LJ Brown

CSE 5337/7337: Information Retrieval and Web Search

Spring 2018, Project 2: Query engine implementation (100 points)

Due: Tuesday May 8, 6:00 pm

Demonstrations are ***encouraged***, but not required.

Deliverables:

1. Complete code in a compressed archive (zip, tgz, etc)
2. A readme file with complete description of used software, installation, compilation and execution instructions to allow me to install and run your program if needed.

*If a demonstration is not scheduled, then I must be able to install and run your program to ensure it works as you described, so test your directions!*

1. A document with the results for the tasks below.

Task:

**Develop a simplified query engine.**Test your data only on the data in:

<https://lyle.smu.edu/~fmoore> aka <https://s2.smu.edu/~fmoore>

1. Use the web crawler you built in Project 1 that crawled a limited space, looking for text, html files and php files. Everyone will need to save the text from the <TITLE> tag. You may need to modify how you saved the information from the documents that you traversed to support the query engine. You may assume a maximum of 60 documents will need to be saved. Describe in detail what was changed to support the second half of the project. [10/10 points]

I had to code to create a document term frequency matrix from the scraped site, along with that I had to implement several tools for operating on matrices and vectors for cluster pruning, kmeans clustering, and other matrix vector arithmetic tools. In addition I created a query engine to handle user input and display results.

1. Produce a clustering of the documents into 5 leader/follower sets. [30/25 points]  
   a) How did you determine the leaders?

I first used k means clustering to find the centroids. Then I found the document vector with the highest cosine similarity to each centroid and chose that as the leader.

b) List the leader-id, document-id, score for each pair.

term frequency clustering information: (cluster leaders chosen using kmeans)

cluster 1:

leader docID: 1

follower docIDs and leader/follower scores:

docID: 12, score: 0.0773067331670823

docID: 3, score: 0.07231920199501247

docID: 2, score: 0.032418952618453865

docID: 11, score: 0.011637572734829594

docID: 7, score: 0.00914380714879468

cluster 2:

leader docID: 22

follower docIDs and leader/follower scores:

docID: 36, score: 0.02493765586034913

docID: 33, score: 0.023275145469659187

docID: 25, score: 0.019118869492934332

docID: 21, score: 0.01828761429758936

docID: 28, score: 0.017456359102244388

cluster 3:

leader docID: 26

follower docIDs and leader/follower scores:

docID: 36, score: 0.02826267664172901

docID: 28, score: 0.00997506234413965

docID: 7, score: 0.00831255195344971

docID: 21, score: 0.00831255195344971

docID: 22, score: 0.005818786367414797

cluster 4:

leader docID: 2

follower docIDs and leader/follower scores:

docID: 1, score: 0.032418952618453865

docID: 4, score: 0.024106400665004156

docID: 12, score: 0.020781379883624274

docID: 3, score: 0.0199501246882793

docID: 11, score: 0.01828761429758936

cluster 5:

leader docID: 34

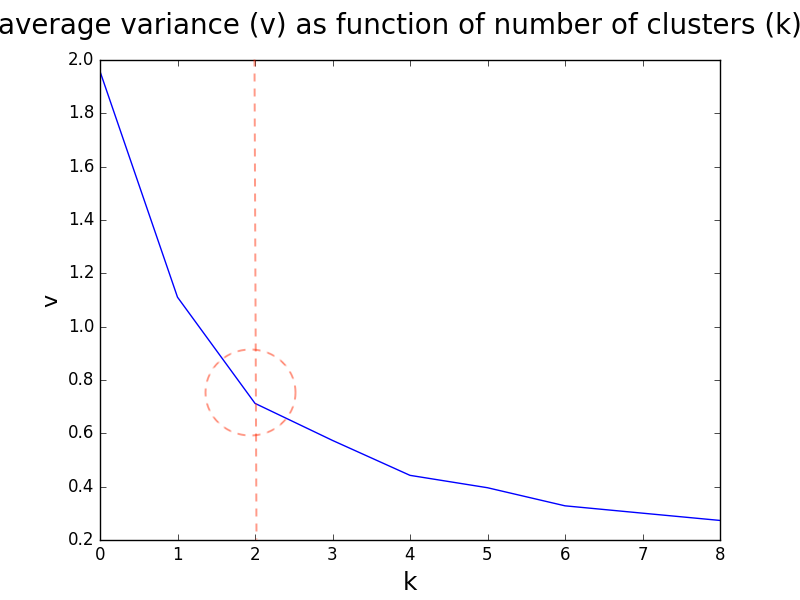
follower docIDs and leader/follower scores:

docID: 7, score: 0.00914380714879468

docID: 20, score: 0.006650041562759767

docID: 25, score: 0.006650041562759767

docID: 37, score: 0.006650041562759767

docID: 14, score: 0.006650041562759767  
c) Do you agree with the clustering? Is 5 reasonable?

Looking at the average variance of the centroids as a function of the number of clusters it looks like the elbow occurs around k=2. So I think 5 clusters is a few to many.

1. The user will be able to enter multiple queries, consisting of one or more query words separated by space. The single word query “stop” will cause your program to stop. [10/10 points]   
   a) What happens if a user enters a word that is not in the dictionary?

If the user enters a word that is not in the dictionary no results are returned.  
b) What happens if a user enters a stop word?

If the user enters a stop word no results are returned.  
c) A set of queries will be provided.

User Query : moore smu

Indexed Tokens : ['moor', 'smu']

Search Type: full\_search, Weighting Type: tfidf

RESULTS:

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

1. URL: https://s2.smu.edu/~fmoore/index\_duplicate.htm,

TITLE: Freeman Moore - SMU Spring 2018,

SCORE: 0.5041562759767249

2. URL: https://s2.smu.edu/~fmoore/index-final.htm,

TITLE: Freeman Moore - SMU Spring 2017,

SCORE: 0.5041562759767249

3. URL: https://s2.smu.edu/~fmoore/index-fall2017.htm,

TITLE: Freeman Moore - SMU Fall 2017,

SCORE: 0.5041562759767249

4. URL: https://s2.smu.edu/~fmoore/schedule.htm,

TITLE: SMU CSE 5337/7337 Spring 2018 Schedule,

SCORE: 0.25166251039068993

5. URL: https://s2.smu.edu/~fmoore/textfiles/index.html,

TITLE: SMU CSE 5/7337 Spring 2018 Textfiles,

SCORE: 0.250831255195345

User Query : Bob Ewell where Scout

Indexed Tokens : ['bob', 'ewel', 'scout']

Search Type: full\_search, Weighting Type: tfidf

RESULTS:

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

1. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird5.html,

TITLE: Mockingbird part 5,

SCORE: 0.012468827930174564

2. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird4.html,

TITLE: Mockingbird part 4,

SCORE: 0.005818786367414797

3. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird3.html,

TITLE: Mockingbird part 3,

SCORE: 0.0033250207813798837

4. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird2.html,

TITLE: Mockingbird part 2,

SCORE: 0.0024937655860349127

5. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird1.html,

TITLE: Mockingbird novel part 1,

SCORE: 0.0016625103906899418

6. URL: https://s2.smu.edu/~fmoore/textfiles/baseball4.txt,

TITLE: NO TITLE,

SCORE: 0.0008312551953449709

User Query : three year story

Indexed Tokens : ['stori', 'three', 'year']

Search Type: full\_search, Weighting Type: tfidf

RESULTS:

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

1. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird4.html,

TITLE: Mockingbird part 4,

SCORE: 0.004156275976724855

2. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird1.html,

TITLE: Mockingbird novel part 1,

SCORE: 0.0024937655860349127

3. URL: https://s2.smu.edu/~fmoore/textfiles/basketball1.txt,

TITLE: NO TITLE,

SCORE: 0.0016625103906899418

4. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird2.html,

TITLE: Mockingbird part 2,

SCORE: 0.0016625103906899418

5. URL: https://s2.smu.edu/~fmoore/textfiles/basketball5.txt,

TITLE: NO TITLE,

SCORE: 0.0008312551953449709

6. URL: https://s2.smu.edu/~fmoore/textfiles/basketball2.txt,

TITLE: NO TITLE,

SCORE: 0.0008312551953449709

User Query : Atticus to defend Maycomb

Indexed Tokens : ['atticu', 'defend', 'maycomb']

Search Type: full\_search, Weighting Type: tfidf

RESULTS:

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

1. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird4.html,

TITLE: Mockingbird part 4,

SCORE: 0.010806317539484621

2. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird5.html,

TITLE: Mockingbird part 5,

SCORE: 0.006650041562759767

3. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird1.html,

TITLE: Mockingbird novel part 1,

SCORE: 0.004156275976724855

4. URL: https://s2.smu.edu/~fmoore/textfiles/basketball5.txt,

TITLE: NO TITLE,

SCORE: 0.0008312551953449709

5. URL: https://s2.smu.edu/~fmoore/misc/text/mockingbird2.html,

TITLE: Mockingbird part 2,

SCORE: 0.0008312551953449709

6. URL: https://s2.smu.edu/~fmoore/textfiles/baseball2.txt,

TITLE: NO TITLE,

SCORE: 0.0008312551953449709

User Query : hocuspocus thisworks

Indexed Tokens : []

Search Type: full\_search, Weighting Type: tfidf

RESULTS:

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

No Results Found.

1. Implement the cosine similarity of the query against each ~~leader~~ document. [40/30 points]   
   a) What document/query weighting scheme did you implement?

I allow the user to choose from pure term frequency weighting and tf-idf weighting schemes for both cluster pruning search and a full cosine similarity search over the entire corpus of documents.  
b) If any of the query words appear in the <title> of any selected document, add 0.25 to its score.  
c) Display the resulting score, document URL, and document title in descending numerical order for the top K=6 results.

See results from 5.c above

1. Explain why you believe the results are correct. [10/10 points]

This appears to be correct: searching document titles will return the correct results. Searching for individual terms that appear in the corpus will return every document it appears in if you ask it to. With the K=6 limitation it does appear to respond properly to title word matches giving them much more weight as can be seen with the first query for “moore smu” and the associated rankings.

**GRADUATE STUDENTS:**

1. Include in the display, the first 20 words of the document (this can be the stemmed version). [0/5 points]
2. If less than K/2 documents are returned for a query, rerun the query (item 4) using thesaurus expansion. A list of words, along with 1 – 3 synonyms will be provided. [0/10 points]

You must show the results of these queries (tentative):  
1. moore smu  
2. Bob Ewell where Scout  
3. three year story  
4. Atticus to defend Maycomb  
5. hocuspocus thisworks

You may certainly include additional queries that reflect your testing.

A small thesaurus is provided here. It is used when a query returns less than 3 documents. When that happens, internally expand the query using thesaurus expansion on each term and then execute the revised query.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| word | alternates | | | |
| beautiful | nice | fancy |  |  |
| chapter | chpt |  |  |  |
| chpt | chapter |  |  |  |
| responsible | owner | accountable |  |  |
| freemanmoore | freeman | moore |  |  |
| dept | department |  |  |  |
| brown | beige | tan | auburn |  |
| tues | Tuesday |  |  |  |
| sole | owner | single | shoe | boot |
| homework | hmwk | home | work |  |
| novel | book | unique |  |  |
| computer | cse |  |  |  |
| story | novel | book |  |  |
| hocuspocus | magic | abracadabra |  |  |
| thisworks | this | work |  |  |

(end)