

【智能音箱基础】唤醒词检测引擎 **Snowboy** 快速搭建指南

Phodal Huang

September 8, 2017

目录

步骤 0: 材料准备 3

步骤 1: 下载 Snowboy 4

步骤 2: 访问麦克风 4

步骤 3: 运行 DEMO 5

 解码器结构 5

 运行 DEMO 5

玩点什么: <https://www.wandianshenme.com>

原文链接: <https://www.wandianshenme.com/play/smart-speaker-hotword-detection-engine-snowboy>

Snowboy 是一款高度可定制的唤醒词检测引擎, 可以用于实时嵌入式系统, 并且始终监听 (即使离线)。当前, 它可以运行在 **Raspberry Pi**、(**Ubuntu**) **Linux** 和 **Mac OS X** 系统上。

一些热门的唤醒词包括, **Amazon Echo** 上的“**Alexa**”, **Android** 设备上的“**OK Google**”和 **iPhone** 上的“**Hey Siri**”。这些唤醒词用于, 发起一个完整的语音交互界面。除了此, 唤醒词还可以用于其他用途, 比如执行简单的命令和控制动作。

在一个棘手 (**hacky**) 的解决方案中, 它可以运行完整的自动语音识别 (**ASR**, **Automatic Speech Recognition**) 来执行热词检测。在这种情况下, 设备将在自动语音识别转录中观察特定的触发词。转录中观察特定的触发词。另外, 当使用基于云的解决方案时, 它也不会保护您的隐私。幸运的是, **Snowboy** 被创造出来, 解决这些问题。

Snowboy 具有以下特性:

- 高度可定制。允许您自由定义自己的魔术词, 如 (但不限于) “芝麻开门” (**open sesame**)、 “打开车库门” (**garage door open**)、或者 “你好, 梦之屋” (**hello dream-house**)。你能想到的, 你就能定制它。
- 一直监听, 但是保护您的隐私。因为 **Snowboy** 没有连接到网络, 因此不需要将你的声音上传到任何地方。
- 轻巧的、可嵌入, 可以让您在 **Raspberry Pi** 上运行。在最小的 **Pi** (单核 **700M Hz ARMv6**) 上消耗少于 **10%** 的 **CPU**。
- **Apache** 协议

当前, **Snowboy** 可以支持:

- 所有的 **Raspberry Pi** (搭载 **Debian Jessie 8.0**)
- 64 位 **Mac OS X**
- 64bit **Ubuntu (12.04 和 14.04)**
- **iOS**
- **Android** (**ARMv7 CPU**)
- **Pine 64**, 搭载 **Debian Jessie 8.5** (内核版本 **3.10.102**)
- **Intel Edison**, 搭载 **Ubilinux (Debian Wheezy 7.8)**

步骤 0: 材料准备

为了使用 **Snowboy**, 我们需要:

- 一个支持 **Snowboy**，并带有麦克风的设备
- 相应的解码器（从步骤 1 下载）
- 一个训练好的模型，可以从 <https://snowboy.kitt.ai/> 获取

如，从<https://snowboy.kitt.ai/hotword/351>，可以获取一个中文的“小白”模型。但是下载的时候，需要自己录入自己的语音。

步骤 1: 下载 **Snowboy**

您可以下载预先打包好的 **Snowboy** 二进制文件，及其 **Python** 封装：

- 64 位 **Ubuntu**: [12.04](#) 和 [14.04](#)
- **Mac OS X**
- 任意运行 **Raspbian 8.0** 的 **Raspberry**, [1/2/3/Zero](#)
- 运行 **Debian Jessie 8.5 (3.10.102)** 的 **Pine 64**, [Pine 64](#)
- 运行 **Ubilinux (Debian Wheezy 7.8)** 的 **Intel Edison**, [Edison](#)

或者自行从 **GitHub** 下载并编译: [Snowboy](#)

步骤 2: 访问麦克风

在这里，我们使用 **PortAudio** 作为音频输入/输出的跨平台支持。我们还使用 **sox** 作为快速实用程序，来检查麦克风设置是否正确。因此，我们需要先安装这个软件，使用麦克风。

1. 安装 **Sox**

Linux 系统执行：

```
1 sudo apt-get install python-pyaudio python3-pyaudio sox
```

Mac OS 系统执行：

```
1 brew install portaudio sox
```

2. 安装 **PortAudio** 的 **Python** 绑定：

```
1 pip install pyaudio
```

3. 要检查是否可以通过麦克风进行录制，请打开终端并运行：

```
1 rec temp.wav
```

步骤 3: 运行 DEMO

解码器结构

解码器 (decoder) 包含了以下的内容:

```
1 |—— README.md
2 |—— _snowboydetect.so
3 |—— demo.py
4 |—— demo2.py
5 |—— light.py
6 |—— requirements.txt
7 |—— resources
8 |   |—— ding.wav
9 |   |—— dong.wav
10 |  |—— common.res
11 |  |—— snowboy.umdl
12 |—— snowboydecoder.py
13 |—— snowboydetect.py
14 |—— version
```

`_snowboydetect.so` 是一个使用 **SWIG** 编译的动态链接库。它依赖于您的系统的 **Python 2** 库。所有与 **Snowboy** 相关的库都在此文件中静态链接。

`snowboydetect.py` 是一个由 **SWIG** 生成的 **Python** 封装文件。因为它很难阅读, 我们创建了另一个高级封装: `snowboydecoder.py`。

您应该已经从 <https://snowboy.kitt.ai> 获得了经过训练的模型文件 (例如 `snowboy.pmdl`), 或者您可以简单地使用 `resources/snowboy.umdl` 中的通用模型。

运行 DEMO

该 **DEMO** 可以在任何设备上运行。但是我们建议您: 在带有扬声器输出的笔记本电脑/台式机上运行它, 因为当您的唤醒词被触发时, **DEMO** 会播放 **Ding** 声音。

1. 为了运行 `snowboydecoder.py` 文件中的 `__main__` 函数的 `demo`, 请在终端中运行以下命令:

```
1 python demo.py snowboy.pmdl
```

这里的 `snowboy.pmdl` 是你从 <https://snowboy.kitt.ai> 下载的训练模型。

2. 当提示 (**prompt**) 时, 向麦克风说话, 看看 **Snowboy** 是否检测到你的魔术短语。

演示非常简单。以下是演示的代码:

```
1 import snowboydecoder
2 import sys
3 import signal
4
5 interrupted = False
6
7 def signal_handler(signal, frame):
8     global interrupted
9     interrupted = True
10
11 def interrupt_callback():
12     global interrupted
13     return interrupted
14
15 if len(sys.argv) == 1:
16     print("Error: need to specify model name")
17     print("Usage: python demo.py your.model")
18     sys.exit(-1)
19
20 model = sys.argv[1]
21
22 signal.signal(signal.SIGINT, signal_handler)
23
24 detector = snowboydecoder.HotwordDetector(model, sensitivity=0.5)
25 print('Listening... Press Ctrl+C to exit')
26
27 detector.start(detected_callback=snowboydecoder.ding_callback,
28               interrupt_check=interrupt_callback,
29               sleep_time=0.03)
30
31 detector.terminate()
```

主程序员在 `detector.start()` 里循环。每 `sleep_time=0.03` 秒, 将会执行下面的函数:

1. 检查填充有麦克风数据的环形缓冲区, 以查看是否检测到一个唤醒词。如果是, 则调用 `detected_callback` 函数。
2. 调用 `interrupt_check` 函数: 如果它返回 **True**, 则断开主循环并返回。

在这里, 我们为 `detected_callback` 分配了一个默认的 `snowboydecoder.ding_callback`, 以便每次听到你的唤醒词时, 电脑都会播放一个声音。

原文链接: <https://www.wandianshenme.com/play/smart-speaker-hotword-detection-engine-snowb>