Information Retrieval Assignment 1

HARSHITHA BENGALURU RAGHURAM
21235396

Question 1

Given:

D1: Shipment of gold damaged in a fire
D2: Delivery of silver arrived in a silver truck
D3: Shipment of gold arrived in a truck

Q1: gold silver truck

Solution:

Tf-idf is a popularly used weighting mechanism. The tf-idf stands for Term Frequency – Inverse Document Frequency. If a term occurs repeatedly in a document, it does not help in distinguishing the document. This is called Term Frequency. Term Frequency helps to quantify how well a term describes the document.

Inverse Document Frequency is a heuristic method used to remove the words that do not add any uniqueness to the document. The IDF value corresponding to a term can be computed by applying logarithm to the ratio of the total number of documents and the number of documents in which the term appears.

tfij = Number of times the term ti appears in the document dj.

idfi = log (Number of documents that contain the term / Number of documents in which the term i occurs)

According to the given question, the word “gold” arrives once in the data D1. Similarly, the word “silver” is found twice in D2. The table given below summarises the term frequency of each word in D1, D2 and D3.

The weightage for all the terms in a document can be computed using the formula given below

$$w_{i;j} = f_{i;j} \log(N/N_i)$$

where
fi;j is the (possibly normalised) frequency of term ti in document dj
N is the number of documents in the collection
In the Ni is the number of documents that contain term ti.

	shipment	of	gold	damaged	in	a	fire	delivery	silver	arrived	truck
IDF	0.176091	0	0.176091	0.477121	0	0	0.477121	0.477121	0.477121	0.176091	0.176091
TF(D1)	1	1	1	1	1	1	1	0	0	0	0
TF(D2)	0	1	0	0	1	1	0	1	2	1	1
TF(D3)	1	1	1	0	1	1	0	0	0	1	1
TF*IDF(D1)	0.176091	0	0.176091	0.176091	0	0	0.477121	0	0	0	0
TF*IDF(D2)	0	0	0	0	0	0	0	0.477121	0.954	0.176091	0.176091
TF*IDF(D3)	0.176091	0	0.176091	0	0	0	0	0	0	0.176091	0.176091
TF(Q)	0	0	1	0	0	0	0	0	1	0	1
TF*IDF(Q)	0	0	0.176091	0	0	0	0	0	0.477121	0	0.176091

According to the given question, the value of N is supposed to be 3 since there are 3 documents D1, D2 and D3.

idfi *= log (3 / Number of documents in which the term i occurs)*

Wij = tfij * idfi

The given query has 3 terms, then the weightage is computed for all the terms.

The next step is to arrange the documents in the descending order of relevance by using cosine similarity. In the Vector Space Model, if each document is considered as a vector in a vector space then the similarity between a query and the document can be found by measuring the angle between the vectors of the document and the query. The formula given below is used to measure the cosine angle between the vectors.

$$\boldsymbol{sim(q,d)} = \frac{\sum q_i\, d_i}{\sqrt{\sum {q_i}^2}\sqrt{\sum {d_i}^2}}$$

sim(q, d1) = **0.0801**

sim(q, d2) =**0.8248**

sim(q, d3) =**0.3272**
Thus, the ranking for the given three documents are **D2 > D3 > D1**.

Question 2

Given:

D1 = Shipment of gold damaged in a fire.

D1 = Shipment of gold damaged in a fire. Fire. Fire.
D1 = Shipment of gold damaged in a fire. Gold
D1 = Shipment of gold damaged in a fire. Gold. Gold.

Solution:

In the first two samples, the appearance of the word "Fire" keeps incrementing. As a consequence, the term frequency for the term "Fire" also increases.

In the last two samples, the appearance of the word "Gold" keeps incrementing. As a consequence, the term frequency for the term "Gold" also increases.

When there is an increase in the term frequency then there is an increase in the weightage of the term.

But for the first two samples, the measure of similarity using Cosine of the angle between the two vectors (query and sample D1) does not change because the query does not contain the term "Fire".

In the last two samples, the measure of Similarity computed using the cosine angle increases because the query contains the word "Gold".

The relevance of the document increases with respect to the query when there is an increment in the occurrence of the term "Gold".

Question 3

Given:

The document collection consists of all the scientific articles published in the Communications of the ACM (www.acm.org/dl).

When the tf-idf scheme is implemented, then a term can occur any number of times in a document. As a result of which irrelevant documents are extracted when a single word has multiple occurrences and high frequency. The length of the entire document is not accounted in tf-idf. There are algorithms like OkapiBM25 which is used to reduce the points or score of the documents which consist of too many words that do not match with the query. The BM25 algorithm implements both the above-mentioned features which enables retrieval of relevant documents.

BM25F is an updated version of the BM25 algorithm. It is specially used when a document consists of multiple fields like headlines, footnotes etc which may have varying degree of importance, length-normalization and relevance.

BM25+ is an extended version of BM25 algorithm. In BM25 algorithm, long documents which contains the terms that match with the query may be scored in a wrong way when compared with the shorter length documents that don't consist any query word. This is caused because term frequency normalization by document length is not properly lower-bounded.

References:

https://en.wikipedia.org/wiki/Okapi_BM25