

# 基于 Ontology 的智能检索系统框架研究与实现

陶 兰, 杨 睿, 陈 冲

(深圳大学信息工程学院, 深圳 518060)

**摘 要:** 提出并实现了一个语义 Web 环境下基于 Ontology 的语义信息检索系统框架。它由信息搜集 Agent、存储 Agent、推理 Agent 和查询 Agent 构成, 这 4 类 Agent 协同工作实现了基于语义的 Web 信息检索服务。该文采用 OWL 描述的 Ontology 作为语义推理的基础, 以 Jena 作为 RDF API, 使用 JADE 作为多 Agent 系统的开发工具和应用平台, 完成了一个针对“Wine”领域进行智能查询的实例。

**关键词:** 语义 Web; 多 Agent 系统; Ontology; 智能检索

## Research and Realization of Intelligent Retrieval System Frame Based on Ontology

TAO Lan, YANG Rui, CHEN Chong

(College of Information Engineering, Shenzhen University, Shenzhen 518060)

**【Abstract】** The paper puts forward and realizes an ontology-based semantic information retrieval system frame in semantic Web. The system is composed of information agent, storing agent, reasoning agent and searching agent. These 4 agents realize the Web information retrieval service by the collaboration. The paper is based on the ontology described by OWL, taking Jena as the RDF API, JADE is used as developing tools and application platform of multi-agent system. As a result, the authors finish an intelligent searching instance for “Wine” field.

**【Key words】** Semantic Web; Multi-agent system; Ontology; Agent retrieval

目前Web上的数据由于缺少语义信息难以实现自动化处理。为解决此问题, W3C组织开展了语义Web项目的研究, 目标是能够对Web上的信息实现智能化和自动化的处理及应用。面对海量信息, 智能信息检索一直是相关领域的重要研究课题。目前Ontology应用于信息检索的著名项目包括(Onto)<sup>2</sup>Agent<sup>[1]</sup>、Ontobroker<sup>[2]</sup>和SKC<sup>[3]</sup>。(Onto)<sup>2</sup>Agent帮助用户在Web上已有的Ontology中检索所需要的Ontology, 并保存各类Ontology的元数据; Ontobroker面向Web上的网页资源, 帮助用户检索到所需的网页内容; SKC解决信息系统语义异构的问题, 希望通过在Ontology上的一个代数系统来实现Ontology间的互操作, 从而实现异构系统之间的互操作。

虽然目前已经出现了上述关于语义 Web 技术的一些应用, 但是利用 Agent 和 Ontology 技术将语义 Web 的各部分结合起来并实现语义 Web 环境下完整的信息检索功能的系统框架还不多见。本文融合语义 Web 及 Agent 等相关技术, 借鉴语义 Web 在知识表示、RDF(Resource Description Framework)数据操作和语义查询以及知识推理等多方面的成果, 提出并实现了一个基于 Ontology 的能够实现语义信息检索的多 Agent 系统框架, 目的在于根据用户的需要自动搜索并存储 Web 内容描述信息, 为用户提供符合条件的基于语义的信息搜集及推荐服务。相比目前的 Web 检索服务, 可以在很大程度上提高检索的查准率和查全率。

### 1 智能检索系统的体系结构和工作流程

语义 Web 环境下, Web 上的信息资源已进行了语义标注。在此环境下, 本系统应完成的任务是: 自动抽取关于 Web 内容描述的元数据, 并将这些元数据进行存储; 对用户提交的关键词进行语义匹配和语义相关性扩展, 形成符合特定

Ontology 语言的查询语句; 通过对数据库中存储的 Web 元数据信息的查询, 得到高效的基于语义的 Web 信息检索结果。

上述任务可划分为元数据收集、元数据存储、基于 Ontology 的知识推理和用户查询等 4 个主要功能。本文将整个系统框架划分为 4 个部分, 每个部分的功能由一类 Agent 实现, 这 4 类 Agent 通过协同工作, 共同完成整个检索任务。本系统的总体结构如图 1 所示, 除 4 类 Agent 外, 还包括一个用于存储 Web 元数据的数据库, 一个为推理 Agent 提供领域知识支持、以 OWL(Web Ontology Language)文件形式存在的 Ontology。

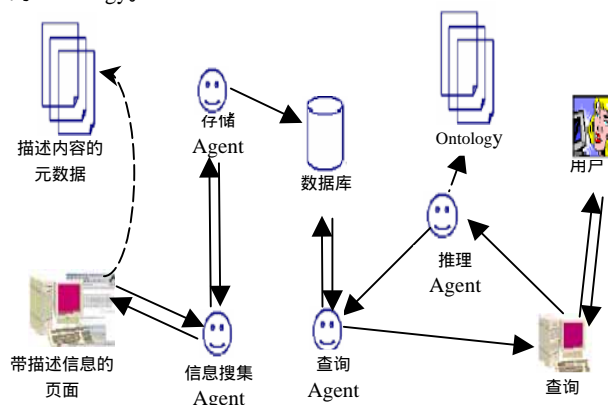


图1 智能检索框架的体系结构

**基金项目:** 深圳市科技计划基金资助项目 (200333)

**作者简介:** 陶 兰(1956 - ), 女, 博士、教授、博导, 主研方向: 语义 Web 与语义网络; 杨 睿、陈 冲, 硕士生

**收稿日期:** 2006-01-30 **E-mail:** szchenchong@21cn.com

系统框架的工作流程如下：

(1)信息收集 Agent 对语义 Web 环境中描述页面内容的元数据进行信息提取和过滤，把收集到的元数据交由存储 Agent。

(2)存储 Agent 接收语义元数据，将这些元数据以合适的方式存储到数据库中，数据库既可以是传统的关系数据库，也可以是专用的 XML/RDF 数据库。

(3)利用描述领域知识的 Ontology，由推理 Agent 对用户提交的查询关键词进行语义匹配和语义相关性扩展，将得到的基于 Ontology 的查询要求交由查询 Agent 执行。

(4)由查询 Agent 接受推理 Agent 产生的针对 Ontology 的查询要求，对存储元数据的数据库采用基于知识的方式进行查询，并将满足用户条件的结果返回给用户。

## 2 基于 Ontology 的知识推理与查询

语义 Web 环境下智能检索的核心技术是知识推理能力。本节以“Wine”领域进行智能查询为例，重点介绍推理 Agent 的知识推理引擎。

### 2.1 关于 Wine 的 Ontology 建模

智能推理引擎的工作基础是表达特定领域知识的 Ontology。本文在 W3C 发表的一个关于 Wine 的 Ontology 的基础上，加入一些关于 Person 和 City 的定义，将此 Ontology 作为推理 Agent 所参照的领域知识，它采用 OWL DL 作为描述语言。下面按照通常开发 Ontology 的步骤介绍关于 Wine 的 Ontology 建模：

(1)定义 Ontology 中的 Class。

(2)按照类别层次排列 Class，图 2 是根据 Wine 的分类方法所定义的 Ontology 的层次结构。

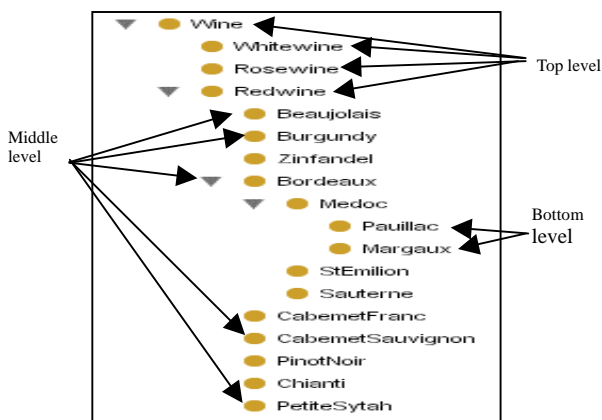


图 2 Ontology 中 Wine 的类层次结构

(3)为每个类定义它的 Property 和 Property 所允许的值，即各种约束条件(如值的个数, Cardinality)、Domain 和 Range 值等。

(4)为 Individual 中的各 Property 填充合适的值：创建 Individual 时首先要选择此 Individual 所属的 Class，然后填上它的 Property 的值，例如 ChateauDYchemSauterne 是一种具体的葡萄酒，它属于 Sauterne 类，原料来自于 SauvignonBlancGrape 和 SemillonGrape，重口味(Strong)，生产厂商为 ChateauDYchem。

### 2.2 基于 Ontology 的知识推理与查询

推理 Agent 的推理引擎应该能根据领域 Ontology 中对各概念的定义推断出这个关键词在查询中的精确语义及与此关键词语义相关的词语和概念。本系统的推理 Agent 使用 Jena<sup>[4]</sup> 中的 Ontology API。推理 Agent 通过调用 Jena 中的 Ontology API，根据 Ontology 对用户输入的关键词进行基于语义的推

理。推理分为两种：关键词的语义匹配和相关性扩展。

#### (1)语义匹配

语义匹配的作用是对用户所输入的关键词进行语义的分析，针对的是具有一词多义的词条。推理 Agent 根据 Ontology 判断此关键词的精确语义，从多个可能的语义选项中选择出最符合用户要求的一项，从而提高检索结果的精确性。传统的搜集引擎只是根据单纯的关键词匹配进行检索，不能区分同一词条的不同含义。例如：用户需要检索有关 Bordeaux 酒的相关信息，检索的结果可能包括关于 Bordeaux 城市的介绍，一个叫 Bordeaux 的个人的主页，也可能包括真正需要的关于 Bordeaux 酒的介绍。在本检索框架中，Ontology 对 Bordeaux 的几个不同语义进行了精确的定义，其中作为 RedWine 子类的 Bordeaux 具有 Body、Color、Flavor、Sugar 等 Property，且 Red 恰好位于 Property Color 的 range 中。属于 Person 类的 Individual 虽然也叫做 Bordeaux，可是它所具有的 Property 的 range 中并无 Red。属于 City 类的名叫 Bordeaux 的 Individual 也是如此。由于三者在本 Ontology 中的 URI 都是独一无二的，因此推理 Agent 参考了 Ontology，知道用户所要查询的不是 Bordeaux 城市，也不是 Bordeaux 的个人，而是一种叫 Bordeaux 的葡萄酒。

#### (2)相关性扩展

在 Ontology 中定义了众多的与查询关键词相关的其它概念，在获得查询词条的精确语义后，根据 Ontology 还可获取更多与此关键词语义关联的词。例如通过 subclass 关系，系统可知道有一种叫 Sauterne 的酒也属于 Bordeaux 酒，而 Sauterne 酒中还有一种叫 ChateauDYchemSauterne 的酒，这是一个 Individual 且属于 Sauterne Class。这样系统就会知道这些 Class 和 Individual 也都属于 Bordeaux 酒。在获得了这些知识后，虽用户的查询中并无 Sauterne 和 ChateauDYchemSauterne 这样的关键词，但系统会知道这些词与用户所要查询的 Bordeaux 酒关系相当密切，在检索时尽管有些页面内容并不包括 Bordeaux 酒的内容，但只要包含了有关 Sauterne 或 ChateauDYchemSauterne 的信息，系统也会把这样的页面返回给用户，这些内容很有可能是用户所需的。

推理 Agent 最后会将语义匹配和相关性扩展所得到的结果(包括 NameSpace、Resource 和 Property)封装起来以 ACLMessage 对象的形式交给查询 Agent，由查询 Agent 根据推理结果对存储有描述网页内容的 RDF 数据库进行查询。

## 3 用 JADE 实现智能检索系统框架

本系统选择 JADE(Java Agent Development Framework)<sup>[5]</sup> 作为 Agent 平台和开发工具。目前尚不存在真实的语义 Web 环境，本系统采用 Omnihttpd 软件作为 Web 服务器构建了一个虚拟的语义 Web 环境，其中所有的页面都含有以 RDF 形式表示的对页面内容进行描述的元数据。图 3 是 JADE 监控界面中看到的搜索 Agent 和存储 Agent 正在运行的情况。

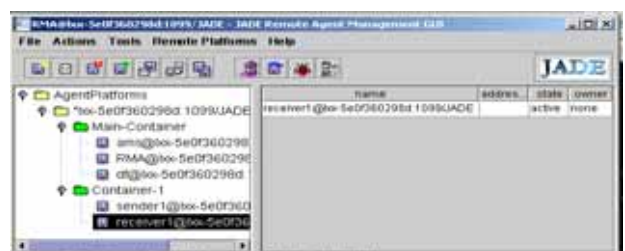


图 3 信息收集 Agent 和存储 Agent 在 JADE 平台中运行

图 4 为本系统运行时,信息搜集 Agent 向存储 Agent 发送所搜集到的元数据的情况,其中 INFORM()表达传送的信息,箭头表达信息传送的方向,双击某个箭头能看到程序传送的某条信息。

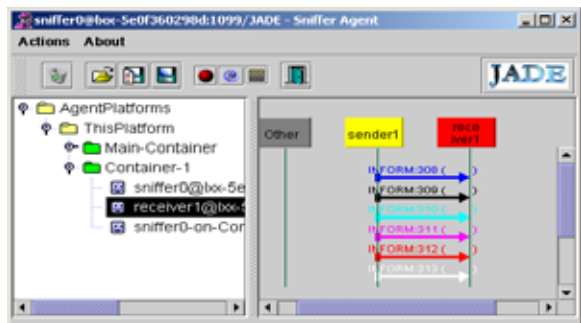


图 4 信息搜集 Agent 向存储 Agent 发送所搜集到的元数据

推理 Agent 可以根据 Ontology 对用户输入的查询要求进行语义匹配和相关性扩展,推理的结果会以 ACLMessage 的形式发送给查询 Agent,以上一节中 Bordeaux 酒为例,其消息内容如图 5 所示,其中包括了名字空间前缀和所查询资源的简写形式。

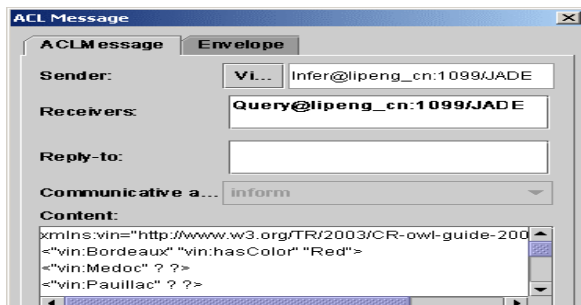


图 5 推理 Agent 发送给查询 Agent 的 ACLMessage

查询 Agent 收到消息后,会将此消息进行解析,并封装成 RDQL 语句。在存储网页元数据的 RDF 数据库中,每一

个模型就是一个 Web 单元,查询出此模型之后,很容易得到它的 URL 和 Title,查询 Agent 把这些信息返回给用户,完成整个的查询过程。例如用户输入 Bordeaux Red 之后进行语义查询所返回的最后结果如图 6 所示。

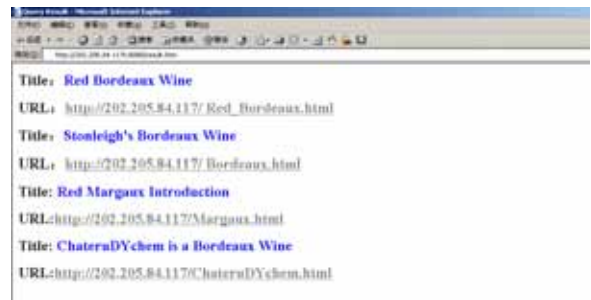


图 6 查询 Agent 返回的最后结果

## 4 结语

由于目前真正的语义 Web 环境尚不存在,本文工作只是验证了在模拟环境下的可行性和有效性。下一步工作是将本系统的主要部件及其功能转变为网格环境下的服务,研究在网格环境下实现基于语义的智能信息查询将面临的问题和求解方法。

## 参考文献

- 1 Apirez J, Perez A G, Lozano A, et al. (Onto)<sup>2</sup>Agent: An Ontology-based WWW Broker to Select Ontologies[C]. Proceedings of the Workshop on Application of Ontologies and Problem-solving Methods, UK, 1998: 16-24.
- 2 Ontobroker[Z]. <http://ontobroker.aifb.uni-karlsruhe.de/>.
- 3 SKC[Z]. <http://www-db.stanford.edu/skc>.
- 4 Jena[Z]. <http://www.hpl.hp.com/semweb/index.html>.
- 5 Nowostawski M, Bush G, Purvis M, et al. Platforms for Agent-oriented Software Engineering[Z]. <http://secml.otago.ac.nz/agents/Assets/documents/dp2000-13.pdf>.

(上接第 190 页)

果要好得多,较神经网络更优良。特别是当所寻找的函数本身比较复杂时,IGEP 方法越发能够显示出其优越性,在变异算子的改进上,由于考虑到了变异的随机性和普遍性,使得种群始终保持着多样性,实验表明这种改进加快了种群的收敛速度,优越于经典 GEP 算法。

## 4 结论

用 IGEP 进行复杂函数自动建模,不需要知道各因素间的因果关系,只需要提供足够的实验或者实验数据,无需知道目标函数,就可以达到准确预测的目的<sup>[3,8]</sup>,并得到准确的函数表达式。实例计算和分析可知,该方法优于传统方法和神经网络方法及经典 GEP 方法。此外,对 IGEP 的程序设计上,本文采用了单指令多数据(SIMD)技术,充分发挥了算法内含的并行性,从而使得算法在求解速度上远远快于普通的算法,并使得改进后的 GEP 方法可以广泛应用于对时间要求比较严格的反问题研究中。

## 参考文献

- 1 Ferreira C. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems[J]. Complex Systems, 2001, 13(2): 87-129.
- 2 Mitchell M. An Introduction to Genetic Algorithms[M]. Mit Press, 1996.
- 3 Kang Lishan, Li Yan, Chen Yuping. A Tentative Research on Complexity of Automatic Programming[J]. Wuhan University Journal of Natural Science, 2001, 6(1/2): 59-62.
- 4 左 劼. 基因表达式编程核心技术研究[D]. 成都: 四川大学, 2004.
- 5 周爱民, 曹宏庆, 康立山等. 用遗传程序设计实现复杂函数的自动建模[J]. 系统仿真学报, 2003, 15(6): 797-799.
- 6 李有法. 数值计算方法[M]. 北京: 高等教育出版社, 1996.
- 7 Candida F. Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence[EB/OL]. <http://www.gene-expression-on-programming.com/>.
- 8 Candida F. About APS 3.0[EB/OL]. <http://www.gepssoft.com/>.