Word Norms Processing Guide

This document details the steps we took to process the data from start to finish by explaining each folder and the files within each folder. We created this guide to explain steps that were not created by a program, and thus, would be difficult to track.

All words.zip contains folders 0-2 for easier downloading.

**Folder: 0 word lists**

This folder contains the raw subject data in the form downloaded from Mechanical Turk or SONA. Additional files were created through email. The perl files in this folder read those .csv or email files and created the individual files of each cue word found in the next folder.

**Folder: 0.5 raw words**

In this folder, we have each target word as a text file. Each line of the file is a participant in the study. These files are “raw” because they have not been spell checked or checked for other issues (see below).

Please note if you see a file or label “word\_con”, the cue word is *con* but a file named con.txt was not readable/useable on Windows because it is a reserve device word. Additionally all files with “ R.txt” – the R was a code we used in the beginning to separate out files, that coding was discarded later.

**Folder: 1 spell checked**

In this folder, we have spell checked all the files (albeit small errors were found later, see below).

**Folder: 1.5 words no wiki**

**\_fixed** includes all words without Wikipedia entries as a subject’s answer. This type of answer was most common on Mechanical Turk participants.

**Word counter**: this folder includes the structure and .php file we used to create word frequencies found in later files.

**Folder: 2 raw frequency files**

**Ours:** This folder contains each cue word as a file with frequencies listed as *feature frequency*. Each file usually starts with a blank space and a number (discarded later) and does include examples where the cue word and the feature are the same (also discarded later).

Countlines.pl was used to create final sample size for each cue.

Getfeatures\_withwords.pl was used to combine all the files in the ours folder into one giant text document words\_features.txt.

**Folder: 3 parsed**

Given the words\_features.txt file, we created an excel document in here called all words.

* All lines with blank spaces were discarded (see above).
* N values from the countlines were added to a separate tab.
* The translation tab was created.
  + This tab includes the information about how we translated full features to their root words. For example, *cats* would be translated to *cat* + *s* for later processing. The affix list word document includes the most common affixes and how we coded them (see the next section).
  + The first column is the original word, the second column was the translated root (or other spelling error or translation like numbers).
* In the words by features tab, we created the sample size column by adding the n values to each cue.
  + Normalized is the frequency / sample size with the caveat that this value cannot be over 100 percent (people often wrote words twice in the same definition).
  + Translated is the column that contains the root or translation for the feature (if there is no translation, the word is just copied over).

The all words.txt file is the all words file in text format. From here, we used 3\_cleanupcrap.pl to do the following:

* Take out a bunch of features that were used to define words, but where not part of the definition, such as *use, do, have, about, describe*.
* Calculate the top features for each word:
  + To be considered a top word, each **translated feature** (not the original written feature) had to be listed 16% across subjects (matching original work by McRae et al. 2005) or be in the top five features listed by **translated feature** frequency.
  + This process was completed on the translated features to capture all the versions of words appearing (i.e., *beauty* could have been *beauty beautiful beautifulness beautifully* etc.). Therefore, the frequency of the translated word normalized was the criterion.
* We put the output in the top words 2016 excel file.
* The top to final.csv file includes a list of small changes that were made between the top words file in folder 3 and final words in folder 4. These small changes included combining lines that were misspelled or had a strange character in them or fixing some translation file typos. We include this file for the sake of transparency since these files do not have the same number of lines.

**Folder: 4 analysis**

Final words 2017 is the completed processed file that includes all versions of words.

* Where is a column that denotes which dataset the cue feature combination came from (b = Buchanan, M = McRae, V = Vinson). Please note that the \_markers from the McRae and Vinson datasets were discarded to match to this dataset (i.e., bat could be an animal or baseball, but here we did not differentiate).
* Part of speech for the cue, feature, and translated feature were created as the most common part of speech (using google define if necessary).
* A1, A2, A3 include all the affixes from the feature to translated feature. These were coded based on the system listed in the affixes file – they were also checked for verb tense (i.e. cat*s* to cat would be numbers while walk*s* to walk would be present participle).
* Lastly, FSG and BSG (forward and backward strength) were added from the Nelson et al. 2004 free association norms.
* Final words.txt is the text version of this file without FSG and BSG.

**M and v data:**

These files include the cue feature frequencies from McRae and Vinson norms.

**Perl calculations:**

Pullallmv\_feautres.pl was used to combine these files together into one place. This information was added to the final words file explained above. We translated words and coded affixes in the same style as described above.

Dotcosine.pl was used to create cosine values as described:

* All raw values.txt – this file calculates the dot product cosine on the raw feature listed and that frequency (feature frequency).
* All root values.txt – this file calculates the dot product cosine on the root values listed and that frequency (frequency translated).
* All affix values.txt – this file calculates the dot product cosine on the affixes and their frequencies (feature frequency). If multiple affixes are the same for each word (i.e. two characteristic labels, they were combined – therefore, if the frequency was 2 and characteristic appeared twice, it was counted as 4).
* In each of these files, the letter b, m, or v appears before both the cue and target word to indicate where the word is from. These are averaged out or included with a dot in separate files.
* With over 4000 unique words, there are more than 16 million possible combinations. Only combinations with **at least two features in common** are printed out. This system cuts off zero and very small cosines, but these values can be calculated if desired from the frequency files.

Combine\_files.pl does the following:

* Uses the following files:
  + All raw values, all root values, and all affix values from the previous step.
  + Usfjcnlsa from Maki 2008 feature sampling norms
  + Usf\_norms.txt from Nelson et al. 2004 free association norms
  + Oldcos.txt from Buchanan et al. 2013 semantic feature production cosine values
  + Items\_spp2.txt is the information about items from the Semantic Priming Project Hutchison et al. 2013
* To create the following files:
  + All where cosine: all cosine values with the tags on which database it came from. Here’s a description of that file:
    - Cue target with b., m., v. to determine which dataset the cue and target originated from.
    - Root cosine, raw cosine, affix cosine from this set of calculations
    - Old cosine from Buchanan et al.
    - JCN, LSA, FSG, BSG
  + All averaged cosine: all cosine values in the same format/order as described above with the exception that repeated cue/target combinations were averaged together.
  + Items\_spp.txt: this file combines the Semantic Priming Project data with the JCN, root, raw, and affix cosine values for further analysis.