

**《基于知识图谱的自动问答系统》**

**设计说明书**



北京航空航天大学

2016-01

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| 1 | 2015.12.31 | 方凯 |  | 完成1、2(部分)、3(部分)、 |
| 2 | 2015.01.01 | 杨东东 |  | 完成2(部分)、3(部分) 、  4、5、6(部分)、7 |
| 3 | 2015.01.03 | 李睿霖 |  | 完成4（部分）、5（部分）、6（部分），修改整个文档 |
| 4 | 2016.01.18 | 杨东东 |  | 修改 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. 范围 1](#_Toc440833840)

[1.1标识 1](#_Toc440833841)

[1.2系统概述 1](#_Toc440833842)

[1.3文档概述 1](#_Toc440833843)

[1.4术语和缩略词 1](#_Toc440833844)

[1.4.1数据库(Database) 1](#_Toc440833845)

[1.4.2自然语言处理(NLP) 1](#_Toc440833846)

[1.4.3矩阵论(Matrix) 2](#_Toc440833847)

[2. 引用文档 2](#_Toc440833848)

[2.1书籍 2](#_Toc440833849)

[2.2批文 2](#_Toc440833850)

[2.3引用资料 2](#_Toc440833851)

[3. 需求概述 3](#_Toc440833852)

[3.1概述系统特性和需求 3](#_Toc440833853)

[3.2加细节 3](#_Toc440833854)

[3.2.1问题分析 3](#_Toc440833855)

[3.2.2信息检索 3](#_Toc440833856)

[3.2.3答案抽取 3](#_Toc440833857)

[3.3 新需求 3](#_Toc440833858)

[4. 体系结构设计 4](#_Toc440833859)

[4.1总体结构 4](#_Toc440833860)

[4.2功能分配 4](#_Toc440833861)

[4.3关键问题及解决方案 5](#_Toc440833862)

[4.3.1问题分析模块及解决方案 5](#_Toc440833863)

[4.3.2问题分类模块及其解决方案 5](#_Toc440833864)

[5. 接口设计 6](#_Toc440833865)

[5.1用户界面设计 6](#_Toc440833866)

[5.2外部接口设计. 6](#_Toc440833867)

[5.3内部接口设计 6](#_Toc440833868)

[6.数据结构设计 7](#_Toc440833869)

[6.1公共数据结构设计 7](#_Toc440833870)

[6.2数据库设计 7](#_Toc440833871)

[7.详细设计 8](#_Toc440833872)

[7.1预处理 8](#_Toc440833873)

# 范围

## 1.1标识

文档标识号：A2015-00-02-00

文档标题：<<基于知识图谱的自动问答系统>>--系统设计说明书

版本号：4.0

## 1.2系统概述

基于知识图谱的自动问答系统，是以中文为载体的系统，其数据库为以百度百科、维基百科、互动百科为主，运用其中的知识性信息进行人机交互以达到自动问答的目的的系统。采用目前发展中的实体分词技术、实体消歧技术、语法分析技术、语义分析技术等作为基础，综合开发而成。

系统尚在开发过程中；

投资方：无；

需方：用户；

用户：问答需求方；

开发方：杨东东，李睿霖，方凯；

运行现场：Android系统。

## 1.3文档概述

本文档用于阐述系统概况以及系统设计说明。该文档使用时保密性一般，由于未涉及核心代码，使用时可以半公开。

## 1.4术语和缩略词

### 1.4.1数据库(Database)

MySQL:关联[数据库管理系统](http://baike.baidu.com/view/68446.htm)

Neo4j:高性能的,NOSQL图形数据库

Redis:内存数据库，以高效的查找存储著称，并具有数据持久的特性

### 1.4.2自然语言处理(NLP)

Entity Linking:实体链接

Page Rank:Google开源的一个搜索算法

Entity Ambiguation:实体歧义

Trie Tree:前缀树

kNN: k近邻算法 (kNN，k-NearestNeighbor)

LSA: 隐式语义分析 (Latent Semantic Analysis)

Markov Model:马尔科夫模型

Lucene: 一个开放源代码的全文检索引擎工具包，是一个全文检索引擎的架构

FudanNLP：一个中国国内做得还算不错的NLP处理开源包

### 1.4.3矩阵论(Matrix)

PCA:主成分分析，用于矩阵维度的降维方法 (Principal Component Analysis)

SVD:矩阵奇异值分解 (singular value decomposition method)

[panion](http://cn.bing.com/dict/search?q=Companion&FORM=BDVSP6&mkt=zh-cn) [Matrix](http://cn.bing.com/dict/search?q=Matrix&FORM=BDVSP6&mkt=zh-cn)：伴随矩阵

[ompanion](http://cn.bing.com/dict/search?q=Companion&FORM=BDVSP6&mkt=zh-cn) [Matrix](http://cn.bing.com/dict/search?q=Matrix&FORM=BDVSP6&mkt=zh-cn)：友矩阵

# 引用文档

## 2.1书籍

《软件项目管理》 朱少民，韩莹 编著，人民邮电出版社

《软件项目管理》 Rajeev T Shandilya编著 科学出版社

## 2.2批文

本项目的经核准的计划任务书和上级机关的批文:第九届《大学生创新创业训练计划》

## 2.3引用资料

《<基于知识图谱的自动问答系统>系统需求规格说明书》

《Open Question Answering Over Curated and Extracted Knowledge Bases》

《syntactic constraints on paraphrases extracted from parallel corpora》

《基于维基百科的自动词义消歧方法》 史天艺

《一个中文实体链接语料库的建设》 舒佳根

《基于角色标注的中国人名自动识别研究》张华平 刘群

《基于层叠隐马尔可夫模型的中文命名实体识别》俞鸿魁 张华平 刘群

《基于角色标注的中文机构名识别》俞鸿魁 张华平 刘群

《基于最大熵的依存句法分析》 辛霄 范士喜 王轩 王晓龙

《社区问答系统中若干关键问题研究》 [廉鑫](http://www.cnki.net/kcms/detail/search.aspx?dbcode=CDFD&sfield=au&skey=%e5%bb%89%e9%91%ab&code=25126167;)

《汉语大词汇量连续语音识别系统研究进展》倪崇嘉 刘文举 徐波

# 需求概述

## 3.1概述系统特性和需求

用户发送文字信息给自动问答系统，然后通过自动系统的分析然后返回给用户尽可能使他所想要的信息。该系统从用户发送信息给客户，然后通过自动问答系统中的问题分析模块，信息检索模块和答案抽取模块共三个算法模板检索出准确率最高的答案返回给用户。能够与用户进行聊天，能够回答事实型问题、列举型问题、定义型问题和交互型问题等。

## 3.2加细节

### 3.2.1问题分析

问题分析是问答系统的第一个模块，它负责对用户提出的中文自然语言问题进行深入细致地理解和分析，为信息检索模块和答案抽取模块奠定良好的基础。问题分析阶段的处理结果直接影响着后续阶段的处理。该阶段的分析主要包括词法分析（分词、词性标注）、问题分类、关键词提取及关键词扩展。其中问题分类是此模块的核心，它能有效减少候选答案的空间，提高系统返回答案的准确率，同时问题的类型也决定了答案抽取中一些策略的选择。由于问句与陈述句的句型结构有着明显差别，为方便在答案抽取时进行相似度计算，必要时还需要对问句进行重写。

### 3.2.2信息检索

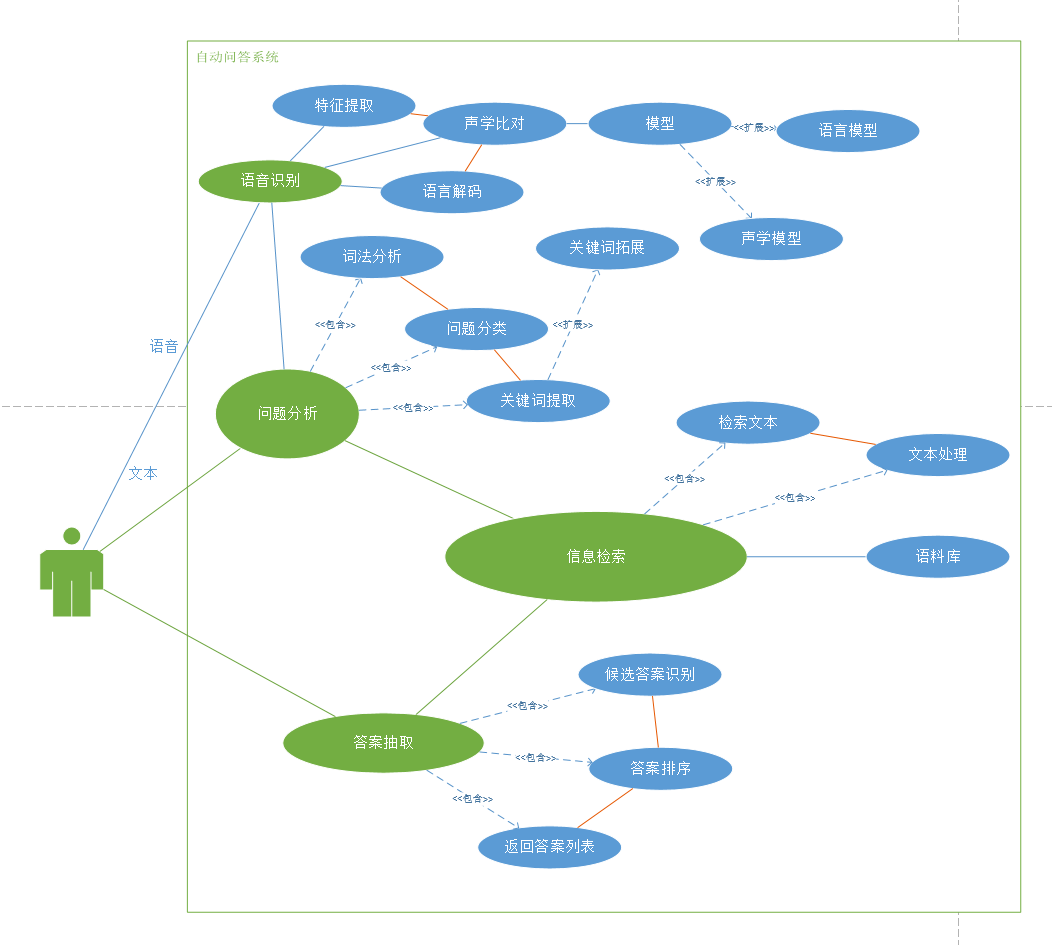
信息检索模块的功能是根据问题分析阶段的关键词提取及扩展生成的查询式在数据库、知识库或互联网[26]上检索所有可能与答案相关的段落、文章或者网页，并做一定的分析处理，将处理结果提交给答案抽取模块进行下一步操作。问答系统中的信息检索同传统的基于关键词匹配的信息检索类似，可以调用现有的比较成熟的各种检索模型。

### 3.2.3答案抽取

答案抽取是问答系统的最终模块，也是最关键的模块。这一模块所应用的答案抽取算法的优劣直接关系着整个问答系统的性能。答案抽取的对象是信息检索模块提交的相关文档，候选答案可以是一句话、一段摘要也可以是一个词，问题类型不同，所得到的答案形式也是不同的，所以答案抽取时还需依据问题类型采取相应的策略。由于信息检索阶段返回的相关文档通常较多，问答系统会抽取出大量的候选答案[27]。因此，候选答案的评估和排序是必不可少的，并最终返回一个最佳答案或按置信度高低排序的答案列表。

## 新需求

之前的部分都对系统如何获取信息，并没有做详细的声明，只是说明了自动问答系统需要一段文本的输入，因此在此，考虑到各种实际情况还有现成的技术，决定将输入自动问答系统的信息媒介允许为语音输入。因此在自动问答系统中需要添加语音转为文字的功能，再进一步接受系统中别的模板的处理。具体的改变的用例模板如图所示。



# 体系结构设计

## 4.1总体结构

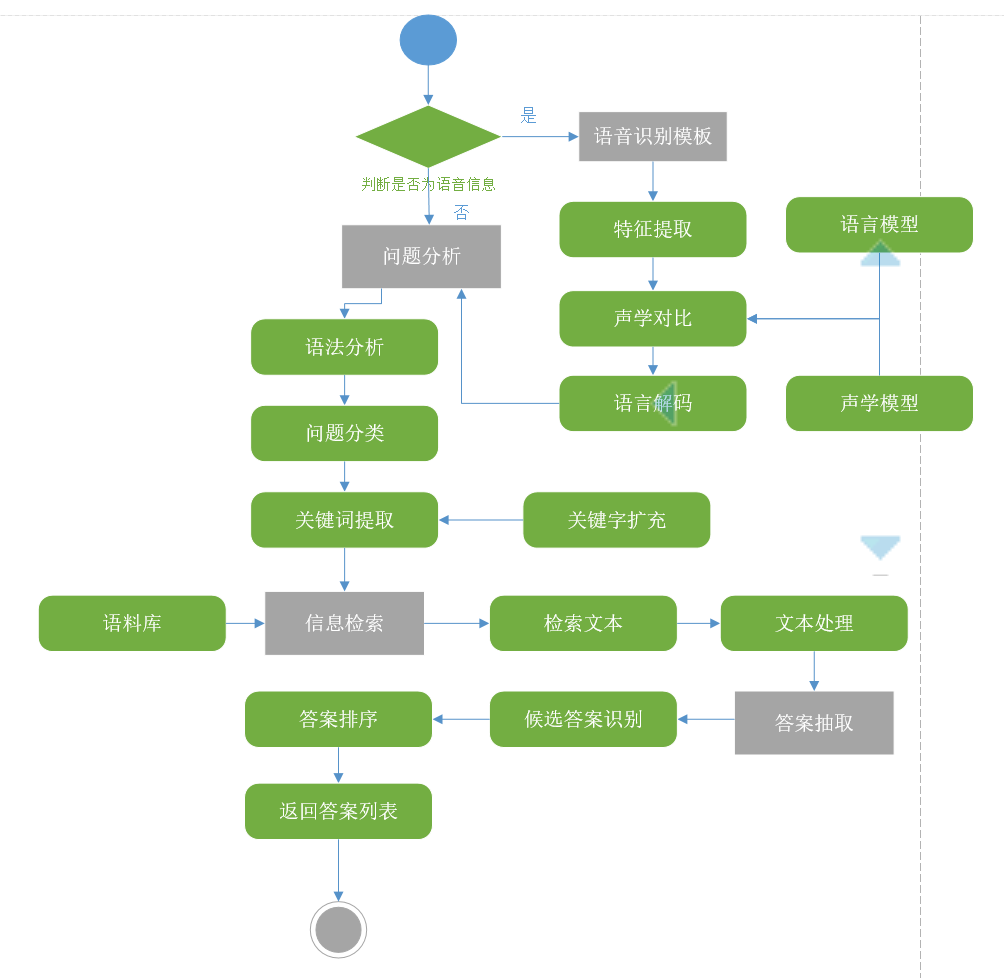
用一系列图表给出系统的总体结构，并对图中的相关内容进行说明。可能包括软件体系结构、硬件体系结构、技术体系结构、支撑体系（部署和实施方案）结构等各个方面，可根据实际情况每个方面分为一小节来写。

采用面向对象分析与设计方法，建议从UML的4+1视图描述软件结构：完善用例图、活动图，给出类图、时序图、状态图。

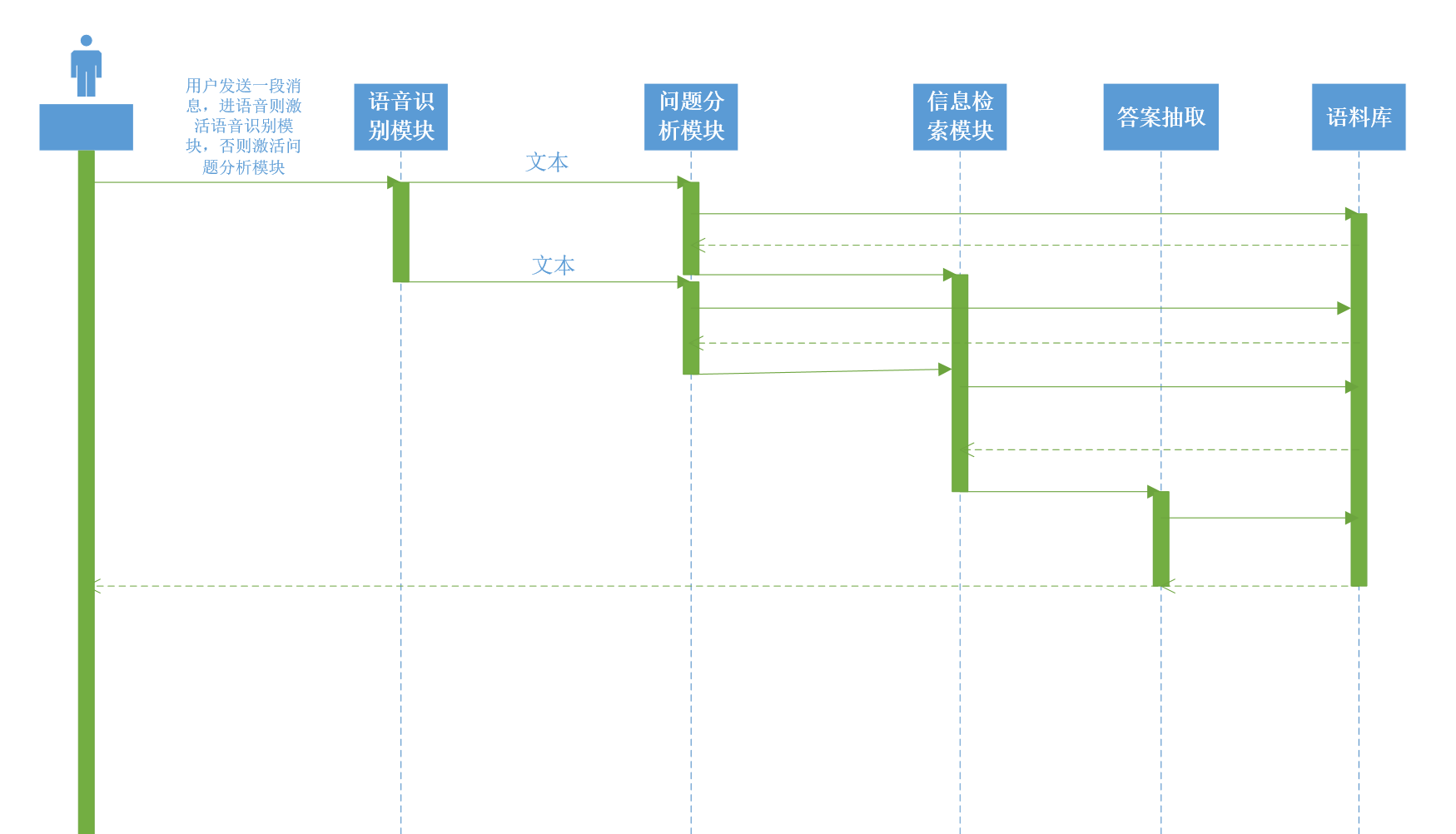
### 4.1.0用例图

如上所示3.3节所示。

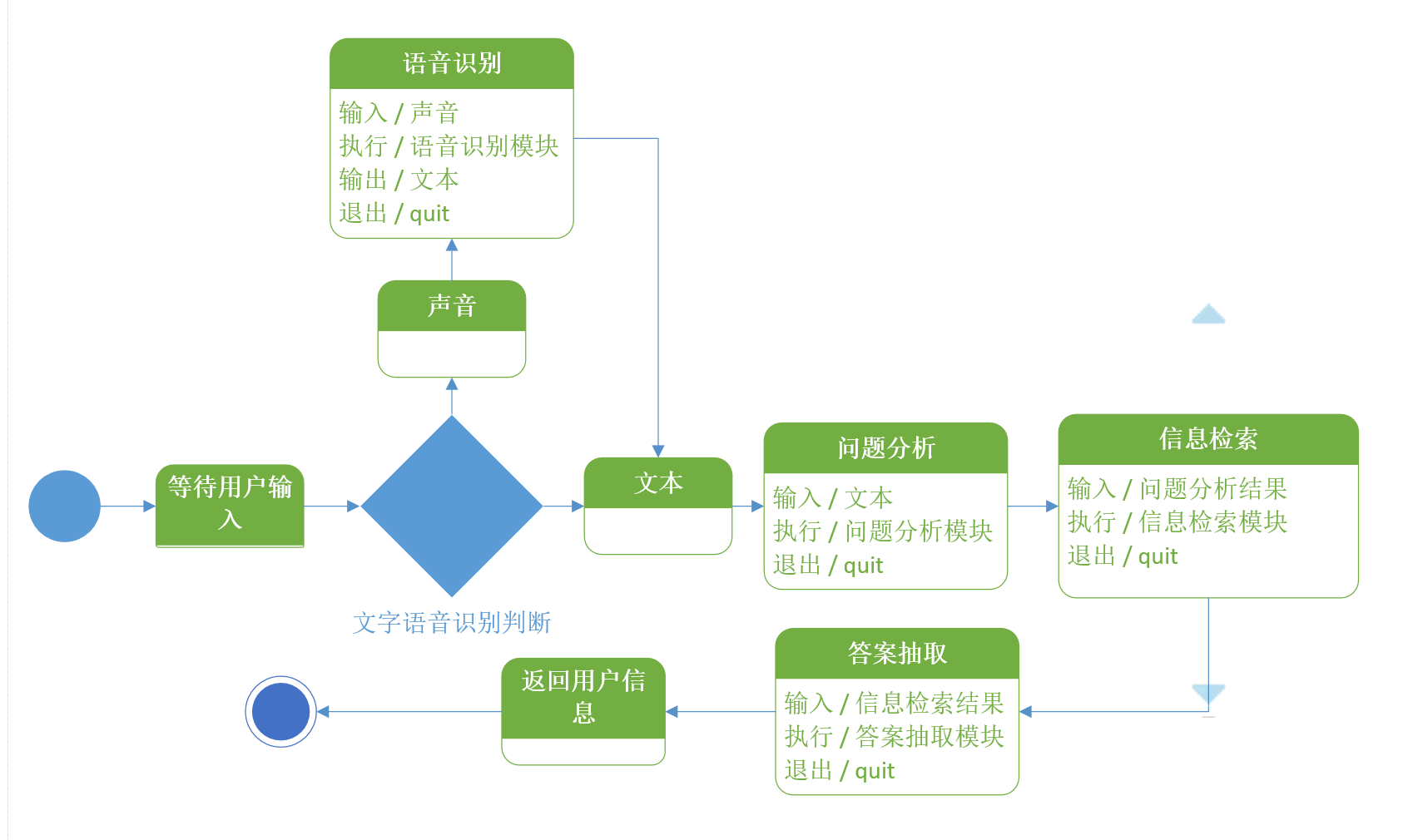
### 4.1.1活动图



### 4.1.2时序图

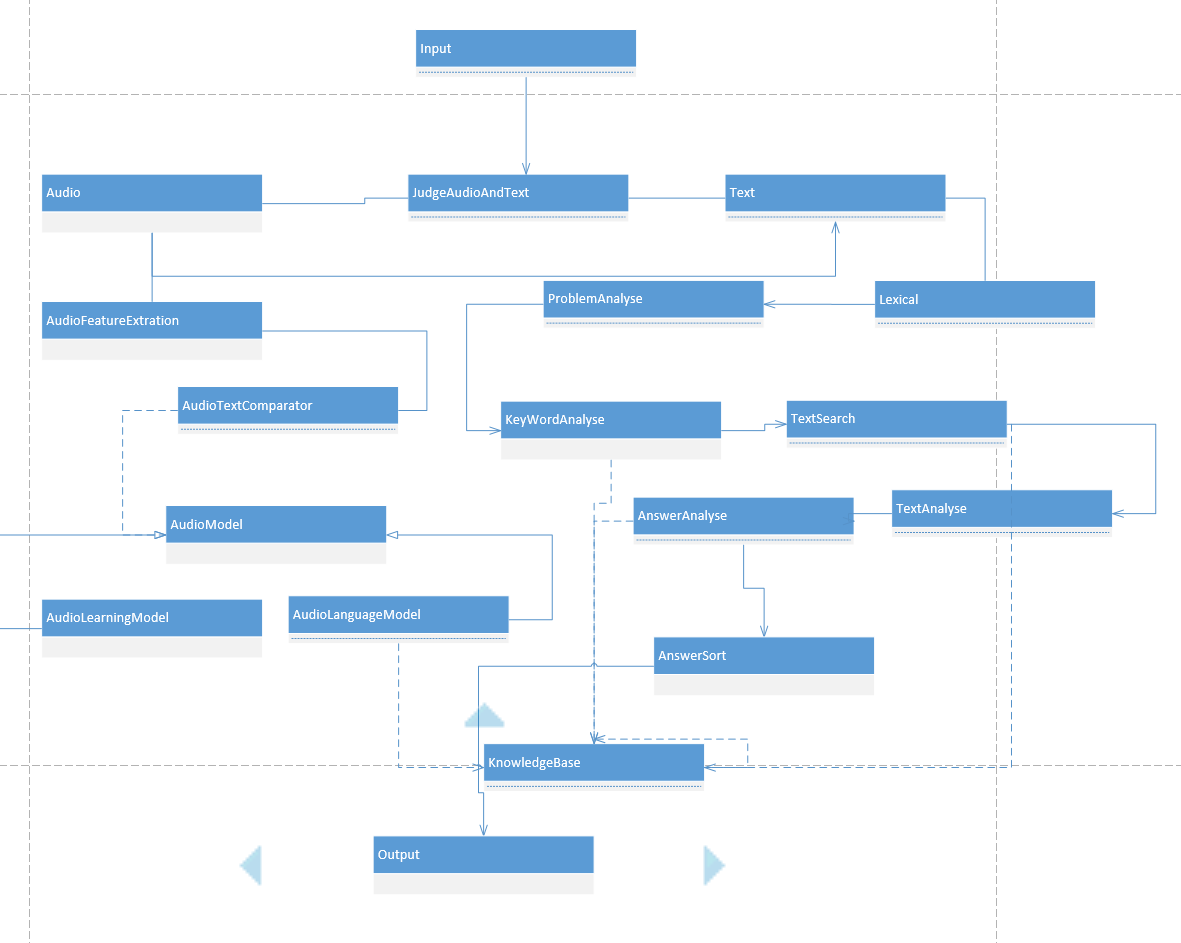


### 4.1.3状态图



### 4.1.4类图

以下类图是从系统层面的类图，是一个具有象征意义的大类。



## 4.2功能分配

说明需求文档当中各项功能要求同总体结构的对应关系。

## 4.3关键问题及解决方案

给出系统总体结构中所涉及的关键问题的设计决策和解决思路。

### 4.3.1问题分析模块及解决方案

由于在用户输入问题中，并非所有的词都能同等的表达相同重要性的含义，因此需要对问题进行预处理，并为后续的问题分类和关键词提取等提供依据。预处理的功能由自然语言处理模块完成，该模块一般包括：中文词法分析模块、命名实体识别模块和句法分析模块。

1. 实现中文词法分析、命名实体识别和句法分析等自然语言处理模块

对用户问句进行分词、词性标注、命名实体识别和句法分析等自然语言处理，可以为问题分类提供依据，同时也使从问句中提取关键词、扩展关键词成为可能。

1. 实现问题分类模块

将用户输入的问题按不同的分类标准归入不同的类别，可以为问题提取关键词提供依据，同时也确定了答案抽取的基本对象。

1. 实现问题关键词提取

根据问题的类别，和自然语言处理的结果，将问题中的关键词提取出来，经过扩展之后构成信息检索的检索词，同时也成为答案排序时的依据。

1. 实现问题关键词的扩展

之所以需要扩展关键词，是因为在正确的答案句子中，很多关键词并不是问题中的关键词，而是问题关键词的同义词、近义词甚至是相关词扩展。

1. 实现查询式的生成

结合一定的启发式规则生成合适的查询，以提高检索模块的召回率。

### 4.3.2问题分类模块及其解决方案

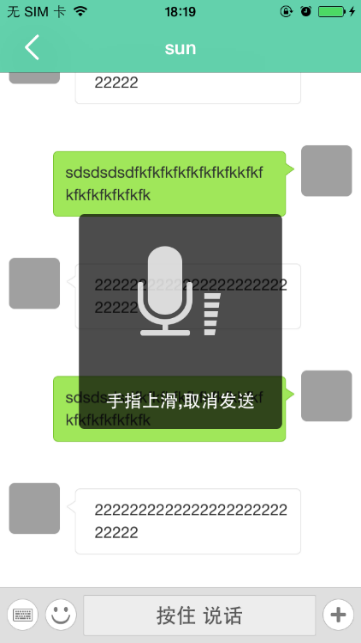
问题分类是问题分析模块的核心，其目的是根据预期的答案类型把问题划分到相应的语义类别，对后续的答案抽取和选择有重要的导向作用。首先，它有效减少了候选答案的空间，其次，决定了答案抽取的策略。该模块的好坏直接影响着问答系统的性能。但是，在一般的中文问答系统中，问题分类是很复杂也是难度很大的领域之一。这主要归因于它与文本分类的两点不同：一、一个问题往往只有一句话，可获取的上下文信息量很少，从而能用于问题分类的特征很有限；二、许多问题并不能严格划分到明确的领域，限制了问题分类的准确性。

# 5. 接口设计

## 5.1用户界面设计

给出系统用户界面的总体设计决策，和典型的用户界面风格。

说明：如果系统提供了对其他系统的接口，如从其他软件系统导入、导出数据，必须在此说明。

图为用户界面友好的聊天界面，支持用户输入模块(包含输入框、录音、表情选择、拍照、从相册选取照片功能)、录音模块。用户可以通过界面发送文字、语音、图片等，进行使用自动问答系统的提问功能和聊天功能。

## 5.2外部接口设计.

对系统的软件和硬件外部接口进行说明。

## 5.2.1用户登录

|  |  |  |  |
| --- | --- | --- | --- |
| 接口类型 | URL | | |
| 接口名称 | Login | | |
| 接口方向 | 网站后台-->手机端 | | |
| 接口说明 | 用户登录 | | |
|  | **参数名称** | **参数类型** | **参数说明** |
| 输入值类型 | Key | String | Key密钥保护数据 |
| Type | String | 调用接口的名称 |
| Uid | String | 用户名 |
| Pwd | String | 密码 |
| 返回值类型 | 返回结果具体结构如下 | | |
| result | Int | 200成功300失败 |
| 说明 | 返回JSON格式数据 | | |
| 返回值范例 | {  "result": 200,  } | | |

## 5.2.2用户注册

|  |  |  |  |
| --- | --- | --- | --- |
| 接口类型 | URL | | |
| 接口名称 | MeInsert | | |
| 接口方向 | 网站后台-->手机端 | | |
| 接口说明 | 添加会员 | | |
|  | **参数名称** | **参数类型** | **参数说明** |
| 输入值类型 | Key | String | Key密钥保护数据 |
| Type | String | 调用接口的名称 |
| Uid | String | 用户名 |
| Pwd | String | 密码 |
| Phone | String | 手机 |
| QQ | String | QQ |
| Mail | String | 邮箱 |
| 返回值类型 | 返回结果具体结构如下 | | |
| result | Int | 200成功300失败 |
| 说明 | 返回JSON格式数据 | | |
| 返回值范例 | {  "result": 200,  } | | |

## 5.2.3获取信息列表

|  |  |  |  |
| --- | --- | --- | --- |
| 接口类型 | URL | | |
| 接口名称 | MesDetail | | |
| 接口方向 | 网站后台-->手机端 | | |
| 接口说明 | 查看商家信息列表 | | |
|  | **参数名称** | **参数类型** | **参数说明** |
| 输入值类型 | Key | String | Key密钥保护数据 |
| Type | String | 调用接口的名称 |
| Field | string | 排序字段（默认“发布时间”） |
| OrderType | String | Desc降序 asc升序 |
| Page | Int | 页数 |
| PageSize | int | 页大小（即返回数据条数） |
| 返回值类型 | 返回结果具体结构如下 | | |
| result | Int | 200成功300失败 |
| PageCount | Int | 总页数 |
| 用户名 | String | 商家的用户名 |
| 店名 | String | 店名 |
| 简介 | String | 简介 |
| 内容 | String | 内容 |
| 价格 | String | 价格 |
| 发布时间 | String | 发布时间 |
| 分类名称 | String | 分类名称 |
| 类型名称 | String | 类型名称 |
| 图片 | String | 图片 |
| 已售 | String | 已售 |
| rowNum | Int | 序号 |
| 说明 | 返回JSON格式数据 | | |
| 返回值范例 | {  "result": 200,  "PageCount": 5,  "MerList": [  {  "rowNum":"1"  "用户名": "用户名",  "店名": "店名",  "简介": "简介",  " 内容": " 内容",  "价格": "价格",  "发布时间": "发布时间",  "分类名称": "分类名称",  "类型名称": "类型名称",  "图片": "图片",  "已售": "已售"  }, {  "rowNum":"2"  "用户名": "用户名",  "店名": "店名",  "简介": "简介",  " 内容": " 内容",  "价格": "价格",  "发布时间": "发布时间",  "分类名称": "分类名称",  "类型名称": "类型名称",  "图片": "图片",  "已售": "已售"  }  ]  } | | |

## 5.2.4提交信息

|  |  |  |  |
| --- | --- | --- | --- |
| 接口类型 | URL | | |
| 接口名称 | OrderDown | | |
| 接口方向 | 网站后台-->手机端 | | |
| 接口说明 | 提交订单 | | |
|  | **参数名称** | **参数类型** | **参数说明** |
| 输入值类型 | Key | String | Key密钥保护数据 |
| Type | String | 调用接口的名称 |
| Josn | String | Josn字符串 |
| 返回值类型 | 返回结果具体结构如下 | | |
| result | Int | 200成功300失败 |
| 说明 | 返回JSON格式数据 | | |
| 返回值范例 | {  "result": 200,  } | | |

## 5.2.5用户获取明细

|  |  |  |  |
| --- | --- | --- | --- |
| 接口类型 | URL | | |
| 接口名称 | UserVoucherMake | | |
| 接口方向 | 网站后台-->手机端 | | |
| 接口说明 | 查看商家列表 | | |
|  | **参数名称** | **参数类型** | **参数说明** |
| 输入值类型 | Key | String | Key密钥保护数据 |
| Type | String | 调用接口的名称 |
| MxID | String | 明细ID |
| 返回值类型 | 返回结果具体结构如下 | | |
| result | Int | 200成功300失败 |
| PageCount | Int | 总页数 |
| 说明 | 返回JSON格式数据 | | |
| 返回值范例 | {  "result": 200,  "PageCount": 5,  "MerList": [  {  ]  } | | |

## 5.3内部接口设计

对系统各构件（模块）之间的接口进行说明。

# 6.数据结构设计

## 6.1公共数据结构设计

对程序当中所定义的全局变量、常量、全局数据结构或类当中Public数据结构进行定义和说明。

Trie树（字典树）为前缀树，是为了支持泛型、遍历、储存、载入，为自然语言处理中最常用的高效检索数据结构。

## 6.2数据库设计

对程序当中所采用的数据库相关的名称和标识符、在数据库当中的位置、定义、度量单位、格式和值域、敏感程度、数据项名、缩写词和代码，包括用于数据库的规格说明等进行定义。建议给出数据库表结构。

说明：数据库表结构可以参考如下，字段值域不够可以自己增加。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库的表结构 | 表名称 | 设计者 | 审核者 | 完成日期 |
|  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 数据项名 | 索引或键 | 备注 | 缩写词 |
|  |  |  |  |  |  |  |  |

文档数据库

主要使用文档进行初步存储。其中词典有两个形态：文本文件(filename.txt)和缓存文件(filename.txt.bin或filename.txt.trie.dat和filename.txt.trie.value)。

文本文件

采用明文储存，UTF-8编码，CRLF换行符。

缓存文件

就是一些二进制文件，通常在文本文件的文件名后面加上.bin表示。有时候是.trie.dat和.trie.value。后者是历史遗留产物，分别代表trie树的数组和值。

修改了任何词典，只有删除缓存才能生效。

词典说明：

词典分为词频词性词典和词频词典。

词频词性词典

每一行代表一个单词，格式遵从[单词] [词性A] [A的频次] [词性B] [B的频次] 。

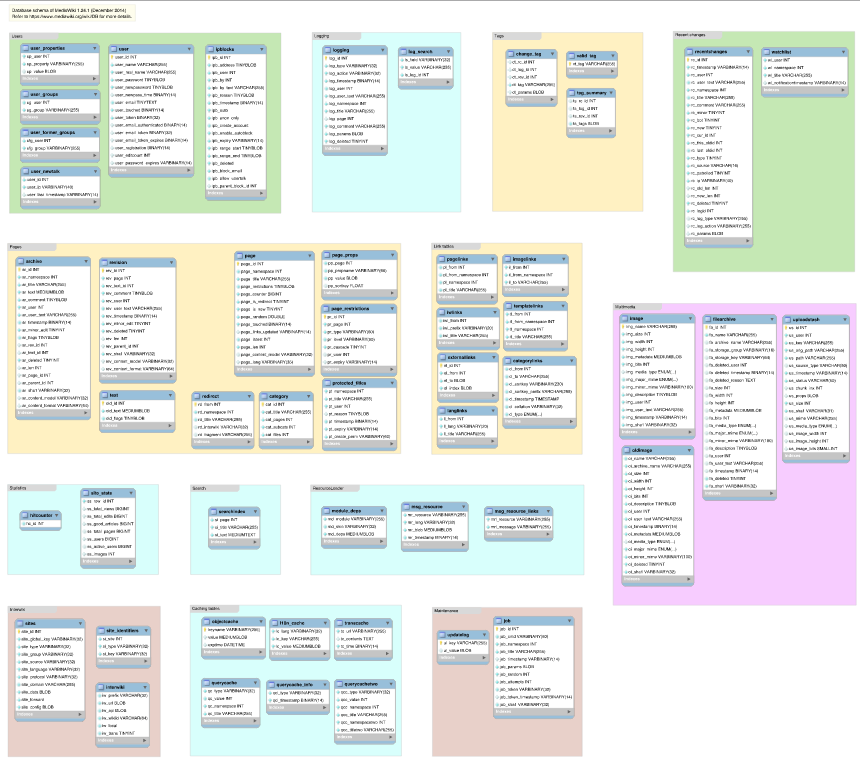
词频词典

每一行代表一个单词，格式遵从[单词] [单词的频次]。

每一行的分隔符为空格符或制表符

少数词典有自己的专用格式，比如同义词词典兼容《同义词词林扩展版》的文本格式，而转移矩阵词典则是一个csv表格。

redis数据库表的设计



# 7.详细设计

每一小节给出一个模块（构件）的详细设计方案。包括模块概述、模块的接口说明（即输入、输出）、以及内部结构设计。其中内部结构又可以考虑从静态、动态结构两个方面阐述；静态结构应给出该模块（构件）的类结构（类图），动态结构应给出该模块关键业务流程的交互模型（顺序图），还可根据实际情况给出状态图（某个构件或对象的状态迁移）和活动图（某个算法的实现流程）等内容。

## 7.1预处理

**自然语言处理模块**

由于在用户输入问题中，并非所有的词都能同等的表达相同重要性的含义，因此需要对问题进行预处理，并为后续的问题分类和关键词提取等提供依据。

预处理的功能由自然语言处理模块完成，该模块一般包括：中文词法分析模块、命名实体识别模块和句法分析模块。

**中文词法分析**

词是承载语义最小的语言单位，但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，所以我们需要将连续的字串按照一定的规范重新组合成词，即所谓的“分词”。词法分析的任务正是对用户问题进行分词、未登录词识别、词性标注等处理。

现有的中文分词算法主要分为四大类：基于字符串匹配的分词方法、基于理解的分词方法、基于统计的分词方法和基于语义的分词方法[43,44]。

一、基于字符串匹配的分词方法。

又称为机械分词方法，或者基于字典的分词方法。其基本原理是按一定的策略将待分析的“汉字串”与一个“充分大的”汉字字典进行匹配，若在字典中匹配到该词，则分词成功。这个方法的三个关键是字典、匹配顺序、匹配策略。根据这三个关键环节的不同，衍生出最大匹配算法（MM），逆向最大匹配法（RMM），最佳匹配法（OM）等等。

二、基于理解的分词方法

又称为基于人工智能的分词方法，基本思想是在分词的同时进行句法、语义分析，利用句法、语义信息来处理歧义现象。通常包含三个部分：分词子系统、句法语义子系统、总控部分。在总控部分的协调下，分词子系统可以获得有关词、句子等的句法和语义信息来对分词歧义进行判断，即它模拟了人对句子的理解过程。但是这种方法存在对句法及语义分析方法的重大依赖，而由于这两项技术目前还不是很成熟，使得这种分词方法还处于研究阶段。目前基于理解的分词方法主要有专家系统分词系统和神经网络分词系统。

三、基于统计的分词方法

该方法的主要思路是：形式上，词是稳定的字的组合，因此在上下文中，相邻的字同时出现的次数越多，就越有可能构成一个词。因此，字和字相邻出现的频率可以反映字之间构成词的可信度，在大量的文档集中通过统计各个字组合出现的频率，可能发现并实现分词。目前应用这种方法的分词算法有：N元语法模型、隐马尔可夫模型和最大熵模型等。这种方法的优点在于不需要大规模的字典，善于发现新词怪词。缺点也很明显，例如容易把一些经常一起出现但不是词误认为是词，对于字典中的常用词精确率也不如采用字典的方法。

四、基于语义的分词方法

语义分析法引入了语义分析，对自然语言本身进行更多的处理，例如扩充转移网络法、知识分词语义分析法、邻接约束法、综合匹配法、后缀分词法、特征词库法等。语义分析目前还在研究中。

词性标注是将特定类别的词用特定的符号标识的过程。经典的实现算法有：基于规则的方法、基于统计的方法、规则和统计相结合的方法及基于神经网络和遗传算法的方法等。基于规则的方法利用语言学制定的内省规则，根据上下文确定词的词性[44]；基于统计的方法通过计算所输入词串所有可能词性标注的概率，选择概率最大的标注串作为词性标注；规则和统计相结合的方法将前两种方法进行融合，一般是先规则过滤，再统计确认，或者先统计，再用规则过滤；基于神经网络和遗传算法的方法模拟人脑识别机制，同时进行上下文知识及逻辑推理[45]，目前该法仍处于研究阶段。

由于时间和效率的关系，本系统没有开发自己的词法分析系统，而是使用了由中科院开发的ICTCLAS汉语分词系统，该系统具有中文分词、词性标注、命名实体识别、未登录词识别等功能，同时支持行业词典及用户自定义词典。ICTCLAS采用层叠隐马尔可夫模型（Hierarchical Hidden Markov Model），将汉语词法分析的所有环节都统一到了一个完整的理论框架中，获得了总体效果最优，分词速度单机996KB/s，分词精度98.45%。在国内973专家组组织的评测活动中，ICTCLAS获得了第一名，在第一届国际中文处理研究机构SigHan组织的评测中也获得了多项第一名。

**命名实体识别**

对中文问答系统来说，命名体识别模块对于问题分析的重要性体现在以下两个方面：

1、提高问答系统中信息检索模块的性能。因为，索引的粒度越大，精确匹配的可能性就越大。实验证明，查询语句中名词短语的识别有助于提高系统的准确率和召回率。

2、是问题分析阶段其他子模块的基础。实现命名实体识别，是词性标注、句法分析的基础，只有更好更准确的命名实体识别，才能使词性标注等后续的自然语言处理任务更有意义。

对于命名体识别的主要方法，国外的研究和国内的研究大相径庭，这是由英文与中文的语言差异造成的。由于英文不需要分词，难度相对要较小一些。主要的方法从最初的基于规则，到后来基于统计，再到目前的以机器学习为主流的方法，经历了长久的发展[44]。

国内进行命名实体识别主要分为两个方向：一种策略是将命名实体分为人名，地名，机构名分别进行识别，该方法充分考虑各类命名实体的特点，并针对不同的类别采取不同识别策略；另外一种是对命名实体识别，采用一体化的方法，对命名实体进行整体识别。目前，将命名实体按照人名，地名，机构名进行划分，分别进行识别的方法在研究中出现较多，占主导地位。

**句法分析**

句法分析是自然语言处理中的中心问题之一，在自动问答、机器翻译、信息检索、信息抽取等领域中有重要的应用。对问题进行句法分析的目的是获取句子的语法结构，即句子所包含的句法单位和这些句法单位之间的关系[46]，进一步为答案抽取提供依据。其中，基于依存文法的句法分析受到越来越多的关注。本系统中也采用了基于依存文法的汉语句法分析器，由斯坦福大学开发。依存文法是由法国语言学家L.Tesiniere在1959年提出来的，该文法认为句子中述语动词是支配其它成分的中心，而它本身却不受其它任何成分的支配，所有的受支配成分都以某种依存关系从属于其支配者。依存文法的句法结构的主要元素是依存关系（dependency relationship），即句子中词对的二元关系，其中一个记为支配词，另一个记为从属词。