Example Coding Approach for Syllable and Pauses

Kevin Hirschi

Prepared for AACL 2022 Pre-Conference workshop on Digital tools for building and analyzing spoken corpora…

Background:

The process outlined below was developed to leverage available scripts for syllable detection and allow for manual correction prior to corpus analysis. I use a very simple coding scheme to store the locations of syllables, pauses, and filled pauses in a TextGrid. The coding scheme can supply data for several common fluency measures.

Code	Meaning
s	Syllable
pause	Pause
fp	Filled pause

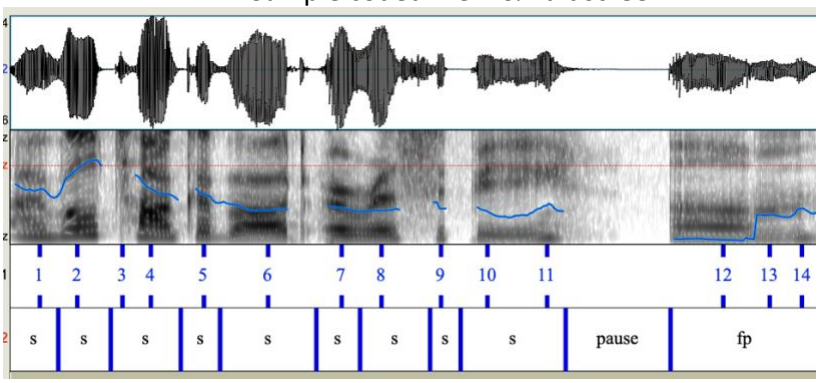
Step 1: Automated syllable and pause marking with de Jong and Wempe's (2009) script

- Install Praat Vocal Toolkit (https://www.praatvocaltoolkit.com/).
- With an audio file selected in the Objects, go to "Process" then "Mark Regions by Syllables." This Praat script by de Jong & Wempe (2009) detects loud and soft segments and marks them as syllables. It only looks at noise intensity. In other words, it does not consider a transcription or differentiate between speech and background noise, hesitations, or other sources of noise.
- The settings window will ask if you would like to adjust thresholds and pause durations. The defaults have seemed to work well in my experience.
- The output of the script will show some summary data, including the total number of syllables, pauses, duration, phonation time, and speech rate. Unfortunately, I'm not aware of an easy systematic way to collect these results when subjecting several files to the script.
- The script will also open the "View & Edit" mode for the audio file and newly created TextGrid.
  *Note: You can run more than one file at a time.

Step 2: Manual correction

- The advantage of viewing the audio with TextGrid information after the script is manual correction. Praat allows for mouse clicks to adjust the boundaries of the syllable region, a small circle to add a region boundary, and control-delete to delete a selected boundary.
- Using the coding scheme above, you can type "s" in the region for each syllable and "fp" for filled pauses. The script automatically marks silent pauses as "pause," but these also may require adjustment.
  *Note: High audio quality and monologic speech will result in fewer modifications. In my experience, 30 seconds of learner speech takes about 7 minutes to correct if the audio quality is high and the speaker is speaking an average pace. Roughly 15% of boundaries needed to be modified.

Sample coded View & Edit screen



Step 3: Save corrected TextGrid and consider analysis options

- Be sure to save the TextGrid outside of Praat in the .TextGrid format.

Step 4: Python Textgrid library

- Install from pip or similar major repository. Documentation is available here.
- Open a TextGrid and process labels and time markings. Particularly easy to use with Pandas.
- Sample script for fluency tag analysis. Complete script available at kevinhirschi.com.

```
## open textgrid library and for loop for textgrid files in directory
import textgrid
for file in os.listdir(directory_of_txts):

    ##open textgrid and look in interval tier
    tg = textgrid.TextGrid.fromFile(directory_of_txts+file)
    for tier in tg.tiers:
        if "<IntervalTier" in str(tier):

            ## create a Pandas dataframe with region details
            df = pd.DataFrame(columns=["int_name", "int_start", "int_end","int_length"])
            i = 1
            for interval in tier:
                df.loc[i] = [interval.mark, interval.minTime, interval.maxTime,(interval.maxTime-interval.minTime)]
                i += 1

            ## create a stat summary (pivot table) df
            df_stats = df[['int_name', 'int_length']]
            result = df_stats.groupby(['int_name']).agg({'count','mean', 'std','sum'})

            ##calculate number of syllables, avg syl lenght, and articulation rate
            num_syl = result['int_length']['count'].loc['s']
            avg_syl_len = result['int_length']['mean'].loc['s']
            arti_rate = round(1 / avg_syl_len, 2)
```