# NLP HW1 Report

## Environment

Running environment: Colab

Python version: Colab

## Discussions

### **Which embedding model do you use? What are the pre-processing steps? What are the hyperparameter settings?**

* I used **word2vec model**
* **Pre-processing:** after sampling 20% of wiki texts, I first filter out the stop words (self-defined). Then, I lowercase all words and split them such that they form a list of lists. Finally, I compute the word occurrence frequencies and remove words whose occurrence is less than or equal to three.
* **Hyperparameter settings:**
  + min\_count=3 # Ignore words with frequency below this
  + window=10 # Maximum distance between current and predicted word
  + vector\_size=300 # word vector dimension
  + workers=multiprocessing.cpu\_count()-1 # Use all available CPU cores to train

### **What is the performance for different categories or sub-categories?**

* Performance varies drastically across different categories. Categories like "family" and "countries" performed quite well, it seems that the word2vec model is suitable for these types of words.
* However, some categories related to “grammar” showed lower accuracy. This suggests that capturing syntactic relationships is more challenging for word embeddings trained on general text data like Wikipedia, which may focus more on semantic similarities.

### 

### **What do you believe is the primary factor causing the accuracy differences for your approach?**

* I think that at the end of the day, the quality and quantity of training data affect accuracy the most.
* Categories with more common, frequent words (like family-related words or country names) tend to perform better because the model has more examples of these relationships during training. And vice versa for rarer words who might not have enough occurrences to generate strong embeddings.
* Additionally, context window size (set to 10 in the current hyperparameters) is crucial for capturing types of relationships. A larger window may capture more semantic relationships, while a smaller window may be better for syntactic relationships.
* Other factors that could have been considered: Lemmatization, stemming, better tokenization, etc.

### **What’s your discovery from your t-SNE visualization plots?**

* From the t-SNE visualization plot, I noticed that similar meaning words within the same category (e.g., family members like "father," "mother," "brother," and "sister"), tend to cluster closely together. This shows that the word2vec model is successful in capturing semantic similarities.
* However, in categories where the relationships are syntactic (e.g., verb tense transformations like "run" and "running"), the clusters were not as well-defined. This shows that the model struggles to capture syntactic differences.
* Additionally, when I was running multiple tests, sometimes the sampled text (20% Wikipedia articles) does not include words in questions-words.csv. Hence, the t-SNE graph sometimes has missing words (e.g. for “family” category)
* This limitation of data, I think, also caused the poor performance of the model

### **What’s the difference in word representations if you increase the amount of training data?**

* Increasing the amount of training data should improve the quality of word representations, especially for rare words. As rare word occurrence increases, the model will be able to generate more accurate embeddings.
* This means better clustering of semantically similar words in the t-SNE plots, especially for words that were previously outliers.
* For frequent words, on the other hand, no big improvement with more training data, as their embeddings would have already stabilized after a certain number of occurrences.

## References

Links to websites I referenced when coding

<https://radimrehurek.com/gensim/models/word2vec.html>

<https://hackmd.io/@gensimDART/SydKygQa_>

<https://www.kaggle.com/code/pierremegret/gensim-word2vec-tutorial>