

Audio Annotator for ROS

Eleni Diamantidou

August 2016

1 Dependencies

- **OS : Ubuntu 16.04 edition**
- **Python 2.7.0**
- **Ros Kinetic**
<http://wiki.ros.org/kinetic/Installation/Ubuntu>
- **NUMPY**
`sudo apt-get install python-numpy`
- **MATPLOTLIB**
`sudo apt-get install python-matplotlib`
- **ffmpeg**
`sudo add-apt-repository ppa:mc3man/trusty-media`
`sudo apt-get update`
`sudo apt-get dist-upgrade`
`sudo apt-get install ffmpeg`
- **PyQt5**
install PyQt5 from Synaptics
*always run *sudo synaptic*
**if gstreamer not automatically installed, install also from synaptics
- **QtMultimedia**
install QtMultimedia from Synaptics

2 Libraries

- **pyAudioAnalysis**

install library from <https://github.com/tyiannak/pyAudioAnalysis.wiki>
the latest edition needs to install *sklearn* and *hmmlearn*

```
sudo apt-get install python-pip
```

```
sudo pip install -U sklearn
```

```
sudo pip install hmmlearn
```

3 Code files

- `audioGlobals.py`
- `rosbagAudio.py`
- `runFunction.py`
- `graphicalInterfaceAudio.py`
- `visualizeAudio.py`
- `onClick.py`
- `ganttChart.py`
- `editAnnotations.py`
- `saveAudioSegments.py`
- `trainAudioFromAnnotations.py`

4 Code file description

- **audioGlobals.py**

Contains all global variables of annotation tool.

- **rosbagAudio.py**

Take as argument a rosbag file. Isolate audio from rosbag. First of all save ros-topic and ros-message and then check if ros-topic is audio. Audio topic is a compressed mp3, so write audio in an .mp3 file. Convert mp3 file to wav. Call runFunction.py

- **runFunction.py**

runFunction has the *main* role. Take as arguments the wavFileName and the bagFileName. Extract Raw Audio from wav file. Calculate duration, frames and rate of audio signal. After that, search for a .csv file. The csv file contains saved annotations from older uses of tool. If csv exists, draw the annotations. Otherwise, call a classifier from pyAudioAnalysis library to estimate possible annotations of audio-signal. Basic condition is that the classifier should have train data (explain below for training and trained audio segments). Call graphicalInterfaceAudio to initialize annotation tool GUI

- **graphicalInterfaceAudio.py**

Create the application window of tool. Application Window contains two *matplotlib* widgets (draw a waveform and a gantt chart) and push buttons. Use QtMediaPlayer to play-pause-stop audio. Waveform widget is clickable.

- **visualizeAudio.py**

This file contains all plotting and drawing methods for audio annotations. Draw audio signal waveform with annotations. Draw cursors for segment selection and make bold selected area in right mouse click. Display annotation menu and add new speaker menu.

- **onClick.py**

This method counts both right and left mouse clicks on waveform. Also keep mouse coordinates.

- **ganttChart.py**

Create a gantt chart with all annotations. It's not clickable. Display only annotation classes.

- **editAnnotations.py**

Save changes in annotations. Change an existing annotation or create a new one. Delete an annotation. Merge annotations in case of overlapping same annotated classes. Create overlaps between different annotation classes. Refresh csv file, waveform and ganttChart plot.

- **saveAudioSegments.py**

Take as arguments .csv file and .wav file from annotation tool. Create folders for each class found in csv. Folder's names based on class labels. Save annotated segments in .wav files. Also save silence, the not annotated audio segments before, between and after annotations. The new files are stored in _ID.wav format.

- **trainAudioFromAnnotations.py**

Train classifier from pyAudioAnalysis Library. Training use data that saveAudioSegments has exported, for example Music, Speech, Activity, Silence audio segments.

5 Run Annotation Tool

Begin with running rosbagAudio.py :

python rosbagAudio.py 2016-07-04-16-19-14.bag

where 2016-07-04-16-19-14.bag the rosbag file. The graphical interface of tool will be initialized. Then you can be able to play and annotate audio files. This process export .mp3 and .wav file from rosbag.

Also note that all annotations are saving in .csv file as [startTime(ms), endTime(mS), Class Label]

eg.

776.3347771904, 3375.6592704286, Speech::George
4044.9464153941, 6862.1783511792, Speech::Eleutheria
6877.743168504, 9461.5028444173, Speech::Maria

To split audio into segments run:

python rosbagAudio.py 2016-07-22-13-24-10_audio.csv 2016-07-22-13-24-10.wav

To train pyAudioAnalysis Classifier run:

python trainAudioFromAnnotations.py

in *aT.featureAndTrain(['Silence/', 'Speech/', 'Music/', 'Activity/'], 1.0, 1.0, aT.shortTermWindow, aT.shortTermSvm, 'svmModelTest', False)* command: Silence, Speech, Music, Activity are folders that saveAudioSegments script has created and they contain audio wav files.

6 Display Annotations

All annotations have a class name/label and a color which corresponds to each class.

Specifically:

- Music - Red
- Activity - Magenta
- Laugh - Yellow
- Speech - Green

To define more than one speakers: each speaker has a unique ID (eg. Name). Annotation Color is a shade of green, so that different speakers can be visually separated.

7 How to use annotation tool

Audio Annotation Tool based its functionality on mouse clicks. Select area on waveform with two clicks: First click for the starting point and second click for the ending point. Then the selected area will appear between two cursors transparent. Right click inside the selected area for annotation choices.

Select an already annotated segment with right click above the segment to change or delete annotation.

Media Player:

In case there is no starting point the player will play the whole signal.

In other case, if only starting point is entered the player will play signal from the starting point until the end.

In the end, if both starting and ending point are entered the player will play the selected segment.