计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(4):93−98 [doi: 10.15888/j.cnki.csa.007924]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

使用贝叶斯分类的高考学业规划智能问答系统①

孙　弋, 李　直

(西安科技大学 通信与信息工程学院, 西安 710054)
通讯作者: 李　直, E-mail: 18710731037@163.com

摘　要: 考生在填报高考志愿时, 针对复杂繁多的各类高校信息数据, 传统的搜索引擎无法根据考生需要的实际信息和搜索结果进行匹配, 考生还需要额外消耗一定精力去筛选数据, 这无疑增加了考生的时间成本. 为此本文提出了基于高考领域知识图谱, 使用中文分词模型和朴素贝叶斯分类算法, 设计并开发了针对高考学业规划的智能问答系统. 与传统的搜索引擎不同的是, 基于人工智能的问答系统能够对考生所关注的问题和搜索结果进行精确匹配, 减少考生重复搜索和筛选数据的次数. 测试结果表明, 本系统可以对高考学业规划中所涉及的大多数问题进行相对准确的针对性回答.

关键词: 高考志愿; 知识图谱; 中文分词; 贝叶斯分类; 问答系统

引用格式: 孙弋,李直.使用贝叶斯分类的高考学业规划智能问答系统.计算机系统应用,2021,30(4):93–98. http://www.c-s-a.org.cn/1003-3254/7924.html

Intelligent Question Answering System for College Entrance Examination Using Bayesian Classification

SUN Yi, LI Zhi

(College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China)

Abstract: The traditional search engine cannot match the actual information needed by the candidates with searching results when they fill the list of preference in college entrance application, consuming extra energy of them to filter the data, which undoubtedly increase the time cost. We design an intelligent question answering system for academic planning of examinees with the knowledge graph of the college entrance examination, a model for Chinese word segmentation and the Bayesian classification algorithm. Unlike traditional search engines, the artificial intelligence-based question answering system can accurately match the candidates' questions with search results, reducing the number of repeated searches and data filtering. The test results demonstrate that the system can offer accurate and targeted answers to most of the questions involved in the academic planning.

Key words: preferred college or university list for college entrance exam; knowledge graph; Chinese word segmentation; naive Bayesian classification; question answering system

随着人工智能技术的快速发展, 各行各业都展开了对人工智能技术在本领域应用的探索, 以求将人工智能技术应用到所属行业. 由于计算成本的迅速降低和新一代无线通讯技术等面向行业领域的广泛应用, 各行业基于网络技术产生了大量结构化和半结构化领域知识数据, 新产生的数据具有数据体量巨大、所含信息量大、整体数据蕴含应用价值巨大的特点, 但对于特定领域的信息检索, 传统的搜索引擎由于涉及面广, 搜索出来的结果过于宽泛, 用户还是需要花费额外时间进行二次查找. 以高考志愿填报为例, 在这种专业性

① 收稿时间: 2020-08-13; 修改时间: 2020-09-03, 2020-10-30; 采用时间: 2020-11-04; csa 在线出版时间: 2021-03-30

较强的领域, 使用传统搜索引擎得到的结果往往会出现二义性问题, 这种差错往往会影响到考生将来的学业规划. 本文利用基于知识图谱构建技术构建出来的垂直领域知识库[1,2] 进行智能问答系统开发, 使用 HanLP (Han Language Processing) 中文分词工具对用户提出的问题进行分词处理, 提取出其中的实体关键信息, 之后采用朴素贝叶斯分类器对提取出的关键信息进行分类[3], 最后根据区分出的问题类别, 从构建好的垂直领域知识库进行信息检索和答案返回, 最终实现面向垂直领域的智能问答系统[4–6].

1 相关研究

知识问答通过自然语言对话的形式帮助人们从知识库中获取知识, 不但是知识图谱的核心应用之一, 也是自然语言处理的重要研究方向. 知识问答系统是一个拟人化的智能系统, 通过接收使用自然语言表达的问题, 理解用户的意图, 获取相关的知识, 最终经过推理计算形成自然语言表达的答案并反馈给用户.

早期的问答系统 NLIDB (Natural Language Interface to Data Base) 是伴随着人工智能的研发逐步兴起的[7], 以 1961 年的 Baseball 系统[8] 和 1972 年的 Lunar 系统[9] 为代表. Baseball 系统回答了有关一年内棒球比赛的问题. Lunar 在阿波罗月球任务期间提供了岩石样本分析数据的界面. 这些系统一般限定在特定领域, 使用的自然语言问题询问结构化知识库. 这些数据库与如今的关系型数据库不同, 更像基于逻辑表达式的知识库. 这一类系统通常为领域应用定制.

基于知识库的问答系统 (Knowledg-Based Question Answering, KBQA) 特指使用基于知识图谱解答问题的问答系统. KBQA 实际上是 20 世纪七八十年代对 NLIDB 工作的延续, 其中很多技术都借鉴和沿用了以前的研究成果, 其中主要的差异是采用了相对统一的基于三元组表示的知识图谱, 并且把予以理解的结果映射到知识图谱的本体后生成查询语句查询解答问题.

2 相关技术

2.1 知识图谱

知识图谱[10] 是一种用图模型来描述知识和建模世界万物之间的关联联系的技术方法. 知识图谱由节点和边组成. 节点可以是实体, 如一个人、一本书等, 或是抽象的概念, 如人工智能、知识图谱等. 边可以是实体的属性, 如姓名、书名, 或是实体之间的关系. 如朋友、配偶. 知识图谱技术是人工智能技术的重要组成部分, 以结构化的方式描述客观世界中的概念、实体及其属性的关系.

知识图谱根据覆盖范围可以分为开放领域知识图谱和垂直领域知识图谱[11]. 开放领域知识图谱通常不被限定于特定的领域中. 它包含大量的常识性知识, 更追求知识的广泛度. 垂直领域知识图谱则面向某个特定的行业领域, 更追求知识的深度与准确度. 高校学业规划知识库属于垂直领域的知识图谱. 本项目从百科类网站、中国教育在线和高校招生网站等结构化数据源中提取出高质量的知识数据, 然后将知识写入图数据库进行持久化, 最终构建出高考领域的知识图谱[12].

2.2 中文分词

中文分词[13], 即 Chinese word segmentation, 即将一个汉字序列进行切分, 得到一个个单独的词. 中文分词与英文分词有很大的不同, 对英文而言, 一个单词就是一个词, 而汉语是以字为基本的书写单位, 词语之间没有明显的区分标记, 需要进行分词处理, 将句子转化为词的表示. 主要的困难在于分词歧义, 此外, 像未登录词、分词粒度都是影响分词效果的重要因素[14]. 众多分词方法主要可以分为基于规则的分词和基于统计的分词两种.

1) 基于规则的分词

规则分词是最早兴起的方法, 简单高效. 主要是通过维护词典, 在切分语句时, 将语句的每个字符串与词表中的词进行逐一匹配, 找到则切分. 按照匹配切分方式, 主要分为正向最大匹配法、逆向最大匹配法以及双向最大匹配法.

2) 基于统计的分词

把每个词看作是由各个字组成, 如果相连的字在不同的文本中出现的次数越多, 就证明这个相连的字很可能就是一个词. 因此我们就可以利用字与字相邻出现的频率来反映成词的可靠度, 当组合频度高于某个临界值时, 我们便可以认为这些字会构成一个词语.

随着 NLP 技术的日益成熟, 中文分词工具越来越多, 常见的有中科院计算所 NLPIR、哈工大 LTP、清华大学 THULAC、斯坦福分词器、HanLP 分词器、jieba 分词等. 本文选择采用基于深度学习的中文分词工具包 HanLP 来进行分词.

2.3 贝叶斯分类器

贝叶斯分类是一类分类算法的总称, 这类算法均以贝叶斯定理为基础, 故统称为贝叶斯分类[15]. 贝叶斯分类器主要有 4 种, 分别是: naive Bayes (朴素贝叶斯)、TAN、BAN 和 GBN, 本文涉及的主要是 naive Bayes. Naive Bayes 分类器在很多真实的分类问题, 例如文档分类、垃圾邮件过滤, 效果很理想. 它只需要少量的训练数据去估计必要的参数, 而且, 相比其它复杂的方法, naive Bayes 分类器执行非常快.

朴素贝叶斯分类是一种十分简单的分类算法[16], 其基础思想为: 对于给出的待分类项, 求解在此项出现的条件下各个类别出现的概率, 哪个概率最大, 就认为此待分类项属于哪个类别. 朴素贝叶斯分类的定义如下:

(1) 设$x=\{x_1,x_2,\cdots,x_m\}$为一个待分类项, 其中每个x都为一个特征属性.

(2) 有类别集合$C=\{y_1,y_2,\cdots,y_n\}$.

(3) 计算$P(y_1|x),P(y_2|x),\cdots,P(y_n|x)$.

(4) 若$P(y_k|x)=\max\{P(y_1|x),P(y_2|x),\cdots,P(y_n|x)\}$, 则$x\in y_k$.

其中, 第 (3) 步中各个条件概率的计算步骤如下:

(1) 找到一个已知分类的待分类项集合, 将这个集合作为训练样本集.

(2) 统计得到在各类别下各个特征属性的条件概率估计:

$$\begin{cases}P(x_1|y_1),P(x_2|y_1),\cdots,P(x_m|y_1)\\P(x_1|y_2),P(x_2|y_2),\cdots,P(x_m|y_2)\\\vdots\\P(x_1|y_n),P(x_2|y_n),\cdots,P(x_m|y_n)\end{cases}$$

(3) 如果各个特征属性是条件独立的, 则根据贝叶斯定理有如下推导:

$$P(y_i|x)=\frac{P(x|y_i)P(y_i)}{P(x)} \tag{1}$$

根据全概率公式, 可以进一步地分解上式中的分母, 得到:

$$P(y_i|x)=\frac{P(x|y_i)P(y_i)}{\sum_i P(x|y_i)P(y_i)} \tag{2}$$

因为分母对于所有的类别为常数, 只要将分子最大化即可, 又因为各特征属性是条件独立的, 所以有:

$$\begin{aligned}P(x|y_i)P(y_i)&=P(x_1|y_i)P(x_2|y_i),\cdots,P(x_m|y_i)P(y_i)\\&=P(y_i)\prod_{j=1}^{m}P(x_j|y_i)\end{aligned} \tag{3}$$

然后将式 (3) 带入到式 (2) 中, 得到:

$$P(y_i|x)=\frac{P(y_i)\prod_{j=1}^{m}P(x_j|y_i)}{\sum_i P(y_i)\prod_{j=1}^{m}P(x_j|y_i)} \tag{4}$$

于是朴素贝叶斯分类器可表示为:

$$\begin{aligned}f(x)&=\arg\max P(y_i|x)\\&=\arg\max\frac{P(y_i)\prod_{j=1}^{m}P(x_j|y_i)}{\sum_i P(y_i)\prod_{j=1}^{m}P(x_j|y_i)}\end{aligned} \tag{5}$$

因为对所有的y_k, 上式中的分母的值都是一样的, 所以可以忽略分母部分, 朴素贝叶斯分类器最终表示为:

$$f(x)=\arg\max P(y_i)\prod_{j=1}^{m}P(x_j|y_i) \tag{6}$$

本文中利用朴素贝叶斯分类器, 首先对分词及抽象化后的句子进行分类, 然后对用户输入的问题与本地的匹配模板进行匹配, 将最终的匹配结果对应到相应的问题模板, 最后根据问题模板去知识库进行查询, 得到最终的问题答案并返回给用户.

3 系统设计

本文将针对高考学业规划中出现的一系列用户可能提出来的问题, 构建面向高考学业规划的智能问答系统. 首先构建出基于高考学业规划领域的知识图谱, 然后利用 HanLP 中文分词工具对用户提出的这一领域的问题进行分词处理, 提取出其中的实体关键信息, 并将实体进行抽象化处理, 之后采用朴素贝叶斯分类器对抽象化后的问题进行分类, 最后根据区分出的问题类别, 从构建好的垂直领域知识库进行信息检索和答案返回, 最终实现面向高考志愿规划领域的智能问答系统.

3.1 系统架构设计

基于以上所介绍的相关技术研究, 本文将基于高考学业规划的相关领域设计并开发智能问答系统, 其系统架构如图 1 所示.

系统的整体架构分为前端用户界面、后端处理系统和数据层. 前端用户界面获得用户输入的问题和最终结果展示, 后端处理部分通过对问题进行分词、句子抽象、问题分类、模板匹配和答案检索等步骤获得最终答案并输出到前端. 系统所用数据包括分词表、分类模板、搜索模板和高校信息知识图谱等. 本文主要

关注于系统后端处理这一部分, 下文将会详细论述后端处理部分的分词模块以及问题分类模块的实现方法.

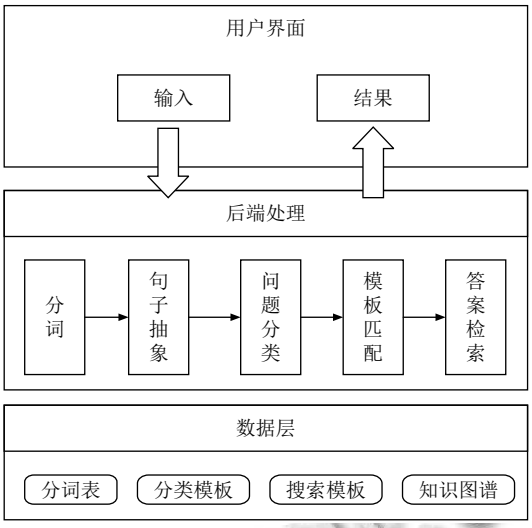

图 1 问答系统架构

3.2 分词模块

HanLP 是由一系列模型与算法组成的 Java 工具包, 目标是促进自然语言处理在生产环境中的应用, HanLP 具备功能完善、性能高效、架构清晰、语料时新、可自定义的特点. HanLP 提供的中文分词包括最短路分词、N-最短路分词、CRF 分词、索引分词、极速词典分词、用户自定义词典等方法, 在提供丰富功能的同时, HanLP 内部模块坚持低耦合、模型坚持惰性加载、服务坚持静态提供、词典坚持明文发布, 使用非常方便, 同时自带一些语料处理工具, 帮助用户训练自己的语料.

HanLP 分词工具进行分词的工作流程如图 2 所示. 首先将词典加载到 HanLP 分词模块, 然后把待分词的句子进行分词输出分词结果, 之后用标注词性对句子进行抽象化, 最后输出最终抽象化后的句子以供分类器进行分类.

本系统针对高考志愿规划领域使用 HanLP 对用户输入的问题进行分词, 并且额外添加了自定义的高校及专业信息词典数据集, 中文分词示例代码如下代码 1.

代码 1. HamLP 中文分词示例代码

```
public void TestA(){
  String lineStr = "电子科学与技术是学什么的? ";
  try{
    Segment segment = HanLP.newSegment();
    segment.enableCustomDictionary(true);
    CustomDictionary.add("电子科学与技术","pr 0");
    List<Term> seg = segment.seg(lineStr);
    for (Term term : seg) {
      System.out.println(term.toString());
    }
    }catch(Exception ex){
    }
}
```

执行这段代码将“电子科学与技术是学什么的? ”这句话进行分词, 得到的结果如下:

电子科学与技术/pr 专业/n 是/vshi 学/v 什么/ry 的/ude1? /w

pr, n, vshi, v, ry, ude1, w 分别代表了当前词的词性, 其中 pr 为我们自定义的词性. 分词完成后对原句子进行抽象, 将其中的专业名称用 pr 替换并抽象句子.

句子抽象化结果: pr 是 学什么 的?

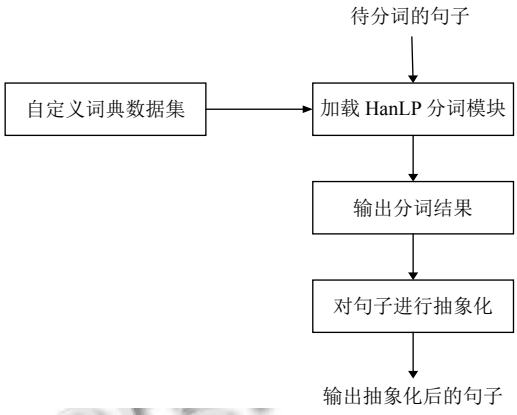

图 2 文本分词流程

3.3 确定问题训练样本

本系统采用 Java 语言调用 Spark 引擎来实现贝叶斯分类器.

首先生成训练样本集, 使用爬虫技术从今日头条、分答以及百度贴吧等网站爬取到针对高考学业规划的相关问题, 共计 5000 多条, 然后通过人工进行标注以及预处理, 总共将这些问题分为了 16 大类, 例如, 询问专业介绍, 其样本语料库如下 (其中“pr”为上文提到的自定义的分词词性):

{pr; pr 的简介; pr 的介绍; pr 专业怎么样; pr 是; pr 是什么专业; pr 的详细信息; pr 的信息; pr 是干什么的; …}

语料库中的数据是对询问专业情况的一些可能会出现的问题的集合, 其余问题的训练样本集也类似, 对样本集进行编号, 并统一汇总, 将样本数据构造成 LabelPoint

类型, 即 double 数组构成的稠密/稀疏向量. 然后再生成测试数据样本, 同样将样本数据构造成 LabelPoint 类型.

对于上面提到的将样本数据构造成 LabelPoint 类型, 即 double 数组构成的稠密/稀疏向量, 比如训练样本数据: “pr 是学什么的? ”, 需要提取训练样本数据里面的关键词, 如: “学”、“什么”, 即前面第 2.3 节提到的待分类特征属性.

对于关键词的提取采用 HanLP 进行分词提取, 构建向量时跟词汇数据集进行比对, 其中词汇数据集也使用 HanLP 分词工具, 将所有的语料样本进行分词, 提取出其中所有非实体词汇的词语构建出一个词汇数据集; 构建向量时首先根据得到的词汇表大小对向量进行初始化, 然后将分词后训练样本与词汇表进行比对, 如果匹配命中就将此项置为 1, 否则默认为 0. 示例代码如代码 2.

代码 2. 设定问题训练样本示例代码

```
public double[] sentenceToArrays(String sentence) {
    double[] vector = new double[vocabuary.size()];
/**模板对照词汇表的大小进行初始化, 全部为 0.0*/
    for(int i = 0;i<vocabuary.size();i++){
      vector[i] = 0;
    }
    Segment segment = HanLP.newSegment();
    List<Term> terms = segment.seg(sentence);
    for(Term term : terms){
      String word = term.word;
      if(vocabulary.containsKey(word)){
        int index = vocabulary.get(word);
        vector[index] = 1;
      }
    }
  return vector;
}
```

3.4 贝叶斯分类模块

本系统采用 Java 语言调用 Spark 引擎来实现贝叶斯分类器.

根据上述敲定的训练样本集合由 SparkContext 实例创建出一个可以被并行操作的分布式数据集 JavaRDD, 再然后将 JavaRDD 类型转换为 RDD 数据并交给贝叶斯分类器进行训练. 示例代码如代码 3.

代码 3. 贝叶斯分类器示例代码

```
public NaiveBayesModel loadClassifierModel() throws Exception {
  SparkConf conf = new SparkConf().setAppName("NaiveBayesTest").
setMaster("local[*]");
  JavaSparkContext sc = new JavaSparkContext(conf);
  List<LabeledPoint> train_list = new LinkedList<>();
  String[] sentences;
  Map<Double, String> seqWithSamples = loadQuestionSamples
("question");
  if(seqWithSamples == null || seqWithSamples.size() == 0){
      throw new Exception("缺少问题训练样本, 请检查! ");
    }
    for(Map.Entry<Double, String> entry : seqWithSamples.
entrySet()) {
        Double seq = entry.getKey();
        String sampleContent = entry.getValue();
        sentences = sampleContent.split("`");
        for (String sentence : sentences) {
          double[] array = sentenceToArrays(sentence);
         LabeledPoint train = new LabeledPoint(seq, Vectors.dense
(array));
          train_list.add(train);
        }
    }
    JavaRDD<LabeledPoint> trainingRDD = sc.parallelize(train_list);
    /**开始训练样本*/
    NaiveBayesModel nb_model = NaiveBayes.train(trainingRDD.
rdd());
    /** 关闭资源*/
    sc.close();
    /** 返回贝叶斯分类器*/
    return nb_model;
}
```

贝叶斯分类器用测试数据样本跟训练的数据进行概率预测, 最终返回我们定义的类别编号. 示例代码如代码 4.

代码 4. 问题分类示例代码

```
public String queryClassify(String sentence) throws Exception {
    double[] testArray = sentenceToArrays(sentence);
    Vector v = Vectors.dense(testArray);
    double index = nbModel.predict(v);
    modelIndex = (int)index;
    System.out.println("the model index is " + index);
    Vector vRes = nbModel.predictProbabilities(v);
    double[] probabilities = vRes.toArray();
    System.out.println("============ 问题模板分类概率
============");
    for (int i = 0; i < probabilities.length; i++) {
      System.out.println("问题模板分类【"+i+"】概率: "+String.
format("%.5f", probabilities[i]));
    }
    System.out.println("============ 问题模板分类概率
============");
    return questionsPattern.get(index);
}
```

通过上面的分词模块和贝叶斯分类模块, 我们最

终得到了用户输入问题的类别编号以及问题的主要实体, 然后利用图数据库查询语言在构建好的知识库中进行查询. 本系统采用的 Neo4j 数据库使用的查询语言是 Cypher 查询语句, 以前文中“电子科学与技术专业是学什么的”问题为例, 经过贝叶斯分类器得到其在“专业介绍”这个问题类型的概率为 65.7%, 则其对应的查询语句如下:

MATCH (n:Profession) WHERE n.name='电子科学与技术' RETURN n.description

最终返回问题中所提到的专业的基本介绍.

4 系统实现与测试

本文根据前文第 3 节的技术路线和方法, 使用 Java 作为开发语言, 采用 Spring Boot 作为系统框架, 使用 Spark 通用计算引擎处理数据集并实现贝叶斯分类算法, 最终完成高考学业规划智能问答系统的设计与开发.

本系统测试运行的硬件环境为 Intel Core i5-8400 处理器, 16 GB 内存, 操作系统为 Windows. 经测试, 本系统能够在以上运行环境下正常运行并实现所设计的主要功能, 响应延时在正常范围之内, 用户体验良好. 能够针对高考学业规划中涉及的普遍问题, 较准确的返回考生用户所需要的信息, 后期根据用户使用次数的增多, 针对这几种情况之外的问题需要继续完善训练样本, 以增加更多问题类别, 实现对更多问题的解答. 系统的运行效果如图 3 所示.

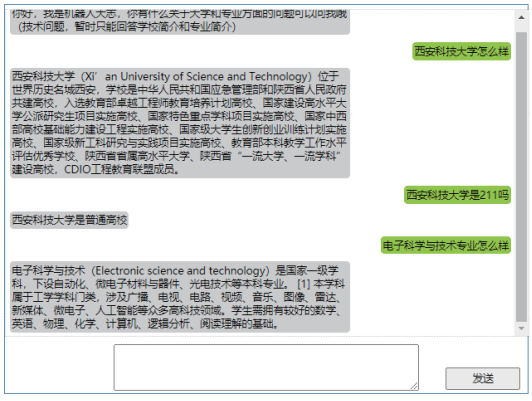

图 3 智能问答系统问答界面

5 总结

随着互联网和人工智能技术在各行业领域应用的进一步深入, 知识图谱中的知识获取与知识推理使自动问答技术能够在细分垂直领域为人们提供更多、更准确的服务. 在本文所设计的高考学业规划领域, 自动问答系统可以在一定范围为考生用户提供更精确、详尽、专业的服务. 但目前还有以下两个方面有待提升: 一是对于知识库和语料库内容的逐步完善, 以求使知识的完整性以及准确性得到进一步的提升; 二是需要通过应用分类更加准确的分类算法, 使分类匹配的精确度得到增强, 以求更快速得到更加精准的答案, 并进一步改善用户的体验.

参考文献

1 刘峤, 李杨, 段宏, 等. 知识图谱构建技术综述. 计算机研究与发展, 2016, 53(3): 582–600. [doi: 10.7544/issn1000-1239.2016.20148228]

2 曹皓伟, 徐建良, 窦方坤. 基于 Neo4j 生物医药知识图谱的构建. 计算机时代, 2020, (6): 35–38.

3 张慧芳. 基于分布式框架下的中文文本特征分类研究 [硕士学位论文]. 包头: 内蒙古科技大学, 2019.

4 张紫璇, 陆佳民, 姜笑, 等. 面向水利信息资源的智能问答系统构建与应用. 计算机与现代化, 2020, (3): 65–71.

5 王加存. 面向领域的问答系统关键技术的研究与应用 [硕士学位论文]. 沈阳: 中国科学院大学 (中国科学院沈阳计算技术研究所), 2020.

6 李敬鑫. 基于深度学习的智能问答系统研究 [硕士学位论文]. 成都: 电子科技大学, 2020.

7 Androutsopoulos I, Ritchie GD, Thanisch P. Natural language interfaces to databases-an introduction. Natural Language Engineering, 1995, 1(1): 29–81. [doi: 10.1017/S1351324900000005X]

8 Green BF, Wolf AK, Chomsky C, *et al*. Baseball: An automatic question-answerer. Proceedings of the Western Joint Computer Conference. New York, NY, USA. 1961. 219–224.

9 Woods WA. Progress in natural language understanding: An application to lunar geology. Proceedings of National Computer Conference and Exposition. New York, NY, USA. 1973. 441–450.

10 黄恒琪, 于娟, 廖晓, 等. 知识图谱研究综述. 计算机系统应用, 2019, 28(6): 1–12. [doi: 10.15888/j.cnki.csa.006915]

11 孙僖. 垂直领域知识图谱构建的关键技术研究 [硕士学位论文]. 北京: 北京邮电大学, 2019.

12 陈德彦, 赵宏, 张霞. 基于领域语义知识库的疾病辅助诊断方法. 软件学报, 2020, 31(10): 3167–3183. [doi: 10.13328/j.cnki.jos.005825]

13 唐琳, 郭崇慧, 陈静锋. 中文分词技术研究综述. 数据分析与知识发现, 2020, 4(S1): 1–17.

14 徐晓芳. 基于条件随机场的中文分词技术的研究与实现 [硕士学位论文]. 南京: 南京邮电大学, 2018.

15 王阳, 周云才. 朴素贝叶斯分类算法的设计与分析. 电脑知识与技术, 2019, 15(11): 206–208.

16 高晓利, 王维, 赵火军. 几种改进的朴素贝叶斯分类器模型. 电子世界, 2018, (21): 40–41.