基于推理的本体映射抽取算法及修正

张庆军 徐德智 陈建二
(中南大学信息科学与工程学院 长沙 410083)

摘 要 针对使用多策略进行本体映射时,其输出的相似度矩阵中往往含有错误的映射对的问题,基于分布式描述逻辑提出了一种Sufferage抽取算法。该算法融入推理技术对候选映射进行抽取,并对抽取结果做进一步修正,从而正确有效地提高了抽取质量。通过和已有的算法进行实验比较,表明该方案能够明显提高查准率,获得更准确的映射结果。

关键词 本体映射,分布式描述逻辑,Sufferage,推理
中图法分类号 TP393;TP391 **文献标识码** A

Extraction Algorithm and Repairing of Ontology Mapping Based on Reasoning

ZHANG Qing-jun XU De-zhi CHEN Jian-er
(College of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract Aiming at the problem that output similarity matrix often involves some error mappings when using Multi-strategies for ontology alignment, a novel sufferage algorithm based on distributed description logical was proposed. Our algorithm combines with reasoning to extract candidate mappings, and then repairs the result of extraction, so improves the quality of extraction correctly and effectively. Comparing with existing algorithms, experimental results show that our algorithm improves the precision significantly, and thus gets a better mapping result.

Keywords Ontology alignment, Distributed description logical, Sufferage, Reasoning

随着本体数量的增多,本体之间的相互协作变得越来越重要,而本体映射是解决本体知识共享和重用的关键。为了取得更好的映射结果,采用多策略进行本体映射已经成为主流。然而由于每种策略都有其固有的优缺点,产生的映射矩阵中往往含有错误的映射对,从而在一定程度上降低了查准率。如何正确有效地抽取映射对,对于最终的匹配质量有很大影响。本文把抽取问题看成最优化问题,考察映射集中存在的不一致,并分为语义冲突不一致和结构冲突不一致,首先使用分布式描述逻辑推理消除映射对之间的语义冲突不一致,然后采用Sufferage算法进行抽取。最后对得到的映射关系做进一步分析,根据修正规则消除结构冲突不一致。

本文第1节是相关工作的介绍;第2节详细阐述Sufferage抽取算法;第3节对抽取得到的映射结果修正;最后是实验设计和结果分析。

1 相关工作

根据抽取性质的不同,本体映射抽取算法可以分为两大类:一是基于局部实体最优的贪心抽取算法,一是基于本体全局最优的抽取算法。文献[1]对传统的局部最优抽取算法进行了分析,该方法的特点是先设定一个合适的阈值,然后对每个实体抽取超过阈值且相似度值最大的进行匹配。文献[2]把映射对的选取看成风险最小化的决策行为最优化问题,以达到全局映射风险最小化,但是没有结合推理来消除可能的不一致,对于全局最优匹配精度的提高并不明显。文献[3]在抽取过程中首次使用了逻辑推理,来检测不一致的映射对,然后使用匈牙利方法(Hungarian)进行全局最优抽取。然而该推理仅仅考虑了映射对之间的不一致,而没有考虑修正抽取结果中可能存在结构冲突造成的不一致,而且Hungarian算法在极端情况下时间复杂度呈指数级,并对映射本体的规模大小和初始映射对的数目敏感。文献[4]对目前映射对的抽取方法进行了分析,对映射对之间可能造成的冲突做了进一步研究,从映射对的一致性和稳定性两个层面来消除可能的错误映射;然而就像该文献所指出的那样,这种扩展存在着稳定性的限制、过于严格而一致性的限制、过于松弛的局限。

针对相关工作的不足,本文提出了一种新的Sufferage抽取算法进行最优化抽取,在推理方面通过标注disjoint关系来扩展、加强一致性,使之相对严格,并制定了移除错误映射对的移除函数;根据对抽取过程的影响把粗粒度的稳定性进行分类,并结合具体的待映射本体的结构特点制定推理规则进行推理,从而弱化其严格性。

2 一致性检测和Sufferage抽取算法

2.1 映射对之间的一致性检测和扩展

本文关注一对一的映射关系抽取。考虑到仅当映射对的

到稿日期:2008-06-24 本文受863国家重点自然科学基金项目(60433020),湖南省自然科学基金(06JJ50142),湖南省国土资源厅科技计划项目(200718)资助。

张庆军(1981-),男,硕士研究生,主要研究方向为本体映射与本体修正,E-mail:zhqj.csu@gmail.com;徐德智(1963-),男,教授,CCF会员,主要研究方向为Web计算、语义网等;陈建二(1954-),男,博士生导师,主要研究方向为计算机高等算法和优化研究等。

相似度超过一定阈值时抽取才有意义，所以在通过多策略本体映射得到的相似度矩阵中，我们选取超过阈值的映射对作为待检测的候选映射。如果映射对之间存在不一致现象，那么待抽取的候选映射中肯定产生了错误的对应关系。而这些错误的映射关系，将影响到其他映射对的正确抽取，所以为了提高抽取质量，必须对错误的匹配予以消除。为了检测映射对之间的不一致，使用分布式描述逻辑[5]进行跨本体检测，即把映射关系作为桥规则，映射对中的概念近似看成等价关系，把这种等价关系转化为前项 into 蕴含关系(⊆)和后项 onto 蕴含关系(⊇)，并附带相似度值作为对应关系的可信度(confidence)，进而在联合本体中检测是否由于引入了桥规则而导致某些概念不可满足的产生。

定义 1(分布式联合本体)　给定本体 O_1、本体 O_2 以及映射集合 M，由映射 M 桥接起来的分布式联合本体定义为：$O_1 \cup_M O_2 = O_1 \cup O_2 \cup \{t(x) \mid x \in M\}$，其中 t 为桥接转换函数：$t(\langle O_1:C, O_2:D, \equiv, conf\rangle) \mid = O_1:C \equiv O_2:D$。

本文用四元组 $\langle e_1, e_2, r, conf\rangle$ 来表示映射关系，其中 $conf$ 为对应关系的可信度。把映射看成等价关系来桥接两个局部本体，是通过 into 和 onto 两个桥规则在局部本体间引入蕴含关系，使之在结构上发生关联，进而把各自的公理集合传递到对方的公理集合上，进行逻辑推理。然而，由于错误映射对的引入，就可能会在映射对之间引起不一致现象。

定义 2(映射对的一致性)　给定本体 O_1 和 O_2 以及它们之间的映射 M，M 是一致的当且仅当不存在概念 $i:C$，使得 $O_i \mid \neq i:C \sqsubseteq \bot$ 且 $O_1 \cup_M O_2 \mid = i:C \sqsubseteq \bot$，否则 M 是不一致的。其中，$i \in \{1,2\}$。

不一致的映射对会导致某些概念的不可满足，因为该映射对在联合本体中引入了错误的对应关系，这种错误会在应用公理集合推理时导致原本可满足的概念变得不可满足。在考察映射对抽取时，我们不需要对映射对中的概念进行完备推理[5]，实验表明只需在映射对所关联的局部范围内考察是否存在不一致，便能保证大多数情况下抽取的正确性，而不必波及整个联合本体。

定义 3(映射集合的一致性)　当且仅当任何两个映射对之间是一致的时候，我们称之为该映射集合是保证一致性的。

候选映射抽取集合只有在保证一致性的情况下，才能较好地提高抽取的质量。然而，这种一致性很大程度上依赖于 disjoint 关系的正确声明。但是，本体中的 disjoint 关系的声明通常较少或是缺失[6]。比如 OAEI 的 benchmarks 测试数据集的本体中，概念之间应有的 disjoint 关系并未定义，从而弱化了检测的效果。所以有必要进行扩展，来显式地标注一些附加的 disjoint 关系。由于本体中概念分类的相对标准化和严格性，通常将一个类定义为一组互不相交的子类的集合，从而兄弟节点之间不应有共同的部分，因此可以把互为兄弟节点间的关系扩展成 disjoint 关系。为此本文进行必要的预处理。首先，用 jena 解析出源本体和目标本体中所有的 subsumption 关系和 disjoint 关系。为了便于算法处理，对于 subsumption 关系以子树的形式保存在邻接表中，其中头节点为具有子类的类，disjoint 关系用矩阵存储。

算法 1　预处理算法

输入：待映射的本体；

输出：待映射本体 subsumption 关系子树的邻接表表示和 disjoint 关系的矩阵表示；

step1　用带 pellet 推理机的 jena 解析方式解析本体，得到本体中所有具有 subClassof()关系的类以及 disjointWith()关系的类；

step2　将具有 subClassof()的类依次作为邻接表 L 的头节点元素，子类分别置于它们所对应的父节点链表中；

step3　计算邻接表中除头节点外的元素数目，并与解析所得的具有不相交关系(disjoint)的元素相加，设置矩阵维数；

step4　遍历邻接表 L，所链接的子链表元素标注成互为 disjointWith()关系，同解析所得的不相交类存储在矩阵中。

通过预处理把源本体和目标本体解析成两个蕴含关系邻接表和两个不相交关系矩阵。为了更有效地检测不一致，我们把包含关系进行扩展，根据实验当扩展超过 5 层时，其不一致几乎不再变化，所以本文把包含关系扩展成 5 层。图 1 为两层扩展进行检测的示意图，其中①表示映射对应关系，②为不相交关系，实线为包含关系。

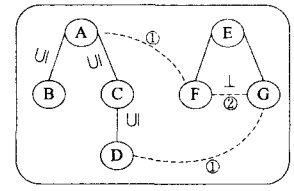


图 1　映射对之间的不一致示意图

当检测到映射对间发生不一致时，需要移除其中一个映射对来保持一致性。然而采用保留相似度值最大的映射对的移除方式并不总是有效的，而借助 WordNet，计算映射对中的元素在 WordNet 中的语义距离，移除语义距离较大的映射对所得的抽取结果更好[5]。但是我们发现，映射对中元素并不总是出现在 WordNet 中，而且当语义距离差超过某一阈值时，采用语义距离的移除方式性能会急剧下降。为此，本文并不是简单地采用语义距离的移除方式，而是定义移除函数来更有效地移除错误的映射对。

定义 4(移除函数)

$$delete(m_1, m_2) = \begin{cases} \min(WSim(m_1), WSim(m_2)), & \text{if } |WSim(m_1) - WSim(m_2)| \leqslant \delta \\ \min(Sim(m_1), Sim(m_2)), & \text{else} \end{cases}$$

其中，m_1 和 m_2 为待检测的两个映射，$WSim()$ 为映射对中的概念在 WordNet 中的语义相似度，$Sim()$ 为由映射策略得出的相似度，δ 为设定的阈值。

对于每一个映射对，如果和另一个映射对中位于源本体中的元素之间为包含关系，且目标本体中的元素之间为不相交关系(或者相反)，则说明这两个映射对之间存在语义不一致，需要移除其中一个映射对。

算法 2　映射集不一致检测算法

输入：包含关系邻接表、不相交关系矩阵和映射对集合 M；

输出：一致性的映射对集合 M′；

```
1：for 每一个 m∈M
2：  for 每一个 m′∈M，且 m≠m′
3：    得到 m 和 m′在目标本体中的概念 c21 和 c22
4：    if c21 和 c22 不相交
5：      得到 m 和 m′在源本体中的概念 c11 和 c12
6：      if 在五层之内 c11 和 c12 构成包含关系
7：        then 从 M 中 delete(m,m′)得到 M′
```

```
8:      end if
9:    end if
10:  end for
11:end for
12:return M′
```

该算法中第六步对时间复杂度的影响较大，要在深度(层数)和宽度(子节点分支数)两个层面进行空间搜索。本文综合考虑时间性能和检测性能，根据实验确定为5层。该算法需要对源本体和目标本体双向各执行一次。

2.2 Sufferage抽取算法

对映射集合消除不一致后，便要进行映射对的抽取。由相关工作的分析可知，把抽取问题看成全局最优化问题，比迭代每一个映射对进行个体最优抽取的效果更好。

定义5(最优化抽取) 给定映射对应关系集合 M，最优化的一对一映射 $M_{opt}\subseteq M$，是指对于其他任一映射集合 $M'\subseteq M$，总有 $\sum_{m\in M_{opt}} conf(m)\geqslant\sum_{m\in M'} conf(m)$。

最优化抽取就是要找到映射集合的一个子集，使得在该子集上抽取得到的相似度之和大于其他任一子集上的相似度之和。本文使用Sufferage算法进行最优化抽取。Sufferage算法是一种启发式的最优化网格调度算法[7]，其调度思想是将任务集合中sufferage值最大的任务分配到性能最优的资源上执行，sufferage值是最优资源与次优资源的性能差距。该算法是一种综合性能较好的调度策略。在本文中，我们把映射对中源本体中的概念类比为调度中的任务，目标本体中的概念类比为调度中的资源，由映射策略所得的相似度值 *conf* 作为初始的效用匹配值，把最大 *conf* 与最小 *conf* 之差作为 *sufferage* 值，在映射集合上进行最优化调度匹配。

算法3 Sufferage抽取算法

```
Sufferage(M):
1:令 M′=∅
2:for 每一个 m∈M
3:  把源本体中的概念置于集合 Ms 中，目标本体中的概念置于集合 Mt 中
4:  if conf(m)=1 或 URI(c1)=URI(c2) //c1 和 c2 为 m 中概念
5:  m→M′
6:for 每一个 m∈M
7:    for 每一个 m′∈M，且 m≠m′
8:      按照 m 中源本体的每个概念所关联的映射进行分组聚类，得到 Ms1，Ms2，…，Msn//n 为总数
9:do until Ms 中的每个概念都已映射指派完毕
10:  对 Mt 中的概念设置标记数组 flag[]，且均初始化为零
11:  for 每个概念 sk∈Ms
12:    在映射分组 Msj 中找到一个 tj∈Mt，使 conf(mkj)值最大，并保留该分组中次最大 conf 值
13:    令 sufferage 值=该最大 conf 值−次最大 conf 值
14:    if 概念 tj 的指派标记 flag[j]=0 //未指派状态
15:      指派 sk 给 tj，将 mkj→M′
16:      从 Ms 中删除 sk，标记 tj 的 flag[j]=1 //已指派
17:    else
18:      if 已指派给 tj 的概念 si 的 sufferage 值<sk 的 sufferage 值
19:        令 si 为未指派状态，即 flag[i]=0
20:        将 si 添加到 Ms 中
21:        指派 sk 给 tj，将 mkj→M′
22:        从 Ms 中删除 sk，标记 tj 的 flag[j]=1
23:  end for
24:end do
25:return M′
```

该算法核心思想是：如果源本体中的概念存在多对一的映射，优先指派最大相似度值与次大相似度值之差(sufferage值)最大。因为一旦该概念被错误地指派，将对整个相似度之和的影响最大。算法把可能的多对一或多对多映射抽取成一对一的映射，使得全局相似度值(即待映射本体的相似度值之和)最大。其中，步骤11到23迭代源本体中的每一个概念。对于任意一个概念 s_k，在目标本体概念中寻找一个具有最大 *conf* 值的概念 t_j，暂时把 t_j 指派给 s_k。如果 t_j 未被指派，则标记 t_j 已指派，并从 M_s 中删除 s_k；如果之前 t_j 已被指派给 s_i，那么从 s_k 和 s_i 中选择sufferage值较大者，然后把 t_j 指派给它，并将它从 M_s 中删除，从而结束一次迭代，重新进入do循环进行下一次指派。这里，我们把最优化的目标函数取为连加和最大，即，$o(M)=\arg\limits_{M}(\sum_{m\in M}Conf(m))$。但是，当有若干sufferage值出现相同时，只能对首次出现sufferage值最大的那个概念进行优先指派。实验发现，这种方式并不总是有效的。当这种情况出现较多时，我们修改目标函数为连乘积最大，即 $o(M)=\arg\limits_{M}(\prod_{m\in M}Conf(m))$。反映在算法中，即要修改步骤13中的sufferage值为最大 *conf* 值乘以次最大 *conf* 值。

3 映射修正

尽管在进行抽取算法之前，已经对本体之间的不一致进行检测，并移除相关的映射对。然而，这种一致性仅仅是从公理集之间是否产生概念的不可满足来检测的，而并未考虑待映射本体结构的特点，从而有可能产生映射关系的不稳定性，即结构性冲突。

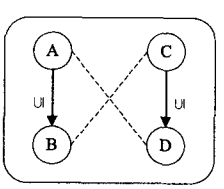
3.1 结构不稳定性冲突

结构不稳定性冲突，是由待映射本体结构特点和部分可能的错误映射两方面的原因造成的。这种冲突有些是合理的，有些却是完全错误的，需要有区别的对待。在局部本体和联合本体中要强调一致的对应关系，即在联合本体中进行分布式描述逻辑推理得出的对应关系不能和局部本体中的关系相冲突，此时我们称该关系是稳定的。映射关系的不稳定性体现在结构上的错位或不一致，本文把这种结构不稳定性分为包含关系的不稳定性和类与属性依附关系的不稳定性。

定义6(映射对间包含关系稳定性) 给定本体 O_1 和 O_2 及其映射 M，M 是包含关系稳定的当且仅当不存在概念 $i:C$，使得 $O_i|\neq i:C\subseteq i:D$ 且 $O_1\cup_M O_2|=i:C\subseteq i:D$，否则 M 是包含关系不稳定的。其中 $i\in\{1,2\}$。

定义7(映射对间类与属性依附关系的稳定性) 给定本体 O_1 和 O_2 以及概念映射 M 和相应属性映射 M'，如果不存在概念 $i:C$，使 $O_i|\neq domainOf(p,c_{i1})\wedge rangeOf(p,c_{i2})$ 且 $O_1\cup_{M\wedge M'}O_2|=domainOf(p,c_{i1})\wedge rangeOf(p,c_{i2})$，则称 M 和 M' 是依附关系稳定的，其中 $i\in\{1,2\}$。

根据结构不稳定性的特点，分以下3种类型进行讨论，其示意图如图2—图4所示(其中虚线表示映射对应关系)。



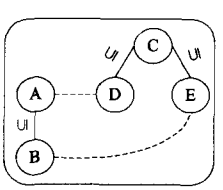


图2 包含关系交叉成环路示意图　图3 映射后包含关系消失示意图

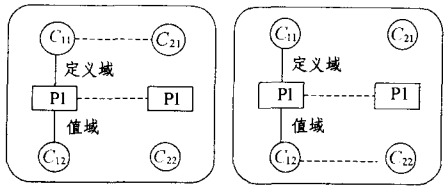


图 4　映射后属性关系(分为值域和定义域依附关系)消失示意图

(1)映射中概念的包含关系交叉而出现环路

一个自身语义一致的本体,如果将其概念视为节点,概念之间的包含关系视为节点与节点之间的有向箭头,则所有概念与概念之间的包含关系应该是个有向无环图。同理,由映射桥接起来的语义一致的联合本体也应该是一个有向无环图。但是交叉映射会破坏这种规则而形成环路,如图 2 所示。应用分布式描述逻辑推理时,可以把映射关系看成双向包含,于是在联合本体中应用 onto 桥规则有:$B\supseteq C\supseteq D\supseteq A$,应用 into 桥规则有 $C\subseteq B\subseteq A\subseteq D$,这样就形成了环路的包含关系,而推理得出的 $B\supseteq A$ 和 $C\subseteq D$ 破坏了局部语义一致性。此时,应该通过启发式规则,删除其中一个映射对来解除环路。

(2)映射之后概念的包含关系消失

如图 3 所示,一个本体中有包含关系的两个概念经过映射后,在另一个本体中对应的两个概念不存在包含关系。然而这种情况不像包含关系的交叉映射那样,一定是不合理的。由于本体结构的差异,这种情况有时是合理的,有时是不合理的,需要根据修正规则来判断并进行相关处理。

(3)映射之后概念与属性之间的关系消失

如图 4 所示,在一个本体中,某个概念有某属性,经过映射后对应概念没有对应属性。这种情况可能是概念之间的对应关系有错,还可能是属性之间的对应关系有错,也可能由于其异构性而视为是合理的。具体要结合概念和属性的其他情况进行辨别,同样根据修正规则进行处理。

3.2　修正规则

本文根据统计的思想,分别统计抽取得到的映射结果中每个映射对引起这 3 种结构不稳定性冲突的次数,并记录与它发生冲突的那些对应关系,以矩阵的形式进行标记,统计出总的冲突次数 *incsTotal*。修正算法根据启发式修正规则进行有区别的处理。规则如下:

①对于交叉映射成环路,这种情况必然导致语义逻辑上的不合理,必须移除其中一个对应关系才能解除冲突。同样使用分布式描述逻辑进行不完备的推理,考察任意两个映射对在 5 层范围之内是否出现交叉情况。若出现,则按照移除函数删除其中一个对应关系。由于本体层次结构的局部耦合紧密性,这种不完备的推理在绝大情况下是非常有效的。

②如果待映射本体的概念结构差异很大,即结构相似度很低,则忽略结构不稳定性情况 2 所引发的所有不一致,因为此时即使是参考映射也包含有一定量的结构不一致的映射对。我们用两个本体的方差来衡量其结构相似度:$S^2=\frac{1}{n}\sum_{i}^{n}(sub_i-\overline{sub})^2$,即根据包含关系邻接表对每个概念统计其子概念个数,然后求方差。上式中,sub_i 为概念 i 的子概念,$\overline{sub}$ 为所有概念的子概念总数的平均值。该方差反应了结构偏离程度,如果该方差相差一倍以上,认为可以忽略该不一致。

③如果待映射本体的对象类型属性的数量相差一倍以上,或数据类型属性的数量相差一倍以上,同样忽略结构不稳定性情况 3 所引发的不一致。

对于规则 1,修正算法的处理类似于算法 2;对于规则 2 和规则 3,当不能忽略该类型的冲突时,修正算法设置阈值 θ,令其为总不一致数量 *incsTotal* 与一个百分比 percentage 的乘积。只要总的不一致数量大于 θ,就找一个错误可能性最大的对应关系(涉及到的冲突数量最多且其值大于 σ)进行删除,同时使得 *incsTotal* 减 1,直到其值小于 θ 为止。同时还要设置参数 σ,如果某个对应关系引起的第二或第三种类型冲突的数量小于概念与概念对应关系的数量与 σ 之积,则认为是在正常范围之内,不应删除此对应关系。

4　实验结果及分析

4.1　测试数据集

本文进行两组实验。实验 1 测试各抽取算法性能的优劣,实验 2 把本文提出的算法应用在我们的 SNAX 系统中,并对比应用前后的实验结果。实验 1 选取 *conference* 测试数据集,该测试集由 14 个会议组织领域的本体组成,OntoFarm project[9] 项目组开发。由于实验目的是检测对多对一映射以及映射对之间的冲突,因此我们只挑选其中的 SIGKDD, EKAW, CMT, PCS, CONFTOOL 5 个本体。实验 2 利用 OAEI2007 的标准测试数据集 *benchmarks*,该数据集包含 51 个本体,其中本体 #101 为参考本体。

实验结果使用查准率和查全率进行评估,定义如下:

查准率(Precision):$p=\frac{|R\cap A|}{|A|}$

查全率(Recall):$r=\frac{|R\cap A|}{|R|}$

其中,A 表示算法识别得到的正确映射结果,R 表示参考映射结果。

4.2　实验结果及对比分析

实验 1 中,本文以 CMT 本体为源本体,其他 4 个本体为目标本体进行映射。由于 OAEI 并未给出参考映射,我们项目小组通过手工验证建立参考映射,如表 1 所列。

表 1　conference 数据集的统计数据

本体	概念	属性	参考映射
CMT	30	59	--
CONFTOOL	39	36	10
EKAW	78	33	11
PCS	24	38	19
SIGKDD	50	28	14

我们以最简单的抽取方式为基准,即在相似度矩阵中,源本体中的每个元素都抽取所在行中相似度最大的元素进行匹配。其他的抽取算法[4] 以该基准为参考,考察其在查准率上的收益(gain)和查全率上的损失(loss)。实验结果如图 5 所示。

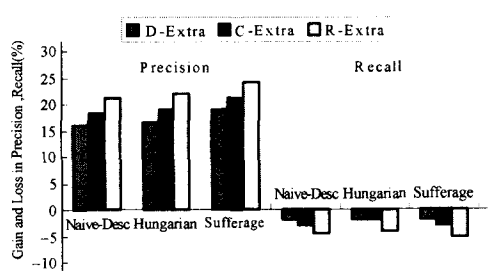


图 5　各抽取算法实验结果对比

在图 5 中,D-Extra 表示直接抽取,C-Extra 表示进行语

义一致性检测后再抽取，R-Extra 表示在 C-Extra 基础上进行抽取修正。可以看出，融入推理后的抽取算法，性能明显得到提高，且其在查准率上的收益大于在查全率上的损失。在各抽取算法中，Sufferage 查准率收益最好，但查全率稍微差些，而 Hungarian 查全率损失最小。通过改变优化目标函数和算法的执行顺序，我们可以得出下面两个结论：

①当遇到 sufferage 值相等的情况较多时，修改优化的目标函数为连乘积后，查准率会略有提高；

②如果把不一致检测和结构冲突修正过程合并放到抽取之前进行，对于 Hungarian 算法，将直接影响抽取过程中的优化选取和指派，抽取质量提高的程度不如其他算法明显。而分离开来，则具有更大的适应性。

在时间复杂度上，Naïve-Desc 和 Sufferage 抽取算法均为 $O(n*m)$，其运行时间为 2～3s，而 Hungarian 为 5s 左右。可以预见，当本体规模变大时(Conference 测试集本体规模均较小)，Sufferage 的时间性能将比 Hungarian 更加好。

实验 2 把本文的映射算法应用在项目 SNAX 系统的升级版本中，使用该系统的映射子模块 SNAX_Mapping[10] 得出候选映射，再用本文的抽取算法进行抽取和修正。

表 2 SNAX_Mapping 映射抽取修正前后结果比较

System	SNAX_Mapping(抽取修正前)		SNAX_Mapping(抽取修正后)	
test	Pre.	Rec.	Pre.	Rec.
1××	1.0000	1.0000	1.0000	1.0000
2××	0.9367	0.8014	0.9445	0.7928
3××	0.8562	0.7301	0.8793	0.7274
total	0.9279	0.8326	0.9396	0.8298

表 2 中 #1××～3×× 表示标准测试数据集 benchmarks 中本体编号，Pre. 表示查准率，Rec. 表示查全率。抽取修正前的映射集的得出详见文献[10]。该实验设置参数 percentage 为 50%，σ 为 10%，从表 2 可以看出，在测试数据集[1××]上，抽取修正之前，查准率和查全率均为 1.0，抽取修正后仍然不变；对于测试数据集[2××]和[3××]，应用本文的算法后查准率都有了一定程度的提高，但查全率略有损失。其中，在[2××]上除本体 210，247，252，261，266 改进比较明显外，大部分没有太大改进，因此总体改进不大。但在[3××]上，除本体 301 外其他改进均较为明显，总体表现最佳。

通过对比各组数据的特点及实验观察，我们发现查准率的提高归结于两个方面：抽取之前的语义消歧及之后的冲突修正；全局最优的抽取算法及移除函数的制定。尤其是融入推理后，精度有了很大提高，分析如下：融入推理前，查准率公式为 $p=\frac{|R\cap A|}{|A|}$，融入推理后 A 可以分解为 A^+ 和 A^-，其中 A^- 是修正过程中去掉的“伪正确”映射对，此时查准率公式应改写为 $p'=\frac{|R\cap A^+|}{|A^+|}$。注意到 $|R\cap A^+|$ 和 $|R\cap A|$ 其值是相等的，因此最终的查准率公式可以写为 $p=\frac{|R\cap A|}{|A^+|}$。比较推理前后的两个公式便可以看出，使用推理技术能够更好地提高映射质量。

结束语 本文提出了 Sufferage 算法对候选映射对进行抽取。为了提高抽取质量，抽取之前先消除映射对间的语义冲突不一致，并对抽取之后的映射结果中可能存在的结构上的不一致进行修正。由于抽取之前的语义消歧和抽取之后的结构修正对抽取的质量影响程度不同，我们分两个阶段进行处理。尽管本文中推理是不完备的，然而实验发现只对映射对(及其涉及的某一局部范围)进行推理是很有效的，而且时间性能有了较大提高。

参 考 文 献

[1] Euzenat J, Shvaiko P. Ontology Matching [M]. Springer Verlag, 2007: 157-187

[2] 唐杰，梁邦勇，李涓子，等. 语义 Web 中的本体自动映射[J]. 计算机学报，2006，29(11)：1956-1976

[3] Meilicke C, Stuckenschmidt H. Applying logical constraints to ontology matching[C]// Proceedings of the 30th Annual German Conference on Artificial Intelligence. Germany: KI 2007, 2007: 99-113

[4] Meilicke C , Stuckenschmidt H . Analyzing mapping extraction approaches[C]// Proceedings of the ISWC'2007 Workshop on Ontology Matching OM-200. Korea, 2007: 25-36

[5] Meilicke C, Stuckenschmidt H, Tamilin A. Repairing ontology mappings[C]// Proceedings of the Twenty-Second Conference on Artificial Intelligence. Canada, 2007: 22-26

[6] Schlobach S. Debugging and semantic clarification by pinpointing [C]// Proceedings of ESWC 2005. Greece, 2005: 226-240

[7] 丁丁，罗四维，高瞻. 网格环境下一种可调目标的启发式调度策略[J]. 计算机研究与发展，2007，44(9)：1572-1578

[8] Kuhn H W. The hungarian method for the assignment problem [J]. Naval Research Logistics, 1955, 2: 83-97

[9] Svab O, Svatek V, Berka P, et al. Ontofarm: Towards an experimental collection of parallel ontologies[C]// Poster Proceedings of the International Semantic Web Conference 2005. Galway, 2005

[10] Zhang Zhiwei, Xu Dezhi, Zhang Tian. Ontology Mapping Based on Conditional Information Quantity[C]// Proceedings of IEEE International Conference on Networking, Sensing and Control . Sanya, 2008: 587-591

(上接第 202 页)

[6] Zadeh L A. Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems [J]. Soft Computing, 1998, 2 (1): 23-25

[7] Lin T Y. Granular Computing on Binary Relations: I: Data Mining and Neighborhood Systems. II: Rough Set Representations and Belief Functions[C]// Skowron A, Polkowski L, eds. Rough Sets in Knowledge Discovery. Physica-Verlag, 1998: 107-140

[8] Yao Y Y. Granular Computing: basic issues and possible solutions [A]// Proceedings of the 5th Joint Conference on Information Sciences[C]. Atlantic, USA: Association for Intelligent Machinery, 2000: 186-189

[9] 苗夺谦，王国胤，刘清，等. 粒计算：过去、现在与展望[M]. 北京：科学出版社，2007

[10] 刘清. Rough 集及 Rough 推理[M]. 北京：科学出版社，2001

[11] Pawlak Z. Rough Sets, Rough Relations and Rough Functions [J]. Fundam. Inform, 1996, 27(2/3): 103-108