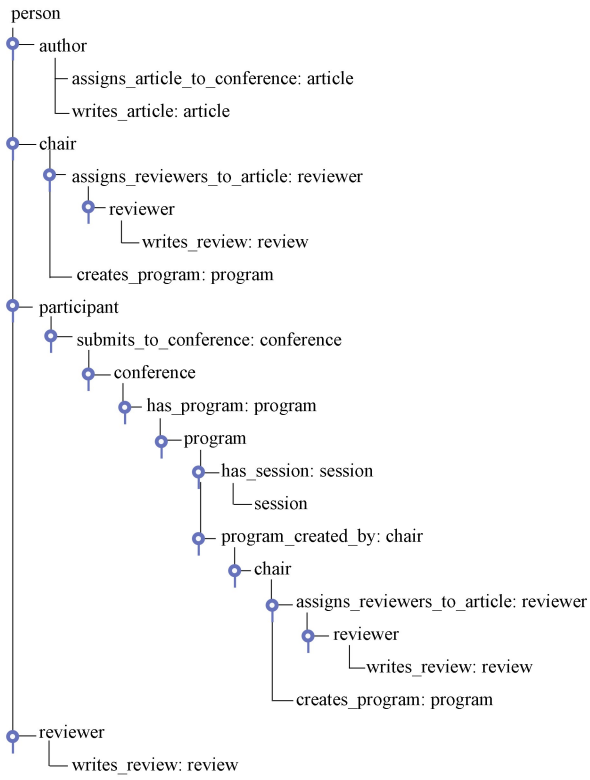
本文以本体对齐评估倡议2021年度评估数据集中会议本体集(https://oaei.ontolgymatching.org/2021/conference/index.html)中的本体模式文件crs_dr.owl为例,图2(A)为该模式文件的解析前owl描述内容部分片段,该部分片段描述了person,author,chair,participant,reviewer等5个类,其中person类是author,chair,participant,reviewer类的父类.图2(B)为其解析后的图表示形式.

2.2 本体分块

为提高本体映射性能、降低本体映射规模和复杂度、减少本体映射过程中的无效映射,在映射前首先对原本体进行分块,然后基于分块后的相似子本体块再执行精确映射.本体分块是基于本体解析后的模式表示图(即一棵倒立的树)进行的,与文献[15]和文献[17]中提出的通过一种分层的聚类算法把本体分割成许多不同的族类不同,本文采用文献[40]中设计的基于树自然分裂的分割算法,与聚类算法相比,该算法简单、高效.算法过程为:分割前先确定源和目标本体,对源本体进行分割,然后基于源本体分割结果分割目标本体.源本体分割过程为:首先计算本体模式树的深度及各节点的入度和出度值,然后判断所有入度为0,即根节点所在树的深度,如果深度值大于3且根节点出度值大于1(避免分割产生孤立节点),则进行1次自然分裂,即删除根节点到各子节点的链接,原来的树自然裂变成几棵独立的子树,重新计算分裂后的子树深度及节点的度,再递归执行分裂算法直到没有可分裂的树时算法结束.由于节点的标签较长,为计算方便,在分割时用节点ID号代替标签表示.ID号由3到4位数字组成,右起第1位表示本节点在该层中位置,第2位表示父亲节点的位置,第3位表示该节点深度,如果深度大于10,则用两位数表示.如ID号为234的节点表示第3层第4号节点,其父亲节点是第2层第3号节点.节点旁标注的是节点度大小,左边表示入度值,右边为出度值.以图2模式片段为例,模式树根节点person编码为001,其子树author,chair,participant和reviewer根节点编码分别为111,112,113,114.为防止分裂次数过多产生大量的子树导致管理上的额外开销,一般情况下分裂次数不超过3次.每次分割后需对节点重新编码.图3为经过重新编码分割后的结果.

```
〈owl: Ontology rdf: about=“”/〉
〈owl: Class rdf:ID=“author”〉
 〈rdfs: subClassOf〉
  〈owl: Class rdf:ID=“person”/〉
 〈/rdfs: subClassOf〉
〈/owl: Class〉
〈owl: Class rdf:ID=“chair”〉
 〈rdfs: subClassOf〉
  〈owl: Class rdf: about=“#person”/〉
 〈/rdfs: subClassOf〉
〈/owl: Class〉
〈owl: Class rdf: ID=“participant”〉
 〈rdfs: subClassOf〉
  〈owl: Class rdf: about=“#person”/〉
 〈/rdfs: subClassOf〉
〈/owl: Class〉
〈owl: Class rdf: about=“#person”〉
 〈owl: disjointWith rdf:resource=“#program”/〉
 〈owl: disjointWith〉
  〈owl: Class rdf: about=“#event”/〉
 〈/owl: disjointWith〉
 〈owl: disjointWith rdf: resource=“#document”/〉
〈/owl: Class〉
〈owl: Class rdf: ID=“reviewer”〉
 〈rdfs: subClassOfrdf: resource=“#person”/〉
〈/owl: Class〉
```

(A)



(B)

图 2　crs_dr. owl 模式片段（A）及其表示（B）

Fig. 2　Schema fragment（A）of crs_dr. owl and its representation（B）

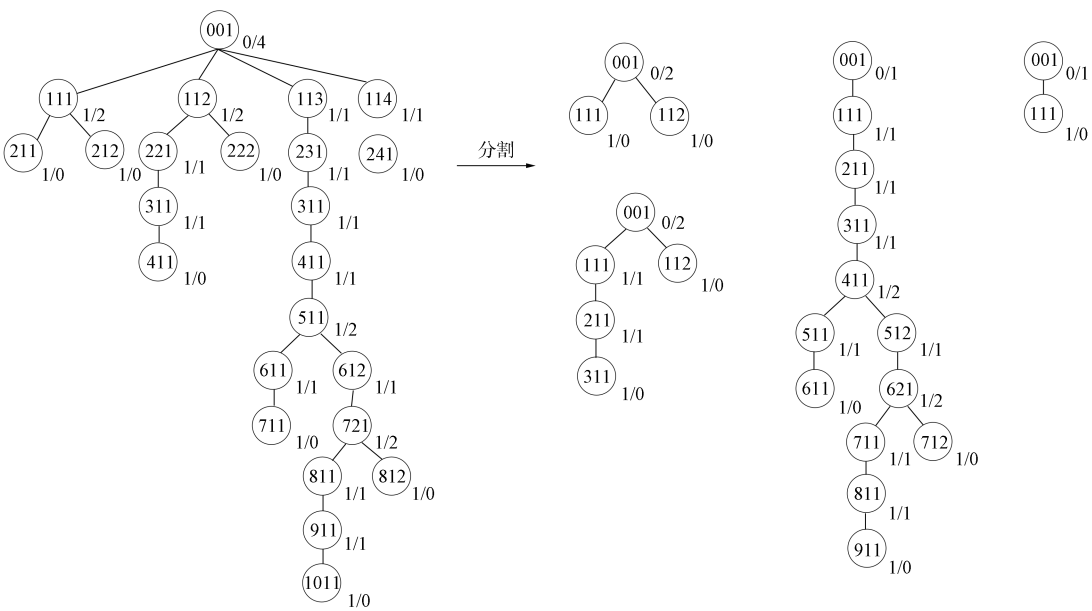


图 3　crs_dr. owl 模式片段分割结果

Fig. 3　Segmentation results of schema fragment of crs_dr. owl

2. 3　相似子本体识别

相似子本体识别的任务主要是发现经过分块后源和目的模式片段中所有相似子片段对，该识别过程是本体映射的粗匹配，只有相似子本体对才会输入到本体映射部件进行精匹配. 为提高相似子本体识别的效率同时保证不会丢失相似子本体对，本文在计算两个子本体块的相似值时只考虑它们根节点

的相似度和子本体的结构相似度的综合相似值，同时，只计算相似子本体对间1∶1的关联关系. 定义两个子本体对间相似值 $\mathrm{sim}(o_1,o_2)$ 为

$$\mathrm{sim}(o_1,o_2)=\alpha\times\mathrm{sim}_{\mathrm{str}}(o_1.\mathrm{Name},o_2.\mathrm{Name})+\beta\times \mathrm{sim}_{syn}(o_1.\mathrm{Name},o_2.\mathrm{Name})+\gamma\times\mathrm{sim}_{\mathrm{cont}}(o_1,o_2), \tag{1}$$

其中：$\mathrm{sim}_{\mathrm{str}}(o_1.\mathrm{Name},o_2.\mathrm{Name})$ 计算两个子本体片段树根节点标签的字符相似值，本文采用编辑距离算法计算字符相似值，即

$$\mathrm{sim}_{\mathrm{str}}(o_1.\mathrm{Name},o_2.\mathrm{Name})=1-\mathrm{editNums}/\mathrm{maxLength}(o_1.\mathrm{Name},o_2.\mathrm{Name}), \tag{2}$$

式中 editNums 为把一个字符转换为另一个字符需要的编辑次数，maxLength( )为返回字符串的最大长度；$\mathrm{sim}_{\mathrm{syn}}(o_1.\mathrm{Name},o_2.\mathrm{Name})$ 计算两个子本体片段树根节点标签的概念相似值，计算方法如下.

定义1(标签义词相似值) 设有标签 A,B，其在 WordNet 中的同义词集合表示为 $A=\{\alpha_1,\alpha_2,\cdots,\alpha_n\}$，$B=\{\beta_1,\beta_2,\cdots,\beta_m\}$，$\alpha_1,\alpha_2,\cdots,\alpha_n$，$\beta_1,\beta_2,\cdots,\beta_m$ 分别为标签 A,B 的原子义词，则标签 A,B 的内积为

$$(A,B)=\sum_{i=1}^{n}\sum_{j=1}^{m}\mathrm{sim}_{\mathrm{asyn}}(\alpha_i,\beta_j), \tag{3}$$

其中 $\mathrm{sim}_{\mathrm{asyn}}(\alpha_i,\beta_j)$ 表示根据两个原子义词在 WordNet 中的词义距离的倒数计算出的义词相似值. 根据内积，即能导出标签的范数和标签的相似值定义.

定义2(标签范数和概念相似值) 设有标签 A，其范数定义为

$$\|A\|=\sqrt{(A,A)}, \tag{4}$$

则两个标签 A,B 间的概念相似值定义为

$$\mathrm{sim}_{\mathrm{syn}}(A,B)=\frac{(A,B)}{\|A\|\cdot\|B\|}=\frac{(A,B)}{\sqrt{(A,A)\cdot(B,B)}}. \tag{5}$$

式(1)中 $\mathrm{sim}_{\mathrm{cont}}(o_1,o_2)$ 用来计算两个子本体片段树结构相似值，定义为

$$\mathrm{sim}_{\mathrm{cont}}(o_1,o_2)=\frac{|C_{o_1}\cap C_{o_2}|}{\sqrt{|C_{o_1}|\cdot|C_{o_2}|}}, \tag{6}$$

其中：$|C_{o_1}\cap C_{o_2}|$ 表示两个子片段树包含的相同节点个数；$\sqrt{|C_{o_1}|\cdot|C_{o_2}|}$ 表示两个子片段树上下文节点数量的几何平均值，用来归一化两个子片段树的结构相似值. 假设子片段树 o_1 包含 v_1,v_2,v_3,v_4 四个节点，o_2 包含 v_1,v_2,v_3,v_5 四个节点，则 $C_{o_1}=\{v_1,v_2,v_3,v_4\}$，$C_{o_2}=\{v_1,v_2,v_3,v_5\}$，根据式(6)得到两个子树的结构相似值为 $\mathrm{sim}_{\mathrm{cont}}(o_1,o_2)=3/4=0.75$.

在两个子片段树的相似值计算中，不同类型相似值对整个片段树相似度的贡献度不同，如对于标签的字符相似值，由于存在同名异义和同义异名等现象，使标签的字符相似值并不能准确表示两个子标签的相似程度，所以对不同类型相似值赋予不同权重. 根据经验，标签概念相似度贡献最大，其次是结构相似值，所以式(1)中的 α,β,γ 取值满足 $\alpha<\gamma\leqslant\beta$，且 $\alpha+\beta+\gamma=1$.

2.4 子本体映射

子本体映射的执行是基于识别后的相似子本体片段对. 如果两个相似片段对各有 n 个节点，则进行相似比较的次数达到 n^2 次，即算法的时间复杂度为 $O(n^2)$. 根据子本体树结构相似值计算方法可知，一对相似子本体树中所有节点不一定都相似，有部分完全相似的公共节点，这些公共节点在相似值计算时不再需要与其他节点进行比较，因此本文提出一种基于节点语义相似度的本体映射方法. 首先把相似片段对的公共节点作为锚节点或称相似度已知节点(相似度值为1.0)，这些节点不参加相似度计算. 如果 n 个节点中有 m 个锚节点，则优化后算法复杂度为 $O((n-m)^2)$，如果有1/2的公共节点，则优化后算法的复杂度为原来的1/4；然后计算非锚节点相似度，在式(1)的基础上进行简化，即节点的相似度为两个节点的标签字符相似度和标签概念相似度的综合值，表示为

$$\mathrm{sim}(o_s,o_t)=\alpha\times\mathrm{sim}_{\mathrm{str}}(o_s.\mathrm{Name},o_t.\mathrm{Name})+\beta\times\mathrm{sim}_{\mathrm{syn}}(o_s.\mathrm{Name},o_t.\mathrm{Name}). \tag{7}$$

式(7)中字符相似度和标签概念相似度值分别使用式(2)和式(5)计算，权重系数 α,β 取值满足 $\alpha<\beta$，且 $\alpha+\beta=1$. 本文中 $\alpha=0.4$，$\beta=0.6$. 当源子本体树中一个节点与对应的目标子本体树中所有节点相似度

值计算完成后，从这组相似值中选择相似值最大、且大于预先设定门限值的两个节点作为匹配的候选映射. 当源子本体树中所有节点与目标本体树中所有节点相似度计算完毕时，则该相似子本体对映射结束，从相似子本体管理器中选择下一个相似子本体对进行映射，直到所有相似子本体对映射结束. 算法的伪代码描述如下.

算法 2　相似子本体映射.

输入：候选相似子本体树对(o_s,o_t)；

输出：候选映射 Mappings；

SubOntologyMapping(o_s,o_t)

步骤 1) OntologyPair$\langle o_{s_i},o_{t_j}\rangle$=getSubOntologyPair( )；//从子本体管理器选择待匹配子本体对

步骤 2) foreach (Node n_s in o_{s_i})

步骤 3) if (!anchor(n_s))；//如果节点不是锚节点

步骤 4) foreach (Node n_t in o_{t_j})

步骤 5) if (!anchor(n_t))；//如果节点不是锚节点

步骤 6) s_i=getSimilarity(n_s,n_t)；//计算两个节点相似度

步骤 7) if (MaxSimilarity$<s_i$)//MaxSimilarity 初始值设为 0，比较其与 s_i 大小

步骤 8) MaxSimilarity=s_i；//更新 MaxSimilarity 值

步骤 9) if (MaxSimilarity>threshold)；//如果源节点 n_s 与所有目标节点最大相似值大于门限值

步骤 10) AddMapping (Mapping$\langle n_s,n_t\rangle$)；//相似值最大的节点对保存为候选映射

步骤 11) goto 1)；//返回步骤 1)继续计算下一组子本体对中的映射

步骤 12) return Mappings. //所有子本体对映射计算完毕，返回所有候选映射，算法结束

3　实验及讨论

下面通过实验验证本文方法在映射性能和精度上的表现，通过实验综合评估提出方法的有效性及存在的不足. 实验使用的计算机配置为：Windows11 64 位操作系统，Intel(R) Core(TM) i7-1260P 处理器，16 GB 运行内存，Sun Java 1.8.0 运行库. 实验数据以本体对齐评估倡议 2021 年度评估数据集中会议本体集中的本体模式文件为例，其中源本体模式文件为 conference.owl，目标本体模式文件包括：confOf.owl,edas.owl,ekaw.owl,iasted.owl,sigkdd.owl，生成的 5 组对应映射结果分别命名为 conf-confOf,conf-edas,conf-ekaw,conf-iasted,conf-sigkdd. 实验任务包括本体映射质量实验和本体映射性能实验.

3.1　本体映射质量实验

为验证本文方法在映射精度上的改进，用通用的模式和本体匹配系统 COMA++[41] 中定义的两种映射方法：AllContext(完全上下文)和 Fragment-based(基于片段)及本文方法(NSS)分别进行映射实验. 由于 COMA++本体匹配系统通用、成熟，该系统中的 AllContext 方法综合了各种匹配器，能产生较高的映射精度，而 Fragment-based 方法又采用了基于分治思想的模块化方法，映射效率高，所以选择这两种方法与本文方法进行实验验证提出方法的有效性. 对 COMA++两种映射方法使用系统默认配置，对 NSS 选择的配置策略为：相似值组合使用加权平均值法，匹配候选者选择使用最大值法，相似子本体识别选定的门限值为 0.6.

本体映射质量采用通用的查全率(recall)和精度(precision)评估，实验结果分别如图 4 和图 5 所示. 由图 4 可见，基于 5 组对应映射数据集应用 3 种映射方法的平均查全率超过了 88%，其中映射 conf-iasted 查全率最高，conf-ekaw 查全率最低，而对于同一组映射，NSS 方法平均查全率最高，AllContext 次之，Frag-based 最低，NSS 方法的平均查全率比 AllContext 高约 2%，比 Frag-based 高约 5%. 由图 5 可见，用 3 种映射方法实验的 5 组映射结果的平均精度达到了 85%，其中映射 conf-iasted精度最高，conf-ekaw 精度最低，而对于同一组映射，NSS 方法平均精度最高，其比 AllContext 方法的平均精度高约 3%以上，比 Frag-based 方法的平均精度高约 6%. 这主要由于：

1) NSS方法使用了分块的方法减少了无效节点的相似度比较，同时降低了映射本体本身结构的复杂度；2) NSS方法由于考虑了节点的语义信息，从而在一定程度提高了节点相似度计算精度；3) AllContext方法和Frag-based方法相比，由于前者综合利用了NAME，PATH，LEAVES和PARENTS等多种匹配器组合计算相似度，且由于Frag-based方法需要人工确定映射本体片段对，如果人工选择有误将影响映射结果，所以Frag-based方法在查全率和精度上均落后于AllContext方法；由于conf-ekaw包含的映射数量最多，conf-iasted映射最少，导致前者计算资源消耗和复杂性比后者高，在一定程度影响了映射质量.

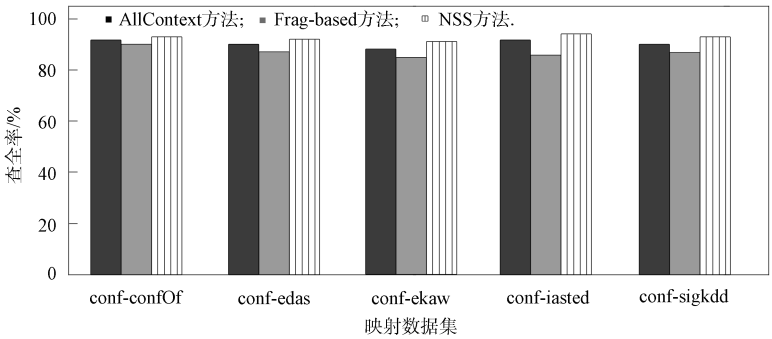


图4 不同方法的匹配查全率比较

Fig. 4 Comparison of matching recall rates of different methods

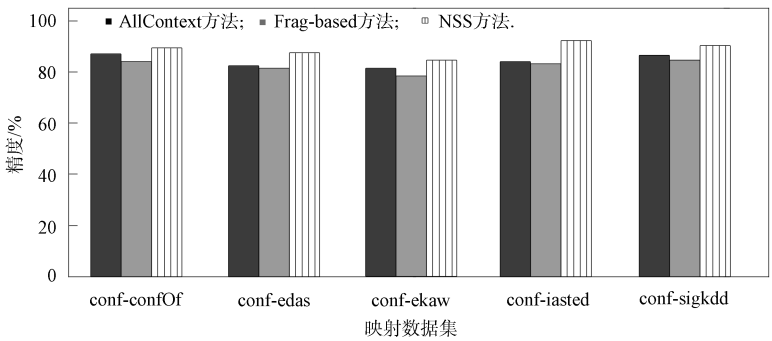


图5 不同方法的匹配精度比较

Fig. 5 Comparison of matching precision of different methods

3.2 本体映射性能实验

为验证本文方法在性能上的优势，基于上述实验数据和3种映射方法，进一步实验比较这3种方法的性能. 3种方法中，NSS和Frag-based方法均采用了分块的策略，区别在于后者需要人工选择需要进行映射的分块，而NSS则是自动对本体模式图进行分块并自动确定候选映射块对，AllContext方法未对模式进行分割，而是通过综合利用名称、路径、叶子、父亲等匹配器从节点标签字符串及所在上下文结构上对源和目的模式进行整体映射. 在给定的5组映射中，conf-ekaw映射节点最多，conf-iasted映射节点最少，5组映射节点数量关系为：conf-ekaw＞conf-edas＞conf-confOf＝conf-sigkdd＞conf-iasted. 3种映射方法应用与上述相同的参数配置进行实验，实验结果如图6所示. 由图6可见，每组映射所耗费的时间一方面与映射节点数量成正比，另一方面，不同组映射及同一组映射应用不同方法耗时各不相同. 对于同组映射任务，NSS方法平均耗时最少，性能最高，Frag-based次之，AllContext最差. NSS方法平均性能比Frag-based方法高约5%，比AllContext方法高约8%. 这主要是由于NSS和Frag-based方法均应用了分块的思想，本体映射的规模和复杂度都进一步降低，提升了性能，同时与Frag-based方法手工确定分块不同，NSS方法在本体分块及相似本体映射过程执行的自动化程度更高，所以性能占优. 对于同种映射方法应用于不同组映射任务，映射conf-iasted平均耗时最少，conf-ekaw平均耗时最多. 这与不同映射任务的工作量有关，映射conf-iasted需要计算的映射节点最少，而conf-ekaw最多，且结构也比前者复杂，导致性能下降.

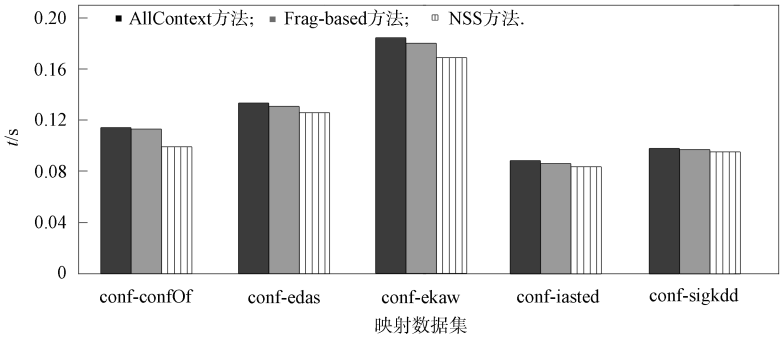


图 6　3 种匹配方法性能比较

Fig. 6　Performance comparison of three matching methods

综上所述，本体虽然在解决知识共享与重用等方面的语义异构问题发挥了重要作用，但由于本体通常是面向领域的，不同领域专家甚至同一领域的不同专家在构建本体时会存在语法和结构上的差异，同时元素语义表达不相同，导致本体映射精度和性能下降. 为改善本体映射性能和精度，本文在分析当前各类本体映射方法及本体模式结构和元素特征的基础上，提出了一种基于节点语义相似度的本体映射方法. 通过本体分块把本体映射问题转化为子本体片段间的映射问题，并在节点比较时通过引入锚的机制进一步减少了本体映射规模，同时在局部映射过程中充分考虑本体元素的标签语义信息. 基于 OAEI 评估数据集的实验结果表明，本文方法在匹配性能上比传统本体映射方法，如 COMA＋＋的 AllContext 方法，有较大提高，同时在精度上优于 Frag-based 方法.

参　考　文　献

[1] STUDER B, BENJAMINS V R, FENSEL D. Knowledge Engineering: Principlesand Methods [J]. Data and Knowledge Engineering, 1998, 25(1/2): 161-197.

[2] 王向前，张宝隆，李慧宗. 本体研究综述 [J]. 情报杂志，2016，35(6)：163-170. (WANG X Q, ZHANG B L, LI H Z. Review of Ontology Research [J]. Journal of Information, 2016, 35(6): 163-170.)

[3] ZHAO L H, ICHISE R. Ontology Integration for Linked Data [J]. Journal on Data Semantics, 2014, 3(4): 237-254.

[4] XU H, ZHANG R. Research on Data Integration of the Semantic Web Based on Ontology Learning Technology [J]. Journal of Electrical Engineering and Computer Science, 2014, 12(1): 167-178.

[5] LAPSHIN E. Levels of Rresenting Ontologies, Languages, Mathematical Models, and Ontology Web-Server Project in Web 2.0 [J]. Automatic Documentation & Mathematical Linguistics, 2012, 46(2): 59-67.

[6] KANG Y B, SHONALI K, WUDHICHART S, et al. Understanding and Improving Ontology Reasoning Efficiency through Learning and Ranking [J]. Information Systems, 2020, 87: 101412-1-101412-17.

[7] VERHOOSEL J P, BEKKUM M V, EVERT F K. Ontology Matching for Big Data Applications in the Smart Dairy Farming Domain [C]//10th International Workshop on Ontology Matching. [S. l.]: CEUR-WS, 2015: 55-59.

[8] 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.

[9] SHVAIKO P, EUZENAT J. Ontology Matching: State of the Art and Future Challenges [J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(1): 158-176.

[10] 王顺，周康达，江东宇. 本体映射综述 [J]. 计算机科学，2017，44(9)：1-10. (WANG S, ZHOU K D, JIANG D Y. Overview of Ontology Mapping [J]. Computer Science, 2017, 44(9): 1-10.)

[11] 宋朋. 本体映射的研究综述 [J]. 图书馆学研究，2016，14(5)：17-21. (SONG P. A Review of Ontology Mapping [J]. Library Science Research, 2016, 14(5): 17-21.)

[12] BABALOU S, KARGAR M J, DAVARPANAH S H. Large-Scale Ontology Matching: A Review of the

Literature [C]//Second International Conference on Web Research. Piscataway, NJ: IEEE, 2016: 158-165.

[13] OCHIENG P, KYANDA S. Large-Scale Ontology Matching: State-of-the-Art Analysis [J]. ACM Computing Surveys 2018, 51(4): 75-1-75-35.

[14] PETER O, SWAIB K. Large-Scale Ontology Matching [J]. ACM Computing Surveys (CSUR), 2018, 51(4): 75-1-75-35.

[15] PATEL A, JAIN S. A Partition Based Framework for Large Scale Ontology Matching [J]. Recent Patents on Engineering, 2020, 14(3): 488-501.

[16] RAHM E. Towards Large-Scale Schema and Ontology Matching [C]//Schema Matching and Mapping. Berlin: Springer, 2011: 3-27.

[17] XUE X, PAN J S. A Segment-Based Approach for Large-Scale Ontology Matching [J]. Knowledge and Information Systems, 2017, 52(2): 1-18.

[18] 陈恒, 李冠宇, 陈鑫影. 模块化思想在大规模本体匹配中的应用 [J]. 计算机工程与应用, 2017, 53(8): 149-153. (CHEN H, LI G Y, CHEN X Y. Application of Modularization in Large-Scale Ontology Matching [J]. Computer Engineering and Application, 2017, 53(8): 149-153.)

[19] 徐德智, 赖雅, 王润梅. 大规模本体分块与映射研究 [J]. 小型微型计算机系统, 2012, 33(3): 542-547. (XU D Z, LAI Y, WANG R M. Research on Large-Scale Ontology Partitioning and Mapping [J]. Small Microcomputer System, 2012, 33(3): 542-547.)

[20] 孙煜飞, 马良荔, 郭晓明, 等. 基于模块化的大规模本体映射方法 [J]. 模式识别与人工智能, 2016, 29(5): 410-416. (SUN Y F, MA L L, GUO X M, et al. Large Scale Ontology Mapping Method Based on Modularization [J]. Pattern Recognition and Artificial Intelligence, 2016, 29(5): 410-416.)

[21] ZHANG H, HU W, QU Y Z. VDoc+: A Virtual Document Based Approach for Matching Large Ontologies Using MapReduce [J]. Journal of Zhejiang University—Science C (Computer & Electronics), 2012, 13(4): 257-267.

[22] ALGERGAWY A, MASSMANN S, RAHM E. A Clustering-Based Approach for Large-Scale Ontology Matching [C]//Proceedings of the 15th International Conference on Advances in Databases and Information Systems. Berlin: Springer-Verlag, 2011: 415-428.

[23] 蒋猛, 禹明刚, 王智学. 多策略自适应大规模本体映射算法 [J]. 计算机工程, 2019, 45(3): 14-19. (JIANG M, YU M G, WANG Z X. Multi Strategy Adaptive Large-Scale Ontology Mapping Algorithm [J]. Computer Engineering, 2019, 45(3): 14-19.)

[24] 叶霞, 许飞翔, 曹军博, 等. 基于主成分分析和 K-Modes 蚁群聚类的本体映射方法 [J]. 计算机应用与软件, 2020, 37(12): 231-237. (YE X, XU F X, CAO J B, et al. Ontology Mapping Method Based on Principal Component Analysis and K-Modes Ant Colony Clustering [J]. Computer Applications and Software, 2020, 37(12): 231-237.)

[25] 仲茜, 李涓子, 唐杰, 等. 基于数据场的大规模本体映射 [J]. 计算机学报, 2010, 33(6): 955-965. (ZHONG Q, LI J Z, TANG J, et al. Large Scale Ontology Mapping Based on Data Field [J]. Journal of Computer Science, 2010, 33(6): 955-965.)

[26] 吕青, 周欣, 李凤莲. 动态分块调节机制下的大规模解剖学本体匹配 [J]. 计算机应用研究, 2023, 40(1): 1-6. (LÜ Q, ZHOU X, LI F L. Large Scale Anatomical Ontology Matching under Dynamic Block Regulation Mechanism [J]. Computer Application Research, 2023, 40(1): 1-6.)

[27] 刘春辰, 刘大有, 王生生, 等. 改进的语义相似度计算模型及应用 [J]. 吉林大学学报(工学版), 2009, 39(1): 119-123. (LIU C C, LIU D Y, WANG S S, et al. Improved Semantic Similarity Calculation Model and Its Application [J]. Journal of Jilin University (Engineering and Technology Edition), 2009, 39(1): 119-123.)

[28] SCHOPMAN B, WANG S H, ISAAC A, et al. Instance-Based Ontology Matching by Instance Enrichment [J]. Journal on Data Semantics, 2012, 1(4): 219-236.

[29] NATH R, SEDDIQUI H, AONO M. An Efficient and Scalable Approach for Ontology Instance Matching [J]. Journal of Computers, 2014, 9(8): 1755-1768.

[30] 王汉博，孙启霖. 基于路径特征的复杂本体匹配 [J]. 计算机工程，2017，43(2)：227-233.（WANG H B，SUN Q L. Complex Ontology Matching Based on Path Features [J]. Computer Engineering，2017，43(2)：227-233.）

[31] 戴大蒙，慕德俊，吴佳斌. 基于概念分层的本体组合匹配策略研究 [J]. 西北工业大学学报，2013，31(1)：14-18.（DAI D M，MU D J，WU J B. Research on Ontology Combination Matching Strategy Based on Concept Stratification [J]. Journal of Northwestern Polytechnical University，2013，31(1)：14-18.）

[32] 徐德智，易晓媛，汤哲. 基于 AHP-熵权决策的本体映射优化算法 [J]. 微电子学与计算机，2017，34(11)：48-52.（XU D Z，YI X Y，TANG Z. Ontology Mapping Optimization Algorithm Based on AHP Entropy Weight Decision [J]. Microelectronics and Computer，2017，34(11)：48-52.）

[33] 孙煜飞，马良荔，周润芝. 一种自适应的多策略本体映射方法 [J]. 海军工程大学学报，2016，28(2)：75-80.（SUN Y F，MA L L，ZHOU R Z. An Adaptive Multistrategy Ontology Mapping Method [J]. Journal of Naval Engineering University，2016，28(2)：75-80.）

[34] FÜRST J，ARGERICH M F，CHENG，B. VersaMatch：Ontology Matching with Weak Supervision [C]// Proceedings of the VLDB Endowment. New York：ACM，2023：1305-1318.

[35] IBRAHIM S，FATHALLA S，LEHMANN J，et al. Toward the Multilingual Semantic Web：Multilingual Ontology Matching and Assessment [J]. IEEE Access，2023，11：8581-8599.

[36] KHAN H，SAQIB M，KHATTAK H A，et al. Ontology Alignment for Accurate Ontology Matching：A Survey [C]//International Conference on Smart Homes and Health Telematics. Berlin：Springer，2023：338-349.

[37] LÜ Z. An Effective Approach for Large Ontology Matching Using Multi-objective Grasshopper Algorithm [C]// ACM International Conference Proceeding Series. New York：ACM，2022：110-116.

[38] AN Y，KALINOWSKI A，GREENBERG J. Exploring Wasserstein Distance across Concept Embeddings for Ontology Matching [EB/OL]. (2022-06-22)[2023-01-10]. https://arxiv.org/abs/2207.11324.

[39] LÜ Q，JIANG C C，LI H. An Interactive Multi-objective Ontology Matching Technique [C]//International Conference on Advanced Machine Learning Technologies and Applications. Berlin：Springer，2021：955-964.

[40] 何杰，屈国兴. 基于 XML Schema 分块的快速本体构建方法 [J]. 吉林大学学报(理学版)，2022，60(5)：1113-1122.（HE J，QU G X. Fast Ontology Construction Method Based on XML Schema Partition [J]. Journal of Jilin University (Science Edition)，2022，60(5)：1113-1122.）

[41] AUMUELLER D，DO H H，MASSMANN S，et al. Schema and Ontology Matching with COMA＋＋ [C]// Proceedings ACM Sigmod International Conference on Management of Data. New York：ACM Press，2005：906-908.

（责任编辑：韩　啸）