!moogle, moogle!

> Proyecto de Programación I.

> Facultad de Matemática y Computación - Universidad de La Habana.

> Cursos 2022, 2023.

>- Estudiante Adrián García Santos

>- Grupo C112

El Inicio

Este es un informe donde se explica los algoritmos básicos del proyecto.

Partes del programa

La carpeta `MoogleServer` contiene el código genera la interfaz gráfica y corre mientras la página funciona...

La carpeta `MoogleEngine` contiene código que ejecuta todos los comandos para devolverte los archivos que corresponden con la búsqueda...

Al iniciar el proyecto

He creado una nueva clase denominada "BaseItem" en la cual se almacena toda la información de los documentos a revisar.

Al iniciarse el proyecto se ejecuta el código:

```
public static BaseItem DB = CargarDatos();
```

"CargarDatos()" es un método en el cual inicia un proceso de recopilación de datos muy importante.

> ⚠ La informacion debe ponerse en la carpeta `/moogle-main /Content/`

Recopila los archivos en un array de string "string[] Archivos", después crea otra variable e la que guarda por cada archivo las palabras que contienen.

Ejecutamos una normalización con el método Normalizar para convertir en minúscula sin tildes y quitar caracteres que no sean letras o números.

Creamos una lista e introducimos todas las palabras sin repetirlas.

Procedemos a crear un diccionario en el que guardamos las palabras y les asignamos una posición.

Creamos un for para poder rellenar una matriz en la que por cada archivo aumentamos un contador en la posición de las palabras del diccionario cada vez que se repiten.

Después hacemos lo inverso, creamos un array en que por cada palabra guardamos en cuantos archivos aparecen.

Ahora viene la parte importante:

###TFIDF:

El TFIDF se divide en dos partes TF e IDF. Para un ahorro de espacio yo ya he creado los parámetros solo falta usarlos.

IDF: N/ni Frecuencia en la que aparece una palabra en los archivos en relación con el total de archivos.

N: cantidad de documentos ("Archivos")

ni: cantidad de documentos donde aparece la palabra("Archivosporpalabra")

TF: F/T Frecuencia en la que aparece una palabra en un archivo en relación a la cantidad de palabra en el archivo.

F: cantidad de veces que aparece la palabra en el archivo("RepeticionPorArchivo[Archivo,Palabra]")

TFIDF=TF*IDF.

Básicamente multiplica la frecuencia de una palabra en un documento(cantidad de repeticiones de este en el documento entre el total de palabras del documento(peso en el documento actual)) [vendría a ser el peso en el documento actual] por la frecuencia inversa de la palabra en la colección de documentos(Es el total de documentos sobre la cantidad de documentos en que aparece la palabra...mientras mayor se aproxima a 1 y es 0(si esta en todos no tiene valor)) [vendría a significar la relevancia en la colección, que de aparecer mucho es menor]

Ha sido creada con éxito la base de información sobre la q va a operar la página...

Realizando una búsqueda:

La búsqueda tiene tres elementos importantes: Los operadores, la sugerencia y las palabras a buscar.

Empezaremos por los Operadores:

Hay cuatro operadores que van delante de las palabras para referirse a ellas.

"!": Operador de no aparición.

Recorremos el TFIDF y eliminamos todos los valores que de los documentos que tengan esa palabra.

“^”: Operador de aparición.

Este es lo contrario. Recorremos el TFIDF y eliminamos todos los valores que de los documentos que no tengan esa palabra.

“*”: Importancia.

Vemos cuantos “*” tiene delante la palabra y le sumamos al TFIDF la cantidad de “*” por 10.

“~”: Cercanía.

Uno de los más difíciles de implementar. Revisamos las palabras antes y después del operador.

Revisamos cada documento y guardamos en que posiciones están las dos palabras, restamos las todas las posiciones de la primera con las de la segunda, cogemos el menor valor lo dividimos a la cantidad de palabras del archivo en el que estamos revisando, le hallamos el logaritmo y lo sumamos al TFIDF del documento.

Búsqueda del query y sugerencia:

Hemos terminado con los operadores por lo que podemos normalizar.

Buscamos en nuestro diccionario cual es la palabra que más se asemeja al query, se te dice cual es en la sugerencia y se busca en la base de datos con ella.

Proceso de evaluación:

Creamos una variable scr la cual es encargada de sumar todos los TFIDF del query en los documentos y si es menor que 0 reinicia el ciclo con otro documento.

Genera Snippet

Revisamos si el documento para ver si tiene 1000 caracteres o menos, si es así lo escribimos tal cual sino escogemos los primeros 1000.

Organizer Resultados:

Recogemos todos los títulos, Snippet y scr y los publicamos organizándolos de mayor a menor eliminando a los que tienen valor cero.

Como se va midiendo la que palabra se asemeja más al query:

Esto se resuelve con el Edit Distance:

Como funciona esta cosa rara, pues asi:

Cogemos las palabras del query y las del diccionario en pares.

Entonces hacemos una matriz. A la primera fila y columna le damos valores desde 0 a el número de la posición.

Escogemos la primera columna y analizo que tengo que hacer para que la letra que representa la posición de la columna se transforme en la que representa la de la fila. Si hay que realizar una operación entonces se le agrega uno al menor los valores de arriba, izquierda o diagonal izquierdo

superior. Si no hay que modificar se coge el valor diagonal izquierdo superior. El valor de la última fila y columna es el que se escoge.

De todos los valores de los pares de palabras se escoge el menor y como resultado tenemos la palabra que más se le asemeja al query.

Gracias por leer