

CECS 451
Assignment 11
Total: 60 Points

General Instruction

- Submit your work in the Dropbox folder via BeachBoard. (Not email or in class)
 - Submit the separate files as they are. (no zip file)
-

1. Evaluate the performance of Google Web Speech API.

- (a) Find the `speech.py`.
- (b) Read `How Speech Recognition Works.txt` which includes 25 sentences, and record your speech as separate WAV(PCM) files using the nomenclature '`Group ID-Sent#.wav`'. For instance, `01-Sent01.wav`. I recommend you to use Audacity to record and edit your speech.
- (c) (10 points) Zip all audio files using the name '`Group ID`' and upload the zip file into 'BeachBoard - Discussions - Lab - How Speech Recognition Works' by clicking 'Start a New Thread'. Write your **Group ID** at the subject line and attach the zip file. **This part is due by the end of this week.**
- (d) (5 points) Complete the `read_original` method that imports the '`How Speech Recognition Works.txt`' into a list of strings, `self.original`, in the order of the sentence number.
- (e) (20 points) Refer *The Ultimate Guide To Speech Recognition With Python – Real Python.mhtml*, and complete the `conv_audio` method that converts audio files into a list of strings, `self.recognized`. The method should convert all audio files (`.wav`) in the folder, `inDir`, in the order of `Sent#`.
- (f) (10 points) Complete the `comp_string` method that compares two lists of strings, `self.original` and `self.recognized`, and calculates the similarities of two strings by using *Levenshtein Distance* (LD). The normalized Levenshtein distance (NLD) is defined by

$$NLD = \frac{LD}{\max(\# \text{ of words in the original}, \# \text{ of words in the recognized})}$$

(You need to convert the strings into the lists of words. For instance, '`I love AI`' to [`'I'`, `'love'`, `'AI'`]. Please refer this *site*.) This method should store the separate distances (NLD) in `self.distances` in the order of `Sent#`.

- (g) Visit 'BeachBoard - Discussions - Lab - How Speech Recognition Works' again, and download all submissions of other groups. Since we have 23 groups, (ideally) you will have 23 audio files for each sentence.

- (h) (15 points) Repeat running `speech.py` for all downloaded audio files, and collect the distances by sentence IDs. Draw a *box-and-whisker plot* of NLD by sentence ID as shown in Figure 1. You may use `Seaborn`, `Pandas`, or `Matplotlib` to draw the plots.
- (i) Submit your `speech.ipynb`.

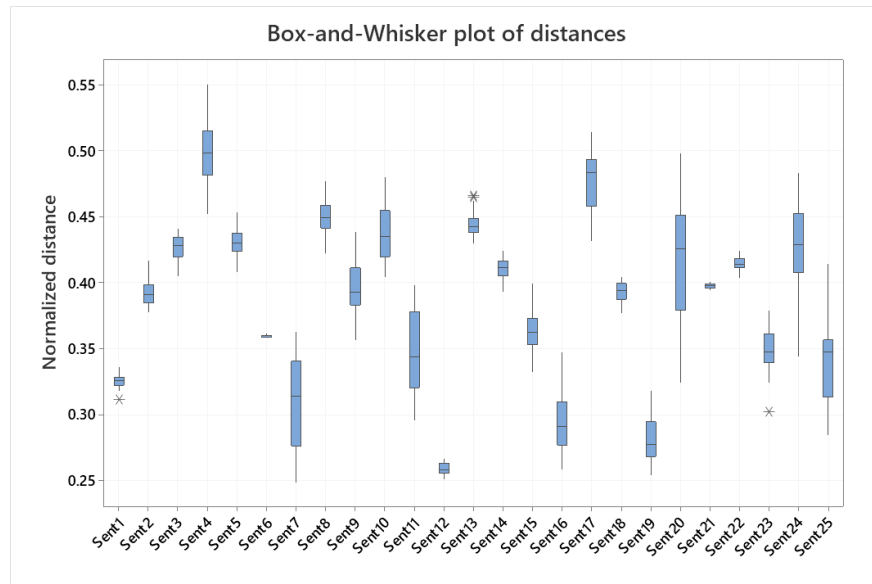


Figure 1: An example of the box-and-whisker plot of normalize distances by sentences. Please note that the plot is generated using random numbers, which will be different than your result.