NLP Prog2 Write Up

Jingwen Zhong

For the first approach I used pre-trained word-vectors build in model "glove-wiki-gigamword-100" from genism-data.

For the second approach, first I use lex-ont.json file to get the if_parents of the word. Note that one word can have more than one if_parents, so we have more than 1 path. Then I use a recursive function I build to find the previous words that the target word stems from. I append all the previous words of the target word in a list, so I can compare one list with one another. I conduct the lists into set, then find the intersection of 2 lists, and that will be one of the LCS. Then I compare the depth of all the LCS and choose the largest depth. In the meantime, I choose the parents sets that contain this LCS, and calculate the depth sets, then I can compute the similarity with the math function.

In my third approach, I limit the words on the trips-brown_NV_overlap.txt list. And then use the find the number(N) of words in trips-brown_NV_overlap.txt and build a $N \times N$ matrix.

My fourth approach is the combination of the first approach and the third one with the Brown corpus, and I think the similarities between 2 words are too high. Also using the Brown corpus, the results between the third approach and the forth one are totally different

I think second approach can get the most reliable result, because it is more reasonable, while the third approach purely depends on the number of samples and I think we need more data.