Birla Institute of Technology and Science-Pilani, Hyderabad Campus

# 

# Information Retrieval (CS F469)

# Design Document

# Assignment 2

Locality Sensitive Hashing (LSH)

S.Devendra Dheeraj Gupta 2017B5A70670H

K.Srinivas 2017B3A70746H

Abhirath singh 2018A7PS0521H

Ritvik C 2018A7PS0180H

Under the guidance of

**Dr. Aruna Malapati**

## **Abstract**:

In this assignment we will use LSH (Locality Sensitive Hashing) technique to build a plagiarism detector tool. The idea of LSH is to hash the similar items/objects into the same bucket. So by this method the higher similar items will be in the same bucket at the end. The procedure can be divided into 3 steps.The steps are :-

1) Shingling

2) Minhashing

3) LSH (Locality Sensitive Hashing)

What we need to know is a shingle , it is nothing but a substring i.e any substring of length of k can be called a k-shingle. For forming the k-shingings the text in all documents is used. In research areas the size of shingle be around 9. We will build a incidence matrix with these shingles and we need minhashing to get the signature matrix. Minhashing willm compress the document vectors to give signatures that have a lesser number of rows.Then comes the LSH part here the signature matrix is divided into bands which each have a  certain number of rows . Each band is hashed and sent into various buckets then at the end which documents end up in the same bucket are more similar than the other. Now we have documents inside the buckets so the probability  of similarity that documents in the same bucket is high.Now we check the documents in the same bucket using different metrics to calculate their similarity.

## **Architecture:**

**Language Used** : Python

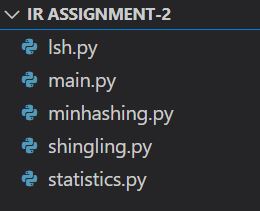
**Dataset Used** : <https://ir.shef.ac.uk/cloughie/resources/plagiarism_corpus.html>

**Used Dataset Detail** : This data is collected from a short answer quiz on Cimpuiter Science There were 5 questions an 20 candidateds out of which 19 copied. This dataset contains 100 documents out of which 95 were copied from the Wikipedia and the other 5 are original answers.

**Libraries used** : Pandas, Numpy, Pickle.

**Data Structures Used** : pandas.DataFrame ,python list, python dictionary, python set

**Modules**: The assignment has been divided into the following modules.Folder structure showed in image below.



1. Shingling
2. Minhashing
3. LSH
4. Statistics
5. Main

## **Procedure Description:**

**Pre-processing:**

         The text documents are given to the shingling model and then perform preprocessing like case normalization, removal of \n, \r and other non important characters. Then from each document shingles are created and added to the incidence matrix. The incidence matrix is a pandas.DataFrame with a shape of number of shingles \* number of documents. Its values are filled with 1 or 0 .Its filled with 1 when a shingle is present in a document and 0 if not.

**Minhashing technique:**

       Some h number of  hash random hash functions are generated. Hashes[k] represents the kth hash function. Every hash function returns the value of  where a and b are >0 and c is the number of shingles. Signature matrix formed using pandas.DataFrame  will have a shape of  number of hash functions \* number of documents .The initial entries of the matrix is infinity.

       For every row of shingle in the incidence matrix, if a document has 1 in that row then the corresponding column in the signature matrix is filled with hash values of the indices only if they are less than the values already present. In this way the number of rows is decreased from the number of shingles to h  which is the number of hash functions. This works because the minhash function for a random permutation of rows gives the same value for two sets equals the Jaccard similarity of those sets. Python set is used for finding Jaccard similarity.

**Locality Sensitive Hashing (LSH):**

We got the signature matrix after the minhashing.In this step we will divide the signature matrix into b bands with r rows in every band. If m is the number of hash functions used in signature matrix making then

We use a hash function  for hashing every band and while hashing a band ,its parts of each document are hashed into a set of buckets. A different set of buckets are used for different bands Each set of buckets is a dictionary having hash values as the keys and list of document ids as values. So documents with the same signature  in the band will land into the same buckets. We consider the similarity between documents using many metrics.

## **Results:**

Incidence matrix: 81 sec

k : 4

size: 11640 x 100

Signature matrix : 1871 sec

Number of hash functions : 100

size: 100 x 100

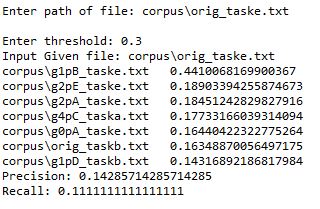
Signature matrix: 936 sec

Number of hash functions: 50

size: 50 x 100

## **Outputs**

**1.**Printing all simiar docs along with metrics to given doc(without threshold)



**2**.Printing only simiar docs above threshold level along with metrics .

