Text Analytics: Practical 5
(for Lecture 5: Similarity)

1) Pick a category of things (e.g., dogs, cats, U2-records*) and describe 5 instances of this category (e.g., Labrador, Poodle, Alsatian, Peke, Borzoi) using 5 feature-words (e.g., "Labrador" might be "shorthair, golden, barks, tail, cute"). Make sure that you re-use some of the feature-words across different instances so that similarities between instances can be found.

    a. Modify the Jaccard-Index program you were given in the lecture to do Jaccard-Distance (see notes) and then find the Jaccard-Distance between all the pairwise-comparisons of the 5 instances. Show the distance-values found in a matrix with rows and columns for the 5 instances.
    b. Check whether property of the triangle inequality holds for the Jaccard-Distance measure on the basis of the values you have found. Show the results of the comparisons you have made in testing for this property.
    c. Now implement the Dice Coefficient and compare its results to those found for Jaccard Distance; comment on the what seems to be the samedifferent in the results from these two measures.

2) Have a look at the Cosine.py program [nb you may need to install the packages its imports]:

    a. Identify 3 online articles about the same event (e.g., Trumpy getting COVID-19*) and 3 online articles about a related, but different, event (e.g., Boris Johnson getting COVID-19*). Select some key paragraphs from the articles an run them through your pre-processing pipeline (use aggressive stop-word removal). Then taking the pre-processed files, find the pairwise cosine-similarity for articles within each group (the Trumpy and the BJ groups) and then between the articles in both groups (i.e., comparing the Trumpy ones and the BJ ones).
    b. Plot these similarity scores in a graph and check if the scores cluster to reflect the two groups. Comment on the clustering/lack-of-clustering (whichever happens) and trace the sources of these effects to the feature-words from each group.
    c. Find the sklearn python package that allows you to compute Cosine Similarity and, also, Manhattan Distance. Use it to process the data you have already. Are the scores you find from the sklearn method the same as the ones from the Cosine.py program? Do the Manhattan Distance scores for the same item-comparisons change markedly from those found for Cosine Similarity; discuss what you find.

* Do not use the ones I have mentioned here !