Entregable elasticsearch

Repositorios de Información

CARLOS ÁLVAREZ RÁBANO [UO252116@uniovi.es](mailto:UO252116@uniovi.es)

SERGIO

ALEJANDRO

# EJERCICIO 1 (SERGIO)

Primero haciendo uso de *bulk-indexer-ngrams.py*, indexamos la colección eliminando palabras vacías.

A continuación, elaboramos un script que contendrá una consulta inicial, cuya temática esta referenciada a la drogodependencia:

Nuestro propósito es conocer cuál es el tipo de droga a la cual la gente es más adicta.

Para la consulta inicial haremos uso de la función ***search*** de ElasticSearch para ver los resultados obtenidos.

La consulta base será “\*drugs\*”, para la cual utilizamos términos significativos en agregaciones, poniendo como máximo para la consulta 25 términos asociados. A su vez eliminamos palabras vacías mediante *“exclude”.*

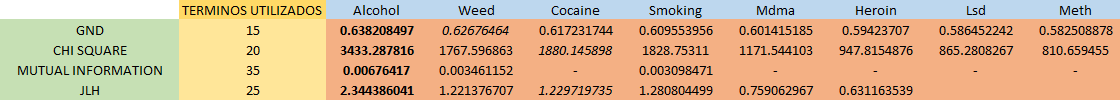
De esta consulta obtenemos los siguientes términos asociados:

1. Drugs
2. Drug
3. Recreational
4. Alcohol
5. Smoking
6. Cocaine
7. Weed
8. Years
9. Life
10. Smoke

Teniendo en cuenta los resultados obtenidos, elaboramos consultas en las que además de palabras vacías eliminamos términos asociados relevantes como “drugs”, “drug”, “recreational”, “years” y “life”, reducimos el número de términos significativos a 10 para no introducir mucho ruido en la consulta y utilizamos diferentes métricas:

1. Google normalized distance (gnd)
2. Chi square
3. Mutual Information
4. Jlh

Los resultados son los siguientes:



Leyenda

Analizando la evaluación sistemática de los resultados anterior, podemos deducir que el alcohol es con claridad, la droga más adictiva. Tras él se encuentran el cannabis o la cocaína, que dependiendo de la métrica utilizada tienen un porcentaje mayor que el otro.

Finalmente creamos una última consulta basada en la consulta inicial, añadiendo los términos relacionados más relevantes, los cuales como hemos dicho anteriormente, son “Alcohol”, “Weed” y “Cocaine”, volcando los resultados a un archivo JSON (mentalhealth-drugs-addition.json), incluyendo autor, fecha de creación y texto haciendo uso de la función ***scan*** de ElasticSearch.

# EJERCICIO 2 (SERGIO)

En primer lugar, tratamos de determinar si hay alguna configuración de consultas More Like This que dé lugar a un funcionamiento equivalente al primer ejercicio.

La documentación de More Like This nos explica que este encuentra documentos que son parecidos a una colección de documentos dada.

Consideramos que no es posible lograr un funcionamiento equivalente al anterior en el que obtengamos los términos más frecuentes ya que lo que nos obtiene son los documentos más recientes.

Tras esto realizamos generamos un código Python con el que obtenemos unos resultados simulares al ejercicio anterior:

1. Primero generamos una consulta inicial con la misma temática escogida en el ejercicio anterior, la cual nos devuelve unos resultados muy similares.
2. Creamos una segunda consulta extendida añadiendo los términos más relevantes de la consulta anterior y nuevamente, el número de documentos encontrados es muy similar al ejercicio anterior.

Para terminar, realizamos la misma consulta extendida volcando los resultados a un fichero JSON (mentalhealth-drugs-addition-MLT.json)

# EJERCICIO 3 (ALEJANDRO)

Este ejercicio trata de obtener una lista exhaustiva de los medicamentos que utilizan los usuarios de la colección. Para ello indexamos la misma con *bulk-indexer-ngrams.py.*

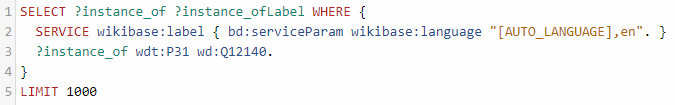
En un principio utilizamos Cerebro para realizar consultas iniciales de prueba, pero sin demasiado éxito en obtener resultados significativos. Usando en la *query* términos como “*alprazolam OR xanax*” u otros medicamentos, los términos relacionados son mas relevantes, pero como indica el ejercicio no utilizaremos los medicamentos para formar la consulta.

Finalmente se descubre que ciertos *subreddits* tienen una concentración muy superior de términos relevantes, como es el caso de *askdocs*. Se toma la decisión de abordar el problema de la siguiente manera, buscamos los *subreddits* mas relevantes y luego por cada uno de ellos buscamos los términos significativos en el *selftext*.

Usamos “*query*\_string” ya que con una sintaxis más compacta nos permite utilizar “*multi match queries”*, “*bool queries”* y “*wildcards queries”*. Tan solo usaremos *“\*medicat\* OR \*prescri\**” ya que al añadir más términos con *OR* como por ejemplo *dose*\*, *pill*\*, *doc*\*, *take*\*…no mejoraban los resultados o incluso añadían más ruido, y usando algunas combinaciones con *AND* resultaba demasiado restrictivo.

Ahora solo queda conseguir una lista de medicamentos mediante la API de Wikidata y validar los resultados obtenidos.

Se intento con la librería *Wikidata 0.6.1* (<https://pypi.org/project/Wikidata/>) pero es incompatible con nuestra versión de Python 2.7, como alternativa utilizamos *WPTools* (<https://github.com/siznax/wptools>) aunque con problemas a la hora de obtener datos del servidor. Finalmente se opto por *SPARQL* (<https://query.wikidata.org/>) que de una forma muy sencilla nos permite recuperar los datos con una consulta.



P31 es el código que hace referencia a la propiedad “*instance of*” y Q12140 a *Medication*

<https://www.wikidata.org/wiki/Property:P31>

<https://www.wikidata.org/wiki/Q12140>

Finalmente comparamos ambas listas y la intersección son la lista de medicamentos pedidos, con un total de 60:

[ "paroxetine", "escitalopram", "sertraline", "methylprednisolone", "cyclobenzaprine", "doxycycline", "hydrocodone", "oxycodone", "ibuprofen", "melatonin", "cocaine", "propranolol", "lisinopril", "bupropion", "dicyclomine", "ciprofloxacin", "fluticasone", "lamotrigine", "fentanyl", "caffeine", "levofloxacin", "gabapentin", "alprazolam", "meloxicam", "naproxen", "omeprazole", "metronidazole", "fluconazole", "trazodone", "metformin", "methylphenidate", "pantoprazole", "diazepam", "prednisolone", "prednisone", "fluoxetine", "clonazepam", "lorazepam", "cephalexin", "hydroxyzine", "water", "amitriptyline", "norepinephrine", "buspirone", "minocycline", "amphetamine", "clindamycin", "montelukast", "guanfacine", "morphine", "clotrimazole", "dopamine", "amoxicillin", "codeine", "progesterone", "citalopram", "azithromycin", "aspirin", "atomoxetine", "duloxetine" ]

Hay algunos términos que podrían cuestionarse si son medicamentos, como “*water*” y “*cocaine*”, pero estan así clasificados en Wikidata.

# EJERCICIO 4 (CARLOS ÁLVAREZ RÁBANO)

En este ejercicio se nos pide realizar varias consultas en las cuales:

1. Debemos encontrar factores comórbidos relacionados con conductas autolesivas y con conductas suicidas.
2. Realizamos un listado sobre enfermedades relacionadas con dichos factores

Para ello, empezamos creando una primera consulta para que nos volcara todos los documentos que estén relacionadas con conductas suicidas o autolesivas en un contenedor (en esta primera consulta solo se buscará una de las dos conductas), para facilitar la búsqueda se nos proporcionan unos palabras o frases claves de las cuáles se puede partir.

Y, por último, después de haber encontrado parte de los factores comórbidos más comunes creamos una segunda consulta para encontrar todos (o gran parte) de los factores comórbidos relacionados con ambas conductas, realizando una búsqueda más específica o especializada. En esta última consulta, a diferencia de la anterior, se nos permite utilizar palabras clave como “comorb” o “comorbidity” para mejorar la búsqueda.

Al realizar dicha búsqueda encontramos distintos factores comórbidos como:

Trastorno obsesivo-compulsivo

Trastorno de ansiedad generalizada

Abuso de medicamentos, alcohol y drogas

Ansiedad (incluida ansiedad social)

Depresión

Bullying

Trastorno de déficit de atención con o sin hiperactividad

Estrés post-traumático (por ejemplo, por un episodio de maltrato o violencia vivido en el pasado)

Enfermedades varias como:

Disnea, bulimia, anorexia, síndrome de distrofia simpática refleja, defecto del tabique auricular…