Similarity

Practical 5: Similarity

Jaccard
Q1

Prac5 Q1(a) Jaccard

- Make up your own set of word features describing 6 different entities; with some obvious overlaps and differences
- Modify the Jaccard-Index python program to do Jaccard-Distance and then compute all pairwise distances between the entities
- Based on results, show empirically, that the property of triangle inequality holds for measure

Prac5 Q1(b) Jaccard

- Now implement the difference function for the Dice Coefficient and show that the property of triangle inequality may not hold for this measure

VSM & Cosine
Q2

Prac5 Q2a Compute Cosine

- Have a look at the Cosine.py program; nb you may need to install the packages its imports
- Find 3 short documents about which you might want to know their similarity
- Produce 5 variants on one of the documents and see how the cosine similarity changes

Prac5 Q2b Compute Cosine

- Plot the similarity differences on a graph showing their cosine similarity score
- Verify that your intuitions about what makes the differing docs less similar does indeed lead to scores that are less similar

Prac5 Q2c Compute Cosine

- Find a python package that computes cosine similarity and euclidean distance
- Use it process the data you have already
- Do the answers correspond and what do the Euclidean Distance scores look like relative to the Cosine ones

Levenshtein
Q3

Prac5 Q3 Levenshtein

- Create or find 5 “normal” tweets from Twitter
- Now take one of these tweets and systematically generate 20 SPAM tweets from it; using the typical techniques of spammers
- Now, perform comparisons between these 20 SPAM tweets each of the 5 Normal Tweets
- Plot their edit-distance scores in a graph and colour code to show how the SPAM v Normal ones
- Are the SPAM tweets obvious, if not why?