Speech Recognition API

Speech recognition (ASR) is based on a deep convolutional neural network to convert long audio files into text.
The recognition service currently provided is used for recorded audio (not real-time).
After the audio file is successfully uploaded, it enters the waiting queue. After the recognition is successful, the user can obtain the result.
The time to return the result is affected by the length of the audio file and the amount of queued tasks.

API requirements

content	Description
Request Protocol	http[s]
Request address	http[s]: //lab_ip /api/xxx
Request method	POST
Encoding	UTF-8
Return Format	JSON
Audio file properties	Sample rate 16k、Mono & Multichannel

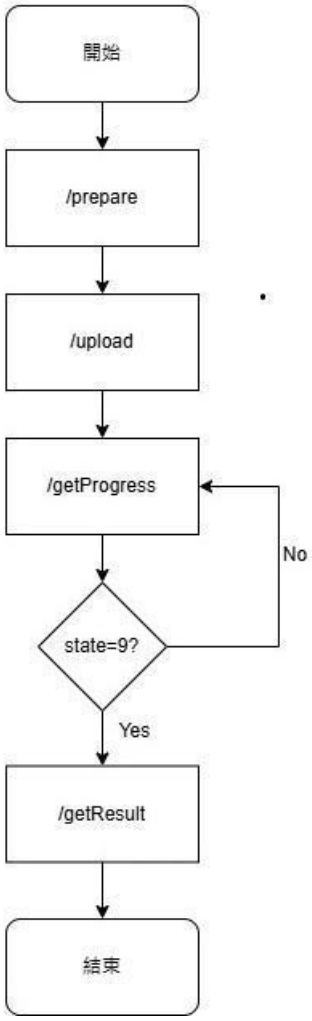
content	Description
file format	wav/flac/opus/m4a/mp3
file size	No more than 500M
Language	Chinese and English

API process:

The API includes the following interfaces: preprocessing, file upload, query processing progress, and get results.
Preprocessing: **/prepare**
File upload :**/upload**
Query processing progress: **/getProgress**
Get Result: **/getResult**

General return description:

parameter	type	description
ok	int	Call success flag (0: success, -1: failure)
err_no	int	Error Code
failed	string	Error description (null: no error)
data	string	Data (null: no return value)
task_id	string	Task id



1、Preprocessing API

#Overview

First, call the pre-processing API to upload the basic information of the audio file to be recognized (file name, size) and other related configurable parameters.

If the call succeeds, the task ID (task_id) is returned.

#URL

POST http[s]://**lab_ip** /api/prepare

 # header fields

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

#Parameter Description

parameter	type	essential	description	example
file_len	string	yes	File size (Byte)	160044
file_name	string	yes	File name	meeting_audio.wav
total_segments	int	yes	Number of segments	5
speaker_number	string	No	Number of speakers, optional value: 0-10, 0 means blind	default：2
has_seperate	string	No	Whether the recognition result contains speaker separation	false or true, default : false
language	string	No	Language default: Chinese and English	default
pd	string	No	Vertical field personalization parameters: court: court Education: edu Finance: finance	prepareParam.put("pd", "edu") pd is a non-essential parameter. If you do not set the parameter,

parameter	type	essential	description	example
			Medical: medical Technology: tech Sports:sports Government: gov Game: game E-commerce: ecom car: car	it will be set to general.
hotWord	string	No	Hot words (Use hot words to improve the recognition rate of professional vocabulary, pay attention to the following points: 1. Single hot word setting: hot word 2. Multiple hot word settings: hot word 1 | hot word 2 | hot word 3 3. The length of a single hot word should not exceed 16 4. The number of hot words is limited to 200	1、prepareParam.put("hotWord", "周杰倫") 2、prepareParam.put("hotWord", "周杰倫|郭台銘")

#Return Value

success:

{
 "ok":0,

"err_no":0,
"failed":null,
"data":"383e72a47557490aa05a344074117a9d"
}

failure:

{
"ok":-1,
"err_no":26000,
"failed":"llegal information",
"data":null
}

#Results description

If the call is successful, data is taskId (task ID), which is a required parameter for subsequent APIs.

#2. File upload API

#Overview

If preprocessing is successful, call the file upload API

#url

POST http[s]://lab_ip /api/upload

 # header fields

Content-Type: multipart/form-data;

#Parameter Description

parameter	type	essential	description	example	
task_id	string	Yes	Task ID	4b705edda27a4140b31b462df0033cfa	
segment_id	int	Yes	Segment order (1-n)	1	

parameter	type	essential	description	example	
segment_len	string	Yes	Segment size (Byte)	204800	
content	Byte data	Yes	Audio file content		

#Return Value

success:

```
{
    "ok":0,
    "err_no":0,
    "failed":null,
    "data":null
}
```

failure:

```
{
    "ok":-1,
    "err_no":26000,
    "failed":"Task ID does not exist",
    "data":null
}
```

#4. Query processing progress API

#Overview

After the caller sends a request to upload a file, the server has scheduled the task. Before getting the result, the caller needs to poll the API to query the current status of the task.

If and only if the task status = 9 (recognition result completed), caller can call the Get Result API to obtain the recognition result.

The polling strategy is determined by the caller, and it is recommended to poll every 10 minutes. See the appendix for status code descriptions.

#url

POST http[s]://lab_ip /api/getProgress

 # header fields

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

#Parameter Description

parameter	type	essential	description	example
task_id	string	Yes	Task ID (preprocessing API return value)	4b705edda27a4140b31b462df0033cfa

#Return Value

success:


```
{
  "ok": 0,
  "err_no": 0,
  "failed": null,
  "data": {
    "task_status": 2,
    "desc": "Audio recognition in progress",
    "segments": {
      "1": {"status": 9, "desc": "Recognition completed"},
      "2": {"status": 2, "desc": "Recognition in progress"}
    }
  }
}
```


failure:


```
{
    "ok":-1,
    "err_no":26640,
    "failed":"File upload failed",
    "data":null
}
```

5、Get result API

#Overview

When the task processing progress status = 9 (see the query processing progress API), call this API to obtain the speech recognition result. This is the final step in the speech recognition process.

For detailed description of each field of the recognition result, please refer to the recognition result description document.

Merge the recognition results of all segments in the order of segment_id and adjust the timestamps (bg and ed)

#url

POST http[s]://lab_ip/api/getResult

 # header fields

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

#Parameter Description

parameter	type	essential	description	example	
task_id	string	Yes	Task ID (preprocessing API return value)	4b705edda27a4140b31b462df0033cfa	

#Return Value

success:


```
{
    "ok":0,
    "err_no":0,
    "failed":null,
    "data":"[{\"bg\":\"0\",\"ed\":\"4950\",\"onebest\":\"Speech recognition results。
\",\"speaker\":\"0\"}]"
}
```


failure:{

```
{
    "ok":-1,
    "err_no":26000,
    "failed":"Illegal messages",
    "data":null
}
```


#Appendix

#recognition result description

Field Name	description
bg	The start time of the sentence relative to the audio, in ms
ed	The end time of the sentence relative to this audio, in ms
onebest	Sentence content
speaker	Speaker number, starting from 1. If speaker separation is not enabled, the speaker number is 0.

#Error code

Error handling will be developed in the next development schedule

Error code	description
0	success
26000	Common Errors

#Task status code

Status ID	description
0	Task created successfully
1	Audio upload completed
2	Audio recognition in progress
9	Recognition completed