CS533 Information Retrieval Systems

Homework – 5

1. Consider the following symmetric similarity matrix for a document collection with four documents. The similarity between d1 and d2 is 0.67, etc. The lower portion of the matrix is not shown.

$$S = \begin{bmatrix} 1.00 & 0.67 & 0.50 & 0.20 \\ - & 1.00 & 0.80 & 0.10 \\ - & - & 1.00 & 0.00 \\ - & - & - & 1.00 \end{bmatrix}$$

Consider the following respective similarities of these documents to a given query:
(d_1, 0.80) (d_2, 0.70), (d_3, 0.40), (d_4, 0.60).

Use the MMR algorithm for selecting the best matching first two documents. Consider the following λ values. After each case give the diversity among the selected documents; where diversity is defined as (1-average similarity among selected documents).

Does the MMR algorithm provide what it promises? For each case please show your steps concisely. (Ref. Carbonell Goldstein SIGIR 1998 paper.)

a. Use λ = 1.00.
b. Use λ = 0.00.
c. Use λ = 0.50.
d. Please repeat the steps a to c three documents.

1. The respective similarities of the documents to a given query is given as (d1, 0.80) (d2, 0.70), (d3, 0.40), (d4, 0.60) In MMR, the document that is the most similar to a given query is chosen as the first element of the reranked set. Thus, d1 is the first element added to the set S, that is the set of selected documents so far, in this question. MMR value is given by:

$$MMR = arg \max_{di \in R/S} \left[\lambda sim_q(di, q) - (1 - \lambda) \max_{dj \in S} sim(di, dj) \right]$$

a. λ=1 means that the documents are reranked according to their similarity to a given query. Thus, our final set S = {d1, d2, d4, d3} where the best matching first two documents are d1 and d2. We were asked to rerank the set up to three documents, so S becomes {d1, d2, d4}. The diversity among the selected documents is 1-(S_{12}+ S_{24} +S_{14})/3= 1-(0.67+0.10+0.20)/3=0.68

b. λ=0 means that the documents are reranked according to diversity. Then,
   - 1st iteration: S={d1}, max {- S_{12}, -S_{13}, -S_{14}}=max{-0.67, -0.50, -0.20}=d4. S becomes {d1, d4}.
   - 2nd iteration: S= {d1, d4}, max { -S_{12}, -S_{13}, -S_{24}, -S_{34}}= max {-0.67, -0.50, -0.1, 0}= d3. S becomes {d1, d4, d3}. The best matching first two documents are d1 and d4, and the diversity of the set S is 1-(0.2+0.5+0)/3=1-0.23=0.77

c. λ=0.5 means that equal importance is given to relevance and diversity.
   - 1st iteration: S={d1}
   - MMR(d2) = 0.5*0.7 – 0.5*0.67=0.015
   - MMR(d3) = 0.5*0.4 – 0.5*0.5=-0.05
   - MMR(d4) = 0.5*0.6 – 0.5*0.2=0.2 → the set S becomes {d1, d4}

- 2nd iteration: S{d1, d4}
  - Here S12 > S24
- MMR(d2) = 0.5*0.7 – 0.5*S12(=0.67) =0.015 → the set S becomes {d1, d4, d2}
  - Here S13 > S34
- MMR(d3) = 0.5*0.4 – 0.5*S13(=0.5) =-0.05

So, the best matching first two documents are d1 and d4, and the diversity among the selected documents is 1-(S_{12}+ S_{24} +S_{14})/3= 1-(0.67+0.10+0.20)/3=0.68

d. In this step, we need to choose the best matching first three documents. λ=1 results in the same result as the one in step (a) since the documents are ranked only according to their similarity to the given query again. Then, the best matching first 3 documents are d1, d2, d4. For λ=0 case, we need to rank the documents according to diversity. Then, the resulting ranked set S becomes {d1, d4, d3}. So, the best matching first three documents are d1, d4, d3. For the last case where λ=0.5, we have d1, d4, d2 as the best matching first three documents since the procedure follows the same equations and similarity coefficients as the ones in step (c).

2. The search result for a query in ranked order are given in the following table. Different meanings of documents d_1, d_2 ... d_{10} are shown by m_1, m_2 , ... m_6.

Rank	Document	Subtopic
1	d_1	m_3
2	d_2	m_4
3	d_3	m_1, m_2
4	d_4	m_5, m_6
5	d_5	m_6
6	d_6	m_5
7	d_7	m_4
8	d_8	m_3
9	d_9	m_2
10	d_{10}	m_1

a. Find s-recall at rank position 5 and 10.

b. Find precision IA at rank position 5 and 10.

2. a. s-recall is given by # of meanings gathered so far / total # of meanings. According to the search results, we have 6 different meanings. So, s-recall@5 = 6/6 = 1 and s-recall@10 = 6/6=1.

b.

Rank	m1	m2	m3	m4	m5	m6
1	0	0	1	0	0	0
2	0	0	0	1	0	0
3	1	1	0	0	0	0
4	0	0	0	0	1	1
5	0	0	0	0	0	1
6	0	0	0	0	1	0
7	0	0	0	1	0	0
8	0	0	1	0	0	0
9	0	1	0	0	0	0
10	1	0	0	0	0	0
Precision@10	2/10	2/10	2/10	2/10	2/10	2/10

Precision IA @5 = 1/6*(1/5 + 1/5 + 1/5 + 1/5 + 1/5 + 2/5) = 0.233

Precision IA @10 = 1/6*(0.2+0.2+0.2+0.2+0.2+0.2) = 0.2

3. Consider the following search engines A, B, C, and D and ranking provided by them for the documents a, b, c, d, e, and f.

A= {b, a, d, c}
B= {b, a, d, f}
C= {b, c, d, a}
D= {a, c, d, e}

Rank the documents according to the following data fusion methods. (Ref. Nuray Can IPM 2006 paper.)

a. Reciprocal rank,

b. Borda count,

c. Condorcet.

3.

a. Reciprocal rank is given by:

$$r(d_i) = \frac{1}{\sum {}^{1}/_{position(d_{ij})}}, j = search\ engine$$

- r(a) = 1/(½ + ½ + ¼ + 1) = 0.44
- r(b) = 1/(1 + 1 + 1 + 0) = 0.33
- r(c) = 1/(¼ + 0 + ½ + ½) = 0.80
- r(d) = 1/(1/3 + 1/3 + 1/3 + 1/3) = 0.75
- r(e) = 1/(0 + 0 + 0 + ¼) = 4
- r(f) = 1/(0 + ¼ + 0 + 0) = 4 → ranks are given as b > a > d > c > e = f

b. This is a 6-way vote so the highest ranked individual gets 6 votes and each subsequent individual gets one vote less (if there are candidates left unmarked by the voter, the remaining points are divided evenly among the unmarked candidates).

- BC(a) = 5 + 5 + 3 + 6 = 19
- BC(b) = 6 + 6 + 6 + 1.5 = 19.5
- BC(c) = 3 + 1.5 + 5 + 5 = 14.5
- BC(d) = 4 + 4 + 4 + 4 = 16
- BC(e) = 1.5 + 1.5 + 1.5 + 3 = 7.5
- BC(f) = 1.5 + 3 + 1.5 + 1.5 = 7.5 → ranks are given as b > a > d > c > e = f

c. In Condorcet method, we build a pairwise comparison matrix that shows how many times a document beats other one, is beaten by the other one and they are tied. So, the pairwise comparison matrix becomes

	a	b	c	d	e	f
a	-	1, 3, 0	3, 1, 0	3, 1, 0	4, 0, 0	4, 0, 0
b	3, 1, 0	-	3, 1, 0	3, 1, 0	3, 1, 0	3, 0, 1
c	1, 3, 0	1, 3, 0	-	2, 2, 0	3, 0, 1	3, 1, 0
d	1, 3, 0	1, 3, 0	2, 2, 0	-	4, 0, 0	4, 0, 0
e	0, 4, 0	1, 3, 0	0, 3, 1	0, 4, 0	-	1, 1, 2
f	0, 4, 0	0, 3, 1	1, 3, 0	0, 4, 0	1, 1, 2	-

- Pairwise winners

	Win	Lose	Tie
a	4	1	0
b	5	0	0
c	2	2	1

d	2	2	1
e	0	4	1
f	0	4	1

Then, the ranks become b>a>c=d>e=f

4. Consider a document collection containing 512,000 objects. The signature of an object requires 1024 bits. What are the signature file sizes using the following signature file organization methods?

a. Sequential Signatures (SS),

b. Bit-sliced Signatures (BS).

4.

a. Sequential signature method assigns 1024 bits to each document in the collection. Thus, signature file size becomes 1024*521000=524288000 bits = 64000 Kbytes.

b. Bit-sliced signature method holds the signature values of the documents in column vectors, e.g., the 1st bit position values of all documents are held in an N*1 vector, where N=512000 in our case. We have 1024 columns for all bits. Then, signature file size becomes 512000*1024=524288000 bits = 64000 Kbytes.

5. In the environment of the above question consider a query with 5 bit positions equal to one. These bit positions are 1, 2, 3, and 4. The leftmost most significant position of a signature is bit position 1. For filtering; i.e., for query signature - document signatures matching; how many pages need to be accessed in the case of SS and BS?

Page size is given as 0.5 K bytes.

Note that in SS we place signatures one after the other and in the case of BS we place bit slices one after the other: Place the first bit slice and then right after that place the second bit slice and if there is room in the page allocated to slice 1 use the remaining space for the second bit slice and carry on like this.

5. a. We know that the first 4 leftmost digits of the query is set which leaves us 1020 unknown bits. The query needs to access all partitions starting with 1111. Thus, it can include one sixteenth of whole collection at most if we assume partitions 0000 to 1111 are equally sized. This narrows down the number of related documents to 512000/16= 32000. We have a page size of 0.5 Kbytes = 4000 bits. Then, the number of pages that need to be accessed becomes 32000/4000 = 8.

6. Consider the following signatures.
S1: 0110 1010
S2: 0100 0110
S3: 1110 0011
S4: 1100 0011
S5: 0011 1010
S6: 1010 0101
S7: 1011 0010
S8: 0000 1111
S9: 1010 0110
S10: 1011 0100

a. Use the fixed prefix, FP, method to partition the above signatures. Take k (key length) as 2. Show the file structure (contents of the pages etc.).

b. Now consider the following queries.
Q1: 1110 0001
Q2: 0110 0011
Q3: 1100 1100

i. Give the partition activation ratio and signature activation ratio for the queries in the FP environment.

ii. Use the partitions of section-a to calculate the average turnaround time to process the queries in sequential and parallel FP environments. Use the assumptions that we used in the class, e.g., the processing of one page signature requires 1 time unit, etc. What is the speed up ratio for the parallel environment?

6. We have the following signatures:
S1: 0110 1010
S2: 0100 0110
S3: 1110 0011
S4: 1100 0011
S5: 0011 1010
S6: 1010 0101
S7: 1011 0010
S8: 0000 1111
S9: 1010 0110
S10: 1011 0100

a.

00	01	10	11
S5, S8	S1, S2	S6, S7, S9, S10	S3, S4

b. We have the following queries Q1: 1110 0001, Q2: 0110 0011, Q3: 1100 1100

i. Partition activation ratio = # of activated partitions / total # of partitions
Signature activation ratio = # of signatures in activated partitions / total # of signatures

Query	K=2	PAR	SAR
Q1 = 1100 0001	1(11)	¼	2/10
Q2 = 0110 0011	2(01, 11)	2/4	4/10
Q3 = 1100 1100	1(11)	¼	2/10

ii. In sequential FP environment, we assume all queries arrive at t=0 and the processing of one page signature requires 1 time unit. Then, sequential processing time becomes as follows:

Q1	Q2	Q2	Q3

0 1 2 3 4

Turnaround time = Time of completion – time of arrival
Average sequential turnaround time = (1+3+4)/3 = 2.66 tu

In parallel FP environment, we follow the same assumptions as in the sequential case. Then, parallel processing time becomes:
00 ← PE1: -
01 ← PE2: Q2
10 ← PE3: -
11 ← PE4: Q1, Q2, Q3. So the average parallel turnaround time=(1+2+3)/3=2. The speed-up ratio = total processing time in sequential case / total processing time in parallel case = 4/3=1.33

7. Partition the signatures of question 6 using the following partitioning methods.

a. EPP (take z= 2).

b. FKP (take k= 2).

c. To process the queries of the above question which pages need to be accessed and why?

Ref. To answer this question consider the paper Lee and Leng 1989 paper in ACM TOIS, "Partitioned Signature Files: Design Issues and Performance Evaluation," or "Signature Files: An Integrated Access Method for Formatted and Unformatted Databases" by Aktug & Can on the web (Can Aktug explains it with a simple figure.

7.

a. In Extended Prefix Partitioning (EPP), we cut all signatures from the position of their z-th 0 and remove the right part. Then, we partition the new signatures accordingly. So, our signatures become:
S1: 0110
S2: 010
S3: 11100
S4: 1100
S5: 0011
S6: 1010
S7: 10110
S8: 00
S9: 1010
S10: 10110

00	010	0011	0110	1010	10110	1100	11100
S8	S2	S5	S1	S6, S9	S7, S10	S4	S3

b. In Floating Key Partitioning (FKP) for k=2, we reduce each signature to 2 bits. Reduction procedure is done down to the two consecutive bits that have minimum number of 1s. The resulting signature also include the starting bit position. In case of ties, the leftmost bit is selected. The signatures become:
S1: 1-10

S2: 3-00
S3: 4-00
S4: 3-00
S5: 1-00
S6: 4-00
S7: 5-00
S8: 1-00
S9: 4-00
S10: 7-00

1-10	1-00	3-00	4-00	5-00	7-00
S1	S5, S8	S2, S4	S3, S6	S7	S9, S10

c. In EPP, Q1 becomes 11100 and in FKP 4-00. Q1 needs to access S3 in EPP, and the pages 3-00 (S2, S4), 4-00 (S3, S6), 5-00 (S7) where page 3-00 is false match as S2, S4 do not match Q1. Q2 becomes 0110 and 4-00 in EPP and FKP, respectively. In EPP, it needs to access 0110 (S1), and in FKP it needs to access 4-00 (S3, S6), 5-00 (S7). Q3 becomes 1100 and 3-00 in EPP and FKP, respectively. Then, it needs to access 1100 (S4) in EPP and 3-00 (S2, S4), 7-00 (S9, S10) in FKP.

8. Partition the signatures of question 6 with the linear hashing algorithm (using suffixes). Assume that each data block can contain three signatures. (Bkfr= 3) and LF= 2/3 as in our in class example.

For Q1 and Q2 of question 3 please specify which pages need to be access and please explain briefly.

For these queries indicate which data pages need to be accessed.

Ref. Zezula et al ACM TOIS 1991 Dynamic partitioning of signature files.

8. We have Bkfr = 3 and LF = 2/3. The signatures are given below.
S1: 0110 1010
S2: 0100 0110
S3: 1110 0011
S4: 1100 0011
S5: 0011 1010
S6: 1010 0101
S7: 1011 0010
S8: 0000 1111
S9: 1010 0110
S10: 1011 0100
Every time we reach LF, we insert LF*Bkfr number of signatures in the corresponding block(s) and update the structure according to the current boundary value, bv.

bv → 0	S1	S2	S5
1	S3	S4	S6

00			
bv → 1	S3	S4	S6
10	S1	S2	S5

00								
bv → 1	S3	S4	S6	→	S8			
10	S1	S2	S5	→	S7			

bv → 00								
01	S6							
10	S1	S2	S5	→	S7			
11	S3	S4	S8					

bv → 00	S10							
01	S6							
10	S1	S2	S5	→	S7	S9		
11	S3	S4	S8					

000								
bv → 01	S6							
10	S1	S2	S5	→	S7	S9		
11	S3	S4	S8					
100	S10							

The resulting LF = 10/15 = 2/3.

9. Consider the following information filtering profiles used in a Boolean environment.

P1= a, b, c, d, e, f
P2= a, b, e, f
P3= b, c, f
P4= b, d, f
P5= a, c, f

Assume that when the terms are sorted in frequency order according to their number of occurrences in documents term a is the least frequently used term in the documents and is also the most frequently used term in the user profiles. The sorted term list continues as b, c … f.

Consider the ranked key method explained in the paper by Yan and Garcia-Molina (Index structures for selective dissemination of information under the Boolean model, *ACM TODS*) and draw the directory and the posting lists for the ranked key method.

9. We have the following information filtering profiles used in a Boolean environment
P1 = a, b, c, d, e, f
P2 = a, b, e, f
P3 = b, c, f
P4 = b, d, f
P5 = a, c, f

a	→	P1	5	b	c	d	e	f		P2	3	b	e	f		P5	2	c	f
b	→	P3	2	c	f		P4	2	d	f									
c																			
d																			
e																			
f																			