# LING571 – Hw7
# Distributional Semantics

## 1. Overview

I worked really hard this week; clocking in more than 26 hours for this course alone.

Despite this, I am not feeling very confident that I've done everything correctly. On the other hand, I've learned a lot, which is, I suppose, the end-goal.

## 1. Notables

When I ran my code, I was not able to complete the generation of hw7_sim_10_PMI_output.txt. A memory error was reported. This error did not manifest on my local machine; and I was later able to run via Condor without error.

## 2. Methodology

I relied a lot on Ryan's implementations of both the PMI calculation as well as his CollocationMatrix matrix; however, as discussed below, his implementations were tweaked where necessary.

## 3. Challenges

It took me a long time to figure out why I needed to calculate cosine similarity. I tried implementing the code without it and discovered that I was getting zero similarity scores when trying to use the PMI by itself. I realize that this was because, without the cosine similarity, the algorithm was only taking into account the exact two words we were looking for; occurring within the specified window-size. In other words, 'car' and 'automobile' would have to have been practically beside one another for window=2.

Eventually, I realized that I needed to use the cosine similarity in order to derive a more generalized similarity. I realized that the cosine similarity is taking into account the common collocating words (not just the two words we were looking for), which resulted in infinitely better results.

## 4. Results

To be honest, I got very strange results.

Using CBOW, I got a non-trivial negative correlation ranging from -0.296 to -0.374. The implication of this is baffling. A positive correlation (near 1 in the interval [-1, 1]) would suggest that the algorithm agreed with the human annotators assessment of which words were similar. A correlation of zero would suggest that no such pattern could be found and that the algorithm was unable to find similar words. A negative correlation, on the other hand, seems to imply that when human annotators found the similarity between the words in our test cases (e.g. car, automobile) to be strong, the algorithm calculated a weak correlation. This pattern was found consistently across the range of the window sizes that I tested; specifically, $window_i \in \{ 2, 10, 50, 100 \}$.

```
+-------------+--------+--------+--------+--------+
| CBOW        |   2    |   10   |   50   |  100   |
+-------------+--------+--------+--------+--------+
| Correlation | -0.296 | -0.359 | -0.374 | -0.356 |
+-------------+--------+--------+--------+--------+
```

I was worried that perhaps I had badly messed up but after conferring with some colleagues, we all had similar results. In fact, one other student's results agreed with mine to 12 decimal places, which was reassuring. Either I wasn't wrong… or, at the very least, I was wrong but in good company.

Perhaps, despite the size of the dataset, there is still not enough data to train the NN. Or perhaps, because we were not seeding randomly (in order to compare results) for our analysis, the NN found a local maximum that was not the global maximum for this problem.

The distributional similarity section went a bit better. Here, I was able to generate the following results:

```
+-----------+--------+-------------------+
|   Type    | Window |   Correlation     |
+-----------+--------+-------------------+
| Frequency |      2 |  -0.0192529856399 |
| PMI       |      2 |   0.606771880235  |
| PMI       |     10 |   0.425933082702  |
| Word2Vec  |      2 |  -0.296267817236  |
+-----------+--------+-------------------+
```

The FREQ variety indicated no correlated with a  window size of 2; however, the PMI variety worked surprisingly well. It demonstrated a relatively strong correlations of 0.61 when using a window size of 2 and a correlation of 0.426 when using a window size of 10. This is by far the strongest correlation measured during my experiments.

## 5. Insights

Certainly, the greater the window, the more slowly the algorithm runs; particularly with PMI. A window of 100 will naturally take almost 100x the time taken to build up the CollocationMatrix; as expected. But the impact on calculating similarity seems non-linear. A window of 100 means that the matrix is 100 times less sparse; which multiplies the work required to calculate the cosine similarities.

## 6. Learning Outcomes

Ryan's code was a great way of accelerating my progress; however, I soon realized that it needed to be tweaked in a number of ways to get better performance. In particular, I recognized an opportunity for caching the row, columns and table sums; which I think drastically improved my implementation's performance.

## 7. Completeness

I completed the assignment.