IR Assignment 1 Report

Name: Rahul Aravind Mehalingam

Net Id: rxm151730

Tokenization:

The following design choices were made for the tokenization process.

- Each xml file is parsed using SAXParser. Each XML element contents are read and stored into a string buffer.
- The tokens of the file contents are split using the following delimiters: \s+, /, \, - and comma.
- In each token, the possessives (‘s) are replaced with a null character.
- All the special characters except dot (.) in a token are replaced with a null character.
- The tokens which contain only the digits are skipped.
- Period in a token is handled in a special way. The token which attributes to the Bibi logical references like C1.245, 1.234, acronyms (U.S, U.N) are retained. The tokens such as ae., j., are skipped. The regex is used to determine the aforementioned pattern. The regex used was ^(\\w+)([\\.])(\\w+)+ Which means the token should start with either alphabets or numbers or both followed by a dot followed by alphabets or numbers.
- Each token is converted into lowercase. Token is trimmed.

The token after the pre-processing is stored in a HashMap with the key as token and the count of occurrence as value.

The token summary for the entire cranfield collection is recorded in a TokenSummary object. The attributes of the TokenSummary object are total token count, unique token count, average token, token hash map and sorted list (for retrieving the top 30 frequently used tokens). The total time taken for the tokenization process was 1.953 s.

Stemming:

The porter stemmer code is downloaded from the link provided in the website. Each token in the token hash map is passed to the stemmer and the stemmed token is maintained in a separate hash map with the key as a stemmed token and count as the value.

The Stem Token Summary is recorded in a TokenSummary object and similarly the attributes such as total stem token count, unique token stem count, average stem token count per document, token stem hashMap and sorted list (the map is converted to a sorted list for retrieving the top 30 most frequent used stems).

RESULT:

The README file contains the instructions on how to run the program. After execution of the program, the results were as follows.

```
{csgrads1:~/IR} java -jar TokenizationAndStemmer.jar

Enter the directory with full absolute path

/people/cs/s/sanda/cs6322/Cranfield

*******************************************

TOKEN SUMMARY OF CRANFIELD DATASET

*******************************************

The number of tokens in the collection: 229713

The number of unique tokens in the collections: 9428

The average number of word tokens per document: 164.0

The number of words that occur only once in the cranfield collection: 3963


Token                    Frequency

---------------------------------------------------


the                       19449


of                       12717


and                        6675


a                        5928
```

in	4636
to	4563
is	4114
for	3493
are	2429
with	2265
on	1944
flow	1848
at	1834
by	1754
that	1570
an	1388
be	1271

pressure	1207
boundary	1156
from	1116
as	1113
this	1081
layer	1002
which	975
number	973
results	885
it	855
mach	824
theory	788
shock	712

```
Time taken for tokenization process: 1924 ms


*****************************************
TOKEN STEM SUMMARY OF CRANFIELD DATASET
*****************************************
The Number of distinct stems in the cranfield text collection: 6731
The average number of word stems per document: 164.0
The Number of stems that occur only once in the document: 2902


Token                    Frequency
-----------------------------------------------------


the                       19449


of                       12717


and                        6675


a                        5928


in                       4636


to                        4563


is                       4114


for                       3493
```

ar	2454
with	2265
on	2262
flow	2079
at	1834
by	1754
that	1570
an	1388
pressur	1382
be	1368
number	1347
boundari	1185
layer	1134

```
from                   1116

as                     1113

result                 1087

thi                    1081

it                     1042

effect                  996

which                   975

method                  887

theori                  881

{csgrads1:~/IR}
```