

## Using convertLM.py

**Purpose:** To convert a .TextGrid file from "Choi style" landmarks ("V", "t-cl", etc) to "Liu style" landmarks ("g", "-b", etc). The resulting file may be saved as a .TextGrid, a .lm (used for WaveShark), or both.

**Usage:** Run the python script with the .TextGrid file as an argument, like so:

```
python convertLM.py path/to/my_file.TextGrid
```

Note: convertLM.py must be run with python 3.x, not 2.7.x or earlier. Check which version of python you have with

```
python --version
```

**Tiers:** First it will ask you which tiers you want to convert. If you select all the tiers (default selection), any tiers that don't contain recognized Choi-style landmarks will be erased. If you select only certain tiers to convert, the other tiers will be saved as is. Enter your selected tiers as a comma-separated list, like so:

**Tiers to convert (default=0,1,2,3):** 2,3

You should select any tiers that have Choi-style landmarks, plus tiers that have information such as <OS>, <breath>, etc.

**Save as:** When it asks you what type to save as, enter a T (or leave blank) to save as a .TextGrid, l (lower case L) to save as .lm, or b to save as both.

Now the converter will run. When it finishes, it will save new files in the specified file types, and it will save **log.txt** which contains the list of landmarks that it didn't recognize.

**Adding unrecognized landmarks:** (Only recommended if you have some experience with python) If you run convertLM.py and see that log.txt has lots of landmarks which you recognize, but the program didn't, you can add them to convertLM.py and run it again to get a better result. Open up convertLM.py in a text editor and find the block of lines that start with `LMTypes.update` (line 83 at time of writing). Each line corresponds to a LM type. For example, the line

```
LMTypes.update({nr : "Nr" for nr in ["n","m","ng"]})
```

corresponds to the type "Nr" (nasal release), and says that the landmarks "n", "m", and "ng"

are of that type. To add a new landmark to this group, put it after the last item in the square-bracketed list, like so:

```
LMtypes.update({nr : "Nr" for nr in ["n","m","ng","new-lm"]})
```

Now, next time you run `convertLM.py`, each instance of "new-lm" will be treated like a nasal release type. **Don't** add the landmark types that end with -x or -+. Those are generated automatically. If you want a landmark with a -x or -+ to be included, add its "non-suffixed" version. e.g. to get "newlm-x" to be converted, just add "newlm" in the correct landmark type category.

**Questions/problems/suggestions/funny stories/etc.** Email me at [jessk@mit.edu](mailto:jessk@mit.edu)