ID2222 Data Mining

Finding Similar Items: Textually Similar Documents

Homework 1 Report

Group 25

Avneesh Vyas <[avneesh@kth.se](mailto:avneesh@kth.se)>

Algirdas Grumuldis <[alggru@kth.se](mailto:alggru@kth.se)>

12th November 2017

# Solution

In this project, we attempt to find similar documents among 10 documents using Jaccard Similarity. For this, we do the following:

1. Perform shingling of the documents by creating a set of unique shingles of length k (k-shingles) from the content of the document.
2. Then we map each shingle to an integer value using hashCode method of String class. Please note that as number of integer values can be less than possible k-shingles, hashing may result in collisions where two distinct shingles get mapped to the same integer value. But we assume that such collisions will be rare and will not significantly impact the overall Jaccard similarity.
3. Jaccard similarity between two documents (A and B) is computed by taking the ratio of unique shingles present in both documents and total distinct shingles in two documents.

For very large documents, space requirement to store k-shingles could be high and prohibitive. So instead, we employ MinHashing technique as follows:

1. Randomly generate n hash functions of the form ( (ax + b) mod c) by randomly generating n values for coefficients a and b. c is chosen to be a constant prime number.
2. For a given shingle set, minhash signature of size n is generated as follows. Each of n hash functions are applied to each element of given shingle set. And then minhash value which is the minimum hash value among all elements for a given hash function is selected. Thus, n minhash value set is generated for the given shingle set.
3. Finally, similarity between two documents is computed by calculating the ratio of identical minhash values among total n minhash values.

# Instructions to run

1. From the root folder run "sbt".
2. (Optional) In sbt shell run "eclipse" to create an eclipse project that can be imported into eclipse.
3. In sbt shell run "compile".
4. In sbt shell run "run".

Functionalities of various classes from the project are briefly described below

* Main.scala - Main entry point to the project
* Shingle.scala - Created hashed k-shingles of all documents present under src/main/resources/documents directory. k=10 defined in Main.scala
* CompareSet.scala - Computes Jaccard similarity of two given sets
* MinHashing.scala - Computes MinHash signature of a given size (n) for a given set of hashed shingles. n=8 used in Main.scala.
* CompareSignature.scala - Compares to given minHash signature set and returns the similarity measure (ratio of matching minhash values and total minhash values)

# Dataset

a0300005.txt, a0300125.txt, a0300128.txt - NSF Awards metadata

rfc2616.txt - HTTP/1.1 DRAFT STANDARD

rfc7231.txt - HTTP/1.1: Semantics and Content PROPOSED STANDARD

rfc7231\_duplicate.txt – the same as rfc7231.txt

rfc7231\_trunc.txt – first 3.4k lines of rfc7231.txt (5.6k lines)

rfc7540.txt - HTTP/2 PROPOSED STANDARD

TimeSync-1.txt – introduction about Wireless time synchronization

TimeSync-2.txt – vendor webpage (Time Synchronization Systems)

# Results

Our program analyzed similarity between documents using Jaccard and MinHashing methods. Metadata of NSF Awards didn’t correlate in between and with other documents.

JaccardSim(a0300128.txt,a0300005.txt) = 0.005115089514066497

JaccardSim(a0300128.txt,rfc7231.txt) = 4.2354934349851756E-4

<…>

Introduction document of time synchronization didn’t corelate with vendors webpage, however similarity was the highest (0.0087) comparing with other documents. It is clear that the webpage wasn’t optimized for search engines…

JaccardSim(TimeSync-1.txt,TimeSync-2.txt) = 0.008762322015334063

Analysis of HTTP/1.1 draft standard (rfc2616.txt) with Semantics and Content section of proposed standard (rfc7231.txt) showed correlation using MinHashing method:

JaccardSim(rfc7231.txt,rfc2616.txt) = 0.058

MinHashSim(rfc7231.txt,rfc2616.txt) = 0.25

MinHashing signature length is 8 here, increasing it to 30 it gives 0.066 similarity between these two documents.

Comparison truncated version with full version of document shows strong correlation:

JaccardSim(rfc7231.txt,rfc7231\_trunc.txt) = 0.6666871203288949

MinHashSim(rfc7231.txt,rfc7231\_trunc.txt) = 0.625

Both methods identified identical documents:

JaccardSim(rfc7231.txt,rfc7231\_duplicate.txt) = 1.0

MinHashSim(rfc7231.txt,rfc7231\_duplicate.txt) = 1.0

## Performance

Valid performance comparison between methods is not possible due Spark overhead and small set of documents. Comparing all shingles using Jaccard method took 154ms, comparing signatures of MinHashing – 2ms, however generation of signatures took 789ms because Spark was involved.

Comparison running Spark on one core:

Generation of shingles(Spark): 3275ms

Jaccard similarity: 154ms

MinHashing signatures(Spark): 789ms

MinHashing comparison: 2ms

Overhead is more noticeable processing using two cores:

Generation of shingles(Spark): 3648ms

Jaccard similarity: 169ms

MinHashing signatures(Spark): 1627ms

MinHashing comparison: 3ms

## Output

Elapsed time: 3736ms

JaccardSim(rfc7540.txt,TimeSync-1.txt) = 0.00220097668340326

JaccardSim(rfc7540.txt,TimeSync-2.txt) = 8.44475721323012E-4

JaccardSim(rfc7540.txt,rfc7230.txt) = 0.03753065358780254

JaccardSim(rfc7540.txt,a0300128.txt) = 5.647324579980235E-4

JaccardSim(rfc7540.txt,a0300005.txt) = 7.097232079489E-5

JaccardSim(rfc7540.txt,rfc7231.txt) = 0.03640726774082962

JaccardSim(rfc7540.txt,rfc7231\_trunc.txt) = 0.03445111778445112

JaccardSim(rfc7540.txt,a0300125.txt) = 4.880769767117557E-4

JaccardSim(rfc7540.txt,rfc2616.txt) = 0.03182088841252828

JaccardSim(rfc7540.txt,rfc7231\_duplicate.txt) = 0.03640726774082962

JaccardSim(TimeSync-1.txt,TimeSync-2.txt) = 0.008762322015334063

JaccardSim(TimeSync-1.txt,rfc7230.txt) = 0.00201765447667087

JaccardSim(TimeSync-1.txt,a0300128.txt) = 0.0

JaccardSim(TimeSync-1.txt,a0300005.txt) = 0.0012674271229404308

JaccardSim(TimeSync-1.txt,rfc7231.txt) = 0.0017157732812684888

JaccardSim(TimeSync-1.txt,rfc7231\_trunc.txt) = 0.0019165432528965938

JaccardSim(TimeSync-1.txt,a0300125.txt) = 9.551098376313276E-4

JaccardSim(TimeSync-1.txt,rfc2616.txt) = 0.002303762812593903

JaccardSim(TimeSync-1.txt,rfc7231\_duplicate.txt) = 0.0017157732812684888

JaccardSim(TimeSync-2.txt,rfc7230.txt) = 4.505664263645726E-4

JaccardSim(TimeSync-2.txt,a0300128.txt) = 0.0

JaccardSim(TimeSync-2.txt,a0300005.txt) = 0.0

JaccardSim(TimeSync-2.txt,rfc7231.txt) = 6.034274680183442E-4

JaccardSim(TimeSync-2.txt,rfc7231\_trunc.txt) = 5.383097075183922E-4

JaccardSim(TimeSync-2.txt,a0300125.txt) = 0.0

JaccardSim(TimeSync-2.txt,rfc2616.txt) = 3.0341851527206526E-4

JaccardSim(TimeSync-2.txt,rfc7231\_duplicate.txt) = 6.034274680183442E-4

JaccardSim(rfc7230.txt,a0300128.txt) = 3.873716831299632E-4

JaccardSim(rfc7230.txt,a0300005.txt) = 1.2978585334198572E-4

JaccardSim(rfc7230.txt,rfc7231.txt) = 0.053628076268073595

JaccardSim(rfc7230.txt,rfc7231\_trunc.txt) = 0.04962230793956927

JaccardSim(rfc7230.txt,a0300125.txt) = 8.942258559018906E-4

JaccardSim(rfc7230.txt,rfc2616.txt) = 0.050029401387745503

JaccardSim(rfc7230.txt,rfc7231\_duplicate.txt) = 0.053628076268073595

JaccardSim(a0300128.txt,a0300005.txt) = 0.005115089514066497

JaccardSim(a0300128.txt,rfc7231.txt) = 4.2354934349851756E-4

JaccardSim(a0300128.txt,rfc7231\_trunc.txt) = 4.5049103522839894E-4

JaccardSim(a0300128.txt,a0300125.txt) = 0.012441679626749611

JaccardSim(a0300128.txt,rfc2616.txt) = 4.728292073356074E-4

JaccardSim(a0300128.txt,rfc7231\_duplicate.txt) = 4.2354934349851756E-4

JaccardSim(a0300005.txt,rfc7231.txt) = 1.2158793847650313E-4

JaccardSim(a0300005.txt,rfc7231\_trunc.txt) = 1.8150467374534896E-4

JaccardSim(a0300005.txt,a0300125.txt) = 0.005309734513274336

JaccardSim(a0300005.txt,rfc2616.txt) = 1.3542795232936077E-4

JaccardSim(a0300005.txt,rfc7231\_duplicate.txt) = 1.2158793847650313E-4

JaccardSim(rfc7231.txt,rfc7231\_trunc.txt) = 0.6666871203288949

JaccardSim(rfc7231.txt,a0300125.txt) = 6.587615283267457E-4

JaccardSim(rfc7231.txt,rfc2616.txt) = 0.05868038656747549

JaccardSim(rfc7231.txt,rfc7231\_duplicate.txt) = 1.0

JaccardSim(rfc7231\_trunc.txt,a0300125.txt) = 5.322451876164287E-4

JaccardSim(rfc7231\_trunc.txt,rfc2616.txt) = 0.05072047614075389

JaccardSim(rfc7231\_trunc.txt,rfc7231\_duplicate.txt) = 0.6666871203288949

JaccardSim(a0300125.txt,rfc2616.txt) = 9.741677584063959E-4

JaccardSim(a0300125.txt,rfc7231\_duplicate.txt) = 6.587615283267457E-4

JaccardSim(rfc2616.txt,rfc7231\_duplicate.txt) = 0.05868038656747549

Elapsed time: 149ms

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Elapsed time: 1250ms

MinHashSim(rfc7540.txt,TimeSync-1.txt) = 0.0

MinHashSim(rfc7540.txt,TimeSync-2.txt) = 0.0

MinHashSim(rfc7540.txt,rfc7230.txt) = 0.0

MinHashSim(rfc7540.txt,a0300128.txt) = 0.0

MinHashSim(rfc7540.txt,a0300005.txt) = 0.0

MinHashSim(rfc7540.txt,rfc7231.txt) = 0.0

MinHashSim(rfc7540.txt,rfc7231\_trunc.txt) = 0.0

MinHashSim(rfc7540.txt,a0300125.txt) = 0.0

MinHashSim(rfc7540.txt,rfc2616.txt) = 0.0

MinHashSim(rfc7540.txt,rfc7231\_duplicate.txt) = 0.0

MinHashSim(TimeSync-1.txt,TimeSync-2.txt) = 0.0

MinHashSim(TimeSync-1.txt,rfc7230.txt) = 0.0

MinHashSim(TimeSync-1.txt,a0300128.txt) = 0.0

MinHashSim(TimeSync-1.txt,a0300005.txt) = 0.0

MinHashSim(TimeSync-1.txt,rfc7231.txt) = 0.0

MinHashSim(TimeSync-1.txt,rfc7231\_trunc.txt) = 0.0

MinHashSim(TimeSync-1.txt,a0300125.txt) = 0.0

MinHashSim(TimeSync-1.txt,rfc2616.txt) = 0.0

MinHashSim(TimeSync-1.txt,rfc7231\_duplicate.txt) = 0.0

MinHashSim(TimeSync-2.txt,rfc7230.txt) = 0.0

MinHashSim(TimeSync-2.txt,a0300128.txt) = 0.0

MinHashSim(TimeSync-2.txt,a0300005.txt) = 0.0

MinHashSim(TimeSync-2.txt,rfc7231.txt) = 0.0

MinHashSim(TimeSync-2.txt,rfc7231\_trunc.txt) = 0.0

MinHashSim(TimeSync-2.txt,a0300125.txt) = 0.0

MinHashSim(TimeSync-2.txt,rfc2616.txt) = 0.0

MinHashSim(TimeSync-2.txt,rfc7231\_duplicate.txt) = 0.0

MinHashSim(rfc7230.txt,a0300128.txt) = 0.0

MinHashSim(rfc7230.txt,a0300005.txt) = 0.0

MinHashSim(rfc7230.txt,rfc7231.txt) = 0.125

MinHashSim(rfc7230.txt,rfc7231\_trunc.txt) = 0.125

MinHashSim(rfc7230.txt,a0300125.txt) = 0.0

MinHashSim(rfc7230.txt,rfc2616.txt) = 0.0

MinHashSim(rfc7230.txt,rfc7231\_duplicate.txt) = 0.125

MinHashSim(a0300128.txt,a0300005.txt) = 0.0

MinHashSim(a0300128.txt,rfc7231.txt) = 0.0

MinHashSim(a0300128.txt,rfc7231\_trunc.txt) = 0.0

MinHashSim(a0300128.txt,a0300125.txt) = 0.0

MinHashSim(a0300128.txt,rfc2616.txt) = 0.0

MinHashSim(a0300128.txt,rfc7231\_duplicate.txt) = 0.0

MinHashSim(a0300005.txt,rfc7231.txt) = 0.0

MinHashSim(a0300005.txt,rfc7231\_trunc.txt) = 0.0

MinHashSim(a0300005.txt,a0300125.txt) = 0.125

MinHashSim(a0300005.txt,rfc2616.txt) = 0.0

MinHashSim(a0300005.txt,rfc7231\_duplicate.txt) = 0.0

MinHashSim(rfc7231.txt,rfc7231\_trunc.txt) = 0.625

MinHashSim(rfc7231.txt,a0300125.txt) = 0.0

MinHashSim(rfc7231.txt,rfc2616.txt) = 0.25

MinHashSim(rfc7231.txt,rfc7231\_duplicate.txt) = 1.0

MinHashSim(rfc7231\_trunc.txt,a0300125.txt) = 0.0

MinHashSim(rfc7231\_trunc.txt,rfc2616.txt) = 0.125

MinHashSim(rfc7231\_trunc.txt,rfc7231\_duplicate.txt) = 0.625

MinHashSim(a0300125.txt,rfc2616.txt) = 0.0

MinHashSim(a0300125.txt,rfc7231\_duplicate.txt) = 0.0

MinHashSim(rfc2616.txt,rfc7231\_duplicate.txt) = 0.25

Elapsed time: 6ms

Process finished with exit code 0