**Ejercicio 4**

En este ejercicio se nos pide encontrar factores comórbidos relativos a la ideación suicida y a las conductas autolesivas.

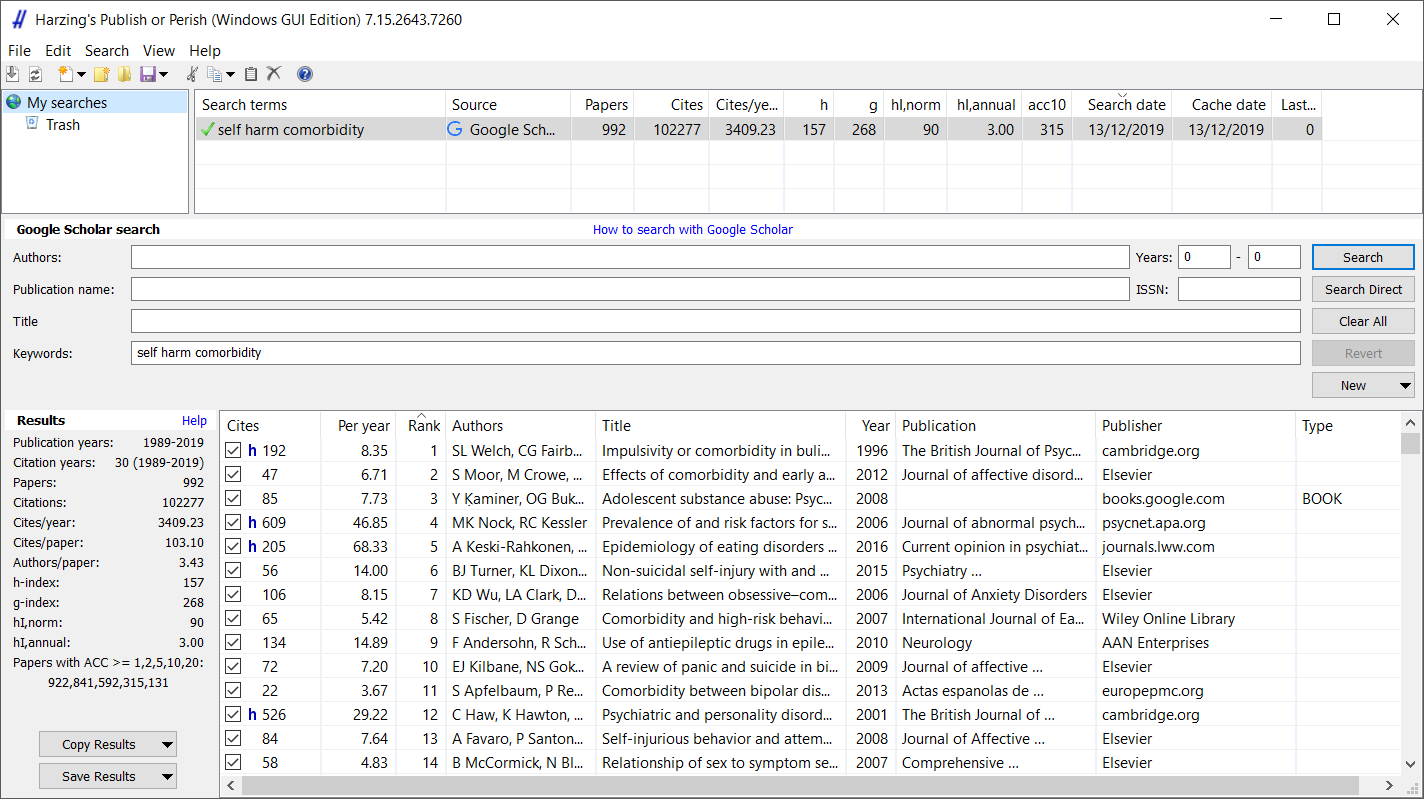
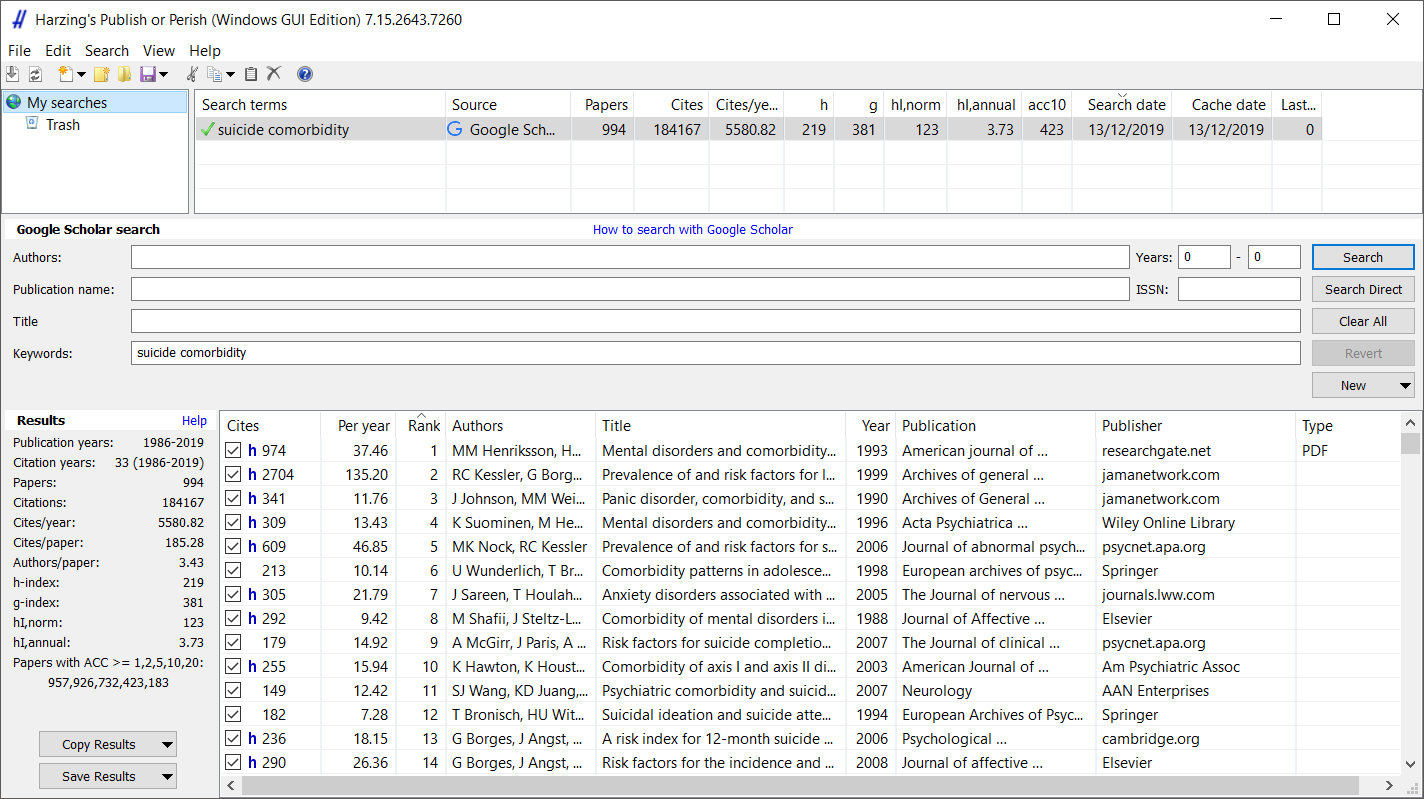
Para ello lo primero que vamos a hacer es encontrar los términos más importantes dentro del conjunto de post de Reddit proporcionado mediante una consulta:

|  |
| --- |
| es = Elasticsearch()  #numero de resultados maximo que dara la consulta  numero\_salidas = 500  results = es.search(  index="reddit-mentalhealth",  body = {  "size": 0,  "query": {  "query\_string": {  "default\_field": "selftext",  "query": query  }  },  "aggs": {  "Title": {  "significant\_terms": {  "field": "title",  "size": numero\_salidas,  "gnd": {}  }  },  "Text": {  "significant\_terms": {  "field": "selftext",  "size": numero\_salidas,  "gnd": {}  }  },  "Subreddit": {  "significant\_terms": {  "field": "subreddit",  "size": numero\_salidas,  "gnd": {}  }  }  }  })  words = []  for j in ["Subreddit", "Text", "Title"]:  for i in results["aggregations"][j]["buckets"]:  if i["key"] not in words and i["key"] not in querywords:  words.append(i["key"])  print("Obtenidas palabras relacionadas con "+query+": "+str(len(words)))  return words |

En esta consulta buscamos los términos dados en el parámetro query ("suicide suicidal \"kill myself\" \"killing myself\" \"end my life\"" en el caso de suicidio y "\"self harm\"" en el caso de conductas autolesivas) y devolvemos los resultados en una lista donde quitamos los términos usados, ya que por definición los factores comórbidos no pueden ser la propia enfermedad.

A continuación necesitamos validar los resultados con un conocimiento experto, para ello utilizaremos la herramienta Publish or Perish ofrecida en <https://harzing.com/resources/publish-or-perish>.

Una vez obtenida la herramienta hacemos las consultas de “suicide comorbidity” y “self harm comorbidity” en Google Scholar obteniendo así dos archivos Json que usaremos mas adelante en el script



Obtenemos los títulos de los artículos obtenidos en los Json mediante el script:

|  |
| --- |
| words = []  #Es necesario especificar el encoding para que no de error  with open(name, encoding='utf-8-sig') as json\_file:  data = json.load(json\_file)  for word in data:  words.append(word['title'])  print("Obtenidas palabras del json "+name+": "+str(len(words)))  return words |

Y una vez hecho esto solo hay que comparar las dos listas de palabras para obtener asi los términos que hacen referencia a las comórbilidades

|  |
| --- |
| wordsElasticSearch = readElasticsearch(query)  wordsJson = readJson(json)  finalWords = []  for we in wordsElasticSearch:  for wj in wordsJson:  #separamos las palabras de los titulos y comprobamos que no se  #hayan añadido ya  if we in wj.split() and we not in finalWords:  finalWords.append(we)  return finalWords |

Nota:

Para realizar la consulta con elasticsearch correctamente es necesario configurarlo anteriormente, para ello se ha optado en meter el siguiente código en el script, aunque también se podría hacer desde cerebro:

|  |
| --- |
| es = Elasticsearch()  mapping = {  "properties": {  "author": {  "type": "text",  "fielddata": "true"  },  "selftext": {  "type": "text",  "fielddata": "true"  },  "title": {  "type": "text",  "fielddata": "true"  },  "subreddit": {  "type": "text",  "fielddata": "true"  }  }  }  response = es.indices.put\_mapping(  index="reddit-mentalhealth",  body=mapping,  ignore=400 # ignore 400 already exists code  ) |