Locality-sensitive Hashing

The task aims at investigating MinHash and Locality-sensitive Hashing (LSH) framework on real-world data sets. In the practice, you write a program to compute *all pairs similarity* on the "bag of words" data set from the UCI Repository using the *Jaccard* similarity. This problem is the core component for detecting plagiarism and finding similar documents in information retrieval.

The bag of words data set contains 5 text datasets which share the same pre-processing procedure. That is, after tokenization and removal of stopwords, the vocabulary of unique words was truncated by only keeping important words that occurred more than 10 times for large data sets. For small data sets, there was not truncation.

It has the format: *docID wordID count*, where *docID* is the document ID, *wordID* is the word ID in the vocabulary, and *count* is the word frequency. Since the Jaccard similarity does not take into account the word frequency, we simply ignore this information. This means that we consider *count* = 1 for each pair (*docID*, *wordID*). We consider a document as a set and each word as a set element, and make use the Jaccard similarity. Note: the dataset is very sparse.

The tasks are specified as follows.

1. **Execute bruteforce computation:** Compute all pairs similarity with Jaccard and save the result into file (since you have to use the bruteforce result for the next tasks). You need to report:
   - **(a) The running of the bruteforce algorithm.**
   - **(b) The average Jaccard similarity of all pairs except identical pairs i.e. J(di, dj) where i ≠ j.**
2. **Compute the MinHash signatures for all documents:** Compute the MinHash signatures (number of hash functions $d = 10$) for all documents. You need to report:
   - **(a) The running time of this step.**
3. **Measure the accuracy of MinHash estimators:** Compute all pairs similarity estimators based on MinHash. Repeat the procedure 2) and 3) with the number of hash functions d ranging from $\{10, 20, \ldots, 100\}$. You need to report:
   - **(a) The running time of estimating all pairs similarity based on Min-Hash with different values of d.**
   - **(b) The figure of MAEs with different values of d on x-axis and MAE values on y-axis.**
4. **Exploit LSH:** Implement LSH framework to solve the subproblem: "Finding

all pairs similar documents with Jaccard ≥ 0.6". In particular, using $d = 100$ hash functions, you need to explain:

(a) **How to tune the parameter b (number of bands) and r (number of rows in one band) so that we achieve the false negatives of 60%-similar pairs at most 10%.**

(b) The space usage affected by these parameters.

Given your chosen setting, **from your experiment**, you need to report

(a) The false candidate ratio.

$$\frac{\text{the number of candidate pairs with exact Jaccard} < 0.6}{\text{number of candidate pairs}}$$

(b) The probability that a dissimilar pair with Jaccard ≤ 0.3 is a candidate pair.

$$\frac{\text{the number of candidate pairs with exact Jaccard} \leq 0.3}{\text{number of pairs with exact Jaccard} \leq 0.3}$$