Lab: Voice Data

The purpose of this lab is to gain familiarity with speech data you might use to train an Automatic Speech Recognition (ASR) system. In the following steps, you'll:

- Explore the LibriSpeech data set and format
- Create your own audio files
- Build your own audio data set

As you complete each step , check it off in the task list that follows:

Lab Task List

- [ ] Clone the AIND-VUI-Lab-Voice-Data repository
- [ ] Visit the LibriSpeech web site
- [ ] Complete the LibriSpeech Corpus Quiz
- [ ] Extract and explore `LibriSpeech_Samples.zip`
- [ ] Complete the LibriSpeech Data Quiz
- [ ] Install the Sonic Visualizer
- [ ] Create five `.wav` files of about a sentence each
- [ ] View a spectrogram of your audio
- [ ] Create data set: Step 1 - convert and structure
- [ ] Create data set: Step 2 - add utterances
- [ ] Create data set: Step 3 - create .json file needed for processing

Lab: Voice Data

Clone the repository

To get started, clone or download the lab repository at AIND-VUI-Lab-Voice-Data. The reposititory contains some data and utility files for you to use in this lab.

Visit the LibriSpeech corpus web site.

Review the information found on the site landing page to answer the following quiz question.

LibriSpeech Corpus Quiz

QUESTION 1 OF 2

Which of the following are true about the LibriSpeech data set?

- The data set consists of 1000 hours of speech
- The data set costs $0.10 per hour of speech downloaded
- The data set is free to use
- The data set is segmented
- The data set includes French and Spanish
- The data set is in English
- The data set is appropriate for large scale speech training

Extract and explore **LibriSpeech_Samples**

Extract the **LibriSpeech_Samples** directory from the `LibriSpeech_Samples.zip` file in the **AIND-VUI-Lab-Voice-Data** directory. This sample includes the `README.TXT`, `BOOKS.TXT`, `CHAPTERS.TXT`, and `SPEAKERS.TXT` information files for you to explore. In addition, it contains a single path of data through **dev-clean/1993/147965/[1993-147965.trans.txt, 1993-147965-0000.wav, ...]**. "1993" is the speaker number and "147965" is the chapter number. You can look up which speakers and chapters these files correspond to in the information files. Within the chapter directory, there are `.wav` audio files and one transcription file.

The full LibriSpeech data sets are much larger, with many more speakers and chapters. There are `.flac` files rather than `.wav` files, which would need to be converted. This has been done for you for the lab. You will work with the larger corpus data set when you get to the Capstone project.

Refer to the **LibriSpeech_Samples** files to answer the following quiz.

LibriSpeech Data Quiz

QUESTION 2 OF 2

Explore the LibriSpeech sampler and check all the following *true* statements about it.

- [ ] The speaker is named Dawn
- [ ] The speaker is named Amanda
- The speaker is named Wendy
- [ ] The book that is being read from is "Frankenstein"

- [ ] The book that is being read from is "Pollyanna"

The transcripts have no punctuation other than apostrophes

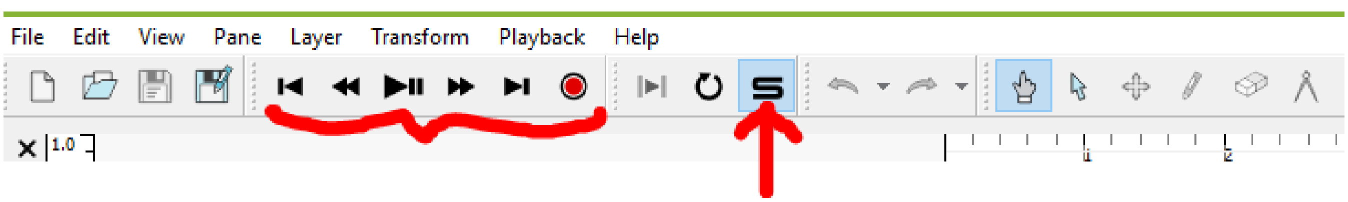
The transcripts are entirely in upper case

SUBMIT

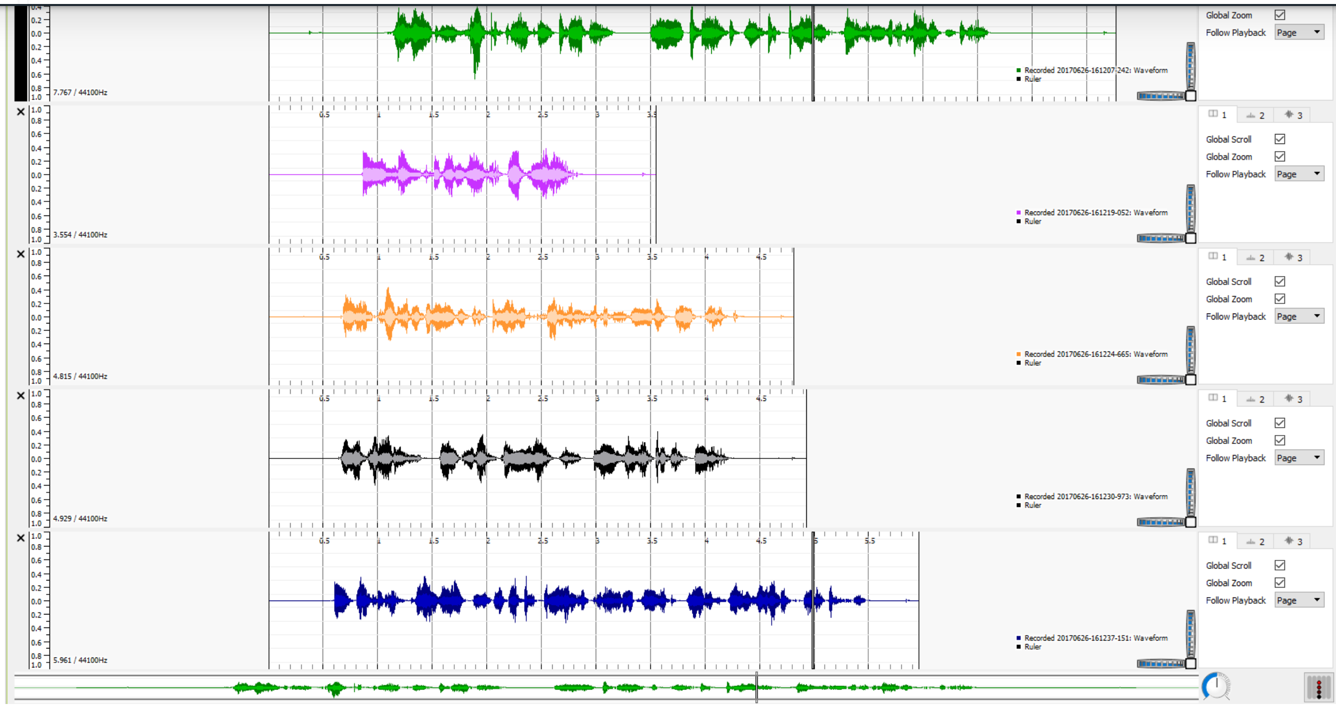
Sonic Visualizer

Download and install the free Sonic Visualizer.

Open the **Sonic Visualizer** application. The controls are fairly straightforward and include a red button for recording and the usual array of play buttons. Note the "Solo Current Pane" button, shown here with a red arrow, which will come in handy if you want to play back a single snippet when several are open.



Choose five sentences from a book or create your own. Record them one at a time with the Sonic Visualizer. You should see something like this:



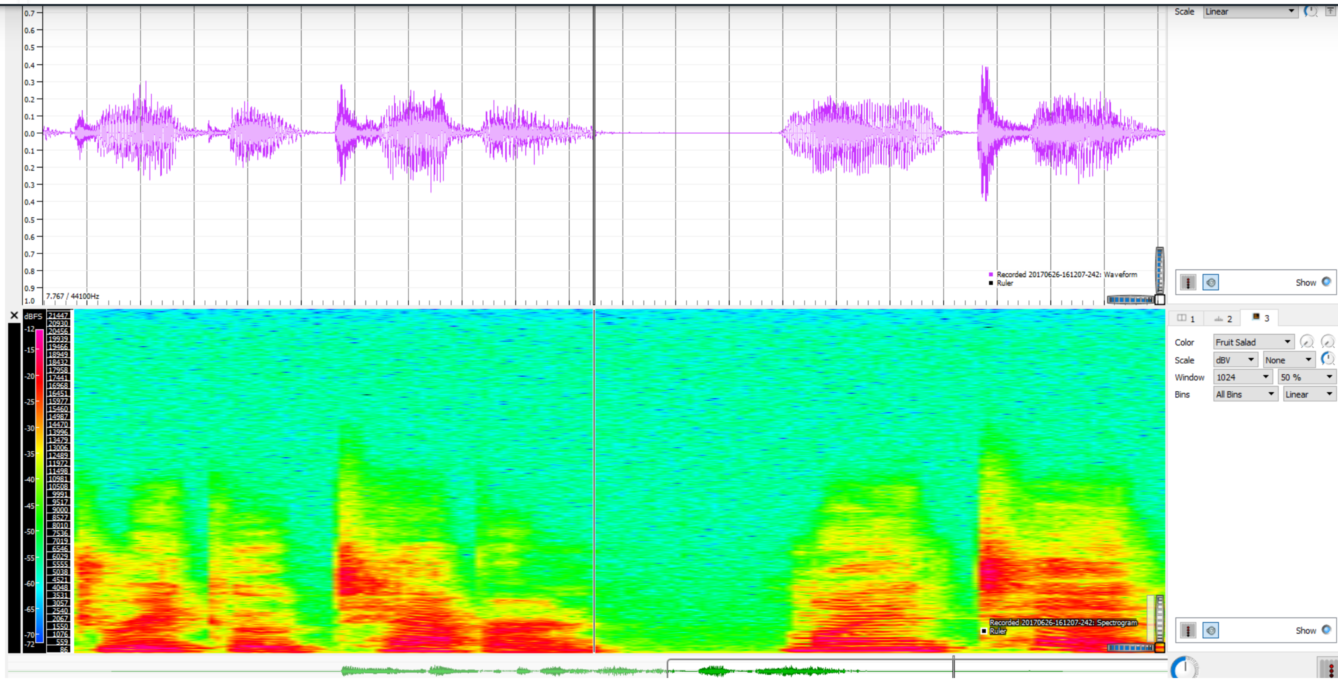
Create `.wav` files

Create a folder for your audio files called **my_audio** in your **AIND-VUI-Lab-Voice-Data** directory, and save each of these recordings there as a `.wav` file. Your audio recordings can be located with the `File->Browse Recorded Audio Folder` command or exported individually with the `File->Export Audio File` command.

Create a spectrogram

Spend as much time as you wish exploring the features of the visualizer. To see a spectrogram of your audio, try the `Pane->Add Spectrogram` command. To see it in multiple colors, change the color choice on the right side to "fruit salad". You may get a better view of the spectrogram by closing some of the panes first.

Lab: Voice Data



Build your data set - Step 1: Convert and structure

Open a terminal window in the **AIND-VUI-Lab-Voice-Data** folder. Set the environment to a python 3 environment, such as the conda **aind** environment created in previous projects, and install the pysoundfile library:

- Mac/Unix

```
$ source activate aind
$ pip install pysoundfile
```

- Windows

```
$ activate aind
$ pip install pysoundfile
```

The `.wav` files need to be converted from an IEEE-FLOAT format produced by Sonic Visualizer to a lower resolution PCM-16 format required in later processing steps. In addition, the audio files need to named and placed in a structure similar to the LibriSpeech file structure, i.e. sorted and identified by speaker and chapter. We need an

```
usage: convert_flt_pcm.py [-h]
                          input_directory data_directory group speaker chapter

positional arguments:
  input_directory  Path to input directory
  data_directory   Path to output data directory
  group            group
  speaker          speaker number
  chapter          chapter number

optional arguments:
  -h, --help       show this help message and exit
```

Convert the files with the following command (you can use different speaker and chapter numbers if you wish).

```
$ python convert_flt_pcm.py my_audio MySpeech my_dev 1 12345
```

Build your data set - Step 2: Add the utterances

You should now have a file structure with renamed `.wav` files in the **MySpeech/my_dev/1/12345** directory. There should also be a file named `1-12345.trans.txt` with the following lines:

```
1-12345-0000
1-12345-0001
1-12345-0002
1-12345-0003
1-12345-0004
```

Note these will have different ID's if you gave different "speaker" and "chapter" numbers during the conversion step. Add sentences that correspond to your `.wav` files with the same ID. You may need to "play" them to be sure of their contents. The utterances should contain all capital letters and no punctuation except for apostrophes where needed. Here's an example:

```
1-12345-0001 I'M A SCIENCE FICTION WRITER AFTER ALL
1-12345-0002 I'M SUPPOSED TO BE ABLE TO DEAL WITH QUESTIONS OF HUGE IMPORT
1-12345-0003 IN ADDITION I'M GOOD AT VIGNETTES AND I WANTED TO GET BETTER
1-12345-0004 I WANTED A FORMAT IN WHICH TO DEAL WITH THE SIMPLEST MOST UNIVERSAL
QUESTIONS
```

Build your data set - Step 3: Create `.json` file needed for processing

In order to use this data to train an ASR, the data generator needs a concise way to access the audio files and match them to the transcription. The following utility walks through the data structure and creates a `.json` description file.

```
usage: create_desc_json.py [-h] data_directory output_file

positional arguments:
  data_directory  Path to data directory
  output_file     Path to output file

optional arguments:
  -h, --help      show this help message and exit
```

In the terminal window, run the following:

```
$ python create_desc_json.py MySpeech/my_dev my_dev.json
```

That's it! Take a look at `my_dev.json` to make sure it contains the file descriptions. The example above yielded the following - yours should be similar but not identical:

Lab: Voice Data


```
estions of a certain type"}
{"key": "MySpeech/my_dev\\1\\12345\\1-12345-0001.wav", "duration": 1.69106575963
71883, "text": "i'm a science fiction writer after all"}
{"key": "MySpeech/my_dev\\1\\12345\\1-12345-0002.wav", "duration": 1.63374149659
86394, "text": "i'm supposed to be able to deal with questions of huge import"}
{"key": "MySpeech/my_dev\\1\\12345\\1-12345-0003.wav", "duration": 1.49043083900
22676, "text": "in addition i'm good at vignettes and i wanted to get better"}
{"key": "MySpeech/my_dev\\1\\12345\\1-12345-0004.wav", "duration": 1.69106575963
71883, "text": "i wanted a format in which to deal with the simplest most univer
sal questions"}
```


Be sure to check all the boxes in the Task List that you have completed!

NEXT