Lab1 - Locality-Sensitive Hashing
Data Mining - KTH

Andrea Scotti, Daniele Montesi

12 November 2019

1 Introduction

The goal of the assignment is to implement the stages of finding textually similar documents based on Jaccard similarity using the shingling, minhashing, and locality-sensitive hashing (LSH) techniques and corresponding algorithms. The implementation is done using Spark in Python (pyspark). To test and evaluate your implementation, we have written a program that uses your implementation to find similar documents in a corpus of 5-10 or more documents such as web pages or emails.

2 Code Explanation

The code is organized in a python notebook file. For every part of the lab, we created one or more functions. You can find each of them in a different cell of the python notebook. The the lab tasks are here explained.

1. Shingling: created 2 functions:
   - hashShingling: Given a shingle, returns its hashed value
   - get_shinglings: Given a document (as String), get list of hashed shinglings
2. computeJaccard: Computes the Jaccard similarity of two sets of integers – two sets of hashed shingles.
3. min_hashing: computes the list of signatures given a list of shingles
4. print_min_hashing_similarities: It computes the similarity of 2 documents given their representation in minhash signatures
5. min_hashing_test: computes the approximate Jaccard similarities with min hashing between all the documents

6. lsh_test: Comprises all the steps in order to compute the approximate Jaccard similarities with min hashing between all the pairs of documents that are likely to have a similarity higher than the chosen threshold.

Note: in order to work, the LSH function needs to define the number of signatures to use, the number of bands and then computes automatically the threshold to use. Since choosing the threshold is a requirement, the system lets the user to choose the signature (or set at 100 by default), then it automatically computes all the possible bends and let the user to choose the possible thresholds at run-time.

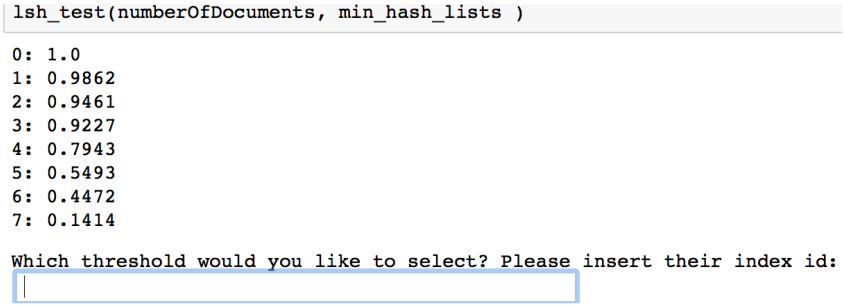

Figure 1: Input text during the threshold choice

3 How to Run the Code

To run the code follow the steps above:

Requirements: Jupyter notebook, Python 3.6, PySpark

1. open the terminal and run

   ```
   > jupyter notebook
   ```

2. open the file "LocalitySentitiveHashing.ipynb" with the notebook
3. run all the cells

You should obtain the following output:

```
lsh_test(numberOfDocuments, min_hash_lists )
```

```
0: 1.0
1: 0.9862
2: 0.9461
3: 0.9227
4: 0.7943
5: 0.5493
6: 0.4472
7: 0.1414
Which threshold would you like to select? Please insert their index id:6
Threshold is set to 0.4472
Candidates are: [(8, 9)]
Similarity between 8 and 9 is 0.56
```

Figure 2: Similar documents found by our algorithm (choosing threshold 6)