Get measurements (Praat script)

Stefano Coretta

January 31, 2017

The script has the following components: a header, preparation code, and a main loop.

get-measurements.praat

```
<<<header>>>

<<<strings>>>

<<<results header>>>

<<<main loop>>>
```

We first need to create a list of the TextGrid files and initialise the result file. The extension `_no` in variable names means `_number`.

"strings"

```
textgrid_directory$ = "../data/derived"
result_file$ = "../results/results.csv"

Create Strings as file list: "list", "'textgrid_directory$'/*.TextGrid"
files_no = Get number of strings
```

We prepare the header of the result file and append it to it. The result file is saved in `results/`. Finally, let's save the indexes of each tier for easier referencing.

"results header"

```
header$ =
    ..."speaker,idx,word,beg_word,end_word,dur_word,beg_voic,end_voic,dur_voic,
    ...beg_mann,end_mann,dur_mann,rels,dur_vowel,dur_geminate,dur_clos,vor,
    ...voffr,mor,norm_voic,norm_mann,norm_vowel,norm_geminate,norm_clos,spread"
appendFileLine: result_file$, header$
```

```
sent = 1
word = 2
voic = 3
mann = 4
rels = 5
```

In the main loop, the script loops through every TextGrid file and for each file it extracts the relevant measurements.

"main loop"

```
for file to files_no
    selectObject: "Strings list"

    file_name$ = Get string: file
    Read from file: "'textgrid_directory$'/'file_name$'"

    intervals_no = Get number of intervals: word
    under_index = index(file_name$, "_")
    speaker$ = left$(file_name$, under_index - 1)

    index = 0

    <<<words loop>>>
endfor
```

The following chunk defines the code that loops through each word in the `word` tier. The variable `int_word` contains the numeric index of the current interval on the word tier. The loop calculates the following:

1. duration of voicing
2. duration of the glottal spreading, nasal, lateral, or rhotic gesture
3. time of release
4. ratio durations as proporsions of the Voice Onset to Release (VOR)

and it appends the results to the result file.

"words loop"

```
for int_word to intervals_no
    lab_word$ = Get label of interval: word, int_word

    if lab_word$ <> ""
        index += 1

        begin_word = Get starting point: word, int_word
        end_word = Get end point: word, int_word
        dur_word = (end_word - begin_word) * 1000
```

```
        <<<voicing>>>

        <<<manner>>>

        <<<release>>>

        <<<calculate normalised>>>

        result_line$ =
        ..."'speaker$','index','lab_word$','begin_word','end_word','dur_word',
            ...'begin_voic','end_voic','dur_voic','begin_mann','end_mann',
            ...'dur_mann','time_rels','dur_vowel','dur_geminate','dur_clos',
            ...'vor','voffr','mor','norm_voic','norm_mann','norm_vowel',
            ...'norm_geminate','norm_clos','son_spread'"
        appendFileLine: result_file$, result_line$
    endif
endfor
```

If the first interval on the `voic` tier is empty, then the next will always be non-empty.

"voicing"

```
int_voic = Get interval at time: voic, begin_word
label$ = Get label of interval: voic, int_voic
if label$ <> ""
    begin_voic = Get starting point: voic, int_voic
    end_voic = Get end point: voic, int_voic
    dur_voic = (end_voic - begin_voic) * 1000
else
    int_voic = int_voic + 1
    begin_voic = Get starting point: voic, int_voic
    end_voic = Get end point: voic, int_voic
    dur_voic = (end_voic - begin_voic) * 1000
endif
```

Since an interval on the `mann` tier will always be exactly below the end of the voicing interval, we can get the index of the `mann` interval with `end_voic - 0.00001` (`-0.00001` is needed since we would get the index of the following interval instead).

"manner"

```
int_mann = Get interval at time: mann, end_voic - 0.00001
label$ = Get label of interval: mann, int_mann
if  label$ <> ""
    begin_mann = Get starting point: mann, int_mann
    end_mann = Get end point: mann, int_mann
    dur_mann = (end_mann - begin_mann) * 1000
```

```
else
    begin_mann = undefined
    end_mann = undefined
    dur_mann = undefined
endif
```

In those cases where there was no clear release, we check that the current release time is between the beginning and the end of the current word.

"release"

```
ind_rels = Get nearest index from time: rels, end_word
time_rels = Get time of point: rels, ind_rels
if time_rels > begin_word and time_rels < end_word
    if label$ <> ""
        dur_clos = (time_rels - end_mann) * 1000
        son_spread = (end_mann - end_voic) * 1000
        dur_geminate = (time_rels - begin_mann) * 1000
    else
        dur_clos = (time_rels - end_voic) * 1000
        son_spread = undefined
        dur_geminate = dur_clos
    endif

    vor = (time_rels - begin_voic) * 1000
    voffr = (time_rels - end_voic) * 1000
    mor = (time_rels - begin_mann) * 1000

else
    dur_clos = undefined
    vor = undefined
    voffr = undefined
    mor = undefined
endif

if label$ <> ""
    dur_vowel = (begin_mann - begin_voic) * 1000
else
    dur_vowel = dur_voic
endif
```

"calculate normalised"

```
norm_voic = dur_voic / vor
norm_mann = dur_mann / vor
norm_vowel = dur_vowel / vor
```

```
norm_geminate = dur_geminate / vor
norm_clos = dur_clos / vor
```

"header"

```
########################################
# get-measurements.praat v1.0.0
########################################
# This script reads the TextGrid files in the specified folder and it extracts
# several measures from them.
########################################
# MIT License
#
# Copyright (c) 2016-2017 Stefano Coretta
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
########################################
```