Automatic data processing with Praat

Deadline: 06.02.2008

8   Exercises

The goal of these exercises is to create a simple text-to-speech (TTS) script that simulates a telephone directory service subsystem. A string of digits is passed as textual input, and the output consists of a sound speaking these digits. The synthesis technique we will use is simple word-based concatenation.

1. Initially, we need some speech for our mini-database. Use Praat to record yourself speaking the utterance, "zero, one, two, three, four, five, six, seven, eight, nine." Make a slight pause before each digit. If you make a mistake, don't worry, you can re-record any mis-spoken digits individually.

2. Segment your recording(s) to obtain ten `Sound`s, each containing one digit, and no leading or trailing silence. Avoid pops by placing boundaries only at zero crossings. Save these `Sound`s as `0.flac`, `1.flac`, etc.

3. Write a TTS script that takes an arbitrary string of digits as textual input and loads and concatenates the corresponding Sound files into one `Sound`, finishing with the command `Play`.

4. Expand this script so that a short pause (200 ms) is inserted between every two consecutive digits. Also change the input parsing so that spaces may be present anywhere in the digit string, and for every space, make the corresponding pause 500 ms (instead of 200 ms) long.

   Either create the silent `Sound`s using appropriate commands or generate two such `Sound`s initially and load them as needed.

5. For every `Sound` in the mini-database, create three new ones. The first should have a flat intonation, the second should rise in pitch at the end, and the third should have falling pitch at the end.

   Create and use a `Manipulation` object for every digit. Each time, replace the pitch tier with a new `PitchTier`. For the flat `Sound`s, place a single pitch point at the start of the `PitchTier`, with a frequency value corresponding to the *overall* mean pitch of *all* your recordings. For rises and falls, the `PitchTier` should contain two points, one at the start (mean pitch) and one at the end (mean pitch $\pm 40\%$). Resynthesize (use the "overlap-add" technique) and save the 30 new `Sound`s as `0_flat.flac`, `0_rise.flac`, `0_fall.flac`, `1_flat.flac`, etc.

   Write a second script that performs all of these tasks for you.

6. Modify the TTS script so that the `flat` `Sound`s are used by default. Before a space in the input (i.e. before a pause in the output), select the `rise` version of the `Sound` instead, and for the last digit only, select the `fall` version.

7. Just to make the output sound more like an analog phone line transmission, add some noise (use Praat's `randomGauss()` function with a standard deviation of `0.01`) and filter the resulting output `Sound` so that only frequencies between 400 Hz and 3400 Hz survive.

Extra. Modify the TTS script so that a slight echo (delay: 750 ms, amplitude factor `0.3`) is introduced before adding the noise.

Running the finished TTS script as shown should sound comparable to the attached file `tts.flac`.

`tts.flac`

```
$ praat tts.praat 0681 302 4691
```