

MeetingStory (A Meeting Visualization Software)

Overview

MeetingStory takes in audio from a meeting and visualizes the meeting in a way that can potentially help with retaining the contents of that meeting. It utilizes voice recognition and machine learning to capture and process data.

The architecture contains four backend components: speech recognition, speaker recognition, topic modeling, and information extraction. The frontend is the web-based storyline system which uses python's Django and Javascript.

The root directory looks like this:

- **meeting** - folder where meeting audio file will go
- **training** - folder where voice training audio files will go
- **project** - folder where resulting will be after program is run
- **main.py** - file that contains code for speaker & speech recognition
- **transcript_generator.py** - file that generates transcript from audio
- **transcript_processor.py** - file that processes transcript
- **transcript_analyzer.py** - helper file

Tutorial

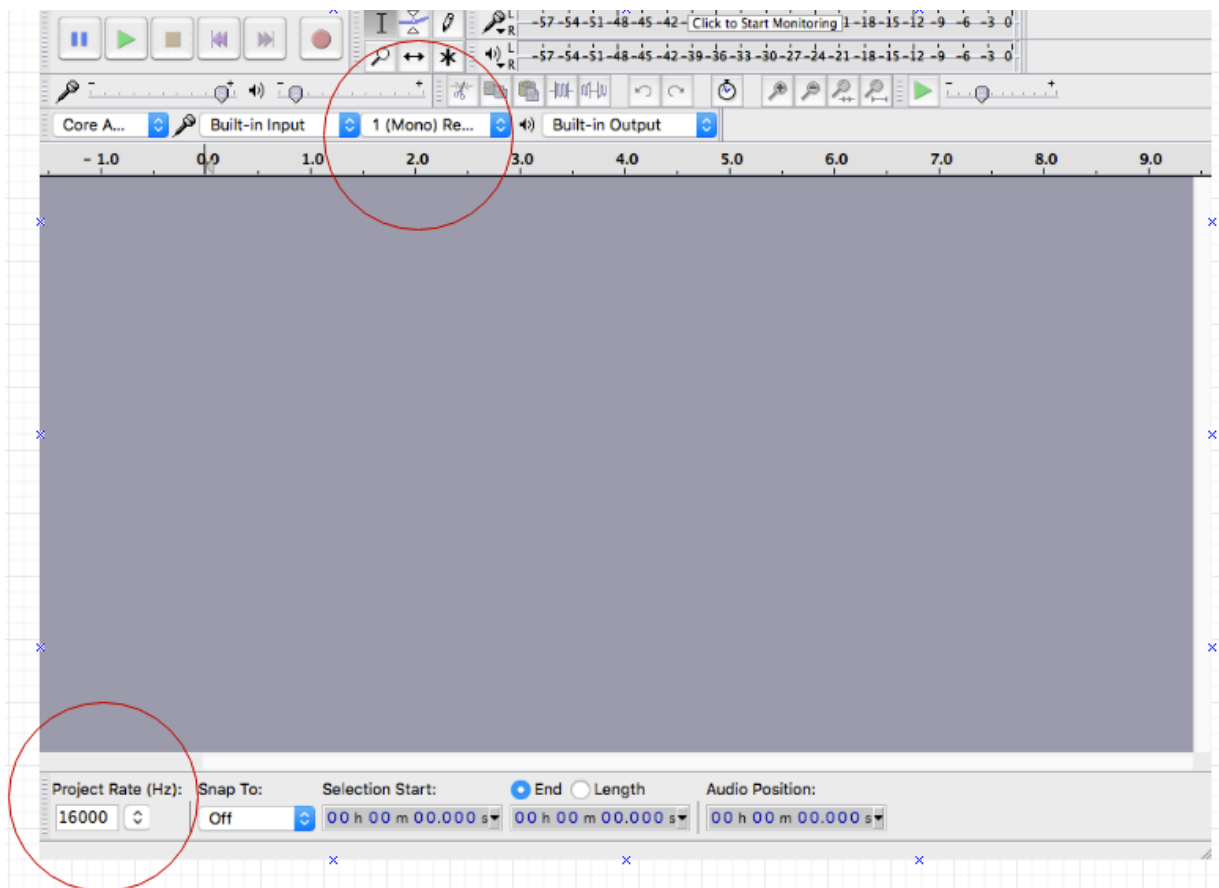
Step 0: Setting up the recording environment

It is important to maintain consistency with the recording device throughout all audio recordings. We recommend using Audacity to record the following audio. Download it [here](#). In order for the audio be compatible with Microsoft Speaker Recognition, keep the audio settings as follows:

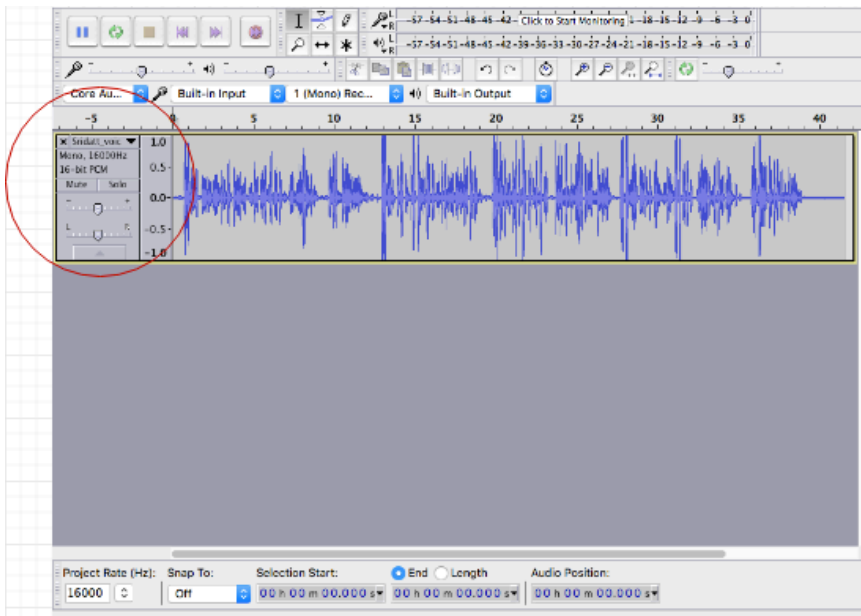
- **Container:** WAV
- **Encoding:** PCM
- **Rate:** 16k
- **Sample Format:** 16 bit
- **Channel:** Mono

The below images show where these settings can be found if you are recording with Audacity.

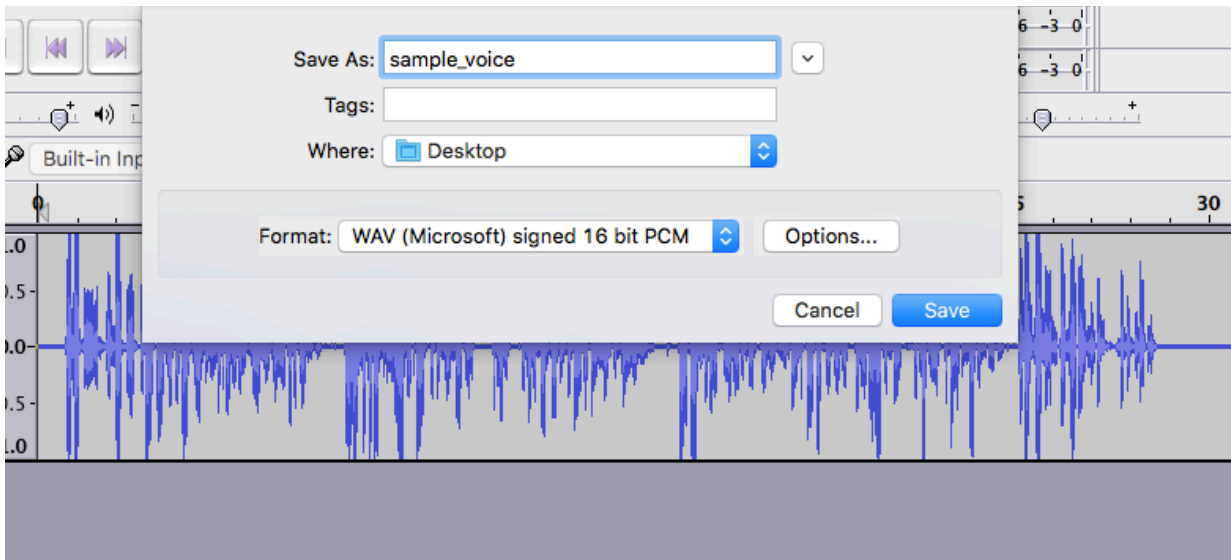
Before the recording, set the Project Rate on the bottom left, as well as the Channel towards the top.



After the recording:, make sure the Sample Format and the Encoding is set properly.



Before exporting, make sure the file is being exported as a WAV.



Step 1: Preparing Training Audio for Speaker Recognition

Record a sample audio file of at least 20 seconds for each member in the meeting, in the audio environment set up in Step 0. Read the following script:

Audacity is free software, developed by a group of volunteers and distributed under the GNU General Public License (GPL). Programs like Audacity are also called open source software, because their source code is available for anyone to study or use. There are thousands of other free and open source programs, including the Firefox web browser, the LibreOffice or Apache OpenOffice office suites and entire Linux-based operating systems such as Ubuntu

Name the audio file in the following format, in all lowercase:

[Person's name]_voice.wav

Place it in the **training** folder. This file will train the Microsoft Speaker Recognition API to recognize the individual's voice. Repeat for each member of the meeting.

Step 2: Preparing Meeting Audio for Speech Recognition

Record the meeting in the audio environment set up in Step 0. Name the audio file in the following format, in all lowercase:

[Meeting_title].wav

Place it in the meeting folder. This file can be converted to text by the Google Speech API and analyzed alongside the training audio.

Step 3: Running MeetingStory

Make sure you are in the root directory, and run the following command:

```
python main.py -m [Meeting_title] -s [Person 1's name] [Person 2's name] ...
```

Results

Inside ./project/data, you will see a folder with the following format:

[year][month][day][hour][min][second]

The folder name is derived from the timestamp of the meeting audio. For example, if the meeting began on June 1st, 2016 at 1:30 PM, the file would have the title “20160601133000”. Inside that folder, you will see the following files:

- **storyline.txt** - storyline of meeting
- **topic.json** - relevant topics extracted from meeting
- **transcript.txt** - transcript of meeting
- **user.json** - each user and their percentage contribution broken down by time segments
- **wordcloud.json** - words that appear on the topic bubble of the visualization