2022/2023

Proyecto de Programación I

FACULTAD DE MATEMATICAS Y COMPUTACIÓN
ALEX DANIEL ARBOALEZ SABATER CC121

"La programación es una carrera en la que uno nunca deja de aprender."

Richard Branson

Que es Moogle?

Es un buscador de documentos del formato *.txt, que dada una busqueda del usuario, devuelve los resultados mas relevantes de dicha busqueda en una base de datos (carpeta).

Como ejecutar el Moogle?

Para poder ejecutar el proyecto:

1 Abrir la carpeta donde se encuentra el proyecto

2 Abrir Terminal / Consola y escribir el siguiente comando:

- La terminal de Linux:

```bash
make dev
```

-Si estás en Windows

```bash
dotnet watch run --project MoogleServer
```

3 - Abrir en el navegador la direccion que ofrece

Como Funciona?

Clase Archivo

{

Se utiliza para representar documentos en un motor de búsqueda. Mediante procesos de calculos logramos extraer informacion de la base de datos que nos seran utiles y esa informacion la devolvemos en un metodo que se llama TF_IDF que contiene la informacion escencial para realizar la busqueda.Para hacer ésto comienza representando cada documento como un vector que almacena solo la informacion necesaria para realizar las búsquedas. Los documentos son normalizados eliminando caracteres complejos de entender por el compilador, los espacios y las tildes, obteniendo solamente la lista de palabras de cada documento. Luego se realiza el cálculo del TF(Termine Frequence) y del IDF(Inverse Document Frequence).

}

Clase Moogle

{

Ésta clase es el núcleo de la funcionalidad del motor de búsqueda y se encarga de coordinar los diferentes componentes del programa. El método "Query" se encarga de realizar la búsqueda. Toma una cadena de consulta como parámetro y devuelve un objeto SearchResult que contiene una lista de elementos de búsqueda y una sugerencia de búsqueda. La búsqueda se realiza utilizando los métodos de las otras clases, incluyendo la normalización de la cadena de consulta y la frecuencia de las palabras, la identificación y el trabajo con los operadores, y el cálculo del puntaje de relevancia de cada documento en función de la consulta. Finalmente, se seleccionan los elementos de búsqueda más relevantes y se devuelven en el objeto SearchResult.Al final, los resultados de la búsqueda se ordenan por puntaje de relevancia y se muestran los 10 primeros resultados (o menos si hay menos de 10 resultados). Mientras compila el programa este se encarga de procesar todas las herramientas necesarias para el correcto funcionamiento de nuestro buscador.

}

Clase Tools

{

Esta contiene una serie de funciones las cuales son necesarias para lograr un mejor ecosistema del proyecto y poder actualizar y mejorar sin dañar la infraestructura del proyecto.

Funciones como:

-ExtraePlabras

-PalabrasSinRepetir

-Direccion

}

Que es Modelo Vectorial?

El modelo vectorial es una técnica utilizada en recuperación de información que se basa en la representación de los documentos y las consultas como vectores en un espacio vectorial. En este espacio, cada dimensión representa un término del vocabulario del corpus (conjunto de documentos) y el valor de cada componente del vector indica la importancia del término en el documento o consulta. Para construir la representación vectorial de un documento, se utiliza una técnica de ponderación de términos, como la frecuencia de término inversa (TF-IDF), que tiene en cuenta la frecuencia de los términos en el documento y en el corpus para asignar un peso a cada término. De esta forma, los términos más importantes para un documento tienen un mayor peso en su representación vectorial. Cuando se recibe una consulta de búsqueda, se representa también como un vector en el espacio vectorial utilizando la misma técnica de ponderación de términos. Entonces, se puede medir la similitud entre el vector de consulta y los vectores de los documentos utilizando una medida de similitud, como la similitud del coseno. La similitud del coseno mide el ángulo entre dos vectores y proporciona una medida de la similitud entre ellos. Cuanto más cercanos sean los vectores, mayor será la similitud del coseno y mayor será la probabilidad de que el documento sea relevante para la consulta.

En resumen, el modelo vectorial es una técnica de recuperación de información que representa documentos y consultas como vectores en un espacio vectorial, donde cada dimensión corresponde a un término del vocabulario del corpus. Esta técnica es útil porque permite medir la similitud entre vectores para determinar la relevancia de los documentos para una consulta dada.

La idea detrás de IDF es que los términos raros son más importantes para la comprensión del contenido de un documento que los términos comunes.

La combinación de TF y IDF se conoce como TF-IDF. TF-IDF es una medida de la importancia relativa de un término en un documento o en una consulta en el contexto de un corpus de documentos. Se calcula multiplicando la frecuencia de término (TF) por la frecuencia inversa de documento (IDF):

En nuestro motor de búsqueda lo utilizamos para medir la relevancia de los documentos en función de las consultas de los usuarios. Se calcula el valor de TF-IDF para todos los términos en la consulta y en cada documento en la colección, y se devuelve una lista de documentos ordenados por su similitud con la consulta.

El objetivo de Moogle! es realizar búsquedas en el interior de varios archivos .txt y en función del contenido de los mismos, mostrar los resultados más relevantes de acuerdo a la búsqueda que usted haya realizado. Para esto, usted debe copiar los archivos .txt a los cuales quiera realizarle la búsqueda en la carpeta Content que aparece en la raiz del proyecto. La cantidad mínima de archivos .txt que el proyecto debe tener en la carpeta Content para funcionar de manera correcta es de 2 archivos. Los cuales ya se encuentran en dicha carpeta. (Siéntase libre de borrarlos y copiar sus propios archivos .txt, el código está preparado para trabajar con cualquier archivo .txt que usted provea, mientras la cantidad mínima de estos sean 2). En cuanto a la cantidad máxima no debería tener ningún problema.

Mi codigo se utiliza mucho el diccionario en C# pues es una estructura de datos que permite almacenar elementos en pares clave-valor. La clave es un valor único que se utiliza para identificar el elemento, mientras que el valor es el elemento en sí mismo.

Hay varias razones por las que podrías querer usar un diccionario en lugar de otro tipo de variable para almacenar elementos:

1. Búsqueda eficiente: Los diccionarios en C# están diseñados para permitir una búsqueda eficiente de elementos por clave. Esto significa que puedes buscar un elemento en el diccionario en tiempo constante, independientemente del tamaño del diccionario.

3. Fácil acceso a los elementos: Los diccionarios en C# proporcionan una forma fácil de acceder a los elementos almacenados por clave. Esto significa que puedes acceder a cualquier elemento en el diccionario simplemente proporcionando su clave.

En resumen, los diccionarios en C# son una estructura de datos muy útil para almacenar elementos en pares clave-valor. Son eficientes en la búsqueda, flexibles en el tipo de elementos que pueden almacenar, y proporcionan un fácil acceso y actualización de los elementos almacenados.

Este documento se deja abierto a próximas actualizaciones pero por ahora este es el resumen del funcionamiento y ecosistema del código…..