HW1: Finding Similar Items: Textually Similar Documents

Authors: Andrei Capastru, Alex Porciani

The task is to implement shingling, minhashing and LSH algorithms in order to compare text-based documents.

1.1 How to run the code

0. Install python3 and install Jupyter Notebook
1. Run:

```
cd ~
git clone https://github.com/andreicap/data-mining.git
cd data-mining/HW1
jupyter notebook
```

2. Open the browser on `http://localhost:8888` and run the `shingling.ipynb` file

1.2 Shingling phase

Here we perform the shinglign phase - Divide the documents in word-touples of size in the variable `shingle_size`

1. Assign shingling size
2. Split the documents in list of words
3. Group words in tuples of `shingle_size`
4. Transform the list of tuples in sets of tuples to avoid duplication and put in in `shingles` column in the dataframe
5. Create a new column with a hashed version of shingles:
    - join the tuples into a string
    - hash the byte-encoded string using `s_hash` function

1.3 MinHash phase

In this phase we calculate the minhashes for each document.

1. Set the number of hashes per document in `numhashes`
2. Generate the two lists of coefficients for the minhash functions
3. Create the signature lists using generated coefficients and save it to the dataframe

1.4 LSH phase

We set the elements per bnad and add all the possible candidate pairs to a set. To do this we iterate over all pairs exactly once, but we itare over the lists of signatures as well and also bandwise. This leads to an inneficient code, but it is ok for small number of documents (<500), like this test.