Modern Information Management

Assignment -1

Student id : 18231267

Name : Chandana

1. **Given document collection :**

   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
   And the query Q : gold silver truck.

To Calculate the relevant documents order with respect to Q(query) need to follow the below steps:

1. **Pre-processing :**

   First we need to construct a (documents)X(words) matrix with a tf-idf weighting.
   The steps to follow before constructing this matrix :
   1. Stop word removal
   2. Lemmatizing the words to find the root word
   3. Lowering all the words(lower case to avoid the ambiguity)
   4. Removal of invalid characters
   5. Removal of Named Entity Recognition(such as person names, Organization names, Places names)

Representing Document and Query as Vector Space Model :

1. Every document consist of set of terms : $t_1 \ldots t_n$
2. weight $w_{i,j}$ is assigned to each term t_i occurring in document d_j
3. Can view a document or query as a vector of weights: dj = (w_1,w_2,w_3..., w_n)
4. Hence we can represent documents and query as n-dimensional vectors.
   $dj = (w_{1,j}, w_{2,j}, \ldots w_{n,j})$
   $q = (w_{1,q}, w_{2,q} \ldots w_{n,q})$

Weighting Technique(tf-idf) :

Idea of tf-idf :

1. We need a means to calculate the term weights in the document and query representation.
2. A term's frequency within a document quantifies how well a term describes a document. The more frequent a term occurs in a document, the better it is at describing that document and vice-versa. This frequency is known as the term frequency or term factor.
3. Also, if a term occurs frequently across all the documents that term does little to distinguish one document from another. This factor is known as the inverse document frequency (idf-frequency).
4. We use the combination of both the above tf-idf, this is one of the most popular technique.
5. For all terms in a document, the weight assigned can be calculated by:

   $$w_{i,j} = f_{i,j} \times \log(N/ni)$$

where
fi,j is the (possibly normalised) frequency of term ti in document **dj**
N is the number of documents in the collection
ni is the number of documents that contain term **ti** .

Query Weighting :

Initially all terms In the query assigned to the weights 0.5.

Tf-Idf weights : each entry is calculated using the tf-idf

	shipment	gold	damage	fire	delivery	silver	arrive	truck
D1	0.176091	0.176091	0.477121	0.477121	0	0	0	0
D2	0	0	0	0	0.477121	0.954243	0.176091	0.176091
D3	0.176091	0.176091	0	0	0	0	0.176091	0.176091
Q	0	0.5	0	0	0	0.5	0	0.5

Similarity Measures :

We can measure this similarity by measuring the cosine of the angle between the two vectors.
Inner product:

$$sim(\vec{d_j}, \vec{q}) = \frac{d_j \bullet q}{|d_j||q|}$$

$$\text{sim}(\vec{d_j}, \vec{q}) = \frac{\sum_{i=1}^{n} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{n} w_{i,q}^2}}$$

By using the above information we calculated the Document-term Matrix :

Similarity(D1,Q) = **0.141353**
Similarity(D2,Q) = **0.595679**
Similarity(D3,Q) = **0.57735**
From the similarities between documents and query :
Relevancy Order of Documents with respect to the Query is : **D2,D3,D1**

Solution 2

Given new collection of documents:

a) D1 = Shipment of gold damaged in a fire. Fire.
b) D1 = Shipment of gold damaged in a fire. Fire. Fire.
c) D1 = Shipment of gold damaged in a fire. Gold.
d) D1 = Shipment of gold damaged in a fire. Gold. Gold.

Tf-idf weights according to the new document set :

	shipment	gold	damage	Fire	delivery	silver	arrive	truck
a	0.176091	0.176091	0.477121	0.954243	0	0	0	0
b	0.176091	0.176091	0.477121	1.431364	0	0	0	0
c	0.176091	0.352183	0.477121	0.477121	0	0	0	0
d	0.176091	0.528274	0.477121	0.477121	0	0	0	0
Q	0	0.5	0	0	0	0.5	0	0.5

By using the above information we calculated the Document-term Matrix :

Similarity($D1_a$,Q) = **0.0927**
Similarity($D1_b$,Q) = **0.0664**
Similarity($D1_c$,Q) = **0.26027**
Similarity($D1_d$,Q) = **0.348628**

From the similarities between documents and query :
Relevancy Order of Documents with respect to the Query is : $D1_d$,$D1_c$,$D1_a$,$D1_b$

We can observe that $D1_a$, $D1_b$ similarities are reduced due to query non-relevant term additions in the document. Similarly $D1_c$, $D1_d$ similarities are increased due to query non-relevant term additions in the document.

Solution 3 :
Given user feedback that document 3 is relevant; show how the query could be modified using the Rocchio method. State any assumptions you make

Relevance Feedback for Vector Space Model :
The idea of relevance feedback is to involve the user in the information retrieval process so as to improve the final result set.
In particular, the user gives feedback on

the relevance of documents in an initial set of results. The basic procedure is:
• The user issues a (short, simple) query.
• The system returns an initial set of retrieval results.
• The user marks some returned documents as relevant or nonrelevant.
• The system computes a better representation of the information need based on the user feedback.
• The system displays a revised set of retrieval results.

$$\vec{q} = \alpha\vec{q} + \frac{\beta}{|D_r|}\sum_{d_j \in D_r} d_j - \frac{\gamma}{|D_n|}\sum_{d_j \in D_n} d_j$$

By using the above formula our modified query q_m = (shipment,gold,silver,arrive,truck) with new weights listed in the below table.

	shipment	gold	damage	fire	delivery	silver	arrive	truck
d1	0.176091	0.176091	0.477121	0.477121	0	0	0	0
d2	0	0	0	0	0.477121	0.954243	0.176091	0.176091
d3	0.176091	0.176091	0	0	0	0	0.176091	0.176091
q	0	0.5	0	0	0	0.5	0	0.5

Observe the below table for modified query weights:

	shipment	gold	damage	fire	delivery	silver	arrive	truck
D_R	0.132068	0.132068	0	0	0	0	0.132068	0.132068
q	0	0.5	0	0	0	0.5	0	0.5
q_new	**0.132068**	**0.632068**	**0**	**0**	**0**	**0.5**	**0.132068**	**0.632068**

D_r: Relavant Document set ={D3}

q : Old query

q_new : modified query using Rocchio

By using the above information we calculated the Similarities :

Similarity(D1,Q) = **0.1797**
Similarity(D2,Q) = **0.5362**
Similarity(D3,Q) = **0.7339**

Document relevancy order : D3,D2,D1

Observation : Using Rocchio Feedback algorithm, we have given more weights to the document3 with respect to querry.

Assumptions in calculation the Rocchio :

1.Considered the Non relevant document set empty
2.Considered $\alpha = 1$, $\beta = 0.5$