流程可定制本体匹配框架:RiMOM2

李 虎 张 啸 仲 茜 侯 磊 王志春
(清华大学计算机科学与技术系 北京 100084)

摘 要 本体作为语义Web中的语义表示形式,是语义Web体系结构中的核心元素,是实现知识共享、协同工作的关键。然而现实世界中本体自身与生俱来的分布性和异构性,又极大地限制了数据的共享与集成。为了实现知识的共享、数据的集成,近年来针对本体匹配方法的研究得到了广泛的重视。随着本体匹配研究的深入,许多有效的本体匹配方法被提出。RiMOM2正是一种集成了多种有效本体匹配方法的多策略本体匹配框架。它尽可能地向初级用户隐藏不必要的阈值设定和参数设置,而向高级用户提供匹配流程的可定制功能,以期针对不同用户实现一种既能适用于普遍本体匹配任务,操作简易,又能达到具有针对性匹配效果的本体匹配工具。同时该框架具有匹配方法组件的易扩展性。

关键词 语义Web,本体,本体匹配,实例匹配,框架
中图法分类号 TP31 **文献标识码** C

RiMOM2: A Customizable Framework of Ontology Matching

LI Hu ZHANG Xiao ZHONG Qian HOU Lei WANG Zhi-chun
(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract As the representation of the semantic Web, the ontology is the key element of the semantic Web system, and the key for knowledge sharing and working together. However, in the real world, the inherent heterogeneous and distribution of ontology have greatly limited the knowledge sharing and data integration. In order to realize the knowledge sharing and data integration, the ontology matching has been widely studied. With the deepening of the research, numerous of effective ontology matching methods have been proposed. RiMOM2 is a multi-strategy framework of ontology matching, which integrated many effective ontology matching methods. For primary users, RiMOM2 hides unnecessary thresholds and parameters settings in order to achieve an easily manipulated ontology matching with common requirements. While for advanced users, it provides a customizable matching process function so that it can be used for ontology matching with specific requirements. The matching method components also are easily extensible in the framework.

Keywords Semantic Web, Ontology, Ontology matching, Instance matching, Framework

1 引言

自"The Semantic Web[1]"一文发表以来,语义Web已经广泛地得到了学术界和工业界的重视。本体作为语义Web中的语义表示形式,是语义Web体系结构中的核心元素,是实现知识共享、协同工作的关键。本体作为共享概念模型的形式化规范说明[2],近年来越来越多的学术机构和企业开始使用本体来描述其所在领域的数据及其语义。比较著名的本体有:OpenCyc本体[1]、OBO(Open Biomedical Ontologies,开放生物医学本体集)本体集[2]、TCM(Traditional Chinese Medicine,传统中医药)本体[3]和DBpedia[4]本体等。这些本体在其相关领域内得到了广泛的应用,促进了信息的共享、集成和系统间的语义互操作。然而,由于本体定义的自发性,及其定义者自身的社会文化背景、对术语规范的使用习惯、对本体组织结构的理解的诸多差异,使得本体在描述同一领域知识时,其具体形式却存在着这样那样的不同,从而直接导致了互联网上本体的分布性和异构性,因而限制了本体数据的共享与集成。为了更好地在语义Web中实现语义互操作,就必须建

到稿日期:2010-06-11 返修日期:2010-09-25 本文受国家自然科学基金(60973102),国家基础研究973项目(2007CB310803),国家高技术研究发展863计划(2009AA01Z138)资助。

李 虎(1979—),男,硕士生,主要研究方向为语义本体映射,E-mail:lihu@keg.cs.tsinghua.edu.cn;张 啸(1986—),男,硕士生,主要研究方向为语义数据集成;仲 茜(1975—),男,博士,主要研究方向为语义数据集成和本体映射;侯 磊(1986—),男,博士生,主要研究方向为语义Web和本体映射;王志春(1983—),男,博士后,主要研究方向为语义Web和本体映射。

1 http://www.cyc.com/

2 http://www.obofoundry.ogr/

3 http://tcmontology.com/

4 http://dbpedia.ogr/

立起异构本体间实体(包括概念、关系、实例等)的匹配关系，其过程称为本体匹配[6]。目前，针对本体匹配的研究主要集中在本体概念以及实例之间的匹配关系。

本体匹配是发现不同本体中具有一致语义实体的过程。本体匹配已成为语义 Web 领域的研究热点[3]。目前，众多的本体匹配策略已经被提出，与之相应，为实现不同本体之间的实际匹配任务而产生了许多本体匹配系统软件，如：Cupid[19]，Rondo[20]，OLA[21]，QOM[22]，S-Match[24]，COMA[25]，COMA++[26]，Falcon-AO[5,12]，RiMOM[4,8,10]等。比较典型的本体匹配系统包括德国莱比锡大学开发的 COMA++系统、东南大学开发的 Falcon-AO 系统以及清华大学开发的 RiMOM 系统。这些系统都开始注重在具体匹配任务中综合使用多种匹配策略，使得在这些系统中都集成了多种本体匹配方法，以期综合多种匹配方法的匹配结果，进而达到较好的最终匹配效果。然而仍然有几个问题需要更深入的研究：1)针对特定的一个本体匹配任务，并不是把所有的匹配方法执行的匹配结果进行累加就能得到更好的匹配结果。那么，针对一个特定的本体匹配任务，如何确定应当综合哪些具体匹配方法的匹配结果？2)又该如何"累加"才能得到较好的最终匹配结果呢？

同时基于已有一些软件共同存在的问题，比如：1)本体匹配的流程基本固定，调整起来比较困难，缺乏灵活性；2)匹配的参数设置偏专业，初级用户较难掌握；3)对于一些数据集，匹配的结果和效率都很难有好的效果。以及较早版本的 RiMOM 系统在实践中暴露出来的问题，比如：1)系统各组件间的关联较为紧密，不利于系统的扩展和移植；2)匹配流程固定，不利于专业用户有针对性地对某些数据集上的本体匹配任务进行个性化的匹配流程定制；3)匹配结果聚合的有限手工处理，不能很好地适应新的匹配器随意扩展的需要。基于以上问题以及需要进一步深入研究的课题，清华大学计算机系知识工程组对相关问题进行了深入研究，对原 RiMOM 系统进行了重新设计、重新开发。新的系统(RiMOM2)将作为一种匹配方法易扩展、匹配流程可定制、匹配结果聚合自动化、面向不同用户群体的本体匹配框架而存在。

目前最新版本的 RiMOM2 具有以下主要特点：

(1)灵活开放的系统框架。RiMOM2 对本体匹配流程进行了合理的抽象和解耦，定义了一系列标准接口。通过实现这些接口，系统开发和维护人员可以很方便地将新的匹配方法集成到系统中。RiMOM2 还实现了匹配流程自定义功能，用户只需要根据系统提供的匹配流程 XML Schema 标准就可以自定义新的匹配流程。用户自定义匹配流程将以 XML 文档的形式存储在系统中，它可以将系统中的各类匹配方法灵活地组合起来，实现具有针对性的面向具体匹配任务的用户定制功能。

(2)较完备的易扩展的本体匹配方法。为了支持常见的本体匹配任务，RiMOM2 实现并集成了多种类型的本体匹配方法，如：基于编辑距离相似度的方法、基于 WordNet 相似度的方法、基于 KNN(K Nearest Neighbors)的方法、基于本体结构的方法、基于数据场和高斯函数的方法、基于已有匹配结果的方法等。为了帮助用户选择适当的本体匹配方法和对不同匹配方法的结果进行聚合，RiMOM2 还实现了相应的匹配策略选择和匹配结果聚合方法。不但如此，用户还可以很容易地将其它匹配方法集成到 RiMOM2 中。

(3)更加友好的用户接口。为了方便用户使用和对匹配结果进行反馈，RiMOM2 提供了图形界面客户端工具。通过它用户可以新建匹配任务、选择匹配方法、定制匹配流程、对系统和匹配方法的参数进行设置。此外，在获得系统返回匹配结果后，用户还可以对其匹配结果进行可视化的调整，作为反馈可重新参与到新一轮匹配过程中，因而具有一定的用户反馈能力。

本文第 2 节对相关概念作了简要说明；第 3 节介绍了 RiMOM2 的框架结构；第 4，5 节较详细地介绍了 RiMOM2 的匹配流程和各功能模块；第 6 节给出了一些具体实验及结果分析；最后对 RiMOM2 及其面对的挑战进行简要总结。

2 相关概念

定义 1(本体) 在语义 Web 中，本体 O 表示为四元组 $O(C,P,A,I)$，其中，C 为概念集合，P 为属性集合，A 为公理的集合，I 为实例集合；$C\cup P\cup A$ 称为本体模式，记为 OS；$C\cup P\cup I$ 中的元素 e 称为本体实体(entity)。

概念又称为类(Class)，是某一领域内具有相同性质对象的集合。在 RDFS 和 OWL 中通过预定义属性 rdfs:Class 和 owl:Class 来定义。另外，RDFS 和 OWL 中还提供了一些表示简单数据类型的预定义类，如字符串(xs:string)、整数(xs:integer)等。

属性又称为关系(Relation)，用以表示概念之间的语义关联。在 RDFS 和 OWL 中，属性 p 可记为 (D,p,R) 或 $p(D,R)$，其中 D,R 为类的集合，分别称为 p 的定义域(Domain)和值域(Range)。在 OWL 中如果属性 p 的值域为简单数据类型，称 p 为数据类型属性(owl:DatatypeProperty)，否则称为对象属性(owl:ObjectProperty)。在 RDFS 和 OWL 中还可以通过属性 rdfs:subClassOf 和 rdfs:subPropertyOf 定义父子概念和父子属性，从而构成实体的层次结构。

在语义 Web 中本体数据采用 RDF 进行表示和存储。RDF 的基本单元称为一个陈述(Statement)，即三元组(Subject,Predicate,Object)，其中 Subject 通常为代表某一资源的 URI；Predicate 为代表某属性的 URI；Object 既可以是代表某一资源的 URI，也可以是一个字面量(Literal)。除此之外，Subject 和 Object 还可以是一个匿名资源(一般无实际意义或尚未定义)，称为空白节点(Blank Node)。

定义 2(本体匹配) 本体匹配是发现不同本体中具有一致语义实体的过程。其形式化定义表示如下：

设 $M(O_1,O_2,F_M)$ 为本体 $O_1(C_1,P_1,A_1,I_1)$ 和本体 $O_2(C_2,P_2,A_2,I_2)$ 的本体匹配，F_M 为建立 O_1,O_2 间本体匹配的匹配函数，定义为

$$F_M(O_1,O_2)\rightarrow\{(\{e_{i1}\},\{e_{j2}\})\mid e_{i1}\in O_1 \text{ 与 } e_{j2}\in O_2 \text{ 匹配}\}$$

其中，$\{e_{i1}\},\{e_{j2}\}$ 均为满足条件的最大实体集合，即对于任意实体 $e\in O_1$，它与 $\{e_{j2}\}$ 中的实体匹配，当且仅当，$e\in\{e_{i1}\}$；同样，对于任意实体 $e\in O_2$，它与 $\{e_{i1}\}$ 中的实体匹配，当且仅当，$e\in\{e_{j2}\}$。

如果 $|\{e_{i1}\}|=|\{e_{j2}\}|=1$，称 M 为单匹配或 1-1 匹配；否则，称为多匹配。

如果 $e_{i1}\in O_1^S,e_{j2}\in O_2^S$，称 M 为本体模式匹配。由于目前本体匹配的研究主要集中在模式层匹配上，因此有时也将本

体模式匹配简称为本体匹配。如果 $e_{i1} \in I_1, e_{j2} \in I_2$，则称 M 为本体实例匹配，有时也将其简称为实例匹配。

3 框架结构

RiMOM2 是清华大学计算机系知识工程组研制开发的一个易扩展、可定制、多策略的本体匹配框架，它基于 Java SE 5.0 实现，具有良好的平台适应性。图 1 描述了 RiMOM2 的框架结构。

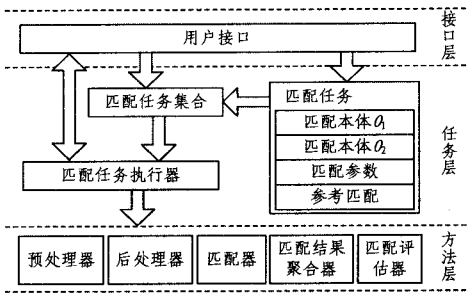

图 1 RiMOM2 框架结构

从图 1 可以看出，RiMOM2 的框架结构可分为 3 个层次：接口层、任务层和方法层。这种分层结构使 RiMOM2 具有良好的灵活性，使得 RiMOM2 不但可以作为本体匹配系统单独使用，也可以方便地集成到其它的应用系统中，作为匹配模块来使用。同时使得各层组件之间的耦合度极大地降低，方便了系统的扩展。下面分别作以简要介绍。

3.1 接口层

为用户与系统之间提供会话接口。用户通过接口层构建匹配任务、选择匹配方法、设定匹配参数，通过提交匹配任务来调用匹配任务执行器以启动该任务的执行，并在任务执行结束后获得匹配结果。此外，用户还可以通过该层对匹配结果进行查看，调整并反馈给系统，进而有效地提高系统重新匹配的准确度。

3.2 任务层

该层维护着用户建立的诸多匹配任务，每个匹配任务包括：待匹配本体、匹配参数，有的还包括参考匹配。其中，待匹配本体为其匹配任务所要处理的本体，在任务执行前这些本体并未被解析，而只是指定了对应本体资源的 URI，实际的解析操作通常在预处理器中完成。匹配参数设定了匹配任务执行的环境参数，这些参数包括：1）匹配任务所采用的预处理器、匹配器、后处理器、匹配聚合器以及匹配评估器等方法；2）相关方法的参数，如：针对某个方法的匹配过滤阈值、进行匹配方法自动选择或计算结果聚合权重所用到的阈值等；3）某些开关变量，如：是否指定为实例匹配，是否指定有参考匹配等；4）匹配流程，用于确定该任务所需要调用的方法以及这些方法的执行顺序。在匹配任务被建立完毕之后，RiMOM2 会将它们放置在匹配任务集合中，直至该任务被删除。图 2 显示了匹配任务的状态转换过程。

匹配任务创建完毕，即进入未提交状态。只有处于未提交状态的任务才可以被用户提交执行。当某一个匹配任务被提交执行后，由匹配任务执行器按照任务匹配参数中指定的匹配流程执行，同时该任务状态被设定为已提交状态。处于已提交状态的任务在执行时才真正意义上使用匹配参数中所设置的参数值，并且处于已提交状态的任务不可删除，其匹配参数亦不可修改。在匹配任务执行完毕后，同样由匹配任务执行器按照匹配参数中的设定对匹配结果进行存储处理，随之该任务状态被设定为执行完成状态。处于未提交和执行完成状态的匹配任务，其匹配参数可随时被用户修改，然而，执行完成状态下的匹配任务的匹配参数在被用户修改后，系统会再次将其所处状态设定为未提交状态。如：在某一匹配任务执行完成后，用户可以通过接口层对匹配结果进行可视化手工调整（纠正匹配结果中存在的错误匹配，或由用户给出一些明确匹配等），继而作为已有参考匹配反馈给系统，以参与该任务的重新匹配过程。由于匹配参数被修改，该匹配任务再次被设定为未提交状态，即可被再次提交执行。同时只有处于未提交或执行完成状态下的匹配任务才可以被用户删除。

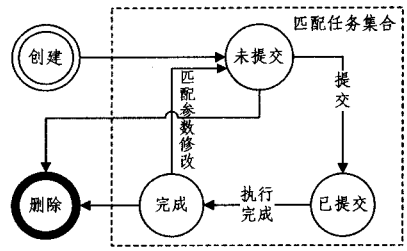

图 2 匹配任务状态转换图

3.3 方法层

该层包含执行匹配任务时所用到的各种具体的方法组件。在第 5 部分将对 RiMOM2 中所实现的主要具体方法组件进行详细介绍。目前，这些方法组件主要包括：

（1）预处理器，主要完成实际的匹配操作执行前的必要初始化工作，例如本体的解析等。RiMOM2 并不提供专门的本体解析器，只定义了对本体数据进行操作的抽象接口，具体的解析工作由第三方本体解析器来完成，如 Jena[5]，OWL API[6] 等。这样做的目的是使 RiMOM2 具有更好的灵活性，用户只要实现了相应的接口就可以使 RiMOM2 处理指定解析器解析的本体数据。另外，本体中术语的规范化、去噪以及具体匹配方法中所用到的特定数据结构的建立，一般也在预处理阶段完成。

（2）后处理器，用于对初始匹配结果进行一些必要处理，如：从初始结果中提取最终的匹配等。

（3）匹配器，具体执行本体匹配操作的方法组件。目前 RiMOM2 中实现的匹配器有基于编辑距离相似度的匹配器、基于 WordNet 相似度的匹配器、基于 KNN 的匹配器、基于本体结构的匹配器等。

（4）匹配结果聚合器，对多个本体匹配方法获得的匹配结果进行聚合。

（5）匹配结果评估器，用于依据指定的评价指标对本体匹配结果进行评估。

3.4 易扩展性

RiMOM2 采用明确的分层设计，使得系统对各功能模块

[5] http://jena.sourceforge.net/

[6] http://owlapi.sourceforge.net/

进行高度的抽象成为可能。RiMOM2 通过定义一系列的标准接口，系统开发和维护人员可以方便地通过实现这些接口来将新的匹配方法集成到系统中，通过这样的方式，使得系统中各功能模块之间的紧密耦合度得以有效释放，达到系统解耦的目的。

另外，为了方便用户掌握和使用系统中的各种方法组件，RiMOM2 对每类组件定义了相应的 XML Schema 文档。依据这些文档，用户能够对系统中的方法组件进行规范化的描述(如：方法参数的定义及说明等)，并以 XML 文档的形式存储在系统中。系统启动时，会根据相关文档的内容将组件的信息注册到系统中，以方便用户使用。

4 匹配流程

4.1 匹配流程组件

RiMOM2 除了提供丰富的与本体匹配相关的原子类型的方法组件外，还对本体匹配的流程进行了抽象，增加了对匹配流程进行控制的组件：串行流程组件和并行流程组件。

(1)串行流程组件

图 3 给出了 RiMOM2 中串行流程组件类型 XML Schema 定义，RiMOM2 中的串行流程组件由若干顺序执行的组件组成，这些组件可以是原子类型的方法组件，如：预处理器组件、后处理器组件、匹配器组件；也可以是流程控制组件，如：并行流程组件。在串行流程组件中的各组件之间有着严格的次序依赖关系，这是因为后一个组件的输入总是依赖于前一个组件的输出。

```
〈xs:complexType name="SequenceType"〉
 〈xs:sequence〉
  〈xs:choice maxOccurs="unbounded"〉
   〈xs:element name="PreProcessor" type="AtomCompentType"
minOccurs="0" maxOccurs="unbounded"/〉
   〈xs:element name="PostProcessor" type="AtomCompentType"
minOccurs="0" maxOccurs="unbounded"/〉
   〈xs:element name="Matcher" type="AtomCompentType"
minOccurs="0" maxOccurs="unbounded"/〉
   〈xs:element name="Parallel" type="ParallelType"
minOccurs="0" maxOccurs="unbounded"/〉
  〈/xs:choice〉
 〈/xs:sequence〉
〈/xs:complexType〉
```

图 3 串行流程组件类型 XML Schema 定义

(2)并行流程组件

图 4 给出了 RiMOM2 中并行流程组件类型的 XML Schema 定义，RiMOM2 中的并行流程组件由若干个独立执行的串行流程组件和 1 个匹配结果聚合器组件组成。其中各个独立执行的串行流程组件可以是单个的原子类型的方法组件，也可以是如图 3 中所定义的任意组合串行流程组件。在各个独立执行的串行流程组件执行完毕后，匹配结果聚合器组件再将其各个串行流程组件执行的结果进行聚合后输出。

```
〈xs:complexType name="ParallelType"〉
 〈xs:sequence〉
  〈xs:choice maxOccurs="unbounded"〉
   〈xs:element name="PreProcessor" type="AtomCompentType"
minOccurs="0" maxOccurs="unbounded"/〉
   〈xs:element name="Matcher" type="AtomCompentType"
minOccurs="0" maxOccurs="unbounded"/〉
   〈xs:element name="PostProcessor" type="AtomCompentType"
minOccurs="0" maxOccurs="unbounded"/〉
   〈xs:element name="Sequence" type="SequenceType"
minOccurs="0" maxOccurs="unbounded"/〉
  〈/xs:choice〉
  〈xs:element name="Aggregator" type="AtomCompentType"/〉
 〈/xs:sequence〉
〈/xs:complexType〉
```

图 4 并行流程组件类型 XML Schema 定义

4.2 基本匹配流程

RiMOM2 采用 XML Schema 对本体匹配流程进行了定义，用户可以方便地根据定义对系统中的方法组件进行任意组合来定制个性的匹配流程。匹配流程将以 XML 文档的形式存储在系统中，并与具体的匹配任务相关联。当该匹配任务提交执行后，RiMOM2 会自动地对其关联的匹配流程进行解析，然后调入到匹配任务执行器的相应执行引擎中执行。图 5 给出了系统的基本匹配流程。

RiMOM2 的基本匹配流程整体上是一个串行流程，下面对图 5 中所描述的基本匹配流程的各部分作详细说明。

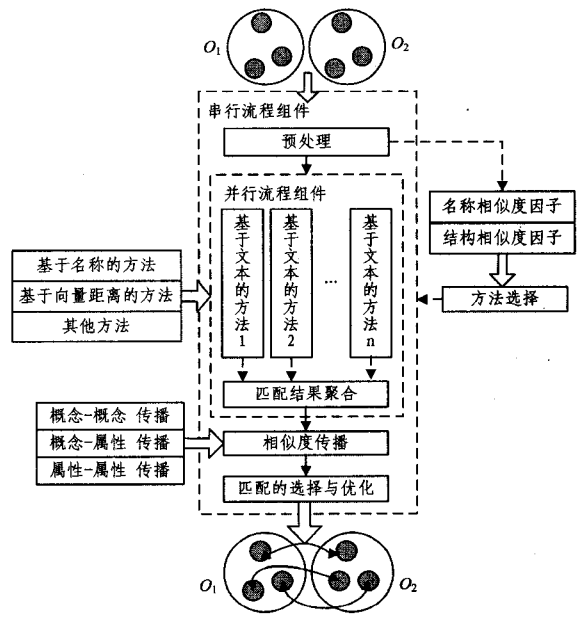

图 5 RiMOM2 基本匹配流程

(1)预处理

通过匹配参数中指定的预处理器实现。目前系统中实现的预处理器组件主要提供两部分功能，一个是通过 OWL API 解析本体的序列化文件；另一个是为自动策略选择计算相关相似度因子[8]。

(2)方法选择

根据用户的设置和本体中数据的特征来选择适当的匹配方法和匹配结果聚合方法，是预处理过程的一部分。在当前的 RiMOM2 实现中，方法选择过程主要通过相关因子[8]以及用户在任务参数中所设定的阈值来选择适当的匹配器组件和匹配结果聚合器组件。

(3)基于文本的匹配方法

这些匹配方法主要使用本体中的文本信息，如：实体的名

称、注释等与实体相关的文本。目前实现的方法主要包括基于编辑距离相似度的方法、基于 WordNet 相似度的方法、基于向量距离的 KNN 方法等。其实，该部分并不限制具体的匹配方法，只是目前的 RiMOM2 主要实现了上述基于本体实体中的文本信息的方法，具体匹配方法由相应的匹配器组件实现。

(4)匹配结果聚合

由匹配结果聚合器实现，通过聚合过程将第(3)点中各个不同的具体匹配方法获得的匹配结果聚合起来，以综合达到更好的匹配效果。具体的聚合方法由 RiMOM2 中集成的聚合器实现。

(5)相似度传播

由专门的匹配器实现。采用 Similarity Flooding[9] 算法，RiMOM2 实现了 3 类相似度的传播策略：CCP(概念-概念传播，Concepts to Concepts Propagation)、CPP(概念-属性传播，Concepts to Properties Propagation)和 PPP(属性-属性传播，Properties to Properties Propagation)。其关键是在构建相似度传播图时，将概念和属性都看成本体图中的节点，并通过相关属性建立节点间的关联。具体匹配时选择哪个策略来实现相似度的传播由用户设定的匹配参数和相关因子决定。

(6)匹配的选择与优化

由后处理器组件实现，用于过滤最终匹配、存储匹配结果。此外，RiMOM2 会根据特定待匹配本体的具体特点，从初始匹配结果中去除掉一部分“不可信”的匹配，得到最终的匹配结果。

5 方法组件

5.1 基于编辑距离的方法

基于编辑距离的方法是一种根据字符串中的字符序列来度量字符串间的相似程度的方法[18]。字符串 s_1，s_2 的编辑距离 $\delta(s_1,s_2)$ 为从 s_1 得到 s_2 所付出的最小代价，其定义如下。

定义 3(编辑距离) 设字符串操作的集合 $Op=\{op_i \mid op_i: S\to S, i\in N, S$ 为字符串集合$\}$，$w: Op\to R$ 为操作的代价函数，则 s_1 与 s_2 的编辑距离：

$$\delta(s_1,s_2)=\min_{(op_i)_I,op_n(\cdots(op_1(s_1)))=s_2}\left(\sum_{i\in I}w(op_i)\right) \tag{1}$$

估算操作代价的方法有许多，一般将操作限定为插入、删除和修改等 3 种，且设定其操作代价均为 1。

在使用编辑距离计算相似度之前，一般需对式(1)中的编辑距离进行归一化处理，如式(2)的形式：

$$edis(s_1,s_2)=\frac{\delta(s_1,s_2)}{\max(length(s_1),length(s_2))} \tag{2}$$

式中，$length(s)$ 为字符串 s 中的字符数量。根据式(2)，文献[7]给出了一个基于编辑距离的相似度：

$$\sigma_{edis}(s_1,s_2)=\frac{1}{1+edis(s_1,s_2)} \tag{3}$$

5.2 基于 WordNet 的方法

基于 WordNet 的方法[7]是从单词自然语义的角度去考虑它们的相似度。获得单词自然语义最常用的方法是使用同义词词典，在英文处理领域最常用的词典是 WordNet[23]。

WordNet 是一个基于同义词集合的电子词典，并对其中的每个概念提供了包括定义和示例的文本描述，并通过概念间父子(SuperConcept/SubConcept)关系和部分整体(partof)关系，形成了一棵以概念为节点的语义树。文献[7]中给出了一个基于 WordNet 语义树的相似度：

$$\sigma_{WN}(t_i,t_j)=\frac{2\times\log(p(n_{ij}))}{\log(p(n_i))+\log(p(n_j))} \tag{4}$$

式中，t_i，t_j 为两个单词，n_i，n_j 为单词对应的 WordNet 语义树节点，n_{ij} 为 n_i 和 n_j 在 WordNet 语义树中公共最近父节点，$p(n_i)$ 为 n_i 的子节点占 WordNet 语义树中全部节点的比例。

5.3 基于 KNN 的方法

KNN(K Nearest Neighbors)，又叫 K-近邻算法，是机器学习算法中最基本的基于实例的学习方法。这个算法假定所有的待分类实例对应于 n 维空间 R^n 中的点。一个实例的最近邻是根据标准欧式距离定义的，即把任意的实例 x 表示为特征向量：

$$\langle a_1(x),a_2(x),\cdots,a_n(x)\rangle$$

式中，$a_r(x)$ 表示元素 x 的第 r 个属性值。那么元素 x_i 和 x_j 间的距离定义为：

$$d(x_i,x_j)=\sqrt{\sum_{r=1}^{n}(a_r(x_i)-a_r(x_j))^2} \tag{5}$$

在本算法中，K-近邻中的实例表示为本体中每个实体的描述信息。这里提取的描述信息主要包括概念/属性中由 RDFS 和 OWL 规范定义的元数据。

5.4 基于本体结构的方法

基于本体结构的方法是利用本体实体的结构信息发现匹配关系的本体匹配方法。本体数据可表示为带标签的有向图，以图中节点和弧分别表示本体实体及其语义关联。通过这些语义关联可以获得本体实体间的匹配关系。

Similarity Flooding 算法是著名的基于图结构的匹配算法[28]。在该算法中，首先建立本体实体间的初始相似度(如使用基于名称方法计算的相似度)。然后，根据待匹配本体的图结构信息建立相似度传播图，图中节点为来自不同本体的实体所构成的实体对。最后，利用不同实体对之间的关系，对实体对中两实体的初始相似度进行重新计算，以达到相似度传播的目的。

5.5 基于数据场和高斯函数的方法

针对大规模不平衡本体匹配问题，提出一个基于数据场和高斯函数的本体匹配算法[27]，并把它作为一种方法组件集成到 RiMOM2 中。

所谓不平衡本体匹配主要是指待匹配的本体双方，其中一个在本体实体或涉及领域的数量上远远大于另一个。包含实体或涉及领域多的本体称为重量级本体，记为 O_h；反之，称为轻量级本体，记为 O_l。该算法首先使用高效的相似度方法，如基于实体名称的方法，建立较小本体 O_l 中实体(如概念)对较大本体 O_h 的初始相关度；然后，基于数据场的理论，利用高斯函数引入周围本体实体对当前实体的影响，修正初始相关度，最终确定并抽取 O_h 中与 O_l 相关的子本体 O_S；最后，利用针对性的匹配方法对 O_l 与 O_S 进行更精确匹配，得到最终匹配结果。

由于 O_h 的规模远远大于 O_l 的规模，通常情况下抽取的子本体 O_s 的规模也会远远小于 O_h 的规模。由于大大降低了匹配规模，过滤掉了不相关实体，该算法在很大程度上提高了本体匹配的效果和针对性匹配方法的效率。同时该算法在初始相似度计算及最后的精确匹配上并不指定任何具体匹配方法，具有较强的灵活性。

该算法在 OAEI[7] 2008 FAO 本体匹配任务评测中取得了第一名的成绩[30]。

5.6 基于已有匹配结果的方法

基于已有匹配结果的本体匹配算法是一种基于本体外部信息，即背景知识[11,17]的匹配方法。由于参与匹配的本体数据本身所存在的局限性，导致了直接利用本体自身信息进行匹配的方法有时难以进一步提高匹配效果。而已有的一些匹配关系通过实践证明往往是较准确的，用它们来修正当前匹配，会在很大程度上提高匹配效果。该算法正是将已有匹配结果作为背景知识，利用多种中间本体选择策略发现中间本体，并以中间本体为桥梁，利用匹配的传递性，获得待匹配本体间的间接匹配结果，最后将待匹配本体间的直接匹配结果与间接匹配结果聚合起来，形成最终的匹配结果。

同时，该算法基于已有的匹配结果通常比较正确这一前提假设，在实际应用中多数情况下是合理的，因为历史匹配结果往往是被人工调整并经过实践检验的。但是，在某些特殊情况下，如果作为背景知识的已有匹配结果中有较多的错误，就有可能导致错误的传递，使匹配结果变坏。针对这一情况，为了克服背景知识不准确对匹配结果造成的不良影响，该算法中使用了迭代修正的方法予以解决。

该算法已经作为一种方法组件被集成到 RiMOM2 中，实验结果表明，上述算法可以有效地发现和利用背景知识，提高匹配的效果[14]。

6 评估方法及实验用例

6.1 评估方法

评估一个匹配方法的优劣，除了考虑其时间复杂度和空间复杂度，匹配效果也是必须要评价的，一般通过比较匹配的结果和标准匹配结果来得到。在实验评估上，本文采用传统的查准率、召回率以及 F1-Measure[13]作为检验标准。

假定给定标准匹配结果为 S，某个结果 R 的查准率、召回率以及 F1-measure 分别定义为：

$$\text{Precision}(R)=\frac{|S\cap R|}{R} \tag{6}$$

$$\text{Recall}(R)=\frac{|S\cap R|}{R} \tag{7}$$

$$\text{F1-measure}=\frac{2\times P\times R}{P+R} \tag{8}$$

6.2 实验用例

Benchmarks 是 OAEI（Ontology Alignment Evaluation Initiative）国际竞赛组织所使用的一个测试用例集，它包含共 51 个不同的本体，本体覆盖的领域是参考文献，表示语言为 OWL-DL，序列化为 RDF/XML。所有本体从 #101－#304 进行编号。其中 #101 作为参考本体，其它本体均与 #101 进行匹配。

测试集中有 43 个本体是在参考本体 #101 的基础上使用系统化方法进行修改得到的。使得 #101 中的某些信息可能被修改、扩展或丢弃。这种修改的目的是为了测试匹配工具在某一方面的性能。修改的对象主要包括：实体名称及其描述信息、层次结构、概念、属性、实例等；修改动作大致为随机替换、删除、断开连接等等。

为了方便测试，通过观察和简单的分析，本文将所有的 51 个本体根据修改方法的不同大致分为以下 5 类：

D1（#101－#104）：除 #102 外，其他 3 个本体在文本和层次结构两方面都基本一样。

D2（#201－#210）：这几个本体都是在 #101 的基础上对元素的名称和/或元素的描述信息进行一定的改变而得到的。

D3（#221－#247）：在这几个本体中，元素的名字和相应的描述信息全部得到保留。所不同的是，这些本体是不同类型的结构性信息缺失的组合。

D4（#248－#266）：这几个本体中，元素的名字全部为随机字符串，没有任何描述信息，只留下少量的层次信息。

D5（#301－#304）：这 4 个都是由第三方组织定义的参考文献领域的本体。通过这几个映射，可以检验整个系统的性能。

6.3 实验结果及分析

本部分主要通过设计 4 个匹配流程针对以上实验数据集进行匹配测试（D4 除外，D5 只选取 #304 单个数据集），并从查准率、召回率和 F1-Measure 三方面进行评价分析。匹配结果如表 1 所列，表中数据均取平均值（D5（#304）除外），其中的 P，R，F1 分别代表查准率、召回率和 F1-Measure。实验中所使用的 4 个匹配流程分别为：1）只使用基于编辑距离的匹配器，表 1 中的 EditDistance；2）只使用基于向量空间的匹配器，表 1 中的 VectorSpace；3）基于编辑距离和向量空间匹配器的简单聚合，表 1 中的 Ed＋Vs；4）基于编辑距离和向量空间匹配器经相似度传播进行聚合，表 1 中的 Ed＋Vs＋SF。

表 1

流程	EditDistance			VectorSpace		
序号	P	R	F1	P	R	F1
D1	60.06%	100.00%	73.21%	88.89%	98.97%	93.66%
D2	52.71%	47.97%	47.47%	51.97%	27.21%	33.67%
D3	57.74%	98.65%	72.89%	78.73%	81.33%	79.70%
D5（#304）	92.11%	86.49%	84.77%	83.12%	47.30%	62.50%
流程	Ed＋Vs			Ed＋Vs＋SF		
序号	P	R	F1	P	R	F1
D1	56.10%	100.00%	70.55%	82.76%	98.97%	90.14%
D2	51.65%	51.08%	48.72%	61.67%	59.19%	59.33%
D3	54.49%	98.65%	69.91%	65.05%	97.02%	77.42%
D5（#304）	74.71%	87.84%	80.75%	60.00%	89.19%	71.74%

6.3.1 查准率分析

图 6 分别给出了不同数据集上 4 个不同匹配流程所得到的查准率的对比结果。从图中可以明显看出基于向量空间的单个方法在数据集 D1 和 D3 上相对于其他匹配流程具有较强的优势，这是由于 D1 和 D3 数据集中文本和层次信息具有良好的均衡优势。D1 数据集上相似度传播对两个单个匹配结果的修正幅度也相当明显。而在实际的本体数据集 D5（#304）上，基于编辑距离的本体匹配方法则充分发挥了实例中文本信息的优势，而结构信息的利用反而影响了实际匹配的准确度。

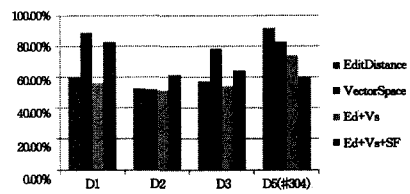

图 6 查准率

[7] http://oaei.ontologymatching.org/

6.3.2 召回率分析

图7分别给出了不同数据集上4个不同匹配流程所得到的召回率的对比结果。从图上很容易看出，对于数据集D1和D3，除D3上基于向量空间的方法显得稍微逊色了一些，4个匹配流程都能得到较好的召回率，这是由于数据集D3相对于D1在数据的层次结构信息上有所丢失。另外，在召回率上，基于向量空间的方法相对于其他匹配流程的结果都显得差了一些，尤其在实际本体数据集D5(#304)上的召回率相较其他匹配流程的结果有较大差距，这也是由于基于向量空间方法的特点所造成的。

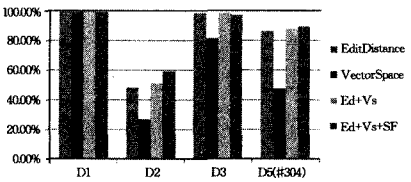

图7 召回率

6.3.3 F1-Measure分析

图8分别给出了不同数据集上4个不同匹配流程所得到的F1-Measure的对比结果。综合来看，基于向量空间的方法在数据集D1和D3上都有较好的表现。对于数据层次结构信息较完备的D1和D2数据集，相似度传播对综合两个单个匹配方法的匹配结果也显示了较好的提升作用。但综合来看，对于D2数据集的匹配结果，几种匹配方法都不能得到令人满意的匹配结果，这是由于数据集D2相对于D1做了大量的文本描述信息的改动，这也是这几个匹配方法都比较依赖于文本信息的缘故。

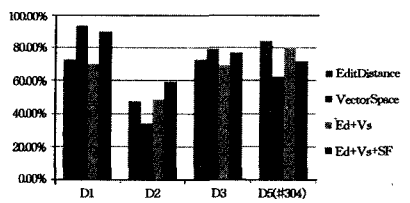

图8 F1-Measure

结束语 RiMOM2是一个严格按照分层设计、具有匹配策略组件易扩展、匹配流程可定制、匹配结果聚合自动化的本体匹配框架。随着本体匹配研究的深入，本体匹配方法日臻完善，实现一个能够应用于实际本体匹配任务的软件系统成为可能。然而，现存的众多本体匹配系统要真正意义上地适用于具有各种特性的实际本体数据的匹配任务，仍然面临着众多挑战。RiMOM2只能对初级用户隐藏不必要的阈值设定及参数设置，同时又只能通过对高级用户提供匹配流程的自定义功能来提高高级用户对具有针对性的本体匹配任务的匹配效果的要求。由此可知，要真正意义上实现能够对具体本体数据特性进行针对性的匹配操作自动化还需要科学工作者们更加努力地研究与探索。

参考文献

[1] Berners-Lee T, Hendler J, Lassila O. The Semantic Web [J]. Scientific American, 2001, 284(5): 35-43

[2] Borst W N. Construction of Engineering Ontologies for Knowledge Sharing and Reuse [D]. Enschede, Netherlands: Centre for Telematics and Information Technology, Twente University, 1997

[3] 邓志鸿，唐世渭，张铭，等. Ontology研究综述[J]. 北京大学学报：自然科学版，2002，38(5)：730-738

[4] Tang J, Li J, Liang B, et al. Using Bayesian decision for ontology mapping[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2006, 4(4): 243-262

[5] Qu Y, Hu W, Chen G. Constructing virtual documents for ontology matching[C]// Proceedings of the 15th International World Wide Web Conference(WWW). 2006: 23-31

[6] Euzenat J, Shvaiko P. Ontology Matching [J]. Berlin Heidelberg, Germany: Springer-Verlag, 2005

[7] Lin D. An Information-Theoretic Definition of Similarity[C]// Proceedings of the 15th International Conference on Machine Learning(ICML). 1998: 296-304

[8] Li J, Tang J, Li Y, et al. RiMOM: A Dynamic Multistrategy Ontology Alignment Framework[J]. IEEE Transactions on Knowledge and Data Engineering(TKDE), 2009, 21(8): 1218-1232

[9] Melnik S, Garcia-Molina H, Rahm E. Similarity flooding: a versatile graph matching algorithm and its application to schema matching[C]// Proceedings of the 18th International Conference on Data Engineering(ICDE). 2002: 117-128

[10] 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

[11] Shvaiko P, Euzenat J. Ten Challenges for Ontology Matching [C]// Proceedings of the 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE). 2008: 1164-1182

[12] Hu W, Jian N, Qu Y, et al. GMO: A graph matching for ontologies[C]// Proceedings of the K-CAP Workshop on Integrating Ontologies. 2005: 43-50

[13] Euzenat J. Semantic precision and recall for ontology alignment evaluation[C]// Proceedings of the 20th International Joint Conference on Artificial Intelligence(IJCAI). 2007: 248-253

[14] 仲茜，李涓子，李毅，等. 基于已有映射结果的本体映射[J]. 清华大学学报：自然科学版，2008，48(7)：1178-1181

[15] Gligorov R, Kate W T, Aleksovski Z, et al. Using Google distance to weight approximate ontology matches [C]// Proceedings of the 16th international conference on World Wide Web (WWW). 2007: 767-776

[16] Cilibrasi R L, Vitanyi P M B. The google similarity distance[J]. IEEE Transactions on knowledge and data engineering, 2007, 19(3): 370-383

[17] Aleksovski Z. Using background knowledge in ontology matching[D]. Eindhoven: Amsterdam and Philips Research, 2008

[18] Levenshtein V. Binary codes capable of correcting deletions insertions and reversals[J]. Doklady akademii nauk SSSR, 1965, 163(4): 845-848

[19] Madhavan J, Bernstein P A, Rahm E. Generic schema matching with Cupid[C]// Proceedings of 27th International Conference on Very Large Data Bases(VLDB). 2001: 48-58

[20] Melnik S, Rahm E, Bernstein P A. Rondo: A Programming Platform for Generic Model Management[C]// SIGMOD. 2003

[21] VEuzenat J, Valtchev P. Similarity-based ontology alignment in OWL-lite[C]// Proceedings of the 15th European Conference on Artificial Intelligence(ECAI). 2004: 333-337

[22] Ehrig M, Staab S. QOM-quick ontology mapping [C]// Procee-

dings of 3rd International Semantic Web Conference(ISWC). 2004:683-697

[23] Miller G A. WordNet: A lexical database for english[J]. Communications of the ACM, 1995, 38(11): 39-41

[24] Giunchiglia F, Yatskevich M, Shvaiko P. Semantic matching: Algorithms and implementation[J]. Journal on Data Semantics IX, 2007(4601): 1-38

[25] Do H H, Rahm E. COMA-A System for Flexible Combination of Schema Matching Approaches[C]// Proceedings of VLDB. Morgan Kaufmann, 2002: 610-621

[26] Aumueller D, Do H H, Massmann S, et al. Schema and ontology matching with COMA++[C]// SIGMOD. 2005: 14-16

[27] Zhong Q, Li H, Li J, et al. A Gauss Function Based Approach for Unbalanced Ontology Matching[C]// SIGMOD 2009

[28] Li Y, Li J, Zhang D, et al. Result of ontology alignment with RiMOM at OAEI'06[C]// Proceedings of the 1st ISWC International Workshop on Ontology Matching(OM). 2006: 181-190

[29] Li Y, Zhong Q, Li J, et al. Result of Ontology Alignment with RiMOM at OAEI'07[C]// Proceedings of the ISWC International Workshop on Ontology Matching(OM). 2007: 227-235

[30] Zhang X, Zhong Q, Li J, et al. RiMOM Results for OAEI 2008 [C]// Proceedings of the ISWC International Workshop on Ontology Matching(OM). 2008: 182-189

[31] Zhang X, Zhong Q, Shi F, et al. RiMOM Results for OAEI 2009 [C]// Proceedings of the ISWC International Workshop on Ontology Matching(OM). 2009: 208-215

(上接第 136 页)

由此产生大量的开销，因此簇首必须控制参与簇维护的网关节点数量。图 6 所示为 3 跳簇维护能耗的估算。由于节点发射功率相等，以发送 1 个 hello 报文的能量为 y 轴单位。图 6 中虚线为网关节点自发参与簇维护产生的能耗；实线为经过簇首协调、优化选择参与维护的网关节点后的能耗。仿真结果表明通过簇首的协调，可以大大降低网关节点进行移动节点管理的能耗。

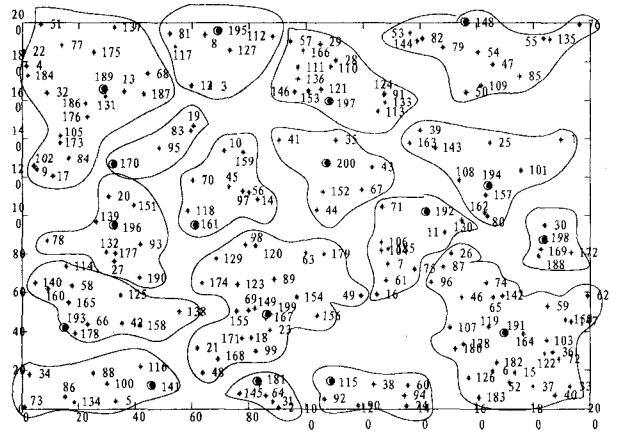

图 4 Max-Min 3 跳分簇的 Ad Hoc 网络

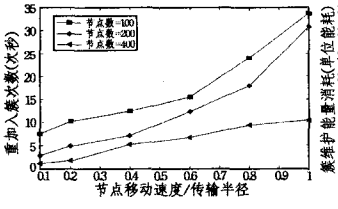

图 5 变更簇首的节点平均数目

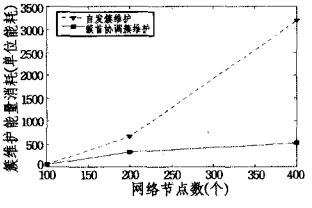

图 6 3 跳簇维护代价估计

结束语 本文提出的 k 跳分簇 Ad Hoc 网络协作框架，首先研究了移动节点的管理，在此基础上进行簇间和基于移动代理的全网络协作。宏观上讲，全网络的节点都参与到基本的簇维护协作中，但是不同类型的节点参与协作的方式不同。网关节点扫描邻居节点信息，管理节点的进出；节点在移动到边界附近开始扫描邻居，准备更换簇首；簇首扫描邻居节点进行簇首交接。

由于 k 跳簇的簇首管理的区域较大，簇首失去大部分簇成员而导致簇重构的概率较小，因此簇首的交接通常是由于簇首自身的原因。同样，成员即使移动速度较快，移出簇区域的过程也相对较长。这样 k 跳簇的簇维护需求，随着簇跳数的增大而减小。

本文提出的协作框架适用于不同的 k 跳分簇算法，但是 k 跳分簇网络的协作开销与分簇的情况密切相关。Max-Min 算法得出的分簇，会出现簇首位于簇边缘的情况，网关节点与簇首通信的开销偏大，因此有必要研究簇内通信开销最小的 k 跳分簇算法。

参 考 文 献

[1] Soltanali S, Pirahesh S, Niksefat S. An efficient scheme to motivate cooperation in mobile ad hoc networks[C]// Networking and Services. Athens, Greece: IEEE Computer Society Press, 2007: 98-103

[2] Buttyan L, Hubaux J. Stimulating cooperation in self-organizing mobile ad hoc networks[C]// Proc. ACM/Kluwer Mobile Networks and Applications (MONET). New York: IEEE Press, 2003: 579-592

[3] Buchegger S, Boudec J Y L. Performance analysis of the confidant protocol: cooperation of nodes fairness in dynamic ad-hoc networks[C]// Proc. of Mobihoc. Lausanne, Switzerland: ACM, 2002: 226-236

[4] Zhong S, Yang Y R, Chen J. Sprite: A simple, cheat-proof, credit-based system for mobile Ad Hoc networks[C]// Proc. of INFOCOM. San Francisco, USA: IEEE Press, 2003: 187-197

[5] 黄蕾，刘立祥. Ad hoc 网络寻路阶段的合作激励机制研究[J]. 计算机学报，2008，31(2)：262-269

[6] Bulut E, Zheng J, Wang Z, et al. Balancing cost-quality tradeoff in cooperative ad hoc sensor networks[C]// IEEE Military Communications Conference, MILCOM. San Diego: IEEE Press, 2008: 1-7

[7] Amis A D, Prakash R, Tai H P. Max-Min D-cluster formation in wireless ad hoc networks[C]// Proc. IEEE INFOCOM. New York: IEEE Press, 1999: 1-10

[8] Leng Su-peng, Zhang Li-ren, Chen Yi-fan. K-hop compound metric based clustering scheme for ad hoc networks[C]// Proc. IEEE International Conference on Communications. Seoul: IEEE Press, 2005: 3396-3400

[9] Shen C C, Srisathapornphat C, Liu R, et al. CLTC: A cluster-based topology control frame-work for ad hoc networks[J]. IEEE Transactions on Mobile Computing, 2004, 3(1): 18-32