Task 1. Search Engine

Part 1: Crawler:

1.2. Information collected about each publication (e.g. links, title, year, author or any additional part)

The below code will crawl the input URL "https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/"which is the Coventry university school of economics and accounting/ publications URL and fetch all the necessary details for each publications like publications links, publications title, publication date, author title, author name, abstract, content (which is abstract + author name).For all the publications I pulled all the above data and I added into a data frame and then at last the data frame is saved into a csv file. The csv file contains 638 rows and 7 columns. (Publications links, publications title, publication date, author title, author name, abstract, content (which is abstract + author name)).

Screenshots of above code:

```
# Import libraries
from urllib.request import urljoin
from bs4 import BeautifulSoup
import requests
from urllib.request import urlparse
import re
import pandas as pd
# # Set for storing urls with same domain

# # Method for crawling a url at next level
def level_crawler(input_url,temp_urls,count_publications,df,j):
    data = []
    current_url_domain = urlparse(input_url).netloc
#   # Creates beautiful soup object to extract html tags
    beautiful_soup_object = BeautifulSoup(requests.get(input_url).content, "lxml")
#   # Access all anchor tags from input
#   # url page and divide them into internal
#   # and external categories
    for anchor in beautiful_soup_object.findAll("a"):
        href = anchor.attrs.get("href")
        title = anchor.string
        if(href != "" or href != None):
            href = urljoin(input_url, href)
            href_parsed = urlparse(href)
            href = href_parsed.scheme
            href += "://"
            href += href_parsed.netloc
            href += href_parsed.path
            final_parsed_href = urlparse(href)
            is_valid = bool(final_parsed_href.scheme) and bool(
                final_parsed_href.netloc)
            if is_valid:
```

```
            if is_valid:
                if current_url_domain not in href and href not in links_extern:
                    #print("Extern - {}".format(href))
                    links_extern.add(href)
                if current_url_domain in href and href not in links_intern:
                     #if re.search("^https://pureportal.coventry.ac.uk/en/publications/.*$", href):
                        #re.search("^The.*Spain$", href)
                    if '/publications/' in href[36:50] and href[50:] != '':
                        #print("Intern - {}".format(href))
                        count_publications += 1
                        links_intern.add(href)
                        temp_urls.append(href)
                        data.append(href)
                        #print('publications title:',title)
                        df.loc[j, ['publication_link']]  = href
                        df.loc[j, ['publication_title']] = title
                        df,j = get_author_details(href,df,j)
                        j += 1
    return temp_urls,count_publications,df,j,data
def get_author_details(link,df,j):
    source_code = requests.get(link)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text,"lxml")
    for link in soup.findAll('a',{'class':'link person'}):
        href = link.get('href')
        name = link.string
        source_code1 = requests.get(href)
        plain_text1 = source_code1.text
        soup1 = BeautifulSoup(plain_text1,"lxml")
        if(soup1.find('a',{'class':'link primary'}) != None):
            sefa = soup1.find('a',{'class':'link primary'}).string
```

```
        elif(soup1.find('a',{'class':'link school'}) != None):
            sefa = soup1.find('a',{'class':'link school'}).string
        if (sefa == 'School of Economics, Finance and Accounting'):
            df.loc[j, ['author_link']]  = href
            df.loc[j, ['author_name']] = name
    publi = soup.find('tr',{'class':'status'})
    date = publi.text.strip('date')
    #print('date:',date)
    #print(abstract,'abs.........')
    if (soup.find('div',{'class':'textblock'})):
        abstract = soup.find('div',{'class':'textblock'}).string
    else:
        abstract = 'Null'
    #print(abstract,'abs.........')
    if type(abstract) == 'NoneType':
        abstract = ' '
    df.loc[j, ['publication_date']]  = date
    df.loc[j, ['abstract']] = abstract
    #df.loc[j, ['content']] = abstract + name
    return df,j

links_intern = set()
url = 'https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/'
depth = 1
p = 0
max_pages = 12
count_publications = 0
# # Set for storing urls with different domain
df = pd.DataFrame(columns = ["publication_link","publication_title","publication_date","author_link","author_name","abstract","co
j = 0
links_extern = set()
```

```
if(depth == 0):
    print("Intern - {}".format(input_url))
elif(depth == 1):
    while p >= 0:
        if p == 0:
            list = []
            print(url,'url page0')
            list_url,count_publications,df,j,data = level_crawler(url,list,count_publications,df,j)
            p += 1
        else:
            new_url = url+'?page='+str(p)
            print(new_url,'url page')
            list_url,count_publications,df,j,data = level_crawler(new_url,list_url,count_publications,df,j)
            p += 1
        |   if (data == []):
                p = -1
else:
#   # We have used a BFS approach
#   # considering the structure as
#   # a tree. It uses a queue based
#   # approach to traverse
#   # links upto a particular depth.
    queue = []
    queue.append(input_url)
    for j in range(depth):
        for count in range(len(queue)):
            url = queue.pop(0)
            urls = level_crawler(url)
            for i in urls:
                queue.append(i)
print('end')
print('total no of publications',count_publications)
```

```
df['content']=df['abstract']+df['author_name']+df['publication_title']
df
```

```
import csv
df.to_csv(path_or_buf ="crawlerdata.csv",sep=',')
```

Output screenshots:

Below screenshots shows the output of crawler. There are total 638 publications in 13 web pages crawler crawls all the web pages. Added a print statement to check whether my crawler is accessing all web pages or not. Also added a count_publications variable to count total number of publications being crawled and at the end of the program added a print statement to print the count_publications.

```
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/ url page0
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=1 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=2 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=3 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=4 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=5 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=6 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=7 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=8 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=9 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=10 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=11 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=12 url page
https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/?page=13 url page
end
total no of publications 638
```

➔ Below screenshots shows the data frame containing all the publications details.

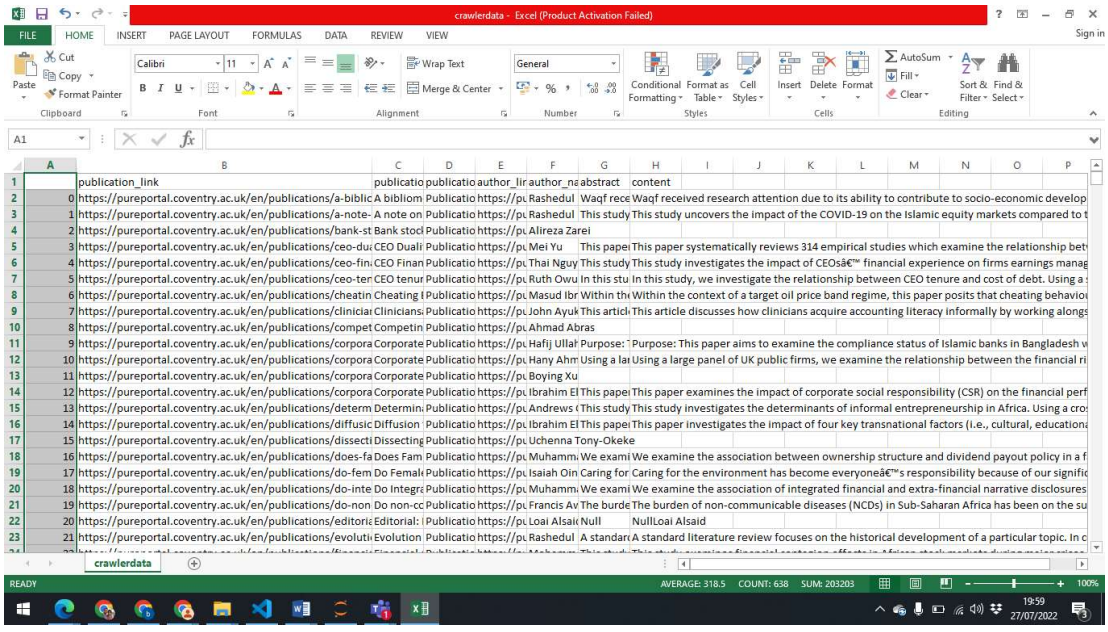
285]:

	publication_link	publication_title	publication_date	author_link	author_name	abstract	content
0	https://pureportal.coventry.ac.uk/en/publicati...	A bibliometric review of the Waqf literature	Publication statusPublished - Jun 2022	https://pureportal.coventry.ac.uk/en/persons/r...	Rashedul Hasan	Waqf received research attention due to its ab...	Waqf received research attention due to its ab...
1	https://pureportal.coventry.ac.uk/en/publicati...	A note on COVID-19 instigated maximum drawdown...	Publication statusPublished - May 2022	https://pureportal.coventry.ac.uk/en/persons/r...	Rashedul Hasan	This study uncovers the impact of the COVID-19...	This study uncovers the impact of the COVID-19...
2	https://pureportal.coventry.ac.uk/en/publicati...	Bank stock valuation theories: do they explain...	Publication statusE-pub ahead of print - 1 Mar...	https://pureportal.coventry.ac.uk/en/persons/a...	Alireza Zarei	None	NaN
3	https://pureportal.coventry.ac.uk/en/publicati...	CEO Duality and Firm Performance: A	Publication statusE-pub ahead of print -	https://pureportal.coventry.ac.uk/en/persons/m...	Mei Yu	This paper systematically reviews 314	This paper systematically reviews 314

		effectiv...	1998				
634	https://pureportal.coventry.ac.uk/en/publicati...	Market Orientation in the UK Higher Education ...	Publication statusPublished - Jul 1998	NaN	NaN	Null	NaN
635	https://pureportal.coventry.ac.uk/en/publicati...	Developing a comprehensive cross-country econo...	Publication statusPublished - 1997	https://pureportal.coventry.ac.uk/en/persons/s...	Sailesh Tanna	This paper describes a framework for setting u...	This paper describes a framework for setting u...
636	https://pureportal.coventry.ac.uk/en/publicati...	Measuring value added in higher education: a p...	Publication statusPublished - Aug 1995	NaN	NaN	Null	NaN
637	https://pureportal.coventry.ac.uk/en/publicati...	Modelling optimal plant size and market equili...	Publication statusPublished - 1992	NaN	NaN	Null	NaN

638 rows × 7 columns

➔ Below screenshot shows all the publications data is being stored into a csv file. From now we can access the csv file every time instead of crawling all the time. Inorder to update the changes in the website, we can run the crawler program once in a week manually and update the data.



1.1. Number of staff whose publications are crawled (approximately) and the maximum number of Publications per staff.

➔ I fetched the below details from the data frame which I got after running the crawler program. Total 80 authors whose publications are crawled.

```
In [9]: df['author_name'].unique()
```

```
Out[9]: array(['Rashedul Hasan', 'Alireza Zarei', 'Mei Yu', 'Thai Nguyen',
               'Ruth Owusu-Mensah', 'Masud Ibrahim', 'John Ayuk Enombu',
               'Ahmad Abras', 'Hafij Ullah', 'Hany Ahmed', 'Boying Xu',
               'Ibrahim Elmghaamez', 'Andrews Owusu', 'Uchenna Tony-Okeke',
               'Muhammad Shahin Miah', 'Isaiah Oino', 'Francis Awuku Darko',
               'Loai Alsaid', 'Mohammad Khaleq Newaz', nan, 'Huston, S.',
               'Ahmed Saleh', 'Mehdi Hosseini', 'Simon Huston', 'Mehul Chhatbar',
               'Tariq Al Montaser', 'Alaa Alhaj Ismail', 'Angelos Synapis',
               'Mehtap Hisarciklilar', 'Dimitris Serenis', 'Daniel Santamaria',
               'Sarkar Kabir', 'Styliani Panetsidou', 'Sandar Win',
               'Salem Alhababsah', 'Piotr Lis', 'Abdurafiu Noah', 'Junyuan Chen',
               'Mahbub Khan', 'Christopher Muganhu', 'Nikhil Sapre',
               'Luda Ruddock', 'Sailesh Tanna', 'Akin Sharimakin',
               'Simon Horsman', 'Olubunmi Ajala', 'Samir Alamad',
               'Daniel Aghanya', 'Alloysius Egbulonu', 'Ullah, H.',
               'Jaliyyah Bello', 'Eliana Lauretta', 'Ken Baldwin',
               'Faisal Shahzad', 'Pythagoras Petratos', 'Ejike Udeogu',
               'Kenneth Baldwin', 'Graham Sadler', 'Lien Luu', 'Randy Silvers',
               'Judith Kabajulizi', 'Amir Khorasgani',
```

```
In [10]: df['author_name'].nunique()
```

```
Out[10]: 80
```

```
array(['Rashedul Hasan', 'Alireza Zarei', 'Mei Yu', 'Thai Nguyen',
       'Ruth Owusu-Mensah', 'Masud Ibrahim', 'John Ayuk Enombu',
       'Ahmad Abras', 'Hafij Ullah', 'Hany Ahmed', 'Boying Xu',
       'Ibrahim Elmghaamez', 'Andrews Owusu', 'Uchenna Tony-Okeke',
       'Muhammad Shahin Miah', 'Isaiah Oino', 'Francis Awuku Darko',
       'Loai Alsaid', 'Mohammad Khaleq Newaz', nan, 'Huston, S.',
       'Ahmed Saleh', 'Mehdi Hosseini', 'Simon Huston', 'Mehul Chhatbar',
       'Tariq Al Montaser', 'Alaa Alhaj Ismail', 'Angelos Synapis',
       'Mehtap Hisarciklilar', 'Dimitris Serenis', 'Daniel Santamaria',
       'Sarkar Kabir', 'Styliani Panetsidou', 'Sandar Win',
       'Salem Alhababsah', 'Piotr Lis', 'Abdurafiu Noah', 'Junyuan Chen',
       'Mahbub Khan', 'Christopher Muganhu', 'Nikhil Sapre',
       'Luda Ruddock', 'Sailesh Tanna', 'Akin Sharimakin',
       'Simon Horsman', 'Olubunmi Ajala', 'Samir Alamad',
       'Daniel Aghanya', 'Alloysius Egbulonu', 'Ullah, H.',
       'Jaliyyah Bello', 'Eliana Lauretta', 'Ken Baldwin',
       'Faisal Shahzad', 'Pythagoras Petratos', 'Ejike Udeogu',
       'Kenneth Baldwin', 'Graham Sadler', 'Lien Luu', 'Randy Silvers',
       'Judith Kabajulizi', 'Amir Khorasgani',
       'Hafiz Ubaid Ur Rahman Rahmani', 'Wei Song',
       'Jaliyyah Ahmadu-Bello', 'S. Huston', 'Dimitrios Serenis',
       'S.H. Huston', 'Dimitris, Jim Serenis', 'Mohamad Nazri Abd Karim',
       'Tanna, S.', 'Hailin Liao', 'Tanna, S. (ed.)', 'Francis Darko',
       'M. Newaz', 'Jun Wang', 'Mark Holmes', 'Mohamad Nazri Mohamad',
       'Yu Wang', 'Liao, H.', 'Graham V. Sadler'], dtype=object)
```

➔ Maximum number of publications per author

```
In [20]: d = df['publication_title'].groupby(df['author_name']).size()

In [21]: d

Out[21]: author_name
         Abdurafiu Noah        18
         Ahmad Abras            7
         Ahmed Saleh            3
         Akin Sharimakin        2
         Alaa Alhaj Ismail      5
                               ..
         Thai Nguyen            4
         Uchenna Tony-Okeke     4
         Ullah, H.              2
         Wei Song               5
         Yu Wang                2
         Name: publication_title, Length: 80, dtype: int64
```

1.3. Pre-processing tasks performed before passing data to Indexer.

Removing stop words, tokenization and stemming are the three pre-processing tasks that are performed before passing data to indexer.

Screen shots of the code and output:

➔ Below screenshot shows reading data into a csv file and storing in a data frame df and prints the df. Data frame is of size 638 rows X 7 columns

```
In [259]: df = pd.read_csv("crawlerdata.csv")
          print(df)

              Unnamed: 0                                   publication_link  \
          0             0  https://pureportal.coventry.ac.uk/en/publicati...
          1             1  https://pureportal.coventry.ac.uk/en/publicati...
          2             2  https://pureportal.coventry.ac.uk/en/publicati...
          3             3  https://pureportal.coventry.ac.uk/en/publicati...
          4             4  https://pureportal.coventry.ac.uk/en/publicati...
          ..          ...                                                ...
          633         633  https://pureportal.coventry.ac.uk/en/publicati...
          634         634  https://pureportal.coventry.ac.uk/en/publicati...
          635         635  https://pureportal.coventry.ac.uk/en/publicati...
          636         636  https://pureportal.coventry.ac.uk/en/publicati...
          637         637  https://pureportal.coventry.ac.uk/en/publicati...

                                           publication_title  \
          0        A bibliometric review of the Waqf literature
          1    A note on COVID-19 instigated maximum drawdown...
          2    Bank stock valuation theories: do they explain...
          3    CEO Duality and Firm Performance: A Systematic...
          4    CEO Financial Experience and Firms' Earnings M...
          ..                                                 ...
          633  Evaluation of clinical interventions: effectiv...
          634  Market Orientation in the UK Higher Education ...
          635  Developing a comprehensive cross-country econo...
          636  Measuring value added in higher education: a p...
          637  Modelling optimal plant size and market equili...

                                           publication_date  \
```

```
635  https://pureportal.coventry.ac.uk/en/persons/s...   Sailesh Tanna
636                                                  NaN             NaN
637                                                  NaN             NaN

                                              abstract  \
0    Waqf received research attention due to its ab...
1    This study uncovers the impact of the COVID-19...
2                                                  NaN
3    This paper systematically reviews 314 empirica...
4    This study investigates the impact of CEOs' fi...
..                                                 ...
633                                               Null
634                                               Null
635  This paper describes a framework for setting u...
636                                               Null
637                                               Null

                                              content
0    Waqf received research attention due to its ab...
1    This study uncovers the impact of the COVID-19...
2                                                  NaN
3    This paper systematically reviews 314 empirica...
4    This study investigates the impact of CEOs' fi...
..                                                 ...
633                                                NaN
634                                                NaN
635  This paper describes a framework for setting u...
636                                                NaN
637                                                NaN

[638 rows x 8 columns]
```

➔ Below screenshot shows the content column in df is stored in a local variable called content and prints the same to perform pre-processing tasks.

```
In [260]: content = df['content'].astype(str)
          print(content)

          0      Waqf received research attention due to its ab...
          1      This study uncovers the impact of the COVID-19...
          2                                                    nan
          3      This paper systematically reviews 314 empirica...
          4      This study investigates the impact of CEOs' fi...
                                     ...
          633                                                  nan
          634                                                  nan
          635    This paper describes a framework for setting u...
          636                                                  nan
          637                                                  nan
          Name: content, Length: 638, dtype: object
```

➔ The pre-processing tasks like stop words removal, tokenizing and stemming are being performed to the variable content.

```
In [261]: import nltk
          nltk.download("stopwords")
          from nltk.corpus import stopwords

          sw = stopwords.words('english')
          nltk.download("punkt")
          from nltk.tokenize import word_tokenize
          from nltk.stem import PorterStemmer

          ps = PorterStemmer()
          filtered_docs = []
          for doc in content:
              tokens = word_tokenize(doc)
              tmp = ""
              for w in tokens:
                  if w not in sw:
                      tmp += ps.stem(w) + " "
              filtered_docs.append(tmp)

          print(filtered_docs)
```

➔ The output is stored in filtered_docs which is after processing content variable. Below screenshot shows the filtered_doc list.

```
[nltk_data] Downloading package stopwords to C:\Users\Bharathi
[nltk_data]     Kondaveeti\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to C:\Users\Bharathi
[nltk_data]     Kondaveeti\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!

['waqf receiv research attent due abil contribut socio-econom develop . while high volum literari evid islam social financ in
strument avail , research motiv studi find divers . therefor , conduct bibliometr analysi waqf literatur understand pattern d
irect research waqf broadli . we collect 319 articl , review waqf extract scopu databas , cover period exceed 100 year 1914 j
une 2020 . we employ rstudio , vosview , microsoft excel citat analysi , content , network analys . a systemat review recent
public complement bibliometr analysi . alongsid reveal relev scientif actor waqf literatur , research waqf malaysia ; conduct
studi wide cash waqf . as result , identifi four research theme waqf studi includ ( 1 ) cash waqf endow , ( 2 ) islam account
waqf , ( 3 ) waqf islam social financ , ( 4 ) govern waqf endow . thi first studi provid bibliometr review waqf literatur add
ress gap exist research offer direct futur research could benefit early-car islam financ researchers.rashedul hasan ', 'thi s
tudi uncov impact covid-19 islam equiti market compar convent counterpart . the extrem large-scal drawdown across market sign
ifi indiscrimin impact . to extent , asian islam market show rel resili counterpart . both islam non-islam asian market signp
ost quicker recoveri rest region , middl east & africa , europ , america . it appear higher return lead smaller maximum drawd
own , higher volatil lead larger maximum drawdown . despit large-scal drawdown , number market secur posit return islam marke
```

➔ The filtered_docs is stored into a new column named ‘Fil_content’ in the data frame.

```
In [262]: df['fil_content'] = filtered_docs

In [263]: df
```

Out[263]:

	publication_link	publication_title	publication_date	author_link	author_name	abstract	content	fil_cor
	pureportal.coventry.ac.uk/en/publicati...	A bibliometric review of the Waqf literature	Publication statusPublished - Jun 2022	https://pureportal.coventry.ac.uk/en/persons/r...	Rashedul Hasan	Waqf received research attention due to its ab...	Waqf received research attention due to its ab...	re rese attent contril
	pureportal.coventry.ac.uk/en/publicati...	A note on COVID-19 instigated maximum drawdown...	Publication statusPublished - May 2022	https://pureportal.coventry.ac.uk/en/persons/r...	Rashedul Hasan	This study uncovers the impact of the COVID-19...	This study uncovers the impact of the COVID-19...	thi u im covi islam

			Performance: A Systematic...	ahead of print - 25 Ma...			reviews empiric
4	4	https://pureportal.coventry.ac.uk/en/publicati...	CEO Financial Experience and Firms' Earnings M...	Publication statusSubmitted - 7 Mar 2022	https://pureportal.coventry.ac.uk/en/persons/t...	Thai Nguyen	This st investiga the impac CEOs'
...	...	...	...	...	...	...	
633	633	https://pureportal.coventry.ac.uk/en/publicati...	Evaluation of clinical interventions: effectiv...	Publication statusPublished - 1998	NaN	NaN	N
634	634	https://pureportal.coventry.ac.uk/en/publicati...	Market Orientation in the UK Higher Education ...	Publication statusPublished - Jul 1998	NaN	NaN	N
635	635	https://pureportal.coventry.ac.uk/en/publicati...	Developing a comprehensive cross-country econo...	Publication statusPublished - 1997	https://pureportal.coventry.ac.uk/en/persons/s...	Sailesh Tanna	This pa describe framework setting
636	636	https://pureportal.coventry.ac.uk/en/publicati...	Measuring value added in higher education: a p...	Publication statusPublished - Aug 1995	NaN	NaN	N
637	637	https://pureportal.coventry.ac.uk/en/publicati...	Modelling optimal plant size and market equili...	Publication statusPublished - 1992	NaN	NaN	N

638 rows × 9 columns

1.4. The crawler operates, e.g. scheduled or run manually

- ➔ My crawler program will run manually by the programmer.

1.5. Explanation of crawler it works.

- ➔ iniitially the user gives input url “https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/” and sends input to “level_crawler” function.
- ➔ Created a data frame columns = ["publication_link","publication_title","publication_date","author_link","author_name","abstract", "content"] to store all publication details in the data frame.
- ➔ Then using beautiful_soup_object.findAll("a") it fetches the publications link and publications title and stores in a data frame and the each publications link is sent as input to “get_author_details” function.
- ➔ In get_author_details function, using the each publication link as input, to fetch author details I used soup.findAll('a',{'class':'link person'}) and got author name and author link. With this we will get all the author details.
- ➔ Inorder to fetch only SEFA authors, we need to check for (sefa == 'School of Economics, Finance and Accounting') and when it’s true only we will store the author details and author link in data frame.
- ➔ Then to get abstract I used (soup.find('div',{'class':'textblock'}) and fetched abstract from the each Publications link and sent to a data frame .
- ➔ Also fetched publication status and date using soup.find('tr',{'class':'status'}) for each publication and Store in a data frame.
- ➔ After fetching and writing all the details of first publication in to the first row of data frame , the code automatically fetches the 2nd publication details and so on till the end of the input page.
- ➔ After fetching all 50 publications in the first page, the input url will be updated automatically With the 2nd page URL using the pages logic. And it will continues to crawl all the pages till the last publication.
- ➔ After completing the crawler program, I added abstract, author_name and publications_title data and stored to the content column [‘content'] = ['abstract'] + ['author_name']+[‘publication_title’].
- ➔ Now the data frame is stored into an excel file named “crawlerdata.csv”. As all the crawled data is sent and stored in to csv file, from now we can access csv file and use the publications data.

- ➔ To perform pre-processing tasks, first we need to read the csv file in to a data frame df which contains all the publication details.
- ➔ I am implementing pre-processing like stop word removal, tokenization and stemming to the 'content' column which contains abstract and author name. So I extracted content column and stored in a list content.
- ➔ Then implemented the stop word removal, tokenization and stemming to the content variable and stored in the data frame into a new column name "fil_content". From now we can use the fil_content column in the data frame.

2. Indexer

2.1. Whether you implemented the index or used Elastic Search (note that if Elastic Search is used you will lose the 15 marks for index construction, but the project becomes easier).

- ➔ In this project I implemented indexer on the fil_content column of data frame df and stored in a dictionary.

2.2. If you implemented it, which data structure is used (for example, incidence matrix or inverted index)

- ➔ In this project I implemented inverted index data structure

2.3. If you implemented it, whether it is incremental, i.e. it grows and gets updated over the time, or it is constructed from scratch every time your crawler is run

- ➔ For each time I run the crawler, the csv file gets updated and the inverted indexer code is executed manually. It is implemented from the scratch every time I run the crawler.

2.4. If you implemented it, show some part of its content (e.g. the constructed dictionary).

Screenshots of code:

```
In [264]: inverted_index = {}

          for i, doc in enumerate(df['fil_content']):
              for term in doc.split(" "):
                  if term in inverted_index:
                      inverted_index[term].add(i)
                  else: inverted_index[term] = {i}
          inverted_index
```

```
Out[264]: {'waqf': {0},
           'receiv': {0, 292, 423, 434, 457, 536},
           'research': {0,
            3,
            9,
            24,
            28,
            32,
            34,
            35,
            38,
            43,
            46,
            49,
            55,
            56,
            59,
            60,
            71,
            72,
            81,
            86,
            107
```

```
 535},
'attent': {0, 117, 197, 277, 316, 434, 495},
'due': {0,
 3,
 5,
 22,
 47,
 54,
 72,
 185,
 187,
 203,
 217,
 231,
 316,
 318,
 459,
 485,
 491,
 508,
 548,
 560},
'abil': {0, 5, 6, 28, 43, 54, 79, 202, 231, 276, 434},
'contribut': {0,
 4,
 16,
 18,
```

```
 626},
'variables.andrew': {13},
'owusu': {13, 28, 52, 81, 165, 183, 185, 249, 430, 441, 442, 536},
'transnat': {14, 316},
'i.e.': {14, 317, 519},
'cultur': {14, 24, 95, 216, 311, 452, 587},
'educ': {14, 17, 24, 30, 58, 79, 81, 155, 175, 210, 216, 301, 318, 477},
'legal': {14, 53, 95, 106, 231, 298, 548},
'diffus': {14, 40, 56},
'isa': {14},
'162': {14},
'1995–2014': {14, 95},
'3240': {14},
'draw': {14, 58, 59, 60, 64, 117, 129, 371, 434, 467, 587},
'distinct': {14, 47, 107, 187, 459},
'theori': {14,
 40,
 49
```

```
 626},
'recruit': {17},
'right': {17, 30, 95, 147, 249, 448},
'age.isaiah': {17},
'oino': {17, 54, 96, 202, 276, 277, 282, 488},
'extra-financi': {18},
'narr': {18, 33},
'commentari': {18, 447},
'section': {18, 30, 58},
'valuat': {18, 141, 295, 389, 448, 459},
'extern': {18, 82, 95, 147, 174, 197, 305, 316, 385, 386, 441, 442},
'31,327': {18},
'63': {18, 488},
'2003–2019': {18},
'pronounc': {18, 32, 106, 141, 229, 317},
'analyst': {18, 47, 187, 317},
'coverag': {18, 243},
'maker': {18, 82, 85, 342, 386, 401, 430, 485, 487, 508},
'throughout': {18, 371},
'world.muhammad': {18},
'burden': {19, 385, 386},
'non-communic': {19},
'diseas': {19, 135, 385, 386},
'ncd': {19},
'sub-saharan': {19, 82},
'surg': {19},
'last': {19, 318, 447, 570},
'decad': {19, 21, 24, 170, 174, 235, 295, 316, 434},
'disabl': {19},
'adjust': {19, 35, 166, 181, 309, 488},
'life': {19, 332},
'dali': {19},
```

2.5. Explanation of how it works

```
In [264]: inverted_index = {}

for i, doc in enumerate(df['fil_content']):
    for term in doc.split(" "):
        if term in inverted_index:
            inverted_index[term].add(i)
        else: inverted_index[term] = {i}
inverted_index
```

```
Out[264]: {'waqf': {0},
 'receiv': {0, 292, 423, 434, 457, 536},
 'research': {0,
  3,
  9,
  24,
  28,
  32,
  34,
  35,
  38,
  43,
  46,
  49,
  55,
  56,
  59,
  60,
  71,
  72,
  81,
  86,
  107
```

- ➔ The output from the crawler program and pre-processing tasks which is the csv file, the column 'fil_content' is used as input to the indexer.
- ➔ An empty dictionary named inverted_index is created. The for loop is implemented on fil_content column for each row of fil_content and splits by space. The term in each doc is checked in inverted index.
- ➔ If it's already present in inverted index, it adds the row number (i.e., publication number) to the list of values in the dictionary where key is the term.
- ➔ If the term is not present in the inverted index dictionary, the term will be added as a key and value is the row number.
- ➔ By the end of the for loop, the inverted index dictionary is created having key as words of the fil_content from all publications and values as list of rows in which the word exists.

3. Query processor

3.1. Pre-processing tasks are applied to a given query

For query processor, I applied pre-processing tasks like stop words removal, tokenization and stemming.

3.2. Do you only support Boolean queries (using AND, OR, NOT, etc.) or accept keywords like Google does (without any need for AND, OR, NOT etc.)

I did not used Boolean queries. My query processor accepts keywords like google does.

3.3. If Elastic Search is used, how you convert a user query to an appropriate query for Elastic Search

Did not used elastic search.

3.4. If Elastic Search is NOT used, whether or not you perform ranked retrieval; if yes, specify whether or not you used vector space and the method used to calculate the ranks

I did not perform ranked retrieval.

3.5. Demonstration of the running system (use screenshots in you report and run your software in your viva). You must run your system on numerous and various input queries to prove the accuracy and robustness of your system. For example, you must use appropriate queries to prove your system performs stop-word removal and stemming and ranked retrieval.

Code and output screenshots of query processor.

Screenshots of code:

```
query = input('enter input query:')

import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords

sw = stopwords.words('english')
nltk.download("punkt")
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer

ps = PorterStemmer()
filtered_docs = []
tokens = word_tokenize(query)
tmp = ""
for w in tokens:
    if w not in sw:
        tmp += ps.stem(w) + " "
filtered_docs.append(tmp)

#print(filtered_docs)

x = filtered_docs[0].split(" ")

q = []
for i in range(len(x)):
    if len(x[i]) > 0:
        q.append(x[i])
#print(q)
```

```
p = []
for i in range(len(q)):
    if (q[i] in inverted_index):
        a = inverted_index[q[i]]
        l = sorted(a)
        p.append(l)
        print('l = ',q[i],l)
    else:
        print(q[i],'not found')
for i in p:
    print(i)
for i  in range(len(p)):
    print(p[i])
    for j in range(len(p[i])):
        print(p[i][j])
        print(df.loc[p[i][j]])
```

Output screenshots for long and short queries:

➔ **Long Query 1:**

- User entered query as input.

```
enter input query:impact of the COVID-19 on the Islamic equity markets compared to their conventional counterparts.
[nltk_data] Downloading package stopwords to C:\Users\Bharathi
[nltk_data]     Kondaveeti\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to C:\Users\Bharathi
[nltk_data]     Kondaveeti\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
l =  impact [1, 4, 6, 9, 12, 13, 14, 17, 22, 24, 25, 26, 30, 32, 34, 41, 45, 53, 54, 58, 59, 60, 62, 64, 65, 71, 72, 75, 87,
89, 90, 95, 106, 131, 136, 141, 147, 152, 153, 157, 161, 166, 167, 168, 170, 182, 197, 202, 203, 210, 216, 220, 229, 231, 23
7, 243, 267, 272, 276, 277, 289, 298, 302, 305, 316, 317, 318, 342, 379, 385, 386, 401, 425, 430, 457, 467, 485, 488, 508, 51
9, 535, 560, 570, 571, 610]
l =  covid-19 [1, 22, 35, 72, 87, 135]
l =  islam [0, 1, 9, 35, 41, 43, 47, 64, 68, 86, 103, 140, 174, 181, 187, 217, 231, 261, 295, 359, 368, 423, 491, 495, 516, 5
19]
l =  equiti [1, 10, 26, 47, 48, 57, 141, 181, 187, 243, 368, 459, 516]
l =  market [1, 5, 6, 12, 22, 27, 32, 33, 35, 40, 41, 47, 48, 49, 54, 59, 60, 62, 68, 70, 71, 73, 79, 82, 87, 94, 103, 140, 1
52, 153, 174, 182, 187, 220, 231, 237, 261, 267, 289, 295, 298, 302, 309, 315, 318, 333, 355, 359, 388, 389, 416, 424, 425, 4
37, 448, 459, 487, 508, 516, 519, 583, 587, 603, 629, 632]
```

- My query processor fetches all the publications related to the query. The first result is the most related query.

```
1
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title    A note on COVID-19 instigated maximum drawdown...
publication_date               Publication statusPublished - May 2022
author_link          https://pureportal.coventry.ac.uk/en/persons/r...
author_name                                              Rashedul Hasan
abstract             This study uncovers the impact of the COVID-19...
content              This study uncovers the impact of the COVID-19...
Name: 1, dtype: object
4
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title    CEO Financial Experience and Firms' Earnings M...
publication_date             Publication statusSubmitted - 7 Mar 2022
author_link          https://pureportal.coventry.ac.uk/en/persons/t...
author_name                                                 Thai Nguyen
abstract             This study investigates the impact of CEOs' fi...
content              This study investigates the impact of CEOs' fi...
Name: 4, dtype: object
6
publication_link     https://pureportal.coventry.ac.uk/en/publicati
```

- Input query sentence is taken from the below publication and got the same publication as output to the query and all the below outputs are the related information.

A note on COVID-19 instigated maximum drawdown in Islamic markets versus conventional counterparts

M. Kabir Hassan, Md Iftekhar Hasan Chowdhury, Faruk Balli, Rashedul Hasan

School of Economics, Finance and Accounting

Research output: Contribution to journal › Article › peer-review

Overview Fingerprint

Abstract

This study uncovers the impact of the COVID-19 on the Islamic equity markets compared to their conventional counterparts. The extremely large-scale drawdown across the markets signifies an indiscriminate impact. To some extent, Asian Islamic markets show relative resilience to their counterparts. Both Islamic and non-Islamic Asian markets signpost a quicker recovery than the rest of the regions, the Middle East & Africa, Europe, and America. It appears that a higher return leads to a smaller maximum drawdown, while higher volatility leads to a larger maximum drawdown. Despite the large-scale drawdown, a

➔ **Long Query 2:**

- User entered query as input.

```
enter input query:a longitudinal research design, and an integrative framework to better control the direct and moderating fa
ctors
[nltk_data] Downloading package stopwords to C:\Users\Bharathi
[nltk_data]     Kondaveeti\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to C:\Users\Bharathi
[nltk_data]     Kondaveeti\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
l =  longitudin [3, 49, 120, 121, 222]
l =  research [0, 3, 9, 24, 28, 32, 34, 35, 38, 43, 46, 49, 55, 56, 59, 60, 71, 72, 81, 86, 107, 117, 120, 121, 129, 136, 16
5, 170, 185, 202, 203, 216, 222, 235, 261, 292, 295, 297, 309, 311, 316, 317, 318, 333, 342, 359, 363, 389, 401, 428, 430, 43
4, 448, 454, 456, 470, 485, 502, 507, 516, 548, 560, 571, 587, 589, 626, 635]
l =  design [3, 9, 42, 50, 71, 161, 165, 170, 182, 200, 261, 332, 363, 370, 487, 629]
l =  , [0, 1, 3, 4, 5, 6, 7, 9, 10, 12, 13, 14, 16, 17, 18, 19, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34, 35, 38, 39, 40, 4
1, 42, 43, 45, 46, 47, 48, 49, 50, 52, 53, 54, 55, 56, 57, 58, 59, 60, 62, 64, 65, 67, 68, 70, 71, 72, 73, 74, 75, 79, 80, 8
1, 82, 85, 86, 87, 89, 90, 91, 93, 94, 95, 96, 103, 105, 106, 107, 114, 117, 120, 121, 129, 131, 135, 136, 137, 140, 141, 14
2, 147, 153, 155, 157, 161, 165, 166, 167, 168, 170, 174, 175, 178, 181, 182, 185, 187, 193, 197, 200, 202, 203, 210, 216, 21
7, 218, 220, 222, 229, 231, 235, 236, 237, 243, 249, 261, 265, 267, 271, 272, 276, 277, 282, 289, 291, 292, 295, 297, 298, 30
```

- My query processor fetches all the publications related to the query. The first result is the most related query.

```
3
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title    CEO Duality and Firm Performance: A Systematic...
publication_date     Publication statusE-pub ahead of print - 25 Ma...
author_link          https://pureportal.coventry.ac.uk/en/persons/m...
author_name                                                    Mei Yu
abstract             This paper systematically reviews 314 empirica...
content              This paper systematically reviews 314 empirica...
Name: 3, dtype: object
49
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title    A Historical Institutionalist Perspective on t...
publication_date             Publication statusPublished - 26 Jan 2021
author_link          https://pureportal.coventry.ac.uk/en/persons/s...
author_name                                                  Sandar Win
abstract             Purpose: Many transition economies are former ...
content              Purpose: Many transition economies are former ...
Name: 49, dtype: object
120
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
```

- Input query sentence is taken from the below publication and got the same publication as output to the query and all the below outputs are the related information.

CEO Duality and Firm Performance: A Systematic Review and Research Agenda

Mei Yu

Research Centre for Financial & Corporate Integrity, School of Economics, Finance and Accounting

Research output: Contribution to journal › Review article › peer-review

Overview Fingerprint

Abstract

This paper systematically reviews 314 empirical studies which examine the relationship between board leadership structure and firm performance. The results show that the mixed research findings are due to different firm performance measurements, research designs, sampling practices and approaches of dealing with endogeneity issues. Studies utilizing multiple-country data, multinational companies, small firms or regions covering Africa, the Middle East, Eastern-Europe and South America are under-represented. It critiques the methodological weaknesses of some empirical studies and proposes that future researchers should use multiple firm performance measurements, a longitudinal research design, and an integrative framework to better control the direct and moderating factors. Future researchers may advance the robustness of research by using multiple theoretical lenses to consider how various governance factors moderate the relationship between board leadership structure and firm performance, particularly the different ownership structures, managerial discretion contexts and national institutional factors.

➔ Long Query 3:

```
enter input query:Revisiting the evidence of earnings management prior to merger announcements: an application of Benford's l
aw
[nltk_data] Downloading package stopwords to C:\Users\Bharathi
[nltk_data]     Kondaveeti\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to C:\Users\Bharathi
[nltk_data]     Kondaveeti\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
l =  revisit [291]
l =  evid [0, 9, 16, 18, 22, 26, 30, 32, 35, 41, 46, 52, 53, 57, 58, 62, 65, 67, 69, 70, 71, 85, 87, 90, 106, 117, 131, 136,
140, 141, 152, 153, 168, 174, 175, 182, 200, 210, 217, 220, 229, 236, 237, 271, 272, 291, 305, 315, 322, 328, 332, 334, 373,
389, 412, 416, 424, 428, 430, 441, 442, 446, 454, 458, 475, 485, 486, 487, 495, 507, 514, 517, 519, 524, 548, 591, 603, 626]
l =  earn [4, 28, 67, 96, 106, 136, 137, 217, 291]
l =  manag [4, 7, 9, 18, 21, 27, 28, 34, 41, 49, 59, 69, 73, 93, 106, 107, 117, 131, 136, 137, 147, 170, 181, 200, 202, 207,
210, 222, 231, 235, 276, 291, 316, 332, 363, 389, 401, 423, 428, 441, 448, 491]
l =  prior [28, 33, 56, 75, 94, 135, 136, 137, 141, 218, 220, 291, 328, 342, 626]
l =  merger [73, 94, 136, 137, 153, 220, 291, 334, 459, 498]
l =  announc [33, 73, 94, 136, 137, 153, 291, 424, 524]
```

```
291
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title    Revisiting the evidence of earnings management...
publication_date           Publication statusSubmitted - 26 Dec 2018
author_link          https://pureportal.coventry.ac.uk/en/persons/t...
author_name                                              Thai Nguyen
abstract             The most popularly-used Jones-type models are ...
content              The most popularly-used Jones-type models are ...
fil_content          the popularly-us jones-typ model current subje...
Name: 291, dtype: object
```

Revisiting the evidence of earnings management prior to merger announcements: an application of Benford's law

Thai Nguyen, Hanh Thi My Le (Editor), Nguyet Nguyen (Editor), Chau Duong (Editor)

School of Economics, Finance and Accounting

Research output: Contribution to conference › Paper › peer-review

➔ **Short Query 1:**

- User entered query as input. My query processor fetches all the publications related to the query. The first result is the most related query.

```
enter input query:Waqf literature
l =  waqf [0]
l =  literatur [0, 5, 10, 16, 21, 30, 35, 38, 40, 42, 43, 46, 47, 53, 55, 65, 67, 75, 86, 93, 107, 120, 121, 129, 136, 137, 1
40, 141, 182, 187, 229, 289, 295, 302, 311, 316, 317, 363, 424, 447, 495, 510, 516, 587, 632]
[0]
[0, 5, 10, 16, 21, 30, 35, 38, 40, 42, 43, 46, 47, 53, 55, 65, 67, 75, 86, 93, 107, 120, 121, 129, 136, 137, 140, 141, 182, 1
87, 229, 289, 295, 302, 311, 316, 317, 363, 424, 447, 495, 510, 516, 587, 632]
[0]
0
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title          A bibliometric review of the Waqf literature
publication_date                 Publication statusPublished - Jun 2022
author_link          https://pureportal.coventry.ac.uk/en/persons/r...
author_name                                            Rashedul Hasan
abstract             Waqf received research attention due to its ab...
content              Waqf received research attention due to its ab...
Name: 0, dtype: object
[0, 5, 10, 16, 21, 30, 35, 38, 40, 42, 43, 46, 47, 53, 55, 65, 67, 75, 86, 93, 107, 120, 121, 129, 136, 137, 140, 141, 182, 1
87, 229, 289, 295, 302, 311, 316, 317, 363, 424, 447, 495, 510, 516, 587, 632]
```

- Input query sentence is taken from the below publication and got the same publication as output to the query and all the below outputs are the related information.

A bibliometric review of the Waqf literature

Muneer M. Alshater, M. Kabir Hassan, Mamunur Rashid, Rashedul Hasan

School of Economics, Finance and Accounting

Research output: Contribution to journal › Article › peer-review

➔ **Short Query 2:**

```
enter input query:Informal Entrepreneurship in Africa
l =  inform [7, 10, 13, 26, 33, 43, 47, 52, 56, 58, 67, 70, 71, 90, 117, 152, 178, 187, 218, 222, 261, 265, 289, 297, 332, 35
9, 373, 388, 389, 437, 485, 583, 587, 632]
l =  entrepreneurship [13, 43, 261]
l =  africa [1, 3, 13, 19, 27, 82, 114, 302, 315, 502]
[7, 10, 13, 26, 33, 43, 47, 52, 56, 58, 67, 70, 71, 90, 117, 152, 178, 187, 218, 222, 261, 265, 289, 297, 332, 359, 373, 388,
389, 437, 485, 583, 587, 632]
[13, 43, 261]
[1, 3, 13, 19, 27, 82, 114, 302, 315, 502]
[7, 10, 13, 26, 33, 43, 47, 52, 56, 58, 67, 70, 71, 90, 117, 152, 178, 187, 218, 222, 261, 265, 289, 297, 332, 359, 373, 388,
389, 437, 485, 583, 587, 632]
7
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title    Clinicians' informal acquisition of accounting...
publication_date     Publication statusE-pub ahead of print - 22 Ju...
author_link          https://pureportal.coventry.ac.uk/en/persons/j...
author_name                                          John Ayuk Enombu
abstract             This article discusses how clinicians acquire ...
content              This article discusses how clinicians acquire ...
```

```
publication_title    Clinicians' informal acquisition of accounting...
publication_date     Publication statusE-pub ahead of print - 22 Ju...
author_link          https://pureportal.coventry.ac.uk/en/persons/j...
author_name                                          John Ayuk Enombu
abstract             This article discusses how clinicians acquire ...
content              This article discusses how clinicians acquire ...
Name: 7, dtype: object
10
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title     Corporate hedging and the cost of equity capital
publication_date                    Publication statusPublished - 2022
author_link          https://pureportal.coventry.ac.uk/en/persons/h...
author_name                                                 Hany Ahmed
abstract             Using a large panel of UK public firms, we exa...
content              Using a large panel of UK public firms, we exa...
Name: 10, dtype: object
13
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title    Determinants of Informal Entrepreneurship in A...
```

Determinants of Informal Entrepreneurship in Africa

Amanze Ejiogu, Obiora Okechukwu, Chibuzo Ejiogu, Andrews Owusu, Ogechi Adeola

Faculty of Business & Law, School of Economics, Finance and Accounting

Research output: Contribution to journal › Article › peer-review

➔ **Short Query 3:**

```
enter input query:corporate governance research
```

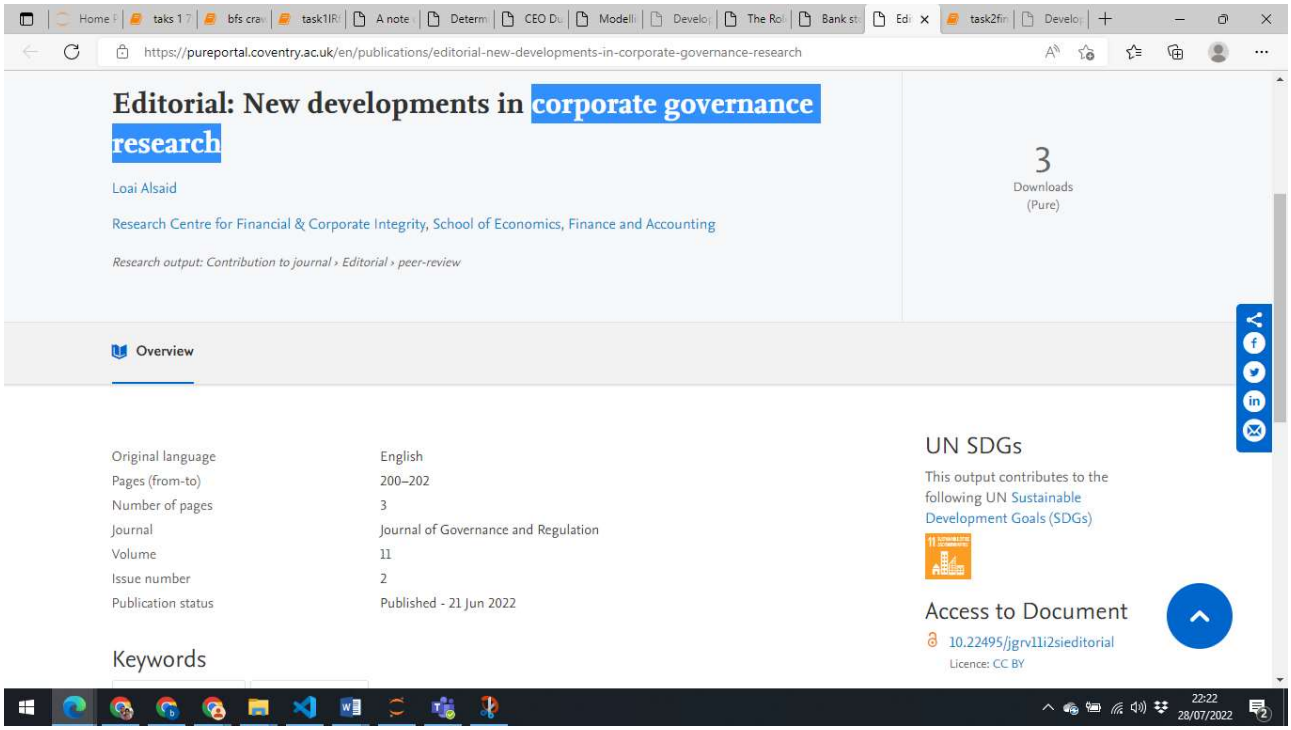
```
[nltk_data] Downloading package stopwords to C:\Users\Bharathi
[nltk_data]     Kondaveeti\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to C:\Users\Bharathi
[nltk_data]     Kondaveeti\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
l =  corpor [4, 5, 9, 12, 17, 18, 20, 25, 30, 32, 40, 49, 53, 56, 58, 59, 60, 62, 73, 75, 85, 129, 131, 157, 161, 167, 175, 1
82, 193, 202, 217, 249, 276, 295, 316, 317, 333, 424, 430, 441, 442, 443, 446, 448, 454, 459, 493, 532, 536, 538, 548, 554, 6
13]
l =  govern [0, 3, 4, 9, 12, 14, 18, 20, 25, 30, 32, 34, 48, 49, 52, 53, 56, 59, 62, 64, 65, 69, 70, 71, 82, 85, 93, 95, 107,
114, 117, 125, 126, 131, 147, 157, 165, 182, 185, 193, 202, 227, 243, 249, 263, 276, 301, 316, 317, 334, 342, 370, 379, 385,
386, 388, 430, 434, 441, 442, 446, 452, 454, 457, 485, 536, 538, 548, 555, 587, 610, 624, 626]
l =  research [0, 3, 9, 20, 21, 24, 28, 32, 34, 35, 38, 43, 46, 49, 55, 56, 59, 60, 71, 72, 81, 86, 107, 117, 120, 121, 129,
136, 165, 170, 185, 202, 203, 216, 222, 235, 261, 292, 295, 297, 309, 311, 316, 317, 318, 333, 342, 359, 363, 389, 401, 428,
430, 434, 448, 454, 456, 470, 485, 502, 507, 516, 548, 560, 571, 587, 589, 626, 635]
[4, 5, 9, 12, 17, 18, 20, 25, 30, 32, 40, 49, 53, 56, 58, 59, 60, 62, 73, 75, 85, 129, 131, 157, 161, 167, 175, 182, 193, 20
2, 217, 249, 276, 295, 316, 317, 333, 424, 430, 441, 442, 443, 446, 448, 454, 459, 493, 532, 536, 538, 548, 554, 613]
```

```
4
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title    CEO Financial Experience and Firms' Earnings M...
publication_date              Publication statusSubmitted - 7 Mar 2022
author_link          https://pureportal.coventry.ac.uk/en/persons/t...
author_name                                                 Thai Nguyen
abstract             This study investigates the impact of CEOs' fi...
content              This study investigates the impact of CEOs' fi...
fil_content          thi studi investig impact ceo ' financi experi...
Name: 4, dtype: object
5
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title                                CEO tenure and cost of debt
publication_date     Publication statusE-pub ahead of print - 21 Ju...
author_link          https://pureportal.coventry.ac.uk/en/persons/r...
author_name                                           Ruth Owusu-Mensah
abstract             In this study, we investigate the relationship...
content              In this study, we investigate the relationship...
fil_content          in studi , investig relationship ceo tenur cos...
Name: 5, dtype: object
```

```
content              We examine the association of integrated finan...
fil_content          we examin associ integr financi extra-financi ...
Name: 18, dtype: object
20
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title    Editorial: New developments in corporate gover...
publication_date            Publication statusPublished - 21 Jun 2022
author_link          https://pureportal.coventry.ac.uk/en/persons/l...
author_name                                                 Loai Alsaid
abstract                                                          Null
content              NullLoai AlsaidEditorial: New developments in ...
fil_content          nullloai alsaideditori : new develop corpor go...
Name: 20, dtype: object
25
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title       Gender Diversity and Financial Statement Fraud
publication_date              Publication statusPublished - 1 Mar 2022
author_link          https://pureportal.coventry.ac.uk/en/persons/m...
author_name                                                     Mei Yu
abstract             This study investigates the role of gender div...
```

- Input is taken from the below publication with no abstract. As the fil_content column contains abstract + publication name + author name , I can able to fetch details of publications.



➔ **<u>Short Query 3:</u>**

```
enter input query:Historical Institutionalist Perspective
l =  histor [21, 49, 55, 120, 121, 261, 295, 309, 328, 371, 447, 510, 570, 626]
l =  institutionalist [49]
l =  perspect [22, 34, 39, 40, 49, 50, 59, 65, 71, 82, 90, 138, 139, 140, 189, 197, 207, 222, 238, 261, 274, 302, 309, 316, 3
22, 387, 415, 424, 434, 510]
[21, 49, 55, 120, 121, 261, 295, 309, 328, 371, 447, 510, 570, 626]
[49]
[22, 34, 39, 40, 49, 50, 59, 65, 71, 82, 90, 138, 139, 140, 189, 197, 207, 222, 238, 261, 274, 302, 309, 316, 322, 387, 415,
424, 434, 510]
[21, 49, 55, 120, 121, 261, 295, 309, 328, 371, 447, 510, 570, 626]
21
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title    Evolution of research in finance over the last...
publication_date               Publication statusPublished - Jan 2022
author_link          https://pureportal.coventry.ac.uk/en/persons/r...
author_name                                          Rashedul Hasan
abstract             A standard literature review focuses on the hi...
content              A standard literature review focuses on the hi...
fil_content          a standard literatur review focus histor devel...
```

```
publication_title    Evolution of research in finance over the last...
publication_date               Publication statusPublished - Jan 2022
author_link          https://pureportal.coventry.ac.uk/en/persons/r...
author_name                                          Rashedul Hasan
abstract             A standard literature review focuses on the hi...
content              A standard literature review focuses on the hi...
fil_content          a standard literatur review focus histor devel...
Name: 21, dtype: object
49
publication_link     https://pureportal.coventry.ac.uk/en/publicati...
publication_title    A Historical Institutionalist Perspective on t...
publication_date             Publication statusPublished - 26 Jan 2021
author_link          https://pureportal.coventry.ac.uk/en/persons/s...
author_name                                              Sandar Win
abstract             Purpose: Many transition economies are former ...
content              Purpose: Many transition economies are former ...
fil_content          purpos : mani transit economi former socialist...
Name: 49, dtype: object
55
```

A Historical Institutionalist Perspective on the Persistence of State Controls during Financial Sector Reforms: The Insightful Case of Myanmar

Sandar Win, Alexander Kofinas

School of Economics, Finance and Accounting

Research output: Contribution to journal › Article › peer-review

➔ **Short Query 4:**

- When I gave a word which is not in any of the publications my query is printing not found message.

```
        q.append(x[i])
#print(q)

p = []
for i in range(len(q)):
    if (q[i] in inverted_index):
        a = inverted_index[q[i]]
        l = sorted(a)
        p.append(l)
        print('l = ',q[i],l)
    else:
        print(q[i],'not found')
for i in p:
    print(i)
for i  in range(len(p)):
    print(p[i])
    for j in range(len(p[i])):
        print(p[i][j])
        print(df.loc[p[i][j]])
```

```
enter input query:bharathi
bharathi not found
```

3.6. Brief explanation of how it works

I took input from the user whether it's long or short query, the pre- processing tasks are applied to the input query and splits all the words in to a list and removes any empty spaces in the list as words. Then the final list with all words will be searched in the dictionary which is inverted_index and gets the row numbers as list which are the publication details of the data frame. Then after retrieving row information I printed all the details of the publications. In my query processor, once the user enters the query it will automatically fetches the related publications. If the user enters a query which is not in any of the publications, user will get the output message "not found".

References:

Web crawling using Breadth First Search at a specified depth - GeeksforGeeks

https://anvil.works/blog/how-to-build-a-search-engine

https://www.youtube.com/user/thenewboston/videos

https://github.com/Dev-Elie/Web-Crawler-for-Google-Scholar/blob/main/app.py

https://elhamamini.medium.com/how-to-build-a-vertical-search-engine-using-python-f09b137b5db

https://towardsdatascience.com/create-a-simple-search-engine-using-python-412587619ff5

http://aakashjapi.com/fuckin-search-engines-how-do-they-work/

https://www.youtube.com/watch?v=bFrO8piASKg

https://www.scrapingbee.com/blog/crawling-python/

https://www.scrapingbee.com/blog/crawling-python/

http://www.cs.put.poznan.pl/alabijak/ezi/lab1/Lab1-Crawling-Python.pdf

https://github.com/wesdoyle/Javelin/blob/master/Notebooks/Inverted%20Index.ipynb

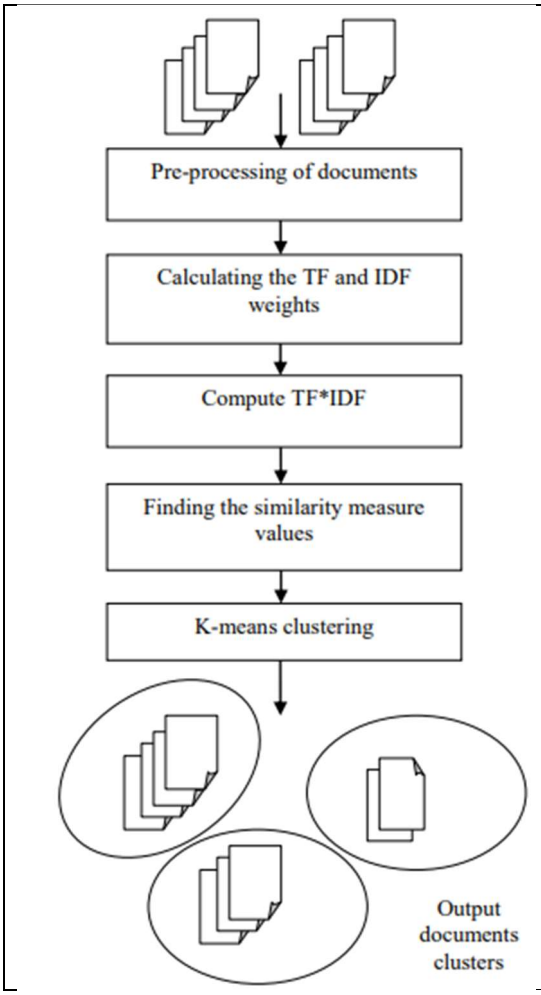
Task 2. Document Clustering

Q1: How and how many input documents are collected

- ➔ I implemented web crawling using Beautiful Soup to pull the text data that belongs to three different categories namely Sport, Health and Politics.
- ➔ I manually collected BBC news website links from the three categories and gave inputs in my crawler function to pull the data automatically from those links and stored all the documents (each document is atleast one sentence) in a list and names as docs.
- ➔ Total number of documents collected are above 110.

Q2: Which document clustering method (e.g. K-means with appropriate K value) has been used and how its performance is measured?

- ➔ Document clustering is a sub area of data clustering which includes concepts from information retrieval, natural language processing, and machine learning. The main aim of document clustering method is find out natural groupings of documents from a given collection of documents.
- ➔ The main goal is to minimize the computational overheard by creating more accurate clusters. There are many kinds of techniques for achieving these desired properties. The two main algorithms that are used in clustering are Hierarchical clustering and K-means clustering techniques. Hierarchical clustering is slower than K-means and sometimes combination of these two also used for good results.
- ➔ Hierarchical clustering techniques produce a nested sequence of partitions or a cluster of hierarchy or tree of clusters. This structure is also called as dendrogram. In this structure every node has child and sibling clusters. The main advantages of hierarchical clustering are their flexibility and ease of handling any forms of similarity. But they suffer from vagueness in the termination criteria.
- ➔ K-means is a partitioning relocation clustering method which divides data into several subsets. When we are using K-means we are using a centroid which is the mean value of all points within the cluster. This centroid represents the cluster formed and this helps the K-means methods to produce clusters in a faster rate than hierarchical methods.
- ➔ K-means provides a faster way to create cluster from a set of random documents. It calculates the vector value for each document from the vector space and based their value new clusters are formed. Since it uses the TDIDF and cosine measure the final produced clusters are always good in terms of both intra and inter cluster similarity
- ➔ So I have used the K-Means algorithm here to generate clusters. K-means clustering is a type of unsupervised learning method, which is used when we don't have labelled data as in our case, we have unlabelled data (means, without defined categories or groups). The goal of this algorithm is to find groups in the data, whereas the no. of groups is represented by the variable K. The data have been clustered on the basis of high similarity points together and low similarity points in the separate clusters.
- ➔ Document clustering can be split in to 3 steps. Pictorial representation and the explanation of the same is shown in the below table.



Cleaning and tokenizing data: this involves lowercasing text, removing non alpha numeric characters/stop words, and stemming the data.

Generating Vector representation of the documents: This concerns with mapping the documents from words into numerical vectors. We can use bag-of-words model (CountVectorizer) or TF.IDF (TfidfVectorizer) to construct vector. Here we used TF.IDF to construct vector.

Applying K-mean algorithm on the document vectors: This requires selecting and applying clustering algorithm to find the best possible groups using the document vectors.

- ➔ Performance Measured: Evaluating the performance of a clustering algorithm is not as trivial as counting the number of errors or the precision and recall of a supervised classification algorithm. In particular any evaluation metric should not take the absolute values of the cluster labels into account but rather if this clustering define separations of the data similar to some ground truth set of classes or satisfying some assumption such that members belong to the same class are more similar than members of different classes according to some similarity metric.
- ➔ Given the knowledge of the ground truth class assignments labels_truth and our clustering algorithm assignments of the same samples labels_pred the **(adjusted or unadjusted) Rand index** is a function that measures the **similarity** of the two assignments, ignoring permutations. Perfect labelling is scored 1.0.
- ➔ I got rand index = 0.7142857142857143 in this task.

Q3: Which type of clustering is used (hierarchical/flat and hard/soft)

- ➔ For this task I used Hard clustering as each document belongs to only one cluster andflat clustering as a cluster won't split itself into several documents.
- ➔ Two types of clustering's
  - ➢ Hard (document belongs to only one cluster) /
  - ➢ Soft (document belongs to more than one cluster).
- ➔ Two types of clustering's
- ➢ Flat (a cluster won't split itself into several clusters)
- ➢ Hierarchical (a cluster itself split into two or more clusters)

Q4: Screenshot and demonstration of its accuracy and robustness for numerous and various inputs

- ➔ Crawled BBC news website and pulled some text related to sports, health and politics. I stored all the Crawled text into a list. Each value in a list is a document. Total length of the list is 121, that means total 121 documents are collected.
- ➔ Below screenshot shows the code for crawling data from news websites.

```
import requests
from bs4 import BeautifulSoup

def craw_test(l):
    count = 0
    url = 'https://www.bbc.co.uk/sport'
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text,"lxml")
    for link in soup.findAll('p',{'class':"ssrcss-6arcww-PromoHeadline e1f5wbog4"}):
            title = link.string
            if(title != None and len(title) > 40 and count < 20):
                #print('news headlines:',title)
                l.append(title)
                count += 1
    url = 'https://www.bbc.co.uk/news/uk-politics-62239950'
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text,"lxml")
    for link in soup.findAll('p',{'class':"ssrcss-1q0x1qg-Paragraph eq5iqo00"}):
            title = link.string
            if(title != None and len(title) > 40 and count < 60):
                #print('news headlines:',title)
                l.append(title)
                count += 1
    count = 0
    url = 'https://www.bbc.co.uk/news/politics'
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text,"lxml")
    for link in soup.findAll('p',{'class':"gs-c-promo-summary gel-long-primer gs-u-mt nw-c-promo-summary gs-u-display-none gs-u-d
        title = link.string
        #print('politics headlines:',title)
        l.append(title)
    for link in soup.findAll('a',{'class':"gs-c-promo-heading gs-o-faux-block-link__overlay-link gel-pica-bold nw-o-link-split__a
        title = link.string
        count = 0
        if(title != None and len(title) > 40 and count < 40):
            #print('politics headlines:',title)
            l.append(title)
            count += 1
    count = 0
    url = 'https://www.bbc.co.uk/news/health-62250899'
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text,"lxml")
    for link in soup.findAll('p',{'class':"ssrcss-1q0x1qg-Paragraph eq5iqo00"}):
        title = link.string
        count = 0
        if(title != None and len(title) > 40 and count < 60):
            #print('health news:',title)
            l.append(title)
            count += 1
    return l


l = []
docs = craw_test(l)
print(len(docs))
print(docs)
```

- ➔ Added print statements in the code to show the length of the list of documents and also printed the docs list. Below screenshots shows the output of the above code.
- ➔ In the below screenshot, it shows the starting of the list.

```
117
['Commonwealth Games: 16 medal events on first day - watch', 'Rossouw hits 96 as South Africa level T20 series', "'Now or neve
r' - hurdler Allen chases NFL dream", 'Watch: Pick from 11 Commonwealth Games streams', 'A decade of development, six landmark
moments', 'Arsenal & Northern Ireland legend Neill dies at 80', "Hamilton & Alonso lead tributes to 'legend' Vettel", "Jamaic
a's ex-GB athlete Tracey ruled out of Games", "Man City abandon Cucurella plans - Friday's gossip", "Stenson does not feel he
'gave up' Ryder Cup role", 'Anger over pro-Putin chants at Turkey football match', 'Motherwell dumped out of Europe by inspired
Sligo', 'Crusaders bow out of Europe after Basel draw', 'Raging bulls & a Nobel laureate - Commonwealth Games open in style',
'Commonwealth Games day-by-day guide & schedule', "'My brothers hung me from doors, but it made me'", 'BBC TV, radio & digital
coverage times for Commonwealth Games', "'I want to leave legacy' - Fachie's quest for glory", 'England aim for gold as cricket
returns to Games', "Flagbearer Gilmour's letter to 16-year-old self", "Watch the key moments from Boris Johnson's final PMQs",
"If Boris Johnson was expecting a teary-eyed send off from all MPs, he didn't get one at his final Prime Minister's Question
s.", "The outgoing prime minister's last weekly grilling at the dispatch box was not unlike his first in September 2019 - acrim
onious, raucous, and littered with his colourful quips.", 'After almost an hour of political theatre, Mr Johnson brought the cu
rtain down in characteristic fashion, with a tongue-in-cheek farewell, borrowed from the script of a sci-fi blockbuster.', '"Ha
sta la vista, baby," Mr Johnson told MPs, reciting the line from Arnold Schwarzenegger\'s cyborg character in the 1991 film Ter
minator 2: Judgment Day.', "Cue uproarious laughter, cheers and a standing ovation from the Conservative benches - although for
mer PM Theresa May didn't appear to join in the applause.", 'Given Schwarzenegger\'s other famous catchphrase in the film - "I
\'ll be back" - Mr Johnson\'s sign-off and what it means may hang in the air in Westminster.', 'For now, at least, Mr Johnson w
ill leave office by September, when the contest to elect his replacement as Tory leader and prime minister is due to conclud
e.', 'In a pithy closing statement Mr Johnson had a few words of advice for his successor, saying that they should stay close t
o the Americans, stick up for the Ukrainians and cut taxes whenever possible.', '"Focus on the road ahead, but always remember
to check rear-view mirror. And remember above all, it\'s not Twitter that counts - it\'s the people who sent us here."', "Held
in the pressure cooker of the House of Commons, Prime Minister's Questions is designed to be an adversarial occasion of high po
litical drama.", 'The first was in 1961, and since then, prime ministers have admitted the event holds some terror.', 'Years af
ter leaving office, former Prime Minister Tony Blair likened PMQs to being marched to his own "execution", while one of his pre
decessors, Harold Macmillan, confided it made him feel "physically sick".', "It's easy to see why as Mr Johnson faced a hostile
audience of MPs, some of whom had been involved in ousting him as Tory leader weeks ago.", 'Before proceedings began, Commons S
peaker Lindsay Hoyle urged MPs to moderate their language and conduct themselves in a respectful manner.', "The plea appeared t
o fall on deaf ears though as Labour MPs - led by leader Sir Keir Starmer - launched a barrage of attacks on Mr Johnson's integ
```

➔ Next screenshot shows the ending of the list.

```
quiry, has said.", 'Opening the inquiry, she promised to be "fair and robust".', 'The former High Court judge said she would co
nduct the inquiry as quickly as possible, without giving a timeframe for its completion.', 'Those who had suffered the most des
erved to know if more could have been done, she said.', 'Lives had been lost, education harmed, businesses folded and mental an
d physical health had suffered.', '"Every person has had their life changed to some extent," Lady Hallett said.', '"Those who h
ave suffered the most will want to know if any more could have been done to reduce their suffering."', 'The inquiry can compel
witnesses to give evidence and release documents, but cannot prosecute or fine anyone.', 'It was a substantial task that would
take time and have a significant cost, Lady Hallett said.', 'But she added: "I am determined to undertake the inquiry as speedi
ly as possible so lessons can be learned before another pandemic strikes."', 'Public hearings will begin in the spring.', 'Befo
re then, Lady Hallettt said, the key topics for the inquiry would include:', 'The inquiry will begin taking evidence from exper
ts in September.', 'There will be many involved in the care sector who will have a great deal to say. They will remember the ea
rly warnings from other countries about the vulnerability of care homes. ', 'Care providers will recount their struggle to get
protective equipment; what they saw as the slowness of government guidance; the rapid discharge of hospital patients into care
homes, some taking with them the virus and above all, their sense of being forgotten. ', 'Families who were unable to see or in
too many cases, say goodbye to their loved ones, are likely to have their say later in the inquiry. ', 'But in weighing up our
preparedness for the pandemic, there is one number that perhaps tells the most powerful story. In the first wave between March
and June 2020, nearly 20,000 care home residents died of Covid in England and Wales. At that time, it represented more than a t
hird of the total number of deaths.', 'Jo Goodman, co-founder of Covid-19 Bereaved Families for Justice campaign, said: "Today
was an emotional day for those of us who have lost loved ones and it meant a lot to hear Baroness Hallet recognise the devastat
ing nature of bereavement and the pain we\'ve been through. ', '"Ultimately, all bereaved families want the same thing, which i
s to make sure that lessons are learnt from our devastating losses to protect others in the future."', 'Charles Persinger, who
is part of the campaign and lost his mother and wife to Covid, added: "We\'ve waited a long time to get to this point - we woul
d have liked the inquiry to start sooner. But what is important now is to really get to the bottom of the mistakes that were ma
de."', "Several reports have already put the UK government's handling of the pandemic under the spotlight.", 'The Covid pandemi
c had a devastating impact on ethnic minority communities in the UK. ', 'In the first wave of the pandemic, black people were a
lmost four times as likely to die of Covid than white people, while Asians were twice as likely to die. About 95% of doctors wh
o died of Covid were ethnic minorities.  ', 'Saleyha Ahsan, a doctor and documentary maker, lost her dad to Covid-19. She tells
me it\'s good inequalities are being considered, but she\'s "keen to see how [Baroness Hallett] does it in real terms".', '"I
\'m not convinced it\'s going to cover it all, so I am wary." ', 'Dr Ahsan is involved with the Covid-19 Bereaved Families for
Justice, which led the campaign for an inquiry to happen.', '"The main thing is is that we have now crossed the start line," sh
e says. "We\'ve been calling for this for a whileâ\x80¦ and today is a definite firing of the starting gun." ', 'From intricate
mosaics to amazing hieroglyphics - what else hides under the water?', 'Matthew Syed is calling for a nuclear awakening...']
```

➔ Downloaded stop words

```
import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords

sw = stopwords.words('english')
print(sw)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'y
ourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself',
'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those',
'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'a
n', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'b
etween', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'of
f', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both',
'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very',
's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'ar
en', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "have
n't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "should
n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

```
[nltk_data] Downloading package stopwords to C:\Users\Bharathi
[nltk_data]     Kondaveeti\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

➔ Implemented tokenizer and stemming to the input list of docs.

```
nltk.download("punkt")
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer

ps = PorterStemmer()
filtered_docs = []
for doc in docs:
    tokens = word_tokenize(doc)
    tmp = ""
    for w in tokens:
        if w not in sw:
            tmp += ps.stem(w) + " "
    filtered_docs.append(tmp)

print(filtered_docs)
```

```
['commonwealth game : 16 medal event first day - watch ', 'rossouw hit 96 south africa level t20 seri ', "'now never ' - hurdle
r allen chase nfl dream ", 'watch : pick 11 commonwealth game stream ', 'a decad develop , six landmark moment ', 'arsen & nort
hern ireland legend neill die 80 ', "hamilton & alonso lead tribut 'legend ' vettel ", "jamaica 's ex-gb athlet tracey rule gam
e ", "man citi abandon cucurella plan - friday 's gossip ", "stenson feel 'gave ' ryder cup role ", 'anger pro-putin chant turk
ey footbal match ', 'motherwel dump europ inspir sligo ', 'crusad bow europ basel draw ', 'rage bull & nobel laureat - commonwe
alth game open style ', 'commonwealth game day-by-day guid & schedul ', "'mi brother hung door , made ' ", 'bbc tv , radio & di
git coverag time commonwealth game ', "' i want leav legaci ' - fachi 's quest glori ", 'england aim gold cricket return game
', "flagbear gilmour 's letter 16-year-old self ", "watch key moment bori johnson 's final pmq ", "if bori johnson expect teary
-ey send mp , n't get one final prime minist 's question . ", "the outgo prime minist 's last weekli grill dispatch box unlik f
irst septemb 2019 - acrimoni , raucou , litter colour quip . ", 'after almost hour polit theatr , mr johnson brought curtain ch
aracterist fashion , tongue-in-cheek farewel , borrow script sci-fi blockbust . ', "`` hasta la vista , babi , '' mr johnson to
ld mp , recit line arnold schwarzenegg 's cyborg charact 1991 film termin 2 : judgment day . ", "cue uproari laughter , cheer s
tand ovat conserv bench - although former pm theresa may n't appear join applaus . ", "given schwarzenegg 's famou catchphras f
ilm - `` i 'll back '' - mr johnson 's sign-off mean may hang air westminst . ", 'for . least . mr johnson leav offic septemb .
```

```
lici hoof - starmer ', 'ex-ambassador sir christoph meyer die 78 ', "dorri defend clair 's accessori sunak attack ", 'sunak say
ye return grammar school ', 'chri mason : truss court johnson loyalist sunak face jibe ', 'mp fire make polici hoof - starmer
', 'ex-ambassador sir christoph meyer die 78 ', "dorri defend clair 's accessori sunak attack ", 'bryant make chariti donat cla
im disprov ', "have wage 'stuck decad ' ? and claim ", 'cut leav militari risk threat rise - mp ', 'govern rule bank holiday eu
ro win ', 'truss vow criminalis street harass ', 'how uk green learn oversea alli ', 'five takeaway heat truss-sunak clash ',
'who tori choos next pm ? ', 'what tax would two tori leadership rival cut ? ', "lesson learn uk 's handl covid , anoth pandem
strike , baro hallett , chair public inquiri , said . ", "open inquiri , promis `` fair robust '' . ", 'the former high court j
udg said would conduct inquiri quickli possibl , without give timefram complet . ', 'those suffer deserv know could done , said
. ', 'live lost , educ harm , busi fold mental physic health suffer . ', "`` everi person life chang extent , '' ladi hallett s
aid . ", "`` those suffer want know could done reduc suffer . '' ", 'the inquiri compel wit give evid releas document , prosecu
t fine anyon . ', 'it substanti task would take time signific cost , ladi hallett said . ', "but ad : `` i determin undertak in
quiri speedili possibl lesson learn anoth pandem strike . '' ", 'public hear begin spring . ', 'befor , ladi hallettt said , ke
y topic inquiri would includ : ', 'the inquiri begin take evid expert septemb . ', 'there mani involv care sector great deal sa
y . they rememb earli warn countri vulner care home . ', 'care provid recount struggl get protect equip ; saw slow govern guida
nc ; rapid discharg hospit patient care home , take viru , sens forgotten . ', 'famili unabl see mani case , say goodby love on
e , like say later inquiri . ', 'but weigh prepared pandem , one number perhap tell power stori . in first wave march june 2020
, nearli 20,000 care home resid die covid england wale . at time , repres third total number death . ', "jo goodman , co-found
covid-19 bereav famili justic campaign , said : `` today emot day us lost love one meant lot hear baro hallet recognis devast n
atur bereav pain 've . ", "`` ultim , bereav famili want thing , make sure lesson learnt devast loss protect other futur . ''
", "charl persing , part campaign lost mother wife covid , ad : `` we 've wait long time get point - would like inquiri start s
ooner . but import realli get bottom mistak made . '' ", "sever report alreadi put uk govern 's handl pandem spotlight . ", 'th
e covid pandem devast impact ethnic minor commun uk . ', 'in first wave pandem , black peopl almost four time like die covid wh
ite peopl , asian twice like die . about 95 % doctor die covid ethnic minor . ', "saleyha ahsan , doctor documentari maker , lo
st dad covid-19 . she tell 's good inequ consid , 's `` keen see [ baro hallett ] real term '' . ", "`` i 'm convinc 's go cove
r , i wari . '' ", 'dr ahsan involv covid-19 bereav famili justic , led campaign inquiri happen . ', "`` the main thing cross s
tart line , '' say . `` we 've call whileâ\x80¦ today definit fire start gun . '' ", 'from intric mosaic amaz hieroglyph - els
hide water ? ', 'matthew sy call nuclear awaken ... ']
```

➔ Implemented TFIDF vectorizer to the above output docs.

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(filtered_docs)
print(X.todense())
print(X)
```

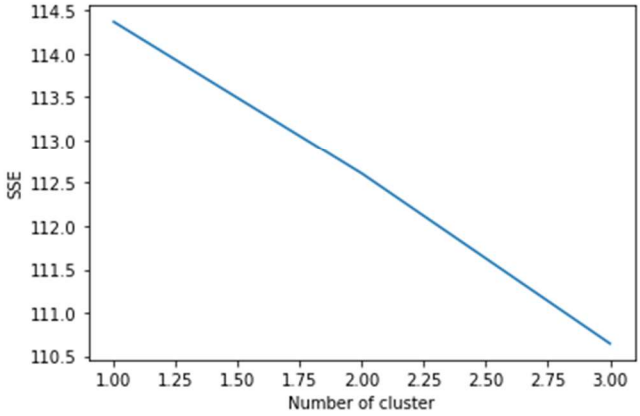
```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
  (0, 769)      0.3435976882426375
  (0, 195)      0.328543241879226
  (0, 283)      0.3158148888410524
  (0, 253)      0.38577699544728233
  (0, 473)      0.419256595060111
  (0, 3)        0.38577699544728233
  (0, 302)      0.3047890879239983
  (0, 150)      0.328543241879226
  (1, 635)      0.35355339059327373
  (1, 689)      0.35355339059327373
  (1, 437)      0.35355339059327373
  (1, 29)       0.35355339059327373
  (1, 656)      0.35355339059327373
  (1, 17)       0.35355339059327373
  (1, 356)      0.35355339059327373
  (1, 610)      0.35355339059327373
  (2, 229)      0.3779644730092272
  (2, 498)      0.3779644730092272
  (2, 127)      0.3779644730092272
```

```
(2, 36)       0.3779644730092272
(2, 371)      0.3779644730092272
(2, 496)      0.3779644730092272
(2, 504)      0.3779644730092272
(3, 675)      0.4557608899406954
(3, 1)        0.4557608899406954
:     :
(114, 185)    0.2653584033848646
(114, 460)    0.2653584033848646
(114, 772)    0.2441682940248652
(114, 710)    0.2441682940248652
(114, 753)    0.22913367216897348
(114, 718)    0.2441682940248652
(114, 282)    0.22913367216897348
(114, 668)    0.45826734433794697
(114, 109)    0.20794356280897405
(114, 619)    0.19988745712161005
(114, 444)    0.2174719134608349
(114, 703)    0.15351589182151953
(115, 770)    0.35355339059327373
(115, 350)    0.35355339059327373
(115, 243)    0.35355339059327373
(115, 351)    0.35355339059327373
(115, 44)     0.35355339059327373
(115, 487)    0.35355339059327373
(115, 388)    0.35355339059327373
(115, 298)    0.35355339059327373
(116, 68)     0.4655404465175178
(116, 505)    0.4655404465175178
(116, 688)    0.4655404465175178
(116, 469)    0.4655404465175178
(116, 109)    0.36481278846153503
```

➔ Elbow method gives us an idea on what a good k number of clusters would be based on the sum of squared distance (SSE) between data points and their assigned clusters' centroids. We pick k at the

spot where SSE starts to flatten out and forming an elbow.

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
sse = {}
for k in range(1,4):
    kmeans = KMeans(n_clusters=k, max_iter=5).fit(X)
    #data["clusters"] = kmeans.labels_
    #print(data["clusters"])
    sse[k] = kmeans.inertia_ # Inertia: Sum of distances of samples to their closest cluster center
plt.figure()
plt.plot(list(sse.keys()), list(sse.values()))
plt.xlabel("Number of cluster")
plt.ylabel("SSE")
plt.show()
```



➔ Implemented K-means to the TFIDF vectorizer output for k = 1 means all the input data is into one cluster '0'.

```
from sklearn.cluster import KMeans
K = 1
model = KMeans(n_clusters=K)#, init='k-means++', max_iter=100, n_init=1)
model.fit(X)

print("cluster no. of input documents, in the order they received:")
print(model.labels_)
```

```
cluster no. of input documents, in the order they received:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0]
```

➔ Implemented K-means to the TFIDF vectorizer output for k = 2.it means all the input data is into two cluster '0 and 1'.

```
from sklearn.cluster import KMeans
K = 2
model = KMeans(n_clusters=K)#, init='k-means++', max_iter=100, n_init=1)
model.fit(X)

print("cluster no. of input documents, in the order they received:")
print(model.labels_)
```

```
cluster no. of input documents, in the order they received:
[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 0
 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
 0 0 1 0 0 0 1 0 0 0]
```

➔ Implemented K-means to the TFIDF vectorizer output for k = 2.it means all the input data is into two cluster ‘0, 1 and 2’.

```
from sklearn.cluster import KMeans
K = 3
model = KMeans(n_clusters=K)#, init='k-means++', max_iter=100, n_init=1)
model.fit(X)

print("cluster no. of input documents, in the order they received:")
print(model.labels_)
```

```
cluster no. of input documents, in the order they received:
[0 0 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 2 1 0 1 1 1 1 1 0 1 0
 1 1 1 1 1 1 1 1 0 0 1 1 0 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 2 1 2 2 0 0 2
 2 2 0 0 2 0 0 0 0 1 0 2 1 0 0 0 1 1 0 0 1 0 0 0 1 0 1 0 0 2 0 0 0 0 0 0 0
 1 0 0 0 0 0]
```

➔ Below code predicting the incoming text documents by using above K-means clustering Output. The list test_doc is the input documents for the predicting to which cluster the input documents belong to. For the test_doc list, I applied all the pre-processing tasks (removing stop words, tokenizing and stemming) which were applied to the clustering input list. Then applied vectorization and model prediction on the processed list. Then we will get the prediction of each document i.e.; to which cluster the document belongs to.

```
test_doc = ["The Covid pandemic had a devastating impact on ethnic minority communities",
             "Boris Johnson was expecting a teary-eyed send off from all MPs",
             "care home residents died of Covid in England and Wales",
            "England on top in first T20 v South Africa",
            "contest to elect his replacement as Tory leader and prime minister is due to conclude",
            "Vingegaard close to Tour victory after stage 18 win",
            "Birmingham attempts to leave 'carbon neutral legacy",
            "Coe says global warming could force move of events",
            "16 medal event first day - watch ",
            "France wins world cup in football",
            "Sunak says yes to return of grammar schools"
           ]
filtered_test_docs = []
for doc in test_doc:
    tokens = word_tokenize(doc)
    tmp = ""
    for w in tokens:
        if w not in sw:
            tmp += ps.stem(w) + " "
    filtered_test_docs.append(tmp)

print(filtered_test_docs)
prediction = []
labels_pred = []
for i in range(len(filtered_test_docs)):
    Y = vectorizer.transform([filtered_test_docs[i]])
    prediction=(model.predict(Y))
    print(prediction)
    labels_pred.append(prediction[0])
```

```
labels_pred = []
for i in range(len(filtered_test_docs)):
    Y = vectorizer.transform([filtered_test_docs[i]])
    prediction=(model.predict(Y))
    print(prediction)
    labels_pred.append(prediction[0])
```

```
['the covid pandem devast impact ethnic minor commun ', 'bori johnson expect teary-ey send mp ', 'care home resid die covid eng
land wale ', 'england top first t20 v south africa ', 'contest elect replac tori leader prime minist due conclud ', 'vingegaard
close tour victori stage 18 win ', "birmingham attempt leav 'carbon neutral legaci ", 'coe say global warm could forc move even
t ', '16 medal event first day - watch ', 'franc win world cup footbal ', 'sunak say ye return grammar school ']
[0]
[1]
[0]
[0]
[1]
[0]
[0]
[0]
[0]
[0]
[2]
```

➔ To measure the performance of K-mean cluster there are many ways, here I calculated rand index to measure the performance of document clustering, I imported metrics from sklearn and applied rand index to get rand score. Input to the rand score are predicted output from above prediction and labels_true which I manually gave input in a list of integer values to which cluster it belongs to. I got 0.714 as rand index. The value of rand index ranges from 0 to 1. Perfect rand index is 1.

```
#rand index - performance measure of k-mean cluster
from sklearn import metrics
labels_true = [0,1,0,2,1,2,0,0,2,2,2]
metrics.rand_score(labels_true, labels_pred)
```

```
0.6363636363636364
```

5. Brief explanation of how k-means document clustering works.

- ➔ The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.
- ➔ The first step in k-means document clustering is to collect the input documents.
- ➔ The second step is to perform pre-processing tasks like removing stop words, tokenizing and stemming to the input documents.
- ➔ The third step is to perform TF.IDF which means converting text document into numerical form called vectorization.
- ➔ The fourth step is to implement K-mean clustering for the vector form of our input document where k is equal to the number of clusters you choose. Given K = 1, 2 and 3 as the input document contains data from three categories.
- ➔ The fifth step is predicting of some incoming documents to the cluster they belongs to.
- ➔ The sixth step is to measure the performance of k-mean i.e.; finding rand index.it ranges from 0 to 1 (perfect).

References:

https://realpython.com/k-means-clustering-python/#understanding-the-k-means-algorithm

https://www.ijert.org/research/improved-clustering-of-documents-using-k-means-algorithm-IJERTV5IS070358.pdf

https://scikit-learn.org/stable/auto_examples/text/plot_document_clustering.html

https://research.ijcaonline.org/volume110/number11/pxc3900929.pdf

https://towardsdatascience.com/performance-metrics-in-machine-learning-part-3-clustering-d69550662dc6

K-means Clustering Elbow Method & SSE Plot - Python - Data Analytics (vitalflux.com)

Appendix:

Task 1:

Source Code: crawled author details with all his publication details.

```
import requests
from bs4 import BeautifulSoup
def craw_test(url,max_pages):
    page = 0
    total_publications = 0
    total_activities = 0
    count_authors = 0
    Q = {}
    while page <= max_pages:
        if (page >0):
            new_url = url+'?page='+str(page)
        else:
```

```
            new_url = url
        source_code = requests.get(new_url)
        plain_text = source_code.text
        soup = BeautifulSoup(plain_text,"lxml")
        for link in soup.findAll('a',{'class':'link person'}):
            links = []
            href = link.get('href')
            title = link.string
            #print('author profile link:',href)
            #print('author name:',title)
            links.append(href)
            count_authors += 1
            links,no_of_publications,total_publications                    =
get_single_author_publications(links,href+'/publications/',total_publications)
            Q[title] = links
            print('total no of publications by',title,'are',no_of_publications)


        page += 1
    print('total no of authors in School of Economics, Finance and Accounting -
Coventry University:' ,count_authors)
    print('total research publications by School of Economics, Finance and
Accounting -Coventry University:',total_publications)
    print('Dictionary',Q)
def get_single_author_publications(q,item_url,total):
    source_code = requests.get(item_url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text,"lxml")
    count = 0
    list_of_results = soup.findAll('div',{'class':'result-container'})
    for each in list_of_results:
        h3_tag = each.find('h3')
        link = h3_tag.find('a').get('href')
        title = h3_tag.text.strip()
        date = each.find('span',{'class':'date'})
        #print('title of publication:',title)
        #print('publications link:',link)
        #print("date of publication:",date.text.strip())
```

```
            q.append(link)
            count +=1
            total +=1
        return q,count,total

seed = 'https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/persons/'
craw_test(seed,2)
```

Screenshots of code:

```
import requests
from bs4 import BeautifulSoup

def craw_test(url,max_pages):
    page = 0
    total_publications = 0
    total_activities = 0
    count_authors = 0
    Q = {}
    while page <= max_pages:
        if (page >0):
            new_url = url+'?page='+str(page)
        else:
            new_url = url
        source_code = requests.get(new_url)
        plain_text = source_code.text
        soup = BeautifulSoup(plain_text,"lxml")
        for link in soup.findAll('a',{'class':'link person'}):
            links = []
            href = link.get('href')
            title = link.string
            #print('author profile link:',href)
            #print('author name:',title)
            links.append(href)
            count_authors += 1
            links,no_of_publications,total_publications = get_single_author_publications(links,href+'/publications/',total_publi
            Q[title] = links
            print('total no of publications by',title,'are',no_of_publications)

        page += 1
```

```
        page += 1
    print('total no of authors in School of Economics, Finance and Accounting -Coventry University:' ,count_authors)
    print('total research publications by School of Economics, Finance and Accounting -Coventry University:',total_publications)
    print('Dictionary',Q)
def get_single_author_publications(q,item_url,total):
    source_code = requests.get(item_url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text,"lxml")
    count = 0
    list_of_results = soup.findAll('div',{'class':'result-container'})
    for each in list_of_results:
        h3_tag = each.find('h3')
        link = h3_tag.find('a').get('href')
        title = h3_tag.text.strip()
        date = each.find('span',{'class':'date'})
        #print('title of publication:',title)
        #print('publications link:',link)
        #print("date of publication:",date.text.strip())
        q.append(link)
        count +=1
        total +=1
    return q,count,total

seed = 'https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/persons/'
craw_test(seed,2)
```

Output:

```
total no of publications by Mohamad Nazri Abd Karim are 3
total no of publications by Ahmad Abras are 8
total no of publications by Mohammed Aderemi Adepoju are 0
total no of publications by Frank Adom are 0
total no of publications by Dami Agbato are 0
total no of publications by Daniel Aghanya are 5
total no of publications by Abel Agoba are 0
total no of publications by Hany Ahmed are 2
total no of publications by Olubunmi Ajala are 2
total no of publications by Sule Akkoyunlu are 0
total no of publications by George Akomas are 0
total no of publications by Samir Alamad are 7
total no of publications by Salem Alhababsah are 6
total no of publications by Alaa Alhaj Ismail are 5
total no of publications by Tariq Al Montaser are 4
total no of publications by Loai Alsaid are 7
total no of publications by Marwan Alssadek are 0
total no of publications by Adnan Aslam are 0
total no of publications by Haseeb Ayaz are 0
total no of publications by Peng Sui are 0
total no of publications by Angelos Synapis are 1
total no of publications by Sailesh Tanna are 46
total no of publications by McFoster Tembo are 0
total no of publications by Uchenna Tony-Okeke are 5
total no of publications by Ejike Udeogu, SFHEA are 14
total no of publications by Obinna Ugwu are 0
total no of publications by Hafij Ullah are 12
total no of publications by Valeria Unali are 0
total no of publications by Ahmed Usman are 1
total no of publications by Jun Wang are 2
total no of publications by Torri Wang are 3
total no of publications by Sandar Win are 7
total no of publications by Alain Wouassom are 0
total no of publications by Di Xiao are 0
total no of publications by Suiwu Xiao are 0
total no of publications by Boying Xu are 2
total no of publications by Mei Yu are 7
total no of publications by Alireza Zarei are 18
total no of authors in School of Economics, Finance and Accounting -Coventry University: 101
```


```
Dictionary {'Mohamad Nazri Abd Karim': ['https://pureportal.coventry.ac.uk/en/persons/mohamad-nazri-abd-karim', 'https://pure
portal.coventry.ac.uk/en/publications/stock-liquidity-and-smes-likelihood-of-bankruptcy-evidence-from-t', 'https://pureporta
l.coventry.ac.uk/en/publications/personalisation-of-power-neoliberalism-and-the-production-of-corr', 'https://pureportal.cove
ntry.ac.uk/en/publications/stock-price-and-volume-effects-associated-with-changes-in-the-com'], 'Ahmad Abras': ['https://pure
portal.coventry.ac.uk/en/persons/ahmad-abras', 'https://pureportal.coventry.ac.uk/en/publications/competing-institutional-log
ics-and-power-dynamics-in-islamic-fina', 'https://pureportal.coventry.ac.uk/en/publications/the-rise-and-fall-of-institutiona
l-entrepreneurship-in-islamic-fi', 'https://pureportal.coventry.ac.uk/en/publications/institutional-entrepreneurship-in-islam
ic-financial-reporting-sta', 'https://pureportal.coventry.ac.uk/en/publications/competing-institutional-logics-and-institutio
nal-embeddedness-of-', 'https://pureportal.coventry.ac.uk/en/publications/political-ideology-and-ngo-accountability-the-case-
of-the-syrian-', 'https://pureportal.coventry.ac.uk/en/publications/competing-institutional-logics-in-islamic-financial-repor
ting-sta', 'https://pureportal.coventry.ac.uk/en/publications/understanding-islamic-financial-reporting-standardisation-proje
ct', 'https://pureportal.coventry.ac.uk/en/publications/islamic-financial-reporting-standardisation-projects-institutiona'],
'Mohammed Aderemi Adepoju': ['https://pureportal.coventry.ac.uk/en/persons/mohammed-aderemi-adepoju'], 'Frank Adom': ['http
s://pureportal.coventry.ac.uk/en/persons/frank-adom'], 'Dami Agbato': ['https://pureportal.coventry.ac.uk/en/persons/dami-agb
ato'], 'Daniel Aghanya': ['https://pureportal.coventry.ac.uk/en/persons/daniel-aghanya', 'https://pureportal.coventry.ac.uk/e
n/publications/corporate-political-strategies-in-weak-institutional-environments', 'https://pureportal.coventry.ac.uk/en/publ
ications/market-in-financial-instruments-directive-mifid-stock-price-infor', 'https://pureportal.coventry.ac.uk/en/publicatio
ns/the-effect-of-government-involvement-and-payment-method-on-merger', 'https://pureportal.coventry.ac.uk/en/publications/the
-impact-of-regulations-on-compliance-costs-risk-taking-and-rep', 'https://pureportal.coventry.ac.uk/en/publications/evaluatio
n-of-mergen-nremium-and-firm-nerformance-in-eurone'], 'Abel Agoba': ['https://pureportal.coventry.ac.uk/en/persons/abel-agob
```

Stored the data in a dictionary having key as name of the author/staff and values are a list Containing author's link, and all his publications.

Source Code: Information collected about each publication

```
# Import libraries
from urllib.request import urljoin
from bs4 import BeautifulSoup
import requests
from urllib.request import urlparse
import re
import pandas as pd
# # Set for storing urls with same domain


# # Method for crawling a url at next level
def level_crawler(input_url,temp_urls,count_publications,df,j):
    data = []
    current_url_domain = urlparse(input_url).netloc
#    # Creates beautiful soup object to extract html tags
    beautiful_soup_object    =    BeautifulSoup(requests.get(input_url).content,
"lxml")
#    # Access all anchor tags from input
#    # url page and divide them into internal
#    # and external categories
    for anchor in beautiful_soup_object.findAll("a"):
        href = anchor.attrs.get("href")
        title = anchor.string
        if(href != "" or href != None):
            href = urljoin(input_url, href)
            href_parsed = urlparse(href)
```

```
            href = href_parsed.scheme
            href += "://"
            href += href_parsed.netloc
            href += href_parsed.path
            final_parsed_href = urlparse(href)
            is_valid = bool(final_parsed_href.scheme) and bool(
                final_parsed_href.netloc)
            if is_valid:
                if current_url_domain not in href and href not in links_extern:
                    #print("Extern - {}".format(href))
                    links_extern.add(href)
                if current_url_domain in href and href not in links_intern:
                     #if
re.search("^https://pureportal.coventry.ac.uk/en/publications/.*$", href):
                        #re.search("^The.*Spain$", href)
                    if '/publications/' in href[36:50] and href[50:] != '':
                        #print("Intern - {}".format(href))
                        count_publications += 1
                        links_intern.add(href)
                        temp_urls.append(href)
                        data.append(href)
                        #print('publications title:',title)
                        df.loc[j, ['publication_link']]  = href
                        df.loc[j, ['publication_title']] = title
                        df,j = get_author_details(href,df,j)
                        j += 1
    return temp_urls,count_publications,df,j,data
def get_author_details(link,df,j):
    source_code = requests.get(link)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text,"lxml")
    for link in soup.findAll('a',{'class':'link person'}):
        href = link.get('href')
        name = link.string
        source_code1 = requests.get(href)
        plain_text1 = source_code1.text
```

```
            soup1 = BeautifulSoup(plain_text1,"lxml")
            if(soup1.find('a',{'class':'link primary'}) != None):
                sefa = soup1.find('a',{'class':'link primary'}).string
            elif(soup1.find('a',{'class':'link school'}) != None):
                sefa = soup1.find('a',{'class':'link school'}).string
            if (sefa == 'School of Economics, Finance and Accounting'):
                df.loc[j, ['author_link']]  = href
                df.loc[j, ['author_name']] = name
        publi = soup.find('tr',{'class':'status'})
        date = publi.text.strip('date')
        #print('date:',date)
        #print(abstract,'abs.........')
        if (soup.find('div',{'class':'textblock'})):
            abstract = soup.find('div',{'class':'textblock'}).string
        else:
            abstract = 'Null'
        #print(abstract,'abs.........')
        if type(abstract) == 'NoneType':
            abstract = ' '
        df.loc[j, ['publication_date']]  = date
        df.loc[j, ['abstract']] = abstract
        #df.loc[j, ['content']] = abstract + name
        return df,j
links_intern = set()
url  =  'https://pureportal.coventry.ac.uk/en/organisations/school-of-economics-finance-and-accounting/publications/'
depth = 1
p = 0
max_pages = 12
count_publications = 0
# # Set for storing urls with different domain
df                  =                  pd.DataFrame(columns                  =["publication_link","publication_title","publication_date","author_link","author_name","abstract","content"])
j = 0
links_extern = set()
list_url = []
```

```
if(depth == 0):
    print("Intern - {}".format(input_url))
elif(depth == 1):
    while p >= 0:
        if p == 0:
            list = []
            print(url,'url page0')
            list_url,count_publications,df,j,data                                    =
level_crawler(url,list,count_publications,df,j)
            p += 1
        else:
            new_url = url+'?page='+str(p)
            print(new_url,'url page')
            list_url,count_publications,df,j,data                                    =
level_crawler(new_url,list_url,count_publications,df,j)
            p += 1
            if (data == []):
                p = -1
else:
#    # We have used a BFS approach
#    # considering the structure as
#    # a tree. It uses a queue based
#    # approach to traverse
#    # links upto a particular depth.
    queue = []
    queue.append(input_url)
    for j in range(depth):
        for count in range(len(queue)):
            url = queue.pop(0)
            urls = level_crawler(url)
            for i in urls:
                queue.append(i)
print('end')
print('total no of publications',count_publications)
#print(list_url)
df['content']=df['abstract']+df['author_name'] +df[`publication_Title']
df
```

```
import csv
df.to_csv(path_or_buf ="crawlerdata.csv",sep=',')
```

Source code for pre-processing tasks:

Reading a csv file and stored in a data frame which contains all the publications data which is being stored at the end of crawler program.

```
df = pd.read_csv("crawlerdata.csv")
print(df)
content = df['content'].astype(str)
print(content)
import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
sw = stopwords.words('english')
nltk.download("punkt")
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
ps = PorterStemmer()
filtered_docs = []
for doc in content:
    tokens = word_tokenize(doc)
    tmp = ""
    for w in tokens:
        if w not in sw:
            tmp += ps.stem(w) + " "
    filtered_docs.append(tmp)
print(filtered_docs)
```

Inverted index Source code:

```
inverted_index = {}
for i, doc in enumerate(df['fil_content']):
    for term in doc.split(" "):
        if term in inverted_index:
            inverted_index[term].add(i)
        else: inverted_index[term] = {i}
inverted_index
```

Query Processor Source code:

```
query = input('enter input query:')

import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
```

```
sw = stopwords.words('english')
nltk.download("punkt")
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer

ps = PorterStemmer()
filtered_docs = []
tokens = word_tokenize(query)
tmp = ""
for w in tokens:
    if w not in sw:
        tmp += ps.stem(w) + " "
filtered_docs.append(tmp)

#print(filtered_docs)

x = filtered_docs[0].split(" ")

q = []
for i in range(len(x)):
    if len(x[i]) > 0:
        q.append(x[i])
#print(q)

p = []
for i in range(len(q)):
    if (q[i] in inverted_index):
        a = inverted_index[q[i]]
        l = sorted(a)
        p.append(l)
        print('l = ',q[i],l)
    else:
        print(q[i],'not found')
for i in p:
    print(i)
for i  in range(len(p)):
    print(p[i])
    for j in range(len(p[i])):
        #print(p[i][j])
        print(df.loc[p[i][j]])
```

Task 2:

Source Code for crawling input docs related to sports, health and politics from BBC news website:

```
import requests

from bs4 import BeautifulSoup

def craw_test(l):

   count = 0

   url = 'https://www.bbc.co.uk/sport'

   source_code = requests.get(url)

   plain_text = source_code.text
```

```
    soup = BeautifulSoup(plain_text,"lxml")
    for link in soup.findAll('p',{'class':"ssrcss-6arcww-PromoHeadline e1f5wbog4"}):
        title = link.string
        if(title != None and len(title) > 40 and count < 20):
            #print('news headlines:',title)
            l.append(title)
            count += 1
    url = 'https://www.bbc.co.uk/news/uk-politics-62239950'
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text,"lxml")
    for link in soup.findAll('p',{'class':"ssrcss-1q0x1qg-Paragraph eq5iqo00"}):
        title = link.string
        if(title != None and len(title) > 40 and count < 60):
            #print('news headlines:',title)
            l.append(title)
            count += 1
    count = 0
    url = 'https://www.bbc.co.uk/news/politics'
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text,"lxml")
    for link in soup.findAll('p',{'class':"gs-c-promo-summary gel-long-primer gs-u-mt nw-c-promo-summary gs-u-display-none gs-u-display-block@m"}):
        title = link.string
        #print('politics headlines:',title)
        l.append(title)
    for link in soup.findAll('a',{'class':"gs-c-promo-heading gs-o-faux-block-link__overlay-link gel-pica-bold nw-o-link-split__anchor"}):
        title = link.string
        count = 0
        if(title != None and len(title) > 40 and count < 40):
            #print('politics headlines:',title)
```

```
            l.append(title)
            count += 1
    count = 0
    url = 'https://www.bbc.co.uk/news/health-62250899'
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text,"lxml")
    for link in soup.findAll('p',{'class':"ssrcss-1q0x1qg-Paragraph eq5iqo00"}):
        title = link.string
        count = 0
        if(title != None and len(title) > 40 and count < 60):
            #print('health news:',title)
            l.append(title)
            count += 1
    return l
l = []
docs = craw_test(l)
print(len(docs))
print(docs)
```

Source Code:

```
import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
sw = stopwords.words('english')
print(sw)
```

Source Code for pre-processing task:

```
nltk.download("punkt")
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
ps = PorterStemmer()
filtered_docs = []
for doc in docs:
```

```
    tokens = word_tokenize(doc)
    tmp = ""
    for w in tokens:
        if w not in sw:
            tmp += ps.stem(w) + " "
    filtered_docs.append(tmp)
print(filtered_docs)
```

Tfidf vectorization Code:

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(filtered_docs)
print(X.todense())
print(X)
```

Elbow graph Code:

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
sse = {}
for k in range(1,4):
    kmeans = KMeans(n_clusters=k, max_iter=5).fit(X)
    #data["clusters"] = kmeans.labels_
    #print(data["clusters"])
    sse[k] = kmeans.inertia_ # Inertia: Sum of distances of samples to their
closest cluster center
plt.figure()
plt.plot(list(sse.keys()), list(sse.values()))
plt.xlabel("Number of cluster")
plt.ylabel("SSE")
plt.show()
```

k-means Code:

```
from sklearn.cluster import KMeans
K = 3
model = KMeans(n_clusters=K)#, init='k-means++', max_iter=100, n_init=1)
model.fit(X)
print("cluster no. of input documents, in the order they received:")
print(model.labels_)
```

k-means test Code:

```
test_doc = ["The Covid pandemic had a devastating impact on ethnic minority communities",
        "Boris Johnson was expecting a teary-eyed send off from all MPs",
```

```
        "care home residents died of Covid in England and Wales",
        "England on top in first T20 v South Africa",
        "contest to elect his replacement as Tory leader and prime minister is due to conclude",
        "Vingegaard close to Tour victory after stage 18 win",
        "Birmingham attempts to leave 'carbon neutral legacy",
        "Coe says global warming could force move of events",
        "16 medal event first day - watch ",
        "France wins world cup in football",
        "Sunak says yes to return of grammar schools"
       ]
filtered_test_docs = []
for doc in test_doc:
   tokens = word_tokenize(doc)
   tmp = ""
   for w in tokens:
      if w not in sw:
         tmp += ps.stem(w) + " "
   filtered_test_docs.append(tmp)


print(filtered_test_docs)
prediction = []
labels_pred = []
for i in range(len(filtered_test_docs)):
   Y = vectorizer.transform([filtered_test_docs[i]])
   prediction=(model.predict(Y))
   print(prediction)
   labels_pred.append(prediction[0])
```

rand index performance measure Code:

```
#rand index - performance measure of k-mean cluster
from sklearn import metrics
labels_true = [0,1,0,2,1,2,0,0,2,2,2]
metrics.rand_score(labels_true, labels_pred)
```

input docs list for clustering:

['Commonwealth Games: 16 medal events on first day - watch', 'Rossouw hits 9
6 as South Africa level T20 series', "'Now or never' - hurdler Allen chases
NFL dream", 'Watch: Pick from 11 Commonwealth Games streams', 'A decade of d
evelopment, six landmark moments', 'Arsenal & Northern Ireland legend Neill
dies at 80', "Hamilton & Alonso lead tributes to 'legend' Vettel", "Jamaica'
s ex-GB athlete Tracey ruled out of Games", "Man City abandon Cucurella plan
s - Friday's gossip", "Stenson does not feel he 'gave up' Ryder Cup role", '
Anger over pro-Putin chants at Turkey football match', 'Motherwell dumped ou
t of Europe by inspired Sligo', 'Crusaders bow out of Europe after Basel dra
w', 'Raging bulls & a Nobel laureate - Commonwealth Games open in style', 'C
ommonwealth Games day-by-day guide & schedule', "'My brothers hung me from d
oors, but it made me'", 'BBC TV, radio & digital coverage times for Commonwe
alth Games', "'I want to leave legacy' - Fachie's quest for glory", 'England
aim for gold as cricket returns to Games', "Flagbearer Gilmour's letter to 1
6-year-old self", "Watch the key moments from Boris Johnson's final PMQs", "
If Boris Johnson was expecting a teary-eyed send off from all MPs, he didn't
get one at his final Prime Minister's Questions.", "The outgoing prime minis
ter's last weekly grilling at the dispatch box was not unlike his first in S
eptember 2019 - acrimonious, raucous, and littered with his colourful quips.
", 'After almost an hour of political theatre, Mr Johnson brought the curtai
n down in characteristic fashion, with a tongue-in-cheek farewell, borrowed
from the script of a sci-fi blockbuster.', '"Hasta la vista, baby," Mr Johns
on told MPs, reciting the line from Arnold Schwarzenegger\'s cyborg characte
r in the 1991 film Terminator 2: Judgment Day.', "Cue uproarious laughter, c
heers and a standing ovation from the Conservative benches - although former
PM Theresa May didn't appear to join in the applause.", 'Given Schwarzenegge
r\'s other famous catchphrase in the film - "I\'ll be back" - Mr Johnson\'s
sign-off and what it means may hang in the air in Westminster.', 'For now, a
t least, Mr Johnson will leave office by September, when the contest to elec
t his replacement as Tory leader and prime minister is due to conclude.', 'I
n a pithy closing statement Mr Johnson had a few words of advice for his suc
cessor, saying that they should stay close to the Americans, stick up for th
e Ukrainians and cut taxes whenever possible.', '"Focus on the road ahead, b
ut always remember to check rear-view mirror. And remember above all, it\'s
not Twitter that counts - it\'s the people who sent us here."', "Held in the
pressure cooker of the House of Commons, Prime Minister's Questions is desig
ned to be an adversarial occasion of high political drama.", 'The first was
in 1961, and since then, prime ministers have admitted the event holds some
terror.', 'Years after leaving office, former Prime Minister Tony Blair like
ned PMQs to being marched to his own "execution", while one of his predecess
ors, Harold Macmillan, confided it made him feel "physically sick".', "It's
easy to see why as Mr Johnson faced a hostile audience of MPs, some of whom
had been involved in ousting him as Tory leader weeks ago.", 'Before proceed
ings began, Commons Speaker Lindsay Hoyle urged MPs to moderate their langua
ge and conduct themselves in a respectful manner.', "The plea appeared to fa
ll on deaf ears though as Labour MPs - led by leader Sir Keir Starmer - laun
ched a barrage of attacks on Mr Johnson's integrity and record in as PM. ",
'"Inflation is up again this morning and millions are struggling with a cost
of living crisis, and he\'s decided to come down from his gold wallpapered b
unker for one last time to tell us that everything\'s fine," Sir Keir said.'
, 'Mr Johnson paid little heed to Sir Lindsay\'s plea either, branding Sir K
eir a "great pointless human bollard".', 'It was the kind of unconventional
one-liner Mr Johnson has reeled off time and again during his 93 PMQs duels,
most of them against Sir Keir.', '"Captain Hindsight" was one of Mr Johnson\
's favourite nicknames for Sir Keir, often used in the context of his calls
for lockdown restrictions during the pandemic.', 'Many of their fiercest cla
shes at PMQs were over Covid-19 rules, and the breach of them by Mr Johnson
and others in Downing Street.', 'A more recent addition to Mr Johnson\'s jib

es at Sir Keir has been "Captain Crasheroonie Snoozefest", and he once descr
ibed former Labour leader Jeremy Corbyn as a "chlorinated chicken" in an exc
hange about post-Brexit trade with the US.', 'When asked why the Tory leader
ship candidates had pulled out of a televised debate this week, Mr Johnson o
nce again conjured an vivid image with his rhetoric.', 'The candidates would
"wipe the floor" with Sir Keir, Mr Johnson said, comparing his Tory colleagu
es to "household detergent".', 'Perhaps not quite what eliminated Tory leade
rship candidate Tom Tugendhat had in mind when he offered a "clean start".',
'BBC political correspondent Ione Wells sat in the press gallery of the Comm
ons watching the spectacle unfold.', 'The atmosphere was "pretty jovial over
all", she said. She said even Mr Johnson\'s critics couldn\'t help but chuck
le at his gags, with Labour and Liberal Democrat MPs laughing at his notorio
us rhetorical flourishes, which have arguably got him in trouble over the ye
ars.', "Mr Johnson's valedictory speech came after veteran Conservative back
bencher Sir Edward Leigh praised his record.", 'Labour MPs could be heard sh
outing "no" as Sir Edward said: "On behalf of the House may I thank the prim
e minister for his three years\' record of service."', "Stepping up to the d
ispatch box for the last time, Mr Johnson thanked his staff and MPs before g
iving a nod to Schwarzenegger's Terminator character.", 'As the heat of PMQs
cooled, Mr Johnson left the chamber, receiving pats on the back and handshak
es as he went.', 'Some tears were apparently shed after all by Conservative
minister Andrea Jenkyns.', "These were the end credits of Mr Johnson's swans
ong PMQs, or, to evoke his Terminator reference, the final parliamentary Jud
gement Day of his premiership.", 'In his first PMQs on 4 September 2019, Mr
Johnson called Mr Corbyn a "great big girl\'s blouse" when the then-Labour l
eader challenged the PM about parliamentary scrutiny.', '"Call an election,
you great big girl\'s blouse"', 'On 12 January 2022, Mr Johnson offered "hea
rtfelt apologies" for attending drinks in Downing Street\'s garden on 20 May
2020, when lockdown restrictions were in force.', 'The prime minister was ac
cused of "body shaming" the SNP\'s Westminster leader Ian Blackford during a
PMQs clash on 26 January 2022.', 'Ian Blackford: "The impending National Ins
urance tax hike hangs like a guillotine while they eat cake."', 'On 25 May 2
022, Mr Johnson repeated his apology for parties held in Downing Street duri
ng lockdown, after senior civil servant Sue Gray published a report into bre
aches of Covid-19 rules.', 'Prime Minister Boris Johnson: I am humbled and I
have learned a lesson', 'But his team later clarifies Mr Sunak was only back
ing the expansion of existing grammar schools.', 'Liz Truss continues to emp
hasise her loyalty to the outgoing prime minister, while Rishi Sunak stands
accused of stabbing him in the back', 'The Labour leader insists he did not
sack his transport spokesman for joining a union picket line.', 'The former
ambassador to the US became famous for his colourful turn of phrase - and co
lourful socks.', 'The culture secretary also accuses the ex-chancellor of le
ading a "ruthless coup" against Boris Johnson.', 'Rampant inflation is leadi
ng to demands for higher pay offers and promises of coordinated industrial a
ction.', 'But his team later clarifies Mr Sunak was only backing the expansi
on of existing grammar schools.', 'Liz Truss continues to emphasise her loya
lty to the outgoing prime minister, while Rishi Sunak stands accused of stab
bing him in the back', 'The Labour leader insists he did not sack his transp
ort spokesman for joining a union picket line.', 'Sunak says yes to return o
f grammar schools', 'Chris Mason: Truss courts Johnson loyalists as Sunak fa
ces jibes', 'MP fired for making up policy on the hoof - Starmer', 'Ex-ambas
sador Sir Christopher Meyer dies at 78', "Dorries defends Claire's Accessori
es Sunak attack", 'Sunak says yes to return of grammar schools', 'Chris Maso
n: Truss courts Johnson loyalists as Sunak faces jibes', 'MP fired for makin
g up policy on the hoof - Starmer', 'Ex-ambassador Sir Christopher Meyer die
s at 78', "Dorries defends Claire's Accessories Sunak attack", 'Bryant makes
charity donation after claims disproved', "Have wages been 'stuck for a deca
de'? And other claims", 'Cuts leave military at risk as threats rise - MPs',

'Government rules out bank holiday for Euro win', 'Truss vows to criminalise street harassment', 'How UK Greens are learning from overseas allies', 'Five takeaways from a heated Truss-Sunak clash', 'Who are the Tories that will ch oose the next PM?', 'What taxes would the two Tory leadership rivals cut?', "Lessons will be learned about the UK's handling of Covid, before another pa ndemic strikes, Baroness Hallett, chairing the public inquiry, has said.", ' Opening the inquiry, she promised to be "fair and robust".', 'The former Hig h Court judge said she would conduct the inquiry as quickly as possible, wit hout giving a timeframe for its completion.', 'Those who had suffered the mo st deserved to know if more could have been done, she said.', 'Lives had bee n lost, education harmed, businesses folded and mental and physical health h ad suffered.', '"Every person has had their life changed to some extent," La dy Hallett said.', '"Those who have suffered the most will want to know if a ny more could have been done to reduce their suffering."', 'The inquiry can compel witnesses to give evidence and release documents, but cannot prosecut e or fine anyone.', 'It was a substantial task that would take time and have a significant cost, Lady Hallett said.', 'But she added: "I am determined to undertake the inquiry as speedily as possible so lessons can be learned befo re another pandemic strikes."', 'Public hearings will begin in the spring.', 'Before then, Lady Hallettt said, the key topics for the inquiry would inclu de:', 'The inquiry will begin taking evidence from experts in September.', ' There will be many involved in the care sector who will have a great deal to say. They will remember the early warnings from other countries about the vu lnerability of care homes. ', 'Care providers will recount their struggle to get protective equipment; what they saw as the slowness of government guidan ce; the rapid discharge of hospital patients into care homes, some taking wi th them the virus and above all, their sense of being forgotten. ', 'Familie s who were unable to see or in too many cases, say goodbye to their loved on es, are likely to have their say later in the inquiry. ', 'But in weighing u p our preparedness for the pandemic, there is one number that perhaps tells the most powerful story. In the first wave between March and June 2020, near ly 20,000 care home residents died of Covid in England and Wales. At that ti me, it represented more than a third of the total number of deaths.', 'Jo Go odman, co-founder of Covid-19 Bereaved Families for Justice campaign, said: "Today was an emotional day for those of us who have lost loved ones and it meant a lot to hear Baroness Hallet recognise the devastating nature of bere avement and the pain we\'ve been through. ', '"Ultimately, all bereaved fami lies want the same thing, which is to make sure that lessons are learnt from our devastating losses to protect others in the future."', 'Charles Persinge r, who is part of the campaign and lost his mother and wife to Covid, added: "We\'ve waited a long time to get to this point - we would have liked the in quiry to start sooner. But what is important now is to really get to the bot tom of the mistakes that were made."', "Several reports have already put the UK government's handling of the pandemic under the spotlight.", 'The Covid p andemic had a devastating impact on ethnic minority communities in the UK. ' , 'In the first wave of the pandemic, black people were almost four times as likely to die of Covid than white people, while Asians were twice as likely to die. About 95% of doctors who died of Covid were ethnic minorities. ', ' Saleyha Ahsan, a doctor and documentary maker, lost her dad to Covid-19. She tells me it\'s good inequalities are being considered, but she\'s "keen to s ee how [Baroness Hallett] does it in real terms".', '"I\'m not convinced it\ 's going to cover it all, so I am wary." ', 'Dr Ahsan is involved with the C ovid-19 Bereaved Families for Justice, which led the campaign for an inquiry to happen.', '"The main thing is is that we have now crossed the start line, " she says. "We\'ve been calling for this for a whileâ\x80¦ and today is a d efinite firing of the starting gun." ', 'From intricate mosaics to amazing h ieroglyphics - what else hides under the water?', 'Matthew Syed is calling f or a nuclear awakening...']