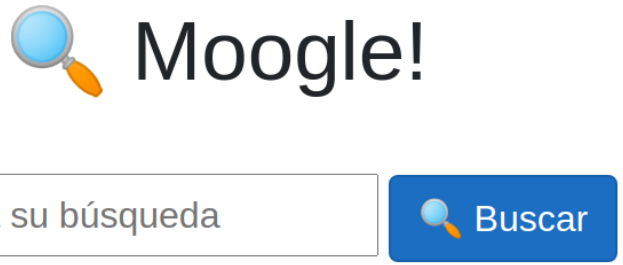
Moogle (Proyecto de Pogramación I)

Laura Alonso Rivero C-122

¿Qué es Moogle?

Es una una aplicación web capaz de encontrar un texto en un conjunto de documentos



El proyecto está estructurado de la siguiente manera:

- Diccionario de palabras (palabra, Diccionario de documentos (docId, DetallesDelDocumento-palabra))
- DetallesDelDocumento-palabra [número de ocurrencias de la palabra en el documento, TF-IDF de la palabra en el documento]
- Doc [id de la ruta, nombre del documento, numero de palabras del documento]
- DirectoryName [id de la ruta, ruta completa del documento]
- Diccionario de Documentos (docId, Doc)

La carga de datos

Los datos son obtenidos a partir de los documentos que se encuentran en la carpeta **‘Content‘** y a partir del método **‘Initialize‘** se cargará la base de datos.

Para cada fichero que se lea se procede a llenar el diccionario **‘wordsDictionary‘** , los valores para esta llave serán otro diccionario llamado **‘docsDictionary‘** donde se encuentra el **TF-IDF** de la palabra en el documento.

TF-IDF (Term Frequency - Inverse Document Frecuency)

$$TF - IDF = \frac{\text{cant de apariciones}}{\text{cant de palabras}} \times \log \frac{\text{total de docs}}{\text{docs con la palabra}} \quad (1)$$

Búsqueda de palabras

Al iniciar la búsqueda el usuario puede elegir una palabra o una frase e incluir en cada palabra operadores tales como:

- ‘¡‘ (exige que la palabra no se encuentre en los documentos devueltos)
- ‘^‘ (exige que la palabra se encuentre en los documentos devueltos)
- ‘*‘ (da mayor importancia a los documentos que contengan esta palabra)

Casos de usos erróneos:

- Si la palabra contiene ‘¡‘ como primer carácter y después le siguen otros operadores
- Si la palabra contiene ‘^‘ como primer carácter y después le siguen otros operadores
- Si la palabra contiene ‘*‘ y después le sigue ‘¡‘

Si se encuentra que alguna palabra contiene ‘¡‘ se eliminan de la lista de búsqueda todos los documentos que la contengan. En cambio, si la palabra posee ‘‘̂ se eliminan todos los documentos que no la contengan

Con estas palabras se procederá a llenar la lista **‘results‘** que devolverá los nombres de los documentos que contengan una o más de estas y los oredenará según el valor del *TF-IDF*

En caso de que no se encuentre la palabra se llenará la lista **‘suggestions‘** que buscará por el método de **‘FindSimilarities‘** las cuatro palabras presentes en el diccionario más similares a esta.

Este método utiliza el algoritmo de Levenshtein (utiliza matrices siendo las filas las letras de la palabra de la query y las columnas las letras de la palabra del diccionario) para dar este criterio de similaridad.

Propuestas para el mejoramiento de la web en el futuro

- Se puede utilizar el algoritmo de Porter a la hora de hacer el diccionario de palabras pues con este se va a eliminar todos los sufijos tales como género, numero, persona, tiempos verbales, diminutivos... y se quedara solamente con las raíces de las palabras lo que reducirá el tamaño del diccionario y hará que la carga de la basa de datos y posterior búsqueda de la query sea más rápida y eficiente.

Propuestas para el mejoramiento de la web en el futuro

- Se puede guardar la data en un fichero estructurado para ahorrar tiempo a la hora de inicializar y recalcularlo solo cada vez que se carguen otros ficheros en la carpeta **'Content'**.
- Se puede realizar una búsqueda exacta de frase de más de dos palabras que estén entre comillas en el criterio de búsqueda.

Gracias por su atención