###写在前面###

这一章将进入机器人语音交互的学习，让机器人能跟人进行语音对话交流。这是一件很酷的事情，本章将涉及到语音识别、语音合成、自然语言处理方面的知识。本章内容：

1.语音交互相关技术

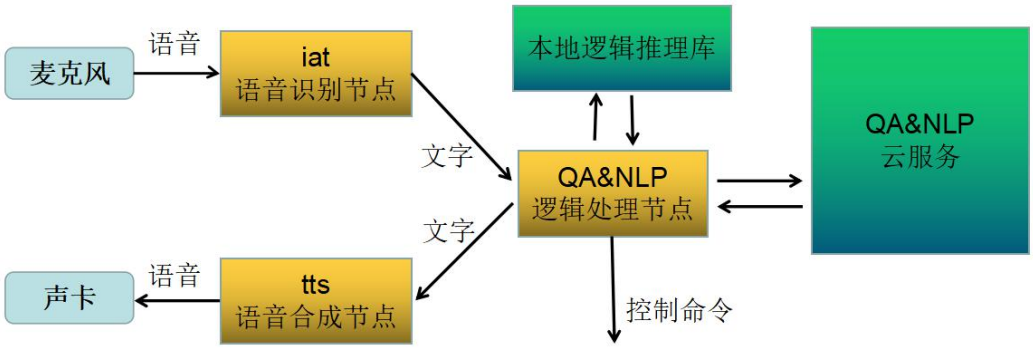
2.机器人语音交互实现

3.自然语言处理云计算引擎

###正文###

1.语音交互相关技术

要机器人能完成跟人对话，涉及到语音识别、语音合成、自然语言处理等技术。简单点说，语音识别就是将人的声音转换成文字便于机器人计算与理解；语音合成就是将机器人要说的文字内容转换为声音；自然语言处理相当于机器人的大脑，负责回答提问。整个语音交互的过程，如图 1。



（图 1）语音交互过程

1.1.语音识别

语音识别技术，也被称为自动语音识别 Automatic Speech Recognition(ASR)，其目标是将人类的语音中的词汇内容转换为计算机可读的输入，例如按键、二进制编码或者字符序列，如图 2。

（图 2）语音识别

语音识别技术所涉及的领域包括：信号处理、模式识别、概率论和信息论、发声机理和听觉机理、人工智能等等。语音识别技术的最重大突破是隐马尔科夫模型 Hidden Markov Model 的应用。从 Baum 提出相关数学推理，经过 Labiner 等人的研究，卡内基梅隆大学的李开复最终实现了第一个基于隐马尔科夫模型的非特定人大词汇量连续语音识别系统 Sphinx。此后严格来说语音识别技术并没有脱离 HMM 框架。当然神经网络方法是一种新的语音识别方法，人工神经网络本质上是一个自适应非线性动力学系统，模拟了人类神经活动的原理，具有自适应性、并行性、鲁棒性、容错性和学习特性，其强的分类能力和输入-输出映射能力在语音识别中都很有吸引力。但由于存在训练、识别时间太长的缺点，目前仍处于实验探索

阶段。

1.2.语音合成

语音合成是语音识别的逆过程，也称为文字转语音（TTS），它是将计算机自己产生的、或外部输入的文字信息转变为可以听得懂的、流利的汉语或其他口语输出的技术。如图 3。

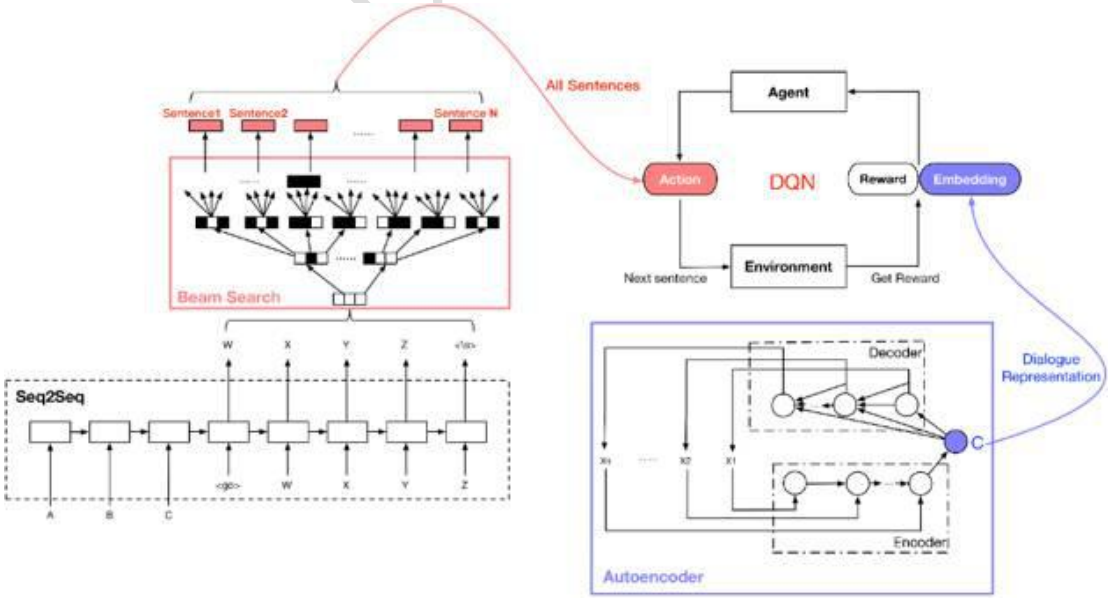
（图 3）语音合成

TTS 过程包括这些步骤：语言处理，在文语转换系统中起着重要的作用，主要模拟人对自然语言的理解过程，文本规整、词的切分、语法分析和语义分析，使计算机对输入的文本能完全理解，并给出后两部分所需要的各种发音提示；韵律处理，为合成语音规划出音段特征，如音高、音长和音强等，使合成语音能正确表达语意，听起来更加自然；声学处理，根据前两部分处理结果的要求输出语音，即合成语音。

1.3.自然语言处理

有了语音识别和语音合成，要让机器人能智能的对答如流的和人交谈，还需要赋予机器人以灵魂。自然语言处理技术（NLP）就是来赋予聊天机器人内在灵魂的。

NLP 是计算机领域与人工智能领域中的一个重要分支。由于数据的大幅度增强、计算力的大幅度提升、深度学习实现端到端的训练，深度学习引领人工智能进入有一个高潮。人们也逐渐开始将如日中天的深度学习方法引入到 NLP 领域，在机器翻译、问答系统、自动摘要等方向取得成功。经过互联网的发展，很多应用积累了足够多的数据可以用于学习。当数据量增大之后，以支持向量机（SVM）、条件随机场（CRF）为代表的传统浅层模型，由于模型过浅，无法对海量数据中的高维非线性映射做建模，所以不能带来性能的提升。然而，以 CNN、RNN 为代表的深度模型，可以随着模型复杂度的增大而增强，更好贴近数据的本质映射关系。一方面，深度学习的 word2vec 的出现，使得我们可以将词表示为更加低维的向量空间。另一方面，深度学习模型非常灵活，使得之前的很多任务，可以使用端到端的方式进行训练。



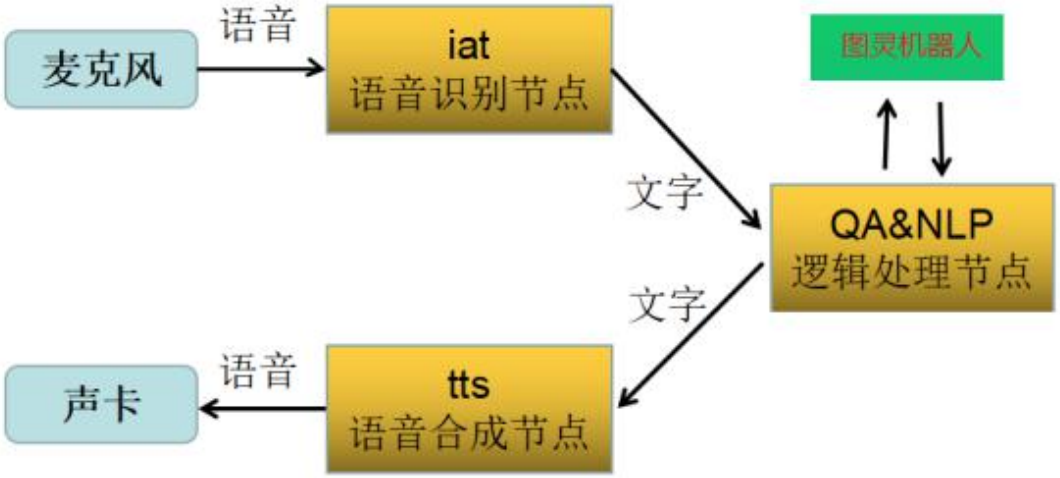
（图 4）基于深度学习的自然语言处理过程

为了让大家更好的理解基于深度学习的自然语言处理过程，举一个比较通用的模

型，如图 4。问题句子通过 Seq2Seq 循环神经网络进行预处理和编码，然后进入答案搜索，接着通过 DQN 强化学习网络对问答策略进程学习。这样，随着时间的推移，问答系统回答问题的水平会越来越高，就达到了不断在线学习的目的了。

2.机器人语音交互实现

其实要自己做一款语音对话机器人还是很容易的，我们只需要选择好语音识别、语音合成、自然语言处理的技术，就可以在一款树莓派 3 开发板上实现了。由于语音交互系统的核心技术是云端自然语言处理技术，所以我们可以选择网上免费提供的语音识别、语音合成等现有方案，将主要精力用在云端自然语言处理技术的研发上。语音识别与语音合成 SDK 有：科大讯飞、百度语音、Google…，对于我们墙内玩家…(Google 头疼)。经过我自己的实测，发现比较好用的免费 SDK 是科大讯飞家的，所以强烈推荐。为了测试方便，我先推荐图灵机器人 API 作为云端自然语言处理技术。等大家将整个语音交互系统的工作原理学会后，随时可以将图灵机器人 API 替换成自己的云端服务器，从而将主要精力转移到云端自然语言处理技术的研发上。说了这么多，我们先来看看咱们的机器人语音交互软硬件实现的真容吧，如图 5。



（图 5）机器人语音交互软硬件实现

USB 麦克风拾取声音，USB 声卡和音响播放声音，树莓派 3 开发板上运行语音识别、语音合成、QA 及 NLP 请求。其中，语音识别和语音合成采用科大讯飞的 SDK，QA 及 NLP 请求调用图灵机器人的 API 接口。

这里特别说明一下，为什么选用 USB 声卡而不用树莓派自带 AV 声卡的原因。你可以直接将耳机插口插入树莓派的 AV 接口试试，肯定很酸爽！杂音太大。这里就需要硬件支持。

杂音原因： 因为树莓派 3 的 AV 接口是音频和视频合并输出的，这个接口是美标接口，而在中国是国标的，接口的接地和音频是相反的，这就导致根本不能用了。另外对播放器的支持并不完善。

2.1.获取科大讯飞的 SDK

科大讯飞提供用于研究用途的语音识别、语音合成的免费 SDK，科大讯飞分发该 SDK 的形式是库文件（libmsc.so）+库授权码（APPID），库文件 libmsc.so 与库授权码 APPID 是绑定在一起的，这也是大多说商业软件分发的方式。

注册科大讯飞账号：

首先，前往讯飞开放平台（https://www.xfyun.cn），注册科大讯飞账号，注册好后，就可以进入自己的控制台进行设置了，如图 6。

（图 6）注册科大讯飞账号及登录

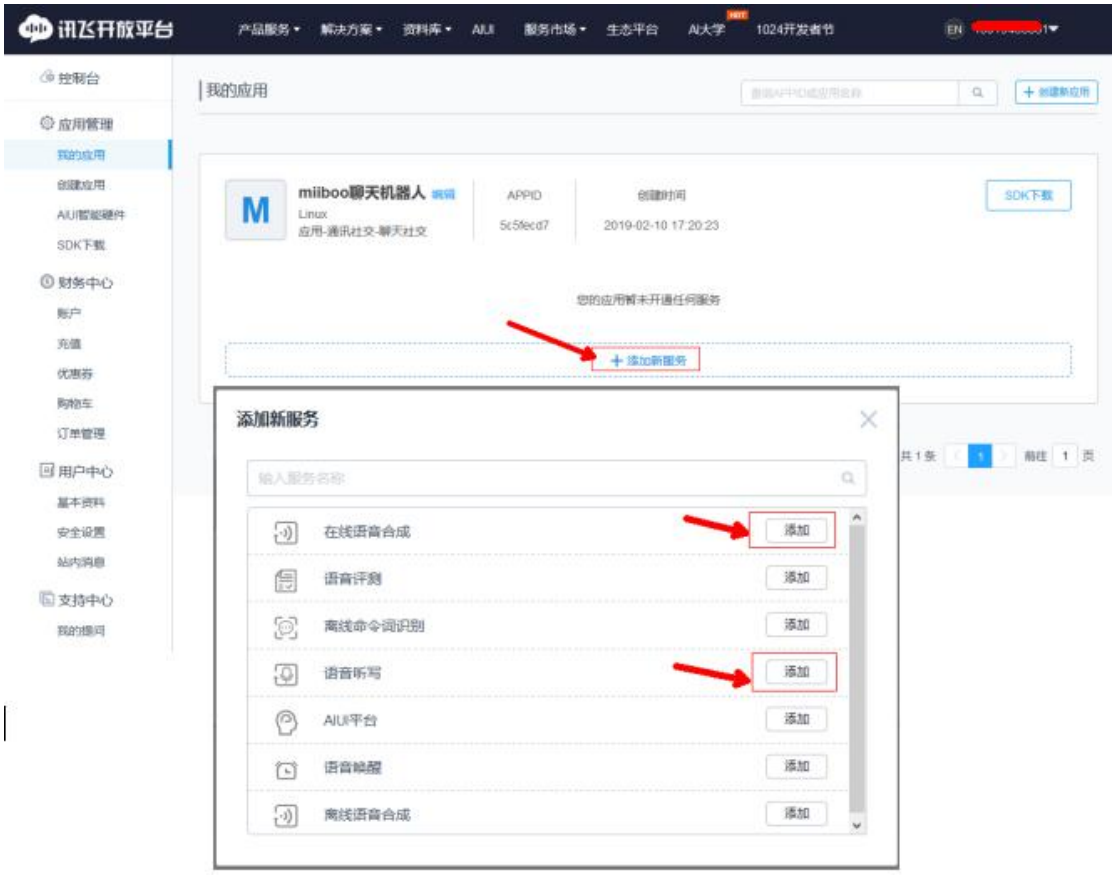
创建应用：

我们要在科大讯飞的开放平台创建我们需要的应用，这样讯飞就根据应用类型给我们生成对应的 SDK 库。

进入讯飞开放平台的控制台后，找到左侧栏的[创建应用]，按要求填写各个选项，注意[应用平台]一栏填 Linux，因为我们用的树莓派 3 开发板装的是 Linux 系统，如图 7。

（图 7）创建应用

创建应用完成后，就要给该应用添加相应的 AI 技能了，由于我们需要讯飞的在线语音合成、在线语音识别（也就是语音听写），所以添加这两个服务就行了。如图 8。



（图 8）添加语音合成与识别服务

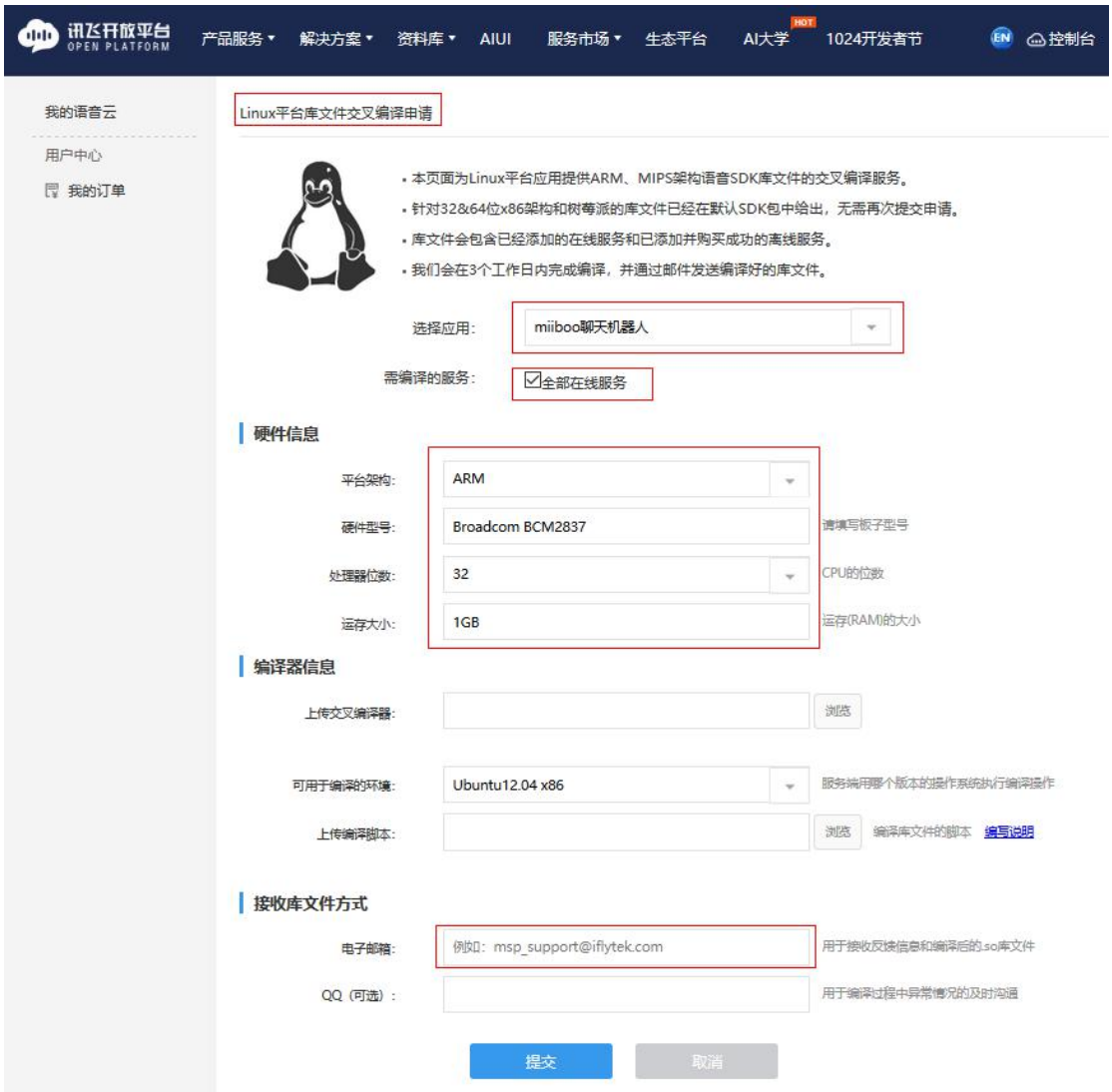
申请树莓派 3 平台对应的 Linux SDK 库：

由于科大讯飞开放平台默认只提供 PC 端 x86 架构的 Linux 库，所以如果我们想在树莓派 3（树莓派 3 为 ARM 架构）上使用科大讯飞的 Linux SDK 库，就需要另外申请。其实申请方法也很简单，进入科大讯飞中我的语音云页面：

http://www.xfyun.cn/index.php/mycloud/app/linuxCrossCompile

进行树莓派 Linux 平台库文件交叉编译申请，选择应用（必须是 linux 平台的应用），按照默认勾选全部在线服务，平台架构 ARM 硬件型号 Broadcom BCM2837（树莓派 3 代 b 型，即树莓派 3 的 SOC，其余版本树莓派，树莓派 2 为 BroadcomBCM2836，更早的版本为 BroadcomBCM2835），处理器位数 32，运行内存填了 1GB。最后记得填上自己的邮箱，提交后，如填写无误正确，你的邮箱将收到可下载库的链接，下载解压后得到 libmsc.so，这个库文件就是我们申请的树莓派 3 平台对应的 Linux SDK 库了。如图 9。关于交叉编译器和编译脚本，从这里 http://pan.baidu.com/s/1pLFPTYr 下载，具体交叉可以参考这一篇

http://bbs.xfyun.cn/forum.php?mod=viewthread&tid=32028&highlight=



（图 9）申请树莓派 3 平台对应的 Linux SDK 库

关于这个库文件对应的库授权码 APPID，可以在[我的应用]界面查看，如图 10。

（图 10）查看库文件对应的库授权码 APPID

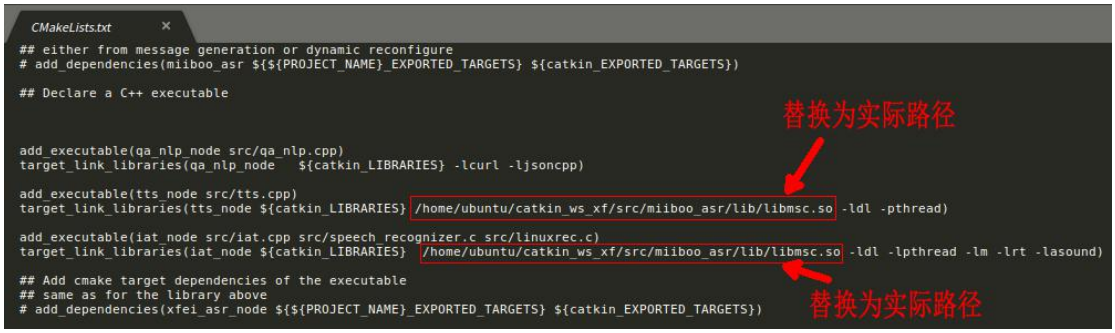
2.2.编译安装讯飞语音交互实例 ROS 版 DEMO

利用科大讯飞提供的 SDK 库文件和官方 API 说明文档，我们就可以开发出自己的语音交互实例程序，当然也可以开发对应的 ROS 程序。在我们的 miiboo 机器人上开发的语音交互 ROS

功能包叫 miiboo_asr。miiboo_asr 功能包文件组织结构，如图 11。其中 lib 文件夹下存放科大讯飞提供的 libmsc.so 库文件，iat.cpp 是语音识别节点源文件，tts.cpp 是语音合成节点源文件，qa_nlp.cpp 是 QA&NLP 逻辑处理节点源文件，其他的文件我们可以不用关心。

（图 11）miiboo_asr 功能包文件组织结构

了解了 miiboo_asr 功能包的基本情况后，我们就开始编译安装吧。首先，将 miiboo_asr 包拷贝到~/catkin_ws_apps/src/目录下。然后将上面申请到的树莓派 3 平台对应的 Linux SDK 库 libmsc.so 文件拷贝到 miiboo_asr/lib/中，并将 miiboo_asr/CMakeLists.txt 文件中有关 libmsc.so 的路径替换为你存放该 libmsc.so 的实际路径。如图 12。



（图 12）CMakeLists.txt 文件中有关 libmsc.so 的路径修改

接着我们需要将 miiboo_asr/launch/xf.launch 文件中的各个 appid、声卡硬件地址、麦克风硬件地址设置成自己实际的值。关于与 libmsc.so 库绑定的 appid 上面已经介绍了查看方法，而声卡硬件地址、麦克风硬件地址的查询也很简单。

麦克风硬件地址的查询直接使用命令 arecord -l，如图 13。

```
robot@robot:~$ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: CameraB409241 [USB Camera-B4.09.24.1], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
robot@robot:~$
```

（图 13）麦克风硬件地址的查询

在这里麦克风录制设备处于卡 1，设备 0，于是我们的麦克风硬件地址就是“plughw::CameraB409241”。

声卡硬件地址的查询直接使用命令 aplay -l，如图 14。

```
robot@robot:~$ aplay -l
**** 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
card 2: Device [USB Audio Device], device 0: USB Audio [USB Audio]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
robot@robot:~$
```

（图 14）声卡硬件地址的查询

在这里声卡播放设备有三个，卡 0 中的设备 0 为 3.5 音频输出，卡 0 设备 1 为 HDMI 音频输出，卡 2 设备 0 为 USB 声卡输出。这里我推荐使用 USB 声卡输出，所以我们的声卡硬件地址就是“plughw:DAC”。

在编译 miiboo_asr 前，我们还需要安装一些依赖项，其实就是麦克风录音和音乐播放工具，安装命令如下：

```
sudo apt-get update

sudo apt-get install libasound2-dev

sudo apt-get install mplayer
```

现在可以编译 miiboo_asr 了，编译命令如下：

```
cd ~/catkin_ws_apps/

catkin_make -DCATKIN_WHITELIST_PACKAGES=”miiboo_asr”
```

编译完成后，就可以运行语音交互节点来实现语音对话了，温馨提醒，请确保树莓派已连接网络，因为语音交互节点运行时需要访问网络。启动语音交互各个节点很简单，直接一条命令：

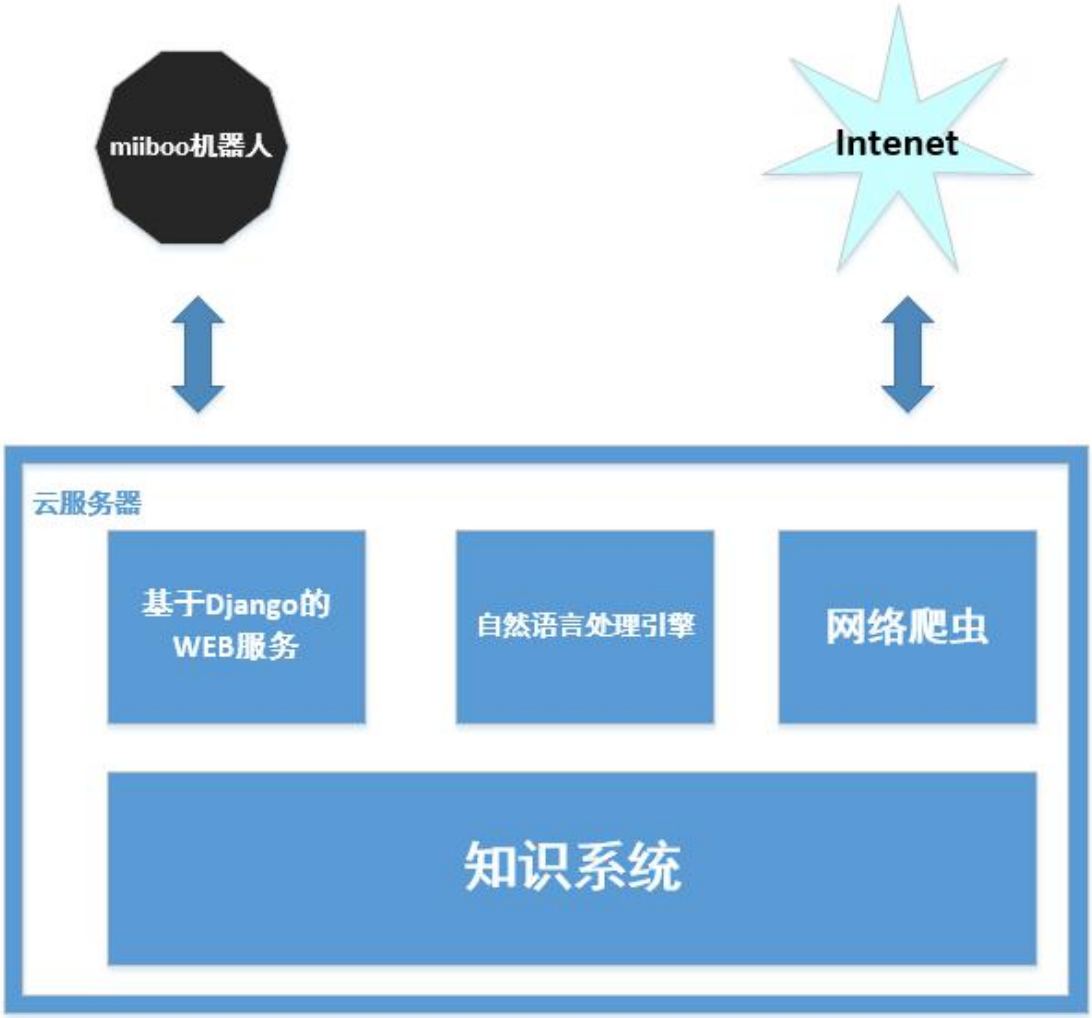
```
roslaunch miiboo_asr xf.launch
```

节点都运行起来后，会听到欢迎语句“你好，欢迎使用 miiboo 机器人语音控制系统”，之后就可以对着麦克风说出自己的指令，经语音识别被转换为文本，文本经图灵机器人得到应答，并通过语音合成使我们能听到回答的声音。这样一个语音交互的聊天机器人就诞生了，尽情享受和机器人聊天的乐趣吧^_^

这里说明一下，如果你使用我们的 miiboo 机器人，那么 miiboo 机器人上已经安装编译好了 miiboo_asr 功能包，所以只需要上面 roslaunch miiboo_asr xf.launch 这条启动命令，就可以开始机器人聊天之旅。但是，miiboo 机器人上安装的 miiboo_asr 功能包的 libmsc.so 的访问次数和频率是有限制的，只能供学习使用。如果大家需要将 miiboo_asr 功能包用来二次开发或实际应用，就需要按照上面的步骤去科大讯飞官网申请自己的 SDK 库了。

3.自然语言处理云计算引擎

这一节的内容作为展望内容，供大家参考和进一步的学习研究。前面也提过，语音交互系统的核心技术是云端自然语言处理技术，等我们采用成熟的方案将语音识别、语音合成等基础问题解决后，就要投入自然语言处理技术的研发了。我的想法是这样的，首先需要在云端服务器上搭建一个 WEB 服务器，然后需要有一个网络爬虫系统不断从互联网上爬去各种训练数据，接着需要搭建一个深度学习框架并运行在线学习算法不断的利用爬到的数据学习，这样自然语言处理云计算引擎同时处于学习和工作状态，图 15 是我构想的系统结构。



（图 15）自然语言处理云计算引擎