Multilingual Hispanic Speech in California Corpus: Corpus Processing Workflow

Julian Vargo
Department of Spanish & Portuguese
University of California, Berkeley

Introduction

In the current state of the MuHSiC corpus, a large quantity of sonix.ai transcriptions have been gathered and are ready to be edited. However, there is still no streamlined workflow to take hand-corrected sonix.ai transcriptions and convert them into usable .TextGrid files for linguistic analysis. Moreover, as linguistic publications have trended towards detailed elaboration of the audio and text processing workflows, having formal documentation of MuHSiC's corpus processing is critical for its usage in future linguistic publications.

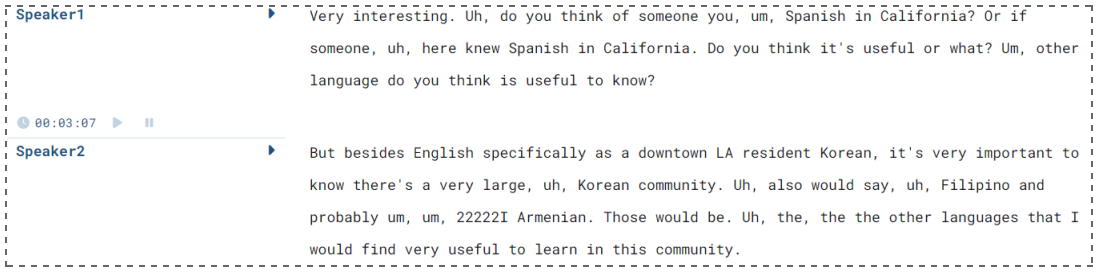
The present guide provides a workflow to effectively get raw sonix.ai into time aligned .TextGrid files which are ready for forced alignment. This guide is meant to be intelligible for Windows and Mac users with and without coding experience. There are two main steps to process all of the corpus's raw audio: (1) turn corpus raw audio into .TextGrid files which can be used on Praat and (2) force-align the .TextGrid files so linguistic analyses can be performed on the files.

Chapter 1: Turning Raw Transcriptions into .TextGrid Files

1.1: Hand-correct AI transcriptions of the corpus interviews

- The first part of this step should be self explanatory. Student researchers will correct any transcription errors found *through the sonix.ai interface* (Not through any other kind of word processor!).
- Manually change all interviewer tags to *Speaker 1*. Manually change all interviewee tags to *Speaker 2*. If the transcription accidentally picks up a third or fourth external voice in the audio, manually switch their *Speaker 3* or *Speaker 4* tag to *Speaker 1* (We don't want any instances of *Speaker 3* or *Speaker 4* in our files, or the final .TextGrids won't parse correctly).

Figure 1: Sonix.ai interface



Note the Speaker1 and Speaker2 tags on the left. Make sure the person conducting the interview is labeled as Speaker1 and mark the interviewee as speaker2. Any other outside speakers should be labeled as speaker1 to ensure the .TextGrids are encoded correctly.

Note how there is a clear transcription error, where the transcript says "22222I". These should be manually corrected. in the sonix.ai editor.

1.2: Export raw transcriptions as an .srt file

- In the top right corner of the sonix.ai interface, click *SubRip subtitle file (*.srt)*
- *Do you want to show speaker names? → Yes, show speaker names (as entered)*
- *How many lines? → Split subtitles by sentence*
- In your File Explorer (Windows) or Finder (Mac), place your raw files into a folder called *Input_SRT* (Make sure you know where you've stored this folder on your computer. You will need to copy the file path of that folder in a future step).
  *The code in this tutorial allows you to bulk convert .srt files, so feel free to include as many .srt files as you want in the Input_SRT folder.
  ***WARNING TO ALL WINDOWS USERS:** MAKE SURE THAT YOUR FOLDERS ARE NOT STORED IN ONEDRIVE. MOVE ANY FOLDERS YOU MAKE TO YOUR COMPUTER'S LOCAL DOWNLOADS OR LOCAL DOCUMENTS FOLDER. (A red flag that you used OneDrive is if you get coding errors that say "file path not found" during future steps)

1.3: Download Python

- Download Python for Windows: https://www.python.org/downloads/windows/
- Download Python for Mac: https://www.python.org/downloads/macos/

*Python is a computer coding language. When you download the language, you are *only installing the coding language*, but are not installing an application which lets you easily edit or run code. We need to separately install an application which lets you run your python code.

1.4: Install an application which runs your code

- Download Visual Studio Code (Versions available for both Windows and Mac): https://code.visualstudio.com/Download

*Visual Studio Code is essentially a word-processor like Microsoft Word or Google Docs, but you type code instead of prose.

1.5: Download .srt to .TextGrid file converter

Unfortunately, the raw .srt files and .TextGrid files are not commonly converted between one-another. .srt files are commonly used in the movie/video production sphere, while .TextGrids are used in the linguistics sphere. This means we need a custom script which cleans up our .srt files and then converts them into a .TextGrid

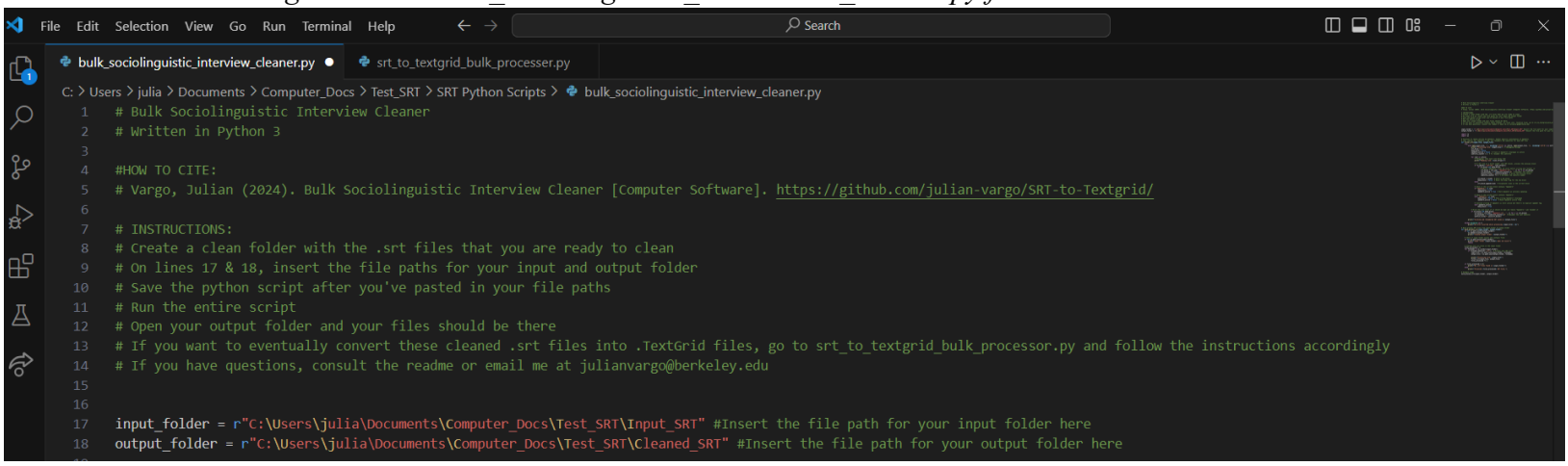
- Go to https://github.com/julian-vargo/SRT-to-Textgrid
- On the GitHub page, you'll see a variety of different files.
- Click on *bulk_sociolinguistic_interview_cleaner.py*
- Once you've opened the page *bulk_sociolinguistic_interview_cleaner.py*, click  on the right side of the screen to download the raw python file
- Go back to https://github.com/julian-vargo/SRT-to-Textgrid and click on *srt_to_textgrid_bulk_processor.py*
- Download *srt_to_textgrid_bulk_processor.py*, click the download arrow  on the right side of the screen to download the raw python file

**bulk_sociolinguistic_interview_cleaner.py* takes your raw .srt files and removes *Speaker1* from the subtitles (removes the transcriptions of the interviewer and any other unwanted external voices). It also removes the unwanted *Speaker2* tag from the .srt (.TextGrids will place the text *Speaker2* in the text section of an interview, so we need to remove this labelling). This is why we need to follow the *Speaker1* and *Speaker2* naming conventions defined in step 1.1
**srt_to_textgrid_bulk_processor.py* should only be run after you are done cleaning your .srt files with *bulk_sociolinguistic_interview_cleaner.py*. This script performs two tasks: it reformats the .srt files in a way that is more compatible with forced aligners, and it converts the .srt file into a .TextGrid file

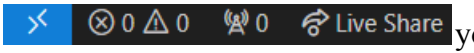
1.6: Open Visual Studio Code & *bulk_sociolinguistic_interviewer_cleaner.py*

- Open Visual Studio Code on your computer
- In the top left of the screen, click *File> Open File > bulk_sociolinguistic_interviewer_cleaner.py*

Figure 2: The bulk_sociolinguistic_interviewer_cleaner.py file on Visual Studio Code



1.7: Run *bulk_sociolinguistic_interviewer_cleaner.py*

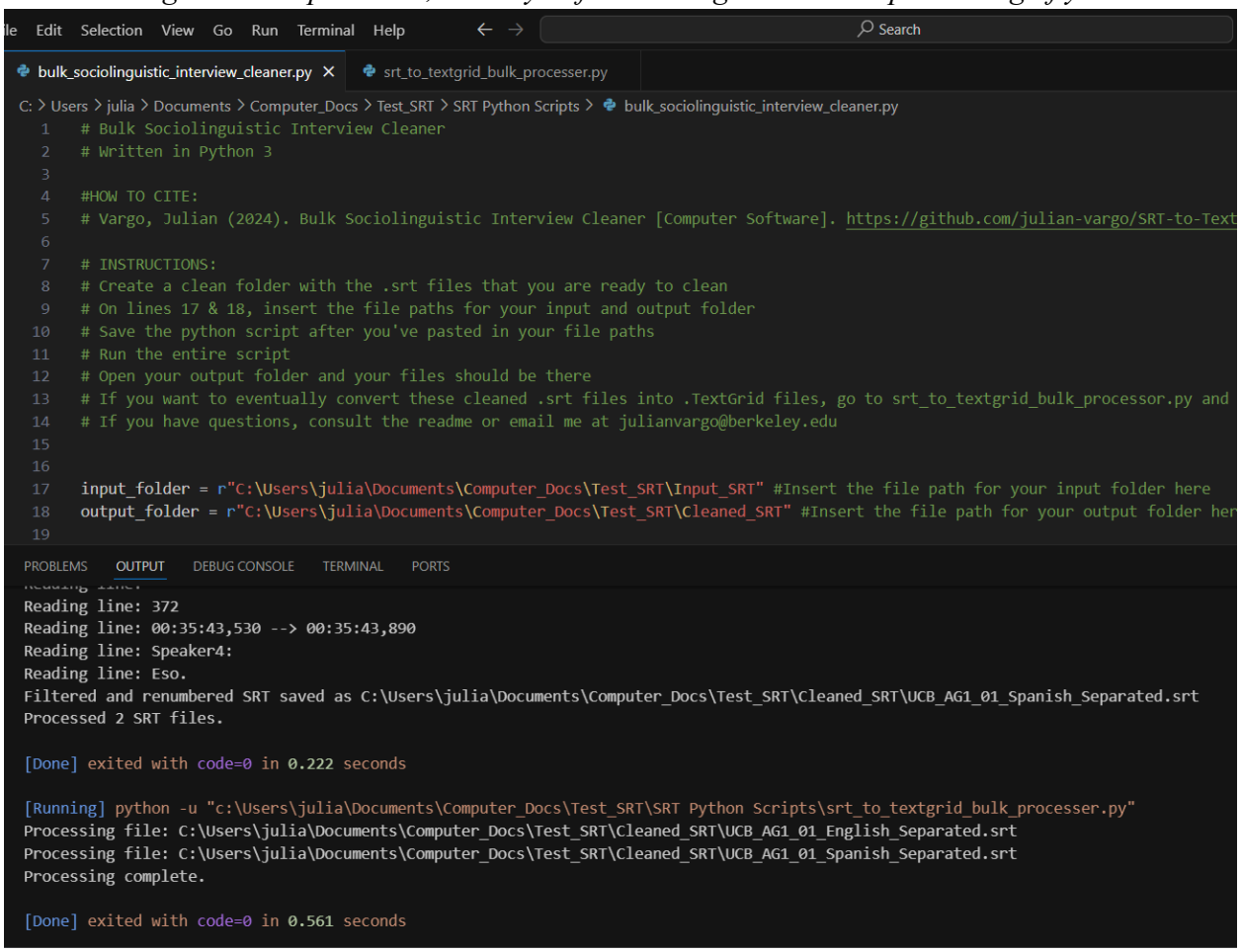
- Create a folder titled *Cleaned_SRT* (this is where your cleaned .srt files will go after you've run the script. Make sure you remember where you've stored this folder on your computer
- Go to line 17. Refer to the red text in Figure 2 where it says **"C:\Users\julia\...\Input_SRT"**. Replace this file path with the file path for your *Input_SRT* folder. To copy and paste a file path, right click on the folder in your File Explorer (Windows) or Finder (Mac), and then click *copy as path* (Windows) or click *copy Input_SRT as Pathname* (Mac).
- Replace **"C:\Users\julia\...\Input_SRT"** with your copied pathname with Ctrl+V (Windows) or Cmd+V (Mac).
- Go to line 18, Refer to the red text in Figure 2 where it says **"C:\Users\julia\...\Cleaned_SRT"**. Replace this file path with the file path for your *Cleaned_SRT* folder. To copy and paste a file path, right click on the folder in your File Explorer (Windows) or Finder (Mac), and then click *copy as path* (Windows) or click *copy Input_SRT as Pathname* (Mac).
- Replace **"C:\Users\julia\...\Cleaned_SRT"** with your copied pathname with Ctrl+V (Windows) or Cmd+V (Mac).
- In the bottom left, 

you may see an issue with the workspace not being trusted. Make sure to click, *Yes I trust the authors*, otherwise your code may not run. More info here are https://code.visualstudio.com/docs/editor/workspace-trust
- In the top left corner of Visual Studio Code, click *File > Save*
- In the top right corner of Visual Studio Code, click the triangular play button  called *run code*.
- Optional step: Open up the output menu (consult Figure 3) to check if you got any error or warning messages. To open this menu, click *Terminal* at the very top of the screen, then click *New Terminal*. In the bottom of the screen, you should see a separate interface

from your code. Click on *Output* to see the output messages after you run your code (this is where any error messages will pop up).

- If the code ran successfully, you should now have cleaned .srt files stored in your *Cleaned_SRT* folder.
  ***WARNING TO ALL USERS:**
  File paths on Windows use backslashes \
  File paths on Mac use forward slashes /

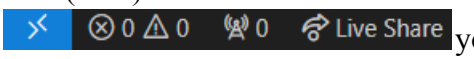
Figure 3: Output menu, where you find messages about the processing of your code



1.8: Open *srt_to_textgrid_bulk_processor.py*

- In your File Explorer (Windows) or Finder (Mac), create a folder titled Output_Textgrid Make sure you know where this folder is stored on your computer.
- In the top left of the screen, click File> Open File > *srt_to_textgrid_bulk_processor.py*
- In Visual Studio Code, you will now have two tabs at the top of the screen. Make sure you have the correct tab opened, titled *srt_to_textgrid_bulk_processor.py* (refer to *Fig. 4*)
- Go to line 16. Refer to the red text in Figure 2 where it says **"C:\Users\julia\...\Cleaned_SRT"**. Replace this file path with the file path for your *Cleaned_SRT* folder. To copy and paste a file path, right click on the folder in your File Explorer (Windows) or Finder (Mac), and then click *copy as path* (Windows) or click *copy Cleaned_SRT as Pathname* (Mac).
- Replace **"C:\Users\julia\...\Cleaned_SRT"** with your copied pathname with Ctrl+V (Windows) or Cmd+V (Mac).
- Go to line 18, Refer to the red text in Figure 2 where it says **"C:\Users\julia\...\Output_Textgrid"**. Replace this file path with the file path for your

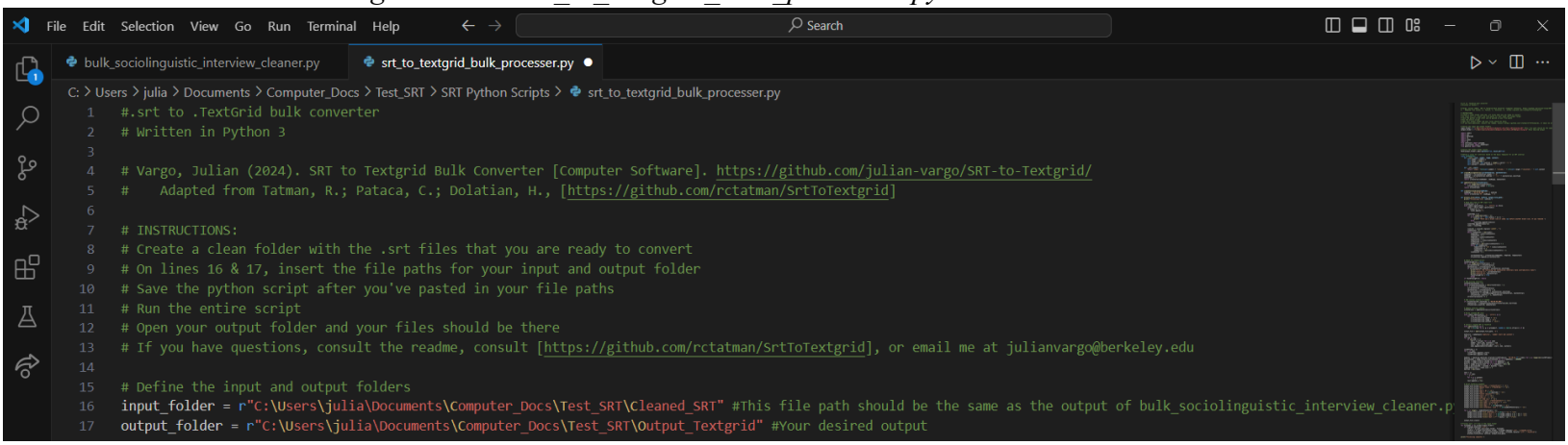
Output_Textgrid folder. To copy and paste a file path, right click on the folder in your File Explorer (Windows) or Finder (Mac), and then click *copy as path* (Windows) or click *copy Output_Textgrid as Pathname* (Mac).

- Replace **"C:\Users\julia\...\Output_Textgrid"** with your copied pathname with Ctrl+V (Windows) or Cmd+V (Mac).
- In the bottom left, 

you may see an issue with the workspace not being trusted. Make sure to click, *Yes I trust the authors*, otherwise your code may not run. More info here are https://code.visualstudio.com/docs/editor/workspace-trust
- In the top left corner of Visual Studio Code, click *File* > *Save*
- In the top right corner of Visual Studio Code, click the triangular play button  called *run code*.
- Optional step: Open up the output menu (consult Figure 3) to check if you got any error or warning messages. To open this menu, click *Terminal* at the very top of the screen, then click *New Terminal*. In the bottom of the screen, you should see a separate interface from your code. Click on *Output* to see the output messages after you run your code (this is where any error messages will pop up).
- If the code ran successfully, **you should now have cleaned .TextGrid files** stored in your *Output_SRT* folder.

*The output of *bulk_sociolinguistic_interview_cleaner* should be the input of *srt_to_textgrid_bulk_processor*. It is important that you run these files in the correct order.

Figure 4: The srt_to_textgrid_bulk_processor.py on Visual Studio Code



Chapter 2: Force Align the Textgrids

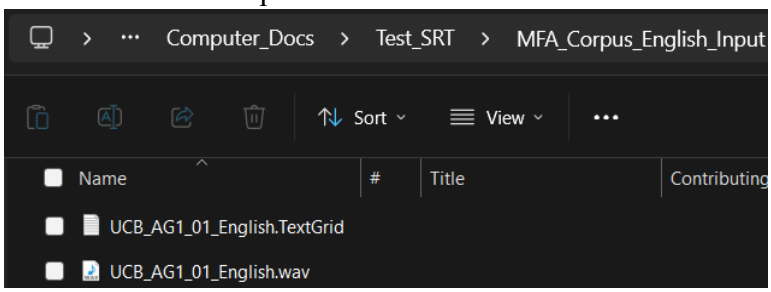
A forced aligner takes .TextGrid input files and automatically breaks entire words or sentences into an output .Textgrid files which have been divided into phonological segments. Forced aligners consist of two parts: a (1) acoustic model (an artificial intelligence model which reads your .TextGrid's spectrograms to tell where each phone starts and stops) and a (2) dictionary containing several likely words in your language. To get a forced alignment up and running, we need to download a couple pieces of software in advance.

2.1 Install Miniconda and Install Montreal Forced Aligner

- Install Miniconda at https://docs.conda.io/en/latest/miniconda.html
- (Windows) Open the app called *Anaconda Prompt*
- (Mac) Open the app called *Terminal*
- Type `conda update conda`
- You will get a prompt which says **`Proceed ([y]/n)?`**
- Type **`y`**
- After you've typed the letter **`y`**, click enter
- Type **`conda create -n aligner -c conda-forge montreal-forced-aligner`**
- You will get a prompt which says **`Proceed ([y]/n)?`**
- Type **`y`**
- Type **`conda activate aligner`**

2.2 Prepare English audios and English .Textgrids

- Place all of the English .wav files and English .TextGrid files in the same folder of your File Explorer (Windows) or the same folder of your Finder (Mac) titled *MFA_Corpus_English_Input*
- Make sure that your .wav and .TextGrid files have the same name. The forced aligner will go through every file in the *MFA_Corpus_English_Input* folder and will mess up if each.wav/.TextGrid pair does not have the exact same name:

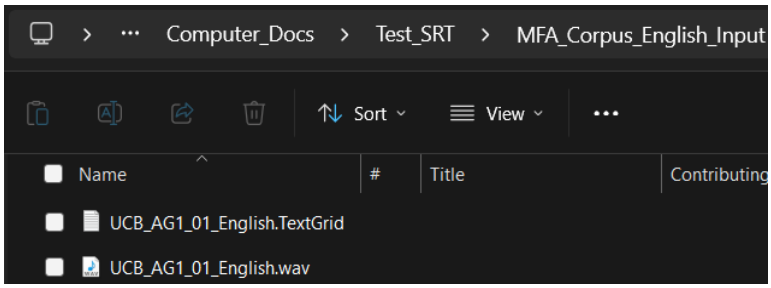


- Create a blank folder in your File Explorer (Windows) or Finder (Mac) titled *MFA_Corpus_English_Output*

*WARNING TO ALL WINDOWS USERS: MAKE SURE THAT YOUR FOLDERS ARE NOT STORED IN ONEDRIVE. MOVE ANY FOLDERS YOU MAKE TO YOUR COMPUTER'S LOCAL DOWNLOADS OR LOCAL DOCUMENTS FOLDER. (A red flag that you used OneDrive is if you get coding errors that say "file path not found" during your future steps)

2.2 Prepare Spanish audios and Spanish .Textgrids

- Place all of the English .wav files and English .TextGrid files in the same folder of your File Explorer (Windows) or the same folder of your Finder (Mac) titled *MFA_Corpus_Spanish_Input*
- Make sure that your .wav and .TextGrid files have the same name. The forced aligner will go through every file in the *MFA_Corpus_Spanish_Input* folder and will mess up if each.wav/.TextGrid pair does not have the exact same name:



- Create a blank folder in your File Explorer (Windows) or Finder (Mac) titled *MFA_Corpus_Spanish_Output*

2.3 Download acoustic and dictionary models

- Return back to *Anaconda Prompt* (Windows) or *Terminal* (Mac)
- Type **`mfa model download acoustic english_mfa`**
- Type **`mfa model download dictionary english_mfa`**
- Type **`mfa model download acoustic spanish_mfa`**
- Type **`mfa model download dictionary spanish_mfa`**

2.4 Align English Data

- Return back to *Anaconda Prompt* (Windows) or *Terminal* (Mac)
- Type **`mfa validate --clean \Users\...\MFA_Corpus_English_Input english_mfa english_mfa \Users\...\MFA_Corpus_English_Output`**
  - For the section highlighted in blue, copy and paste your computer's file path for your *MFA_Corpus_English_Input* folder. In yellow, copy and paste your computer's file path for *MFA_Corpus_English_Output*
  - This script checks for any problems in the unprocessed input .Textgrids. If you get any error messages which you do not know how to troubleshoot, please contact julianvargo@berkeley.edu with attached screenshots of error messages for assistance.
- Type **`mfa align --clean \Users\...\MFA_Corpus_English_Input english_mfa english_mfa \Users\...\MFA_Corpus_English_Output`**
- Your processed .TextGrid files should now be in the *MFA_Corpus_English_Output* folder

***WARNING TO ALL USERS:**
File paths on Windows use backslashes \
File paths on Mac use forward slashes /

2.5 Align Spanish Data

- Return back to *Anaconda Prompt* (Windows) or *Terminal* (Mac)
- Type **`mfa validate --clean \Users\...\MFA_Corpus_English_Input english_mfa english_mfa \Users\...\MFA_Corpus_English_Output`**
  - For the section highlighted in blue, copy and paste your computer's file path for your *MFA_Corpus_English_Input* folder. In yellow, copy and paste your computer's file path for *MFA_Corpus_English_Output*
  - This script checks for any problems in the unprocessed input .Textgrids. If you get any error messages which you do not know how to troubleshoot, please contact julianvargo@berkeley.edu with attached screenshots of error messages for assistance.
- Type **`mfa align --clean \Users\...\MFA_Corpus_English_Input english_mfa english_mfa \Users\...\MFA_Corpus_English_Output`**
- Your processed .TextGrid files should now be in the *MFA_Corpus_English_Output* folder

***WARNING TO ALL USERS:**
File paths on Windows use backslashes \
File paths on Mac use forward slashes /