Weather Robot

This code enables the robot to speak whatever the user wants and then through natural language understanding (NLU) it has the ability to understand the intent and act accordingly.

Packages used:

- ROS lex_node
- google text to speech(gTTS python)
- SpeechRecognition(python)
- pygame

Requirements

- Ubuntu 18.04
- python 3.x(code working on 3.6)
- ROS melodic

ROS lex_node

This node allows the robot to communicate with a bot created in Amazon lex console which gives it the ability to understand an intent and respond accordingly .In my application specifically the robot simply responds to replies on the weather such as 'The weather today is sunny' or 'sunny' and the response is 'Thank you for your help!'. First you need to create an IAM user in your aws account with the following permissions:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "lex:PostContent",
                "lex:PostText",
                "lex:PutSession",
                "lex:GetSession",
                "lex:DeleteSession"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

and follow this aws guide to configure your user on your machine. To install simply follow the instruction in the official github of the package lex-ros1 which also contains the guide on how to configure your lex bot on this node.

gTTS and pygame

google text to speech library uses google translate's voice to transform any text input into speech. The package however doesn't play sound directly from code it only creates the mp3 file to be played. for this reason i also installed pygame to be able to play media directly, but feel free to use other media players if required. to install both:

```
$ pip3 install gTTS
$ pip3 install pygame
```

use pip3 in order to have the packages linked to the path of python3 in order for the code to work. for reference: gTTS

SpeechRecognition

The library supports several engines and APIs for performing speech recognition you can check those in the official site of SpeechRecognition. The API used in this project is Google Speech Recognition and since we want to capture audio from the microphone we also need to install Pyaudio to install this package:

```
$ pip3 install SpeechRecognition
```

#ROS service client The lex_node requires a rosservice call such as:

```
rosservice call /lex_node/lex_conversation "{content_type:'text/plain;charset=utf-
8',accept_type: 'text/plain; charset=utf-8', text_request: 'make a
reservation',audio_request: {data: ''}}"
```

these inputs however could be sent through code with the implementation of a ros service client which is exactly what my code does in this part:

```
content ='text/plain; charset=utf-8'
accept ='text/plain; charset=utf-8'
txtreq =r.recognize_google(audio)
rospy.init_node('service_client')
rospy.wait_for_service('lex_node/lex_conversation')
lex_conv = rospy.ServiceProxy('lex_node/lex_conversation', AudioTextConversation)
try:lex_resp=lex_conv(content,accept,txtreq,None)
except rospy.ServiceException as e:print("Service did not Process request: %s"%str(e))
text= lex_resp.text_response
```

you can follow the ros wiki for creating a rosservice client in python: http://wiki.ros.org/ROS/Tutorials/WritingServiceClient%28python%29

Notes

in case of facing errors in ROS and python version follow this article which was really helpful in solving the version errors i faced. The code can also work on python 2.7 however for the sake of our application we are working with python 3.

Todos

- Create a more advanced lex bot.

- Possibly introduce wake word functionality.(available also in [SpeechRecognition](#) snowboy)

Architecture Diagram