Problem Set 3: Inverted Index and Query Processor

Renae Fisher
CS 4903: Text Mining — Professor Mackey
July 27, 2019

Methodology

Text Normalization

The UAInvertedIndex class transforms any text that it will read or write. But, it expects the tokens to be lowercase and devoid of most punctuation. Its text cleaning method replaces all non-ASCII characters with a single-byte character and returns a trimmed String. It's necessary to replace such characters to ensure that each record will have a fixed length. The UAQuery class performs the same steps as UAInvertedIndex, except it uses additional techniques from the pre-processing phase. Its method convertText removes some punctuation, and it uses other forms of punction as a delimiter. The method also reduces the case of the word, performs stemming, and trims the word to a specific length.

BUILDINVERTEDINDEX

Time complexity
$$O(DT \log V_D) + O(D \log D) + O(T \log^2 D) + O(V)$$

Space complexity $O(V_D) + O(D) + O(\log D) + O(V)$

The BuildinvertedIndex method first makes a call to AlgoOne. This method uses a loop to process a list of files in an input directory. At each iteration, AlgoOne uses a loop and a TreeMap to count the frequencies of tokens in a document. The two loops take O(DT) time total. In this case, D and T refer to the documents and tokens in the input directory. The TreeMap is comprised of a red-black tree and it maintains order for all items in the tree. It takes $log(V_D)$ time, where V_D represents the distinct

tokens in a document, to retrieve and add items to the map. Altogether, it takes ALGOONE exactly $O(DT \log V_D)$ time to count the frequencies of tokens and arrange them in sorted order. After the inner loop expires, ALGOONE calls WRITETEMPFILE. This method uses a loop to write the distinct tokens in TreeMap to separate files. This adds an insignificant cost of $O(V_D)$.

Next, BuildinvertedIndex calls the method MergeSort, which uses an iterative sort to combine most of the temporary files. It first creates an array to hold all temporary files, then it uses two loops to merge the contents of those files. The outer loop divides the file array into segments. Its control variable increases by a multiple of two after each iteration, which approximates a logarithmic run time. The code in the inner loop, including the call to Merge, processes a segment delineated by the outer loop in linear time. Altogether, the method MergeSort takes $O(D \log D)$ time and O(D) space to merge the files in the temporary directory.

Last, the method BuildinvertedIndex uses algoTwo to construct the post.raf and dict.raf files. First, algoTwo opens the directory of merged files, which have been reduced to $O(\log D)$ space by MERGESORT. Next, the method uses two loops to determine the appropriate order of the tokens as it writes them to post.raf. The outer loop runs until there are no empty BufferedReaders, and there are $\log D$ total readers. The inner loop examines each reader as it finds the token alphabetically first in the buffer. Together, the two loops take $O(T \log^2 D)$ time to create the postings file from the directory of merged files. Last, AlgoTwo uses a loop to write the global hash table to the hard drive as dict.raf. This process takes O(V) time and space, where V represents the distinct terms between all documents.

RUNQUERY

Time complexity
$$O(QD_TT) + O(V_TD_T)$$
 Space complexity
$$O(V_T + V_D + V_Q) + O(V_TV_D) + O(\log V_D)$$

The method RUNQUERY uses two methods to build a term-document matrix. First, it creates three hash tables that each hold the number of distinct terms, documents, or words in the query. The termMap hash table takes V_T space, the docMap hash

table takes V_D space, and the q hash table takes V_Q space. Next, runQuery calls MAPROWSCOLS, which maps distinct terms and document IDs to the rows and columns of a matrix. The method MAPROWSCOLS first iterates over the words in the query. It locates each word in the dictionary file, which would take constant time at best. If the word exists, the method looks for more information in the postings file. It takes D_T time, the number of documents with term T, to examine each posting. For each posting, MAPROWSCOLS opens the document and adds each term in it to the termMap. Altogether, MAPROWSCOLS takes $O(QD_TT)$ time and $O(V_T + V_D + V_Q)$ space total.

Once the method runQuery is finished, RUNQUERY makes a call to buildTDM. This method creates a term-document matrix that is V_T rows by V_D columns, which corresponds to the number of items in termMap and docMap. Then, buildTDM iterates over the items in termMap. Like MAPROWSCOLS, the method retrieves information from the dictionary file and postings. But, it stores the rtf-idf values of a term and document in the term-document matrix. Overall, it takes buildTDM $O(V_TD_T)$ time and $O(V_TV_D)$ space to construct the term-document matrix.

After the term-document matrix is complete, the method RUNQUERY uses getDocs to find documents that match the query vector. It iterates over the docMap and uses its information to calculate the cosine similarity between each document and the query vector. It takes V_D time to iterate over docMap, and V_T time to calculate cosine similarity between two vectors. The method stores each result in a priority queue, whose add operation takes logarithmic time. Altogether, the method getDocs takes $O(V_DV_T)$ time and uses $O(\log V_D)$ space to find the k documents that are most similar to a query.

Results

The following pages demonstrate the performance of the Buildinvertedindex and RunQuery methods against a portion of tokenized data from the Common Crawl data set. Each section shows the run-time of the application as it performs an indexing or querying task. The RunQuery section shows a sample of the top ten documents that the search-engine returned for each query.

BUILDINVERTEDINDEX

 $real \quad 4m37.982s$

 $user \quad 3m38.643s$

sys 1m7.179s

RUNQUERY

Computer

```
computer raspberr
         minnesot institut hardware unlimited how extra models malware study
  free linkedin requirs submenu pure
           section county get click will tech
when now science courss software graz foundation out busins miles soluts educs vision
not itm solutions in the solution of the solution fonts web rated solutions earth systems constructed the solution fonts web rated solutions earth systems costeachers repair run universi interacticon teaching learn thomas page weeks tel security technolo one topics can work other system cookis appropriate and security systems.
computin skills lit understa datawho works share blue computs educatio online
    problems computr short
      learning
      upgrade compons coure
                  benefits
coding
       \#22963
                      022963uafs.html.out
                                                       0.5634
       \#13962
                      013962 uafs.html.out
                                                       0.4854
                      034826uafs.html.out
       \#34826
                                                       0.4805
      \#33335
                      033335uafs.html.out
                                                       0.4293
       \#39357
                      039357uafs.html.out
                                                       0.3856
       \#26145
                      026145 uafs.html.out
                                                       0.3856
       \#26146
                      026146uafs.html.out
                                                       0.3856
       \#21331
                      021331uafs.html.out
                                                       0.3827
       \#15852
                      015852uafs.html.out
                                                       0.3807
```

0 m 37.702 s

#36639

real

036639uafs.html.out

0.3260

Big Data

sentry parts governme
competit storis globalhe
pro customr posts
static analysis companis experien
ldkstrategifocus come info
our actions not cdn will how innovati
interpre journeis 2017 analyts what cases
find leads get view some info
converss uri their touch blog hatena icon
michal story registr hatenabl jamie they
potentia like need but help might
that postd cata comments fblog
image strategy nus employr use
consulti make advancd othr
organisa informat governments globalhe
informat governments
big parastor
big parastor
what cases
mail ret website
touch blog hatena icon
michal story registr hatenabl jamie they
potentia like need but help might
add
comments fblog
image strategy nus employr use
consulti make advancd othr
organisa read
generati understa

#23529	023529uafs.html.out	0.6231639
#42494	042494 uafs.html.out	0.6147945
#42495	042495 uafs.html.out	0.6147945
#438	000438 uafs.html.out	0.5418439
#1794	001794 uafs.html.out	0.5348657
#43510	043510 uafs.html.out	0.4739626
#35849	035849 uafs.html.out	0.43092135
#1795	001795 uafs.html.out	0.42124513
#7360	007360 uafs.html.out	0.4105744
#42770	042770 uafs.html.out	0.39694285

 $\begin{array}{ll} \text{real} & 0\text{m}48.793\text{s} \\ \text{user} & 0\text{m}32.625\text{s} \\ \text{sys} & 0\text{m}20.384\text{s} \end{array}$

Big Data and Data Analytics

```
marketin analyze
journeis static question storis
analyts add tools articls
story into like consulti journy info
blogldkwhere analysis companis pro
make analysts might data permalin cdn
big analit what bettr find actions
analit what bettr find actions
analit what insights when domain that
strategi jamie bad insights when domain that
wexperiengetret may mail hatenabl
lead customr customs read image deepr
identify comments converss use right
their visualiz platform but using they have
wixstati behavior hatena registr not innovati
understa 2017
website linkedin mailto will can
need take postd cases feed how
article advancd potentia demo
governme
our
through
```

#1794	001794 uafs. html. out	0.7212369
#42494	042494 uafs.html.out	0.66177726
#42495	042495 uafs.html.out	0.66177726
#34892	034892 uafs.html.out	0.5747477
#438	000438 uafs.html.out	0.5336642
#1795	001795 uafs. html. out	0.52219814
#43510	043510 uafs.html.out	0.510183
#23529	023529 uafs.html.out	0.48249996
#35849	035849 uafs.html.out	0.46385247
#35976	035976 uafs.html.out	0.43618143

 $\begin{array}{ll} {\rm real} & 0{\rm m}50.788{\rm s} \\ {\rm user} & 0{\rm m}34.912{\rm s} \\ {\rm sys} & 0{\rm m}20.473{\rm s} \end{array}$

Text Mining

squaresp our drilling markt canada comment million take gold website markets inc datats where philippe years any developm get sign undergro can rights they article but now mine one some year blog its job does countyher san two were servics people been stone browe coal had reports mines cdn work his fournir out uncertai data honduras alo than time asx their contamin australi equipmen posts grant mitchell them what security know company communit has

#17094	017094 uafs. html. out	0.84758025
#17093	017093 uafs.html.out	0.768427
#27844	027844 uafs.html.out	0.7282029
#2745	002745 uafs.html.out	0.52677673
#1005	001005 uafs.html.out	0.46474215
#32854	032854 uafs.html.out	0.43216804
#31332	03133 uafs. html. out	0.40972403
#19839	019839 uafs.html.out	0.33095247
#11745	011745 uafs.html.out	0.28476787
#17117	017117 uafs.html.out	0.24425438
real	0 m 31.389 s	
user	0 m 18.427 s	

 $0\mathrm{m}14.111\mathrm{s}$

sys

uafs.edu

admisss
augut psycholo
service college academic
january clevel resourcs
faculty session manageme universi
week novembr middleto february
science scholars 2018 february
science scholars 2018 school main
wordprs dept tuition school main
director hall athlets net specialibody like march
director hall programs students collegs vally
sciencs transfrinternat blog based
centrawards englih events
online alpha administ studis study
library chaptrcampus icon our communitservics
april 2020 office mailto event visit web
medicine graduate leadersh aid column health
fairfax presiden year mind
informat nutritio how info financia social
building program news apply arts
educatio support alumni content
academs integrat departme
calendar student admisss calendar student

#22711	uafs.html.out	0.96920776
#25562	025562 uafs. html. out	0.9411387
#27083	027083 uafs.html.out	0.9291655
#33572	033572 uafs. html. out	0.9002196
#33571	033571 uafs. html. out	0.8953159
#41777	0.87936157 uafs.html.out	041777
#30227	030227 uafs. html. out	0.87801754
#22872	02287 uafs.html.out	0.8082933
#35015	035015 uafs.html.out	0.7995487
#12063	012063 uafs. html. out	0.78565556
real	0 m 35.274 s	
user	0 m 20.472 s	
sys	0 m 16.331 s	

Data Science, Artificial Intelligence, and 123 Cake Bakeries

wedding marketin
website content connecti
disrupt 2018 birthday currency
theme
how shape online friendnatural cream nov
feel beautifu talks intervie
cakes where aprilera serisniche writeattentio
wishs pops day making take ordr
western markt peterson dec what feb
gold expert jan bakeryplatform story whats
jason shop which follow icm download much
donuts talk cake encyclop sep busins bet disrupti
permalin purple memorand psycholo 2019 recipe
monopoly onl our crown
puh program expertie into share
cookis chocolat
upon black jordan
intellic greeting

#29905	029905 uafs.html.out	0.25840327
#12101	012101 uafs.html.out	0.25634882
#5573	005573 uafs.html.out	0.24116346
#7849	007849 uafs.html.out	0.23287354
#2662	002662 uafs.html.out	0.21350399
#28585	028585 uafs.html.out	0.20389752
#28717	028717 uafs.html.out	0.1999392
#28586	028586 uafs.html.out	0.19864368
#27943	027943 uafs.html.out	0.18948705
#22705	022705 uafs.html.out	0.18580127
real	0 m 47.304 s	
user	$0 \mathrm{m} 31.457 \mathrm{s}$	

0 m 19.926 s

sys