Name: Prathusha JS Naidu

Project Phase 4
Information Retrieval

I. Introduction

The objective of this assignment is to build a command line retrieval engine on top of the inverted files. The retrieval engine should take in queries which are lists of words and display the ten top-ranking document identifiers or filenames to the user. This project has been built on Phase 3 and the following format should be used for executing the program.

python3 *<program_name> <input directory> <query along with term weights>*

Example: python3 retrieve.py input “1.0 International 1.0 affairs”

II. Input

The input for the program is a set of HTML documents and the query terms along with term weights.

III. Output

The program displays the top 10 highest ranked documents that match with the query terms. If the query terms aren’t found in any files, then the message “No documents found” is displayed.

IV. Implementation

The indexing results from Phase 3 have been used to build the search engine. Hence all of the preprocessing on the raw HTML files is covered in this part. Additionally, the query from the user is preprocessed by removing stop words and down-casing. The query is stored in the form of a vector which is later used for calculating cosine similarity.

Steps for calculating document-query similarity scores:

(TDM – Term Document Matrix)

1) Scores:

Cosine product or the dot product is used for calculating similarity scores.

→ For each term ‘t’ in query ‘q’

→ For each document in postings list of term ‘t’

→ Score[doc d]+= TDM[t][d] * weight

→ Magnitude[doc d]+= TDM[t][d] * TDM[t][d]

2) Normalize:

To compare scores, documents are normalized.

→ Ranking[doc d]= Score[doc d] / sqrt(Magnitude[doc d])

3) Highest Rankings:

"Ranking" is sorted to display the top 10 documents for each query. If no such documents exist, then an appropriate message is displayed.

The most important data structure used is a nested dictionary which provides us with the postings list. Dictionaries have also been used for hashing the document similarity scores.

V. Running Time

The outer loop in "Scores" is iterated for each term in the query and the inner loop is executed for every document in the postings list. Hence time complexity is approximately |q|*|d|, I.e number of query terms multiplied by number of documents.

VI. Results

query - **"1.0 international 0.4 affairs"**

```
prathusha@prathusha-Flex-3-1480:~/IR/hw4$ python3 retrieve.py input "1.0 interna
tional 0.4 affairs"
Document ID      Similarity Score
179              0.68
215              0.36
243              0.33
287              0.18
161              0.17
269              0.05
125              0.05
188              0.05
022              0.05
492              0.05
```

query - **"0.4 computer 1.0 network"**

```
prathusha@prathusha-Flex-3-1480:~/IR/hw4$ python3 retrieve.py input "0.4 compute
r 1.0 network"
Document ID      Similarity Score
161              0.67
181              0.37
315              0.22
135              0.14
016              0.14
221              0.09
446              0.08
223              0.05
140              0.05
087              0.05
prathusha@prathusha-Flex-3-1480:~/IR/hw4$
```

query- **“1.0 diet”**

```
prathusha@prathusha-Flex-3-1480:~/IR/hw4$ python3 retrieve.py input "1.0 diet"
Document ID      Similarity Score
009              1.00
263              0.47
018              0.39
252              0.18
353              0.13
050              0.11
152              0.10
prathusha@prathusha-Flex-3-1480:~/IR/hw4$
```

query - **“0.4 hydrotherapy”**

```
prathusha@prathusha-Flex-3-1480:~/IR/hw4$ python3 retrieve.py input "0.4 hydroth
erapy"
Document ID      Similarity Score
273              1.00
prathusha@prathusha-Flex-3-1480:~/IR/hw4$
```

query - **“1.0 identity 1.0 theft”**

```
prathusha@prathusha-Flex-3-1480:~/IR/hw4$ python3 retrieve.py input "1.0 identit
y 1.0 theft"
Document ID      Similarity Score
309              1.00
245              0.53
348              0.37
298              0.33
332              0.19
328              0.15
397              0.14
379              0.12
380              0.10
301              0.09
prathusha@prathusha-Flex-3-1480:~/IR/hw4$
```

query - **“0.9 Zimbabwe”**

```
prathusha@prathusha-Flex-3-1480:~/IR/hw4$ python3 retrieve.py input "0.9 Zimbabw
e"
No documents found
prathusha@prathusha-Flex-3-1480:~/IR/hw4$
```

query - **“0.7 peacemeal 0.4 popular 1.0 information”**

```
prathusha@prathusha-Flex-3-1480:~/IR/hw4$ python3 retrieve.py input "0.7 peaceme
al 0.4 popular 1.0 information"
Document ID      Similarity Score
292              1.00
447              0.54
352              0.41
064              0.23
163              0.23
351              0.16
157              0.15
029              0.13
017              0.13
133              0.12
prathusha@prathusha-Flex-3-1480:~/IR/hw4$
```

query - **“0.1 The 0.3 topic 0.7 is 0.4 about 0.3 international 1.0 affairs”**

```
prathusha@prathusha-Flex-3-1480:~/IR/hw4$ python3 retrieve.py input "0.1 The 0.3
 topic 0.7 is 0.4 about 0.3 international 1.0 affairs"
Document ID      Similarity Score
199              0.41
326              0.19
254              0.19
252              0.11
292              0.11
250              0.07
174              0.05
010              0.05
133              0.05
119              0.05
prathusha@prathusha-Flex-3-1480:~/IR/hw4$
```