2021 届硕士专业学位论文

分类号：_____ 学校代码：10269

密　级：_____ 学　　号：71174500156

East China Normal University

硕士专业学位论文

MASTER'S DISSERTATION

论文题目：基于知识图谱的食疗健康问答机器人的研究与实现

院　　系：软件工程学院

专业学位类别：工程硕士

专业学位领域：软件工程

论文指导教师：刘献忠　副教授

论 文 作 者：吴浩锋

2021 年 11 月 11 日

Dissertation for master degree in 2021

Student ID:71174500156
University code:10269

East China Normal University

Title: Research and implementation of food therapy health Q&A robot base on knowledge graph

Department: Software Engineering

Type: Master of Engineering

Domain: Software Engineering

Supervisor: Associate Prof. Xianzhong Liu

Candidate: Haofeng Wu

11 , 2021

华东师范大学学位论文原创性声明

郑重声明：本人呈交的学位论文《基于知识图谱的食疗健康问答机器人的研究与实现》，是在华东师范大学攻读硕士/博士（请勾选）学位期间，在导师的指导下进行的研究工作及取得的研究成果。除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名： 吴浩锋 **日期：** 2021 年 11 月 11 日

华东师范大学学位论文著作权使用声明

《基于知识图谱的食疗健康问答机器人的研究与实现》系本人在华东师范大学攻读学位期间在导师指导下完成的硕士/博士（请勾选）学位论文，本论文的研究成果归华东师范大学所有。本人同意华东师范大学根据相关规定保留和使用此学位论文，并向主管部门和相关机构如国家图书馆、中信所和“知网”送交学位论文的印刷版和电子版；允许学位论文进入华东师范大学图书馆及数据库被查阅、借阅；同意学校将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于（请勾选）

（ ）1.经华东师范大学相关部门审查核定的“内部”或“涉密”学位论文*，于 年 月 日解密，解密后适用上述授权。

（√）2.不保密，适用上述授权。

导师签名 本人签名 吴浩锋

2021 年 11 月 11 日

* “涉密”学位论文应是已经华东师范大学学位评定委员会办公室或保密委员会审定过的学位论文（需附获批的《华东师范大学研究生申请学位论文“涉密”审批表》方为有效），未经上述部门审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权）。

吴浩锋硕士学位论文答辩委员会成员名单

姓名	职称	单位	备注
杨智应	教授	上海海事大学	主席
肖波	副教授	华东师范大学	
陈伟婷	副教授	华东师范大学	

摘要

近年来，人们已经开始重视如何在日常生活中调养自己的身体,食疗这个话题也时常被提起。科学的进行食疗养生对于身体和人精神的益处是显而易见的，最重要的是它不会对身体造成一些负担。相较于药物治疗，食疗更多的在于提高人体的身体素质，起到更好的预防的作用。食疗的素材例如食物水果等在生活中更容易获得而且相较于药物来说更加物美价廉。再者食疗养生能在潜移默化中改善自身的身体而不会让患者有什么不舒服以及疼痛的感觉。

但是现阶段的有关食疗健康的数据还是比较离散，用户无法在有效的时间内找到针对自身调养的食疗方案。本文将围绕食疗知识图谱的构建和智能问答系统的设计来展开：

（1）研究如何构建有关于食疗健康方面的知识图谱，本论文将分为三个方面进行详细的介绍。知识抽取主要包括基于本体的抽取（包括知识挖掘）以及基于模型的抽取。本文的数据主要来源于中华养生网以及医疗信息科普中心的结构化和半结构化的数据，知识抽取的操作流程是对数据进行本体构建、特征抽取以及监督学习。知识融合主要是实体向量相似度计算来进行实体消歧等。知识存储本文使用的是 Neo4j 和 MySQL 混合的方式实现。

（2）对于智能问答的设计，在建立食疗知识图谱后，使用 jieba 进行分词以及词性标注的预处理，其次使用 Word2vec 模型计算词向量的相似度，进而生成特征向量，然后通过朴素贝叶斯预排序以及 CNN 排序学习反馈到用户更准确的答案。

最后，本文通过上述内容的研究构建了食疗健康的知识图谱，并通过朴素贝叶斯以及 CNN 的算法提升了食疗健康问答的精确度，最终构建了一个满足用户需求的食疗健康问答系统。

关键词：问答系统，食疗健康，知识图谱，实体识别

ABSTRACT

In recent years, people have begun to pay attention to how to adjust their bodies in daily life, and the topic of diet therapy is often mentioned because of this.The benefits of scientific dietary regimen for the body and human spirit are obvious.The most important thing is that it does not cause any burden on the body. Compared to medication,diet therapy is more about improving the physical fitness of the human body and playing a better preventive role. Secondly, historical materials such as food and fruits are more readily available in life and are more affordable and cheaper than drugs.Furthermore, diet therapy can improve one's own body in a subtle way without making patients feel uncomfortable and painful.

However, the current data of dietary health are still relatively discrete, and users cannot find a dietary solution for their own rehabilitation within an effective time. This article will focus on the construction of knowledge graph of diet therapy and the design of an intelligent question answering system:

(1)To study how to construct a knowledge graph about dietary health, this paper will be divided into four aspects for detailed introduction. Knowledge extraction, mainly includes ontology-based extraction (including knowledge mining) and model-based extraction. The data in this paper mainly come from structured and semi-structured data of China Health Network and the medical information popularization Center, etc.. The process is feature extraction, knowledge fusion, and supervised learning. Knowledge fusion mainly focuses on entity vector similarity calculation for entity disambiguation. Knowledge processing mainly focuses on knowledge reasoning (AMIE algorithm).This article implements knowledge storage by using Neo4j and MySQL .

(2)About the design of intelligent question answering, after the establishment of the diet knowledge graph, jieba is used to pre-process word segmentation and part-of-speech tagging.

Secondly,the Word2vec model is used to calculate the similarity of the word vector,and then the feature vector is generated,and then learn feedback to users through naive Bayes pre-sorting and CNN sorting to more accurate answers.

Finally, this article builds a knowledge spectrum of dietary health through the research of the above content, and improves the accuracy of the dietary question through various algorithms, and finally builds a dietary health question and answer system that meets user needs.

Keywords: Q&A system, food therapy health, KG, entity recognition

目录

第一章 绪论

1.1 研究背景与意义

食疗健康指的是不通过药物治疗而通过食物的均衡搭配来达到调养身体机能以及预防疾病。其中涉及到了很多的关于营养学方面的知识。合理的食物搭配[1]不仅能有效的促进细胞组织的新陈代谢，还能为细胞成长提供较均衡的营养。了解更多的有关于食疗健康方面的知识有助于我们在不使用药物的情况下提高机体免疫能力，进而促进身体的代谢循环，达到增强身心，促进健康的目的。

随着网络的发展与更新，网上关于食疗方面的知识越来越多，但是人们并不能十分有效的获取到自己想要得到的需求方面的答案，其主要原因有：1、现阶段的有关食疗健康的数据还是比较离散，用户需要自己去评论区寻找和回复。2、网站没有对特定问题“对症下药”，反而给出来的都是一些模凌两可的食疗方案，令用户难以判断。

面对用户的食疗健康方面的需求，知识图谱在这方面展现出了巨大的潜力。

知识图谱在很多的行业中都扮演着重要的作用，就语义而言，其实质是通过处理各种复杂繁琐的数据而得到的<实体，关系，实体>三元组组成的巨大的知识库。知识图谱的应用从 google 搜索至聊天机器人[2]涉及的领域不仅仅只是人工智能，还包括其他方面。它不但能为我们提供对于海量数据知识的处理能力，使得我们能够更迅速的了解互联网的信息表达方式，还能更精准的捕捉到海量信息背后所隐藏的更大的价值。并且，基于知识图谱的智能问答也正逐渐成为一种自然交互的新趋势，针对垂直领域的智能问答系统所具备的精确定位问题并快速给出回复的能力已经成为发展的新目标。

问答系统处理问题分为几步：

1. 获取用户的问题需求，通过处理用户输入的自然问句，然后生成关键词，确定答案的类型以及问题的向量矩阵。

2. 检索，负责根据所转换出来的语义表示，使用强化学习以及检索排序筛选出候选答案。

3. 评分，返回得分最高的答案。而其中要重点关注的问题主要是如何进行

数据知识化和知识数据化，然后进行有效的知识抽取以及知识推理以及语言处理等各方面，得到用户问句的最优解然后反馈。

如今的智能问答正蓬勃发展，本文的论述主要通过两方面展开：一是如何构建一个良好的有关食疗健康的知识图谱，换而言之是知识数据库的构建。当今互联网是一个拥有巨大信息量的信息聚合体，所以需要进行有效的信息整合，即构建知识数据库。二是对用户所提出的自然问句进行语义解析，即解析问题并且反馈用户所对应的答案。这样既可以节省大部分的查阅文献资料的时间，还可以快速得到已经整合分析过的数据资料。

综上，针对当前的背景，研究中文场景的一个关于食疗健康的智能问答系统有利于人群更具有针对性的对自己的身体健康与安全饮食进行管控，具有重要的现实意义。

1.2 研究现状

在食疗健康领域的知识图谱构建方面，从 2012 的 google 搜索开始确定 Knowledge graph(KG)[3]的概念并致力于完善一个智能检索领域的搜索引擎，之后又很多的公司都加入到了研究 KG 的领域当中，例如百度的“知心”，与此同时各大领域也看到了自身发展在 KG 中潜藏的巨大的前景，知识图谱的发展进入到了喷薄式发展时期。随着科研机构等各方面的推动下，截止到 2016 年，KG 已经出现了超过千万级的巨型概念类知识库。其代表是 Probase。再往后，知识图谱开始往各领域普及，包括智能问答，医疗知识图谱等方面，KG 都显示出了出色的处理能力，它能提供更快捷更精确的定位问题的能力，为各领域驳杂庞大的数据中发现潜藏的价值指出了一条明路。

国内 KG 构建最早的是 zhishi.me，它整合了 Wiki,Hudong Wiki,以及 Baidu Wiki 的海量数据，并为我们提供了一个 sparql 终端用于获取 Linking opendata。目前 zhishi.me 是一个具有超过 125000000 三元组的超大型语义网络。

在各大行业领域中引用广泛的是的漆桂林教授等联合发布的 openKG 知识图谱开放社区，它旨在让 KG 不再仅仅被广泛引用于搜索或者 bot 问答，而可以在更多的垂直领域发挥其出色的处理能力，本文所要解决的问题之一就是知识图

谱在食疗健康方面的应用[4]，如何以 NLP 的视角从文本中抽取语义以及获取到一个结构化的有关于食疗健康的知识库。

在智能问答[5]方面，随着深度学习以及神经网络的快速发展，基于知识图谱的智能问答系统相关的技术越来越成熟，这也成为了当前的热门研究之一。在卷积神经网络(CNN)等训练模型在文本处理中相继获得优异的成果后，智能问答在语义匹配和特征获取方面的精确度越来越高，所获得的问答质量也日益优良。

在上述的基础上，研发一个基于食疗健康的智能问答系统具有很大的发展前景。

1.3 论文主要研究内容

本文研究的主要研究内容是：

1. 研究食疗健康方面的知识图谱的构建。

本文构建知识图谱主要经过了知识抽取，知识融合以及知识存储三个方面。知识抽取使用 KBC 系统针对来源于中华养生网以及医疗信息科普中心的结构化和半结构化的数据进行知识抽取操作。知识融合指的是通过实体向量相似度计算来进行实体消歧，最后把数据存入到图数据库 Neo4j 和关系数据库中。

2. 研究基于食疗健康方面的智能问答的构建以及如何更精确的获取用户需要的数据。

在建立食疗知识图谱后，通过对比研究选择较好的方法进行分词以及词性标注的预处理，进而生成问题特征向量，然后通过分类训练学习，研究 CNN 分类训练与朴素贝叶斯预分类等算法，对比各模型的准确率与召回率等参数，从而选择一个更佳的模型，从而反馈用户更准确的答案。

3. 研究一种基于食疗健康方面的智能问答的系统实现。

本文将基于智能问答的构建，实现基于食疗健康方面的智能问答的系统，获取到最终结果反馈到用户。

1.4 论文组织结构

本文将论文分为六个部分，下面对每一章进行简单介绍如下。

第一章，绪论。通过对论文的研究背景的介绍以及分析，获得研究这方面的实际意义，在此基础上介绍了现阶段的有关于知识图谱以及智能问答的现状与发展。这种健康的饮食，近年来，越来越多的人在关注，但这方面的数据相对离散，关系复杂，如何对此领域方面的数据进行有效的整合以及数据存储，如何更有效地获取数据也是将要解决的任务。本文构建了食疗健康领域的知识图谱以及相关的智能问答系统，为快速有效的获取数据提供服务。

第二章，关键技术综述。主要对本文需要用到的各种技术以及算法进行简单的介绍。主要包括知识图谱以及相关方面的知识，智能问答以及相关方面的知识，包括神经网络方面的知识。

第三章，面向食疗健康领域的知识图谱构建。本章通过知识抽取，知识融合以及知识存储三个方面构建了基于食疗健康的知识图谱。

第四章，基于知识图谱的智能问答系统模型设计。主要讲解实体识别以及关系抽取方面的分类模型（包括 CNN 模型）以及实验结果分析。

第五章，系统整体实现与测试。主要简述了系统的工作流程、开发环境以及系统测试。

第六章，总结与展望。总结论文所进行的工作，并对其中所包含的一些不足进行分析，然后对未来这个领域的发展进行展望。

第二章 关键技术综述

2.1 知识图谱及相关技术

知识图谱是由大量的三元组构成的关系型知识库。我们在构建知识图谱的时候主要是从海量的数据中抽取实体以及实体的特征属性。构建知识图谱的方法步骤：首先在连续向量空间中表示实体与关系，然后构建一个用于评估三元组的合理性的评分函数，接着可以通过最大化观察到的三元组的合理性构建模型[6]，三元组模型有 RDF、带标志的 RDF 等。用于查询 RDF 的 SPARQL 以及用于处理多元关系 Freebase（CVT）也将涉及，最后将抽取出来的实体以及属性的三元组导入到数据库中以供接下来的智能问答使用。

知识抽取[7]主要是疾病和食物的实体抽取、事件抽取以及关系抽取，其中需要进行处理的数据主要指的是来自百科的开放书库以及半结构化数据等。结构化数据包括连接数据以及数据库数据等，半结构化数据包括网页列表等。食疗健康的数据主要来自网页列表数据以及开放数据。知识抽取可以分为基于本体的抽取（如 TransE）以及基于模型抽取（如基于触发词的 Pattern）。关系抽取[8]主要分为传统关系抽取以及开放域关系抽取，涉及监督学习，其中包括 CNN 以及 BI-LSTM-CRF，图推理等。实体链接是通过构建实体向量然后进行相关的相似度计算，从候选实体集中找出实体指称及目标实体的过程。在此过程中，可通过 TF-IDF 算法[10]进行实体消歧[12]，评估抽取的实体对文章的重要程度。知识监督指的是对于获取到的数据质量评估。通过开放性的数据库进行知识抽取以及知识融合等各个步骤[13]获取到的数据也是存在着一定的实体以及关系抽取错误的，为了对自己的数据负责，所以需要对所获得的数据进行质量评估[14]。增高知识库中数据的可信度。

知识融合是在知识抽取的基础上先进行数据的预处理，然后通过相似度计算机型本体对齐的过程。

2.2 分类算法

分类算法是数据挖掘的核心之一，其主要是用于识别事务属于哪一类。

2.2.1 卷积神经网络 CNN

卷积神经网络[15]中最关键的几个因素是：卷积计算、关联权重共享参数和池化函数，对文本进行卷积计算一般只选择一维卷积。关联权重和池化层这一组合让卷积神经网络可以迅速利用所输入的数据。

卷积神经网络已经在文字领域以及图片领域获得了巨大的突破，是常用的分类算法之一。Hubel 等学者在猫的神经元上定义了“感受野”。与此同时，日本的 Fukushima 教授受到了 Hubel 教授的启发，在 Hubel 的研究基础上，提出了奠定性神经元理论基础的神经感知机。大江东去浪淘尽，一代又一代的研究学者站在巨人的肩膀上不断前行，在 1998 年，Lecun 设计的 LeNet-5,被广大深度学习领域和神经网络领域的学者认为是 CNN 网络的最初始形态，但由于当时社会还没有出现当今时代的拥有强大计算能力的计算机，所以在当时这个模型并没有得到应有足够的重视。

在 2006 年，Hioton 解决了“局部最优解被困扰”的相关问题，他是通过预训练的方式不断重复的设计与实验，才使得网络逐渐具备深度，并由此开辟了深度学习之路的大门。

2012 年，卷积神经网络 AlexNet 在 TextNet 数据集上大获全胜，并以绝对的准确率拿下了当时的冠军，此后卷积神经网络便与文字识别领域紧紧的联系在了一起，往后流行的卷积模型都是在这个模型上发展而来的。同时，卷积神经网络又分为卷积层，线性整流层，池化层和损失函数层。

2.2.2 朴素贝叶斯

贝叶斯的实质是条件概率，是通过获取先验知识来进行概率推断的生成模型。我们假设给定的目标的各个特征都是相互独立的，那我们只要通过相似度计算获取到哪个类别出现的相对概率最大，那就可以认为目标就属于哪个类别。由此可得朴素贝叶斯是一个高偏差低方差的模型，其对于处理一些小的训练集有明显的优势。

使用朴素贝叶斯[16]分类的流程如下：

1. 进行数据的预处理，获取到训练样本的实体以及其对应的一系列特征属

性。

2. 通过计算每个类别下的 P（y）及每个特征属性在相互独立的情况下的条件概率。

3. 然后对每个类别计算最大项 P（x|y）以及 P(y)。其中 P(x|y)条件概率指的是 y 类别样本出现 x 的概率,P(y)先验概率指训练集中类别为 y 的样本组成的集合出现的概率。

其中对于第二步中，如果使用的特征集是离散类型的时候可以选择多项式分布；而如果特征集是连续的并且符合正态分布的时候，可以使用高斯分布即计算每个类别中特征项的标准差；如果集合适合二项分布的特征时，可以选择贝努利模型。

2.3 智能问答及相关技术

现在是一个数据的时代，丰富的知识都通过结构化以及半结构化等的方式存储在知识库中，所以我们需要通过各种方式去获取到这些数据以及数据背后潜藏着的价值，智能问答由此而生。

基于语义解析的智能问答(KBQA)[17][18]现在是研究领域的热点之一。语义解析[21]是在获得用户的自然问句的基础上对其进行一系列的转换，最终使得其可以在知识库中快捷高效的获取知识。智能问答[19][20]就是在进行语义解析以后，将获取到的信息通过机器语言能够识别的语义表示进行查询检索，最终将用户需要的相关知识进行反馈展现。目前的研究热点已经不再是早期的相关规则的系统，而是在进行浅显查询的同时通过神经网络进行一定的知识推理，使得 KBQA 在应用领域以及灵活性上大大提升。

2.3.1 实体识别

实体识别[22]主要是通过获取到的自然问句进行分词以及标注的过程。

我们需要高效的将用户输入的问句中的关键词获取出来，应该先去除一些毫无关系的词汇，然后对分离出来的关键词进行词性的标准化处理，同时对于多关键词的实体冲突的处理也是现在一直面临着挑战之一。比如同一实体在文章中出现的地点不同以及其所附带的属性的不同也会有不同的词性分类。

2.3.2 词的向量化

词的向量化[23]主要是使用 ship-gram 模型将文本向量化，生成问题向量矩阵，然后使用卷积神经网络将问题向量矩阵成成问题特征向量，接着计算用户问题特征向量与候选答案特征向量相似度，通过排序学习反馈给用户答案。

2.3.3 关系抽取

关系抽取是一种用来获取数据信息的典型任务。我们需要在实体识别以后，对实体向量进行分类训练[24]，获取到实体所具有的特征集以及有效特征，然后进行关系/属性映射。

2.4 本章小结

本章是对本文所使用到的基本技术进行了介绍，是下面实验章节的过渡。知识图谱的相关技术是构建基于食疗健康的知识图谱的基础，分类算法以及智能问答的相关技术是智能问答以及相关实验的重要组成部分，所以在此分别进行了介绍。

第三章 面向食疗健康领域的知识图谱构建

上一章节对于知识图谱的相关知识进行了详细的介绍，本章节在此基础上分成四部分来阐述如何构建一个基于食疗健康的知识图谱，首先需要创建食疗健康的相关实体库，然后通过知识抽取、知识融合以及知识存储等相关技术来构建基于食疗健康的知识图谱。

3.1 整体流程

由于知识的相对离散，构建食疗领域的知识图谱首先要通过百科以及HTML 中的半结构化知识进行知识抽取，然后将这些知识进行筛选梳理，并进行知识融合（构建同义实体库），接着通过知识推理以及质量评估等方面进行知识加工，最后存储到知识库中（MySQL,Neo4j），这就是构建知识图谱的总体流程。

该模块的整体流程如图 3.1 所示：

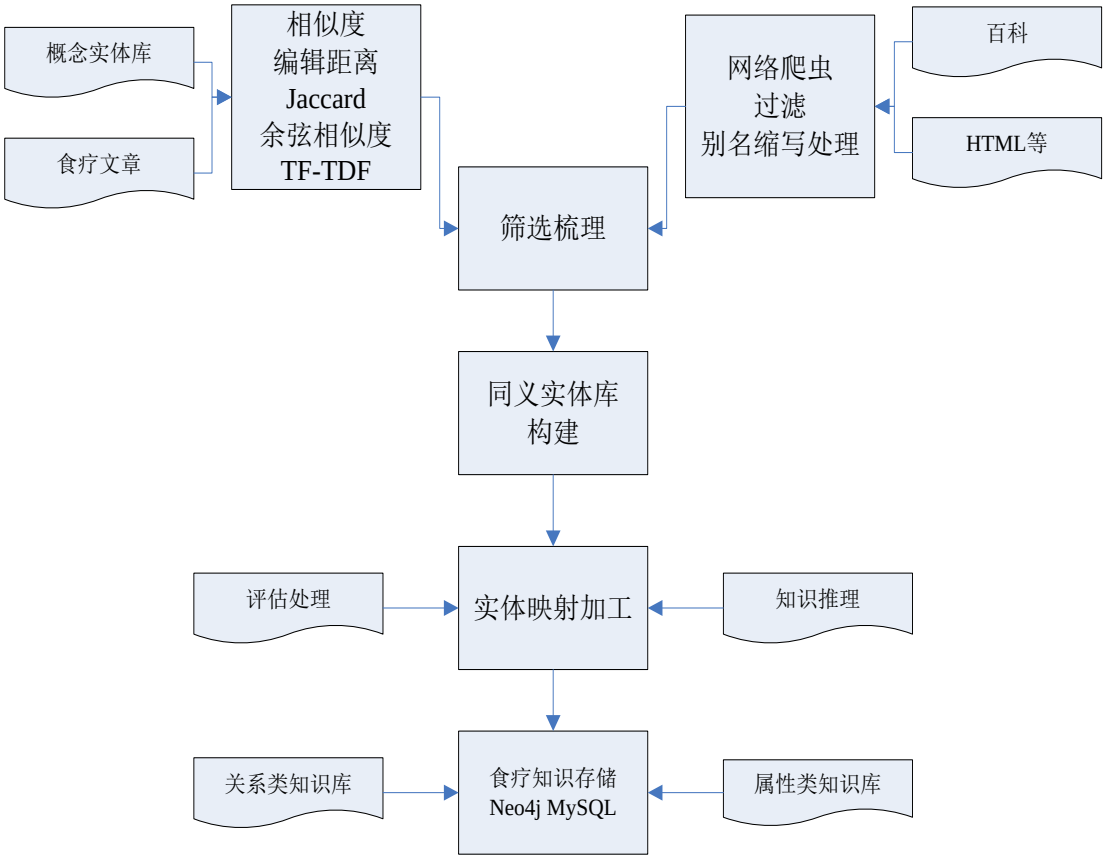


图 3.1 构建知识图谱总体流程图

3.2 创建本体

在构建知识图谱之前，要在数据库中提前创建本体。首先创建概念实体库 fruit、vegetable 和 disease 等，概念实体库中主要用来存储有关于食疗健康的相关实体以及实体所关联的属性关系。例如 vegetable，除了设置默认属性 name 以外，编辑 fruit/vegetable 与 disease 的关系:prevent、remission、resist。对于 disease 添加新属性 disease_result、disease_symptoms、treatment,同时编辑其与 fruit/vegetable 的关系:eat_more 等，然后将创建的本体以及相关信息存入到 MySQL 和 Neo4j 关系数据库中。相对应的数据库中的实体对应属性/关系的部分表如表 3.1 所示：

表 3.1 属性/关系表

food_health_test001_158328272079715692_attribute_definition 0.006 sec. 0 50

	_id	name	alias	type	domain_value	data_type	data_uni	is_functional	direction	constraints
1	ObjectId(...	治疗方法		0	3	5		0	0	{}
2	ObjectId(...	症状		0	3	5		0	0	{}
3	ObjectId(...	病因		0	3	5		0	0	{}
4	ObjectId(...	预防		1	19	0		0	0	{}
5	ObjectId(...	缓解		1	19	0		0	0	{}
6	ObjectId(...	抗		1	20	0		0	0	{}
7	ObjectId(...	多吃		1	3	0		0	0	{}

构建此知识谱图的数据主要来源于中华养生网以及医疗信息科普中心中的数据，从这些数据中通过知识抽取等操作把相关的实体以及属性存入到数据库中。逻辑模型图如图 3.2 所示：

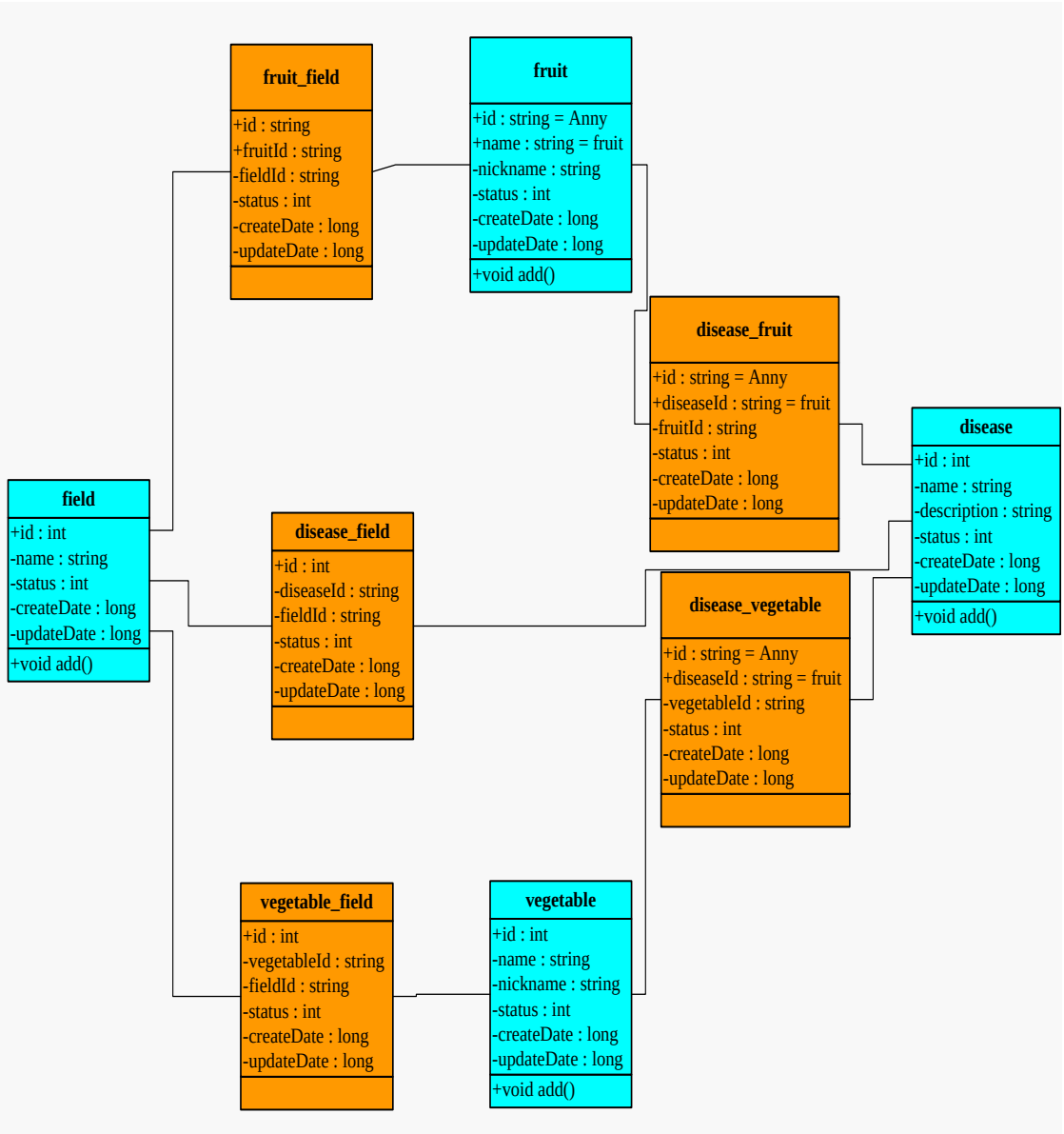


图 3.2 食疗领域的数据库逻辑模型图

3.3 知识抽取

3.3.1 知识抽取相关流程

构建食疗健康的知识图谱的主体任务是进行食疗知识抽取，也就是先要抽取知识库。知识抽取根据数据结构化与开放型等类型的不同分为传统知识抽取与开放域知识抽取。对于不同类型的数据可以使用相对应的知识抽取的方式。如：

当需要从结构化的知识库中获取数据时，可以选择通过 D2R/Virtuoso 等来处理一些复杂的数据关系；当面对半结构化数据时，会使用一些包装器来进行网页

清洗等（百科知识一般会使用 DBpedia 进行模板匹配来做属性对齐等），而本文主要针对来源于中华养生网以及医疗信息科普中心的结构化和半结构化的数据，可以选择使用 KBC 来进行相关的知识抽取对应的知识抽取。方法分析如表 3.2 所示：

表 3.2 知识抽取的方法比较图

系统	知识抽取方法	抽取对象
DBpedia	模板匹配	百科知识
D2R/Virtuoso	关系映射	结构化的知识库
KBC	特征向量	半结构化数据

相比较上述其他的抽取系统而言具有模块化的特点，便于进行相关分析，而且分布式的数据库也有助于提高知识抽取的性能。其流程如图 3.3 所示。

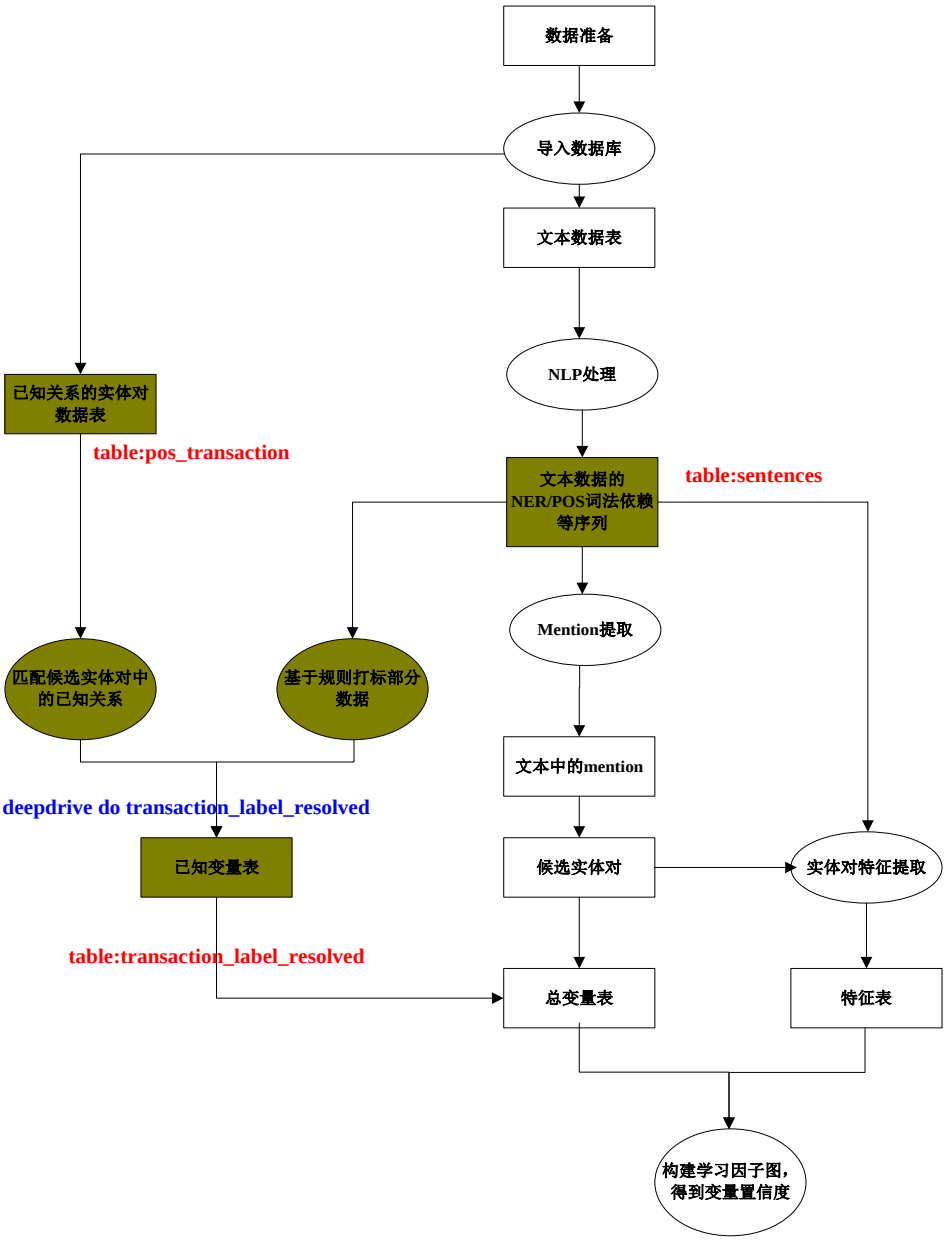


图 3.3 KBC 系统的总体流程图

如图 3.3 所示，首先准备抽取所需的先验数据，存入到数据表$DEEPDIVE DO POS_TRANSACTION。其次定义文章数据表，并对数据库中的文章数据进行 NLP 解析。接着定义 sentences 表，将解析出的数据写入到 sentences 表中。然后进行候选实体抽取，在表结构中定义一张候选实体表，定义候选实体的函数 map_fruit_weakness_mention，导入 sentence 表中的的 token 以及 NER 等，并返回实体和在句子中的位置。再接着特征抽取，定义特征抽取的相关函数以及特征表，将 mention 表中的 NLP 数据进行特征组合然后存入到特征表中。最后进行

数据标注，定义标签表，导入候选实体对集，这些实体对的初始标注统一为 0，如果有相关的 db 数据和候选实体对相关联，则权重+3，并为其标注正负例，在 app.ddlog 中定义标签表，将候选实体对导入并且标注。最终在标记候选实体对中生成最终 note。

3.3.2 知识抽取流程优化

在模型训练的时候，可以对其进行一定的优化。

（1）对数据进行过滤，有利于候选实体的抽取。

非实体进行过滤：通过获取标签表中的标签的头指针并从 0 开始往后遍历直到结束位置为止）。函数调用，从 sentences Table 读入，输出到 Join 实体表中，实体表位于 map_fruit_weakness_mention 中，在此可以进行筛选获取候选实体对。

（2）对所获得的数据进行质量评估。

选择关系三元组构建实体消歧的样本，利用头实体-关系、尾实体、头实体-尾实体等多种表达方式构建大量的正负样本，来学习特征-实体语义关系。在此构建因子图，然后将上述结果注入已知的变量中。因子图（通过因子分解得到局部函数的乘积所构成的双向图）使用一种二模图用来表示函数因式分解后的结果，是概率图的一种。最后通过与先验变量的对比获得采样，并且在样本上做随机梯度下降的权重训练[26]

$$\arg\max \frac{\sum_{I \in T_e} Z(I)}{\sum_{I \in T} Z(I)} \qquad (3\text{-}1)$$

随机梯度下降每次只计算一个样本损失函数，其中 Z(I)表示第 I 个样本的标签值，T 指的是各个特征的极值。

迭代调试的模型效果如图 3.4 所示，其中输出的分数和分数段的正利率分别对应着横轴与竖轴。当分数线和标准线越趋近代表着模型的使用效果越好。

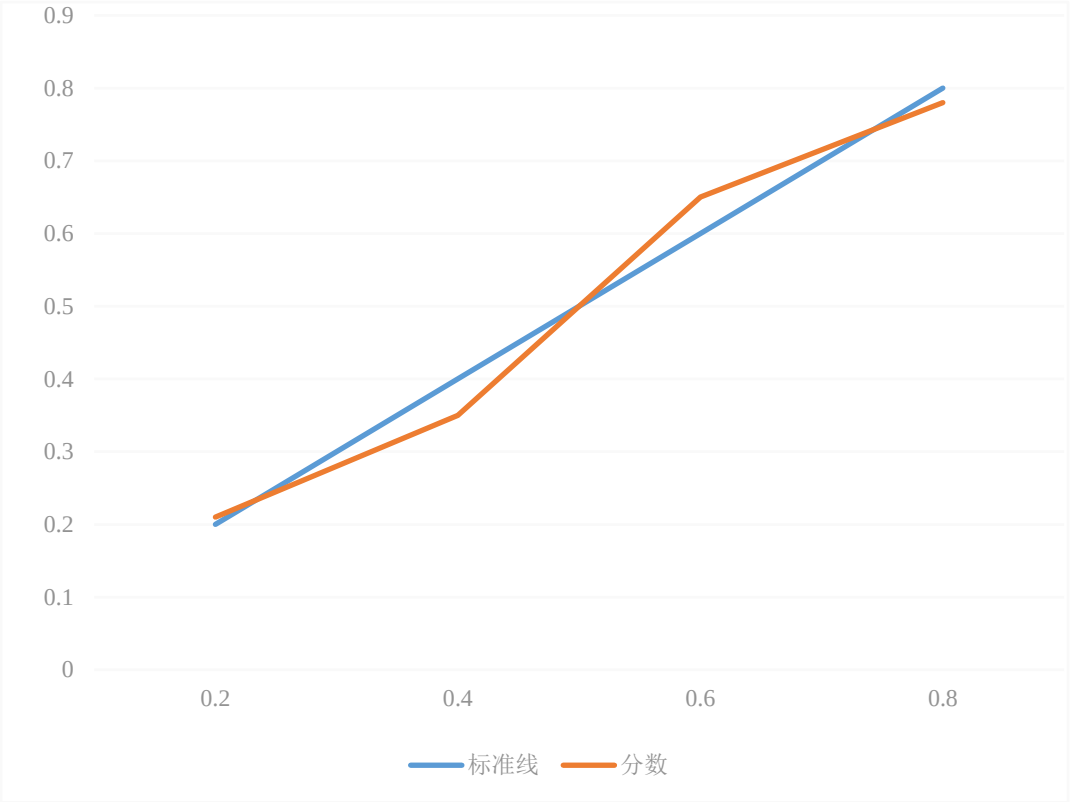


图 3.4 模型输出图

3.4 知识融合

对于进行上一节知识抽取之后得到的数据，我们开始对这些结构化知识进行知识融合。其主要步骤是先进行初步筛选，然后去判断属性的相似度，最后进行知识融合。

首先要对获取到的食疗健康的数据进行初步筛选，这里使用的初步筛选的过滤算法为三角不等式过滤：

对于给定（H,h）,其中 h 是度量的标准，假设 H 中有 x、y、z 三条信息，则三角不等式 $h(x,y)\leqslant h(x,z)+h(y,z)$推理可得到 $h(x,y)-h(y,z)>\theta\rightarrow h(x,z)>\theta$，其中 θ 是阀值。

当 y 远小于目标的集合数量的时候，通过三角不等式过滤[28]可以有效的降低相似度比较的次数。

然后分块，实体对齐后记录记录连接构建实体应用表：假设比较实体 L 和 S，L 和 S 在第 i 个属性上的值是 Li 和 Si, 那么：[L1,S1),sim(L2,S2),…,sim(Ln,Sn)] ->综合单个相似度得到属性相似度向量以及实体相似度。接着计算余弦相似度 Sim(Vec1,Vec2)。

余弦相似度计算公式：

$$\cos\_sim = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} * \sqrt{\sum_{i=1}^{n} y_i^2}} \quad (3\text{-}2)$$

通过余弦公式来计算相似度，计算出来的余弦夹角不受向量的长度影响，余弦的夹角越大，两个向量的相似度越高。但是如果数据库很大，则等待的时间也会相对漫长，对此我们可以先将分母部分进行预计算，然后保存起来，等到需要使用的时候再去获取。

构建词向量矩阵，得到两个词之间的相似度，具体代码如下：

```
public float wordDis(String queryword1, String queryword2) {
    float[] vector1 = wordMap.get(queryword1);
    float[] vector2 = wordMap.get(queryword2);
    float dist = 0;
    if (vector1 == null || vector2 ==null) {
    return 0;
    }
    for (int i = 0; i < vector1.length; i++) {
    dist += vector1[i] * vector2[i];
    }
  return dist;
}
```

当相似度满足所设置的阀值，则进行数据融合。

3.5 知识存储

知识存储是通过上述的步骤获取出了实体以及相对应的属性关系然后存储到知识库中。一般而言我们使用图数据库来存储获取出来的三元组，它不仅能够让我们更直观的进行数据存储，而且在解决负责繁琐的关系问题上有着简单高效的处理方式。

对于知识存储，我们使用 Neo4j 来存储转换后的三元组，其操作的总体流程如图 3.5 所示：

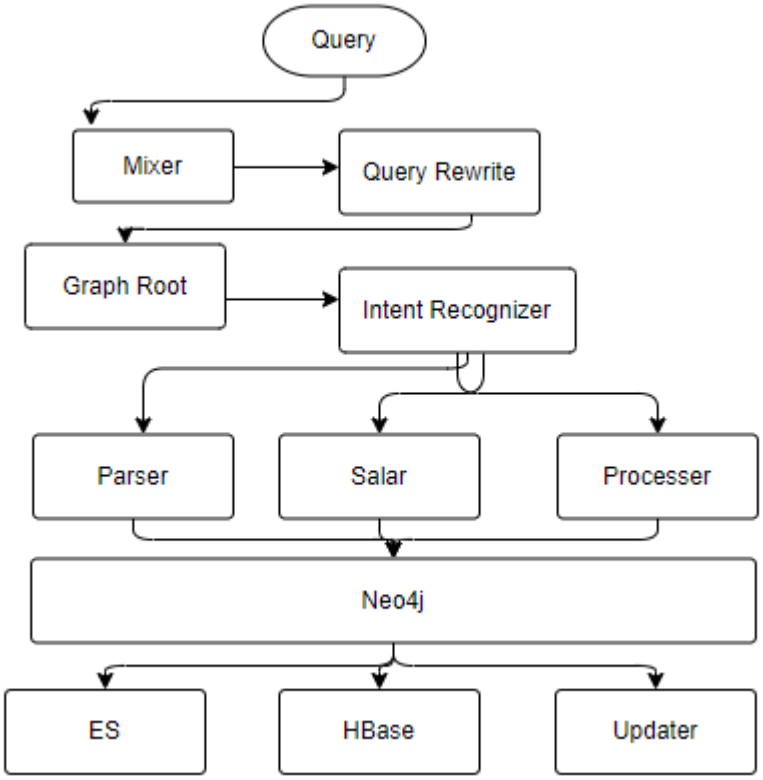


图 3.5 Neo4j 总体流程图

其中 Graph RPC（Parser 解析、Salar 生成器以及 Processer 编辑器）以及 Neo4j 的相关操作（ES、HBase、Updater）都属于 Graph Service。

它不仅查询的速度较快，而且在处理大数据方面的性能也相当不错。Neo4j 是使用 java 开发的开放型知识库，它把所有的写入的数据都依据图形的方式存放在节点与相对应的关系中，简单的属性图如图 3.6 所示：

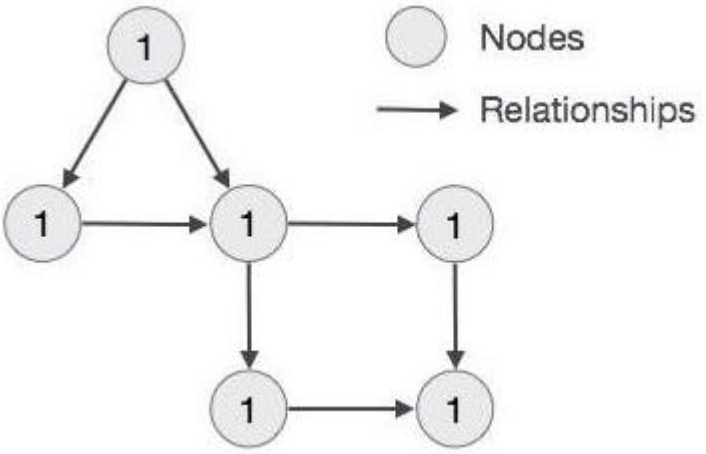


图 3.6 Neo4j 属性图

文件存储结构如图 3.7 所示：

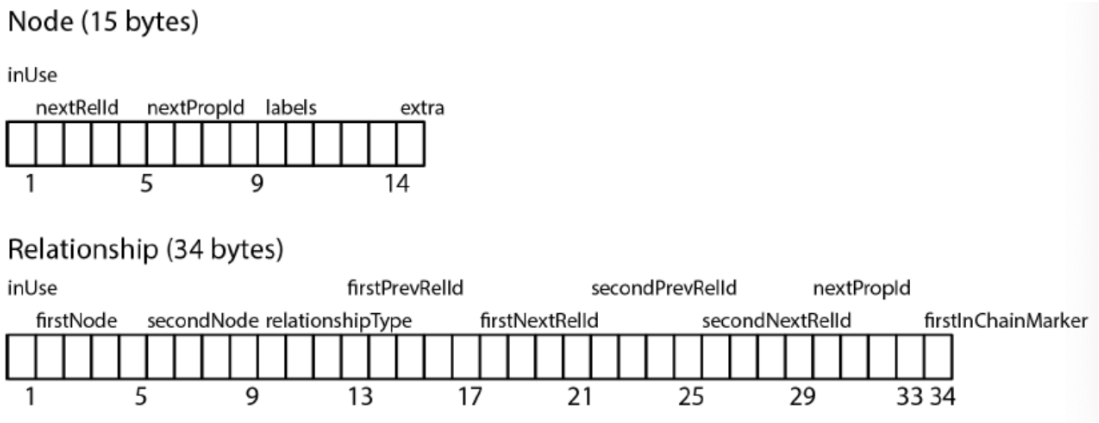


图 3.7 Neo4j[29]文件存储结构图

当存储文件 nodestore.db 时，我们会在第一个字节设置一个表示是否被使用的 Flag，然后在下 4 个字节设置第一个关系的 ID，在紧接着的 4 个字符设置第一个属性 ID，然后在下 5 个字符中设置当前借点的 Label，指向 Label 存储的 LABEL。最后设置用来标识相邻点的标志字符。网络拓扑图如图 3.8 所示。

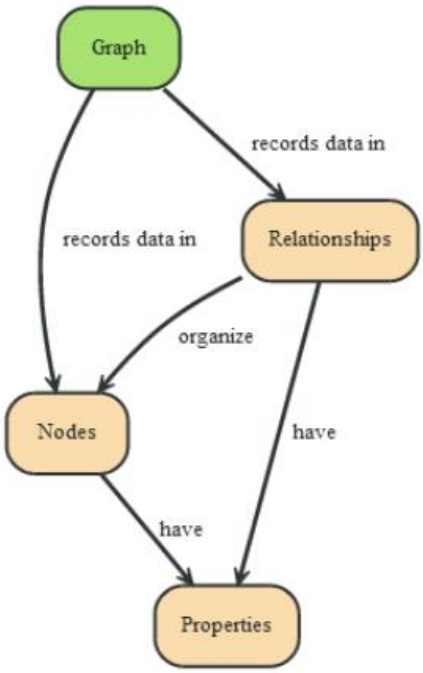


图 3.8 Neo4j 存储文件的网络拓扑图

Neo4j 的数据导入一般我们会使用 Cypher，Batch Inserter 等等，其主要步骤是

1. 创建或者连接一个数据库

   db = GraphDatabase('neodb')

2. 创建节点

   fruits =db.node()

3. 连接到参考节点，方便查找

   db.reference_node.FRUITS(fruits)

4.为 db 组建立索引，便于快速查找

   idx=db.node.indexes.create('fruits')

5.为节点添加关注关系

   with db.transaction:

   get_user('fruit2').FOLLOWS(get_user('fruit1'))

在导入数据之后，数据库的部分知识图谱如图 3.9 所示。

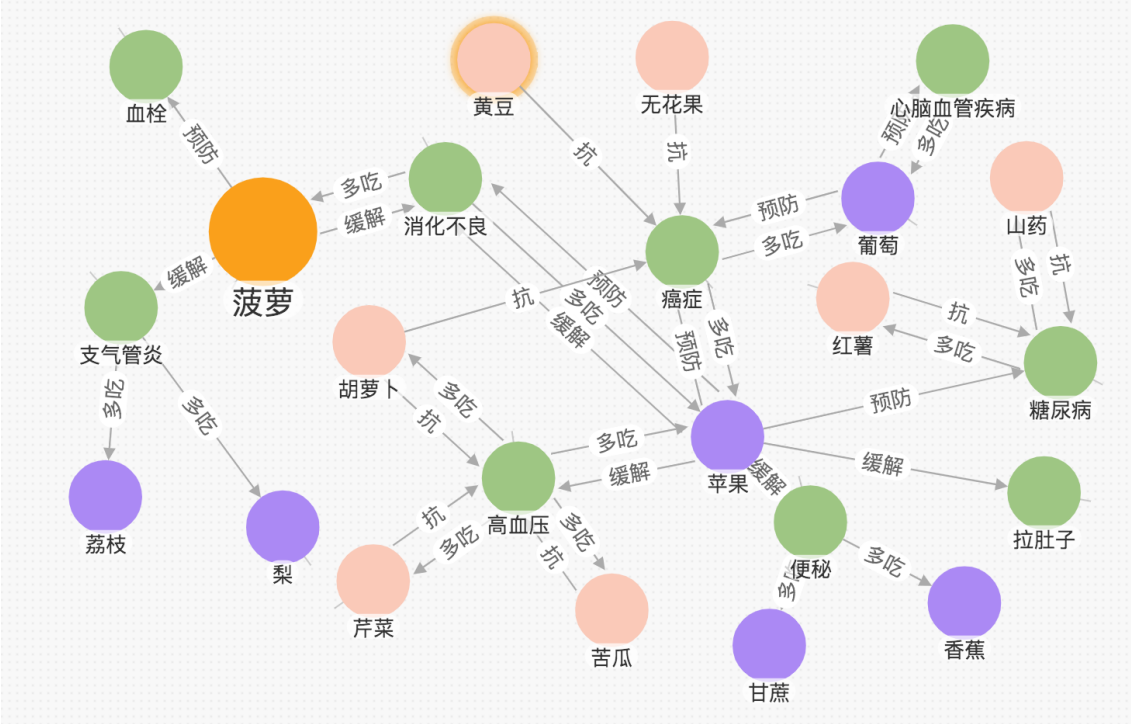


图 3.9 知识图谱图示

Neo4j 可以进行可视化展示导入后的图数据库，将各类食物以及治疗方面的概念之间的关系通过图标的形式进行展示。

表 3.3 为相对应的数据库中的对应实体的部分表：

表 3.3 实体表

basic_info 0.037 sec.

	_id	name	type	meta_data
1	0	食疗图谱	0	
2	3	疾病	0	{ 1 field }
3	6	哮喘		{ 1 field }
4	7	冻疮		{ 1 field }
5	9	秃发		{ 1 field }
6	11	贫血		{ 1 field }
7	14	消化不良		{ 1 field }
8	15	支气管炎		{ 1 field }
9	16	癌症		{ 1 field }
10	17	食物	0	{ 1 field }
11	19	水果	0	{ 1 field }
12	20	蔬菜	0	{ 1 field }
13	21	菠萝		{ 1 field }
14	22	胡萝卜		{ 1 field }
15	23	血栓		{ 1 field }
16	24	黄豆		{ 1 field }
17	25	无花果		{ 1 field }
18	26	苹果		{ 1 field }
19	27	葡萄		{ 1 field }
20	28	心脑血管...		{ 1 field }
21	29	便秘		{ 1 field }
22	30	高血压		{ 1 field }
23	31	糖尿病		{ 1 field }

3.6 本章小结

本章通过对来源于中华养生网以及医疗信息科普中心的结构化和半结构化的数据构建概念库。针对这些数据，首先通过使用 KBC 系统进行知识抽取，然后通过三角不等式过滤和余弦相似度计算进行知识融合，接着将数据存储到 Neo4j 关系数据库以及 MySQL 数据库中，最终构建成基于食疗健康的知识图谱。

第四章 基于知识图谱的智能问答系统模型设计

智能问答的主要工作是分析出自然语言问句中所包含的关键信息，进而获取其包含的语义后，再基于食疗问答领域的知识图谱查询相关的知识，最后获取到准确的答案返回。本章将进行详细的阐述。

4.1 总体流程

智能问答系统模型主要分为实体识别和关系抽取两部分。

实体识别实质是对序列化数据进行标注的过程，其中包含了分词以及各种词性标注。就理论上而言，从自然问句中分离而出的各个词语都可以单独构成一类：人名（Person）、水果名（Fruit）、疾病（Illness）等。

其中构建实体的三元组词典的时候，需要考虑到实体命名的类别根据不同的特征应该有不同的处理方式，不能使用统一的模型去刻画人名。例如疾病比较适合基于词的三元组作为模型，而人名可以基于字的三元模型。

实体识别的流程如图 4.1 所示：

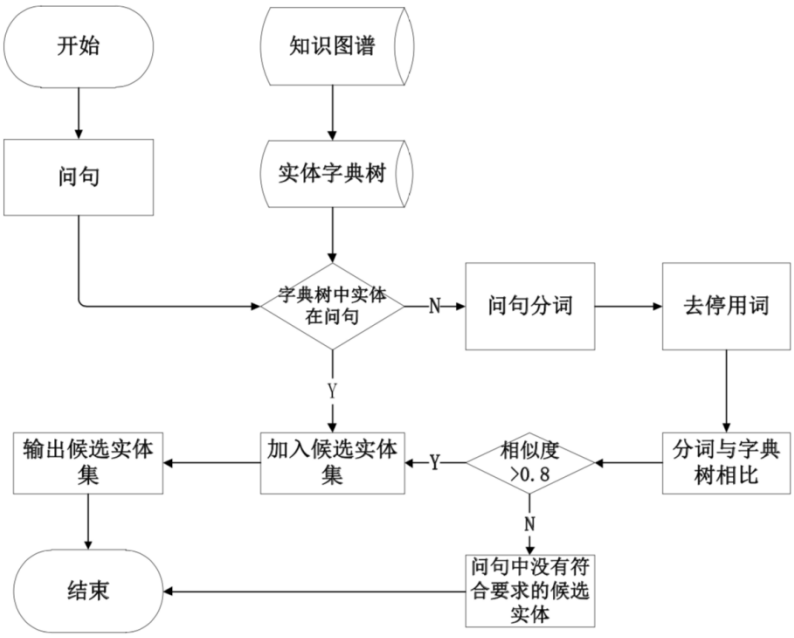


图 4.1 实体识别流程图

关系抽取是一种用来获取数据信息的典型任务，主要分为四大类，分别是有监督的，半监督的，弱监督的以及无监督学习。

有监督的学习通常采用分类算法[33]，以获得通过训练特征数据的有效信息，然后用训练过的分类器进行关系抽取。

半监督的学习指的是 Bootstrapping，它实质是用标注反复迭代的过程。

本文选择的分类器是 CNN 分类器，在进行卷积池化等一系列训练操作前，我们可以对问句进行预分类，本文通过朴素贝叶斯将问句预分类为定义型、列表问答型以及事实型。这样可以降低 CNN 执行所占用的资源，同时也能提高检索的精确度和效率。

当获取到自然问句中的实体以及其对应的关键特征后，会映射到知识图谱中去进行属性关系的匹配，最终获取到答案返回，下面对所用到的关键技术进行介绍。

4.2 分词

问句分词本文对 jieba 分词与 Bi-LSTM+CRF 分词器进行了对比。jieba 分词是采用了动态规划查找路径的方式来进行分词标注，对于知识库中所没有的词汇也可以使用 HMM 来生成写入。相比于 Bi-LSTM+CRF 分词器而言 jieba 分词更适用。相似度计算在本文主要使用的是 Word2vec 模型，通过计算两个语义向量（自然语句向量以及特征向量）的余弦相似度进行对比（其中余弦夹角与向量相似度成正比）。

4.2.1 jieba 分词

jieba 分词的分词以及相关的词性标注如图 4.2 所示：

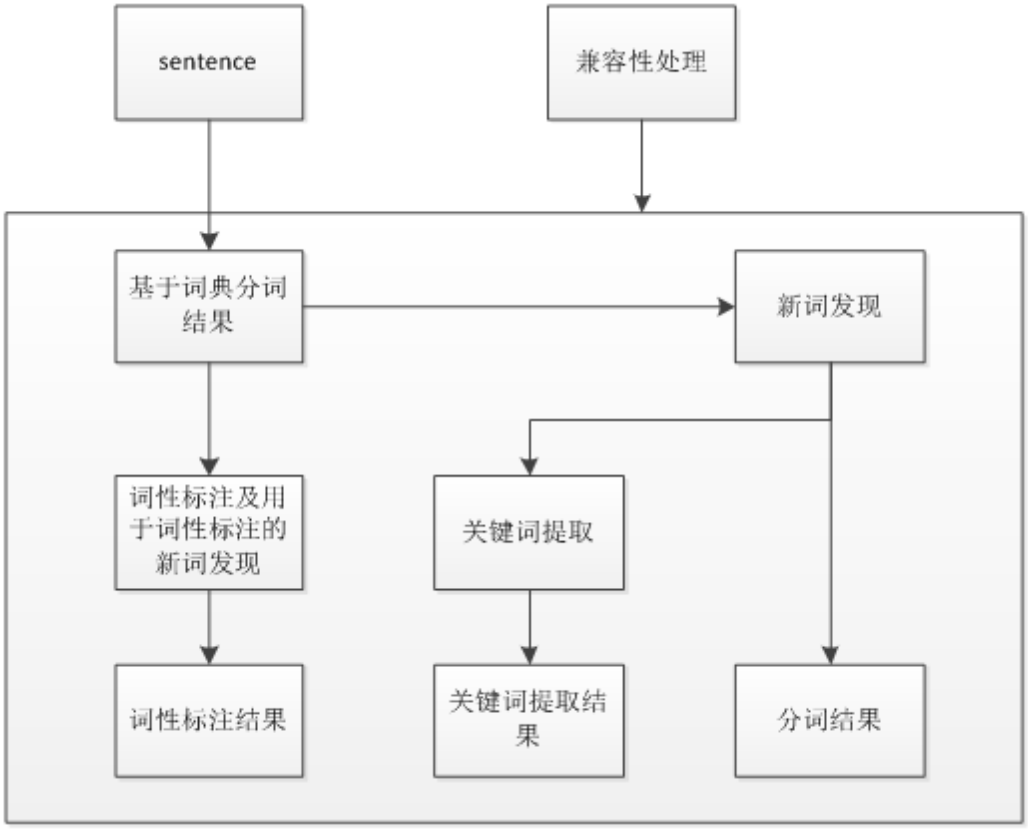


图 4.2 jieba 分词的工作流程图

jieba 分词的三种分词模式如下：

1）全模式下 输入“白血病需要多吃哪些水果”

返回：白血病 病需 需要 多 吃 哪些 水果

2）精确模式下，输入“白血病需要多吃哪些水果”

返回：白血病 需要 多吃 哪些 水果

模式的执行原理如图 4.3 所示：

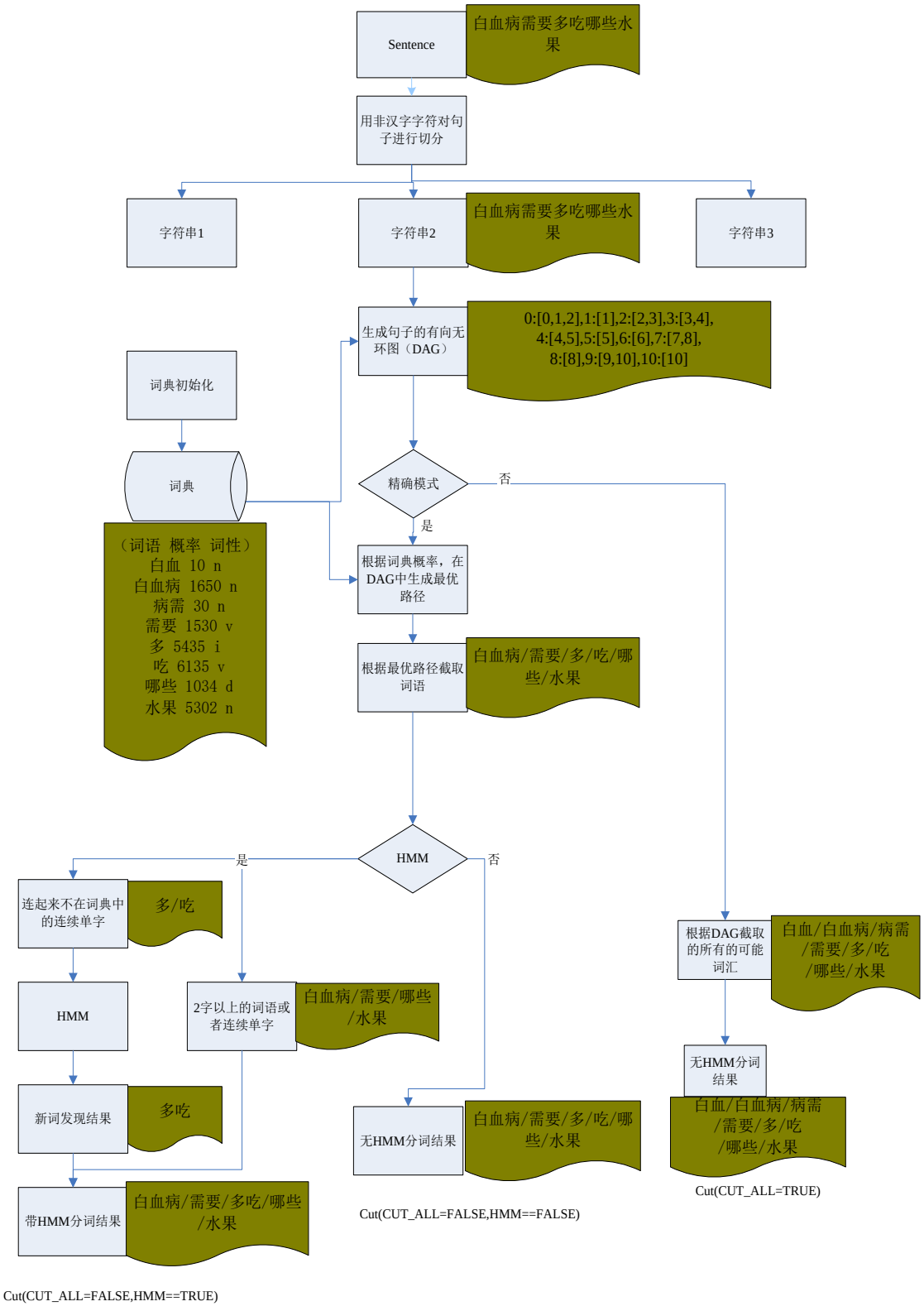


图 4.3 jiaba 分词全模式与精确模式下的执行原理

其中如果开启了 HMM[30]，在进行截取词语的时候会去发现新词，把没有在三元组词典中出现的连续的单字组成词汇存储。

3）搜索引擎模式

seg_list = jieba.cut_for_search（“病毒性感冒需要多吃哪些水果”）

返回：病毒 毒性 性感 感冒 病毒性 病毒性感冒 需要 多吃 哪些 水果

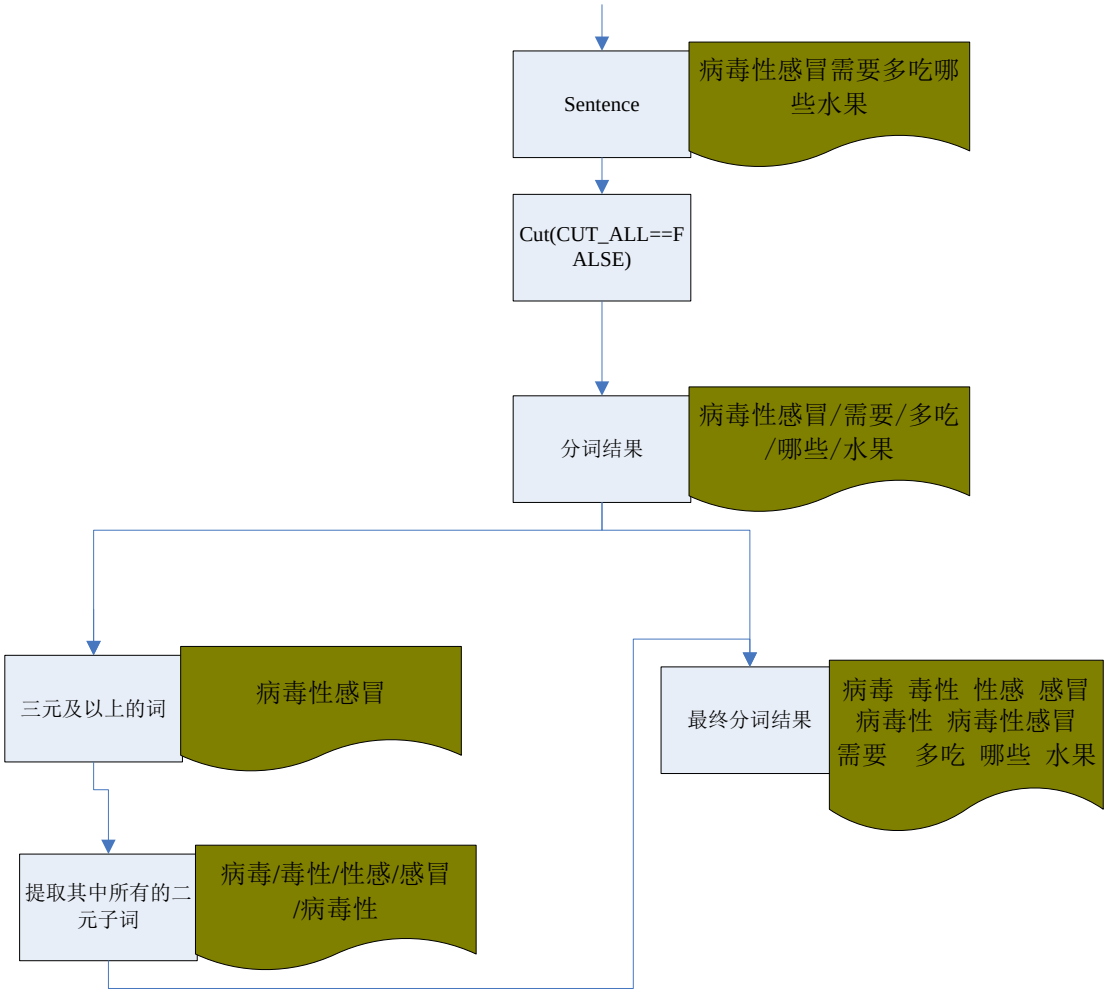


图 4.4 jiaba 分词搜索引擎模式下的执行原理

jieba 分词后我们需要对语料进行编号。因为语料中语句长度不一，所以采用长句裁减短句补零的方式使其统一化（采用 pad_sequences），使得其语句规范，产生的数据如图 4. 5 所示：

```
array([[   2,    1,   74, ..., 4471,  864,    4],
       [   0,    0,    0, ...,    9,   52,    6],
       [   0,    0,    0, ...,    1, 3154,    6],
       ...,
       [   0,    0,    0, ..., 2840,    1, 2240],
       [   0,    0,    0, ...,   19,   44,  196],
       [   0,    0,    0, ...,  533,   42,    6]], dtype=int32)
```

图 4.5 统一化的词向量图示

4.2.2 Bi-LSTM-CRF 分词

Bi-LSTM-CRF[31]是一种深度学习模型，它是一种特殊的循环神经网络模型，LSTM 有遗忘门，输入们，输出门三种来控制 cell 的状态。BiLSTM 会融合句子正序以及逆序两种方向的 LSTM 层，使得当前词能够获取到过去以及未来的信息。接着会把所获取到的信息拼接成 2 个 LSTM 隐含层变量大小并输出到 CRF 层中。因为 Bi-LSTM 虽然学习到了上下文的信息，并最大概率的序列，但是在预测过程中考虑序列彼此间的影响方面还有所欠缺，所以引入 CRF 很好的弥补了这方面的问题。

对于每一个输入 X，我们都会得到一个预测的序列假设为 Y（$y_1, y_2, \ldots y_n$），模型会通过序列计算最优标注序列。公式如下：

$$P(X \mid y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \qquad (4\text{-}1)$$

其中 A 为转移概率矩阵[34]，P 为 Bi-LSTM 的预测结果，在使用的时候可以使用 viterbi 算法动态规划对矩阵进行解码，获取 tag 序列 y*。

$$y^* = \underset{\hat{y} \in Y_x}{\operatorname{argmax}}\, s(X, \hat{y}) \qquad (4\text{-}2)$$

4.2.3 Bi-LSTM-CRF 与 jieba 分词的对比分析

在使用了一些句子进行分词后对比后，得到结果如表 4.1 所示

表 4.1 测试分词对比表

测试用句	Bi-LSTM-CRF	jieba
中暑需要吃西瓜	中暑/需要/吃/西瓜	中暑/需要/吃/西瓜
中暑需要吃大西瓜	中暑/需要/吃/大/西瓜	中暑/需要/吃/大西瓜
老人家中有感冒药	老人家/中/有/感冒药	老人家/中/有/感冒药
天真的你	天真/的/你	天/真的/你
…	…	…

在进行了大量的中文分词实验后，得到 Bi-LSTM-CRF 在某些情况下的分词效果要略优于 jieba 分词，但是在新词的处理上，jieba 分词的处理能力（通过 HMM）要强于 Bi-LSTM-CRF，而且一些专业的名词在 Bi-LSTM-CRF 中如未收录也很难处理，所以本文选择的是 jieba 分词。

4.3 Word2vec

4.3.1 词向量

对于分词标注并去除词性标注以及停用词后获取到的候选实体集，本文使用 Word2vec 模型[32]来比较词向量。对于 CBOW 算法，如果是两层结构的话，输入层传入的不再是负责的交叉权值的计算，而是获取到上下文的词向量，然后进行 ave 求值。如果是三层的话输入层传入的将是 one-hot 向量，隐藏层是为了将输入层的词向量进行累加与处理，输出层是一颗哈夫曼树，其每次进行节点的分裂都是二分类的过程，其结构如图 4.6 所示：

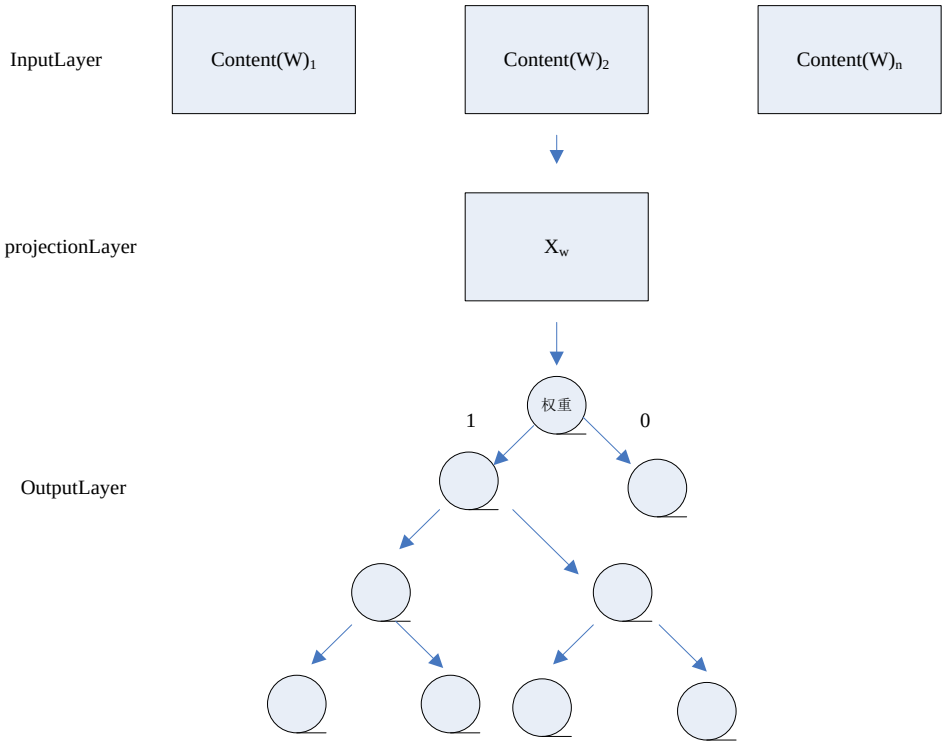


图 4.6 CBOW 算法结构图

CBOW 算法并不注重上下文的词语距离，他通过输入语句周围一定步数的的词向量获得相关词的词向量，其值可作为上下文中目标词通过神经网络训练得到的最大平均概率。其公式为：

$$V(W(n))=\frac{1}{N}\sum_{n-1}^{N}\Big(\sum_{\substack{-k\le j,\\ j\le k,\\ j\neq 0}}\big(\log\big(p\big(W(n+j)\,|\,W(n)\big)\big)\big)\Big) \qquad (4\text{-}3)$$

其中包含 k 个词，V 是用来描述单词向量 W(N)(n 是单词向量所在的地方)的描述值。N 用于计数上下文相关词的出现频率（开始时上下文相关词具有相同的权重），p 为先验概率，j 为相对位置。

Skip-gram 是通过输入当前的相关词来预测向下文的改路。输入是一个随机词向量，它的训练思想和 CBOW 算法类似，训练过程如图 4.7 所示：

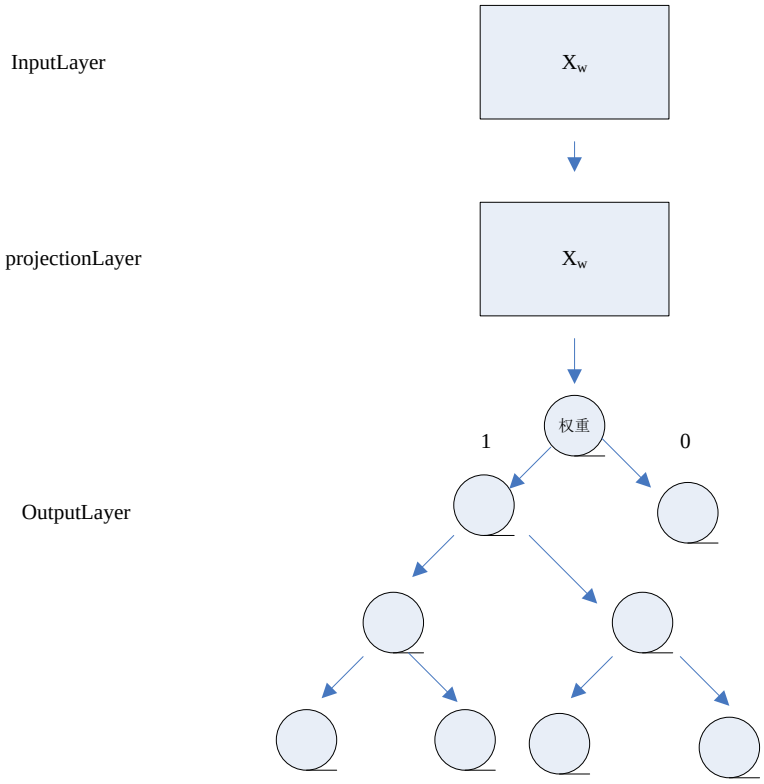


图 4.7 Skip-gram 算法结构图

经过训练之后，该算法 RDF 并获得了分词的最优向量。

词向量矩阵如图 4.8 所示：

[-0.7688807 -0.25586388 0.1608192 -0.8175718 -0.49182913 0.3053461 -0.8933297
0.5435378 0.15263264 -0.6505136 0.58177656 0.2610467 0.60386753 -0.37077782
-0.121123776 -0.63709927 -0.084013075 -0.32030246 0.37618414 1.0853177 -0.23331262
-0.1330202 0.19853982 -0.26166525 0.21209513 -0.43921936 -0.13753673 -0.28911144
-0.07992798 -0.11276782 0.8588074 -0.16840978 -0.066087775 -1.4498862 -0.116305925
-0.50329906 -0.25005418 0.85132086 -0.7688807 -0.25586388 0.1608192 -0.8175718
-0.49182913 0.3053461 -0.8933297 0.5435378 0.64792275 0.46061665 -0.12162254]

图 4.8 词向量矩阵示例图

4.3.2 相似度比较

如果问句中的实体在字典树中未查询到，则需要对候选实体进行相似度比较，如果相似度大于 0.8 则输出此候选实体，不然则抛弃返回 NO。

具体的代码如下：

```
# -*- coding: utf-8 -*-
```

```
import warnings
warnings.filterwarnings(action='ignore',category=UserWarning,module='gensim')
from gensim.models import word2vec
import logging
    # 主程序
logging.basicConfig(format='%(asctime)s:%(levelname)s:%(message)s',level=loggin
g.INFO)
sentences = word2vec.Text8Corpus(u"D:\wiki\语料.txt")  # 加载语料
n_dim=300
# 训练 skip-gram 模型;
model = word2vec.Word2Vec(sentences, size=n_dim, min_count=5,sg=1)
# 计算两个词的相似度/相关程度
y1 = model.similarity(u"西红柿", u"番茄")
```

4.4 预分类

4.4.1 朴素贝叶斯分类

智能问答进行语义检索时，响应时间有很大一部分是在执行 CNN。我们可以此之前对所获取到的语料进行预分类，这样做可以减小对于处理器的资源占用，同时也可以在一定层度上提高模型的精确度。对于这种小规模文本的分类，朴素贝叶斯效果上佳。

朴素贝叶斯[35]的算法逻辑如图 4.9 所示：

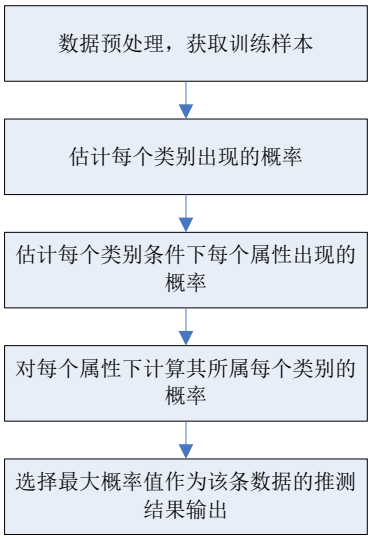


图 4.9 朴素贝叶斯算法逻辑图

1) 获取输入集 y=(Y1,Y2,Y3...Yn)，其中 Y1,Y2,Y3...Yn 为 y 所对应的 n 个对应的特征属性

2) 获取标注集合 z=(Z1,Z2,Z3...Zm)，共 m 个标注。

3) 计算 y 在各标注中的条件概率。

$$P(Z_m \mid y) = \frac{P(y \mid Z_m)P(Z_m)}{P(y)} = \frac{P(Zm)}{P(y)} \prod_{i=1}^{n} P(Y_i \mid Z_m) \qquad (4\text{-}4)$$

4) 选择最大概率值输出

P(Z|y)：具有特征向量 y 样本属于 Z 类别的概率，其最大概率即是计算目标。

P(y|Z)：指的是训练中的数据，指的是 Z 类别中有 y 向量出现的概率。

P(Z)：指的是训练样本中 Z 出现频率

P(y)：指的是 y 特征属性出现的概率

朴素贝叶斯常用的有三种模型，其中高斯模型需假设输入集的特征属性连续且符合正态分布，多项式分布模型适合离散的类型，贝努利模型适合二项分布特征的类型。

在这选择使用多项式模型，主要通过以下方式来实现：

1. 前置步骤是分词和标注，这部分在 4.2 中 jieba 方法已经进行了详细的描述，此处不做展开。但是很多时候一个词语出现的多少并不能明确的指定这个词语所占有的比重，比如词语“啊”、“了”等。对此我们需要对分词进行评估，

测评其对于文章的关键程度。这些对于结果造成不了任何影响的词汇，需要对其进行去停用词操作。

2. 其次要统计词汇在文章中出现的概率即占比。记 X 是要统计词汇的占比，记 Y1,Y2,Y3...Yn 为要输入的词汇，则输入集合 y=(Y1,Y2,Y3...Yn), 记 Z1,Z2,Z3...Zm 为文本处理后的标注，则标注集合 z=(Z1,Z2,Z3...Zm)。

3. 接着计算关键词出现在分类文本中的概率（（目标记录在分类文本的次数+目标同时记录在所有文本中的次数）/总文档记录次数）以及关键词出现在总文本中的概率（目标记录的次数/总文本数）。

4. 最后根据出现在分类文本中的概率以及同时出现在所有文本中的概率通过朴素贝叶斯算法求出分类的概率。

通过以上步骤的预分类，减少了问答的响应时间，系统的可用性将会得到增加。

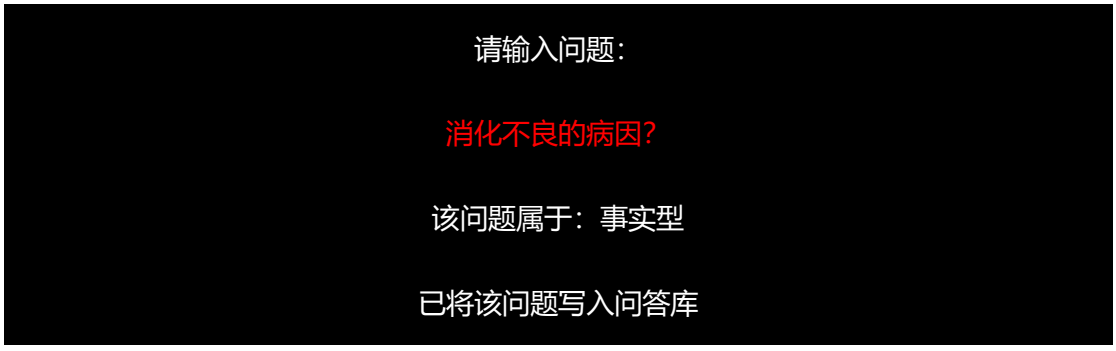
其核心代码如图 4.10 所示：

```
getKeyWord();//获取关键词
readData();//从文件中读取出分类的数据
getCountKeyWordClass();//统计关键词出现在分类文本中的次数
getChanceKeyWord();//计算关键词出现在分类文本中的概率
getCountKeyWordAllText();//统计关键词出现在总文本中的次数
getChanceKeyWordSameTime();//统计关键词同时出现在所有文本中的概率
classByBayes();//使用朴素贝叶斯算法求出分类的概率
```

图 4.10 朴素贝叶斯核心代码图

其分类测试效果如表 4.2 所示：

表 4.2 预分类测试效果图示



4.4.2 SVM 与 LR 分类

支持向量机 SVM 和逻辑回归 LR 都是应用于分类的监督学习算法，其本质都是为了获得一个最优解，但两者在处理逻辑等方面还是有一定区别的，具体如表 4.3 所示：

表 4.3 训练模型的对比表

SVM	LR
基于距离分类	基于概率分类
少数样本参与核函数计算	每个样本点都参与分类决策
适用小规模数据集	适用海量数据
考虑分类边界线附近的样本	受所有数据点的影响

4.4.3 预分类实验结果

朴素贝叶斯进行预分类最终用自带的模型测试的准确率与召回率如图 4.11 所示：

```
2020-03-28 17:48:34.698520+0800 Enn[64139:44481843] 5s 3ms/step - loss:0.2604 - acc:0.8943 -val_loss:0.6034
    -val_acc:0.8034---8
2020-03-28 17:48:34.698665+0800 Enn[64139:44481843] 5s 3ms/step - loss:0.2542 - acc:0.9055 -val_loss:0.5024
    -val_acc:0.8057---9
2020-03-28 17:48:34.698784+0800 Enn[64139:44481843] 5s 3ms/step - loss:0.2135 - acc:0.9223 -val_loss:0.4833
    -val_acc:0.8074---10
```

图 4.11 预分类测试参数图示

当获取到重要特征后，将该特征分别放入到朴素贝叶斯，SVM 以及 LR 中进行训练，从而挑选出最优的结果作为输出。

将特征放入分类算法中训练后得到的各项指标如表 4.4 所示：

表 4.4 算法实验指标对比表

指标 / 分类算法	precision	recall	F1-score
朴素贝叶斯	92.1%	67.0%	77.6%
SVM	90.3%	62.5%	73.9%
LR	89.9%	66.5%	76.5%

所以本文选择朴素贝叶斯进行预分类。

造成朴素贝叶斯模型准确率偏差的原因有：

1. 在很多时候，我们分类的结果并不能只考虑某一个词汇，而应该考虑某些词汇的组合。以后可以考虑进行传入词汇组合的训练。

2. 朴素贝叶斯算法是建立在其相关的特性相互独立的情况下，但是往往有一部分的特征时相互之间有所关联的，所以当特征的关联性较高的时候所得到的分类效果不佳。

4.5 CNN 分类

通过 4.1 实体识别获取到准确的食疗信息的实体后，还需要理解自然问句中用户的用意，他的意图主要的表现形式为食疗信息的关系或者属性，即需要获取到自然问句中的用意到语料库中关系的映射。本文使用 CNN[34]来训练词向量。

通过 4.1 实体识别中的词性标注可以获取到带关系标签的标注词汇集，其可以作为 CNN 分类器分类任务训练所需要的数据集。这些词汇集拥有单标记以及多标记两种形式。标签中涵盖了实体关系以及属性，如病-病症，食物-特性等。

CNN 神经网络和全连接神经网络类似，但是 CNN 相邻层之间并不是所有的节点都连接，其擅长提取局部特征，所以对短文本数据表现出了良好的识别能力。

CNN 模型包括四个层次，分别为

1. 数据输入层：上述中的训练所需要的数据集即若干个数据矩阵。

2. 卷积层：卷积层中可以进行多次卷积，通常会选择一个 3*3 或者 5*5 的神经元进行特征映射，每卷积一次矩阵的深度就会变深。卷积层主要用于特征提

取。矩阵的计算公式如式 *4-7*：

$$S(i,j)=(X,Y)*(i,j)=\sum_{m}\sum_{n}X(i+m,j+n)y(m,n) \qquad (4\text{-}5)$$

3. 池化层：池化层主要是用来去除没必要的一些数据，本文在池化层中使用了最大值池化即 Max-pool，在保证了减少模型参数的同时，又给卷积层的不定长度的输出给与了相同长度的输入。

4. 全连接层：全连接层+softmax 是对上面经过卷积池化的特征向量进行分类，然后比较不同类别的概率进行输出。

CNN 的算法结构如图 4.12 示：

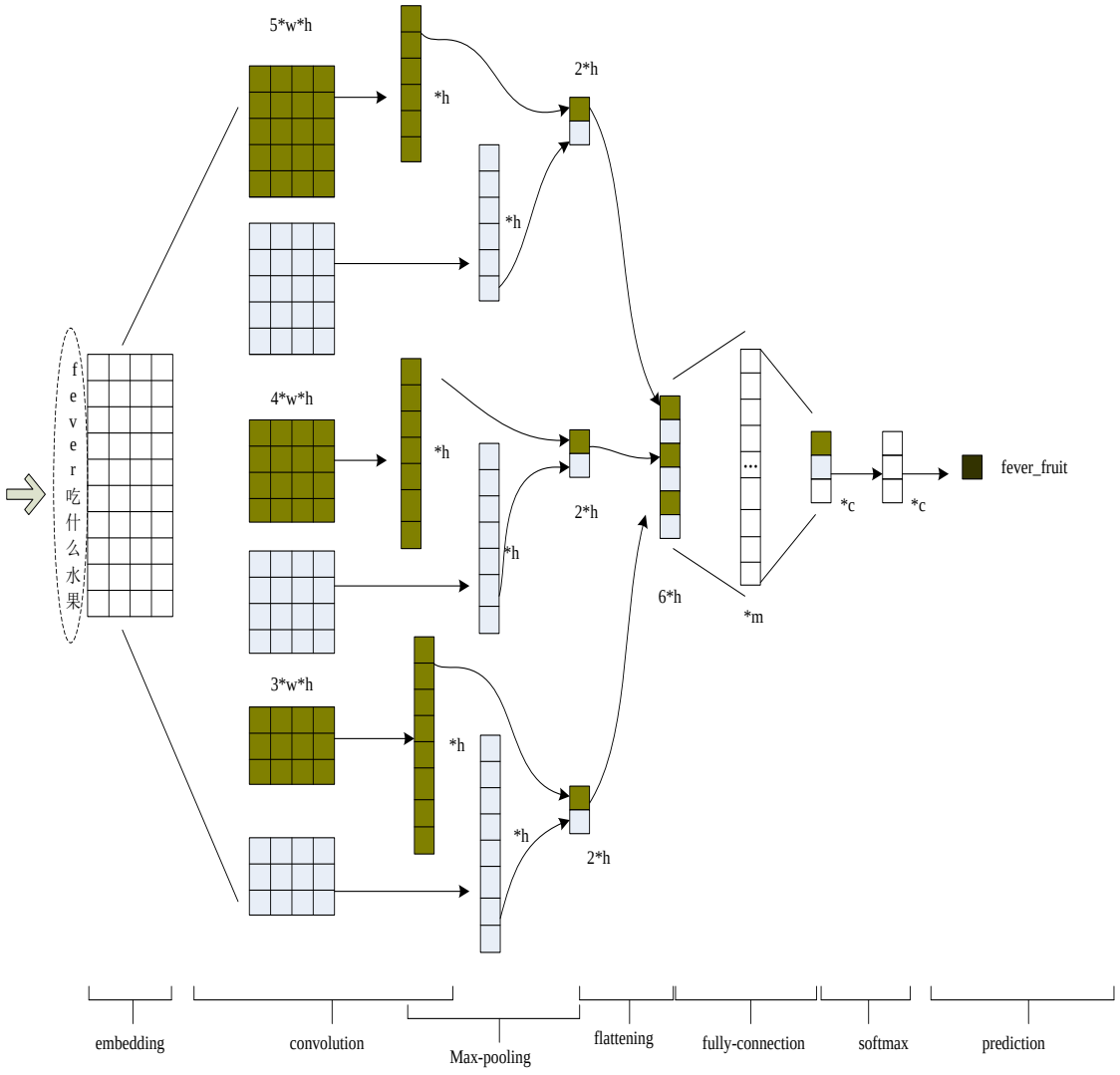


图 4.12 基于卷积神经网络的问句关系/属性映射模型示意图

如上图所示，首先我们会把自然问句中获取到的实体转换为他的对应的类别

标注（发烧-fever），然后作为输入加入到 CNN 的分类模型里面，接着经过 CNN 模型的四个层次的处理，最终得到自然问句中关系与属性的映射。

其中 Embedding（嵌入表示层）也指的是语句矩阵，在这里分别采用 char-CNN、Word-CNN、Char-Word-CNN 进行训练。我们需要设定句子的最大长度和向量维度，如果不符合长度就进行多则截取少则补零的方式处理，且需要设置不同维度来区分语句中的特征。

Convolution：因为文本是一维的，所以我们需要使用一维卷积，在此基础上，我们需要设计不同高度（一般为 2-8）的卷积核（其实质为一个和词向量同宽度的向量），通过不同的高度，我们可以获取到不同的向量特征。

Max-Pooling：因为定义不同高度的卷积核所得到的特征大小也是不尽相同的，所以我们要对每个特征进行池化使其统一。Max-Pooling 是通过选取特征的最大值做采样，也就是捕获其的显著特征。

Flattening：这一层是指获取到经过卷积等一系列训练后得到的输出特征向量后，将之平铺获得一个整体的特征向量。最后通过 softmax 全连接层处理以后实现食疗信息关系/属性的多分类任务。

对比卷积与池化如图 4.13 所示：

```
初始化矩阵:
[[0.0, 0.0, 0.0, 0.84003717, 0.0, 0.0, 0.473217, 1.000001],
[0.90521437, 0.86672086, 0.6687838, 0.0, 1.0000002, 0.929654, 0.06073203, 0.0],
[0.7697354, 0.7243773, 0.4113286, 0.0, 7334271, 0.8083423, 0.479932, 0.0],
[0.8285432, 0.7113442, 0.5210232, 0.0, 0.7642221, 0.8134234, 0.042421, 0.114231],
[0.8142343, 1.0, 0.6348756, 0.0, 0.5632134, 0.9034212, 0.052423, 0.0],
[0.6632423, 0.6432453, 1.0, 0.0, 0.6234658, 0.7043251, 0.0, 0.0]]
卷积一次:
[[0.4429844, 0.3838769, 0.3772052, 0.4632423, 0.4823452, 0.3693842, 0.382342],
[0.827933, 0.3842456, 0.3788974, 0.4698782, 0.8732421, 0.4692891, 0.0258932],
[0.754332, 0.7145875, 0.3194823, 0.4268542, 0.7832452, 0.5398573, 0.1598932],
[0.829945, 0.6184588, 0.2733142, 0.3786876, 0.8492742, 0.5792834, 0.1703928],
[0.902345, 0.7856442, 0.4198424, 0.3288553, 0.8942187, 0.4987683, 0.0482294],
[0.783553, 0.8295358, 0.3598777, 0.3819421, 0.7895432, 0.4187283, 0.0134532]]
池化一次:
[[0.824983, 0.7199092, 0.4699982, 0.8779129, 0.8779129, 0.4621913],
[0.827532, 0.7209421, 0.4487666, 0.8739293, 0.8739293, 0.5298384],
[0.829435, 0.6509981, 0.4085731, 0.8492893, 0.8492893, 0.5799895],
[0.902342, 0.7893284, 0.7809007, 0.8945986, 0.8945986, 0.5793829],
```

图 4.13 卷积池化对比图

CNN 会出现过拟合现象，在此我们使用了 L1 正则化来尝试，发现未对模型性能的影响造成影响，所以使用：

1. 标准差的方式来进行处理，当拟合度达到一定值时，将不再进行卷积池化而直接进入到 softmax 全连接层处理。

2. 采用 RELU 算法，在进行 CNN 训练执行时使用 RELU 来剔除不必要的特征。

该函数公式为式 4-8 所示：

$$F(X)=\begin{cases} X=0, X<0 \\ X=X, X>0 \end{cases} \quad (4\text{-}6)$$

图 4.14 为防止过拟合效果展示：

```
[[-0.3552867, -0.3699886, -0.69738835, 0.84003717, -0.44
[[0.0, 0.0, 0.0, 0.84003717, 0.0, 0.0, 0.473217, 1.00000
[[0.4429838, 0.38387614, 0.37720525, 0.46000934, 0.48242
[[0.827977, 0.7101677, 0.46000934, 0.8779129, 0.8779129,
标准差0.22256148
句子相似度：0.72090775
[[0.76907235, 0.5803425, 0.6642151, 0.8779129, 0.6882453
[[0.76907235, 0.6642151, 0.8779129, 0.8779129], [0.79432
标准差0.09059572
句子相似度：0.8080365
```

图 4.14 防过拟合效果展示图

其分类测试效果如表 4.5 所示：

表 4.5 分类测试表

输入问题：	输出答案
吃哪些水果可以缓解消化不良	菠萝，苹果
消化不良应该多吃哪些水果型	菠萝，苹果

4.6 实验结果与分析

4.6.1 实验数据集

实验数据来源于一起食疗网中获取到的食疗养生列表 30088 条以及医疗信息科普中心的有问必答集合前 18000 条，通过实体以及特征随机生成补全 50000 条，训练集和测试集按八二比例进行实验。

4.6.2 实体识别实验结果与分析

实验结果如表 4.6 所示：

表 4.6 实体识别实验结果表

实体识别类别	实体识别准确率
水果实体	92.3%
疾病实体	89.9%
蔬菜实体	91.4%

造成实体识别的结果有所偏差的原因有：

1.部分实体名称存在别名（例：桂圆别名龙眼，番茄别名西红柿等）

2.部分实体相同的词汇在不同的场景具有不同的意思。（例如：苹果可以指代水果也可以指代手机。）

3.部分实体输入时缺少或者错别字等都有可能造成实体识别出现偏差。

4.6.3 CNN 分类实验结果与分析

在 CNN 的关系属性映射的实验中，通过在 Embedding 这里进行训练所完成的分类任务来看，CNN 对于短文本的自然问句具有比较好的分类效果，且先寻找到最佳的卷积核以后，取该值尺寸附近的合适值来混合使用往往效果更佳。对于较长的文本因为卷积核限制其相关特征的获取而效果不佳。

其实验结果（关系属性映射的准确率）如表 4.7 所示：

表 4.7 CNN 分类实验结果表

问题类型	卷积核为 4	卷积核为 3,4,5 混合
食疗信息关系类别（未预分类）	90.5%	91.1%
食疗信息属性类别（未预分类）	91.3%	91.9%
食疗信息关系类别（预分类）	91.2%	92.4%
食疗信息属性类别（预分类）	91.8%	92.3%

造成模型准确率偏差的原因有：

1.数据不够多，需要添加更多的数据来降低过拟合 。

2.选用 Max-pooling 来进行池化虽然能减少模型参数的数量，使得模型的过拟合问题降低，但是最大特征值多次出现时，有部分强特征将会丢失导致模型的精确度降低。

4.7 本章小结

本章首先通过比较 jieba 分词和 Bi-LSTM-CRF 的分词性能的对比，选择了 jieba 分词来进行分词标注的工作，然后通过 CNN 卷积神经网络来对数据进行分类训练。其中实验表明卷积核的混合使用能提升分类算法 0.6%的准确率，朴素贝叶斯的预分类也能够提升 CNN 分类算法的准确率。

第五章 系统整体实现与测试

前文主要对使用的方法模型以及算法进行了详细的阐述，这一章主要针对系统的开发环境、系统的具体实现以及系统测试三方面进行阐述。

5.1 系统开发环境与结构介绍

5.1.1 系统开发环境

本文主要使用了 Python 和 Java 语言来进行此食疗问答系统，其中 Python 在智能问答和深度学习这一块领域具有相当大的优势，我们可以很轻松地引用一些经典算法，减少代码的冗余，Java 我主要用来构建知识图谱和相关数据的处理。下图是我进行系统构建时使用的实验环境。

表 5.1 实验环境展示表

System	macOS 10.13.6
Development language	Python 3.7，Java
Deep Learning Platform	Tensorflow frame
Memory	8G
Hard disk	256G
DataBase	Neo4J，MySQL

系统部署流程如表 5.2 所示，先安装语言环境和平台，然后倒入相关的学习模型，再部署相关的 JDK 开发环境，最后部署相关服务。

表 5.2 系统部署流程表

安装 Python 3.7，Java 语言环境
安装 Tensorflow 框架
安装 Neo4J，MySQL 数据库和相关可视化插件
倒入训练好的深度学习模型
Hard 启动提供深度学习 API 的 Python 程序

启动相关服务

5.1.2 系统架构

针对上面所述，本文基于 MVC 设计并实现了一个基于知识图谱的有关食疗健康的问答系统。其中数据层负责数据存储与处理，逻辑层负责算法与思维处理，展示层负责页面的整体展示。

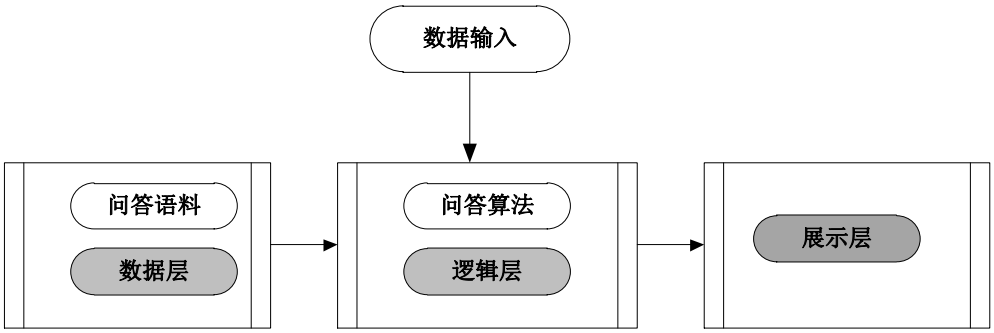


图 5.1 展示层结构图

如果按照系统的整体功能构建，也可以细分为数据存储与处理层、食疗知识图谱构建层、知识图谱问答层、应用展示层。

1.数据存储与处理层：该层主要是通过网络爬虫、OCR，NLP 工具以及自定义脚本的方式对结构化、半结构化数据进行知识抽取，存储有关食疗健康方面的相关信息，并对数据进行预处理等工作，为下面的知识图谱的构建提供基础数据支撑。

2.食疗知识图谱构建层：其具体的构建如第三章的研究内容所示，首先，通过基于食疗健康领域术语以及属性规则的人工抽取以及基于实体识别和关系抽取的自动抽取的方式实现知识获取，然后确认知识库中对应实体对象，连接对应实体并进行相似度计算。其次是特征抽取（1.OCR，NLP 工具等 2.用户自定义脚本）、专业知识融合、监督学习以及迭代优化，以此来进行知识融合。关系类知识存储到 Neo4j 知识中，属性类知识通过键值对存入 MySQL 中，该层为上层的问答层提供数据支撑。

3.知识图谱问答层：首先，获取自然文具以后通过构造数据自动转换以及标注工具等生成知识图谱问答任务训练集，然后通过训练问答数据来进行数据增强。其次，通过分词标注以及分类算法完成自然问句的语义解析，然后通过规则器将问句语义逻辑转化为查询逻辑，然后在对应的知识库中完成知识的检索，该层为上层的应用展示层提供了核心业务能力。

4.应用展示层：根据输入的自然问句，在问答层的处理下，从知识库中获取到准确的答案呈现在前端。

5.2 系统实现与核心结构

经过对知识图谱的构建以及智能问答的设计进行了详细的设计，其中图 5.2 展示的是相关的代码结构图：

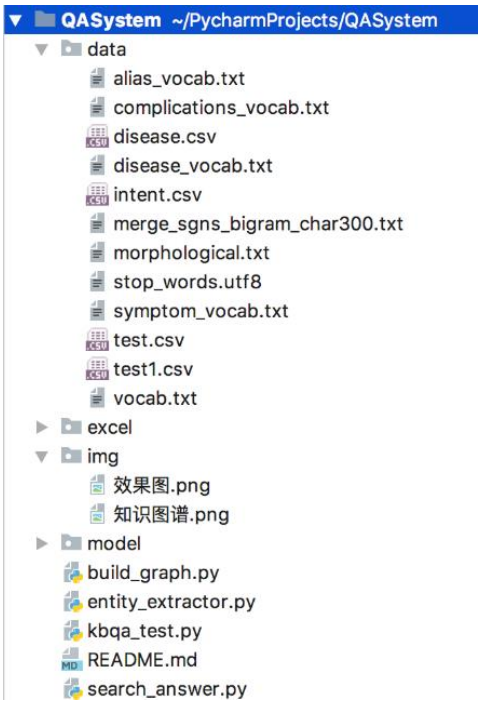


图 5.2 相关代码展示图

实体类之间的 ER 图已经在知识存储中标明，系统主要实现类图如图 5.3 所示：

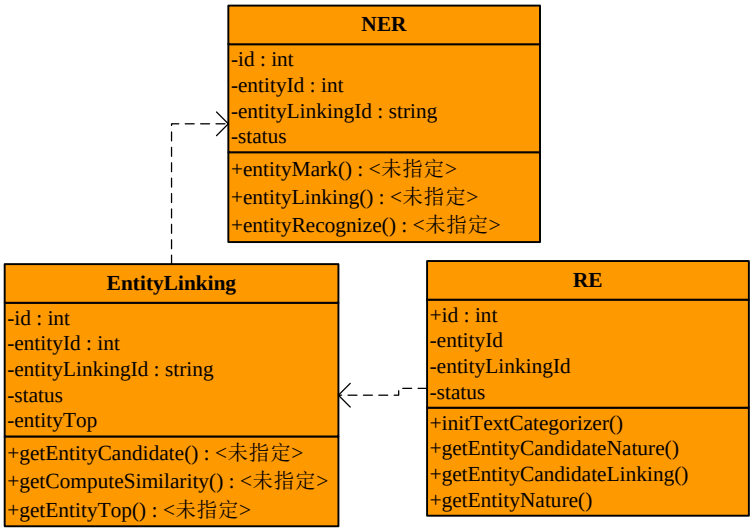


图 5.3 核心类图

其中实体链接类 Entity-Linking 依赖于实体识别类 NER，关系抽取类 RE 依赖于 Entity-Linking。

主要包括的方法如表 5.3 所示

表 5.3 核心类方法注释表

NER 方法名	解释
entityMark()	实体标注
entityRecognize()	实体识别
entityLinking	实体链接
Entity-Linking 方法名	**解释**
getEntityCandidate()	获取候选实体
entityComputeSimilarity()	相似度计算
getEntityTop	通过相似度获取 top1 的实体
RE 方法名	**解释**
initTextCategorizer()	构建文本分类器
getEntityCandidateNature()	锁定实体获取属性作为候选属性

getEntityCandidateLinking()	锁定实体获取关系作为候选属性
getEntityNature()	通过分类获取实体属性

NER 的部分核心代码如图 5.4 所示：

```
action = config.FLAGS.action
# 获取词的总数。
vocab_size = get_src_vocab_size()
src_unknown_id = tgt_unknown_id = vocab_size
src_padding = vocab_size + 1

src_vocab_table, tgt_vocab_table = create_vocab_tables(src_vocab_file, tgt_vocab_file, src_unknown_id,
                                                       tgt_unknown_id)
embedding = load_word2vec_embedding(vocab_size)

if action == 'train':
    iterator = get_iterator(src_vocab_table, tgt_vocab_table, vocab_size, BATCH_SIZE)
elif action == 'predict':
    BATCH_SIZE = 1
    DROPOUT_RATE = 1.0
    iterator = get_predict_iterator(src_vocab_table, vocab_size, BATCH_SIZE)
else:
    print 'Only support train and predict actions.'
    exit(0)

tag_table = tag_to_id_table()
net = NER_net("ner", iterator, embedding, BATCH_SIZE)
with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    sess.run(iterator.initializer)
    tf.tables_initializer().run()

    if action == 'train':
        train(net, iterator, sess)
    elif action == 'predict':

        predict(net, tag_table, sess)
```

图 5.4 NERServer 类中的部分核心代码

获取 NER Result 的部分核心代码如图 5.5 所示：

```
Map<Integer, List<Element>> elementIndexMap = new TreeMap<>(
        new Comparator<Integer>() {
            @Override
            public int compare(Integer obj1, Integer obj2) { return obj1 - obj2; }
        });
Map<Long, Element> entityMap = new LinkedHashMap<>();
Map<Long, Element> conceptMap = new HashMap<>();
Map<Integer, Element> attributeMap = new HashMap<>();
List<Element> attributeMetaList = new ArrayList<>();
Map<String, String> askContentMap = new LinkedHashMap<>();
Map<Long, Long> entityConceptMap = new HashMap<>();
Map<String, JSONObject> skillConfigMap = new HashMap<>();
NERResultBean nerResultBean = new NERResultBean(input, elementIndexMap, entityMap, entityConceptMap, conceptMap, attribu
String originalInput = input;

Map<Long, Double> entityScoreMap = new HashMap<>();
Map<Long, String> entityWordMap = new HashMap<>();
boolean isPath = false;
for (String key : skillWakeUpWordMap.keySet()) {
    if (input.contains(key)) {
        Intent intent = skillWakeUpWordMap.get(key);
        skillConfigMap.put(intent.getExpression(), new JSONObject());
        if (Intent.SEMANTICSEARCH.equals(intent)) {
            skillConfigMap.get(Intent.SEMANTICSEARCH.getExpression()).put("query", input.replace(key, ""));
        }
        if (Intent.PATH.equals(intent)) {
            isPath = true;
        }
        int index = input.indexOf(key);
        Element skillElement = new SkillElement(index, 10, -1L, null, null, null, 0.0, key, null, intent.getExpression()
        if (!elementIndexMap.containsKey(index)) {
            elementIndexMap.put(index, new ArrayList<>());
        }
        elementIndexMap.get(index).add(skillElement);
        String replace = "";
        for (int i = 0; i < key.length(); i++) {
            replace += "_";
        }
        input = input.replace(key, replace);
    }
}
```

图 5.5 NERServerImpl 类中的部分核心代码

其中对应语料库进行语义分析的部分核心代码如图 5.6 所示：

```
List<SemanticSegWord> wordList = NERUtil.recognizeAll(kgName, input, 1, true);
nerResultBean.setWordList(wordList);
Set<Long> entityIdSet = new HashSet<>();
int pos = 0;
if(input.indexOf("近期")!= -1){
    pos = input.indexOf("近期");
    SemanticSegWord specialWord = new SemanticSegWord();
    specialWord.setWord("近期");
    specialWord.setType(71);
    specialWord.setNormalValue("2019-01-01 00:00:00");
    specialWord.setPos(pos);
    wordList.add(specialWord);
}
for (SemanticSegWord word : wordList) {
    logger.info(JSONObject.toJSON(word));
    input = input.replaceFirst("{71}"+word.getWord(), "{}");
    if (!elementIndexMap.containsKey(word.getPos())) {
        elementIndexMap.put(word.getPos(), new ArrayList<>());
    }
    if (word.getType() == 0) {
        Element conceptElement = new Element(word.getPos(), 0, conceptProjectionMap.get(word.getConceptIdList().get(0)), null, w
        conceptMap.put(word.getConceptIdList().get(0), conceptElement);
        elementIndexMap.get(word.getPos()).add(conceptElement);
    } else if (word.getType() == 1) {
        List<Long> entityIdList = word.getEntityIdList();
        entityIdSet.addAll(entityIdList);
        for (int i = 0; i < entityIdList.size(); i++) {
            entityIndexMap.put(entityIdList.get(i), word.getPos());
            entityScoreMap.put(entityIdList.get(i), word.getEntityScoreList().get(i));
        }
        Set<Long> toDisambiguateIdSet = new HashSet<>();
        if (entityIdList.size() > 1) {
            toDisambiguateIdSet.addAll(entityIdList);
            toDisambiguateWordMap.put(word.getWord(), toDisambiguateIdSet);
        } else {
            entityWordMap.put(entityIdList.get(0), word.getWord());
        }
    } else if (word.getType() == 3) {
        List<Integer> attrIdList = word.getAttributeIdList();
        Set<Integer> attrIdSet = new HashSet<>(attrIdList);
        Element attributeElement = null;
        for (Integer attrId : attrIdSet) {
            attributeElement = new Element(word.getPos(), 2, attrProjectionMap.get(attrId), null, null, attrId, 0.0, word.getWor
```

图 5.6 实体语义分析的部分核心代码

然后要对进行处理后的数据进行实体消歧，其核心代码如图 5.7 所示：

```
/**实体消歧***/
    private Map<String, Long> disambiguate(String kgName, Map<String, Set<Long>> toDisambiguateIdMap, Map<Long, String> entityMap, Map<
        List<Long> allowAttList = new ArrayList<>();
        Map<Integer, Integer> attrTypeMap = new HashMap<>();
        Map<String, Long> resultMap = new HashMap<>();
        int disambiguateSwitch = 0;
        int disambiguateConcept = 0;
        Map<Long, Map<Integer, Integer>> disambiguateAttrMap = null;
        if (config != null) {
            disambiguateSwitch = config.getDisambiguateSwitch();
            disambiguateConcept = config.getDisambiguateConcept();
            disambiguateAttrMap = config.getDisambiguateAttribute();
        }
        if (disambiguateSwitch == 1) {
            if (disambiguateConcept == 1) {
                Set<Long> toRemoveConceptSet = new HashSet<>();
                for (String word : toDisambiguateIdMap.keySet()) {
                    Set<Long> toDisambiguateIds = toDisambiguateIdMap.get(word);
                    Set<Long> toRetainIdSet = new HashSet<>();
                    for (Long entityId : toDisambiguateIds) {
                        long conceptId = entityConceptMap.get(entityId);
                        if (conceptMap.containsKey(conceptId)) {
                            toRetainIdSet.add(entityId);
                            toRemoveConceptSet.add(conceptId);
                        }
                    }
                    if (toRetainIdSet.size() != 0) {
                        toDisambiguateIds.clear();
                        toDisambiguateIds.addAll(toRetainIdSet);
                    }
                }
                for (Long conceptId : toRemoveConceptSet) {
                    Element conceptElement = conceptMap.get(conceptId);
                    elementIndexMap.get(conceptElement.getIndex()).remove(conceptElement);
                    conceptMap.remove(conceptId);
                }
            }

            if (disambiguateAttrMap != null && disambiguateAttrMap.size() > 0) {
                for (String word : toDisambiguateIdMap.keySet()) {
                    Set<Long> toRetainIdSet = new HashSet<>();
                    Set<Long> toDisambiguateIds = toDisambiguateIdMap.get(word);
```

图 5.7 实体消歧的部分核心代码

Robot 实体类中所包含的主要属性如图 5.8 所示：

```
@ApiModelProperty(notes = "未知说辞，可能有多个")
@FormParam("unknownWords")
private String unknownWords;

@ApiModelProperty(notes = "机器人系统编号，兼容云问接口")
@FormParam("sysNum")
private String sysNum;

@ApiModelProperty(notes = "机器人密码")
@FormParam("pwd")
private String pwd;

@ApiModelProperty(notes = "排列序号")
@FormParam("orderNo")
private Integer orderNo;

@ApiModelProperty(notes = "appKey")
@FormParam("appKey")
private String appKey;

@ApiModelProperty(notes = "是否私有，1私有 0公有 公有的机器人只能使用，不能编辑")
private Integer isPrivate;

private String isQuote;

@ApiModelProperty(notes = "是否为多意图机器人")
@FormParam("multiIntent")
private Boolean multiIntent;

@ApiModelProperty(notes = "意图规则，多意图机器人专用,1模型分类,2优先级,3触发词")
@FormParam("intentRule")
private Integer intentRule;
@ApiModelProperty(notes = "分类模型id")
@FormParam("classifyModelId")
private String classifyModelId;
@ApiModelProperty(notes = "识别元素规则")
@FormParam("readRule")
private String readRule;

public String getIsQuote() { return isQuote; }

public void setIsQuote(String isQuote) { this.isQuote = isQuote; }
```

图 5.8 RobotNER 类的部分核心代码

当用户输入自然问句时，系统首先会分析用户问句中的意图，即从自然问句

中获取实体与关系，其主要是两个流程：实体识别和链接。其中使用 wec2Vec 去进行相似度模型和分类模型训练，通过语法分析和关键字提取，判断相对应的实体类型。

表 5.4 中主要记录了有关食疗以及有利病症的相关实体。

表 5.4 相关实体表

attribute_string 0.004 sec. 0 50

	_id	attr_id	entity_id	entity_type	attr_value
1	ObjectId(...	4	6	3	桃仁和杏仁各6克，生糯米10粒，上药共为末，用鸡蛋清调匀，外敷双脚心和...
2	ObjectId(...	6	11	3	红细胞生成减少、溶血、失血
3	ObjectId(...	5	11	3	头昏、耳鸣、失眠、面色苍白
4	ObjectId(...	4	11	3	可以通过补充缺乏的营养物质进行治疗，如缺铁性贫血补铁及治疗导致缺铁的...
5	ObjectId(...	6	9	3	内分泌功能障碍性疾病如脑垂体前叶功能减退症、性腺功能减退症、甲状腺功...
6	ObjectId(...	5	9	3	局部疾患往往发生疤痕而引起永久性秃发。药物引起的秃发常常是暂时性的，...
7	ObjectId(...	4	9	3	药物治疗可服维生素B2、维生素B6、和胱氨酸
8	ObjectId(...	6	6	3	由变应原等引起，环境、药物及生理因素促发
9	ObjectId(...	5	6	3	突发性喘息、气促、胸闷、咳嗽，多在夜间或凌晨发生
10	ObjectId(...	6	14	3	不良饮食习惯，包括刺激性食物（咖啡、浓茶、甜食、油腻、生冷等）和不良...
11	ObjectId(...	5	14	3	伴有失眠，焦虑，抑郁，头痛，注意力不集中
12	ObjectId(...	4	14	3	改善生活方式，养成规律的饮食习惯，并调整饮食结构，避免食用可能诱发症...

我们需要从目标问句中分析出关键信息，然后利用训练模型对候选实体进行属性处理，获取到属性向量后，来和知识图谱属性向量进行相似度比较，最后选择相似度最高的属性。

5.3 系统功能测试

针对各类问题系统所给出的相关回答，在这里开始演示效果：

表 5.5 定义型与事实型问答表

问句（定义型和事实型）	返回结果
不良消化的简介	图 5.9
不良消化的病因	图 5.9
消化不良的症状	图 5.10
消化不良的治疗方法	图 5.10

图 5.8 主要展示的是定义型以及事实型的问答，在经过朴素贝叶斯的预分

类后，查询回结果的速度有一定的提升。

图 5.9 食疗健康事实型问答结果展示

图 5.10 食疗健康事实型问答结果展示

表 5.8 中主要罗列了一些不同表达方式的自然问句，从获得的结果来看，系统是相当稳定的。

表 5.8 列表型问答表

问句（列表型）	返回结果
吃葡萄预防什么疾病	图 5.11
心脑血管疾病应该多吃什么	图 5.11
吃哪些水果可以缓解消化不良	图 5.12
吃菠萝可以缓解哪些疾病	图 5.12
消化不良应该多吃哪些水果	图 5.13
吃什么蔬菜可以抗癌症	图 5.13

图 5.11 食疗健康列表型问答结果展示

从图 5.12 中，输入问句“吃哪些水果可以缓解消化不良”和“消吃菠萝可以缓解哪些疾病问题”两者的答案相互包含可以从侧面反映该系统为用户的自然问句

提供一个准确可靠的答案。

图 5.12 食疗健康列表型问答结果展示

图 5.13 食疗健康列表型问答结果展示

从上图 5.12 和 5.13 中，输入问句“吃哪些水果可以缓解消化不良”和“消化不良应该多吃哪些水果”两者都表示是“化不良应该吃什么水果”，两种相同含义不同的表达方式，返回的都是一样的结果，对于不同的表达方式但是含义相近

的自然问句给出相同的答案。

5.4 本章小结

本章基于上面章节的研究结果，对其在实际生产环境中进行了应用。本章首先介绍了开发此应用所使用的开发环境以及系统的核心架构(包括实体链接，实体识别以及关系抽取等)，然后对此系统的功能测试进行了说明，最终可以看出此问答系统具有一定的语义识别能力以及比较不错的稳定性。

第六章 总结与展望

6.1 总结

本文主要内容是实现了食疗知识图谱的构建和智能问答系统的设计。

（1）本文构建知识图谱主要经过了知识抽取，知识融合以及知识存储。知识抽取主要包括基于本体的抽取（包括知识挖掘）以及基于模型的抽取，本文使用的是 KBC 系统，主要针对结构化和半结构化的数据的知识抽取操作，其流程是特征抽取、监督学习以及迭代优化。知识融合主要着手于实体向量相似度计算来进行实体消歧等。最后把数据存入到图数据库 Neo4j 和关系数据库中。

（2）对于智能问答的设计，在建立食疗知识图谱后，使用 jieba 进行分词以及词性标注的预处理，进而生成问题特征向量，然后通过分类训练学习反馈到用户更准确的答案。从实验结果中，可以为用户的自然问句提供一个准确可靠的答案，对于相近语义的问句也基本能够识别出准确的答案，此食疗健康问答系统定的语义识别能力以及比较不错的稳定性。

最后，本文通过上述内容的研究构建了食疗健康的知识图谱，并通过各算法的精确度对比提升了食疗健康问答的精确度，最终构建了一个满足用户需求的食疗健康问答系统。

6.2 展望

通过上述的文章的归纳总结，我们在更好地实现基于知识图谱与神经网络的食疗健康问答系统，可以做以下方面的改进：

1）基于 CNN 神经网络进行分类算法的时候，单独考虑某个词汇并不能很好的获取到它所具备的显著特征，以后可以考虑传入词汇组合进行关系抽取，提高问答系统的精确度。

2）知识图谱很不完整，对于很多的有关于食疗方面的问题都无法得到有效的解答，希望以后能拥有更为庞大的有关于此专业领域的知识图谱。

3）有关于知识加工中知识推理方面的实现还很欠缺，技术有限。

4）如果基于此文章基础上，实现自然问句的重新确认，在答案的精确度以及各个方面应该会有更好的进步。例：问：当我起床头疼感觉浑身发冷该起床做点

什么吃的更有利于身体健康？答：那请问你是否有咳嗽或者一些其它症状呢？通过反复的确认，可以给到用户更精确的有关食疗健康的答案，也可以让用户有更好的使用体验，为此我们可以扩展出更多的应用。

参考文献

[1] 李婧. 基于体质调理的食疗咨询系统设计研究[D].湖南:中南大学，2014,15-20.

[2] 王一鸣. 基于知识图谱的推荐技术研究及应用[D].杭州:电子科技大学,2018,32-52.

[3] 李涛, 王次臣, 李华康. 知识图谱的发展与构建[J]. 南京理工大学学报: 自然科学版, 2017, 41(1): 22-34.

[4] 杨笑然. 基于知识图谱的医疗专家系统[D].浙江:浙江大学大学,2018.40-55.

[5] 曾帅,王帅,袁勇,等. 面向知识自动化的自动问答研究进展[J]. 自动化学报,2017,(9):1491-1508.

[6] Wu Y, Mu T, Goulermas JY.Translating on Pairwise Entity Space for Knowledge Graph Embedding[J].Neurocomputing, 2017, 260:411-419.

[7] 宋园园 SYY. 一种基于领域知识的特征提取算法[J]. 云南民族大学学报: 自然科学版, 2017, 26 (3): 252-257.

[8] 王蕾, 谢云, 周俊生, 等. 基于神经网络的片段级中文命名实体识别[J]. 中文信息学报, 2018, 32(3): 84-90.

[9] 曹倩，赵一鸣. 知识图谱的技术实现流程及相关应用[J]. 情报理论与实践，2015，38（12）: 13-18.

[10] Wu H C, Luk R W P, Wong K F, et al. Interpreting TF-IDF term weights as making relevance decisions[J]. Acm Transactions on Information Systems, 2008, 26(3):55-59.

[11] 官赛萍, 靳小龙, 贾岩涛, et al. 面向知识图谱的知识推理研究进展[J]. 软件学报, 2018, 29(10):74-102.

[12] 杜婧君, 陆蓓, 谌志群. 基于中文维基百科的命名实体消歧方法[J]. 杭州电子科技大学学报, 2012, 32(6):57-60.

[13] 赵军.命名实体识别、排歧和跨语言关联[J]. 中文信息学报,2009,(2):3-17.

[14] 尹榕慧,姚祖发.面向多领域标准的数据质量评估框架研究[J].标准科学,2020(01):92-95.

[15] 王毅, 谢娟, 成颖. 结合 LSTM 和 CNN 混合架构的深度神经网络语言模型[J].

情报学报, 2018, 37(2): 194-205.

[16] 陆向艳,苏崇,刘峻.基于朴素贝叶斯的敏感信息识别方法研究[J].网络安全技术与应用,2021(07):56-57.

[17] 杜泽宇,杨燕,贺樑.基于中文知识图谱的电商领域问答系统[J].计算机应用与软件,2017,34(05):153-159.

[18] 陈程,翟洁,秦锦玉,江嘉,武海霞,蔡婷婷. 基于中医药知识图谱的智能问答技术研究[J]. 中国新通信,2018,20(2):204-207.

[19] Foy N F, Fergerson R W, Musen M A. The knowledge model of Protege-2000: Combining interoperability and flexibility[C]//International Conference on Knowledge Engineering and Knowledge Management. Springer, Berlin, Heidelberg, 2000: 17-32.

[20] Wang P, Wu Q, Shen C, et al. Explicit Knowledge-based Reasoning for Visual Question Answering[J]. Computer Science, 2015.

[21] Ckash Bharadwaj, David Mortensen, Chris Dyer, Jaime G Carbonell. Phonologically aware neural model for named entity recognition in low resource transfer settings. EMNLP, pages 1462–1472, 2016.

[22] 李枫林, 柯佳. 基于深度学习框架的实体关系抽取研究进展[J]. 情报科学, 2018, 36(3): 169-176.

[23] 鲍玉来,耿雪来,飞龙.基于卷积神经网络的旅游信息关系抽取研究[J].现代情报,2019,39(08):132-136.

[24] 王新磊. 受限领域内基于中文问句语义相关度计算的智能问答系统研究[D]. 山东财经大学,2014.25-45.

[25] William Leeson, Adam Resnick, Daniel Alexander, et al. Natural Language Processing (NLP) in Qualitative Public Health Research: A Proof of Concept Study. 2019, 18

[26] Hyeyoung Park, Kwanyong Lee. Adaptive Natural Gradient Method for Learning of Stochastic Neural Networks in Mini-Batch Mode. 2019, 9(21)

[27] Bizer C, Schultz A. The berlin sparql benchmark[J]. International Journal on Semantic Web and Information Systems (IJSWIS), 2009, 5(2): 1-24.

[28] 代文强,李晓荣,冯毅.最大和搜索结果多样性问题及其贪婪算法分析[J].系统工程理论与实践,2016,36(03):706-711.

[29] Androutsopoulos I , Koutsias J , Chandrinos K V , et al. An evaluation of Naive Bayesian anti-spam filtering[J]. 2000.Webber, Jim. [ACM Press the 3rd annual conference - Tucson, Arizona, USA (2012.10.19-2012.10.26)] Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity - SPLASH \"12 - A programmatic introduction to Neo4j[C]// Conference on Systems. 2012:217.

[30] 朱咸军,洪宇,黄雅琳,张馨予,肖芳雄.基于 HMM 的算法优化在中文分词中的应用[J].金陵科技学院学报,2019,35(03):1-7.

[31] 孙雅婧,李成华,杨斌,江小平,艾提日也古丽·艾尼瓦尔.基于 BI-LSTM-CRF 模型的维吾尔语分词研究[J].青海师范大学学报(自然科学版),2019,35(04):5-12.

[32] Samer Abdulateef;Naseer Ahmed Khan;Bolin Chen;Xuequn Shang Multidocument Arabic Text Summarization Based on Clustering and Word2Vec to Reduce Redundancy[J] Information,2020

[33] 赵姗姗. 深度学习与多元特征相结合的答案选择排序研究[D]. 哈尔滨工业大学,2016:1-62.

[34] Chiu J P C, Nichols E. Named entity recognition with bidirectional LSTM-CNNs. arXiv preprint arXiv:1511.08308, 2015.

[35] 王军强. 基于数据挖掘在社交网络中热点话题的研究[D].浙江:浙江理工大学,2016.22-28

致谢

在华东师范大学的三年研究生就读让我快速成长，想起当初刚入校园的我和现在的我相比，我能感受到时光在我身上作用着的点点滴滴，知识伴随着我的成长，让我的生活过的充实而多彩。回首这三年的学习经历，我的心中洋溢着感激以及激动。

在这里我很感谢我的导师刘献忠，每当我有疑惑以及无法找到方向的时候，导师总能在最关键的时候给予我方向，让我在图书馆中找到最适合自己的书籍。在我学习与成长期间，深深受益于导师的关心与淳淳教导。他作为老师，为我点亮了前行之路，作为长辈，对我无微不至。能成为导师的学生是我一辈子值得庆幸的事情，在此我对导师表示我最真挚的感谢。

我还要在此感谢那些在我学习和生活中给予我帮助以及鼓励的老师以及同学们，感谢你们一直的关心与支持。我最要好的朋友彭玉林、马威锋，三年来我们共同成长，共同进步，感谢你们给予我的帮助。同窗友谊，我将终身难忘。

我还要感谢我早一届毕业的朋友郑小林，感谢他能在我为论文焦头烂额时给予帮助，在我论文遇到瓶颈的时候也是他及时给出他意见，并且给我介绍他受找到的最新的实验方法。此外我还想感谢一下平时和我一起研究机器学习、深度学习以及自然语言处理的小伙伴，非常感谢你们的支持。

最后我还要感谢我的母校---华东师范大学，感谢你给了我一个成长和学习的平台，让我不断吸取新知识，充实自己。