Computer Engineering Department
Bilkent University

CS433-CS533: **Information Retrieval Systems**
Assignment No. 2
November 1, 2024   Version 1
Due date: November 11 Monday 11:59 pm

Notes: **1**. Submissions prepared by a word processor is expected: Use latex or word: handwritten neat submissions are also accepted. Solve the questions in the order they are listed. For all, give reasonable amount of detail. Upload a pdf file. **2**. Sloppy submissions will not be graded. **3**. If you see an incomplete explanation or a problem in wording introduce your solution, explain, and proceed. **4**. You can be late for two days and for each late day 10 points will be taken off. (Any time on Nov. 12: -10, Any time on Nov. 13: -20.)

1. Consider the following ranked 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. Give the R-Precision (See TREC-6 Appendix A for definition) for Q1 and Q2.
b. 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.
c. 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. The Manning et al. book *Introduction to Information Retrieval* has some explanation.

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

3. 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. This is the most efficient approach we studied.

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

4. Three proofs for C3M, all tiny: they all are a matter of knowing the concepts or definitions.
m: number of documents, n: number of terms.

a. 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.

b. For a binary D matrix if all $c_{ii} = 1/m$ then all $c_{ij}= 1/m$ ($1\leq j\leq m$, jǂ i).

c. According the C3M concepts show that δ= (n/m) x δ',where δ and δ' are average decoupling coefficient of documents and terms.

5. Consider the D matrix given above in question 3.

a. Calculate the c_{34} entry of the C matrix using the double stage experiment and by drawing the tree like structure for this entry of the C matrix.

b. Obtain the clusters according to C3M for the D matrix. How many entries of the C matrix do we need to calculate?

Draw the IISD: inverted index for seed documents. Show how you select the seed documents.

Please show your work for clustering with some detail.

6. Consider the double stage probability experiment of the C3M algorithm. 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 for each case. See Fig. 1 of the C3M TODS paper.

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

b. What is the minimum number of active branches for a C matrix entry?

7. How can we use the indexing clustering relationships implied by the cover coefficient concept: What can be its practical uses? See C3M TODS paper, Section 2.7.

The indexing clustering relationship is expressed as follows

$$n_c= (m x n) / t = m / t_g= n / x_d$$

n_c: approximate number of clusters
m: number of documents
n: number of terms
t: number of non zero entries in the D matrix
x_d: depth of indexing (average number of terms per document)
t_g: term generality (average number of documents per term)

8. Consider the following document by document similarity matrix.

$$\begin{bmatrix} 1 & 0.4 & 0.8 & 0.5 & 0.3 \\ X & 1 & 0.6 & 0.3 & 0.4 \\ X & X & 1 & 0.5 & 0.2 \\ X & X & X & 1 & 0.6 \\ X & X & X & X & 1 \\ & & & & \end{bmatrix}$$

a. Obtain the corresponding single-link dendrogram and the similarity matrix implied by the dendrogram. Give the similarity matrix implied by the dendrogram.

b. Obtain the corresponding complete-link dendrogram and the similarity matrix implied by the dendrogram. Give the similarity matrix implied by the dendrogram.

9. Consider the following specifications for a document database:

m (No. of documents)	= 360
n_c (No. of clusters)	= 24
k (No. of relevant documents for a given query)	= 3

Assume that (1) documents are randomly distributed among the clusters; (2) each cluster has the same size.

What is the expected number of clusters to be accessed to retrieve all relevant documents of the query? (Use Yao's formula, see the related paper: Yao, S. B., "Approximating block accesses in database organizations." *Communication of the ACM*, Vol. 20, No. 4, 1977, pp. 260-261.

10. Consider five data items a, ... e. The first partitioning structure P_1 has two clusters: {a, b, c}, {d, e}. The other partitioning structure is given as P_2= {a}, {b, c, d}, {e}.

a. Assume that P_2 is the ground truth (gold standard) an calculate the agreement between P_1 and P_2 using the Rand index (give TP, TN, FP, FN values).

b. Do the same calculation and this time assume that P_1 is the ground truth.

11. In this part consider the paper J. Zobel, A. Moffat, "Inverted files for text search engines." *ACM Computing Surveys*, Vol. 38, No. 2, 2006. Also look at the slides (see also Moodle, Handwritten Class Notes 2023, item no. 5: Inverted IndexNotes).

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> <37, 4>
<40, 5> <43, 4> <55, 3> <64, 2> <68, 4> <72, 3> <75, 1> <88, 2>

term-b: <15, 7> <66, 3> <75, 1> <90, 2>

In a posting list tuple like <1, 2> the first number indicates the document number, and the next number indicates the frequency of the term in this document ($f_{d,t}$). The posting list for term-a, for example, indicates that term-a appears in d1 twice and in d10 three times, etc.

a. 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? Give the number and show the comparisons.

b. Introduce a skip structure with 3 items in each subgroup i.e. data chunk (data chunks will also have the lowest document number of the following data chunk).

Draw the corresponding figure.

Give the number of comparisons involved to process the same query using this skipping structure (for an example see the related notes p. 2).

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 (queries based on **or** operations)? Please explain.

12. Consider the "**original**" posting lists given for two terms below.
term-1 ==> <16, 4> <34, 3> <47, 4> <109, 7>
term-2 ==> <15, 3> <22, 3> <33, 2> <34, 6> <86, 4> <108, 7>

a. Construct the posting list structure and organize the posting list in such a way that documents that contain the term most frequently will be at the beginning of the posting lists and it will be followed by the document that contains the term with the next highest frequency. This is "**frequency ordered**" posting list. If we have two documents that contain the term with the same frequency keep these documents in ascending order according to their document number.

b. Consider *ranked* query processing for the Google-like query Q: <term-1 term-2>. For ranking the documents use the summation of $f_{d,t}$ values of the query terms for the documents. Assume that term-1 is more important.

Give ranked documents for the following posting list structures. For limited accumulator cases assume that we have 5 accumulators. Using
i. Original posting lists(this involves no limit for the accumulators),
ii. Frequency ordered posting lists with limited accumulators,
iii. Frequency ordered posting lists with limited accumulators and interleaved processing

Show your work with reasonable detail so that one can follow it.