

# 山西大学论文

编号: 201601001144

## 论文题目 基于知识图谱的疾病知识问答系统的设计与实现

姓 名 张 威

院 系 计算机与信息技术学院

专 业 计算机科学与技术

学 习 年 限 2016 年 9 月至 2020 年 7 月

指 导 教 师 黄 利

学 位 级 别 学 士

2020 年 4 月 10 日

# 基于知识图谱的疾病知识问答系统的设计与实现

## 摘要

随着人工智能时代的到来，知识图谱受到了广泛的关注。传统的基于关键词匹配的搜索引擎技术存在返回结果冗余，查询效率低下等问题。基于知识图谱的问答系统是一种新型信息服务方式，其返回的结果往往更简洁，准确，因而查询更高效。随着计算机算力的不断提升，智能问答系统技术也得到了长足的发展。

本文首先阐述了知识获取、实体链接、关系抽取等知识图谱关键技术理论，随后介绍了现有的智能问答关键技术，并举例分析其优缺点；然后，针对疾病知识领域，探究从网上获取半结构化的数据，并将其转化为知识图谱中的三元组结构。接着，基于上述的三元组结构，阐述了构建特定领域知识图谱的具体方法及流程。最后，基于构建的疾病知识领域知识图谱，结合自然语言处理技术，实现了基于知识图谱的疾病知识问答系统。

**关键词：**知识图谱；疾病领域；命名实体识别；Neo4j；问答系统；

# Design and Implementation of Disease Knowledge Question Answering System Based on Knowledge Graph

## Abstract

With the advent of the era of artificial intelligence, the knowledge graph has received extensive attention. The traditional search engine technology based on keyword matching has the problems of redundant results and low query efficiency. The question answering system based on knowledge graph is a new type of information service method. The returned results are often more concise and accurate, so the query is more efficient. With the continuous improvement of computer computing power, the technology of intelligent question answering system has also been greatly developed.

This article first explains the key technology theories of knowledge graphs such as knowledge acquisition, entity linking, and relationship extraction, and then introduces the existing key technologies of intelligent question answering and analyzes their advantages and disadvantages; then, for the field of disease knowledge, explore the semi-structure obtained from the Internet Data and transform it into a triple structure in the knowledge graph. Then, based on the above-mentioned triple structure, the specific method and process of constructing the knowledge graph of a specific domain are explained. Finally, based on the constructed knowledge graph of the disease knowledge domain, combined with natural language processing technology, a disease knowledge question answering system based on the knowledge graph is realized.

**Keywords:** knowledge graph; disease domain; named entity recognition; Neo4j; question answering system;

# 目录

摘要 .....	I
Abstract .....	II
1 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	1
1.3 本文研究内容和研究目标 .....	2
1.4 论文结构.....	2
2 相关理论与关键技术概述 .....	4
2.1 网络爬虫技术.....	4
2.2 知识图谱技术.....	4
2.2.1 知识获取 .....	4
2.2.2 命名实体识别 .....	4
2.2.3 实体链接 .....	5
2.2.4 关系抽取 .....	5
2.3 智能问答技术.....	6
2.3.1 语义解析方法 .....	6
2.3.2 信息抽取方法 .....	7
2.3.3 向量建模方法 .....	7
2.3.4 深度学习方法 .....	8
3 疾病知识图谱的构建 .....	9
3.1 数据处理 .....	9
3.1.1 目标数据分析 .....	9
3.1.2 数据获取 .....	9
3.1.3 数据预处理 .....	9
3.2 知识图谱构建.....	10

3.2.1 本体构建 .....	11
3.2.2 实体及其关系 .....	11
3.2.3 构建知识图谱 .....	11
3.2.4 知识图谱展示 .....	12
3.2.5 本章小结 .....	14
4 智能问答系统算法设计与实现 .....	15
4.1 系统环境配置 .....	15
4.2 智能问答系统算法设计 .....	15
4.3 系统架构设计 .....	19
4.4 系统分析 .....	21
4.5 系统实现 .....	21
4.6 系统演示 .....	22
4.7 本章小结 .....	28
5 总结与展望 .....	29
5.1 总结 .....	29
5.2 展望 .....	30
参考文献 .....	31
致谢 .....	33

# 1 绪论

## 1.1 研究背景及意义

随着互联网的飞速发展，信息产生与传播速呈几何速度增加，如文字，声音以及视频数据。在这些海量数据中充斥着许多“垃圾”数据。因为传统的搜索引擎大多是基于关键字进行搜索的，这种信息检索方式存在很多问题：（1）返回的是相关文档集，需要用户自己寻找相关问题答案。（2）因为是根据关键词进行匹配，所以返回的答案质量不一，很多答案对用户来说是无意义的。传统的信息检索方式还存在语义理解方面的问题，这种信息检索方式不能很好的理解用户需求，也就不可避免的导致检索效率低下问题。因此，如何才能够提高人们获取信息的效率成为等下人们面临的主要问题。人们希望尽可能的提高信息检索的效率。与此同时，携高效性与便捷性的智能问答系统的出现有效的解决了这个问题。

基于知识图谱的智能问答系统，是一种新型的信息服务方式。不同于现有的搜索引擎，问答系统是以精确的自然语言形式返回答案，而不再是搜索引擎中返回的基于关键词匹配的相似文档排序。华盛顿大学图灵中心主任 Etzioni 教授曾明确指出：“以直接而准确的方式回答用户自然语言提问的自动问答系统将构成下一代搜索引擎的基本形态”<sup>[1]</sup>。因此，智能问答系统技术被认为是未来信息服务领域的颠覆性技术之一。

身体健康是人们最关注的问题，但是人们对于各种疾病的症状、用药、预防措施、治疗方法及费用等信息却知之甚少，传统信息化服务有着信息杂乱多样，获取方式低效等弊端。面对上述问题，能够更好的理解用户意图、更针对地给用户提供疾病健康知识是关键。目前，越来越多的科研人员投入这一领域，研究如何利用人工智能技术，为人类提供更智能的服务。

## 1.2 国内外研究现状

最早知道的问答系统有 BASEBALL<sup>[2]</sup>和 LUNAR<sup>[3]</sup>。BASEBALL 可以回答关于日期、地点、和美国棒球比赛时间等问题。LUNAR 是最早的科学问答系系统。它的设计初衷是支持阿波罗任务对岩石进行地址分析。在对其评测中发现，它能正确回答人类提出 90%的问题。BASEBALL 和 LUNAR 的共同点是都使用了由领域专家手动编写的领域知识库。早期的问答系统限定于特定领域，可扩展性差。同时，需要领域专家对大量非监督数据进行手动收集和标注，耗费大量的时间和精力，因此很难进行较大范围的推广。

Google 公司在 2012 年提出了知识图谱的这一概念<sup>[7]</sup>。近年来，随着知识图谱技术

的不断发展,研究者开始尝试将知识图谱技术应用于问答系统。希望通过信息抽取、实体链接、实体融合等方法,将文本知识转换为计算机易于理解和表示的结构化知识,利用实体及实体间语义关系对文本数据进行更深层次的表示,从而实现对数据背后的信息进行深度挖掘与理解。之后,随着大规模知识库的出现,如 YAGO<sup>[4]</sup>、Freebase<sup>[5]</sup>、DBpedia<sup>[6]</sup>等,进一步推动了基于知识图谱问答的发展。

在 2016 年,美国 Google 公司率先推出的人工智能阿尔法狗战胜世界围棋冠军李世石后,人工智能引起了人们的广泛关注。随着人工智能的发展,智能问答系统技术也迅速发展起来。纵观问答系统的发展过程,近些年,问答系统取得了许多丰硕的成果。苹果公司在 iPhone4s 中就嵌入了其公司的人工智能产品 Siri, Siri 能够通过自然语言 and 用户进行沟通交流。Siri 作为一款智能问答产品,其不仅可以通过语言 and 用户直接对话,还能完成用户的特定需求,例如,给某个人打电话、定个闹钟等功能。除此之外,美国微软公司也在 Windows 电脑中嵌入了其智能问答产品——Cortana 个人助手。同谷歌的 Siri 一样, Cortana 一样可以满足用户这些特定的需求。近年来,国内各大互联网公司也相继推出了自己的智能问答产品,譬如阿里巴巴旗下的天猫精灵,能够和人们通过自然语言进行对话。小米公司也相继推出了小爱智能音箱,能够满足用户语音点歌,讲故事等需求。科大讯飞等公司也相继推出了智能问答个人助手相关产品。随着这些优秀产品的相继推出,人们的生产生活更加便利,人类的双手也进一步得到解放。

随着深度学习在自然语言处理领域的广泛应用,近年来,开始涌现了许多利用深度学习技术进行问答的方法。Dong L 等人<sup>[15]</sup>在 2015 年提出用卷积神经网络(CNN)对上述向量建模方法进行提升, Yih S W 等人<sup>[16]</sup>在 2015 年提出用卷积神经网络对语义解析方法进行提升,该方法获得了当时的最好成绩。随后又出现了使用 LSTM 结合 Attention 机制进行问答的方法<sup>[17]</sup>。深度学习方法所拥有的端到端的优势有着很好的前景,伴随着 AI 技术的发展,知识库问答技术也将得到更好、更广的应用。

### 1.3 本文研究内容和研究目标

本文研究的主要目标是针对限定领域内的疾病知识领域,从医药健康网站上获取相关数据,结合自然语言处理技术构建以疾病为中心的领域知识图谱;之后,使用该图谱,设计并实现一个功能可用,界面美观的疾病知识智能问答系统,从而实现用户输入自然语言问句,系统返回问答结果。

### 1.4 论文结构

随着知识图谱技术的快速发展,问答系统迅速进入人们的视野,借助知识图谱的强

大知识表示与存储能力，以及知识图谱的高效信息检索能力，本文提出的基于知识图谱的疾病知识问答系统的设计与实现由五个部分组成，其组织结构如下：

第一部分 绪论：介绍了本文的研究背景与意义，总结了国内外问答系统的发展状况，同时提出本文的研究内容与目标。

第二部分 相关理论与关键技术：介绍了原始数据获取阶段所使用的爬虫技术。介绍了领域知识库的本体设计与知识库构建，其中包括使用自然语言处理相关方法进行实体识别、意图识别等。介绍了自然语言问句的解析与处理方法，分析了其优缺点以及本文适应方法。

第三部分 疾病领域知识图谱的构建：首先对数据爬取技术进行介绍，接着介绍了知识图谱构建过程的技术细节。然后，对构建的知识图谱进行数据统计分析，如包含实体及关系数。最后，结合 Neo4j 对知识图谱进行可视化。

第四部分 智能问答系统的设计与实现：首先，介绍了系统开发与生产环境。接着给出系统架构，并分析其流程以及问答算法具体实现细节。最后，使用 Flask 开发了一个功能完善，界面友好的问答系统，并对使用方法进行详细演示。

第五部分 总结与展望：先对本文的实现成果做了详细的总结。然后，对基于知识图谱的问答系统的发展趋势做出展望。



## 2 相关理论与关键技术概述

### 2.1 网络爬虫技术

网络爬虫<sup>[18]</sup>是一个自动提取网页的程序，它为搜索引擎从万维网上下载网页，是搜索引擎的重要组成部分。爬虫以某个或者多个网页链接为起点，在爬取网页过程中，不断从爬取到的页面获取新的链接并将其放入队列，这一过程不断重复，直到满足某一条件或者队列为空。爬虫会对爬取到的网页进行存储（如存储到 Redis, MongoDB 等），同时对网页内容进行解析、过滤，保留对用户有用的信息。

### 2.2 知识图谱技术

#### 2.2.1 知识获取

用于构建知识图谱的数据可以分为非结构化数据、半结构化数据、以及结构化数据。对于非结构化数据，首先要提取用户感兴趣的知识，这个过程需要过滤掉文本中广告内容而保存对用户有用的信息。之后使用自然语言处理技术对提取的文本进行实体识别。这里分为两种情况，如果用户拥有自己的知识库，则可以通过实体链接技术把正文中可能的候选实体链接到用户的知识库。否则，用户需要利用命名实体识别技术对文章中的实体进行识别。实体可能存在同义词的情况，此时需要构建同义词表。实体识别需要用到的自然语言处理相关技术在 2.2.2 命名实体识别部分详细阐述。对于半结构化数据的处理上，一般是使用包装器对半结构化数据进行知识抽取。因为半结构化数据有许多重复性的结构，可以利用机器学习方法学习出其中的规则或着手动编写知识抽取规则，进而对半结构数据进行知识获取。

#### 2.2.2 命名实体识别

命名实体识别技术(Named Entity Recognition,NER)技术，通俗的说，就是识别这些实体所指称的边界和类别，最早主要关注人名、地名、组织机构名这三类专有名词识别方法，是自然语言处理中非常基础的一项任务。命名实体识别技术也是信息提取、问答系统、及其翻译等许多自然语言处理任务的重要基础。命名实体识别的准确程度，直接决定下游任务的效果。命名实体识别主要有以下三种方法<sup>[19]</sup>：

（1）基于规则和词典的方法。基于规则的方法大多采用语言学专家构造的规则模板，以模式匹配和字符串匹配为主要方，依赖于知识库和词典的建立。该方法是命名实体识别中使用最早的方法，虽然性能一般较统计方法更优，但是其规则编造耗时巨大，且覆盖面有限，并且可移植性较差。

(2) 基于统计的方法。基于统计的方法主要包括：隐马尔可夫模型、最大熵、支持向量机、条件随机场等。核心思想是通过对语料库语料进行训练，获取其包含的隐藏语言信息，然后对这些语言信息进行统计分析，进而挖掘出语料特征（包括上下文特征、词性特征、停用词特征等）。

(3) 基于深度学习方法。基于神经网络的方法主要是 LSTM+CRF，其主要思想是把词进行分布式表示（word embedding），不再像之前的独热编码(one-hot)方法编码词向量，取而代之的是把词映射成更稠密的 embedding 词向量。然后把这些词向量依次输入到 RNN，依次经过神经网络全连接层和 CRF 层，最后经过 Softmax 来预测每个词的标签。这种方法体现了深度学习方法的端到端的处理思想，不依赖特征工程，但是可解释性差。

### 2.2.3 实体链接

实体链接技术是指将实体链接到相应的知识库中即和知识库中的知识关联起来的一种技术。由于自然语言存在歧义性和多样性的特征，实体消歧是实体链接面临的最主要问题，也是其根本难点。目前的实体消歧方法可以分为基于概率生成模型、基于主题模型、基于图模型、基于深度神经网络的方法<sup>[20]</sup>。

### 2.2.4 关系抽取

实体关系定义为两个或多个实体间的关系，关系抽取是指从文本中检测识别出两个或多个实体间的语义关系。实体间的关系常表示为三元组（Entity1, Relationship, Entity2）。例如“江苏的省会是南京”可以用三元组表示为（江苏，省会，南京）。

关系抽取技术是知识图谱构建过程的关键技术之一，具有重要的理论研究价值。关系抽取是许多其他知识图谱相关技术的基础，具体表现有：（1）自动化构建大规模知识图谱。现有的知识图谱如 WordNet、CYC 等都依赖于人工专家，构建过程费时费力。利用关系抽取技术，可以自动化构建知识图谱，大大节约了人力成本，如 Freebase、DBpedia、Yago。（2）在问答系统方面，利用关系抽取技术可以有效的找出与问题类型相关联的答案类型。（3）关系抽取技术在自然语言理解领域也有着巨大的作用，合理的利用关系抽取技术能够有效的改进许多自然语言处理领域任务的性能，典型的有实体链接。

关系抽取可以根据内容划分为限定域关系抽取和开放域关系抽取。限定域关系抽取中所抽取的关系是确定的，预先定义好的。因此可以，通过有监督学习方法进行规则学习。而开放域关系抽取的关系是事先未定义的、不确定的。因此，其主要研究方法是使用非监督学习方法进行未知关系发现。

与实体链接等知识图谱构建技术一样，关系抽取技术也面临着许多挑战。其中包括

自然语言表达具有多样性。自然语言多样性表现在同一种实体关系可能存在多种不同的表达方式，例如“家乡”可以表示为“A 的家乡是 B”，“B 是 A 的家乡”，“作为 A 的家乡，B.，”等不同的表达方式。但是其本质含义是相同的。实体关系具有隐含性。隐含性体现在文本中没有明确的表现出关系类型。例如，“李嘉诚与英国政府相关人士共同商讨合作事宜”，其潜在含义是李嘉诚希望与英国政府进一步扩大房地产开发市场。但是，这段话中并没有直接给出李嘉诚和其公司的关系，但是从这段话中我们可以推测出其潜在的关系。实体关系具有复杂性，实体关系存在一对多的情况，而且有些实体关系可以并存。比如：江苏和南京的关系有多个，南京坐落于江苏，南京是江苏省的省会。这些关系是同时存在的，可见实体关系具有复杂性。

## 2.3 智能问答技术

随着知识图谱的迅速发展，智能问答技术也得到了长足的发展，传统的知识库问答技术大体上可以分为基于语义解析方法、基于信息抽取方法、基于向量建模方法、深度学习方法。

### 2.3.1 语义解析方法

语义解析方法<sup>[8][9][10]</sup>的主要思想是把自然语言转化为一系列形式化的逻辑形式，通过自底向上地对这些逻辑形式的解析，得到一种可以表达自然语言问题的逻辑形式，进而转换为查询语句（如 Cypher、Sparql 等），然后利用这些语句查询知识库，从而得到问题的答案。Berant J 等人通过该方法构建的知识库问答系统，获得了当时的最好成绩，其主要思路如图 2.1 所示。

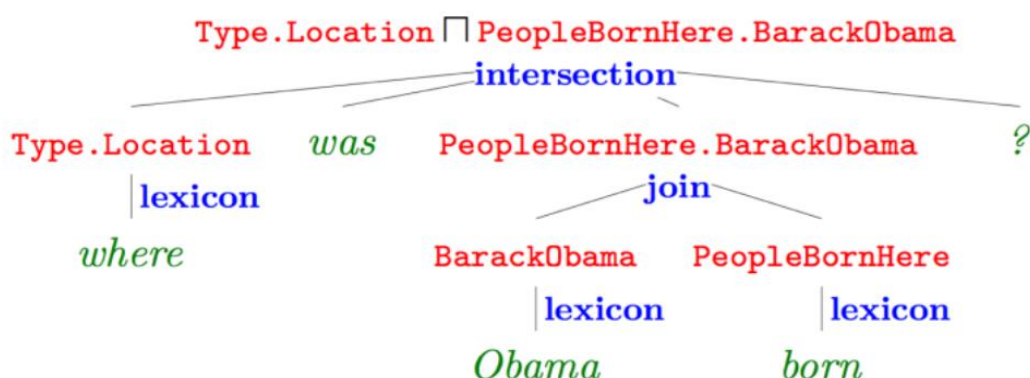


图 2.1 语义解析方法

图 2.1 中，红色部分即逻辑形式，绿色部分表示来自用户的问题，蓝色部分为语义

解析进行的相关操作，而形成的语义解析树的根节点则是最终的语义解析结果，可以通过查询语句直接在知识库中查询并得到最终答案。该方法效果较好，但是非常依赖相关领域专家，并且需要花费巨大的时间和精力。

### 2.3.2 信息抽取方法

信息抽取方法<sup>[11]</sup>，使用自然语言处理方法提取问题中的实体，然后查询知识库获取以该实体为中心的子图，子图中的每一个节点或边都可以作为候选答案，对问题进行建模，获取问题特征，使用机器学习方法训练分类器对特征进行分类，进而得到问题的答案。

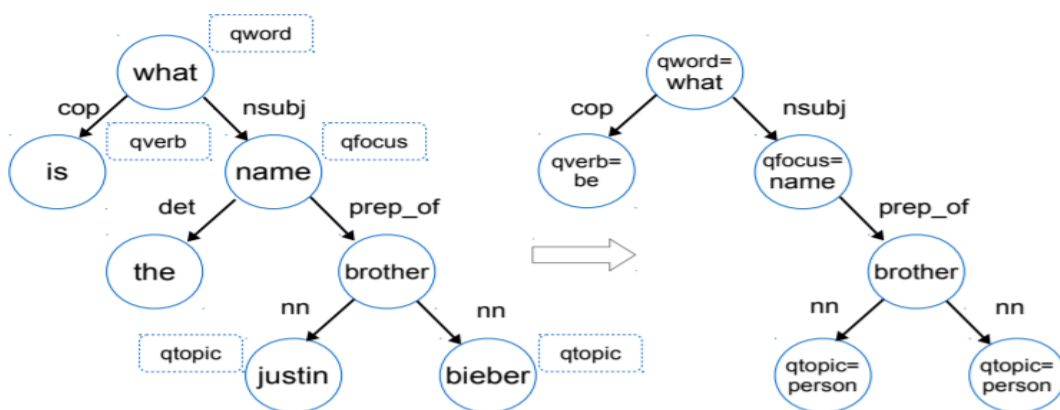


图 2.2 信息抽取方法

图 2.2 是 Yao X 等人在 2014 年提出的通过信息抽取方法。其思想是先提取问题的主要特征词及其依赖关系（左），然后转换为特征图（右），特征图删除了不重要信息，只保留原始问题相关信息，本质上是一个信息抽取的过程。该方法相比语义解析方法，减少了对人工定义规则的依赖，但是能否对问题构建良好的特征决定了最终问答系统的好坏。

### 2.3.3 向量建模方法

向量建模方法<sup>[12][13][14]</sup>主体思想是把用户问题和答案全部向量化，通过训练数据进行训练，使得问题与答案的向量尽可能接近（通常是以向量点乘的形式）。使用模型对问题和候选答案的得分进行筛选，分数最高的答案为最终答案。

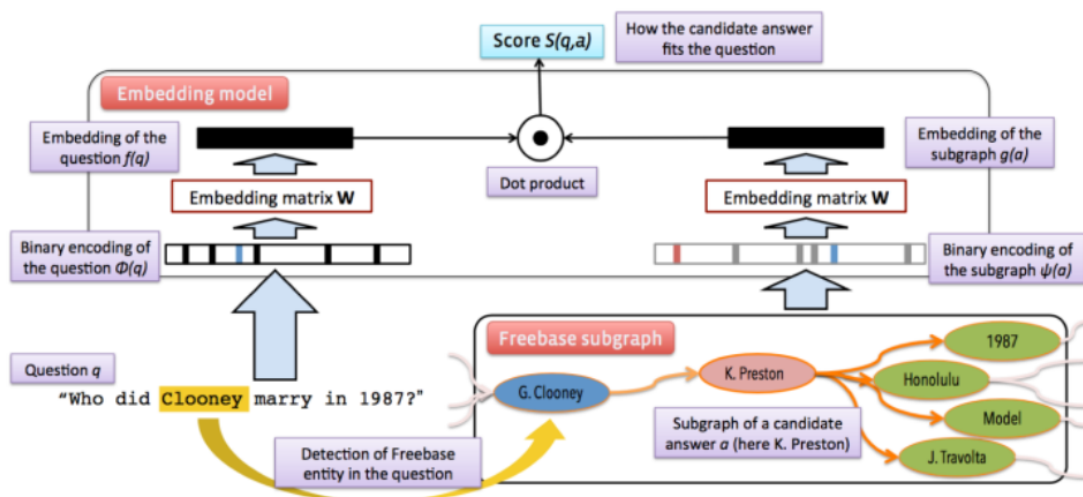


图 2.3 向量建模方法

图 2.3 展示了 Bordes A 等人在 2014 年提出的向量建模法用于问答系统的模型框架，该方法不但免去了人工构建规则和特征，并且从应用角度来说更加易于实现。

#### 2.3.4 深度学习方法

随着计算机硬件的快速发展，计算机算力得到了大幅提升。使用深度学习进行问答逐渐成为了问答系统的主流方法。其中包括卷积神经网络 (CNN)<sup>[21]</sup>、循环神经网络 (RNN)、长短期记忆神经网络 (LSTM)，以及注意力机制 (Attention) 与 RNN 相结合的方法<sup>[22]</sup>。这些方法有着传统方法所不具有的优点，如不需要人工构建特征，可以实现端到端的系统等。可见，深度学习方法有着很好的前景。

## 3 疾病知识图谱的构建

知识图谱是智能问答系统的核心，本章将详细介绍如何构建一个以疾病为中心的知识图谱。

### 3.1 数据处理

#### 3.1.1 目标数据分析

本文的数据源自医疗健康网站——寻医问药网，考虑到人们大多以疾病作为检索关键字进行检索的习惯，本文将以疾病作为整个知识图谱的中心节点。

疾病按照其所属类别可以分为内科、外科、妇产科、男科、皮肤科、肿瘤科、五官科、儿科、传染科、精神科以及其它科室。又可以根据疾病常见程度、患病年龄、性别等分类为常见病、职业病、男性病、女性病、儿童病等。

对于具体的某种疾病，可以从网站上获取疾病简介、疾病病因、预防措施、并发症、诊断方法、治疗方案以及护理方法和饮食保健等疾病信息。这些信息是整个疾病知识问答系统的基础。

#### 3.1.2 数据获取

考虑到网站内容是一种半结构化的信息，我们可以利用已有的网页解析技术对目标网页进行解析，从而获取相关知识。

Python 是一种面向对象、解释型、动态数据类型的高级语言。其中有许多与网页解析相关的函数库，如 requests、urllib2、spider 等。本文采用 spider 作为网页爬取的主要技术。

爬取的主页面如图 3.2，对爬取的 Html 页面，使用 xpath 以及 css 等元素选择器提取目标文本内容，如 `selector.xpath('//div[@class="jib-article-con jib-lh-article"]//p/text()')[0].strip()`。

#### 3.1.3 数据预处理

对爬取的文本内容进行空值检查，以前去重处理。为了方便后续数据的读取等操作，本文以疾病为单位把疾病相关知识以 json 形式存入 MongoDB 数据库。其中爬取的数据如图 3.1 所示。

```
{ "gaishu_info": { "name": "百日咳", "desc": "百日咳(pertussis, whooping cough)是由百日咳杆菌所致的急性呼吸道传染病。其特征为阵发性痉挛性咳嗽, "name": "苯中毒", "desc": "苯(benzene)是从煤焦油馏分及石油裂解所得的一种芳香烃化合物, 系无色有芳香气味的油状液体。挥发, "name": "喘息样支气管炎", "desc": "喘息样支气管炎(asthmatooid bronchitis)又称哮喘性支气管炎, 泛指一组喘息表现的婴幼儿, "name": "成人呼吸窘迫综合征", "desc": "成人呼吸窘迫综合征简称ARDS, 是一种继发的, 以急性呼吸窘迫和低氧血症为特征的综合, "name": "大量羊水吸入", "desc": "胎儿在宫内或分娩过程中吸入较大量羊水称大量羊水吸入(massive amniotic fluid aspirati, "name": "单纯性肺嗜酸粒细胞浸润症", "desc": "单纯性肺嗜酸粒细胞浸润症, 又名吕弗琉综合征, 是吕弗琉于1932年首先描述本病, "name": "大叶性肺炎", "desc": "大叶性肺炎(lobar pneumonia), 又名肺炎球菌肺炎, 是由肺炎双球菌等细菌感染引起的呈大叶性, "name": "大疱病综合征", "desc": "大疱病综合征有多种表现, 均因接触各种有害物质所致。好发于办公室内工作的人群, 或人员密, "name": "二硫化碳中毒", "desc": "二硫化碳(carbon disulfide, CS2)是工业上应用广泛的化学溶剂, 也用于粘胶纤维、四氯化碳, "name": "肺-胸膜阿米巴病", "desc": "肺-胸膜阿米巴病是溶组织阿米巴原虫感染所致的肺及胸膜化脓性炎症, 肝原性病变多发生在, "name": "肺出血-肾炎综合征", "desc": "肺出血-肾炎综合征, 又称抗基底膜性肾小球肾炎, Goodpasture综合征或Goodpasture病, "name": "肺放线菌病", "desc": "肺放线菌病(pulmonary actinomycosis)是由厌氧的以色列放线菌感染肺部引起的慢性化脓性肉
```

图 3.1 爬取的疾病数据

疾病介绍

疾病常识

病因

预防

并发症

诊断方法

症状

检查

诊断鉴别

治疗方案

治疗

护理

饮食保健

本词条由 **衢州市人民医院 耳鼻咽喉科 张宏伟**（副主任医师、讲师）提供内容并参与编辑

感冒简介

感冒, 总体上分为普通感冒和流行性感冒, 在这里先讨论普通感冒。普通感冒, 祖国医学称“伤风”, 是由多种病毒引起的一种呼吸道常见病, 其中30%-50%是由某种血清型的鼻病毒引起, 普...[详情>](#)

常识

易感人群: 无特定人群

患病比例: 0.6%

传染方式: 呼吸道传播

常用检查: [白细胞计数 \(WBC\)](#) [更多>](#)

症状表现: [头痛](#) [发烧](#) [更多>](#)

并发疾病: [鼻窦炎](#) [鼻炎](#) [更多>](#)

治疗

就诊科室: 内科 呼吸内科

治疗方式: [对症治疗](#) [中医治疗](#)... [更多>](#)

治疗周期: 7-14天

治愈率: 97%

常用药品: [盐酸氨溴索分散片](#) [喷托维林氯化铵糖浆](#)

治疗费用: 根据不同医院, 收费标准不一致, 市三甲医院约(500-1000元)

温馨提示: 重在预防, 加强锻炼、增强体质、生活饮食规律、改善营养。避免受凉和过度劳累, 有助于降低易感性, 是预防感冒最好的办法。

图 3.2 疾病信息主页

3.2 知识图谱构建

知识图谱的构建技术主要有自顶向下和自底向上两种。自顶向下构建指的是借助网站等结构化数据源，从高质量的数据中提取出本体及其模式信息，加入知识库，从而实现知识库的构建。而自底向上的构建方式则是从公开的数据集中提取出其中的资源模式，选中置信度较高的信息加入知识库，实现知识库的构建。本文的数据源是来自医疗健康网站的半结构化数据，因此本文采用自顶向下的知识库构建方式。

10

### 3.2.1 本体构建

本体根据其应用主题可以分为顶层本体、领域本体、任务本体、应用本体。本文研究的本体属于领域本体，即研究特定领域内概念及概念之间的关系。本体构建可按照确定领域本体范畴、复用现有本体、列出领域内的术语、定义类和类的等级关系、定义类的属性、填充实例的思路进行构建。

### 3.2.2 实体及其关系

本文的实体类型共 6 种，包括疾病 (Disease)、部门 (Department)、检查项目 (Check)、治疗药品 (Drug)、食物 (Food)、疾病症状 (Symptom)。

实体间的关系共 8 种，包括<Disease, belongs\_to, Department>疾病所属科室、<Disease, inspection\_item, Check>疾病检查项目、<Disease, common\_drug, Drug>疾病常用药物、<Disease, has\_symptom, Symptom>疾病症状、<Disease, good\_food, Food>疾病宜吃食物、<Disease, avoid\_food, Food> 疾病忌吃食物、<Disease, recommend\_recipes, Food>推荐食谱、<Disease, has\_complication, Symptom>疾病并发症。

### 3.2.3 构建知识图谱

本文使用 Neo4j 图数据库存储数据，将数据处理部分的数据以三元组的形式插入数据库中，进而完成知识图谱的构建。

具体而言，使用 Python 中的函数库 py2neo 可以方便的创建节点 (Node) 及关系 (Relationship)。在构建知识图谱过程中，把 “Disease”，” Department”，” Check”，” Drug”，” Food”，” Symptom” 作为 Node；把 ” belongs\_to”，” inspection\_item”，” common\_durg”，” has\_symptom”，” good\_food”，” avoid\_food”，” recommend\_recipes”，” has\_complication” 作为实体间关系，从而完成疾病知识图谱的构建。

构建过程的核心代码如下图 3.3：



```

def create_graph_nodes(self):
    Diseases, Symptoms, Departments, \
    Checks, Drugs, Foods, \
    diseases_info, \
    rels_depart_depart, rels_disease_eat, rels_disease_no_eat, rels_disease_recipes, rels_disease_drug, \
    rels_disease_check, rels_disease_symptom, rels_disease_complication, rels_disease_department = self.read_nodes()
    self.create_disease_nodes(diseases_info)
    self.create_general_nodes("Symptom", Symptoms)
    self.create_general_nodes("Department", Departments)
    self.create_general_nodes("Check", Checks)
    self.create_general_nodes("Drug", Drugs)
    self.create_general_nodes("Food", Foods)
def create_graph_rels(self):
    Diseases, Symptoms, Departments, \
    Checks, Drugs, Foods, \
    diseases_info, \
    rels_depart_depart, rels_disease_eat, rels_disease_no_eat, rels_disease_recipes, rels_disease_drug, \
    rels_disease_check, rels_disease_symptom, rels_disease_complication, rels_disease_department = self.read_nodes()
    self.create_relationship("Department", "Department", rels_depart_depart, "belongs_to", "属于")
    self.create_relationship("Disease", "Food", rels_disease_eat, "good_food", "宜吃食物")
    self.create_relationship("Disease", "Food", rels_disease_no_eat, "avoid_food", "忌吃食物")
    self.create_relationship("Disease", "Food", rels_disease_recipes, "recommand_recipes", "推荐食谱")
    self.create_relationship("Disease", "Drug", rels_disease_drug, "common_drug", "常用药物")
    self.create_relationship("Disease", "Check", rels_disease_check, "check_item", "检查项目")
    self.create_relationship("Disease", "Symptom", rels_disease_symptom, "has_symptom", "疾病症状")
    self.create_relationship("Disease", "Disease", rels_disease_complication, "has_complication", "并发症")
    self.create_relationship("Disease", "Department", rels_disease_department, "belongs_to", "所属科室")

```

图 3.3 构建知识图谱核心代码

### 3.2.4 知识图谱展示

在 Windows 系统上打开 Neo4j 桌面版，启动 Neo4j 数据库，进入目标数据库，点击 Open Browser 即可看到构建的知识图谱可视化结果。下面图 7 是以肝炎为例的简单展示，浅绿色节点表示疾病节点，深绿色节点表示疾病症状，蓝色节点表示疾病相关检查，黄色节点表示疾病所属类别，粉色节点表示疾病药物信息（部分实体类型未在图中显示）。实体节点之间的有向边表示实体间的关系。

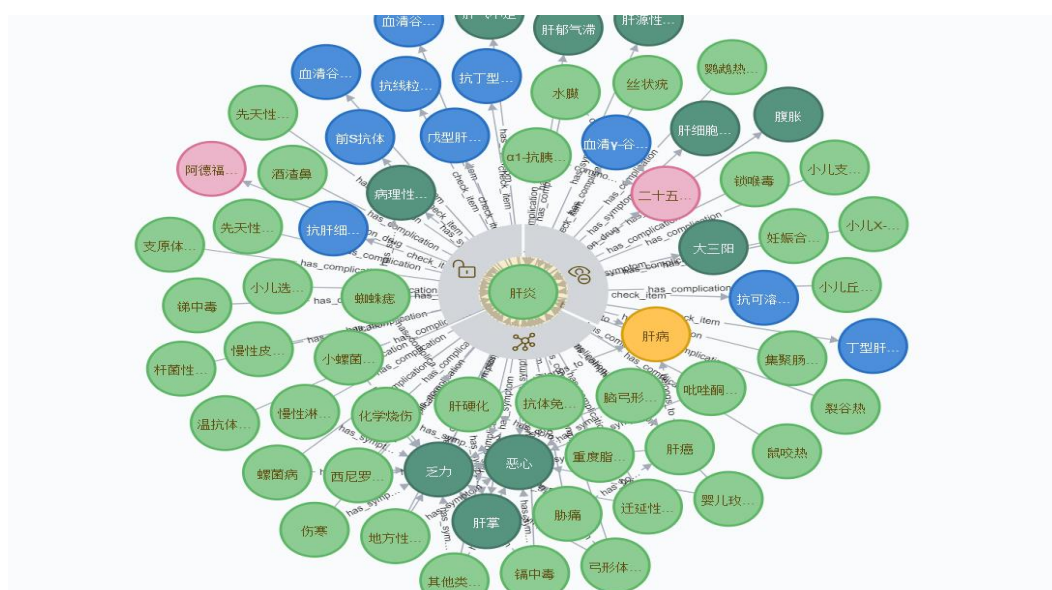


图 3.4 知识图谱可视化

构建的知识图谱的实体及其关系的统计信息分别如下表 3.1、表 3.2，疾病中心节点包含的属性如表 3.3。由图可知，本文构建的知识图谱规模较大，为后续问答系统提供了数据支撑。

表 3.1 疾病知识图谱实体统计信息

实体类型	中文含义	实体数量	举例
Disease	疾病	8792	感冒
Department	科室	54	内科
Check	检查项目	3342	血常规
Drug	治疗药品	1204	肠炎宁
Food	食物	4854	蜂蜜
Symptom	症状	6556	腹腔出血
Total	总计	24802	约 2.5 万实体

表 3.2 疾病知识图谱关系统计信息

实体关系类型	中文含义	关系数量	举例
belongs_to	属于	8784	〈哮喘,belongs_to,内科〉
common_drug	常用药物	13477	〈小儿肺炎,common_drug,小儿肺热平胶囊〉
good_food	宜吃食物	34221	〈胸椎骨折,good_food,黑鱼〉
avoid_food	忌吃食物	34215	〈感冒,avoid_food,猪油〉
check_item	检查项目	39098	〈肾结石,check_item,尿液颜色〉
recommend_recipes	推荐食谱	39663	〈肝病,recommend_recipes,小米粥〉
has_complication	并发症	19151	〈痔疮,has_complication,直肠癌〉
has_symptom	疾病症状	58398	〈冠心病,has_symptom,心慌〉
Total	总计	247007	近 25 万实体关系

表 3.3 疾病中心节点属性信息

属性类型	中文含义	举例
name	疾病名称	感冒
desc	疾病描述	发热伴寒战；咽痛；流鼻涕
cause	疾病病因	当有受凉，淋雨，过度疲劳
prevent	预防措施	补充维生素 E、维生素 C
treat_cycle	治疗周期	7-14 天

treat_way	治疗方式	中医治疗;支持性治疗
cure_prob	治愈概率	97%
susceptible_people	易感人群	无特定人群
medical_insurance	是否医保疾病	未知
transmission_way	传染方式	呼吸道传播
treat_cost	治疗费用	约 500-1000 元
nursing	护理方法	日常护理

### 3. 2. 5 本章小结

本文按照知识图谱的构建顺序,从数据获取到本体设计方法,再到构建知识库构建,比较详细的阐述了如何获取半结构化数据,如何进行本体设计以及如何构建知识图谱。结合 Neo4j 数据库的可视化展示以及对实体及关系的统计信息,完整地展现知识库构建的整个流程。

## 4 智能问答系统算法设计与实现

基于上述构建的以疾病为中心的知识图谱，结合问句模板匹配的方法，实现智能问答系统的构建。核心在于对问句进行语义解析与意图识别。

### 4.1 系统环境配置

本文使用目前非常受欢迎的 Python 网页开发轻量级框架——Flask, 使用 Bootstrap 进行前端页面美化，实现了功能完善，界面美观的基于知识图谱的疾病知识问答系统。生产环境与开发环境分别如表 4.1、表 4.2。

表 4.1 开发环境

操作系统	Windows 10
CPU	Intel Core i5 2.4GHz
内存	20G
数据库	Neo4j
开发语言	Python 3.6.4

表 4.2 生产环境

操作系统	Ubuntu 18.04
CPU	Intel Core i7 2.5GHz
内存	40G
数据库	Neo4j
开发语言	Python3.6.4

### 4.2 智能问答系统算法设计

本文的问答算法核心点包括实体识别、意图识别、查询语句转换。

- (1) 实体识别。首先，根据爬取的获取的实体构建领域关键词词典，领域词词典由 Disease、Department、Check、Drug、Symptom、Food 这六类实体类型所包含的实体构成。然后，基于该词典构建基于 Trie 树的关键词快速查询树(核心代码如图 4.1)，用 Trie 树匹配输入的自然语言问句，找出潜在的关键词作为候选实体从而实现实体

识别（核心代码如图 4.2）。

```
def build_actree(self, word_list):  
    actree = ahocorasick.Automaton()  
    for ind, word in enumerate(word_list):  
        actree.add_word(word, (ind, word))  
    actree.make_automaton()  
    return actree
```

图 4.1 构建 AC Tree

```
def get_keyword_from_question(self, question):  
    region_words = []  
    for item in self.region_tree.iter(question):  
        keyword = item[1][1]  
        region_words.append(keyword)  
  
    region_words = list(set(region_words)) # 去重  
  
    stop_words = []  
    for wd1 in region_words:  
        for wd2 in region_words:  
            if wd1 in wd2 and wd1 != wd2:  
                stop_words.append(wd1)  
  
    final_words = [word for word in region_words if word not in stop_words]  
    final_word_types = {word: self.word_type_dict.get(word) for word in final_words}  
    return final_word_types
```

图 4.2 获取潜在实体类型

（2）意图识别。采用基于查询词模板匹配的方法对查询意图进行分类，从而实现意图识别。首先，手动构建问句疑问词词库（如图 4.3），然后把识别出的潜在实体与疑问词词库进行匹配，获得问句类型（部分问句匹配代码如图 4.4）。

```

# 问句疑问词
self.symptom_qwds = ['症状', '表征', '现象', '症候', '表现', '什么样']
self.cause_qwds = ['原因', '成因', '为什么', '怎么会', '怎样才', '咋样才', '怎么会', '如何会', '为啥', '为何',
                  '会造成', '怎么回事', '总是', '老是']
self.complication_qwds = ['并发症', '并发', '一起发生', '一并发生', '一起出现', '一并出现', '一同发生', '一
self.food_qwds = ['饮食', '饮用', '吃', '食', '伙食', '膳食', '喝', '菜', '忌口', '补品', '保健品', '食谱',
self.drug_qwds = ['药', '药品', '用药', '胶囊', '口服液', '炎片', '什么药']
self.prevent_qwds = ['预防', '防范', '抵制', '抵御', '防止', '躲避', '逃避', '避开', '免得', '逃开', '避开',
                    '怎样才能不', '怎么才能不', '咋样才能不', '咋才能不', '如何才能不',
                    '怎样才不', '怎么才不', '咋样才不', '咋才不', '如何才不',
                    '怎样才可以不', '怎么才可以不', '咋样才可以不', '咋才可以不', '如何可以不',
                    '怎样才可不', '怎么才可不', '咋样才可不', '咋才可不', '如何可不']
self.treat_cycle_qwds = ['周期', '多久', '多长时间', '多少时间', '几天', '几年', '多少天', '多少小时', '几个
self.treat_way_qwds = ['怎么治疗', '如何医治', '怎么医治', '怎么治', '怎么医', '如何治', '医治方式', '疗法']
self.cure_prob_qwds = ['多大率能治好', '多大几率能治好', '治好希望大么', '几率', '几成', '比例', '可能性']
self.susceptible_qwds = ['什么样的', '易感人群', '容易感染', '易发人群', '什么人', '哪些人', '感染', '染上',
self.check_qwds = ['检查', '检查项目', '查出', '检查', '测出', '试出', '怎么查', '查什么']
self.belong_qwds = ['属于什么科', '属于', '什么科', '科室', '挂号', '挂什么', '挂什么科室']
self.effect_qwds = ['治疗什么', '治啥', '治疗啥', '医治啥', '治愈啥', '主治啥', '主治什么', '有什么用', '有
                    '有什么好处', '有什么益处', '有何益处', '用来', '用来做啥', '用来作甚', '需要', '要', '能干
self.treat_cost_qwds = ['多少钱', '开销', '花费', '治疗费用', '多少钱', '费用']
self.medical_insurance_qwds = ['报销', '医保', '报销比例', '合作医疗']
self.transmission_way_qwds = ['怎么传播', '传播', '传染']
self.nursing_qwds = ['护理', '如何护理', '护理方式', '护理方法', '保养']

```

图 4.3 构建疑问词词库

```

# 已知疾病判断症状
if self.check_qwds_type(self.symptom_qwds,question) and ('Disease' in types):
    question_type = 'disease_symptom'
    question_types.append(question_type)

# 已知症状判断疾病
if self.check_qwds_type(self.symptom_qwds,question) and ('Symptom' in types):
    question_type = 'symptom_disease'
    question_types.append(question_type)

# 疾病原因
if self.check_qwds_type(self.cause_qwds,question) and ('Disease' in types):
    question_type = 'disease_cause'
    question_types.append(question_type)

# 并发症
if self.check_qwds_type(self.complication_qwds,question) and ('Disease' in types):
    question_type = 'disease_complication'
    question_types.append(question_type)

# 疾病常用药品
if self.check_qwds_type(self.drug_qwds,question) and ('Disease' in types):
    question_type = 'disease_drug'
    question_types.append(question_type)

```

图 4.4 获取问句类型核心代码

(3) 查询语句转换。基于实体识别与意图识别得到的问句类型，根据自定义的问题类型转换为 Cypher 语句规则，把自然语言问句转换为相应的 Cypher 查询语句（转换部分核心代码如图 4.5）。

```

def sql_transfer(self, question_type, entities):
    if not entities:
        return []

    sql = []
    #TODO 1 已知疾病查询症状 disease_symptom
    if question_type == 'disease_symptom':
        sql = ["MATCH (m:Disease)-[r:has_symptom]->(n:Symptom) where m.name = '{0}' " \
            "return m.name, r.name, n.name".format(i) for i in entities]

    #TODO 2 已知症状判断疾病 symptom_disease
    elif question_type == 'symptom_disease':
        sql = ["MATCH (m:Disease)-[r:has_symptom]->(n:Symptom) where n.name = '{0}' " \
            "return m.name, r.name, n.name".format(i) for i in entities]

    #TODO 3 疾病原因 disease_cause
    elif question_type == 'disease_cause':
        sql = ["MATCH (m:Disease) where m.name = '{0}' " \
            "return m.name, m.cause".format(i) for i in entities]

    #TODO 4 并发症 disease_complication
    elif question_type == 'disease_complication':
        sql = ["MATCH (m:Disease)-[r:has_complication]->(n:Disease) where m.name = '{0}' " \
            "return m.name, r.name, n.name".format(i) for i in entities]
    # sql = f

```

图 4.5 问句转换为 Cypher 语句

### 4.3 系统架构设计

系统的整体架构如图 4.6。

首先，对用户输入的自然语言问句进行实体识别与意图识别。其次，根据上步处理结果对问句进行分类。然后对问句进行解析并转换为 Cypher 语句。最后，使用 Cypher 语句查询 Neo4j 数据库，返回查询结果。



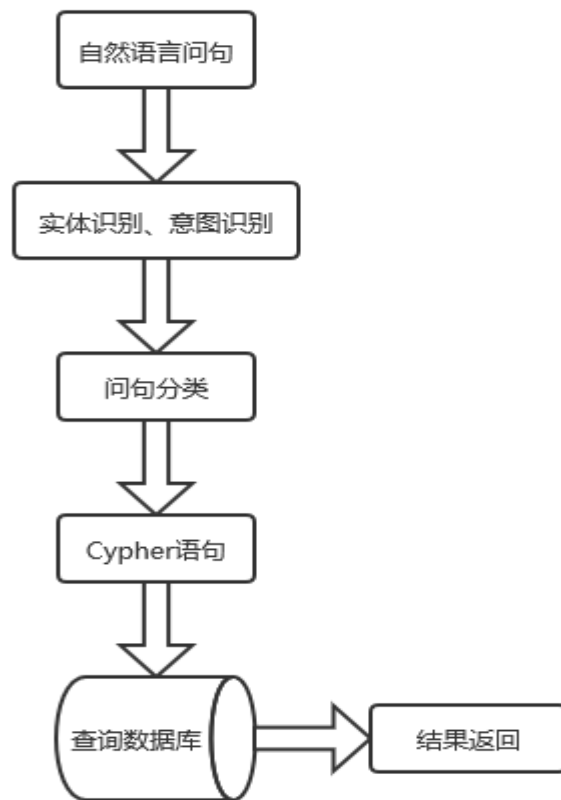


图 4.6 问答系统架构图

问答系统用例图如图 4.7。

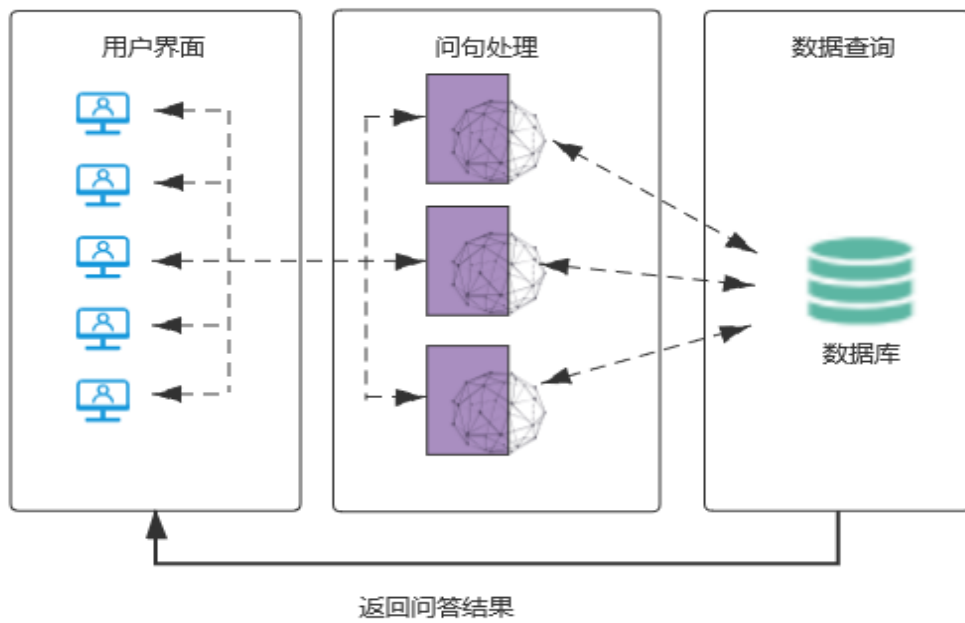


图 4.7 问答系统用例图

## 4.4 系统分析

系统共包含 21 种问题模板类型，能够很好的满足人们对于疾病知识查询的需求。系统支持的问答具体类型如表 4.3。

表 4.3 系统支持的问答类型

Question Type	问题类型	举例
disease_symptom	已知疾病查看症状	小儿肺炎有什么症状
symptom_disease	已知症状查看可能的疾病	最近老是流鼻涕怎么办
disease_cause	疾病原因	总是失眠是什么原因
disease_complication	疾病并发症	感冒有哪些并发症
disease_drug	疾病常用药物	白内障一般吃什么药
drug_disease	药物能治疗啥疾病	阿莫西林胶囊能治疗啥
disease_avoid_food	疾病忌口	肝病不能吃什么
disease_good_food	疾病宜吃	肺结核吃什么好
food_avoid_disease	什么疾病不能吃的食物	什么人最好不要吃蜂蜜
food_good_disease	食物适合哪些人吃	腰果适合哪些人吃
disease_check	疾病检查项目	怎么查出来是不是脑膜炎
check_disease	已知检查找疾病	血常规能查出来啥病
disease_prevent	疾病预防方法	怎么样才能防止肾虚
disease_treat_way	疾病治疗方法	高血压要怎么治
disease_cure_prob	疾病治愈概率	肺结核能治好吗
disease_susceptible_people	疾病易感人群	什么人容易得高血压
disease_department	疾病去哪个科室	痔疮属于哪个科室的
disease_treat_cost	疾病治疗费用	治疗肾结石要多少钱
disease_medical_insurance	某病是医保疾病吗	肾结石是医保疾病吗
disease_treat_cycle	某疾病的治疗周期	感冒要多久才能好
disease_desc	疾病概述	抑郁症

## 4.5 系统实现

本文基于 Python 的 Flask 微框架，结合 Bootstrap、jQuery、AJAX 进行系统实现。Flask 框架有如下优点：

(1) 简单：Flask 的路由以及路由函数由修饰器设定，开发人员不需要借助其他文件匹配；

(2)配置灵活:有多种方法配置,不同环境的配置也非常方便;环境部署简单,Flask运行不需要借助其他任何软件。

(3)扩展库多:Flask有大量的第三方扩展插件。

(4)低耦合,Flask可以兼容多种数据库、模板。

系统采用客户端与服务端分离的方式,方便后期对项目进行维护。客户端是普通用户与系统进行交互的窗口,服务端对客户端提交的问题进行解析、查询,然后把查询结果返回给客户端。

## 4.6 系统演示

图 4.8 是基于知识图谱的疾病知识问答系统客户端的主界面。用户通过输入框输入问题。用户提交问题后,系统通过 AJAX 跨域请求从服务端获得该问题的查询结果并在查询结果框内进行展示,同时给出此次查询的响应时间。

由于领域问答系统能回答的问题是有限的,当系统无法回答用户的问题时,将返回如图 4.9 的界面。当网络出现故障时,将返回 4.10 的界面。

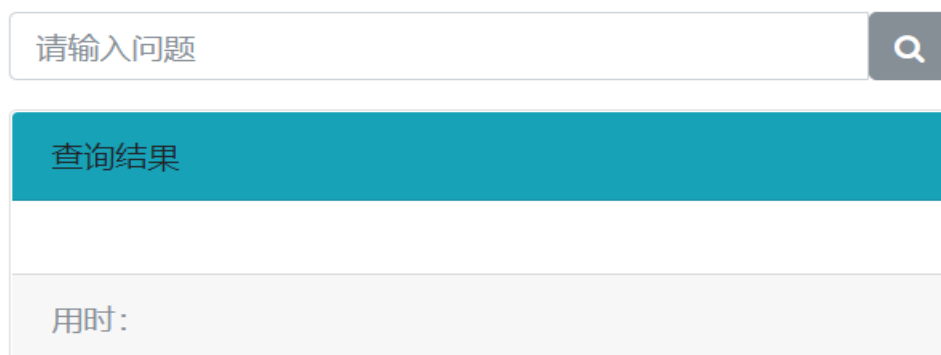


图 4.8 问答系统客户端主界面

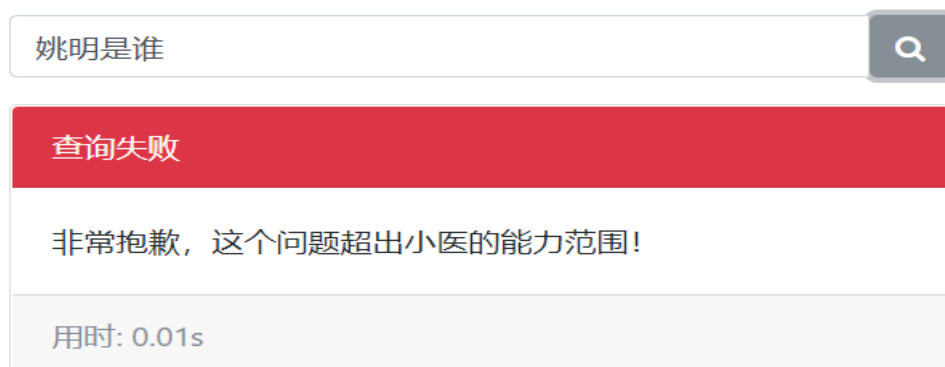


图 4.9 系统无法回答界面



图 4.10 网络故障界面

本文在系统分析部分介绍了该系统能够回答的问题类型，其中共有 21 种问题类型，由于文章篇幅有限，下面将对部分问题类型进行演示。

- (1) disease\_symptom (根据疾病查询疾病相关症状), 如图 4.11, 用户输入小儿肺炎有什么症状。问答系统返回小儿肺炎的症状描述信息。

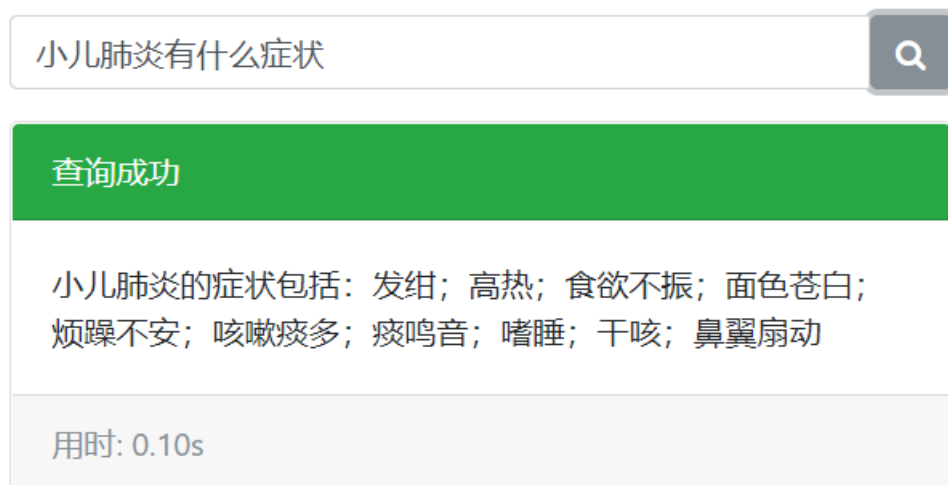


图 4.11 根据疾病查询症状演示图

- (2) symptom\_disease (根据症状推测疾病), 如图 4.12, 用户输入最近老是流鼻涕怎么办, 系统通过分析, 得出其问句类型为已知疾病症状推测可能患上的疾病信息, 然后系统给出可能的疾病类型。

最近老是流鼻涕怎么办

Q

查询成功

症状流鼻涕可能染上的疾病有：感冒；小儿流行性感  
冒；慢性额窦炎；鼻源性头痛；下呼吸道感染；副流行性感  
冒；小儿感冒；蝶窦炎；干酪性鼻窦炎；小儿急性上呼吸道感染；  
硫化氢中毒；风寒感冒；筛窦炎；风热犯肺；H7n7；麻疹；  
禽流感；鼻炎；原发性鼻腔淋巴瘤；鼻病

用时: 0.06s

图 4.12 根据症状推测疾病演示图

(3) disease\_cause (查询疾病病因)，如图 4.13，用户输入总是感冒是什么原因，系统分析出其类型为查询疾病病因，然后给出感冒的成因。

总是感冒是什么原因

Q

查询成功

感冒可能的成因有：感冒有70%-80%由病毒引起，主要有流  
感病毒(甲，乙，丙)，副流感病毒，呼吸道合胞病毒，腺病  
毒，鼻病毒，埃可病毒，柯萨奇病毒，麻疹病毒，风疹病  
毒。细菌感染可直接或继病毒感染之后发生，以溶血性链球  
菌为多见，其次为流感嗜血杆菌，肺炎球菌和葡萄球菌等，  
偶见革兰阴性杆菌，其感染的主要表现为鼻炎，咽喉炎或扁  
桃腺炎 全身酸痛等。当有受凉，淋雨，过度疲劳等诱发因  
素，使全身或呼吸道局部防御功能降低时，原已存在于上呼  
吸道或从外界侵入的病毒或细菌可迅速繁殖，引起发病，尤  
其是老幼体弱或有慢性呼吸道疾病如鼻旁窦炎，扁桃体炎  
者，更易罹病。鼻腔及咽粘膜充血，水肿，上皮细胞破坏，  
少量单核细胞浸润，有浆液性及粘液性炎性渗出，继发细菌  
感染后，有中性粒细胞浸润，大量脓性分泌物。

用时: 0.04s

图 4.13 查询疾病原因演示图

(4) disease\_complication (查询疾病并发症), 如图 4.14, 用户输入肺炎有哪些并发症, 系统分析其类型为查询疾病并发症, 然后给出具体查询结果。




查询成功

肺炎的并发症包括: 支气管肺炎; 肺脓肿

用时: 0.06s

图 4.14 查询疾病并发症演示图

(5) disease\_drug (疾病常用药物), 如图 4.15, 用户输入白内障一般吃什么药, 系统分析问句, 得出其类型为白内障, 然后给出问句查询结果。



查询成功

白内障通常的使用的药品包括: 苄达赖氨酸滴眼液; 调元大补二十五味汤散

用时: 0.06s

图 4.15 查询疾病常用药物演示图

(6) disease\_avoid\_food (疾病忌口), 如图 4.16, 用户输入肝病不能吃什么, 系统分析其问句类型为查询疾病忌口, 并给出查询结果。

肝病不能吃什么

Q

查询成功

肝病忌食的食物包括有：松子仁；白酒；鸭肉；鸭蛋

用时: 0.05s

图 4.16 查询疾病忌口演示图

(7) disease\_good\_food (疾病宜吃食物)，如图 4.17，用户输入肺结核吃什么好，系统分析其类型为疾病宜吃食物，并给出问答结果。

肺结核吃什么好

Q

查询成功

肺结核宜食的食物包括有：推荐食谱包括有：白果炒百合;豌豆绿豆粥;三丝发菜;炒苋菜;银耳炖木瓜;白扁豆粥;素炒小白菜;木耳拌豆芽

用时: 0.10s

如图 4.17 查询疾病宜吃食物演示图

(8)disease\_department(疾病所属科室)，如图 4.18，用户输入痔疮属于哪个科室的，系统分析其问句类型为查询疾病所属科室，然后给出问答结果。

痔疮属于哪个科室的

Q

查询成功

痔疮所属科室为： 外科

用时: 0.05s

如图 4.18 查询疾病所属科室演示图

(9) disease\_treat\_cost (疾病治疗费用), 如图 4.19, 系统, 用户输入治疗肾结石要多少钱, 系统分析其问题类型为疾病治疗费用, 并给出问答结果。

治疗肾结石要多少钱

Q

查询成功

肾结石的治疗费用相关信息: 根据不同医院, 结石大小不同, 收费标准不一致, 市三甲医院约 (1000 - 6000元)

用时: 0.04s

图 4.19 查询疾病治疗费用演示图

(10) disease\_desc (疾病描述), 如图 4.20, 用户输入高血压, 系统分析其问题类型为查询疾病描述, 并给出高血压的描述信息。



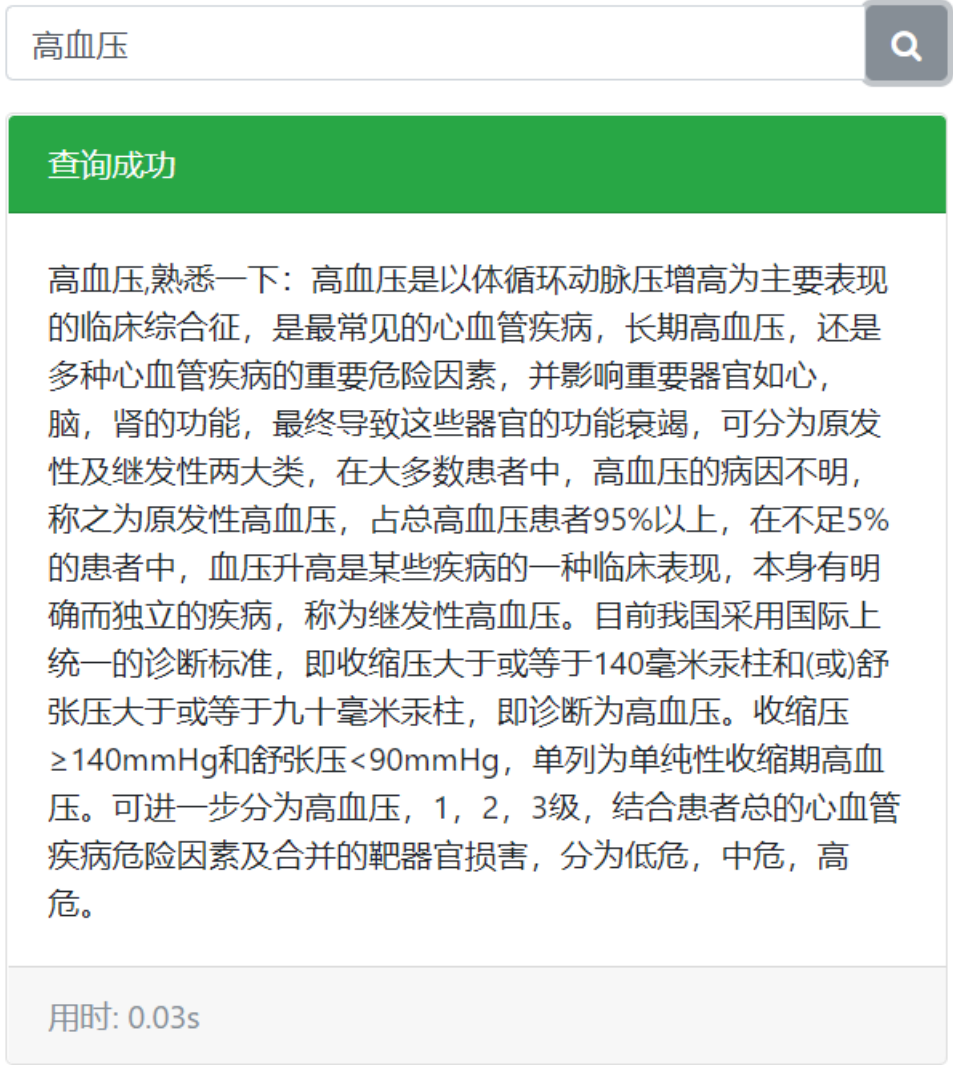


图 4.20 查询疾病描述演示图

## 4.7 本章小结

本章主要介绍了基于知识图谱的疾病知识问答系统的系统设计相关信息，如问答系统算法设计、系统开发环境、系统设计架构以及系统分析。重点介绍了如何基于传统的基于模板匹配的语义解析方法实现问答系统，从最终的问答效果来看，本文实现的系统功能完善，界面友好，响应迅速、准确，具有一定的应用价值。

## 5 总结与展望

### 5.1 总结

问答系统是指让计算机自动回答用户所提出的自然语言问题，是信息服务的一种高级形式。不同于传统的搜索引擎基于关键词匹配并返回相关文档的检索方法。基于知识图谱的问答系统返回的是精准的自然语言形式的答案。因此，基于知识图谱的问答系统有着很好的发展前景。目前，各大高校以及研究机构都在该方向进行积极的探索。

本文的主要贡献如下：

- (1) 介绍了爬成技术，为知识图谱的构建提供数据支撑。
- (2) 详细的介绍了知识图谱构建的各个环节及其关键技术。阐述了知识获取、实体链接、关系抽取等理论。同时介绍了目前智能问答技术的主要实现方法。
- (3) 使用 Neo4j 图数据库构建了疾病领域的知识图谱，其中包含近 2.5 万实体，约 25 万实体关系，并对构建的知识图谱进行可视化展示。
- (4) 构建基于知识图谱的疾病知识问答系统。使用 Flask 作为后端开发框架，Bootstrap 作为前端模板构建了一个简单易用的用户交互界面。并对系统支持的问答类型进行分析。最后，结合部分具体问题类型进行可视化展示。从最终的展示结果来看，本文实现了功能完善，界面友好，响应迅速的基于知识图谱的疾病知识问答系系统。

在工业界的实际生成环境种，一个好的问答系统是由多个部门相互合作，共同完成的。因此，他们实现的系统具有良好的鲁棒性。鉴于个人能力有限，同时考虑到时间等因素，本文是对智能问答系统的尝试，这是有益的一次探索，我在该问答系统的实现过程也收获了很多宝贵的经验。我将在研究生阶段继续深入探索相关工作。

## 5.2 展望

基于知识图谱的问答系统有着传统搜索引擎不可比拟的优势。但是，基于知识图谱的问答技术仍面临着许多关键问题，如复杂问句的问答方法，面向问答的深度推理能力以及下文关联的多轮问答等问题。随着计算机算力的不断提升，深度学习、知识工程等技术得到了长足的发展。随着自然语言处理、深度学习、知识工程等相关技术的飞速发展，基于知识图谱的问答技术将会得到很大的突破。我们也期待基于知识库的智能问答技术相关应用的落地实践，这将进一步为人们的生产生活提供便利。

## 参考文献

- [1] Etzioni O. Search needs a shake-up[J]. *Nature*, 2011, 476(7358): 25-26.
- [2] Green Jr B F, Wolf A K, Chomsky C, et al. Baseball: an automatic question-answerer[C]//Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference. 1961: 219-224.
- [3] Woods, William A. "Progress in natural language understanding: an application to lunar geology." *Proceedings of the June 4-8, 1973, national computer conference and exposition*. 1973.
- [4] Suchanek F M, Kasneci G, Weikum G. Yago: a core of semantic knowledge[C]//Proceedings of the 16th international conference on World Wide Web. 2007: 697-706.
- [5] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]//Proceedings of the 2008 ACM SIGMOD international conference on Management of data. 2008: 1247-1250.
- [6] Lehmann J, Isele R, Jakob M, et al. DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia[J]. *Semantic Web*, 2015, 6(2): 167-195.
- [7] 刘 峤, 李 杨, 段 宏, 刘 瑶, 秦 志 光. 知识图谱构建技术综述 [J]. 计算机研究与发展, 2016, 53(03): 582-600.
- [8] Berant J, Chou A, Frostig R, et al. Semantic parsing on freebase from question-answer pairs[C]//Proceedings of the 2013 conference on empirical methods in natural language processing. 2013: 1533-1544.
- [9] Cai, Qingqing, and Alexander Yates. "Large-scale semantic parsing via schema matching and lexicon extension." *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2013.
- [10] Fader A, Zettlemoyer L, Etzioni O. Open question answering over curated and extracted knowledge bases[C]//Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014: 1156-1165.

- [11] Yao X, Van Durme B. Information extraction over structured data: Question answering with freebase[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2014: 956-966.
- [12] Bordes A, Chopra S, Weston J. Question answering with subgraph embeddings[J]. arXiv preprint arXiv:1406.3676, 2014.
- [13] Yang M C, Duan N, Zhou M, et al. Joint relational embeddings for knowledge-based question answering[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 645-650.
- [14] Bordes A, Weston J, Usunier N. Open question answering with weakly supervised embedding models[C]//Joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin, Heidelberg, 2014: 165-180.
- [15] Dong L, Wei F, Zhou M, et al. Question answering over freebase with multi-column convolutional neural networks[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015: 260-269.
- [16] Yih S W, Chang M W, He X, et al. Semantic parsing via staged query graph generation: Question answering with knowledge base[J]. 2015.
- [17] Zhang Y, Liu K, He S, et al. Question answering over knowledge base with neural attention combining global knowledge information[J]. arXiv preprint arXiv:1606.00979, 2016.
- [18] Python 网络爬虫技术[M]. 人民邮电出版社 , 江吉彬, 2019.
- [19] <https://www.jianshu.com/p/3c2b18920616>
- [20] 漆桂林, 高桓, 吴天星. 知识图谱研究进展[J]. 情报工程, 2017, 3(1): 4-25.
- [21] Chen K, Wang J, Chen L C, et al. Abc-cnn: An attention based convolutional neural network for visual question answering[J]. arXiv preprint arXiv:1511.05960, 2015.
- [22] Qu Y, Liu J, Kang L, et al. Question answering over freebase via attentive RNN with similarity matrix based CNN[J]. arXiv preprint arXiv:1804.03317, 2018, 38.

## 致谢

时光荏苒，本科美好时光即将结束。

在此，我首先向我的两位指导老师黄利老师和何铁科老师表示感谢，两位老师在我的论文选题、撰稿、修改过程中给出了许多宝贵的意见，这让我收获良多，也正是在两位老师的悉心指导下，我才得以顺利完成这篇论文。

再者，感谢我的女朋友，在她的陪伴与支持下，我才得以取得今天的成绩。

最后，学院的老师及同学们对我的鼓励和支持让我终生难忘，伴随着老师的谆谆教诲与同学们的陪伴，我的大学生活变得丰富多彩。