Examen RI Julio 2017-Soluciones

Recuperación da Información (Universidade da Coruña)

Examen **Recuperación de Información** Julio 2017

Apellidos: _____Nombre: _____
Examen sin libros, apuntes, ni dispositivos electrónicos.
Tiempo: 2h. 15m.

1. (1 punto) Considere relevancia gradual (0, 1, 2; 0 es No Relevante) y una query con un total de 5 relevantes y un ranking (de izquierda a derecha las posiciones del ranking son de la 1 a la 10) donde la relevancia de los documentos es la que se muestra:

   0 2 2 0 1 2 1 0 0 0

   a) Compute MAP con corte 10 y relevancia binaria
   Respuesta: 0.63

MAP = (½ + ⅔ + 3/5 + 4/6 + 5/7)/5 = (0.50 + 0.67 + 0.60 + 0.67 + 0.71) / 5 = 0.63

   b) Compute F1 (F con beta=1) con P y R computados con corte 10 y relevancia binaria
   Respuesta: 0.67

F1= 2PR / (P+R) = 2 * 0.5 * 1 / (0.5 + 1) = 2/3

   c) Compute NDCG@10 con relevancia gradual y la siguiente formulación de DCG

$$DCG@p = rel_1 + \sum_{i=2}^{p} (rel_i / \log_2 i)$$

   Respuesta: 0.78

Dado el ranking

0 2 2 0 1 2 1 0 0 0

La secuencia rel1, reli/$\log_2$i con i=2,...10 es:
0 2 1.27 0 0.43 0.78 0.36 0 0 0
Por tanto DCG@p con p=1,...10 es
0 2 3.27 3.27 3.7 4.48 4.84 4.84 4.84 4.84

Para el ranking ideal:
2 2 2 1 1 0 0 0 0 0
La secuencia rel1, reli/$\log_2$i con i=2,...10 es:
2 2 1.27 0.5 0.43 0 0 0 0 0
Por tanto DCG@p con p=1,...10 es
2 4 5.27 5.77 6.2 6.2 6.2 6.2 6.2 6.2

Por tanto NDCG@10= 4.84/6.2 = 0.78

2. (1 punto) Considere un grafo web con cuatro páginas. De 1 sale un enlace a 1 y otro a 4; de 2 sale un enlace a 2 y otro a 4; de 3 sale un enlace a 3 y otro a 4; de 4 sale un enlace a 4. Compute el Page Rank después de dos iteracciones por el método de potencia y suponiendo un estado inicial con una distribución uniforme. Se considera también un teleporting del 20%.

Respuesta: 0.11 0.11 0.11 0.67

La matriz de adyacencia es:
1 0 0 1
0 1 0 1
0 0 1 1
0 0 0 1
y la matriz de transición de probabilidad MTP es:
0.5 0 0 0.5
0 0.5 0 0.5
0 0 0.5 0.5
0 0 0 1

la matriz de transición de probabilidad con teleporting 20%, MTPalfa0.2 es
MTPalfa0.2 = 0.8 x MTP + (0.2/4) x [1s]

donde [1s] es la matriz 4 x 4 con todos los elementos de valor 1

MTPalfa0.2=

0.45 0.05 0.05 0.45

0.05 0.45 0.05 0.45

0.05 0.05 0.45 0.45

0.05 0.05 0.05 0.85

(0.25 0.25 0.25 0.25) x MTPalfa0.2= (0.15 0.15 0.15 0.55)

(0.15 0.15 0.15 0.55) x MTPalfa0.2=(0.11 0.11 0.11 0.67)

3) (1 punto) Considere los documentos d1, d2 y d3 cuyo contenido se muestra
d1: Se aproxima verano templado
d2: Se aproxima verano lluvioso
d3: Se aproxima otoño

y la query q: verano templado
Considere el modelo de espacio vectorial con esquemas de pesado idf $\log_{10}$ en la query, raw tf x idf $\log_{10}$ en los documentos, con normalización euclídea sólo en los documentos, y similaridad computada con el producto interior de los vectores. Compute la similaridad de la query con cada uno de los documentos:

Respuesta: d1 0,51; d2 0,21; d3 0

los idf log10 se calcula con log10(3/df(t))
para los distintos términos resulta el idf log que aparece en la segunda columna

Los pesos de los términos de la query aparecen en la tercera columna. Obviamente los términos que no aparecen en la query tienen un peso cero, es decir realmente es (tf en la query x idf log10 del término)

En la cuarta columna aparece el raw tf del término en el documento.
En la quinta columna aparece el raw tf del término en el documento multiplicado por el ifd log10 del término. Una vez calculados estos pesos se tiene que normalizar por la norma del vector, por ejemplo para d1 la norma euclídea de d1 es sqrt($0^2+0^2+0.18^2+0.48^2$) = 0.51. Finalmente los pesos del término en el documento que aparecen en la quinta columna se dividen por la norma euclídea así calculada, y se obtienen los valores de las sexta columna.

Por últimos se calcula el producto interior con los vectores de la tercera y sexta columna.

Para D1 Se aproxima verano templado

t	Idf log10	Query weights	Raw tf en D1	Raw tf x idflog10	D1 weights
se	0	0	1	0	0
aproxima	0	0	1	0	0
verano	0.18	0.18	1	0.18	0.18/0.51=0.35
templado	0.48	0.48	1	0.48	0.48/0.51=0.94
lluvioso	0.48	0	0	0	0
otoño	0.48	0	0	0	0
				|D1| = 0.51	

sim(q, D1) = 0.18 x 0.35 + 0.48 x 0.94 = 0.51

Para D2 Se aproxima verano lluvioso

t	Idf log10	Query weights	Raw tf en D2	Raw tf x idflog10	D2 weights
se	0	0	1	0	0
aproxima	0	0	1	0	0
verano	0.18	0.18	1	0.18	0.18/0.51=0.35
templado	0.48	0.48	0	0	0
lluvioso	0.48	0	1	0.48	0.48/0.51=0.94
otoño	0.48	0	0	0	0
				|D2| = 0.51	

sim(q, D2) = 0.18 x 0.35 = 0.06

Para D3 Se aproxima otoño

t	Idf log10	Query weights	Raw tf en D3	Raw tf x idflog10	D3 weights
se	0	0	1	0	0
aproxima	0	0	1	0	0
verano	0.18	0.18	0	0	0
templado	0.48	0.48	0	0	0
lluvioso	0.48	0	0	0	0
otoño	0.48	0	1	0.48	1
				|D2| = 0.48	

Sim(q, D3) = 0

4) (1 punto) Considere un índice invertido donde se codifican en las listas invertidas los docID, tf y posiciones de los términos en los documentos. Los docID y posiciones se codifican con d-gaps y todo con variable byte encoding. Un término t tiene la siguiente lista invertida_

00000010 10000011 10000011 00000001 10000000 10000001 10000001 10000010 10000001 10000100

Indique en que documentos y posiciones aparece el término t
Respuesta: _____

Al codificar con variable byte encoding, si el primer bit (el bit mas significativo, es decir, el bit a la izquierda al representar así los bytes en papel) es 1, indica que es el último byte del número a codificar, mientras que si el primer bit es 0 indica que no es el último byte del número a codificar. Por tanto el bit mas significativo de cada byte es una marca y sólo se pueden usar los otro 7 bits para codificar los valores de los números. Por tanto al leer esa secuencia de bytes, los números en decimal que se corresponden son:

259 3 128 1 1 2 1 4

el primer documento es docID 259 con tf=3 y el término aparece en las posiciones 128, 129. 130 después aparece en el documento con docID 261 con tf=1 y aparece en la posición 4

5) (1 punto) En el modelo de RI de Language Models

a) El ranking producido por P(D|Q) ¿es siempre igual al producido por P(Q|D)? Marque la respuesta correcta y de la razón:

Si / No

$P(D|Q) =^{rank} P(Q|D) \times P(D)$

P(Q|D) es el Query Likelihood que es lo que estimamos con Language Models. Entonces ambos son iguales a efectos del ranking producido (el orden en el que aparecen los documentos es el mismo) si el prior del documento P(D) es uniforme, si no es así no hay garantía de que ambos rankings sean iguales.

b) Explique como se computa el Query Likelihood P(Q|D), primero sin suavización y después con suavización de Jelinek-Mercer y explicando también por qué es necesaria.

El Query Likelihood para unigram Language Models y suponiendo independencia de términos se computal tal y como viene al final de las slide 30 del tema Retieval Models. Después se explica que se puede estimar la probabilidad de cada término de la query en el documento con estimación MLE pero eso tiene el problema de que si un término del documento no aparece en la query, la probabilidad estimada es cero. La suavización resuelve ese problema y es explicada en esas slides, el text book y en las clases. La suavización lleva también a tener un mejor modelo de lenguaje del documento que con estimacion MLE y produce mejor calidad de ranking aúnque no tuviésemos el problema mencionado.