elasticsearch

Documentación

SOBRE LOS EJERCICIOS HECHOS DE ELASTIC SEARCH

Othmane Bakhtaoui | UO259323 | R.I. (mayo)

Tabla de Contenidos

1 – Ejercicio 1: 2

2 – Ejercicio 2: 3

3 – Ejercicio 3: 3

4 – Ejercicio 4: 4

1 – EJERCICIO 1:

La temática escogida es el tema de “addiction”. Los términos que se van a buscar en la consulta son **addiction** OR **craving** OR **obsession** que son todas palabas relacionadas con la adicción.

El tamaño de la colección que uso es 20.

Se comparan 4 métricas al inicio que son **gnd**, **jlh**, **chi_square** y **mutual_information** obteniendo 6 términos por cada una. Gnd y chi_square son parecidos, jlh también pero en mutual_information no.

Después, me quedé con los 3 primeros ya que el mutual_information no ofrece resultados similares con los 3 restantes. Ahora con los 3, obtengo 12 términos por cada métrica y ahora los resultados se diferencian y los mejores son **gnd** y **chi_square**.

Luego, con las dos métricas que quedan, aumento el tamaño de términos a 25 y ahora se nota la diferencia y podemos comprobar que el **gnd** es mucho mejor que el **chi_square**. Ya que el chi_square tiene un acierto de más de la mitad y el gnd de más de **70%.**

Finalmente, después de comparar las métricas, decidí sacar 12 términos relacionados con “adicción” ya que después de **12**, aparecen unos que tienen poca relación con el tema como “housefire” pero después de 14 o 15 aparecen algunos más relacionados como “**nicotine**”. Por lo tanto, 12 me pareció lógico y obtenemos los siguientes términos: **porn, pmo, nofap, masturbation, drug,heroin,dopamine,alcohol,pornography,sexual,smoking,masturbating ,smoke,masturbate, masturbation.** Asi que tenemos un buen porcentaje de acierto.

2 – EJERCICIO 2:

Las consultas "**More Like This**" sirven para encontrar documentos similares a un conjunto de documentos dado. Para hacer esto, selecciona un conjunto de términos representativos de estos documentos de entrada, y después, realiza una consulta utilizándolos. Una vez obtenidos los resultados, los devuelve. La labor del usuario es seleccionar los documentos de entrada, el método de selección de los términos y cómo se forma la consulta.
El caso de uso más simple consiste en solicitar documentos similares a un texto proporcionado, pero, existen otros casos de uso más complicados, consistentes en mezclar textos con documentos que ya existen en el índice, o también existe la opción de mezclar algunos textos, un conjunto elegido de documentos y otro conjunto de documentos que no es necesario que estén presentes en el índice.

En resumen, More Like This es una forma de crear consultas que nos ofrece resultados parecidos sin usar gnd en las agregaciones en el parámetro like.

Y tengo pensado que la consulta del ejercicio 1 se puede hacer algo parecido que está en el fichero **Ejercicio 2.py**

3 – EJERCICIO 3:

A través del ***es.search()*** buscamos la lista de medicamentos usando la query '*was prescribed' u*sando la métrica **gnd** después de la tasa de acierto notada en el ejercicio 1 y una lista de stopwords igual que en el ejercicio 1 pero añadiendo terminos para excluir como el tamaño de dosis "**.*mg.***", "**.*cm.***", "**.*[1-1000].***" y cualquier palabra que contiene "**.*presc.***". Además de un tamaño máximo de terminos de 1000.

Y para sacar los términos relacionados, recorremos los resultados y los validamos a través de una lista externa extensiva de medicamentos obtenida a través de ***Wikidata*** y otros como **OMS** y dejamos sólo los términos que existen en dichas listas (los medicamentos) y escribimos dichos términos en un fichero llamada *ej3_result.txt conteniendo 62 medicamentos.*

Hay una posibilidad de al lugar de la lista, usar el api de Wikidata obteniendo 71 medicamentos pero un poco menos relevantes comparada con la lista usada (*por ejemplo sale acid y doce y drugs*) y la búsqueda es muy lenta pero aún asi está incluida en el fichero pero comentada. Su resultado está en el fichero *ej3_result_using_api.txt*

4 – EJERCICIO 4:

En el principio como había que buscar artículos relacionados con los temas propuestos, usaba en el principio la librería **scholarly** de Python que ofrece los servicios de buscar por títulos, autores y resúmenes de artículos de Google Scholar. Pareció una solución ideal; Sacaba los factores relacionados con el self harm en primero buscándolos en títulos, selftexts y subreddits y combino cada término con la palabra "**comorbid**" y busco cada pareja usando dicha librería en títulos o bien resúmenes de cada artículo para ver si aparecerá. Pero al final no era lo esperado; Daba muchos errores de timeouts y errores de formato de unicode además que tardó mucho en escribir los factores comórbidos además de los términos sacados de las consultas.

Así al final, usé el programa Publish or Perish para buscar artículos relacionados con "comorbidity" y obtuve un fichero **article_titles.txt** con los títulos de dichos artículos. Asi que para temas por ejemplo del self harm busco los términos en los campos de título, selftext y subreddit, luego, compruebo si existen en la lista obtenida de PoP de títulos de los articulos y si está, los añado a un set y los saco por fichero. Y lo mismo para el tema de suicidio cambiando las consultas.

Un ejemplo de terminos en los resultados para hurt myself en los primeros 30 factores aparecen **fat, anxiety, drinking, cancer, sat, school, anxious.** Hay unos términos que describen perfectamente el tema y otros un poco.