Represent (Team 028)

March 27, 2021

We start by importing “Yet Another Keyword Extractor (Yake)” - an open source unsupervised automatic Keyword extraction program.

[6]:

```
import yake
import requests
from PIL import Image
```

Next, we read in an input file with a large quantity of text to be analysed

[7]:

```
filename = "data.txt"

f = open(filename, mode='r')
text = f.read()
```

Our first step of analysis is to summarise the text using the open source summary engine, SMMRY

[8]:

```
key = "1D69298B9E"
endpoint = 'https://api.smmry.com/'

data = {
    "sm_api_input":text
}
params = {
    "SM_API_KEY": key,
    "SM_LENGTH": "7"
}
headers = {"Expect":"100-continue"}
response = requests.post(url=endpoint, params=params, data=data,␣
 ↪headers=headers)

summary = response.json()['sm_api_content']

print(summary)
```

```
In participants who received two standard doses, vaccine efficacy was 62·1% and
in participants who received a low dose followed by a standard dose, efficacy
was 90·0%. Overall vaccine efficacy across both groups was 70·4%. From 21 days
after the first dose, there were ten cases hospitalised for COVID-19, all in the
```

```
control arm; two were classified as severe COVID-19, including one death. Phase
2 cohort in COV002 in older adults6 have been published and show an acceptable
safety profile for the vaccine with induction of binding and neutralising
antibodies as well as generation of interferon- enzymelinked immunospot
responses, with higher antibody titres after a second dose of vaccine. Two
dosage groups were included in COV002: participants who received a low dose of
the vaccine as their first dose and were boosted with a standard dose, and
subse quent cohorts who were vaccinated with two standard-dose vaccines. The
trial staff administering the vaccine pre pared vaccines out of sight of the
participants and syringes were covered with an opaque material until ready for
administration to ensure masking of participants. Were 30 cases among 5807
participants in the vaccine arm and 101 cases among 5829 par ticipants in the
control group, resulting in vaccine efficacy of 70·4%. In participants who
received two standard-dose vaccines, vaccine efficacy was 62·1%, whereas in
those who received a low dose as their first dose of vaccine, efficacy was
higher at 90·0%. In England and Wales, 129 529 weekly self-swabs were processed
by the DHSC, of which 126 324 were matched to study participants. Similar
results have been seen for other vaccines where a reduced number or type of
priming dose in infancy can lead to higher responses to a booster vaccine.
Thelancet.com Vol 397 January 9, 2021  Articles Other coronavirus vaccine
developers have released preliminary high-level results in public statements,
including more than 90% efficacy reported for the lipid nanoparticle mRNA
vaccine BNT162b2,11 92% efficacy for the Sputnik V vaccine,12 and 94·5% for the
Moderna lipid nanoparticle mRNA-1273 vaccine.
```

Our second step of analysis is to extract the key ideas of the article in the form of keywords by using the YAKE module we have previously imported.

[9]:

```
language = "en"
max_ngram_size = 3
deduplication_thresold = 0.9
deduplication_algo = 'seqm'
windowSize = 1
numOfKeywords = 4

custom_kw_extractor = yake.KeywordExtractor(lan=language, n=max_ngram_size,␣
 ↪dedupLim=deduplication_thresold,
                                            dedupFunc=deduplication_algo,␣
 ↪windowsSize=windowSize, top=numOfKeywords, features=None)

keywords = [word for word, score in custom_kw_extractor.extract_keywords(text)]

print(keywords)
```

```
['vaccine efficacy', 'vaccine', 'efficacy', 'dose']
```

Our next step is to associate each of these key phrases (also known as ngrams) with an image obtained from the google custom search API

[16]:

```
key = "AIzaSyDfAQR2Q3KIOjJUT-0_kJgmTCcB2_lcQPY"
search_engine_id = "f980e922532fe1c1a"

for k in keywords:
    payload = {'key': key, 'cx': search_engine_id, 'q': k, 'searchType':␣
 ↪"image"}
    query = 'https://www.googleapis.com/customsearch/v1?'
    response = requests.get(query, params=payload)

    i = 0
    uri = ""

    if 'error' in response.json().keys():
        print("Google API Quota for today exceeded. Please obtain new keys or␣
 ↪try again tomorrow")
    else:
        while(not(uri.endswith(".jpg"))):
            uri = response.json()['items'][i]['link']
            i += 1

        img_PIL = Image.open(requests.get(uri, stream=True).raw)
        img_PIL.save("images/"+k+".jpg")
        img_PIL.show()
```

Finally, we identify where the keywords and their corresponding ideas appear in the tex

[11]:

```
def find_most_frequent_word(data, window_size, keyword):
    most_frequent_window = -3
    count_max = 0
    for i in range(0, len(data), window_size):
        curr_count = 0
        for j in range(i, min(i+window_size, len(data))):
            curr_count += data[j].count(keyword)
            # do all the counting here
        if count_max < curr_count:
            count_max = curr_count
            most_frequent_window = i+2

    return most_frequent_window

data = text.splitlines()
window_size = 5

for k in keywords:
    print("\"" + k + "\" is mentioned around line ", end="")
    print(find_most_frequent_word(data, window_size, k))
```

```
"vaccine efficacy" is mentioned around line 37
```

```
"vaccine" is mentioned around line 2262
"efficacy" is mentioned around line 37
"dose" is mentioned around line 27
```

[12]:

```
f.close()
```

Citations

In-depth journal paper at Information Sciences Journal

Campos, R., Mangaravite, V., Pasquali, A., Jatowt, A., Jorge, A., Nunes, C. and Jatowt, A. (2020). YAKE! Keyword Extraction from Single Documents using Multiple Local Features. In Information Sciences Journal. Elsevier, Vol 509, pp 257-289. pdf

ECIR'18 Best Short Paper

Campos R., Mangaravite V., Pasquali A., Jorge A.M., Nunes C., and Jatowt A. (2018). A Text Feature Based Automatic Keyword Extraction Method for Single Documents. In: Pasi G., Piwowarski B., Azzopardi L., Hanbury A. (eds). Advances in Information Retrieval. ECIR 2018 (Grenoble, France. March 26 – 29). Lecture Notes in Computer Science, vol 10772, pp. 684 - 691. pdf

Campos R., Mangaravite V., Pasquali A., Jorge A.M., Nunes C., and Jatowt A. (2018). YAKE! Collection-independent Automatic Keyword Extractor. In: Pasi G., Piwowarski B., Azzopardi L., Hanbury A. (eds). Advances in Information Retrieval. ECIR 2018 (Grenoble, France. March 26 – 29). Lecture Notes in Computer Science, vol 10772, pp. 806 - 810. pdf