ACTO1 – SAR

(10/04/2018)

Apellidos y Nombre:.....

(IMPORTANTE: todos los cálculos se mostrarán truncados a dos decimales)

1)) Sea una colección de documentos con 40 documentos. Se realizan dos consultas (Q1 y Q2) para las cuales hay 10 y 12 documentos relevantes, respectivamente.
Un sistemas de recuperación de información devuelve los resultados indicados en la fila Relevante de las tablas Q1 y Q2. Se pide:

a) Calcular la eficacia (Precisión, Recall y la F-medida con β=1) para cada consulta. **(0,2 puntos)**

Consulta	Precisión	Recall	F-medida
Q1	5/12=0,41	5/10=0,5	0,45
Q2	4/12=0,33	4/12=0,33	0,33

b) Completar las Tablas de Precision y Recall (expresando la operación de división realizada y el resultado en decimales, p.e. 2/3 = 0,66) e Interpoladas. **(0,6 puntos)**

Tabla Precision&Recall Reales

Q1	1	2	3	4	5	6	7	8	9	10	11	12
Relevante	Yes	Yes	Non	Yes	Non	Non	Yes	Non	Non	Non	Yes	Non
Precisión	1	1	2/3	3/4	3/5	3/6	4/7	4/8	4/9	4/10	5/11	5/12
Recall	0,1	0,2	0,2	0,3	0,3	0,3	0,4	0,4	0,4	0,4	0,5	0,5

Q2	1	2	3	4	5	6	7	8	9	10	11	12
Relevante	Yes	Non	Non	Non	Yes	Non	Yes	Yes	Non	Non	Non	Non
Precisión	1	1/2	1/3	1/4	2/5	2/6	3/7	4/8	4/9	4/10	4/11	4/12
Recall	0,08	0,08	0,08	0,08	0,16	0,16	0,25	0,33	0,33	0,33	0,33	0,33

Tabla Precision&Recall Interpoladas

Precisión Q1	1	1	1	3/4	4/7	5/11	0	0	0	0	0
Precisión Q2	1	4/8	4/8	4/8	0	0	0	0	0	0	0
Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0

c) Calcular la Precisión Media de cada consulta y la MAP. **(0,2 puntos)**

Pmedia(Q1)=(1+1+3/4+4/7+5/11+0+0+0+0+0)/10=0,37

Pmedia(Q2)=(1+2/5+3/7+4/8+0+0+0+0+0+0+0+0)/12= 0,19

MAP=(0,37+0,19)/2=0,28

2) Considérese una colección de 1.000 documentos entre los cuales se encuentran los siguientes:

Doc1: **programmers** build **computer software**

Doc2: most **software** has **bugs**, but good **software** has less **bugs** than bad **software**

Doc3: some **bugs** can be found only by executing the **software**, not by examining the source **code**

Los términos a considerar se han indicado en negrita.

Se pide calcular la similitud coseno entre la consulta "computer software programmers" y cada uno de los documentos (esquema de pesado lnc.ltc). En la tabla se indica el *df* de cada término considerado. **(1 punto)**

Term			Consulta				Doc1				Doc2				Doc3			
	df_t	idf_t	$f_{t,q}$	tf_{tq}	$W_{t,q}$=tfxidf	L-Norm	$f_{t,d}$	$tf_{t,d}$	$w_{t,d}$=tf_xidf	L-Norm	$f_{t,d}$	$tf_{t,d}$	$w_{t,d}$=tf_xidf	L-Norm	$f_{t,d}$	$tf_{t,d}$	$w_{t,d}$=tf_xidf	L-Norm
bugs	50	1,3	0	0	0	0,00	0	0	0	0,00	2	1,3	1,3	0,66	1	1	1	0,57
code	20	1,69	0	0	0	0,00	0	0	0	0,00	0	0	0	0,00	1	1	1	0,57
computer	100	1	1	1	1	0,45	1	1	1	0,57	0	0	0	0,00	0	0	0	0,00
programmers	20	1,69	1	1	1,69	0,76	1	1	1	0,57	0	0	0	0,00	0	0	0	0,00
software	100	1	1	1	1	0,45	1	1	1	0,57	3	1,47	1,47	0,74	1	1	1	0,57

Esquema de pesado lnc.ltc:

- para los **documentos** log-pesado, no idf y normalización coseno;
- para la **consulta** log-pesado, idf y normalización coseno.

Similitud coseno(consulta,Doc1)= 0.93= 0+0+(0.45x0.57)+(0.76x0.57)+(0.45x0.57)

Similitud coseno(consulta,Doc2)= 0.33=0+0+0+0+(0.45x0.74)

Similitud coseno(consulta,Doc3)= 0.25=0+0+0+0+(0.45x0.57)

3) Los posting lists implementados con skip pointers, ¿son útiles para las operaciones?:

a) t1 and t2

b) t1 or t2

c)(t1) and (not t2)

siendo t1 y t2 los términos que se buscan. Justifica la respuesta. **(0,5 puntos)**

a) Es útil para la operación **AND** porque al buscar sólo coincidencias evita comparaciones que se saltan con los skip pointers.

b) Damos como buenos los dos siguientes razonamientos :
   - No es útil para la operación OR porque deben incluirse todos los documentos de ambas listas (evitando repeticiones).
   - Es útil si el algoritmo recorre las sublistas entre skips y vuelca los IDs en el resultado final. Con ello se ahorrarían comparaciones, aunque no recorridos.

c) En este caso podrían ser útiles los skip pointers de t2 ya que permitirían acceder rápidamente al documento de t1 que queremos saber si existe en t2. Se podrían añadir skips en la posting de t1 si el algoritmo recorre las sublistas entre skips en las postings de t1 y vuelca los IDs en el resultado final.

4) Realizar la inserción de los elementos (29, 33, 25, 7, 15, 36) en una tabla hash de tamaño B=11, con función hash H(x)= x MOD B, y con estrategia de redispersión que usa una 2ª función hash

$h_i(x) = (h_{i-1}(x) + k(x))$ MOD B siendo $k(x) = (x \text{ MOD } (B-1)) + 1$ **(0,5 puntos)**

0	33
1	
2	
3	25
4	7
5	
6	36
7	29
8	
9	
10	15