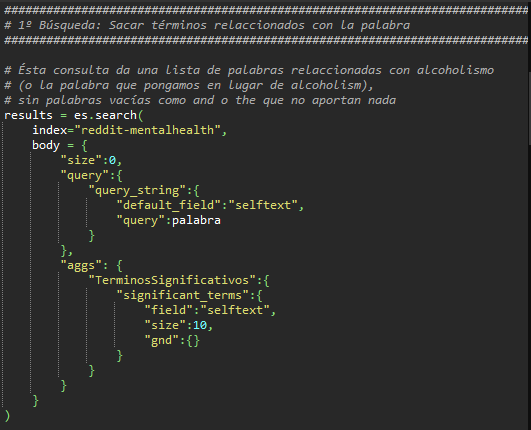
**Ejercicio 1**

El objetivo del ejercicio 1 es obtener documentos que traten sobre un determinado tema que introducirá el usuario (obviamente sin tener que contener esa palabra necesariamente). Para las pruebas de esta práctica se ha utilizado el alcoholismo (alcoholism) aunque el programa soporta cualquier término.

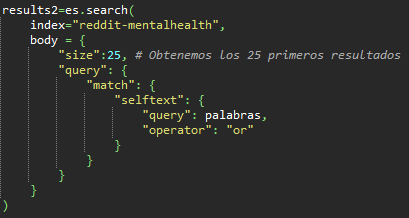
El script comienza haciendo una serie de consultas utilizando el parámetro gnd (Google normalized distance) para obtener palabras que se almacenarán en un set (se ha utilizado esa estructura de datos para que no se repitan las palabras). Las palabras más relacionadas con la palabra que ha introducido el usuario es lo que contendrá el set, que se imprimirá por pantalla al ejecutar el script.





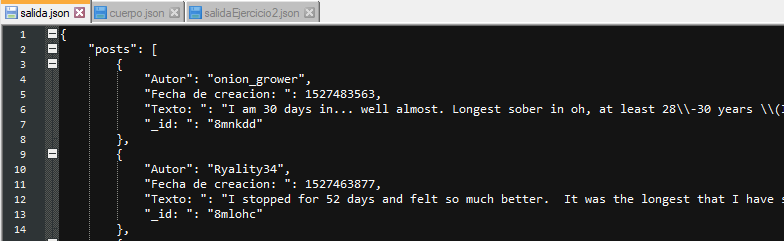
El resultado (results) está en formato JSON.

Una vez sacados los términos, lo que se hace es simplemente pasarlos a un string (palabras) para poder introducirlos en una consulta que nos devolverá los 25 posts que más de esas palabras coincidan.



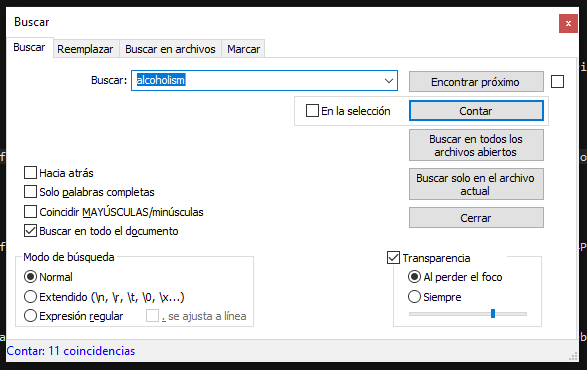
Una vez que ya tenemos los términos ya es cuestión de utilizar la librería json para crear el fichero de salida.

**Análisis del fichero de salida y su precisión:**



Una vez examinados los 25 documentos del fichero de salida, se comprueba que la precisión es del 100% ya que no hay ningún documento que no hable sobre un problema con la bebida.

Cabe destacar que hay bastantes documentos que no contienen la palabra que introdujo el usuario (alcoholism). De hecho, sólo hay 11 veces que aparece la palabra en el documento:



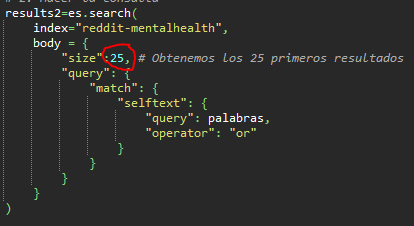
Para obtener la precisión del 100% de documentos relevantes, se han utilizado los siguientes valores:

* Número de palabras que se obtienen en la primera extracción de términos relacionados con el alcoholismo: 10



(En las siguientes dos expansiones también se utiliza 10. Al final no salen 30 porque obviamente se repiten términos y acaban saliendo unos 20 distintos para el término alcoholism)

* Número de documentos que se retornan: 25



Se ha probado ha retornar 100 documentos pero ya había alguno que no trataba de alguien que está hablando del alcoholismo, así que se dejo en 25 para lograr una precisión del 100% de documentos relevantes

**Ejercicio 2**

Las consultas MLT (More like this) por debajo funcionan de la siguiente manera:

* Se les pasa como parámetros una serie de documentos (ya sea por ejemplo en forma de un \_id de un documento que se encuentra en el índice, en este caso reddit-mentalhealth, o en forma de texto plano)
* Obtienen de ese documento o serie de documentos una serie de palabras representativas de esos documentos.
* A partir de esa serie de palabras, hacen una consulta que devuelve los documentos que más de esas palabras coincidan.

Por tanto, MLT lo que hace por debajo es una expansión de términos, pero en lugar de hacerla a partir de un término como en el ejercicio 1, lo hace a partir de una serie de documentos.

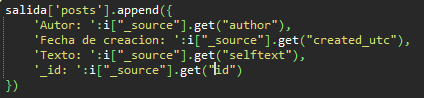
Para emular la expansión de términos del ejercicio 1 con MLT se puede coger uno o varios documentos obtenidos en la salida del ejercicio 1 (hablan sobre el alcoholismo, o la palabra que introduzca el usuario) y pasárselos como entrada a la consulta MLT para que saque otra serie de documentos relacionados.

Los términos que la consulta MLT ha considerado representativos no aparecen porque es un proceso interno, lo que nos devuelve la consulta MLT son los documentos relacionados.

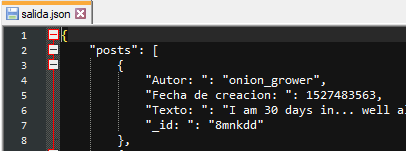
No hay problema porque el objetivo del ejercicio no es saber qué términos cree la consulta MLT que son representativos de los documentos de entrada, sino simplemente obtener los documentos de salida.

Por tanto, los pasos a seguir en éste ejercicio son los siguientes:

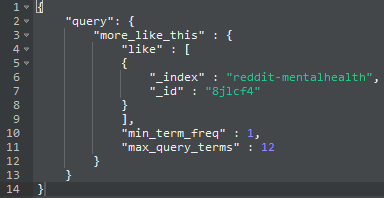
* Obtener del archivo JSON generado por el ejercicio 1 los \_id de los documentos (Para ello hay que modificar ligeramente el ejercicio 1 para que también añada los \_id de cada documento, el objetivo es poder hacer el ejercicio 1 y el 2 para cualquier término, no solo para alcoholismo.)



Muestra del archivo generado por el ejercicio 1:

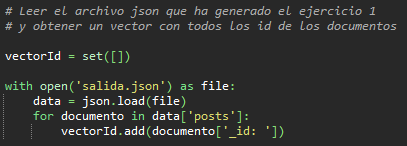


* Pasarle los \_id a la consulta MLT, un pequeño ejemplo:

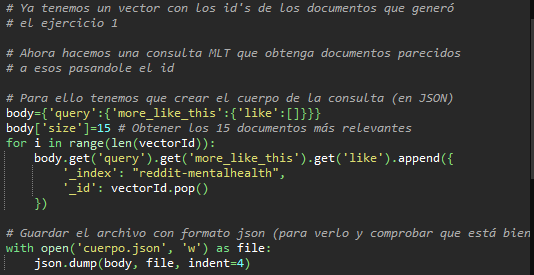


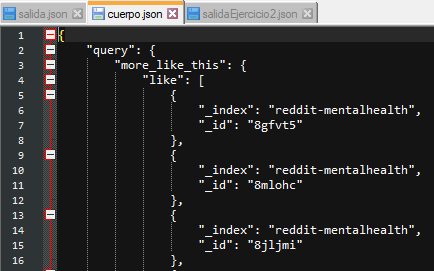
Esa consulta devuelve documentos parecidos al “8jlcf4”

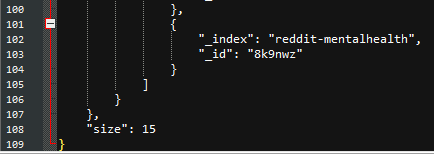
En el script lo que se hace es leer los id’s del fichero generado por el ejercicio1 mediante la librería json:



Una vez ya tenemos los id’s de los documentos, hay que crear el cuerpo de una consulta parecida a la que hay más arriba de ejemplo en cerebro. El cuerpo es en json:

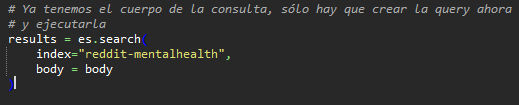






(Se le añade el parámetro size al final para que sólo retorne los 15 primeros archivos)

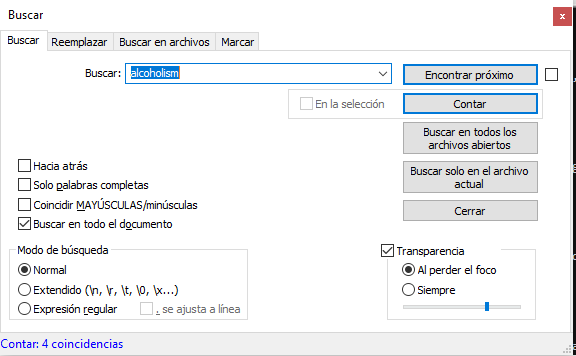
Como se puede ver en el archivo json que ha generado, es una consulta con el mismo formato que la que hay más arriba en cerebro. Ya solo es cuestión de ejecutarla:



* Una vez ya obtenidos los documentos que genera la consulta MLT hay que almacenarlos en un archivo JSON, lo cual se hace exactamente igual que en primer ejercicio, mediante la librería json se crea un archivo y se van añadiendo los documentos, no he creído conveniente explicar esta parte ya que el objetivo de este ejercicio es entender el funcionamiento de las consultas more like this.

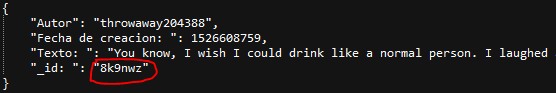
**Análisis del fichero de salida y su precisión:**

Mediante el parámetro “size”: 15 en el cuerpo de la consulta MLT (para que devuelva 15 documentos diferentes) se obtiene una precisión del 100% de documentos relevantes, con un total de 4 coincidencias de la palabra alcoholism:

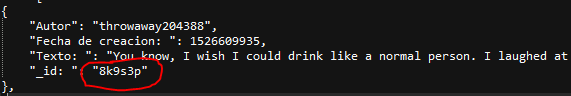


Cabe destacar que hay documentos que se repiten, pero es porque salen repetidos en el índice de reddit-mentalhealth, pero en realidad son posts distintos porque tienen un atributo \_id en el índice distinto, lo que me hace pensar que la persona ha copiado y pegado el contenido y publicado varios post, probablemente en varios subforos, ya que también las fechas son distintas. Eso sí, en cada una de las salidas no se repiten, cada post de cada salida es diferente.

Por ejemplo:



En salida.json (la salida del ejercicio 1)

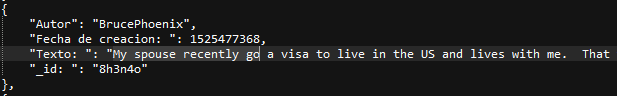


En salidaEjercicio.json

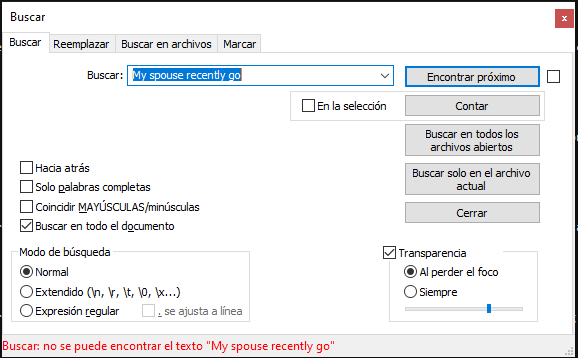
Esto se podría resolver controlando que los selftext sean distintos, pero realmente lo son, sólo cambia que hay algún carácter que cambia de los dos textos, así que sería bastante difícil diferenciar si un post es el mismo o no.

**Conclusión**

Por otra parte, la consulta more like this se podría considerar exitosa ya que realmente ha encontrado documentos que hablan sobre problemas con la bebida que no estaban en el primer documento, como por ejemplo el siguiente post:



No aparece en la salida del ejercicio 1 porque buscando las cuatro primeras palabras, no sale:



Y trata sobre el alcoholismo, habla de un amigo suyo que bebe mucho y tiene problemas.