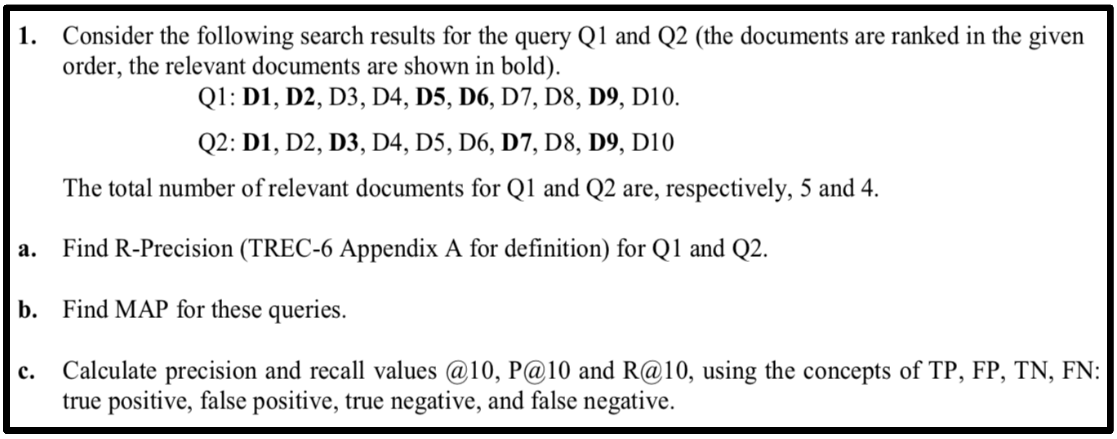
Duygu Durmuş

Computer Engineering Department
Bilkent University
CS533 - Information Retrieval Systems



1. Consider the following search results for the query Q1 and Q2 (the documents are ranked in the given order, the relevant documents are shown in bold).

Q1: **D1**, **D2**, D3, D4, **D5**, **D6**, D7, D8, **D9**, D10.

Q2: **D1**, D2, **D3**, D4, D5, D6, **D7**, D8, **D9**, D10

The total number of relevant documents for Q1 and Q2 are, respectively, 5 and 4.

a. Find R-Precision (TREC-6 Appendix A for definition) for Q1 and Q2.

b. Find MAP for these queries.

c. Calculate precision and recall values @10, P@10 and R@10, using the concepts of TP, FP, TN, FN: true positive, false positive, true negative, and false negative.

Solution for question 1:

Q1: **D1**, **D2**, D3, D4, **D5**, **D6**, D7, D8, **D9**, D10 The total number of relevant documents: 5
Q2: **D1**, D2, **D3**, D4, D5, D6, **D7**, D8, **D9**, D10 The total number of relevant documents: 4

a. R-Precision

R-precision is the precision after R-documents have been retrieved, where R is the number of relevant documents for the topic [1]. There are 5 and 4 relevant documents for Q1 and Q2, respectively. Therefore,

R-Precision(Q1) = 3/5 = 0.6
R-Precision(Q2) = 2/4 = 0.5

b. Mean Average Precision (MAP)

$$\boldsymbol{Recall} = \frac{\boldsymbol{number\ of\ relevant\ items\ retrieved}}{\boldsymbol{number\ of\ relevant\ items\ in\ collection}}\ [1]$$

$$\boldsymbol{Precision} = \frac{\boldsymbol{number\ of\ relevant\ items\ retrieved}}{\boldsymbol{total\ number\ of\ items\ retrieved}}\ [1]$$

Document Number	1	2	3	4	5	6	7	8	9	10
Relevance	✓	✓	✗	✗	✓	✓	✗	✗	✓	✗
Precision	1/1	2/2	2/3	2/4	3/5	4/6	4/7	4/8	5/9	5/10
Recall	1/5	2/5	2/5	2/5	3/5	4/5	4/5	4/5	5/5	5/5

Table 1: The table showing the relevance, precision and recall values at 10 for Q1

Document Number	1	2	3	4	5	6	7	8	9	10
Relevance	✓	✗	✓	✗	✗	✗	✓	✗	✓	✗
Precision	1/1	1/2	2/3	2/4	2/5	2/6	3/7	3/8	4/9	4/10
Recall	1/4	1/4	2/4	2/4	2/4	2/4	3/4	3/4	4/4	4/4

Table 2: The table showing the relevance, precision and recall values at 10 for Q2

$$\textbf{Average Precision(AveP)} = \frac{\sum_{k=1}^{n} p(k) * rel(k)}{number\ of\ relevant\ documents}$$

where p(k) is precision at position k and rel(i) is the relevancy of the kth document that is 1 for ✓(relevant) and 0 for ✗(not relevant) [2].

Average Precision(Q1) = (1 + 1 + 3/5 +4/6 +5/9)/5 = 0.764
Average Precision (Q2) = (1 + 2/3 + 3/7 + 4/9)/4 = 0.634

$$\textbf{Mean Average Precision (MAP)} = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$$

where Q is the number of queries [2].

MAP = (0.764 + 0.634) /2 = 0.699

c. Precision and Recall @ 10 by using TP, FP, TN and FN

$$Precision = \frac{True\ Positives}{True\ Postives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

where *the number of true positives* is the number of correctly labeled items belonging to the relevant class, *the number of false positives* is the number of incorrectly labeled items belonging to the non-relevant class and *the number of false negatives* is the number of incorrectly labeled items but belonging to the relevant class [3] .

Therefore, true positives are retrieved and relevant, false positives are retrieved but not relevant and false negatives are not retrieved but relevant. In our case, all documents(10) for both queries are retrieved so the number of false negatives are 0.

For Q1:

The number of false negatives = 0
The number of true positives = 5
The number of false positives = 5

Precision@10 = 5/(5+5) = 5/10 = 1/2 = 0.5
Recall@10 = 5/(5+0) = 1

For Q2:

The number of false negatives = 0
The number of true positives = 4
The number of false positives = 6

Precision@10 = 4/(4+6) = 4/10 = 2/5 = 0.4
Recall@10 = 4/(4+0) = 1

2. For the queries given above draw the recall precision graph using the TREC interpolated approach (See TREC 6 Appendix A). Explain the purpose of interpolation. Find related articles that may explain it and provide it/their citations.

Solution for question 2:

Precision	1	1	1	1	1	0.66	0.66	0.66	0.66	0.55	055
Recall	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

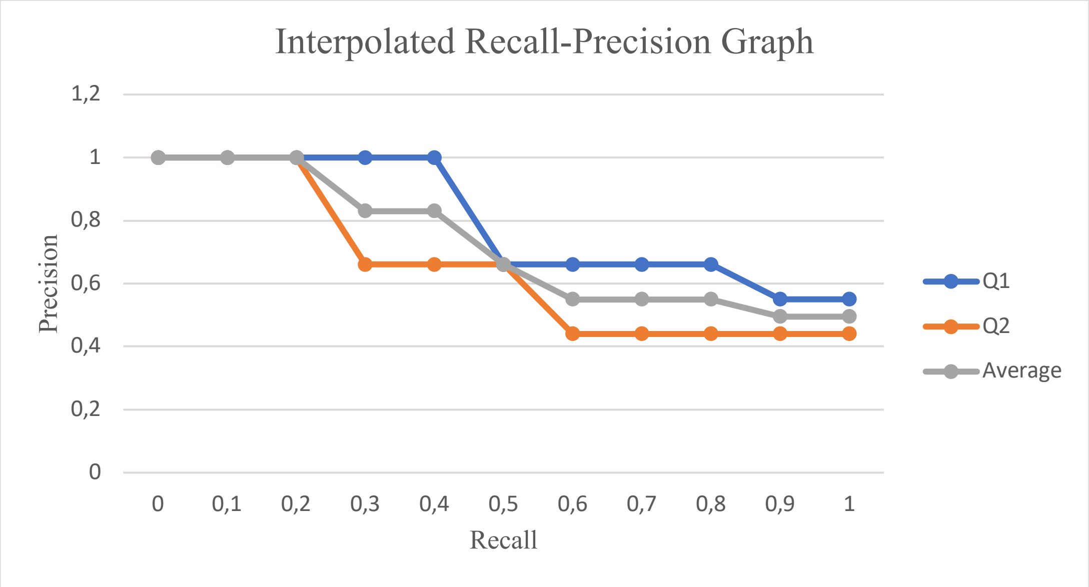
Table 3: The table showing the interpolated precision and recall for Q1

Precision	1	1	1	0.66	0.66	0.66	0.44	0.44	0.44	0.44	0.44
Recall	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

Table 4: The table showing the interpolated precision and recall for Q2

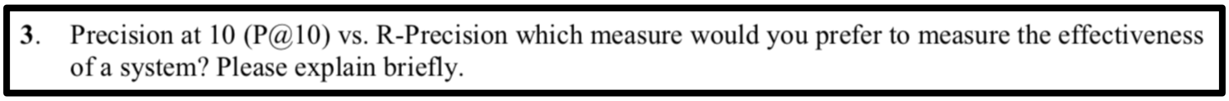
Precision	1	1	1	0.83	0.83	0.66	0.55	0.55	0.55	0.50	0.50
Recall	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

Table 5: The average of interpolated precision and recall for both queries



To obtain the precision values at all of the standard recall levels from 0.0 to 1.0 in increments of 0.1, the recall-precision data points is required to be interpolated. The rule of interpolation is for each standard recall level i from 0.0 to 1.0, use the maximum precision obtained for any actual recall level greater than or equal to i [1]. Therefore, this method of interpolation defines the precision at any recall level as the maximum precision observed in any recall-precision point at a higher recall level. It turns precision-recall graph into a step-function with the jumps at the observed point [4].

Since search engines are not perfect and they always retrieve some non-relevant documents, precision tends to decrease with increasing recall. This interpolation produces a step-function that is monotonically decreasing which means that precision values always go down or stay the same with the increasing recall [5]. Based on Croft, Metzler and Strohman's book, the general purpose of interpolation is that the recall-precision values are defined by the sets of documents in the ranking with the best possible precision values [5]. Rijsbergen also confirms this idea by saying that the linear interpolation estimates the *best* possible performance between any two adjacent observed points [4]. Thus, the rationale for interpolation is that the user wants to look at more stuff if both precision and recall get better [6].



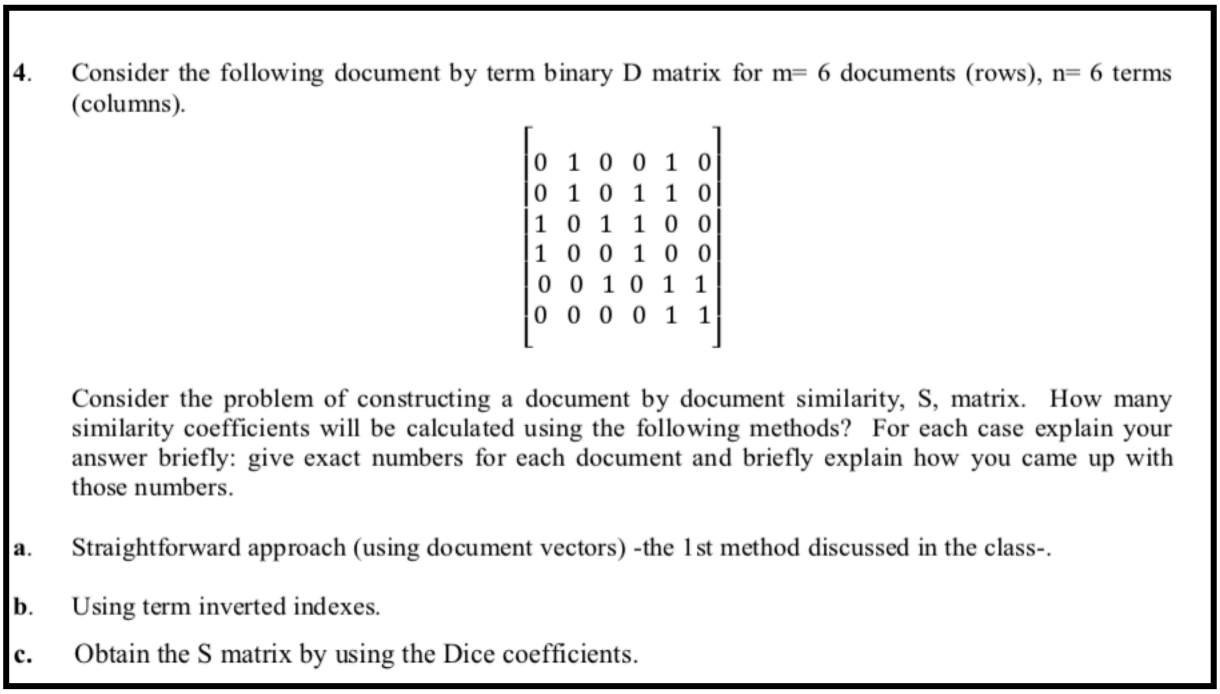
3. Precision at 10 (P@10) vs. R-Precision which measure would you prefer to measure the effectiveness of a system? Please explain briefly.

As discussed in the class, effectiveness and efficiency are important for performance evaluation of a system. Effectiveness is measured with the precision and recall. Precision at 10 and R-precision are two different precision measures for information retrieval evaluation methods.

Precision at 10 shows the number of relevant documents/total number of documents retrieved at rank 10 while R-precision is the precision after R-documents have been retrieved, where R is the number of relevant documents.

For example, in the question 1, R-Precision(Q1) = 3/5 = 0.6 and Precision at 10=0.5. As it could be seen from example, Precision@10 considers 10 documents for determining precision which is a low precision value. R-precision takes all relevant documents into account and in the first 5 relevant documents(R=5), most of the relevant documents(3) are retrieved so it gives a better precision value.

It means that R-precision is an evaluation method which enables evaluation of entire system by considering the total number of relevant documents while Precision@10 takes the documents until 10 into consideration. The value of 10 could change so measuring the effectiveness of the whole system might not possible. However, since R-Precision considers all relevant documents in the system, it gives a better measuring result. Thus, selecting the measure depends on the situation but in the case of measuring the effectiveness of whole system, I would prefer R-Precision.



4. Consider the following document by term binary D matrix for m= 6 documents (rows), n= 6 terms (columns).

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Consider the problem of constructing a document by document similarity, S, matrix. How many similarity coefficients will be calculated using the following methods? For each case explain your answer briefly: give exact numbers for each document and briefly explain how you came up with those numbers.

a. Straightforward approach (using document vectors) -the 1st method discussed in the class-.

b. Using term inverted indexes.

c. Obtain the S matrix by using the Dice coefficients.

Solution of question 4:

a. Straightforward approach

$$\boldsymbol{D} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix}$$

The straightforward approach constructs the similarity matrix S by calculating the similarity between every pair of the documents. Since similarity matrix is symmetric in other words Sij = Sji and Sii means the same document, only the given pairs below are calculated:

$$\boldsymbol{S} = \begin{bmatrix} \mathbf{1} & \boldsymbol{S12} & \boldsymbol{S13} & \boldsymbol{S14} & \boldsymbol{S15} & \boldsymbol{S16} \\ X & \mathbf{1} & \boldsymbol{S23} & \boldsymbol{S24} & \boldsymbol{S25} & \boldsymbol{S26} \\ X & X & \mathbf{1} & \boldsymbol{S34} & \boldsymbol{S35} & \boldsymbol{S36} \\ X & X & X & \mathbf{1} & \boldsymbol{S45} & \boldsymbol{S46} \\ X & X & X & X & \mathbf{1} & \boldsymbol{S56} \\ X & X & X & X & X & \mathbf{1} \end{bmatrix}$$

Syntax: S_{ij} means the similarity between the documents i and j

Since the similarity coefficients of S_{12}, S_{13}, S_{14}, S_{15}, S_{16}, S_{23}, S_{24}, S_{25}, S_{26}, S_{34}, S_{35}, S_{36}, S_{45}, S_{46} and S_{56} are required to be calculated, in total **15 similarity coefficients** will be calculated. In other words, $\frac{m*(m-1)}{2} = \frac{6*5}{2}$= 15 where m is the number of documents.

b. Using term inverted indexes

$$\boldsymbol{D} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix}$$

2 documents can have similarity value > 0 if they have one or more common terms.

Syntax: $t_i \rightarrow d_j$ means term i is found in the document j

$t_1 \rightarrow d_3, d_4$
$t_2 \rightarrow d_1, d_2$
$t_3 \rightarrow d_3, d_5$
$t_4 \rightarrow d_2, d_3, d_4$
$t_5 \rightarrow d_1, d_2, d_5, d_6$
$t_6 \rightarrow d_5, d_6$

Consider d_1:

d_1 contains terms t_2 and t_5.

$t_2 \rightarrow d_1, d_2$
$t_5 \rightarrow d_1, d_2, d_5, d_6$
$t_2 \cup t_5 \rightarrow d_1, d_2, d_5, d_6$ Calculate S_{12}, S_{15}, S_{16}

Consider d_2:

d_2 contains terms t_2, t_4 and t_5.

$t_2 \rightarrow d_1, d_2$
$t_4 \rightarrow d_2, d_3, d_4$
$t_5 \rightarrow d_1, d_2, d_5, d_6$
$t_2 \cup t_4 \cup t_5 \rightarrow d_1, d_2, d_3, d_4, d_5, d_6$ Calculate S_{23}, S_{24}, S_{25}, S_{26}

Consider d_3:

d_3 contains terms t_1, t_3 and t_4.

$t_1 \rightarrow d_3, d_4$
$t_3 \rightarrow d_3, d_5$
$t_4 \rightarrow d_2, d_3, d_4$
$t_1 \cup t_3 \cup t_4 \rightarrow d_2, d_3, d_4, d_5$ Calculate S_{34}, S_{35}

Consider d_4:

d_4 contains terms t_1 and t_4.

$t_1 \rightarrow d_3, d_4$
$t_4 \rightarrow d_2, d_3, d_4$
$t_1 \cup t_4 \rightarrow d_2, d_3, d_4$ –

Consider d_5:

d_5 contains terms t_3, t_5 and t_6.

$t_3 \rightarrow d_3, d_5$
$t_5 \rightarrow d_1, d_2, d_5, d_6$
$t_3 \cup t_5 \cup t_6 \rightarrow d_1, d_2, d_3, d_5, d_6$ Calculate S_{56}

Consider d_6: No calculation needed

Since the similarity coefficients of S_{12}, S_{15}, S_{16}, S_{23}, S_{24}, S_{25}, S_{26}, S_{34}, S_{35} and S_{56} are required to be calculated, in total **10 similarity coefficients** will be calculated by using term inverted indexes.

c. S-matrix by using the Dice coefficients

$$\boldsymbol{D} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix}$$

$d1 = [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0] \qquad |d1| = 2$
$d2 = [0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0] \qquad |d2| = 3$
$d3 = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0] \qquad |d3| = 3$
$d4 = [1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0] \qquad |d4| = 2$
$d5 = [0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1] \qquad |d5| = 3$
$d6 = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1] \qquad |d6| = 2$

Calculate given similarity coefficients using the Dice coefficient given below:

$$\boldsymbol{Dice\ Coefficient} = \frac{\mathbf{2}|\boldsymbol{X} \cap \boldsymbol{Y}|}{|\boldsymbol{X}| + |\boldsymbol{Y}|} \text{ [7]}$$

$$\boldsymbol{S} = \begin{bmatrix} \mathbf{1} & \boldsymbol{S12} & \boldsymbol{S13} & \boldsymbol{S14} & \boldsymbol{S15} & \boldsymbol{S16} \\ X & \mathbf{1} & \boldsymbol{S23} & \boldsymbol{S24} & \boldsymbol{S25} & \boldsymbol{S26} \\ X & X & \mathbf{1} & \boldsymbol{S34} & \boldsymbol{S35} & \boldsymbol{S36} \\ X & X & X & \mathbf{1} & \boldsymbol{S45} & \boldsymbol{S46} \\ X & X & X & X & \mathbf{1} & \boldsymbol{S56} \\ X & X & X & X & X & \mathbf{1} \end{bmatrix}$$

From previous part, we know similarity coefficients given below which requires calculation: S_{12}, S_{15}, S_{16}, S_{23}, S_{24}, S_{25}, S_{26}, S_{34}, S_{35} and S_{56}.

Other similarity coefficients will be 0 and they are not calculated below since these documents do not have any common terms. For instance,

For S_{46}:

$d4 = [1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0] \qquad |d4| = 2$
$d6 = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1] \qquad |d6| = 2$

$$Dice\ Coefficient = \frac{2 * (0 + 0 + 0 + 0 + 0 + 0)}{2 + 2} = \frac{2 * 0}{2 + 2} = \frac{0}{4} = 0$$

Calculation of similarity coefficients using Dice coefficients:

For S_{12}:

$d1 = [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0] \qquad |d1| = 2$
$d2 = [0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0] \qquad |d2| = 3$

$$Dice\ Coefficient = \frac{2 * (0 + 1 + 0 + 0 + 1 + 0)}{2 + 3} = \frac{2 * 2}{2 + 3} = \frac{4}{5} = 0.8$$

For S_{15}:

$d1 = [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0] \qquad |d1| = 2$
$d5 = [0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1] \qquad |d5| = 3$

$$Dice\ Coefficient = \frac{2 * (0 + 0 + 0 + 0 + 1 + 0)}{2 + 3} = \frac{2 * 1}{2 + 3} = \frac{2}{5} = 0.4$$

For S_{16}:

$d1 = [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0] \qquad |d1| = 2$
$d6 = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1] \qquad |d6| = 2$

$$Dice\ Coefficient = \frac{2 * (0 + 0 + 0 + 0 + 1 + 0)}{2 + 2} = \frac{2 * 1}{2 + 2} = \frac{2}{4} = 0.5$$

For S$_{23}$:

$d2 = [0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0] \qquad |d2| = 3$

$d3 = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0] \qquad |d3| = 3$

$$Dice\ Coefficient = \frac{2*(0+0+0+1+0+0)}{3+3} = \frac{2*1}{3+3} = \frac{2}{6} = 0.33$$

For S$_{24}$:

$d2 = [0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0] \qquad |d2| = 3$

$d4 = [1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0] \qquad |d4| = 2$

$$Dice\ Coefficient = \frac{2*(0+0+0+1+0+0)}{3+2} = \frac{2*1}{3+2} = \frac{2}{5} = 0.4$$

For S$_{25}$:

$d2 = [0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0] \qquad |d2| = 3$

$d5 = [0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1] \qquad |d5| = 3$

$$Dice\ Coefficient = \frac{2*(0+0+0+0+1+0)}{3+3} = \frac{2*1}{3+3} = \frac{2}{6} = 0.33$$

For S$_{26}$:

$d2 = [0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0] \qquad |d2| = 3$

$d6 = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1] \qquad |d6| = 2$

$$Dice\ Coefficient = \frac{2*(0+0+0+0+1+0)}{3+2} = \frac{2*1}{3+2} = \frac{2}{5} = 0.4$$

For S$_{34}$:

$d3 = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0] \qquad |d3| = 3$

$d4 = [1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0] \qquad |d4| = 2$

$$Dice\ Coefficient = \frac{2*(1+0+0+1+0+0)}{3+2} = \frac{2*2}{3+2} = \frac{4}{5} = 0.8$$

For S$_{35}$:

$d3 = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0] \qquad |d3| = 3$

$d5 = [0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1] \qquad |d5| = 3$

$$Dice\ Coefficient = \frac{2*(0+0+1+0+0+0)}{3+3} = \frac{2*1}{3+3} = \frac{2}{6} = 0.33$$

For S56:

$d5 = [0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1] \qquad |d5| = 3$

$d6 = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1] \qquad |d6| = 2$

$$Dice\ Coefficient = \frac{2*(0+0+0+0+1+1)}{3+2} = \frac{2*2}{3+2} = \frac{4}{5} = 0.8$$

The resulting S-matrix using Dice coefficient is given below:

$$S = \begin{bmatrix} 1 & 0.8 & 0 & 0 & 0.4 & 0.5 \\ X & 1 & 0.33 & 0.4 & 0.33 & 0.4 \\ X & X & 1 & 0.8 & 0.33 & 0 \\ X & X & X & 1 & 0 & 0 \\ X & X & X & X & 1 & 0.8 \\ X & X & X & X & X & 1 \end{bmatrix}$$

5. In this part consider the paper J. Zobel, A. Moffat, "Inverted files for text search engines." *ACM Computing Surveys*, Vol. 38, No. 2, 2006.

a. Assume that we have the following posting list for
term-a: <1, 2> <3, 2> <9, 2> <10, 3> <12, 4> <18, 4> <20, 3>, <23, 3> <25, 4> <33, 4> <40, 5> <43, 4> <55, 3><64, 2> <68, 4> <72, 3> <75, 1> <88, 2>.
The posting list indicates that term-a appears in d1 twice and in d10 three times, etc.

Assume that we have the following posting list for
term-b: <12, 7> <66, 3> <75, 1>.

Consider the following conjunctive Boolean query: term-a **and** term-b. If no skipping is used how many comparisons do you have to find the intersection of these two lists?

b. Introduce a skip structure such data we have a pointer to next 5th entry (it will also have the lowest document number of the following data chunk) , for example for term-a at the beginning of the inverted index we have a pointer that indicates that the next document number at the beginning of the next data chunk as 18. Similarly at the beginning of 18 there will be a skip pointer that indicates that next data chunk starts with document number 40 and we will also have a pointer to that data chunk. Draw the corresponding figure then give the number of comparisons involved to process the same query using this skipping structure.

c. State the advantages and disadvantages of large and small skips in the posting lists. Please give it in a tabular form. Note that in the paper it is assumed that compression will be used. The skip idea is applicable in an uncompressed environment too.

d. Can we take advantage of the skipping structure for disjunctive queries? Please explain.

Solution of question 5:

a. Number of comparisons of two lists with no usage of skipping

term-a → < 1, 2 > < 3, 2 > < 9, 2 > < 10, 3 > < 12, 4 > < 18, 4 > < 20, 3 > < 23, 3 > < 25, 4 > < 33, 4 > < 40, 5 > < 43, 4 > < 55, 3 > < 64, 2 > < 68 ,4 > < 72, 3 > < 75, 1 > < 88, 2 >

term-b → < 12, 7 > < 66, 3 > < 75, 1 >

Boolean query: term-a **and** term b
As the document numbers in the posting list is in increasing order, only one scan is enough for intersection.

Compare < 12, 7 > of term-b's posting list with < 1, 2 >, < 3, 2 >, < 9, 2 >, < 10, 3 > and < 12, 4 > from term-a's posting list. After **5 comparisons**, we know where to place < 12, 7>. Increment indexes of both term-a and term-b.

Compare < 66, 3 > of term-b's posting list with < 18, 4 >, < 20, 3 >, < 23, 3 >, < 25, 4 >, < 33, 4 >, < 40, 5 >, < 43, 4 >, < 55, 3 >, < 64, 2 > and < 68 ,4 > from term-a's posting list. After **10 comparisons**, we know where to place < 66, 3 >. Increment indexes of both term-a and term-b.

Compare < 75, 1 > of term-b's posting list with < 72, 3 >, < 75, 1 > and < 88, 2 > from term-a's posting list. After **3 comparisons**, we know where to place < 75, 1 >. Increment index of both term-a and term-b.

Therefore, we can find intersection of term-a and term-b lists without using skipping in **18 (5+10+3)** comparisons.

b. Introducing a skipping structure

I introduce a skipping structure with **chunk size=5**. Posting lists of term-a is splitted into chunks in total of 4.

The skipping structure works as follows:
1)Split the posting list of term-a into chunk size of 5.
2)Select highest number document in the chunk as the descriptor of the chunk.
3) chunk-comparison=0 (comparing chunk descriptor with posting list element of term-b)
element-comparison=0 (comparing chunk element of term-a with posting list element of term-b)

```
Compare each posting list element of term-b with the chunk i descriptor of term-a
    If chunk i descriptor <= posting list element of term-b
        chunk-comparison++
        insert into chunk i
        If chunk i element of term-a <= posting list element of term-b
            element-comparison++
            move to next posting list element of term-b
        else
            element-comparison++
            go to next element of chunk i of term-a
    else
        chunk-comparison++
        go to next chunk i+1
```

Total comparison = element-comparison + chunk-comparison

1)
term-a:
Chunk 1: < 1, 2 > < 3, 2 > < 9, 2 > < 10, 3 > < 12, 4 >
Chunk 2: < 18, 4 > < 20, 3 > < 23, 3 > < 25, 4 > < 33, 4 >
Chunk 3: < 40, 5 > < 43, 4 > < 55, 3 > < 64, 2 > < 68 ,4 >
Chunk 4: < 72, 3 > < 75, 1 > < 88, 2 >

term-b: < 12, 7 > < 66, 3 > < 75, 1 >

2 and 3)
term-a:
Chunk 1: < 1, 2 > < 3, 2 > < 9, 2 > < 10, 3 > < 12, 4 >   Chunk 1 Descriptor: < 12, 4 >

Chunk 2: < 18, 4 > < 20, 3 > < 23, 3 > < 25, 4 > < 33, 4 >	Chunk 2 Descriptor: < 33, 4 >
Chunk 3: < 40, 5 > < 43, 4 > < 55, 3 > < 64, 2 > < 68 ,4 >	Chunk 3 Descriptor: < 68 ,4 >
Chunk 4: < 72, 3 > < 75, 1 > < 88, 2 >	Chunk 4 Descriptor: < 88, 2 >

term-b:< 12, 7 > < 66, 3 > < 75, 1 >

3)
chunk-comparison = 0
element-comparison = 0
Compare < 12, 7 > with Chunk 1 Descriptor < 12, 4 >:
12 <= 12? YES, chunk-comparison = 1
Insert < 12, 7 > into Chunk 1
12 <= 1? NO, element-comparison = 1
12 <= 3? NO, element-comparison = 2
12 <= 9? NO, element-comparison = 3
12 <= 10? NO, element-comparison = 4
12 <= 12? YES, element-comparison = 5
Go to next element of term-b
Compare < 66, 3 > with Chunk 1 Descriptor < 12, 4 >:
66 <= 12? NO, chunk-comparison = 2
Go to Chunk 2
Compare < 66, 3 > with Chunk 2 Descriptor < 33, 4 >:
66 <= 33? NO, chunk-comparison = 3
Go to Chunk 3
Compare < 66, 3 > with Chunk 3 Descriptor < 68, 4 >:
66 <= 68? YES, chunk-comparison = 4
Insert < 66, 3 > into Chunk 3
66 <=40? NO, element-comparison = 6
66 <=43? NO, element-comparison = 7
66 <=55? NO, element-comparison = 8
66 <=64? NO, element-comparison = 9
66 <=68? YES, element-comparison = 10
Go to next element of term-b
Compare < 75, 1 > with Chunk 1 Descriptor < 12, 4 >:
75 <= 12? NO, chunk-comparison = 5
Go to Chunk 2
Compare < 75, 1 > with Chunk 2 Descriptor < 33, 4 >:
75 <= 33? NO, chunk-comparison = 6
Go to Chunk 3
Compare < 75, 1 > with Chunk 3 Descriptor < 68, 4 >:
75 <= 68 NO, chunk-comparison = 7
Go to Chunk 4
Compare < 75, 1 > with Chunk 4 Descriptor < 88, 2 >:
75 <= 88 YES, chunk-comparison = 8
Insert < 75, 1 > into Chunk 4
75 <= 72? NO, element-comparison = 11
75 <= 75? YES, element-comparison = 12

Therefore, we can find intersection of term-a and term-b lists with using skipping structure of chunk size 5 in **20(element-comparison + chunk-comparison = 12 + 8) comparisons**.

c. Advantages and Disadvantages of small and large skips in the posting lists

	Advantages	Disadvantages
Large Skips	The total number of chunks decreases since the number of documents within the chunk increases.	The total number of comparisons within the chunk increases since more documents found in each chunk.
	The total number of comparisons with the chunk descriptors decreases because are less chunks with more documents.	Since the chunk size is bigger, the search within the chunk takes more time.
	Since the chunk number is less, the search between chunks takes less time.	
	Unnecessary comparisons between chunks are eliminated.	
	Fewer skip yield larger skip spans so it tends to do less skips.	
	Requires less comparisons and fewer space.	

Table 6: The advantages and disadvantages of large skips

	Advantages	Disadvantages
Small Skips	The total number of comparisons within each the chunk decreases since the number of chunks increases	The total number of chunks increases since less documents found in each chunk.
	Since the chunk size is less, the search between chunks will take more time.	The total number of comparisons with the chunk descriptors increases because more chunks with less documents.
	Unnecessary comparisons within a chunk is eliminated.	Since the chunk number is more, the search between chunks takes more time.
	More chunks can be skipped	More skips yield shorter skip spans, so it tends to do more skips.
		Requires many skip pointer comparisons and pointer storage.

Table 7: The advantages and disadvantages of small skips

I benefited from [8] and [9] in order to construct the tables above.

d. Advantage of the skipping structure for disjunctive queries

I think it is not possible to take advantage of a skipping structure for disjunctive queries. In the case of a conjunctive (AND) queries, since it takes the intersection of posting queries in order to find the common terms which can also be seen in part c of question 4, it is useful. However, disjunctive (OR) queries take the union of posting lists, in other words, it is useless for finding common terms because the union of posting queries is the set of documents which are in one query, in another query, or both one query and another query. For instance, (term-a OR term-b) is the set of documents which are in term-a posting list, in term-b posting list, or both term-a and term-b posting lists. This set also contains the documents either in term-a or either in term-b. Therefore, it is not useful in the case of finding the common terms of posting lists.

6. Give a posting list of term-a (above it is given in standard sorted by document number order) in the following forms: 1) a) ordered by $f_{d,t}$, b) ordered by frequency information in prefix form. What are the advantages of the approaches a and b? Please again give these pros and cons in a tabular form. Do they have any practical value?

Solution of question 6:

term-a → < 1, 2 > < 3, 2 > < 9, 2 > < 10, 3 > < 12, 4 > < 18, 4 > < 20, 3 > < 23, 3 > < 25, 4 > < 33, 4 > < 40, 5 > < 43, 4 > < 55, 3 > < 64, 2 > < 68 ,4 > < 72, 3 > < 75, 1 > < 88, 2 >

a. Ordered by $f_{d,t}$(The frequency of term t in document d)

term-a → < 40, 5 > < 12, 4 > < 18, 4 > < 25, 4 > < 33, 4 > < 43, 4 > < 68 ,4 > < 10, 3 > < 20, 3 > < 23, 3 > < 55, 3 > < 72, 3 > < 1, 2 > < 3, 2 > < 9, 2 > < 64, 2 > < 88, 2 > < 75, 1 >

b. Ordered by Frequency Information in Prefix Form

term-a → < 5:1: 40 > < 4:6: 12, 18, 25, 33, 43, 68> <3:5: 10, 20, 23, 55, 72> <2:5: 1,3,9,64,88> <1:1: 75>

c. Advantages and Disadvantages of $f_{d,t}$ and Frequency Information in Prefix Form

	Advantages	Disadvantages
$f_{d,t}$	It improves the query processing by ordering frequency values in descending order because if large frequency values are interesting, it is useful to store them at the beginning rather than somewhere in the middle.	It requires more comparisons in the posting list to find the document.
	It is practical because if threshold for frequency is used, some frequencies below this threshold can be ignored which results in increasing the performance.	

Table 8: The advantages and disadvantages of $f_{d,t}$

	Advantages	Disadvantages
Frequency Information in Prefix Form	It provides better query processing as the size of the posting list is smaller than $f_{d,t}$.	The prefix form is difficult to construct so it is costly.
	It is practical because it saves memory and the prefix form provides compression since it does not require to store same frequency information by avoiding repeated frequencies, so it is useful when there is a long posting list.	The d-gaps are on average larger when document identifiers are sorted so the document number part of each pointer increases cost.

Table 9: The advantages and disadvantages of $f_{d,t}$

I benefited from [9] in order to construct the tables above.

7. What are the components of an information retrieval test collection? Explain the pooling approach? Please read the paper by Zobel (How Reliable Are the Results of Large-Scale Information Retrieval Experiments?) and give some reflections of his criticism of this approach.

Solution of question 7:

The components of an information retrieval test collection according to Zobel are:

- A set of documents
- A set of queries
- Relevance information about each document with respect to each query

Pooling Approach:

The pooling approach retrieves the top i documents and collect them in a pool for identification of relevant documents for each query where i is the document number. Thus, pooling enables identifying documents to be considered for relevance assessment, so it determines relevant documents for each query. Only the documents in the pool are treated as relevant. The problem about this approach is that unseen documents are assumed to be irrelevant which means that the relevant documents that are not evaluated are considered as irrelevant.

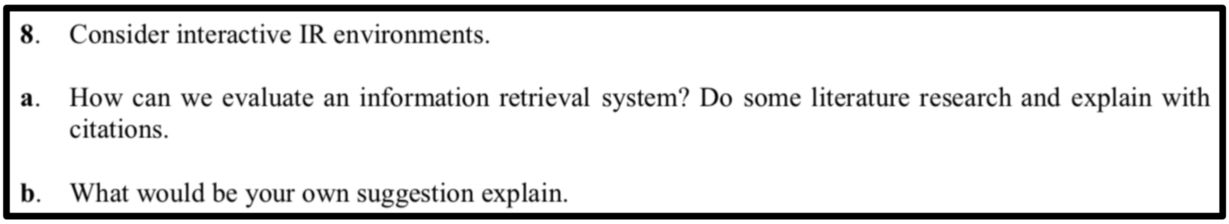
With information retrieval test collection, we need to utilize techniques such as pooling but Zobel has some criticism about this approach. First, Zobel claims that there are some difficulties related to pooling depth. He mentions that pooling might introduce bias by giving the example of fixed-depth pool which favors the numerical performance of a new system that is a simple combination of two other successful methods [10].

Secondly, there is a large database, the limitation of documents judged might be required. Pooling is used for this purpose and each system makes contribution to the same number of documents for evaluation so Zobel claims that sufficiently deep pool is fair enough to find most of relevant documents [10].

Thirdly, Zobel says that another potential disadvantage occurs when depth m exceeds pool depth p which causes similar systems to reinforce each other. This system reinforcement might introduce small distortions related to performance for systems that contributed to the pool. In addition, Zobel criticizes that pooling only identifies part of relevant documents so the effectiveness of a technique that does not contribute to the pool is underestimated. His observations show that the use of adequate pool depth is significant for both system reinforcement and system omission.

Lastly, Zobel believes that for the systems designed to maximize recall, the pooling cannot be used. It is because of the fact that it is impossible to be sure that pooling identifies most of relevant documents which results in highly uncertain results.

In his paper, Zobel considers determining the degree to which the use of pooling provides reliable results by decreasing this bias. In addition, he shows that if the pool depth, the number of identifiers taken from each run, is increased; it is possible to obtain useful estimates of numbers of new relevant documents that can be discovered for each query [10]. Thus, his aim is to demonstrate variation of standard pooling strategies which increases the number of relevant documents discovered for given judgement effort without compromising fairness and introducing bias [10].



8. Consider interactive IR environments.

a. How can we evaluate an information retrieval system? Do some literature research and explain with citations.

b. What would be your own suggestion explain.

Solution of question 8:

a. Evaluation of an information retrieval system

IIR (Interactive Information Retrieval) is information retrieval with users. Classic IR environment isolates humans from evaluation model, IIR focuses on users' cognitive and affective behaviors and experiences such as interactions between user and system, user and information. In other words, IR checks if the system retrieves relevant documents while IIR checks if people use the system to retrieve relevant documents. Salton and Cleverdon identify user effort measures and presentation issues as important component of IR evaluation, including attitudes and perceptions of users along with recall and precision [11].

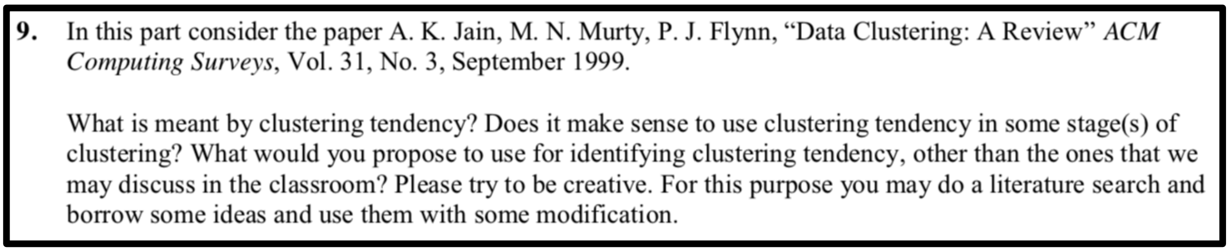
The evaluation techniques involve indexing, retrieval and ranking algorithms, user interfaces or interactive techniques [12]. IIR evaluation examines how differences in systems or interfaces (independent variable such as interface-type and task-type) impacts outcome measures (dependent variable). Developing a valid baseline in IIR evaluation includes identifying and blending the status quo and the experimental system [11]. Rotation and counterbalancing are important for experimental design. Then, sampling is required for different items in IIR such as users, tasks, topics and documents. Also, collections are needed which are known as test collections in IR. For collecting data in IIR evaluations, instruments like loggers and questionnaires are also helpful but there are also some data collection techniques such as think-aloud, stimulated recall, observation, spontaneous and prompted self-report, interviews etc. The final method for collecting data during IIR evaluation focuses on the outcome of the search. IIR researchers benefit from several approaches to examine the end products such as examination of references, expert assessments and cross-evaluation. Then, mostly quantitative data analysis is applied to the collected data.

There are four basic measures for the evaluation of IIR emerged from standard: contextual(individual differences, information needs), interaction(number of queries, number of search results viewed, number of documents viewed, query length), performance(interactive recall, interactive TREC precision, interactive user precision, relative relevance for modified version of recall and precision for IIR, cumulated gain, discounted cumulated gain, ranked half-life for cumulated gain measures, search speed, qualified search speed for time-based measures and cost-utility measures) and usability(effectiveness, efficiency, satisfaction).

There are also a few guidelines and models for conducting an evaluation for IIR evaluations [12]. Thomas and Hawking proposes an evaluation method based on preference. Subjects are presented with a split screen each displaying search results from two different search engines. Subjects are asked to make holistic evaluations basing their preferences on entire lists rather than individual documents [11]. The Cranfield model is based on the principle of test collections and measurement of recall and precision ratios as indicators of system performance and it is utilized for evaluation of both IR and IIR models [13]. However, it has some limitations based on assumptions on the cognitive and behavioral features of the environment in which (I)IR systems functions [13]. Borlund introduces an IIR evaluation model to facilitate IIR evaluation as close as possible to actual information searching and IR processes [14].

b. Own Suggestion

There is no single best evaluation method for IIR. It is more than system evaluation and retrieval effectiveness, so it requires more than one approach or method for evaluation. I think Borlund's IIR evaluation model is good for evaluation since it meets the requirements of three revolutions put forward by Robertson and Hancock-Beaulieu: the involvement of potential users as test participants, the application of dynamic and information needs (real and simulated individual information needs) and the employment of multidimensional and dynamic relevance judgements [13]. The reason behind my idea is that these requirements include most of the important features of an IIR system.



9. In this part consider the paper A. K. Jain, M. N. Murty, P. J. Flynn, "Data Clustering: A Review" *ACM Computing Surveys*, Vol. 31, No. 3, September 1999.

What is meant by clustering tendency? Does it make sense to use clustering tendency in some stage(s) of clustering? What would you propose to use for identifying clustering tendency, other than the ones that we may discuss in the classroom? Please try to be creative. For this purpose you may do a literature search and borrow some ideas and use them with some modification.

Solution of question 9:

Clustering tendency means examining the input data to see if it is feasible to a cluster analysis so there might be many clusters obtained from clustering algorithm but some of them might be useless. Clustering tendency can be used in some stages of clustering. Some clustering algorithms may produce clusters even if the data does not contain any clusters since it directly divides the data into clusters. Thus, before applying a clustering algorithm on the input data, we should evaluate the data set to see if it contains meaningful clusters and if they are meaningful how many clusters are there. After clustering tendency is evaluated, the optimal clusters in the input data can be found. There are also methods for evaluating the clustering tendency such as Hopkins Statistic and Visual Assessment of cluster Tendency (VAT) algorithm [15]. In addition, the idea of Cover Coefficient concept can be utilized since it is useful for tuning and optimization along with indexing-clustering relationship [16]. A modified version of VAT which utilizes similarity matrix in the input data for checking the high correlation between the documents using Euclidean distance measure rather than using dissimilarity matrix can be used.

10. Give a small similarity matrix of your choice with 5 documents and show that the complete link algorithm may produce different, non unique, clustering structures. Draw the corresponding dendrogram.

Propose a method that would prevent this non uniqueness problem.

Solution of question 10:

$$S = \begin{bmatrix} 1 & 0.1 & 0.2 & 0.6 & 0.6 \\ X & 1 & 0.3 & 0.4 & 0.5 \\ X & X & 1 & 0.3 & 0.5 \\ X & X & X & 1 & 0.2 \\ X & X & X & X & 1 \end{bmatrix}$$

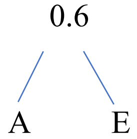
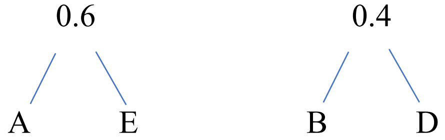
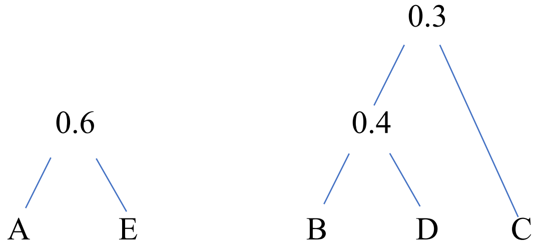
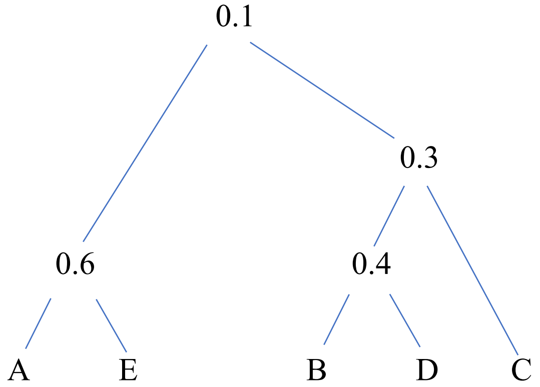
Document 1= A
Document 2= B
Document 3= C
Document 4= D
Document 5= E

Based on the similarity matrix of 5 documents above,

Pair	Similarity Pair
AE	0.6
AD	0.6
BE	0.5
CE	0.5
BD	0.4
BC	0.3
CD	0.3
DE	0.2
AC	0.2
AB	0.1

First Complete-Link Clustering:

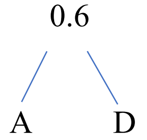
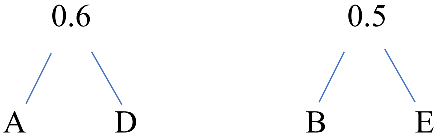
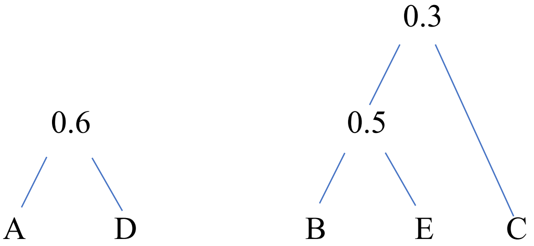
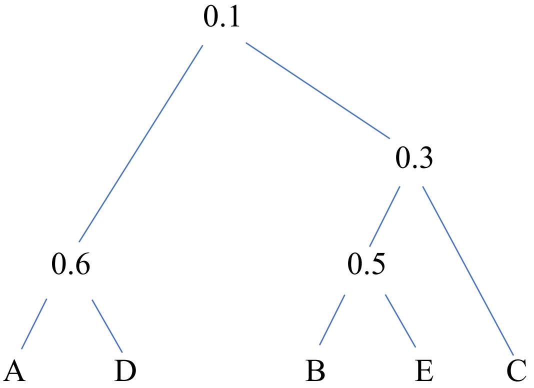
Step	Similarity Pair	Similarity
1	AE	0.6
2	AD	0.6
3	BE	0.5
4	CE	0.5
5	BD	0.4
6	BC	0.3
7	CD	0.3
8	DE	0.2
9	AC	0.2
10	AB	0.1

Step	Similarity Pair	Complete-Link Structure	Items Covered
1	AE, 0.6		AE
2	AD, 0.6	Too early to connect since we do not know DE	AE, AD
3	BE, 0.5	Too early to connect since we do not know AB	AE, AD, BE
4	CE, 0.5	Too early to connect since we do not know AC	AE, AD, BE, CE
5	BD, 0.4		AE, AD, BE, CE, BD
6	BC, 0.3	Too early to connect since we do not know CD	AE, AD, BE, CE, BD, BC
7	CD, 0.3		AE, AD, BE, CE, BD, BC, CD
8	DE, 0.2	Too early to connect since we do not know AB, AC	AE, AD, BE, CE, BD, BC, CD, DE
9	AC, 0.2	Too early to connect since we do not know AB	AE, AD, BE, CE, BD, BC, CD, DE, AC
10	AB, 0.1		AE, AD, BE, CE, BD, BC, CD, DE. AC, AB

$$S' = \begin{bmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.6 \\ X & 1 & 0.3 & 0.4 & 0.1 \\ X & X & 1 & 0.3 & 0.1 \\ X & X & X & 1 & 0.1 \\ X & X & X & X & 1 \end{bmatrix}$$

Second Complete-Link Clustering:

Step	Similarity Pair	Similarity
1	AD	0.6
2	AE	0.6
3	BE	0.5
4	CE	0.5
5	BD	0.4
6	BC	0.3
7	CD	0.3
8	DE	0.2
9	AC	0.2
10	AB	0.1

Step	Similarity Pair	Complete-Link Structure	Items Covered
1	AD, 0.6		AD
2	AE, 0.6	Too early to connect since we do not know DE	AD, AE
3	BE, 0.5		AD, AE, BE
4	CE, 0.5	Too early to connect since we do not know CB	AD, AE, BE, CE
5	BD, 0.4	Too early to connect since we do not know AB, DE	AD, AE, BE, CE, BD
6	BC, 0.3		AD, AE, BE, CE, BD, BC
7	CD, 0.3	Too early to connect since we do not know AB, AC, DE	AD, AE, BE, CE, BD, BC, CD
8	DE, 0.2	Too early to connect since we do not know AB, AC	AD, AE, BE, CE, BD, BC, CD, DE
9	AC, 0.2	Too early to connect since we do not know AB	AD, AE, BE, CE, BD, BC, CD, DE, AC
10	AB, 0.1		AD, AE, BE, CE, BD, BC, CD, DE. AC, AB

$$S'' = \begin{bmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.1 \\ X & 1 & 0.3 & 0.1 & 0.5 \\ X & X & 1 & 0.1 & 0.3 \\ X & X & X & 1 & 0.1 \\ X & X & X & X & 1 \end{bmatrix}$$

As it could be seen from two different complete-link clustering structure, the complete-link algorithm may generate two different clustering structures from similarity matrix S which results in different similarity matrices S' and S''.

In my example, changing the step order of AD and AE having the same similarity value results in different clustering structure. To determine step number for each pair in complete-link structure, the similarity pairs are ordered according to similarity value in descending order. However, when two pairs have the same similarity value, the problem is determining which one to insert to the structure first.

The clustering structure of complete-link depends on the order of inserting terms [16]. A method for determining which pair should be inserted first if equal similarity values have ties can be proposed. A rule for the method: the document pair having smaller document numbers is inserted first. For instance,

If **similarity value of document1-document5 = similarity value of document1-document4**, **insert document1-document4** to the structure first.

If **similarity value of document1-document5 = similarity value of document5-document1**, **insert document1-document5** to the structure first.

If **similarity value of document1-document2 = similarity value of document2-document3**, **insert document1-document2** to the structure first.

11. For a weighted D matrix according to the cover coefficient concept prove that diagonal entries of the C matrix can be smaller than the off diagonal entries, c_{ii} can be smaller than c_{ij} where i ǂ j. Proof by example is acceptable, but I prefer a general proof.

Solution of question 11:
Proof by example:

$$D = \begin{bmatrix} 1 & 0 \\ 5 & 2 \end{bmatrix}$$

First row sum = 1, Second row sum = 7
First column sum = 6, Second column sum = 2

$$S = \begin{bmatrix} 1/1 & 0 \\ 5/7 & 2/7 \end{bmatrix} \; S' = \begin{bmatrix} 1/6 & 0 \\ 5/6 & 2/2 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.1667 & 0.8333 \\ 0.1190 & 0.8810 \end{bmatrix}$$

As it could be seen from C matrix above, C_{11}(0.1667) is smaller than C_{12}(0.8333). Therefore, the proof is done.

12. Consider the D matrix given above.

a. Calculate the c_{12} entry of the C matrix using the double stage experiment and by drawing the tree like structure for this entry of C.

b. Obtain the clusters according to C3M for the D matrix given above.

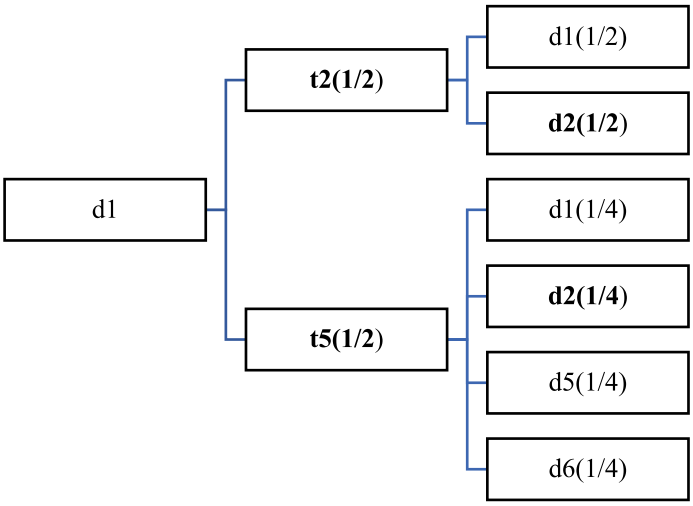
How many entries of the C matrix do you need to calculate?

Draw the IISD, inverted index for seed documents.

Please show your work for clustering.

Solution of question 12:

a. C12 entry of C Matrix using the double stage experiment



$C_{12} = 1/2*1/2 + 1/2*1/4 = 0.375$

b. Obtain clusters according to C3M

$$\boldsymbol{D} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix}$$

α: Inverse of row sums
α1 = 1/2, α2 =1/ 3, α3 =1/ 3, α4 =1/2, α5 = 1/3, α6 =1/ 2

β: Inverse of column sums
β1 = 1/2, β2 =1/2, β3 = 1/2, β4 = 1/3, β5 = 1/4, β6 =1/2

m: Number of documents
m = 6

n: Number of terms
n = 6

$$\boldsymbol{S} = \begin{bmatrix} \mathbf{0} & \mathbf{1/2} & \mathbf{0} & \mathbf{0} & \mathbf{1/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{1/3} & \mathbf{0} & \mathbf{1/3} & \mathbf{1/3} & \mathbf{0} \\ \mathbf{1/3} & \mathbf{0} & \mathbf{1/3} & \mathbf{1/3} & \mathbf{0} & \mathbf{0} \\ \mathbf{1/2} & \mathbf{0} & \mathbf{0} & \mathbf{1/2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1/3} & \mathbf{0} & \mathbf{1/3} & \mathbf{1/3} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1/2} & \mathbf{1/2} \end{bmatrix} \quad \boldsymbol{S'} = \begin{bmatrix} \mathbf{0} & \mathbf{1/2} & \mathbf{0} & \mathbf{0} & \mathbf{1/4} & \mathbf{0} \\ \mathbf{0} & \mathbf{1/2} & \mathbf{0} & \mathbf{1/3} & \mathbf{1/4} & \mathbf{0} \\ \mathbf{1/2} & \mathbf{0} & \mathbf{1/2} & \mathbf{1/3} & \mathbf{0} & \mathbf{0} \\ \mathbf{1/2} & \mathbf{0} & \mathbf{0} & \mathbf{1/3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1/2} & \mathbf{0} & \mathbf{1/4} & \mathbf{1/2} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1/4} & \mathbf{1/2} \end{bmatrix}$$

S for rows S' for columns

$$\boldsymbol{C} = \begin{bmatrix} \mathbf{0.3750} & \mathbf{0.3750} & \mathbf{0} & \mathbf{0} & \mathbf{0.1250} & \mathbf{0.1250} \\ \mathbf{0.2500} & \mathbf{0.3611} & \mathbf{0.1111} & \mathbf{0.1111} & \mathbf{0.0833} & \mathbf{0.0833} \\ \mathbf{0} & \mathbf{0.1111} & \mathbf{0.4444} & \mathbf{0.2778} & \mathbf{0.1667} & \mathbf{0} \\ \mathbf{0} & \mathbf{0.1667} & \mathbf{0.4167} & \mathbf{0.4167} & \mathbf{0} & \mathbf{0} \\ \mathbf{0.0833} & \mathbf{0.0833} & \mathbf{0.1667} & \mathbf{0} & \mathbf{0.4167} & \mathbf{0.2500} \\ \mathbf{0.1250} & \mathbf{0.1250} & \mathbf{0} & \mathbf{0} & \mathbf{0.3750} & \mathbf{0.3750} \end{bmatrix}$$

C = S x S'^{T}

Number of clusters = $n_c = \sum_{i=1}^{m} Cii$ = 0.375 + 0.3611 + 0.4444 + 0.4167 + 0.4167 + 0.375 = 2.3889 ≈ 2. Therefore, the number of clusters is 2.

Seed power of document di is given by $P_i = C_{ii} * (1 - C_{ii}) * X_{d_i}$

$P_1 = C_{11} * (1 - C_{11}) * X_{d_1}$ = 0.375*(1-0.375) *2 = 0.46875

$P_2 = C_{22} * (1 - C_{22}) * X_{d_2}$ = 0.3611*(1-0.3611) *3 = 0.69212

$P_3 = C_{33} * (1 - C_{33}) * X_{d_3}$ = 0.4444*(1-0.4444) *3 = 0.74073

$P_4 = C_{44} * (1 - C_{44}) * X_{d_4}$ = 0.4167*(1-0.4167) *2 = 0.48612

$P_5 = C_{55} * (1 - C_{55}) * X_{d_5}$ = 0.4167*(1-0.4167) *3 = 0.72918

$P_6 = C_{66} * (1 - C_{66}) * X_{d_6}$ = 0.375*(1-0.375) *2 = 0.46875

We concluded that the number of clusters is 2. The obtained cluster power seeds showed that d_3 and d_5 are cluster seeds since P_3 and P_5 returned the highest values in terms of cluster seed power. Thus, **d_3** and **d_5** are **cluster seeds** and **d_1, d_2, d_4** and **d_6** are **non-seeds**.

$$\boldsymbol{D} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix}$$

Inverted Index for Seed Documents:

t_1→<3,1>

t_2→<>

t_3→<3,1><5,1>

t_4→<3,1>

t_5→<5,1>

t_6→<5,1>

Cluster 1(cluster seed: d_3) → d_2, d_3 (C_{23}>C_{25})

Cluster 2(cluster seed: d_5) → d_1, d_4, d_5, d_6 (C_{15}>C_{13}, C_{43}>C_{45}, C_{65}>C_{63})

In total, **8 entries of C matrix**(C_{23},C_{25},C_{15},C_{13},C_{43},C_{45},C_{65},C_{63}) should be calculated in order to assign non-seeds to seeds and **6 entries of C matrix** (C_{11}, C_{22}, C_{33}, C_{44}, C_{55}, C_{66}) should be calculated for seed power of documents. In total, **14 entries of C matrix**(8+6) should be calculated.

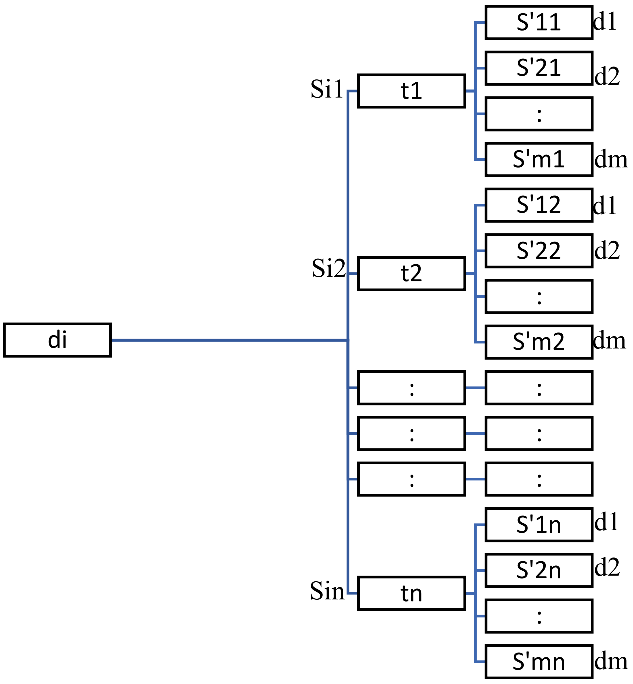
13. Consider the double stage probability experiment of the cover coefficient experiment. The D matrix size is given as m (no. of rows) by n (no. of columns). Consider the construction of the C, document by document, matrix. Explain your answer with a simple figure.

a. What is the possible maximum number of active branches (branches that exist, i.e. with non-zero values)?

b. What is the minimum number of active branches?

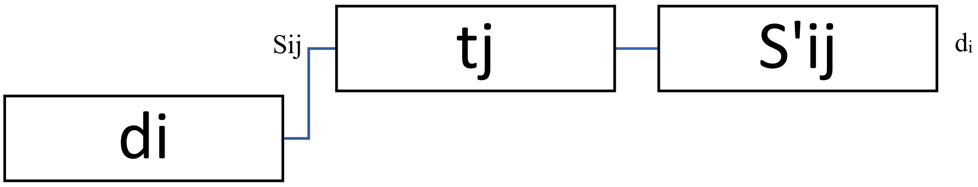
Solution of question 13:

a. Maximum number of active branches



The maximum number of active branches is obtained when all documents includes all terms which can be seen in the figure including all these branches. Therefore, for a document, n number of active branches for n terms and m*n number of active branches for m documents. For document i, n+m*n active branches are obtained. For all documents, **m(n+m*n) active branches in total** since total number of documents is m.

b. Minimum number of active branches



The minimum number of active branches is obtained when all documents do not contain any of terms. Since each document should contain at least one term, there should be 1 active branch for a term and 1 active branch for a document. Thus, there must be 2 active branches for each document. In total **2*m** active branches for all documents since total number of documents is m.

14. How can we use the indexing clustering relationships implied by the cover coefficient concept: What can be its practical uses?

Solution of question 14:

Indexing is used in order to represent documents by structures using index terms while retrieving documents. As it is demonstrated in the paper "Concepts and Effectiveness of the Cover-Coefficient-Based Clustering Methodology for Text Databases", there is a significant relationship between indexing and cover coefficient-based clustering [17]. This relationship indicates change in the size of indexing vocabulary affects the number of clusters. Therefore, it can be used to adjust the number and size of clusters based on parameters of the computer system searching the clusters. In addition, indexing-clustering relationships hold very closely in the case of binary indexing [17], so it is useful for the areas benefiting from binary indexing. However, even if the aim is not to cluster documents, the clustering methodology can be useful in producing adequate index vocabularies [18]. In the paper, Can and Özkarahan, narrow down and tune initial set of terms by the use of frequency and weight thresholds in conjunction with certain relationships derived between indexing and the cover coefficient-based clustering [18]. This shows that the indexing-clustering relationship has a practical usage in automatic tuning and optimization.

References

[1] N. I. T. L. (. R. G. o. t. I. A. Division(IAD), "The Sixth Text REtrieval Conference (TREC 6)-Appendix A".

[2] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval). [Accessed 19 October 2018].

[3] "Wikipedia," [Online]. Available: http://www.wikizeroo.com/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvUHJlY2lzaW9uX2FuZF9yZWNhbGwjUHJlY2lzaW9u. [Accessed 20 October 2018].

[4] C. J. v. RIJSBERGEN, "Chapter 7-Evaluation," in *Information Retrieval*, 1975, pp. 117-118.

[5] W. B. Croft, D. Metzler and T. Strohman, Search Engines Information Retrieval in Practice, Pearson Education, Inc., 2015.

[6] S. Teufel, "Lecture 5: Evaluation," [Online]. Available: https://www.cl.cam.ac.uk/teaching/1415/InfoRtrv/lecture5.pdf. [Accessed 20 October 2018].

[7] "Sørensen–Dice coefficient," [Online]. Available: http://www.wikizeroo.com/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvU8O4cmVuc2Vu4oCTRGljZV9jb2VmZmljaWVudA. [Accessed 22 October 2018].

[8] C. Hauff, "Indexing and booelan retrieval," 2012. [Online]. Available: https://chauff.github.io/documents/ir-2011_12/lecture3.pdf. [Accessed 21 October 2018].

[9] J. Zobel and A. Moffat, "Inverted Files for Text Search Engines," *ACM Computing Surveys,* 2006.

[10] J. Zobel, "How Reliable are the Results of Large-Scale Information Retrieval Experiments?," *ACM,* 1998.

[11] D. Kelly, "Methods for Evaluating Interactive Information Retrieval Systems with Users," *Foundations and Trends in Information Retrieval,* 2009.

[12] D. Kelly and C. R. Sugimoto, "A Systematic Review of Interactive Information Retrieval Evaluation Studies, 1967-2006," *Journal of the American Society for Information Science and Technology,* 2011.

[13] P. Borlund, "The IIR evaluation model: a framework for evaluation of interactive information retrieval systems," *Information Research,* 2003.

[14] P. Borlund, "Interactive Information Retrieval:Introduction," *Journal of Information Science Theory and Practice,* 2013.

[15] A. C. Tendency. [Online]. Available: https://www.datanovia.com/en/lessons/assessing-clustering-tendency/. [Accessed 22 October 2018].

[16] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Englewood Cliffs, New Jersey: Prentice Hall Advanced Reference Series, 1988.

[17] F. Can and E. A. Ozkarahan, "Concepts and Effectiveness of the Cover Coefficient-based Clustering Methodology for Text Databases," *ACM Transactions on Database Systems,* 1990.

[18] F. Can and E. Ozkarahan, "An Automatic and Tunable Document Indexing System," *ACM Conference on Research and Development in Information Retrieval,* 1986.

[19] W. Frakes and R. Baeza-Yates, Information Retrieval: Data Structures & Algorithms.

[20] C. D. Manning, P. Raghavan and H. Schütze, "Introduction to Information Retrieval".

[21] K. Zuva and T. Zuva, "Evaluation of Information Retrieval Systems," *International Journal of Computer Science & Information Technology,* 2012.