CS172: Information Retrieval

UC Riverside

Prof. Mariam Salloum Winter 2019

Problem Set 1

You encouraged to discuss the assignment with your classmates, but eventually each student should sit-down and work on their own code.

1 Vector Space Model (Non-Programming)

In the vector space model, the input query and the documents in the collection are represented as vectors in V-dimensional space, where V denotes the size of the indexed vocabulary (i.e., the number of unique terms in the collection). Given a query, documents are scored (and ranked) based on their vector-space similarity to the query. In class, we talked about two vector space similarity measures: (1) the inner product and (2) the cosine similarity. The goal of this question is to understand their differences.

Suppose we have a collection of 8 documents (denoted as $D_1 \ldots D_8$ below). Answer the following questions. Assume a binary text representationa vectors value for a particular dimension (i.e., a particular index term) equals 1 if the term appears at least once and 0 otherwise. **Note: Please show your work for full credit**

- D_1: jack and jill went up the hill
- D_2: to fetch a pail of water
- D_3: jack fell down and broke his crown
- D_4: and jill came tumbling after
- D_5: up jack got and home did trot
- D_6: as fast as he could caper
- D_7: to old dame dob who patched his nob
- D_8: with vinegar and brown paper

(1) Given a query-vector q and a document-vector d, the inner product (i.e, the score given to document d for query q) is given by,

$$inner - product(q, d) = \sum_{i=1}^{V} (q_i * d_i)$$

Using the inner product, what is the score given to each document $D_1 \ldots D_8$ in response to the query 'jack'?

(2) Given a query-vector q and a document-vector d, the cosine similarity (i.e, the score given to document d for query q) is given by,

$$CosSim(q,d) = \frac{\sum_{i=1}^{V}(q_i * d_i)}{\sqrt{\sum_{i=1}^{V} q_i^2}\sqrt{\sum_{i=1}^{V} d_i^2}}$$

Using the cosine similarity, what is the score given to each document $D_1 \ldots D_8$ in response to the query 'jack'?

(3) For this particular query, scoring documents $D_1 \ldots D_8$ using the inner-product and the cosine similarity would result in equal rankings (HINT: if theyre not, you made a mistake). Why?

(4) Give an example of a query for which scoring documents $D_1 \ldots D_8$ using the inner-product and the cosine similarity would result in different rankings.

(5) The vector space model has the flexibility that it can accommodate different term-weighting schemes. Different term-weighting schemes make different assumptions about which terms are most important. Compute TF-IDF for terms in D_1. Use $D_1 \ldots D_8$ to compute corpus statistics such as df_1.

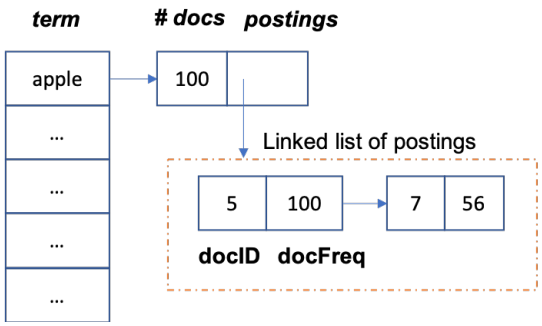
2 TF-IDF (Programming)

In this problem, you are asked to write code to compute the TF-IDF for terms in a document collection.

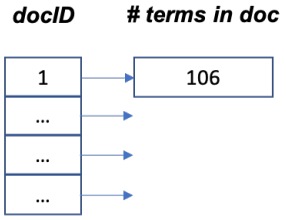
2.1 Inverted Index Creation

Write a program/class called **docIndex** that reads the provided input data and creates two indices, as defined below:

(a) Word / Posting Index: An inverted index (that excludes stop words from the file stoplist.txt) of terms/words and the corresponding frequency and documents in which the term is found. The index data structure should have some form such that you have random look-up of terms and allows new terms to be added without rebuilding it completely. The index should contain the document frequency of each term (i.e. the number of documents that contain the term).



(b) Document index: An inverted index that contains the number of terms in each document.



2.2 Test Program

Write a program/class called **test** which will utilize the created indices to compute the tf-idf term weightings.

Prompt the user for a term. If the term is in the word index, the program should display a list of the postings for that term. For each posting, it should generate the term weightings, tf, idf, and tf-idf in that order separated by commas.

If the term is not in the inverted file, it should display a suitable message.

The program should loop until the user enter "QUIT".

2.3 README

Include a readme file that describes the design of your code and detailed instructions for running your program. We will not grade programs that don't compile, and won't attempt to run your program if the instructions are not provided.

3 Submission

You will be submitting your solution to part 1 (word doc, text, PDF, etc.) and source code for part 2 via Github. Your submission should include solution to problem 1, all your source files for problem 2, a README files, and the output data (for the provided data files). The README file should include a short description of what is included in the submission and how to run your program.

NOTE: its your responsibility to verify that your code was checked-in to the Github repo correctly and ontime.