**技术文档**

**功能一：**语音识别

**功能描述：**模拟人机对话，并在特定词的触发下退出该功能

**引用模块：**

Baidu-aip

requests

json

speechRecognition

图灵机器人api：<http://openapi.tuling123.com/openapi/api/v2>

**关键部分代码：**

# 将语音转文本STT

def listen():

    # 读取录音文件

    with open(path, 'rb') as fp:

        voices = fp.read()

    try:

        # 参数dev\_pid：1536普通话(支持简单的英文识别)、1537普通话(纯中文识别)、1737英语、1637粤语、1837四川话、1936普通话远场

        result = client.asr(voices, 'wav', 16000, {'dev\_pid': 1537, })

        # result = CLIENT.asr(get\_file\_content(path), 'wav', 16000, {'lan': 'zh', })

        # print(result)

        # print(result['result'][0])

        # print(result)

        result\_text = result["result"][0]

        print("you said: " + result\_text)

        return result\_text

    except KeyError:

        print("KeyError")

        speaker.Speak("我没有听清楚，请再说一遍...")

**功能二：**通过关键词查询查询书籍名称

**功能描述：**对书名进行分词，匹配用户输入的关键词，处理得到相关度，降序返回书名列表 **引用模块：**

jieba,

random

py2neo

re

**关键部分代码：**

def book\_name\_search(elementDict,input):

    '''

    朴素相似度比较，通过检查书名中的关键词在检索内容中是否出现，得到(0,n,1)/length的相似度

    '''

    rel\_dict = {}

    for n in elementDict.keys():

        rel\_list = elementDict[n]

        length = len(rel\_list)

        rel\_val = 0 #定义相关度

        for i in rel\_list:

            if  re.search(i,input,re.IGNORECASE) or re.search(input,i,re.IGNORECASE):

                rel\_val += 1

        if rel\_val != 0:

            rel\_dict[n] = rel\_val/length

        else:

            continue

    res\_raw = sorted(zip(rel\_dict.values(), rel\_dict.keys()))

    res\_tag = res\_raw[::-1]

    res = []

    for i in res\_tag:

        res.append(find\_book(i[1]))

    return res

**功能三：**藏书馆节点查询

**功能描述：**输入馆藏地的名称，返回该馆藏点的全部书籍(默认选择15本) **引用模块：**同上

**关键部分代码：**

def site\_storage(siteName,src=graph,Num=15):

    '''

    查询某一节点所有图书，并以元组形式逐条返回。(generator mode)

    Num是返回数目上限，默认返回15本书

    '''

    site = find\_site(siteName, output=None)

    tmp = []

    try:

        for n in src.match(end\_node=site,rel\_type='藏于'):

            startNode = n.start\_node()#字典形式储存图书的信息

            tmp.append(dict(startNode))

        minNum = min(len(tmp),Num)

        random.shuffle(tmp)

        res = tmp[:minNum]

        return res

    except:

        return '输入有误，查询失败'