Align audio files and transcriptions

using Lo Congrès alignment programs

1 Installation

You have to do these steps only once, to install the softwares you need if you want to align.

1.1 Install numpy

Install the numpy package with the command:
pip3 install numpy

1.2 Install aeneas

To install in Linux:

1. In a terminal, type:
wget https://raw.githubusercontent.com/readbeyond/aeneas/master/install_dependencies.sh

2. When the processing is done, type:
bash install_dependencies.sh

3. Open a terminal in the directory you want the application repository to be (ex.: Documents)

4. Type:
git clone https://github.com/ReadBeyond/aeneas.git

5. Type:
cd aeneas

6. Type:
sudo pip3 install -r requirements.txt

7. Type:
python3 setup.py build_ext --inplace

8. Type:
python3 aeneas_check_setup.py

9. To check if the program works, type:
python3 -m aeneas.tools.execute_task

In order for the programs to work, you will need to open the terminal **in the aeneas repository**.

1.3 Download and install Praat (optional)

If you want to check aeneas alignment (strongly recommended), you need to install Praat. You can download it here: https://www.fon.hum.uva.nl/praat. All the instructions to the installation are in the "download" pages.

1.4 Get the programs

Get the programs of Lo Congrès automatic alignment kit:
- add_variety.py
- align_aeneas
- align_variety.py
- convert_json_textgrid.py
- rename_files.py
- split_audio.py
- split_sentence.py

Copy and paste the program align_aeneas.py inside the aeneas repository.

2 Prepare files

Prepare a repository with all the audio files (in wav format). Be careful, **you can't have capital letters**, diacritical signs or whitespaces in files or repositories names.

2.1 Rename files if needed

If you want to automatically rename all the files in a repository to avoid uppercase and special characters, you can use the program rename_files.py with the command:
```
python3 rename_files.py repository_of_the_files_to_rename
```

This program will rename all of the repository files with removing special characters and lowercasing capital letters.

2.2 Format transcriptions

For every audio file, you need to have a .csv transcription file. It has to have the same name (except for the extension ; ex.: cercapais.wav and cercapais.csv). In the file, there must be one sentence per line, without any other information. Be careful, **you can't have capital letters**, diacritical signs or whitespaces in files names.

If you want to automate the sentence by sentence splitting from **txt** files, you can use the program split_sentence.py with the command:
```
python3 split_sentence.py repository_with_files_to_split
```

This program will create, for each txt file of the repository, a csv file with a sentence per line.

3 Align

Copy the repository containing the files inside the aeneas repository you created during installation at step 1.

Open a terminal in the aeneas repository and type the command:
```
python3 align_aeneas.py name_of_the_files_repository
```

The program creates .json files which are now in the repository with audio files and their transcriptions.

4 Convert in TextGrid

Convert json in TextGrid with the command:
python3 convert_json_textgrid.py files_repository

The files repository doesn't need to be in the aeneas repository anymore.

You get TextGrid files you can open with Praat software.

5 Correct Praat files

This step is optional but strongly recommended. You mostly need to check the beginning and the end of the file, and where there are blanks in the middle, because the aligner doesn't detect silences.

1. Open Praat.
2. In the Praat Objects windows, click on "Open" then "Read from file"
3. Navigate to the folder with the files. Select the .wav file and the corresponding .TextGrid file. Click on "Open".
4. In the box under "Objects:", select the two files.
5. Click on "View & Edit".
6. Correct: correct the texts when the transcription does not matches exactly what is said, move the limits around the text when they are not in the right place, add an interval if needed...
7. Save the TextGrid file overwriting the old one.

If you need a tutorial to use Praat, you can go here:
https://llacan.cnrs.fr/fichiers/manuels/Praat/PRAAT%20TutorialFR.pdf.

6 Make a file per sentence

Use the program split_audio.py with the command:
python3 split_audio.py files_repository

You get in a repository named "exports" all the audio files corresponding, each one, to a sentence. You also have, for each original file, a .csv file which matches the audio files name to the sentences uttered in them.

7 Add the variety information

If your language has dialectal variety or accents, you might want to indicate which one applies to your sentences. You can also use this field to indicate any other information (gender, age..).

7.1 If all the sentences in all the repository files are in the same variety

To add the variety at the end of each line of the linking files, use the program add_variety.py with the command:
python3 add_variety.py files_repository sentences_variety

Be careful, you need to specify the repository with the splitted files (the "**exports**" repository, if you didn't rename it since step 6).

You get, in the "exports" repository, files ended by "_var.csv" which contains:

- the audio file name

- the written transcription
- the sentence variety

7.2 If there are more than one variety in each file

Get back to the csv files created in step 2 and create a second column in which you will indicate, for each sentence, its variety. Save this file with the same name as the original file and **_var** at the end (ex.: cercapais_var.csv), field separator: §, no characters string separator. Let this file in the same repository than the original file (that is the parent repository of the "exports" repository).

Launch the program align_variety.py with the command:
```
python3 align_variety.py exported_files_repository
```

You get, in the "exports" repository, files ended with "_var.csv" which contains:
- the audio file name
- the written transcription
- the sentence variety

CC-BY-NC-ND