Ranking Retrieval

Profesor Heider Sanchez

P1. Normalización de la longitud:

Se tiene la siguiente tabla en donde se muestra un conjunto de cuatro términos y su matriz de conteo (frecuencia) tanto para la consulta como para ambos documentos Doc1 y Doc2.

		Frequency (tf)		
i	term	Q	Doc1	Doc2
1	affection	115	15	40
2	jealous	10	5	0
3	gossip	2	20	22
4	wuthering	0	25	0

Se le pide aplicar ambas técnicas de scoring:

a) El score se resuelve como la sumatoria del *log-frecuency weigth* de los términos comunes.

$$Score1(Q,D) = \sum_{i \in Q \cap D} qi \cdot di$$

En donde, $qi = log_{10}(1 + tf_{i,Q})$ y $di = log_{10}(1 + tf_{i,D})$

Nota: no se está haciendo ponderación del idf para simplificar el ejercicio.

b) El score es normalizado por la norma de cada vector.

$$Score2(Q,D) = \sum_{i \in Q \cap D} nq_i \cdot nd_i$$

En donde, $nq_i = qi/\|\vec{Q}\|_2$, lo mismo con ndi.

$$\|\vec{Q}\|_2 = \sqrt{\sum_{i=1}^{|Q|} qi^2}$$

Llene el siguiente cuadro, analice los resultados y de una explicación de dicho comportamiento.

	(Q, Doc1)	**(Q, Doc2)**
Score1		
Score2		

P2. TF-IDF:

Dada la siguiente tabla en donde se destribuye los pesos TF-IDF para dos documentos de la colección y para la consulta, se pide calcular el score sin normalizar (dot product) y el score normalizado (cosine similarity) entre Q y cada documento.

Id	Término	Doc2 (TF-IDF)	Doc2 (TF-IDF)	Q (TF-IDF)
T1	Clima	1,452	0	0
T2	Biblioteca	0	2,093	1,345
T3	Universidad	2,122	0	1,453
T4	Alcalá	3,564	0	1,987
T5	España	4,123	4,245	0
T6	Libros	0	1,234	2,133
T7	Geografía	0	0	0
T8	Población	2,342	0	0
T9	Electricidad	0	0	0
T10	Ciencia	0	0	0
T11	Social	0	2,345	0
T12	Luz	1,975	0	0
T13	Unamuno	4,543	2,135	3,452
T14	Física	0	0	0
T15	Fluidos	6,134	0	0
T16	Literatura	2,234	3,456	4,234

$$DotProduct(Q, D) = \sum_{i \in Q \cap D} qi \cdot di$$

$$cosine(Q, D) = \frac{\sum_{i \in Q \cap D} qi \cdot di}{\sqrt{\sum_{i=1}^{|Q|} {q_i}^2 \times \sum_{i=1}^{|Q|} {d_i}^2}}$$

En donde qi es el peso tf-idf del termino Ti respecto al documento Q. Lo mismo con di.

Llene el siguiente cuadro, analice los resultados y de una explicación de dicho comportamiento.

	(Q, Doc1)	**(Q, Doc2)**
DotProduct		
cosine		

P3. Implementación (Tarea):

- Implemente el índice invertido usando la recuperación por ranking para consultas de texto libre (la consulta es solo una o más palabras en lenguaje natural).
- Para probar el desempeño de su implementación. Se proveerá una colección de aproximadamente 20mil tweets de Twitter.
- Para construir el diccionario de términos, debe usar el contenido del atributo "text" y filtrar los stopwords encontrados. El docID vendría a ser el Id del tweet.
- Proponga tres consultas y muestre el **top-10 de los tweets** que se aproximan a dicha consulta.
- Analice el performance de su implementación y proponga una la solución algorítmica para el uso de memoria secundaria ante grandes colecciones de datos.

Ejemplo de tweets:


```
[
    {
        "id": 1046263368691023873,
        "date": "Sun Sep 30 05:00:00 +0000 2018",
        "text": "#VotaBien @EstherCapunay  Necesito esa via expresa sur pero yaaaaaa !!!",
        "user_id": 1042868076327231488,
        "user_name": "@AlcantaraYasuri",
        "location": {},
        "retweeted": false
    },
    {
        "id": 1046263372675788800,
        "date": "Sun Sep 30 05:00:01 +0000 2018",
        "text": "RT @PeruanoComunica: Jorge Muñoz era la voz, hasta que vi esta foto. Vitocho el
ultra KeikoAlanista que recibirá ordenes de la #SeñoraK y #…",
        "user_id": 861714162979934209,
        "user_name": "@Emperilluminati",
        "location": {},
        "retweeted": true,
        "RT_text": "Jorge Muñoz era la voz, hasta que vi esta foto. Vitocho el ultra
KeikoAlanista que recibirá ordenes de la #SeñoraK y #AG. Ni que hablar de los otros dos. 👇  NO
PODEMOS DAR NI UN MILÍMETRO DE PASO A KEIKO CON MIRAS AL 2021 https://t.co/DGT8QBw0Zl",
        "RT_user_id": 2977539507,
        "RT_user_name": "@PeruanoComunica"
    },
]
```