语音识别开发手册

(内部资料，请勿外传)

版本	修改内容	修改时间	修改人
V1.0	调整文档领域等信息	2018-07-13	全刚
V1.1	增加NeuHub域名	2018-08-02	全刚
V1.2	增加参数说明	2018-08-03	全刚
V1.3	增加音频格式参数并修改了一些描述错误	2018-09-20	崔午阳
V2.0	重新梳理文档架构	2018-09-27	崔午阳

接口描述

功能描述

语音转换成文字

能力说明

语音识别API根据不同的使用场景，使用在对应领域场景下训练的模型，以提高识别准确率

音频要求

> 1. 目前支持的音频格式：wav、mp3、amr
> 2. 目前支持的采样率：16000
> 3. 一次请求的音频最大时长：20秒

请求说明

HTTP调用方法

`POST`

请求URL

URL	说明
http://ai-api.jd.com/asr	线上NeuHub平台域名
http://asrapi-base.jd.com/asr	联调测试域名

请求参数

Header

- 示例：

```
Domain:search
Application-Id:9a2eaac5-4114-4edf-9036-89d1813b684d
Request-Id:56a847e6-84c0-4c01-bf4b-d566f2d2dd11
Sequence-Id:1
Asr-Protocol:0
Net-State:2
Applicator:0
Property:{
    "autoend" : false,
    "encode" : {
        "channel" : 1,
        "format" : "wav",
        "sample_rate" : 16000
    },
    "platform" : "Linux",
    "version" : "0.0.0.1"
}
```

- 字段说明：

字段	说明
Domain	领域。跟语音识别场景有关，目前可选值为： - search, - jingme_mia
Application-Id	产品ID。由Server端统一分配，用于在请求时做鉴权： - 申请方式：业务方发邮件给jdai-speech@jd.com - 测试token：9a2eaac5-4114-4edf-9036-89d1813b684d
Request-Id	请求语音串标识码。由客户端生成，代表完整的语音识别请求过程，需要注意： - 需要全局唯一 - 对于同一次请求Request-Id需要保持一致 - 每一次识别请求都要新生成Request-Id，若多次请求使用同一个将 会产生不可预知的错误。 生成方法: - libuuid 库可以直接生成。Android 及 Ios 也有相关的生成函数。
Sequence-Id	语音分段传输的分段号。从 1 开始，为负表示最后一段语音。例如：一次语音识别请求分10个请求，则Sequence-Id依次为：1,2,3,4,5,6,7,8,9,-10
Asr-Protocol	通信协议版本号，使用 0 即可
	客户端网络状态： - NETSTATE_NO_NETWORK = 0;

Net-State	- NETSTATE_NO_WIFI_MOBILE = 1;
	- NETSTATE_WIFI = 2;
	- NETSTATE_CDMA = 3;
	- NETSTATE_EDGE = 4; //移动 2.5G
	- NETSTATE_EVDO_0 =5;
	- NETSTATE_EVDO_A = 6;
	- NETSTATE_GPRS = 7;
	- NETSTATE_HSDPA = 8; //联通 3G
	- NETSTATE_HSPA = 9;
	- NETSTATE_HSUPA = 10;
	- NETSTATE_UMTS = 11;
	- NETSTATE_EHRPD = 12;
	- NETSTATE_EVDO_B = 13;
	- NETSTATE_HSPAP = 14;
	- NETSTATE_IDEN = 15;
	- NETSTATE_LTE = 16;
	- NETSTATE_UNKOWN_MOBILE = 20;

Applicator	应用者，SDK 会提供给不同的应用者(渠道，例如：内部业务(0)，外部业务(1))
*Property	包含跟语音识别相关的请求参数，例如采样率、音频格式等。此参数要在第一个请求包的时候传过来，即Sequence-Id为 1 或者 -1 (-1 表示一次语音识别请求只有一个http请求包的情况)

- Property参数

字段	说明
autoend	bool类型，是否开启服务端自动断尾，即服务端vad，默认填false
encode	语音识别的请求格式： - channel：int类型，音频声道数，目前只支持单声道，填1 - format：字符串类型，音频格式，目前支持wav，amr，mp3 - sample_rate：int类型，采样率，目前只支持16000
platform	字符串类型，各平台的机型信息，格式为：{平台}&{机型}&{系统版本号} - 平台：Android，iOS，Linux，windows - 机型：设备的机型名称，如GalaxyNexus，iPhoneX等 - 系统版本号：设备系统版本，如2.3.3，4.2.1等 - 示例：Android&Pixel&7.1；iOS&iPhoneX&11.4.1
version	字符串类型，客户端版本号

请求示例

发送第一包数据需要上传Property：

```
Host: aiasr.jd.com
Accept: */*
Domain:search
Application-Id:9a2eaac5-4114-4edf-9036-89d1813b684d
Request-Id:56a847e6-84c0-4c01-bf4b-d566f2d2dd11
Sequence-Id:1
Asr-Protocol:0
Net-State:2
Applicator:0
Property:{"autoend":false,"encode":{"channel":1,"format":"mp3","sample_rate":16000
},"platform":"Linux","version":"0.0.0.1"}
Content-Length: 3200
Content-Type: application/x-www-form-urlencoded
Expect: 100-continue
```

发送其他数据包或最后一包，不需要上传Property，最后一包seq_id取负值：

```
Host: aiasr.jd.com
Accept: */*
Domain:search
Application-Id:9a2eaac5-4114-4edf-9036-89d1813b684d
Request-Id:56a847e6-84c0-4c01-bf4b-d566f2d2dd11
Sequence-Id:-2
Asr-Protocol:0
Net-State:2
Applicator:0
Content-Length: 3200
Content-Type: application/x-www-form-urlencoded
Expect: 100-continue
```

HTTP调用示例代码

C 示例代码：

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include<sys/time.h>
#include "curl/curl.h"
#include "curl/easy.h"
#include "uuid/uuid.h"

#define MAX_BUFFER_SIZE 512
```

```
#define MAX_BODY_SIZE 1000000

int g_package_size = 32000;
struct timeval g_start, g_end;

static size_t writefunc(void *ptr, size_t size, size_t nmemb,
                        char **result) {
    gettimeofday(&g_end, NULL);
    printf("recv-send=%ld ms\n", (g_end.tv_sec-g_start.tv_sec)*1000+\
            (g_end.tv_usec-g_start.tv_usec)/1000);

    size_t result_len = size * nmemb;
    *result = (char *)realloc(*result, result_len + 1);
    if (*result == NULL) {
        printf("realloc failure!\n");
        return 1;
    }
    memcpy(*result, ptr, result_len);
    (*result)[result_len] = '\0';
    printf("%s\n", *result);
    return result_len;
}

static void sendReq(char *domain, char *appid, char *reqid, int seqid,
                    char *property, char *audiodata, int content_len) {
    char host[MAX_BUFFER_SIZE];
    memset(host, 0, sizeof(host));
    snprintf(host, sizeof(host), "%s", "aiasr.jd.com/voice");
    char tmp[MAX_BUFFER_SIZE];
    memset(tmp, 0, sizeof(tmp));

    CURL *curl;
    CURLcode res;
    char *resultBuf = NULL;
    struct curl_slist *headerlist = NULL;
    snprintf(tmp, sizeof(tmp), "Domain:%s", domain);
    headerlist = curl_slist_append(headerlist, tmp);
    snprintf(tmp, sizeof(tmp), "Application-Id:%s", appid);
    headerlist = curl_slist_append(headerlist, tmp);
    snprintf(tmp, sizeof(tmp), "Request-Id:%s", reqid);
    headerlist = curl_slist_append(headerlist, tmp);
    snprintf(tmp, sizeof(tmp), "Sequence-Id:%d", seqid);
    headerlist = curl_slist_append(headerlist, tmp);
    snprintf(tmp, sizeof(tmp), "Asr-Protocol:%d", 0);
    headerlist = curl_slist_append(headerlist, tmp);
    snprintf(tmp, sizeof(tmp), "Net-State:%d", 2);
    headerlist = curl_slist_append(headerlist, tmp);
    snprintf(tmp, sizeof(tmp), "Applicator:%d", 0);
```

```
    headerlist = curl_slist_append(headerlist, tmp);
    if (seqid==-1||seqid==1) {
        snprintf(tmp, sizeof(tmp), "Property:%s", property);
        headerlist = curl_slist_append(headerlist, tmp);
    }
    curl = curl_easy_init();
    curl_easy_setopt(curl, CURLOPT_URL, host);
    curl_easy_setopt(curl, CURLOPT_POST, 1);
    curl_easy_setopt(curl, CURLOPT_TIMEOUT, 3000);
    curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headerlist);
    curl_easy_setopt(curl, CURLOPT_POSTFIELDS, audiodata);
    curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE, content_len);
    curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, writefunc);
    curl_easy_setopt(curl, CURLOPT_WRITEDATA, &resultBuf);
    gettimeofday(&g_start, NULL);
    res = curl_easy_perform(curl);
    if (res != CURLE_OK) {
        printf("perform curl error:%d.\n", res);
        return ;
    }
    curl_slist_free_all(headerlist);
    curl_easy_cleanup(curl);
}

int main (int argc,char* argv[]) {
    if (argc != 4) {
        printf("Usage: %s domain package_size audio_file\n", argv[0]);
        return -1;
    }

    FILE *fp = NULL;
    fp = fopen(argv[3], "r");
    if (NULL == fp) {
        return -1;
    }
    fseek(fp, 0, SEEK_END);
    int content_len = ftell(fp);
    fseek(fp, 0, SEEK_SET);
    char *audiodata = (char *)malloc(content_len);
    fread(audiodata, content_len, sizeof(char), fp);

    char *domain = argv[1];
    char reqid[64];
    uuid_t uuid;

    uuid_generate(uuid);
    uuid_unparse(uuid, reqid);
```

```
    char tmp[MAX_BUFFER_SIZE];
    memset(tmp, 0, sizeof(tmp));
    snprintf(tmp, sizeof(tmp), "%s", "{\"autoend\":false,\"encode\":{\
            "channel\":1,\"format\":\"wav\",\"sample_rate\":16000},\
            "platform\":\"Linux\",\"version\":\"0.0.0.1\"}");

    g_package_size = atoi(argv[2]);
    if (g_package_size <= 0) {
        printf("package_size <= 0!\n");
        return 0;
    }

    char *package_data = (char *)malloc(g_package_size);
    int package_len = g_package_size;
    int seqid = 1;

    for (int pos=0;pos<content_len;pos+=g_package_size,++seqid)
    {
        package_len = g_package_size<content_len-pos?
                        g_package_size:content_len-pos;
        usleep(1000*package_len/32);
        seqid = g_package_size<content_len-pos?seqid:-seqid;
        memcpy(package_data, audiodata+pos, package_len);
        sendReq(domain, "linux_demo", reqid, seqid, tmp,
                package_data, package_len);
    }

    fclose(fp);
    free(audiodata);
    free(package_data);
    return 0;
}
```

返回说明

返回参数

服务器返回的识别结果采用json格式：

字段名	说明
request_id	请求id，一次请求相同
err_no	错误码，0表示识别正确
index	返回结果编号，负数表示全部最终结果
err_msg	错误码信息
content	text：识别结果的文本内容

返回示例

```
{
    "request_id":"c3e2-d139-67e7-f4d7-e64b",
    "err_no": 0,
    "index": -44,
    "err_msg": "",
    "content": [{
        "text": "今天天气 怎么样",
    }]
}
```

错误码

错误编号	content	错误信息
0	{识别结果}	正常
-7001	/	语音数据为空
-7002	/	语音数据过长，一次请求音频不能超过一分钟
-7003	/	语请求参数出错，缺少Request-Id、Sequence-Id或Application-Id等。
-7004	/	音频头部格式解析错误，例如wav头部
-7005	/	音频采样率或通道数错误
-7006	/	音频格式错误
-8001	/	服务内部音频解码错误
-8002	/	服务内部am-server模块错误
-9001	/	服务内部decoder-server模块错误
-9002	/	服务内部语音识别解码失败