基于本体技术的高血压知识库平台构建*

杨美洁 张 兴 熊相超

（重庆医科大学医学信息学院 重庆 400016）

〔摘要〕 详细阐述基于本体技术的高血压知识库平台构建，包括本体库以及 Web 端界面等，指出该平台可以为高血压诊断和治疗提供依据，为其他慢病知识库平台构建提供借鉴。

〔关键词〕 高血压；知识库平台；本体；Python；精准化

〔中图分类号〕R－056 〔文献标识码〕A 〔DOI〕10.3969/j.issn.1673－6036.2019.04.013

Building of Hypertension Knowledge Base Platform Based on Ontology Technology *YANG Meijie, ZHANG Xing, XIONG Xiangchao, Chongqing Medical Informatics College, Chongqing Medical University, Chongqing 400016, China*

〔**Abstract**〕 The paper elaborates on the building of hypertension knowledge base platform based on ontology technology, including the ontology base and Web interface, etc., points out that the platform can provide basis for the diagnosis of and treatment for hypertension and provide references for the building of knowledge base of other chronic diseases.

〔**Keywords**〕 hypertension; knowledge base platform; ontology; Python; precisely

1 引言

高血压是最常见的慢性病，也是心脑血管病最主要的危险因素[1]。随着我国经济的发展和人口的老龄化，高血压患病率持续增加，高血压引起的冠心病、脑卒中等疾病的致残率、致命率高[2]，在我国心脑血管疾病死亡的第 1 位危险因素是高血压[3]。目前我国医疗资源紧张导致看病难等问题，在人工智能、大数据时代背景下，将新兴信息技术应用到医疗服务中，使患者在家中通过网络就能得到医疗建议，缓解就医压力。

目前关于高血压本体构建和知识库平台的研究主要包括：张宇[4]等构建高血压非药物治疗知识库 Web 端界面，从 Web 获取的大量文档，利用文本分类技术以及词频（Term Frequency，TF）和文件频率（Document Frequency，DF）方法提取文档和类别特征，通过支持向量机（Support Vector Machine，SVM）方法对文档分类，最后建立本地高血压非药物治疗知识库。吴昊[5]等提出基于本体和案例推理的高血压诊疗系统的框架结构。巩沐歌[6]等将高血压疾病、知识库和本体结合起来，构建具有推理功能的高血压知识库。张巍[7]等提出基于本体和案例推理的高血压诊疗系统模型。构建高血压领域本体及推理规则，使用 Jess 推理机进行推理操作，使用 Jena 实现对本体库和案例库并行的查询。李博[8]等结合本体方法将文本临床指南转变成临床指南知识库。本文利用 Python 爬虫技术爬取网络高血压数据，通过本体技术和 Protege 工具构建高血压本体

〔修回日期〕 2018－11－07

〔作者简介〕 杨美洁，硕士，讲师，发表论文 8 篇。

〔基金项目〕 重庆市社会事业与民生保障科技创新专项（项目编号：cstc2015shms－ztzx10003）；2018 年医学信息学院学生科研与创新实验（项目编号：2018C003）。

库，描述领域概念及其之间的约束和联系，将其存储在 Mysql 数据库中，本体构建完成后以 RDF/XML 形式存储，用于网络本体语言（Web Ontology Language，OWL）或规则推理。使用 Jieba 分词与正则化技术对用户输入的自然语言进行分词处理，Jena 推理引擎返回结果，采用 Python Web 的 Django 框架进行构建前台可视化界面。

2 高血压本体库构建

2.1 构建流程

高血压知识库平台构建流程，见图 1。Studer 等在 1998 年对本体定义为本体是共享概念明确的形式化规范说明[9]。本文参照《中国高血压防治指南 2017》版，结合 Python 爬取的高血压网络数据、相关文献图书等资料，借鉴 7 步法和骨架法，采用美国斯坦福大学开发的本体编辑软件 Protege 5.0 软件进行本体的构建[10]。主要构建高血压的症状体征、检查检验、药物等。

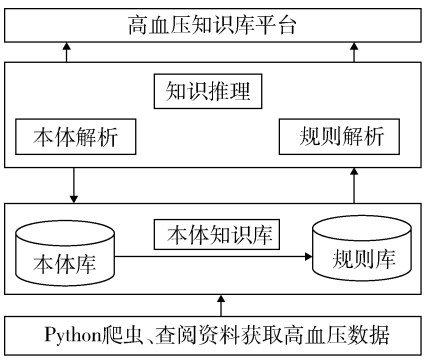

图 1 高血压知识库平台构建流程

2.2 本体模型

高血压本体模型，见图 2。本文构建高血压的领域本体包括症状体征、检查检验和药物。其中症状体征主要表现为：头晕、恶心、呕吐、咳嗽、心悸、尿频、四肢麻木、下肢水肿等。检验检验主要包括血压、血尿素氮、肌酐、低高密度脂蛋白、胆固醇、三酰甘油等。抗高血压药物主要包括 ACE 和 ARB、α 受体阻滞剂、β 受体阻滞剂、抗高血压药物、拮抗剂、利尿剂等。利用 Protege 5.0 为高血压本体构建 3 大类，分别是检查检验、药物、症状。Protege 中有两个属性定义，分别是类属性和关系属性。检查检验类属性项目、结果、单位、参考值；药物类属性：药物名、副作用；症状类属性：症状名、症状概述。构建 3 个类之间的关系属性完成本体的构建。本体构建完成后以 RDF/XML 形式存储，用于 OWL 或规则推理。Jena 是一个开源的 Java 语义网框架，可构建语义网和链接数据应用。Jena 利用 TDB 组件将上述构建的 RDF 形式的高血压知识本体存储起来，再通过资源描述框架定义集（Resource Description Framework Schema，RDFS）、OWL 以及 Jena 的 Rule Reasoner 进行本体推理，进一步自动识别补全数据，避免数据缺失、失真等情况。最后使用 Fuseki 组件，通过 SPARQL 语言对 RDF 数据进行查询，实现高效的知识提取。

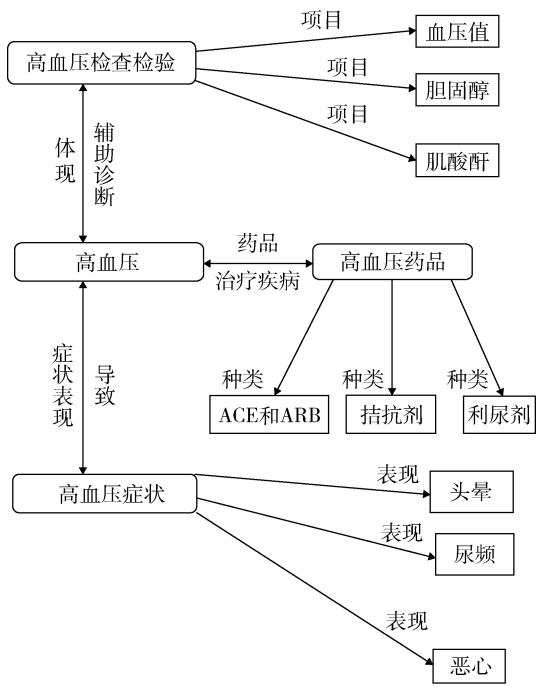

图 2 高血压本体模型

3 高血压知识库 Web 端

3.1 概述

利用 Python Web 框架构建高血压知识库。通过 Python 的 Django 框架开发高血压知识库的 Web 端界面[11]。用户在使用 Web 端进行查询时需要将输入的自然语言转换成计算机识别的 SPARQL 语句，因

此要用 Python 正则 Refo 模块、中文分词 Jieba 模块，实现对高血压知识中字符串及词句段切、关键字提取等，将自然语言转化为 SPARQL 语句，解析返回查询结果。

3.2 Jieba 分词

在 Python 的 Jieba 模块中加载自定义字典可实现对自然语言较准确的分词。以输入“高血压症状体征有哪些?”为例，利用 Python 的 Jieba 模块分词的部分代码和结果如下：

```
# jieba 自动分词
words = jieba. cut (hyper_ str)
print ('- - - - - - - - - - - - - - - - 默认分词效果 jieba - - - - - - - - - - - - - - - - ')
print ('/'join (words))
# 加载自定义字典
jieba. set_ dictionary ('sym. txt')
words = jieba. cut (hyper_ str)
print ("- - - - - - - - - - - - 加载自定义字典后，分词效果 - - - - - - - - - - - - - - - - -")
print ('/'join (words))
- - - - - - - - - - - - - - - - - - - - jieba 默认分词效果 - - - - - - - - - - - - - - - - - - - - - - -
高血压/疾病症状/有/哪些/?
- - - - - - - - - - - - - - - - 加载自定义字典后，分词效果 - - - - - - - - - - - - - - - -
高血压/疾病/症状/有/哪些/?
```

3.3 词性标注及关键字提取

为提高检索查询结果的效果和效率，需要对自然语言进行词性标注[12]。词性标注（Part - of - Speech Tagging）是指为分词结果中每个字符串标注一个词性，避免出现汉语歧义问题，进一步提高分词效率、精确度。对上述例句进行词性标注和关键词提取的部分代码和结果如下：

```
##词性标注及关键字提取
print ('- - - - - - - - - - - - - - - - 词性标注及关键字提取 - - - - - - - - - - - - - - - - ')
import jieba. posseg as pseg
words = pseg. cut (hyper_ str)
for word, flag in words:
    print ('% s % s'% (word, flag))
- - - - - - - - - 词性标注及关键字提取结果 - - - - - - - - -
高 a
血压 n
疾病 n
症状 n
有 v
哪些
? x
```

3.4 正则 Re 及 REfO

用户在 Web 端进行检索时会输入某些问题，本文采用正则为每个问题设定语义模板，主要使用 Re 和 REfO 两种正则模块，两者的区别是 REfO 适于任意序列的对象，而 Re 则是匹配字符串。用户在 Web 端进行检索时，平台首先利用 Re 模块将用户的问题分词处理后与 Jena 后端数据进行匹配，如果匹配成功则返回相应结果，否则失败。Re 和 REfO 模块代码如下：

```
class W (Predicate):
def_ init_ (self, token = ". * "pos = " * )
    # 正则表达式
    self. token = re. compile (token + " $ "
    self. pos = re. compile (pos + " $ "
    super (W, self). _ init_ (self. match)
    def match (self, word):
        m1 = self. token. match (word. token. decode ('utf - 8'))
        m2 = self. pos. match (word. pos)
        return m1 and m2
def apply (self, sentence):
    match = [ ]
    for m in finditer (self. condition, sentece):
        # m. span () 从头部匹配
        i, j = m. span ()
        matches. extend (sentence [i: j])
    return self. action (matches), self. condition_ num
#规则集合
rules = [
    Rule (condition_ num = 2, condition = disease_ entity + Star (Any ( ), greedy = False) + zhengzhuang_ keyword
```

```
+ Star（Any（），greedy = False），action = QuestionSet. has_ zhengzhuang_ question），
        Rule（condition_ num = 2，condition = disease_ entity + Star（Any（），greedy = False） + bingfazheng_ keyword + Star（Any（），greedy = False），action = QuestionSet. has_ bingfazheng_ question），
        Rule（condition_ num = 2，condition = disease_ entity + Star（Any（），greedy = False） + yufang_ keyword + Star（Any（），greedy = False），action = QuestionSet. has_ yufang_ question），
        Rule（condition_ num = 2，condition = disease_ entity + Star（Any（），greedy = False） + gaishu_ keyword + Star（Any（），greedy = False），action = QuestionSet. has_ gaishu_ question），
        Rule（condition_ num = 2，condition = disease_ entity + Star（Any（），greedy = False） + zhiliao_ keyword，action = QuestionSet. has_ zhiiao_ question），
    Rule（condition_ num = 2，condition = Star（Any（），greedy = False） + yufang_ keyword + disease_ entity，action = QuestionSet. has_ yufang_ question），
        .....
]
for rule in self. rules：
    # print（rule） #
    word_ objects 是一个列表，元素是包含词语和词语对应词性的对象 query，num = rule. apply（word_ objects）
```

最后利用 Pycharm 平台的 Django 项目来进行高血压知识库平台 Web 端界面的开发。利用腾讯云服务器部署 LNMP 环境。将所有项目数据上传，成功后启动 Apache Jena Fuseki 服务，在 Python 项目中启动 manage. py，界面成功运行。

4 结语

本文利用本体技术构建高血压知识图谱，人工智能大数据技术处理自然语言，Python 语言实现基于本体的高血压知识库平台开发。此平台可以辅助医生进行医疗活动，对公众进行高血压知识的普及，减缓就医难和医疗资源紧张等问题。基于本体的高血压知库平台构建为其他慢病（糖尿病等）知识库平台构建提供借鉴。后续的研究将对重庆市某医院的电子病历数据进行采集，进一步获取高血压的相关资料以对高血压本体进行完善。

参考文献

1 刘力生. 中国高血压防治指南 2010［J］. 中华高血压杂志，2011，19（8）：701 - 743.

2 周亚东，刘晓红，张永强，等. 陕西省农村老年人高血压患者知晓率治疗率和控制率的现况调查研究［J］. 中国预防医学杂志，2016，17（3）：170 - 172.

3 中华预防医学会慢性病预防与控制分会. 慢性病的流行形势和防治对策［J］. 中国慢性病预防与控制杂志，2005，13（1）：2 - 3.

4 张宇，汪丰，黄海诚，等. 基于 Web 的高血压非药物治疗知识库构建［J］. 工业控制计算机，2014，27（5）：99 - 100.

5 吴昊，谢红薇. 基于本体和案例推理的高血压诊疗系统的研究［J］. 计算机应用与软件，2013，30（12）：155 - 159，206.

6 巩沐歌，温有奎. 基于本体的高血压疾病诊断知识库［J］. 情报杂志，2010，29（S1）：169 - 172.

7 张巍，张绚，陈俊杰. 基于本体的高血压诊疗系统推理模型研究［J］. 计算机工程与设计，2013，34（11）：4016 - 4020.

8 李博，李科，曾东，等. 基于语义关系的高血压临床指南知识库构建［J］. 中国数字医学，2013，8（9）：64 - 67.

9 黄果，周竹荣. 基于领域本体的概念语义相似度计算研究［J］. 计算机工程与设计，2007，28（10）：2460 - 2463.

10 刘智锋，夏晨曦，黄梨，等. 糖尿病领域本体构建与语义推理实现［J］. 中华医学图书情报杂志，2017，26（9）：7 - 11.

11 朱赟. Python 语言的 Web 开发应用［J］. 电脑知识与技术，2017，13（32）：95 - 96.

12 张卫. 中文词性标注的研究与实现［D］. 南京：南京师范大学，2007.