# CS533 Information Retrieval – Fall 2017

## HW5 Solutions

### Q1

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

($d_1$, 0.80) ($d_2$, 0.70), ($d_3$, 0.40), ($d_4$, 0.60).

MMR = $argMax_{d_i \in R \backslash S} \left[ \lambda sim_1(d_i, q) - (1-\lambda) max_{d_j \in S} sim_2(d_i, d_j) \right]$

C = Collection
$d_i$, $d_j$ ∈ $C$

$\lambda = [0, 1]$ $\quad \lambda \uparrow$ *higher accuracy and relevance,* $\quad \lambda \downarrow$ *higher diversity*

**a)** For λ = 1, we select documents based on similarity
Thus, $d_1 > d_2 > d_4 > d_3$
Start with $d_{1\,,}$ S = {d1} R\S = { $d_2$, $d_4$, $d_3$}

MMR = $argMax_{d_i \in R \backslash S} \left[ \boxed{\lambda sim_1(d_i, q)} - \cancel{(1-\lambda) max_{d_j \in S} sim_2(d_i, d_j)} \right]$

**MMR($d_2$) = 0.7 ← Maximum. Therefore S={** $d_1$, $d_2$ **} = 0.67**
MMR($d_3$) = 0.4
MMR($d_4$) = 0.6

**b)** For λ = 0 : Rank documents based on diversity

MMR = $argMax_{d_i \in R \backslash S} \left[ \cancel{\lambda sim_1(d_i, q)} \boxed{- (1-\lambda) max_{d_j \in S} sim_2(d_i, d_j)} \right]$

S= { $d_1$} without MMR ; R\S = { $d_2$, $d_3$, $d_4$}
MMR ($d_2$ )= –0.67 = -0.67
MMR ($d_3$ )= –0.5 = -0.5
MMR ($d_4$ )= -0.2 = -0.2 ← Maximum. Therefore S={ $d_1$, $d_4$ } = 0.20

**c)** For λ = 0.5 :
S= { $d_1$} without MMR ; R\S = { $d_2$, $d_3$, $d_4$}
MMR ($d_2$ )= 0.5 * 0.7 – 0.5 * 0.67 = 0.015
MMR ($d_3$ )= 0.5 * 0.4 – 0.5 * 0.5 = -0.05
MMR ($d_4$ )= 0.5 * 0.6 – 0.5 * 0.2 = 0.2 ← Maximum. Therefore S={ $d_1$, $d_4$ } = 0.20

**d)** For λ = 1, we select documents based on similarity
From answer Q1a → S={ $d_1$, $d_2$ } R\S = { $d_4$, $d_3$}

$$MMR = argMax_{d_i \in R \setminus S}\left[\boxed{\lambda sim_1(d_i, q)} - \cancel{(1-\lambda) max_{d_j \in S} sim_2(d_i, d_j)}\right]$$

MMR($d_3$) = 0.4
MMR($d_4$) = 0.6 ← **Maximum. Therefore S={** $d_1$, $d_2$, $d_4$ **}**

S = { $d_1$, $d_2$} + { $d_2$, $d_4$ } + { $d_1$, $d_4$ } = 0.67 + 0.10 + 0.20 = **0.97**

For λ = 0 : Rank documents based on diversity

$$MMR = argMax_{d_i \in R \setminus S}\left[\cancel{\lambda sim_1(d_i, q)}\boxed{- (1-\lambda) max_{d_j \in S} sim_2(d_i, d_j)}\right]$$

From answer Q1b → S={ $d_1$, $d_4$ } R\S = { $d_2$, $d_3$ }

MMR ($d_2$ )= −0.67 = -0.67
MMR ($d_3$ )= −0.5 = -0.5 ← Maximum. Therefore S={ $d_1$, $d_4$, $d_3$ } = 0.7

S = { $d_1$, $d_4$} + { $d_1$, $d_3$ } + { $d_4$, $d_3$ } = 0.2 + 0.50 + 0.0 = **0.70**

For λ = 0.5 :
From answer Q1c → S={ $d_1$, $d_4$ } R\S = { $d_2$, $d_3$ }

MMR ($d_2$ )= 0.5 * 0.7 – 0.5 * 0.67 = 0.015 ← Maximum. Therefore S={ $d_1$, $d_4$, $d_2$} = 0.97
MMR ($d_3$ )= 0.5 * 0.4 – 0.5 * 0.5 = -0.05

S = { $d_1$, $d_4$} + { $d_1$, $d_2$ } + { $d_4$, $d_2$ } = 0.2 + 0.67 + 0.1 = **0.97**

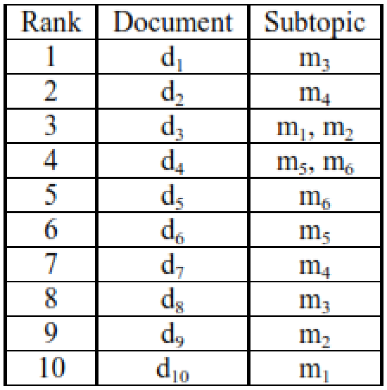
| λ | **K=2** | **K=3** |
|---|---|---|
| **1.0** | **0.67** | **0.97** |
| **0.5** | **0.20** | **0.97** |
| **0.0** | **0.20** | **0.70** |

**Remarks:**
As the below table shows, λ changes the relevance / diversity balance of results. As λ increases, the total similarity increases, and as λ decreases, the total similarity decreases and diversity increases. The results can better be observed with more number of documents and terms.

$$\lambda = [0, 1] \quad \lambda \uparrow higher\ accuracy\ and\ relevance, \quad \lambda \downarrow higher\ diversity$$

| λ | **Similarity K=2** | **Similarity K=3** |
|---|---|---|
| **1.0** | **0.67** | **0.97** |
| **0.5** | **0.20** | **0.97** |
| **0.0** | **0.20** | **0.70** |

## Q2



| 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) S-Recall can be defined as**:
The unique number of topics covered until n<sup>th</sup> rank / Total number of unique topics

S-Recall @ 5 = 6 / 6 = 1.0
S-Recall @ 10 = 6 / 6 = 1.0

**b)** Precision -IA

| Rank | $m_1$ | $m_2$ | $m_3$ | $m_4$ | $m_5$ | $m_6$ |
|---|---|---|---|---|---|---|
| 1 | | | 1 | | | |
| 2 | | | | 1 | | |
| 3 | 1 | 1 | | | | |
| 4 | | | | | 1 | 1 |
| 5 | | | | | | 1 |
| 6 | | | | | 1 | |
| 7 | | | | 1 | | |
| 8 | | | 1 | | | |
| 9 | | 1 | | | | |
| 10 | 1 | | | | | |
| P@5 | 1/5 | 1/5 | 1/5 | 1/5 | 1/5 | 2/5 |
| P@10 | 2/10 | 2/10 | 2/10 | 2/10 | 2/10 | 2/10 |

**Precision-IA @ 5**

$$\frac{1}{6} \cdot \left( \frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{2}{5} \right) = 0.23$$

**Precision-IA @ 10**

$$\frac{1}{6} \cdot \left( \frac{2}{10} + \frac{2}{10} + \frac{2}{10} + \frac{2}{10} + \frac{2}{10} + \frac{2}{10} \right) = 0.2$$

## Q3

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

a) **Reciprocal Rank**
*(Divisions are based on the rank in each result set)*

$$R(a) = \frac{1}{\frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{1}} = 0.44$$

$$R(b) = \frac{1}{\frac{1}{1} + \frac{1}{1} + \frac{1}{1}} = 0.33$$

$$R(c) = \frac{1}{\frac{1}{4} + \frac{1}{2} + \frac{1}{2}} = 0.80$$

$$R(d) = \frac{1}{\frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3}} = 0.75$$

$$R(e) = \frac{1}{0 + 0 + 0 + \frac{1}{4}} = 4$$

$$R(f) = \frac{1}{0 + \frac{1}{4} + 0 + 0} = 4$$

$$R(e) = R(f) > R(c) > R(d) > R(a) > R(b)$$

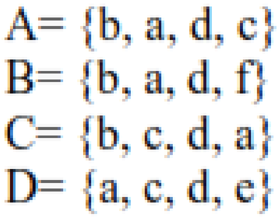
$$b > a > d > c > e = f$$

### b) Borda Count Method

The highest rank individual (in an n-way vote) gets n votes and each subsequent gets one vote less (so #2 get n-1 ...).

$BC(a) = BC_A(a) + BC_B(a) + BC_C(a) + BC_D(a) = 3 + 3 + 1 + 4 = \mathbf{11}$
$BC(b) = BC_A(b) + BC_B(b) + BC_C(b) + BC_D(b) = 4 + 4 + 4 = \mathbf{12}$
$BC(c) = BC_A(c) + BC_B(c) + BC_C(c) + BC_D(c) = 1 + 3 + 3 = \mathbf{7}$
$BC(d) = BC_A(d) + BC_B(d) + BC_C(d) + BC_D(d) = 2 + 2 + 2 + 2 = \mathbf{8}$
$BC(e) = BC_A(e) + BC_B(e) + BC_C(e) + BC_D(e) = 0 + 0 + 0 + 1 = \mathbf{1}$
$BC(f) = BC_A(d) + BC_B(d) + BC_C(d) + BC_D(d) = 0 + 1 + 0 + 0 = \mathbf{1}$

Hence the result is **b>a>d>c > e=f**



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

### c) Condorcet Method

| * | 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 | - |

| * | 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 |

Final ranking of documents: **b>a>c=d > e=f**

## Q4

**a)**

File size = N * F

Each object is assigned with 1024 bits.

Filesize= 1024 * 512 000 /8 /1024= 64 000 Kbytes = 62.5 Mb

**b)**

Each memory element holds the bits of associated columns of object signatures.

There are 512000 objects with 1024 bits long.

Filesize = F*N = 1024 * 512 000 /8 /1024= 64 000 Kbytes = 62.5 Mb

## Q5

Page size = 0.5 Kbytes

1024 bit signatures

512000 objects

0.5 Kbytes / 1024 bits = 4 signatures in each page

Sequential signature method requires all pages to be accessed.

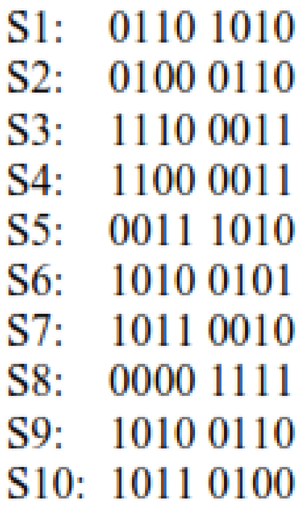
512000 / 4 = **128 000 pages** needs to be accessed in SS.

As long as the page signatures comply with the query signatures, the page must be accessed. As the query is mostly 0, a high portion of the pages will be accessed. In the worst case, all 128000 will be accessed.

In BS, there are 1024 memory objects / 4 = **256 pages**

Since there are 5 positive bits (1s) 5 of these pages will be accessed in the next iterations.

## Q6



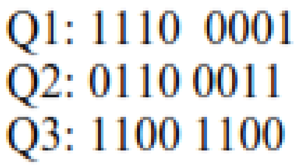
S1: 0110 1010<br>
S2: 0100 0110<br>
S3: 1110 0011<br>
S4: 1100 0011<br>
S5: 0011 1010<br>
S6: 1010 0101<br>
S7: 1011 0010<br>
S8: 0000 1111<br>
S9: 1010 0110<br>
S10: 1011 0100

**a)**

**K=2**

| **00** | **01** | **10** | **11** |
|---|---|---|---|
| S5<br>S8 | S1<br>S2 | S6<br>S7<br>S9<br>S10 | S3<br>S4 |

**b)**



Q1: 1110 0001<br>
Q2: 0110 0011<br>
Q3: 1100 1100

**i)** & AND first 2 bits of each query with partition representative bits.

| | K=1 | K=2 | K=3 |
|---|---|---|---|
| Q1 = 1110 | 1 (1) | 1 (11) | 2 (110, 111) |
| Q2 = 0110 | 2 (0.1) | 4 (00,01,10,11) | 8 (011) |
| Q3 = 1100 | 1 (1) | 1 (11) | 2 (110.111) |

| | **PAR (k=1)** | **PAR k=2** | **PAR k=3** |
|---|---|---|---|
| **Q1** | ½ | ¼ | 2/8 |
| **Q2** | 2/2 | 4/4 | 8/8 |
| **Q3** | 4/2 | 1/4 | 2/8 |

**ii)** Turnaround time = completion time –arrival time.
Each query takes 1 unit time.
In sequential processing all queries arrives at time 0.
K=2 all queries arrives at t=0

| Q1 | Q2 | Q2 | Q2 | Q2 | Q3 |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |

| | Sequential | Parallel |
|---|---|---|
| Q1 | 1 | 1 |
| Q2 | 2 | 2 |
| Q3 | 6 | 3 |

Sequential Processing Avg. Turnaround Time = 1 + 3 +4 /3 = /2.66 tu

Parallel Avg. Turnaround Time: 1+2+3/3 = 2 tu
Parallel Speedup ratio =3/2 = 1.5

## Q7

a) In EPP (extended prefix partitioning) the key length is chosen to be the shortest prefix which contains a predefined number of zeros described by z.

For Z=2 partition structure is as follows:

| **Partition** | **Signatures** |
|---|---|
| P1 | **1110 0**001 (Q1) |
| P2 | **0110** 0011 (Q2) |
| P3 | **1100** 1100 (Q3) |

b) In FKP (fixed key partitioning) we examine each of the consecutive no overlapping k-substrings of a signature and selects the leftmost substring that has the least amount of 1s.

For k=2 partition structure is as follows:
Q1 = 11 10 **00** 01
Q2 = 01 10 **00** 11
Q3 = 11 **00** 11 00

| **Partition** | **Signatures** |
|---|---|
| P1 | 11 10 **00** 01<br>01 10 **00** 11 |
| P2 | 11 **00** 11 00 |

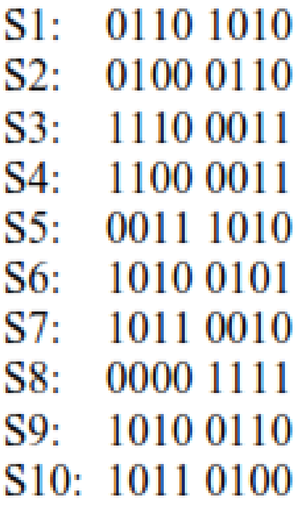
c)

For signature query Q1 since the prefix of the query is ‘11100’ will have to access 1 partition for EPP and 2 partitions for FKP

For signature query Q2 since the prefix of the query is 0110 we have to access 2 partitions (p1 , p2)

For signature query Q3 because of the prefix we can access only P1 and P3 of EPP, but we access 1 partitions of FKP.

## Q8



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

Blocksize = 3
LoadFactor = 2/3
When we reach LF = 2/3 we can add 2 more bits and update the signature file.

Bv = 0, h=1

| | | | |
|---|---|---|---|
| 0 | S1 | S2 | |
| 1 | S3 | S4 | |

**Lf = 4 / (2*3) = 2/3 insert 2 more**

| | | | |
|---|---|---|---|
| 0 | S1 | S2 | S5 |
| 1 | S3 | S4 | S6 |

**Bv = 1, h =1**

| | | | |
|---|---|---|---|
| 01 | S6 | | |
| 1 | S3 | S4 | |
| 10 | S1 | S2 | S5 |

**Lf = 6 / (6*3) = 2/3 insert 2 more**

| | | | |
|---|---|---|---|
| 01 | S6 | | |
| 1 | S3 | S4 | S8 |
| 10 | S1 | S2 | S5 |

**Bv = 0, h =1**

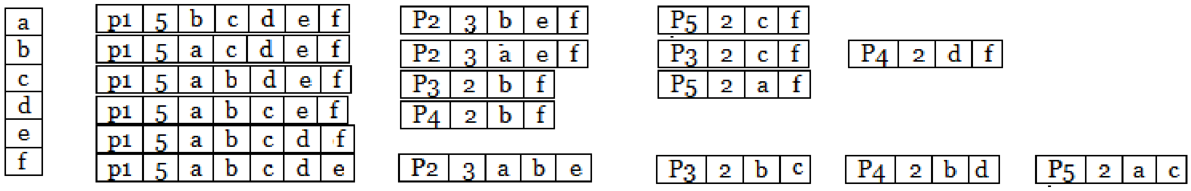
| | | | | | |
|---|---|---|---|---|---|
| 00 | S10 | | | | |
| 01 | S6 | | | | |
| 10 | S1 | S2 | S5 | S7 | S9 |
| 11 | S3 | S4 | S8 | | |

**Lf = 4 / (4*3) = 1/3 insert 1 more (S8 inserted)**

| | |
|---|---|
| Q1 | 1110 0001 |
| Q2 | 0110 0011 |

Since las two bits of Q1 is 01 we need to access 01 and 11 and for Q2 we need to access page 11.

## Q9



In ranked key method, least frequent term in user profiles are more likely to appear frequently in documents.