Zalkar Ziiaidin uulu
Justin Vasseli
CS 4122: Computational Linguistics

Project#5

Time Complexity

It was my first time to deal with a very high time complexity in Python. My very first version, which used Python lists instead of NumPy arrays, showed time complexity of ~600s (10 mins) for one cycle over all test cases. Which would result in ~60 mins to run all six different cases.

NumPy

After integrating NumPy I was able to cut the running time considerably. Particularly, I was to cut it ~6 times, so now my average run on CatLab machine is ~100s for one cycle.

Scipy

Even though Scipy and NumPy have common roots, I found Scipy to be ~3 times slower than NumPy built-in functions. Particularly, Scipy showed slower performance in calculating an Euclidean distance.

Files with higher accuracy

gram6-nationality-adjective.txt
capital-common-countries.txt
Family.txt

All three have relatively higher accuracy of >20%, while other files might have accuracy rates as low as 1%. I think that their higher rates come from the reason that used vector model has most of the words needed.

Normalization

I didn’t get a chance to see the difference, since our particular case. However, I was also able to implement fast normalization using NumPy

Similarity Metrics

All three of them have around ~14% accuracy rate. Interesting fact is that Cosine distance is highly similar to Euclidean distance, which might be reasonable or just my bug.

Overall

It was quite an interesting challenge. I was able to get the idea of working with relatively bigger data and to learn to cut the run time in Python.