Stefanie Reed
LING83600
Fall 2020

Lab 1 Write Up

Part 1

The first step was to read in the .tsv file, which I did by using the pandas method "read_csv". Then I needed to get the lemmas for each word. I created a function that used nltk's wordnet synset feature and returned the first synset for each word.

Next, I made a function called "get_synset" to return the synset to feed into the word similarity functions I made. For path similarity, Leacock-Chodorow, and Wu-Palmer measures, I made the arguments for those functions the tokens from the word pair columns. For the remaining similarity measures that require information content, I downloaded that from the Brown data.

Once all the functions for measuring similarity were defined, I initialized empty lists for their scores. Then I made a loop that would go through the data frame and locate the i^{th} word from each column, which I then passed into all my similarity functions and appended those to the scores lists. Once those were all computed, I made those into columns that I added to the main data frame.

Once I populated my data frame with all of the scores, I computed the correlation by indexing only the generated scores from my data frame then running a loop over those six columns.

For coverage, I used numpy to sum over the scores and divided that by the length of that number and got 200.

Issues

When trying to compute the Leacock-Chodorow similarity, I got an error that the parts of speech didn't match for their synsets. I tried to fix this for a couple hours and could not figure it out so I just ended up deleting the two entries that gave me the error. Those were "eat" and "drink", and "stock" and "populate".

I know I did this incorrectly because my coverage was off.
Despite being aware my numbers are totally off, I really enjoyed this part of the assignment because it was extremely helpful for familiarizing myself with indexing in pandas and using pandas more in general.

Part 2 and 3

My biggest problem with this portion of the assignment was that I misunderstood the instructions but confidently thought I knew what to do, then spent days working on the wrong thing.

My first issue was with tokenizing the data. At first, I casefolded, lemmatized, and removed stop words before tokenizing. I realized that I shouldn't remove the stop words because it affects the distance score.
I had to run the tokenization for this portion several times because of mistakes pre-processing. I also could not figure out how to get the news data file into the right format to be accepted by the provided ppmi.py script. I know it was an issue with my tokenization, which I kept trying to fix. I wasn't sure how to get around the fact that one mistake can cost an hour. And I kept making many mistakes with this part.

I then tried to create word pairs of all of the tokens in the news crawl data. This was incredibly time consuming and regrettably so when I realized the word pairs were supposed to come from the ws353.tsv.

When I ran the ppmi script, the results indicated that there were zero word pairs found. I understand that the script wants a sentence with a space between all the elements. I used the command "less -s filename" on my tokenized .txt file and it looked like that was what I had, so I'm not sure what I did wrong there.

Since I could not get this to run properly, I couldn't figure out how to get the PPMI score. I also could not run the word2vec script either due to the same issues.

If I had been able to, the next steps I would have taken would be to match the order of the word pairs by using if statements. Then I would have used a function similar to Part 1 to compute the correlation.