Elasticsearch assignment

UO258425, Carlos Manrique Enguita
UO236405, Daniel Rückert García
UO258454, Violeta Ruiz Martínez

December 15, 2019

Contents

1 Exercise 1

We choose "Rehab" as topic to develop this first exercise.

To recover all the posts related to the topic we do a first query in order to find the most significant words related to it. To do that we look for the most significant words in posts that contain the words **Rehab** and **Rehabilitation**. We perform an aggregation in the selftext field. Whe tried this first query using three different similarity metrics: Chi square, Google normalized distance and percentage. Whe have created an index without stopwords so we perform this first query in this index to get the most optimum results.

1.1 Percentage

When using percentage we set the maximum number of significant terms to 100 because it is the one that gets the fewer results. With this metric we get 10 terms where we consider 5 as relevant:

- Rehabilitation
- Rehab
- Rehabs
- Criminals
- Librium (a medical drug)

1.2 Google normalized distance

When we use the Google normalized distance we set the maximum number of significant terms to 20 as this metric and the Chi square get the most number of results. With this metric we got 20 terms where we consider 9 as relevant:

- Rehab
- Rehabilitation
- Jail
- Facility
- Criminals
- Outpatient
- Heroin
- Homicide
- Detox

1.3 Chi square

When we used the Chi square metric we also set the maximum number of significant terms to 20 as we did for the Google one. Using this metric we also get 20 terms and we consider 9 as relevant:

- Rehab
- Rehabilitation
- Detox
- Alcoholism
- Sober
- Drinking

- Criminals
- Outpatient
- Hospital

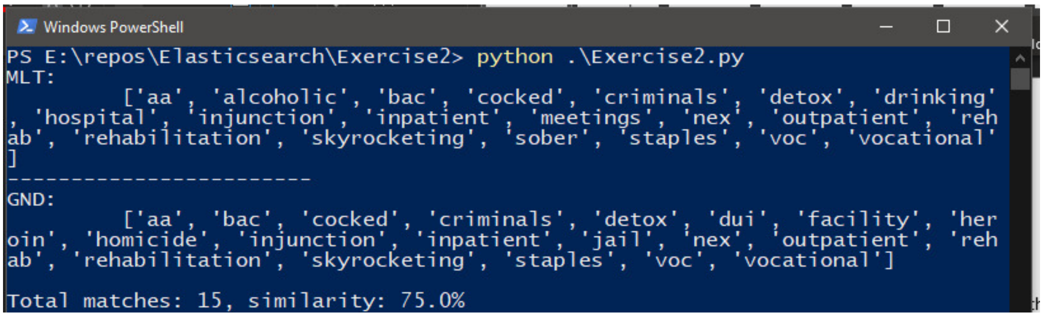
After getting the significant terms we perform a second query searching for these words in the selftext.

2 Exercise 2

MLT queries do not directly produce a list of significant terms, they produce however a list of documents including the relevant terms used to perform the query. This is simlar to how Google produces its search results. As we know GND uses the documents resturned by google to generate the list of significant terms.

Therefore, to perform a similar operation to that of GND we just have to do a query for significant terms over the MLT query in order to get the most significant terms of those documents.

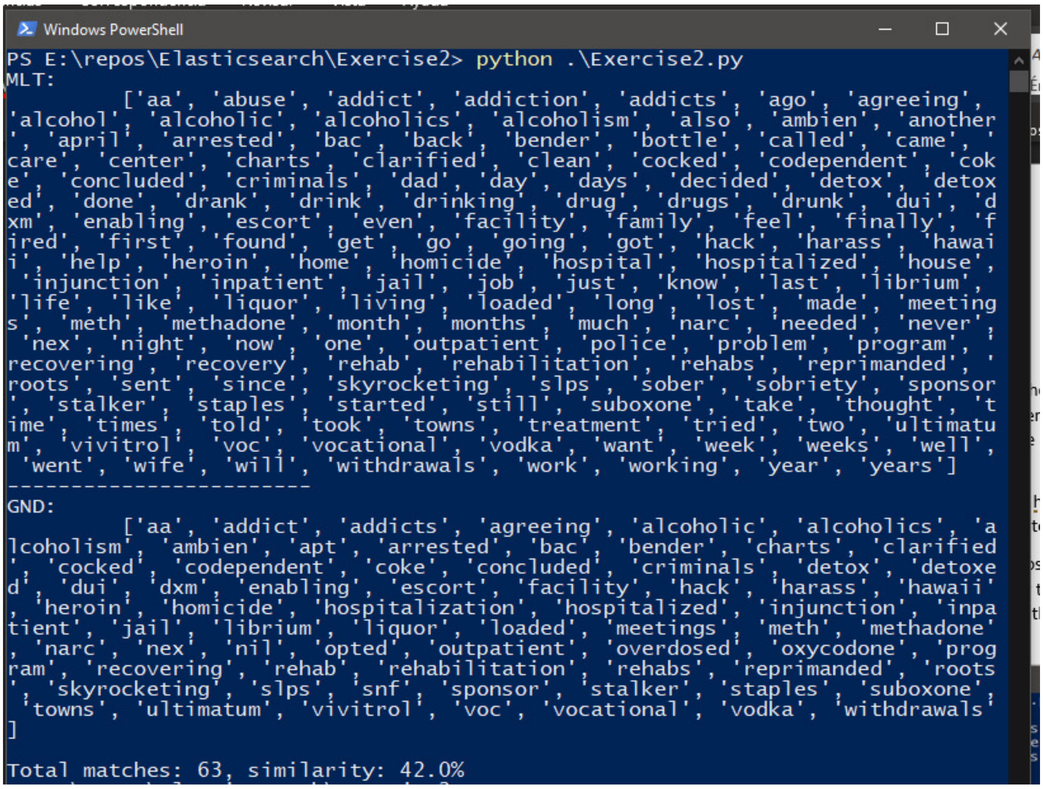
After performing such operation , we compared the results to those obtained by a GND query, obtaining a list of 20 significant terms for each query. Comparing the results, we obtained that 75% of the significant terms were identical. However, increasing the number of significant terms to more than 100 reduced this accuracy down to 40%.



```
PS E:\repos\Elasticsearch\Exercise2> python .\Exercise2.py
MLT:
        ['aa', 'alcoholic', 'bac', 'cocked', 'criminals', 'detox', 'drinking'
, 'hospital', 'injunction', 'inpatient', 'meetings', 'nex', 'outpatient', 'reh
ab', 'rehabilitation', 'skyrocketing', 'sober', 'staples', 'voc', 'vocational'
]
------------------------
GND:
        ['aa', 'bac', 'cocked', 'criminals', 'detox', 'dui', 'facility', 'her
oin', 'homicide', 'injunction', 'inpatient', 'jail', 'nex', 'outpatient', 'reh
ab', 'rehabilitation', 'skyrocketing', 'staples', 'voc', 'vocational']

Total matches: 15, similarity: 75.0%
```

Figure 1: Results with 20 significant terms



```
PS E:\repos\Elasticsearch\Exercise2> python .\Exercise2.py
MLT:
         ['aa', 'abuse', 'addict', 'addiction', 'addicts', 'ago', 'agreeing',
'alcohol', 'alcoholic', 'alcoholics', 'alcoholism', 'also', 'ambien', 'another
', 'april', 'arrested', 'bac', 'back', 'bender', 'bottle', 'called', 'came', '
care', 'center', 'charts', 'clarified', 'clean', 'cocked', 'codependent', 'cok
e', 'concluded', 'criminals', 'dad', 'day', 'days', 'decided', 'detox', 'detox
ed', 'done', 'drank', 'drink', 'drinking', 'drug', 'drugs', 'drunk', 'dui', 'd
xm', 'enabling', 'escort', 'even', 'facility', 'family', 'feel', 'finally', 'f
ired', 'first', 'found', 'get', 'go', 'going', 'got', 'hack', 'harass', 'hawai
i', 'help', 'heroin', 'home', 'homicide', 'hospital', 'hospitalized', 'house',
 'injunction', 'inpatient', 'jail', 'job', 'just', 'know', 'last', 'librium',
'life', 'like', 'liquor', 'living', 'loaded', 'long', 'lost', 'made', 'meeting
s', 'meth', 'methadone', 'month', 'months', 'much', 'narc', 'needed', 'never',
 'nex', 'night', 'now', 'one', 'outpatient', 'police', 'problem', 'program', '
recovering', 'recovery', 'rehab', 'rehabilitation', 'rehabs', 'reprimanded', '
roots', 'sent', 'since', 'skyrocketing', 'slps', 'sober', 'sobriety', 'sponsor
', 'stalker', 'staples', 'started', 'still', 'suboxone', 'take', 'thought', 't
ime', 'times', 'told', 'took', 'towns', 'treatment', 'tried', 'two', 'ultimatu
m', 'vivitrol', 'voc', 'vocational', 'vodka', 'want', 'week', 'weeks', 'well',
 'went', 'wife', 'will', 'withdrawals', 'work', 'working', 'year', 'years']
------------------------
GND:
         ['aa', 'addict', 'addicts', 'agreeing', 'alcoholic', 'alcoholics', 'a
lcoholism', 'ambien', 'apt', 'arrested', 'bac', 'bender', 'charts', 'clarified
', 'cocked', 'codependent', 'coke', 'concluded', 'criminals', 'detox', 'detoxe
d', 'dui', 'dxm', 'enabling', 'escort', 'facility', 'hack', 'harass', 'hawaii'
, 'heroin', 'homicide', 'hospitalization', 'hospitalized', 'injunction', 'inpa
tient', 'jail', 'librium', 'liquor', 'loaded', 'meetings', 'meth', 'methadone'
, 'narc', 'nex', 'nil', 'opted', 'outpatient', 'overdosed', 'oxycodone', 'prog
ram', 'recovering', 'rehab', 'rehabilitation', 'rehabs', 'reprimanded', 'roots
', 'skyrocketing', 'slps', 'snf', 'sponsor', 'stalker', 'staples', 'suboxone',
 'towns', 'ultimatum', 'vivitrol', 'voc', 'vocational', 'vodka', 'withdrawals'
]

Total matches: 63, similarity: 42.0%
```

Figure 2: Results with 150 significant terms

3 Exercise 3

To develop this exercise we perform a query pretty similar to the one in the first exercise, we look for the most significant terms in the **selftext** field of the post that match *”prescribed me”*, *”prescr*”*, or *”antidepressant*”*. We just use this three possible matches because we noticed that adding too much terms got worse results for us.

In the aggregation we specified a maximum size of 100 and used Google Normalized Distance as metric because after trying with all of the ones that we used to develop the previous exercises we noticed that GND was the one that was working best in this case.

When though we are working upon an index where the stop words have been removed we added some other words that must not be taken into account as significant terms in the aggregation, which are words that are usually related to the taking of a medication such as types of doctors who are usually the ones allowed to make a prescription, illnesses and symptoms, side effects, or medication types such as pills, tablets and so on.

After performing this query we end up with a list of a hundred words where some are medication names and some are not and for that matter we validate it by using Wikidata.

In order to make this validation automatically we use the MediaWiki API to get the data from Wikidata. First for each significant term that we have obtained we perform a first query to Wikidata in order to get all of the entities which name matches a given significant term. As this can recover more than one result we go through it getting the code for each of the entities recovered and perform the last query to Wikidata using that code. This last query recovers all the information about the entity. In this last step we check if the entity is an *”instance of”* (”P31”) medication(”Q12140”) ir pharmaceutical product (”q28885102”) and if it is we take that result as valid.

After the validation we get a list of 49 medication names which are:

- Zoloft
- lexapro
- amoxicillin
- fluexetine
- prednisolone
- prozac
- sertraline
- metronidazole
- norepinephrine
- klonopin
- doxycycline
- amitriptyline
- levofloxacin
- escitalopram
- hydroxyzine
- trazodone
- effexor
- paroxetine

- dexamphetamine
- concerta
- duloxetine
- flagyl
- ritalin
- buspar
- valium
- vitamin
- lisinopril
- clindamycin
- intuniv
- lamotrigine
- atarax
- prevacid
- fluticasone
- cipro
- miralax
- gabapentin
- focalin
- methylprednisolone
- alprazolam
- naproxen
- venlafaxine
- strattera
- cyclobenzaprine

4 Exercise 4

The main purpose of this exercise is to obtain a list with the greatest number of comorbidity factors with respect to suicidal ideation and self-injuring behaviors.

4.1 Retrieving the Reddit results

As we cannot use any expert knowledge as a starting point, we have decided to begin using a multi match query about **”kill myself”** by searching the fields selftext, subreddit and title. We take the 15 most relevant subreddits from this query which are:

- SuicidalWatch
- depression
- offmychest
- Advice
- NoFap
- stopdrinking
- BPD
- mentalhealth
- Anxiety
- AskDocs
- ADHD
- schizophrenia
- leaves
- OCD
- stopsmoking

4.2 Publish or Perish

The exercise is formed by two python scripts called **Exercise4ES.py** and **Exercise4ES-PoP.py**. In order to get the results the second one must be executed first. The first script computes the final results.

In order to obtain reliable data on the true comorbities of suicidal ideation and self-injuring behaviors, **Publish or Perish** tool has been used. It allows us to obtain the main scientific papers that deal with this topic and are available in Google academics. Any document that mentions comorbidities of these topics” seems relevant to our propouses. After indexing the documents, we export them in a JSON file called **pop.json** that will be treated later as follows:

1. We have realized that the only relevant information is contained in the title, so we removed anything else from the json objects.

2. Remove all the titles that are not written in english.

3. All English stopwords have been removed.

4. The list of titles has become a list of words without duplicates.

5. That list is converted to a dictionary [key, value] containing the word and the frequency of appearance.
6. The dictionary is sorted in descending order.
7. We take the words with a frequency greater than 10.

Then we store the output in a file called **popOutput.txt**.

The keywords used for the Publish or Perish search were: ”self-injuring behavior”, ”suicidal comorbidity”, ”suicide comorbidity”, ”suicidal comorbidites”, ”suicide comorbidities” and ”suicidal ideation”.

4.3 Expanding the first Elastic query

After retrieving the main subreddits that have information about suicide we perform a query in order to get the 20000 most frequent words by using GND in each subreddit. As by default ElasticSearch dont allow to get more than 1000 results at once, it is mandatory to set a bigger limit using cerebro:

```
PUT \_cluster/settings
{
    "persistent":{
        "search.max\_buckets":20000
    }
}
```

Then we compare these words with the ones in **popOutput.txt**. If it exists in that file, its added to the result and stored in a file **result.txt**

4.4 Understanding the results

This process give us some false positive results, but most of them are pretty accurate. Some examples of comorbidites contained in the result file are:

- Schizophrenia
- Cancer
- drug
- lupus
- scitalopram
- winter
- lesbian
- pain
- old
- stress
- work
- school
- alcohol
- eczema
- health
- pregnant