《人机交互技术》课程实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业名称 | 计算机科学与技术(非师范) | 年级 | 2019 | 班级 | 计工本1901 |
| 学生姓名 | 隋远 | 指导老师 | 杜萍 | 时间 | 2021/11/05 |

|  |  |
| --- | --- |
| 实验名称 | 掌握实现语音合成技术 |
| 实  验  目  的  及  要  求 | 目的：了解语音交互的基本原理，掌握自然语音交互中语音识别功能的实现方法；  要求：熟悉和掌握声音文件的录制和保存的基本过程；能够搭建智能开放平台的基本编程环境；实现语音识别功能的开发及演示。 |
| 实  验  环  境 | 具有登陆互联网功能的Windows 10操作系统，Python开发环境，AI开放平台系统语音识别模块。 |
| 实  验  内  容 | 百度短语音识别可以将 60 秒以下的音频识别为文字。适用于语音对话、语音控制、语音输入等场景。  接口类型：通过 REST API 的方式提供的通用的 HTTP 接口。适用于任意操作系统，任意编程语言  接口限制：需要上传完整的录音文件，录音文件时长不超过 60 秒。浏览器由于无法跨域请求百度语音服务器的域名，因此无法直接调用API接口。  支持音频格式：pcm、wav、amr、m4a  音频编码要求：采样率 16000、8000，16 bit 位深，单声道 |
| 实  验  步  骤  或  实  验  方  案 | 调用流程：  1. 创建账号及应用： 在 ai.baidu.com 控制台中，创建应用，勾选开通 ”语音技术“-”短语音识别、短语音识别极速版“ 能力。获取AppID、API Key、Secret Key，并通过请求鉴权接口换取 token ，详细见 “接入指南”。  2. 创建识别请求： POST 方式，音频可通过 JSON 和 RAW 两种方式提交。JSON 方式音频数据由于 base64 编码，数据会增大1/3。其他填写具体请求参数 ，详见 ”请求说明“。  3. 短语音识别请求地址： http://vop.baidu.com/server\_api  4. 返回识别结果：识别结果会即刻返回，采用 JSON 格式封装，如果识别成功，识别结果放在 JSON 的 “result” 字段中，统一采用 utf-8 方式编码。详见 ”返回说明“。 |
| 调  试  过  程  及  实  验  结  果 | 1. 从网页中申请的应用获取appKey和appSecret    2. 设置要识别的文件，文件格式    3. 设置采样率    4. JSON格式POST上传本地文件    5. 返回结果    若识别成功返回case    若识别失败返回case    6. 测试结果    正确识别音频信号“北京科技馆”。 |
| 总  结 | 借助百度开放云平台，调用API接口，上传JSON格式音频文件，进行语音识别。 |
| 附  录 | # coding=utf-8  import sys  import json  import base64  import time  IS\_PY3 = sys.version\_info.major == 3  if IS\_PY3:  from urllib.request import urlopen  from urllib.request import Request  from urllib.error import URLError  from urllib.parse import urlencode  timer = time.perf\_counter  else:  from urllib2 import urlopen  from urllib2 import Request  from urllib2 import URLError  from urllib import urlencode  if sys.platform == "win32":  timer = time.clock  else:  # On most other platforms the best timer is time.time()  timer = time.time  # API\_KEY = 'kVcnfD9iW2XVZSMaLMrtLYIz'  # SECRET\_KEY = 'O9o1O213UgG5LFn0bDGNtoRN3VWl2du6'  API\_KEY = '4QSHUs3o7KPGV6UWqw3fKYQf'  SECRET\_KEY = 'C4BMG8ngBNcKzzBSkwVRluZZQ2kkQIND'  # 需要识别的文件  AUDIO\_FILE = './audio/16k.pcm' # 只支持 pcm/wav/amr 格式，极速版额外支持m4a 格式  # 文件格式  FORMAT = AUDIO\_FILE[-3:] # 文件后缀只支持 pcm/wav/amr 格式，极速版额外支持m4a 格式  CUID = '123456PYTHON'  # 采样率  RATE = 16000 # 固定值  # 普通版  DEV\_PID = 1537 # 1537 表示识别普通话，使用输入法模型。根据文档填写PID，选择语言及识别模型  ASR\_URL = 'http://vop.baidu.com/server\_api'  SCOPE = 'audio\_voice\_assistant\_get' # 有此scope表示有asr能力，没有请在网页里勾选，非常旧的应用可能没有  #测试自训练平台需要打开以下信息， 自训练平台模型上线后，您会看见 第二步：“”获取专属模型参数pid:8001，modelid:1234”，按照这个信息获取 dev\_pid=8001，lm\_id=1234  # DEV\_PID = 8001 ;  # LM\_ID = 1234 ;  # 极速版 打开注释的话请填写自己申请的appkey appSecret ，并在网页中开通极速版（开通后可能会收费）  # DEV\_PID = 80001  # ASR\_URL = 'http://vop.baidu.com/pro\_api'  # SCOPE = 'brain\_enhanced\_asr' # 有此scope表示有极速版能力，没有请在网页里开通极速版  # 忽略scope检查，非常旧的应用可能没有  # SCOPE = False  class DemoError(Exception):  pass  """ TOKEN start """  TOKEN\_URL = 'http://openapi.baidu.com/oauth/2.0/token'  def fetch\_token():  params = {'grant\_type': 'client\_credentials',  'client\_id': API\_KEY,  'client\_secret': SECRET\_KEY}  post\_data = urlencode(params)  if (IS\_PY3):  post\_data = post\_data.encode( 'utf-8')  req = Request(TOKEN\_URL, post\_data)  try:  f = urlopen(req)  result\_str = f.read()  except URLError as err:  print('token http response http code : ' + str(err.code))  result\_str = err.read()  if (IS\_PY3):  result\_str = result\_str.decode()  print(result\_str)  result = json.loads(result\_str)  print(result)  if ('access\_token' in result.keys() and 'scope' in result.keys()):  print(SCOPE)  if SCOPE and (not SCOPE in result['scope'].split(' ')): # SCOPE = False 忽略检查  raise DemoError('scope is not correct')  print('SUCCESS WITH TOKEN: %s EXPIRES IN SECONDS: %s' % (result['access\_token'], result['expires\_in']))  return result['access\_token']  else:  raise DemoError('MAYBE API\_KEY or SECRET\_KEY not correct: access\_token or scope not found in token response')  """ TOKEN end """  if \_\_name\_\_ == '\_\_main\_\_':  token = fetch\_token()  speech\_data = []  with open(AUDIO\_FILE, 'rb') as speech\_file:  speech\_data = speech\_file.read()  length = len(speech\_data)  if length == 0:  raise DemoError('file %s length read 0 bytes' % AUDIO\_FILE)  speech = base64.b64encode(speech\_data)  if (IS\_PY3):  speech = str(speech, 'utf-8')  params = {'dev\_pid': DEV\_PID,  #"lm\_id" : LM\_ID, #测试自训练平台开启此项  'format': FORMAT,  'rate': RATE,  'token': token,  'cuid': CUID,  'channel': 1,  'speech': speech,  'len': length  }  post\_data = json.dumps(params, sort\_keys=False)  # print post\_data  req = Request(ASR\_URL, post\_data.encode('utf-8'))  req.add\_header('Content-Type', 'application/json')  try:  begin = timer()  f = urlopen(req)  result\_str = f.read()  print ("Request time cost %f" % (timer() - begin))  except URLError as err:  print('asr http response http code : ' + str(err.code))  result\_str = err.read()  if (IS\_PY3):  result\_str = str(result\_str, 'utf-8')  print(result\_str)  with open("result.txt","w") as of:  of.write(result\_str) |