Pedro D Perez

Moogle es una aplicación web original que tiene como objetivo buscar de manera inteligente un texto en un conjunto de documentos. Fue desarrollada con tecnología .NET Core 6.0, específicamente usando Blazor como framework web para la interfaz gráfica y en el lenguaje C#. La aplicación se divide en dos componentes fundamentales:

- MoogleServer: es un servidor web que renderiza la interfaz gráfica y sirve los resultados.
- MoogleEngine: es una biblioteca de clases donde está implementada la lógica del algoritmo de búsqueda.

Para correr el proyecto, se debe usar el comando "dotnet watch run --project MoogleServer" en Windows y "make dev" en Linux. En la carpeta "Content" deben aparecer los documentos (en formato .txt) en los que el usuario va a realizar la búsqueda. En la casilla donde aparece "Introduzca la búsqueda", el usuario va a escribir lo que desea buscar y basta con apretar el botón "Buscar" para que Moogle! haga su trabajo.

Clases

En la clase Moogle, se convierte la consulta en un array de String separándola palabra a palabra y luego se convierte en un vector calculándole su TF. Este vector se multiplica término a término por el vector que contiene el IDF de los textos y con este vector resultante se calcula el score de cada texto multiplicándolo por la matriz donde se tienen los TF-IDF de todos los textos.

Se busca el Snippet revisando cada texto relevante y buscando la palabra más relevante de ese texto para la búsqueda y se devuelve una cadena de 20 palabras antes y después de esta. Finalmente, se crea el objeto SearchResult y se ordena según su score con la función Sort de la clase Ordenar y se devuelve este objeto.

En la clase TFIDF, se accede a los textos usando la ruta al contenido. Luego, se crea un diccionario donde se hacen corresponder las palabras diferentes en todos los archivos de texto con el orden en que van apareciendo. Se genera una lista de matrices (TFIDF)

Este sistema de valoración puede ser útil para evaluar la relevancia de palabras en un conjunto de textos ypuede ser usado en diferentes aplicaciones, como motores de búsqueda o análisis de texto.

Funciones Extras
El símbolo ! indica que la aparición de una palabra en un texto debe reducir su valoración. Es decir, si una palabra aparece en un texto, la valoración de ese texto se reduce.

El símbolo ~ se utiliza para guardar las distancias entre dos palabras y aumentar la valoración en función de su cercanía. Es decir, si dos palabras están cerca en un texto, se aumenta su valoración.

El símbolo ^ se utiliza para aumentar la valoración de una palabra en particular y reducirla en los textos donde no aparece. Es decir, si se quiere que una palabra tenga

más peso en la evaluación, se aumenta su valoración y se reduce en los textos donde no aparece.

El símbolo * se utiliza para aumentar la valoración en función de cuántas veces aparece una palabra o un conjunto de palabras en los textos. Es decir, si una palabra o un conjunto de palabras aparecen varias veces en un texto, su valoración se incrementa.

TFIDF Explicación

En la descripción del proyecto se habla sobre los términos TF e IDF, estos serían:
TF: La relación entre la cantidad de palabras de un documento en específico(N) y la cantidad de veces que aparece una palabra en ese documento(n), de la forma: TF=n/N.
IDF La relación entre la cantidad de documentos existentes(D) y la cantidad de documentos que contienen la palabra en cuestión(d): log(1+D/d).
De esta forma al realizar la multiplicación del TF y el IDF, obtendríamos la relevancia de esa palabra en esos documentos.