Informe Escrito

Proyecto Moogle!

Por: Laura Martir Beltrán Grupo: C-113

Estudiante de Ciencias de la Computación de la Universidad de La Habana (Facultad MatCom).

TEMAS:

- Qué es “Moogle!”?
- Clases
- Procesamiento de Datos
- Operadores
- Modelo Vectorial

Que es “Moogle!”?

Moogle! Es un motor de búsqueda el cual dada cierta información ofrecida por el usuario, devuelve todos los documentos relacionados con ella gracias a una biblioteca de datos que tiene almacenados y procesados.

Clases

Para una mejor implementación es necesario crear una nueva clase que formará una parte muy importante del proyecto.

- TXT (documento):

  Características: -Título del documento(title).

-Contenido del Documento(text).

Procesamiento de datos

- Procesamiento de documentos y su contenido:

Para hacer el procesamiento de documentos es necesario crear un directorio donde se guarda toda la información que está en la carpeta Content.

De esta información extraemos todos los archivos ".txt" para luego procesarlos y guardar su título y su contenido en el objeto de clase TXT.

- Procesamiento de las palabras del contenido de los documentos:

Es necesario tener un conjunto de palabras que agrupe a todas estas contenidas en los documentos. Son procesadas de manera tal que no posean signos innecesarios para la búsqueda:

("~","!","@","#","$","%","^","&","*","(",")","_","+","-","=","`","[","]","{","}",";","|",":","'","<",">",",","?","/"),

Cabe mencionar que en este conjunto de palabras no conviene tener ninguna repetida, ya que es poco conveniente para futuros procesamientos.

Operadores

Estos son signos o símbolos que indican cómo se debe manipular ciertos elementos en la búsqueda:

-(*): afecta el score de la palabra que lo precede en la búsqueda y lo aumenta para así darle mas importancia en los archivos por encima de las demás.

-(^): la palabra a la que preceda este símbolo debe estar en todos los documentos que devuelva la búsqueda.

-(¡): la palabra a la que preceda este símbolo no puede aparecer en ningun documento devuelto en la búsqueda.

-(~): se encuentra entre dos palabras del query y define que para todos los documentos de la base de datos, mientra más cerca entre sí estén las palabras que preceden este símbolo, más importancia se le dará a este documento en la entrega de un resultado.

Modelo Vectorial

Antes de trabajar con el query, debemos obtener el peso de cada palabra en nuestra bibliotea en cada uno de nuestros documentos. Para ello utilizamos una matriz bidimensional que contiene valores flotantes que representan el resultado obtenido tras aplicar el cálculo de una fórmula conocida como “Term Frequency * Inverse Document Frequency”. Como no conocemos a ciencia cierta qué numero de columna corresponde a cada palabra y que fila a cada archivo, empleamos diccionarios para mantener bien catalogados nuestros datos en la matriz.

Tras llenar la matriz, comenzamos a trabajar con el query: separamos en palabras y lo llevamos a vector TXT para hallar el peso de cada palabra de este. Posteriormente aplicamos el cálculo de la similitud del coseno entre todos los documentos que contengan al menos una palabra del query y del vector TXT que lo representa. Esto resultará en la obtención de una lista de valores correspondientes a cada documento evaluado, y estos valores son el “score” respectivo de cada uno de ellos.

Paralelamente a este proceso, se ejecuta la obtención de las palabras recomendadas:

Analizamos cada palabra del query y buscamos si se encuentra en nuestra lista. En ese caso, se le asigna como recomendada ella misma. En caso que la palabra no se encuentre entonces busca la más parecida a través de la implementación del Edit-Distance (o distancia de Levenshtein). De esta forma siempre va a devolver la palabra exacta o la más parecida.

Finalmente se aplica el trabajo con operadores sobre la lista de documentos y flotantes obtenida para posteriormente llevar esa lista al tipo de dato SeacrchItem. Aquí la obtención del Snippet se ejecuta respecto a la palabra más importante de la búsqueda que se encuentre en dicho documento. El Snippet es un fragmento de texto que muestra el contexto de la palabra encontrada. Luego de esto solo queda organizar los elementos por Score y devolver en orden de mayor a menor.

Cabe destacar que si la lista de recomendaciones es exactamente igual al query, significa que todas las palabras estaban bien escritas. Sin embargo si no lo es, entonces se mostrará una recomendación con la corrección más apropiada de la palabra.