基于OwlReady2的智能人机对话系统

**宋丛威**1

1(浙江工业大学之江学院 理学院 绍兴 312030)

**摘要** OWL是用于本体论知识管理的描述语言，而OwlReady2是它的Python接口，OwlReady2封装了推理引擎HermiT和Pellet, 可以进行基于描述逻辑 (DLs) 的推理。利用Python的动态编程功能，实现具有推理能力的人机对话系统。为使用自然语言，还建立了一个文法解析模块。系统可以通过简单对话任务的测试。

**关键词** OwlReady2；描述逻辑；推理引擎；人机对话系统

**中图法分类号：**TP311 **文献标识码：**A

**Intelligent Human-Machine Dialogue System based on OwlReady2**

SONG Cong-wei1

*1(Department of Sience，Zhijiang College Zhejiang University of Technology， Shaoxing 312030)*

**Abstract** OWL is a description language for the management of ontological knowledge, and OwlReady2 is its Python interface, that encapsulates the reasoning engine HermiT and Pellet, and can make reasoning based on description logics (DLs). Utilizing Python's dynamic programming, the human-machine dialogue system with reasoning ability based on OwlReady2 is realized. In order to use natural language, a grammar parsing module is built. The system can pass the testing of simple dialogue task.

**Keywords**  OWLReady2; Description Logics; Reasoning Engine; Human-Machine Dialogue System

引言

本研究的最终目的是建立一个具备逻辑推理能力和知识存储的人机对话系统。这种系统已经被创造出来，并可以被更多的软件或计算机语言实现，比如可以实现一阶逻辑推理的Prolog，CLIPS，和Python实现的PyKE，还有基于描述逻辑和语义网络的推理引擎[1]。不过使用它们，需要学会一种计算机语言和现代逻辑学基本原理。有一些网络机器人[2,3]能够实现自然语言交互，但是不具有逻辑推理能力。这些网络机器人主要依赖基于统计学的自然语言处理方法，对语法要求不大，但问答的逻辑性不强。人们自然会想到设计一套解析程序，将自然语言翻译成计算机语言，然后交给软件进行推理。这样的解析程序和软件构成的系统具备了两种功能：既能用自然语言交流，又能进行逻辑推理。

本文的特点是用OwlReady2（Python对OWL的封装）[4]实现自然语言人机交互。我们会写一个解析器，将自然语言翻译成OwlReady2对象，然后交给一个对话流程，在这个流程中，读取并处理这些对象。每一次交互，程序至少应该完成下面几个任务：

1. 解析陈述句，并作为知识存储，如果包含未知信息，应该反问用户。还允许检验与原有知识的一致性。
2. 解析一般疑问句，判断真假。同样的，应该反问用户未定义词语。
3. 解析特殊疑问句，计算疑问词对应的对象。

设计系统最初的动机是，通过人机交互的形式存储专业知识，同时提供自动推理、一致性检验。因此本系统既可以进行日常式对话，也能构建各种领域的专家系统。

DLs、OWL与OWLOready2简介

DLs一般被认为是一阶逻辑的子集，尽管术语、符号风格上有不小的区别，其最大的特点是可判定性。DLs由三个主要概念构成：个体、概念、关系，分别对应于一阶逻辑中的个体常项、一元谓词、二元谓词。[4-9]

列出几个公式的对应关系（语义等价）及汉语例子，仅供参考。完整的对应关系见文献[4,7]。

表 1 一阶逻辑与DLs的对应关系

|  |  |  |
| --- | --- | --- |
|  |  | “八公”是狗 |
|  |  | “八公”是“教授”的朋友 |
|  |  | “八公”喜欢（某些）人类 |
|  |  | “八公”只喜欢同类 |
|  |  | 狗是一种动物 |

OWL是一种服务于网络的本体标记语言，用于构建本体论知识库[5,7,10]。它和网络标记语言HTML/XML风格上非常相似，其实就是XML的一种具体实现。OWL可以表示DLs的每一种表达式，而OWL推理器也能实现DLs的推理。

OwlReady2是用Python实现的OWL封装。熟悉Python的读者来说，可以少学一种新语言。当然，它也实现了DLs的推理，以及“封闭世界假说”。内置的推理引擎是HermiT和Pellet（基于Java）。[1,3]即使不能保证OwlReady2实现了DLs的所有推理形式，也因为它是用Python写的开源软件，我们可以进行功能上的几乎无限制的扩展。本文就可以看做是一个扩展，即增加一个解析模块和动态化本体信息存储模块。动态化本体信息存储模块负责信息的动态读写，因为信息都是在人机交互中读取、修改和添加的；解析模块可以让我们采用自然语言与机器沟通，而不是直接用计算机语言或逻辑公式。

DLs和OwlReady2中的表达式（函数、类、运算符）都是一一对应的[4]。因此，如果用户熟悉DLs，一种更为简单的选择是把DLs翻译成OwlReady2对象。此外，OwlReady2已经支持SWRL规则。新版本的开发会考虑利用SWRL规则[11]。

人机对话系统架构

在引言中，我们已经提到了建立对话系统的思路和任务，本节介绍系统架构。

对话过程类似于计算机语言的读-写-打印循环（REPL）。每一个输入的字符串都被计算机解释成可执行的命令，而这些命令的涵义是知识库状态的变换。如输入，则知识库自动添加 即

。

在变换过程中，我们可以设计各种扩展程序，记录元信息，丰富推理系统。如记录对话的历史，此时命令表示

,

其中是命令历史列表,在历史列表末尾添加元素。还可以将分为成功执行的和未成功执行的。

下图是本系统的架构图。系统设计遵循MVC模式。用户通过一个操作界面与系统各模块互动。用户只需输入表示语句或命令的字符串。解析器会把它们解析成有意义的数据。交给OwlReady2和Python的其他模块处理。

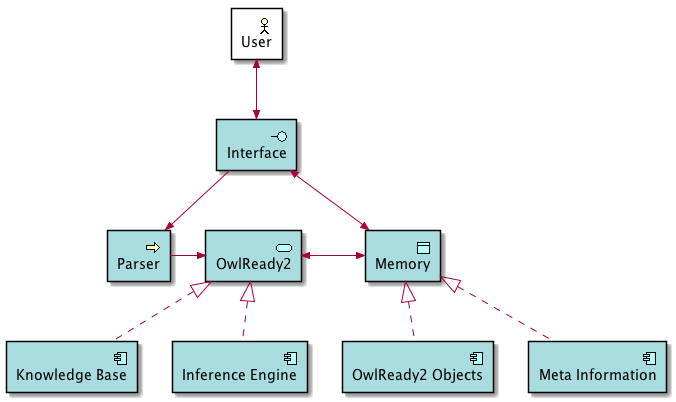


图1 人机对话系统架构图

Fig. 1 Architecture Diagram of Human-Machine Dialogue System

接着是用户与系统交互的流程图。用户除了可以输入可被解析的语句，还可以输入命令，便于在交互中，主动控制系统，比如查看变量的值与类型，删除或修改之前输入的表达式解析成的知识和本体。

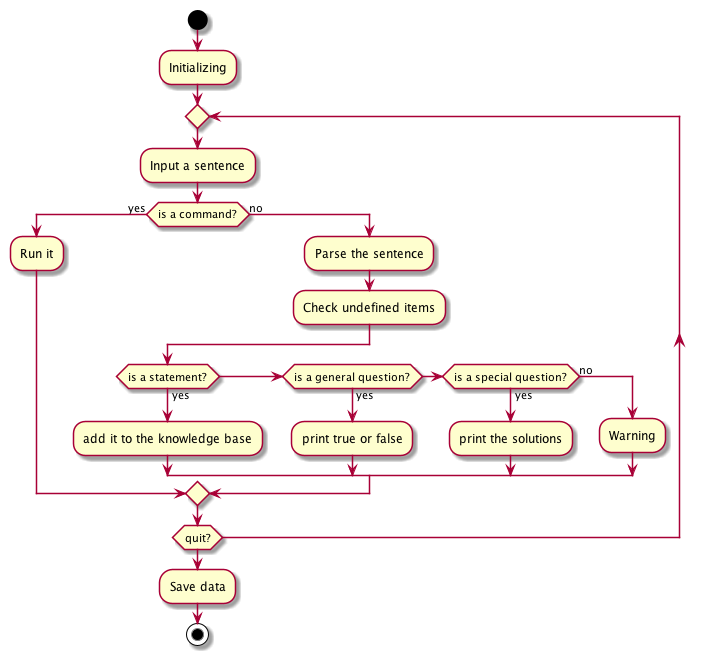


图2 人机交互流程图

Fig. Human-machine interaction flow chart

对人机对话系统实现

本节讲解实现思路和一些技术细节，主要包含语义学和文法解析两个方面。

逻辑推理是由OwlReady2完成的，我们只负责调用，OwlReady2也能自动存储知识，管理知识库。剩下最困难的是解析过程。我们采用纯Python实现的文法解析库pyparsing，它具有良好的扩展性，可以根据需要改造解析结果。语义本质上是上文提到的知识库变换。语义学关系到如何改造解析结果。

除此之外，对话系统还具有如下功能：历史记录，提供元信息，以及命令交互。

3.1 对话语义学

形式上，每一次输入的句子都被理解成是本体知识库（DLs表达式的集合）到自身的变换。对于陈述句，这个变换就是加入新的表达式，但不必有返回值；而对于疑问句，这个变换是恒等变换，同时返回推理引擎给出的该问题的答案。本文的目的是建立一个实用的人机对话程序，并不打算对这个变换进行严谨的形式化的讨论，只是给出变换的大致形式。所有形式均为：状态状态，返回值。

1. 陈述句：

如果进行一致性检验，则定义如下，

1. 一般疑问句：
2. 特殊疑问句：

其中是本体知识库,代表的解析结果，即DLs表达式，是个体变量或概念变量。如果采用封闭世界假设，那么一般疑问句的回答通常只会是真或假。而特殊疑问句牵涉变量，而变量的取值不一定是唯一的。这实际上是动态DLs的操作语义解释[12-15]。更严格的形式化理论会留到以后讨论。

3.2自然语言解析

最大的负担是解析自然语言。自然语言和DLs表达式之间没有明显的对应关系。比如，“狗是人类的朋友”和“人类喜欢狗”在自然语言中是两种很不相同的句式，但在DLs中都可能表述为

.

而“狗是忠诚的朋友”又有不同表达。为了简化系统设计，也只能限制可被解析的自然语言。

受DLs的ABox和TBox启发，本文对自然语言（汉语）分成如下几类：

个体+（不）是+概念，如“八公”是（一只）狗。

概念+（不是）是一种+概念，如秋田犬是一种狗。

概念/个体 +（不/只）关系+ 概念/个体，如人类喜欢狗。

注意，本文对第3类语句的语义解释有如下规定。“人类喜欢狗”表示人类喜欢某些狗，即“人类喜欢.狗”。只有“人类只喜欢狗”才被解释为“人类喜欢.狗”。如果把个体理解成只含一个个体的概念，那么可以一定程度上简化分类。

我们只在这些语句类型上建立形式文法。它们都是简单的“主宾谓”结构。因此，真正能被解析的汉语语句并不多。此外，代表个体的名词必须加上引号，动词和形容词会注上词性标签，便于识别；词语之间需加空格，进行手动分词。目前分词技术已经很成熟了，增加一个分词模块是非常容易的。

3.3 疑问句的处理

从句式上看，汉语的陈述句与疑问句没有什么不同。这是汉语解析的简单之处。一般疑问句仅仅是在陈述句最后加上疑问语气词。然而，正如你在上一节看到的，其语义却较为复杂。疑问句表达式翻译成OwlReady2中的代码是

A in a.is\_instance\_of 或 A in a.is\_a

即使这很不自然。这仅能判断直接的从属关系。如果只知道，那么要用

A in a.INDIRECT\_is\_instance\_of

甚至不得不递归判断。

特殊疑问句的解答有一定困难。经典的DLs并没有变量的概念，不能直接表达一个包含变量的公式。特殊疑问句虽然和陈述句语序一致，但是不能直接利用陈述句解析结果。解答，可用A.instances，而这种含“高阶”变量的疑问解答，可用a.is\_a，如“地球是什么？”。其他类型的疑问句还有

等等。

疑问句有一个特殊问题，回答不能是没有用的知识，比如，“地球是什么样的行星？”，回答不应该是“天体”，因为天体的范围比行星还大。形式地，的答案不应该包括的所有父概念，也不能是形如的概念，DLs中这些概念使天然地成立。

3.4 测试

为了便于测试和调试，输入的语句及其对应的回答都事先准备好，然后让程序自动检验，并不是交互模式。交互模式下的对话测试也已录屏并上传至<https://www.bilibili.com/video/av56821908>。输入文本对语法要求极其严格，目前均由作者编制，不能选用一般的测评语料[3]。图3是其中两个测试对话字典。

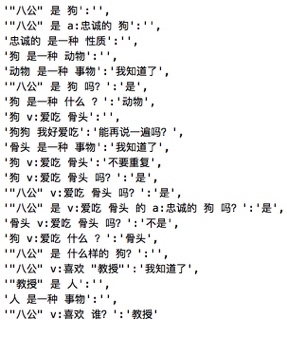
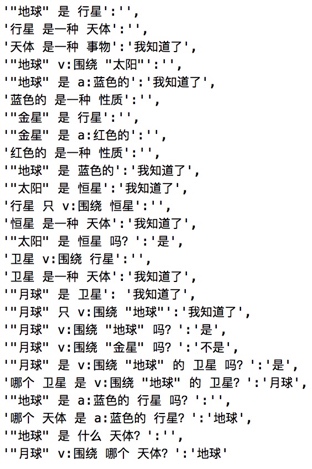


图3 问答字典（空字符串表示不必判断）

Fig. 3 Dialogue Dictionary (the empty string means it need not be judged)

这里已经实现了历史记录的功能：系统会记录所有未被正确求值但解析成功的句子，如第一句“‘八公’是狗”，此时“狗”还没有定义，因此整个句子不能计算语义，一旦给出“狗”的定义后，该语句会自动求值，避免重复。系统也允许给关系下定义，如“喜欢是一种对称关系”。

正如前文提到的那样，第一个测试只采用无修饰成分的主谓宾结构，词语之间用空格分隔，个体词语要用引号标记，动词和形容词词性会被标注。测试任务基本包含了所有句型，包括个体和概念之间的关系以及相应的疑问句，甚至还使用否定词。如果有错误，测试会自动报错，也就是说程序通过了测试。程序执行时间基本上由OwlReady2及其内置推理引擎决定，语法解析和一些封装代码不会对执行速度有显著影响。

源代码、测试程序（包括编制的测试对话）以及系统设计图均已上传至<https://github.com/Freakwill/gimbiseo>，供分享和交流。开发还在继续，源文件会不定期更新。

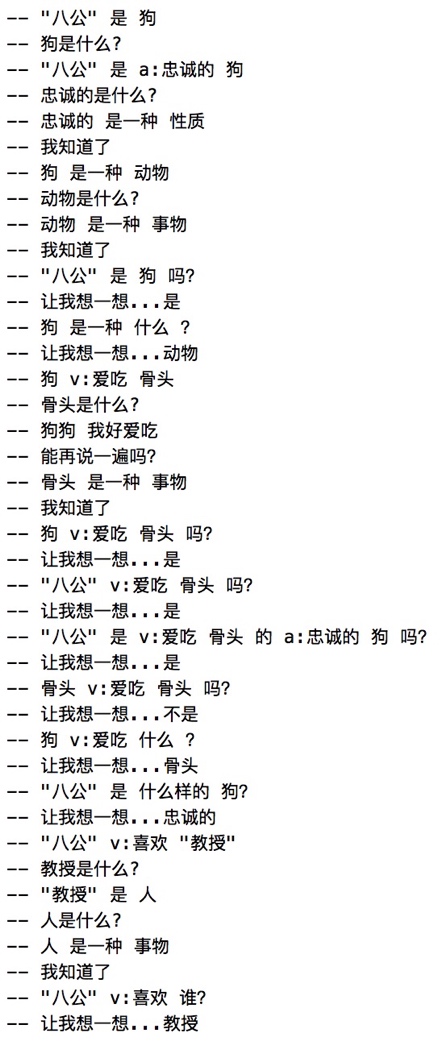


图4 人机交互演示

Fig. 4 Demonstration of human-machine interaction

1. 总结

根据测试结果，该系统能够顺利完成几个基本的对话任务。成功为后续开发奠定基础。

目前，该系统目前处于初期开发阶段，功能还比较简单，主要受限于自然语言语法的特殊性。目前通过测试的语句只限于主谓宾的简单结构，允许使用复合名词。可使用的量词还很有限，但不难进行扩展。以后的工作重心是研制更强大的语言解析器。

系统界面目前还停留在命令行的形式上，需要根据系统特点设计一个用户友好的界面[16]。

系统一开始处于“空断言”的状态，不存在专业知识。对对话的主题也不敏感。当系统在特殊领域投入使用时，应该事先建造相关的知识库。

另一个更有野心的计划是，和机器学习方法进行结合，如神经网络[17]。这样做可以拓展程序的能力，但远远超出本文研究范围。

此外，我们也正在开发适合编程的形式语言，避免引入自然语言中非逻辑因素。人机互动过程则是这种语言的REPL。

参考文献

[1] Priyanka P, Priyanka T. Interpreting Inference Engine for Semantic Web[J]. International Journal of Advanced Research in Computer Engineering & Technology, 2013, 2(2): 2278-1323.

[2] 图灵机器人, [http://www.turingapi.com/[OL](http://www.turingapi.com/%5BOL)].

[3] 王昊奋, 劭浩等. 自然语言处理实践：聊天机器人技术原理与应用[[M](http://www.turingapi.com/%5BOL)]. 北京: 电子工业出版社, 2019.

[4] Lamy J B. OwlReady: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies[J]. Artificial Intelligence in Medicine, 2017, 80:11-28.

[5] Baader F, Hanschke P. A scheme for integrating concrete domains into concept languages[C]. In: proceedings of International Joint Conference on Artificial Intelligence. Sydney, Australia, 1991, 452-457.

[6] 戴伟民. 语义网信息组织技术与方法[M]. 上海：学林出版社, 2008.

[7] 陆建江，张亚非，苗壮等. 语义网原理与技术[M]. 北京：科学出版社, 2007.

[8] 王昌龙. 基于描述逻辑的大规模本体推理关键技术研究[D]. 天津大学, 2017.

[9] 王金环, 李宝敏. 基于本体DL的语义推理研究[J]. 计算机技术与发展, 2009, 19(11): 94-96.

[10] Segaran T, Evans C, Taylor J. Programming the Semantic Web[M]. O’Reilly Media, Inc. 2009.

[11] 初林洁. 基于本体和描述逻辑的SWRL推理研究[D]. 广西师范大学, 2014.

[12] Asher N, Lascarides A. Logics of Conversation[M]. Cambridge University Press, 2003.

[13] 常亮, 史忠植, 陈立民等. 一类扩展的动态描述逻辑[J]. 软件学报, 2016, 21(01).

[14] 郝国舜, 马世龙, 眭跃飞. 一种扩展的动态描述逻辑语言及其 Tableau 算法[J]. 智能系统学报, 2009, 4(03): 226-233.

[15] 史忠植. 高级人工智能[M]. 北京：科学出版社, 2011.

[16] 薛静. 计算机软件用户界面特点及其应用[J]. 微型电脑应用, 2019, 35(1): 110-112.

[17] Mao, Jiayuan, Gan, Chuang, Kohli, Pushmeet, et al.. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision, 2019.

电话：13575464594

详细地址：绍兴市柯桥区越州大道958号，浙江工业大学之江学院

电子邮箱地址：williamzju@yahoo.com

身份证号码：330283198607050050