Proyecto de programacion de primer año de ciencias de la computacion
de la Facultad Matcom de la Universidad de La Habana.

Nombre:Ernesto Castellanos Vasquez
Grupo:C 121

Para ejecutar el proyecto escribir en la consola de LINUX desde el directorio de moogle-main el comando:
- bash makefile

#Operadores:
-Sintaxis:

<operador>palabra

! - colocado antes de la palabra indica que esta no debe aparecer.
^ - colocado antes de la palabra indica que esta debe aparecer .
* - colocado antes de la palabra indica que esta posee mayor valor entre mas * se coloquen.
~ - colocado entre dos palabras indica que entre mas cerca aparecen mas valor tiene el documento y si ambas no aparecen ese documento no es devuelto.

La aplicacion Moogle es un buscador creado en el lenguaje de programacion C# con NET Core 6.0 cuyo objetivo es buscar inteligentemente en el contenido de un conjunto de archivos de texto para esto utiliza el criterio de la similitud cosenica entre vectores de tf-idf de cada documento el tf-idf es una medida de la relevancia de una palabra en cada documento de un conjunto que permite junto con la similitud cosenica dar un valor de la similitud entre una busqueda y un documento, con la formula :

Cada sumatoria representa lo que se conoce como magnitud.
q: matriz de tf-idf de la query.
d: matriz de tf-idf del documento.

$$\cos = \frac{\sum_{i=1}^{n} Q_i * D_i}{\sqrt{\sum_{i=1}^{n} Q_i^2} * \sqrt{\sum_{i=1}^{n} D_i^2}}$$

ademas de el codigo incluido inicialmente el programa utiliza las siguientes clases

Loader:
Posee el metodo mainLoader que solo se ejecuta una vez al iniciar el proyecto y rellena un array de nombre matrix, que contiene diccionarios de pares key = palabra y value = lista de objetos de tipo Position el length de esta lista es la cantidad de ocurrencias(tf) esto se realiza de esta manera ya que los documentos se deben acceder de forma secuencial y se conoce su cantidad pero la lista de posiciones se accede segun la palabra a la que corresponde y no se conoce cuantas palabras son y ademas de esto guarda el nombre y ruta de cada documento y la matriz de pesos segun palabra y documentos en el array weight como un array de diccionarios por la misma razon anterior(los documentos se acceden secuencialmente y las palabras segun sea cada una).

Moogle:
Esta clase estaba incluida anteriormente pero fue modificada a grandes rasgos.
En primer lugar el metodo Query recibe un string que representa la busqueda del usuario luego primero para cada palabra en la busqueda comprueba si se repite y almacena este valor
y despues recorre cada documento y calcula su score mediante una llamada al metodo score de la clase Score ,si la palabra existe en el guarda esa linea como el snippet de ese documento, si la palabra no se encontro en ningun documento llama al metodo minDist de la clase distancia que retorna la palabra mas parecida, luego elimina los documentos con score nulo y ordena el resto de mayor a menor
y retorna estos resultados al metodo que hizo la llamada.

Score:
Posee el metodo score que calcular el calculo de la similitud cosenica de los documentos con la busqueda mediante los metodos TFIDF(que multiplica el tf por el resultado de la llamada al metodo IDF el cual calcula el IDF de los documentos mediante una llamada al array de diccionarios y array de documentos de la clase Loader) y el metodo sobrecargado M que calcula la magnitud
Tambien posee el metodo que trae el snippet ese documento.

Distance:
Mediante el metodo minDist encuentra la palabra mas parecida a la buscada en los documentos.
Este metodo usa el metodo lev de la misma clase para calcular la distancia de Levenshtein con cada palabra.

Operator:
Calcula el cambio en el score del documento segun la existencia de operadores

ResultComparer:
Se utiliza como criterio de comparacion pra ordenar los documentos