Farhat Abbas University
College of Sciences
Department of Computer Sciences

**IR/IRW**
**Information Retrieval**
**Instructor: Dr Fouzi Harrag**

**Homework**
**First Semester 2018/2019**
**Weight: 20 marks**

**Due: Wednesday, December 19th (by 12:30 pm)**

## 1: Vector Space Model (7 marks)

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 the vector-space similarity measures: Cosine similarity. The goal of this question is to understand the use of this measure.
Suppose we have a collection of 8 documents (denoted as D1-D8 below). Answer the following questions. Assume a binary text representation—a vector's value for a particular dimension (i.e., a particular index term) equals 1 if the term appears at least once and 0 otherwise.
_ $D_1$: 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

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

$$Euclidean\ Distance\ (q, d) = \sqrt{\sum_{i=1}^{v}[q_i - d_i]^2}$$

Using the Euclidean distance, what is the score given to each document D1-D8 in response to the query "jack"?

(b) 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,

$$\text{cosine similarity}(q, d) = \frac{\sum_{i=1}^{\mathcal{V}} (q_i \times d_i)}{\sqrt{\sum_{i=1}^{\mathcal{V}} q_i^2} \times \sqrt{\sum_{i=1}^{\mathcal{V}} d_i^2}}$$

Using the cosine similarity, what is the score given to each document D1-D8 in response to the query "jack"?

(c) For this particular query, scoring documents D1-D8 using the Euclidean distance and the cosine similarity would result in equal rankings (HINT: if they're not, you made a mistake). Why?

(d) Give an example of a query for which scoring documents D1-D8 using the Euclidean distance and the cosine similarity would result in different rankings. Explain your choice.

**2: Term Weighting (6 marks)**

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. Answer the following questions.

(a) According to a binary weighting scheme (1 if the term occurs, 0 if it doesn’t), which are the most descriptive terms within a document?

(b) According to the TF (term-frequency) weighting scheme, which are the most descriptive terms within a document?

(c) According to the IDF (inverse-document frequency) weighting scheme, which are the most descriptive terms within a document?

(d) According to the TF.IDF (term-frequency multiplied by inverse document frequency) weighting scheme, which are the most descriptive terms within a document?

(e) Compute the TF.IDF weights for all seven terms in D1. Use D1-D8 to compute corpus statistics such as d ft. Do you notice anything strange? Why does this happen? Is it likely to happen in a more ‘realistic’ document collection?

**3: Document Representation (7 marks)**

Oftentimes, the documents we want to search have some amount of structure. Scholarly articles, for example, usually have a title, a set of authors, an abstract, a main body, a references section, and possibly an appendix. It turns out that weighting some parts of a document (e.g., the title) more heavily than other parts (e.g., the appendix) improves retrieval performance.

The general idea is that a document with many of the query-terms appearing in the title should be scored and rank higher than a document with many of the query-terms appearing in the appendix—the title describes the main content of the document better than the appendix.

Suppose you have a collection of documents with two non-overlapping fields: a TITLE field and a BODY field. And, suppose you have access to an out-of-the-box search engine that performs vector-space-model based retrieval using a binary text representation (1’s and 0’s) and Cosine similarity scoring.

Your goal is to design a solution that weights the TITLE field more than the BODY field. In other words, if you have a query with a single query term (e.g., “jack”), you want a document that has “jack” in the title (and nowhere else) to be scored and ranked higher than a document that has “jack” in the body (and nowhere else). How would you do this? (HINT: there are many right answers. Be creative and have fun!).