Computational Semantics 2019-2020
2nd graded assignment

This week you will experiment with how distributional representations can account for categorisation; for instance, whether they capture the fact that strawberries and plums are fruits and buses and planes are vehicles. We will use a pre-defined list of words and categories. The solutions to Week 4's exercises can guide your coding.

Work collaboratively. Submit, per group:

- one pdf file with your answers (including all necessary tables and figures). Include a section "Contributions" that specifies who in the team has contributed to what.
- one zip file with your code and its output.

Exercise 1. You have one file with words and another with categories (words_battig.txt, categories_battig.txt). Your first task will be to use WordNet to assign the words to their categories. For some words, WordNet doesn't contain enough information to place them in one of the pre-specified categories; these words we will ignore in the remainder of the exercise. Create a script that takes the two data files as input and produces another file as output, wordnet_categorisation.txt, that lists each word with its corresponding category according to WordNet. Format: one line per word, words and categories separated by a tab ("\t" in Python). For instance (also see file sample_categorisation_file.txt):

```
dog       mammal
elephant  mammal
cat       mammal
```

Note for inexperienced Pythoners: If you struggle too much with the code, you can also create this file semi-automatically: Retrieve the hypernyms of the words with code, and create and fill in the text file with the categorisation manually. (But do try to do as much as possible automatically, of course.)

Exercise 2. Now we will see how the nouns in our mini-taxonomy (wordnet_categorisation.txt) get grouped in distributional space. There are three options for this exercise. Groups with DTIC students should go for Option C. Other groups are welcome to do Option C, should attempt Option B, and are allowed to revert to Option A if they struggle too much.

A) Create a t-sne graph with two dimensions by completing script exercise2A_visualisation.py. This graph should depict the word vectors for both the target words ("apple", "dog") and their hypernyms ("fruit", "animal"). Do 10 different runs to obtain different graphs (save them with different names) so you can see general tendencies.

B) Create a t-sne grah with two dimensions by completing script exercise2B_visualisation.py. This graph should depict the word vectors for the target words ("apple", "dog"), their hypernyms ("fruit", "animal"), and the prototype vectors that depict categories as averages of the target words: For instance, the prototype category FRUIT will be the average of the word vectors for "apple", "orange", "grape", "peach", and "strawberry". Use different colors for target words, hypernyms, and prototype categories. Do 10 different runs to obtain different graphs (save them with different names) so you can see general tendencies.

   - BONUS: See if you can modify the function plot_with_tsne() such that it only plots labels for hypernyms and prototype categories; that is, it should plot dots for all elements, plus words in lowercase for hypernyms ("fruit", "animal") and words in uppercase for prototype categories ("FRUIT", "ANIMAL"). Or try other variations of the plots.

- BONUS: Compare the cosine similarities between the hypernyms and, on the one hand, their prototype category; on the other, their hyponyms individually. What do you conclude about averaging individual vectors as a way to represent a class of words?

C) Complete script exercise2C_clustering.py, in which you will do the same as in Option B plus a clustering exercise. Clustering is an unsupervised Machine Learning techniques that creates groups of datapoints based on their similarity. It is very useful for exploratory purposes. You will use the k-means algorithm with k=11, evaluate the results via purity, and qualitatively examine the results. For background on all this, see `https://nlp.stanford.edu/IR-book/html/htmledition/flat-clustering-1.html`

- BONUS: As part of the analysis, create plots that compare clusters with categories. Suggestion: barplot with clusters as bars, categories as colors.

Exercise 3. Comment on the results that you obtain: what did you expect, what results are strange, and how can you explain them? For Option C, include quantitative results and qualitative analysis of the clustering results.