Report of Locality Sensitivity Hashing

Tianling Feng

**Task 0:**

The following graph depicts the similarity of two input strings using Jaccard Similiarity and MinHash functions. There is about 0.035 difference.

Text

Description automatically generated with medium confidence

**Task 1:**

The following graph shows the query time, quality of all candidates in terms of Jaccard Similarity and quality of top-10 candidates in terms of Jaccard Similarity. The mean Jaccard Similarity of all URL retrieved is 0.2986. The mean Jaccard Similarity of all top-10 URLs in each candidate set is 0.652058. The query time for this process is 266 seconds. The quality of overall candidates is relatively low. However, by selecting the top-10 candidates, we are able to get a relatively good candidate that’s similar to the sample url.

Text

Description automatically generated

**Task 2:**

The following graph shows the time if we use brute-force way to calculate 200 urls’ Jaccard Similarity with other urls in the list. It will take roughly 757 seconds to compute. It is significantly than the query time in task 1.****

**Task 3:**

In summary, when the K increases, the quality of all candidate urls increases and the query time decreases significantly. When L increases, the query time increases as well, and the mean of Jaccard similarity decreases.

**K is 2, L is 20**

Query time = 191.69583082199097 seconds

Quality of all candidate urls: 0.3572813877669406

**K is 2, L is 50**

Query time = 209.32425022125244 seconds

Quality of all candidate urls: 0.3574632893650256

**K is 2, L is 100**

Query time = 193.26170682907104 seconds

Quality of all candidate urls: 0.3574632893650256

**K is 3, L is 20**

Query time = 17.580888986587524 seconds

Quality of all candidate urls: 0.4696390480316102

**K is 3, L is 50**

Query time = 17.695492029190063 seconds

Quality of all candidate urls: 0.4578114348090657

**K is 3, L is 100**

Query time = 17.590622901916504 seconds

Quality of all candidate urls: 0.45359489361401195

**K is 4, L is 20**

Query time = 2.3601620197296143 seconds

Quality of all candidate urls: 0.6114669902295689

**K is 4, L is 50**

Query time = 2.538541078567505 seconds

Quality of all candidate urls: 0.617855243105583

**K is 4, L is 100**

Query time = 2.2908742427825928 seconds

Quality of all candidate urls:

0.6166179192125812

**K is 5, L is 20**

Query time = 0.12875795364379883 seconds

Quality of all candidate urls: 0.7192508691941653

**K is 5, L is 50**

Query time = 0.13515520095825195 seconds

Quality of all candidate urls: 0.7149573519331667

**K is 5, L is 100**

Query time = 0.13062214851379395 seconds

Quality of all candidate urls: 0.7108147897310647

**K is 6, L is 20**

Query time = 0.06156802177429199 seconds

Quality of all candidate urls: 0.796732462596608

**K is 6, L is 50**

Query time = 0.06453990936279297 seconds

Quality of all candidate urls: 0.7652171829084561

**K is 6, L is 100**

Query time = 0.06892704963684082 seconds

Quality of all candidate urls: 0.7689937785484657

**See Task 4 on next page.**

**Task 4:**

Plot 1:

When K increases, the quality of candidate sets increase as well (less likely to retrieve element with low jaccard similarity).Chart, histogram

Description automatically generated

Plot 2:

When L increases, the quality of candidate sets decrease. However, it is more likely to find the exact match (since we are retrieving from a larger candidate set).

Chart, diagram

Description automatically generated