Snips 和 Raspberry Pi 3 构建您自己的天气语音助手

Phodal Huang

September 8, 2017

目录

原文链接：https://www.wandianshenme.com/play/snips-raspberry-pi-build-privacy-voice-command

你可能听说过像 Amazon Echo 和 Google Home 这样的语音助手。也许，你也对通过声音控制你的家的想法感到好奇。然而，您的家庭也是您最亲密的空间，您在安装一个永远监听的云端连接设备之前，会进行深思熟虑地，将您的数据传输到远程服务器上...不要再担心，我们为您提供了一些特别的东西！☐

在 Snips，我们正在面对一个挑战，将人工智能添加到我们的日常生活中，就意味着放弃隐私的想法。这是为什么我们最近推出了 Snips 语音平台的信念，这是一个 Amazon Alexa 的 100％替代品，允许任何人轻松地在连接的设备上，添加强大的语音助手，而不会影响其隐私。

在这篇文章中，我将引导您完成 3 个简单的步骤，以建立您自己的“私人设计”的 Amazon Echo。

在一小时之内，您应该在您面前有一个功能性的语音助手，用于回应天气查询。主要步骤如下：

1. 建立你自己的助手
2. 在你的 Raspberry Pi 3 上设置 Snips。
3. 处理天气要求。

步骤 0：材料准备

1 2 3

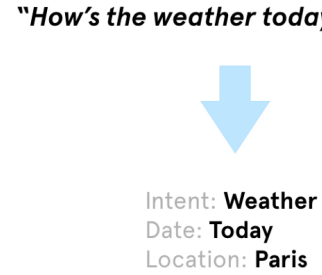
我们需要的有：

1. Raspberry Pi 3
2. 麦克风
3. 扬声器

如果您准备好了这个材料，我们继续前进！

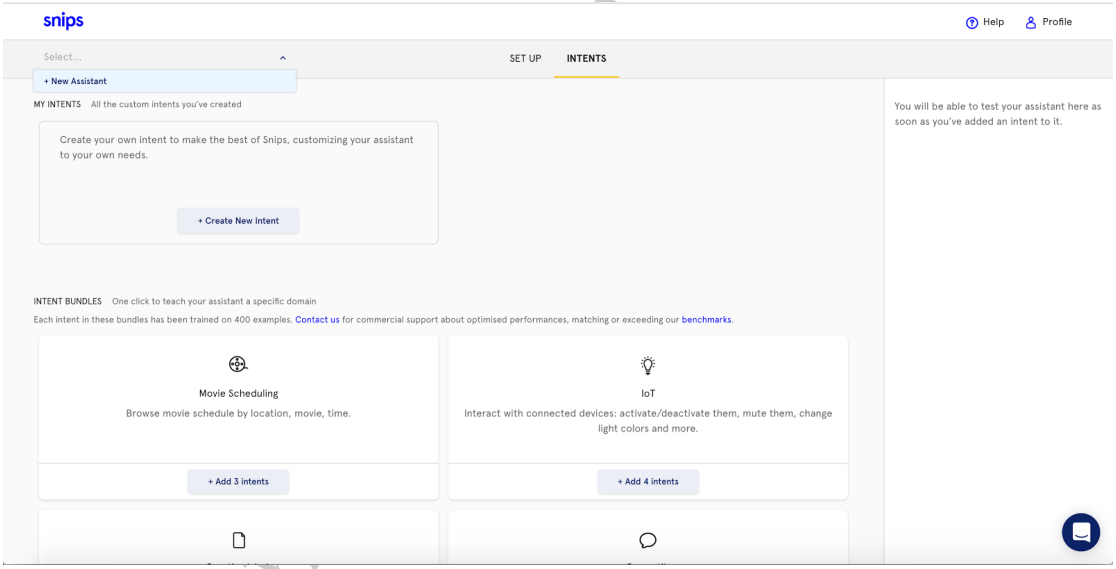
步骤 **1**：建立你自己的助手

目标：在第一步中，我们将构建一个组件，它能够从天然语言表达的天气查询中，提取明确的意图、数据和位置。



跳转到 https://console.snips.ai/intents，然后登录。

你现在在 Snips 的控制台上。在这里，您可以教导您的助理，了解您希望处理的任何命令。现在，我们对于与天气相关的查询感兴趣。因此，让我们创建一个助手，点击左上角的菜单，并将其命名为 WeatherAssistant。



在 SET UP （“设置”）选项卡下，我们将保留当前的参数：我们使用将在您的设备上运行的 Snips 的自动语音识别（ASR）解决方案，以英语为对象。

您现在可以点击 INTENTS 选项卡。

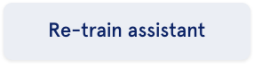
在这里，您可以提供由助理处理的意图（intents）。对于天气相关的查询，我们已经有一套准备好了的意图（intents）。我们称之为捆绑（bundle）。向下滚动并将其添加到您的助手。

向上滚动。你可以看到 3 个意图（intents）被添加到助理：

- SearchWeatherForecast：一般天气相关的问题

- SearchWeatherForecastCondition：针对天气条件的问题
- SearchWeatherForecastTemperature：针对温度问题

要完成这一步，只需训练你的模型：



现在，在右侧的控制台中，测试它以进行验证。

☐验证步骤 **1:**

在控制台中输入『How's the weather today in Paris?』。这时应该检测到 SearchWeatherForecast 的意图，与相应的时间和位置相关联。

步骤 **2**：在 **Raspberry Pi 3** 设置 **Snips**

目标：在你的 Raspberry Pi 3 上搭建并运行 Snips。我们希望它在你叫 "Hey Snips" 时做出反应，当我们大声问候天气时，我们希望它能够理解。

由于，我们已经在其他玩法里，介绍如何进行基本的 Raspberry Pi 搭建，这里仅列一下步骤：

1. 在 SD 卡上安装 Raspbian Jessie Lite
2. 允许 SSH
3. 连接 Raspberry Pi 到互联网中

接下来，我们就可以进入下一步。

安装 **Snips**

如上所述连接到您的 Raspberry Pi 3。然后运行：

```
$ curl https://install.snips.ai -sSf | sh
```

此命令将安装 Snips 及其依赖项，以及一些方便的工具。当被问及时，请确认您愿意安装 Docker 和 python。

☐ 如果您获得权限被拒绝的错误，请重新登录您的 Raspberry Pi 3，并再次运行安装脚本。

设置声音的输入和输出

从您的 Raspberry Pi 3，运行以下命令：

```
$ aplay -l
```

应该触发如下的响应：

```
**** List of PLAYBACK Hardware Devices ****
card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IEC958/HDMI]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

从该命令的输出，我们知道音频输出链接到设备（card 0 ）和设备 0（device0）。这就是提到的 hw：0,0。指标在您的设置中可能会有所不同，这就是为什么我们要执行这些步骤。

同样，要知道音频输入的声卡和设备索引：

```
$ arecord -l
```

如：

```
**** List of CAPTURE Hardware Devices ****
card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]
 Subdevices: 1/1
 Subdevice #0: subdevice #0
```

音频输入连接到声卡 1 和设备 0，这将是提到的 hw：1,0。

有了这些信息，我们可以相应地更新 □/.adoundrc 文件：

```
$ nano ~/.asoundrc
```

安装你的助手

现在，在验证 Snips 是搭建好、运行之前的最后一步，便是安装您在步骤 1 中准备的助理。从您的计算机执行：

```
$ scp <PATH_TO_ASSISTANT_ZIP> pi@raspberrypi.local:assistant.zip
```

然后从你的 Raspberry Pi 上执行：

```
$ snips-install-assistant assistant.zip
```

□验证步骤 **2**：

我们现在准备好验证 Snaps 在您的 Raspberry Pi 3 上运行。在一个终端窗口中，从您的 Raspberry Pi 3 运行

```
$ snips
```

这将启动 Docker 容器中的所有 Snips 组件。从另一个窗口中，仍然从您的 Raspberry Pi 3，运行以下：

```
$ snips-watch
```

这将简明扼要地告诉你发生了什么。一旦应用完成所有的加载，只需要唤醒 “Hey Snips” 就能唤醒助手，它应该引起一些反应。它被称为唤醒词，它只是与你的助手沟通，然后你才能多问一些事情。

你现在可以说 “How’s the weather today in Paris?”。您应该从日志中看到 `SearchWeatherForecast` 意图已被触发。

让我花一点时间解释，刚刚发生在您的设备上的事情：唤醒词检测、语音识别和自然语言理解。如果你想确保这是真的，关掉你的 WiFi，再试一次！

他们现在只剩下一步，就是告诉你的助手：如何回应这样的查询。

步骤 3：处理天气请求

目标：在这里，我们只是要求我们的助手，在问及天气时做出回应。

这一步很容易。登录到 Open Weather Map。我们将依靠他们的 API，来获取有关天气的信息。一旦登录后，请转到 API 密钥选项卡，然后复制您的密钥。

现在，下载以下脚本，并替换 `<WEATHER_API_KEY>` 和 `<CITY_NAME>` 参数：

```
# encoding: utf-8
from __future__ import unicode_literals
```

```
3
4 import datetimeimport json
5
6 import paho.mqtt.client as mqtt
7 import requests
8
9 fromtimestamp = datetime.datetime.fromtimestamp
10
11 # MQTT client to connect to the bus
12 mqtt_client = mqtt.Client()
13 HOST = "localhost"
14 PORT = 9898
15 WEATHER_TOPICS = ['hermes/intent/SearchWeatherForecastTemperature',
16                   'hermes/intent/SearchWeatherForecastCondition',
17                   'hermes/intent/SearchWeatherForecast']
18
19 # WEATHER API
20 WEATHER_API_BASE_URL = "http://api.openweathermap.org/data/2.5"
21 WEATHER_API_KEY = "<YOUR_APP_ID>"
22 DEFAULT_CITY_NAME = "<YOUR_CITY>"
23 UNITS = "metric"
24
25
26
27 # Subscribe to the important messages
28 def on_connect(client, userdata, flags, rc):
29     for topic in WEATHER_TOPICS:
30         mqtt_client.subscribe(topic)
31
32
33 # Process a message as it arrives
34 def on_message(client, userdata, msg):
35     print msg.topic
36
37     if msg.topic not in WEATHER_TOPICS:
38         return
```

```
39
40 slots = parse_slots(msg)    weather_forecast = get_weather_forecast(slots)
41
42
43     if msg.topic == 'hermes/intent/SearchWeatherForecast':
44         '''
45         Complete answer:
46             - condition
47             - current temperature
48             - max and min temperature
49             - warning about rain or snow if needed
50         '''
51         response = ("It will be mostly {0}{1} today. "
52                     "Current temperature is {2} degrees. "
53                     "Max temperature will be {3}. "
54                     "Minimum will be {4}.").format(
55             weather_forecast["mainCondition"],
56             weather_forecast["inLocation"],
57             weather_forecast["temperature"],
58             weather_forecast["temperatureMax"],
59             weather_forecast["temperatureMin"]
60         )
61         response = add_warning_if_needed(response, weather_forecast)
62
63     elif msg.topic == 'hermes/intent/SearchWeatherForecastCondition':
64         '''
65         Condition-focused answer:
66             - condition
67             - warning about rain or snow if needed
68         '''
69         response = "It will be mostly {0}{1} today.".format(
70             weather_forecast["mainCondition"],
71             weather_forecast["inLocation"]
72         )
73         response = add_warning_if_needed(response, weather_forecast)
74
```

```
75      elif msg.topic == 'hermes/intent/SearchWeatherForecastTemperature':
76  '''         Temperature-focused answer:
77              - current temperature
78              - max and min temperature
79          '''
80          response = ("Current temperature{0} is {1} degrees. "
81                      "Today, max temperature will be {2}. "
82                      "Minimum will be {3}.").format(
83              weather_forecast["inLocation"],
84              weather_forecast["temperature"],
85              weather_forecast["temperatureMax"],
86              weather_forecast["temperatureMin"]
87          )
88
89      say(response)
90
91
92  def parse_slots(msg):
93      '''
94      We extract the slots as a dict
95      '''
96      data = json.loads(msg.payload)
97      return {slot['slotName']: slot['rawValue'] for slot in data['slots']}
98
99
100 def say(text):
101     '''
102     Print the output to the console and to the TTS engine
103     '''
104     print(text)
105     mqtt_client.publish('hermes/tts/say', json.dumps({'text': text}))
106
107
108 def parse_open_weather_map_forecast_response(response, location):
109     '''
110     Parse the output of Open Weather Map's forecast endpoint
```

```
111     '''
112 today = fromtimestamp(response["list"][0]["dt"]).day    today_forecasts =
        filter(lambda forecast: fromtimestamp(forecast["dt"]).day==today, response["list"])
113
114     all_min = [x["main"]["temp_min"] for x in today_forecasts]
115     all_max = [x["main"]["temp_max"] for x in today_forecasts]
116     all_conditions = [x["weather"][0]["main"] for x in today_forecasts]
117     rain = filter(lambda forecast: forecast["weather"][0]["main"] ==
            "Rain", today_forecasts)
118     snow = filter(lambda forecast: forecast["weather"][0]["main"] ==
            "Snow", today_forecasts)
119
120     return {
121         "location": location,
122         "inLocation": " in {0}".format(location) if location else "",
123         "temperature": int(today_forecasts[0]["main"]["temp"]),
124         "temperatureMin": int(min(all_min)),
125         "temperatureMax": int(max(all_max)),
126         "rain": len(rain) > 0,
127         "snow": len(snow) > 0,
128         "mainCondition": max(set(all_conditions),
                key=all_conditions.count).lower()
129     }
130
131
132 def get_weather_forecast(slots):
133     '''
134     Parse the query slots, and fetch the weather forecast from Open Weather
            Map's API
135     '''
136     location = slots.get("weatherForecastLocality", None) \
137             or slots.get("weatherForecastCountry", None)  \
138             or slots.get("weatherForecastRegion", None)  \
139             or slots.get("weatherForecastGeographicalPOI", None) \
140             or DEFAULT_CITY_NAME
141     forecast_url = "{0}/forecast?q={1}&APPID={2}&units={3}".format(
```

```
142         WEATHER_API_BASE_URL, location, WEATHER_API_KEY, UNITS)
143 r_forecast = requests.get(forecast_url)    return
        parse_open_weather_map_forecast_response(r_forecast.json(), location)
144
145
146 def add_warning_if_needed(response, weather_forecast):
147     if weather_forecast["rain"] and weather_forecast["mainCondition"] !=
          "rain":
148         response += ' Be careful, it may rain.'
149     if weather_forecast["snow"] and weather_forecast["mainCondition"] !=
          "snow":
150         response += ' Be careful, it may snow.'
151     return response
152
153
154 if __name__ == '__main__':
155     mqtt_client.on_connect = on_connect
156     mqtt_client.on_message = on_message
157     mqtt_client.connect(HOST, PORT)
158     mqtt_client.loop_forever()
```

复制到你的 Raspberry Pi 3 上：

```
1 $ scp <PATH_TO_HANDLER> pi@raspberrypi.local:.
```

并在，第三个终端窗口中运行你的处理程序：

```
1 $ python handler.py
```

☐验证步骤 **3**：

我们现在可以重现我们已经做了什么，以此来验证步骤 2，说：“Hey Snips”，然后“How’s the weather today in Paris?”。你的助手现在应该回应你了！

步骤 4：Next

您一直使用的所有代码都是非专用的免费代码。随意个性，欢迎在 www.console.snips.ai 上为您的助手添加新意图，并探索新的黑客！特别是，如果你进入家庭自动化，你可能会喜欢 Snips 和家庭助理平台可以一起使用！

对于我们来说，我们有一些新功能即将推出，我们认为您会喜欢：法语，西班牙语，意大利语和德语的端到端支持，定制唤醒词以及对 Raspberry Pi 3 等其他平台的支持！我们也在努力给我们的助手一个更好的声音。

在此期间，我们很乐意通过我们的 Slack 社区 听取您的意见。如果您有反馈意见，请随时给我们打电话！ □ □ □

原文链接：https://medium.com/snips-ai/build-a-weather-assistant-with-snips-4253541f1684

原文链接：https://www.wandianshenme.com/play/snips-raspberry-pi-build-privacy-voice-commanc