Information Retrieval (TDT4117)

Assignment 1

Submitted by

Name : Md Anwarul Hasan
Student ID : 583233

Task 1

A) Answer:

The Boolean model of information retrieval has some limitations and considering those Web Search Engines not employed the model. Those are,

i) The term document matrix that is a sparse matrix, which requires huge amount of memory to store. The matrix stores the value “0” in approximately its 982% space and which has no impact in this model.
ii) The model does not make any ranking of the document, as a result too many or too few documents could be retrieved,
iii) The model only does exact matching. It does not work with partial matching.

Some limitations of Vector Space Model (VSM) for which Web Search Engines not employed the model are,

i) Vector space model is works with keywork query, rather than phrase query. For example if anyone search for a song, i.e which has an order of words, in such case VSM is not good.
ii) This model will not incorporate the documents that used different vocabulary for the same topic.
iii) Keywords need to match in exact format, otherwise a lot of non-related matching will hamper the final result.

B) Answer:

C) Answer:

With respect to VSM, document vectors are normalized to unit length through a procedure call document length normalization. If each term of the document is in the t-dimensional space (here t is the total number of terms in the particular document) and thus each document can be represent as a vector of weighted terms. The term k_i of document d_j generated the vector $w_{i,j}$ x $\overrightarrow{k_i}$ which is the contribution of the term in the document. So the vector $\overrightarrow{d_j}$ is the composed value of all its term vector components and thus we can computer it as ,

$$|\vec{d_j}| = \sqrt{\sum_{i}^{t} w_{i,j}^2}$$

Here $w_{i,j}$ is the weight of the term of the document for pair (k_i, d_j). If TF-IDF weights is calculated then we can easily compute this.

Document vector normalization is important because, larger of big documents may contain more terms than small documents and regardless of their relevancy the big sized documents are likely to retrieve simply because they had more terms than small documents. To eliminate this document vectors are normalized to unit length.

Task 2

Sub task 1

Here,

So, the presence of the term in the corresponding document, (terms are in columns and documents are in rows and Boolean 1 and 0 represent presence and absence of the term in the corresponding document)

	Cloudy	Sunny	Rainy
Doc1	1	1	1
Doc2	1	0	1
Doc3	1	1	0
Doc4	1	1	1
Doc5	0	1	0
Doc6	0	0	1
Doc7	1	1	1
Doc8	1	0	1
Doc9	0	1	1
Doc10	1	1	1

So term vector for Cloudy, Sunny and Rainy is, (representing documents in columns and terms in rows)

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7	Doc8	Doc9	Doc10
Cloudy	1	1	1	1	0	0	1	1	0	1
Sunny	1	0	1	1	1	0	1	0	1	1
Rainy	1	1	0	1	0	1	1	1	1	1

1) Answer:

For q1 = “Rainy and Cloudy”

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7	Doc8	Doc9	Doc10
Cloudy	1	1	1	1	0	0	1	1	0	1
Rainy	1	1	0	1	0	1	1	1	1	1
And (operator)	1	1	0	1	0	0	1	1	0	1

so returned documents are those that has value 1 in “and”-operator row.

For q2 = “Cloudy and Sunny”

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7	Doc8	Doc9	Doc10
Cloudy	1	1	1	1	0	0	1	1	0	1
Sunny	1	0	1	1	1	0	1	0	1	1
And (operator)	1	0	1	1	0	0	1	0	0	1

so returned documents are those that has value 1 in “and”-operator row.

For q3 = "Sunny OR Rainy"

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7	Doc8	Doc9	Doc10
Sunny	1	0	1	1	1	0	1	0	1	1
Rainy	1	1	0	1	0	1	1	1	1	1
Or (operator)	1	1	1	1	1	1	1	1	1	1

so returned documents are those that has value 1 in “and”-operator row.

For q4 = "Cloudy NOT Rainy"

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7	Doc8	Doc9	Doc10
Rainy	1	1	0	1	0	1	1	1	1	1
Not Rainy	0	0	1	0	1	0	0	0	0	0

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7	Doc8	Doc9	Doc10
Cloudy	1	1	1	1	0	0	1	1	0	1
Not Rainy	0	0	1	0	1	0	0	0	0	0

And (operator)	0	0	1	0	0	0	0	0	0	0

so returned documents are those that has value 1 in “and”-operator row.

For q5 = "Sunny"

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7	Doc8	Doc9	Doc10
Sunny	1	0	1	1	1	0	1	0	1	1

so returned documents are those that has value 1 in “Sunny” row.

2) Answer:

The dimension of the vector space representing this document collection is

	Cloudy	Sunny	Rainy
Doc1	1	1	2
Doc2	1	0	1
Doc3	1	2	0
Doc4	4	2	2
Doc5	0	1	0
Doc6	0	0	2
Doc7	1	1	1
Doc8	1	0	2
Doc9	0	1	2
Doc10	1	2	1

This is achieved by counting the number of times a term presence in that particular document.

3) Answer:

If we assume that $W_{i,j}$ is the weight of a term in the pair (k_i, d_j), where k_i is the term and d_j is the document then we can apply the TF-IDF weighting scheme,

$$w_{i,j} = \begin{cases} (1 + \log f_{i,j}) \times \log \frac{N}{n_i} & \text{if } f_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Here,

$f_{i,j}$ = is the frequency of the term "I" in the document "j"

N = is the total number of the documents = 4

n_i = is the document frequency, i.e. number of the document containing the i term

so by applying this to the matrix we get from the answer no 2,

	Cloudy	Sunny	Rainy
Doc1	0.514573	0.514573	0.643856
Doc2	0.514573	0	0.321928
Doc3	0.514573	1.029146	0
Doc4	1.54372	1.029146	0.643856
Doc5	0	0.514573	0
Doc6	0	0	0.643856
Doc7	0.514573	0.514573	0.321928
Doc8	0.514573	0	0.643856
Doc9	0	0.514573	0.643856
Doc10	0.514573	1.029146	0.321928

4) Answer:

We have found the document term matrix from the answer 3.

Using this we can easily calculate the similarity of document 5 to other four documents, i.e. document 1, 2, 4 and 10

So, distance between Document 1 and document 5 is,

Distance (Doc1, Doc2) = SQR $((0.514573-0)^2 + (0.514573-0.514573)^2 + (0.643856-0)^2)$

= SQR(0.264786+0+0.414551)

= 0.824219

So the similarity between Document 1 and document 5 is 0.824219

Again the distance between Document 2 and document 5 is,

Distance (Doc2, Doc5) = SQR $((0.514573-0)^2 + (0-0.514573)^2 + (0.321928095-0)^2)$

= SQR (0.264786 + 0.264786 + 0.103638)

= 0.795744

So the similarity between Document 2 and document 5 is 0.795744

Again the distance between Document 4 and document 5 is,

Distance (Doc4, Doc5) = SQR $((1.54372-0)^2 + (1.029146- 0.514573)^2 + (0.643856-0)^2)$

= SQR (2.38307 + 0.264786 + 0.414551)

= 1.749973

So the similarity between Document 4 and document 5 is 1.749973

Again the distance between Document 5 and document 10 is,

Distance (Doc4, Doc5) = SQR $((0-0.514573)^2 + (0.514573-1.029146)^2 + (0-0.321928)^2)$

= SQR(0.264786 + 0.264786 + 0.103638)

= 0.795744

So the similarity between Document 10 and document 5 is 0.795744.

5) Answer:

Formula of cosine similarity is given below,

$$\cos(\theta) = \frac{\sum_{j=1}^{|\mathcal{V}|} m_{i,j} \cdot m_{q,j}}{\sqrt{\sum_{j=1}^{|\mathcal{V}|} m_{i,j}^2} \cdot \sqrt{\sum_{i=1}^{|\mathcal{V}|} m_{i,q}^2}}$$

Eliminating $\sqrt{\sum_{j=1}^{|\mathcal{V}|} m_{i,j}^2}$ and $\sqrt{\sum_{i=1}^{|\mathcal{V}|} m_{i,q}^2}$ from the formula (as they are common for all) we get,

Cos (θ) = $\sum_{j=1}^{|\mathcal{V}|} m_{i,j} \cdot m_{q,j}$

SubTask 2:

The main difference between BM25 and probabilistic model are,

i. BM25 model does not require to initially divide the documents into relevant or non-relevant sets by guessing
ii. Probabilistic model can not answer phrase query because the indexes are occurring independently as a result it can only answer keywork query but BM25 model overcomes that
iii. To build good ranking function term frequency and document length normalization has significant role. These two component is present in BM25 model but probabilistic model did not use those.
iv. To be computed the probabilistic model require relevance information but BM25 model does not require relevance information to calculate.