Moogle!

Luis Daniel Silva Martínez

21 de Julio, 2023

Presentación

¿Alguna vez ha deseado buscar entre todos sus archivos de texto una palabra o frase particular? ¿Ha deseado organizar sus documentos de texto según su relevancia respecto a una búsqueda?

Pues usted debería probar **Moogle!**, el nuevo buscador de texto actualmente sensación entre decenas de estudiantes de la Facultad de Matemática y Computación de la Universidad de La Habana.

Introducción

Moogle! es un proyecto de programación totalmente original cuyo propósito es buscar inteligentemente un texto en un conjunto de documentos. Dada una búsqueda introducida por el usuario, este programa es capaz de leer archivos de texto en formato *.txt* de una colección y devolver los documentos relevantes.

Qué ofrece **Moogle!**?

1. De una colección de archivos *.txt* predefinida y personalizable en la carpeta *Content* del directorio del proyecto, **Moogle!** devuelve los documentos relevantes (en orden descendente según la relevancia, el más relevante al inicio) respecto a una query (búsqueda) introducida por el usuario.
2. Ofrece sugerencias en caso de que los términos de la query no se encuentren en el corpus de documentos pero existan coincidencias con términos semejantes. Especialmente útil en caso de que haya ocurrido algún error ortográfico al ingresar la búsqueda.

Estructura

Moogle! está desarrollada con tecnología .NET Core 6.0, específicamente usando Blazor como framework web para la interfaz gráfica, y en el lenguaje C#. La aplicación está dividida en dos componentes fundamentales:

- MoogleServer: Servidor web encargado de renderizar la interfaz gráfica y sirve los resultados.
- MoogleEngine: Biblioteca de clases que tiene implementada la lógica del algoritmo de búsqueda.

Cómo se Ejecuta?

Para ejecutar el programa puede abrir el archivo *MoogleServer.sln* o ejecutar el comando dotnet watch run –project MoogleServer

TF-IDF

El programa hace uso de un modelo vectorial con el algoritmo TF-IDF (Term Frequency - Inverse Document Frecuency) el cual expresa como vector cuán relevante es una palabra para un documento.

$$TFIDF_{(t,d)} = TF_{(t,d)} \times IDF_{(t)} \tag{1}$$

- t: término
- d: documento

TF

El TF (Term Frequency) de un término es su frecuencia relativa en un documento.

$$TF_{(t,d)} = \frac{tf}{tw} \tag{2}$$

- t: término
- d: documento
- tf: frecuencia del término t en d
- tw: total de términos en el documento d

IDF

El IDF (Inverse Document Frequency) de un término es su relación entre su frecuencia entre los documentos del corpus y el total de documentos.

$$IDF_{(t)} = \log_2 \frac{1+N}{1+df} \tag{3}$$

- t: término
- df: cantidad de documentos que contienen a t
- N: total de documentos en el corpus

Similitud del Coseno

La fórmula de Similitud del Coseno, permite obtener la relevancia del documento respecto a la búsqueda introducida por el usuario, comprobando el coseno del ángulo entre los vectores de cada documento con la query.

$$SimCos(\theta) = \frac{D \cdot Q}{||D||\ ||Q||} \tag{4}$$

- D: vector del documento
- Q: vector de la query (búsqueda)

Sugerencia

En caso de haber encontrado pocos documentos relevantes o haber introducido un término en la query que no se encuentra en la colección, el programa ofrece una sugerencia para una mejor búsqueda. Esto gracias a la fórmula de Distancia de Levenshtein, buscando los términos más similares a la query dentro del corpus.

Especificaciones

- El programa está cableado para encontrar la carpeta *Content* dentro de la carpeta que contiene a todo el proyecto. Para realizar búsquedas entre los documentos *.txt* de su elección, deberá copiarlos a la carpeta *Content* ubicada en el directorio principal del proyecto.
- Para introducir la query puede presionar la tecla `Enter` luego de escribir para que haga la búsqueda sin tocar el botón de `Buscar`.
- Para introducir la sugerencia dada por el programa puede presionarla directamente y llevar a cabo la búsqueda con esa sugerencia como query.