The design of an artificial intelligence service assistant

Jinyong Li, Jianzhi Deng* and Peipei Zhi

Guangxi key Laboratory of Embedded Technology and Intelligent System, Guilin University of Technology, Guilin, 541004, China

*Corresponding author’s e-mail: dengjianzhi@163.com

Abstract. In order to meet people’s requirement for smart-home and life assistant, this paper presents a design of artificial intelligence (AI) service assistant which adopts NAO robot and raspberry Pi. The design incorporates Python language with the technology of web crawler that realized the function of human-computer interaction of voice such as the weather and ticket inquiry, the infrared control of air-condition. The design can achieve expected results by verifying several functions experimentally in a laboratory, which can provide a powerful reference for the robot-assistant of smart-home, daily-service and related experimental projects.

1. Introduction
With the development of artificial intelligence, automation and computer network, people desire an intelligent life and a powerful life-assistant. Hence, researching on the smart service robots is being designed. Humanoid robot is the mainstream of service robots with the structure and appearance as human beings. The serviceable robot can provide various services as information inquiry and assistive work, such as an assistant or a secretary [1, 2]. This project aimed at the design of an artificial intelligence service assistant with humanoid robot NAO and raspberry Pi. It can be realized the information query and device control in human-robot interaction (HRI). This current study provides a reference design of interactive robot assistant.

2. System Design
2.1 General Principle
By using the NAO as the platform, the functions of weather query, ticket schedules and air-condition (AC) control via voice are realized. It realizes that human-computer dialogue with the help of built-in speech recognition and playback function by NAO robot and its Wi-Fi module. The Raspberry Pi-3B was connected by TCP/IP. The General Purpose Input/Output (GPIO) of Pi-3B controls AC by IR module. The design was shown in Figure 1. The NAO robot received human speech by built-in microphone. The recognized voice signal is compared to the vocabulary database of NAO. And then, the robot did the corresponding action to the judgement of voice recognition. When controlling the AC, NAO was made a connection to Pi-3B server at first. Secondly, Pi-3B sent the IR command to control the AC and accomplished the right action according to the receiving signal. Pi-3B needs to record infrared signal from remote controller, so that it can realize the control of target device.

2.2 System Framework
When inquiring network information, NAO robot is connected to the router via built-in WIFI to access the special website on the Internet, gained and preserved the data on the webpage. Then it analyses
webpage to get the query’s data, and plays the content by loud speaker. When controlling AC, NAO robot is connected to the Pi-3B in the same WLAN to send the control command to Pi-3B. After that, the Pi-3B converts the commands into remote signal, and sends the infrared signal.

Figure 1. The diagram of design

Figure 2. The transmitting and receiving circuit diagram of infrared

3. The Hardware Design

NAO, a humanoid robot platform for robocup competition, is developed by Aldebaran with high performance CPU and 25 degrees of freedom, more than 100 sensors, and other hardware resources. It can be utilized as supporting voice recognition functions, converting and speaking in 21 languages [3-5]. Dr. Eben Upton designed the Pi-3B by a multifunctional open source micro Linux Computer with CPU, RAM, USB controller, HDMI, Ethernet, WIFI and GPIO [6-8]. The system controls ACs through infrared. Figure 2 shows the schematic of infrared module. The transmitter TSAL6200 is driven by triode. And the 38KHz signals is received by the HS0038 receiving diode.

4. Software Design

4.1 General Overview

NAO is programmed via Choregraphe. It can be setup under any platform (Windows, Linux, Mac OS) and used to monitor and control the robot. NAO runs on NAOqi operating system which provided the motion control, video, voice of robot and the communication, programming between devices and information sharing. The design realizes the HRI through voice dialogues. NAO receives and recognizes the speech signal, then the received voice data was compared with preload database to output the vocabularies with the highest similarity, and the language spoken by person will be judged to perform relevant action, as shown in Figure 3.

4.2 Weather Query

There are two methods to show the weather online, webpage or API. The webpage data is captured by Python in the first way. Uniform Resource Location (URL) is the string that describes location of information resource. The URL link of Beijing Weather on Weather China webpage which provides server about weather, www.weather.com.cn/weather/101010100.shtml. It consists of three parts, hypertext transfer protocol, domain name and the specific address of the webpage.

Data of the weather is displayed as a HTML webpage. So its code matches regular expression to withdraw the information that you need. The partial webpage and corresponding HTML text are shown in Figure 4. If the command of querying weather is recognized, it would execute the querying program by using urlib library urlopen (ULU) in order to download and analyze the relevant data from the server.
When looking over the weather code of Beijing, the weather, temperature and other data are also listed in the HTML tag. In order to improve the robustness, when searching the information queried in webpage code, the stable label were selected to extract information. When querying the city’s name, the corresponding name extracted from the “title” tag. The “title” tag was as follows:

\[
\text{<title>}[\text{Beijing Weather}] \text{ Weather forecast in Beijing, ...}, \text{weather query for the next 15 days</title>}
\]

According to the syntax rules, the regular expression of extracting city information was

\[
\text{“reg = r’<title>[(.+?)weather].+?’</title>’”}
\]

which means ‘<title>[(.+?)weather].+?’</title>’ does not string-escaping, so the information was to be matched accurately, among them, the “(.+?)” represents what the regular expression extracted. It needs to intercept the data by using regular expression for extracting the weather that day, such as “<h1>25(today)</h1>” and “<h1>26(tomorrow)</h1>”; then the weather, temperature and wind index of the day were matched and extracted. The webpage source code of the day was show in Figure 5.

4.3 Tickets Query

Another function of this design is ticket query, and broadcasting by the speaker. It obtains train-ticket data from 12306.cn. The train ticket is queried, beginning with the request interface of train ticket data on the 12306.cn is found.

For example, the train tickets were from Yulin to Guilin on May 30, 2018 by 12306.cn. The interface was, https://kyfw.12306.cn/otn/leftTicket/query?leftTicketDTO.train_date=2018-05-30&leftTicketDTO.from_station= YLZ&leftTicketDTO.to_station= GLZ&purpose_codes=ADULT, which contains date train_date, starting station from_station, destination station to_station and ticket type purpose_codes.

The starting and destination stations display in station code, rather than Chinese characters. The station has a certain relationship between each station name and the station codes of 12306.cn and is stored in .js files, which is returned to the client. Chinese name of the railway station was converted to corresponding code. And the URL link of requesting was constructed, so the ticket information can be queried.

URL codes of the railway-station requests are shown in Figure 7. The code of Beijing North station is "VAP" and the Beijing East station code is "BOP". In a range of data for querying railway tickets, the displayed data are obtained from the .js file. When receives the instruction of ticket query, NAO enters the train ticket query process.

The program requests the URL link that encompass station codes by the requests of the get() method:  https://kyfw.12306.cn/otn/resources/js/framework/station_name.js?station_version=1.9053. The server returns the .js file containing station codes, and finds the relationship between the station’s Chinese name and code from the string of response object, then the relationship will be converted into a Python dictionary format by the dict() method, which was convenient for language call. Furthermore, using a method that Python built-in function called dict.get (key, default= None) to return
corresponding station code, which was converted from the Chinese name of origin and destination station.

There is no corresponding and no state between the two access requests of client. When querying the train ticket, you need to give an access to the 12306 website for several times via Cookie or Session, wherein Cookie is used on the 12306 website. The design uses requests library Session() method to process and save local Cookie.

By using the corresponding station code to construct the URL link, it is accessed by the method mentioned above. And the .js files returned by the server are converted to the Python dictionary format using the json() method of Python standard json library. Then the data about train number are extracted and encoded into “utf-8” format through encoding conversion. The .js file containing the train data is visualized in Figure 8.

In order to realize the flight ticket query, it needs to find the interface to store the flight ticket data. Then according to query time and the URL interface, the flight ticket data of the Ctrip.com are requested and captured in a manner of simulating the browser accessing the website. The difference between the airline ticket query and train ticket query process is that the .js file returned by Ctrip.com contains fares, and the specific implementation process is no longer exhaustive.

4.4 Infrared Remote Control
Pi-3B will play a significant role when the system controls AC: Pi-3B receives the instructions from NAO by TCP/IP protocol; then the infrared transmitter tube sends signal to ACs, through Pi-3B’s programmable GPIO pin.

The TCP/IP protocol is a protocol of computer network. In this design using the TCP connection C/S mode, the Pi-3B as a server and the NAO as a client, then it could establish a connection via "three-way handshake", first NAO makes a request to Pi-3B, and after receiving the request, Pi-3B will provide the corresponding service and close the connection. Namely the Pi-3B controls the infrared transmitting tube to emit a corresponding infrared signal according to the command of controlling ACs sent by NAO. On the other hand, utilizing Python’s standard library Socket to realize the communication and connection of NAO, and the functions in the Socket library was used to create socket objects, configure TCP connection protocol, IP address and port, and so on.

The design realizes the control of ACs for two brands of Gree and Midea. Due to the existing AC still uses mainstream infrared remote control [10], the Pi-3B first needs a learning process for the infrared signal of the original remote controller. Through the oscilloscope, you can observe the Gree’s remote control signal received by the Pi-3B infrared: the infrared remote controller transmits a 9ms high level and a 4.5ms low level as the boot code, and then transmits it one by one. At the same time, the signal level logic value "0" is represented by a 660us burst plus a low level of 540us; the logic
value "1" is represented by a 660us burst plus a 1640us low level. Figure 9 shows the logic value “0” captured by the oscilloscope, which does not strictly adhere to the NEC protocol.

With the help of the Linux system open source package Linux Infrared Remote Control (LIRC), the infrared signal of the remote control will be recorded in Pi-3B and copied into the lircd configuration file. Then lired controls the infrared transmitter module to transmit the infrared remote control signal similar to the air conditioner remote controller. Pi-3B acquires infrared remote signal mainly by interruption mechanism, its Linux infrared remote control originates from Aron Szabo’s original LIRC driver which was used to receive the IR signal and transmit to the user process Lircd.

Due to the complexity of the infrared signal, the LIRC software package does not support this device, and the generated configuration file only refers to the format of the LIRC remote control configuration file. First, the lirc-rpi module was enabled to trigger the Pi-3B’s GPIO, to receive and send infrared signals, to modify the driver configuration in the file “/etc/lirc/lirc_options.conf”: GPIO17 pin controls infrared transmission, GPIO18 pin receives the infrared signal. Then Pi-3B combines the command line to record the remote command one by one, and transmits the corresponding infrared signal by inputting the irsend command in Pi-3B. In addition, the irsend was the basic LIRC program that sends infrared command in the format: “irsend SEND_ONCE remote control name button name”.

![Figure 9. The remote control signal](image)

Pi-3B as a server should have the function of running the service program itself after power-on which avoids the trouble of manual operation. After Raspbian OS is powered-on, it logs into the system automatically as a "pi" user, then runs "startx" function automatically and starts up the window system, runs a custom program automatically. The path of the service program that needs to be run automatically has been stored in the file called /home/pi/.config/autostart/my.desktop.

5. System Test
Weather and ticket query, remote control air conditioner were tested, and the test equipment is shown in Table 1. The operation results of human-computer interaction were tested through the dialogue between the tester and NAO, and the synchronous debugging was performed in the Choregraphe development environment.

| Device Name | Model of Device | Notes |
|-------------|-----------------|-------|
| NAO robot   | V5              | NAOqi OS |
| Raspberry Pi 3B | ----- | Raspbian OS |
| TP-LINK Router | TL-WR740N | Wireless Routing |
| Wall-mounted Gree Air-conditioner | KFR- | Remote Control of YAPOF |
|             | 50GM/(50556)NhAd-3 | type 1 |

NAO will query and tell the weather of the XX city when accepted the word, "What is the weather like in XX". Figure.10 (a) presents a request to test the weather in Beijing and Washington. The
weather, maximum temperature, minimum temperature, wind index, and so on could be correctly reported.

When NAO enters the train ticket query process and informs some information such as start and destination station, departure date, NAO can correctly broadcast the ticket number and the information of each type of seat. The train ticket inquiry from Guilin to Yulin on June 7 as shown in Figure 10(b). When the tester tells NAO to enter the AC control mode of “Gree” or “Midea”, and performs switching and adjusting. The experimental results show that NAO can control the AC normally, and Figure 10(c) shows the test about controlling the AC.

6. Conclusion
In this paper, a robot service assistant was designed. The hardware platform was built by using NAO and Raspberry Pi, which realized webpage analysis, data extraction and matching, and the transmission and reception of AC remote control signal. Python was used to realize weather and ticket queries, home-appliance control and other functions under human-machine voice interaction, and achieved the expected experimental results. The design is used as a reference for resolving service-robot with human-computer interaction, which is realized by artificial intelligence and related application technologies. At the same time, it could be the experimental projects based on platform tools such as NAO, Raspberry Pi and Python.

Acknowledgments
This work was supported by Guangxi key laboratory foundation of embedded technology and Intelligent System, and Innovation and entrepreneurship training program for College Students.

References
[1] Guo Yu. (2017) Artificial Intelligence and Home Design. Packaging Engineering. 38 (16): 12-15
[2] Dang Hongshe, Hou Jinliang, Zhang Chao. (2017) Design and implementation of home assistant robot system based on Kinect. Application of Electronic Technique, 43(3):70-72
[3] Wang Zhen, Zhang Xing. (2016) Design and implement of innovation experiments based on NAO robot platform. Research and exploration in laboratory.35(12):156-159
[4] Liang Fuxin, Liu Hongbin, Zhang Fulei and Chang Faliang. (2017) Target recognition method based on NAO robot. Computer Engineering and Design. 38(8):2235-2239
[5] Yu Jianjun, Men Yusen, Yuan Xiaogang, Zhao Shaqiong. (2016) The research and implementation of behavior imitation system about NAO robot based on Kinect. CAAI Transactions on Intelligent Systems. 11(2):180-188
[6] Du Feixiang, Yang Jing, Shi Jianfen. (2018) Intelligent traffic control platform based on Flask and Raspberry Pi. Modern Electronics Technique. 31(13):88-91
[7] Yin Xianhua, Liu Mingyuan, Wang Ning. (2016) Design of smart home system based on Raspberry[J]. Modern Electronics Technique.39(23): 161-164
[8] Wang Yuanjie, Lin Du, Xian Hao. (2019) Analysis of various programming methods based on NAO robot. 3(38):42-46

[9] Liu Aiqin, Wang Youlin, Shang Shan. (2018) Optimization and Implementation of Keywords Association Recommendation Algorithm Based on Crawling Technology. 41(4):133-138