Homework 1

Finding Textually Similar Documents

How to run:

Run main.ipynb file. The data files are in a separate folder called 'Data'

Things implemented:

Functions:

-hash_shingle:

- Open files from a certain path (file_path).
- Convert the files to shingle sets with a length (k) of 5.
- Hash the shingles by using "hash" command of Python.
- Return the set of hashed shingles by using the "add" command to add value of hashed shingle to the sets of shingles.

In the file, each sentence includes a number and a date at the beginning, and a web address at the end. But we are not interested in the numbers, dates and web addresses. So we split the numbers, dates and web addresses by using "line.split" command. For example, in the sentence "586266687948881921|Thu Apr 09 20:37:25 +0000 2015|Drugs need careful monitoring for expiry dates, pharmacists say http://www.cbc.ca/news/health/drugs-need-careful-monitoring-for-expiry-dates-pharmacists-say-1.3026749?cmp=rss", only "Drugs need careful monitoring for expiry dates, pharmacists say" will be output to be hashed.

-jacard_similarity:

- Calculate the intersection and union of two sets(s1 and s2).
- Divide the intersection by union and return the Jaccard similarity of the two sets.

-hash_function:

- The hash function can be expressed as (a*(value of shingle)+b) mod c. The coefficients a and b will be random number less than the maximum value of x (2^32-1=4294967295). c is a prime number(4294967311) slightly bigger than the maximum value of x.

get_signature:

- For i in range of the number of hash function, calculate the returned value of hash function for the variable x (x will be the value of hash shingle from the sets of shingles) by using the lambda function and the map function.
- Add the minimum value of the signature to the end of the signature list by using the append method.

find_sim_all:

- Find similar documents by searching all pairs of documents

find_sim_LSH:

- Find similar documents by only searching candidate pairs.

Results:

We looked at documents of Health News in Twitter Data Set, this dataset contains health news from major health news agencies such as BBC, CNN, and NYT. We also combined two datasets together, the Reuters and CBC datasets were combined into one (marked by + in the table below). This was done to see if the algorithm will be able to find high similarity with a document that contains a mixture of two documents. The list of all documents is given below.

bbchealth
cbcandreuters (+)
cbchealth
cnnhealth
everydayhealth
foxnewshealth
msnhealthnews
nytimeshealth
reuters_health

We firstly divided each data set into 5-shingles. We chose k=5 because the text entries are from twitter tweets which are rather short in length. Then each shingle was hashed to a number.

In the next stage a signature matrix was created. We used 100 hash functions to make a signature for each document.

Next, we made the LSH function which used 20 bands and 5 rows and it found all the candidate pairs. We used the LSH function to compare efficiency of it to that of simply comparing all pairs of signatures. We considered two documents to be similar if their fraction of signatures that were equal was at least 0.5. We picked 20 bands and 5 rows for LSH because that implied a threshold of $(1/20)^{(1/5)}$=0.54, so we focusing on speed and avoiding false positives.

We found that comparing all the signatures together, two pairs of documents were found similar. These were:

Pair	Document 1	Document 2	Similarity SIgnature
1	`cbcandreuters`	`cbchealth`	0.52
2	`cbcandreuters`	`reuters_health`	0.53

Time taken to find these pairs was: 0.05 seconds

We found that comparing only candidate pairs that were found by LSH only one pair of documents were found similar. This was:

Pair	Document 1	Document 2	Similarity SIgnature
1	`cbcandreuters`	`reuters_health.txt`	0.53

Time taken to find this pair: 0.02 seconds

The results show that although the LSH method ran faster, it missed one pair of documents that were supposed to be similar (false negative). We checked the true Jaccard similarity of the two document pairs to be very similar to their signature similarities as theory predicts.

Pair	Document 1	Document 2	Jaccard
1	`cbcandreuters`	`cbchealth`	0.53
2	`cbcandreuters`	`reuters_health`	0.56

Finally, we ran one more test by changing the number of bands in the LSH method. This time we picked 25 bands and 4 rows, for implied threshold of 0.447. This way we are picking bands and rows so that the threshold is lower, meaning we try to avoid false negatives at the expense of false positives. The result of this is that 5 pairs of documents were picked to be candidate pairs.

Pair	Document 1	Document 2	Similarity SIgnature
1	`cbcandreuters`	`cbchealth`	0.52
2	`cbcandreuters`	`reuters_health`	0.53
3	`cbchealth`	`nytimeshealth`	0.23
4	`everydayhealth`	`foxnewshealth`	0.22
5	`foxnewshealth`	`reuters_health`	0.33

By changing the band and row number to implicate a lower threshold we found 3 pairs of false positive results, however this time we did not miss any pairs of documents that were similar (did not miss false negatives)