面向OWL知识的问答系统:Agile

赵喜清[1] 张利明[1] 高明霞[2]
(河北北方学院信息科学与工程学院 张家口 075000)[1] (北京工业大学计算机学院 北京 100124)[2]

摘 要 问答系统因能提供方便的输入模式与更精确的答案而成为获取网络信息的重要手段。介绍了一个面向OWL知识的问答系统Agile,并着重阐述了其在问题规范和字典生成方面的技术方案。为了得到合适的映射单位,Agile定义了两个数据结构用于规范自然语言问题和OWL本体知识字典,并借助一些自然语言处理工具和OWL解析工具将两种源知识进行了形式化。和现有网络问答系统比较,Agile不需要用户参与,能够处理的问题领域和形式更加丰富。

关键词 问答系统,OWL知识,问题规范,字典生成
中图法分类号 TP311 **文献标识码** A

Question-Answering System Based on OWL Knowledge: Agile

ZHAO Xi-qing[1] ZHANG Li-ming[1] GAO Ming-xia[2]
(College of Information Science & Engineering, Hebei North University, Zhangjiakou 075000, China)[1]
(College of Computer Science, Beijing University of Technology, Beijing 100124, China)[2]

Abstract Existing Question-Answering systems for the Web focus on source of text. The Web Ontology Language (OWL) is recommended by W3C as the standard for knowledge representation and exchange on the Internet in 2004. So QA based on OWL knowledge becomes an important research. A QA system named Agile was presented in this paper, and the detail schemes of formulating questions and indexing OWL were described. In order to acquire mapping unit, Agile defined two data structures for formulating questions and OWL and formatted them through existing natural language processing technology and OWL parsing method. Agile is auto and can deal with more kinds of questions than former QA based on OWL systems.

Keywords Question-answering, OWL knowledge, Formulating questions, Building dictionary

1 引言

问答系统(Question Answering System, QA)[1]支持自然语言格式的用户问题并能返回简洁、准确的答案,又称为人机对话系统(Human-machine conversation, HMC)。伴随着不同知识表示方式的出现和发展,研究、开发针对不同知识源和不同应用的各类问答系统吸引了众多研究者的兴趣。网络本体语言(Web Ontology Language, OWL)[2,3]是2004年W3C推荐的用于表示网络本体知识的工业标准。随着OWL语言规范的标准化,大量的个人和学术团体专注于自动学习和手动建立各种各样OWL版本的本体知识(简称为OWL知识)并将其发布于网络。基于OWL知识的问答系统[4-6]成为面向Web问答系统研究的新探索,这类系统兼顾了基于知识库问答系统的推理能力和基于Web问答系统的知识获取方式。QACID[4], Aqualog[5], Agile[6]是其中的重要成果。QACID只能处理电影领域的问题。它自动建立了一个问题模式(question pattern)和SPARQL匹配的数据库,但是需要手动收集大量问题作为学习实例。Aqualog专门处理英语问题,对问题进行了手工分类,但是其问题理解过程中概念和关系的识别过程是半自动化的,需要用户的参与。由于使用了问题手工分类模板,其可处理的问题范围只有带"who"和"what"问题标记词的问题。Agile可以用OWL知识回答用户的面向事实的句子级英语问题,能够实现字典生成、问题规范、语义转换、语义重组、答案推理等主要步骤。为了得到合适的映射单位,Agile定义了两个数据结构用于规范自然语言问题和OWL知识,并借助一些自然语言处理工具和OWL解析工具将两种知识进行了形式化;为了减少用户干预并利用源问题中的语义信息,Agile将问题语义转换形式化为一个模糊约束满足问题,并将不同的影响因素表示为不同的软约束,利用最小-最优解答和Leximin排序将包含语义信息的问题成分翻译成了OWL知识元素;语义重组以OWL语言规范生成的模版为基础,采用模式匹配方式生成了有效的RDF图模式;答案推理借助了推理引擎Pallet。和现有OWL知识问答系统比较,Agile不需要用户参与,能够处理的问题领域

到稿日期:2012-09-25 返修日期:2012-12-22 本文受国家自然科学基金(60496322,60496327),国家科技支撑计划课题"农村医疗卫生知识库及远程医学服务系统及应用"(2012BAH05F04)资助。
赵喜清(1970—),男,硕士,副教授,主要研究方向为无线网络、知识数据库,E-mail:zxqlytqq@163.com;张利明(1974—),男,硕士,讲师,主要研究方向为图像处理、人工智能;高明霞(1975—),女,博士,副教授,主要研究方向为知识工程、数据挖掘。

和形式更加丰富。因为篇幅所限，本文着重介绍系统的整体结构以及问题规范和字典生成技术，其中第 2 节介绍 Agile 的系统结构和各部分的功能；第 3 节介绍问题规范和字典生成技术；第 4 节验证了 Agile 在几个真实 OWL 知识库上的效果；最后对系统进行总结并指出进一步的研究方向。

2 Agile 系统结构

如图 1 所示，Agile 整个系统分为字典生成、问题规范、语义转换、语义重组、答案推理 5 个功能部分。这 5 个功能部分的处理流程可以分为字典处理模式和问题处理模式两种形式。字典处理模式主要完成 OWL 知识到领域字典的知识索引，这个过程是知识预处理过程，通常离线完成。问题处理模式需要将用户的英语自然语言问题规范成带属性集的词集，通过语义转换和语义重组将自然语言问题翻译为等价的 RDF 元组，答案推理改写这些元组成为有效的 OWL 查询（SPARQL 等）并借助现有推理引擎 Pallet 从 OWL 知识库中获取到了 OWL 格式的答案。

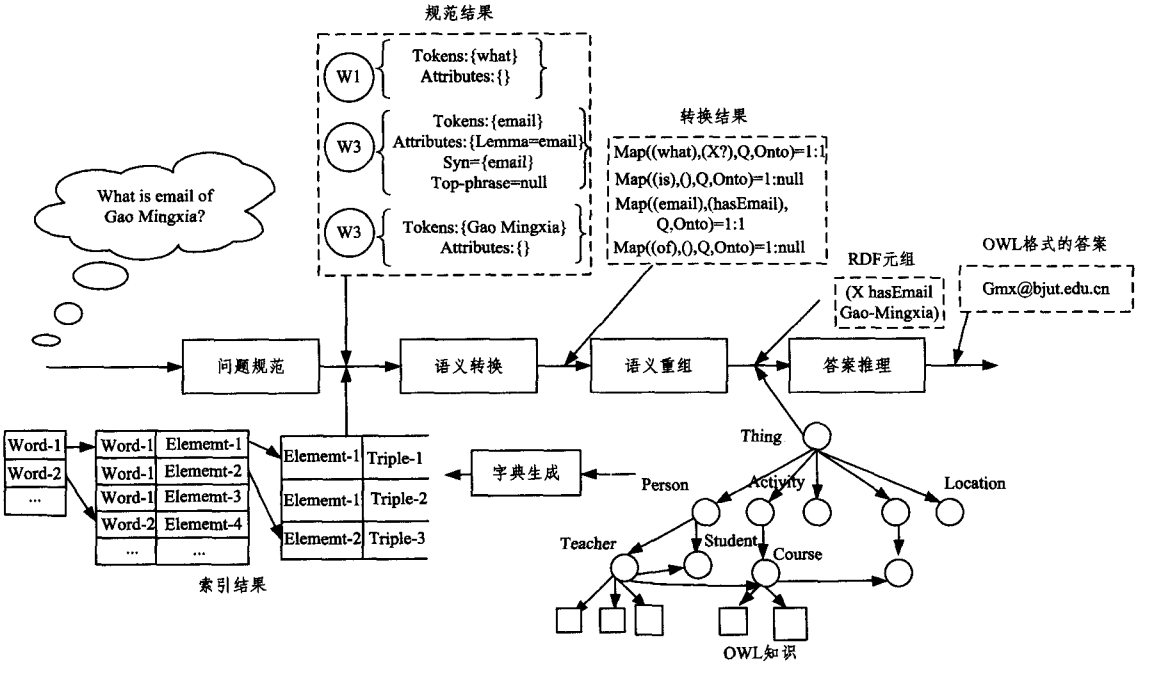

图 1 Agile 的系统结构

2.1 问题规范

问题规范是将用户输入的自然语言问题形式化为 Agile 问答系统内部知识的过程。Agile 的内部知识是 OWL 知识形式，因此 Agile 的问题规范过程就是将自然语言问题表达成 OWL 知识形式的过程。从 OWL 知识库角度看，元素（通常指类、对象属性、数据属性和个体）是组成知识的基本单位；而从自然语言体系的语法组成看，词（或者实体）是表达实际语义的最小单位，元素由词（或者实体）按一定语法规则构成。因此，问题规范过程是 Agile 系统对问题最小化处理的过程，由此获得自然语言中的词和 OWL 知识库中的元素，是处理用户输入自然语言问题到带属性词的重要步骤。

2.2 字典生成

字典生成将 OWL 形式的多层知识（例如公理、元素）分解索引成自然语言的词，最终目的就是根据已有 OWL 知识库生成一个用于检索的领域字典，领域字典的最小索引形式就是词，用于初步检索问题词的语义信息。详细技术见第 3 节。

2.3 语义转换

在问题规范和字典生成的基础上可以进行两种知识形式的横向映射，即语义转换。具体地说，就是在问题词集和元素集之间建立语义等价映射的过程。对一个问题词语义的理解受很多因素影响，常见的有语法因素和词形因素等。这些因素的影响程度和影响优先级各不相同，为了统一、量化地评估这些影响因素，Agile 将这个映射过程形式化为一个模糊约束最优问题。

2.4 语义重组

OWL 知识库的查询是以 RDF 图模型为基础的，语义重组的关键是将语义转换结果重新组合成 RDF 图模式（RDF 元组）。特定类型的 OWL 元素可以组合成几种有限的 RDF 图模式，利用这些规范形成的模板和一些隐含属性识别方法就可以完成 Agile 中的语义重组。

2.5 答案推理

答案推理主要完成 RDF 图模式到选定查询语言的重写与提交符合规范的 OWL 查询给 OWL 知识推理机这两项任务。SPARQL 是 W3C 推荐的标准 RDF 查询语言，语法类似于 SQL，可以通过图形模式匹配实现对多个 RDF 图的查询。因此 Agile 选择 SPARQL 作为 OWL 知识库的查询语言。RDF 图模式和 SPARQL 有各自的语法规范，答案推理根据这些规范将查询组合结果重写成了格式有效的 SPARQL 查询，并将其提交给支持 SPARQL 查询的推理引擎 Pallet[1]。

1) http://pellet.owldl.com/

3 Agile 系统实现

问题规范的最终目的就是将问题中蕴含的各种词法、句法、语法和语义信息转换为合适的形式提供给 Agile 的后续功能模块。为此，Agile 定义了一个统一的数据结构用于表示各种自然语言问题，即：

问题是一个满足顺序关系的词集 $W=\langle w_1,\cdots,w_i,\cdots,w_n\rangle$，而 $w_i:=(T,A,TP)$ 表示词集中词在具体输入问题中的语言特征。其中，$T:=\{character_j\}_1^i$ 是组成该词的字符序列集；A 是该词通过标注获取到的属性集，目前的 Agile 版本主要关注词性（POS）、词干（Stem）、同义词（Syn）、等价词（Equ）、命名实体类型（NEType）等；TP 指向包括该词的词组。

问题规范的主要任务就是获得这个结构，整个过程需要各种各样的自然语言处理技术[7]，大致流程分为：分割和标注两部分，详细过程见算法 1。图 2 提供了一个具体输入问题被规范成有序词集的实例。从图 2 可知，一个原始问题首先通过分割（tokenization）获得一组候选词。然后，通过不同的标注技术增加这些词的语法和语义属性，例如使用一个预先定义的 stop 词典为候选词中的 stop 词作标注，或者利用实体识别技术标注各类实体，比如人名、地名、数字、日期等。最后，通过各种语义字典，比如 Gate[8]、WordNet[9] 等工具扩展同义词、等价词、缩写词等属性。

算法 1 问题规范

//Input：Q 是自然语言问题。
//Ouput：W 是词集
1. 借助自然语言处理工具 API 分割 Q 成为候选词集 W；
2. for W 中每个 w do
  2.1. 词性标注；
  2.2. stop 标注；
  2.3. 命名实体标注；
  2.4. 同义词标注；
  ……
end
5. 返回 W

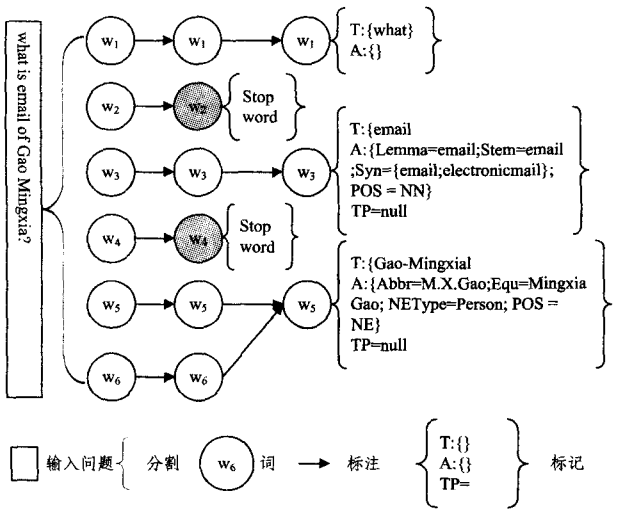

图 2 问题规范实例

和问题规范类似，字典生成任务需要将 OWL 知识库中潜在的一些知识组织成一个特定数据结构用于理解问题。图 3 是 Agile 中使用的平面字典结构。这个结构将以 RDF 三元组（triple）形式表示的公理知识、元素（class，individual，DatatypeProperty，ObjectProperty，value）等 OWL 知识库中的知识特征以自然语言中的最小语义单位、个体词进行索引，并形成一个链式数据结构。

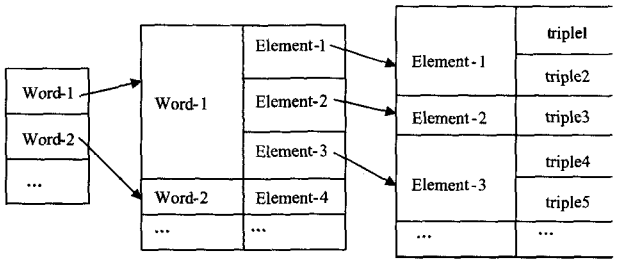

图 3 知识库索引结构

为了获取 OWL 知识库特征，Agile 使用了现存的 OWL 知识解析器：Jena[9]。一个 OWL 元素通常由一个词或多个词组成。单个词就是自然的索引词。由多个词组成的元素通常都遵循一定的命名规则，常见的命名规则有大写或连字符“—”等。为了分割这些多词元素，可以统计一下目前最常用的命名形式，并根据具体使用情况选择相反的过程进行分解。OWL 知识中有些成分例如数据属性可使用的值（value）经常是一个较完整的自然语言片段，例如词组、短语甚至于句子等。处理这些成分就需要将和问题规范类似的自然语言处理技术用于建立索引词。详细情况可参考算法 2。

算法 2 字典生成

//Input：KB 是 OWL 格式的知识库。
//Ouput：D 是图 3 格式的字典。
1. 根据字典结构建立空字典 D；
2. 借助 Jena 解析 KB 获取类集、属性集、个体集、值集；
3. 对类集中每个类
  3.1. 如果是单个词，建立〈词，类〉数据项并插入字典；
  3.2. 如果是多词组合
    3.2.1 根据具体命名规则分词
    3.2.2 为每个词建立〈词，类〉数据项并插入字典；
4. 对属性集中每个属性
  4.1. 如果是单个词，建立〈词，属性〉数据项并插入字典；
  4.2. 如果是多词组合
    4.2.1. 根据具体命名规则分词
    4.2.2. 为每个词建立〈词，属性〉数据项并插入字典；
  ……
end
5. 返回 D。

4 实验

4.1 实验数据

为了验证 Agile 的效果，利用不同渠道获取了 3 个不同主题的 OWL 知识库。第一个是 institute. owl 知识库，是国际 WIC 研究院[2] 研究的数据结构模型及其手工扩充，具体包含 91 个类、70 个属性、120 个个体、1768 个公理。第二个知识库——people-pets. owl 描述的是人、动物和宠物等概念及它们间的关系，具体包含 59 个类、16 个属性、27 个个体、565

[2] http://www.iwici.org/

个公理。它的原始概念来自于 Racer[3] 的例子知识库。实例和关系是根据一些介绍宠物的 Web 站点扩充的。第三个知识库——references. owl 描述的是和参考文献相关的概念和关系，具体包含 37 个类、71 个属性、114 个个体、864 个公理。它是 EON Ontology Alignment Contest[4] 所用的参考本体。它的实例更像是自动生成的，实例名称使用了人类很难理解的符号，例如“a456080390”。这些实例通常附属有两个和名称等价的属性“title”或是“label”。为了消除符号名称，预处理过程中用等价属性值替换了符号名称。

表 1 提供了和上述知识库匹配的 3 个问题集。第一个问题集的问题包括两个来源。一部分是参考 Webclopedia[5] 中问题类型模拟生成的，另一部分是直接从 WIC 研究院的学生那里收集到的。其余两个问题集中的问题都是参考 Webclopedia 中问题类型手工模拟生成的。从表 1 可见，问题集中包含着 18 个祁使类问题（im 表示祁使类问题），在进行问题规范实验时，这些问题要手工重写为带问题标记词的问题。

一个自然语言问题能否被一个给定知识库中的知识正确理解，除了理解技术的影响外，更重要的条件是：知识库中是否包含有足够理解这个问题的知识。为了消除由于知识不足引起的误差，表 1 中的问题通过人类专家进行了手工检查，确认是可以由对应知识库回答的。

表 1　问题集的分类

问题集	what	which	who	whose	how	when	where	im	sum
institute	21	24	22	7	6	5	4	4	93
people-pets	26	11	22	5	6	0	0	5	75
references	23	21	6	2	7	5	3	9	76

4.2　实验结果及分析

Agile 的问题规范模块采用了现有的自然语言处理工具，这些现有技术在分割和标注问题时不可避免地会产生少量错误。Agile 的后续模块会将问题规范结果作为输入，问题规范引入的错误就会影响其他模块的精度，并最终影响 Agile 系统的精度。通过模块的单独测试得出 3 个问题集（institute，people-pets，references）在问题规范模块的精度分别是：92.3%，96.05%，96.05%。

为了消除问题规范的影响，手工纠正了这些分割和标注错误，形成了一个无规范错误的问题集。图 4 提供了 Agile 在 3 个数据库上针对两个问题集的精度结果。从图中可知：Agile是领域无关系统，在消除了问题规范错误后，3 个问题集的精度都达到了 80%。由于 3 个知识库的结构特点相差较远，其中第二和第三个知识库中包含了大量的匿名个体和匿名类，这些通过属性限制声明的类在知识库索引时很难生成一致的三元组公理知识，因此可能会影响语义转换模块的精度，并最终影响整个 Agile 的精度。

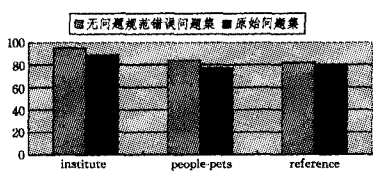

图 4　Agile 在 3 个数据集上的精度比较

结束语　本文介绍的基于 OWL 知识的问答系统 Agile，和以前系统相比有以下贡献：(1)系统是全自动的，在问题规范、语义转换以及字典生成等关键步骤无需用户和领域专家参与。(2)系统是领域无关的，在 3 个不同领域的知识库和问题集上的实验，其最优准确率都达到了 80%。开发面向 Web 的问答系统是一个具有相当难度的前沿课题，在以下方面 Agile 也没有取得完全令人满意的结果：(a)在多变量问题处理方面，语义转换精度仍然需要提高，语义重组需要新的技术；(b)在问题领域定位方面，Agile 仍然是静态的，一次部署只适合于解决一个领域的问题，领域实时动态选择仍然是个难题；(c)在知识库获取和更新方面，Agile 也没有实现动态化，一次部署只适合于利用现有的 OWL 知识库，不能增量式更新。

参 考 文 献

[1] 王树西. 问答系统：核心技术、发展趋势[J]. 计算机工程与应用，2005，18(41)：1-3

[2] Horrocks I, Patel-Schneider P, Harmelen F. From SHIQ and RDF to OWL: The Making of a Web Ontology Language[J]. Journal of Web Semantics, 2003, 1(1): 7-26

[3] Smith M K, Welty C, McGuinness D L. OWL Web Ontology language guide[OL]. http://www.w3.org/TR/owl-guide/

[4] ′Oscar, Ferr′andez, Izquierdo R, et al. Addressing ontology-based question answering with collections of user queries[J]. Information Processing and Management, 2009, 45: 175-188

[5] Lopez V, Pasin M, Motta E. AquaLog: An Ontology-Portable Question Answering System for the Semantic Web[C]// ESWC 2005, LNCS 3532. 2005: 546-562

[6] Gao M X, Liu J M, Zhong N, et al. Semantic Mapping from Natural Language Questions to OWL Queries[J]. Computational Intelligence, 2011, 27(2): 280-314

[7] Jurafsky D, Martin J H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition[M]. Prentice Hall, 2000

[8] Cunningham H. Software Architecture for Language Engineering[D]. Department of Computer Science, University of Sheffield, June 2000

[9] Fellbaum C, et al. WordNet: An Electronic Lexical Database [M]. MIT Press, 1998

[10] Brian M. Jena: A Semantic Web Toolkit[J]. IEEE Internet Computing, 2002, 6(6): 55-59

3) http://www.sts.tu-harburg.de/ r. f. moeller/racer/

4) http://oaei.ontologymatching.org/2004/Contest/

5) http://www.isi.edu/natural-language/projects/webclopedia/