En el proyecto se debe agregar documentos al content antes de iniciar el proyecto, de ser posible que los documentos estén en ingles (esto es muy importante puesto que así mejora mucho la precision a la hora de buscar, debido al uso de un diccionario con una gran cantidad de palabras en ingles lematizadas llamado SteammedWords.json el cual no puede ser borrado)

Mi proyecto cuenta con 4 lugares principales: MoogleEngine, MoogleServer, MoogleLibrary y Algebra Library. En Moogle Library se encuentran las clases creadas por mi para realizar toda la parte de la lógica de obtener documentos procesarlos y todo lo demás….

En Moogle Library:

La clase Obteiner va a ser la clase que se llama dentro del Moogle server para llevar a cabo todo desde procesar los textos hasta calcular el tf-idf de los mismos. Ademas se encarga de saber si están guardados los json, el deserealizado de los json y saber si se ha cambiado algún archivo txt.

Para obtener los textos el Obteiner llama a la clase Process y esta se encarga de primeramente agregar espacios en blanco al final de cada texto y tenerlos de forma organizada en otra carpeta ContentwithSpaces para una mayor organizacion y ademas poder permitir cambios del usuario en los documentos originales de una major manera, luego se encarga de leer todos los documentos, guardar los nombres y los textos en listas ademas procesa los textos y también los guarda y recorre cada documento para contar las apariciones de cada palabra en cada txt y las apariciones de cada palabra tematizada en cada txt.

Y esta clase Process se encarga de serializar en json los valores de los nombres de los archivos, los textos de cada archivo, los textos procesados, las apariciones de las palabras, las apariciones de las palabras lematizadas, y la ultima fecha en que fueron cambiados los archivos txt antes de procesar todo el proyecto.

Luego la clase Obteiner al ejecutar su constructor es capaz de analizar, primeramente si existen los archivos json con la información necesaria y ademas los textos no se han cambiado y entonces decidir si es necesario o no procesar de nuevo todo.

“Para guardar los datos se utiliza un serializador de json, y para obtenerlos luego se utiliza un deserializador, esto permite no tener que Volver a realizar operaciones cuando no es necesario, pero cuando desde la clase obteiner se verifica que falta algun archive json o que las fechas de ultima vez de update de los documentos de los txt han sido cambiadas o que falta algun archivo o hay alguno nuevo entonces la clase obteiner manda a la clase process a procesar todo de nuevo, pense en solo procesar los archivos nuevos o viejos pero la parte del idf si habria que calcularla de nuevo y era basicamente el mismo tiempo de demora que calcular todo de nuevo”

Cuando es necesario simplemente se ejecuta procesar de nuevo (en caso de ser necesario procesar de nuevo significa que hay que calcular de nuevo el tf-idf puesto que ademas es una de las necesidades la existencia del json del tf-idf para no serializar), si no no se ejecuta procesar y ya.

Se utiliza la siguiente formula de tf-idf:

$$tfidf = (xi/ni)(ln(d + 1/t)$$

xi representa la cantidad de apariciondes de una palabra en un documento

ni representa la cantidad de palabras de dicho document

d es la cantidad de documentos

t es la cantidad de documentos en que aparece la palabra

El tf-idf y las apariciones de cada palabra en cada txt se guarda en una estructura de datos creada por mi la cual llame matrix y se encarga de guardar en una array de diccionarios (lo cual es bastante eficiente) indexando por numero de archivo y por palabra.

Luego se deserializa la información y se guarda en el objeto. El objeto obtiene solo es creado al inicio del proyecto y solo muta o cambia sus propiedades si se detecta mientras se hace una query que algún documento cambio o se quito algún documento o se añadió algún documento.

Luego dentro de la clase CosineSimilarityCalculator se realiza todo lo relacionado a obtener todos los operadores presentes dentro del query y según que tipo de operadores es se agrega a una estructura para procesarlos de mejor manera.

1. A la hora de ver los operadores dentro del cuero si es de tipo * se guarda cuantos * hay antes de la palabra por cada palabra con *
2. Si es de tipo ^ se guarda en una lista de yesWords para comprobar que la palabra aparezca en la query
3. Si es de tipo ! Se agrega a una lista de noWords pero ademas se retira de la query para evitar las apariciones de palabras semejantes
4. Si es de tipo ~ se agrega a un diccionario las palabras a sus lado para verificar que estén cerca

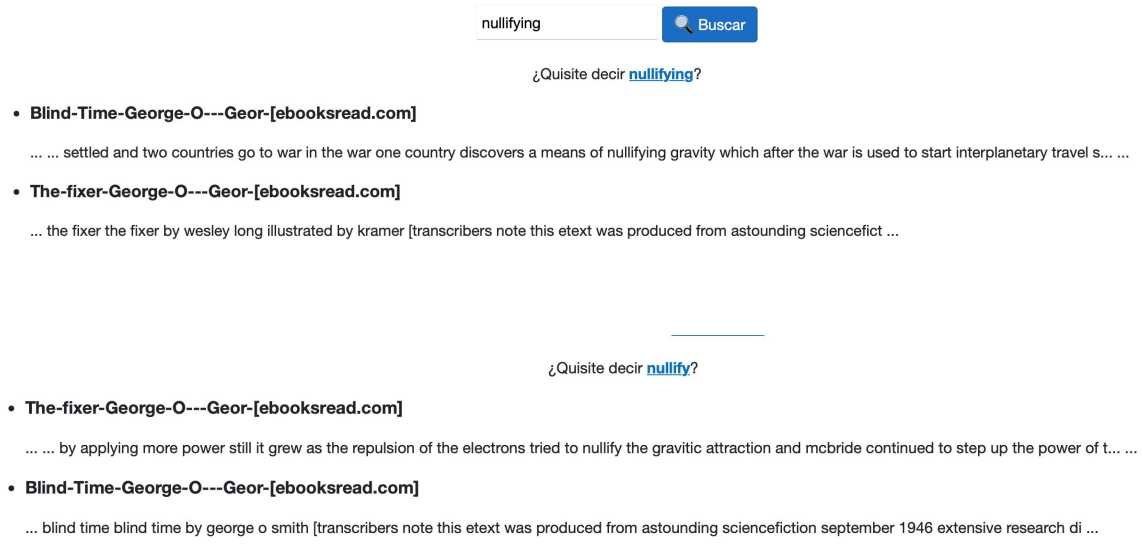
Luego es que se obtiene el query procesado quitando todos esos símbolos.
Luego se cuentan las palabras del query (cuando se están contando las palabras del query se analiza por cada palabra escrita en el query cual es la palabra mas parecida dentro de mi universo de palabras y con esa palabra es con la que se trabaja, esto mediante el uso de distancia de Levinstein), se calculan los dos tf-idf del query el normal y el de palabras lematizadas. Y después de calcular los tf-idf se multiplica por la cantidad de asteriscos mas 1 al valor de cada palabra con aster.

Luego se calcula la semejanza entre el tfidf de la query con el de cada documento mediante el cosine similarity al igual que con los tf-idf de palabras lematizadas pero estos últimos añaden al score solo la mitad.

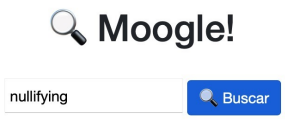
Al final se verifica las apariciones de las yesWords y que no aparezcan las noWords en los documentos o la puntuación es 0, y ademas en cuanto a la distancia entre palabras se calcula iterando por el array y viendo con cual palabra estoy trabajando y busco por la aparición de la otra o de ella misma y si me encuentro la otra palabra calculo distancia y guardo con la nueva que trabajo y el nuevo indice y si me encuentro con la misma solo cambio el indice, luego si esa distancia es mayor o igual int.MaxValue que ademas es el valor por defecto el score del documento es 0, si no se le suma 0.1/ min_distance.

Ademas con el Snippet calculator busco las apariciones de las palabras que ya son las mas parecidas a las de mi universo de palabras y las busco dentro de los textos cuyo score sea diferente de 0 ademas y devuelvo un pedazo de texto que las contenga. Y si lo que pasa es que el documento no contiene apariciones de dichas palabras pero si contiene a alguna palabra tematizada y por eso tiene puntuación pues se devuelve el inicio del texto.

En el proyecto puedes buscar palabras y te da los documentos que contengan esas palabras y ademas documentos que contengan palabras parecidas. Pero le da prioridad a los documentos que contengan a las palabras exactas.



Operador de * :

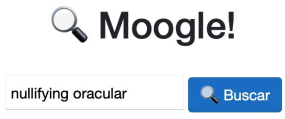


- **Blind-Time-George-O---Geor-[ebooksread.com]**

  ... ... settled and two countries go to war in the war one country discovers a means of nullifying gravity which after the war is used to start interplanetary travel s... ...

- **The-fixer-George-O---Geor-[ebooksread.com]**

  ... the fixer the fixer by wesley long illustrated by kramer [transcribers note this etext was produced from astounding sciencefict ...



- **Gold-and-glory-Grace-Stebbing-[ebooksread.com]**

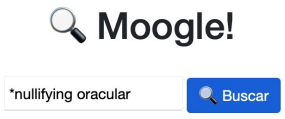
  ... ... _a powerful friend_ come with me and ask no questions such was the oracular order addressed by master pedro to his friend master sancho the mornin... ...

- **Blind-Time-George-O---Geor-[ebooksread.com]**

  ... ... settled and two countries go to war in the war one country discovers a means of nullifying gravity which after the war is used to start interplanetary travel s... ...

- **The-fixer-George-O---Geor-[ebooksread.com]**

  ... the fixer the fixer by wesley long illustrated by kramer [transcribers note this etext was produced from astounding sciencefict ...



- **Blind-Time-George-O---Geor-[ebooksread.com]**

  ... ... settled and two countries go to war in the war one country discovers a means of nullifying gravity which after the war is used to start interplanetary travel s... ...

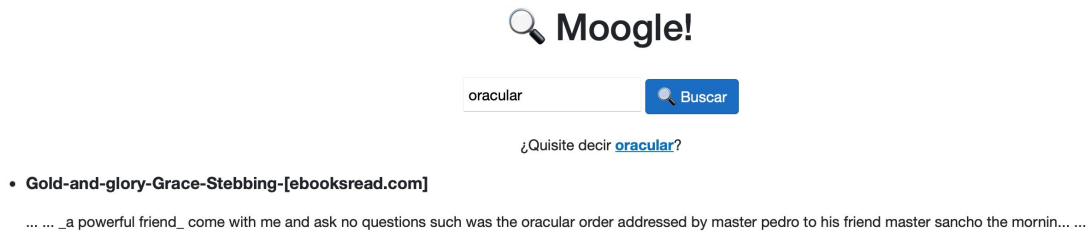
- **Gold-and-glory-Grace-Stebbing-[ebooksread.com]**

  ... ... _a powerful friend_ come with me and ask no questions such was the oracular order addressed by master pedro to his friend master sancho the mornin... ...

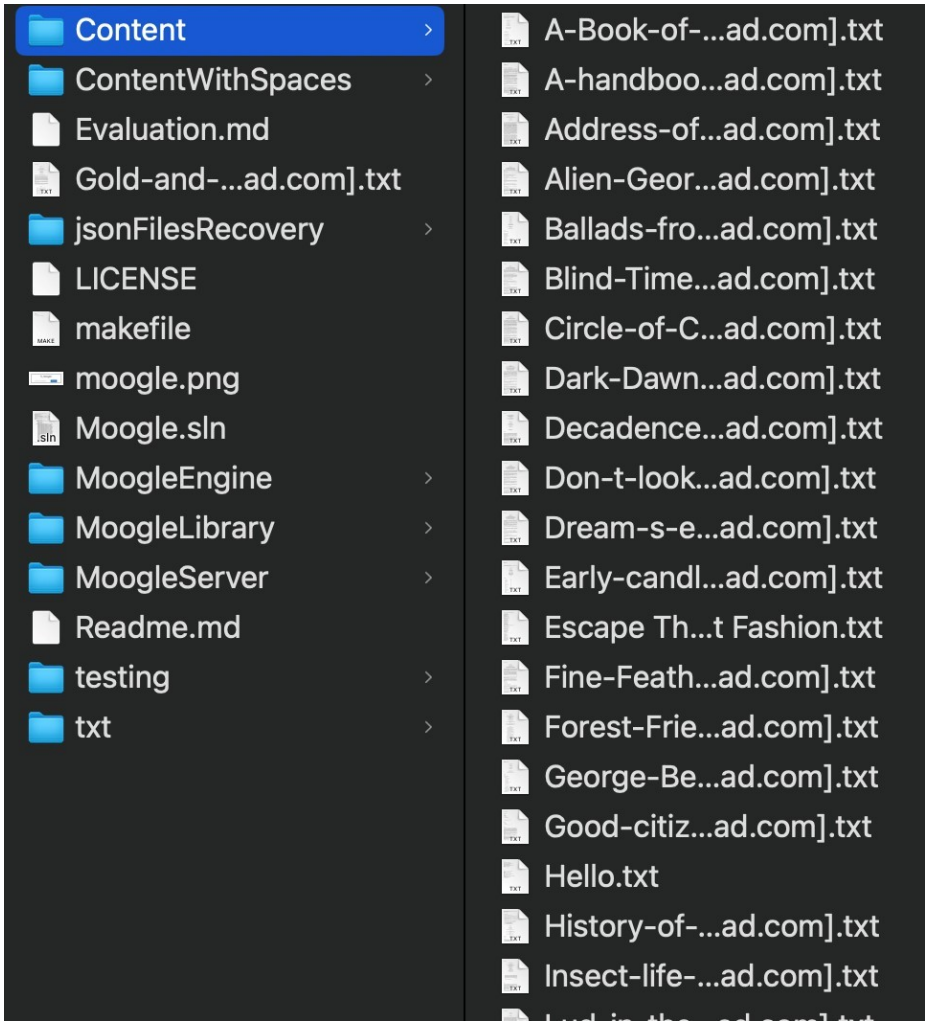
- **The-fixer-George-O---Geor-[ebooksread.com]**

  ... the fixer the fixer by wesley long illustrated by kramer [transcribers note this etext was produced from astounding sciencefict ...

Quitar documentos en tiempo real:(Los documentos se extraen o se eliminan o se agregan o se cambian , los que están dentro de Content, no del content whit spaces)



Extraer el documento de Content y ponerlo afuera para probar la búsqueda.



Si buscas de nuevo aparece:

🔍 Moogle!

oracular 🔍 Buscar

¿Quisite decir **oracular**?