Introduction

This is a documentation for 3 scripts whose combined purpose is to reduce the manual workload in text-to-speech alignment when some words are not found in the dictionary being used. Here, I'll refer to the Montreal Forced Aligner (MFA) but you may use other similar aligners relying on .lab and .wav files to run. The individual use of each script is laid out below:

1. GenerateLab.py serves to create .lab files from .wav files and a .docx file. The .lab files can then be used with the .wav files by running the MFA.
2. GenerateDict.py creates a French grapheme-to-phoneme (G2P) dictionary from the out-of-vocabulary (OOV) words following an attempt to use the MFA. It is based on the pronunciation of Basaa speakers. Note that this can be replaced by another program generating a G2P dictionary for another variety of French or for another language.
3. MergeDict.py combines multiple G2P dictionaries to get a broader dictionary which can then be used by the MFA.

I've written these as a Research Assistant under Dr Fatima Hamlaoui at the University of Toronto from June to August 2018.

As of 31 August 2018, the 'front' part of some French words containing apostrophes (such as qu', c', m' etc) are classified by the MFA as OOV even though they are in the dictionary being used. This prevents the generation of TextGrid files but hopefully, this situation will be resolved.

The code can be found at: https://github.com/Dragionic/automate-alignment

Should you need any clarification, you can contact me at: matthieuchanchee@gmail.com
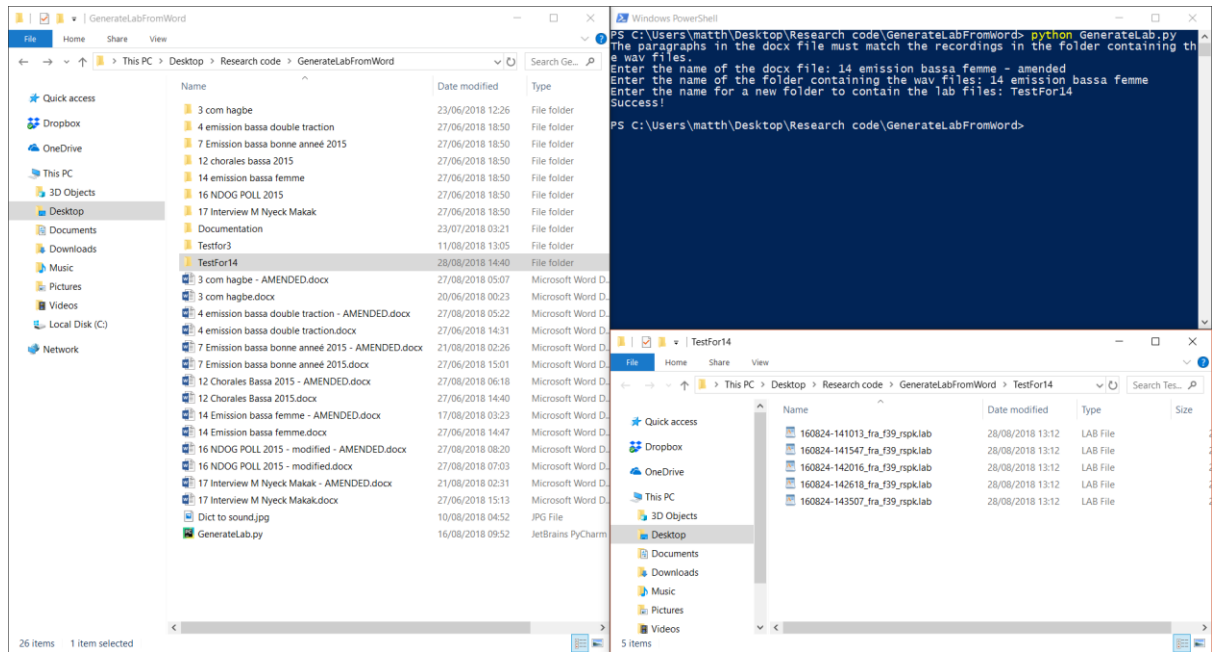

Requirements

You must have Python 3 installed for all three scripts.

1. GenerateLab.py:
   - The .docx file, the folder containing the .wav files, and GenerateLab.py must be in the same folder.
   - The paragraphs in the .docx file must correspond to the .wav files.
2. GenerateDict.py:
   - The file containing the list of OOVs, the file containing the original G2P dictionary, and GenerateDict.py must be in the same folder.
3. MergeDict.py:
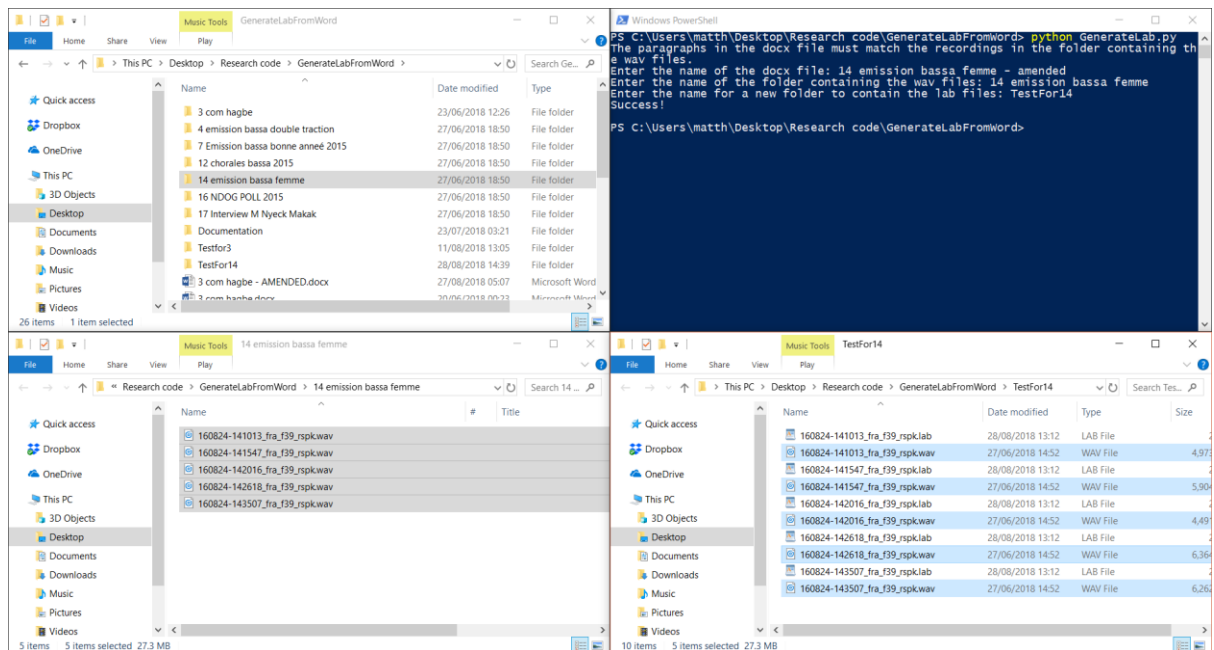   - The G2P dictionaries to be merged must be in the same folder as MergeDict.py.

Use

1. **GenerateLab.py:**
   a) Open your cmd/PowerShell/Terminal in the directory containing both the .docx file and the folder consisting of the .wav files. Run GenerateLab.py and follow the steps. A new folder containing the .lab files corresponding to the paragraphs in your .docx file and bearing the same names as the corresponding .wav recordings should be created.
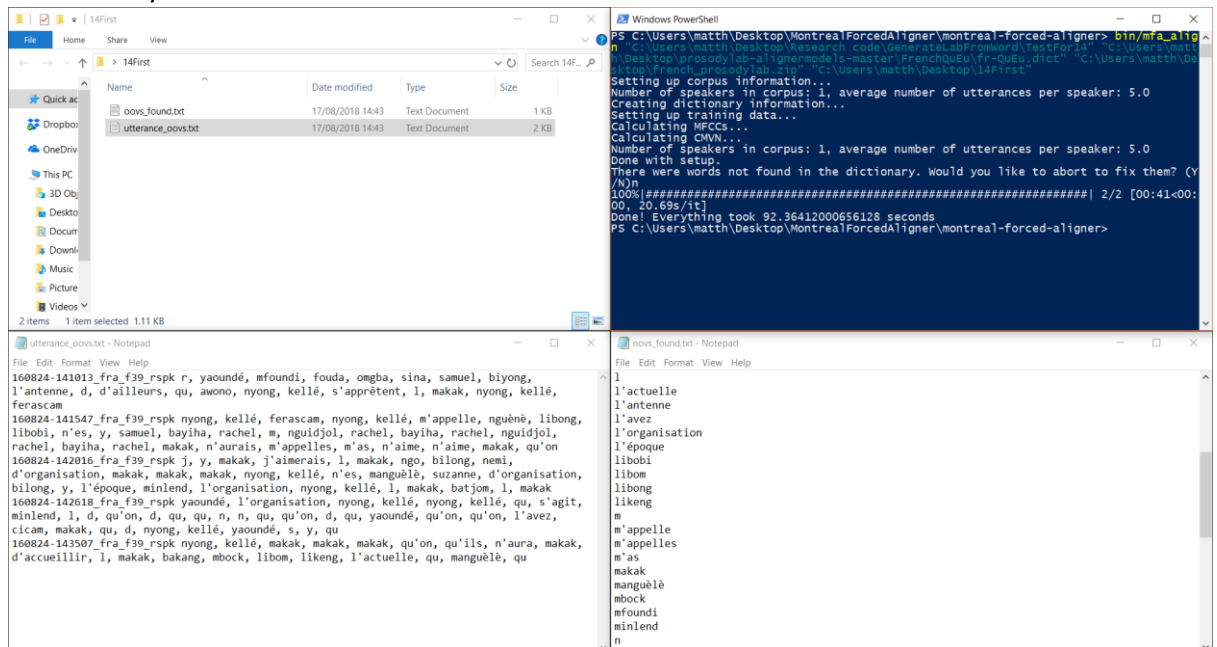

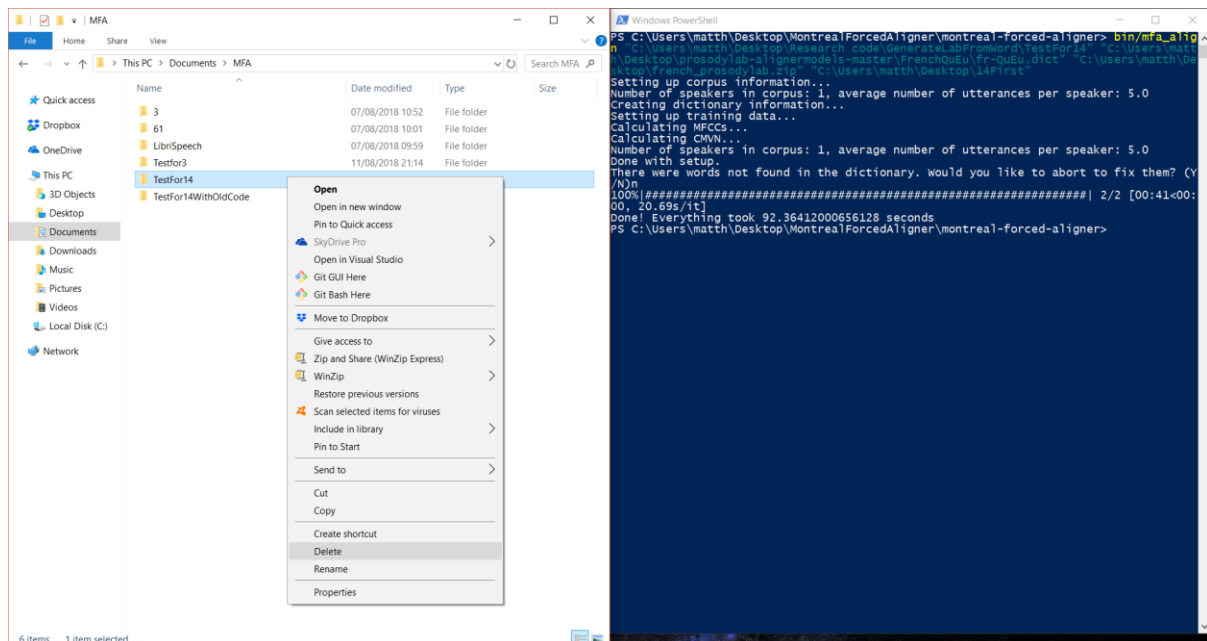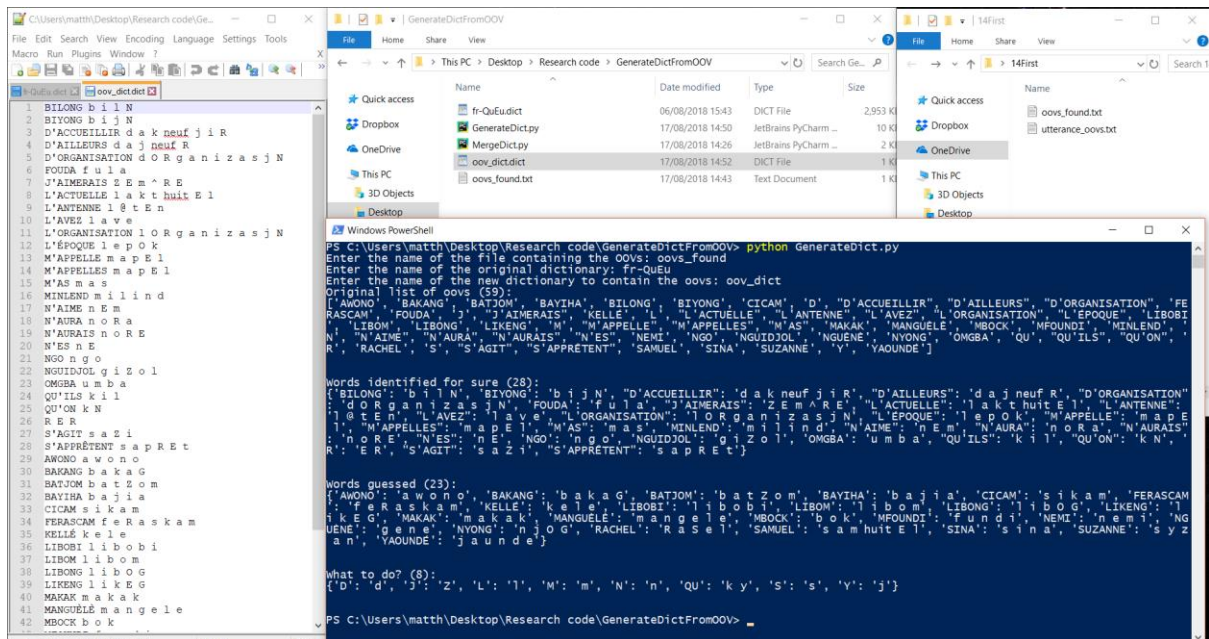   b) Copy and paste the .wav files into the newly created folder.

2. GenerateDict.py:

   a) Run the MFA. Here, the Prosodylab G2P dictionary
   (https://github.com/prosodylab/prosodylab-alignermodels/tree/master/FrenchQuEu) and
   acoustic model (https://montreal-forced-
   aligner.readthedocs.io/en/latest/pretrained_models.html) are being used. The folder 14First
   is created by the MFA.

   b) Delete the folder containing the logs generated by the MFA. This is because we'll use the
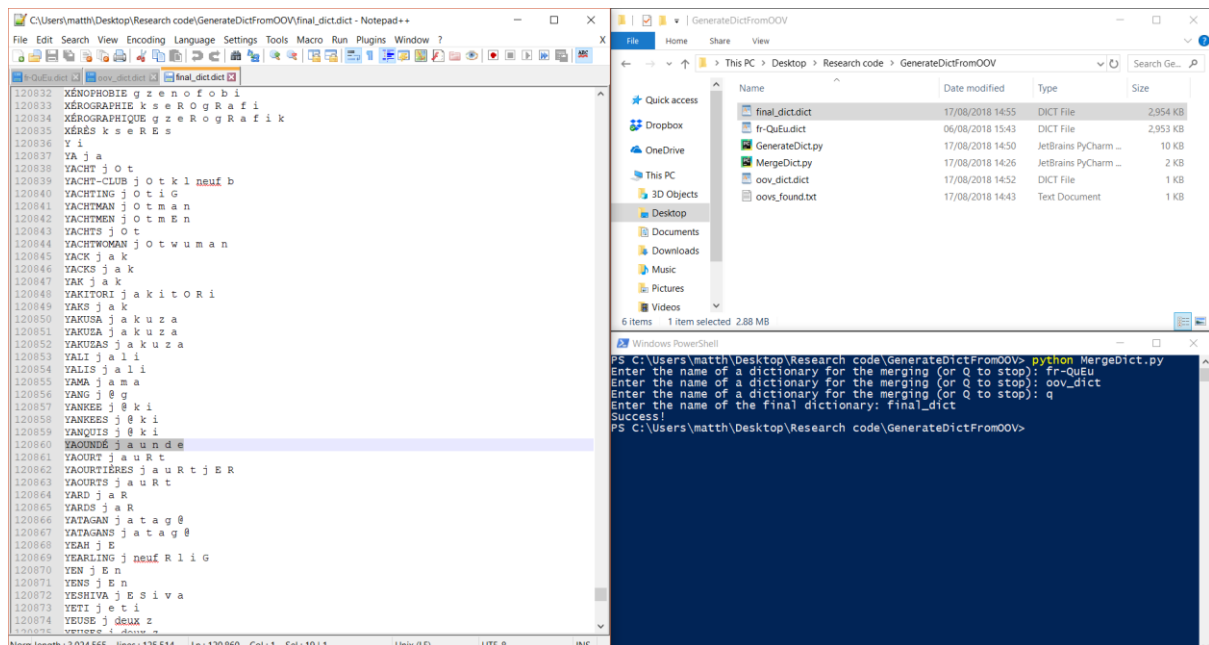   name 'TestFor14' again later.

c) Copy and paste our original dictionary (here, fr-QuEu) and the file containing the list of OOVs (here, oovs_found) into the folder containing GenerateDict.py. Then, run the latter to create a G2P dictionary for these OOVs (here, oov_dict).
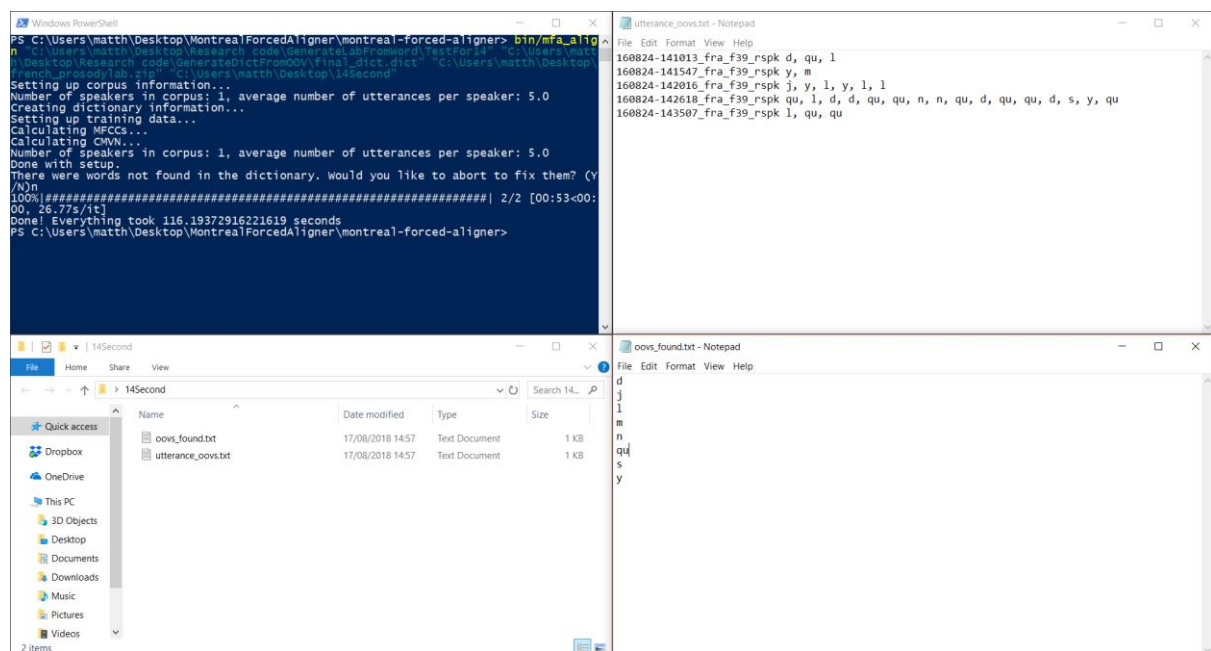

3. MergeDict.py:
a) Run MergeDict.py to combine our original dictionary and our OOV dictionary. This gives final_dict here.

b) Finally, run the MFA with our newly generated dictionary.

Notes

On all 3 scripts

In the following cases, an error message will be displayed and no file will be generated:

- A file or folder (e.g. the .docx file and the folder containing .wav files on step 1a) whose name was input does not exist.
- Any of the following characters was used to name a new file or folder:

$$\backslash, /, :, *, ?, ", <, >, |$$

- The name you entered for a new file or folder is already used.

On GenerateLab.py

An error message will be displayed and no file will be generated if number of paragraphs in the .docx file is not equal to the number of .wav files.

Some people end their .docx files by pressing several times on the Enter/Return key, thus creating 'empty' paragraphs. This program ignores empty paragraphs I've been able to predict but should other types of empty paragraphs show up, the code will have to be updated. As of 31 August 2018, an empty paragraph is defined as any of:

- Nothing (empty string)
- 1 to 9 spaces (' ', ' ', …, '   ')