



Proyecto de Programación I. Moogle!

Nombre: Ana Paula González Muñoz

Grupo: 121

El proyecto Moogle! consiste en un sistema de recuperación de información que basa su funcionamiento central en el algoritmo TF-IDF y en la implementación de modelos vectoriales para garantizar búsquedas más eficientes y acertadas. Se emplearon un conjunto de lenguajes de programación para lograr la mejor vinculación entre los elementos que conforman el código. Como principal motor en la creación y el trabajo con los algoritmos, se empleó el lenguaje C#, mientras que la interfaz gráfica se encuentra codificada en una plataforma web.

Para garantizar una edición sencilla y la legibilidad del código desarrollado, se utilizaron en el proyecto las clases: preSearch y Search.

Al ejecutarse el proyecto carga la clase preSearch que se encarga de hacer todos los cálculos pertinentes antes de realizar la búsqueda a través de los métodos:

- *ObtenerRuta*: se encarga de obtener la ruta de cada txt que se encuentran en la base de datos.
- *Normalizar*: se encarga de normalizar los textos (eliminar tildes, espacios, caracteres que puedan hacer ruido)
- *GuardarTxt*: se encarga de guardar los txt normalizados de la base de datos para poder trabajar con ellos.
- *Tfidf*: se encarga de calcular Frecuencia Inversa – Frecuencia Inversa del Documento ($Tf \cdot Idf$) para darle un valor a las palabras según su importancia en la base de datos y en los documentos.

Al introducirse la búsqueda se ejecuta la clase Moogle que se encarga de llamar a los métodos de la clase Search. Esta clase tiene como onjetivo realizar los cálculos pertinentes para determinar que documentos son más relevantes. Para ello se emplean los siguientes métodos:

- *obtener query*: se encarga de cargar y normalizar (sin eliminar los operadores) la búsqueda que introduce el usuario.
- *guardarOperadores*: se encarga de guardar las palabras con operadores (~, ^, *, !) y eliminar los operadores de la búsqueda.
- *Tfquery*: se encargad de calcular la Frecuencia Inversa (Tf) de las palabras de la búsqueda que introduce el usuario.

- *querySinRepetir*: se encarga de eliminar las palabras repetidas de la búsqueda que introduce el usuario
- *queryTfIdf*: previo a la ejecución de este método se eliminan los operadores de la búsqueda para ser utilizada en caso de que sea necesario realizar sugerencias. Esta función se encarga de multiplicar la Frecuencia Inversa (Tf) de cada palabra la búsqueda por la Frecuencia Inversa del Documento (Idf) general de la misma palabra.
- *OperadorDeImportancia*: se encarga de modificar el Tf*Idf de las palabras de la búsqueda que contengan dicho operador.
- *magnitudDeLaQuery*: se encarga de calcular la magnitud (donde A,B...N es el TF*IDF de la búsqueda)
- *diccionarioDeSimilitudDeCoseno*: se encarga de convertir los documentos y la búsqueda del usuario en vectores del plano para saber cuales documentos son mas relevantes. Se divide por partes:
 - vector documento: crea el vector documento multiplicando el TF*IDF de las palabras del documento
 - suma punto: sumatoria de los TF*IDF de la búsqueda y del documento
 - magnitud del documento: se encarga de calcular la magnitud (donde A,B...N es el TF*IDF del documento)
- *Operadores*: se encarga de realizar las operaciones pertinentes según los operadores que tenga la búsqueda
- *similitudDeCosenoOrdenado*: en una lista se guardan los documentos que más coincidencias tienen con la búsqueda del usuario
- *Snippets*: se encarga de mostrar una parte del documento donde se encuentra la palabra con mayor coincidencia de la búsqueda en el mismo.
 - Este método utiliza el resultado de la función *OrdenarQuerySegunTfIdf* que se encarga de ordenar las palabras sin repetir de la query según su importancia. Es un método auxiliar para los snippets
- *DistanciaHamill*: durante la llamada a este método se crea una nueva búsqueda con las palabras sugeridas. Esta función se ejecuta si en la búsqueda hay una palabra que no se encuentra en nuestro content, se encarga de comparar dicha palabra con todas las palabras de la base de datos y de buscar la palabra con menor cambio y la sugiere para realizar una nueva búsqueda.