ID2222
Lab Report 1
Philip Claesson & Miguel Maricalva

Finding Similar Items: Textually Similar Documents

How to run

The assignment is handed in as a Jupyter Notebook. In the data directory, some sample data is provided.

Simply run by sequentially executing each block of the code.

Code structure

The code starts with the reading of data, followed by the main data structures: Document (for keeping and manipulating the read documents), Hasher (for creating, maintaining and using a large set of hash function) and LSH (for executing Locality-Sensitive hashing). All of the actual manipulation is contained within methods of these three classes, which are then executed sequentially in the code.

The execution is divided into three parts: 1. Finding the jaccard similarity, 2. Finding the minhash similarity and 3. Using LSH to find similar pairs.

A class OrderedSet is imported and used from a separate file.

Hyperparameters

Hyperparameters can be set for part 1, 2 and 3 under each parts respective section *Hyperparameters*. The hyperparameters are explained in the file, but also here:

n_buckets
The number of hash buckets (or the number of different possible output values for the hash function). A high n_buckets will yield low similarity measures, because more possible hash values means fewer shingles will be hashed to the same buckets.

shingle_length
The number of characters in each shingle. Large value will yield low similarity measures, as there will be more combinations of shingles.

(2.)
n_hashes
The number of hash functions used to create the min hashing signature.

(3.)

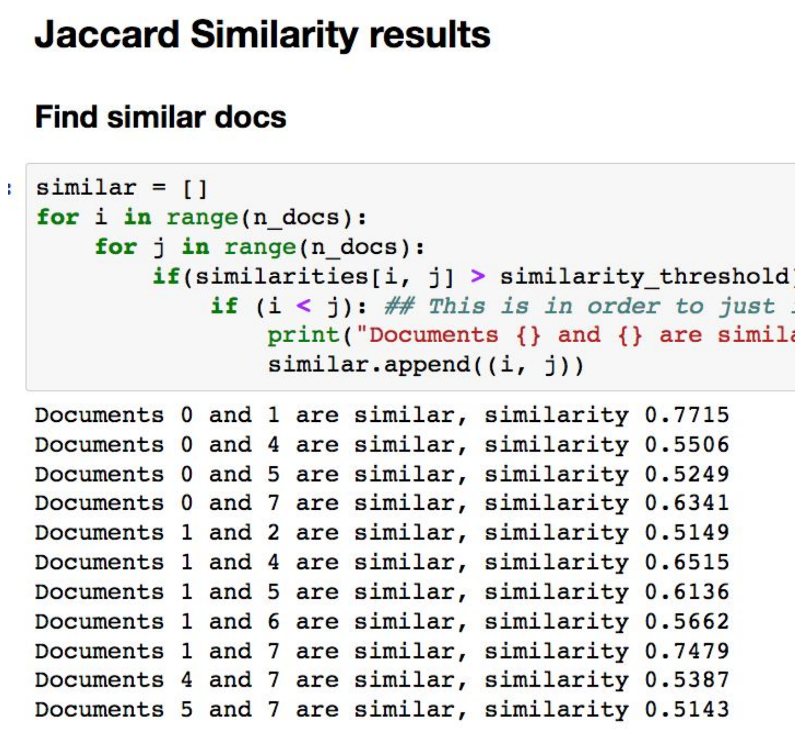
n_bands and n_rows
The number of bands, and number of rows in each band. b * r should be equal to the number of integers in the signature of a document, n.

t
The threshold of similarity to filter out not false positives from candidate pairs.

n_lsh_buckets
The number of buckets in the hash function which hashes the bands of the signatures. Should be as large as possible, but increases the runtime!

Results

All results are displayed in the notebook. In this report, the top similar documents between the different methods are displayed. As can be seen, some documents are reoccurring in all methods, for example (0, 1), (4, 7).



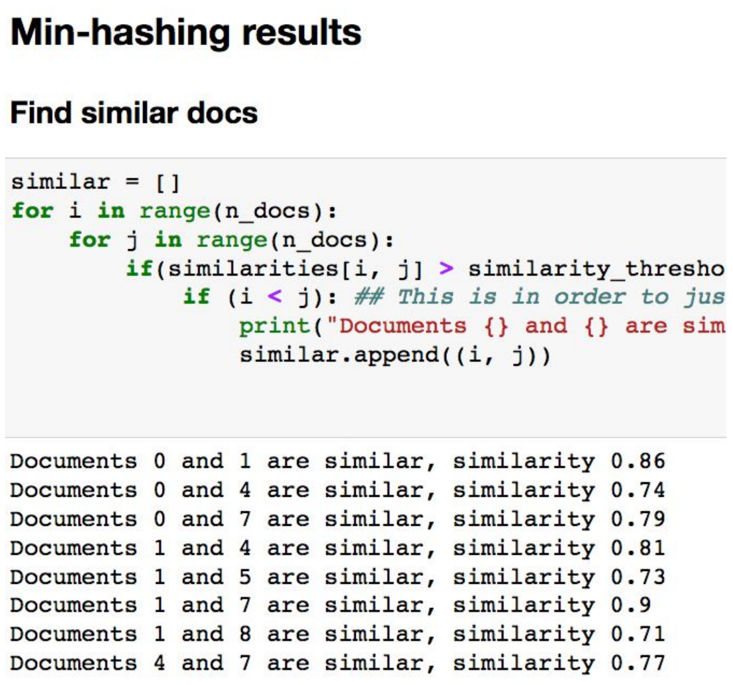
Jaccard Similarity results

Find similar docs

```
similar = []
for i in range(n_docs):
    for j in range(n_docs):
        if(similarities[i, j] > similarity_threshold
            if (i < j): ## This is in order to just
                print("Documents {} and {} are simil
                similar.append((i, j))
```

```
Documents 0 and 1 are similar, similarity 0.7715
Documents 0 and 4 are similar, similarity 0.5506
Documents 0 and 5 are similar, similarity 0.5249
Documents 0 and 7 are similar, similarity 0.6341
Documents 1 and 2 are similar, similarity 0.5149
Documents 1 and 4 are similar, similarity 0.6515
Documents 1 and 5 are similar, similarity 0.6136
Documents 1 and 6 are similar, similarity 0.5662
Documents 1 and 7 are similar, similarity 0.7479
Documents 4 and 7 are similar, similarity 0.5387
Documents 5 and 7 are similar, similarity 0.5143
```

ID2222
Lab Report 1
Philip Claesson & Miguel Maricalva

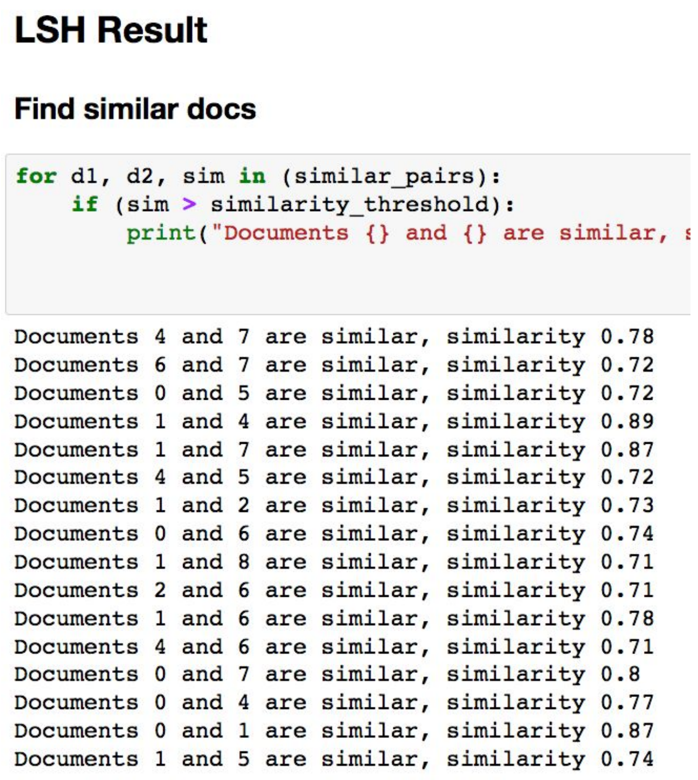


Min-hashing results

Find similar docs

```
similar = []
for i in range(n_docs):
    for j in range(n_docs):
        if(similarities[i, j] > similarity_thresho
            if (i < j): ## This is in order to jus
                print("Documents {} and {} are sim
                similar.append((i, j))
```

```
Documents 0 and 1 are similar, similarity 0.86
Documents 0 and 4 are similar, similarity 0.74
Documents 0 and 7 are similar, similarity 0.79
Documents 1 and 4 are similar, similarity 0.81
Documents 1 and 5 are similar, similarity 0.73
Documents 1 and 7 are similar, similarity 0.9
Documents 1 and 8 are similar, similarity 0.71
Documents 4 and 7 are similar, similarity 0.77
```



LSH Result

Find similar docs

```
for d1, d2, sim in (similar_pairs):
    if (sim > similarity_threshold):
        print("Documents {} and {} are similar, s
```

```
Documents 4 and 7 are similar, similarity 0.78
Documents 6 and 7 are similar, similarity 0.72
Documents 0 and 5 are similar, similarity 0.72
Documents 1 and 4 are similar, similarity 0.89
Documents 1 and 7 are similar, similarity 0.87
Documents 4 and 5 are similar, similarity 0.72
Documents 1 and 2 are similar, similarity 0.73
Documents 0 and 6 are similar, similarity 0.74
Documents 1 and 8 are similar, similarity 0.71
Documents 2 and 6 are similar, similarity 0.71
Documents 1 and 6 are similar, similarity 0.78
Documents 4 and 6 are similar, similarity 0.71
Documents 0 and 7 are similar, similarity 0.8
Documents 0 and 4 are similar, similarity 0.77
Documents 0 and 1 are similar, similarity 0.87
Documents 1 and 5 are similar, similarity 0.74
```