Estructuras de Datos 2021-2

Proyecto Final

Profesor: Pedro Ulises Cervantes González confundeme@ciencias.unam.mx
Ayudante: Yessica Janeth Pablo Martínez yessica_j_pablo@ciencias.unam.mx
Ayudante: América Montserrat García Coronado ame_coronado@ciencias.unam.mx
Ayud. Lab: Emmanuel Cruz Hernández emmanuel_cruzh@ciencias.unam.mx
Ayud. Lab: Adrián Felipe Vélez Rivera adrianf_velez@ciencias.unam.mx

Introducción

Un motor de búsqueda sencillo en el que a partir de una consulta q se quiere saber cuál documento de un conjunto de N documentos es el más relevante, consiste en convertir cada documento y la consulta en vectores de términos para posteriormente analizar cuál documento d es más similar a q.

Para la elaboración de este proyecto se le asignará un peso TF-IDF a cada documento, que es el producto de una función de la frecuencia del término por una función del inverso de la frecuencia en los documentos. Dicho de otra forma, la función TF registra la relevancia de cierto término en un documento en específico, mientras que IDF registra la relevancia del término en el conjunto total de documentos.

Sus fórmulas correspondientes son:

$$TF(t,d) = \begin{cases} log_2(f_{t,d}) + 1 & \text{si } f_{t,d} > 0 \\ 0 & \text{en otro caso} \end{cases}$$

$$IDF(t) = \begin{cases} log_2 \dfrac{N+1}{N_t} & \text{si } N > 0 \text{ y } N_t > 0 \\ 0 & \text{en otro caso} \end{cases}$$

Donde $f_{t,d}$ es el número de ocurrencias de t en el documento d, N_t es el número de documentos que contiene al término t y N es el número total de documentos.

Para saber qué tan relevante es un documento d respecto a nuestra consulta q, se usa la función de similitud:

$$sim(d,q) = \frac{\sum_{t \in q} TF(t,d) \cdot IDF(t)}{\sqrt{\sum_{t \in d} (TF(t,d) \cdot IDF(t))^2}}$$

Nota: Al final de este documento viene un ejemplo de cómo usar las fórmulas.

Sobre el proyecto

El alumno implementará un motor de búsqueda que utilice TF-IDF para que al ingresar una consulta q, le devuelva al usuario los 10 documentos más relevantes ordenados de mayor a menor relevancia.

Se deberá contar además con un historial de búsquedas para saber todo lo que el usuario ha buscado. En el historial solo deberán estar las consultas que se han hecho, no los resultados de las búsquedas.

Por último, su programa debe incluir un caché de las últimas 10 búsquedas. En el caché se guardará el resultado de las últimas búsquedas para no tener que recalcular sus documentos más relevantes.

El proyecto deberá ser **robusto**.

Datos de entrada

- La ruta del directorio donde están los archivos de texto que utilizará el motor de búsqueda.

  Esta ruta se pedirá al iniciar el programa solamente. Se cargarán **todos** los archivos de texto (.txt) dentro del directorio.

  Como es probable que el proceso de carga pueda ser algo lento si son muchos archivos o si estos son muy largos, se sugiere mostrar una barra de progreso o un porcentaje que le indique al usuario cómo va el avance de esta tarea.

- Las consultas. Cadenas de no más de 200 caracteres.

  El programa debe permitir hacer tantas consultas como sea necesario.

Datos de salida

- Los 10 documentos más relevantes para la consulta. Se debe devolver el nombre de los archivos ordenados de mayor a menor según la similitud con la consulta.

  No devolver archivos que tengan similitud 0 con la consulta.

  Es posible que se devuelvan menos de 10 documentos si no hay suficientes de ellos que tengan una similitud mayor a 0 con la consulta.

- El historial de búsqueda en caso de solicitarlo.

  Devolver solo las consultas que se han hecho, no los resultados de las búsquedas.

Sobre la implementación

No debe haber distinción entre mayúsculas y minúsculas ni entre caracteres acentuados y no acentuados.
Se deben ignorar caracteres especiales como ¿ ¡ ? ! , . ; :
Para este proyecto no es necesario usar interfaz gráfica, pero si la agregan habrá calificación adicional.

Entrega

- El proyecto sigue los mismos lineamientos de entrega que las prácticas.
- La elaboración del proyecto puede ser en parejas.
- **NO INCLUIR** los archivos de texto que usa el motor de búsqueda.
- Incluir un pdf en el que expliquen por cada estructura de datos que usaron: en qué parte del proyecto fue usada y por qué la escogieron.

Ejemplo

Considera los siguientes documentos:

Documento	Contenido
$d1$	Do you quarrel, sir?
$d2$	Quarrel sir! no, sir!
$d3$	If you do, sir, I am for you: I serve as good a man as you.
$d4$	No better.
$d5$	Well, sir.

La frecuencia de los términos sería:

Documento	Parejas $(t,\ f_{t,d})$
$d1$	$(do, 1), (you, 1), (quarrel, 1), (sir, 1)$
$d2$	$(quarrel, 1), (sir, 2), (no, 1)$
$d3$	$(if, 1), (you, 3), (do, 1), (sir, 1), (i, 2), (am, 1), (for, 1), (serve, 1), (as, 2), (good, 1), (a, 1), (man, 1)$
$d4$	$(no, 1), (better, 1)$
$d5$	$(well, 1), (sir, 1)$

Dado que hay 5 documentos en total y el término "sir" aparece en 4 de ellos, el IDF para"sir" es:

$$IDF(sir) = log_2(\frac{6}{4}) \approx 0.58$$

El TF para el término "sir" en el documento $d2$ dado que aparece dos veces en él es:

$$TF(sir, d2) = log_2(2) + 1 = 1 + 1 = 2$$

Por lo tanto, el valor TF-IDF para "sir" con $d2$ es el producto de ambos resultados:

$$TF(sir, d2) \cdot IDF(sir) \approx 2 \cdot 0.58 = 1.16$$

Si tenemos la consulta $q =$ "quarrel sir" y queremos saber qué tan similar es $d2$ con q, primero tendríamos que calcular el valor TF-IDF para "quarrel" y "no" en el documento $d2$. Aplicando las fórmulas correspondientes obtenemos que:

$$TF(quarrel, d2) \cdot IDF(quarrel) \approx 1.58$$

$$TF(no, d2) \cdot IDF(no) \approx 1.58$$

Con esto, ahora sí podemos calcular la similitud entre q y $d2$ aplicando la fórmula:

$$sim(d2, q) = \frac{\sum_{t \in q} TF(t, d2) \cdot IDF(t)}{\sqrt{\sum_{t \in d2} (TF(t, d2) \cdot IDF(t))^2}}$$

$$= \frac{1.58 + 1.16}{\sqrt{1.58^2 + 1.16^2 + 1.58^2}} = \frac{2.74}{2.52} = 1.08$$