

Intelligent Action Choreography

Intelligent Action Choreography

[Experiment Objective](#)

[Experiment Quick Start Steps](#)

[Experimental Results](#)

[Main Source Code Analysis](#)

[Overall flow chart for this experiment](#)

(Selected section)

[1. Customizing the English Version](#)

Experiment Objective

Use language to communicate with a smart car, enabling it to perform a specific action based on relevant semantics.

Experiment Quick Start Steps

1. Power on the computer, wait for the IP address to appear on the OLED screen, and then log in remotely via VNC to the Raspberry Pi desktop.
2. Open a new terminal and enter the command:

```
cd /home/pi/project_demo/09.AI_Big_Model/
```

3. Modify the file configuration:

```
gedit API_KEY.py
```

Set **AI_Agent = Flase**, as shown in the image.

```
1 #Whether to use local data or online data - Agent True: Online False: Local
2 AI_Agent = False #The English version needs no attention
3
```

4. Then, fill in **TONYI_KEY** for the Chinese version and **openAI_KEY** for the English version. Save and exit the **API_KEY.py** file, then run the following command:

```
#Start command for the Chinese version
python3 Car_Choreography/AIagent_main.py
```

```
#Start command for the English version
python3 Car_Choreography/AIagent_main_en.py
```

5. The car enters the wake-up state. For international users, use the wake-up word: "Hi, yahboom."
6. After successfully waking up, the car will respond with a horn sound. After waiting for 1 second, you can then state the desired action sequence.
7. The smart car will analyze the language based on semantics, execute a series of action commands, and provide some dialogue feedback.

8. This concludes the execution of the actions in the dialogue sequence. To continue the conversation, repeat steps 5-8.

Experimental Results

1. Waiting for wakeup

```
pi@yahboom:~/project_demo/09.AI_Big_Model $ python3 Car_Choreography/AIagent_main.py
serial /dev/myspeech open
start
Waiting for keyword...
```

2. Start recording

```
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
Current volume: 63034.0, boot threshold: 3000, End threshold: 1500
start recording
3000 63034.0
Current volume: 71399.0, boot threshold: 3000, End threshold: 1500
3000 71399.0
Current volume: 57223.0, boot threshold: 3000, End threshold: 1500
3000 57223.0
Current volume: 50633.0, boot threshold: 3000, End threshold: 1500
3000 50633.0
Current volume: 40563.0, boot threshold: 3000, End threshold: 1500
```

3. After speech recognition, provide speech feedback and execute corresponding action commands

```
Q: After turning the light red, it turns blue finally turn off a light.
Car Agent Start
A:First, I'll glow red like a traffic light, then turn blue like the sky, and finally, I'll turn off the lights. It's like a mini light show! Enjoy the show~
Start executing action Car_RGB_Control(255, 0, 0)
Waiting for keyword...
Start executing action time.sleep(1)
Start executing action Car_RGB_Control(0, 0, 255)
Start executing action time.sleep(1)
Start executing action Close_RGB()
```

Main Source Code Analysis

```
def play_agent():
    print("start")
    global response, agent_plan_output, xuanxin
    while True:
        if detect_keyword():
            xuanxin = 1
            os.system("kill mplayer")
            Car_Reset()
            time.sleep(.2)

            if os.path.exists('./myrec.wav'):
                os.remove('./myrec.wav')
            time.sleep(0.2)

            start_recording()

            time.sleep(0.2)
            if TTS_IAT_Tongyi:
                rectext = rec_wav_music_Tongyi()
            else:
```

```

rectext = rec_wav_music()

if rectext != "":
    print("Q:" + rectext)
    try:
        if AI_Agent:
            agent_plan_output =
eval(Car_tonyi_agent_online(rectext))
        else:
            agent_plan_output = eval(Car_Agent_Plan(rectext))

        response = agent_plan_output['response']

    except:
        display_text = "There was an error in obtaining the action
information, please try again..."
        print(display_text)
        continue

    print("A:" + response)

```

detect_keyword: Wake-up function for the wake-up word

- Chinese version-specific configuration
 - rec_wav_music_Tongyi:** Tongyi Qianwen voice recognition. Valid only when TTS_IAT_Tongyi = True and the Chinese version is used.
 - rec_wav_music:** iFlytek Spark voice recognition solution. Valid only when TTS_IAT_Tongyi = False and the Chinese version is used.
 - TTS_IAT_Tongyi: Configured in API_KEY.py

Car_tonyi_agent_online: This API executes actions based on the agent configured on the Tongyi Qianwen platform. This API is effective when AI_Agent = True. This API also configures the viewing and selection section.

Car_Agent_Plan(rectext): This API executes actions based on information from the locally deployed **Car_agent.py**. This API is effective when AI_Agent = False.

- The English version's speech synthesis and recognition are already packaged, so you don't need to worry about them here.

Modifying the recording duration, start threshold, and end threshold

1. In the terminal, enter:

```

cd /home/pi/project_demo/09.AI_Big_Model/Car_Choreography/
gedit Car_audio.py

```

2. Find the source code shown below.

```

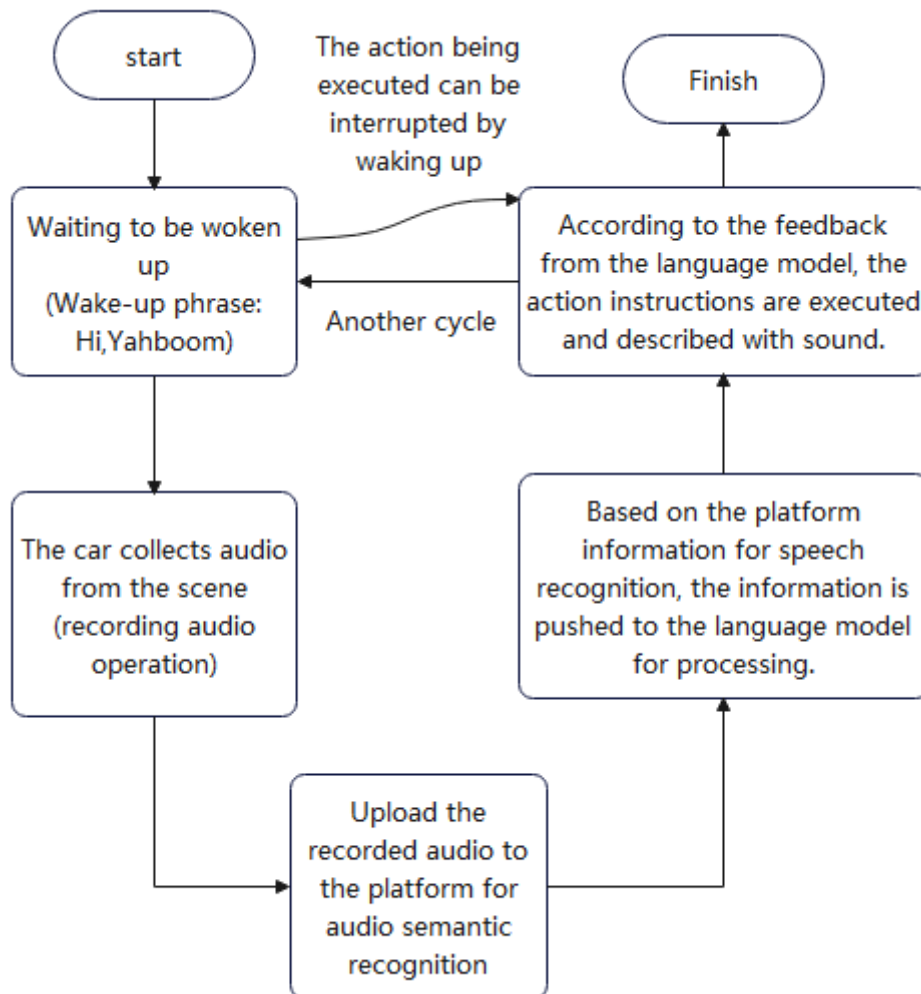
194 def start_recording(timel = 3, save_file=SAVE_FILE):
195     global automark, quitmark
196     start_threshold = 3000 #30000
197     end_threshold = 1500 #20000
198     endlast = 15
199     max_record_time = 8
200
201     CHUNK = 1024
202     FORMAT = pyaudio.paInt16
203     CHANNELS = 1
204     RATE = 16000
205     WAVE_OUTPUT_FILENAME = save_file
206

```

- start_threshold: The threshold for starting recording when sound is detected (this can be lowered to 5000 in quiet environments, and increased to 150000+ in noisy environments).
- end_threshold: The threshold for stopping recording when sound is detected. A value of 30-50% of start_threshold is recommended.
- endlast: The number of times to stop recording. Here, it is 15. For example, if 15 consecutive sound values meet the stop threshold, recording will automatically terminate.
- max_record_time: Recording duration, set to 5 here.

Note: start_threshold > end_threshold. This is a required rule, and its value can be determined by the environment.

Overall flow chart for this experiment



(Selected section)

1. Customizing the English Version

1. Enter the following command in the terminal:

```
cd /home/pi/project_demo/09.AI_Big_Model/Car_Choreography/
gedit Car_agent_en.py
```

Simply modify this file. Note: Do not change the content in the green box. You can add to it, but you cannot delete or modify it.

```
Open  Car_agent_en.py  Save  -  +  x
~project_demo\09_AI_Big_Model\Car_Choreography

1 from Car_Online_API import Api_action_en
2
3
4 AGENT_SYS_PROMPT = '''
5 You are an experienced car with unique insights into smart cars and Mecanum wheels, and have a humorous and witty way of speaking. The car has some built-in functions. Please output the corresponding
6 functions to be run and your reply to me in JSON format according to My instructions.
7
8 [here is an introduction to all built-in functions]
9 forward movement:Car_Forward(speed,mytime)#speed:Representing speed,mytime:A few seconds ahead of the representative,For example, advancing by 2 seconds:Car_Forward(mytime=2);For example, moving forward
10 quickly for 1 second:Car_Forward(speed=100,mytime=1)
11 step back action:Car_back(speed,mytime)#The meaning of its parameters is the same as the parameters of the forward action
12 left turn action:Car_left(speed,mytime)#The meaning of its parameters is the same as the parameters of the forward action
13 right turn action:Car_right(speed,mytime)#The meaning of its parameters is the same as the parameters of the forward action
14 left translation action:Car_left_translation(speed,mytime)#The meaning of its parameters is the same as the parameters of the forward action
15 right translation action:Car_right_translation(speed,mytime)#The meaning of its parameters is the same as the parameters of the forward action
16 nodding motion:Car_servo_nod()
17 shaking head action:Car_servo_sayno()
18 control the color of the small car light:Car_RGB_Control(R,G,B)#of which R:Representing the color red,G:green,B:blue#Their range of values0-255,The higher the value,The darker the color;For example, the
19 car lights appear yellow:Car_RGB_Control(255,255,0)
20 the action of turning off the car lights:Close_RGB()
21 interface for playing music:Car_Music_API(strname,strmusic) #of which strname:It's the singer's name. strmusic:It's the title of the song,For example, the boy of Mengran
22 Car_Music_API(strname="Mengran",strmusic="teenager")
23 obtain the distance of the object/obstacle in front of you:Get_dis_obstacle_en()
24 perform relevant actions based on the corresponding distance:car_avoid_api(fardis,Stopflag)#Fardis is the stopping distance from the object, measured in millimeters; Stopflag: Whether to stop after reaching
25 the distance of fardis, Stopflag=0 means not to stop.
26 interface for resting and waiting:time.sleep(time) #The unit of time is 1 second,For example, waiting is time.sleep(1)
27 there are also some speed related meanings:Fast:speed=100,Low speed:speed=25.If there is no specified direction, the default is to perform the action to the left.
28 there are also some color related meanings:Eggplant is purple, kiwi is brown on the outside, peach is pink, and leaves are green.
29 there are also expressions from some music stars:Jay Chou is Jay Zhou, Old Xue is Jacky Xue, God E is Eason Chan, Jolin is Jolin, etc.
30
31 Output JSON format]
32 You can directly output JSON, starting from {, do not output the beginning or end containing JSON
33 in the 'function' key, output a list of function names, where each element is a string representing the name and parameters of the function to be run. Each function can run independently or sequentially with
34 other functions. The order of list elements represents the order in which functions are executed
35 in the 'response' key, output your reply to me in the first person according to My instructions and your actions. It needs more than 20 words but not more than 60 words. It can be humorous and divergent,
36 using lyrics, lines, Internet hotspots, and famous scenes.
37
38 There are some specific examples]
39 my instructions: Forward for 3 seconds and then backward for 0.5 seconds. You output: {'function': ['Car_Forward(mytime=3)', 'Car_back(mytime=0.5)'], 'response': 'Forward for 3 seconds and then backward for 0.5
40 seconds, this operation is amazing or not'}
41 my instructions: Quickly advance for 2 seconds, then slowly retreat for half a second, and finally quickly turn left for one lap. You output: {'function':
42 ['Car_Forward(speed=100,mytime=2)', 'Car_back(speed=25,mytime=0.5)', 'Car_left(speed=100)'], 'response': 'Fast forward, graceful backward, and another gorgeous left turn, am I your dream car?'}
43 my instructions: Quickly translate for 1 second, then turn right for 1 turn. You output: {'function': ['Car_right_translation(speed=100,mytime=1)', 'Car_right(mytime=1)'], 'response': 'Looking at my lightning
44 fast translation and cool right turn, do you feel like I'm about to take off?'}
45 my instructions: Move forward for 2 seconds and then tell me the distance to the obstacle in front of you. You output: {'function': ['Car_Forward(mytime=2)', 'Get_dis_obstacle_en()'], 'response': 'Move forward for
46 2 seconds, the obstacle distance will be revealed soon. Are you a little nervous?'}
47 my instructions: Nodding, then the car lights turn dark red. You output: {'function': ['Car_servo_nod()', 'Car_RGB_Control(125,0,0)'], 'response': 'Nodding, the small car light is dark red, creating a sense of
48 atmosphere. Isn't it a bit romantic?'}
49 my instructions: Play me a song of Jay Chou's rice fragrance. You output: {'function': ['Car_Music_API(strname="周杰伦",strmusic="稻香")'], 'response': 'As soon as the fragrance of rice rings, memories strike! The
50 car takes you back to your youthful years~'}
51 my instructions: Detected a 20cm obstacle ahead and stopped, then turned left half a circle. You output: {'function': ['car_avoid_api(fardis=200)', 'Car_left(mytime=0.5)'], 'response': 'Obstacle appears, emergency
52 brake 20cm, then make a magnificent left turn, this operation is 6 or 6?'}
53 my instructions: Detected a 28cm obstacle ahead and kept moving forward for 2 seconds, then turned right half a circle. You output: {'function':
54 ['car_avoid_api(fardis=200,Stopflag=0)', 'time.sleep(2)', 'Car_right(mytime=0.5)'], 'response': 'Obstacles ahead? It doesn't exist! Look at me rushing straight over and then making a handsome right turn for half
55 a circle. This operation is either 6 or 6?'}
Python2  Tab Width: 8  Ln 1, Col 1  INS
```

You can modify this file to use the action interface. This requires some basic knowledge, and it is not recommended for beginners to modify it directly.

2. After completing the modification, you can replace the large model by editing **Car_Online_API.py**.

```
cd /home/pi/project_demo/09.AI_Big_Model/Car_Choreography/
gedit Car_online_API.py
```

Then modify the model indicated by the red box.

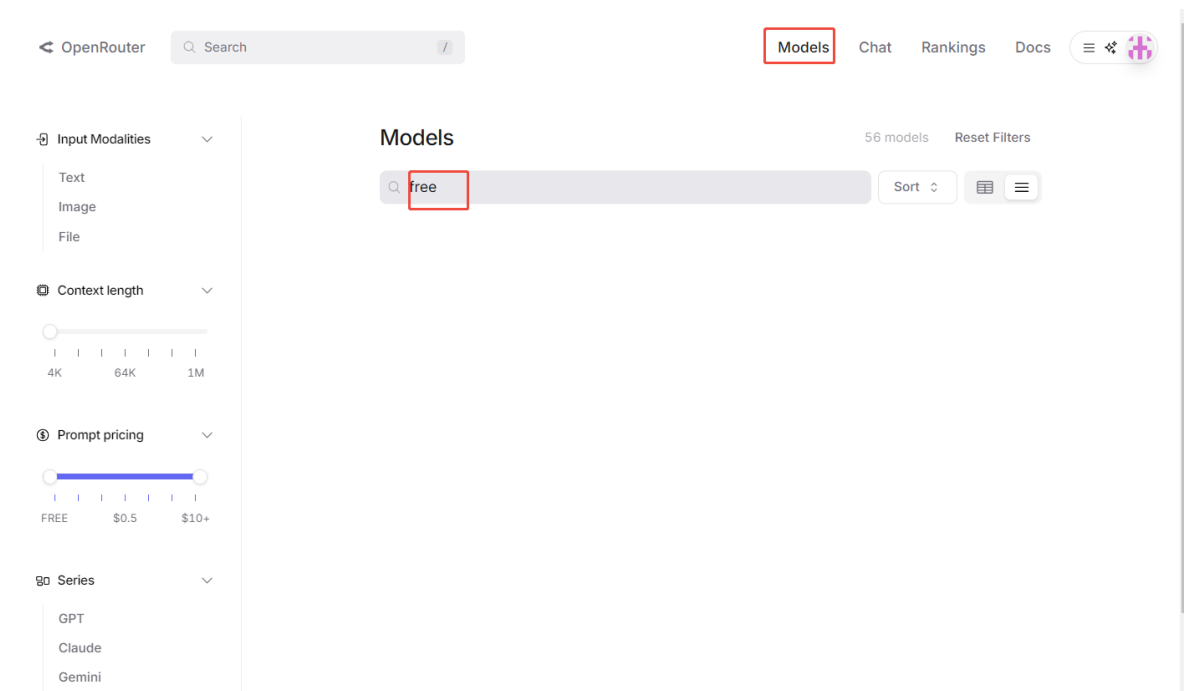
```
Car_Online_API.py
~/project_demo/09_AI_Big_Model/Car_Choreography

45 completion = client.chat.completions.create(
46     # Module List: https://help.aliyun.com/zh/model-studio/getting-started/models
47     model="qwen-plus",
48     messages=[
49         {"role": "system", "content": "You are a helpful assistant."},
50         {"role": "user", "content": PROMPT},
51     ],
52 )
53 #print(completion.model_dump_json())
54 result = completion.choices[0].message.content
55 #print(result)
56 return result
57
58
59
60 def Api_action_en(PROMPT='Car agent'):
61     client = OpenAI(
62         base_url="https://openrouter.ai/api/v1",
63         api_key=openAI_KEY,
64     )
65
66     completion = client.chat.completions.create(
67
68         #model="google/gemini-2.5-pro-exp-03-25:free",
69         model="qwen/qwen2.5-vl-32b-instruct:free",
70         #model="meta-llama/llama-4-maverick:free",
71         #model="nvidia/llama-3.1-nemotron-ultra-253b-v1:free",
72         messages=[
73             {
74                 "role": "user",
75                 "content": [
76                     {
77                         "type": "text",
78                         "text": PROMPT
79                     }
80                 ]
81             }
82         ]
83     )
84
85     result = completion.choices[0].message.content
86     #print(result)
87     return result
```

Learn about the models that can be replaced

First, visit the website

[openrouter](https://openrouter.ai)



Using deepseek as an example:

OpenRouter
Search
Models
Chat
Rankings
Docs

Input Modalities
Text
Image
File
Context length
4K 64K 1M
Prompt pricing
FREE \$0.5 \$10+
Series
GPT

Models

56 models Reset Filters

free
Sort

DeepSeek: DeepSeek V3 0324 (free) 259B tokens

DeepSeek V3, a 685B-parameter, mixture-of-experts model, is the latest iteration of the flagship chat model family from the DeepSeek team. It succeeds the DeepSeek V3 model and performs ...

by [deepseek](#) | 33K context | \$0/M input tokens | \$0/M output tokens

DeepSeek: R1 0528 (free) 66.6B tokens

May 28th update to the original DeepSeek R1 Performance on par with OpenAI o1, but open-sourced and with fully open reasoning tokens. It's 671B parameters in size, with 37B active in an ...

by [deepseek](#) | 164K context | \$0/M input tokens | \$0/M output tokens

Overview Providers Versions Apps Activity Uptime **API**

OpenRouter provides an OpenAI-compatible completion API to 400+ models & providers that you can call directly, or using SDK. Additionally, some third-party SDKs are available.

In the examples below, the [OpenRouter-specific headers](#) are optional. Setting them allows your app to appear on the OpenR leaderboards.

openai-python python typescript openai-typescript curl

```

from openai import OpenAI

client = OpenAI(
    base_url="https://openrouter.ai/api/v1",
    api_key="OPENROUTER_API_KEY",
)

completion = client.chat.completions.create(
    extra_headers={
        "HTTP-Referer": "<YOUR_SITE_URL>", # Optional. Site URL for rankings on openrouter.ai.
        "X-Title": "<YOUR_SITE_NAME>", # Optional. Site title for rankings on openrouter.ai.
    },
    extra_body={
        model="deepseek/deepseek-chat-v3-0324:free",
    },
    messages=[

```

3. Finally, run the startup command.

```

```shell
cd /home/pi/project_demo/09.AI_Big_Model/
python3 Car_Choreography/Alagent_main_en.py

```

#### \*\*Tips\*\*

List some dialogues

1. Drive forward for 2 seconds, then change the taillights to gold, turn left halfway, and nod.
2. Change the taillights to a more romantic color and play Jay Chou's "Rice Fragrance."
3. Stop 15 cm away from the obstacle.